TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

MẠNG PHÁT HIỆN ĐỐI TƯỢNG 1

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Nhóm: **Fantastic4**

Tp. Hồ Chí Minh, tháng 01/2021

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Nguyễn Hoàng Thái Duy - 1712019
Bùi Văn Hợp - 1712046
Âu Dương Tấn Sang - 1712145

MẠNG PHÁT HIỆN ĐỐI TƯỢNG 1

CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN
PGS.TS. Lý Quốc Ngọc

Tp. Hồ Chí Minh, tháng 01/2021

Mục lục

Phần 1

Giới thiệu

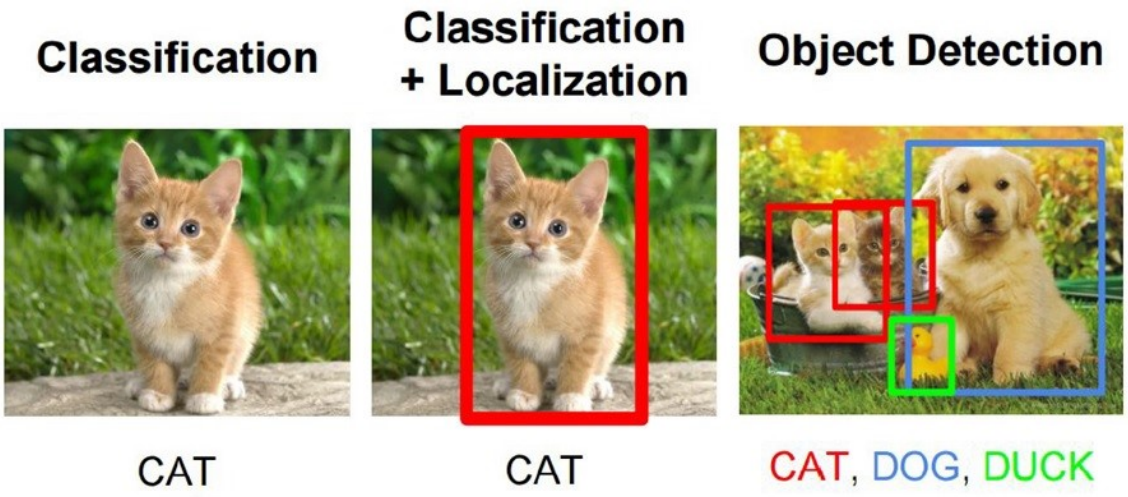
1.1 Động lực nghiên cứu

Nhận dạng đối tượng là một trong những hướng nghiên cứu nên tảng trong lĩnh vực Khoa học dữ liệu Thị giác. Nhiệm vụ tìm và định danh đối tượng ở mức độ định danh (định dạng chính xác chó, mèo, chuột, ...). Nhận dạng đối tượng được ứng dụng trong nhiều lĩnh vực trong cuộc sống như hệ thống an ninh, hệ thống xe tự hành, ...; trong công nghiệp như phân loại sản phẩm, phát hiện sản phẩm lỗi, ...; trong nghiên cứu khoa học, nhân dạng đối tượng mở ra nhiều hướng nghiên cứu liên quan như phân đoạn hình ảnh ở mức độ từng vật thể, captioning hình ảnh, tracking vật thể, ...Tuy nhiên vẫn có một số vấn đề tồn tại, ví dụ như độ ổn định của mô hình, chất lượng hình ảnh và khả năng áp dụng mô hình trên nhiều thiếu bị khác nhau (thiết bị nhúng, thiết bị cầm tay, ...).

1.2 Tổng quan bài toán phát hiện đối tượng

Bài toán phân loại (Classification)

Với bài toán phân loại ảnh, kết quả trả ra khi đưa 1 ảnh đơn vật thể vào là xác suất ảnh đó thuộc về object nào.



Hình 1.1: Classification - Localization - Object Detection

Input: Ảnh $I \in R^{W,H,D}$
Output: $R = \{(Class1 : Confidence), (Class2 : Confidence), ...\}$ với $Class$ là loại vật thể, $Confidence$ tương ứng với xác suất vật thể trong ảnh thuộc về Class tương ứng.

Bài toán định vị đối tượng(Localization)

Đối với bài toán định vị đối tượng, ta đã xác định loại đối tượng trên ảnh. Phần còn lại ta cần xác định vị trí chính xác của vật thể trên ảnh.

Input: Ảnh $I \in R^{W,H,D}$
Output: $R = \{(x, y, w, h), ...\}$ với x, y, w, h tương ứng với tọa độ góc trái trên, chiều dài và rộng của bounding-box.

Bài toán phát hiện đối tượng(Object Detection)

Bài toán Phát hiện đối tượng là tổng hợp của 2 bài toán Phân loại ảnh và Định vị đối tượng. Có thể được mô tả như sau

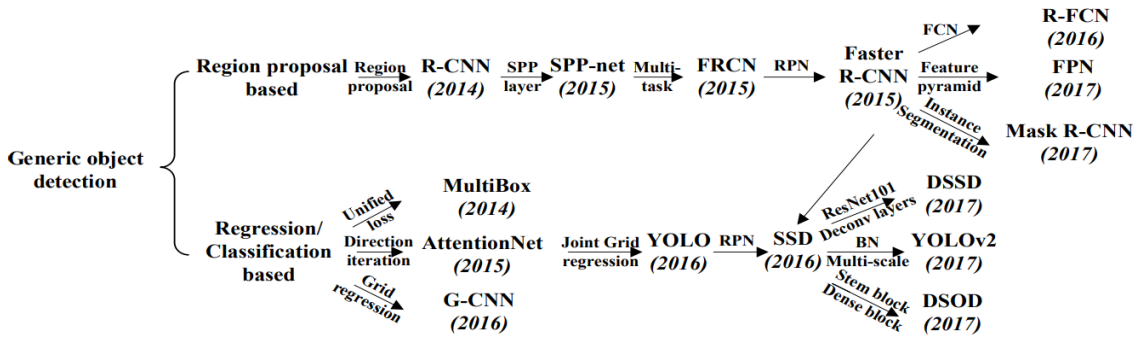
Input: Ảnh $I \in R^{W,H,D}$
Output: $R = \{(x, y, w, h, Class, Confidence), ...\}$ với x, y, w, h tương ứng

với tọa độ góc trái trên, chiều dài và rộng của bounding-box tương ứng với $Class$ là loại vật thể, $Confidence$ tương ứng với xác suất vật thể trong ảnh thuộc về Class tương ứng.

Phần 2

Tổng quan

2.1 Các mạng nhận dạng đối tượng



Hình 2.1: Phân loại các mô hình Nhận dạng đối tượng tính tới 2019[6] theo hướng tiếp cận

- **Thời kỳ đầu - các giải thuật truyền thống** Trước 2014, khi chưa ghi nhận sự phát triển của mạng nơ-ron tích chập. Những giải thuật phát hiện đối tượng chủ yếu dựa vào các đặc trưng cấp thấp và trung. Viola Jones Detector là một trong những giải thuật đầu tiên được ghi nhận về phát hiện gương mặt thời gian thực. Giải thuật VJ sử dụng những các tiếp cận cổ điển nhất như cửa sổ trượt: tìm tất cả các vùng ứng viên có thể trên ảnh với kích thước cố định. Ngoài ra, Histogram of Oriented Gradients (HOG) - trích xuất đặt trưng cũng

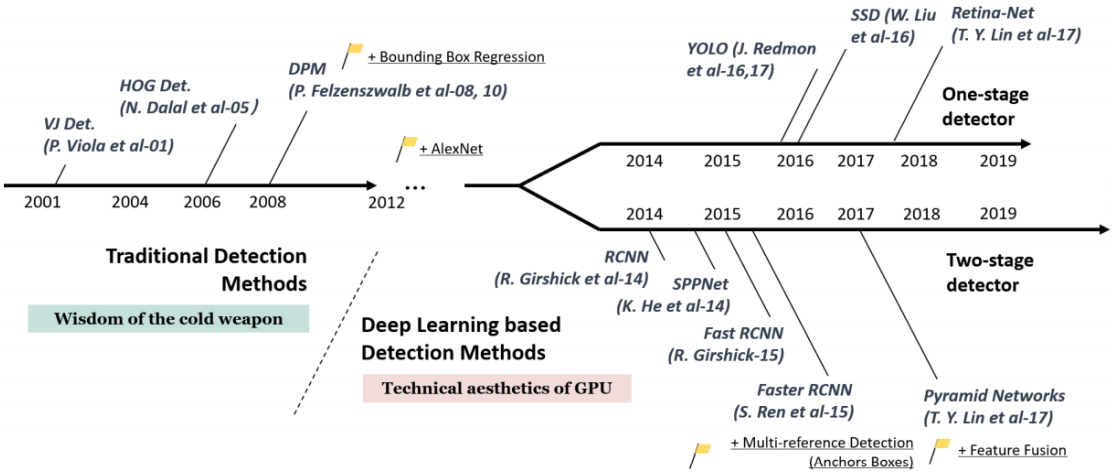
được khai thác vào bài toán phát hiện vật thể, đặc trưng này hiệu quả hơn bởi sự bất biến với phép scale và hình dạng đối tượng.

