Aero2Astro

Task 5

Report

By

Anshul Singh

Research intern-Inspect

Object Detection:

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs. Image classification involves predicting the class of one object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around their extent.

Object Detection Method

There are many deep learning methods that are available for deep learning task:

1. RCNN
2. Fast-RCNN
3. Faster RCNN
4. YOLO

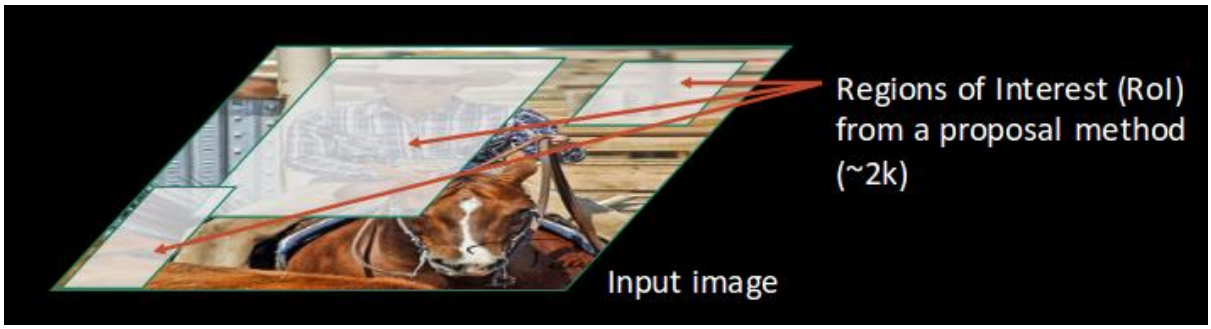
RCNN

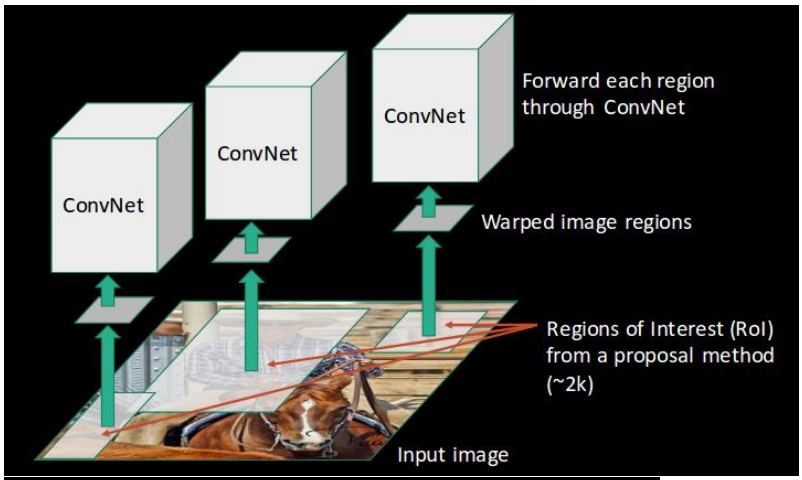
Instead of working on a massive number of regions, the RCNN algorithm proposes a bunch of boxes in the image and checks if any of these boxes contain any object. RCNN uses selective search to extract these boxes from an image (these boxes are called regions).

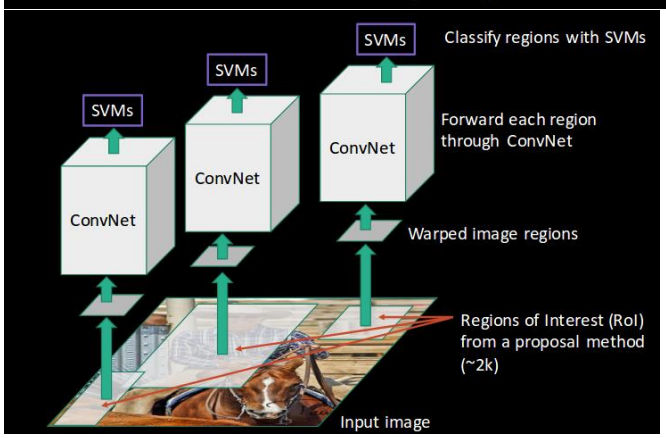
The steps which are used in RCNN to detect the objects are a follows:

