TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐHQG TPHCM

KHOA CÔNG NGHỆ THÔNG TIN (CLC)

REPORT

YOLO TRONG OBJECT DETECTION

Chuyên đề Thị giác máy tính

GVHD: Thầy Võ Hoài Việt

HUỲNH THỊ MỸ THANH

MSSV: 19127273

Mục lục:

I. Giới thiệu chung

Trong lĩnh vực thị giác máy tính, bài toán Object detection đang được ứng dụng rất nhiều trong các thiết bị thông minh trên thế giới. YOLO có khả năng xác định các đối tượng trong hình ảnh tức là YOLO sẽ đưa ra bounding boxes xác định vị trí và kích thước của đối tượng đó. Trong ngành công nghiệp Object detection thường được dùng để phát hiện những sản phẩm lỗi hay không giống mẫu, hay trong việc giám sát thì có những camera giám sát thông minh... Với nhu cầu và tính cần thiết của bài toán này, mà nhiều nhà nghiên cứu đã và đang phát triển các thuật toán để làm cho việc phát hiện vật thể trở nên nhanh và chính xác hơn.

YOLO xuất hiện vào năm 2016 với tác giả chính là ông Joseph Redmon và đã tạo sự chú ý đáng kể từ cộng đồng nghiên cứu nhờ vào việc dễ sử dụng (end-to-end) và tốc độ chạy nhanh (real-time). Sau đó YOLO đã phát triển thêm 2 version mạnh hơn là YOLO-v2 vào năm 2017 và YOLO-v3 vào năm 2018. Ngoài ra sau này còn có các nhà nghiên cứu khác cũng đã cải tiến và phát triển theo mục tiêu của các tác giả, và hiện chúng ta có: YOLO-v4, YOLO-v5, YOLO-v7 ...

II. Thách thức

Trước khi nói về những thách thức hiện đang tồn tại, song song đó vẫn là rất nhiều ưu điểm đến từ YOLO dành cho bài toán Object Detection, sau đây là những điểm chính mà YOLO đã mang lại:

- Tốc độ xử lý nhanh: Tốc độ xử lý đáp ứng được thời gian thực, phát hiện đối tượng và xác định vị trí của chúng một cách nhanh chóng.
- Độ chính xác khá tốt.
- Dễ dàng triển khai mô hình.

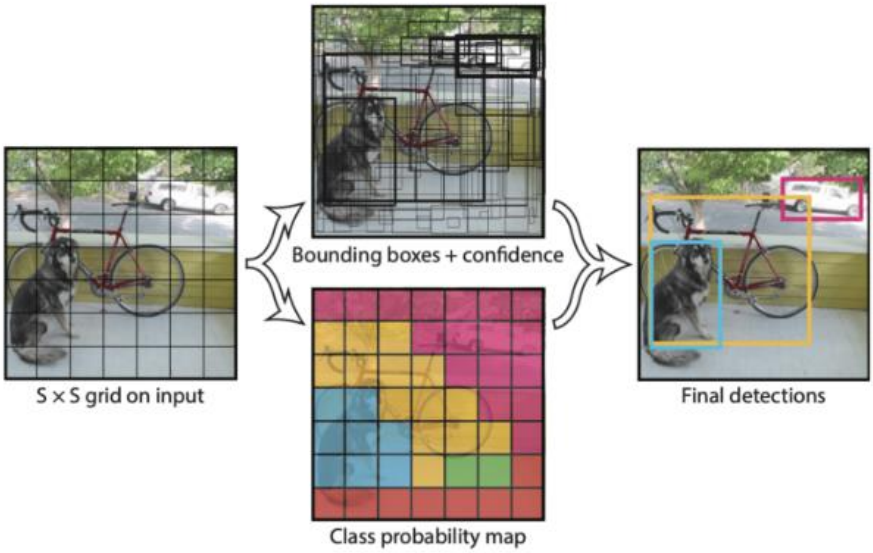
Mặc dù YOLO có rất nhiều ưu điểm và được sử dụng rộng rãi trong nhiều ứng dụng phát hiện đối tượng, nhưng vẫn phải đối mặt với một số thách thức:

- Tuy YOLO có độ chính xác khá tốt nhưng nó vẫn chưa thể sánh bằng được với các phương pháp như Fast R-CNN.
- YOLO thường bỏ sót những đối tượng nhỏ trong hình ảnh (những đối tượng có kích thước rất nhỏ so với kích thước hình ảnh) và gặp khó khăn bởi việc phát hiện những đối tượng bị che khuất.
- Thời gian training khá dài.

III. Hệ thống detection của YOLO:

1. Phát biểu bài toán

Tìm hiểu sâu bên trong YOLO-v1.



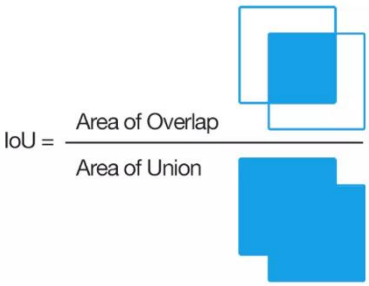
Input: YOLO chia ảnh đầu vào thành kích thước S x S ô lưới. Nếu tâm của object nằm trong 1 ô lưới nào đó, thì ô đó có nhiệm vụ phát hiện đối tượng đó.

Output: Là một ma trận 3 chiều có kích thước S x S x (5 x N + M) với số lượng tham số của mỗi ô là (5 x N + M) với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán. Như vậy, mỗi một ô có nhiệm vụ dự đoán bounding box và confidence cho các box đó. Mỗi bounding box dự đoán sẽ chứa 5 thông tin box(x,y,w,h) và confidence, trong đó:

- (x,y) là tọa độ trọng tâm của đối tượng
- (w,h) chiều rộng, chiều cao của bounding box
- Confidence: độ tin cậy. Được tính bằng cách so sánh thông tin predict ra và thông tin được label. Công thức:

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

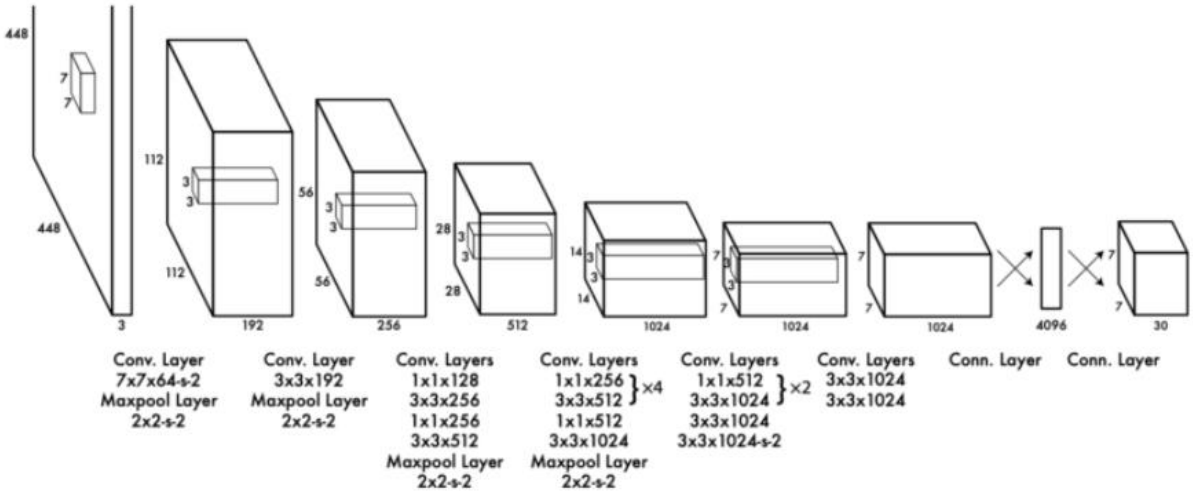
- Pr(Classi| Object): xác suất của mỗi class cho ô.
- Pr(Object): xác suất có vật thể trong ô.
- IOU: phần giao nhau của bounding box predict và bounding box được label.