- **Sự phát triển của các kiến trúc mạng học sâu** Khi các mạng học sâu tích chập phát triển, bài toán phát hiện đối tượng được chia thành 2 nhóm theo hướng tiếp cận này "one-stage-detection" và "two-stage-detecion".

  Ở hướng "one-stage-detection", YOLOv1 là giải thuật được phát triển đầu tiên theo hướng này khi kiến trúc mạng neural tích chập bắt đầu phát triển. Một mạng neural tích chập được áp dụng trong xuyên suốt quá trình học, với kiến trúc mạng Yolo sử dụng, ảnh sẽ được chia theo từng vùng, dự đoán bounding-box và xác suất của từng vùng một cách đồng thời. Ở các phiên bản Yolov2,3 được phát triển sau này. Tỷ lệ chính xác được tăng lên nhưng vẫn đảm bảo được tốc độ. "two-stage-detecion", các kiến trúc họ nhà R-CNN đi theo hướng tiếp cận này. Các framework đi theo kiến trúc này tuân theo hướng "coarse-to-fine", bắt đầu với xây dựng các vùng ứng viên, sau đó trích xuất đặc trưng từ các vùng ứng viên để lấy feature map. Từ đó mới giải quyết bài toán Bounding-box regression, Confident score. Ở các bước như lựa chọn vùng ứng viên, trích xuất đặc trưng, bounding-box regression là các mạng khác nhau hoàn toàn nhưng được chia sẻ trọng số. Khác so với "one-stage-detection" chỉ dùng 1 mạng xuyên suốt quá trình học.

2.2 Bộ dữ liệu

Dữ liệu trong lĩnh vực Khoa học dữ liệu thị giác luôn là vấn đề đáng quan tâm trong nhiều tác vụ nghiên cứu. Phát hiện đối tượng trên ảnh là một trong số những hướng nghiên cứu được chú ý phát triển số lượng dataset lớn, đa dạng về số lượng, chất lượng và chủng loại. Một trong số đó là:



Hình 2.2: Phân loại các mô hình Nhận dạng đối tượng [7] theo năm

- PASCAL VOC
  The PASCAL Visual Object Classes (VOC) Challenges1 (từ 2005 đến 2012) là một trong những cuộc thi lớn đời đầu tiên trong lĩnh vực Khoa học dữ liệu Thị giác. Có nhiều task nhỏ trong PASCAL VOC, bao gồm phân loại hình ảnh, phát hiện đối tượng,phân đoạn ngữ nghĩa and nhận diện hành động. Bộ dataset bao gồm 5000 ảnh huấn luyện với hơn 12000 vật thể được gán nhãn (VOC07). Ở bọ VOC12, có hơn 11000 ảnh huấn luyện và 27000 đối tượng gán nhãn. Các nhãn bao gồm người, động vật, phương tiện giao thông, nội thất - những vật dụng trong đời sống hằng ngày.

- ILSVRC
  The ImageNet Large Scale Visual Recognition Challenge là một trong những cuộc thi thúc đẩy sự phát triển, qua đó tạo ra hàng loạt những mô hình state-of-the-art. Bộ dữ liệu bao gồm hơn 200 vật thể với hơn 517k ảnh và 534k vật thể được gán nhãn.

- MS-COCO
  MS-COCO là cuộc thi hằng nằm được tổ chức từ 2015. Với số lượng dữ liệu cực lớn và tăng theo từng năm. Đến nay đã đạt 164k ảnh và 897k vật thể được gán nhãn thuộc 80 loại khác nhau. Điểm mạnh

của MS-COCO là gần với ảnh thực tế hơn vì chứa nhiều vật thể nhỏ hơn(diện tích chiếm ít hơn 1% ảnh tổng thể)

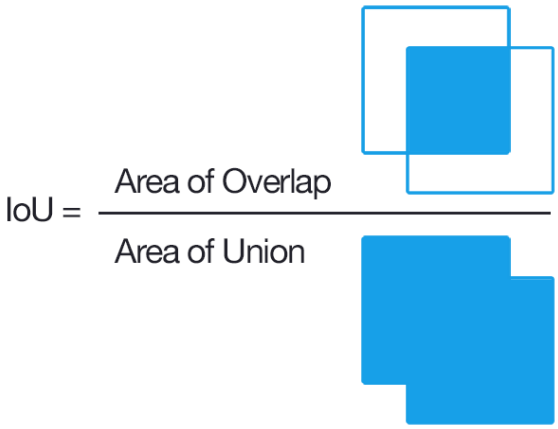
2.3 Phương pháp đánh giá

2.3.1 Điểm IoU

IoU (Intersection over Union) score là một điểm để đánh giá phần chồng lên giữa 2 bounding box.

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

Trong đó A, B là 2 bounding box.



Hình 2.3: Cách tính toán trực quan của điểm IoU giữa 2 bounding box

2.3.2 Precision - Recall

Precision đo mức độ chính của dự đoán, tương đương trong số những dự đoán, có bao nhiêu dự đoán là chính xác.

Recall đo mức độ tốt của dự đoán của bạn, tương đương với việc trong số những dự đoán là dương (positive) thì có bao nhiêu trong đó thực sự là

dương.

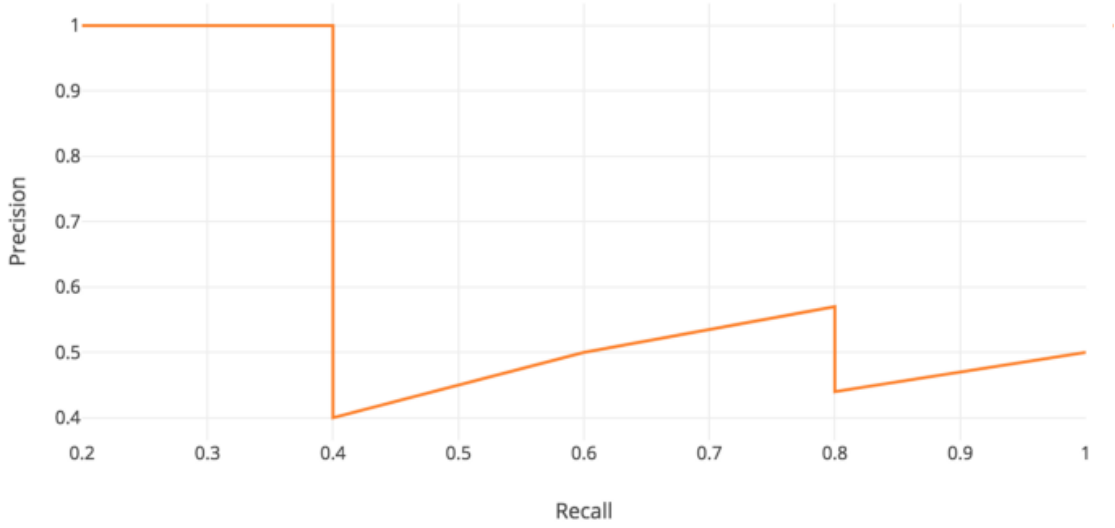
$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Trong đó: TP là True Positive, FP là False Positive, FN là False Negative, FP là False Positive.

2.3.3 Điểm mAP

Với một mức độ đánh giá bounding box, có thể ta sẽ coi những bounding box được dự đoán có điểm IoU score lớn hơn 0.75, ta có thể vẽ được một đồ thị biểu diễn giữa Precision và Recall.



