

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO ĐỒ ÁN MÔN HỌC
TOÁN CHO KHOA HỌC MÁY TÍNH
CS115.M13.KHCL

GIẢNG VIÊN HƯỚNG DẪN: LƯƠNG NGỌC HOÀNG
SINH VIÊN THỰC HIỆN: ĐÀO TRẦN ANH TUẤN – 20522107
PHẠM TRẦN ANH TIÊN - 20522012
TRẦN PHÚ VINH – 20522161

TP. HỒ CHÍ MINH, 12/2021

Mục lục

1. Giới thiệu	3
2. Grid Cell.....	3
3. Cấu trúc của YOLOv3	5
3.1 Feature Extractor	5
3.2 Feature Detector	5
4. Intersection over Union (IOU)	7
5. Khung neo (Anchor box)	8
6. Dự đoán bounding box	9
7. Non-max suppression.....	10
8. Hàm độ lỗi (Loss function).....	11
8.1 Classification loss	11
8.2 Localization loss.....	12
8.3 Confidence loss.....	12
9. Tổng kết	13
10. Tài liệu tham khảo	13

1. Giới thiệu

Object Detection là một kĩ thuật bao gồm việc định vị những đối tượng có trong ảnh kết hợp nhận dạng những đối tượng đó. Đây cũng là một bài toán quan trọng trong thị giác máy tính cũng như là máy học. Object Detection có khả năng ứng dụng cao trong nhiều lĩnh vực của đời sống. Kĩ thuật này có thể dùng để đếm số lượng xe, người đi bộ đang lưu thông, được ứng dụng trong hệ thống bảo mật... YOLO là một thuật toán nhận dạng đối tượng nổi tiếng với tốc độ nhanh cùng với độ chính xác khá tốt.

Khác với các thuật toán vào thời điểm 2015-2016, thay vì dùng cửa sổ trượt hay regional proposal và tính toán nhiều lần, YOLO sẽ nhìn toàn bộ bức ảnh được đưa vào trong suốt thời gian huấn luyện, kiểm tra và chỉ tính toán một lần duy nhất. Điều này không những khiến mô hình YOLO có thể hiểu được thông tin của cả bức ảnh mà còn tăng tốc độ thuật toán.

2. Grid Cell

Ảnh đầu vào được chia thành một lưới với kích thước $S \times S$ (S dòng, S cột), mỗi ô lưới sẽ dự đoán B bounding box. Mỗi bounding box gồm các thành phần: tọa độ (coordinate) x, y, w, h , độ tin cậy p_c (box confidence score) và xác suất dự đoán C lớp (conditional class probability) mà đối tượng trong bounding box thuộc về. Kết quả sau khi dự đoán sẽ được mã hóa dưới dạng tensor với kích thước $S \times S \times (B * (5 + C))$. Trong đó mỗi bounding sẽ gồm các thành phần:

