SO SÁNH THUẬT TOÁN SSD VÀ YOLO TRONG PHÁT HIỆN ĐỐI TƯỢNG

COMPARE SSD ALGORITHM AND YOLO IN OBJECT DETECTION

TRÀ VĂN ĐỒNG(*), NGUYỄN THU NGUYỆT MINH(**) và HUỲNH CHÍ NHÂN(**)

TÓM TẮT: *Phát hiện đối tượng có thể chia thành hai nhóm là: 1) Phát hiện một đối tượng cụ thể và 2) Phát hiện chủng loại đối tượng. Hầu hết các phương pháp đều dựa trên họ R-CNN (Regions with Convolutional Neural Network Family) như R-CNN, Fast R-CNN [4], Faster R-CNN [2], Mask R-CNN,... gồm một chuỗi tiến trình nhiều lớp xen kẽ nhau rất phức tạp và chi phí cao. Năm 2016, Joseph Redmon và đồng sự đề xuất phương pháp phát hiện đối tượng YOLO (You Only Look Once) [1] và Wei Liu và đồng sự đề xuất phương pháp phát hiện đối tượng SSD (Single Shot Detector) [3] dựa trên cách tiếp cận khác.*

Từ khóa: *phát hiện đối tượng; deep learning; mạng neuron tích chập CNN; YOLO; SSD.*

ABSTRACT: *Object detection can be divided into two groups: 1) detecting a specific object, and 2) detecting categories of the object. Most of methods based on R-CNN family (Regions with Convolutional Neural Network Family) such as R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN,... it comprises of a sequence of processing of alternated layers which is very complex and expensive. In 2016, Joseph Redmon et al. proposed a method of object detection named YOLO (You Only Look Once), and Wei Liu et al. proposed a method of object detection named SDD (Single Shot Detector) based on a different approach.*

Key words: *object detection; deep learning; convolutional neural network; YOLO; SSD.*

1. ĐẶT VẤN ĐỀ

Thuật toán SSD và YOLO đều thuộc nhóm single shot detectors. Cả hai đều sử dụng convolution layer để rút trích đặc trưng và một convolution filter để đưa quyết định và đều dùng feature map có độ phân giải thấp (low resolution feature map) để dò tìm đối tượng, chỉ phát hiện được các đối tượng có kích thước lớn. Phát hiện đối tượng với mục tiêu là phát hiện đối tượng có hay không trong một hoặc nhiều ảnh, định vị đối tượng đó trong ảnh, là một trong những bài toán cơ bản và thử thách nhất trong thị giác máy tính. Các kỹ thuật deep learning [8] phát triển gần đây được xem như là những phương pháp học khá hữu hiệu các đặc trưng được rút trích trực tiếp từ dữ liệu.

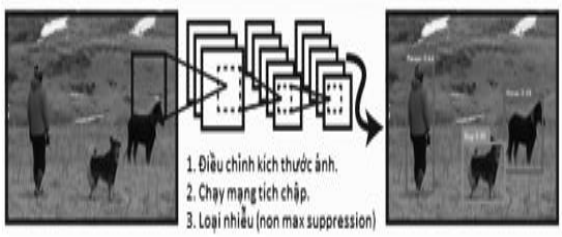
Khi chúng ta muốn phát hiện ra object trong một bức ảnh, sau đó đánh nhãn cho object đó, các phương pháp cũ quá chậm để phục vụ trong real-time, hoặc đòi hỏi thiết bị phải mạnh cho đến khi YOLO và SSD ra đời, có khả năng gán nhãn cho toàn bộ object trong khung hình với chỉ duy nhất một operation và mô hình sử dụng một mạng neural duy nhất. Có thể nói YOLO, SSD đã xây dựng một hướng tiếp cận đầu tiên giúp đưa Object detection thực sự khả thi trong cuộc sống. Trong bài viết này, chúng tôi xin trình bày hai thuật toán nói trên để làm rõ hơn vấn đề tìm kiếm đối tượng.

(*) ThS. Trường Đại học Văn Lang, *dong.tv@vlu.edu.vn*
(**) ThS. Trường Đại học Văn Lang, *Mã số: TCKH22-10-2020*

2. NỘI DUNG

2.1. Thuật toán YOLO [5]

YOLO là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đó, các convolutional layers sẽ trích xuất ra các feature của ảnh; full-connected layers sẽ dự đoán ra xác suất và tọa độ của đối tượng. YOLO được đề xuất từ ý tưởng con người có thể phát hiện và định vị được một đối tượng nào đó khi đã nhìn qua một lần. Do đó kiến trúc một hệ thống phát hiện đối tượng YOLO tương đối đơn giản như minh họa trong hình 1.