Hình 2.4: Ví dụ về một đường cong Precision - Recall

Điểm mAP (mean Average Precision) hay là AP là một điểm để đánh giá mức độ chính xác trong tất cả các lớp cân bằng giữa Precision - Recall, ở đây chính là diện tích của phần dưới đường cong Precision - Recall.

$$mAP = \int_0^1 p(r)dr$$

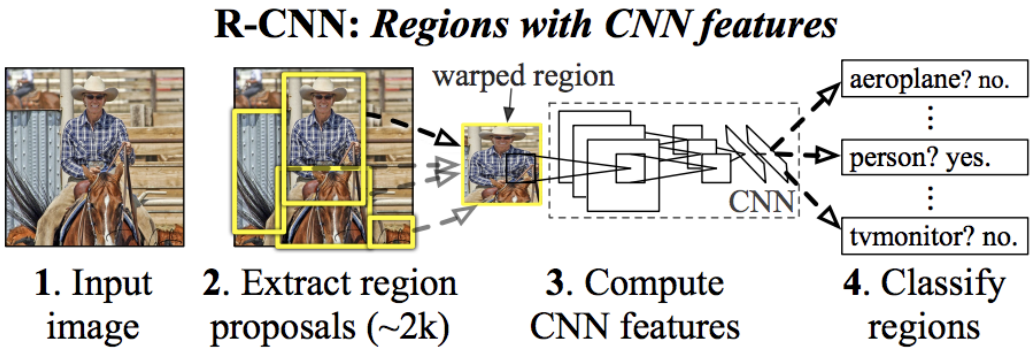
Ta có $p(r)$ là một hàm biểu diễn giá trị của Precision tại một giá trị Recall. Giá trị mAP càng cao và tiệm cận 1 thì mô hình càng có độ chính xác cao.

Phần 3

Các mạng phát hiện đối tượng

3.1 Region Based Convolutional Neural Networks (R-CNN)

3.1.1 Tổng quan



Hình 3.1: Mô hình R-CNN [3] với 3 phần độc lập: (1) Ảnh đầu vào, (2) lựa chọn các vùng ứng viên, (3) Trích xuất đặc trưng, (4) Phân loại vùng và xác định vị trí

3.1.2 Lựa chọn các vùng ứng viên

Việc lựa chọn các vùng ứng viên có thể thực hiện bằng các phương pháp sau: objectness, selective search, category-independent object proposal, constrained paraetric min-cuts, ... Trong kiến trúc R-CNN, tác giả lựa chọn phương pháp Selective Search [2].

Giải thuật Selective Search

Algorithm 1 Tìm vùng ứng viên với Selective Search

Input: Ảnh $I \in R^{W,H,3}$

Output: Tập $S = \{R_1, R_2, ..., R_{2000}, ...\}$ trong đó $R_i = [x, y, w, h]$ là tọa độ bao đóng của vùng ứng viên.

1: Áp dụng giải thuật [1] để phân đoạn ảnh nhằm tạo ra các vùng phân đoạn ban đầu.
2: Lặp lại để gộp các vùng giống nhau
3: Xác định tọa độ của các vùng ứng viên

Hình 3.2: Tìm các vùng ứng viên

3.1.3 Trích xuất đặc trưng

Ở giai đoạn trích xuất đặc trưng, ảnh đầu vào được quy định kích thước cố định $227x227$ được đưa vào một mạng CNN bất kỳ (ở bài gốc, tác giả sử dụng AlexNet[8]) như VGG, Resnet, InceptionNet, ...

Mạng CNN được áp dụng ở bước này, nhằm trích xuất vector đặc trưng 4096 chiều. Do mạng CNN được sử dụng, bộ trọng số không được chia sẻ các thông tin về các vùng ứng viên ở bước 1 và phân loại ở bước sau. Vì thế cả mô hình phụ thuộc vào mạng pretrained-CNN để lấy được đặc trưng tốt.

3.1.4 Phân loại

Sau khi đưa bức ảnh qua bước chọn các vùng ứng cử viên và đưa vào một bộ trích xuất đặc trưng, ta sẽ có được một vector đặc trưng mô tả thông tin của vùng dữ liệu đó.
Tiếp theo ta cần đưa chúng vào một bộ phân loại để biết chúng thuộc lại nào trong K + 1 lớp (ở đây có lớp số 0 là lớp phân loại vùng là nền ảnh và K loại vật thể mà ta đặt ra trước).

Bài báo gốc của R-CNN sử dụng mô hình SVM nhị phân để phân loại cho vector đặc trưng của từng vùng. Cụ thể với K + 1 lớp cần phân loại, ta cần sử dụng đúng K + 1 bộ phân loại SVM nhị phân được huấn luyện độc lập với nhau.

Với mỗi bộ phân lớp cho riêng một lớp trong K + 1 lớp, ta sử dụng một ngưỡng điểm IoU là 0.3 giữa bounding box và ground truth. Ta tiếp tục đưa vector đặc trưng và bounding box của vùng được phân vào lớp này vào tiếp một bước để tinh chỉnh lại bao đóng.

3.1.5 Hồi quy cho các bao đóng (bounding box regressor)

Sau khi có bounding box được đề xuất từ bước trên và có cả ground truth tương ứng với nó với cùng lớp đã được phân vào. Ta tiến hành tinh chỉnh lại bounding box (bao gồm 4 tham số cụ thể 2 tham số là tọa độ của điểm trung tâm và 2 tham số là chiều rộng và chiều cao của bounding box).

Ta có bounding box được dự đoán hiện tại là $p = (p_x, p_y, p_w, p_h)$ và grouth truth cho bounding box là $g = (g_x, g_y, g_w, g_h)$.

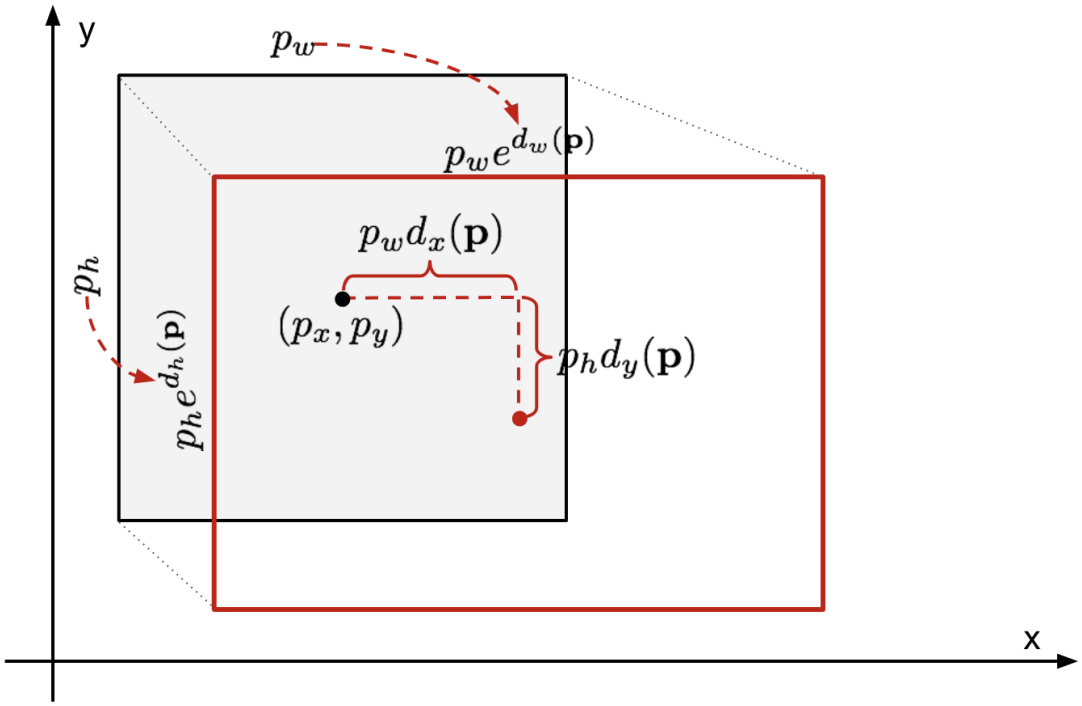
Ta sẽ có một bài toán học cách tinh chỉnh bounding box p như sau: Gọi $\hat{g}$ là bounding box được tinh chỉnh từ bounding box p. Ta có:

$$\begin{aligned}
\hat{g}_x &= p_w d_x(\mathbf{p}) + p_x \\
\hat{g}_y &= p_h d_y(\mathbf{p}) + p_y \\
\hat{g}_w &= p_w \exp(d_w(\mathbf{p})) \\
\hat{g}_h &= p_h \exp(d_h(\mathbf{p}))
\end{aligned}$$

Mục đích cho việc dùng hàm biến đổi này là ta muốn học những tham số để biến đổi tọa độ trung tâm của bounding box sẽ bất biến với phép co giãn, và chiều rộng, chiều dài sẽ biến đổi theo tỉ lệ log.

