# Bài 12 - Các thuật toán Object Detection

**29 Sep 2019 - phamdinhkhanh**

*Menu*



# 1. Object detection là gì?

Trước khi đi vào tìm hiểu object detection là gì, bạn đọc cần nắm vững một số khái niệm về mô hình phân loại ảnh (image classification), kiến trúc Convolutional neural network Pham Dinh Khanh (https://www.kaggle.com/phamdinhkhanh/convolutional-neural-network-p1), quá trình hình thành và phát triển mạng CNN đến nay Blog dlapplication (https://dlapplications.github.io/2018-07-06-CNN/).

Sau khi đã đọc các bài hướng dẫn trên hãy quay lại bài viết này, bạn đọc sẽ hiểu những gì mà tôi trình bày sau đây dễ dàng hơn. Chúng ta cùng bắt đầu:

Sẽ khá khó cho người mới bắt đầu để phân biệt các nhiệm vụ khác nhau của computer vision. Chẳng hạn như phân loại hình ảnh (image classification) là gì? Định vị vật thể (object localization) là gì? Sự khác biệt giữa định vị vật thể (object localization) và phát hiện đối tượng (object detection) là gì? Đây là những khái niệm có thể gây nhầm lẫn, đặc biệt là khi cả ba nhiệm vụ đều liên quan đến nhau. Hiểu một cách đơn giản:

- Phân loại hình ảnh (image classification): liên quan đến việc gán nhãn cho hình ảnh.
- Định vị vật thể (object localization): liên quan đến việc vẽ một hộp giới hạn (bounding box) xung quanh một hoặc nhiều đối tượng trong hình ảnh nhằm khoanh vùng đối tượng.
- Phát hiện đối tượng (object detection): Là nhiệm vụ khó khăn hơn và là sự kết hợp của cả hai nhiệm vụ trên: Vẽ một bounding box xung quanh từng đối tượng quan tâm trong ảnh và gán cho chúng một nhãn. Kết hợp cùng nhau, tất cả các vấn đề này được gọi là object recognition hoặc object detection.

Bài viết này sẽ giới thiệu một cách khái quát các vấn đề của object detection và các mô hình deep learning state-of-art được thiết kế để giải quyết nó.

Sau khi đọc bài này, bạn đọc sẽ biết:

- Phân biệt được các tác vụ cơ bản trong computer vision: Image classification, object localization, object detection, object segmentation, image captioning.
- Lịch sử hình thành, phát triển và đặc điểm cấu trúc của các thuật toán object detection bao gồm 2 nhóm chính:

Top

    - Họ các mô hình R-CNN (Region-Based Convolutional Neural Networks) giải quyết các nhiệm vụ định vị vật thể và nhận diện vật thể.
    - Họ các mô hình YOLO (You Only Look Once), là một nhóm kỹ thuật thứ hai để nhận dạng đối tượng được thiết kế để nhận diện vật thể real time.

# 2. Như thế nào là nhận dạng đối tượng?

Nhận dạng đối tượng là một thuật ngữ chung để mô tả một tập hợp các nhiệm vụ thị giác máy tính có liên quan liên quan đến việc xác định các đối tượng trong ảnh kỹ thuật số.

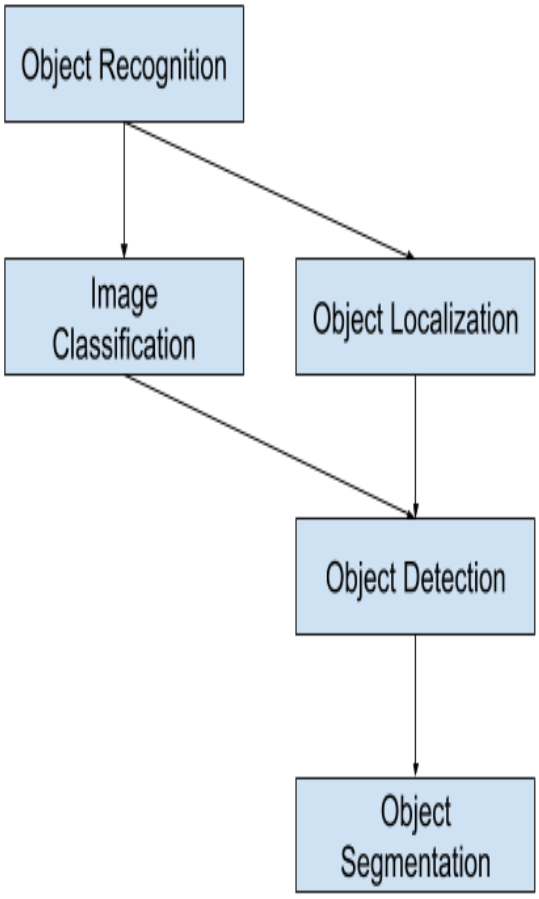
Phân loại hình ảnh liên quan đến việc dự đoán lớp của một đối tượng trong một hình ảnh. Định vị vật thể đề cập đến việc xác định vị trí của một hoặc nhiều đối tượng trong một hình ảnh và vẽ bounding box xung quanh chúng. Phát hiện đối tượng kết hợp hai nhiệm vụ trên và thực hiện cho một hoặc nhiều đối tượng trong hình ảnh. Chúng ta có thể phân biệt giữa ba nhiệm vụ thị giác máy tính cơ bản trên thông qua input và output của chúng như sau:

- **Phân loại hình ảnh**: Dự đoán nhãn của một đối tượng trong một hình ảnh.
    - Input: Một hình ảnh với một đối tượng, chẳng hạn như một bức ảnh.
    - Output: Nhãn lớp (ví dụ: một hoặc nhiều số nguyên được ánh xạ tới nhãn lớp).
- **Định vị đối tượng**: Xác định vị trí hiện diện của các đối tượng trong ảnh và cho biết vị trí của chúng bằng bounding box.
    - Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.
    - Output: Một hoặc nhiều bounding box được xác định bởi tọa độ tâm, chiều rộng và chiều cao.
- **Phát hiện đối tượng**: Xác định vị trí hiện diện của các đối tượng trong bounding box và nhãn của các đối tượng nằm trong một hình ảnh.
    - Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.
    - Output: Một hoặc nhiều bounding box và nhãn cho mỗi bounding box.

Một số định nghĩa khác cũng rất quan trọng trong computer vision là phân đoạn đối tượng (object segmentation), trong đó các đối tượng được nhận dạng bằng cách làm nổi bật các pixel cụ thể của đối tượng thay vì bounding box. Và image captioning kết hợp giữa các kiến trúc mạng CNN và LSTM để đưa ra các lý giải về hành động hoặc nội dung của một bức ảnh.

Bên dưới là sơ đồ tổng hợp các tác vụ của computer vision.

Top




**Hình 1**: Sơ đồ các mối liên hệ giữa các tác vụ trong computer vision.

Chúng ta cũng có thể hiểu object recognition tương tự như object detection theo một cách tương đối nào đó. Gần đây thì Object Recognition đã trở thành một phần của của cuộc thi ILSVRC (http://image-net.org/challenges/LSVRC/), một trong những cuộc thi nhận diện ảnh lớn nhất hành tinh.

Điểm khác biệt nữa trong các mô hình image classification so với Object Recognition đó là mô hình image classification có hàm loss function chỉ dựa trên sai số giữa nhãn dự báo và nhãn thực tế trong khi object detection đánh giá dựa trên sai số giữa nhãn dự báo và sai số khung hình dự báo so với thực tế.

# 3. Các thuật ngữ sử dụng trong bài

- region proposal: Vùng đề xuất, là những vùng mà có khả năng chứa đối tượng hoặc hình ảnh ở bên trong nó.
- bounding box: Là hình chữ nhật được vẽ bao quan đối tượng nhằm xác định đối tượng.
- offsets: Là các tham số giúp xác định bounding box bao gồm tâm của bounding box $(x, y)$ và chiều dài, chiều rộng $(w, h)$.
- anchor box: Chính là một bounding box cơ sở để xác định bounding box bao quanh vật thể dựa trên các phép dịch tâm và scale kích thước chiều dài, rộng. Mỗi loại anchor box sẽ phù hợp để tìm ra bounding box cho 1 loại vật thể đặc trưng. Chẳng hạn vật thể là con người thường có chiều cao > chiều rộng trong khi đoàn tàu sẽ có chiều rộng lớn hơn nhiều lần chiều cao.

Top

- feature: Các đặc trưng được tạo ra từ một mạng deep CNN chẳng hạn Alexnet hoặc VGG16 giúp nhận diện nhãn của hình ảnh.
- pipeline: Là một tợp hợp các bước xử lý liên tiếp nhận đầu vào là dữ liệu (ảnh, âm thanh, các trường dữ liệu) và trả ra kết quả dự báo ở output.

# 4. Lớp các mô hình họ R-CNN

R-CNN (regions with CNN features) là lớp các mô hình xác định vùng đặc trưng dựa trên các mạng CNN được phát triển bởi Ross Girshick (http://www.rossgirshick.info/) và các cộng sự. Lớp các mô hình này gồm 3 mô hình chính là R-CNN, Fast R-CNN và Faster-RCNN được thiết kế cho các nhiệm vụ định vị vật thể và nhận diện vật thể.

Chúng ta sẽ đi vào tìm hiểu các mô hình này.

## 4.1. R-CNN (2014)

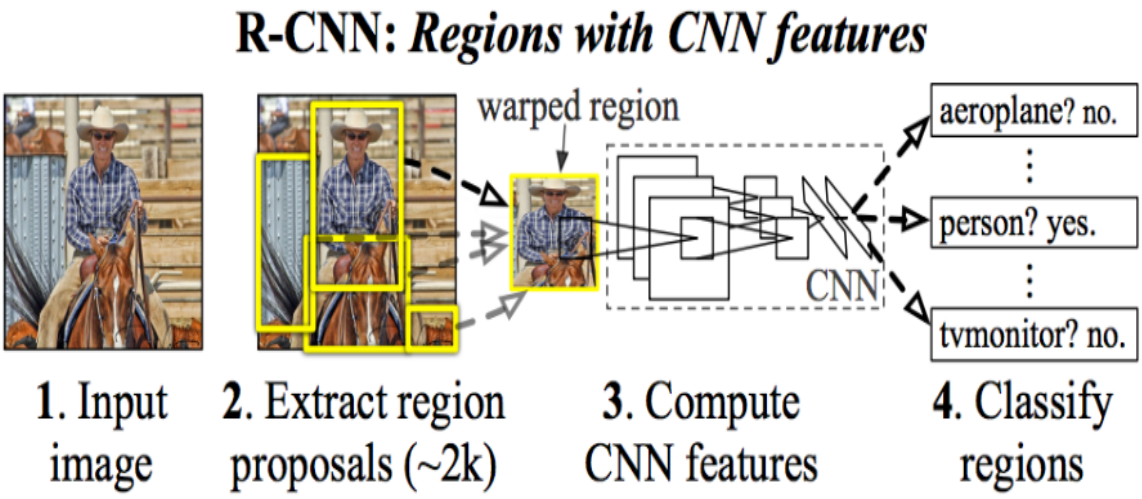
R-CNN được giới thiệu lần đầu vào 2014 bởi Ross Girshick và các cộng sự ở UC Berkeley một trong những trung tâm nghiên cứu AI hàng đầu thế giới trong bài báo Rich feature hierarchies for accurate object detection and semantic segmentation (https://arxiv.org/abs/1311.2524).