$$[p_c, (b_x, b_y, b_w, b_h), (c_2, c_1, \dots, c_n)]$$

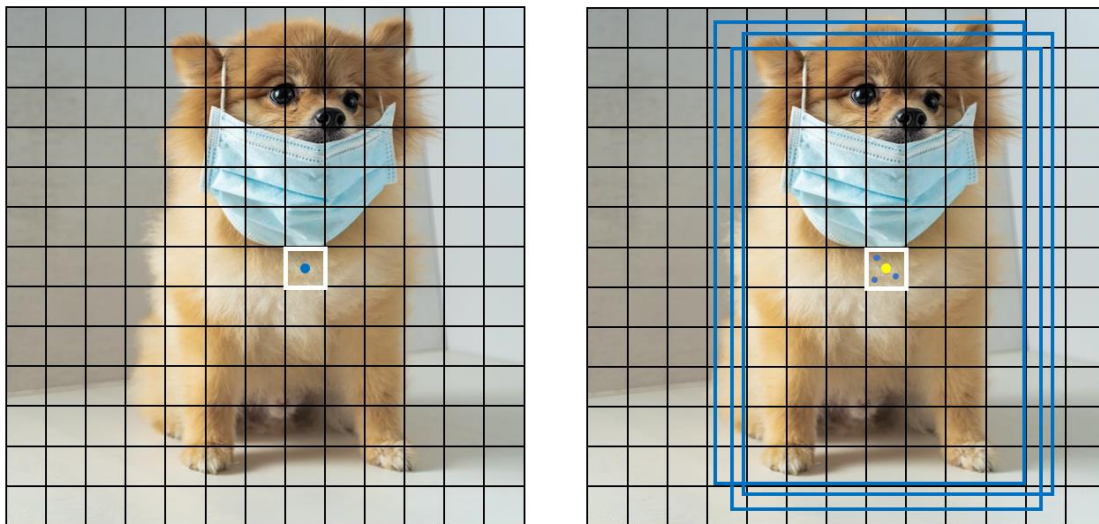
Tọa độ bounding box gồm 4 thành phần (b_x, b_y, b_w, b_h) :

- (b_x, b_y) là tọa độ tâm đối tượng so với grid cell
- (b_w, b_h) lần lượt là chiều rộng và chiều cao của bounding box

Độ tin cậy p_c được định nghĩa là $Pr(Object) * IOU_{pred}^{truth}$, là giá trị phản ánh xem một bounding box có chứa đối tượng hay không và độ chính xác của bounding box đó. Nếu không có đối tượng trong ô thì confidence score bằng 0. Ngược lại, nếu có đối tượng trong ô lưới thì confidence score sẽ bằng giá trị IOU giữa bounding box được dự đoán và ground truth bounding box.

Conditional class probability (c_1, c_2, \dots, c_n) là xác suất mà đối tượng được phát hiện trong box thuộc về các lớp nhất định, được định nghĩa là $Pr(Class_i|Object)$. Tùy vào bộ dữ liệu mà các giá trị xác suất có thể sử dụng logistic hoặc softmax. Khi sử dụng softmax mỗi đối tượng sẽ mang một nhãn thực, còn khi sử dụng các hàm logistic độc lập với nhau, một đối tượng có thể mang nhiều nhãn thực. Ví dụ một đối tượng có hai nhãn thực là “người” và “đàn ông”.

Ví dụ cụ thể dưới đây, ảnh đầu vào được chia thành lưới các ô vuông kích thước $S \times S$. Trong đó ô màu trắng sẽ chịu trách nhiệm dự đoán đối tượng do trọng tâm của đối tượng (dấu chấm vàng) rơi vào ô đó. Mỗi ô lưới sẽ dự đoán số lượng bounding box cố định. Như ví dụ, mô hình dự đoán 3 điểm trọng tâm và từ đó dự đoán 3 bounding box trong ô lưới màu trắng chứa trọng tâm đối tượng.



