Computer Vision — A journey from CNN to Mask R-CNN and YOLO -Part 1

towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04

November 22, 2019

Renu Khandelwal

Jul 22, 2019

·

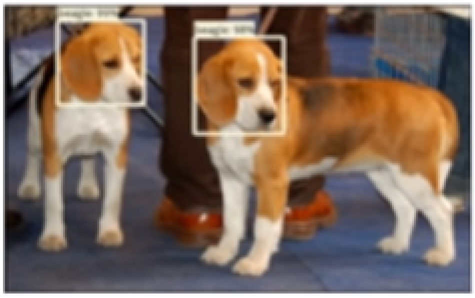
12 min read

·

·

In this article we will explore and understand the architecture and workings of different computer vision algorithm CNN, Region-based CNN(R-CNN), Fast R-CNN, Faster R-CNN. In the next article, we will explore Mask R-CNN and YOLO(You only look once)

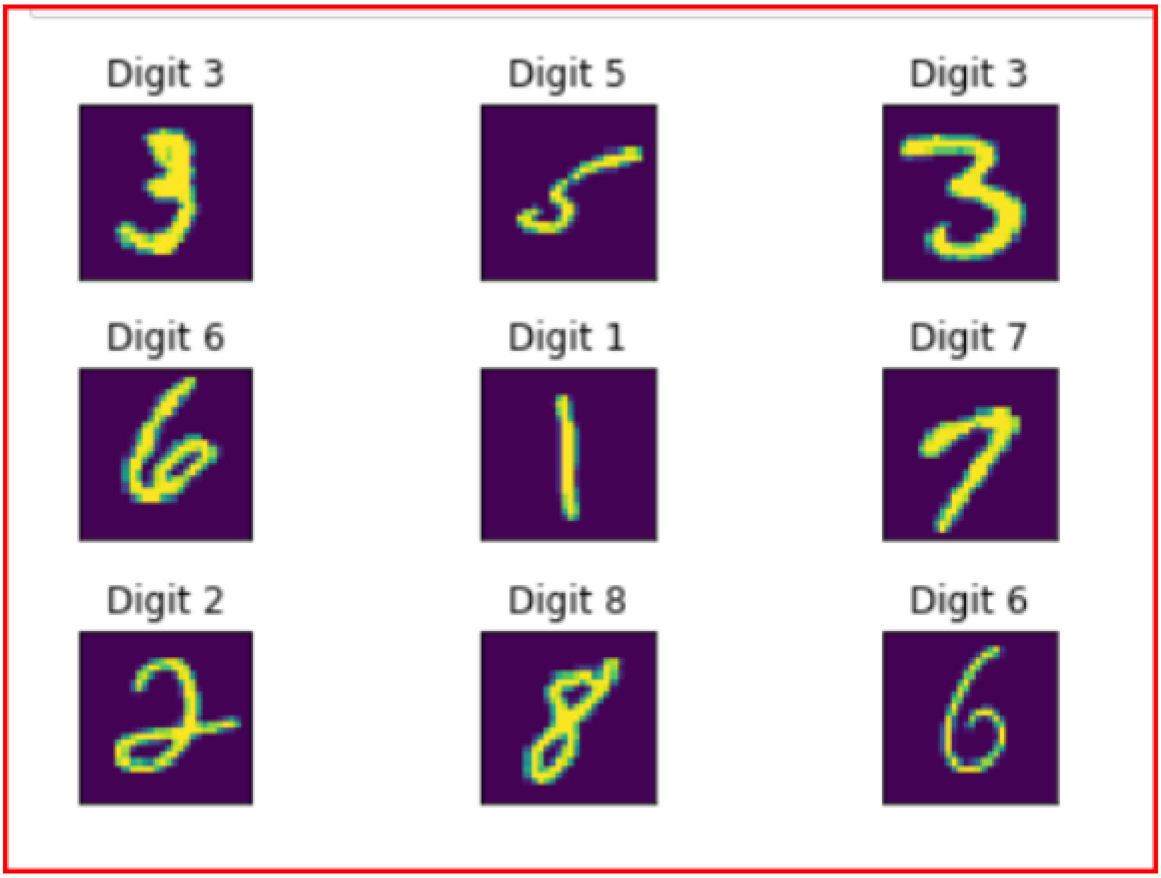
Computer vision is a subfield of AI. It is used to enable computers to understand, identify and generate intelligent understanding of the digital images the same way human vision does.



