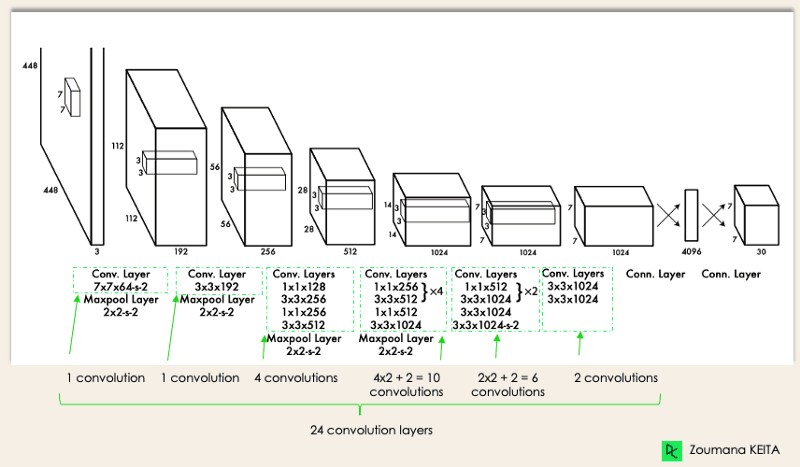
Cách YOLO hoạt động:

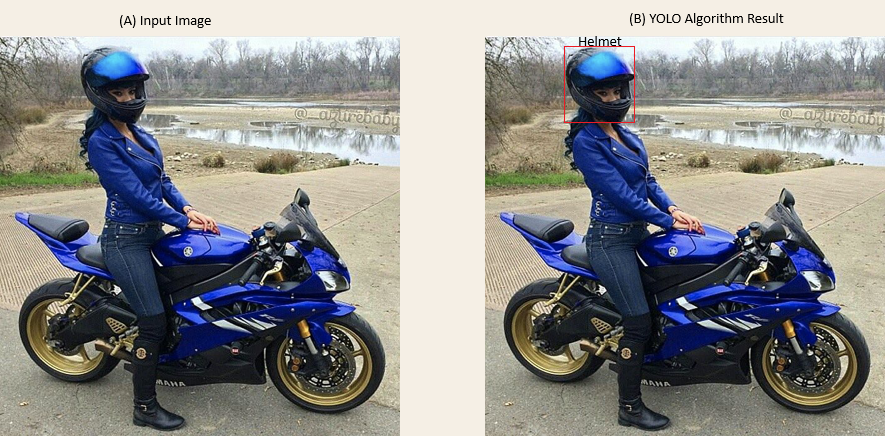


Như hình bên trên, ta thấy tổng cộng có 24 lớp tích chập, 4 lớp tổng hợp tối đa và 2 lớp kết nối đầy đủ, vậy theo cấu trúc này YOLO sẽ hoạt động theo trình tự:

* Thay đổi kích thước hình ảnh thành 448x448 trước khi vào mạng tích chập
* Tích chập 1x1 được áp dụng đầu tiên để giảm số lượng kênh, sau đó là tích chập 3x3 để tạo đầu ra hình khối
* Hàm kích hoạt trong mạng chủ yếu là ReLU ngoại trừ lớp cuối sử dụng hàm tuyến tính
* Một số kĩ thuật bổ sung ví dụ như chuẩn hóa hàng loạt và loại bỏ tương ứng sẽ chuẩn hóa mô hình và ngăn overfitting

Vậy YOLO phát hiện đối tượng như thế nào?

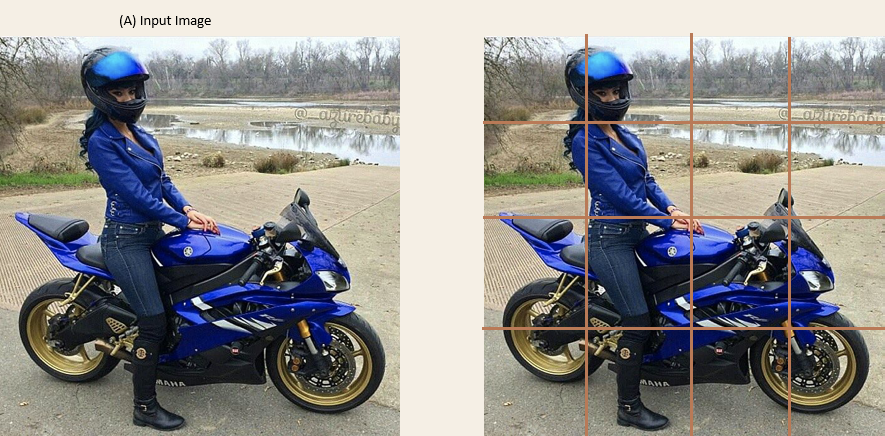
* Mục tiêu bài toán là xây dựng mô hình từ YOLO để phát hiện ra người đội mũ bảo hiểm



* Thuật toán này hoạt động dựa trên 4 phương pháp:
  + Residual blocks
  + Bounding box
  + IOU
  + Non-maximum Suppresion

1. Residual Block

Bước đầu tiên là chia hình ảnh (A) thành ô lưới NxN hình bằng nhau, với N trong trường hợp này mình sẽ tính là 4. Mỗi ô trong lưới chịu trách nhiệm định vị và dự đoán lớp của vật thể chứa trong đó, dựa trên giá trị probability/confidence.



