

TRƯỜNG ĐẠI HỌC MỎ ĐỊA CHẤT
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

BÁO CÁO SEMINAR

Nghiên cứu ứng dụng một số mô hình học máy trong
bài toán nhận dạng đối tượng trong không ảnh

Người trình bày: ThS. Đặng Hữu Nghị

1. Mở đầu

Việc phát hiện vật thể trong hình ảnh, với mục tiêu phát hiện các vật thể thuộc một tập hợp các danh mục vật thể được xác định trước (ví dụ: ô tô và người đi bộ), là một vấn đề đã có lịch sử lâu dài. Việc phát hiện vật thể chính xác sẽ có tác động tức thời và sâu rộng đến nhiều ứng dụng, chẳng hạn như hiểu hình ảnh, giám sát video và phát hiện dị thường. Mặc dù phát hiện vật thể thu hút nhiều nghiên cứu và đã đạt được những tiến bộ đáng kể với các kỹ thuật học sâu trong những năm gần đây, nhưng các thuật toán này thường không tối ưu để xử lý các chuỗi hoặc hình ảnh được chụp bởi các nền tảng dựa trên máy bay không người lái, do nhiều thách thức như thay đổi góc nhìn, tỷ lệ và che khuất.

2. Tổng quan về các phương pháp phát hiện đối tượng

Các bộ phát hiện đối tượng cổ điển:

Ban đầu, các phương pháp phát hiện đối tượng được xây dựng dựa trên mô hình cửa sổ trượt, sử dụng các đặc trưng và bộ phân loại được tạo thủ công trên các lưới ảnh dày đặc để định vị đối tượng. Là một trong những khuôn khổ phổ biến nhất trước đây, Viola và Jones [9] sử dụng đặc trưng Haar và thuật toán Adaboost để học một loạt các bộ phân loại xếp tầng cho phát hiện khuôn mặt, giúp đạt được kết quả chính xác với hiệu suất cao. Felzenszwalb và cộng sự [2] phát triển một phương pháp phát hiện đối tượng hiệu quả dựa trên sự kết hợp của các mô hình bộ phận biến dạng đa tỷ lệ. Cụ thể, họ tính toán các đặc trưng Biểu đồ Độ dốc Định hướng (HOG) trên mỗi bộ phận của đối tượng và huấn luyện SVM tiềm ẩn (một công thức cải tiến của MI-SVM về các biến tiềm ẩn) để có hiệu suất mạnh mẽ. Tuy nhiên, các bộ phát hiện đối tượng cổ điển không hoạt động tốt trong các tình huống khó khăn. Trong những năm gần đây, với sự phát triển của Mạng nơ-ron tích chập (CNN) sâu, lĩnh vực phát hiện đối tượng được thống trị bởi các bộ phát hiện dựa trên CNN, có thể được chia thành hai loại: phương pháp hai giai đoạn và phương pháp một giai đoạn.

Phương pháp dựa trên CNN hai giai đoạn:

Phương pháp hai giai đoạn đầu tiên tạo ra một nhóm các đề xuất đối tượng bằng một trình tạo đề xuất riêng biệt, sau đó dự đoán các vùng đối tượng chính xác và các nhãn lớp tương ứng, chẳng hạn như R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN, R-FCN, Mask R-CNN và FPN.

R-CNN là một trong những công trình tiên phong sử dụng mô hình CNN được đào tạo trước trên ImageNet, trích xuất một vector đặc trưng có độ dài cố định từ mỗi đề xuất bằng CNN, sau đó phân loại từng vùng bằng SVM tuyến tính theo danh mục cụ thể.

SPP-Net đề xuất lớp SPP gộp các đặc trưng và tạo ra các đầu ra có độ dài cố định để loại bỏ ràng buộc kích thước đầu vào cố định của mô hình CNN. Ngược lại với SPP, Fast R-CNN thiết kế một thuật toán huấn luyện một giai đoạn học chung để phân loại các đề xuất đối tượng và tính chỉnh vị trí không gian của chúng theo cách đầu cuối. Faster R-CNN cải tiến Fast R-CNN hơn nữa bằng cách sử dụng mạng đề xuất vùng thay vì thuật toán tìm kiếm chọn lọc để trích xuất các đề xuất vùng. Phương pháp R-FCN phát triển một mạng tích chập hoàn toàn (FCN) để giải quyết vấn đề phát hiện đối tượng, xây dựng một tập hợp các bản đồ nhạy cảm với vị trí bằng cách sử dụng một ngân hàng các lớp tích chập chuyên biệt để kết hợp phương sai dịch chuyển vào FCN. Gần đây, Lin và cộng sự [5] khai thác hệ thống phân cấp kim tự tháp đa tỷ lệ vốn có của các mạng tích chập sâu để xây dựng các kim tự tháp đặc trưng với chi phí bổ sung không đáng kể nhằm cải thiện hiệu suất phát hiện. Trong [4], phần đầu của mạng được thiết kế nhẹ nhất có thể để giảm chi phí tính toán, bằng cách sử dụng một bản đồ đặc trưng mỏng và một mạng con R-CNN giá rẻ (gộp và một lớp kết nối đầy đủ duy nhất). [10] đề xuất một R-CNN nhận biết tắc nghẽn mới để cải thiện khả năng phát hiện người đi bộ trong các cảnh đông đúc, thiết kế một tổn thất tổng hợp để buộc các đề xuất phải gần và định vị chặt chẽ với các đối tượng tương ứng. Nhìn chung, các phương pháp đã đề cập ở trên có chung một quy trình phát hiện đối tượng (tức là tạo đề xuất đối tượng, trích xuất đặc trưng, phân loại đối tượng và hồi quy hộp giới hạn). Giai đoạn tạo đề xuất vùng là nút thắt để cải thiện hiệu suất chạy.