Using Computer vision we can identify

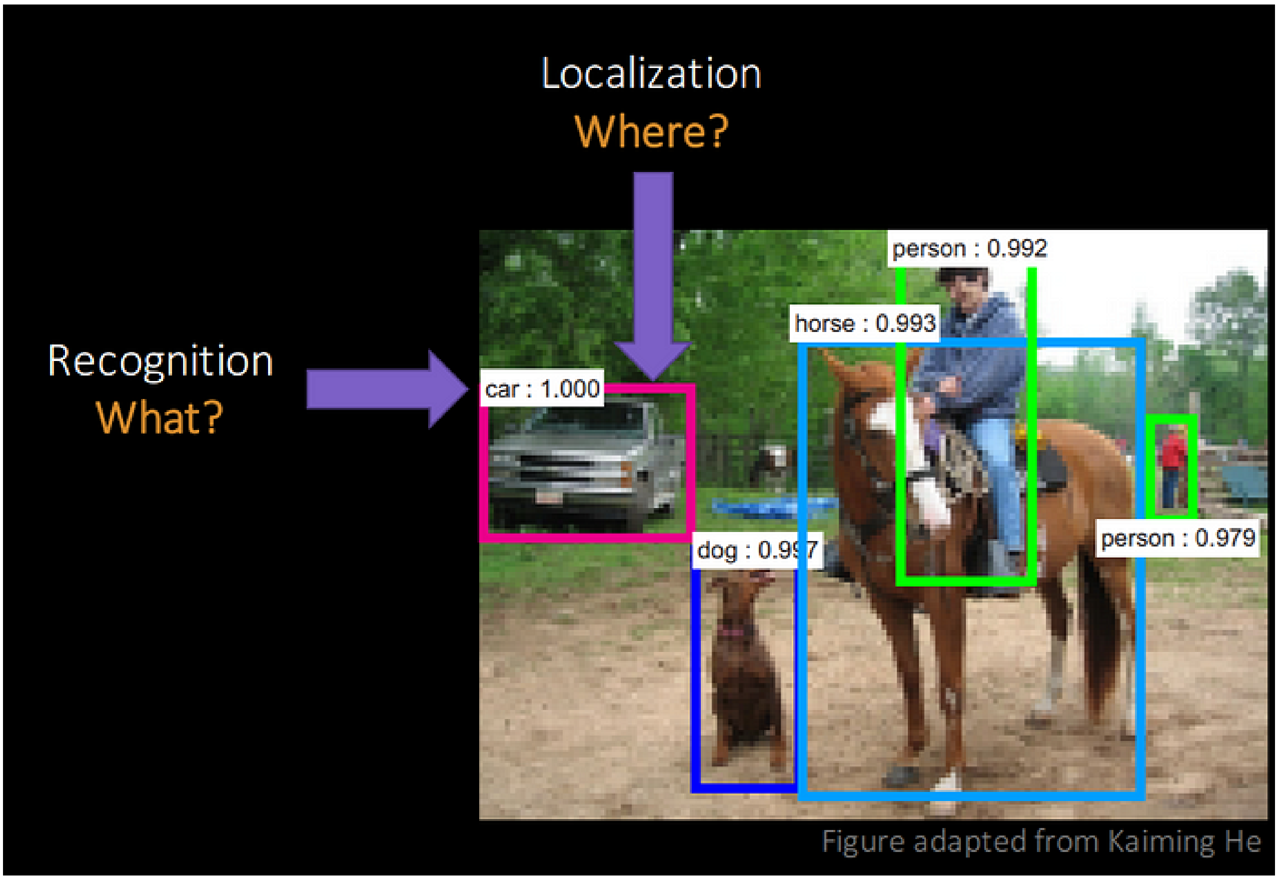
like edge detection.

. Assigning label to an image like identifying cats and dogs in an image or classifying digits



. This involves classifying an image along with identifying the location of the object in a bounding box.

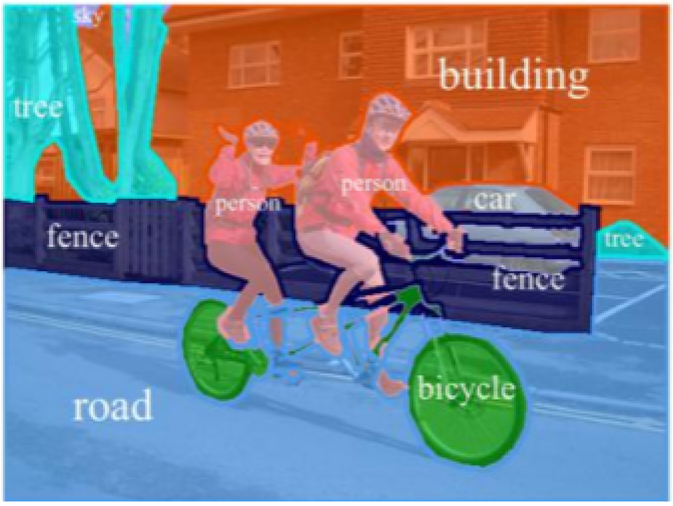
to identify all the different objects present in an image along with their locations. Drawing a bounding box around all the objects present in an image. Detection is recognizing what is present in the image. Localization is where is the object present in the image

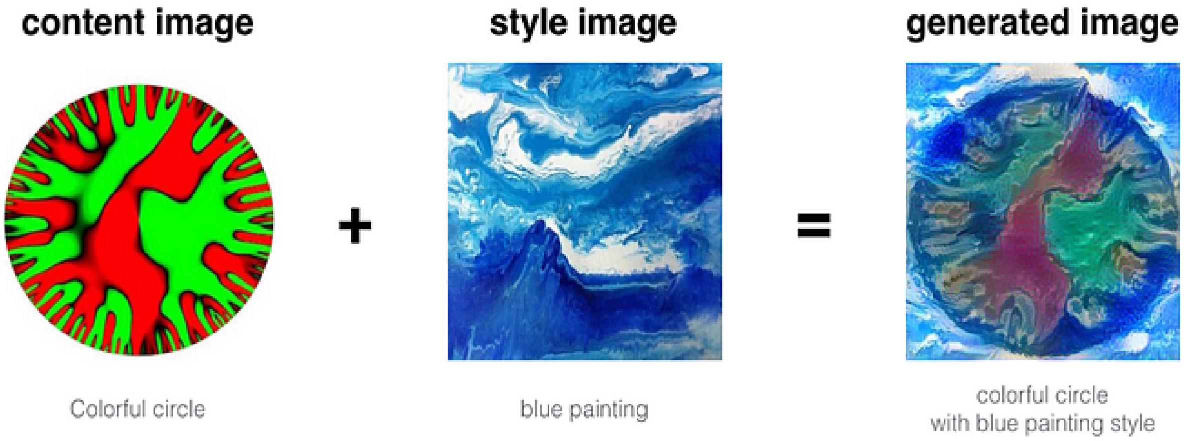


Source:

detects all the objects present in an image at the pixel level. Outputs regions with different classes or objects

where we generate a new image by learning the style from one image and applying to another image





