Faster RCNN

Towards Real-Time Object Detection with Region Proposal Networks

Charulata Lodha | Feb 23,2021

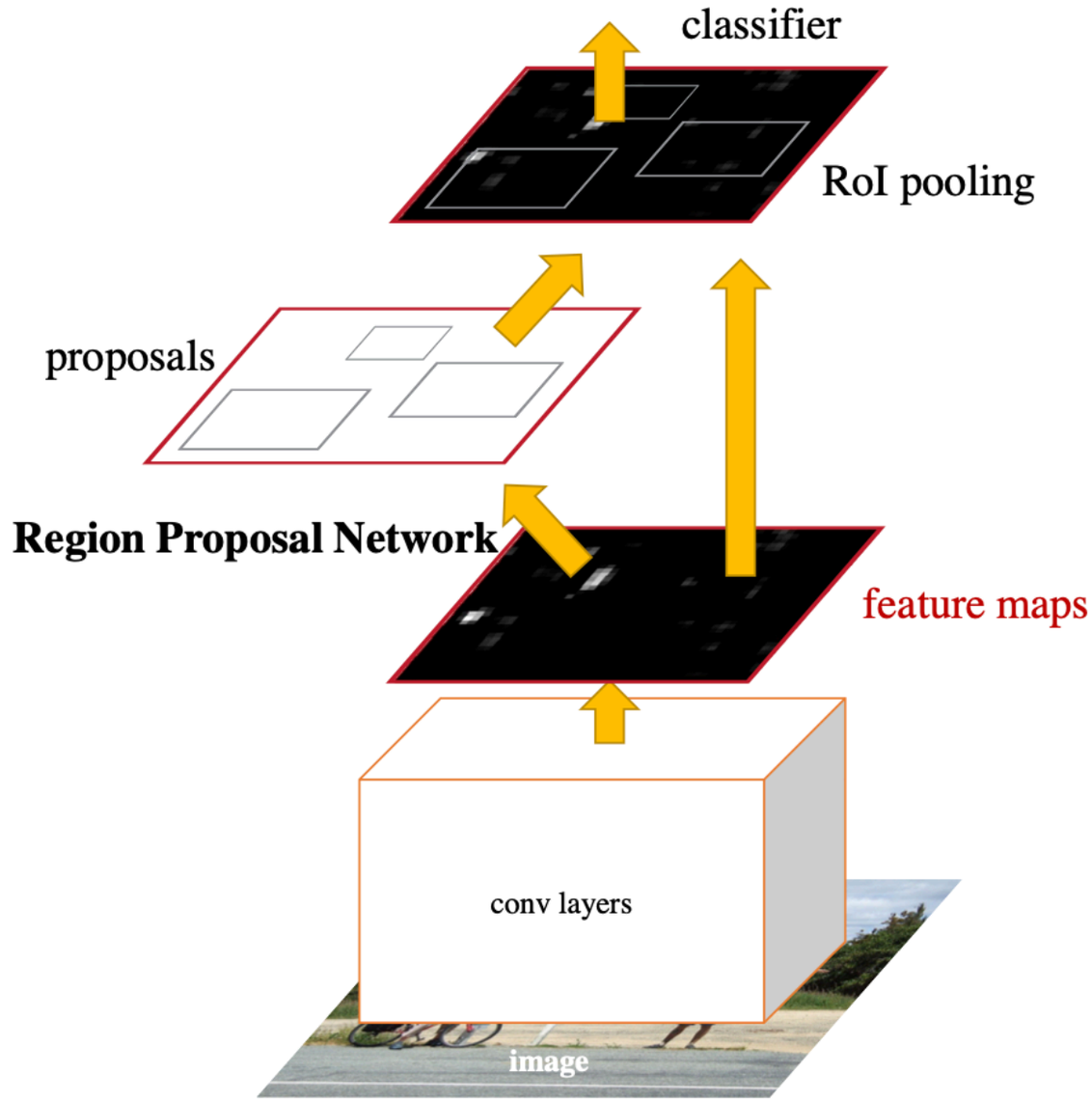
INTRODUCTION

- Region Proposal Networks

  Fully convolutional network (FCN) can be trained end-to- end specifically for the task for generating detection proposals.