Phương pháp dựa trên CNN một giai đoạn:

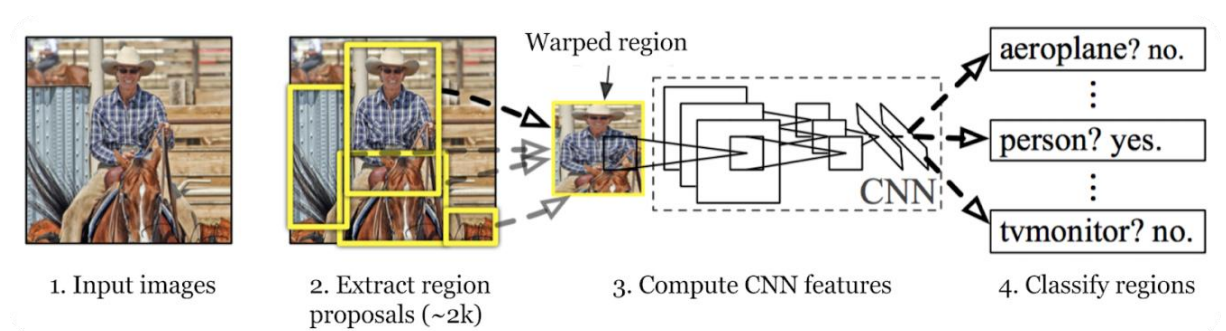
Khác với phương pháp hai giai đoạn, phương pháp một giai đoạn dự đoán trực tiếp vị trí, hình dạng và nhãn lớp của đối tượng mà không cần giai đoạn trích xuất đề xuất, có thể chạy với hiệu suất cao. Cộng đồng đã chứng kiến những cải tiến đáng chú ý theo hướng này, bao gồm YOLO, SSD, DSSD, RefineDet và RetinaNet.

Cụ thể, YOLO [7] xây dựng bài toán phát hiện đối tượng như một bài toán hồi quy đối với các hộp giới hạn phân tách không gian và các xác suất lớp liên quan. Sau đó, Redmon và cộng sự [8] đã cải thiện YOLO ở nhiều khía cạnh, chẳng hạn như thêm chuẩn hóa hàng loạt trên tất cả các lớp tích chập, sử dụng hộp neo để dự đoán hộp giới hạn và sử dụng huấn luyện đa tỷ lệ. SSD tận dụng một tập hợp các hộp neo mặc định với các tỷ lệ khung hình và tỷ lệ khác nhau để rời rạc hóa không gian đầu ra của các hộp giới hạn và hợp nhất các dự đoán từ nhiều bản đồ đặc trưng với các độ phân giải khác nhau. DSSD [3] tăng cường SSD với các lớp giải tích chập để đưa thêm ngữ cảnh quy mô lớn vào phát hiện đối tượng nhằm cải thiện độ chính xác, đặc biệt là đối với các đối tượng nhỏ. Zhang et al. [11] làm giàu ngữ nghĩa của các đặc điểm phát hiện đối tượng trong SSD, bằng một nhánh phân đoạn ngữ nghĩa và một mô-đun kích hoạt toàn cục. Lin et al. [6]

sử dụng Focal Loss (RetinaNet) để giải quyết vấn đề mất cân bằng lớp trong phát hiện đối tượng bằng cách định hình lại mất entropy chéo chuẩn sao cho nó giảm trọng số mất mát được gán cho các ví dụ được phân loại tốt. Ngoài ra, Zhang et al. [11] đề xuất một bộ dò RefineDet một lần. Nó được hình thành bởi hai mô-đun được kết nối với nhau, tức là mô-đun tinh chỉnh neo và mô-đun phát hiện đối tượng, đạt được độ chính xác và hiệu quả cao. Hơn nữa, Chen et al. [1] đề xuất một mạng tinh chỉnh kép để tăng hiệu suất của các bộ dò một giai đoạn, trong đó xem xét tinh chỉnh neo và tinh chỉnh bù đặc trưng trong phát hiện bù neo.

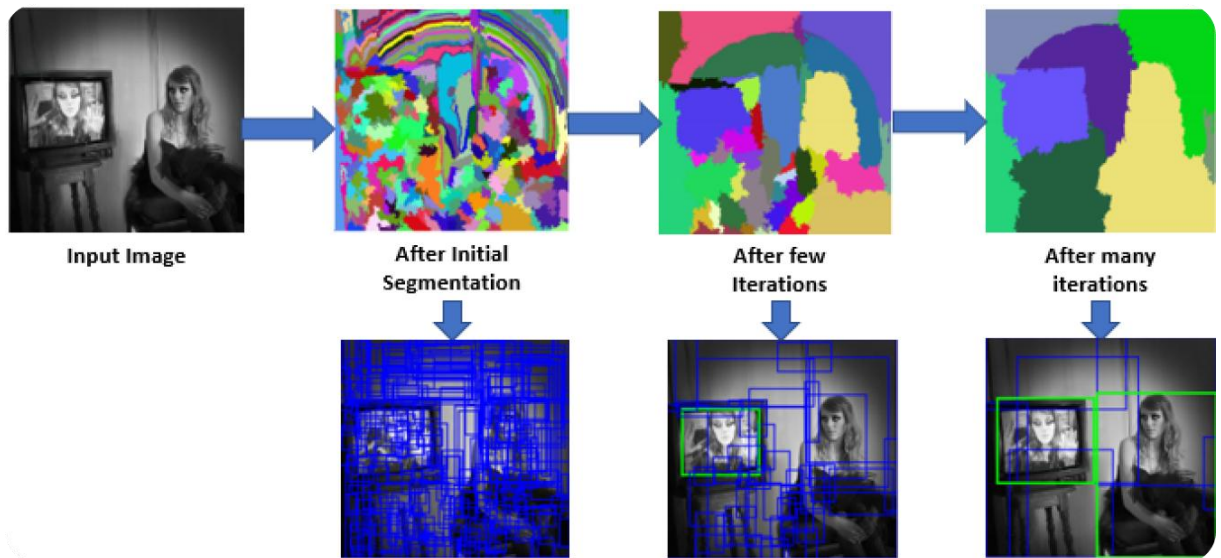
3. Mô hình R-CNN

Quy trình phát hiện đối tượng của mô hình R-CNN bao gồm ba bước chính: tạo đề xuất vùng, trích xuất đặc trưng và phân loại đối tượng đồng thời tinh chỉnh các hộp giới hạn của chúng. Chúng ta hãy cùng xem xét từng bước.