Hình 2.1: Ví dụ về grid cell và dự đoán bounding box

3. Cấu trúc của YOLOv3

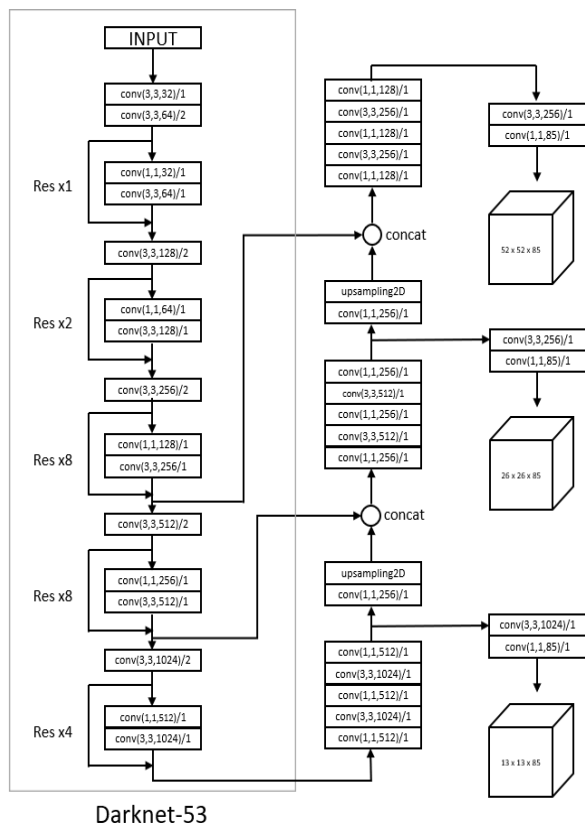
Để có thể dễ dàng hiểu rõ được cấu trúc mạng của YOLOv3, ta sẽ chia toàn bộ cấu trúc ra làm 2 phần chính: **trích xuất đặc trưng (Feature Extractor)** và **phát hiện đặc trưng (Feature Detector)**. Trước tiên, ảnh đầu vào được đưa qua lớp trích xuất đặc trưng để trích xuất ra những đặc trưng trên ảnh. Tiếp theo, lớp phát hiện đặc trưng sử dụng những đặc trưng đó để dự đoán ra những khung chứa (bounding box) và nhận dạng đối tượng.

3.1 Feature Extractor

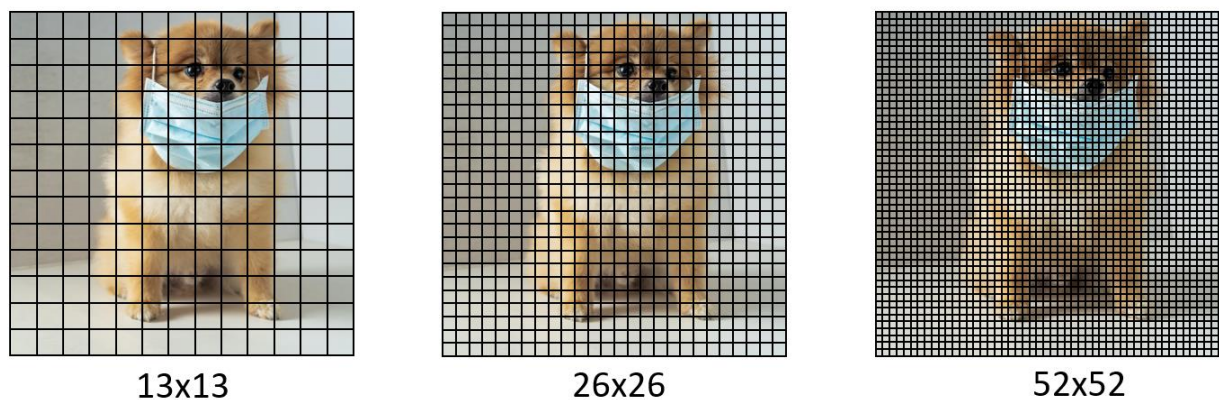
YOLOv3 sử dụng kiến trúc Darknet-53 cho lớp trích xuất đặc trưng (Feature Extractor). Darknet-53 bao gồm 53 lớp tích chập, trong đó có những lớp tích chập với bộ lọc (filter) có kích thước 3x3 và 1x1 xếp xen kẽ lên nhau. Khác với các phiên bản trước đó, YOLOv3 thay thế các lớp max-pooling bằng các lớp tích chập với hệ số trượt (stride) bằng 2. Lớp tích chập sẽ đóng vai trò như lớp max-pooling để giảm mẫu nhưng vẫn giữ lại các đặc trưng cấp thấp mà max-pooling thường bỏ qua, chính điều này sẽ giúp cho mô hình dự đoán tốt hơn với các đối tượng nhỏ. Thêm vào đó, các khối phần dư (residual block) cũng được dùng để tránh trường hợp tiêu biến đạo hàm (vanishing gradient) trong quá trình huấn luyện.