- Input : takes an image (of any size) as input

- Output : a set of rectangular object proposals

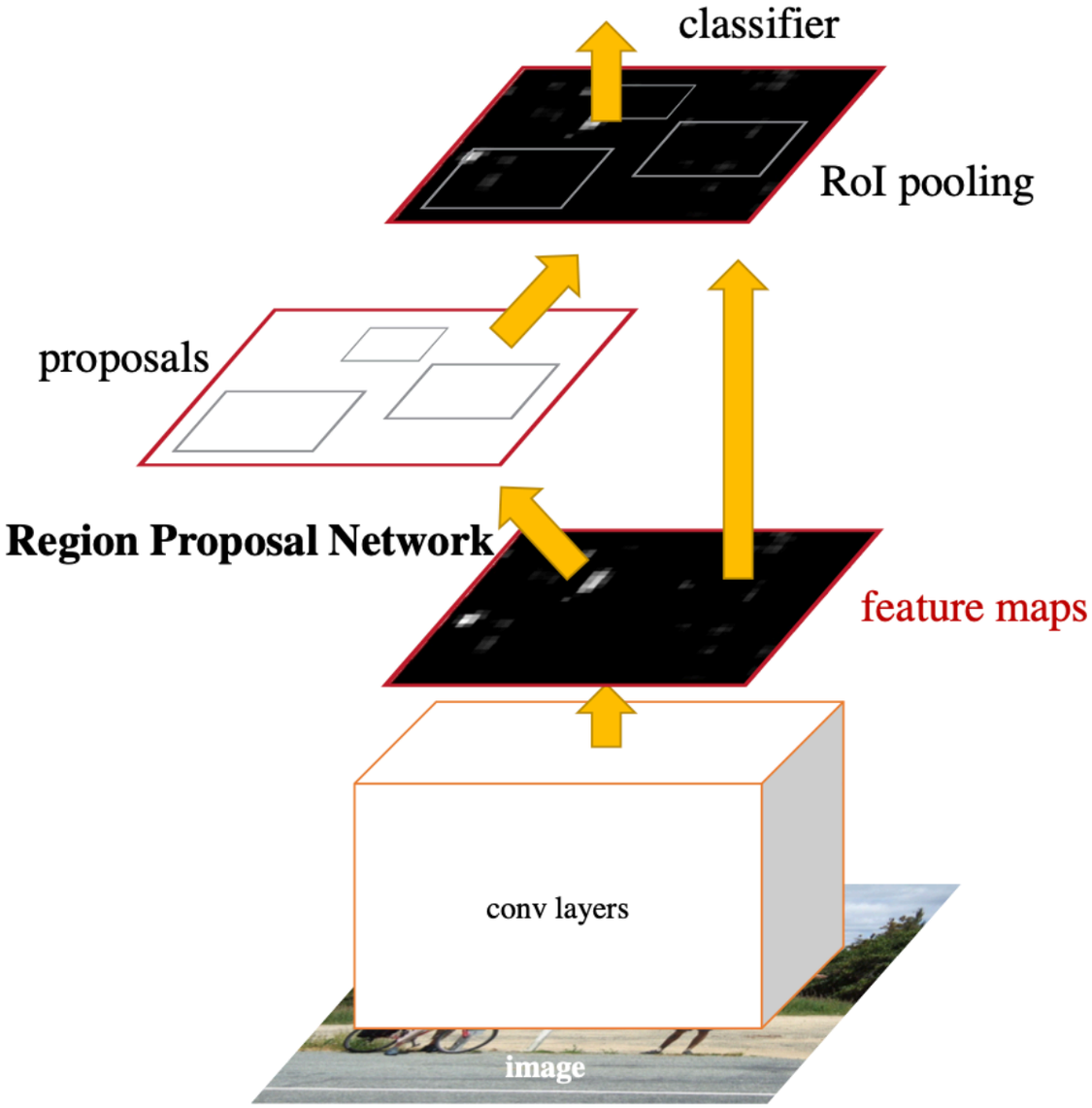


Faster RCNN

2 Modules

- Deep fully convolutional network : to propose region
- Fast R-CNN detector : uses the proposed regions

-> RPN module tells the Fast R-CNN module where to look.