Nó có thể là một trong những ứng dụng nền móng đầu tiên của mạng nơ ron tích chập đối với vấn đề định vị, phát hiện và phân đoạn đối tượng. Cách tiếp cận đã được chứng minh trên các bộ dữ liệu điểm chuẩn, đạt được kết quả tốt nhất trên bộ dữ liệu VOC-2012 và bộ dữ liệu phát hiện đối tượng ILSVRC-2013 gồm 200 lớp.

Kiến trúc của R-CNN gồm 3 thành phần đó là:

- Vùng đề xuất hình ảnh (Region proposal): Có tác dụng tạo và trích xuất các vùng đề xuất chứa vật thể được bao bởi các bounding box.
- Trích lọc đặc trưng (Feature Extractor): Trích xuất các đặc trưng giúp nhận diện hình ảnh từ các region proposal thông qua các mạng deep convolutional neural network.
- Phân loại (classifier): Dựa vào input là các features ở phần trước để phân loại hình ảnh chứa trong region proposal về đúng nhãn.

Kiến trúc của mô hình được mô tả trong biểu đồ bên dưới:




Top

> **Hình 2**: Sơ đồ pipeline xử lý trong mô hình mạng R-CNN (được trích xuất từ bài báo gốc). Ta có thể nhận thấy các hình ảnh con được trích xuất tại bước 2 với số lượng rất lớn (khoảng 2000 region proposals). Tiếp theo đó áp dụng một mạng deep CNN để tính toán các feature tại bước 3 và trả ra kết quả dự báo nhãn ở bước 4 như một tác vụ image classification thông thường.

Một kỹ thuật được sử dụng để đề xuất các region proposal hoặc các bounding box chứa các đối tượng tiềm năng trong hình ảnh được gọi là “selective search”, các region proposal có thể được phát hiện bởi đa dạng những thuật toán khác nhau. Nhưng điểm chung là đều dựa trên tỷ lệ IoU giữa bounding box và ground truth box mà bạn đọc sẽ được tìm hiểu ở bài viết tiếp theo giới thiệu về mạng SSD.

Trích xuất đặc trưng về bản chất là một mạng CNN học sâu, ở đây là AlexNet, mạng đã giành chiến thắng trong cuộc thi phân loại hình ảnh ILSVRC-2012. Đầu ra của CNN là một vectơ 4096 chiều mô tả nội dung của hình ảnh được đưa đến một mô hình SVM tuyến tính để phân loại.

Đây là một ứng dụng tương đối đơn giản và dễ hiểu của CNN đối với vấn đề object localization và object detection. Một nhược điểm của phương pháp này là chậm, đòi hỏi phải vượt qua nhiều module độc lập trong đó có trích xuất đặc trưng từ một mạng CNN học sâu trên từng region proposal được tạo bởi thuật toán đề xuất vùng chứa ảnh. Đây là một vấn đề chính cần giải quyết vì bài viết mô tả mô hình hoạt động trên khoảng 2000 vùng được đề xuất cho mỗi hình ảnh tại thời điểm thử nghiệm.

Mã nguồn Python (Caffe) và MatLab cho R-CNN như được mô tả trong bài viết đã được cung cấp trong kho GitHub repository của R-CNN (https://github.com/rbgirshick/rcnn).

# 4.2. Fast R-CNN (2015)

Dựa trên thành công của R-CNN, Ross Girshick (lúc này đã chuyển sang Microsoft Research) đề xuất một mở rộng để giải quyết vấn đề của R-CNN trong một bài báo vào năm 2015 với tiêu đề rất ngắn gọn Fast R-CNN (https://arxiv.org/abs/1504.08083).

Bài báo chỉ ra những hạn chế của R-CNN đó là:

- Training qua một pipeline gồm nhiều bước: Pipeline liên quan đến việc chuẩn bị và vận hành ba mô hình riêng biệt.
- Chi phí training tốn kém về số lượng bounding box và thời gian huấn luyện: Mô hình huấn luyện một mạng CNN học sâu trên rất nhiều region proposal cho mỗi hình ảnh nên rất chậm.
- Phát hiện đối tượng chậm: Tốc độ xử lý không thể đảm bảo realtime.

Trước đó một bài báo đã đề xuất phương pháp để tăng tốc kỹ thuật được gọi là mạng tổng hợp kim tự tháp - Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition (https://arxiv.org/abs/1406.4729), hoặc SPPnets vào năm 2014. Phương pháp này đã tăng tốc độ trích xuất features nhờ lan truyền thuận trên bộ nhớ đệm.

Điểm đột phá của Fast R-CNN là sử dụng một single model thay vì pipeline để phát hiện region và classification cùng lúc.

