Nhóm 13: Phát hiện vật thể trong ảnh

Nội dung YOLO v3 (You Only Look Once)

# YOLO v3 có thể

* Phát hiện nhiều vật thể trên một hình ảnh duy nhất
* Dự đoán các lớp của các vật thể
* Phát hiện vị trí của các vật thể này trên hình ảnh

# YOLO v3

* Áp dụng 1 mạng NN (Neural Network) cho toàn bộ hình ảnh
* Mạng NN chia hình ảnh thành các vùng
* Tạo ra xác suất cho mọi vùng
* Dự đoán một số hộp giới hạn bao phủ 1 số vùng trên hình ảnh & chọn những vùng tốt nhất theo xác suất

# Kiến trúc

* Bao gồm 53 CNNs layer (Darknet-53)
* Đối với các tác vụ phát hiện, kiến trúc gốc được xếp chồng lên nhau với 53 layer nữa
* Cung cấp 106 layer kiến trúc
* Bắt đầu bất kỳ lệnh nào trong khuôn khổ Darknet:
* Tải kiến trúc bao gồm 106 layer
* Các phát hiện được thực hiện ở 3 layer: 82, 94, 106

# Input

* Đầu vào của mạng: 1 loạt các hình ảnh có hình dạng sau (n, 416, 416, 3), trong đó:
  + n: 1 số hình ảnh
  + 416: chiều rộng
  + 416: chiều cao
  + 3: RGB
* Đầu vào của hình ảnh:
* Có thể thay đổi size trước khi đưa vào mạng
* Được thay đổi kích thước theo kích thước mạng đầu vào
* Tỷ lệ khung hình:
  + Thử nghiệm việc giữ hoặc không giữ tỷ lệ khung hình bằng cách điều chỉnh thông số khi train & thử nghiệm trong khung Darknet gốc, trong Tensorflow Keras hoặc bất kỳ khung nào khác mà ta muốn sử dụng
  + So sánh & chọn cách tiếp cận phù hợp nhất với mô hình tùy chỉnh của mình

# Cách mạng phát hiện các đối tượng

* YOLOv3 thực hiện phát hiện ở cả 3 quy mô khác nhau & ở 3 vị trí riêng biệt trong mạng (để phát hiện các lớp 82, 94 và 106)
* Hình ảnh đầu vào của các ví dụ về mạng bằng các yếu tố sau: 32, 16 & 8 (bước tiến của mạng) tại các vị trí riêng biệt đó của các mạng tương ứng cho biết đầu ra ở 3 vị trí riêng biệt của mạng nhỏ hơn đầu vào của mạng ntn

VD: Size mạng đầu vào 416x416 &

* bước tiến 32 => output (Feature maps) có kích thước 13x13 => phát hiện vật thể lớn
* bước tiến 16 => output có kích thước 26x26 => phát hiện vật thể vừa
* bước tiến 8 => output có kích thước 52x52 => phát hiện vật thể nhỏ

# Detection Kernels: giải thích các filter được train

* YOLOv3 áp dụng các detection kernels 1x1 tại 3 vị trí riêng biệt trong mạng
* YOLOv3 được đào tạo trên tập dữ liệu COCO có 80 lớp
* Phương trình: trong đó:
* số ô giới hạn của Feature maps đã tạo có thể dự đoán b = 3, YOLOv3 dự đoán 3 ô giới hạn
* mỗi hộp giới hạn có thuộc tính: tx ty là tọa độ tâm tw th là rộng, dài của hộp giới hạn, p0 là điểm vật thể (xác suất dự đoán vật thể xuất hiện trong hộp giới hạn), p0 là confidences (vectơ phân phối xác suất dự đoán của các lớp)
* c = 80 (COCO)
* (thuộc tính)
* Mỗi Featuer Maps được tạo ra bởi các detection kernels tại 3 vị trí riêng biệt trong mạng, có thêm 1 depth nữa kết hợp 255 thuộc tính của các hộp giới hạn cho tập dữ liệu COCO. Hình dạng của các Feature maps này như sau: 13x13x255, 26x26x255, 52x52x255