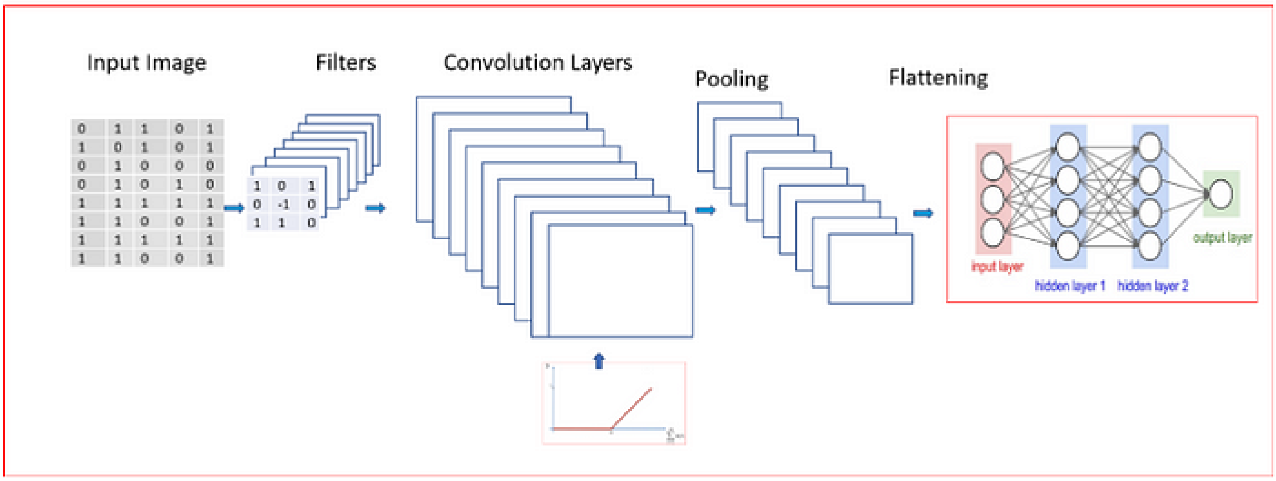
When we view an image, we scan the image. We may view an image from left to right or top to bottom to understand the different features of the image. Our brain combines different local features that we scanned to classify the image. This is exactly how CNN works.

CNN takes input as an image “x”, which is a 2-D array of pixels with different color channels(Red,Green and Blue-RGB).

To the input image we apply different **filters or feature detector** to output **feature maps.** Filters or feature detectors arespatially small compared to the input image. These filters extend through the full depth of the input image.

Multiple convolutions are performed in parallel by applying nonlinear function ReLU to the convolutional layer.

Multiple feature detector identifies different things like edge detection, different shapes, bends or different colors etc.



We apply Pooling to the convolutional layer. We can apply Min Pooling, Max Pooling or Average Pooling. Max pooling function provides better performance compared to min or average pooling.

Pooling helps with **Translational Invariance.** Translational invariance means that when we change the input by a small amount the pooled outputs does not change.

Invariance of image implies that even when an image is rotated, sized differently or viewed in different illumination an object will be recognized as the same object.

In the next step, we flatten the pooled layer to input it to a fully connected(FC) neural network.

We use a **softmax** activation function for multi class classification in the final output layer of the fully connected layer.

For a binary classification we use a **sigmoid** activation function in the final output layer of the fully connected layer.

Strength of CNN

CNN is used for

- Image classification
- Objection detection using bounding boxes

Limitations of CNN

Don't work well when multiple objects are in the visual field due to interference.

We now explore Region-based CNN's that will help solve the problem of multiple objects present in an image and draw bounding boxes around all the different objects.

Region based CNN- R-CNN

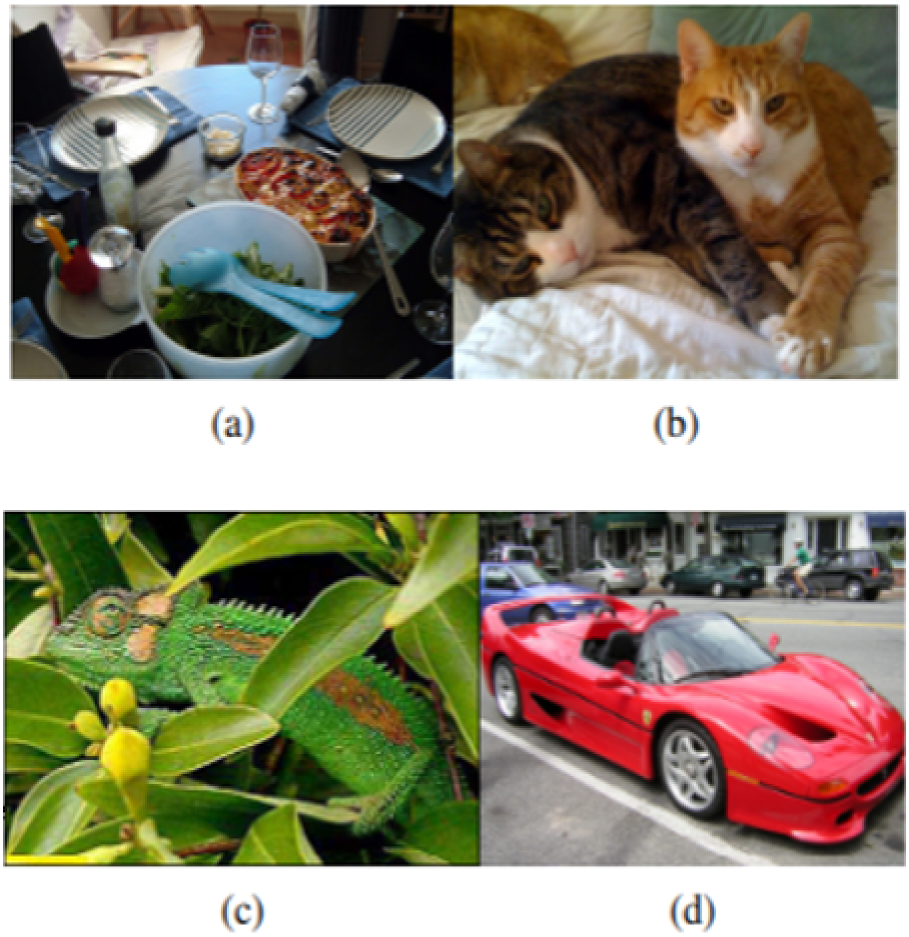
R-CNN is used for classification as well as objection detection with bounding boxes for multiple objects present in an image

R-CNN works on a premise that only a single object of interest will dominate in a given region.

R-CNN uses selective search algorithm for object detection to generate region proposals.

