HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thị Thu Thủy

PHÁT HIỆN ĐỐI TƯỢNG TỪ VỰNG MỞ CÓ KÍCH THƯỚC NHỎ TRONG ẢNH CHỤP TỪ DRONE SỬ DỤNG OWL-VIT KẾT HỢP SAHI

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI – 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thị Thu Thủy

PHÁT HIỆN ĐỐI TƯỢNG TỪ VỰNG MỞ CÓ KÍCH THƯỚC NHỎ TRONG ẢNH CHỤP TỪ DRONE SỬ DỤNG OWL-VIT KẾT HỢP SAHI

Chuyên ngành: Khoa học máy tính
Mã số: 8.48.01.01

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC :
PGS.TS PHẠM VĂN CƯỜNG

HÀ NỘI – 2024

LỜI CAM ĐOAN

Tôi cam đoan đề án " Phát hiện đối tượng từ vựng mở có kích thước nhỏ trong ảnh chụp từ drone sử dụng OWL-ViT kết hợp SAHI" là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin chịu hoàn toàn trách nhiệm về lời cam đoan nêu trên.

Học viên

Nguyễn Thị Thu Thủy

LỜI CẢM ƠN

Lời đầu tiên em xin cảm ơn đến các thầy cô giảng viên của Học viện Công nghệ Bưu chính Viễn thông đã truyền đạt cho em bao kiến thức vô cùng quý báu và cần thiết trong thời gian học tập ở trường. Những tri thức ấy chính là nền tảng vững chắc cho sự phát triển của em sau này. Xin kính chúc thầy cô có nhiều sức khỏe và thành công hơn nữa trong sự nghiệp trồng người.

Em cũng xin gửi lời cảm ơn PGS.TS Phạm Văn Cường, giảng viên đã tận tình hướng dẫn em trong suốt quá trình nghiên cứu để hoàn thành đề án này. Nhờ sự hướng dẫn chỉ bảo tận tình của thầy, em đã có thêm nhiều kiến thức về trí tuệ nhân tạo và thị giác máy tính. Vốn kiến thức quý giá vô cùng quan trọng cho định hướng của em ở tương lai.

Qua những năm tháng sinh viên dưới mái trường đại học, em đã gặp vô vàn khó khăn nhưng thật may mắn khi gia đình và thầy cô, bạn bè luôn ở bên và động viên giúp đỡ. Em xin gửi lời cảm ơn tới tất cả mọi người.

Dù rất cố gắng nhưng do kiến thức của em đôi chỗ còn chưa vững nên Đề án của em không thể không tránh khỏi những thiếu sót. Mong thầy cô xem xét và đóng góp ý kiến giúp em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Hà Nội, ngày 19 tháng 02 năm 2024

Học viên

Nguyễn Thị Thu Thủy

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Viết tắt	Tiếng anh	Tiếng việt
OWL-ViT	Vision Transformer for Open-World Localization	Học máy biến đổi thị giác phát cho việc định vị trong thế giới mở.
SAHI	Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection	Cắt nhỏ hình ảnh hỗ trợ suy luận và tinh chỉnh tốt hơn cho việc phát hiện đối tượng nhỏ
OVD	Open Vocabulary Detection	Phát hiện đối tượng từ vựng mở
Drone	Drone	Máy bay không người lái
AI	Artificial Intelligence	Trí tuệ nhân tạo
TOOD	Task-aligned One-stage Object Detection	Phát hiện đối tượng một giai đoạn được điều chỉnh cho nhiệm vụ
FCOS	Fully Convolutional One-Stage Object Detection	Phát hiện đối tượng một giai đoạn hoàn toàn dựa trên tích chập
VFNet	Vari Focal Net	Mạng Vari Focal
YOLO	You only look once	Mạng nhìn một lần
Transformer	Transformer	Mô hình học máy biến đổi
ViT	Vision Transformer	Mô hình học máy biến đổi thị giác
SSD	Single Shot Detector	Phát hiện đối tượng một lần chạy
Faster-RCNN	Faster Region Convolution Neural Network	Mạng nơ-ron tích chập khu vực nhanh hơn
RPN	Region Proposal Network	Mạng đề xuất vùng quan tâm
CNN	Convolution Neural Network	Mạng nơ ron tích chập
DETR	Detection Transformer	Mô hình học máy biến đổi phát hiện đối tượng
Encoder	Encoder	Bộ mã hóa
Decoder	Decoder	Bộ giải mã
NMS	Non-maximum Suppression	Phương pháp chống trùng lặp không tối đa
AP	Average Precision	Độ chính xác trung bình một lớp đối tượng
mAP	Mean Average Precision	Độ chính xác trung bình các lớp trong tập dữ liệu
IoU	Intersection over Union	Tỷ lệ giao trên hợp của hai vùng
GIoU	Generalized IoU	Tỷ lệ giao trên hợp tổng quát của hai vùng
SGD	Stochastic Gradient Descent	Giảm gradient ngẫu nhiên
Adagrad	Adaptive Gradient Algorithm	Thuật toán gradient tuỳ biến

RMSprop	Root Mean Square Propagation	Thuật toán lan truyền trung bình căn bậc hai
Adam	Adaptive Moment Estimation	Thuật toán ước lượng mô-men tuỳ biến
AdamW	Adam with Decoupled Weight Decay	Thuật toán Adam với điều chỉnh trọng số phân tách
GPU	Graphics Processing Unit	Bộ xử lý đồ họa

DANH MỤC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại công nghệ thông tin phát triển như hiện nay, đặc biệt là lĩnh vực AI với các hệ thống nhận diện thông qua camera ngày càng nhiều. Trong đó phát hiện đối tượng là một tính năng được được nhiều hệ thống sử dụng, nhất là trong các hệ thống phân tích khách hàng của cửa hàng, hệ thống giám sát an ninh. Hơn thế nữa, tính năng này còn được áp dụng vào để phân tích ảnh từ các thiết bị bay không người lái được lắp camera như drone. Ảnh chụp từ drone chụp được rất nhiều vật thể, góc camera chụp rộng. Tuy nhiên một trong những vấn đề gặp phải của nó khi sử dụng những phương pháp phát hiện đối tượng truyền thống hiện nay đó là những đối tượng loại nhỏ thường hay bị bỏ sót.

Những năm gần đây, phát hiện đối tượng (object detection) theo hướng từ vựng mở (open-vocabulary - OV) [1] đã thu hút sự quan tâm nghiên cứu ngày càng nhiều. Khác với phát hiện đối tượng truyền thống chỉ nhận dạng các đối tượng thuộc các danh mục cố định, phát hiện đối tượng từ vựng mở nhằm mục tiêu phát hiện các đối tượng trong một tập hợp danh mục mở. Các mô hình huấn luyện cả hình ảnh-ngôn ngữ (vision-language) để phát hiện các đối tượng từ vựng mở. Trong đó, Vision Transformer for Open-World Localization (OWL-ViT) [2] là một trong những mô hình phát hiện tốt nhất hiện nay. Tuy nhiên thực tiễn cho thấy mô hình này phát hiện những đối tượng có kích thước nhỏ không tốt.

Để giải quyết các vấn đề này, đề án nghiên cứu cải tiến mô hình Vision Transformer for Open-World Localization (OWL-ViT) kết hợp với kỹ thuật Slicing Aided Hyper Inference (SAHI) [3] để phát hiện đối tượng từ vựng mở cho các đối tượng kích thước nhỏ trong ảnh chụp từ drone.

2. Tổng quan về vấn đề nghiên cứu

Bài toán phát hiện đối tượng (object detection) là một bài toán trong lĩnh vực thị giác máy tính, nhằm tìm ra và xác định vị trí cũng như loại đối tượng xuất hiện trong một hình ảnh hoặc video. Cụ thể, bài toán phát hiện đối tượng yêu cầu mô hình học máy có khả năng nhận diện và phân loại các đối tượng khác nhau trong hình ảnh và đồng thời đưa ra thông tin về vị trí của chúng. Điều này thường được thực hiện bằng cách tạo ra các hộp bao (bounding boxes) xung quanh các đối tượng và gán nhãn cho chúng. Bài toán này từ lâu đã được nghiên cứu rất nhiều tuy nhiên các phương pháp trước đem lại hiệu quả chưa cao như phát hiện thiếu, nhầm đối tượng, nhất là với các đối tượng kích thước nhỏ. Đặc biệt các loại đối tượng chỉ có thể được phát hiện nếu nó nằm trong tập huấn luyện, muốn phát hiện thêm lớp mới ta phải thu thập dữ liệu cho chúng và tiến hành đào tạo lại mô hình từ đầu, và có rất nhiều loại đối tượng rất khó để thu thập dữ liệu.

Với sự phát triển của công nghệ kèm theo sự phát triển của dữ liệu, những ý tưởng mới kết hợp giữa văn bản, để phát hiện đối tượng trong ảnh tốt hơn. Và một hướng đi mới cho bài toán này chính là phát hiện đối tượng từ vựng mở (open-vocabulary detection- OVD). Trong phát hiện đối tượng truyền thống, mô hình chỉ phát hiện được những đối tượng cụ thể đã được đào tạo (tập đối tượng cố định). Ngược lại, nhờ sự kết hợp đào tạo giữa cả dữ liệu văn và hình ảnh, phát hiện đối tượng từ vựng mở, mô hình có thể phát hiện được cả những đối tượng chưa được huấn luyện. Nhận đầu vào là một cặp hình ảnh – văn bản (image-text), văn bản gồm những danh từ cần phát hiện trong bức ảnh, sau đó mô hình phát hiện đối tượng từ vựng mở sẽ cho ra kết quả phát hiện gồm các hộp bao vật thể (bounding boxes) và tên ứng với các danh từ của đối tượng. Như vậy, phát hiện đối tượng từ vựng mở đã khắc phục được vấn đề tập đối tượng phát hiện bị hạn chế, sự kết giữa văn bản và hình ảnh sẽ giúp tăng độ chính xác cho phát hiện đối tượng trong ảnh. Một trong những mô hình cho kết quả tốt nhất với phát hiện đối tượng từ vựng mở hiện nay đó là Vision Transformer for Open-World Localization (OWL- ViT) . Mô hình OWL-ViT được thiết kế dựa trên kiến trúc cơ bản của Vision Transformer [4] và Text Transformer [5], huấn luyện trước nó với một tập dữ liệu lớn gồm các cặp hình ảnh-văn bản. Để phát hiện đối tượng từ vựng mở, loại bỏ token pooling và thêm vào hai đầu phân loại và xác định vị trí đối tượng (object classification head và object localization head) làm đầu ra của Vision Transformer encoder. Đầu xác định vị trí đối tượng là một mạng nơ ron truyền thẳng nhiều lớp-Multi-Layer Perceptron (MLP) [6], cho ra tọa độ các đối tượng là các hộp bao (bounding boxes), số hộp bao bằng số lượng từ của đầu vào văn bản (mỗi từ ứng với một danh từ là một lớp đối tượng cần được phát hiện). Phần văn bản đầu vào được xử lý qua Text Transformer encoder sẽ được sử dụng kết hợp với đầu phần loại để phân loại đối tượng, gán nhãn cho đối tượng đã được xác định với bounding box. OWL-ViT cho kết quả phát hiện đối tượng rất ấn tượng như đạt 34.6% AP tổng thể và 31.2% AP lớp hiếm cho các class không được huấn luyện trước trên tập LVIS. Đây là một mô hình có kiến trúc đơn giản, rất dễ để mở rộng phát triển.

Tuy nhiên thực nghiệm cho thấy OWL-ViT phát hiện các đối tượng kích thước nhỏ không tốt, cụ thể là rất nhiều đối tượng nhỏ trong bước ảnh không được phát hiện. Và đây là một vấn đề của mô hình cần được cải thiện. Có rất nhiều giải pháp giúp mô hình phát hiện đối tượng kích thước nhỏ trong ảnh được tốt hơn. Đề án này sử dụng một trong những kỹ thuật mới và tốt nhất hiện nay đó là Slicing Aided Hyper Inference (SAHI) để giải quyết vấn đề này cho OWL-ViT. Kỹ thuật SAHI có thể áp dụng với bất kì mô hình phát hiện đối tượng nào. Trong quá trình tinh chỉnh, phương pháp này chia bức ảnh thành các nhiều phần chồng lấn nhau (overlapping patches). Các patches này được thay đổi kích thước, tuy nhiên vẫn giữ nguyên tỷ lệ khung hình,

tạo ra các bức ảnh tăng cường, nhằm mục đích tăng kích thước của đối tượng so với trong hình ảnh gốc. Đối với quá trình suy luận, ảnh đầu vào cũng được SAHI chia thành các overlapping patches. Sau đó, thực hiện phát hiện đối tượng cho từng patches, và hợp nhất chúng lại thu được kết quả cho ảnh đầu vào. Bằng cách này, SAHI đã giúp tăng AP cho phát hiện đối tượng với các mô hình phát hiện đối tượng truyền thống 6.8%, 5.1%, 5.3% lần lượt cho FCOS, VFNet, TOOD [7][8][9]. SAHI cũng được triển khai kết hợp rộng rãi với các mô hình phổ biến YOLO [10].

Từ trên, có thể thấy rằng OWL-ViT đã sử dụng kết hợp giữa thông tin giữa văn bản và hình ảnh để phát hiện đối tượng từ vựng mở. Điều này giúp cho việc phát hiện đối tượng chính xác hơn, phát hiện được các lớp đối tượng không qua huấn luyện. Đây cũng là môt mô hình với kiến trúc cơ bản, rất dễ để mở rộng và phát triển thêm. Trong khi đó, kỹ thuật SAHI với phương pháp chia cắt hình ảnh giúp cho việc phát hiện đối tượng kích thước nhỏ hiệu quả hơn, đã khắc phục một vấn đề thường xuyên gặp phải trong các mô hình phát hiện đối tượng. Bằng việc tận dụng ưu điểm của mô hình OWL-ViT và kỹ thuật SAHI, đề án sẽ trình bày phương pháp kết hợp OWL-ViT để phát hiện đối tượng từ vựng mở có kích thước nhỏ trong ảnh chụp từ drone.

3. Mục đích nghiên cứu

Đề án này nghiên cứu phát hiện đối tượng từ vựng mở có kích thước nhỏ trong ảnh chụp từ drone. Giải pháp sử dụng mô hình OWL-ViT để phát hiện đối tượng từ vựng mở, kết hợp với kỹ thuật SAHI giúp mô hình phát hiện đối tượng tốt hơn. Với việc phát hiện đối tượng nhỏ trong ảnh tốt hơn sẽ giúp cho các hệ thống AI giám sát, phân tích thông tin qua camera nhận phát hiện nhiều đối tượng hơn, giúp cho việc phân tích hình ảnh được chi tiết và chính xác hơn.

Khía cạnh lý thuyết:

- Nghiên cứu: Hiểu sâu hơn về hướng đi mới trong phát hiện đối tượng là phát hiện đối tượng từ vựng mở. Nghiên cứu mô hình OWL-ViT và kỹ thuật SAHI, khả năng kết hợp áp dụng vào bài toán phát hiện đối tượng kích thước nhỏ trong ảnh chụp từ drone.
- Phân tích so sánh: Để thực hiện nghiên cứu đánh giá phương pháp kết hợp giữa OWL-ViT + SAHI sẽ giúp phát hiện đối tượng kích thước nhỏ tốt hơn so với chỉ sử dụng OWL-ViT hoặc SAHI kết hợp với phương pháp khác bằng cách lập bảng so sánh kết quả trên tập dữ liệu VisDrone2019-Detection [11].
- Hiểu vấn đề: Để hiểu được những vấn đề, thách thức trong bài toán phát hiện đối tượng kích thước nhỏ trong ảnh chụp từ drone. Từ đó có những ý tưởng để nghiên cứu giải pháp.

Khía cạnh thực tiễn:

- Cài đặt mô hình: Lập trình mô hình kết hợp giữa OWL-ViT + SAHI để nhận diện những đối tượng kích thước nhỏ trong ảnh chụp từ drone.
- Ứng dụng trong tạo bộ dữ liệu: Phương pháp OWL-ViT + SAHI sẽ giúp thực hiện đánh nhãn dữ liệu tự động hiệu quả hơn khi có thể phát hiện các đối tượng nhỏ tốt hơn. Từ đó, có thể ứng dụng xây dựng công cụ đánh nhãn dữ liệu tự động với phương pháp này.
- Ứng dụng trong sản phẩm: Phát hiện đối tượng trong ảnh chụp từ drone có tính ứng dụng cao trong các hệ thống giám sát, phân tích ở không gian rộng phát hiện vật thể từ trên cao.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Đối tượng: Đối tượng chính của nghiên cứu này là dữ liệu ảnh, cụ thể là các đối tượng có kích thước nhỏ được chụp từ drone.
- Công nghệ: Các công nghệ được nghiên cứu là mô hình phát hiện đối OWL-ViT và kỹ thuật SAHI.

Phạm vi nghiên cứu:

- Phạm vi: Ảnh chụp từ drone. Cụ thể là bộ dữ liệu VisDrone2019- Detection.
- Khung thời gian: Dự án dự kiến sẽ được hoàn thành trong khoảng thời gian bốn tháng. Nghiên cứu sẽ bao gồm các bài báo và bộ dữ liệu tính đến cuối năm 2023.

5. Phương pháp nghiên cứu

Đề án được nghiên cứu dựa trên cả lý thuyết và thực nghiệm. Xây dựng mô hình dựa trên các phương pháp được nghiên cứu từ các bài báo khoa học về phát hiện đối tượng trong ảnh.

Thu thập dữ liệu:

- Bộ dữ liệu drone: Bộ dữ liệu được chụp từ drone VisDrone2019-Detection, gồm các hình ảnh chụp từ trên cao, góc rộng với rất nhiều đối tượng kích thước nhỏ trong ảnh. Bao gồm 10,209 ảnh với 2,6 triệu đối tượng được chia làm 3 tập đào tạo-xác thực-kiểm thử. Các loại đối tượng trong ảnh: người đi bộ, đám đông người, xe đạp, ô tô, xe bán tải, xe tải hạng nặng, xe ba bánh, xe ba gác, xe bus.
- Tiền xử lý dữ liệu: Lọc bỏ ảnh mờ, nhiễu. Chỉnh sửa lại nhãn đánh sai. Đưa về dạng nhãn dữ liệu phù hợp cho mô hình.

Thiết lập thử ngiệm:

- Môi trường: Thực nghiệm sẽ được tiến hành trên một máy tính được kiểm soát để đảm bảo khả năng thử nghiệm nhiều lần.
- Công cụ và thư viện: Ngôn ngữ lập trình Python sẽ được sử dụng cùng với các thư viện hỗ trợ ngôn ngữ này.

Phương pháp:

- Nghiên cứu tài liệu: nghiên cứu các bài báo khoa học về OWL-ViT và SAHI để hiểu sâu hơn về mô hình và kỹ thuật trên.
- Xây dựng phương pháp: kết hợp giữa mô hình phát hiện đối tượng từ vựng mở OWL-ViT và kỹ thuật hỗ trợ phát thiện đối tượng kích thước nhỏ SAHI.
- Đào tạo mô hình: Xử lý dữ liệu, tiến hành tinh chỉnh (fine-tuning) với các bộ dữ liệu VisDrone2019-Detection.
- Số liệu đánh giá: Mô hình sẽ được đánh giá dựa trên điểm AP thu được trên các bộ dữ liệu.
- Đánh giá: Đánh giá kết quả mô hình dự đoán trên các tập dữ liệu. So sánh OWL-ViT với trước và sau khi sử dụng thêm SAHI. Ngoài ra cũng so sánh thêm với các kết quả trên các tập dữ liệu đã có của những mô hình phát hiện đối tượng khác như TOOD, FCOS, VFNet.
- Xây dựng demo cho phương pháp OWL-ViT+SAHI để thấy rõ kết quả phát hiện đối tượng từ vựng mở có kích thước nhỏ trong ảnh.

Từ mục tiêu, nhiệm vụ nghiên cứu, đề án sẽ được cấu trúc với ba chương nội dung chính như sau:

Chương 1:Tổng quan vấn đề nghiên cứu

Chương 2: Phương pháp OWL-ViT kết hợp SAHI

Chương 3: Thực nghiệm và đánh giá

Chương I - TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

1.1 Bài toán phát hiện đối tượng

1.1.1 Tổng quan phát hiện đối tượng

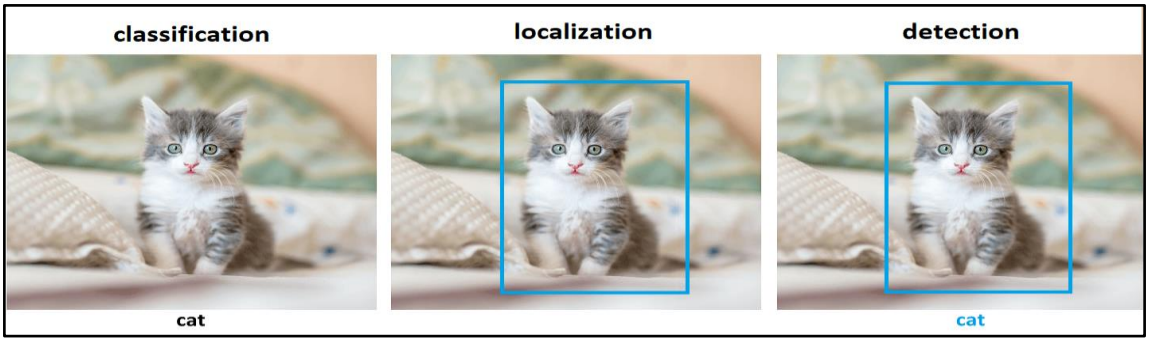
Một trong những lĩnh vực quan trọng của Trí tuệ nhân tạo (Artificial Intelligence) là thị giác máy tính (Computer Vision). Computer Vision là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh, phát hiện các đối tượng, tạo ảnh, siêu phân giải hình ảnh và nhiều hơn vậy. Phát hiện đối tượng-Object detection có lẽ là khía cạnh sâu sắc nhất của thị giác máy vì được áp dụng nhiều trong thực tế.