# Ô lưới

* YOLOv3 dự đoán 3 ô giới hạn cho mỗi ô của Featuer Maps -> mỗi ô dự đoán 1 vật thể thông qua 1 trong các ô giới hạn của nó nếu tâm của vật thể thuộc trường tiếp nhận của ô này
* Nhiệm vụ của YOLOv3 trong khi train là xác định ô này rơi vào tâm của vật thể -> là 1 trong những ô của Feature maps được tạo ra bởi các detection kernels
* Training YOLOv3:
* Có 1 hộp giới hạn cơ bản chịu trách nhiệm phát hiện 1 đối tượng
* Ô trung tâm được YOLOv3 chỉ định chịu trách nhiệm dự đoán vật thể này
* Điểm vật thể cho ô này = 1 -> là 1 trong những ô của Featuer Maps tương ứng chịu trách nhiệm phát hiện vật thể
* Trong quá trình train tất cả các ô dự đoán mỗi ô có 3 giới hạn

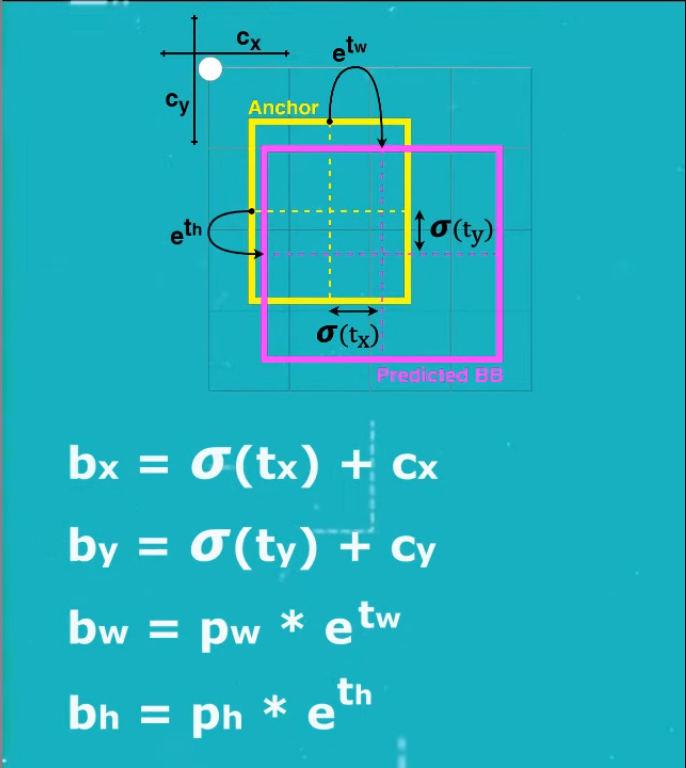
*#Trong 3 ô giới hạn chọn cái nào để dự đoán tốt nhất?*

# Anchor Box

* Để dự đoán các hộp giới hạn YOLOv3 sử dụng các hộp giới hạn mặc định được xác định trước được gọi là anchor/prior
* Các anchor được sử dụng để tính toán chiều rộng thực & chiều cao của hộp giới hạn được dự đoán
* Tổng có 9 anchor box được sử dụng, 3 anchor box cho mỗi thang đo
* Ở mỗi tỷ lệ, mỗi ô lưới của Featuer Maps có thể dự đoán 3 ô giới hạn bằng cách sử dụng 3 anchor
* Tính toán anchor, YOLOv3 áp dụng phân cụm k-means
* Chiều rộng và chiều cao cho tập dữ liệu COCO: chúng được nhóm lại theo tỷ lệ tại 3 vị trí riêng biệt tại mạng
* VD: Có hình ảnh input của hình dạng 416x416x3
* hình ảnh đi qua kiến trúc sâu CNN (mạng nơ ron tích tụ/ mô hình của Deep Learning) YOLOv3 cho đến vị trí riêng biệt đầu tiên & có bước tiến 32
* hình ảnh input được thu nhỏ thành kích thước 13x13 & 255 depth của Featuer Maps được tạo ra bởi các detection kernels
* có 3 anchor box nên mỗi ô mã hóa thông tin về 3 hộp giới hạn được dự đoán
* mỗi hộp giới hạn có các thuộc tính tx ty tw th, p0 , p1 p2 … pc & tập dữ liệu COCO có 80 lớp
* ô giới hạn xác suất:
* trích xuất xác suất trong số 3 ô giới hạn dự đoán của ô này để xác định ô này chứa 1 số lớp nhất định
* tính toán sản phẩm theo từng yếu tố của điểm vật thể & danh sách lớp
* tìm xác suất tối đa, vd hộp giới hạn 1 phát hiện ra lớp chanh với xác suất 0,55
* các phép tính này được áp dụng cho tất cả 13x13 ô trên 3 ô dự đoán & trên 80 lớp. Số ô dự đoán ở tỷ lệ đầu tiên này trong mạng là 507. Các tính toán này cũng áp dụng cho các tỷ lệ khác trong mạng cho ta 2028 & 8112 ô dự đoán. Tổng cộng YOLOv3 dự đoán 10 647 ô được lọc ra bằng kỹ thuật triệt tiêu không tối đa