Regions in an image can be identified by

- varying colors
- varying scales
- varying textures
- varying enclosures



Identifying different objection based on regions

Figure(a), spoons, bowls are in different scales. Figure(b), kittens are distinguishable based on colors and not texture. Figure(c), Chameleon is distinguishable by texture, but not color. Figure(d), Wheels are part of the car, but not similar in color or texture. They are part of an enclosure.

Selective search

- Uses bottom-up grouping of image regions to generate a hierarchy of small to large regions
- The goal is to generate a small set of high-quality object locations

- Combines the best of the intuitions of segmentation and exhaustive search.
- Image segmentation exploits the structure of the image to generate object locations
- Exhaustive search aims to capture all possible object locations

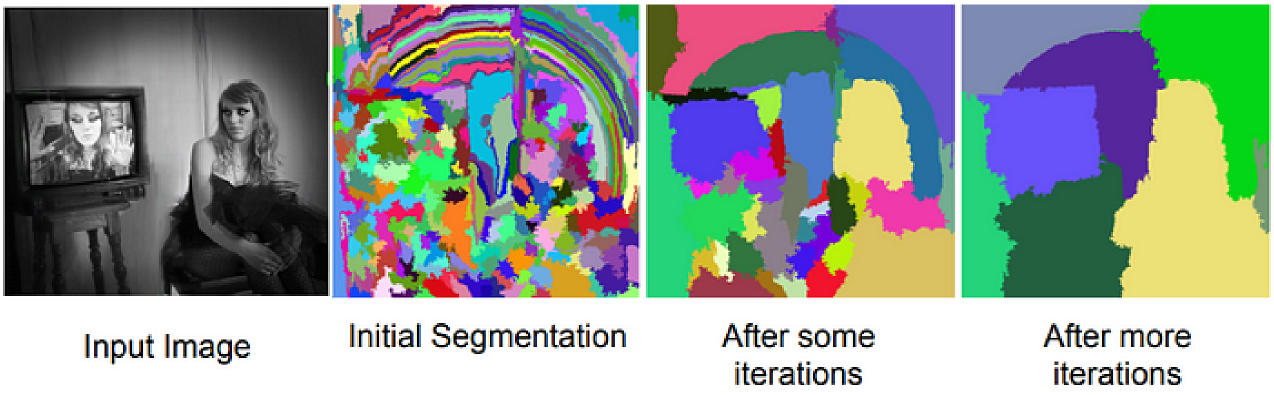
Selective Search with Exhaustive Search Step by Step working

Step 1: **Generate initial sub-segmentation**. We generate as many regions, each of which belongs to at most one object.

Step 2: **Recursively combine similar regions into larger ones.** Here we use Greedy algorithm.

- From the set of regions, choose two regions that are most similar.
- Combine them into a single, larger region.
- Repeat until only one region remains.

This yields a hierarchy of successively larger regions, just like we want



Selective search Algorithm to generate regions for object locations

Step 3: **Use the generated regions to produce candidate object locations**.

Now that we know how Selective Search works, let's get into the details of R-CNN

R-CNN combines region proposal with CNN.

What is Region Proposal?

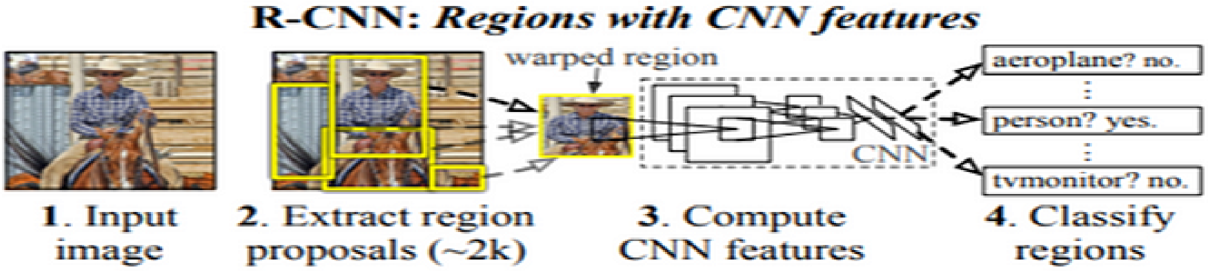
Region proposals is a set of candidate detection available to the detector. CNN runs the sliding windows over the entire image however R-CNN instead select just a few windows. R-CNN uses 2000 regions for an image.

Region proposals run an algorithm called a segmentation algorithm which uses selective search.

1. CNN acts as a feature extractor that extracts a fixed-length feature vector from each region. After passing through the CNN, R-CNN extracts a 4096-dimensional feature vector for each region proposal

To all scored regions in an image, apply a greedy non-maximum suppression.

Non-Max suppression rejects a region if it has an intersection-over union (IoU) overlap with a higher scoring selected region larger than a learned threshold.

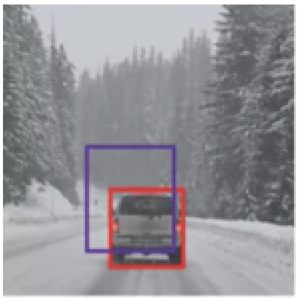


Our objective with object detection is to detect an object just once with one bounding box. However, with object detection, we may find multiple detections for the same objects. **Non-Max suppression ensures detection of an object only once**

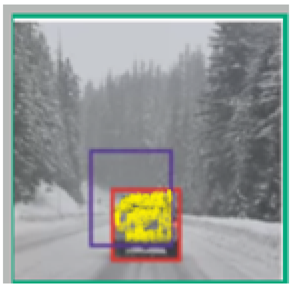
To understand Non-Max suppression, we need to understand IoU.

Intersection over Union — IoU

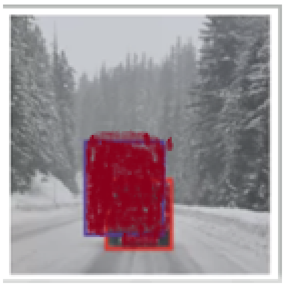
IoU computes intersection over the union of the two bounding boxes, the bounding box for the ground truth and the bounding box for the predicted box by algorithm



Blue is predicted bounding box and red is ground truth bounding box



Intersection



Union

$$\text{IoU}= \frac{\text{Overlapped area between ground truth and predicted bounding boxes}}{\text{Total area of ground truth and predicted bounding boxes}}$$

When IoU is 1 that would imply that predicted and the ground-truth bounding boxes overlap perfectly.