3.2 Feature Detector

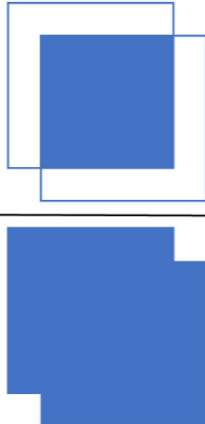
YOLOv3 sử dụng kiến trúc Feature Pyramid Networks (FPN) để đưa ra các dự đoán từ nhiều tỉ lệ khác nhau của ánh xạ đặc trưng (feature map). YOLOv3 cũng thêm các liên kết giữa các lớp dự đoán. Mô hình sẽ tăng mẫu (upsample) các lớp dự đoán trước đó và sau đó nối với các lớp ở Darknet-53. Việc này giúp YOLOv3 tận dụng các feature map với độ thô - tinh khác nhau cũng như tận dụng triệt để thông tin có trên ảnh cho việc dự đoán và tăng độ chính xác khi dự đoán các đối tượng nhỏ.



YOLOv3 sẽ dự đoán trên 3 feature map khác nhau. Các feature map nhỏ sẽ phát hiện và dự đoán vật thể có kích thước lớn, còn các feature map lớn sẽ dự báo được các đối tượng vừa và nhỏ ở trong ảnh. Mỗi feature map sẽ sử dụng 3 anchor box để dự đoán vật thể.



4. Intersection over Union (IOU)

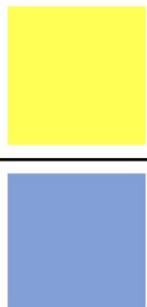
$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$


Hình 4.1: Intersection over Union (IoU)

Intersection over union (IOU) được sử dụng để đánh giá độ tương đồng giữa hai box được chọn. Thuật toán này được sử dụng trong nhiều bước tính toán của YOLOv3.

Area of Intersection là diện tích giao nhau giữa hai box cần tính. Area of Union là diện tích phân hợp nhau giữa hai box đó. Giá trị IoU càng cao thì hai box có độ tương đồng càng cao.

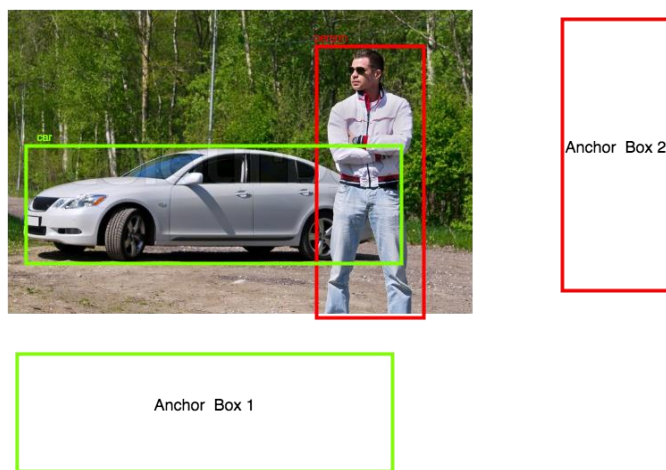


$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$


Hình 4.2: Ví dụ về Intersection over Union (IoU)

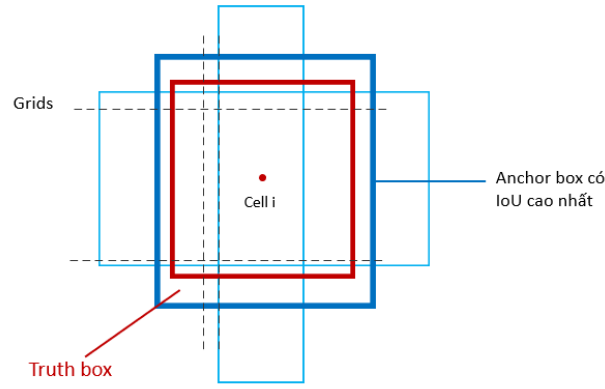
5. Khung neo (Anchor box)

Anchor box là những bounding box đã được xác định từ trước với kích thước cố định. Anchor box được thiết kế riêng cho từng bộ dữ liệu có sẵn dựa trên thuật toán phân cụm (K-means clustering). Anchor box được sử dụng từ việc các vật thể có một số bounding box tương đồng ví dụ như ô tô, xe đạp có bounding box dạng hình chữ nhật nằm ngang, người có bounding box dạng hình chữ nhật đứng. YOLOv3 có tổng cộng 9 anchor box, mỗi feature map sẽ sử dụng 3 anchor box.