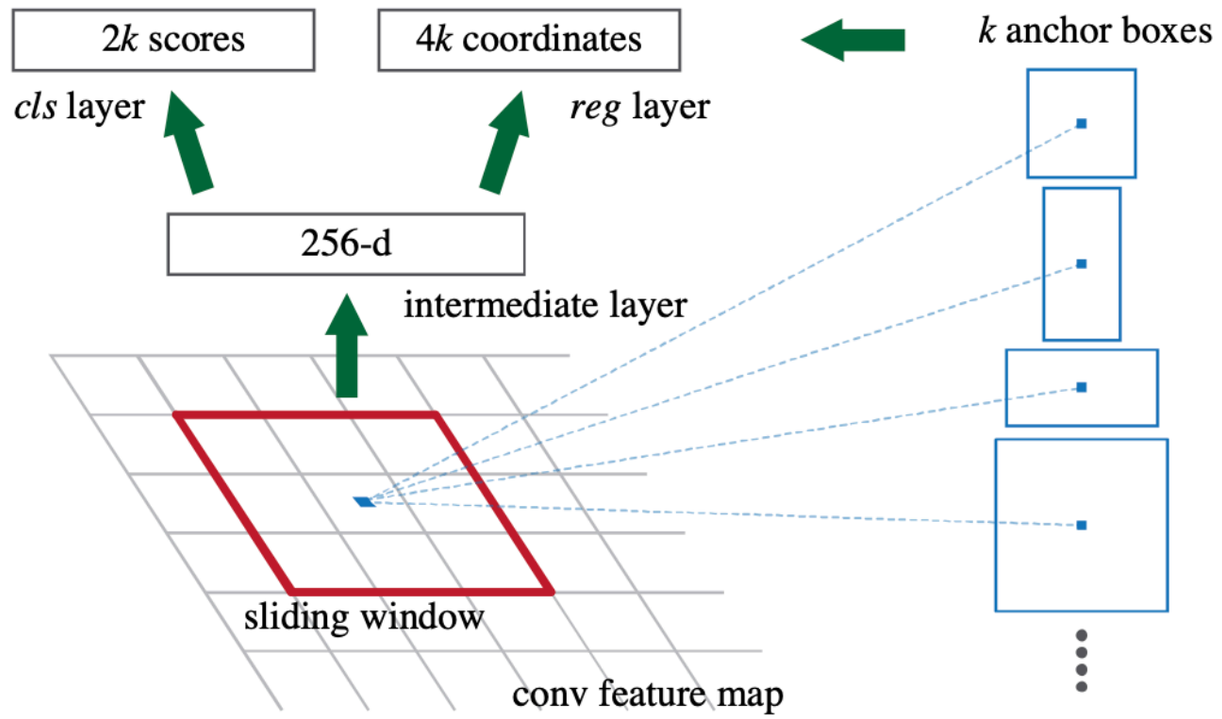


Designs and Properties

Region Proposal Network

- Region Proposal Generation : slide a small network over the feature map which was the output from previous shared layer.

- This feature is fed into two sibling fully-connected layers :
    - a box-regression layer (reg) and
    - a box-classification layer (cls).