Ta thấy rằng nhưng tham số cần học sẽ là $d_x(p), d_y(p), d_w(p), d_h(p)$, chúng sẽ có miền giá trị thực không bị giới hạn $[-\infty, +\infty]$, nhưng giá trị mục tiêu cần học đó là:

$$\begin{aligned}
t_x &= (g_x - p_x)/p_w \\
t_y &= (g_y - p_y)/p_h \\
t_w &= \log(g_w/p_w) \\
t_h &= \log(g_h/p_h)
\end{aligned}$$



Hình 3.3: Mô phỏng lại cách tinh chỉnh bounding box với các tham số từ $d(p) = \{d_x(p), d_y(p), d_w(p), d_h(p)\}$

Vậy nên từ đây ta có hàm lỗi dựa trên $(d(p), t)$ như sau:

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2$$

Trong đó: giá trị λ là một siêu tham số được lựa chọn cho Regularization, và được chọn dựa trên Cross-Validation trong quá trình lựa chọn mô hình và các tham số.

3.1.6 Dữ liệu huấn luyện

Với dư liệu huấn luyện, chúng ta sẽ tạo một tập các ảnh và bounding box chứa tất cả chứa đủ tất cả các lớp mà phần selective search có thể tìm ra. Dữ liệu huấn luyện được cần cho cả 3 bước trong R-CNN: (1) Fine-tune các tham số trong mạng học sâu trích xuất đặc trưng, (2) Bộ phân loại SVM và (3) Huấn luyện cho giai đoạn tinh chỉnh bounding box.

3.1.7 Hạn chế

- Ở bước trích xuất đặc trưng, bộ trọng số của mạng CNN không được chia sẻ với bước tạo các vùng ứng viên và phân loại lớp ở bước sau. Vì vậy toàn bộ mô hình bao gồm 3 cấu trúc rời rạc nhau. Gây hạn chế cho việc fine-tune mô hình vì có nhiều siêu tham số riêng biệt ở từng cấu trúc.

- Ở bước tạo ra các vùng ứng viên, số lượng vùng ứng viên được tạo ra là rất lớn (xấp xỉ 2000 vùng) vì vậy tăng chi phí tính toán.

3.2 Fast R-CNN

3.2.1 Hướng cải thiện

Tác giả của bài báo R-CNN cũng đưa ra những hạn chế của mô hình R-CNN như sau:

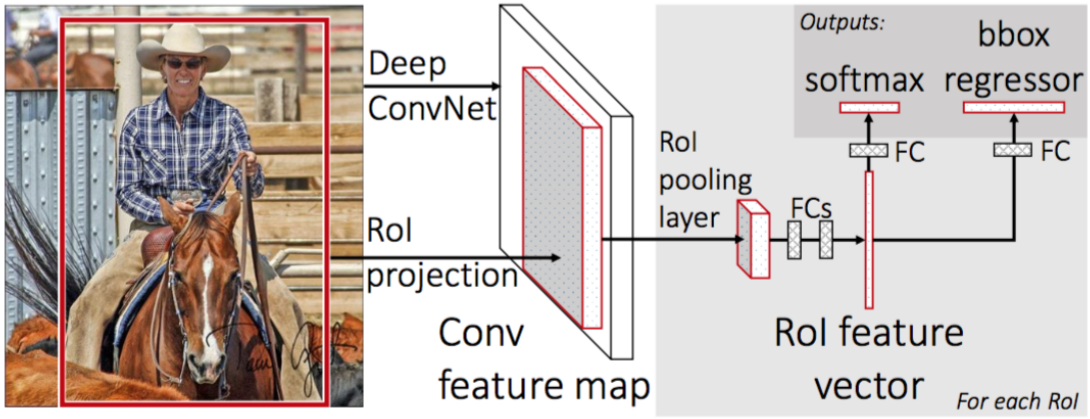
1. Phải đưa từng vùng vào trong 2000 vùng được đề xuất vào phân lớp cho từng ảnh

2. Ở bước Regional Proposal với Selective Search, đây là một thuật toán cố định, không có các bước học để cải thiện và sử dụng những mô hình tiên tiến, nên có thể dẫn tới việc đưa ra các vùng không chính xác.

3. Thời gian chạy thực tế (in realtime) quá lâu.

3.2.2 Mô hình Fast R-CNN

Điểm thay đổi so với R-CNN, Fast-RCNN [4] sẽ đưa toàn bộ ảnh vào một ConvNet, để từ ảnh ban đầu, ta tạo ra 1 feature map. Và tất cả những vùng được đề xuất sẽ dùng chung feature map này.



Hình 3.4: Mô hình cho Fast R-CNN

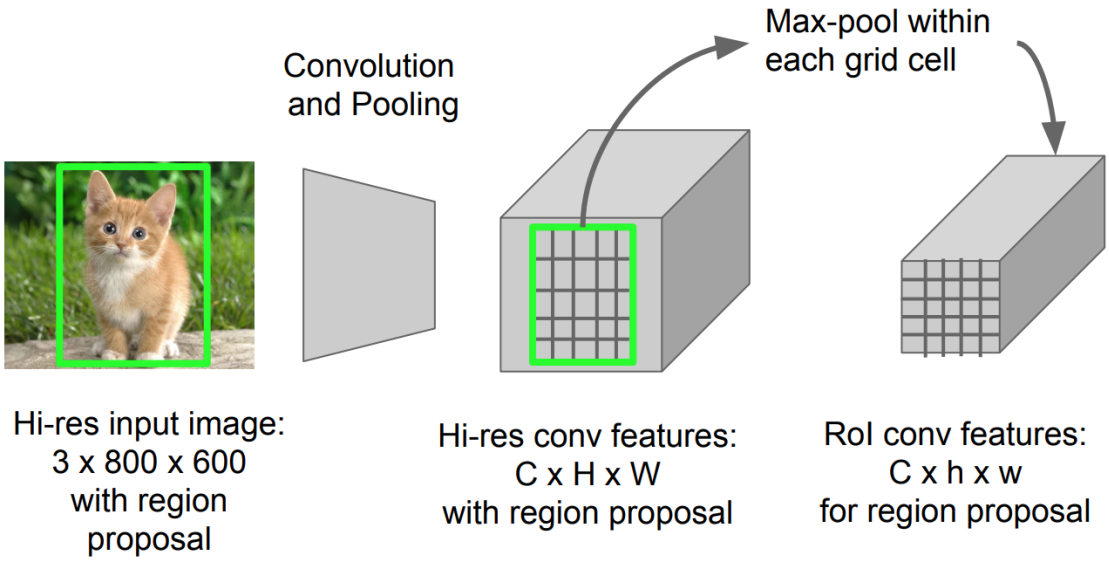
Tiếp theo chúng ta đưa feature map qua một lớp RoI pooling layer. Ý nghĩa của lớp ROI Pooling Layer sẽ được trình bày ở phần tiếp theo.

3.2.3 ROI Pooling Layer

ROI Pooling Layer là một loại Max-Pooling để đưa một vùng feature-map với kích thước $C \times H \times W$ bất kì về thành feature-map với kích thước $c \times h \times w$.

Ta coi một vùng được đề xuất trong ảnh ban đầu, sẽ được ánh xạ trong feature-map có kích thước là $C \times H \times W$ sau khi đưa ảnh qua một ConvNet ở bước trước.

Với mục tiêu đưa mọi vùng được đề xuất của mô hình ConvNet ở trên với cùng kích thước (tương tự như việc chuyển tất cả các vùng về cùng 1 kích thước của R-CNN), kích thước kết quả của ROI Pooling Layer là $c \times h \times w$, ta chia lưới của feature-map thành các vùng h / H và w / W rồi áp dụng max pooling vào từng vùng để rút gọn feature map thành kích thước $c \times h \times w$.