Đề xuất vùng: Nền tảng của RCNN

Trong bước đầu tiên, mô hình R-CNN quét hình ảnh để tạo ra nhiều đề xuất vùng. Đề xuất vùng là các khu vực tiềm năng có thể chứa các đối tượng. Các phương pháp như Tìm kiếm có chọn lọc được sử dụng để xem xét các khía cạnh khác nhau của hình ảnh, chẳng hạn như màu sắc, kết cấu và hình dạng, chia nó thành các phần khác nhau. Tìm kiếm có chọn lọc bắt đầu bằng cách chia hình ảnh thành các phần nhỏ hơn, sau đó hợp nhất các phần tương tự để tạo thành các khu vực quan tâm lớn hơn. Quá trình này tiếp tục cho đến khi khoảng 2.000 đề xuất vùng được tạo ra.



Các đề xuất khu vực này giúp xác định tất cả các vị trí có thể có nơi một đối tượng có thể xuất hiện. Trong các bước tiếp theo, mô hình có thể xử lý hiệu quả các khu vực liên quan nhất bằng cách tập trung vào các khu vực cụ thể này thay vì toàn bộ hình ảnh. Sử dụng các đề xuất khu vực cân bằng giữa tính kỹ lưỡng và hiệu quả tính toán.

Trích xuất đặc trưng hình ảnh: Nắm bắt các chi tiết

Bước tiếp theo trong quy trình phát hiện đối tượng của mô hình R-CNN là trích xuất đặc trưng từ các đề xuất vùng. Mỗi đề xuất vùng được thay đổi kích thước thành một kích thước nhất quán mà CNN yêu cầu (ví dụ: 224x224 pixel). Việc thay đổi kích thước giúp CNN xử lý từng đề xuất một cách hiệu quả. Trước khi biến dạng, kích thước của mỗi đề xuất vùng được mở rộng một chút để bao gồm thêm 16 pixel ngữ cảnh bổ sung xung quanh vùng để cung cấp thêm thông tin xung quanh nhằm trích xuất đặc trưng tốt hơn.

Sau khi thay đổi kích thước, các đề xuất khu vực này được đưa vào CNN như AlexNet, thường được đào tạo trước trên một tập dữ liệu lớn như ImageNet CNN xử lý từng vùng để trích xuất các vector đặc trưng đa chiều, nắm bắt các chi tiết quan trọng như cạnh, kết cấu và hoa văn. Các vector đặc trưng này cô đọng thông tin thiết yếu từ các vùng. Chúng chuyển đổi dữ liệu ảnh thô thành định dạng mà mô hình có thể sử dụng cho các phân tích sâu hơn. Việc phân loại và định vị chính xác các đối tượng trong các giai đoạn tiếp theo phụ thuộc vào quá trình chuyển đổi quan trọng này từ thông tin hình ảnh thành dữ liệu có ý nghĩa.

Phân loại đối tượng: Xác định các đối tượng được phát hiện

Bước thứ ba là classify Các đối tượng trong các vùng này. Điều này có nghĩa là xác định danh mục hoặc lớp của từng đối tượng được tìm thấy trong các đề xuất. Các vector đặc trưng được trích xuất sau đó được chuyển qua bộ phân loại học máy.

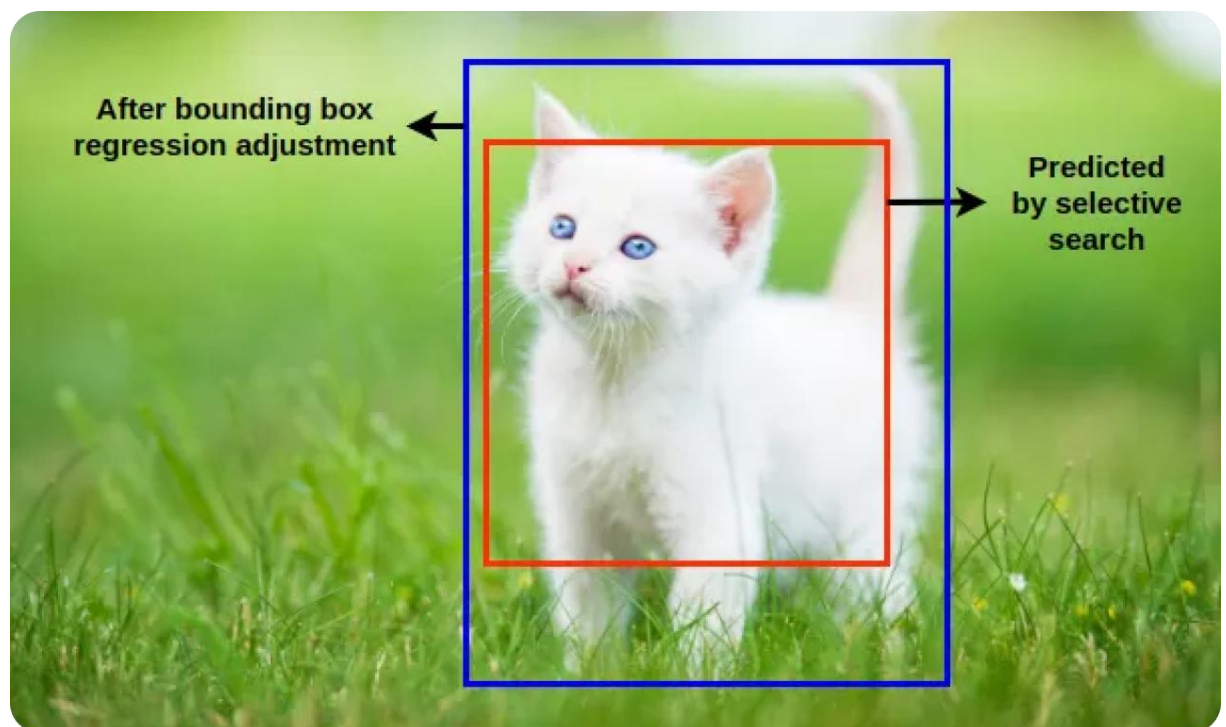
Trong trường hợp của R-CNN, Máy Vector Hỗ trợ (SVM) thường được sử dụng cho mục đích này. Mỗi SVM được huấn luyện để nhận ra một lớp đối tượng cụ thể bằng

cách phân tích các vector đặc trưng và quyết định xem một vùng cụ thể có chứa một thể hiện của lớp đó hay không. Về cơ bản, đối với mỗi danh mục đối tượng, có một bộ phân loại chuyên dụng kiểm tra từng đề xuất vùng cho đối tượng cụ thể đó.