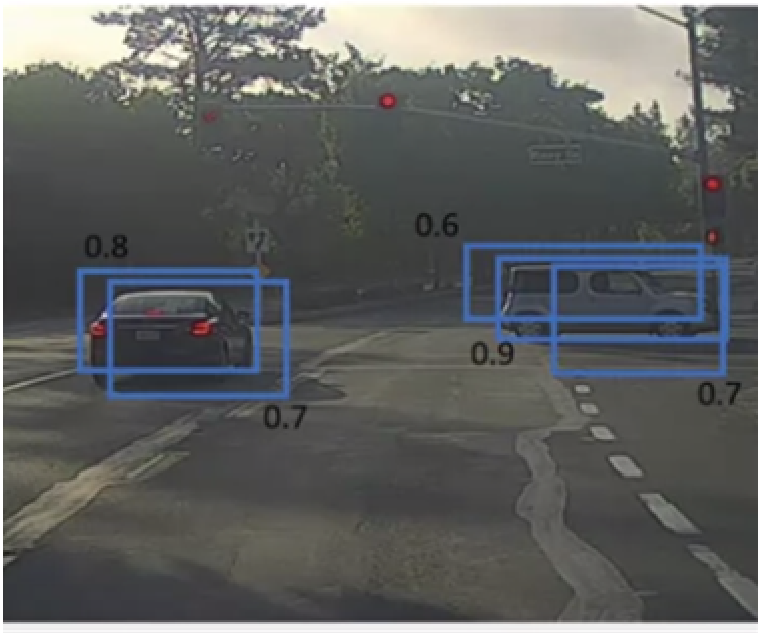
To detect an Object once in an image, **Non-Max suppression considers all bounding boxes with IoU >0.5**

Non-Max Suppression

- Non-Max Suppression will remove all bounding boxes where IoU is less than or equal to 0.5
- Pick the bounding box with the highest value for IoU and suppress the other bounding boxes for identifying the same object

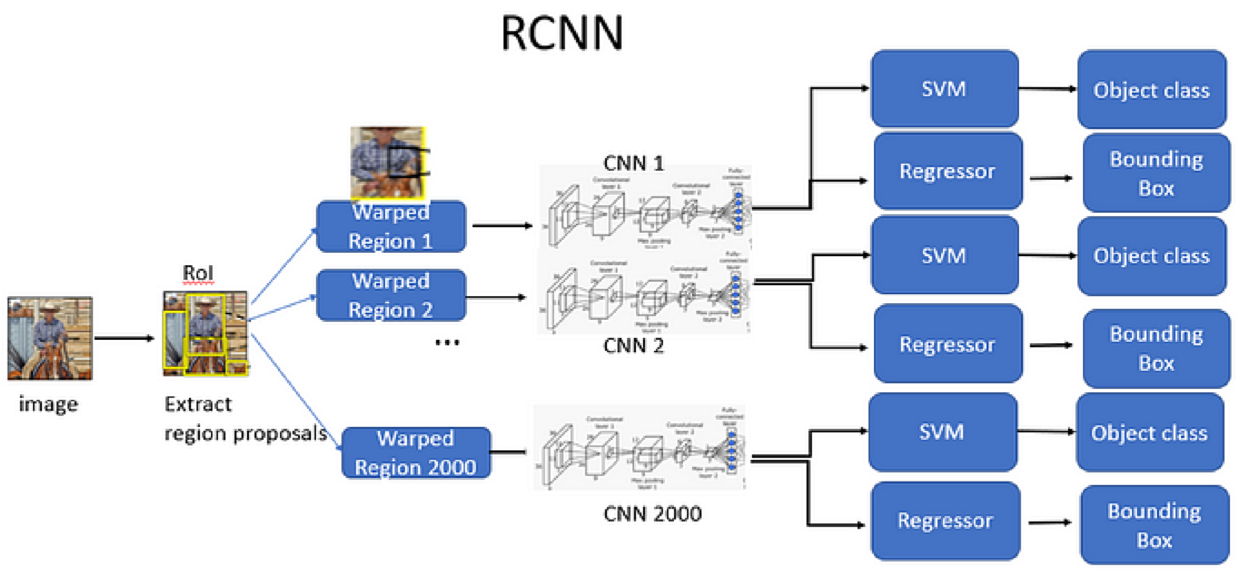
For e.g. if we have three rectangles with the 0.6 and the 0.7 and 0.9. For IoU to identify the vehicle in the image below, Non-Max Suppression will keep the bounding box with IoU 0.9 and will suppress the remaining bounding boxes of 0.6 and 0.7 IoU.

For the car in the image below, Non-Max Suppression will keep IoU with 0.8 and suppress or remove IoU bounding box with 0.7



> Biggest Challenges with R-CNN is that **Training is slow and expensive**

Architecture of R-CNN



- We extract 2000 regions for every image based on selective search.
- Extracting features using CNN for every image region. For N images, we will have N*2000 CNN features.
- R-CNN's Object detection uses three models:

> •CNN for feature extraction
>
> •Linear SVM classifier for identifying objects
>
> •Regression model for tightening the bounding boxes