Hình 3.5: ROI Pooling Layer được áp dụng trong mô hình

3.2.4 Hàm lỗi kết hợp

Ta có thể xem Fast R-CNN là một mô hình đa nhiệm (multi-tasks neural network), việc sử dụng một hàm độ lỗi để đánh giá chính xác tác vụ học từ mô hình cần phải cân bằng giữa việc phân lớp đúng vật thể vào lớp của chúng (classfication task) và việc tinh chỉnh lại bounding box chính xác (refine bounding box task). Vậy nên hàm lỗi mà mô hình đề xuất như sau.

Kí hiệu:

- u là giá trị mà vùng thuộc về lớp nào trong $K+1$ lớp, $u = 0..K$ với 0 được quy định là lớp background.
- p là một vector $K+1$ chiều là kết quả của 1 lớp softmax thể hiện xác suất thuộc về lớp thứ i với$p_i, p = (p_0, p_1, p_2...p_K)$.
- v là bounding box chính xác, $v = (v_x, v_y, v_w, v_h)$ được lấy ra từ tập dữ liệu để huấn luyện
- t^u là bounding box được dự đoán $t^u = (t^u_x, t^u_y, t^u_w, t^u_h)$

.

Ta sẽ có hàm lỗi của mô hình là:

$$L(p, u, t^u, v) = L_{cls}(p, u) + 1[u >= 1]L_{box}(t^u, v)$$

Trong đó:

$$L_{cls}(p, u) = -log(p_u)$$

$$L_{box}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{smooth}(t_i^u - v_i)$$

Và hàm $1[u >= 1] = 1 if u >= 1 otherwise 0$, đây là hàm để kiểm tra trường hợp $u \geq 1$ thì mới thêm độ lỗi của tác vụ bounding box, vì ta đã quy định $u = 0$ là xác định bounding box đó là background, nên sẽ không được tính L_{box} vào trong độ lỗi tổng L.

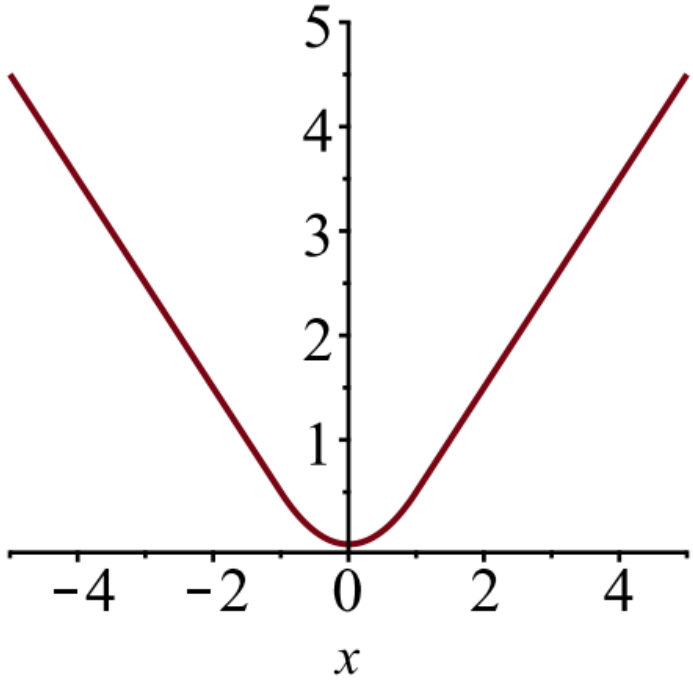
Ngoài ra hàm $L_1^{smooth}(x)$ được sử dụng để tăng khả năng ít bị nhiễu bởi những giá trị ngoại lai, do độ lỗi của tác vụ tinh chỉnh bounding box khá dễ gặp những trường hợp nhiễu.

$$L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

3.2.5 Cách train mô hình

1. Bước 1: Train một mô hình ConvNet phân lớp với tập dữ liệu hiện có, hoặc sử dụng một pretrained ConvNet model từ ImageNet. Ta chú ý chỉ cần đưa bài toán về việc phân K + 1 lớp chúng ta cần, với 1 lớp thêm vào cho những vùng background (không có object quan tâm).

2. Bước 2: Sử dụng Selective search để lấy được 2000 vùng được đề



Hình 3.6: Hàm L_1^{smooth}

xuất.

3. Bước 3: Áp dụng mô hình phân lớp ở Bước 1, đưa ảnh đầu vào qua mô hình. Sau đó ta chú ý thay đổi những phần sau của mô hình:

   - Thêm lớp ROI Pooling Layer (nếu là những pretrained model thì ta loại bỏ lớp max-pooling cuối cùng) để đưa N vùng đề xuất ở bước 2 vào và đưa chúng về 1 feature map chung có kích thước là Cx7x7.
   - Đưa feature map Cx7x7 ở trên qua một vài lớp trước khi ra một feature vector cuối cùng.

4. Bước 4: Kết quả sẽ bao gồm 2 nhánh output như sau:

   - Một nhánh để phân lớp cho vùng thuộc 1 trong K + 1 lớp được định nghĩa.

- Một nhánh để tinh chỉnh bounding-box cho vùng được đề xuất ban đầu.

3.2.6 Đánh giá

- Tốc độ chạy đã được cải thiện hơn so với R-CNN bởi vì ta sử dụng chung 1 feature map cho toàn bộ các vùng được đề xuất
- Tuy nhiên Fast R-CNN vẫn còn một điểm chưa tối ưu hoàn toàn là việc đề xuất các vùng vẫn phải do một thuật toán khác, nghĩa là chưa hoàn toàn có thể đưa mô hình End-to-end sử dụng ConvNet được.

	R-CNN	Fast R-CNN
Training Time:	84 hours	**9.5 hours**
(Speedup)	1x	**8.8x**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**

Faster!

FASTER!

Hình 3.7: So sánh tốc độ giữa R-CNN và Fast R-CNN

.

3.3 Faster R-CNN

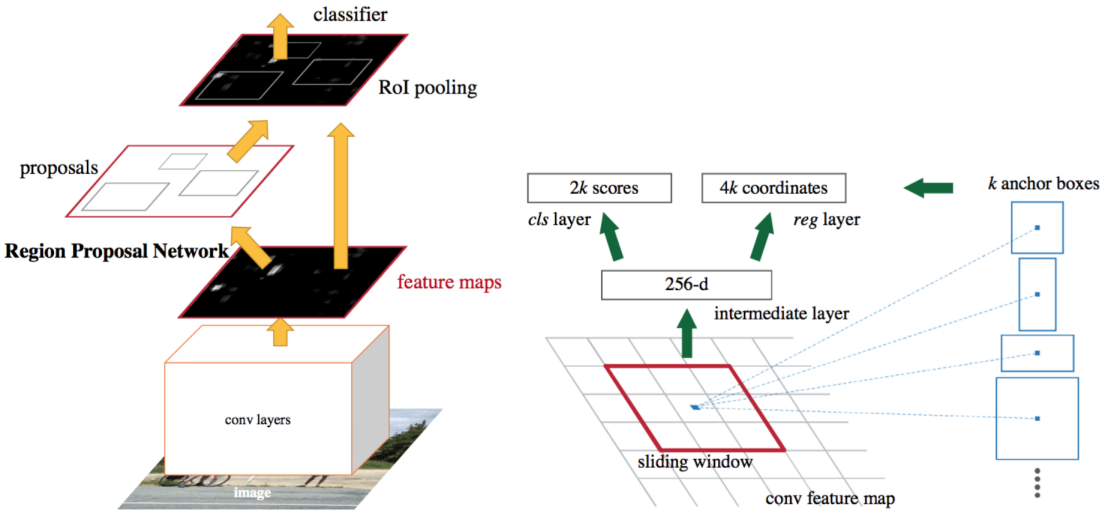
3.3.1 Hướng cải thiện

So với 2 mô hình tiền nhiệm, mô hình Faster R-CNN [9] có những điểm cải thiện sau:

- Thêm vào một mạng học sâu là Regional Proposal Network để đưa ra các bounding box có khả năng là vật thể thay cho thuật toán Selective Search trước đây.
- Regional Proposal Network là một mạng học sâu được huấn luyện đồng thời chung với thành phần khác và sử dụng feature map chung.

Với những bước cải tiến thay hoàn toàn thuật toán Selective Search, và vẫn giữ nguyên các tham số để phân loại vật thể cũng như tinh chỉnh bounding box như Fast R-CNN, nên thời gian huấn luyện cũng như đưa ra 1 kết quả rất nhanh, có thể được ứng dụng trong thời gian thực.