Trong quá trình huấn luyện, các bộ phân loại được cung cấp dữ liệu được gắn nhãn với các mẫu dương tính và âm tính:

- Các mẫu dương tính: Các vùng chứa đối tượng mục tiêu.
- Các mẫu âm tính: Các vùng không có đối tượng.

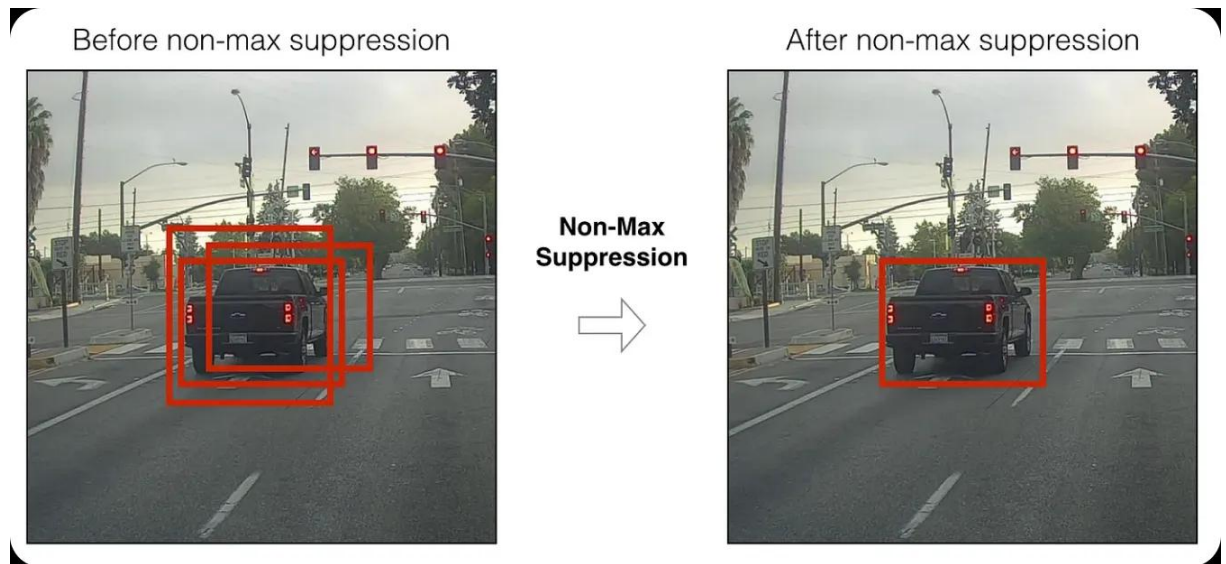
Các bộ phân loại học cách phân biệt giữa các mẫu này. Hồi quy hộp giới hạn (Bounding box regression) tiếp tục tinh chỉnh vị trí và kích thước của các đối tượng được phát hiện bằng cách điều chỉnh các hộp giới hạn được đề xuất ban đầu để phù hợp hơn với ranh giới đối tượng thực tế. Mô hình R-CNN có thể xác định và định vị chính xác các đối tượng bằng cách kết hợp phân loại và hồi quy hộp giới hạn.



Kết hợp tất cả lại với nhau: Tinh chỉnh các phát hiện bằng NMS

Sau các bước phân loại và hồi quy hộp giới hạn, mô hình thường tạo ra nhiều hộp giới hạn chồng lấn cho cùng một đối tượng. Loại bỏ không tối đa (NMS) được áp dụng để tinh chỉnh các phát hiện này, giữ lại các hộp chính xác nhất. Mô hình loại bỏ các hộp trùng lặp và dư thừa bằng cách áp dụng NMS và chỉ giữ lại những phát hiện đáng tin cậy nhất.

NMS hoạt động bằng cách đánh giá điểm tin cậy (chỉ ra khả năng thực sự có một vật thể được phát hiện) của tất cả các hộp giới hạn và loại bỏ những hộp có điểm cao hơn chồng chéo đáng kể.



Một ví dụ về kỹ thuật non-maximum suppression.

Sự phân tích các bước trong NMS :

- **Sắp xếp:** Các hộp giới hạn được sắp xếp theo điểm tin cậy của chúng theo thứ tự giảm dần.
- **Lựa chọn:** Hộp có điểm cao nhất sẽ được chọn và tất cả các hộp chồng lên nhau đáng kể (dựa trên Giao trên Hộp, IoU) cùng với nó sẽ bị xóa bỏ.
- **Vòng lặp:** Quá trình này lặp lại cho hộp có điểm số cao nhất tiếp theo và tiếp tục cho đến khi tất cả các hộp đã được xử lý.

Nói tóm lại, mô hình R-CNN phát hiện các đối tượng bằng cách tạo ra các đề xuất vùng, trích xuất các đặc điểm bằng CNN, phân loại các đối tượng và tính chỉnh vị trí của chúng bằng hồi quy hộp giới hạn và sử dụng Non-Maximum Suppression (NMS) chỉ giữ lại những phát hiện chính xác nhất.

4. Mô hình YOLO

YOLO, hay “You Only Look Once,” là một thuật toán phát hiện đối tượng trong thị giác máy tính. Điểm đặc biệt của YOLO nằm ở cách nó đưa ra dự đoán về các hộp giới hạn (bounding box) và xác suất của đối tượng trong một lần truyền hình ảnh duy nhất.

Trước đây, các thuật toán phát hiện đối tượng thường sử dụng trình phân loại đã đào tạo sẵn để xác định đối tượng sau khi đã tạo ra các khu vực quan tâm có thể có. Tuy nhiên, YOLO làm khác đi bằng cách đưa ra dự đoán toàn bộ quá trình trong một lần chạy, từ việc xác định vị trí của đối tượng đến việc phân loại chúng.