Hình 1. Hệ thống phát hiện đối tượng YOLO đơn giản

Ghi chú: 1) Ảnh được điều chỉnh thành ảnh có kích thước phù hợp (chẳng hạn 448x448); 2) Chạy một mạng tích chập để rút trích đặc trưng; 3) Kết quả phát hiện dựa trên ngưỡng rút ra từ độ tin cậy của mô hình, có thể xử lý và phát hiện đối tượng 45 frames/s với độ chính xác cao.

2.1.1. Vector dự đoán (The Predictions Vector)

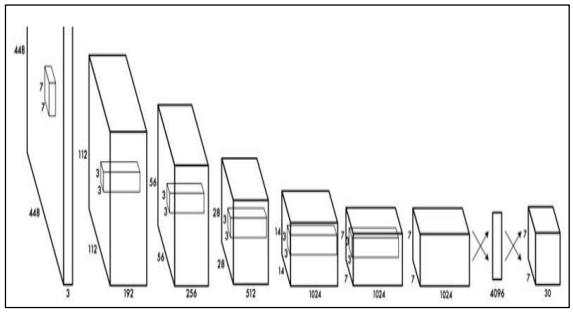
Đây là vector đầu ra của YOLO. Ảnh đầu vào được chia thành một lưới gồm *SxS* ô. Với mỗi đối tượng là một ô, một ô trong lưới được xem là ứng viên để dự đoán đối tượng. Mỗi ô như vậy dự đoán cho *B* bounding box (gọi tắt là box) và *C* xác xuất cho lớp của đối tượng. Mỗi box có 5 thành phần: *(x,y,w,h,conf).* Trong đó *(x,y)* là tọa độ tương đối giữa tâm của box so với ô (điều đó có nghĩa là nếu tâm của box không roi vào trong ô thì ô đó không được xem là ô ứng viên). Các tọa độ này được chuẩn hóa [0,1]. *(w,h)* là chiều rộng và chiều dài tương đối của box và cũng được chuẩn hóa [0,1]. Thành phần *conf* được gọi là độ tin cậy của box. *conf* được tính như sau:

$$conf = Pr(Object) * IOU(pred, truth) \quad (1.1)$$

Trong đó: IOU là chỉ số IOU [7] (Intersect Over Union). Nếu đối tượng không có trong box thì *conf* = 0, ngược lại *conf = IOU(pred, truth).* Điều này có nghĩa là *Pr(Object)* = 0 nếu đối tượng không có trong box và *Pr(Object)* = 1 nếu đối tượng có trong box. Chỉ số *conf* phản ánh có hay không một đối tượng thuộc một chủng loại (lớp) trong box.IOU là tỷ lệ giữa diện tích vùng giao nhau và diện tích vùng hợp nhất của 2 box, thông thường nếu IOU > 50% thì được xem là box dự đoán đó có đối tượng. Trường hợp muốn dự đoán xác xuất đối tượng thuộc chủng loại nào thì sử dụng thêm xác xuất có điều kiện *Pr(Class(i) | Object).*

2.1.2. Mạng neuron tích chập

YOLO xây dựng một mạng CNN để dự đoán các tensor kích thước (7,7,30). Mạng CNN này có tác dụng làm giảm kích thước không gian mỗi vị trí thành 7 x 7 với 1024 một kênh đầu ra. Hình 2 minh họa kiến trúc một mạng YOLO. Kiến trúc mạng CNN trong YOLO có 24 lớp tích chập kết hợp với các lớp max pooling và 2 lớp fully connected. Lần lượt mỗi lớp tích chập sẽ giảm kích thước không gian đặc trưng từ lớp trước đó.



Hình 2. Kiến trúc mạng CNN trong YOLO

Bảng 1. Liệt kê lớp trong mạng CNN (gọi là mạng CNN đầy đủ - *full CNN*) của một hệ thống YOLO

Tên	Bộ lọc	Kích thước đầu ra
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
..	..	..
Conv 15	1 x 1 x 512	28 x 28 x 512

Một mạng CNN trong một hệ thống YOLO không nhất thiết phải đầy đủ 24 lớp mà tùy từng đối tượng có thể điều chỉnh giảm số lớp cho phù hợp vì số lớp càng ít, tốc độ YOLO càng nhanh.