3.3.2 Mô hình Fast R-CNN



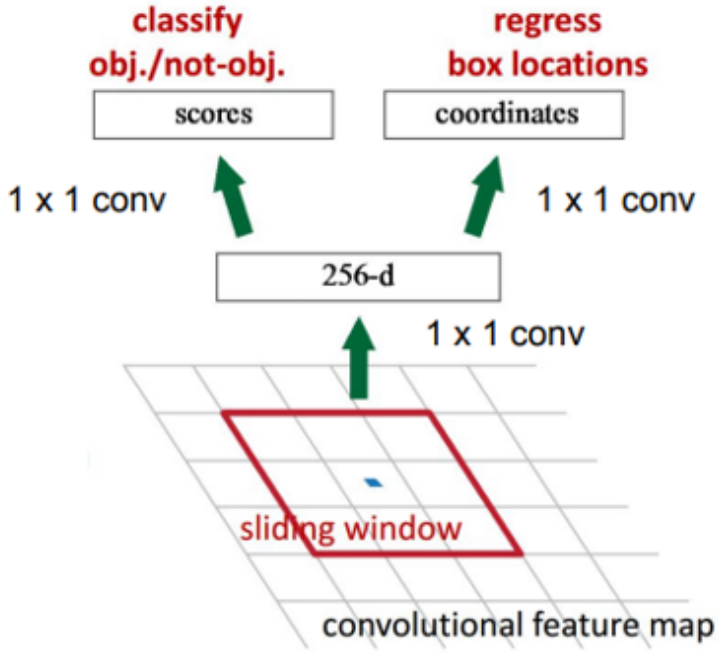
Hình 3.8: Mô hình của Faster R-CNN được đưa ra trong bài báo gốc[9]

Về cơ bản mô hình Faster R-CNN gần như tương tự với Fast R-CNN, điểm khác biệt lớn nhất là Mạng đề xuất vùng (Regional Proposal Network) và một hàm lỗi mới để phù hợp với thêm một tác vụ đánh giá các vùng được đề xuất có đủ tốt hay không. Hai sự thay đổi chính đó sẽ được trình bày trong 2 phần tiếp theo.

3.3.3 Regional Proposal Network

Ta đã biết Faster R-CNN sẽ đưa toàn bộ ảnh qua 1 ConvNet để tạo ra 1 feature map chung. Regional Proposal Network cũng như ROI Pooling đều xài chung feature map này.

Để xây dựng Regional Propal Network này, người ta sẽ trượt (slide) một cửa sổ nhỏ trên feature map này và xây dựng 1 mạng học sâu nhỏ để làm 2 tác vụ: phân loại bounding box này có phải là 1 vật thể hay không, và một hàm hồi quy để sinh ra các tọa độ của bounding box. Một cửa sổ

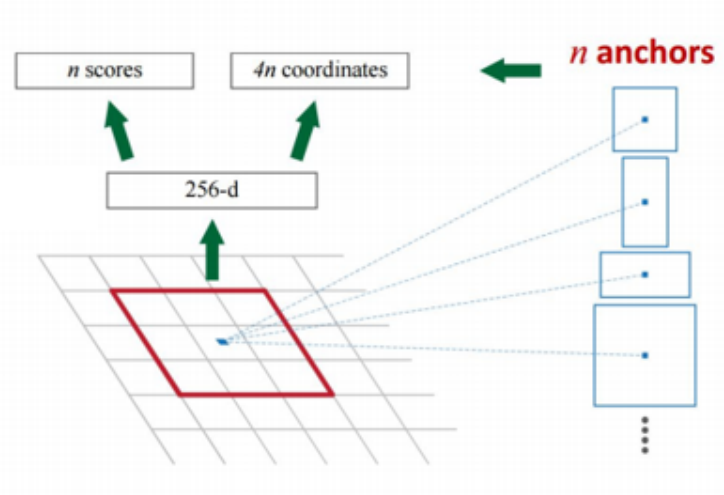


Hình 3.9: Áp dụng slide window trên feature map để sinh ra các bounding box tiềm năng

được chọn tuần tự trên feature map, cửa sổ này sẽ cung cấp thông tin của một vùng trên ảnh ban đầu, từ đó chúng ta có thể học và đánh giá được trong vùng đó có vật thể hay không, có thể tinh chỉnh bounding box thêm không.

Ngoài ra Faster R-CNN còn cung cấp thêm 1 cơ chế đó gọi là anchor box (khung mỏ neo). Đây là những khung được định nghĩa sẵn ban đầu. Đây chính là một cải tiến dành cho việc xử lí các trường hợp có nhiều vật thể

cùng nằm trong 1 bounding box và có nhiều tỉ lệ khác nhau. Cụ thể ở bài báo gốc, người ta sử dụng 3 kích thước khác nhau và 3 tỉ lệ khác nhau cho mỗi kích thước, tổng cộng có 9 anchor box ở mỗi cửa sổ trượt trên feature map tương ứng với các vùng của ảnh ban đầu. Cách tạo dữ liệu để huấn



Hình 3.10: Sử dụng anchor box để tăng tính đa dạng cho các bounding box

luyện cho mạng Regional Proposal Network, với bounding box từ tập ảnh ground truth ta sẽ chỉ chọn ra những anchor box có điểm IoU lớn hơn 0.7 để đánh kết quả là 1, và điểm IoU nhỏ hơn 0.3 để đánh kết quả là 0 đây là cho tác vụ phân loại bounding box có vật thể hay không; cũng như các thông số của bounding box cho tác vụ tinh chỉnh bounding box.

3.3.4 Hàm lỗi kết hợp

Hàm lỗi của Faster R-CNN được thay đổi một chút so với Fast R-CNN. Thêm vào đó là một hàm lỗi của mạng học sâu Regional Proposal Network dựa vào việc: phân loại n anchor box có phải là vật thể / không phải vật thể, tinh chỉnh $4 \times n$ tọa độ của n bounding box chính xác hơn.
Ta sẽ định nghĩa thêm các vector sau:

- p_i : Dự đoán xác suất anchor box i có phải là vật thể

- p_i^* : Nhãn 0/1 của ảnh groundtruth xác định anchor box i có phải là vật thể.
- t_i : Chứa 4 giá trị x, y, w, h định nghĩa một bounding box
- t_i^* : 4 giá trị x, y, w, h chứa các giá trị của bounding box trong ảnh groundtruth.
- N_{cls} là một tham số để chuẩn hóa độ lỗi cho các mini-batch, ở trong bài báo gốc được đặt $N_{cls} = 256$
- N_{box} là một tham số để chuẩn hóa độ lỗi cho các bounding box, ở trong bài báo gốc được đặt N_{box} xấp xỉ 2400.
- λ là một tham số để cân bằng độ lỗi giữa L_{cls} và L_{box}, λ thường được set khoảng 10 (vì số lượng bounding box thường gấp 10 lần số lượng lớp).

Ta sẽ có n cặp (p_i, t_i) để biểu thị cho một bounding box, hàm lỗi $L = L_{cls} + L_{box}$ được chuyển thành tổng độ lỗi cho n bounding box.

$$L(\{p_i, t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{box}} \sum_i p_i^* \times L_1^{smooth}(t_i - t_i^*)$$

$$L_{cls}(p_i, p_i^*) = -p_i^* \log(p_i) - (1 - p_i^*) \log(1 - p_i)$$

3.3.5 Cách train mô hình

Các bước huấn luyện mô hình

1. Pre-train a CNN network on image classification tasks. Huấn luyện một mạng học sâu ConvNet cho tác vụ phân loại ảnh hoặc sử dụng một mô hình pretrain.

2. Huấn luyện mạng Regional Proposal Network cho tác vụ đề xuất những vùng.Tập dữ liệu được tạo dựa trên điểm IoU đã được nói ở phần chi tiết về mạng Regional Proposal Network ở trên.

3. Huấn luyện một mạng Fast R-CNN sử dụng mạng Regional Proposal Network vừa được huấn luyện ở bước trước.Ở bước này 2 mạng của Fast R-CNN và Regional Proposal Network chưa share chung feature map.

4. Huấn luyện cả 2 mạng cùng một lúc sau khi chia sẻ chung feature map.

5. Cuối cùng chúng ta cũng đã có một mạng Faster R-CNN hoàn chỉnh và tiếp tục huấn luyện cũng như fine-tune các tham số.

6. Lặp lại bước 4 và 5 để train lần lượt mạng Fast R-CNN để phân loại hoặc Regional Proposal Network nếu cần thiết.

3.3.6 Đánh giá

Đây là bảng đánh giá chung về thời gian huấn luyện cũng như thời gian để chạy cho một tấm ảnh.