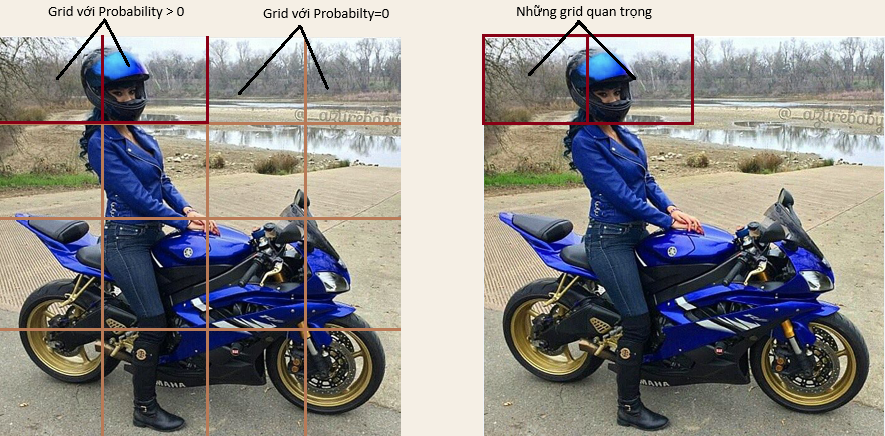
2. Bounding box

Bước tiếp theo là tính toán bounding box tương ứng với hình chữ nhật đánh dấu cho tất cả các vật thể ở trên ảnh. Ta có thể có bất kì số lượng bounding box tương ứng với số lượng vật thể trên ảnh.

YOLO sẽ xác định thuộc tính của các bounding box bằng cash sử dụng module hồi quy theo định dạng dưới đây, với Y là vector cuối biểu diễn cho mỗi bounding box

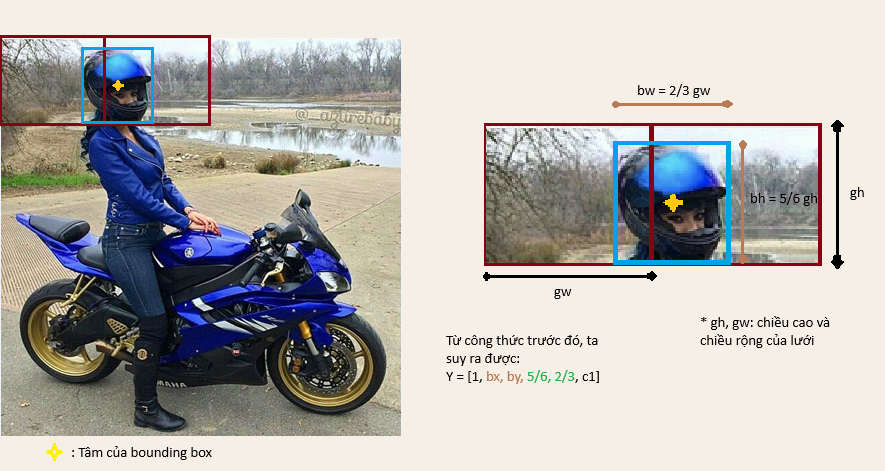
Y = [pc, bx, by, bh, bw, c1, c2]

* pc tương ứng điểm probability của lưới chứa một đối tượng. Ví dụ, tất cả lưới màu đỏ sẽ có điểm probability cao hơn 0. Ảnh bên phải là phiên bản đơn giản vì xác suất của các ô màu cam đều bằng 0 (không đáng kể)



* bx, by là những tọa độ x và y của trung tâm bounding box đối với ô lưới bao quanh
* bh, bw tương ứng là chiều cao và chiều rộng bounding box đối với ô lưới bao quanh
* c1, c2 tương ứng là 2 class Helmet và No Helmet (nếu có trong ảnh). Ta có thể có bao nhiêu class tùy thuộc vào trường hợp ảnh

Ta có thể xem ảnh dưới để hiểu rõ hơn:



3. IOU

Trong hầu hết tất cả các trường hợp, một đối tượng đơn lẻ trong một cảnh có thể nằm trong nhiều ô lưới để dự đoán, nhưng mà không phải tất cả chúng đều liên quan tới nhau. Mục đích của IOU ( 0 < IOU < 1) là để loại bỏ các hộp có lưới như vậy chỉ để giữ những ô có liên quan. Đây là logic của IOU:

* Người dung xác định ngưỡng IOU của mình, ví dụ là 0,5
* Sau đó YOLO tính toán IOU của mỗi ô lưới trong vùng giao nhau, chia cách với vùng hợp nhất.
* Cuối cũng nó bỏ qua dự đoán những ô có điểm IOU < ngưỡng và xem xét những ô có IOU > ngưỡng

Bên dưới là minh họa về quá trình áp dụng lựa chọn lưới cho hình ảnh. Ta có thể thấy đối tượng ban đầu nằm trong 2 lưới và cuối cũng chỉ có Grid 2 được chọn.



4. Non-Max Suppression

Việc đặt ngưỡng cho IOU không phải lúc nào cũng đủ vì đối tượng có thể có nhiều lưới có IOU vượt quá ngưỡng và việc xét tất cả các lưới đó sẽ gây ra nhiễu. Ta có thể sử dụng Non-Max Suppresion để giữ lại những hộp có điểm probability cao nhất.