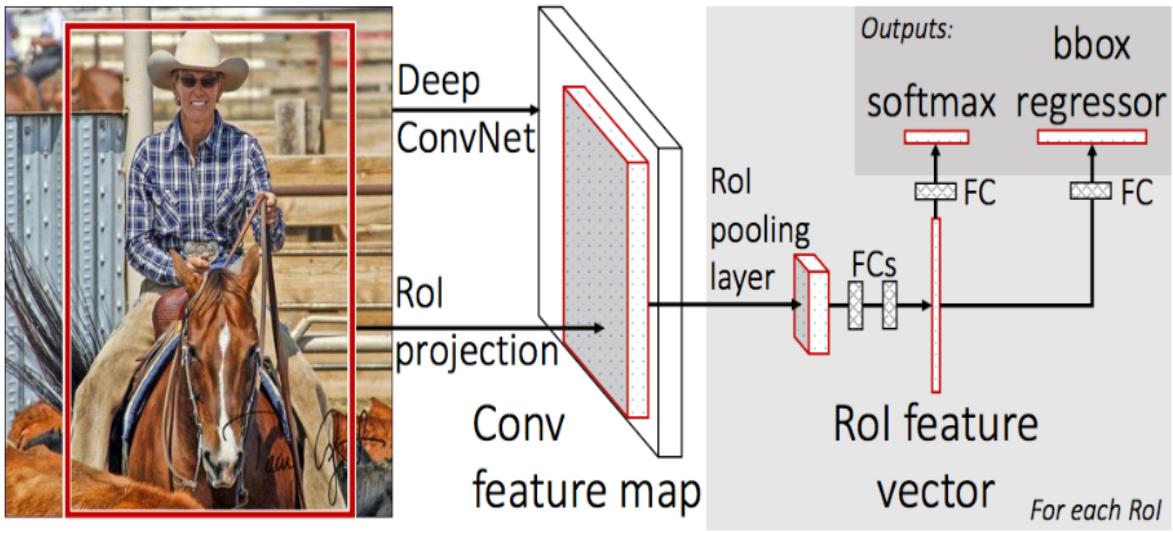
Kiến trúc của mô hình trích xuất từ bức ảnh một tập hợp các region proposals làm đầu vào được truyền qua mạng deep CNN. Một pretrained-CNN, chẳng hạn VGG-16, được sử dụng để trích lọc features. Phần cuối của deep-CNN là một custom layer được gọi là layer vùng quan tâm (Region

Top

of Interest Pooling - RoI Pooling) có tác dụng trích xuất các features cho một vùng ảnh input nhất định.

Sau đó các features được kết bởi một lớp fully connected. Cuối cùng mô hình chia thành hai đầu ra, một đầu ra cho dự đoán nhãn thông qua một softmax layer và một đầu ra khác dự đoán bounding box (kí hiệu là bbox) dựa trên hồi qui tuyến tính. Quá trình này sau đó được lặp lại nhiều lần cho mỗi vùng RoI trong một hình ảnh.

Kiến trúc của mô hình được tóm tắt trong hình dưới đây, được lấy từ bài báo.




**Hình 3**: Kiến trúc single model Fast R-CNN (được trích xuất từ bài báo gốc). Ở bước đầu ta áp dụng một mạng Deep CNN để trích xuất ra feature map. Thay vì warp image của region proposal như ở R-CNN chúng ta xác dịnh ngay vị trí hình chiếu của của region proposal trên feature map thông qua phép chiếu RoI projection. Vị trí này sẽ tương đối với vị trí trên ảnh gốc. Sau đó tiếp tục truyền output qua các layer RoI pooling layer và các Fully Connected layers để thu được RoI feature véc tơ. Sau đó kết quả đầu ra sẽ được chia làm 2 nhánh. 1 Nhánh giúp xác định phân phối xác suất theo các class của 1 vùng quan tâm RoI thông qua hàm softmax và nhánh còn xác định tọa độ của bounding box thông qua hồi qui các offsets.

Mô hình này nhanh hơn đáng kể cả về huấn luyện và dự đoán, tuy nhiên vẫn cần một tập hợp các region proposal được đề xuất cùng với mỗi hình ảnh đầu vào.

Mã nguồn Python và C ++ (Caffe) cho Fast R-CNN như được mô tả trong bài báo xem tại Fast - RCNN (https://github.com/rbgirshick/fast-rcnn).

# 4.3. Faster R-CNN (2016)

Kiến trúc mô hình đã được cải thiện hơn nữa về cả tốc độ huấn luyện và phát hiện được đề xuất bởi Shaoqing Ren và các cộng sự tại Microsoft Research trong bài báo năm 2016 có tiêu đề Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks Top

(https://arxiv.org/abs/1506.01497). Dịch nghĩa là “Faster R-CNN: Hướng tới phát hiện đối tượng theo thời gian thực với các mạng đề xuất khu vực”.

Kiến trúc này mang lại độ chính xác cao nhất đạt được trên cả hai nhiệm vụ phát hiện và nhận dạng đối tượng tại các cuộc thi ILSVRC-2015 và MS COCO-2015.

Kiến trúc được thiết kế để đề xuất và tinh chỉnh các region proposals như là một phần của quá trình huấn luyện, được gọi là mạng đề xuất khu vực (Region Proposal Network), hoặc RPN. Các vùng này sau đó được sử dụng cùng với mô hình Fast R-CNN trong một thiết kế mô hình duy nhất. Những cải tiến này vừa làm giảm số lượng region proposal vừa tăng tốc hoạt động trong thời gian thử nghiệm mô hình lên gần thời gian thực với hiệu suất tốt nhất. Tốc độ là 5fps trên một GPU.