Sẽ khá khó cho người mới bắt đầu để phân biệt các nhiệm vụ khác nhau của computer vision. Chẳng hạn như phân loại hình ảnh (image classification) là gì? Định vị đối tượng (object localization) là gì? Sự khác biệt giữa định vị vật thể (object localization) và phát hiện đối tượng (object detection) là gì? Đây là những khái niệm có thể gây nhầm lẫn, đặc biệt là khi cả ba nhiệm vụ đều liên quan đến nhau. Hiểu một cách đơn giản:

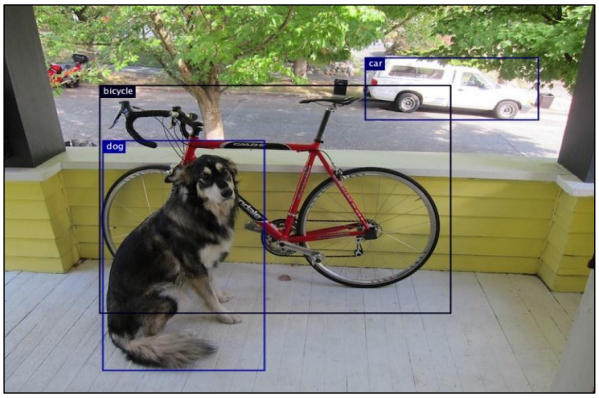
- Phân loại hình ảnh (image classification): liên quan đến việc gán nhãn cho hình ảnh. Mô hình trí tuệ nhân tạo sẽ dự đoán nhãn của một đối tượng trong một hình ảnh. Đầu vào: một hình ảnh với một đối tượng. Đầu ra: nhãn lớp của đối tượng trong ảnh. Ví dụ như mô hình nhận ảnh đầu vào chỉ chứa một con vật và cho ra kết quả tên con vật đó.

- Định vị đối tượng (object localization): liên quan đến việc vẽ một hộp giới hạn (bounding box) xung quanh một hoặc nhiều đối tượng trong hình ảnh nhằm khoanh vùng đối tượng. Nghĩa là mô hình sẽ xác định vị trí hiện diện của các đối tượng trong ảnh và cho biết vị trí của chúng bằng bounding box. Đầu vào: Một hình ảnh có một hoặc nhiều đối tượng. Đầu ra: Một hoặc nhiều bounding box được xác định bởi tọa độ tâm, chiều rộng và chiều cao. Chẳng hạn đầu vào là một bức ảnh chứa cả hai con vật chó và mèo, đầu ra sẽ là hộp bao hình chữ nhật bao quanh mỗi con vật.

- Phát hiện đối tượng (object detection): Là nhiệm vụ khó khăn hơn và là sự kết hợp của cả hai nhiệm vụ trên: Vẽ một bounding box xung quanh từng đối tượng quan tâm trong ảnh và gán cho chúng một nhãn. Kết hợp cùng nhau, tất cả các vấn đề này được gọi là object recognition hoặc object detection. Đầu vào: một hình ảnh có một hoặc nhiều đối tượng. Đầu ra: một hoặc nhiều bounding box tương ứng với nhãn đối tượng. Ví dụ như mô hình nhận đầu vào là một bức ảnh chứa cả hai con vật chó và mèo, kết quả đầu ra sẽ là hai hộp bao hình chữ nhật bao quanh mỗi con vật và tên con vật tương ứng cạnh mỗi hộp.



Hình 1.1: Ba nhiệm vụ phân loại hình ảnh, định vị đối tượng, phát hiện đối tượng



Hình 1.2: Phát hiện nhiều đối tượng trong một ảnh

Bài toán phát hiện đối tượng đề cập đến khả năng của hệ thống máy tính và phần mềm để định vị các đối tượng trong một hình ảnh và xác định từng đối tượng. Object Detection đã được sử dụng rộng rãi để phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật và xe không người lái,…

1.1.2 Phát hiện đối tượng trong ảnh chụp từ Drone

Trong thời đại công nghệ thông tin phát triển như hiện nay, đặc biệt là lĩnh vực AI với các hệ thống nhận diện thông qua camera ngày càng nhiều. Trong đó phát hiện đối tượng là một tính năng được được nhiều hệ thống sử dụng, nhất là trong các hệ thống phân tích khách hàng của cửa hàng, hệ thống giám sát an ninh. Hơn thế nữa, tính năng này còn được áp dụng vào để phân tích ảnh từ các thiết bị bay không người lái được lắp camera như drone. Ảnh chụp từ drone chụp được rất nhiều vật thể, góc camera chụp rộng. Phát hiện đối tượng từ ảnh chụp của drone có nhiều ứng dụng quan trọng, bao gồm:

- Giám sát và An ninh: Drone có khả năng bay trên cao và thu thập dữ liệu ảnh từ các góc độ khác nhau. Các mô hình phát hiện đối tượng có thể được sử dụng để phát hiện các đối tượng xâm nhập lạ tại các khu vực quan trọng như biên giới, các cơ sở quân sự. Điều này giúp nâng cao an ninh và đảm bảo sự an toàn công cộng.

- Quản lý môi trường và tài nguyên: Phát hiện đối tượng từ ảnh chụp của drone có thể hỗ trợ trong việc giám sát môi trường tự nhiên như phát hiện vùng đất rừng bị tàn phá, theo dõi sự biến đổi của các khu vực đất và đánh giá tình trạng đại dương. Nó cũng có thể được sử dụng để theo dõi tài nguyên nông nghiệp, như đánh giá mật độ cây trồng, giám sát sự phát triển và phát hiện bất thường trong vườn trồng.
- Quản lý thiên tai và khắc phục hậu quả: Drone có thể được sử dụng để chụp ảnh từ không gian sau các thiên tai như động đất, lũ lội và cơn bão. Mô hình AI có thể giúp phát hiện các vùng bị tổn thương, đánh giá mức độ thiệt hại và hỗ trợ quyết định khắc phục hậu quả.
- Quản lý công trình và xây dựng: Bằng ảnh chụp từ Drone, các mô hình phát hiện đối tượng sẽ phát hiện các công trình xây dựng trái phép trái với quy hoạch. Từ đó hỗ trợ việc giám sát các công trình xây dựng và hạ tầng.

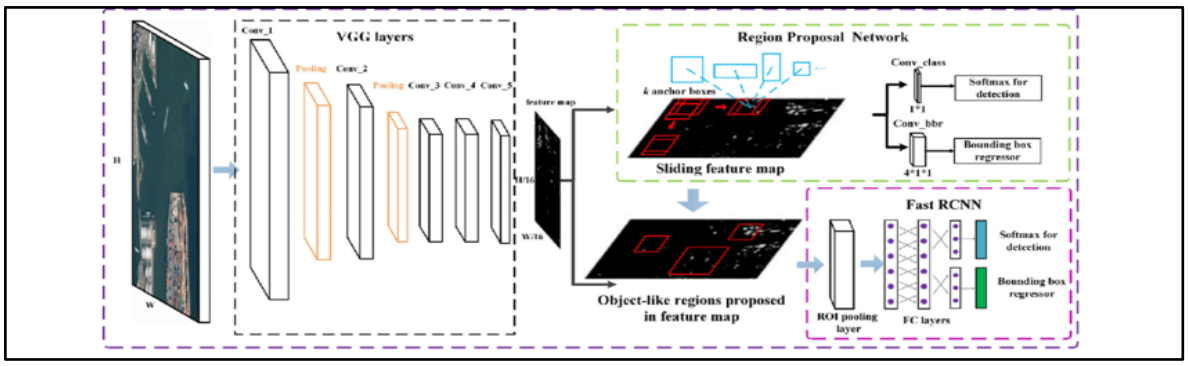
Hình 1.3: Ảnh chụp thành phố từ Drone

1.2 Các nghiên cứu liên quan

1.2.1 Một số phương pháp phát hiện đối tượng từ vựng đóng

Phát hiện đối tượng từ vựng đóng là các mô hình đã được cố định hóa tên các đối tượng là các từ vựng nhất định và từ ngữ, tên loại đối tượng không được sử dụng trong quá trình học tập của mô hình.

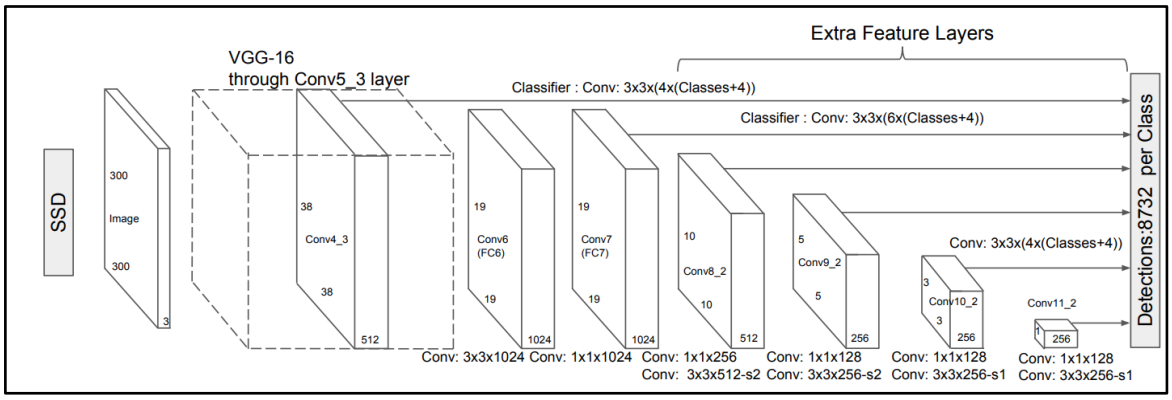
Ban đầu, các mô hình "one-stage" và "two stage" như SSD [12] và Faster-RCNN [13] rất được ưa chuộng. Đây đều là các mô hình thuần CNN [14], kiến trúc mô hình phần lớn xây dựng đều dựa trên các lớp CNN. Kiến trúc mạng Faster-RCNN có 2 phần chính là “two stage”:



Hình 1.4: Kiến trúc mạng Faster-RCNN

Trong phần đầu tiên, các bản đồ đặc trưng của ảnh gốc được tạo bởi các mạng trích xuất đặc trưng (VGG, ResNet, Inception, Inception Resnet -v2, v.v.). Và bản đồ đặc trưng từ một số lớp tích chập trung gian đã chọn được sử dụng để dự đoán các vùng đề xuất có điểm số đối tượng và vị trí theo Mạng đề xuất khu vực (RPN). Giai đoạn này chỉ đưa ra điểm số ước tính xác suất của đối tượng hoặc không đối tượng và bounding box cho mỗi vùng đề xuất bằng lớp two-class softmax và hàm mất mát mạnh mẽ (robust loss function) (Smooth L1). Trong phần thứ hai, vị trí của các vùng đề xuất được sử dụng để cắt các vùng đặc trưng từ cùng một bản đồ đặc trưng trung gian bằng cách sử dụng lớp ROI pooling (Region of Interest Pooling - RoI Pooling). Và bản đồ đặc trưng cho từng vùng đề xuất được cung cấp cho phần còn lại của mạng để dự đoán điểm số cụ thể của lớp và tinh chỉnh vị trí bounding box. Mạng này được chia sẻ một phần tính toán bằng cách cắt các vùng đề xuất từ bản đồ đặc trưng được tạo bởi cùng một lớp tích chập trung gian trong phần đầu tiên. Vấn đề của phương pháp này là mỗi vùng đề xuất vẫn phải được cho vào phần còn lại của mạng để tính toán riêng. Do đó, tốc độ phát hiện phụ thuộc vào số vùng đề xuất từ RPN.

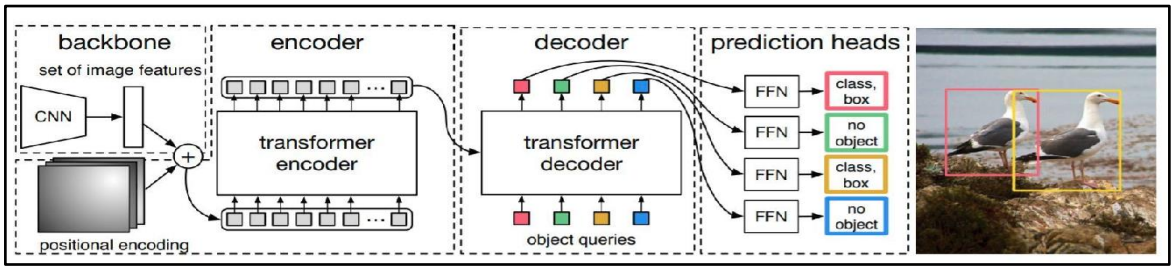
Một trong những mô hình "one-stage" được sử dụng phổ biến là SSD. Kiến trúc mạng SSD được chia làm 3 phần:



Hình 1.5: Kiến trúc mạng SSD

Phần 1, Các lớp tích chập cơ bản bao gồm mạng trích xuất tính năng như VGG ResNet, Inception, Inception Resnet -v2, v.v. Lớp tích chập trung gian của phần này tạo ra một bản đồ đặc trưng có tỷ lệ lớn, có thể được chia thành nhiều có kích thước nhỏ hơn, ít đặc trưng hơn để phát hiện các đối tượng nhỏ hơn. Phần 2, Các lớp tích chập bổ sung nối với lớp cuối cùng của mạng tích chập cơ bản. Phần này của các lớp tạo ra các bản đồ đặc trưng đa tỷ lệ có kích thước lớn hơn của các trường tiếp nhận để phát hiện đối tượng lớn hơn. Phần 3, Các lớp tích chập dự đoán sử dụng bộ lọc tích chập nhỏ dự đoán các vị trí bounding box và độ tin cậy cho các lớp đối tượng. Để giữ nguyên translation variance, SSD chọn các lớp trước đó để tạo bản đồ đặc trưng quy mô lớn, được sử dụng để phát hiện các đối tượng nhỏ. Tuy nhiên, các đặc trưng trong các bản đồ này từ các lớp trước đó không đủ phức tạp, điều này dẫn đến hiệu suất kém hơn trên các đối tượng nhỏ hơn.

Gần đây, mô hình DETR (Detection Transformer) [15] được phát triển, sử dụng Transformer để phát hiện đối tượng. Khác với những phương pháp truyền thống ở trên dựa trên việc hiệu chỉnh phân loại các loại đối tượng và độ tin cậy của hộp bao vật thể trên anchor boxes được định nghĩa từ trước. Vì Transformer thực chất biến đổi chuỗi nên DETR có thể coi như là quá trình biến đổi từ chuỗi hình ảnh đến đối tượng truy vấn.Kiến trúc mạng DETR bao gồm 3 thành phần chính Backbone, Encoder và Decoder:



Hình 1.6: Kiến trúc mạng DETR

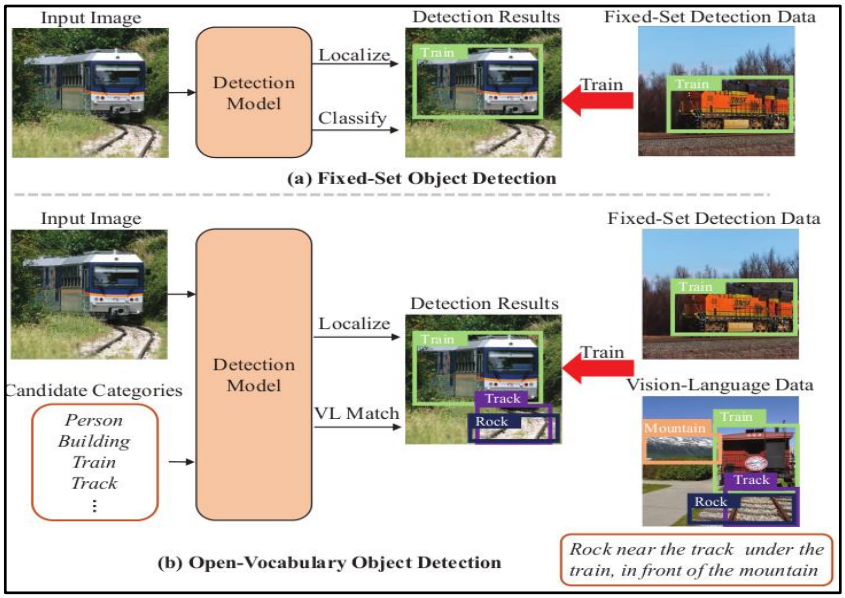
Đầu tiên, DETR sử dụng một Backbone (như là ResNet, ViT,..) để trích xuất đặc trưng, ta thu được bộ đặc trưng của ảnh có chiều là C×H×W sau đó một lớp tích chập 1×1 được sử dụng để giảm số chiều của C giảm xuống còn d, ta được các đặc trưng mới có chiều là d×H×W. Sau khi giảm chiều, các đặc trưng sẽ được thêm vị trí (Spatial Position Encoding) rồi sau có đưa vào Encoder. Cấu trúc của DETR Decoder về cơ bản cũng tương tự như Transformer, khác biệt là Decoder decode N đối tượng một cách song song và cách thêm vị trí (Position Encoding) ở Decoder. DETR Decoder có 2 đầu vào: 1 là đầu ra của Encoder, 2 là đối tượng truy vấn. Đối tượng truy vấn cũng được tác giả gọi là learned positional embedding. Có thể hiểu đơn giản là dựa vào các đặc trưng đã được encode bởi DETR Encoder, DETR Decoder sẽ chuyển đổi 100 truy vấn thành 100 mục tiêu. Cuối cùng, các lớp dự đoán được đưa ra thông qua lớp Linear và Box prediction được đưa ra bởi MLP. Mô hình DETR là

một hướng tiếp cận mới so với hướng tiếp cận CNN truyền thống. Vấn đề của DETR cũng như các mô hình trước đó là khá tệ trong việc phát hiện đối tượng nhỏ.

1.2.2 Phát hiện đối tượng từ vựng mở

Phương pháp phát hiện đối tượng truyền thống xác định vị trí và phân loại các đối tượng trong một tập hợp danh mục cố định, như được thể hiện trong Hình 1.7. Do đó, người lập trình phải liên tục huấn luyện lại mô hình để phù hợp với các ứng dụng thực tế khác nhau, vì các ứng dụng khác nhau thường có các tập hợp danh mục các đối tượng khác nhau. Điều này gây những khó khăn về mặt thu thập dữ liệu cho các đối tượng mới không phải lúc nào cũng dễ kiếm và đủ đa dạng. Cùng với đó việc mất thêm nhiều thời gian thu thập dữ liệu và đào tạo lại mô hình cũng có thể ảnh hưởng đến tiến độ của dự án.

Với sự phát triển của công nghệ kèm theo sự phát triển của dữ liệu, những ý tưởng mới kết hợp giữa văn bản, để phát hiện đối tượng trong ảnh tốt hơn. Và một hướng đi mới cho bài toán này chính là phát hiện đối tượng từ vựng mở (open-vocabulary- OV). Nhờ sự kết hợp đào tạo giữa cả dữ liệu văn và hình ảnh, phát hiện đối tượng từ vựng mở, mô hình có thể phát hiện được cả những đối tượng chưa được huấn luyện. Nhận đầu vào là một cặp hình ảnh – văn bản (image-text), văn bản gồm những danh từ cần phát hiện trong bức ảnh, sau đó mô hình phát hiện đối tượng từ vựng mở sẽ cho ra kết quả phát hiện gồm các hộp bao vật thể (bounding boxes) và tên ứng với các danh từ của đối tượng. Như vậy, phát hiện đối tượng từ vựng mở đã khắc phục được vấn đề tập đối tượng phát hiện bị hạn chế, sự kết giữa văn bản và hình ảnh sẽ giúp tăng độ chính xác cho phát hiện đối tượng trong ảnh.

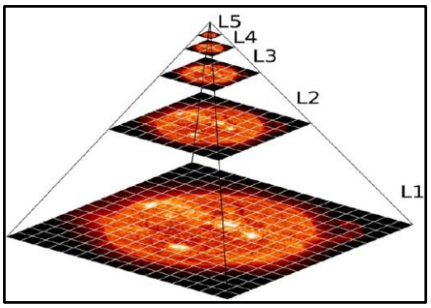


Hình 1.7: Phát hiện đối tượng truyền thống và phát hiện đối tượng từ vựng mở

1.2.3 Một số kỹ thuật hỗ trợ phát hiện đối tượng nhỏ.

Có một số kỹ thuật hỗ trợ phát hiện đối tượng nhỏ trong thị giác máy tính. Các phương pháp này sử dụng các chiến lược và thuật toán khác nhau để cải thiện hiệu suất phát hiện, đặc biệt là cho các đối tượng có kích thước nhỏ. Dưới đây là một số kỹ thuật phổ biến hiện tại.

Kim tự tháp ảnh (Image Pyramid): Phương pháp này tạo ra nhiều phiên bản với tỷ lệ khác nhau của ảnh đầu vào bằng cách thực hiện giảm mẫu hoặc tăng mẫu. Các phiên bản này, gọi là các cấp độ kim tự tháp, cung cấp các độ phân giải khác nhau của ảnh. Các mô hình phát hiện đối tượng có thể áp dụng thuật toán phát hiện trên mỗi cấp độ pyramid để xử lý các đối tượng ở các tỷ lệ khác nhau. Trong hình dưới đây, kỹ thuật kim tự tháp ảnh đã được áp dụng cho một hình ảnh của mặt trời. Phương pháp này cho phép phát hiện các đối tượng nhỏ bằng cách tìm kiếm chúng ở các cấp độ kim tự tháp thấp hơn, nơi chúng có thể nổi bật và có thể phân biệt tốt hơn.