Lớp cuối cùng sử dụng hàm kích hoạt tuyến tính, trong khi các lớp khác sử dụng hàm leaky RELU:

$$\theta(x) = \begin{cases} x & nếu\ x > 0 \\ 0.1x & ngược\ lại \end{cases} \quad (1.2)$$

2.1.3. Hàm chi phí (Loss function)

Hàm chi phí được dùng để tối ưu hóa trong quá trình huấn luyện và có công thức

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \quad (1.3)$$

Trong đó: λ_{coord} là hằng số cho trước.

(x,y) là tọa độ tương đối của box

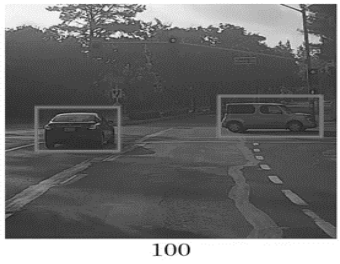
$\mathbb{1}^{obj}$ được định nghĩa như sau;

1, nếu một đối tượng có trong ô thứ *i* và box thứ *j*, mà box này là box ứng viên. 0, đối với các trường hợp khác.

2.1.4. Quy trình phát hiện đối tượng

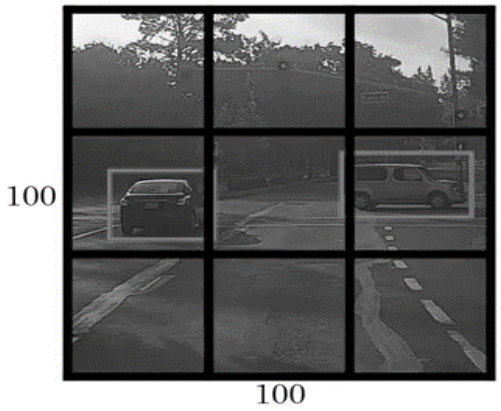
Quá trình phát hiện đối tượng trong một hệ thống YOLO có thể tóm lược trong các bước sau:

YOLO nhận ảnh đầu vào, chẳng hạn như hình 3



Hình 3. Ảnh đầu vào cho YOLO

Hệ thống thực hiện chia ảnh đầu vào thành lưới, để đơn giản chẳng hạn 3x3.



Hình 4. Hệ thống lưới được YOLO chia cho ảnh đầu vào

Phân lớp và định vị ảnh được thực hiện cho mỗi ô trong lưới. Khi đó YOLO dự đoán các box và xác suất thuộc chủng loại nào cho đối tượng (nếu có). Dữ liệu đã gán nhãn sẽ được đưa vào mô hình để huấn luyện. Trong hình 4 hệ thống đã chia ảnh đầu vào thành ma trận 3 x 3, và giả sử trong hệ thống có 3 lớp là người đi bộ, xe hơi và xe máy. Do đó mỗi ô trong lưới, nhãn y sẽ là một vector 8 chiều.

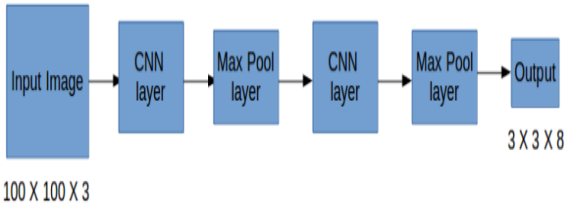
Bảng 2. Bảng thể hiện nhãn y khi phát hiện và không phát hiện đối tượng

y	pc	y	0	y	1
	bx		?		bx
	by		?		by
	bh		?		bh
	bw		?		bw
	c1		?		0
	c2		?		1
	c3		?		0
Vector 8 chiều		Khi không phát hiện đối tượng trong ô		Khi ô có đối tượng	

Trong đó: pc: xác định một đối tượng có hay không trong ô (xác suất); bx, by, bw, bh: các chỉ số của box; c1, c2, c3: tượng trưng cho các lớp. Nếu đối tượng là xe hơi thì c2=1, c1=c3=0.

Xét ô đầu tiên trong ví dụ trên: Vì không có đối tượng nào trong ô này, pc=0 và nhãn y sẽ là: Dấu? hàm ý các giá trị bx, by, bw, by… không có ý nghĩa khi không có đối tượng nào trong ô.