	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Hình 3.11: Thời gian huấn luyện cũng như chạy trên mọt tấm ảnh của 3 mô hình R-CNN, Fast R-CNN, Faster R-CNN

3.4 Mask R-CNN

3.4.1 Giới thiệu

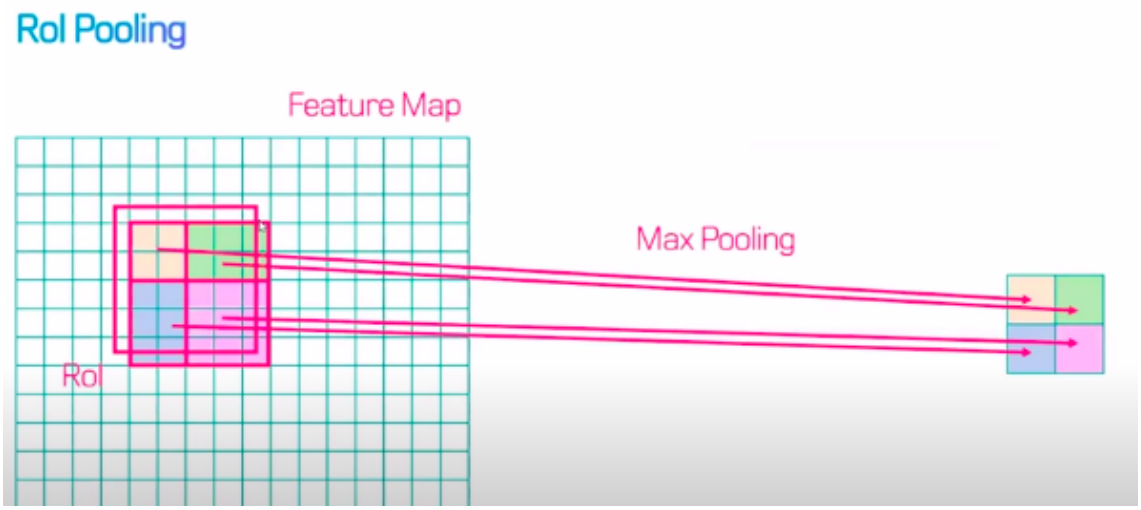
Mask R-CNN [5] là sự kết hợp giữa Faster R-CNN và Fully Convolutional Network (FCN). Ngoài hai output của Faster R-CNN là bounding box và label cho bounding box đó, Mask R-CNN còn sử dụng thêm một nhánh FCN hoạt động song song với nhánh Fast R-CNN để thực hiện phân đoạn đối tượng trong bounding box tương ứng.

Nhánh Faster R-CNN hoạt động gần như không khác gì với phiên bản trước đó, chỉ có một thay đổi nhỏ là sử dụng RoI Align thay cho RoI Pooling. Nhánh thứ hai chính là nhánh phân đoạn, hay tách đối tượng trong vùng RoI ra khỏi nền.

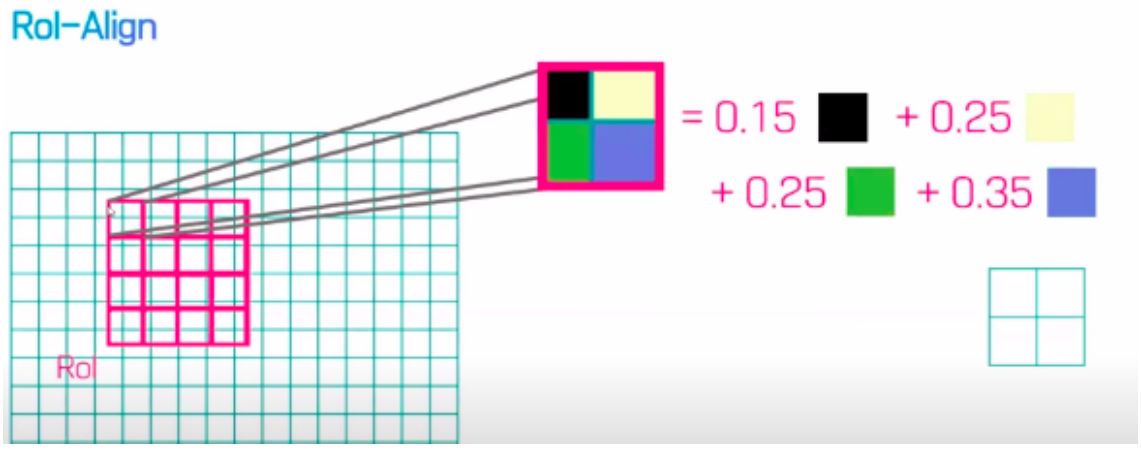
3.4.2 RoI Align

Tác vụ phân lớp RoI trong Faster R-CNN ít bị ảnh hưởng bởi lớp pooling. Tuy nhiên việc phân đoạn đối tượng rất nhạy cảm với các lớp pooling do đây là tác vụ trên pixel. Nếu pooling không cẩn thận, khi downsample làm mất nhiều thông tin, dẫn đến khi tìm ra mask và upsample trở lại, mask thu được sẽ khác rất xa so với GroundTruth. Nhóm tác giả đề xuất RoI Align thay thế cho RoI Pooling. Các giá trị feature map sau khi scale sẽ được nội suy từ feature map gốc chứ không còn pooling như trước. Nhờ đó mà ta không bỏ đi bất kì thông tin nào.

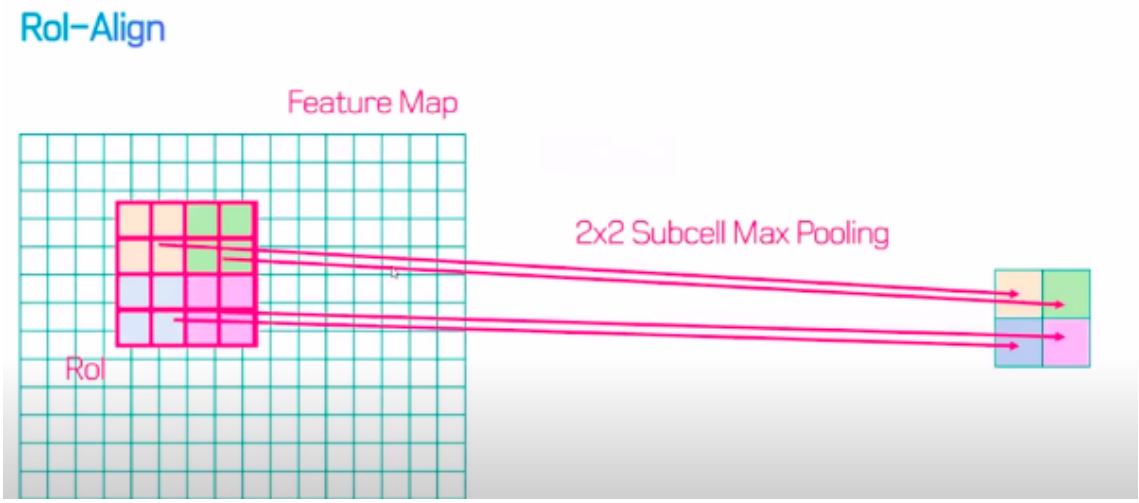
Feature map sau khi scale sẽ được đưa vào hai nhánh: thứ nhất là nhánh Faster R-CNN giống hệt phiên bản cũ để đóng bounding box và phân loại cho đối tượng, thứ hai là nhánh FCN để tách đối tượng trong vùng RoI ra khỏi nền



Hình 3.12: RoI Pooling, vùng RoI được làm tròn cho khớp với các tọa độ rời rạc trên ảnh



