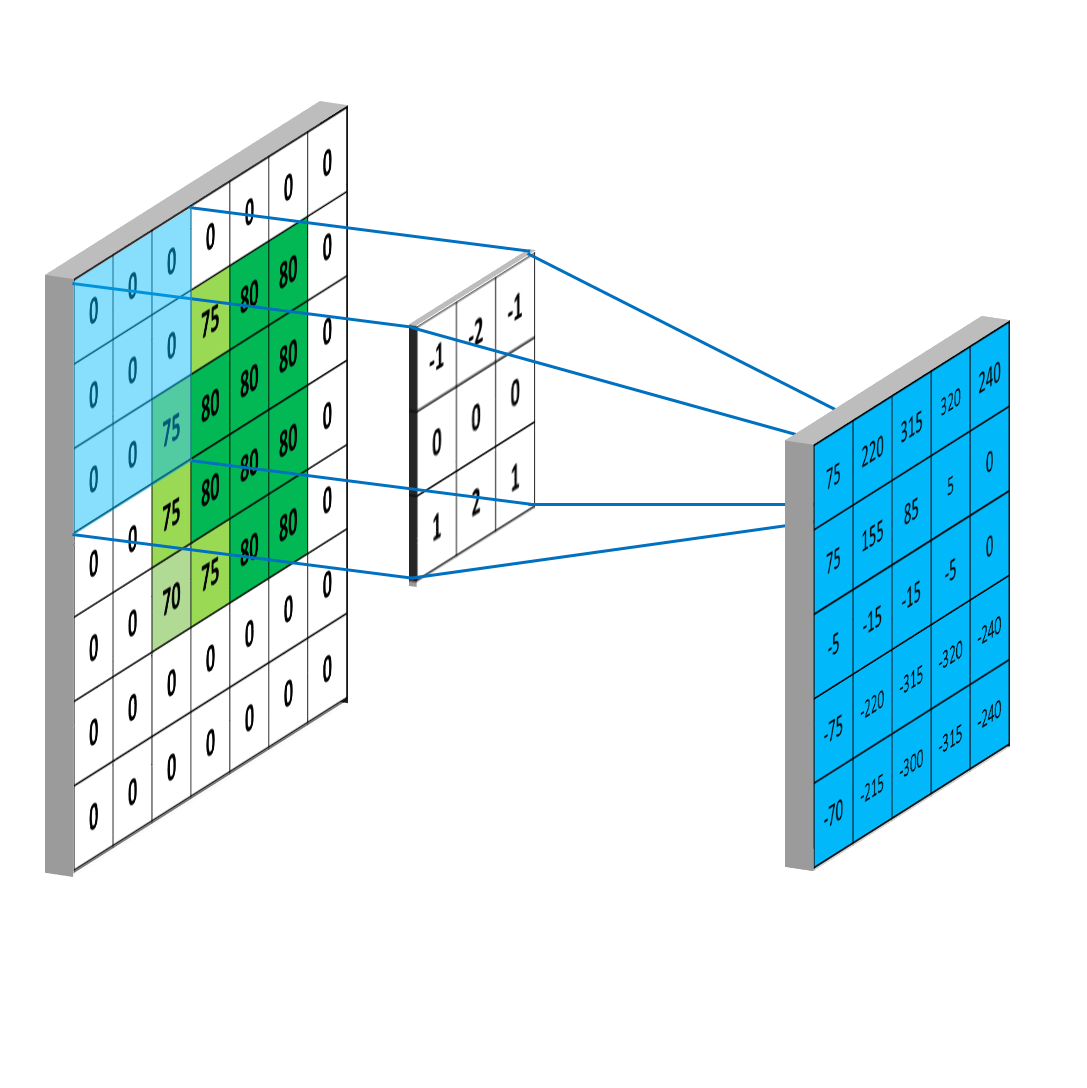
1. Định nghĩa

YOLO (You Only Look Once) một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng.

YOLO là thuật toán object detection nên ngoài việc có thể phân loại vật thể mà còn xác định được vị trí của vật thể

YOLO được coi là thuật toán nhanh nhất trong các lớp mô hình nên rất thích hợp xử lý trong khoảng real-time.

Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đóp các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.



1. Cách hoạt động của YOLO

Để quan tâm về YOLO hoạt động như thế nào thì ta cần quan tâm tới 4 vấn đề:

+ Grid System

+ Bounding box

+ CNN

+ Loss function

1. Grid System

* Chia ảnh gốc thành một lưới S x S (S là grid size), kích thước của ảnh phải là bội số của của S
* Mỗi ô chịu trách nhiệm dự đoán các bounding box nếu tâm của bb nằm trong ô, mỗi ô lưới chỉ dự đoán 1 đối tượng

Nếu nhiều đối tượng trong 1 ô thì ntn ?



1. Bounding Box (Ô biên)

Nói đơn giản là nó bao quanh một đối tượng mình xác định, nó có thể bao đúng hoặc sai hoặc thiếu.