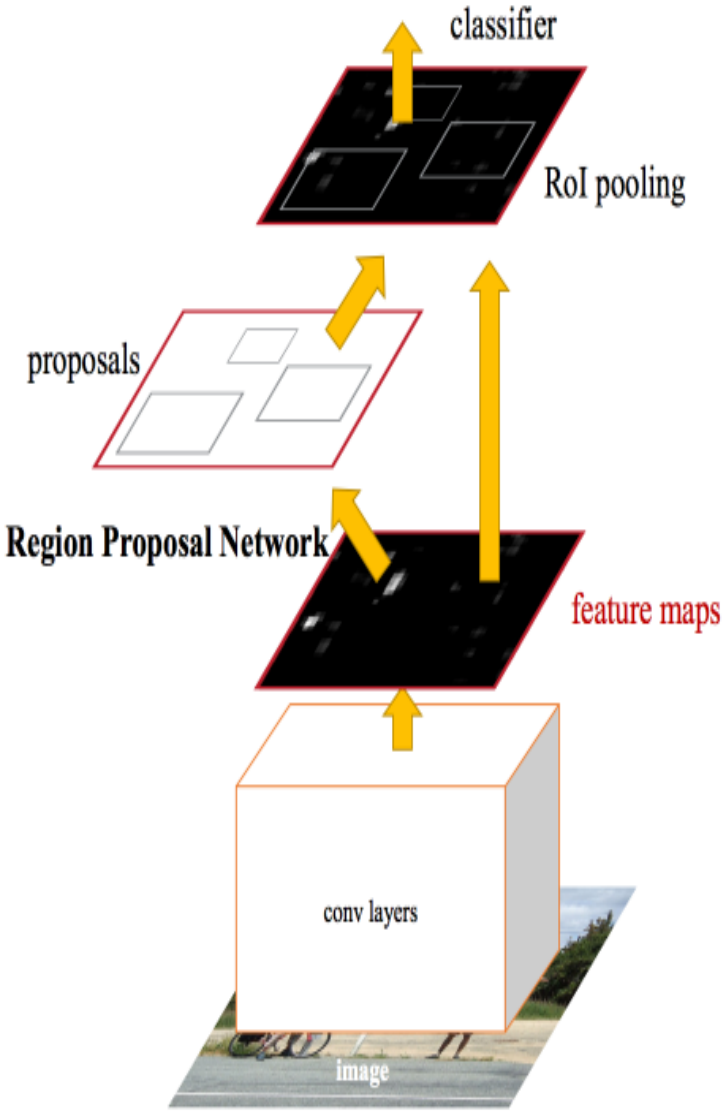
Mặc dù là một mô hình đơn lẻ duy nhất, kiến trúc này là kết hợp của hai modules:

- 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.

Cả hai modules hoạt động trên cùng một output của một mạng deep CNN. Mạng RPN hoạt động như một cơ chế attention cho mạng Fast R-CNN, thông báo cho mạng thứ hai về nơi cần xem hoặc chú ý.

Kiến trúc của mô hình được tổng kết thông qua sơ đồ bên dưới:

Top




**Hình 4**: Kiến trúc mô hình Faster R-CNN (được trích xuất từ bài báo gốc). Ở giai đoạn sớm sử dụng một mạng deep CNN để tạo ra một feature map. Khác với Fast R-CNN, kiến trúc này không tạo RoI ngay trên feature map mà sử dụng feature map làm đầu vào để xác định các region proposal thông qua một RPN network. Đồng thời feature maps cũng là đầu vào cho classifier nhằm phân loại các vật thể của region proposal xác định được từ RPN network.

RPN hoạt động bằng cách lấy đầu ra của một mạng pretrained deep CNN, chẳng hạn như VGG-16, và truyền feature map vào một mạng nhỏ và đưa ra nhiều region proposals và nhãn dự đoán cho chúng. Region proposals là các bounding boxes, dựa trên các anchor boxes hoặc hình dạng được xác định trước được thiết kế để tăng tốc và cải thiện khả năng đề xuất vùng. Dự đoán của nhãn được thể hiện dưới dạng nhị phân cho biết region proposal có xuất hiện vật thể hoặc không.

Một quy trình huấn luyện xen kẽ được sử dụng trong đó cả hai mạng con được đào tạo cùng một lúc. Điều này cho phép các tham số trong feature dectector của deep CNN được tinh chỉnh cho cả hai tác vụ cùng một lúc.

Top

Tại thời điểm viết, kiến trúc Faster R-CNN này là đỉnh cao của họ model R-CNN và tiếp tục đạt được kết quả gần như tốt nhất trong các nhiệm vụ nhận diện đối tượng. Một mô hình mở rộng hỗ trợ cho phân đoạn hình ảnh, được mô tả trong bài báo năm 2017 có tựa đề Mask R-CNN (https://arxiv.org/abs/1703.06870).

Mã nguồn Python và C ++ (Caffe) cho Fast R-CNN như được mô tả trong bài báo có thể tham khảo tại Faster R-CNN (https://github.com/rbgirshick/py-faster-rcnn).

# 5. Lớp các mô hình họ YOLO

Một họ mô hình nhận dạng đối tượng phổ biến khác được gọi chung là YOLO. YOLO không phải là bạn chỉ sống một lần đâu nhé, nó có nghĩa là bạn chỉ nhìn một lần (you only look one), được phát triển bởi Joseph Redmon (https://pjreddie.com/), và các cộng sự.