Lưu ý: trong 1 ô lưới thì chỉ có tâm của 1 object, ví dụ 2 object cùng có tâm trên 1 ô lưới, thì chỉ phát hiện được 1 object có score cao nhất. Đây cũng là điểm yếu của YOLO v1, vì chỉ có tối đa 7×7=49 object được phát hiện trong 1 ảnh.

2. Network design

YOLO đã thiết kế một network có 24 layers convolutional để trích xuất feature trên mỗi ảnh đầu vào, và tiếp theo sau là 2 layer fully connected có nhiệm vụ dự đoán ra xác suất và tọa độ của từng object có trong ảnh.



Về cơ bản, YOLO chứa 3 loại layer: Convolutional, Maxpool, và Fully Connected. Tác giả của YOLO đã kế thừa kiến trúc của GoogleNet, nhưng thay thế inception modules của GoogleNet bằng module chứa 1×1 convolutional layer kết hợp với 3×3 convolutional layer (ý tưởng dựa trên paper Network In Network), mục đích của 1×1 convolutional layer là để giảm số chiều của feature.

Với mục đích nhanh hơn nữa, một biến thể của của YOLO v1 dùng để tăng tối đa tốc độ cho việc phát hiện đối tượng là Fast YOLO cũng được giới thiệu. Khác với YOLO thì Fast YOLO chỉ sử dụng 9 layer convolutional thay vì 24 layer convolutional, kết hợp với dùng filter size nhỏ hơn đã giúp tốc độ infer của Fast YOLO đạt được 155 fps.

Trên các layer convolutional sử dụng Leaky ReLU để làm activation function trên toàn bộ network, ngoại trừ lớp cuối cùng cần phân loại thì network sử dụng linear activation function.

3. Loss function

Loss function của YOLO được định nghĩa theo công thức :

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned}
$$

YOLO sử dụng Sum-Squared Error (SSE) để tính loss của giá trị predict và ground truth. SSE giúp việc tối ưu dễ dàng hơn, nhưng việc này đồng thời cũng không đúng với mục tiêu là tối đa hóa việc trung bình độ chính xác, vì thế Tác giả còn thêm 2 trọng số vào hàm loss là λcoord và λnoobj . Cụ thể loss của YOLO là tổng của 5 loss thành phần, theo thứ tự từ trên xuống:

- Loss của tâm (x,y) của bbox được predict và bbox được label (ground-truth),
- Loss của chiều dài và rộng (h,w) của bbox được predict và bbox được label (ground-truth),

- Loss của confidence score khi có object trong ô (cell),
- Loss của confidence score khi không có object trong ô (cell),
- Loss của xác suất tại ô có object.

IV. Các version YOLO và họ R-CNN:

1. Họ R-CNN:

RCNN và các biến thể của nó (Fast R-CNN, Faster R-CNN) được coi là phương pháp đạt hiệu suất tốt nhất trong các phương pháp dựa trên mạng neural. RCNN cung cấp một bước phát hiện đối tượng rất chính xác và có khả năng phát hiện các đối tượng nhỏ và phức tạp. Tuy nhiên, RCNN có thể yêu cầu phải huấn luyện trên một tập dữ liệu lớn và sử dụng phần cứng mạnh để đạt được hiệu suất tốt. Dưới đây là bảng so sánh một số mặt giữa RCNN và YOLO.

	RCNN	YOLO
Phương pháp tiếp cận	Region proposal	End to end
Tốc độ xử lý	Chậm hơn	Nhanh hơn
Số lượng tham số	Nhiều	Ít
Độ chính xác	Cao hơn	Thấp hơn

Sự chọn lựa giữa RCNN và YOLO phụ thuộc vào các yêu cầu cụ thể của bài toán như tốc độ, độ chính xác, số lượng tham số...