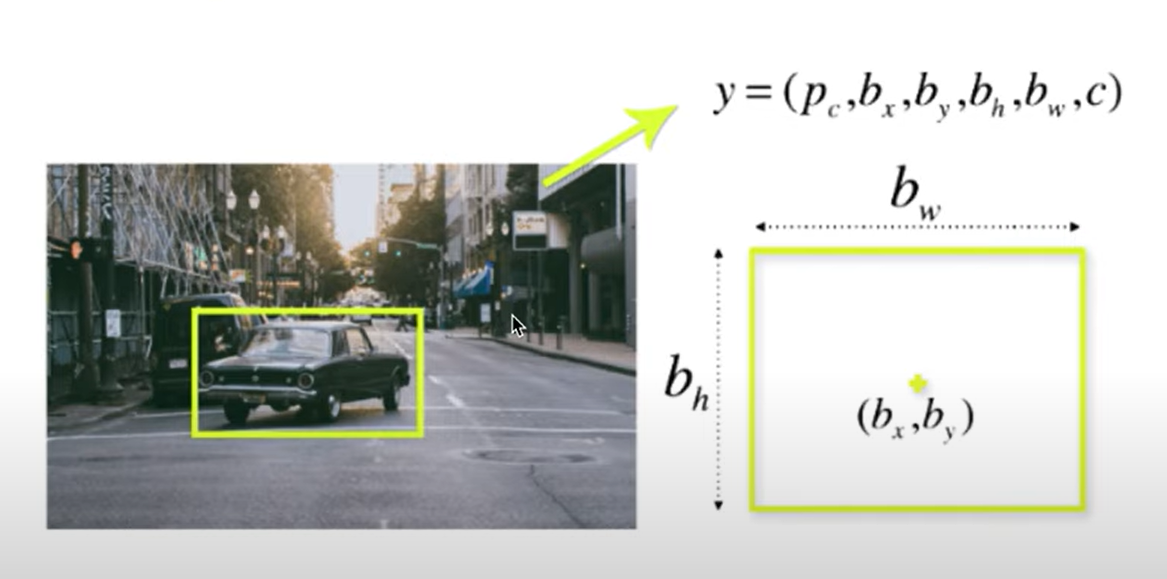
Mỗi hộp được giới hạn thường bao gồm 4 thuộc tính

+ (bx, by) : tọa độ tâm

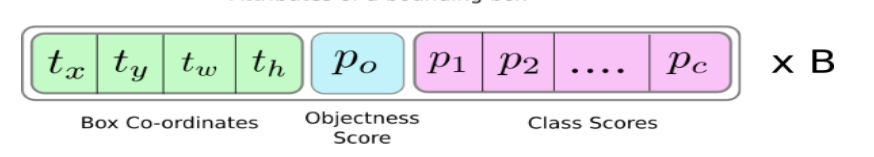
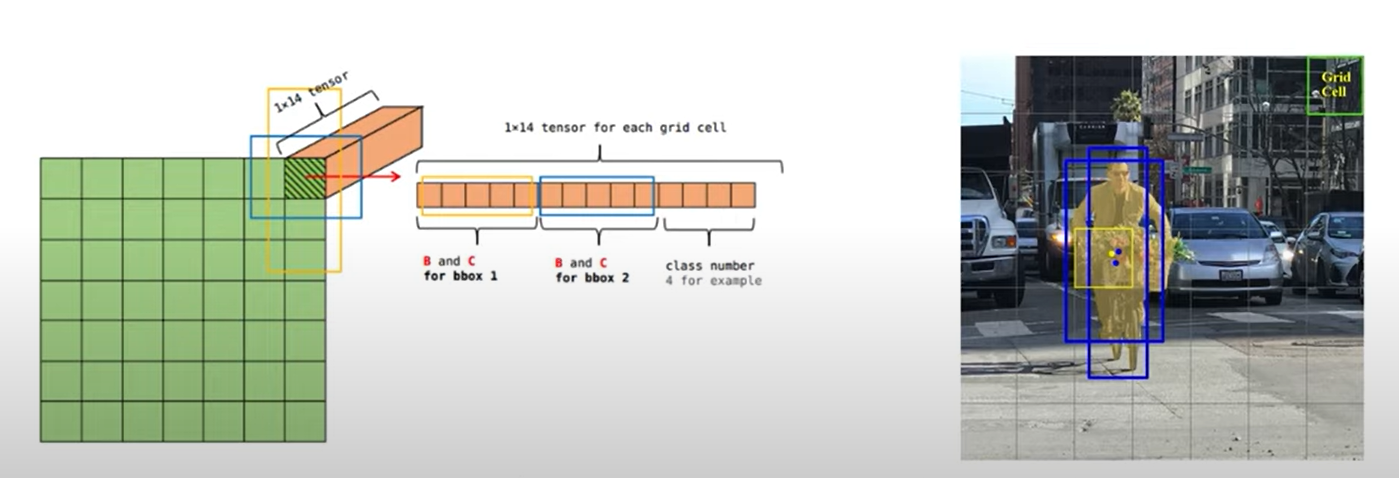
+ bh, bw : kích thước của bounding box

