

# Faster R-CNN và Mask R-CNN

Đại học Bách Khoa Hà Nội  
Nguyễn Đức Thắng  
KSTN Toán Tin K61  
Báo cáo tại nhóm nghiên cứu

06/2020

# Nội dung

- 1 Giới thiệu về các bài toán trong Computer vision
- 2 Một số mạng CNN hiện đại (giới thiệu)
- 3 Transfer Learning (giới thiệu)
- 4 Mô hình R-CNN
- 5 Mô hình Fast R-CNN
- 6 Mô hình Faster R-CNN
- 7 Mô hình Mask R-CNN

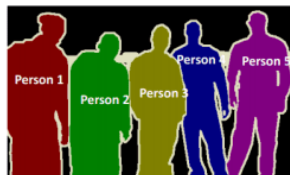
# Giới thiệu về các bài toán trong Computer Vision



Object Detection



Semantic Segmentation



Instance Segmentation

Một số bài toán cơ bản trong Computer vision:

- Phân loại hình ảnh (image classification)
- Định vị vật thể (object localization)
- Phát hiện đối tượng (object detection)
- Phân đoạn đối tượng (object segmentation): semantic segmentation và instance segmentation
- Chú thích cho ảnh (image captioning)

# Một số mạng CNN hiện đại

Một số mạng CNN hiện đại (theo thời gian phát hành) (giới thiệu ở các bài sau):

- AlexNet
- VggNet
- GooLeNet
- ResNet
- SqueezeNet

# Transfer learning

Các loại transfer learning:

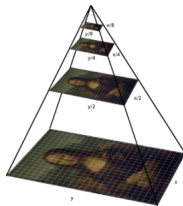
- Feature extractor
- Fine tuning

Tham khảo thêm:

- Đồ án II - Nguyễn Đức Thắng
- nttuan8.com

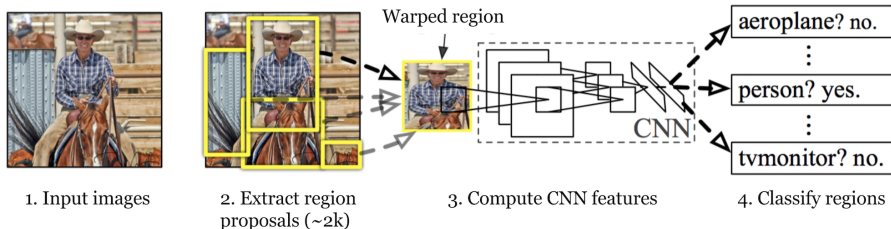
# Mô hình R-CNN

Phương pháp cửa sổ trượt:



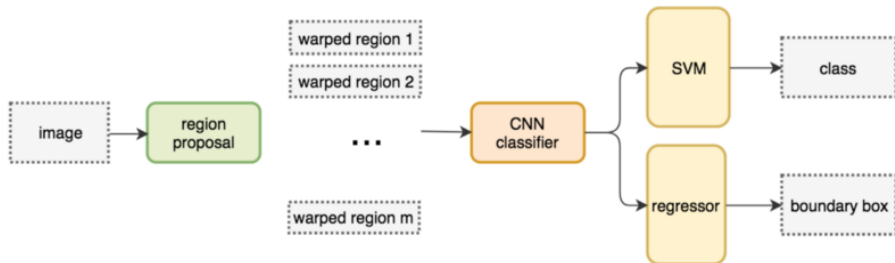
# Mô hình R-CNN

- 1 Dùng thuật toán Selective Search để lấy ra khoảng 2000 bounding box.
- 2 Với mỗi bounding box, ta xác định xem nó là đối tượng nào.



Hình: Pipeline mạng R-CNN

# Mô hình R-CNN



Hình: Mô phỏng mô hình R-CNN



# Thuật toán Selective Search



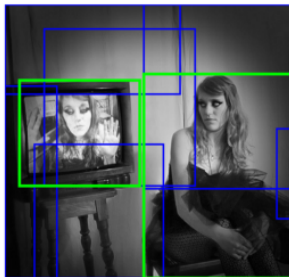
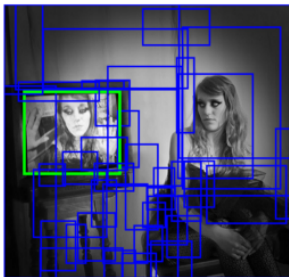
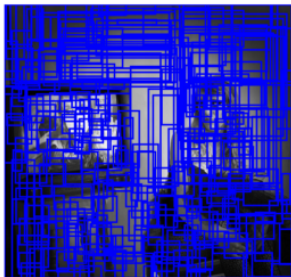
Input Image