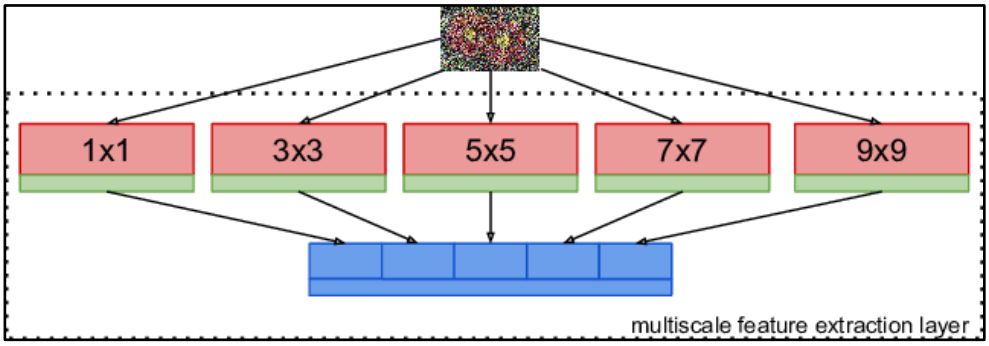
Hình 1.8: Kỹ thuật kim tự tháp ảnh

Cửa sổ trượt (Sliding Window): Phương pháp này bao gồm việc trượt một cửa sổ có kích thước cố định trên ảnh ở các vị trí và tỷ lệ khác nhau. Tại mỗi vị trí cửa sổ, bộ phát hiện đối tượng áp dụng một mô hình phân loại để xác định xem có đối tượng nào xuất hiện hay không. Bằng cách xem xét các kích thước và vị trí cửa sổ khác nhau, bộ phát hiện có thể tìm kiếm hiệu quả các đối tượng nhỏ trên toàn bộ ảnh. Tuy nhiên, phương pháp sliding window có thể tốn kém về mặt tính toán, đặc biệt là khi xử lý ảnh lớn hoặc nhiều tỷ lệ khác nhau.

Hình 1.9: Phương pháp cửa sổ trượt

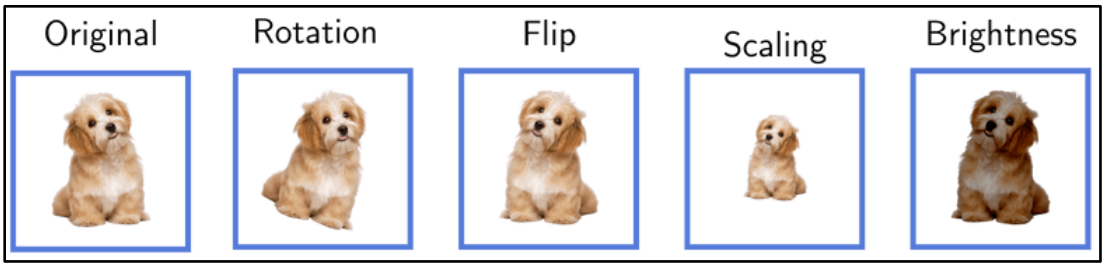
Kỹ thuật trích xuất đặc trưng đa tỷ lệ (Multiple Scale Feature Extraction): Phương pháp này xử lý ảnh ở nhiều độ phân giải khác nhau hoặc áp dụng các lớp tích

chập với các lĩnh vực nhận thức khác nhau. Bằng cách kết hợp các đặc trưng từ các tỷ lệ khác nhau, mô hình có thể tăng hiệu quả nhận diện các đối tượng nhỏ và lớn trong ảnh. Phương pháp này giúp bảo tồn các chi tiết nhỏ liên quan đến việc phát hiện các đối tượng kích thước nhỏ.



Hình 1.10: Kỹ thuật trích xuất đặc trưng đa tỷ lệ

Tăng cường dữ liệu (Data augmentation): Đây là một trong những kỹ thuật nổi tiếng nhất trong thị giác máy tính có thể cải thiện hiệu suất phát hiện đối tượng nhỏ bằng cách tạo ra các mẫu huấn luyện bổ sung. Các phương pháp tăng cường như cắt ngẫu nhiên, thay đổi kích thước, xoay, hoặc thêm nhiễu nhân tạo có thể giúp tạo ra các biến thể trong tập dữ liệu, cho phép mô hình học các đặc trưng mạnh mẽ cho các đối tượng nhỏ. Các kỹ thuật tăng cường cũng có thể mô phỏng các tỷ lệ, quan điểm và che phủ khác nhau của đối tượng, giúp mô hình phát hiện tổng quát tốt hơn với các tình huống thực tế.



Hình 1.11: Tăng cường dữ liệu

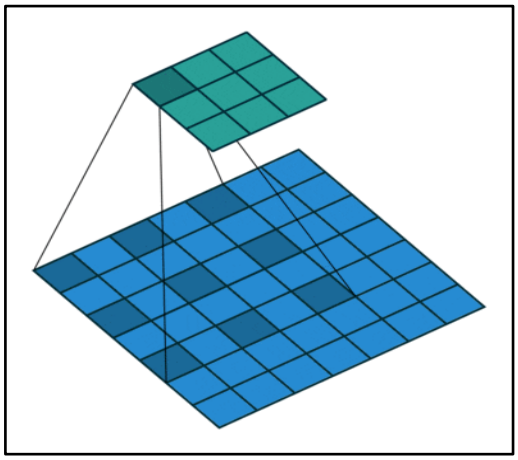
Học chuyển giao (Transfer learning): Phương pháp này liên quan đến việc tận dụng kiến thức đã học từ việc tiền huấn luyện trên các bộ dữ liệu quy mô lớn (ví dụ: ImageNet) và áp dụng nó vào các nhiệm vụ phát hiện đối tượng. Các mô hình được tiền huấn luyện, đặc biệt là những mô hình với kiến trúc mạng CNN sâu, ghi lại các đặc trưng phân cấp phong phú có lợi cho việc phát hiện đối tượng nhỏ. Bằng cách điều chỉnh mô hình được tiền huấn luyện trên các bộ dữ liệu mục tiêu, mô hình hiện đối tượng có thể nhanh chóng thích nghi với các nhiệm vụ mới, sử dụng các biểu diễn đã học và cung cấp khả năng phát hiện tốt hơn cho các đối tượng nhỏ.

1.3 Vấn đề phát hiện đối tượng kích thước nhỏ trong ảnh

1.3.1 Nguyên nhân phát hiện đối tượng kích thước nhỏ không tốt trong ảnh Drone

Phát hiện đối tượng trong ảnh drone được ứng dụng trong thực tế rất nhiều và nó cũng có rất nhiều vấn đề cần khắc phục. Một trong những vấn đề gặp phải của nó khi sử dụng những phương pháp phát hiện đối tượng truyền thống hiện nay đó là những đối tượng loại nhỏ thường hay bị bỏ sót. Ảnh chụp từ drone chụp được rất nhiều vật thể, góc camera chụp rộng. Nhiệm vụ phát hiện đối tượng gặp nhiều khó khăn do kích thước nhỏ và độ phân giải thấp của các đối tượng, cũng như các yếu tố khác như che khuất, nhiễu nền và biến đổi trong điều kiện ánh sáng. Ngoài ra còn rất nhiều lý do khiến các phương pháp phát hiện đối tượng truyền thống phát hiện đối tượng kích thước nhỏ kém được nêu ra ở dưới đây.

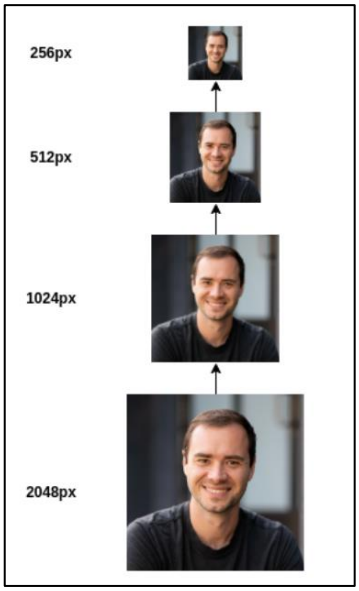
Đầu tiên là do giới hạn vùng tiếp nhận (Limited Receptive Field). Khái niệm này dùng để chỉ phạm vi không gian của ảnh đầu vào (trường nhìn) có tác động đến đầu ra của một nơron hoặc bộ lọc cụ thể trong mạng nơron tích chập (CNN). Mỗi nơron trong một lớp tích chập sẽ có một trường nhìn, tức là vùng của ảnh đầu vào mà nó nhận thức. Kích thước của trường nhìn thường nhỏ hơn kích thước toàn bộ ảnh đầu vào. Khi đi sâu vào mạng, trường nhìn của các nơron sẽ càng nhỏ đi do các phép tích chập và lớp gộp. Trong các mô hình phát hiện đối tượng thông thường, vùng tiếp nhận có thể bị hạn chế, điều này có nghĩa là mạng không có đủ thông tin ngữ cảnh xung quanh các đối tượng nhỏ. Kết quả là, mô hình có thể gặp khó khăn trong việc phát hiện và xác định vị trí chính xác các đối tượng này do vùng tiếp nhận không đủ.



Hình 1.12: Vùng tiếp nhận trong CNN

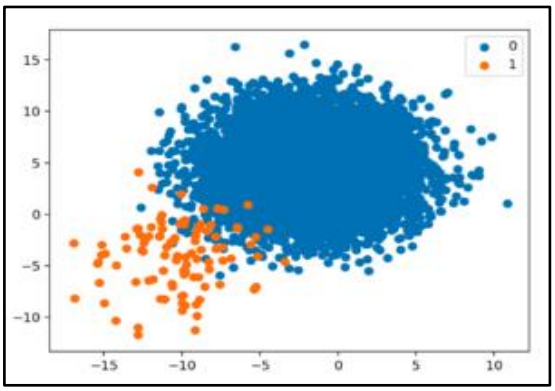
Nguyên nhân thứ hai là do biến đổi tỷ lệ (Scale Variation). Các đối tượng nhỏ thể hiện sự biến đổi tỷ lệ đáng kể so với các đối tượng lớn trong một hình ảnh. Các bộ phát hiện đối tượng được huấn luyện trên các bộ dữ liệu chủ yếu gồm các đối tượng lớn, chẳng hạn như ImageNet [16] hoặc COCO [17], có thể gặp khó khăn trong

việc tổng quát hóa cho các đối tượng nhỏ do sự chênh lệch về tỷ lệ. Trong hình dưới đây, biến đổi tỷ lệ đã được áp dụng. Sự biến đổi về kích thước có thể gây khó khăn trong việc phù hợp với các biểu diễn đối tượng đã học, dẫn đến hiệu suất phát hiện giảm đi đối với các đối tượng nhỏ.



Hình 1.13: Biến đổi tỷ lệ của đối tượng trong ảnh

Lý do thứ ba, thiên hướng dữ liệu huấn luyện (Training Data Bias). Các mô hình phát hiện đối tượng thường được huấn luyện trên các bộ dữ liệu quy mô lớn, có thể chứa các thiên hướng về các đối tượng lớn do sự phổ biến của chúng. Thiên hướng này có thể không cố ý ảnh hưởng đến hiệu suất của bộ phát hiện đối tượng khi xử lý các đối tượng nhỏ. Kết quả là, mô hình có thể chưa được tiếp xúc đủ với các ví dụ huấn luyện đa dạng về các đối tượng nhỏ. Điều này dẫn đến sự thiếu ổn định và độ chính xác phát hiện giảm đi đối với các trường hợp đối tượng nhỏ.



Hình 1.14: Mất cân bằng dữ liệu trong bộ dữ liệu

Cuối cùng là việc xác định vị trí chính xác của các đối tượng nhỏ có thể gặp khó khăn do độ phân giải không gian hạn chế của các bản đồ đặc trưng trong kiến

trúc mạng nơron tích chập. Các chi tiết tinh tế cần thiết để xác định vị trí chính xác có thể bị mất hoặc trở nên không thể phân biệt ở độ phân giải thấp hơn. Các đối tượng nhỏ có thể bị che khuất bởi các đối tượng lớn khác hoặc các nền nhiễu, làm khó khăn thêm cho việc xác định vị trí. Những yếu tố này có thể góp phần làm cho các bộ phát hiện đối tượng thông thường không thể xác định và phát hiện các đối tượng nhỏ một cách chính xác.

1.3.2 Phương pháp phát hiện đối tượng đề xuất

Các phương pháp được nêu ở mục 1.2.2 giúp mô hình cải thiện độ chính xác khi phát hiện các đối tượng kích thước nhỏ. Tuy nhiên những kỹ thuật đó có các nhược điểm như: Pyramid Image và Sliding Window gây tốn kém chi phí tính toán và bộ nhớ. Multiple Scale Feature Extraction cần phải sửa lại kiến trúc mạng khi áp dụng, khó khăn thay đổi các kiến trúc mạng phức tạp. Data augmentation có thể gây mất thông tin khi áp dụng các biến đổi không phù hợp, giảm chất lượng và độ tin cậy của dữ liệu. Transfer learning không cải thiện nhiều kết quả cho các đối tượng nhỏ hiếm gặp. Bên cạnh đó các phương pháp phát hiện đối tượng từ vựng đóng cũng còn nhiều vấn đề đối với các đối tượng kích thước nhỏ. Nhất là các mô hình dựa trên kiến trúc CNN thường bị Limited Receptive Field như đã nêu ở trên.

Để khắc phục những vấn đề được nêu trên, đề án đề xuất sử dụng phương pháp kết hợp giữa mô hình phát hiện đối tượng từ vựng mở Vision Transformer for Open-World Localization (OWL-ViT) và kỹ thuật Slicing Aided Hyper Inference (SAHI) giúp phát hiện các đối tượng kích thước nhỏ tốt hơn. Mô hình OWL-ViT được thiết kế dựa trên kiến trúc cơ bản của Vision Transformer và Text Transformer chứ không sử dụng kiến trúc CNN. Mô hình sẽ được huấn luyện trước nó với một tập dữ liệu lớn gồm các cặp hình ảnh-văn bản. Nhờ sự kết hợp đào tạo giữa cả dữ liệu văn và hình ảnh, phát hiện đối tượng từ vựng mở, mô hình có thể phát hiện được cả những đối tượng chưa được huấn luyện. Như vậy, phát hiện đối tượng từ vựng mở OWL-ViT đã khắc phục được vấn đề tập đối tượng phát hiện bị hạn chế, sự kết giữa văn bản và hình ảnh sẽ giúp tăng độ chính xác cho phát hiện đối tượng trong ảnh. Bên cạnh đó, sử dụng thêm kỹ thuật SAHI, một trong những kỹ thuật mới và tốt nhất hiện nay. Với phương pháp này chia cắt hình ảnh giúp cho việc phát hiện đối tượng kích thước nhỏ hiệu quả hơn, đã khắc phục một vấn đề thường xuyên gặp phải trong các mô hình phát hiện đối tượng.

1.4 Kết luận chương

Tại chương này, đề án cung cấp một cái nhìn tổng quan về bài toán phát hiện đối tượng. Phân biệt các nhiệm vụ phân loại hình ảnh, định vị đối tượng với phát hiện đối tượng. Trình bày bài toán phát hiện đối tượng trong ảnh chụp từ drone và các ứng dụng thực tế như giám sát an ninh, quản lý tài nguyên, giám sát xây dựng,... Đề án cũng trình bày các nghiên cứu liên quan đến vấn đề phát hiện đối tượng kích thước

thước nhỏ trong ảnh chụp Drone. Cụ thể là phát hiện đối tượng từ vựng đóng như SSD, Faster-RCNN, DETR và các vấn đề của chúng. Bên cạnh đó là trình bày các kỹ thuật phổ biến hỗ trợ phát hiện đối tượng nhỏ trong ảnh: kim tự tháp ảnh, cửa sổ trượt, trích xuất đặc trưng đa tỷ lệ, tăng cường dữ liệu, học chuyển giao . Đề án chỉ ra vấn đề hiện tại của phát hiện đối tượng kích thước nhỏ trong ảnh Drone. Nêu ra nguyên nhân của vấn đề, nhược điểm của các phương pháp trước. Từ đó đề xuất giải pháp đề xuất kết hợp giữa OWL-ViT và SAHI cho bài toán. Các chương sau sẽ đi vào khía cạnh kỹ thuật của phương pháp này.

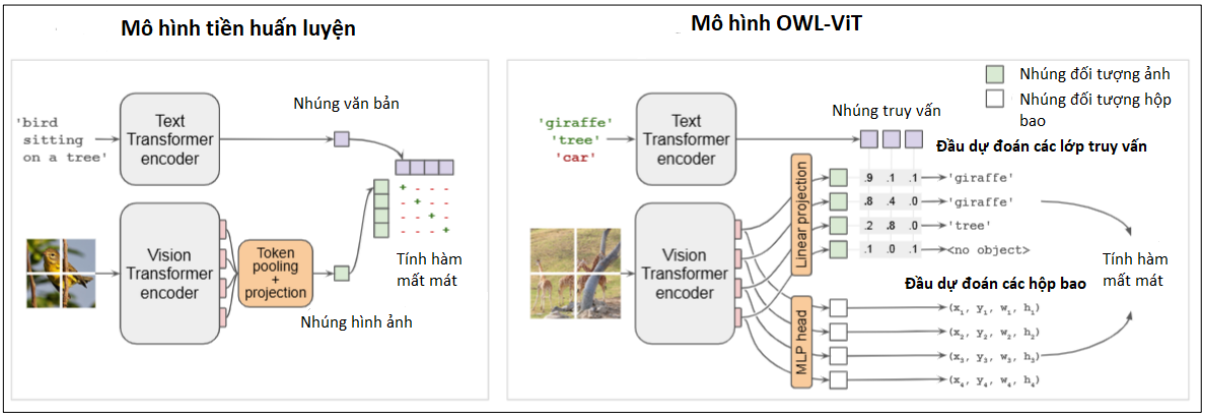
Chương 2 - PHƯƠNG PHÁP OWL-VIT KẾT HỢP SAHI

2.1 Mô hình Vision Transformer cho Open-World Localization (OWL-ViT)

2.1.1 Tổng quan mô hình OWL-ViT

Gần đây, các mô hình phát hiện đối tượng truyền thống gặp vấn đề thường bị giới hạn loại đối tượng trong một tập hợp nhỏ, cố định các từ ngữ, vì việc thu thập dữ liệu huấn luyện với không gian nhãn lớn hoặc mở là tốn kém và tốn thời gian. Tuy nhiên, điều này đã thay đổi với sự phát triển của các bộ mã hóa ngôn ngữ mạnh mẽ và huấn luyện hình ảnh-văn bản trái ngược. Các mô hình này học một biểu diễn chung của hình ảnh và văn bản từ các cặp hình ảnh-văn có sẵn phong phú trên web. Bằng cách tận dụng lượng lớn dữ liệu hình ảnh-văn bản, các mô hình đã huấn luyện đối ngẫu. (contrastive training) đã mang lại cải tiến đáng kể trong nhiệm vụ phân loại.

Với ý tưởng trên, OWL-ViT bắt đầu với kiến trúc Vision Transformer, đã được chứng minh là có khả năng mở rộng cao, và tiến hành tiền huấn luyện đối ngẫu trên một tập dữ liệu hình ảnh-văn bản lớn. Sau đó để chuyển giao mô hình sang nhiệm vụ phát hiện đối tượng, thực hiện một số thay đổi. Đầu tiên, loại bỏ lớp pooling token cuối cùng và thay vào đó gắn một đầu phân loại nhãn và một đầu dự đoán bounding box cho mỗi token đầu ra của Transformer Encoder. Phân loại từ vựng mở mở được kích hoạt bằng cách thay thế trọng số của lớp phân loại cố định bằng class-name embeddings được thu được từ mô hình văn bản. Tiếp theo, điều chỉnh lại mô hình đã được tiền huấn luyện trên các tập dữ liệu phát hiện tiêu chuẩn bằng cách sử dụng hàm mất mát bipartite matching. Như vậy, cả mô hình hình ảnh và văn bản đều được điều chỉnh lại từ đầu đến cuối giống như hình 2.1. Bên trái là mô hình tiền huấn luyện, bên phải là mô hình OWL-ViT sau khi được điều chỉnh lại.



Hình 2.1: Kiến trúc mô hình OWL-ViT

Để phân loại các đối tượng đã phát hiện với từ vựng mở, mô hình sử dụng text embeddings, thay vì class embeddings đã học, trong lớp đầu ra của đầu phân loại.

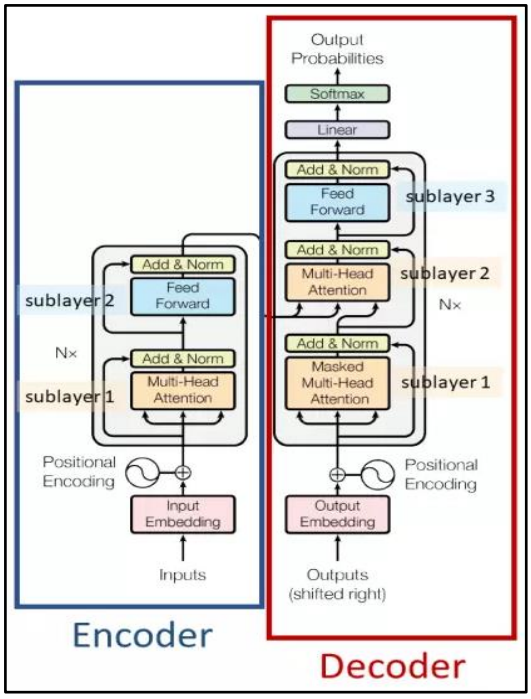
Các text embeddings, được gọi là truy vấn (queries), được tạo ra bằng cách đưa tên đối tượng hoặc các mô tả đối tượng văn bản khác qua text encoder. Nhiệm vụ của mô hình sau đó là dự đoán, đối với mỗi đối tượng, một bounding box và xác suất mà mỗi truy vấn áp dụng cho đối tượng đó. Các truy vấn có thể khác nhau cho mỗi hình ảnh. Kết quả là, mỗi hình ảnh sẽ có không gian nhãn riêng biệt, được xác định bởi một tập hợp các chuỗi văn bản. Phương pháp này bao gồm cả việc phát hiện đối tượng với từ vựng đóng, trong đó toàn bộ tập hợp tên danh mục đối tượng được sử dụng làm tập truy vấn cho mỗi hình ảnh.