Xét ô có đối tượng xe trong đó: Nhãn y sẽ là: Như vậy với mỗi ô trong 9 ô sẽ có một vector 8 chiều là đầu ra. Đầu ra sẽ là một ma trận có dạng 3 x 3 x 8.

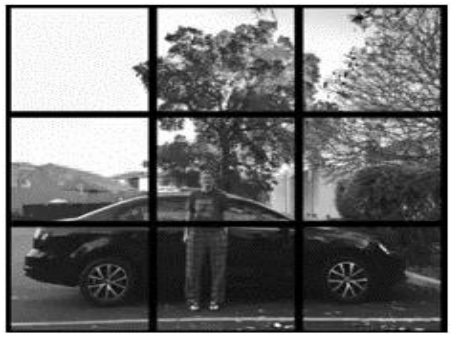


Hình 5. Mô hình huấn luyện trong YOLO

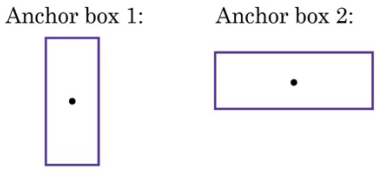
Tính toán các chỉ số cho box: trong YOLO tọa độ tương đối của đối tượng trong ô và loại bớt các box bằng phương pháp Non-max Suppression.

Trường hợp đối tượng nằm trong nhiều ô. Các ô trong YOLO không dự đoán đối tượng có trong ô một cách độc lập nhau mà có sự liên kết giữa các ô do không chỉ sử dụng dữ liệu có trong ô mà còn sử dụng dữ liệu ô lân cận khi trong mạng CNN.

Anchor box: trường hợp chỉ có một đối tượng trong ô thì việc tính toán các chỉ số box tương đối đơn giản nhưng một ô chứa nhiều đối tượng thì việc tính chỉ số box phức tạp hơn. Do đó phát sinh ra khái niệm anchor box. Xét hình sau đây:



Hình 6. Ví dụ về anchor box



Hình 7. Hai loại anchor box

Bảng 3. Nhãn y được thể hiện khi có 2 anchor box

pc	bx	by	bh	bw	c1	c2	c3	pc	bx	by	bw	c1	c2	c3
y														

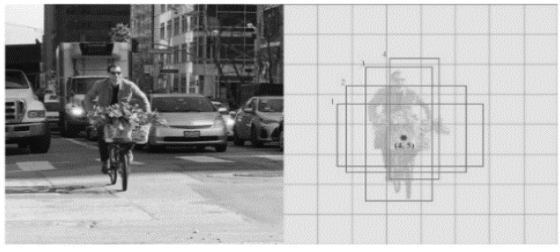
YOLO sẽ lấy tâm của đối tượng và dựa trên vị trí của tâm sẽ gán đối tượng đó thuộc ô nào. Trong ví dụ này tâm cả 2 đối tượng đều nằm chung một ô. Trường hợp này YOLO sẽ tính chỉ số box cho từng đối tượng như sau: YOLO định nghĩa 2 loại anchor box như minh họa trong Hình 7. Khi đó nhãn y sẽ có dạng như sau: 8 dòng trên thuộc anchor box 1, 8 dòng dưới thuộc anchor box 2. Đối tượng được gán cho anchor box nào là dựa trên tính tương đồng hình dáng của anchor box với box (bounding box). Một cách tổng quát số anchor box sẽ bằng với số đối tượng có trong ô.

2.2. THUẬT TOÁN SSD [3]

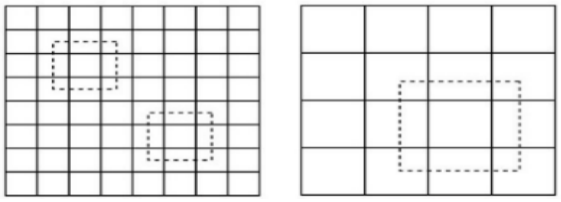
Tương tự như YOLO, việc phát hiện đối tượng hoặc nhiều đối tượng trong ảnh chỉ cần lướt qua ảnh một lần, do đó SSD phát hiện đối tượng nhanh hơn so với cách tiếp cận dựa trên họ R-CNN. Hoạt động SDD gồm 2 giai đoạn: Rút trích các ánh xạ đặc trưng; Sử dụng bộ lọc tích chập để phát hiện đối tượng.