Hình 5.1: Ví dụ về anchor box. Bounding box của vật thể “xe ô tô” có dạng hình chữ nhật nằm ngang, còn đối tượng “người” có dạng hình chữ nhật đứng

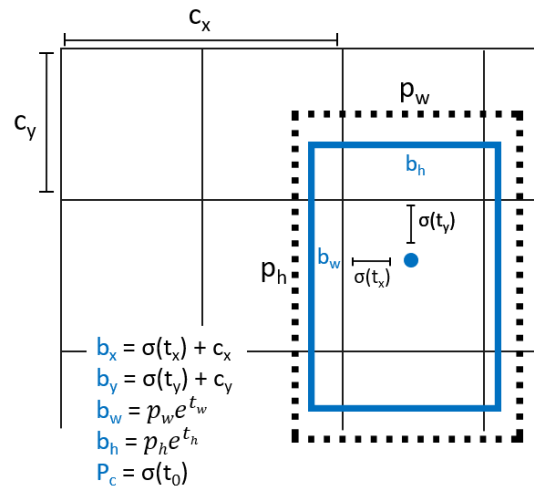
Để tìm được bounding box cho vật thể, YOLO sử dụng những anchor box làm cơ sở ước lượng. Với mỗi ground truth box, chúng ta sẽ dự đoán một bounding box dựa trên một anchor box, mà anchor box đó có giá trị IOU cao nhất với ground truth box. Bằng việc dự đoán dựa trên anchor box có sẵn, bounding box được dự đoán sẽ có kết quả ổn định hơn.



Hình 5.2: Xác định anchor box cho một vật thể. Từ ô lưới i ta có 3 anchor box. Cả 3 anchor box này đều giao nhau với ground-truth box của đối tượng. Tuy nhiên, chỉ anchor box có đường viền dày nhất màu xanh được lựa chọn làm anchor box cho vật thể bởi nó có IoU so với ground truth bounding box là cao nhất.

6. Dự đoán bounding box

Như đã đề cập trước đó, chúng ta sẽ tìm một anchor box có giá trị IoU cao nhất với ground truth box, từ đó tiến hành dự đoán bounding box dựa trên anchor box đó. Các bounding box sẽ được dựa trên những phép biến đổi từ anchor box và ô chứa đối tượng.



Hình 6.1: Minh họa về cách tính bounding box

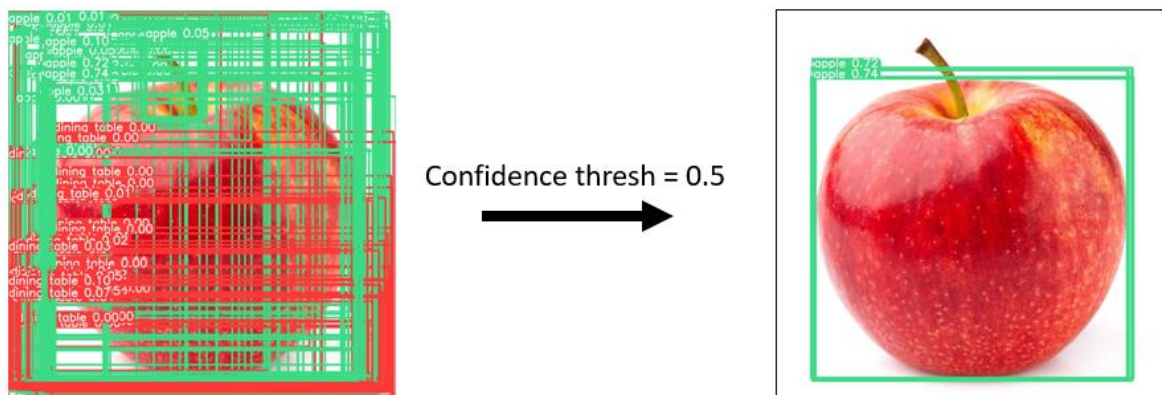
Trong đó:

- t_x, t_y, t_w, t_h : được dự đoán bởi YOLO
- c_x, c_y : độ lệch (offset) từ grid cell chứa trọng tâm đối tượng đến góc trái trên của ảnh
- p_w, p_h : kích thước anchor box
- b_x, b_y, b_w, b_h : các thông số của predicted bounding box
- $\sigma()$: hàm sigmoid với mục đích làm cho giá trị dự đoán chỉ nằm trong khoảng $[0,1]$

7. Non-max suppression

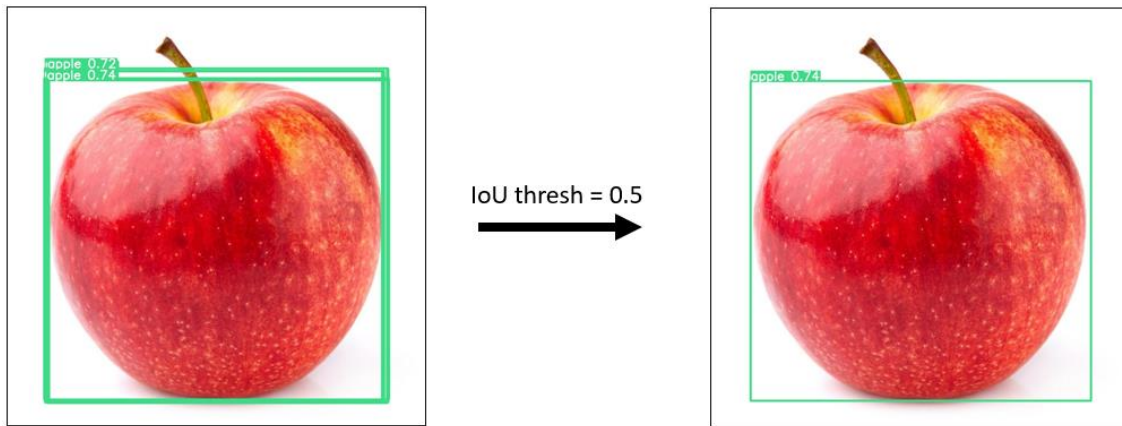
Với mỗi đối tượng trong ảnh, YOLO có thể dự đoán một hoặc nhiều trọng tâm cho một đối tượng, dẫn đến có rất nhiều bounding box dư thừa. Để có thể tìm được bounding box tốt nhất cho đối tượng, YOLO sẽ sử dụng một phương pháp gọi là non-max suppression (NMS). NMS là sẽ loại bỏ những bounding box dư thừa và chỉ giữ lại một bounding box tốt nhất cho mỗi đối tượng, NMS sẽ xem xét 2 yếu tố để lựa chọn bounding box: confidence score và IoU.

Đầu tiên, ta xác định một ngưỡng (threshold) confidence score nhất định để NMS có thể xóa đi một số bounding box (thông thường threshold sẽ là 0.5).



Hình 7.1: Ví dụ về kết quả bounding box sau khi lọc confidence score

Sau đó, NMS sẽ chọn ra những bounding box có confidence score cao nhất. Tiếp theo, NMS sẽ loại bỏ những box còn lại nếu những box đó có chỉ số IoU với box được chọn vượt ngưỡng threshold nhất định. NMS sẽ dừng lại khi số bounding box bằng với số đối tượng được phát hiện.