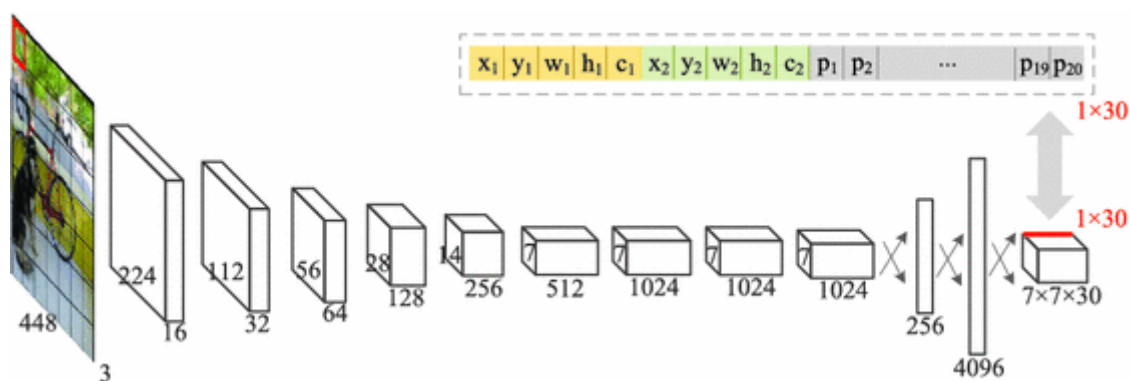
Với cách tiếp cận đột phá này, YOLO đã đạt được sự cải tiến đáng kể và vượt trội so với các thuật toán phát hiện đối tượng khác, đặc biệt trong việc đảm bảo tính thời gian thực.

Các thuật toán phát hiện đối tượng trước đây, như Faster RCNN, thường hoạt động theo các bước gồm việc xác định khu vực quan tâm và sau đó tiến hành phân loại trên từng khu vực riêng biệt. Trong khi đó, YOLO hoàn tất tất cả các dự đoán chỉ trong một lần chạy, giúp nó đạt được hiệu quả tính toán cao.

Với sự đột phá của YOLO, đã có nhiều phiên bản mới và cải tiến được giới thiệu kể từ lần ra mắt đầu tiên vào năm 2015, mang lại sự phát triển đáng kể cho lĩnh vực phát hiện đối tượng. Hãy cùng điểm qua các mốc thời gian thể hiện sự tiến bộ của YOLO trong những năm qua.

Cơ chế hoạt động của YOLO

YOLO có kiến trúc khá cơ bản, bao gồm base network và extra layers. Base network là Convolution network có nhiệm vụ trích xuất đặc trưng ảnh. Extra layers là các layer cuối dùng để phân tích các đặc trưng và phát hiện vật thể. Base network thường được dùng là Darknet.



YOLO hoạt động theo cơ chế detect vật thể trên từng vùng ảnh. Ảnh được chia thành $S \times S$ ô, ô hay còn gọi là cell ... Thực ra là chia trong tưởng tượng, không phải cắt ảnh ra hay thực hiện bất kì bước image-processing nào. Bản chất của việc chia ảnh là việc chia output, target thành dạng matrix A kích thước $S \times S$. Nếu trong ảnh, tâm của vật thể nằm trong cell thứ (i,j) thì output tương ứng nó sẽ nằm trong $A[i,j]$.

Luồng xử lí như sau:

- Convolution network tiến hành trích xuất đặc trưng ảnh
- Extra layers (fully connected layers) phân tích, phát hiện vật thể. Trả về output là một matrix A có kích thước:

$$Shape(A) = S * S * (5 * B + C)$$

Trong đó, B là số bound box, mỗi bound box gồm 5 số liệu: $(x,y,w,h, confidence_score)$. confidence score xác suất tại ô đó có object hay không. Cuối cùng là C phần tử - đại diện cho phân bố xác suất về loại object, tức class distribution. Vì C phần tử này là một phân phối xác suất nên cần đảm bảo:

$$\sum_0^c p_i = 1$$

Như vậy, bạn đã thấy rằng YOLO tính toán tọa độ bbox, xác suất có xuất hiện vật thể, phân bố xác suất để phân loại vật thể... tất cả đều được thực hiện trong 1 lần. Đó là lý do khiến cho YOLO nhanh hơn nhiều so với các 2-stage model, tạo nên lợi thế cho YOLO.

Loss function

Loss function của YOLO khá đơn giản nếu cần giải thích, nhưng để code lại là một vấn đề khác =))

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \quad (1)$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \quad (2)$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \quad (3)$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \quad (4)$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (5)$$

Loss_total được tổng hợp từ 5 phần, mỗi phần nhằm tối ưu một vấn đề riêng biệt:

- p1: xy_loss khi object tồn tại ở box_j trong $cell_i$
- p2: wh_loss khi object tồn tại ở box_j trong $cell_i$
- p3: confidence_loss khi object tồn tại ở box_j trong $cell_i$
- p4: confidence_loss các box không tồn tại object
- p5: class_probability_loss tại cell có tồn tại object.

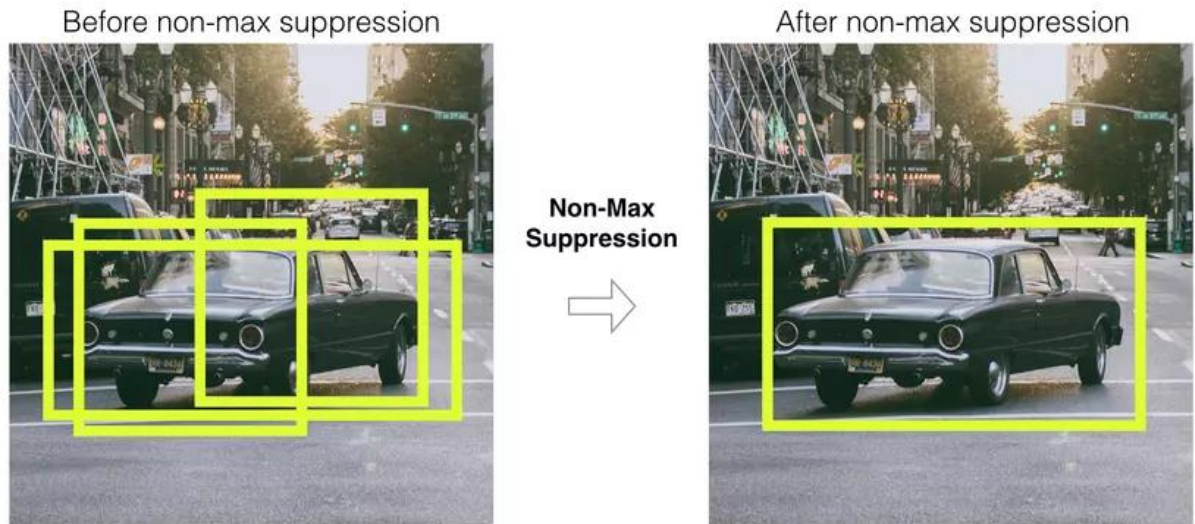
Chú thích:

- $l_{ij}^{\text{obj}} = 1$ nếu trong ô thứ i, có box thứ j có chứa object.
- l_{ij}^{noobj} ngược lại với l_{ij}^{obj}

- $l_i^{obj} = 1$ Nếu ô thứ i có chứa object (ngược lại = 0).
- λ_{coord} và λ_{noobj} là trọng số thành phần.