# Tính toán hộp giới hạn được dự đoán

* Anchor là mồi nhữ của hộp giới hạn & được tính toán bằng cách sử dụng k-means clustering
* Đối với tập dữ liệu COCO, dự đoán chiều rộng thực & chiều cao thực của các hộp giới hạn:
* YOLOv3 tính toán hiệu số cho các anchor được xác định trước (phép biến đổi không gian log)
* Để dự đoán tọa độ trung tâm của các hộp giới hạn, YOLOv3 chuyển các kết quả đầu ra thông qua hàm sigmoid
* Phương trình được sử dụng để nhận dự đoán về tọa độ tâm, chiều rộng & chiều cao của hộp giới hạn:



bx by bw bh là tâm, chiều rộng, chiều cao của hộp dự đoán

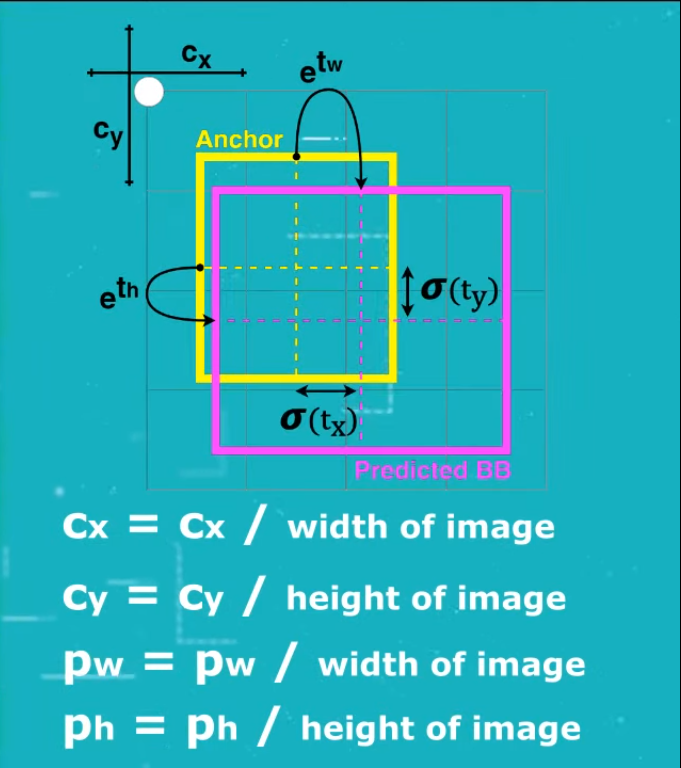
tx ty tw th là các kết quả output của mạng sau khi train

cx cy là tọa độ của góc trên cùng bên trái của ô trên lưới của anchor box thích hợp