+ c : lớp, đối tượng của bb

+ pc : xác suất là đối tượng đó của bb

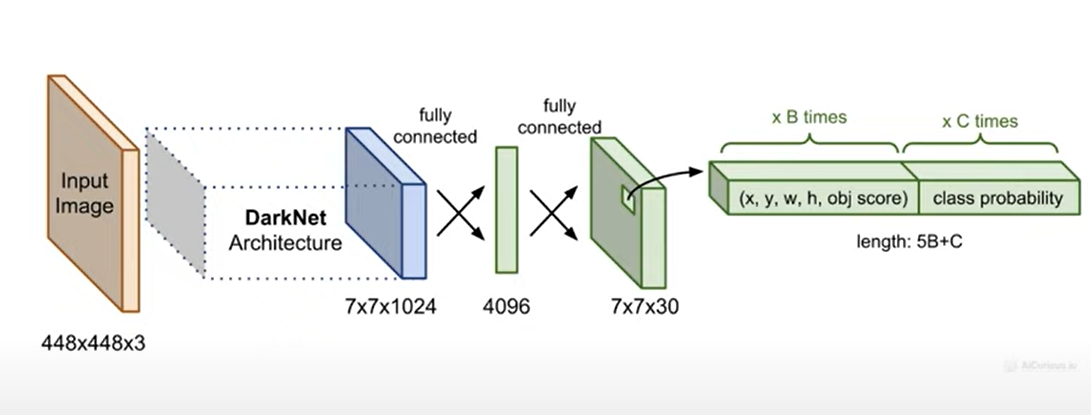


* Mỗi ô lưới dự đoán số bb cố định (tự cho sẵn)
* Mỗi ô lưới cần phải biết được có đối tượng hay không, độ lệch của bb tạo ra so với ô lưới hiện tại và lớp của cái đối tượng đó
* Số chiều của output: n\_box + 4\*n\_box + number\_classes\*n\_box



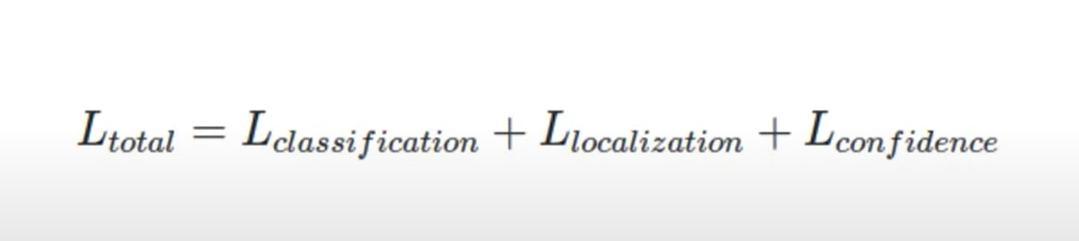
1. Kiến trúc CNN

* Sử dụng linear regression để dự đoán các thông tin ở mỗi ô vuông
* Layer cuối cùng không sử dụng hàm kích hoạt



1. Loss function (Hàm mất mát)

* Classification loss: Loss function cho dự đoán loại nhãn của object
* Localization loss: Loss function cho dự đoán tọa độ, chiều dài, rộng của bb
* Confidence loss: Loss function dự đoán box có chứa object nào hay không



Ngoài ra còn có non-max suppression khi có nhiều ô lưới chồng chéo nhau

Đọc thêm :

<https://phamdinhkhanh.github.io/2020/03/09/DarknetAlgorithm.html>

<https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P> (nói kĩ hơn về công thức)

<https://viblo.asia/p/yolov1-nhan-dien-vat-the-chi-ngo-1-lan-khoi-nguon-aNj4v0b0L6r> (nói kĩ hơn về mạng CNN trong YOLO)

1. Các bước tiền xử lý dữ liệu (image preprocessing)

Ảnh được chụp đang khá rõ nét và bắt sáng tốt nên cần cho nó những điều kiện khó khăn hơn để nó học được một cách tối ưu nhất.

Lưu ý: + Đừng chạy hết dữ liệu mà test qua 1, 2 tấm ảnh

+ Chia phần train – test ra làm 70-30 80-20

B1: Các dịch chuyển hình học như dịch chuyển phải, trái, lên, xuống, xoay ảnh, lật ảnh, biến đổi phối cảnh.

B2: Các phương pháp lọc nhiễu, làm mờ thông qua bộ lọc. (argumentation)

B3: Phương pháp phát hiện edge sử dụng bộ lọc canny hoặc ngưỡng threshold.

B4: Xác định các contours theo cường độ màu sắc tương đồng.

B5: Xác định bounding box của vật thể.

B6: Sử dụng non-max suppression để triệt tiêu các bounding box chồng lấn.

Tham khảo:

<https://phamdinhkhanh.github.io/2020/01/06/ImagePreprocessing.html>

<https://www.youtube.com/watch?v=g3I3ougTLAs&t=769s>