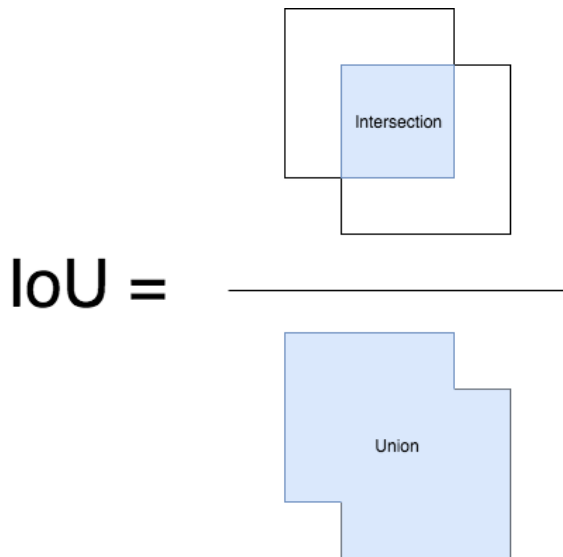
Non-max suppression

Giả sử trải qua quá trình huấn luyện mô hình, bạn thu được một model đủ tốt và muốn đưa vào ứng dụng. Tuy nhiên model dù tốt vẫn xảy ra trường hợp: dự đoán nhiều bounding box cho cùng một vật thể



Để giải quyết vấn đề này, ta lọc bớt các bounding box dư thừa (trùng nhau và cùng class) bằng non-max suppression. Ý tưởng:

- Các box có confidence_score được xếp hạng từ cao xuống thấp. [box_0, box_1, ... box_n]
- Duyệt từ đầu danh sách, với từng box_i, loại đi các box_j có $IOU(box_i, box_j) > threshold$. Trong đó $j > i$. threshold là một giá trị ngưỡng được lựa chọn sẵn. IOU là công thức tính độ overlap - giao thoa giữa hai bounding box
- Công thức tính IOU được minh họa rất dễ hiểu theo hình:



Các phiên bản của YOLO

YOLO v2 (YOLO9000)

Phiên bản này được giới thiệu vào năm 2016 và được biết đến với tên YOLO9000. YOLO v2 được thiết kế để nhanh hơn và chính xác hơn phiên bản gốc. Nó cung cấp khả năng phát hiện nhiều loại đối tượng hơn. Cải tiến quan trọng trong YOLO v2 bao gồm việc sử dụng anchor boxes, chuẩn hóa hàng loạt, và chiến lược đào tạo đa quy mô.

YOLO v3

YOLO v3, ra mắt vào năm 2018, tập trung vào việc tăng độ chính xác và tốc độ của thuật toán. Nó sử dụng một kiến trúc CNN mới có tên Darknet-53 và anchor boxes được điều chỉnh để phù hợp với kích thước và hình dạng của đối tượng. YOLO v3 cũng giới thiệu feature pyramid networks (FPN) để cải thiện phát hiện ở nhiều tỷ lệ khung hình. Điều này cải thiện hiệu suất phát hiện trên các đối tượng nhỏ.

YOLO v4

YOLO v4, giới thiệu vào năm 2020, sử dụng một kiến trúc CNN mới có tên là CSPNet và sử dụng anchor boxes với k-means clustering. Nó giới thiệu một loss function mới gọi là GHM loss và cải thiện kiến trúc của FPN so với YOLO v3.

YOLO v5

YOLO v5, được giới thiệu vào năm 2020, sử dụng kiến trúc phức tạp hơn gọi là EfficientDet và đào tạo trên tập dữ liệu lớn hơn (D5) bao gồm nhiều danh mục đối tượng. Nó sử dụng dynamic anchor boxes, spatial pyramid pooling (SPP), và CIoU loss để cải thiện hiệu suất đối tượng phát hiện trên nhiều loại đối tượng.

YOLO v6

Phiên bản này được đề xuất vào năm 2022 bởi Li và cộng sự. YOLO v6 sử dụng một biến thể của kiến trúc EfficientNet có tên là EfficientNet-L2 và đưa ra dense anchor boxes.

Mỗi phiên bản YOLO đã mang lại cải tiến đáng kể trong việc phát hiện đối tượng với việc sử dụng các kiến thức mới, kiến trúc mạng nâng cấp và phương pháp đào tạo tốt hơn. Sự phát triển liên tục của YOLO cho thấy cam kết của cộng đồng nghiên cứu trong việc cải thiện thuật toán phát hiện đối tượng và thị giác máy tính.

YOLO V7

YOLO v7 sử dụng một tập hợp gồm chín anchor boxes, một công cụ quan trọng giúp cải thiện việc xác định đối tượng. Các anchor boxes với các tỷ lệ khung hình khác nhau cho phép YOLOv7 phát hiện hiệu quả các đối tượng có hình dạng và kích thước đa dạng.

Một cải tiến quan trọng trong YOLO v7 là việc sử dụng focal loss function. Trong khi các phiên bản trước sử dụng cross-entropy loss, focal loss tập trung vào việc giảm trọng số cho các ví dụ được phân loại đúng, hỗ trợ việc phát hiện đối tượng khó hơn. Điều này đóng góp đáng kể vào độ chính xác tổng thể.