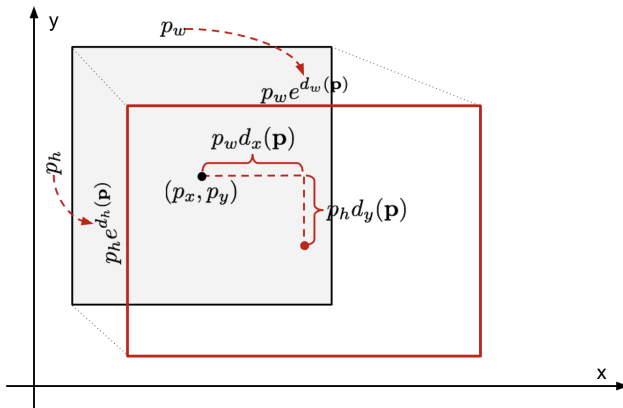
Output Image

Hình: Thuật toán graph based image segmentation

# Thuật toán Selective Search

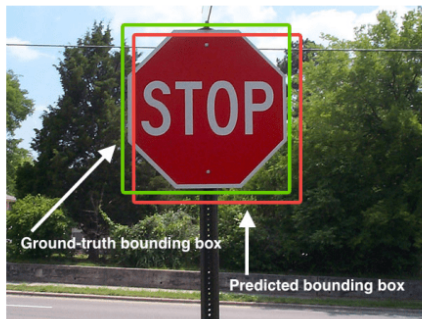


# Bounding box regression



Hình: Mô tả công thức chuyển đổi của bounding box

# Intersection over Union (IoU)



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

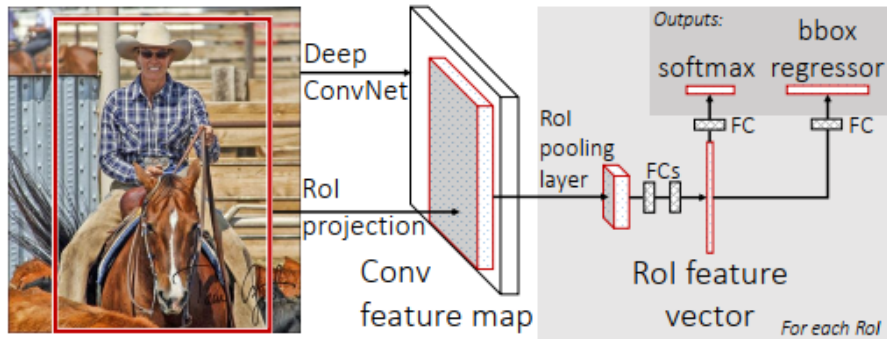
Hình: Intersection over Union

# Mô hình Fast R-CNN

Nhược điểm của R-CNN:

- 1 **Training qua nhiều giai đoạn**
- 2 **Training tốn kém không gian và thời gian**
- 3 **Phát hiện đối tượng chậm**

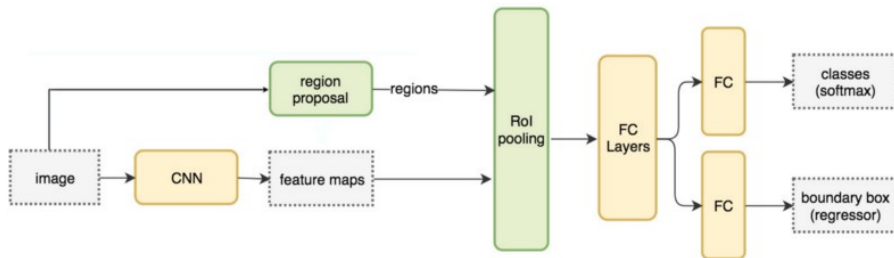
# Mô hình Fast R-CNN



Hình: Mô hình Fast R-CNN

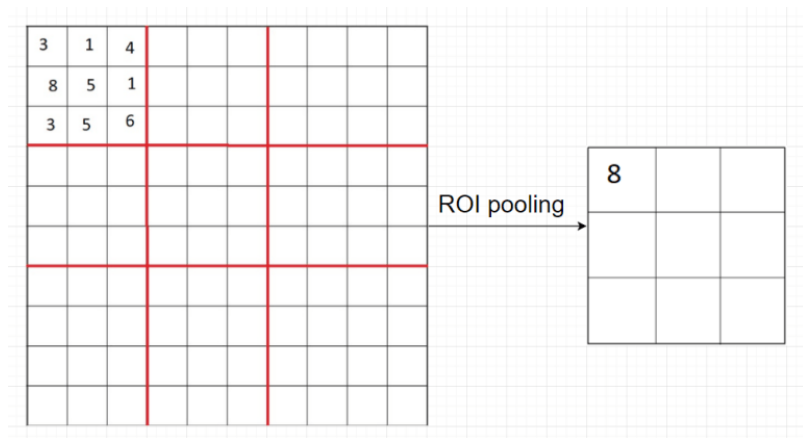
# Mô hình Fast R-CNN

Luồng xử lý của Fast R-CNN có thể tóm tắt như sau:



Hình: Luồng hoạt động của mô hình Fast R-CNN

# Roi pooling



Hình: Hoạt động của Roi Pooling



# Hàm mất mát của Fast R-CNN

Hàm mất mát bằng tổng của chi phí phân lớp và bounding box regression:

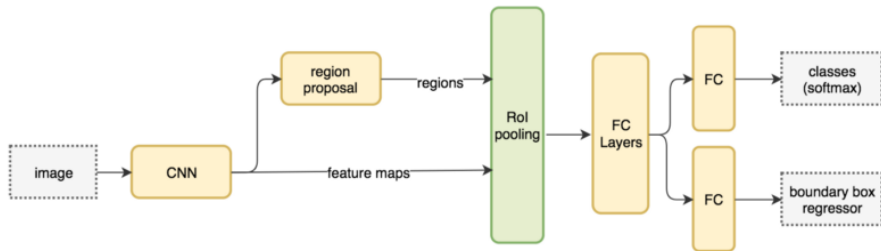
$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}}$$

Xem thêm trong báo cáo đồ án II - Nguyễn Đức Thắng

# Mô hình Faster R-CNN

Mặc dù là một mô hình đơn lẻ duy nhất, kiến trúc này là kết hợp của hai modules:

- Mạng đề xuất khu vực (Region Proposal Network, viết tắt là RPN). Mạng CNN để đề xuất các vùng và loại đối tượng cần xem xét trong vùng.
- Fast R-CNN: Mạng CNN để trích xuất các features từ các region proposal và trả ra các bounding box và nhãn.



# Anchor



Hình: Anchor

Các anchor được gán mức là positive/negative (foreground/background) dựa vào IoU với ground-truth bounding box theo quy tắc sau:

- Các anchor có tỉ lệ IoU lớn nhất với ground-truth box sẽ được gọi là positive.
- Các anchor có tỉ lệ  $\text{IoU} \geq 0.7$  sẽ được coi là positive.
- Các anchor có tỉ lệ  $\text{IoU} < 0.3$  sẽ được coi là negative.
- Các anchor có IoU nằm trong khoảng  $0.3 \leq x < 0.7$  sẽ được coi là trung tính (neutral) và sẽ không được sử dụng trong quá trình huấn luyện mô hình.

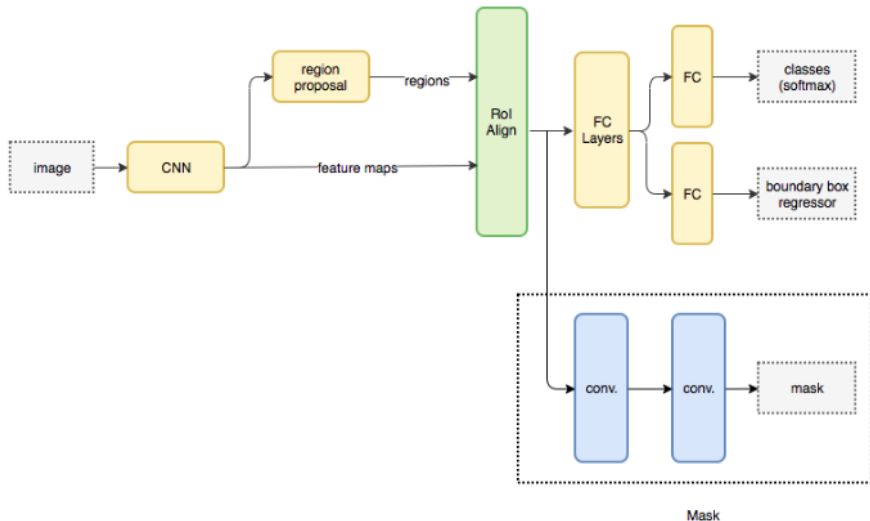
Hàm mất mát: Xem thêm tại báo cáo

# Non-maxima suppression

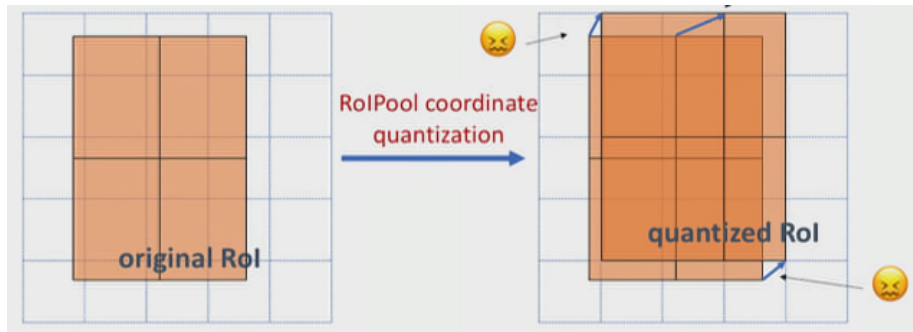
Các region proposal thu được sau RPN sẽ bị chồng chéo lên nhau rất nhiều nên cần một cơ chế để lọc các vùng đó, và đồng thời cũng tăng tốc độ xử lý của mô hình. Một phương pháp được đề xuất đó là Non-maxima suppression. Các bước thực hiện như sau:

- **Bước 1:** Chọn ra anchor box (A) có xác suất là foreground lớn nhất trong tập Input.
- **Bước 2:** Thêm A vào tập Output.
- **Bước 3:** Loại bỏ A và các anchor box trong tập Input mà có hệ số IoU với A lớn hơn 0.5 ra khỏi tập Input.
- **Bước 4:** Kiểm tra nếu tập Input rỗng hoặc tập Output đủ anchor mong muốn thì dừng lại, nếu không quay lại bước 1.

# Mô hình Mask R-CNN

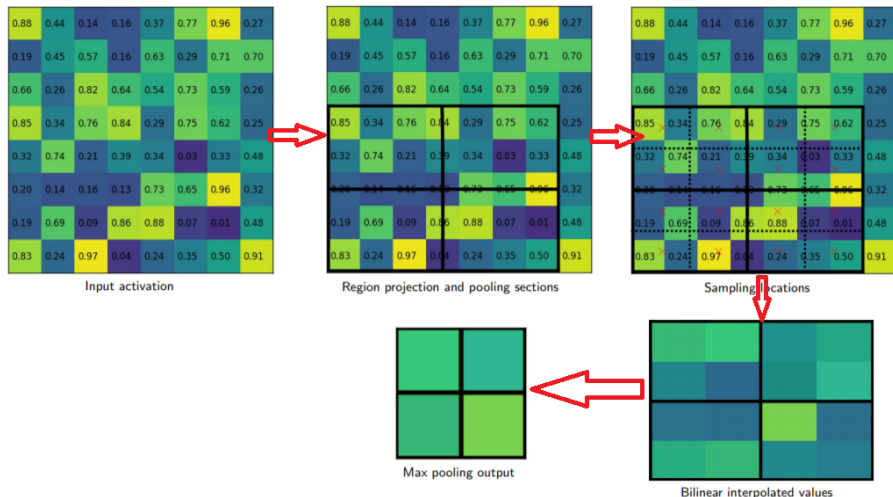


# Roi Align



Hình: Vấn đề lượng tử hoá của RoIPooling

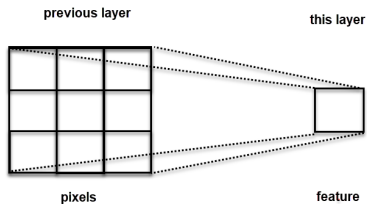
# Roi Align



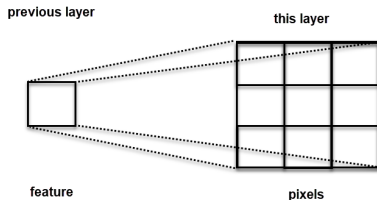
Hình: Mô tả phương pháp RoiAlign



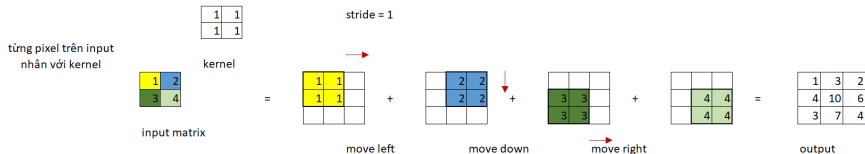
# Deconvolution



ConvNet



DeconvNet



# Unpooling

0.1	0.5	<b>1.2</b>	-0.7
<b>0.8</b>	-0.2	-0.5	0.3
0.4	<b>0.9</b>	-0.1	-0.2
-0.6	0.1	<b>0.5</b>	0.3

max-pooling

0.8	1.2
0.9	0.5

		x	
x			
	x		
		x	

max locations

0	0	<b>0.5</b>	0
<b>1.3</b>	0	0	0
0	<b>0.4</b>	0	0
0	0	<b>0.1</b>	0

unpooling

1.3	0.5
0.4	0.1

# Tham khảo thêm

- 1 Image segmentation:  
<https://phamdinhhkhanh.github.io/2020/06/10/ImageSegmentation.html>
- 2 Các mạng CNN hiện đại:  
<https://phamdinhhkhanh.github.io/2020/05/31/CNNHistory.html>
- 3 Transfer learning:  
<https://phamdinhhkhanh.github.io/2020/04/15/TransferLearning.html>