Các mô hình R-CNN nói chung có thể chính xác hơn, tuy nhiên họ mô hình YOLO nhanh hơn rất rất nhiều so với R-CNN, và thậm chí đạt được việc phát hiện đối tượng trong thời gian thực.

## 5.1. YOLO (2015)

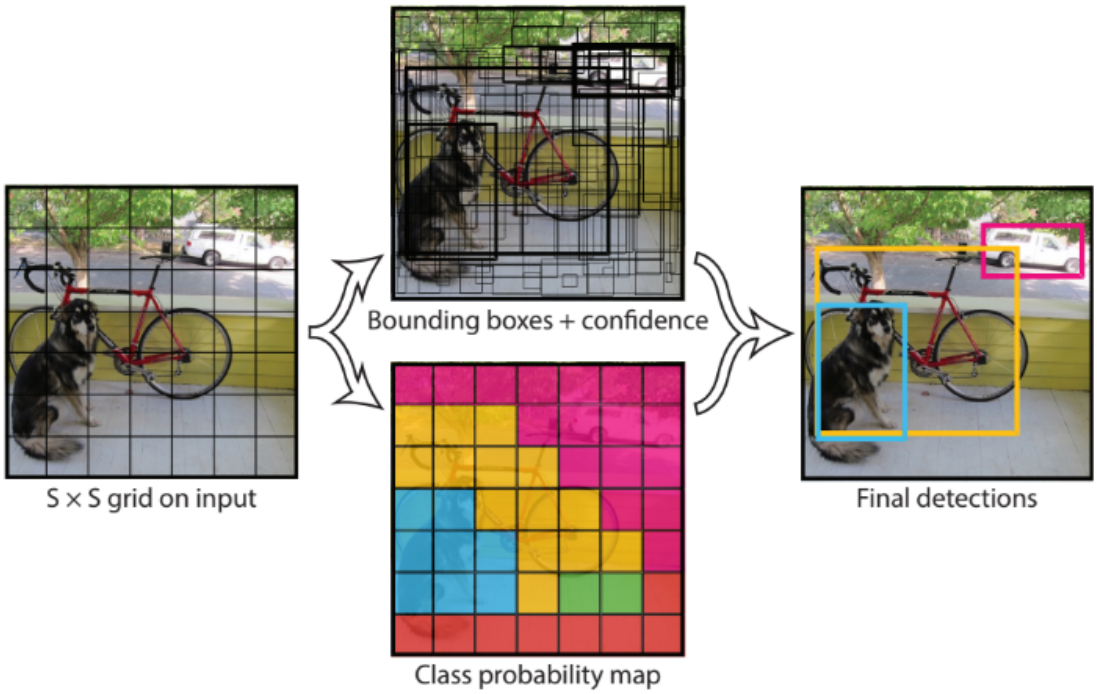
Mô hình YOLO được mô tả lần đầu tiên bởi Joseph Redmon, và các cộng sự. trong bài viết năm 2015 có tiêu đề Bạn chỉ nhìn một lần: Phát hiện đối tượng theo thời gian thực - You Only Look Once: Unified, Real-Time Object Detection (https://arxiv.org/abs/1506.02640). Trong công trình này thì một lần nữa Ross Girshick, người phát triển mạng R-CNN, cũng là một tác giả và người đóng góp khi ông chuyển qua Facebook AI Research (https://research.fb.com/category/facebook-ai-research/).

Phương pháp chính dựa trên một mạng neural network duy nhất được huấn luyện dạng end-to-end model. Mô hình lấy input là một bức ảnh và dự đoán các bounding box và nhãn lớp cho mỗi bounding box. Do không sử dụng region proposal nên kỹ thuật này có độ chính xác thấp hơn (ví dụ: nhiều lỗi định vị vật thể - localization error hơn), mặc dù hoạt động ở tốc độ 45 fps (khung hình / giây) và tối đa 155 fps cho phiên bản tối ưu hóa tốc độ. Tốc độ này còn nhanh hơn cả tốc độ khung hình của máy quay phim thông thường chỉ vào khoảng 24 fps.

Mô hình hoạt động bằng cách trước tiên phân chia hình ảnh đầu vào thành một lưới các ô (grid of cells), trong đó mỗi ô chịu trách nhiệm dự đoán các bounding boxes nếu tâm của nó nằm trong ô. Mỗi grid cell (tức 1 ô bất kì nằm trong lưới ô) dự đoán các bounding boxes được xác định dựa trên tọa độ x, y (thông thường là tọa độ tâm, một số phiên bản là tọa độ góc trên cùng bên trái) và chiều rộng (width) và chiều cao (height) và độ tin cậy (confidence) về khả năng chứa vật thể bên trong. Ngoài ra các dự đoán nhãn cũng được thực hiện trên mỗi một bonding box.

Ví dụ: một hình ảnh có thể được chia thành lưới 7 × 7 và mỗi ô trong lưới có thể dự đoán 2 bounding box, kết quả trả về 98 bounding box được đề xuất. Sau đó, một sơ đồ xác suất nhãn (gọi là class probability map) với các confidence được kết hợp thành một tợp hợp bounding box cuối cùng và các nhãn. Hình ảnh được lấy từ bài báo dưới đây tóm tắt hai kết quả đầu ra của mô hình.