Khác với một số phương pháp khác, mô hình không kết hợp tất cả các truy vấn cho một hình ảnh thành một chuỗi token (token sequence) duy nhất . Thay vào đó, mỗi truy vấn bao gồm một token sequence riêng biệt đại diện cho một mô tả đối tượng riêng biệt và được xử lý riêng biệt bởi bộ mã hóa văn bản. Ngoài ra, kiến trúc của mô hình không bao gồm việc kết hợp giữa image encoder và text encoder. Mặc dù việc kết hợp sớm có vẻ có lợi theo nhiều suy đoán nhưng thực tế nó làm giảm hiệu suất suy luận đáng kể vì việc mã hóa một truy vấn yêu cầu một quá trình chuyển tiếp qua toàn bộ mô hình hình ảnh và cần được lặp lại cho mỗi kết hợp hình ảnh/truy vấn. Trong OWL-ViT có thể tính toán các nhúng truy vấn độc lập với hình ảnh, cho phép sử dụng hàng ngàn truy vấn cho mỗi hình ảnh, nhiều hơn nhiều so với việc kết hợp sớm.

Các head đặc thù cho việc phát hiện đối chỉ chiếm tối đa 1,1% (tùy thuộc vào kích thước của mô hình) số lượng tham số trong mô hình. Điều này có nghĩa là phần lớn tham số của mô hình tập trung vào việc image encoder và text encoder, trong khi chỉ một tỷ lệ nhỏ được sử dụng cho các nhiệm vụ cụ thể liên quan đến phát hiện đối tượng. Dưới đây, đề án sẽ trình bày rõ hơn về image encoder và text encoder của mô hình.

2.1.2 Text Encoder của mô hình

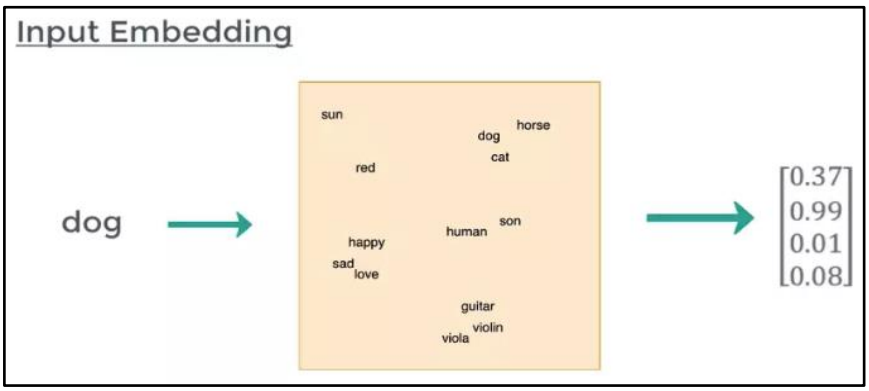
Trong mô hình OWL-ViT sử dụng Encoder của Transformer tiêu chuẩn để mã hóa phần văn bản. Transformer đã được giới thiệu trong bài báo "Attention Is All You Need" của Vaswani et al. vào năm 2017. Nó đã trở thành một trong những kiến trúc quan trọng nhất và phổ biến nhất cho các tác vụ như dịch máy, dự đoán từ vựng tiếp theo, tổng hợp văn bản. Một trong những đặc điểm quan trọng của Transformer là khả năng xử lý đầu vào dài và tổng quát hóa tốt trên các ngôn ngữ khác nhau. Mô hình này đã đạt được nhiều kết quả ấn tượng trong các nhiệm vụ xử lý ngôn ngữ tự nhiên và đã được ứng dụng rộng rãi trong các công cụ và ứng dụng liên quan đến xử lý ngôn ngữ.



Hình 2.2: Kiến trúc của Transformer

Nhìn vào sơ đồ kiến trúc của Transformer ở trên có thể thấy rằng nó được chia thành hai phần rõ ràng đó là encoder và decoder. Do mô hình OWL-ViT chỉ sử dụng phần encoder nên đề án sẽ tập trung phân tích các phần của Transformer encoder và bỏ qua phần decoder.

Đầu tiên của phần encoder là Input Embeddings, máy tính không hiểu câu chữ mà chỉ đọc được số, vectơ, ma trận; vì vậy ta phải biểu diễn câu chữ dưới dạng vectơ, gọi là input embedding. Điều này đảm bảo các từ gần nghĩa có vectơ gần giống nhau. Hiện nay đã có khá nhiều pretrained word embeddings như GloVe, Fasttext, gensim Word2Vec,... cho chúng ta lựa chọn.



Hình 2.3: Input embedding trong Transformer

Word embeddings phần nào cho giúp ta biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau. Đó là lý do

Transformer có thêm một phần Positional Encoding để thêm thông tin về vị trí của một từ.

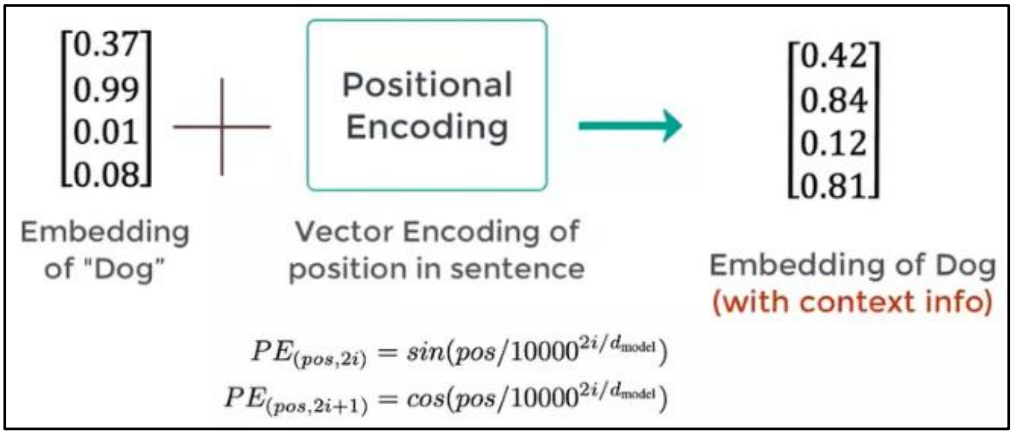
$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i})} = \sin\left(\frac{\mathrm{pos}}{10000^{\frac{2\mathrm{i}}{\mathrm{d}_{\mathrm{model}}}}}\right) \tag{2.1}$$

$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i}+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{\frac{2\mathrm{i}}{\mathrm{d}_{\mathrm{model}}}}}\right) \tag{2.2}$$

Trong đó:

- Pos là vị trí của từ trong câu,
- PE là giá trị phần tử thứ i trong embeddings có độ dài d_{model}

Sau đó, ta cộng vectơ Positional Encoding (PE) vào vectơ Input Embeddings.

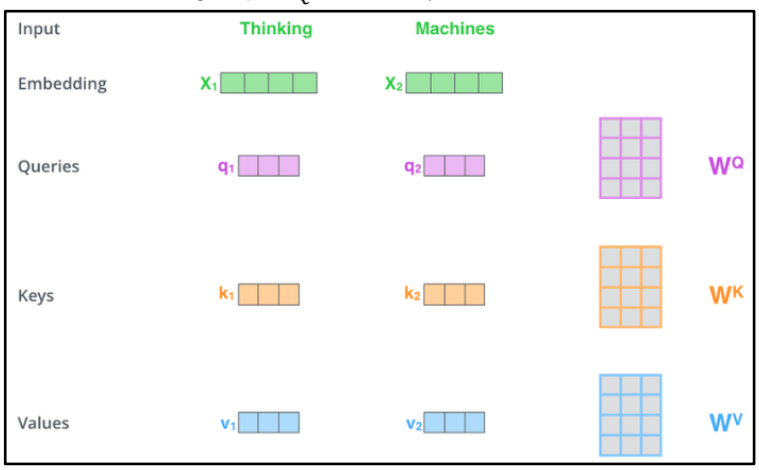


Hình 2.4: Positional Encoding của Transformer

Tiếp đến là Self-Attention-cơ chế giúp Transformer "hiểu" được sự liên quan giữa các từ trong một câu. Ví dụ như từ "kicked" trong câu "I kicked the ball" (tôi đã đá quả bóng) liên quan đến các từ khác như thế nào? Rõ ràng nó liên quan mật thiết đến từ "I" (chủ ngữ), "kicked" là chính nó lên sẽ luôn "liên quan mạnh" và "ball" (vị ngữ). Ngoài ra từ "the" là giới từ nên sự liên kết với từ "kicked" gần như không có. Vậy làm sao Self-Attention trích xuất những đặc trưng "liên quan" này? Ta nhìn lại với kiến trúc tổng thể ở trên, có thể thấy đầu vào của các module Multi-head Attention (bản chất là Self-Attention) có 3 mũi tên, đó chính là 3 vectơ Querys (Q), Keys (K) và Values (V). Từ 3 vectơ này, ta sẽ tính vectơ attention Z cho một từ theo công thức sau:

$$\mathrm{Z} = \mathrm{softmax}\left(\frac{\mathrm{Q}.\mathrm{K}^{\mathrm{T}}}{\sqrt{d_{Q,K,V}}}\right).\ \mathrm{V} \tag{2.3}$$

Công thức này khá đơn giản và được thực hiện như sau. Đầu tiên, để có được 3 vectơ Q, K, V, input embeddings được nhân với 3 ma trận trọng số tương ứng (được điều chỉnh trong quá trình huấn luyện) W_Q, W_K, W_V.

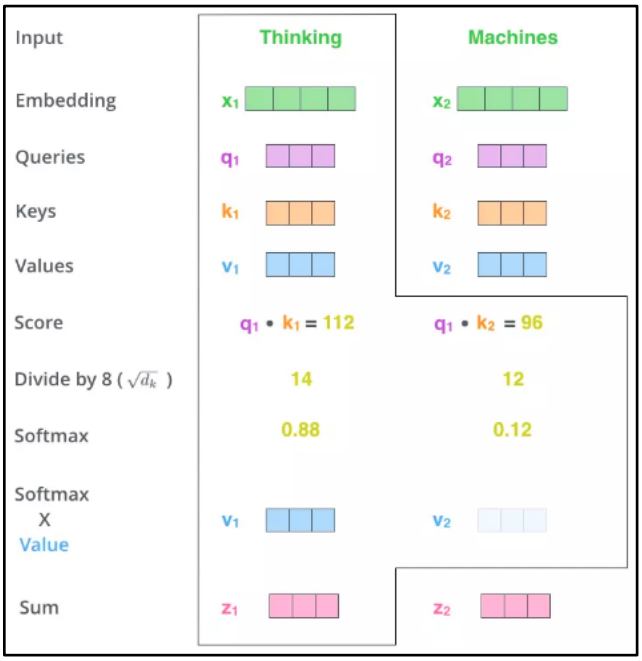


Hình 2.5. Ba vectơ Querys (Q), Keys (K) và Values (V) và W_Q, W_K, W_V trong cơ chế Self-Attention của Transformers

Lúc này, vectơ K đóng vai trò như một khóa đại diện cho từ, và Q sẽ truy vấn đến các vectơ K của các từ trong câu bằng cách nhân chập với những vectơ này. Mục đích của phép nhân chập để tính toán độ liên quan giữa các từ với nhau. Theo đó, 2 từ liên quan đến nhau sẽ có "Score" lớn và ngược lại. Bước thứ 2 là bước "Scale", đơn giản chỉ là chia "Score" cho căn bậc hai của số chiều của Q/K/V (trong hình chia 8 vì Q/K/V là 64-D vectơ). Việc này giúp cho giá trị "Score" không phụ thuộc vào độ dài của vectơ Q/K/V. Bước thứ 3 là softmax các kết quả vừa rồi để đạt được một phân bố xác suất trên các từ.

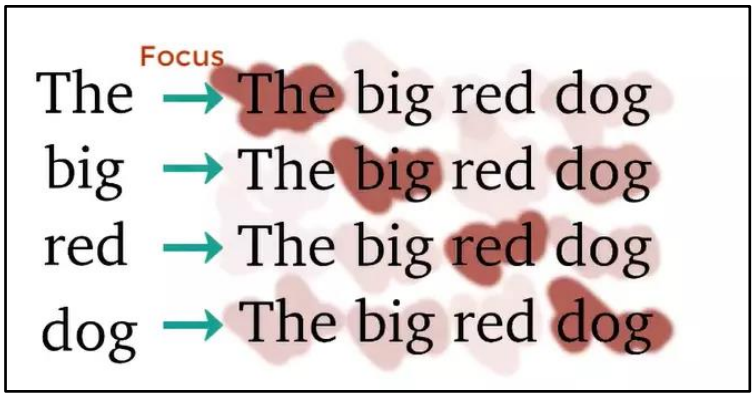
Bước thứ 4 ta nhân phân bố xác suất đó với vectơ V để loại bỏ những từ không cần thiết (xác suất nhỏ) và giữ lại những từ quan trọng (xác suất lớn).

Ở bước cuối cùng, các vectơ V (đã được nhân với softmax output) cộng lại với nhau, tạo ra vectơ attention Z cho một từ. Lặp lại quá trình trên cho tất cả các từ ta được ma trận attention cho 1 câu.



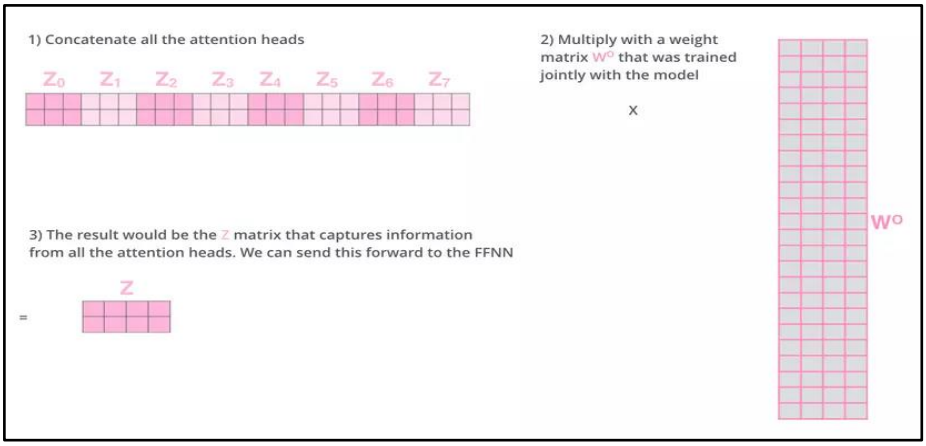
Hình 2.6: Vectơ attention Z được tạo ta cho một từ trong cơ chế Self-Attention

Vấn đề của Self-attention là attention của một từ sẽ luôn "chú ý" vào chính nó. Điều này rất hợp lý thôi vì rõ ràng từ đó phải liên quan đến từ đó nhiều nhất. Minh họa như hình dưới đây:



Hình 2.7: Vấn đề chú ý vào một từ của Self-attention

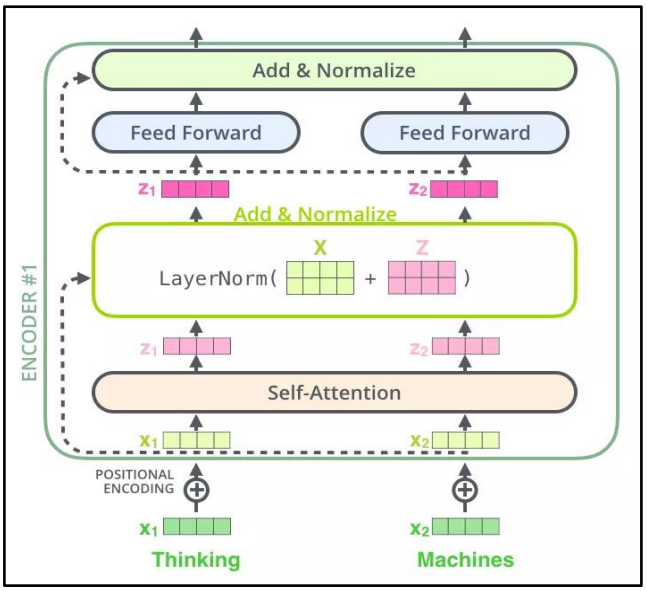
Để tránh xảy ra điều này vì cái ta mong muốn muốn là đặc trưng liên quan giữa các từ khác nhau trong câu. Tác giả đã giới thiệu một phiên bản nâng cấp hơn của Self-attention là Multi-head attention. Ý tưởng rất đơn giản là thay vì sử dụng 1 Self-attention (1 head) thì ta sử dụng nhiều Attention khác nhau (multi-head) và biết đâu mỗi Attention sẽ chú ý đến một phần khác nhau trong câu. Vì mỗi "head" sẽ cho ra một ma trận attention riêng nên ta phải concat các ma trận này và nhân với ma trận trọng số W_O để ra một ma trận attention duy nhất (weighted sum). Và tất nhiên, ma trận trọng số này cũng được điều chỉnh trong khi quá trình huấn luyện.



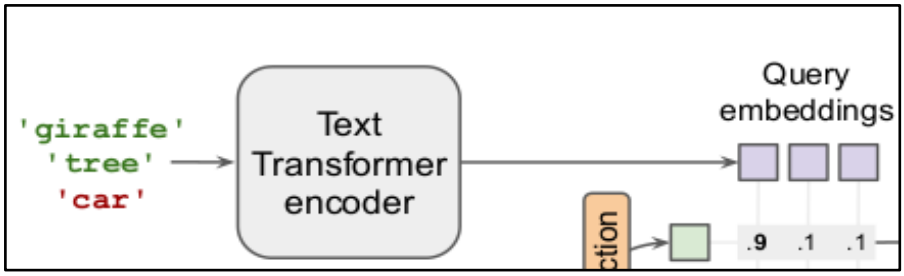
Hình 2.8: Mutil Multi-head Attention trong Transformer

Sau khi đi qua Mutil Multi-head Attention cho ra các vectơ attention Z, chúng sẽ đi qua phần Add & Normalize tạo thành sub-layer 1 như trong mô hình tổng quan về Transformer ở trên. Mỗi sub-layer đều là một khối dư (residual block). Cũng giống như residual blocks trong Computer Vision, các kết nối tắt (skip connections) trong Transformer cho phép thông tin đi qua sub-layer trực tiếp. Thông tin này (x) được cộng với attention (z) của nó và thực hiện chuẩn hóa (normalization) với Layer Normalization.

Cuối cùng là Feed Forward, sau khi được chuẩn hóa, các vectơ z được đưa qua mạng kết nối đầy đủ (fully connected) và cho ra các text/query embeddings. Vì các vectơ này không phụ thuộc vào nhau nên ta có thể tận dụng được tính toán song song cho cả câu.



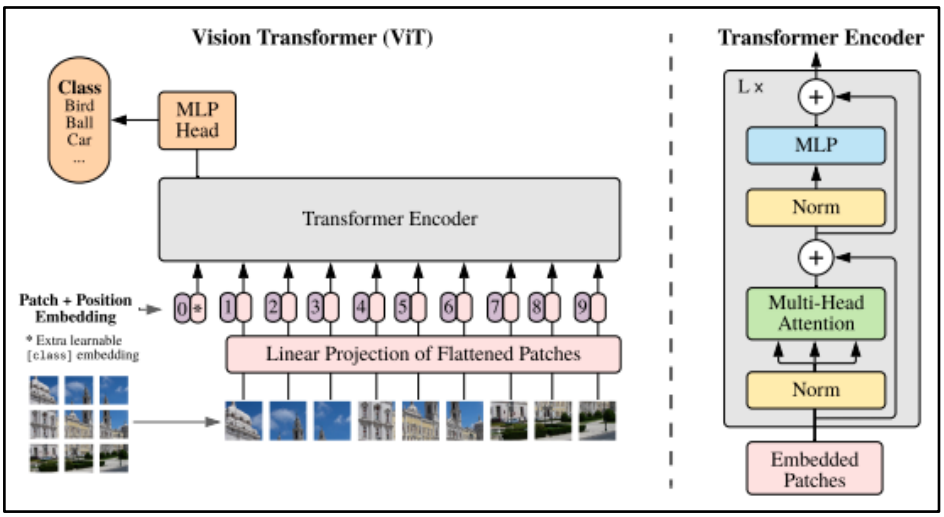
Hình 2.9: Chi tiết Encoder trong Transformer



Hình 2.10: Phần query embeddings sau khi đi qua Text Transformer encoder

2.1.3 Vision Encoder của mô hình

Vision Transformer (ViT) là một kiến trúc mạng nơ-ron sử dụng cơ chế Transformer cho các nhiệm vụ xử lý hình ảnh. Trước đây, kiến trúc Transformer thường được sử dụng cho xử lý văn bản trong lĩnh vực xử lý ngôn ngữ tự nhiên. Tuy nhiên, với sự ra đời của Vision Transformer, nó đã trở thành một trong những phương pháp tiên tiến nhất trong lĩnh vực thị giác máy tính. Trong truyền thống, các mô hình thị giác sử dụng các lớp tích chập (convolutional layers) để trích xuất đặc trưng từ hình ảnh. Tuy nhiên, các lớp tích chập có thể gặp khó khăn trong việc mô hình hóa các mối quan hệ xa giữa các vùng không gian trong hình ảnh. Điều này có thể làm giảm khả năng mô hình tổng quát hóa và xử lý những mẫu mới. Vision Transformer thay thế các lớp tích chập bằng một số lớp Transformer. Mô hình Vision Transformer chia hình ảnh thành các mảnh nhỏ hơn gọi là "điểm chú ý" (patches) và biến đổi chúng thành các vectơ. Các vectơ này sau đó được đưa vào một mạng Transformer, cho phép mô hình học được sự tương tác giữa các điểm chú ý và xử lý thông tin không gian rộng hơn



Hình 2.11: Kiến trúc Vision Transformer

Nhìn vào hình 2.11, dễ thấy kiến trúc của mô hình gồm 3 thành phần chính:

- Linear Projection of Flattened Patches
- Transformer encoder.
- Classification head.