Hình 7.2: Ví dụ về kết quả bounding box sau khi lọc IoU score

8. Hàm độ lỗi (Loss function)

YOLOv3 sử dụng tổng bình phương độ lỗi và binary cross entropy giữa kết quả dự đoán với nhãn thực để tính toán giá trị độ lỗi. Loss function có thể được chia thành những phần sau:

- Lỗi phân loại nhãn (Classification loss)
- Lỗi dự đoán vị trí bounding box (Localization loss)
- Lỗi phát hiện đối tượng (Confidence loss)

$$L_{total} = L_{classification} + L_{localization} + L_{confidence}$$

8.1 Classification loss

Classification loss là độ lỗi phân loại của việc dự đoán loại nhãn thực của đối tượng. Hàm lỗi này chỉ tính trên những ô vuông có xuất hiện đối tượng, còn những ô vuông khác ta sẽ bỏ qua. Classification loss được tính bằng công thức sau:

$$L_{classification} = \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in class} [p_i(c) \log(\hat{p}_i(c)) + (1 - p_i(c)) \log(1 - \hat{p}_i(c))]$$

Với:

- $\mathbb{1}_i^{obj}$: bằng 1 nếu ô thứ i có chứa đối tượng, ngược lại bằng 0.
- $\hat{p}_i(c)$: xác suất mà mô hình dự đoán
- $p_i(c)$: nhãn thực của đối tượng, nếu đối tượng thuộc về lớp thứ c thì $p_i(c)$ sẽ bằng 1, ngược lại bằng 0

8.2 Localization loss

Localization loss là độ lỗi dự đoán vị trí bao gồm tọa độ tâm, chiều rộng, cao của bounding box bao quanh đối tượng (x, y, w, h). Trong đó, (x, y) là tọa độ tâm, (w, h) là chiều rộng và chiều cao của bounding box. Giá trị độ lỗi dự đoán tọa độ tâm bounding box dự đoán và ground truth box được tính như sau:

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

Giá trị độ lỗi dự đoán kích thước của bounding box dự đoán và ground truth box được tính như sau:

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

- $\mathbb{1}_{ij}^{obj}$: bằng 1 nếu bounding box thứ j của ô thứ i có chứa đối tượng, ngược lại bằng 0
- λ_{coord} : hằng số bằng 5.0 làm tăng giá trị hàm lỗi ở các vị trí có đối tượng

8.3 Confidence loss

Confidence loss là lỗi dự đoán của khung chứa đối tượng trên ảnh so với nhãn thực tại vùng đó

$$L_{Confidece} = \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [C_i^j \log(\hat{C}_i^j) + (1 - C_i^j) \log(1 - \hat{C}_i^j)]$$

$$+ \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} [C_i^j \log(\hat{C}_i^j) + (1 - C_i^j) \log(1 - \hat{C}_i^j)]$$

Trong đó:

- $\mathbb{1}_{ij}^{noobj}$: bằng 1 nếu bounding box thứ j của ô thứ i không chứa đối tượng, ngược lại bằng 0
- $\lambda_{noobject}$: hằng số bằng 0.5 làm giảm giá trị hàm độ lỗi ở các vị trí không có đối tượng
- \hat{C}_i^j : độ tin cậy của box thứ j của ô thứ i
- C_i^j : là giá trị thực tại ô thứ i, nếu ô đó chịu trách nhiệm dự đoán đối tượng thì C_i^j sẽ bằng 1, ngược lại bằng 0

9. Tổng kết

Khả năng nhận diện chính xác của mô hình YOLOv3 rất có ích trong nông nghiệp, công nghiệp và dịch vụ. Ở một số môi trường đòi hỏi không những độ chính xác cao mà còn cần tốc độ nhanh như giám sát, xe tự hành, phương tiện không người lái... thì YOLOv3 chính là một ứng cử viên sáng giá.

10. Tài liệu tham khảo

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, arXiv:1506.02640, 2016.

[2] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger,” arXiv:1612.08242, 2016.

[3] Joseph Redmon, Ali Farhadi, “YOLOv3: An Incremental Improvement,” arXiv:1804.02767, 2018.