Top




**Hình 5**: Các bước xử lý trong mô hình YOLO (hình ảnh trích xuất từ bài báo gốc). Đầu tiên mô hình chia hình ảnh thành một grid search kích thước $S \times S$. Trên mỗi một grid cell ta dự báo một số lượng $B$ bounding boxes và confidence cho những boxes này và phân phối xác suất của $C$ classes. Như vậy output các dự báo là một tensor kích thước $S \times S \times (B \times 5 + C)$. Giá trị 5 là các tham số của offsets của bounding box gồm $x, y, w, h$ và confidence. $C$ là số lượng tham số của phân phối xác suất.

# 5.2. YOLOv2 (2016) và YOLOv3 (2018)

Mô hình YOLOv2 được Joseph Redmon và Ali Farhadi cập nhật nhằm cải thiện hơn nữa hiệu suất trong bài báo năm 2016 có tựa đề là YOLO9000: Better, Faster, Stronger (https://arxiv.org/abs/1612.08242).

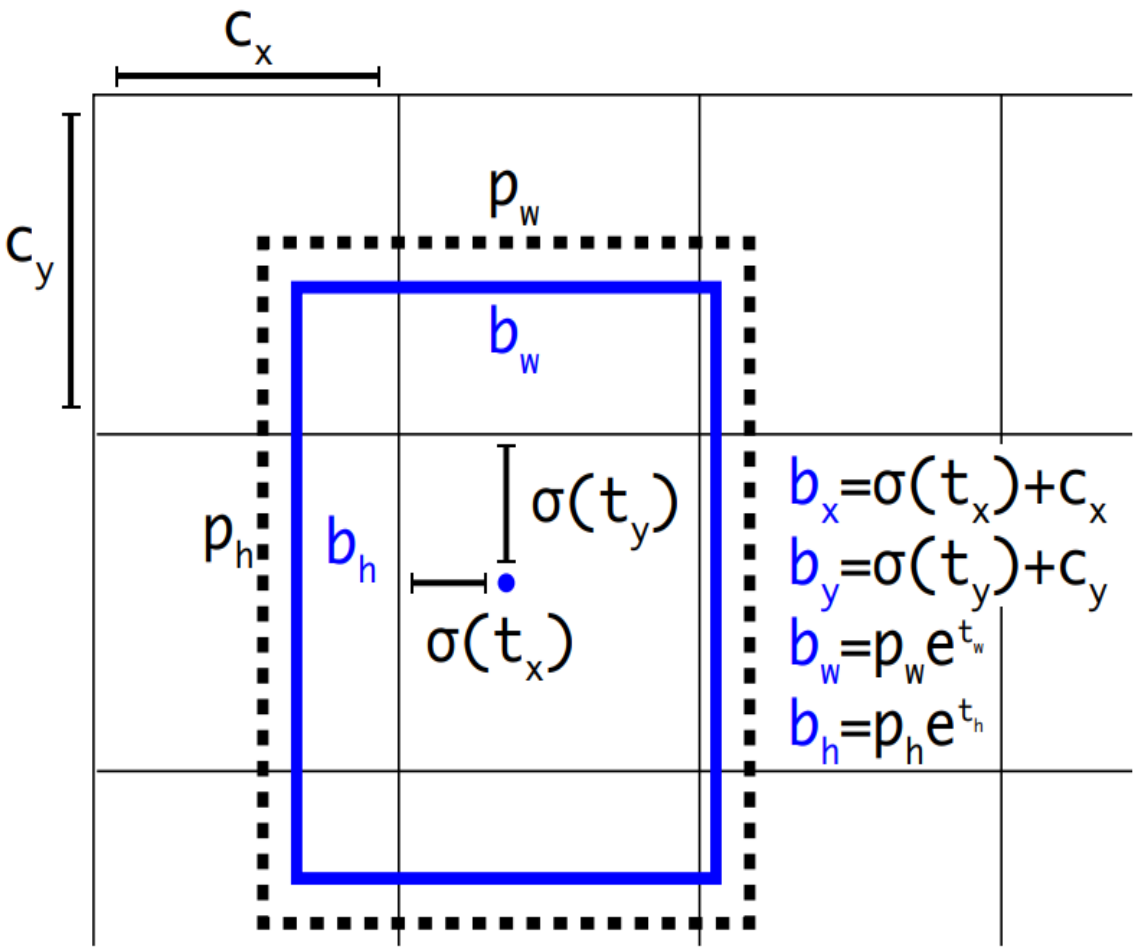
Mặc dù biến thể của YOLO được gọi là YOLOv2, một instance của mô hình theo như mô tả đã được đào tạo trên hai bộ dữ liệu nhận dạng đối tượng, và có khả năng dự đoán lên tới 9000 loại đối tượng khác nhau, do đó được đặt tên là YOLO9000. Với con số này thì mô hình này đã tiến xa hơn rất nhiều so với mọi mô hình trước đó về số lượng các loại đối tượng có khả năng phát hiện.

Một số thay đổi về huấn luyện và kiến trúc đã được thực hiện, chẳng hạn như việc sử dụng batch normalization cho hàng loạt và hình ảnh đầu vào phân giải cao.

Giống như Faster R-CNN, mô hình YOLOv2 sử dụng anchor boxes, bounding box được xác định trước với hình dạng và kích thước hợp lý được tùy chỉnh trong quá trình huấn luyện. Sự lựa chọn các bounding boxes cho hình ảnh được xử lý trước bằng cách sử dụng thuật toán phân cụm k-mean trên tập dữ liệu huấn luyện.