Few things to be improved in R-CNN would be

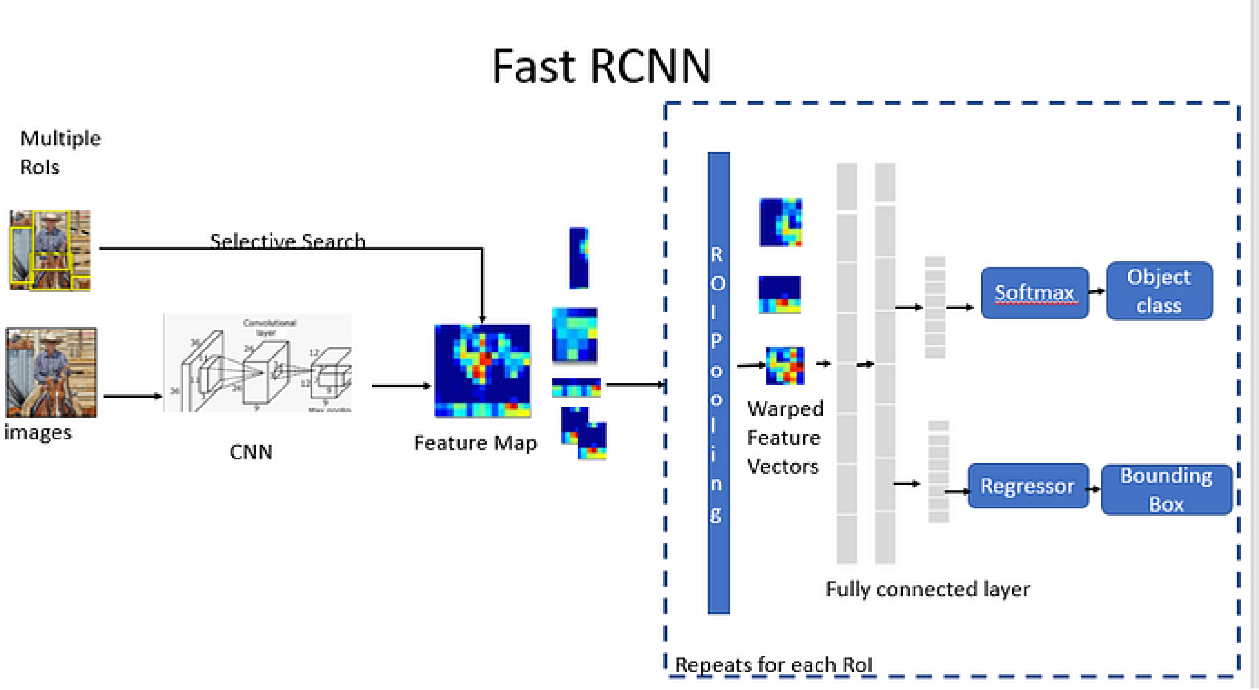
.

All this is done in Fast R-CNN.

Fast R-CNN

Fast R-CNN is a fast framework for object classification and object detection with deep ConvNets

Architecture and working of Fast R-CNN



Fast R-CNN network takes image and a set of object proposals as an input.

Unlike R-CNN, **Fast R-CNN uses a single deep ConvNet to extract features for the entire image once.**

We also create a set of **ROI(Region of Interest)** for the image using selective search. **Region of interest (RoI) layer extracts a fixed-length feature vector from the feature map for each object proposal for object detection**. RoI layer is a special-case of the spatial pyramid pooling layer with only one pyramid level

Fully Connected layers(FC) needs fixed-size input. Hence we use ROI Pooling layer to warp the patches of the feature maps for object detection to a fixed size.

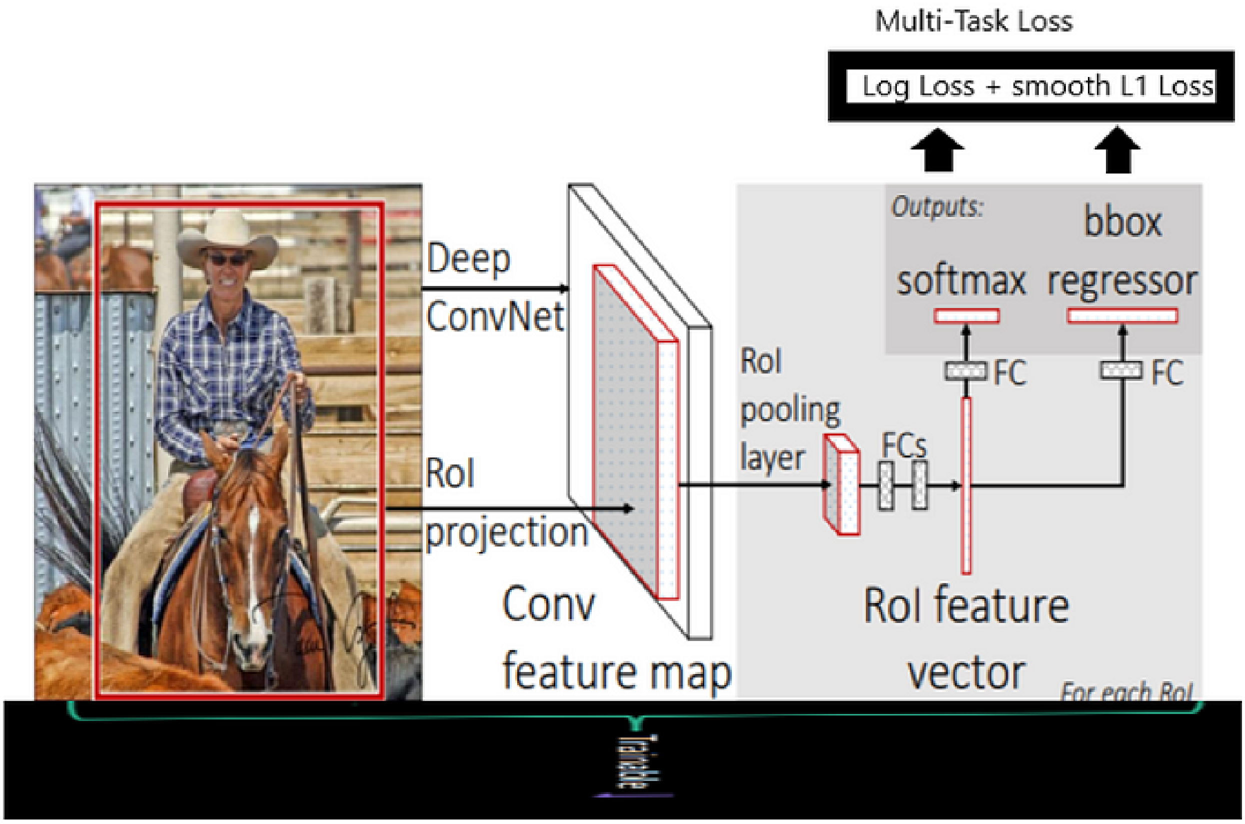
ROI pooling layer is then fed into the FC for **classification** as well as **localization.** RoI pooling layer uses max pooling. It converts features inside any valid region of interest into a small feature map.