Hình 3.13: RoI Align, giá trị trong vùng RoI sẽ được nội suy



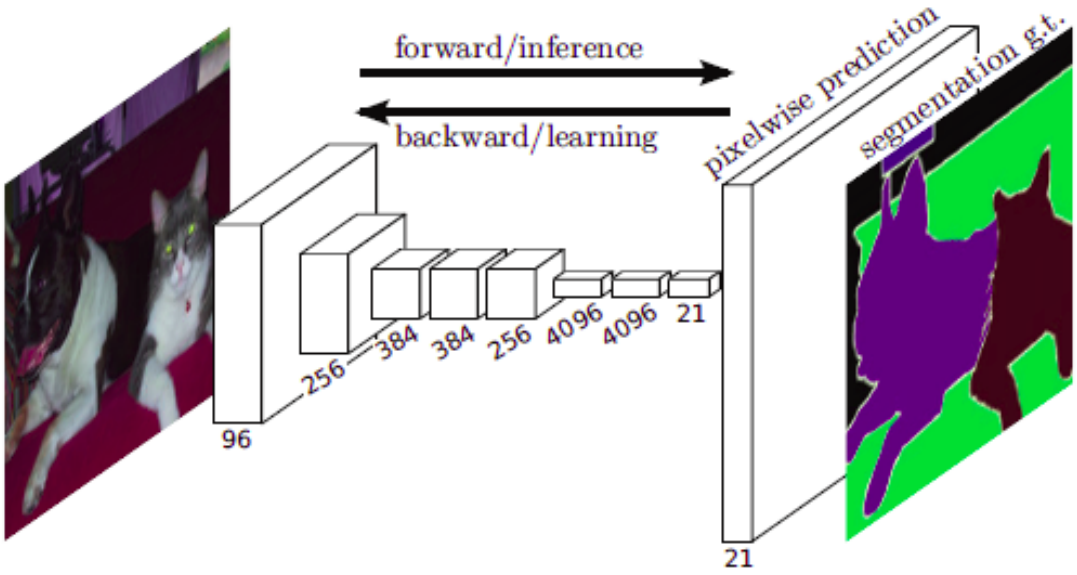
Hình 3.14: RoI Align, sau đó mới thực hiện pooling vùng RoI

3.4.3 Fully Convolutional Network - FCN

FCN thực chất chính là một mạng CNN thông thường, nhưng thay vì sử dụng các lớp FC ở cuối cùng để phân lớp hay gán nhãn cho bức ảnh, thì FCN lại không sử dụng các lớp kết nối đầy đủ, mà nó thực hiện upsample đầu ra của lớp tích chập cuối cùng, hay nói cách khác là feature map, để được feature map có chiều dài và rộng giống như ảnh input, rồi tiến hành tổng hợp đặc trưng cho từng điểm ảnh, nhằm mục đích phân lớp hay gán nhãn cho từng điểm ảnh này.

Cụ thể, giả sử input có số chiều là $W * H * D$, sau khi truyền qua các lớp convolution, W và H ngày càng nhỏ lại, đồng thời D cũng tăng lên. Ta thu được kết quả sau các lớp convolution là $W' * H' * D'$. FCN sau đó sẽ upsample (bằng scale, transpose convolution,...) để feature map có chiều dài và rộng giống với ảnh input: $W * H * D'$. Điều này có nghĩa rằng: với mỗi điểm ảnh, ta thu được một vector đặc trưng có D' chiều.

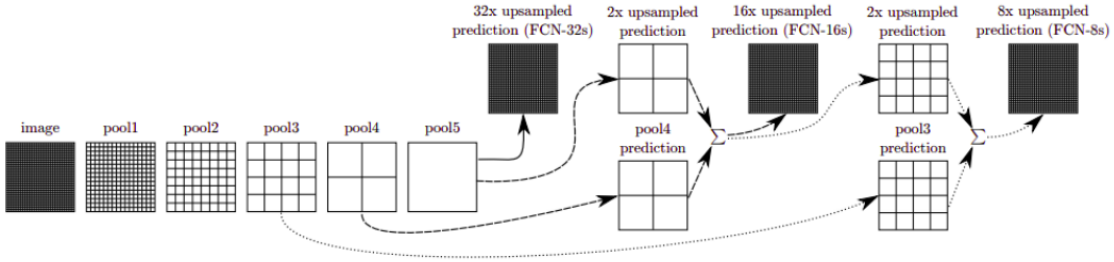
Vector đặc trưng của mỗi điểm ảnh được đưa qua một hàm regression để phân lớp (trong trường hợp Mask R-CNN, chỉ có 2 lớp là object và background - binary classification hay 2-class regression). Trong Mask R-CNN, kết quả cuối cùng sẽ được scale ngược về kích thước ban đầu (do bước RolAlign scale các feature map về cùng kích thước).



Hình 3.15: FCN - Fully Convolutional Network

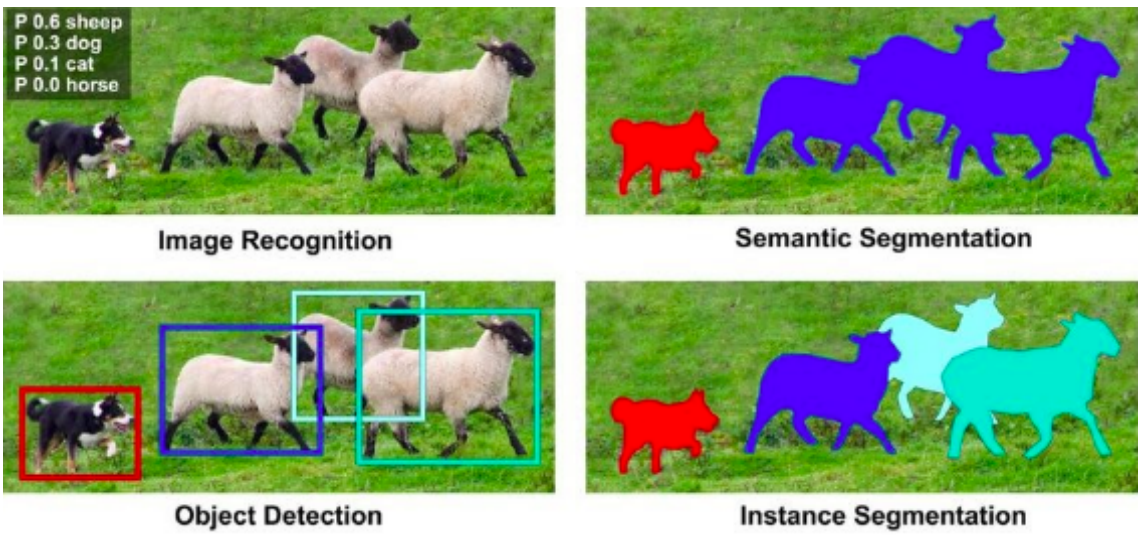
Hình 3.16: Mask R-CNN, pixel-to-pixel classification

Trong bài báo gốc FCN, tác giả còn cải thiện kết quả segmentation bằng cách cho dừng convolution sớm, để tránh việc feature map có chiều dài và rộng trở nên quá nhỏ, dẫn đến việc upsample lên không còn đạt chất lượng tốt.



Hình 3.17: FCN, dừng sớm để có đội phân giải tốt hơn

Cuối cùng, do trước đó, các RoI tiềm năng đã được đề xuất bởi RPN, công việc của FCN lúc này rất nhẹ nhàng (object đã được đóng box, FCN chỉ cần segmentation object vs background). Cũng nhờ RPN, các object khác nhau có khả năng cao sẽ thuộc các RoI khác nhau, nên cuối cùng ta thu được kết quả là Instance Segmentation (FCN gốc khó hoặc thậm chí không làm được điều này).



Hình 3.18: Instance segmentation

Tài liệu tham khảo

[1] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *Int. J. Comput. Vision* 59.2 (Sept. 2004), 167–181. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000022288.19776.77. URL: https://doi.org/10.1023/B:VISI.0000022288.19776.77.

[2] J.R.R. Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* (2013). DOI: 10.1007/s11263-013-0620-5. URL: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf.

[3] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation.* 2014. arXiv: 1311.2524 [cs.CV].

[4] Ross B Girshick. "Fast R-CNN. CoRR abs/1504.08083 (2015)". In: *arXiv preprint arXiv:1504.08083* (2015).

[5] Kaiming He et al. "Mask r-cnn. corr abs/1703.06870 (2017)". In: *arXiv preprint arXiv:1703.06870* (2017).

[6] Zhong-Qiu Zhao et al. *Object Detection with Deep Learning: A Review.* 2019. arXiv: 1807.05511 [cs.CV].

[7] Zhengxia Zou et al. *Object Detection in 20 Years: A Survey.* 2019. arXiv: 1905.05055 [cs.CV].

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances*

in Neural Information Processing Systems. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105.

[9] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [ ]". In: *: https://arxiv. org/pdf/1506.014 pdf (: 06.01. 2019)* ().