2.2.1. Rút trích đặc trưng

SSD sử dụng mạng tích chập để rút trích đặc trưng, cụ thể là sử dụng lớp Conv4_3. Hình 8 trực quan hóa kết quả rút trích lớp Conv4_3 bằng một ma trận 8x8 (thực tế là 38x38). Với mỗi vị trí sẽ có 4 đối tượng được dự đoán.



Hình 8. Ảnh trái: ảnh gốc. Ảnh phải: 4 bounding box được dùng dự đoán tại mỗi điểm



Hình 9. SSD sử dụng ma trận nhỏ để phát hiện đối tượng trong ma trận lớn hơn

2.2.2. Bộ dự đoán tích chập dùng dự đoán đối tượng

SSD tính các chỉ số cho vị trí và lớp bằng cách sử dụng bộ lọc tích chập. Sau khi có các ma trận đặc trưng, SSD sử dụng bộ lọc tích chập 3x3 để dự đoán. Các bộ lọc này thực hiện dự đoán tương tự như các bộ lọc CNN. Mỗi bộ lọc sẽ cho 25 kênh đầu ra bao gồm: 21 chỉ số cho mỗi lớp và một bounding box cho mỗi vị trí.

2.2.3. Sử dụng nhiều tỷ lệ cho ánh xạ đặc trưng để phát hiện đối tượng

SSD sử dụng các lớp có độ phân giải thấp hơn để phát hiện đối tượng có tỷ lệ lớn. Việc sử dụng ma trận đặc trưng nhiều tỷ lệ cải thiện đáng kể độ chính xác của thuật toán. *Bounding box mặc định;*

2.2.4. Lựa chọn box mặc định

Các box mặc định được chọn thủ công, sau đó SSD định nghĩa một giá trị tỷ lệ ảnh (scale) cho mỗi lớp đặc trưng. Đi từ trái sang, lớp Conv4_3 bắt đầu phát hiện đối tượng từ tỷ lệ nhỏ nhất 0.2 (đôi khi là 0.1) và tăng tuyến tính cho đến lớp tận cùng bên phải sẽ đạt tỷ lệ là 0.9. Kết hợp giá trị tỷ lệ scale với tỷ lệ khung hình, công thức (2.1a) và (2.1b) tính chiều dài w và chiều cao h của box mặc định.

$$w = \text{tỷ lệ scale}.\sqrt{\text{tỷ lệ khung hình}} \quad (2.1a)$$

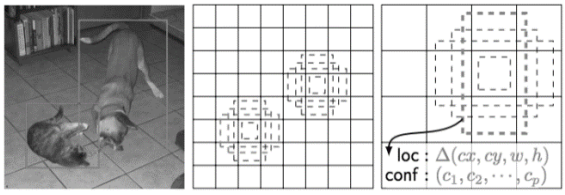
$$h = \frac{tỷ\ lệ\ scale}{\sqrt{tỷ\ lệ\ khung\ hình}} \quad (2.1b)$$

2.2.5. Phát hiện đối tượng

Các dự đoán của SSD được phân loại so khớp dương và so khớp âm. SSD chỉ sử dụng so khớp dương để tính toán chi phí định vị. Nếu chỉ số IoU lớn hơn 0.5, so khớp là dương, ngược lại là âm. Hình 8 minh họa tỷ lệ IOU giữa bounding box với các box mặc định và chỉ một bounding box được chọn cho đối tượng.

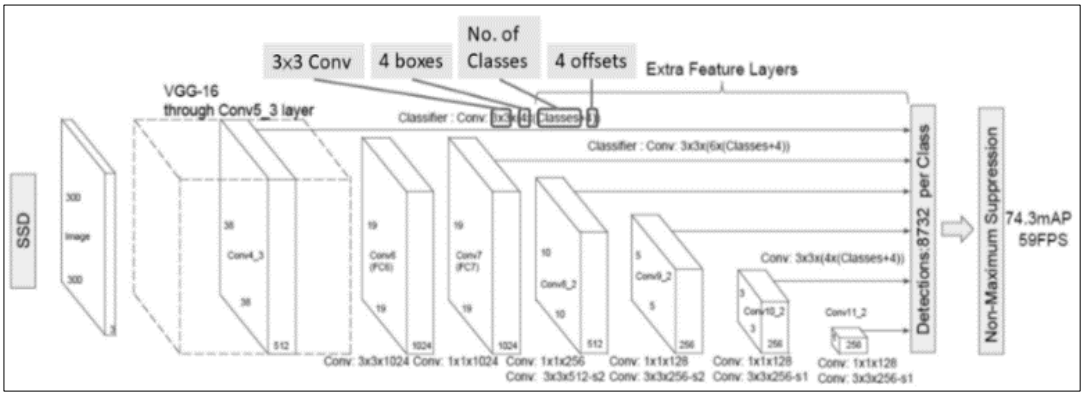
2.2.6. Box mặc định và ma trận đặc trưng nhiều tỷ lệ