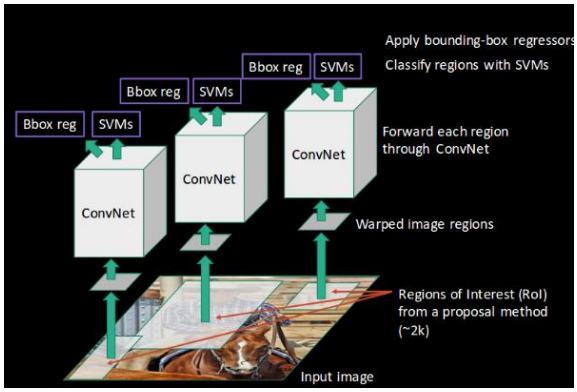
1. We first take a pre-trained convolutional neural network.
2. Then, this model is retrained. We train the last layer of the network based on the number of classes that need to be detected.
3. The third step is to get the Region of Interest for each image. We then reshape all these regions so that they can match the CNN input size.
4. After getting the regions, we train SVM to classify objects and background. For each class, we train one binary SVM.
5. Finally, we train a linear regression model to generate tighter bounding boxes for each identified object in the image.


Input image


Regions of Interest (RoI)
from a proposal method
(~2k)
Input image


ConvNet
ConvNet
ConvNet
Forward each region
through ConvNet
Warped image regions
Regions of Interest (RoI)
from a proposal method
(~2k)
Input image


SVMs
SVMs
SVMs
Classify regions with SVMs
ConvNet
ConvNet
ConvNet
Forward each region
through ConvNet
Warped image regions
Regions of Interest (RoI)
from a proposal method
(~2k)
Input image



Limitation of RCNN:

Training of RCNN model is expensive and slow because:

1. Extracting 2,000 regions for each image based on selective search
2. Extracting features using CNN for every image region. Suppose we have N images, then the number of CNN features will be N*2,000

Some limitation of RCNN is reduced by Fast RCNN.

Fast-RCNN

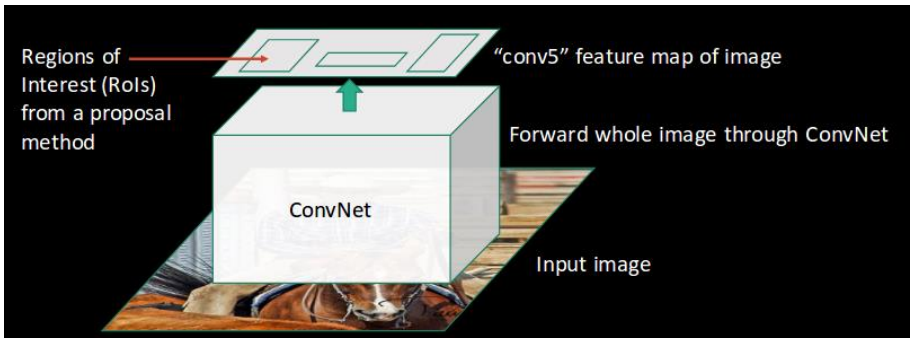
What else can we do to reduce the computation time a RCNN algorithm typically takes? Instead of running a CNN 2,000 times per image, we can run it just once per image and get all the regions of interest. This is the basic idea of fast RCNN.

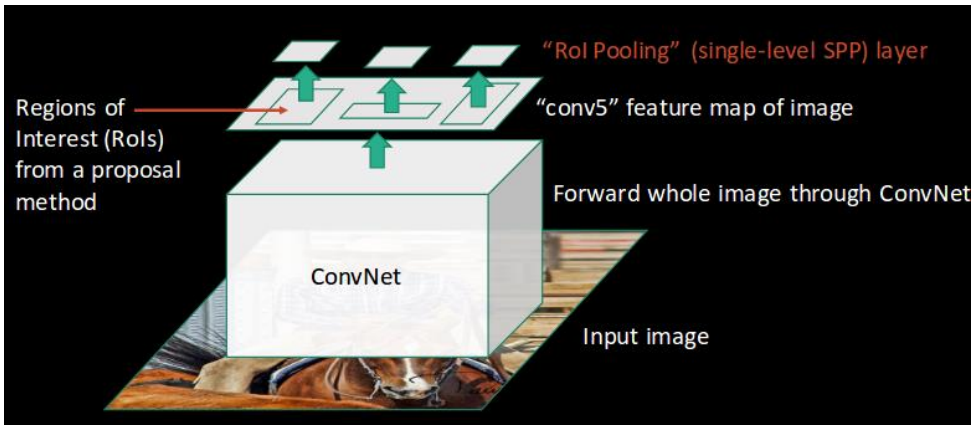
The steps which are used in Fast RCNN to detect the objects are as follows:

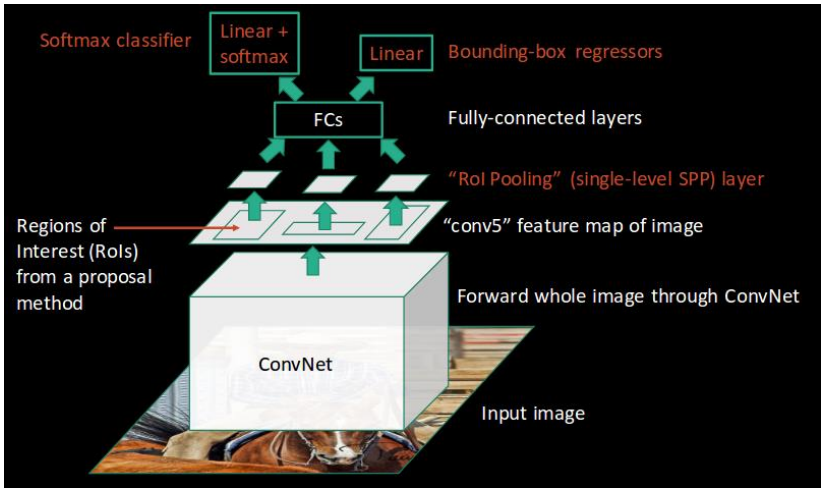
1. As with the earlier two techniques, we take an image as an input.
2. This image is passed to a ConvNet which in turns generates the Regions of Interest.
3. A RoI(Region of intrest) pooling layer is applied on all of these regions to reshape them as per the input of the ConvNet. Then, each region is passed on to a fully connected network.
4. A softmax layer is used on top of the fully connected network to output classes. Along with the softmax layer, a linear regression layer is also used parallely to output bounding box coordinates for predicted classes.

Instead of using three different models ,fast RCNN uses a single model which extracts features from the region and divides them in different classes and returns the boundary boxes.


Input image


Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image


"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image


Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Limitation of Fast CNN

Fast RCNN has some problems too. It also uses a selective search method which is very slow. It takes 2 seconds to detect object which is far better than normal CNN but in the real case datasets it is still less.

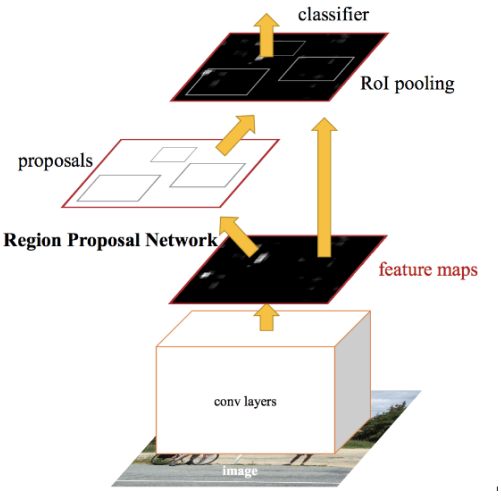
But there is another object detection method which is Faster RCNN.

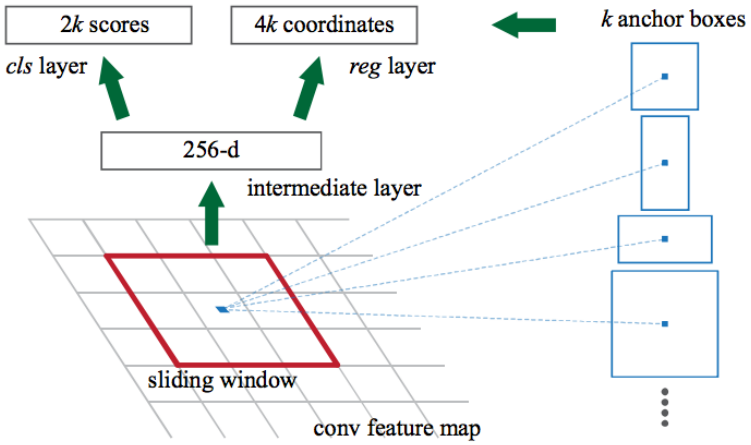
Faster-RCNN:

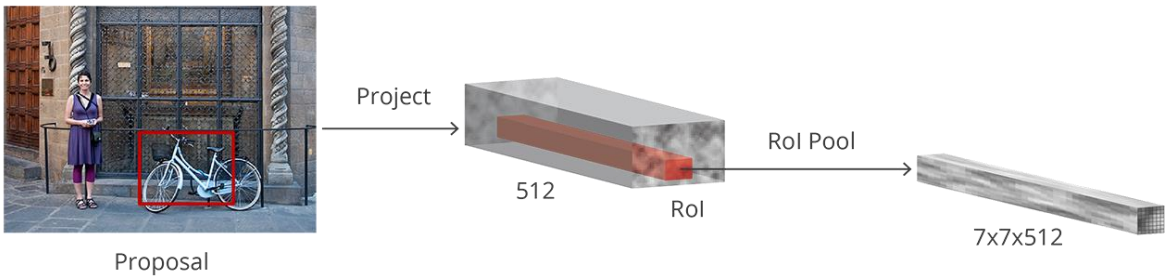
Faster RCNN is the modified version of Fast RCNN. The major difference between them is that Fast RCNN uses selective search for generating Regions of Interest, while Faster RCNN uses “Region Proposal Network (RPN)”.

The steps which are used in Faster RCNN to detect the objects are as follows:

1. We take an image as input and pass it to the ConvNet which returns the feature map for that image.
2. Region proposal network is applied on these feature maps. This returns the object proposals along with their objectless score.
3. A ROI pooling layer is applied on these proposals to bring down all the proposals to the same size.
4. Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

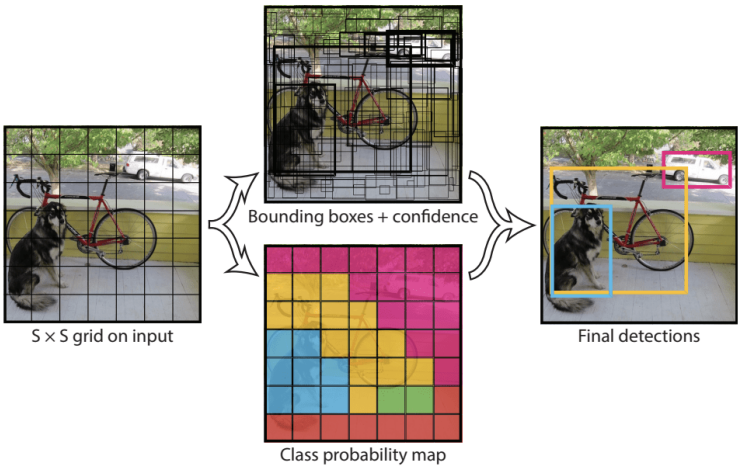






YOLO

The approach involves a single neural network trained end to end that takes a photograph as input and predicts bounding boxes and class labels for each bounding box directly. The technique offers lower predictive accuracy (e.g. more localization errors), although operates at 45 frames per second and up to 155 frames per second for a speed-optimized version of the model.



Data augmentation:

In image processing field, the widely used approaches for data augmentation include reflections, rotation and colorcasting etc. Object detection methods benefit from these data augmentation techniques. Fast and Faster-RCNN use horizontal flip to augment data during train. Randomly erasing patches of the image to reduce the risk of over-fitting. Although the above techniques are available, these methods are primarily empirical and cannot be transfer to other datasets as effectively. So,recent works have focused on learning how to generate good data augmentations. Smart augmentation is an attempt at learned data augmentation strategy. It creates a network that learns how to generate augmented data. AutoAugment uses reinforcement learning to optimize for accuracy. More recently, Population Based Augmentation (PBA) learned non stationary augmentation policy schedules instead of a fixed augmentation policy.

While the above approaches have focused on classification problems, Zoph et al. extend the previous works to object detection tasks . Dvornik et al. Leveraged context modeling to increase the number of object instances. Zhou[30] proposed a slot-based image augmentation to generate images with more learn-able object detection related features. These learned methods are usually expensive in computation.

Another powerful data augmentation method is Generative Adversarial Nets (GAN). The generative model in GAN framework tries to produce realistic images to fool the discriminative model, while the discriminative model attempts to distinguish the generated samples from the real images. The emergence of GAN has attracted the attention of many researchers, and many variants of GAN have been proposed to improve the quality of the synthetic image. The GAN approaches are rarely used for real-world scene detection problems, as it becomes much more difficult to generate an image with many object instances placed in a relevant background than to generate an image with only one object.