Độ phân giải của YOLO v7 đã được nâng lên. Nó xử lý hình ảnh ở độ phân giải 608 x 608 pixel, cao hơn so với 416 x 416 pixel trong YOLO v3. Điều này giúp YOLOv7 phát hiện các đối tượng nhỏ hơn và tăng độ chính xác tổng thể của thuật toán.

YOLO v7 vẫn duy trì tốc độ ấn tượng của dòng sản phẩm YOLO. Với khả năng xử lý hình ảnh ở tốc độ 155 khung hình mỗi giây, YOLO v7 vượt trội so với nhiều thuật toán phát hiện đối tượng khác. Điều này làm cho nó trở thành một lựa chọn lý tưởng cho các ứng dụng thời gian thực như giám sát và ô tô tự lái, nơi tốc độ xử lý là yếu tố quan trọng.

YOLO v7, dù mạnh mẽ và hiệu quả, vẫn tồn tại một số hạn chế như sau:

- **Phát hiện đối tượng nhỏ còn hạn chế:** YOLOv7 gặp khó khăn trong việc phát hiện các đối tượng nhỏ hoặc ở khoảng cách xa. Điều này đặc biệt quan trọng trong các tình huống đông đúc hoặc khi đối tượng cần phát hiện không nổi bật trong bối cảnh.
- **Khó khăn với tỷ lệ và kích thước khung hình khác nhau:** YOLOv7 có thể gặp khó khăn trong việc phát hiện đối tượng ở các tỷ lệ khung hình khác nhau. Điều này đôi khi gây khó khăn khi có sự biến đổi về kích thước giữa các đối tượng trong cùng một hình ảnh.
- **Nhạy cảm với biến đổi ánh sáng và môi trường:** YOLOv7 có thể không ổn định khi ánh sáng hoặc điều kiện môi trường thay đổi. Nó có thể dẫn đến việc

không thể phát hiện đối tượng hoặc tạo ra các dự đoán không chính xác trong điều kiện môi trường biến đổi.

- **Yêu cầu tính toán nhiều:** YOLOv7 yêu cầu nhiều tính toán để xử lý ảnh, điều này có thể gây khó khăn khi cố gắng chạy thuật toán trong thời gian thực trên các thiết bị có tài nguyên hạn chế như điện thoại thông minh. Việc cải thiện tốc độ suy luận có thể đòi hỏi sự đầu tư vào phần cứng mạnh mẽ.

YOLO v8

YOLOv8 là phiên bản mới nhất trong dòng mô hình YOLO. Các mô hình YOLO được huấn luyện trước trên các bộ dữ liệu lớn như COCO và ImageNet. Điều này cho phép chúng vừa có khả năng cung cấp dự đoán cực kỳ chính xác với các lớp đã được huấn luyện, vừa có thể học các lớp mới một cách tương đối dễ dàng.

Các mô hình YOLO cũng có tốc độ huấn luyện nhanh hơn, với độ chính xác cao và kích thước mô hình nhỏ. Chúng có thể được huấn luyện trên các GPU đơn, do đó dễ tiếp cận hơn đối với các nhà phát triển.

Được công bố vào đầu năm 2023, YOLOv8 đã mang lại nhiều điểm tích cực so với phiên bản tiền nhiệm, như phát hiện không dùng anchor, giới thiệu lớp tích chập C3 và tăng cường mosaic.

5. Kết quả thực nghiệm

5.1 Bộ dữ liệu VisDrone-DET2018

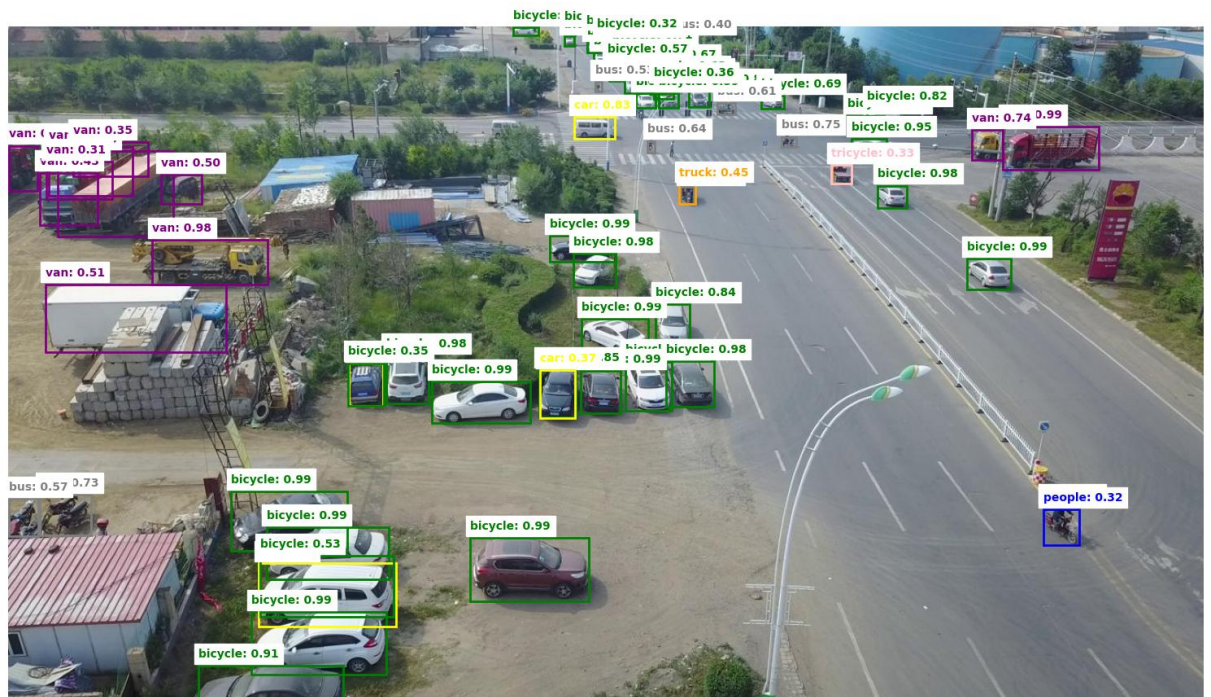
Bộ dữ liệu VisDrone-DET2018, tập trung vào việc phát hiện mười loại vật thể được xác định trước (ví dụ: người đi bộ, người1, ô tô, xe tải nhỏ, xe buýt, xe tải, xe máy, xe đạp, xe ba bánh có mái che và xe ba bánh) trong hình ảnh từ máy bay không người lái. Chúng tôi yêu cầu mỗi thuật toán tham gia dự đoán các hộp giới hạn của các vật thể trong các lớp được xác định trước với độ tin cậy giá trị thực. Một số phương tiện đặc biệt hiếm khi xuất hiện (ví dụ: xe tải xương cơ khí, xe nâng và xe bồn) bị bỏ qua trong

VisDrone-DET2018, tập trung vào việc phát hiện mười loại vật thể được xác định trước (ví dụ: người đi bộ, người1, ô tô, xe tải nhỏ, xe buýt, xe tải, xe máy, xe đạp, xe ba bánh có mái che và xe ba bánh) trong hình ảnh từ máy bay không người lái.