Top

Điều quan trọng, các predicted bounding box được tinh chỉnh để cho phép các thay đổi nhỏ có tác động ít hơn đến các dự đoán, dẫn đến mô hình ổn định hơn. Thay vì dự đoán trực tiếp vị trí và kích thước, các offsets (tức tọa độ tâm, chiều dài và chiều rộng) được dự đoán để di chuyển và định hình lại các pre-defined anchor boxes tại mỗi một grid cell thông qua hàm logistic.




**Hình 6**: Sơ đồ giúp tạo prior bounding box có chiều rộng $p_w$ và chiều cao $p_h$ đã xác định từ grid cell có tọa độ $(c_x, c_y)$. Khi đó tọa độ tâm $(b_x, b_y)$ được tính theo mức độ tịnh tiến hàm sigmoid. Đồng thời, chiều rộng và chiều cao $(b_w, b_h)$ được tính như công thức scale số mũ của cơ số tự nhiên $e$.

Những cải tiến xa hơn của mô hình đã được đề xuất bởi Joseph Redmon và Ali Farhadi trong bài báo năm 2018 với tiêu đề YOLOv3: An Incremental Improvement (https://arxiv.org/abs/1804.02767). Những cải tiến này khá là nhỏ, chủ yếu là thay đổi mô hình deep CNN trong trích xuất feature.

# 6. Tổng kết

Trong bài viết này chúng ta đã tìm hiểu một cách khái quát các khái niệm cơ bản trong computer vision và lịch sử hình thành, phát triển của các lớp mô hình ứng dụng trong object detection. Tôi xin tổng kết lại:

- Phân biệt các khái niệm về image classification, object localization, object detection

Top

- Họ các mô hình object detection dựa trên Region-Based Convolutional Neural Network (R-CNNs) gồm các lớp mô hình: R-CNN, Fast R-CNN và Faster R-CNN là những mô hình sơ khai, có tốc độ xử lý chậm. Thuật toán dựa trên 2 phần xử lý riêng biệt là phát hiện các region proposal và phân loại hình ảnh.
- Lớp các mô hình YOLO có tốc độ thời gian xử lý thực. Là công nghệ state-of-art nhất hiện nay có tốc độ xử lý realtime, phát hiện được lên tới 9000 loại đối tượng.
- Nhìn chung, các kiến trúc object detection đều dựa trên một deep CNN network chẳng hạn như VGG16 hoặc Alexnet ở giai đoạn đầu giúp trích lọc features và nhận diện các region proposal. Sau đó phát triển thuật toán nhằm tìm ra bounding box và confidence của đối tượng chứa trong bounding box. Tùy vào thiết kế mà các mô hình có thể theo dạng pipeline hoặc trong một single model. Tốc độ xử lý của mô hình phụ thuộc vào số lượng bounding box mà nó tạo ra.

# 7. Tài liệu

Tất nhiên các phần trình bày bên trên mới chỉ là tổng hợp khái quát nhất về đặc điểm chính của các lớp mô hình object detection. Để hiểu được nguyên lý hoạt động thực sự bên dưới của các mô hình không phải là dễ dàng. Bên dưới là tổng hợp danh sách các bài báo theo từng họ model, danh sách các tài liệu mà tôi đã tham khảo để viết bài viết này và các khóa học để bạn đọc có thể tìm hiểu sâu hơn.

**R-CNN Family Papers**

1. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013 (https://arxiv.org/abs/1311.2524).
2. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, 2014 (https://arxiv.org/abs/1406.4729).
3. Fast R-CNN, 2015 (https://arxiv.org/abs/1504.08083).
4. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2016 (https://arxiv.org/abs/1506.01497).
5. Mask R-CNN, 2017 (https://arxiv.org/abs/1703.06870).

**YOLO Family Papers**

1. You Only Look Once: Unified, Real-Time Object Detection, 2015 (https://arxiv.org/abs/1506.02640).
2. YOLO9000: Better, Faster, Stronger, 2016 (https://arxiv.org/abs/1612.08242).
3. YOLOv3: An Incremental Improvement, 2018 (https://arxiv.org/abs/1804.02767).

**Code project**

1. R-CNN: Regions with Convolutional Neural Network Features, GitHub (https://github.com/rbgirshick/rcnn).
2. Fast R-CNN, GitHub (https://github.com/rbgirshick/fast-rcnn).
3. Faster R-CNN Python Code, GitHub (https://github.com/rbgirshick/py-faster-rcnn).
4. YOLO, GitHub (https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection).

**Info tác giả**

1. Ross Girshick, Homepage (http://www.rossgirshick.info/).
2. Joseph Redmon, Homepage (https://pjreddie.com/).
3. YOLO: Real-Time Object Detection, Homepage (https://pjreddie.com/darknet/yolo/).

Top

**Các tài liệu tham khảo khác**

1. A Gentle Introduction to Object Recognition With Deep Learning - Jason Brownlee (https://machinelearningmastery.com/object-recognition-with-deep-learning/)
2. Convolutional Neural Network - Course 4 - Andrew Ng (https://www.youtube.com/watch?v=ArPaAX_PhIs&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF)
3. A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN, 2017 (https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4).
4. Object Detection for Dummies Part 3: R-CNN Family, 2017. (https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html)
5. Object Detection Part 4: Fast Detection Models, 2018. (https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html)

Top