Fully connected layer branches into two sibling output layers

- One with
- Another layer with a .

Key differences between R-CNN and Fast R-CNN

- A single deep ConvNet speeds us the image processing significantly unlike R-CNN that uses 2000 ConvNets for each region of the image.
- . Softmax slightly outperforming SVM for objection classification



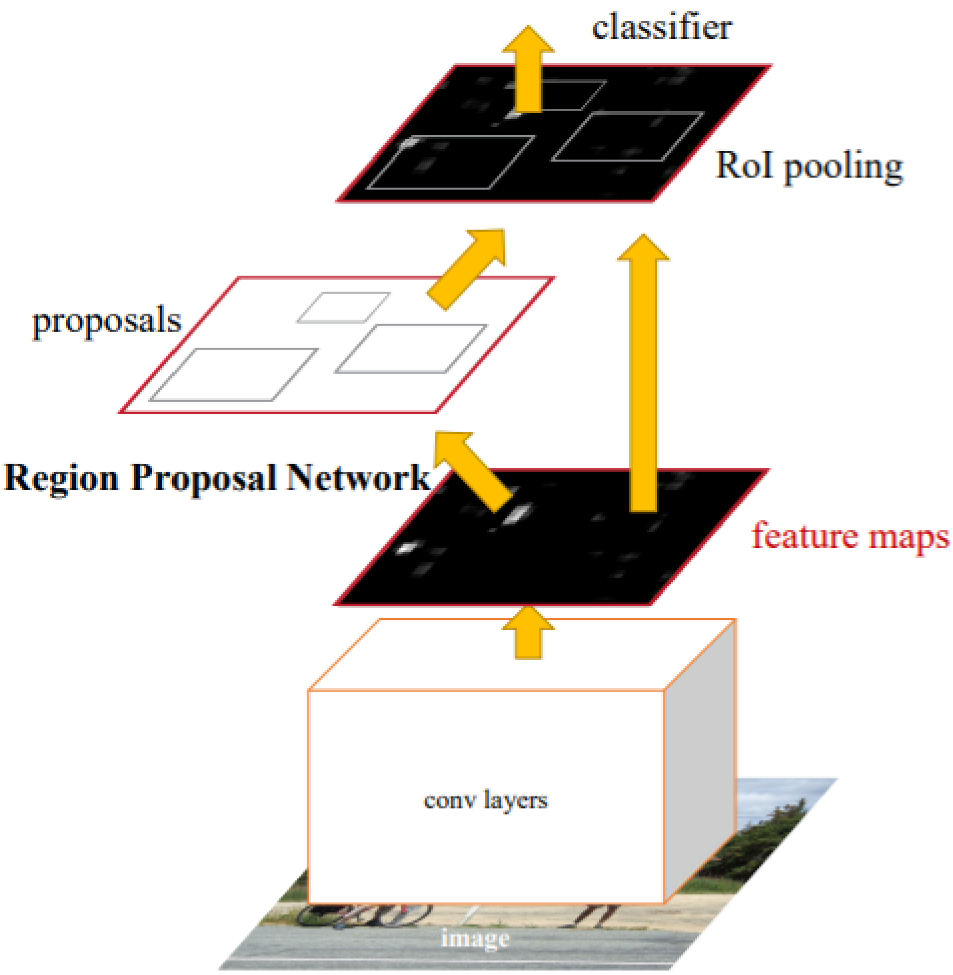
> Fast R-CNN uses selective search as a proposal method to find the Regions of Interest, which is slow and time consuming process. Not suitable for large real-life data sets

Faster R-CNN

Faster R-CNN does not use expensive selective search instead uses Region Proposal Network.

It is a single, unified network for object detection

Faster R-CNN consists of two stages



Faster R-CNN

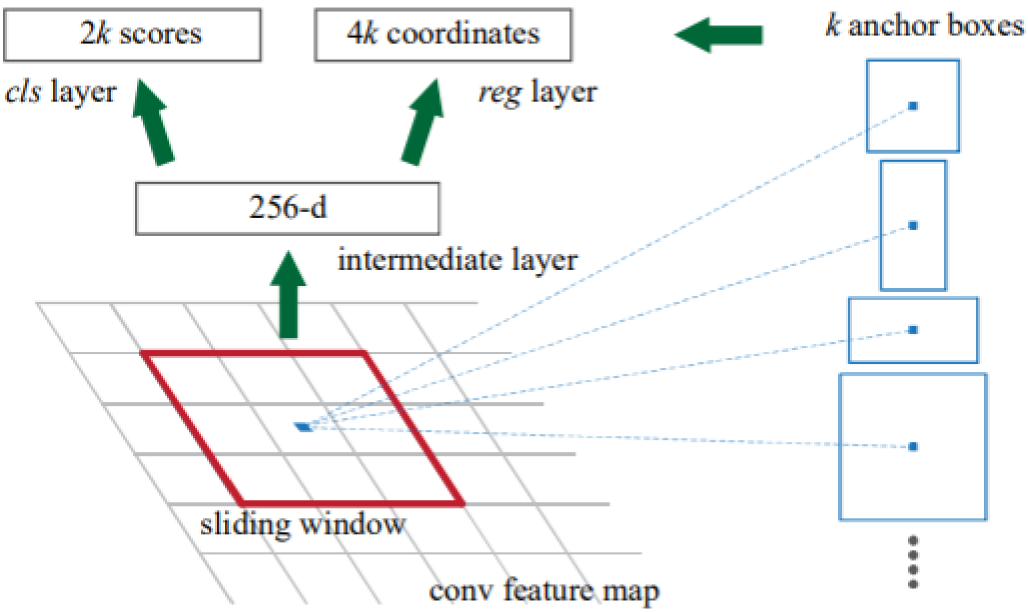
Region Proposal Network(RPN)

Region Proposal Network takes an image of any size as input and outputs a set of rectangular object proposals each with an objectness score. It does this by sliding a small network over the feature map generated by the convolutional layer

RPN shares computation with a Fast R-CNN object detection network.