Trong phần đầu tiên Linear Projection of Flattened Patches, khác với các mô hình CNN cho bài toán image classification, ảnh input đầu vào cho mô hình CNN đó là toàn bộ ảnh với kích thước cố định. Tuy nhiên Vision Trans có một cách xử lý khác. Với mỗi ảnh đầu vào, ViT xử lý bằng cách chia ảnh ra thành các phần có kích thước bằng nhau (patch) và sau đó thêm các thông tin cần thiết, quá trình này gọi là Patch Embedding (hình 2.11). Cách chia hình ảnh thành các patch như sau. Với một hình ảnh 3 chiều:

$$\text{3D Image (X)} \ \epsilon \ \text{resolution} \ R^{HxWxC} \tag{2.4}$$

Chuyển hình dạng ảnh 3D thành các patch 2D đã được làm phẳng:

$$\text{Patch Image } (X_p) \ \epsilon \ \text{resolution} \ R^{Nx(P\hat{}2.C)} \tag{2.5}$$

Trong đó:

- 3D Image là ảnh gốc 3 chiều ban đầu với 3 chiều HxWxC tương ứng với chiều cao (height), chiều rộng (width), số kênh màu (channels)
- Patch Image là patch của hình ảnh sau khi chia với độ dài chuỗi N = H . W / P^2 và (P, P) là độ phân giải của mỗi patch.

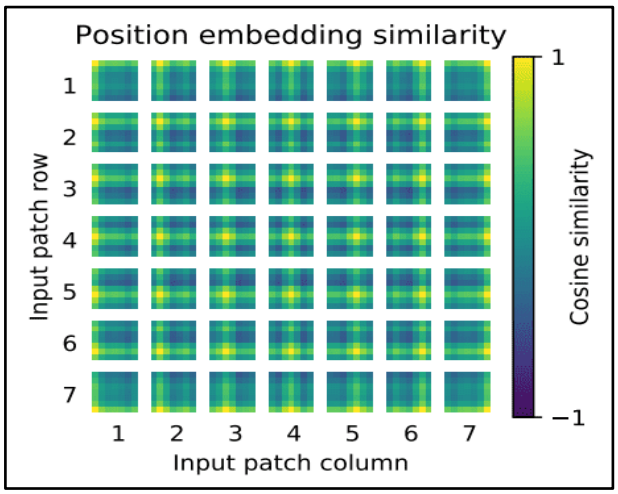
Bước tiếp theo, đưa các patches này về dạng vector bằng cách duỗi thẳng (flatten) các patches này ra. Hình 2.11 trên mô tả phần Linear Projection. Thực chất Linear Projection là một lớp Dense với đầu vào là flattened vector của các patches, đầu ra sẽ là vectơ embedding tương ứng với từng patch.

$$z_i = W * x_i + b \tag{2.6}$$

Trong đó:

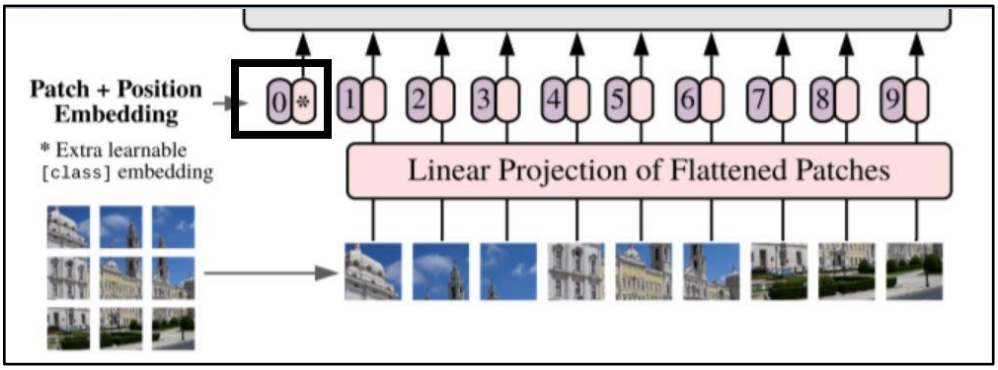
- x_i là flattened vectơ của patch thứ i
- z_i là vectơ embedding đầu ra tương ứng của x_i khi đi qua Linear Projection
- W được gọi là ma trận embedding

Tương tự như với mô hình Transformer gốc. Positional embedding trong mô hình ViT sẽ chứa thông tin về vị trí của patch trong ảnh (spatial information). Nếu như chỉ embedding các patch và đưa vào mô hình Transformer thì với 2 ảnh ở bên trên sẽ hoàn toàn không có sự khác biệt. Do đó ta cần thêm thông tin về vị trí cho mỗi patch. Sau khi có vectơ positional embedding cho mỗi patch ta sẽ cộng các vectơ này tương ứng với embedding vector của từng patch đã tính ở trên và thu được các vector embedding vừa chứa thông tin của vùng ảnh vừa chứa thông tin về vị trí của nó trong ảnh. Vector vị trí này có kích thước 1D giúp giảm kích thước lưu trữ so với vector 2D.



Hình 2.12: Độ tương đồng của các position embedding trong ảnh

Nhìn vào hình 2.12, những gói nào ở cùng hàng/cột sẽ có embedding giống nhau hay có biểu diễn giống nhau. Các thông tin vị trí trong patch embedding là rất cần thiết để mô hình học được thứ tự từng gói trong ảnh qua đó đảm bảo được ngữ nghĩa của từng object trong ảnh.



Hình 2.13: Patch Embedding trong Vision Transformer

Các position embedding được cộng vào các vectơ patch của ảnh tương ứng. Giống như minh họa trong hình 2.13, với 9 patch ảnh được chia tương ứng với 9 position embedding từ 1 đến 9. Tuy nhiên lại còn thừa vị trí 0*. Phần * ở đây chính

là Class Embedding. Chúng ta thêm một embedding có thể học ($z^0 = x_{class}$) vào đầu chuỗi các patch embedding như sau:

$$z^0 = [x_{class};\ x_p^1 E;\ x_p^2 E;\ \dots\ ;\ x_p^N E] + E_{pos} \quad (2.7)$$

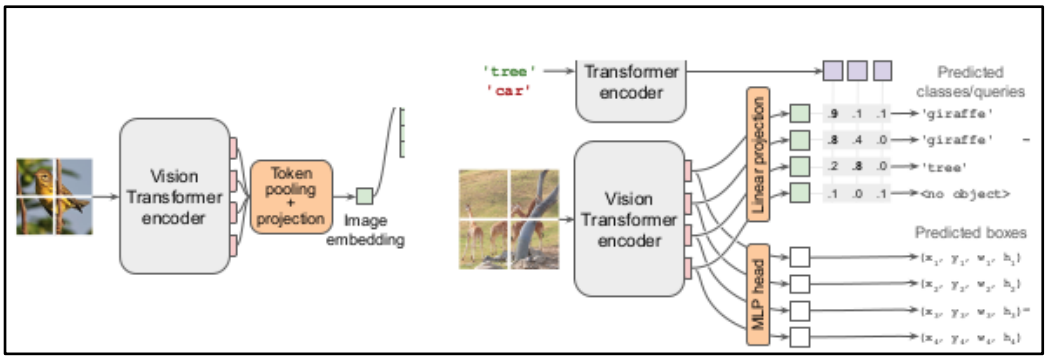
Trong đó:

- $E \in R^{(P\hat{}2.C).D}$
- $E_{pos} \in R^{(N+1).D}$

Ở đây, x_{class} là một nhãn lớp và x_p^N là hình ảnh patch tương ứng với N trong khoảng từ 1 đến n.

Trong quá trình tiền huấn luyện bằng bộ mã hóa Transformer, ta luôn cần một nhãn lớp ở vị trí 0. Khi chúng ta truyền các hình ảnh patch làm đầu vào, luôn cần thêm một token phân loại ở vị trí đầu tiên như được thể hiện trong hình 2.13.

Sau các phần trên ta được các vectơ embedding có kèm vị trí của các patch đã được chia, và nhãn của hình ảnh. Sau đó chúng được đưa vào Transformer Encoder, về phần này đã được trình bày kỹ tại mục 2.1.2 của Transformer.



Hình 2.14: Vision Transformer encoder trong OWL-ViT

Kiến trúc Vision Transformer dùng trong phân loại ở hình 2.14 với các vectơ sau khi đi qua Transformer encoder sẽ đi đến đầu classification là một khối Multilayer perceptron đưa ra kết quả cuối cùng là xác suất tương ứng với các class. Trong hình 2.1 bên trái là kiến trúc tiền huấn luyện (pre-trained) của Vision Transformer encoder, các vectơ đầu ra của encoder cũng được đưa qua token pooling + projection để tạo ra một image embedding tổng quát cho toàn bộ hình ảnh sau đó kết hợp với phần text encoder để phân loại ảnh. Khác với những thứ trên, OWL-ViT sử dụng để phát hiện đối tượng nên các vectơ sau khi đi qua vision Vision Transformer encoder sẽ được đồng thời đi qua hai đầu MLP và đầu Linear projection. Đầu Linear projection dùng để dự đoán ra nhãn của đối tượng. Đầu MLP (Multilayer perceptron) để dự đoán ra các bounding box của đối tượng.

2.1.4 Hàm mất mát

Mô hình OWL-ViT sử dụng hàm mất mát khớp hai phía (bipartite matching loss) [20] được giới thiệu trong mô hình DETR. Hàm mất mát khớp hai phía thực hiện thực chất là so sánh các lớp và hộp giới hạn được dự đoán của mỗi truy vấn đối tượng N = 100 với các nhãn đúng, được đệm đến cùng độ dài N (vì vậy, nếu một hình ảnh chỉ chứa 4 đối tượng, 96 nhãn sẽ chỉ có "không có đối tượng" làm lớp và "không có hộp giới hạn" làm hộp giới hạn). Thuật toán Hungarian matching được sử dụng để tìm một ánh xạ một-đến-một tối ưu cho mỗi truy vấn N tới mỗi nhãn N. Tiếp theo, ta sử dụng hàm mất mát chéo entropy chuẩn (cho các lớp) và tổ hợp tuyến tính của hàm mất mát L1 và hàm mất mát Generalized IoU [25] (đối với hộp giới hạn) để tối ưu các tham số của mô hình.

Mô hình OWL-ViT điều chỉnh nó cho việc phát hiện đối tượng vựng mở như sau. Do công sức cần thiết để gán nhãn cho các tập dữ liệu phát hiện một cách toàn diện, các tập dữ liệu với số lượng lớn danh mục được gán nhãn theo cách liên minh. Các tập dữ liệu như vậy có không gian nhãn không phân biệt, có nghĩa là mỗi đối tượng có thể có nhiều nhãn. Do đó, mô hình sử dụng hàm mất mát focal sigmoid cross-entropy thay vì softmax cross-entropy làm hàm mất mát phân loại. Hơn nữa, vì không phải tất cả các đối tượng đều được gán nhãn trong mỗi hình ảnh, các tập dữ liệu liên minh cung cấp cả những nhận dạng tích cực (xuất hiện) và tiêu cực (không xuất hiện) của các đối tượng truy vấn cho mỗi hình ảnh. Trong quá trình huấn luyện, đối với một hình ảnh cụ thể, mô hình sử dụng tất cả các nhận dạng tích cực và tiêu cực của nó như các truy vấn. Hơn nữa, mô hình còn ngẫu nhiên chọn mẫu các đối tượng theo tỷ lệ tần suất của chúng trong dữ liệu và thêm chúng như các "giả-tiêu-cực" để có ít nhất 50 tiêu cực cho mỗi hình ảnh.

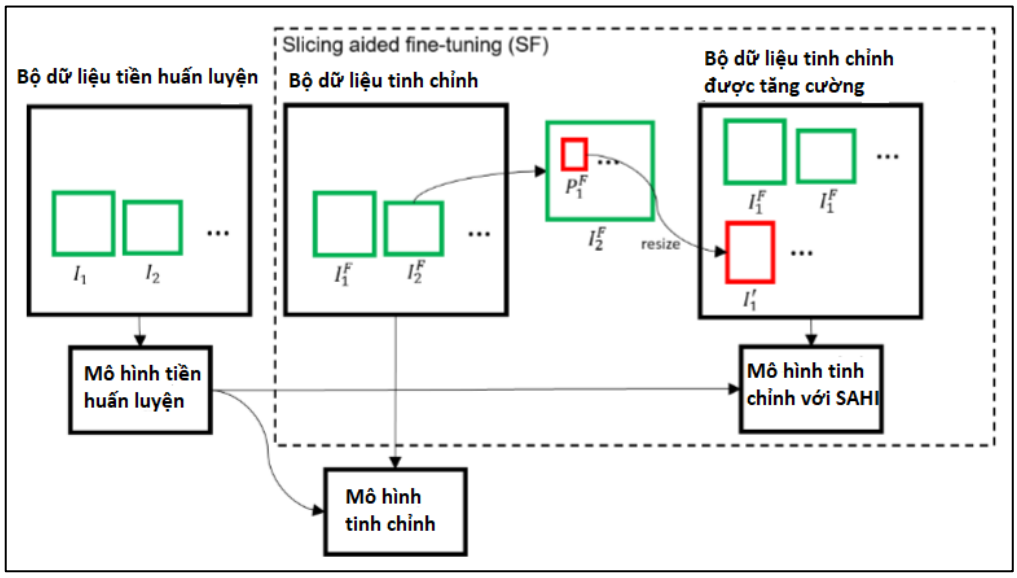
2.2 Kỹ thuật Slicing Aided Hyper Inference (SAHI)

Phát hiện các đối tượng nhỏ và các đối tượng ở xa trong cảnh quan sát là một thách thức lớn trong các ứng dụng giám sát. Những đối tượng như vậy được đại diện bởi một số lượng nhỏ pixel trong hình ảnh và thiếu chi tiết đủ, làm cho việc phát hiện chúng bằng các bộ phát hiện thông thường trở nên khó khăn. Kỹ thuật Slicing Aided Hyper Inference (SAHI) được đề xuất, hỗ trợ các mô hình phát hiện đối tượng nhỏ bằng việc chia nhỏ hình ảnh và fine-tuning. Phương pháp này chia hình ảnh đầu vào thành các phần chồng lấn nhau (overlap patches), gây ra khu vực pixel tương đối lớn hơn cho các đối tượng nhỏ so với hình ảnh được đưa vào mạng.

2.2.1 Phương pháp SAHI cho tinh chỉnh mô hình

Các mô hình phát hiện đối tượng phổ biến thường được cung cấp trọng số được huấn luyện trước trên các tập dữ liệu như ImageNet và MS COCO. Điều này

cho phép chúng ta tinh chỉnh mô hình bằng cách sử dụng các tập dữ liệu nhỏ hơn và trong khoảng thời gian đào tạo ngắn hơn so với việc huấn luyện từ đầu với các tập dữ liệu lớn. Tuy nhiên, các tập dữ liệu thông thường này thường chứa hình ảnh có độ phân giải thấp (640 x 480) với các đối tượng lớn chiếm phần lớn khu vực pixel (trung bình chiếm 60% chiều cao của hình ảnh). Mô hình được huấn luyện trước trên các tập dữ liệu này cho kết quả phát hiện tốt cho các hình ảnh tương tự. Tuy nhiên, hiệu suất phát hiện đối tượng nhỏ trên các hình ảnh có độ phân giải cao được tạo bởi drone và camera giám sát chất lượng cao thường thấp hơn đáng kể.

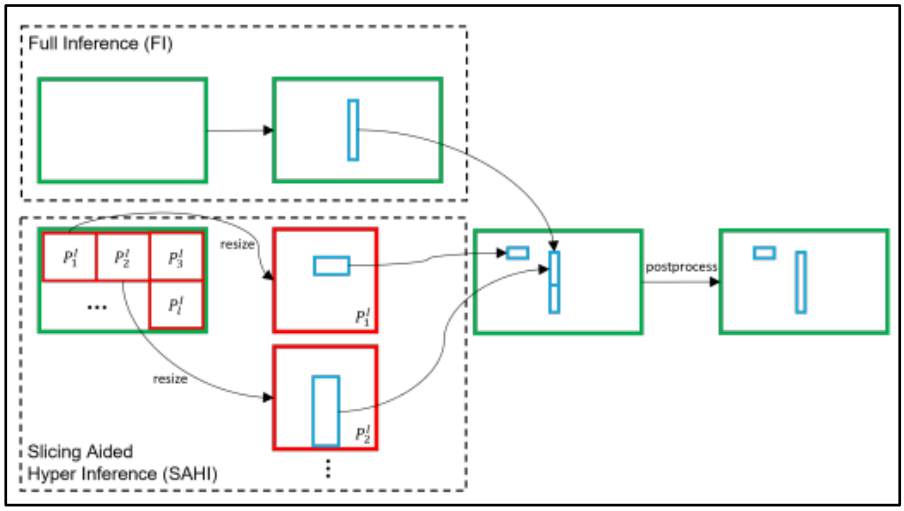


Hình 2.15: Phương pháp SAHI cho tinh chỉnh mô hình (Slicing aided fine-tuning)

Để khắc phục vấn đề trên, SAHI hỗ trợ quá trình tinh chỉnh mô hình phát hiện đối tượng như sau. Đầu tiên, mở rộng tập dữ liệu bằng cách trích xuất các patch từ tập dữ liệu fine-tuning như được thể hiện trong Hình 2.15. Mỗi hình ảnh I_1^F, I_2^F, ..., I_j^F được chia thành các patch chồng lấn nhau P_1^F, P_2^F, ..., P_k^F với kích thước hai chiều M và N được lựa chọn trong khoảng $[M_{min}, M_{max}]$ và $[N_{min}, N_{max}]$ là các siêu tham số. Trong quá trình fine-tuning, các patch được điều chỉnh kích thước để giữ nguyên tỷ lệ khung hình sao cho chiều rộng hình ảnh nằm trong khoảng từ 800 đến 1333 pixel, từ đó tạo ra các hình ảnh mở rộng I'_1, I'_2, ..., I'_k, trong đó kích thước đối tượng tương đối lớn hơn so với hình ảnh gốc. Những hình ảnh này I'_1, I'_2, ..., I'_k, cùng với hình ảnh gốc I_1^F, I_2^F, ..., I_j^F (để hỗ trợ phát hiện các đối tượng lớn), được sử dụng trong quá trình fine-tuning. Bởi vì, khi kích thước mảnh giảm, các đối tượng lớn có thể không vừa vào một mảnh và các khu vực giao nhau, điều này có thể dẫn đến hiệu suất phát hiện kém cho các đối tượng lớn.

2.2.2 Phương pháp SAHI cho suy luận mô hình

Trong giai đoạn suy luận, ảnh truy vấn gốc I được chia thành l phần chồng lên nhau M × N: P_1^I, P_2^I, ... P_l^I. Sau đó, mỗi phần cắt được điều chỉnh kích thước mà vẫn giữ tỷ lệ khung hình ban đầu. Tiếp theo, quá trình phát hiện đối tượng được thực hiện độc lập trên từng phần chồng lên nhau. Một quá trình suy luận đầy đủ (có thể dùng thêm tinh chỉnh với SAHI) sử dụng ảnh gốc có thể được áp dụng để phát hiện các đối tượng lớn hơn. Cuối cùng, kết quả dự đoán của các phần chồng lên nhau và (nếu được sử dụng) kết quả tính chỉnh được hợp nhất trở lại kích thước ban đầu bằng cách sử dụng thuật toán Non-maximum Suppression (NMS). Trong quá trình NMS, các hộp có tỷ lệ giao nhau (Intersection over Union - IoU) [26] cao hơn ngưỡng khớp đã được xác định trước Tm được khớp và đối với mỗi khớp, các kết quả phát hiện có xác suất phát hiện thấp hơn ngưỡng đã xác định T_d sẽ bị loại bỏ.



Hình 2.16: Phương pháp SAHI cho suy luận mô hình (Slicing aided hyper inference)

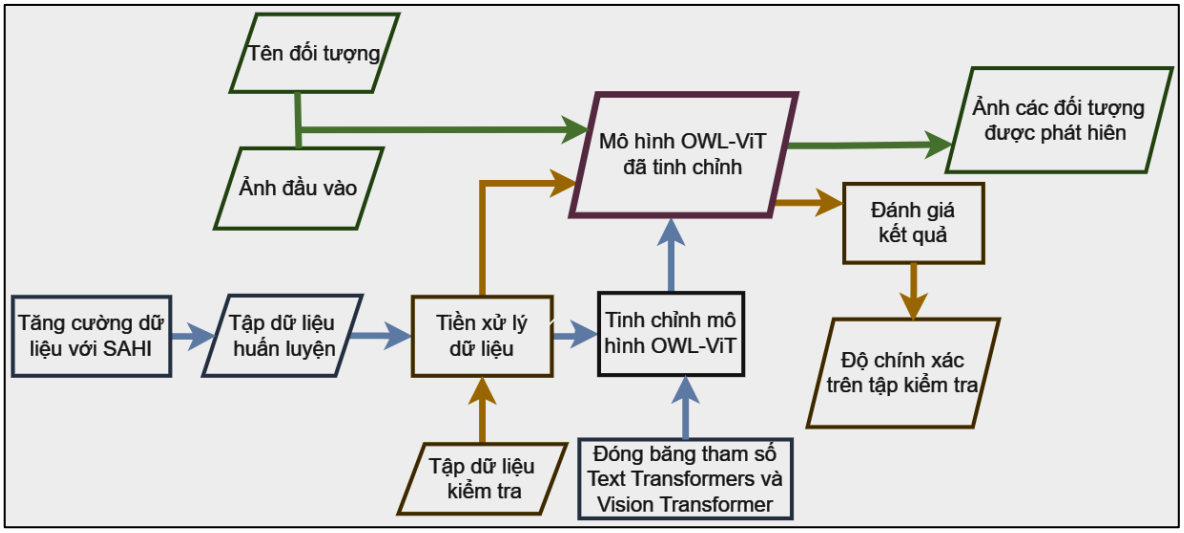
2.3 Phương pháp kết hợp giữa OWL-ViT và SAHI cho phát hiện đối tượng

2.3.1 Tổng quan phương pháp

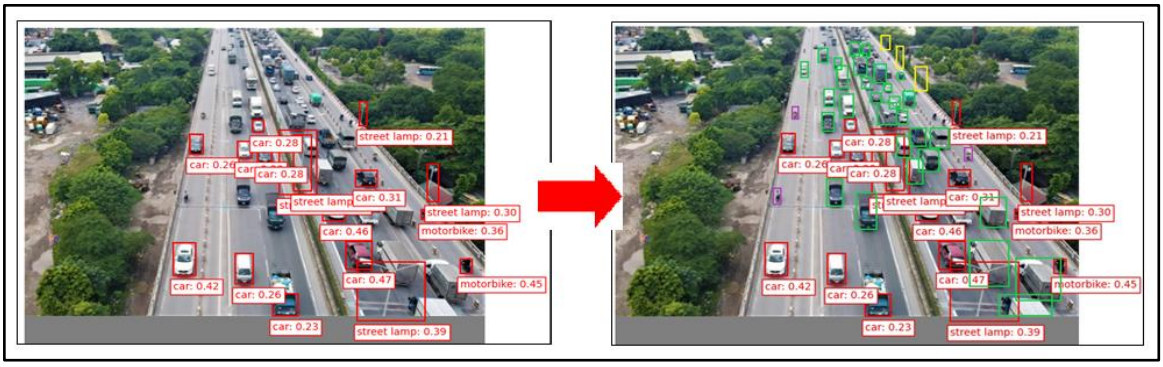
Mô hình OWL-ViT tuy được đào tạo với tập dữ liệu lớn với nhiều loại vật thể nhưng đối với những vật thể có kích thước nhỏ trong ảnh mô hình phát hiện vẫn còn kém. Điều này có thể do nhiều nguyên nhân như đã nêu ở chương I, tuy nhiên nguyên nhân chủ yếu là do mô hình chưa được đào tạo nhiều với vật thể nhỏ như trong tập dữ liệu Visdrone2019. Vậy nên đề án sử dụng kỹ thuật SAHI để tăng cường dữ liệu cho bộ dữ liệu đào tạo. Tận dụng sức mạnh đã được đào tạo với một nguồn dữ liệu cực lớn từ OWL-ViT, tiến hành tinh chỉnh mô hình này với bộ dữ liệu đã được tăng cường. Trong quá trình tinh chỉnh này có thay đổi hàm mất mát để đạt kết quả tốt hơn.