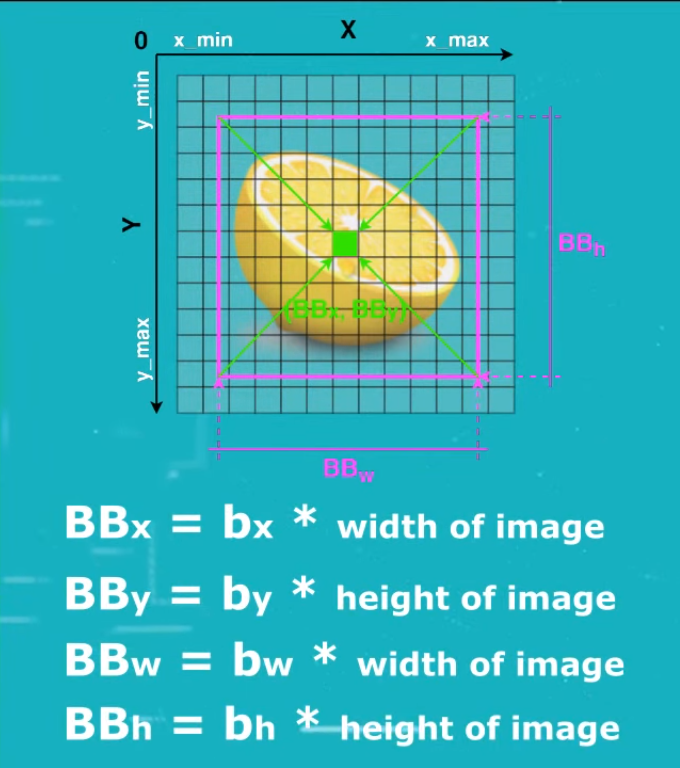
pw ph là chiều rộng, chiều cao của anchor

* Cách YOLOv3 train có 1 hộp giới hạn nền & 1 ô trung tâm chịu trách nhiệm cho vật thể. Chiều rộng của mạng được dự đoán chính xác nhất có thể tọa độ của ô trung tâm & ô giới hạn này. Sau khi train và chuyển tiếp mạng xuất ra tọa độ tx ty tw th
* YOLOv3:
* Không dự đoán giá trị tuyệt đối của chiều rộng & chiều cao => dự đoán hiệu số đối với anchor
* Giúp loại bỏ độ dốc không ổn định trong quá trình train => cx cy pw ph được chuẩn hóa

thành chiều rộng & chiều cao hình ảnh thực



tx ty được chuyển qua hàm sigmoid cho giá trị từ 0 đến 1. Nhận giá trị tuyệt đối sau dự đoán:



# Điểm vật thể (p0)

* Mỗi ô YOLOv3 sẽ xuất ra các hộp giới hạn với các thuộc tính của chúng (tx ty tw th p0) & 80 confidences cho mọi lớp mà hộp giới hạn này có thể thuộc về
* Kết quả output sẽ được sử dụng để chọn các anchor box bằng cách tính điểm, chiều rộng, chiều cao của hộp giới hạn dự đoán bằng các anchor đã chọn
* YOLOv3 khi train chỉ định ô trung tâm của hộp giới hạn chịu trách nhiệm dự đoán vật thể -> ô này & các ô lân cận có p0 gần bằng 1 còn các ô góc có p0 gần như bằng 0. Điểm vật thể biểu thị xác suất để ô này là ô trung tâm chịu trách nhiệm dự đoán một vật thể cụ thể & hộp giới hạn thích hợp chứa vật thể bên trong.
* Sự khác nhau giữa điểm vật thể với các xác suất dự đoán của 80 lớp (p1 p2 … pc):
* p1 p2 … pc đại diện cho xác suất mà vật thể được phát hiện thuộc về 1 lớp cụ thể như person, cat, lemon,…
* p0 biểu thị xác suất hộp giới hạn chứa vật thể bên trong

# Các ý chính về cách hoạt động của YOLOv3

* YOLOv3 áp dụng CNN (mạng nơ ron tích tụ) cho hình ảnh input
* Để dự đoán các hộp giới hạn: lấy ví dụ hình ảnh tại 3 vị trí riêng biệt của mạng này: 82, 94, 106 (còn gọi là thang đo)
* Khi train sử dụng từng kernels detection được áp dụng cho lưới các ô tại 3 vị trí riêng biệt này của mạng
* Mạng được train để chỉ định 1 ô chịu trách nhiệm phát hiện 1 vật thể nếu ô đó rơi vào trung tâm của vật thể này
* 9 hộp giới hạn xác định trước (anchor) được sử dụng để tính toán kích thước không gian & tọa độ của các hộp giới hạn dự đoán. 3 anchor cho mỗi thang đo
* Tổng cộng YOLOv3 dự đoán 10 647 ô giới hạn được lọc bằng kỹ thuật triệt tiêu không tối đa chỉ để lại những ô phù hợp