Hình 9 minh họa cách SSD kết hợp các ma trận đặc trưng theo nhiều tỷ lệ với các bounding box mặc định để phát hiện đối tượng theo nhiều tỷ lệ. Điều này chứng tỏ lớp ma trận đặc trưng có độ phân giải càng cao thì càng phù hợp cho việc phát hiện các đối tượng có kích thước nhỏ.



Hình 10. Các ma trận tỷ lệ khác nhau được dùng phát hiện đối tượng khác nhau

2.2.7. Kiến trúc mạng SSD



Hình 11. Kiến trúc mạng SSD

Để việc thực hiện được chính xác, các lớp ánh xạ đặc trưng khác nhau đều phải đi qua lớp tích chập 3x3, được sử dụng để phát hiện đối tượng. Như vậy mỗi bounding box sẽ có (c+4) đầu ra. Khi qua bộ lọc 3x3 lớp Conv4_3 sẽ có 38x38x4x(21+4) = 144.400 đầu ra.

Nếu tính riêng bounding box, số lượng bounding box sẽ là 38x38x4 = 5776.

Tương tự cho các lớp khác:

Lớp Conv7: 19x19x6 = 2166 bounding box (6 box cho mỗi vị trí).

Lớp Conv8_2: 10x10x6 = 600 bounding box (6 box cho mỗi vị trí).

Lớp Conv9_2: 5x5x6 = 150 bounding box (6 box cho mỗi vị trí).

Lớp Conv10_2: 3x3x4 = 36 bounding box (4 box cho mỗi vị trí).

Lớp Conv11_2: 1x1x4 = 4 bounding box (4 box cho mỗi vị trí).

Nếu cộng lại sẽ có tất cả: 5776 + 2166 + 600 + 150 + 36 +4 = 8732 bounding box.

2.2.8. Hàm chi phí (Loss Function)

Hàm chi phí trong SSD có công thức sau:

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) \quad (22)$$

Như vậy hàm chi phí trong SSD gồm 2 yếu tố L_{conf} và L_{loc}, trong đó N là số box mặc định dùng để so khớp.

L_{loc} của các box mặc định là khớp là chi phí định vị cho hàm $smooth_{L1}$ để box dự đoán được (l) với box mặc định (g) gần trùng khớp nhau.

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N}\sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m) \quad (2.3)$$

Trong đó:

$$\hat{g}_j^{cx} = \frac{g_j^{cx} - d_i^{cx}}{d_i^w} \quad và \quad \hat{g}_j^{cy} = \frac{g_j^{cy} - d_i^{cy}}{d_i^h}$$

$$\hat{g}_j^{w} = log\left(\frac{g_j^w}{d_i^w}\right) \quad và \quad \hat{g}_j^{h} = log\left(\frac{g_j^h}{d_i^h}\right)$$

Các tham số (cx, cy) là tọa độ tâm của bounding box, w và h là chiều rộng và chiều cao của bounding box.

L_{conf} là chi phí cho độ tin cậy của hàm softmax trên c lớp. (α = 1 trong trường hợp kiểm tra chéo).

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} \log(\hat{c}_i^p) - \sum_{i\in N} log\,(\hat{c}_i^0) \quad (2.4)$$

Trong đó: $\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$

$x_{ij}^p = \{1,0\}$ là yếu tố chỉ thị khớp hay không giữa box mặc định thứ i và box thực tế thứ j của lớp p.

2.2.9. Tỷ lệ co giản và tỷ lệ khung hình của box mặc định

Tỷ lệ co giản box mặc định được tính như sau:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(k-1), k \in [1,m] \quad (2.5)$$

Giả sử có m ánh xạ đặc trưng để dự đoán, s_k được dùng để tính cho đặc trưng k. S_{min} và S_{max} lần lượt là 0.2 và 0.9, nghĩa là tỷ lệ co giản tối thiểu là 0.2 và tối đa là 0.9.