Sau khi mô hình được tinh chỉnh, đánh giá mô hình trên tập kiểm tra của bộ dữ liệu Visdrone2019 với độ đo mAP. Mô hình này cũng nhận đầu vào là ảnh và tên đối tượng dưới dạng văn bản, cho ra kết quả suy luận là ảnh đầu ra với xác suất dự đoán có đối tượng, tên đối tượng và bounding box tương ứng. Bounding box có dạng (cx, cy, w, h) với (cx, cy) là tọa độ tâm và (w, h) là chiều rộng và chiều cao của bounding box.

Từng bước của phương pháp phát hiện đối tượng kết hợp OWL-ViT và SAHI được miêu tả chi tiết trong sơ đồ hình 2.17 dưới đây:



Hình 2.17: Phương pháp phát hiện đối tượng kết hợp OWL-ViT và SAHI

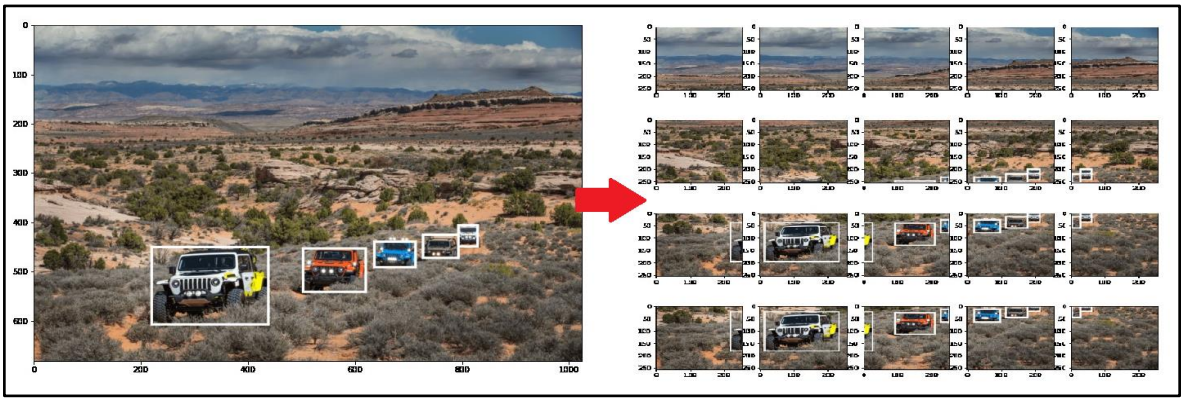


Hình 2.18: Minh họa phương pháp giúp phát hiện vật thể nhỏ tốt hơn

2.3.2 Xử lý trong tinh chỉnh mô hình

Trong phần tinh chỉnh mô hình, bước đầu tiên đó là chuẩn bị dữ liệu. Đối với phần chuẩn bị dữ liệu, đề án sử dụng kỹ thuật SAHI để tăng cường dữ liệu. SAHI cắt những ảnh trong tập đào tạo của tập Visdrone2019 ra thành các phần nhỏ hơn. Trộn lẫn chúng với tập đào tạo ban đầu, ta được bộ dữ liệu đào tạo mới với kích thước ảnh đa dạng hơn. Điều này sẽ giúp cho những vật thể nhỏ của bộ dữ liệu được đào tạo

nhiều hơn với các ảnh kích thước khác nhau. Như trong hình 2.19, có thể thấy rằng ảnh gốc ban đầu được chia thành các ảnh nhỏ, và giữ nguyên được nhãn ban đầu của vật thể. Những vật thể nhỏ trong những ảnh mới này được phóng to ra so với kích cỡ của bức ảnh mới. Nhờ vậy khi cho vào đào tạo, mô hình sẽ học được những đặc rõ ràng của các đối tượng cần phát hiện trong bức ảnh. Và việc gia tăng số lượng ảnh sẽ giúp mô hình được học đi học lại các đối tượng, giúp phát hiện đối tượng tốt hơn. Trong tập dữ liệu Visdrone2019 các ảnh có kích thước lớn, đồng thời độ phân giải cao nên việc cắt nhỏ ảnh hoàn toàn phù hợp vì không làm ảnh hưởng đến chất lượng ảnh. Sau khi cắt ra bằng SAHI, các bức hình nhỏ không hề bị mờ và đảm bảo chất lượng cho mô hình khi đào tạo.



Hình 2.19: Dùng SAHI để tăng cường dữ liệu

Sau khi xử lý xong dữ liệu, tiếp đến là tinh chỉnh mô hình OWL-ViT. Vì OWL-ViT là một mô hình có kích cỡ mạng rất lớn, để đào tạo cả mô hình cần một nguồn tài nguyên lớn cả về thiết bị máy móc lẫn dữ liệu. Vậy nên đề án thực hiện tinh chỉnh mô hình với tập dữ liệu đã tăng cường bằng SAHI ở trên. Tận dụng sức mạnh của mô hình đã được đào tạo với các tập dữ liệu lớn, ta đóng băng phần Text Encoder và Vision Encoder, tiến hành tinh chỉnh trên phần còn lại của mạng, bao gồm hai phần quan trọng là đầu dự đoán bounding box và dự đoán tên lớp cho đối tượng. Trong quá trình tinh chỉnh sử dụng hàm mất mát theo OWL-ViT Adaptation. Thuật toán tối ưu tham số được sử dụng là AdamW [18], một phiên bản cải tiến hơn so với Adam [19].

2.3.3 Thuật toán tối ưu

Thuật toán tối ưu là cơ sở để xây dựng mô hình neural network với mục đích "học " được các đặc trưng của dữ liệu đầu vào, từ đó có thể tìm được cặp trọng số và độ lệch (weights-bias) phù hợp để tối ưu hóa mô hình. Thuật toán tối ưu cơ bản nhất đó chính là gradient descent [20]. Qua nhiều sự nghiên cứu, từng thuật toán tối ưu khác ra đời, khắc phục những vấn đề còn tồn tại đã lâu. SGD[21] hiệu quả hơn gradient descent khi giải các bài toán tối ưu, ví dụ, nó chịu ít ảnh hưởng xấu gây ra

bởi dữ liệu dư thừa. Minibatch SGD mang lại hiệu quả đáng kể nhờ việc vector hóa, tức xử lý nhiều mẫu quan sát hơn trong một minibatch. Đây là chìa khóa để xử lý dữ liệu song song trên nhiều GPU và nhiều máy tính một cách hiệu quả. Phương pháp (động lượng) (momentum) [22] bổ sung cơ chế gộp các gradient quá khứ, giúp quá trình hội tụ diễn ra nhanh hơn. Adagrad [23] sử dụng phép biến đổi tỉ lệ theo từng tọa độ để tạo ra tiền điều kiện hiệu quả về mặt tính toán. RMSprop tách rời phép biến đổi tỉ lệ theo từng tọa độ khỏi phép điều chỉnh tốc độ học.

Thuật toán Adam kết hợp tất cả các kỹ thuật đã nói ở phần trên thành một thuật toán học hiệu quả. Dựa trên RMSProp, Adam cũng sử dụng trung bình động trọng số mũ cho gradient $\boldsymbol{g}_t$ ngẫu nhiên theo minibatch như dưới đây:

$$\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-2} + (1-\beta_1)\boldsymbol{g}_t \tag{2.8}$$

$$\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2 \tag{2.9}$$

Nếu khởi tạo $m_o = v_o = 0$, thuật toán sẽ có độ chệch ban đầu đáng kể về các giá trị nhỏ hơn và Adam chuẩn hóa lại chúng như sau:

$$\widehat{\boldsymbol{m}_t} = \frac{\boldsymbol{m}_t}{1-\beta_1^t} \tag{2.10}$$

$$\widehat{\boldsymbol{v}_t} = \frac{\boldsymbol{v}_t}{1-\beta_2^t} \tag{2.11}$$

Điều chỉnh lại giá trị gradient, tương tự như ở RMSProp, tuy nhiên ở đây Adam sử dụng phép hiệu chỉnh độ chệch (bias correction), cộng thêm một hằng số nhỏ để điều chỉnh cho trường hợp khởi động chậm khi ước lượng động lượng và mô-men bậc hai:

$$\boldsymbol{g}'_t = \frac{\eta \widehat{\mathbf{m}}_t}{\sqrt{\widehat{\mathbf{v}}_t} + \varepsilon} \tag{2.12}$$

Đối với gradient có phương sai đáng kể, chúng ta có thể gặp phải những vấn đề liên quan tới hội tụ. Những vấn đề này có thể được khắc phục bằng cách sử dụng các minibatch có kích thước lớn hơn. Với sự hiệu quả trên, Adam đã trở thành phương pháp mặc định được chọn để huấn luyện các mạng nơ-ron.

Phổ biến và thường được sử dụng là vậy, tuy nhiên khi được kiểm tra trên một loạt các nhiệm vụ học sâu khác nhau như phân loại hình ảnh, mô hình ngôn ngữ cấp ký tự và phân tích cú pháp, Adam không có hiệu suất tổng quát tốt như SGD với động lượng (SGD with momentum). Sự khác biệt của Adam nằm ở cách thức triển khai không hiệu quả của trọng số tiêu biến (weight decay). Để giải quyết vấn đề này, chúng ta có một thuật toán khác đó là AdamW. AdamW sửa đổi đơn giản một chút so với Adam bằng cách tách riêng trọng số tiêu biến (weight decay) khỏi các bước tối ưu hóa được thực hiện trong hàm mất mát.

Weight decay là giảm giá trị trọng số theo hàm mũ với hệ số giảm trọng số θ:

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha\nabla f_t(\theta_t) \quad (2.13)$$

Ở đây, λ xác định tốc độ giảm trọng số theo mỗi bước và $\nabla f_t(\theta_t)$ là độ dốc của batch thứ t sẽ được nhân với tốc độ học α. Đối với Stochastic gradient descent weight decay tương đương với L_2 regularization. Tuy nhiên điều này ngược lại với các thuật toán adaptive gradient. Việc sử dụng L_2 regularization trong Adam giống như Stochastic gradient descent đã cho các kết quả không tốt khi thực nghiệm.

Điểm khác biệt lớn nhất của AdamW và Adam được thể hiện rõ ràng qua phần weight decay trong hình so sánh hai thuật toán ở dưới đây:

Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

1: **given** $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$
2: **initialize** time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second moment vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$
3: **repeat**
4: $t \leftarrow t + 1$
5: $\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$ ▷ select batch and return the corresponding gradient
6: $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) + \lambda\boldsymbol{\theta}_{t-1}$
7: $\boldsymbol{m}_t \leftarrow \beta_1\boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$ ▷ here and below all operations are element-wise
8: $\boldsymbol{v}_t \leftarrow \beta_2\boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
9: $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t/(1 - \beta_1^t)$ ▷ β_1 is taken to the power of t
10: $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t/(1 - \beta_2^t)$ ▷ β_2 is taken to the power of t
11: $\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$ ▷ can be fixed, decay, or also be used for warm restarts
12: $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left(\alpha\hat{\boldsymbol{m}}_t/(\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) + \lambda\boldsymbol{\theta}_{t-1}\right)$
13: **until** *stopping criterion is met*
14: **return** optimized parameters $\boldsymbol{\theta}_t$

Hình 2.20: Mã giả thuật toán tối ưu của Adam và AdamW

Nhìn vào mã giả thuật toán ở trên, khi Adam chạy trên hàm mất mát f, và cộng với regularization, những trọng số có xu hướng có gradient lớn trong f sẽ không được điều chỉnh nhiều. Bởi vì trong L_2 regularization, cả hai gradient của hai số hạng (hàm mất mát và regularization) đều được điều chỉnh theo độ lớn của chúng. Do đó, trọng số x với độ lớn gradient lớn s được điều chỉnh theo một lượng tương đối nhỏ hơn so với các trọng số khác.

Ngược lại, việc tách riêng weight decay của AdamW sẽ điều chỉnh tất cả các trọng số với cùng tốc độ λ, điều chỉnh hiệu quả các trọng số x với s lớn hơn so với chuẩn hóa L_2 tiêu chuẩn. Như trong hình trên, AdamW tách riêng weight decay khỏi bước tối ưu hóa được thực hiện trong hàm mất mát (dòng 6) bằng cách chuyển nó xuống phần cập nhật trọng số (dòng 12).

AdamW đem lại hiệu suất tổng quát tốt hơn đáng kể với việc tách riêng weight decay so với Adam L_2 regularization. Điều này đã được chứng minh qua thực nghiệm, việc tách riêng weight decay làm cho các cài đặt tối ưu của tốc độ học tập và weight decay độc lập hơn nhiều, nhờ đó giảm bớt tối ưu hóa siêu tham số.

2.3.4 Hàm mất mát

Hàm mất mát (loss function) là một hàm số được sử dụng để đo lường mức độ sai lệch giữa đầu ra dự đoán của mô hình và giá trị thực tế (ground truth). Mục tiêu của hàm mất mát là tối thiểu hóa sai lệch này, từ đó cung cấp một đánh giá về hiệu suất của mô hình.

Hàm mất mát được sử dụng cho OWL-ViT trong đề án như sau:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{Unmatched}} + \mathcal{L}_{\text{GIoU}} + \mathcal{L}_{\text{Box}} \tag{2.14}$$

Trong đó $\mathcal{L}_{\text{class}}$ là hàm mất mát phân loại tương phản (contrast classification loss) được tính dựa trên những truy vấn đúng (positive queries) và truy vấn sai (negative queries) cho từng bounding box được dự đoán trong ảnh với p_{pos}, p_{neg} là xác suất của bounding box cho truy vấn đúng và truy vấn sai.

$$\mathcal{L}_{\text{class}} = -\left[\log(p_{pos}) + \frac{1}{n_{neg}} \sum_{n_{neg}} \log(1 - p_{neg})\right] \tag{2.15}$$

Ngoài ra cũng có một hàm mất mát $\mathcal{L}_{\text{Unmatched}}$ để loại bỏ các bounding box giả được phát hiện với xác suất cao cho truy vấn đúng. Nghĩa là bounding box được dự đoán đúng nhãn phân loại nhưng kích thước bounding box này không khớp với ground truth box. Số bounding box không khớp được đặt là U. Công thức của hàm $\mathcal{L}_{\text{Unmatched}}$ như sau:

$$\mathcal{L}_{\text{Unmatched}} = -\frac{1}{U} \sum_{u} \log(1 - p_{pos}) \tag{2.16}$$

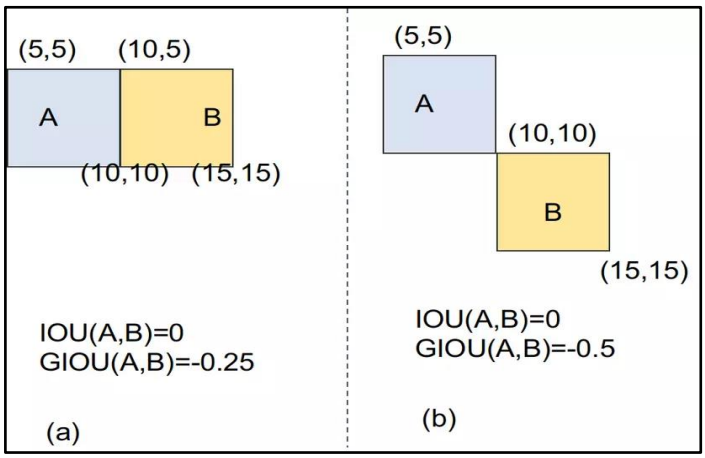
Hàm mất mát của mô hình sử dụng cho bounding box là hàm mất mát L1 (L1 loss) và hàm mất mát GIoU (GIoU loss) cho phần hàm mất mát định vị vật thể (localization loss). Hàm mất mát L1 còn được gọi là tổn thất tuyệt đối trung bình (mean absolute loss) hoặc sai số tuyệt đối trung bình (mean absolute error). Nó chỉ đơn giản là tổng của sự khác biệt tuyệt đối giữa giá trị thực tế y_{true} và giá trị dự đoán $y_{predict}$. Công thức của L1 loss như sau:

$$\mathcal{L}_{L1} = \sum_{i=1}^{n} |y_{true} - y_{predict}| \tag{2.17}$$

Khi thực hiện phát hiện đối tượng, mô hình sẽ dự đoán các thông số của bounding box như tọa độ (cx, cy) của tâm hộp, chiều rộng và chiều cao. Để đo lường sự sai khác giữa các bounding box dự đoán và thực tế, L1 loss được tính toán bằng cách tính tổng giá trị tuyệt đối của hiệu giữa các thông số của hai bounding box đó. Cụ thể, để tính L1 loss cho phát hiện đối tượng, ta lấy tổng giá trị tuyệt đối của hiệu giữa các thông số của bounding box dự đoán và bounding box thực tế cho tất cả các đối tượng trong ảnh. Sau đó, L1 loss được tính toán bằng cách lấy trung bình của các giá trị

tuyệt đối này. L1 loss giúp đo lường mức độ sai khác giữa các bounding box dự đoán và các bounding box thực tế. Nếu L1 loss càng nhỏ, tức là các bounding box dự đoán và thực tế càng gần nhau, thì mô hình phát càng chính xác.

GIoU (Generalized IoU) là một phiên bản cải tiến của IoU trong việc đánh giá độ tương đồng giữa hai bounding box trong phát hiện đối tượng. GIoU được đưa ra để giải quyết nhược điểm của IoU khi không có sự giao nhau giữa các hộp giới hạn. Tại những lần dự đoán lúc đầu, khi mô hình chưa được "học nhiều", các box không hề giao nhau là chuyện hoàn toàn có thể xảy ra. Khi sử dụng IoU, việc không có sự giao nhau giữa các hộp giới hạn làm cho tỷ lệ chồng lấn (overlay) luôn bằng 0, và do đó không thể phân biệt được hộp nào tốt hơn. Ví dụ như ở hình 2.21 dưới đây:

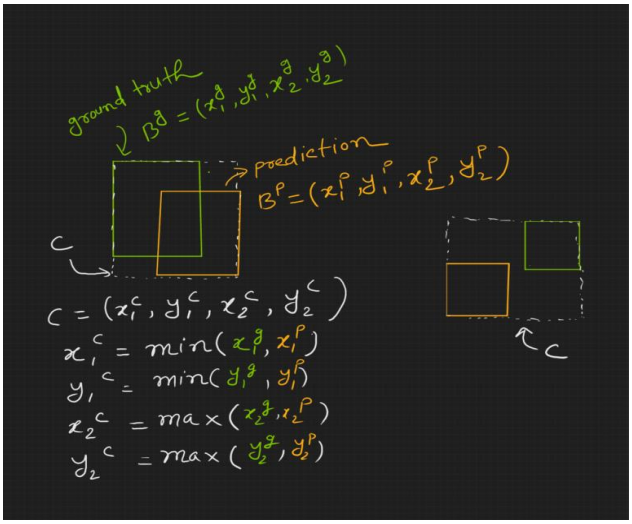


Hình 2.21: So sánh giữ IoU và GIoU

Để giải quyết vấn đề này, GIoU tính toán không chỉ tỷ lệ chồng lấn mà còn đánh giá sự gần nhau của hai bounding box. Nó sẽ tính toán tỷ lệ chồng lấn (overlap ratio) cùng với một thành phần khoảng cách (distance term) giữa hai hộp giới hạn. Kết quả là, GIoU có thể phân biệt được giữa các hộp không giao nhau và cho kết quả tốt hơn so với IoU trong các trường hợp này. Công thức tính của GIoU như sau:

$$\mathrm{GIoU} = \mathrm{IoU} - \frac{\mathrm{C}(\mathrm{A} \cup \mathrm{B})}{\mathrm{C}} \qquad (2.18)$$

Với $\mathrm{IoU} = \frac{\mathrm{A} \cap \mathrm{B}}{\mathrm{A} \cup \mathrm{B}}$ và C là hình chữ nhật nhỏ nhất bao quanh hai bounding box A và B. Ta có diện tích hình chữ nhật C được tính như hình dưới đây:



Hình 2.22: Hình chữ nhật C bao quanh hai bounding box A va B trong GIoU

Nhìn vào công thức ở trên, ta có IoU có giá trị [0,1] nên suy ra giá trị GIoU sẽ nằm trong (-1, 1). Công thức của hàm mất mát GIoU sẽ như sau:

$$\mathcal{L}_{\text{GIoU}} = 1 - \text{GIoU} = 1 - \text{IoU} + \frac{C \setminus (A \cup B)}{C} \tag{2.19}$$

2.4 Kết luận chương

Trong chương 2, đề án trình bày từ tổng quan đến chi tiết các nghiên cứu liên quan của phương pháp. Đầu tiên là trình bày về mô hình phát hiện đối tượng từ vựng mở OWL-ViT với các phần Text Encoder, Vision Encoder và hàm mất mát của mô hình. Tiếp đến là mô tả chi tiết về kỹ thuật SAHI với hai phần sử dụng trong tinh chỉnh và sử dụng trong suy luận. Đề án cũng trình bày chi tiết ý tưởng kết hợp hai phần lại và đưa ra giải pháp cho vấn đề phát hiện đối tượng kích thước nhỏ trong ảnh chụp từ Drone. Phương pháp sử dụng SAHI để tăng cường dữ liệu cho tập dữ liệu đào tạo. Sau đó tinh chỉnh mô hình OWL-ViT với tập dữ liệu đào tạo đã được tăng cường. Đề án mô tả thuật toán tối ưu AdamW được cải tiến từ Adam và hàm mất mát sử dụng trong quá trình tinh chỉnh. Tiếp đến chương 3 viết về việc triển khai và đánh giá hiệu quả của phương pháp.