Feature generated from RPN is fed into two sibling fully connected layers — a box-regression layer for the bounding box and a box-classification layer for object classification.

RPN is efficient and processes 10 ms per image to generate the ROI's.



Region Proposal Network

Anchors

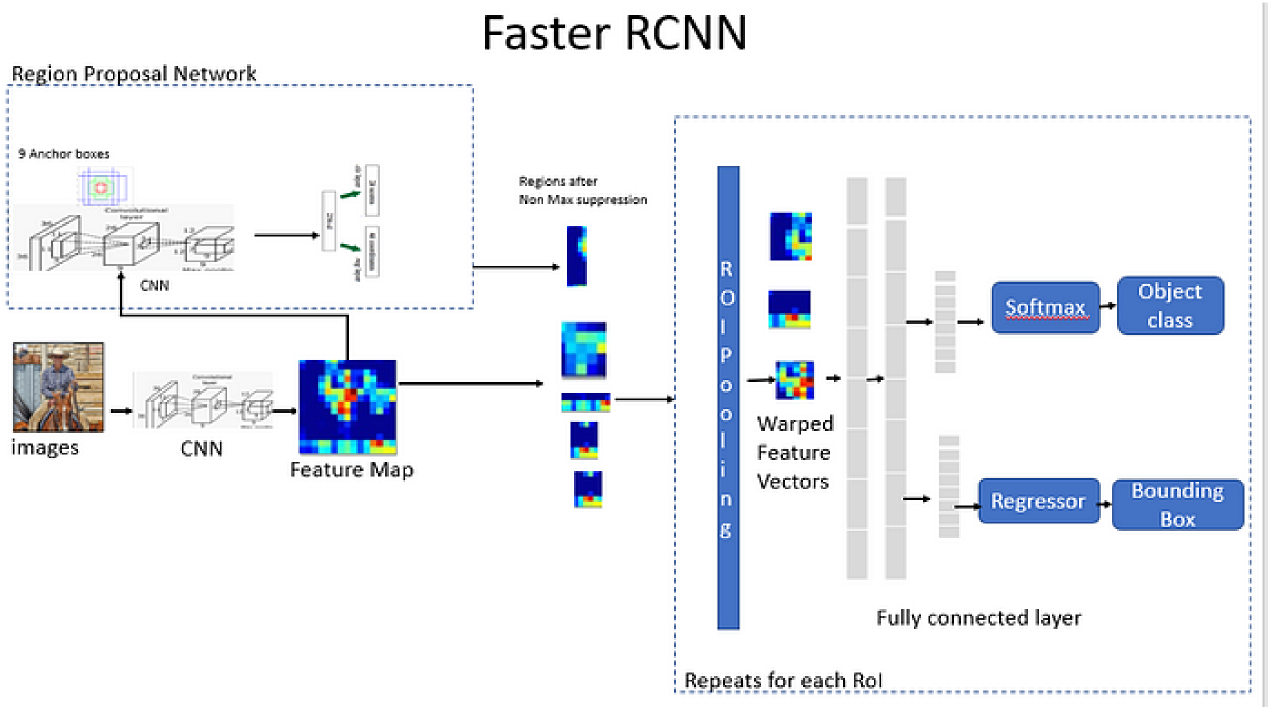
An anchor is centered at the sliding window in question and is associated with a scale and aspect ratio. Faster R-CNN uses 3 scales and 3 aspect ratio, yielding 9 anchors at each sliding windows.

Anchors help with translational invariance.

At each sliding window location, we simultaneously predict multiple region proposals. The number of maximum possible proposals for each location is denoted as k.

Reg layer has 4k outputs encoding the coordinates of k boxes, and the cls layer outputs 2k scores that estimate the probability of object or not object for each proposal

Architecture and working of Fast R-CNN



Faster R-CNN is composed of 3 different neural networks

1. Feature Network which generates feature maps from the input image using deep convolutional layer
2. Region Proposal Network (RPN) is used to identify different regions which uses 9 anchors for each sliding window. This helps with translational invariance. RPN generate a number of bounding boxes called Region of Interests (ROIs) with a high probability for the presence of an object
3. Detection Network is the R-CNN which takes input as the feature maps from the convolutional layer and the RPN network. This generates the bounding boxes and the class of the object

Faster R-CNN takes image as an input and is passed through the Feature network to generate the feature map.

RPN uses the feature map from the Feature network as an input to generate the rectangular boxes of object proposals and the objectness score.

The predicted region proposals from RPN are then reshaped using a RoI pooling layer. Warped into a fixed vector size.

Warped fixed-size vector is then fed into two sibling fully connected layers, a regression layer to predict the offset values for the bounding box and a classification layer for object classification

Usage of Faster R-CNN

- 3-D object detection
- part-based detection
- instance segmentation

- image captioning

Summary

We started with a simple CNN used for image classification and object detection for a single object in the image.

R-CNN is used for image classification as well as localization for multiple objects in an image.

R-CNN was slow and expensive so Fast R-CNN was developed as a fast and more efficient algorithm. Both R-CNN and Fast R-CNN used selective search to come up with regions in an image.

Faster R-CNN used RPN(Region Proposal Network) along with Fast R-CNN for multiple image classification, detection and segmentation.

In the next article, we will explore YOLO and Mask R-CNN.

References:

http://vision.stanford.edu/teaching/cs231b_spring1415/slides/ssearch_schuyler.pdf

https://arxiv.org/pdf/1406.4729.pdf

https://arxiv.org/pdf/1506.01497.pdf

https://arxiv.org/pdf/1311.2524.pdf

http://www.cs.toronto.edu/~tingwuwang/semantic_segmentation.pdf

http://www.cs.toronto.edu/~tingwuwang/semantic_segmentation.pdf

https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf

http://www.robots.ox.ac.uk/~tvg/publications/talks/fast-rcnn-slides.pdf