2. Các version YOLO (v1-v3):

YOLOv1:

Mô hình YOLO ban đầu có 20 lớp và được huấn luyện trên tập dữ liệu VOC 2012 với độ chính xác khoảng 63.4%, có tốc độ xử lý chậm nhất trong các version.

Điểm yếu của YOLOv1:

- Độ chính xác kém hơn so với các mô hình Region-based detector (kỹ thuật sử dụng two-stage).
- Recall thấp, khả năng phát hiện bị thiếu/sót các đối tượng.
- Dự đoán tối đa một object trong 1 cell, tối đa 49 object.

YOLOv2:

Có một số cải tiến như:

- Batch Normalization(BN): thêm BN vào các lớp convolutional để tối ưu hóa quá trình huấn luyện.
- Sử dụng kỹ thuật anchor box (cho phép mô hình dự đoán nhiều bbox thay vì duy nhất 1 bbox).
- YOLOv2 cũng sử dụng mô hình Darknet-19 với một số cải tiến.

Với các cải tiến này, YOLOv2 đã tăng mAP từ 63.4 lên 78.6 so với YOLOv1 (độ chính xác và sai sót ít hơn).

YOLOv3:

YOLOv3 đã thực hiện một loạt các thay đổi nhỏ về thiết kế để làm cho hệ thống chạy tốt hơn, với một số thay đổi:

- Sử dụng Darknet-53 (học được các đặc trưng phức tạp hơn so với Darknet-19).
- Sử dụng kỹ thuật Multi-scale prediction để tăng khả năng nhận dạng đối tượng có kích thước khác nhau trong cùng một ảnh.

Với những thay đổi này, YOLOv3 đã chạy nhanh hơn đáng kể so với các phương pháp khác với hiệu suất tương đương.

Sau khi phát hành YOLOv3, Joseph Redmon không còn nghiên cứu thị giác máy tính. Các nhà nghiên cứu như Alexey Bochkovskiy và Glenn Jocher đã tiếp tục và phát triển lên YOLOv4 và YOLOv5.

V. Kết luận

- YOLO là một trong những phương pháp object detection nhanh nhất hiện nay, đặc biệt là trên các ứng dụng thời gian thực. Tốc độ xử lý của YOLO vượt trội so với các phương pháp truyền thống như Faster R-CNN và RetinaNet.
- YOLO hiện tại đã có khả năng phát hiện đa hướng và các đối tượng nhỏ hơn, nhờ phương pháp phân tích hình ảnh toàn cục (global image analysis) và việc sử dụng anchor box có kích thước đa dạng.
- Tuy nhiên, độ chính xác của YOLO vẫn chưa thể so sánh được với các phương pháp state-of-the-art khác như RetinaNet và EfficientDet, đặc biệt là trên các bộ dữ liệu yêu cầu độ chính xác cao.

- Các phiên bản YOLO mới như YOLOv4 và YOLOv5, YOLOv6 đã cải thiện đáng kể độ chính xác và hiệu suất so với phiên bản ban đầu.
- Trong tương lai, có thể YOLO sẽ tiếp tục được cải tiến và phát triển với các tính năng mới nhằm đạt được độ chính xác cao hơn và tốc độ xử lý nhanh hơn, đồng thời có thể sẽ được sử dụng rộng rãi trong các ứng dụng thực tế như xe tự hành, hệ thống an ninh, và các sản phẩm liên quan đến thị giác máy tính.

VI. Tài liệu tham khảo

- https://arxiv.org/pdf/1506.02640.pdf
- https://arxiv.org/pdf/1612.08242.pdf
- https://arxiv.org/pdf/1804.02767.pdf
- https://arxiv.org/pdf/2004.10934.pdf
- YOLO You Only Look Once