Chương 3 - THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1 Mô tả các bộ dữ liệu

3.1.1 Nguồn dữ liệu

Các thiết bị bay không người lái (UAVs), hay còn gọi là drone, được trang bị máy ảnh và đã được triển khai nhanh chóng trong một loạt các ứng dụng, bao gồm nông nghiệp, nhiếp ảnh từ không gian, giao hàng nhanh và giám sát. Do đó, việc hiểu tự động dữ liệu hình ảnh thu thập từ các nền tảng này trở nên rất đòi hỏi, đồng thời đưa thị giác máy tính gần gũi hơn với các drone. Vậy nên một bộ dữ liệu chuẩn đoán quy mô lớn với đánh dấu chính xác cho các nhiệm vụ quan trọng của thị giác máy tính, mang tên đã ra đời, nhằm kết hợp thị giác và drone lại với nhau. Bộ dữ liệu VisDrone2019 được thu thập bởi nhóm AISKYEYE tại Laboratoire of Machine Learning and Data Mining, Đại học Thiên Tân, Trung Quốc. Bộ dữ liệu gồm 288 đoạn video được tạo thành từ 261,908 khung hình và 10,209 ảnh tĩnh, được chụp bởi các máy ảnh gắn trên drone khác nhau, bao gồm nhiều khía cạnh khác nhau bao gồm vị trí (chụp từ 14 thành phố khác nhau cách nhau hàng nghìn km tại Trung Quốc), môi trường (đô thị và nông thôn), đối tượng (người đi bộ, xe cộ, xe đạp, v.v.), và mật độ (cảnh vắng và tắc nghẽn). Lưu ý rằng, dữ liệu được thu thập bằng cách sử dụng các nền tảng drone khác nhau (tức là drone với các mô hình khác nhau), trong các tình huống khác nhau và dưới các điều kiện thời tiết và ánh sáng khác nhau. Những khung hình này được gán nhãn thủ công với hơn 2.6 triệu bounding box đối tượng quan tâm thường xuyên, chẳng hạn như người đi bộ, xe ô tô, xe đạp và xe ba bánh. Một số thuộc tính quan trọng bao gồm khả năng hiển thị cảnh, lớp đối tượng và che khuất cũng được cung cấp để tận dụng dữ liệu tốt hơn.

3.1.2 Chi tiết dữ liệu

Bộ dữ liệu được sử dụng trong đề án là bộ dữ liệu ảnh tĩnh từ chụp từ drone sử dụng để phát hiện đối tượng VisDrone2019-Detection, gồm các hình ảnh chụp từ trên cao, góc rộng với rất nhiều đối tượng kích thước nhỏ trong ảnh. Bao gồm 10,209 ảnh 3 tập đào tạo-xác thực-kiểm thử được với số lượng ảnh lần lượt là 6471, 548, 1610. Số lượng đối tượng là 2,6 triệu đối tượng bao gồm các loại sau: người đi bộ, đám đông người, xe đạp, ô tô, xe bán tải, xe tải hạng nặng, xe ba bánh, xe ba gác, xe bus. Chi tiết mô tả bộ dữ liệu ở bảng 3.1 dưới đây:

Bảng 3.1: Chi tiết bộ dữ liệu Visdrone2019-Detection

STT	Thuộc tính	Mô tả
1	Số lượng ảnh	10,209
2	Số lượng đối tượng	2.6 triệu

3	Các loại đối tượng	Người đi bộ, đám đông người, xe đạp, ô tô, xe bán tải, xe tải hạng nặng, xe ba bánh, xe ba gác, xe bus.
4	Dung lượng	1.81 GB
5	Loại ảnh	Ảnh màu RGB
6	Kích thước	Các ảnh có chiều dài >1000 pixel, chiều rộng >700 pixel
7	Nhãn	Mỗi ảnh được gán với một tệp nhãn .txt tương ứng gồm các đối tượng và bounding box trên từng dòng
8	Tỷ lệ phân chia tập huấn luyện-xác thực- đào tạo	Tỷ lệ tương ứng 63.38%, 5.37%, 31,25%
9	Đường liên kết tải	https://github.com/VisDrone/VisDrone-Dataset

Bộ dữ liệu gồm 10 lớp đối tượng, chi tiết số lượng từng loại đối tượng trong tập dữ liệu như bảng 3.2.

Bảng 3.2: Chi tiết số lượng từng loại đối tượng trong tập dữ liệu

	Tập đào tạo	**Tập xác thực**	**Tập kiểm tra**
Car	144.866	14.064	28.074
Pedestrian	79.337	8.844	21.006
People	27.059	5.125	6.376
Motor	29.647	4.886	5.845
Van	24.956	1.975	5.771
Bicycle	12.875	1.287	1.302
Tricycle	4.812	1.045	530
Truck	12.875	750	2.659
awning-tricycle	3.246	532	599
Bus	5.926	251	2.940

Hình 3.1: Một ảnh trong bộ dữ liệu chụp từ drone

Nhìn vào bức ảnh trên, có thể thấy rằng các đối tượng chụp từ drone trong bộ dữ liệu VisDrone2019-Detection khá nhỏ. Với một số ảnh, nhất là đối với các đối tượng ở góc xa, vừa nhỏ lại còn gần nhau, mặt độ dày đặc.

3.2 Quá trình tinh chỉnh mô hình OWL-ViT kết hợp SAHI

3.2.1 Thông số, độ đo

Đề án sử dụng độ đo trong phát hiện đối tượng là mAP viết tắt của mean Average Precision (độ chính xác trung bình). Một AP (Average Precision) được tính toán cho mỗi lớp đối tượng riêng lẻ trong bài toán object detection. mAP là giá trị trung bình của tất cả các AP này. Để tính toán mAP, các bước sau được thực hiện:

1. Tính toán AP cho mỗi lớp đối tượng: Đầu tiên, tính toán giá trị Precision và Recall cho mỗi ngưỡng ngưỡng xác suất (threshold) để xác định bounding box dự đoán là chính xác hay không. Trong bài toán phát hiện đối tượng thì cách tính Precision và Recall cũng tương tự như bài toán phân loại nhưng cách định nghĩa True Positive, True Negative, False Positive, False Negative có sự khác biệt, nó sẽ sử dụng một thông số khác là IoU (intersection over union) nó đơn giản là tỉ số của diện tích trùng lặp của 2 bounding box trên diện tích hợp của 2 bounding box. Khi tính các giá trị True Positive, False Positive của Precision và Recall, ta có sử dụng một thông số là ngưỡng IoU (IoU threshold). Tùy thuộc vào tập dataset mà ta đánh giá nên đặt ngưỡng cho phù hợp. Vậy nên AP có thể được kí hiệu là AP@α với α là giá trị IoU threshold. Dựa vào thông số IoU này người ta sẽ xác định True Positive, True Negative, False Positive, False Negative như sau:

- True Positive (TP): Khi IoU của hộp dự đoán vật thể (predict box) và hộp nhãn thật (ground truth) lớn hơn hoặc bằng ngưỡng IoU dùng để đánh giá.

- False Positive (FP): Khi IoU của hộp dự đoán vật thể (predict box) và hộp nhãn thật (ground truth) nhỏ hơn ngưỡng IoU dùng để đánh giá.
- True Negative (TN): Thông số này có thể hiểu nó như là background và sẽ không cần quan tâm thông số này.
- False Negative (FN): Bounding box của đối tượng không được phát hiện (phát hiện sót)

Sau đó, xây dựng đường cong Precision-Recall bằng cách sắp xếp các cặp (Precision, Recall) theo thứ tự giảm dần của Recall. Cuối cùng, tính toán diện tích dưới đường cong Precision-Recall, tức là AP cho lớp đối tượng đó.

2. Tính toán mAP: Sau khi tính toán AP cho mỗi lớp đối tượng, mAP được tính bằng cách lấy giá trị trung bình của tất cả các AP này. Điều này đảm bảo rằng sự đánh giá hiệu suất của mô hình phát hiện đối tượng không chỉ dựa trên một lớp đối tượng cụ thể, mà là sự kết hợp của hiệu suất trên tất cả các lớp đối tượng có trong bộ dữ liệu. Công thức tính mAP cho n lớp đối tượng trong tập dữ liệu như sau:

$$\mathrm{mAP} = \frac{1}{n}\sum_{n=1}^{n} \mathrm{AP_i} \tag{3.1}$$

mAP là một phép đo quan trọng để đánh giá tổng thể hiệu suất của mô hình object detection. Nó là một chỉ số phổ biến và được sử dụng rộng rãi trong cả nghiên cứu và ứng dụng thực tế.

3.2.2 Tăng cường dữ liệu với SAHI

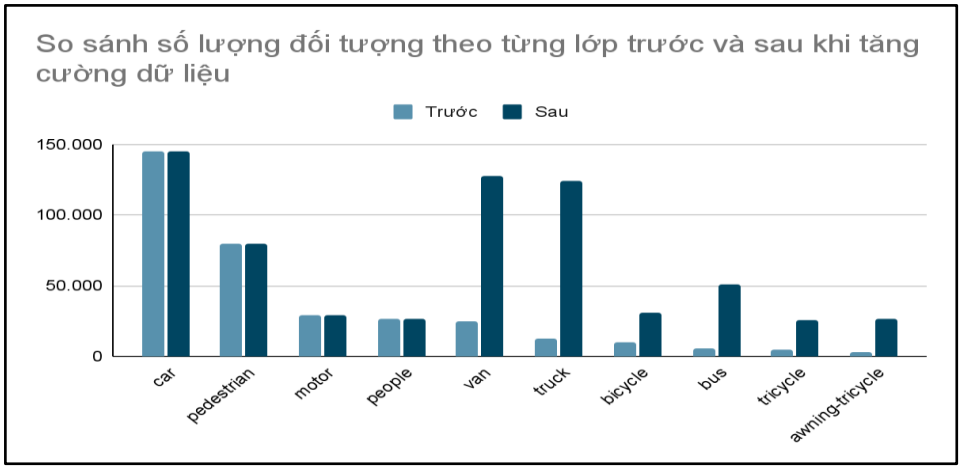
Ở chương II, đề án đã việc tăng cường dữ liệu đào tạo cho bộ dữ liệu bằng kỹ thuật SAHI. Tại phần thực nghiệm này, ta chia các ảnh trong tập đào tạo thành các ảnh có kích thước nhỏ hơn (ảnh slice) là 840x840, 960x960 với tỷ lệ chồng lấn là 0 và 0.25. Từ đó ta được 4 tập dữ liệu nhỏ hơn là 840x840 tỷ lệ chồng lấn 1, 840x840 tỷ lệ chồng lấn 0.25, 960x960 tỷ lệ chồng lấn 0 và 960x960 tỷ lệ chồng lấn 0.25. Nhìn vào số lượng đối tượng từng lớp trong tập đào tạo được mô tả tại bảng 2 ở trên, có thể thấy rằng số lượng đối tượng trong các lớp không được cân bằng, điều này dẫn tới việc mất cân bằng trong phát hiện đối tượng, nghĩa là mô hình sẽ dự đoán nghiêng về lớp có số lượng đối tượng được học nhiều hơn, vì nó có nhiều hơn trong tập đào tạo. Để giải quyết vấn đề đề này ta áp dụng SAHI để tăng cường dữ liệu. Sau khi chia ảnh trong tập đào tạo thành các ảnh có kích thước nhỏ hơn, ta tăng cường những lớp ít dữ liệu bằng cách lấy thêm các đối tượng trong ảnh slice trong quá trình tinh chỉnh. Như trong bảng 3, lớp car (ô tô) trong tập đào tạo ban đầu đã đủ nhiều nên không lấy thêm đối tượng để tăng cường ảnh. Và lớp van hay truck cần bổ sung thêm ảnh để cân bằng hơn với lớp car, vì các loại ô tô dễ bị nhầm với nhau (xe truck hay bị nhầm phần đầu xe thành car). Tương tự với lớp motor, bicycle, tricycle, awning-tricycle, các dòng xe mô tô, xe ba bánh, xe đạp rất dễ nhầm lẫn nên tăng cường chúng lên số

lượng đều bằng nhau. Bảng 3.3 dưới đây mô tả chi tiết số lượng các đối tượng của các lớp tăng trong việc tăng cường dữ liệu, những số in đậm là những số được dùng để cho vào tập đào tạo mới.

Bảng 3.3: Mô tả chi tiết về số lượng đối tượng được tăng cường theo lớp

Lớp	**Tập đào tạo ban đầu**	**Slice 840-0**	**Slice 840-025**	**Slice 960-0**	**Slice 960-025**	**Tập đào tạo mới**
car (3)	**144.866**	280.028	299.263	328.313	359.478	***144.866***
pedestrian (0)	**79.337**	145.894	154.630	167.470	179.868	***79.337***
motor (9)	**29.647**	59.054	62.814	65.710	75.213	***29.647***
people (1)	**27.059**	52.687	57.093	57.047	66.631	***27.059***
van (4)	**24.956**	**45.081**	48.579	52.629	**57.374**	***127.411***
truck (5)	**12.875**	**25.088**	**26.231**	**28.988**	**31.362**	***124.544***
bicycle (2)	**10.480**	**20.641**	23.538	23.077	26.387	***31.121***
bus (8)	**5.926**	**10.087**	**10.335**	**11.750**	**12.538**	***50.656***
tricycle (6)	**4.812**	**9.612**	10.331	9.290	**11.597**	***25.921***
awning-tricycle (7)	**3.246**	**6.965**	**7.470**	6.997	**9.259**	***26.940***

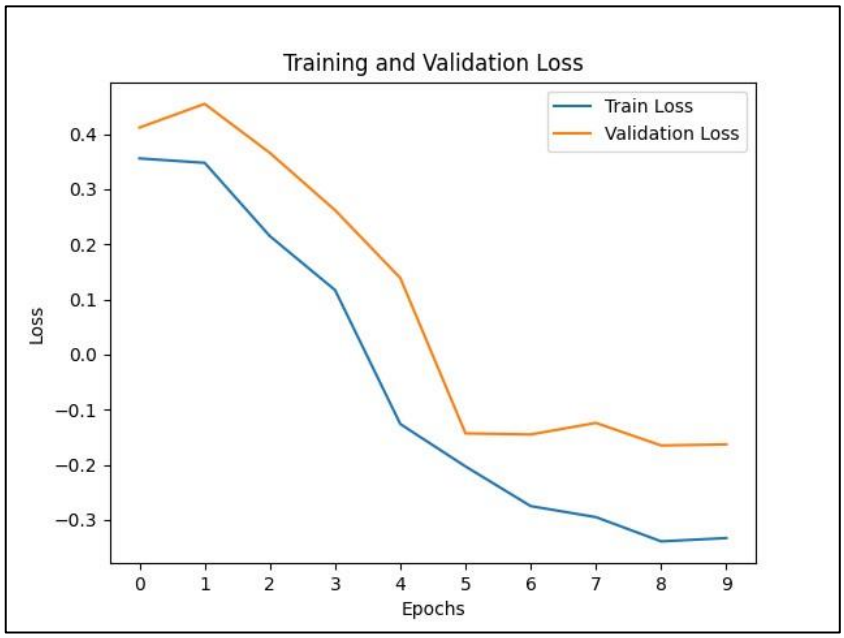
Biểu đồ tại hình 3.2 dưới đây là mô tả trực quan sự tăng cường dữ liệu đối với các lớp. Nhìn vào biểu đồ có thể thấy ngay được các lớp car, predestrian, motor, people được giữ nguyên vì chúng đã có nhiều trong tập đào tạo. Lớp bus và lớp truck được tăng cường lên nhiều nhất vì chúng có rất ít trong tập đào tạo và lại bị lớp hay nhầm là car áp đảo về số lượng. Sau khi tăng cường có những lớp được tăng số lượng đối tượng lên hơn 10 lần.



Hình 3.2: Biểu đồ số lượng đối tượng được tăng cường theo các lớp bằng SAHI

3.2.3 Tinh chỉnh mô hình với bộ dữ liệu Visdrone

Sau khi tăng cường dữ liệu của tập đào tạo ở trên bằng SAHI, tinh chỉnh mô hình OwL-ViT với 10 epochs, tốc độ học là 3e-7. Mô hình được đánh giá độ mất mát trên tập val của VisDrone2019-Detection, kết quả đào tạo mô hình được trình bày ở hình 3.3 và bảng 3.4 dưới đây.



Hình 3.3: Kết quả train loss và val loss trong quá trình huấn luyện

Bảng 3.4: Kết quả train loss và val loss trong quá trình huấn luyện

Epoch	**Train loss**	**Val loss**
0	0.356	0.412
1	0.348	0.455

2	0.215	0.366
3	0.117	0.262
4	-0.126	0.139
5	-0.203	-0.143
6	-0.275	-0.145
7	-0.295	-0.124
8	-0.339	-0.165
9	-0.333	-0.163

Nhận xét:

-Giá trị giữa train loss và val loss đều có xu hướng giảm dần trong quá trình huấn luyện. Từ những epoch 7 về sau, val loss và train loss không giảm nhiều.

-Mô hình đạt kết quả tốt nhất trên tập val với val loss = -0.165 khi train loss = -0.339 tại epoch thứ 8.

3.3 Đánh giá kết quả

3.3.1 Kết quả Owl-ViT kết hợp SAHI

Từ kết quả của huấn luyện trên, chọn mô hình ở epoch thứ 8 có kết quả tốt nhất để đánh giá trên tập kiểm thử của bộ dữ liệu Visdrone2019. Sử dụng độ đo mAP với ngưỡng IoU = 0.5 theo 3 loại đối tượng lớn, vừa và nhỏ ở bảng

Bảng 3.5: Kết quả đánh giá mô hình OwL-ViT + SAHI trên tập kiểm tra

STT	**Độ đo**	**Kết quả**	**Ghi chú**
1	mAP@0.5	28.5	Cho toàn bộ đối tượng
2	mAP@0.5s	17.5	Đối tượng kích thước nhỏ
3	mAP@0.5m	43.2	Đối tượng kích thước vừa
4	mAP@0.5l	48.7	Đối tượng kích thước lớn

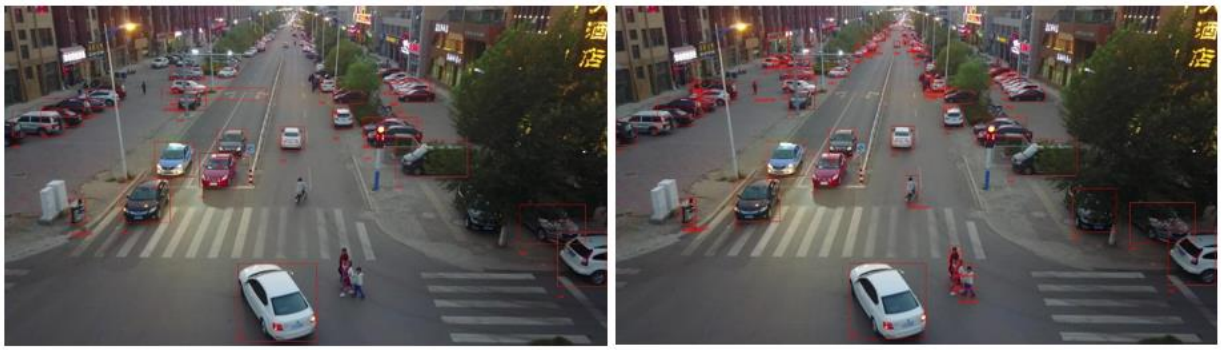
Bảng 3.6: So sánh OwL-ViT + SAHI và OwL-ViT ban đầu trên tập kiểm tra

	mAP@0.5	mAP@0.5s	mAP@0.5m	mAP@0.5l
OwL-ViT ban đầu	21.3	10.7	38.5	45.2
OwL-ViT + SAHI	28.5	17.5	43.2	48.7

Nhận xét:
- Nhìn chung OwL+ SAHI tăng kết quả lên đáng kể so với OwL ban đầu chưa được fine-tune. Tăng mAP@0.5 trên toàn tập kiểm tra là 7.2%.
- Đối với đối tượng loại nhỏ mAP@0.5s tăng 6.8%. So với mô hình gốc ban đầu, đây là một kết quả khá tốt.

Dưới đây là một số hình ảnh kết quả phát hiện đối tượng trong tập kiểm tra giữa mô hình OwL-ViT ban đầu chưa được fine-tune và mô hình OwL-ViT+SAHI.

Hình 3.4: Phát hiện đối tượng ô tô với OwL-ViT ban đầu (trái) và OwL-ViT kế hợp SAHI (phải)



Hình 3.5: Phát hiện đối tượng ô tô và người với OwL-ViT ban đầu (trái) và OwL-ViT kế hợp SAHI (phải)

3.3.2 So sánh với các mô hình khác kết hợp SAHI

Với kết quả trên, ta so sánh mô hình OwL-ViT kết hợp SAHI với một số mô hình khác cũng kết hợp với SAHI được đánh giá cùng trên tập kiểm tra của bộ dữ liệu VisDrone2019.