Bộ dữ liệu VisDrone-DET2018 bao gồm 8.599 hình ảnh (6.471 để huấn luyện, 548 để xác thực, 1.580 để kiểm tra) với các chú thích phong phú, bao gồm hộp giới hạn vật thể, danh mục vật thể, tỷ lệ che khuất và tỷ lệ cắt cụt. Với các kịch bản thực tế đa dạng, bộ dữ liệu được thu thập bằng nhiều nền tảng máy bay không người lái (tức là máy bay không người lái với các mô hình khác nhau), trong các tình huống khác nhau (trên 14 thành phố khác nhau trải dài hàng nghìn km) và trong các điều kiện thời tiết và ánh sáng

khác nhau. Dữ liệu thực tế cơ bản được chú thích thủ công trong các tập huấn luyện và xác thực được cung cấp cho người dùng, nhưng dữ liệu thực tế cơ bản của tập kiểm tra được giữ lại để tránh hiện tượng (quá) khớp các thuật toán.

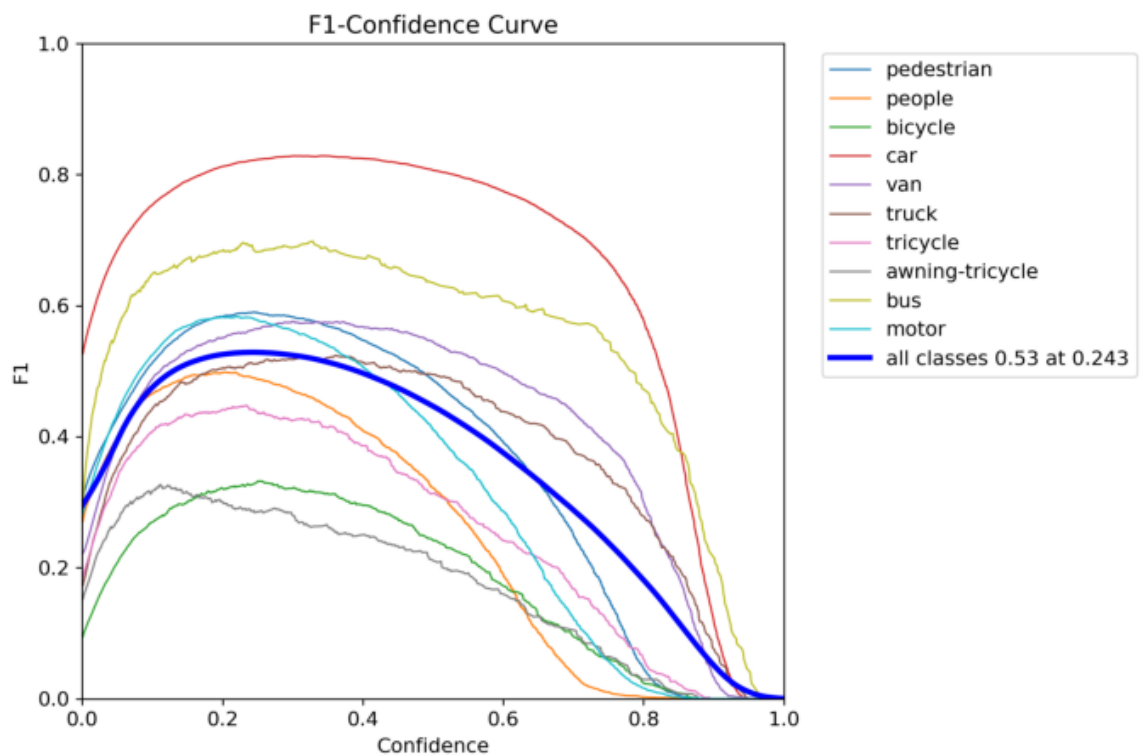
5.2. Ứng dụng R-CNN



5.3. Ứng dụng YOLO v8m



F1 Curve



6. Kết luận

Trong bài toán phát hiện đối tượng có trong ảnh từ trên không, chúng tôi đã chạy thực nghiệm phương pháp R-CNN và YOLO v8m trên bộ dữ liệu VisDrone-DET2018. Từ kết quả thực nghiệm, chúng tôi thấy rằng phương pháp có khả năng phát hiện đa dạng trên các lớp đối tượng khác nhau.

TÀI LIỆU THAM KHẢO

- [1]. Chen, X., Wu, Z., Yu, J.: Dual refinement network for single-shot object detection. CoRR abs/1807.08638 (2018), <http://arxiv.org/abs/1807.08638>
- [2]. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI 32(9), 1627–1645 (2010)
- [3]. Fu, C., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD : Deconvolutional single shot detector. CoRR abs/1701.06659 (2017), <http://arxiv.org/abs/1701.06659>
- [4]. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head R-CNN: in defense of two-stage object detector. CoRR abs/1711.07264 (2017)
- [5]. Lin, T., Doll'ar, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR. pp. 936–944 (2017)
- [6]. P. Zhu, L. Wen, D. Du, X. Bian, B. Ling, et al.
- [7]. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
- [8]. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: CVPR. pp. 6517–6525 (2017)
- [9]. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR. pp. 511–518 (2001)
- [10]. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Occlusion-aware R-CNN: detecting pedestrians in a crowd. In: ECCV (2018)
- [11]. Zhang, Z., Qiao, S., Xie, C., Shen, W., Wang, B., Yuille, A.L.: Single-shot object detection with enriched semantics. In: CVPR (2018)