Với mỗi tỷ lệ co giản s_k, có 5 tỷ lệ khung:

$$a_r \in \left\{1,2,3,\frac{1}{2},\frac{1}{3}\right\} \; (w_k^a = s_k\sqrt{a_r}) \; (h_k^a = \frac{s_k}{\sqrt{a_r}}) \quad (2.6)$$

Do đó có thể tính tối đa 6 bounding box với tỷ lệ khung khác nhau. Đối với các lớp chỉ có 4 bounding box, SSD bỏ 2 tỷ lệ 3 và 1/3.

2.3. SO SÁNH YOLO VÀ SSD

2.3.1. Về phương pháp

Cả SSD và YOLO đều chung ý tưởng dự đoán đối tượng chỉ cần qua một lần quét ảnh, đều dựa trên mạng CNN trong deep learning để rút trích đặc trưng ảnh, đều chia ảnh thành lưới gồm nhiều ma trận tỷ lệ khác nhau. Sự khác nhau chính giữa 2 thuật toán này là cách thức xây dựng các box để định vị và xác định vùng biên của các đối tượng trong ảnh (2.2). Nếu YOLO sử dụng anchor box thì SSD sử dụng box mặc định. Cách thức dò tìm cũng khác nhau, YOLO xét sự tương quan giữa các vùng ảnh bằng cách gán nhãn còn SSD sử dụng box với nhiều tỷ lệ khác nhau và tỷ lệ khung ảnh cho các đối tượng có box cùng hình dạng. Kết quả là SSD tỏ ra kém hơn YOLO trong việc phát hiện những đối tượng có kích thước nhỏ và ảnh có độ phân giải kém.

2.3.2. Về thực nghiệm

Thực nghiệm so sánh được thực hiện cho 2 thuật toán: YOLO, SSD [3], trên máy Jupyter Notebook chạy trên Google Cloud có GPU Nvidia Tesla K80 14Gb. Chỉ số đánh giá được sử dụng là AP (Average Precision) trên bộ dữ liệu COCO.

Bảng 4. So sánh chỉ số AP giữa 2 thuật toán YOLO và SSD

YOLO	SSD
33	31.2

So sánh về thời gian phát hiện đối tượng trên 3 ảnh.

Bảng 5. So sánh thời gian phát hiện đối tượng giữa 2 thuật toán YOLO và SSD

YOLO	SSD
	
0:00:02.204303	0:00:03.170221

Qua thực nghiệm trên nhận thấy chỉ số AP giữa 2 thuật toán YOLO và SSD là tương đương nhau, về tốc độ cũng chênh lệch không đáng kể. Giữa YOLO và SSD tốc độ nhanh gần như tương đương nhau nhưng YOLO tỏ ra vượt trội hơn SSD về đối tượng được phát hiện trong ảnh nhất là đối tượng có tỷ lệ nhỏ.

3. KẾT LUẬN

Mặc dù có hạn chế về độ chính xác so với Faster RCNN nhưng trong những trường hợp cần phát hiện tương đối một đối tượng nào đó một cách tức thời (theo thời gian thực) hoặc số đối tượng cần phát hiện tương đối ít, YOLO và SSD thực sự là một thuật toán đáng sử dụng.

TÀI LIỆU THAM KHẢO

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016), *You Only Look Once: Unified, Real-Time Object Detection*, Computer Vision and Pattern Recognition, arXiv:1506.02640.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun (2016), *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Computer Vision and Pattern Recognition, arXiv:1506.01497v3 [cs.CV].

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg (2016), *SSD: Single Shot MultiBox Detector*. Computer Vision and Pattern Recognition, arXiv:1512.02325v5.

[4] Ross Girshick (2015), *Fast R-CNN*, Computer Vision and Pattern Recognition, arXiv:1504.08083v2.

[5] https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/, ngày truy cập: 06-12-2018.

[6] https://github.com/topics/tensorflow-yolo, ngày truy cập: 29-8-2019.

[7] https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/, ngày truy cập: 07-11-2016.

[8] https://nttuan8.com/bai-10-cac-ky-thuat-co-ban-trong-deep-learning/.

Ngày nhận bài: 06-01-2020. Ngày biên tập xong: 30-6-2020. Duyệt đăng: 24-9-2020