Bảng 3.7: So sánh OwL-ViT kết hợp SAHI với các mô hình khác

	mAP@0.5	**mAP@0.5s**	**mAP@0.5m**	**mAP@0.5l**
FCOS + SAHI	25.8	14.2	39.6	45.1
VFNet + SAHI	28.8	16.8	44.0	47.5
TOOD + SAHI	29.4	18.1	44.1	50.0
OwL-ViT + SAHI	28.5	17.5	43.2	48.7

Nhận xét:

- So sánh 4 mô hình trên, có thể thấy độ chính xác mAP của mô hình OwL-ViT kết hợp SAHI đứng thứ 3, chỉ hơn FCOS. Tuy nhiên so về loại đối tượng nhỏ mAP@0.5 thì mô hình lại đứng thứ 2, chỉ sau TOOD 0.6%.

-Với loại đối tượng lớn mAP@0.5l, mô hình cũng đứng thứ 2. Loại đối tượng vừa mAP@0.5m tuy cũng đứng thứ 3 nhưng chỉ kém TOOD là 0.9% và hơn khá nhiều so với FCOS là 3.6%.

3.4 Demo

3.4.1. Giới thiệu về Hugging Face Gradio

Hugging Face Gradio là một công cụ mã nguồn mở được phát triển bởi Hugging Face, nhằm giúp xây dựng giao diện người dùng tương tác cho mô hình Trí tuệ Nhân tạo (AI) một cách dễ dàng. Gradio kết hợp giữa khả năng xử lý ngôn ngữ tự nhiên và hình ảnh của Hugging Face với Gradio, một thư viện giao diện người dùng tương tác mã nguồn mở.

Gradio cung cấp các công cụ để tạo ra các giao diện tương tác cho mô hình AI một cách nhanh chóng, mà không yêu cầu nhiều kiến thức về lập trình hoặc giao diện người dùng. Với Gradio, bạn có thể xây dựng giao diện người dùng cho mô hình AI trong vài dòng mã, cho phép người dùng tương tác với mô hình và xem kết quả trực tiếp. Các tính năng chính của Hugging Face Gradio bao gồm:

1. Tự động tạo giao diện: Gradio tự động tạo giao diện dựa trên các đối số đầu vào và đầu ra của mô hình AI. Bạn chỉ cần xác định các kiểu dữ liệu và mô hình sẽ tự động tạo giao diện tương ứng.

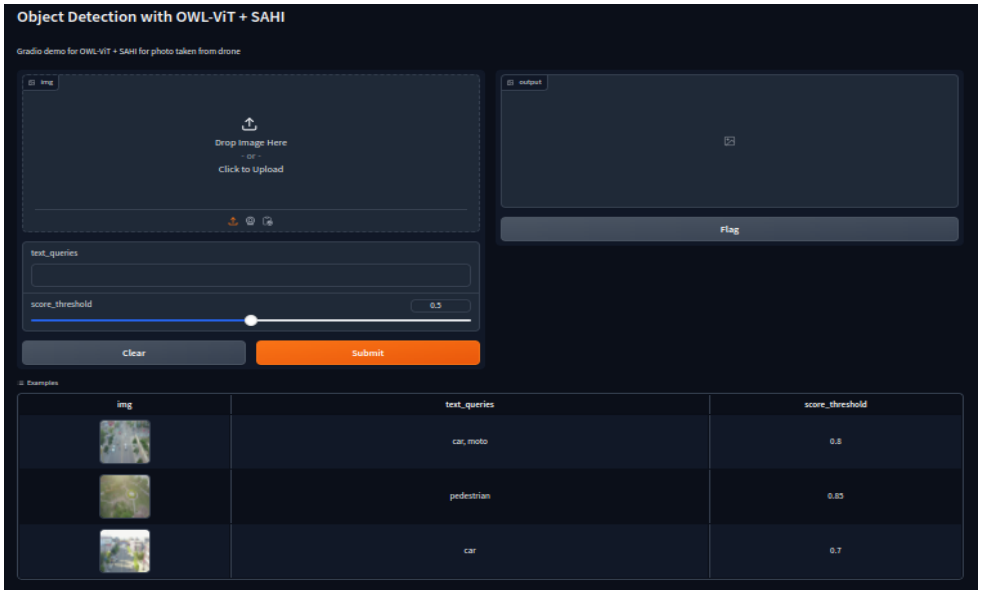
2. Hỗ trợ đa loại đầu vào: Gradio hỗ trợ nhiều kiểu đầu vào, bao gồm văn bản, hình ảnh, âm thanh và video. Điều này cho phép bạn xây dựng giao diện cho mô hình AI với nhiều loại dữ liệu đầu vào khác nhau.
3. Giao diện tương tác linh hoạt: Gradio cho phép bạn tạo các thành phần giao diện như hộp văn bản, nút nhấn, hộp kiểm, trình chọn và nhiều hơn nữa. Bạn có thể tùy chỉnh giao diện để phù hợp với nhu cầu của ứng dụng của mình.
4. Hiển thị kết quả trực tiếp: Gradio hiển thị kết quả từ mô hình AI ngay trên giao diện người dùng, cho phép người dùng xem kết quả một cách trực quan và tương tác với chúng.

Gradio cung cấp một cách dễ dàng và nhanh chóng để xây dựng giao diện người dùng tương tác cho mô hình AI. Nó là một công cụ hữu ích cho việc thử nghiệm, triển khai và chia sẻ mô hình AI với người dùng một cách trực quan và hấp dẫn.

3.4.2 Xây dựng giao diện demo

Xây dựng demo trên Hugging Face với mô hình OWL-ViT kết hợp SAHI đã được huấn luyện ở trên để phát hiện các đối tượng trong ảnh chụp từ Drone. Giao diện có hai phần chính là Input (bên trái) và Output(bên phải). Chi tiết được mô tả dưới đây:

- Input:
  -Ảnh đầu vào. Có thể tải ảnh lên từ thiết bị hoặc kéo ảnh từ phần examples bên dưới.
  -Từ truy vấn (text queries): tên loại đối tượng muốn phát hiện.
  -Thanh trượt điều chỉnh score threshold: điều chỉnh ngưỡng confidence cho phát hiện đối tượng.
- Output: Ảnh kết quả. Phát hiện đối tượng từ ảnh đầu vào.
- Nút Submit: Tiến hành phát hiện đám cháy từ ảnh.
- Nút Clear: Xóa ảnh, xóa kết quả. Trở về giao diện ban đầu.
- Examples: Các ảnh ví dụ kèm ngưỡng có sẵn, người dùng có thể kéo vào để demo nhanh kết quả.

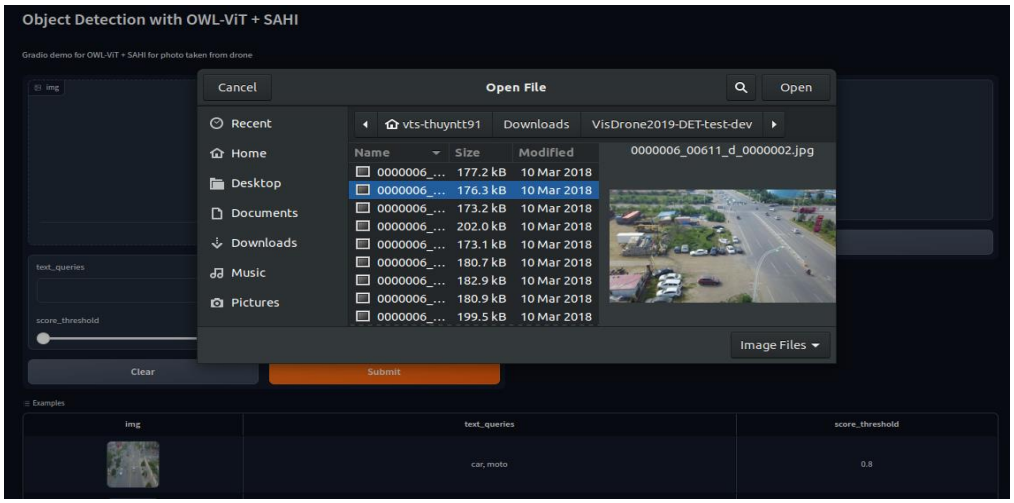


Hình 3.6: Giao diện demo của mô hình OWL-ViT kết hợp SAHI

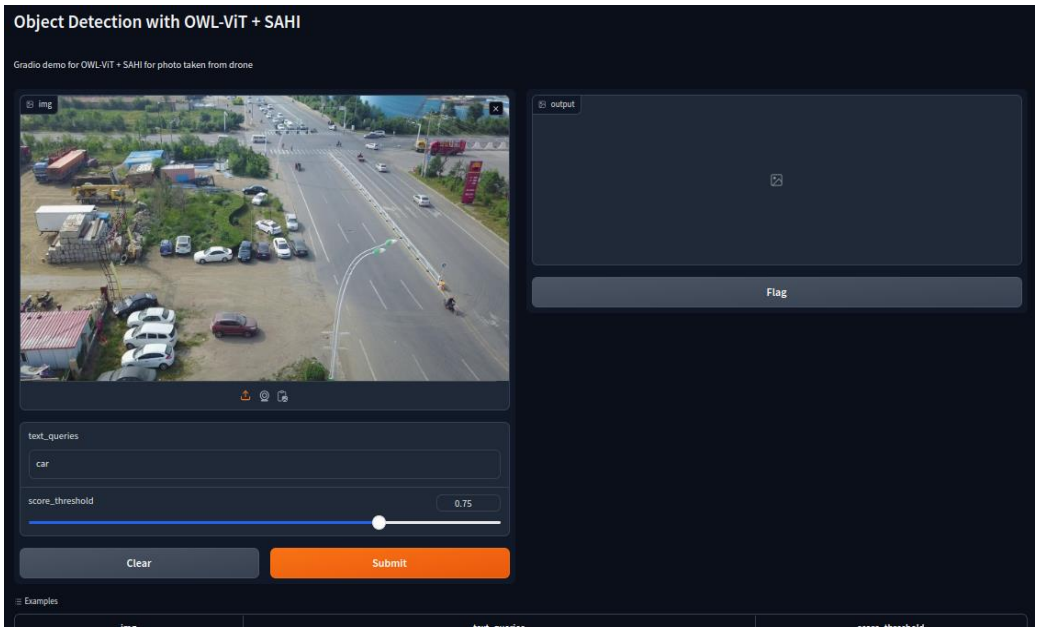
3.4.3 Kết quả demo

Sau khi xây dựng giao diện demo xong, dưới đây là kết quả sử dụng các chức năng trên giao diện. Đầu tiên là với hình ảnh được tải lên từ máy tính cá nhân.

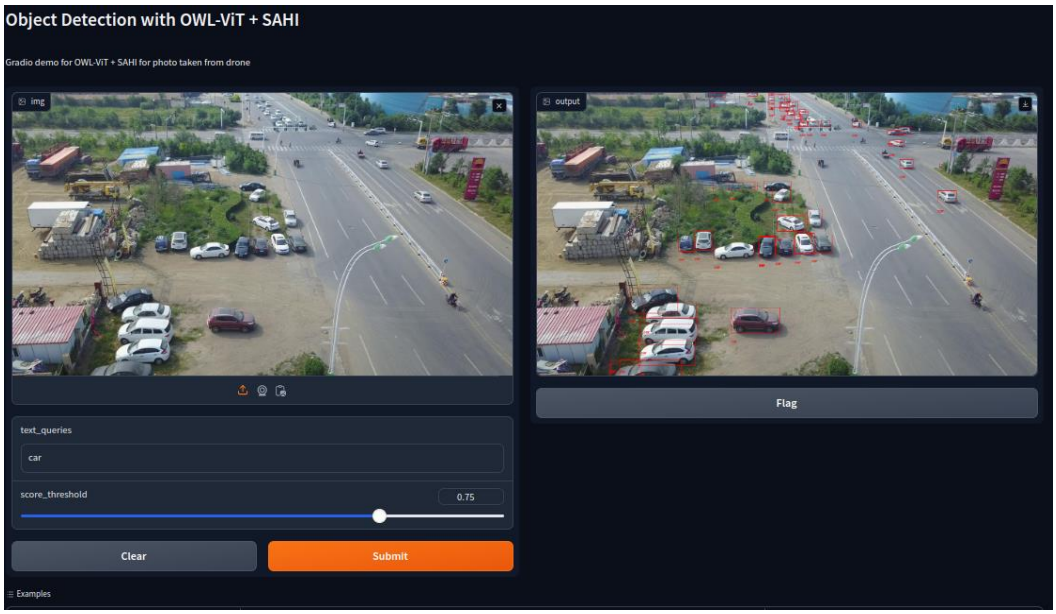
- Bước 1: Click chuột vào kí hiệu tải lên ở phần giao diện Input. Chọn ảnh và ấn tải lên.
- Bước 2: Nhập phần text queries: tên lớp đối tượng cần phát hiện
- Bước 3: Điều chỉnh thanh trượt score threshold về ngưỡng mong muốn
- Bước 4: Click nút Submit để tiến hành phát hiện đối tượng trong ảnh. Sau khi kết quả được hiển thị ở phần Output, có thể ấn nút Clear để xóa toàn bộ Input đã nhập và kết quả



Hình 3.7: Chọn ảnh tải lên với giao diện Demo

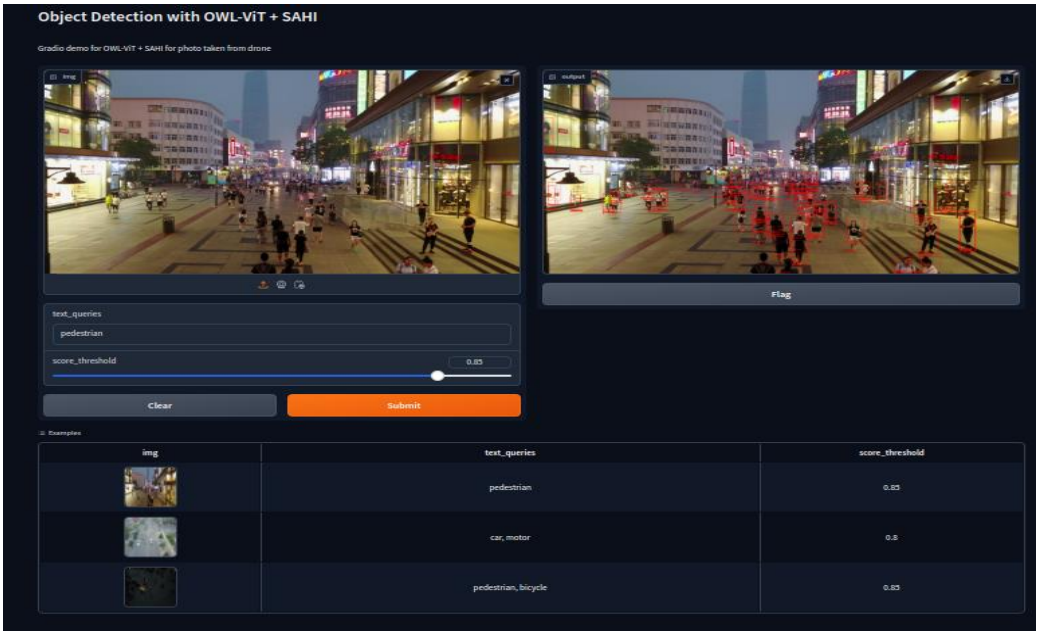


Hình 3.8: Nhập các thông tin đầu vào để phát hiện đối tượng trên giao diện Demo



Hình 3.9: Kết quả phát hiện đối tượng trên giao diện Demo

Ngoài ra, có thể sử dụng phần example, phát hiện đối tượng cùng ảnh ngưỡng có sẵn để demo kết quả. Chỉ cần click vào ảnh muốn demo trong danh sách và ấn Submit để xem kết quả.



Hình 3.10: Phần example trong giao diện demo

3.5 Kết luận chương

Với chương 3, đề án trình bày cụ thể về quá trình thực nghiệm và đánh giá hiệu quả của phương pháp được đề xuất. Mô tả chi tiết về các bộ dữ liệu được sử dụng trong quá trình tinh chỉnh và đánh giá. Trình bày cách thức tăng cường bộ dữ liệu đào tạo của tập dữ liệu Visdrone2019 bằng SAHI. Đồng thời, đề án cũng trình bày về quá trình tinh chỉnh mô hình OWL-ViT với các bước cụ thể. Đánh giá các kết quả của mô hình OWL+SAHI với mô hình gốc OWL-ViT để thấy được sự hiệu quả. So sánh thêm với các kết quả đã có với các phương pháp khác trên các bộ dữ liệu sử dụng. Xây dựng demo cho phương pháp để dễ dàng sử dụng mô hình phát hiện đối tượng cho ảnh, trực quan hóa kết quả.

KẾT LUẬN

Các kết quả đạt được của đề án tốt nghiệp:

- Nghiên cứu các bài toán phát hiện đối tượng, phát hiện đối tượng từ vựng mở và phát hiện đối tượng có kích thước nhỏ trong ảnh chụp từ drone. Trình bày kiến trúc mô hình OWL-ViT, kỹ thuật SAHI.
- Nghiên cứu phương pháp kết hợp giữa mô hình phát hiện đối tượng từ vựng mở OWL-ViT và kỹ thuật SAHI giúp phát hiện đối tượng kích thước nhỏ trong ảnh hiệu quả hơn. Từ đó tận dụng được kết hợp giữa văn bản và hình ảnh để phát hiện được nhiều đối tượng kích cỡ nhỏ trong ảnh chụp từ drone.
- Tiến hành thực nghiệm và đánh giá độ hiệu quả của phương pháp OWL-ViT cho việc phát hiện đối tượng kích thước nhỏ trong ảnh chụp từ drone. Xây dựng demo cho phương pháp để dễ dàng sử dụng mô hình phát hiện đối tượng và trực quan hóa kết quả.

Hướng nghiên cứu tiếp theo:

- Mô hình: nghiên cứu mô hình giúp phát hiện đối tượng với tốc độ nhanh và chính xác hơn cho video. Nghiên cứu thêm các kỹ thuật có thể áp dụng để giải quyết những vấn đề khác trong phát hiện đối tượng.
- Ứng dụng: mở rộng nghiên cứu tính ứng dụng của phương pháp có thể áp dụng vào những lĩnh vực nào khác trong cuộc sống để xây dựng sản phẩm thực tế.

TÀI LIỆU THAM KHẢO

[1] Zareian, A., Rosa, K.D., Hu, D.H., Chang (2020), "Open-vocabulary object detection using captions", arXiv:2011.10678 [cs.CV].

[2] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, Neil Houlsby (2023), "Simple Open-Vocabulary Object Detection with Vision Transformers", ArXiv: 2205.06230 [cs.CV].

[3] Fatih Cagatay Akyon, Sinan Onur Altinuc, Alptekin Temizel (2022), "Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection", *In 2022 IEEE International Conference on Image Processing (ICIP),* DOI: 10.1109/ICIP46576.2022.9897990.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, GeorgHeigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby (2020), "An Image is Worth 16x16: Transformers for Image Recognition at Scale", arXiv:2010.11929 [cs.CV].

[5] Valentina Emilia Balas, Liliana Perescu-Popescu, Nikos E Mastorakis (2009), "Multilayer perceptron and neural networks", ResearchGate.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017), "Attention Is All You Need", arXiv:1706.03762 [cs.CL].

[7] Zhi Tian, Chunhua Shen, Hao Chen, Tong He (2019), "FCOS: Fully Convolutional One-Stage Object Detection", arXiv:1904.01355 [cs.CV].

[8] Haoyang Zhang, Ying Wang, Feras Dayoub, Niko Sünderhauf (2020), "VarifocalNet: An IoU-aware Dense Object Detector", arXiv:2008.13367 [cs.CV].

[9] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R. Scott, Weilin Huang (2021), "TOOD: Task-aligned One-stage Object Detection", arXiv:2108.07755 [cs.CV].

[10] Juan R.Terven, Diana M.Cordova-Esparza (2023), "A Comprehensive Review of YOLO: From YOLOv1 and Beyond", arXiv: 2304.00501 [cs.CV].

[11] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, Qinghua Hu (2018), "Vision Meets Drones: A Challenge", arXiv:1804.07437 [cs.CV].

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg (2015), "SSD: Single Shot MultiBox Detector", arXiv: 1512.02325 [cs.CV].

[13] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN (2015): Towards Real-Time Object Detection with Region Proposal Networks", arXiv: 1506.01497 [cs.CV].

[14] Keiron O'Shea, Ryan Nash (2015), “An Introduction to Convolutional Neural Networks”, arXiv: 1511.08458 [cs.CV].
[15] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko (2020), “End-to-End Object Detection with Transformers”, arXiv: 2005.12872 [cs.CV].
[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei (2014), “ImageNet Large Scale Visual Recognition Challenge”, arXiv: 1409.0575 [cs.CV].
[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár (2014), “Microsoft COCO: Common Objects in Context”, arXiv: 1405.0312 [cs.CV]
[18] AdamW: Ilya Loshchilov, Frank Hutter (2017), “Decoupled Weight Decay Regularization”, arXiv: 1711.05101 [cs.CV].
[19] Adam: Diederik P. Kingma, Jimmy Ba (2014), “Adam: A Method for Stochastic Optimization”, arXiv: 1412.6980 [cs.CV].
[20] Sebastian Ruder (2017), “An overview of gradient descent optimization algorithms”, arXiv: 1609.04747 [cs.CV].
[21] Stephan Wojtowytsch (2021), “Stochastic gradient descent with noise of machine learning type. Part I: Discrete time analysis”, arXiv: 2105.01650 [cs.CV].
[22] Jingwen Fu, Bohan Wang, Huishuai Zhang, Zhizheng Zhang, Wei Chen, Nanning Zheng (2023), “When and Why Momentum Accelerates SGD:An Empirical Study”, arXiv: 2306.09000 [cs.CV]
[23] Adagrad: N. Zhang, D. Lei, J.F. Zhao (2018), “An Improved Adagrad Gradient Descent Optimization Algorithm”, *In 2018 Chinese Automation Congress (CAC)*, DOI: 10.1109/CAC.2018.8623271
[24] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, Silvio Savarese (2018), “Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression”
[25] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, Ruigang Yang (2019), “IoU Loss for 2D/3D Object Detection”, arXiv: 1908.03851 [cs.CV]