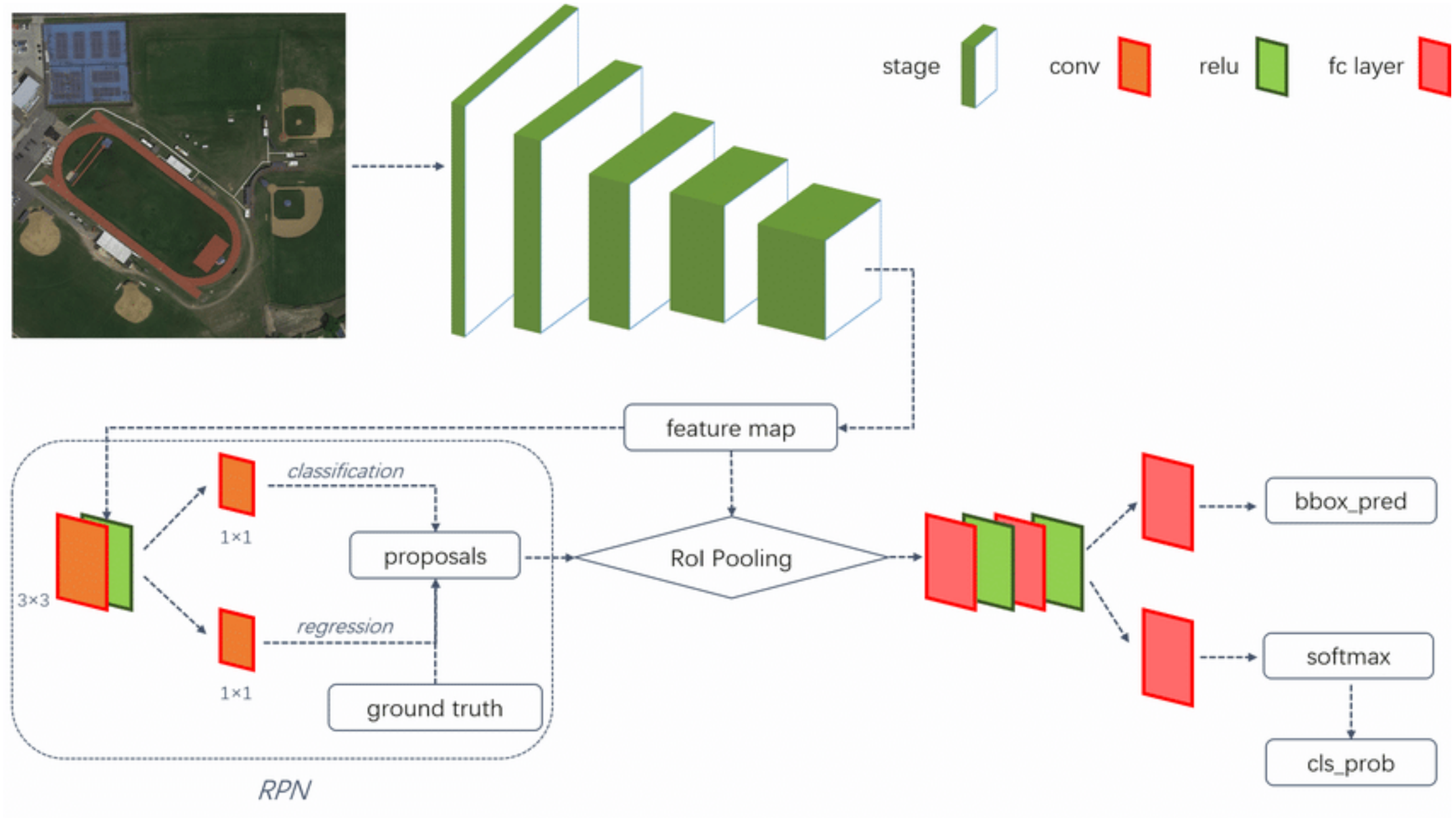
—> This architecture is naturally implemented with an n×n convolutional layer followed by two sibling 1 × 1 convolutional layers (for reg and cls, respectively).

Anchors

- Reference Box is known as Anchors (scale & aspect ratio).
    - Default 3 scale, 3 aspect ratio -> k=9 anchors at each sliding position.

- k -> maximum possible proposals for each location of sliding window
- Reg Layer = 4k encoding coordinates : k boxes
- Clas Layer = 2k score : Probability Object or not?
    - 2 class Softmax Layer
    - Logistic Regression

eg : Feature Map of size W * H. -> W*H*k anchors.



The architecture of Faster R-CNN. The "conv" represents convolutional layer, the "relu" represents activation function and the "fc layer" represents fully connected layer. The network outputs intermediate layers of the same size in the same "stage". The "bbox_pred" represents the position offset of the object and the "cls_prob" represents the probability of the category.

Translation-Invariant Anchors

Important property of this approach

- It is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors.

  If one translates an object in an image, the proposal should translate and the same function should be able to predict the proposal in either location.

- Result :
  Output layer has 2.8×104 parameters ($512 \times (4 + 2) \times 9$ for VGG-16), two orders of magnitude fewer than MultiBox's output layer that has 6.1×106 parameters ($1536 \times (4 + 1) \times 800$ for GoogleNet.
  Less risk of overfitting on small datasets.

Multi-Scale Anchors as Regression References

Two popular ways for multi-scale predictions

1. image/feature pyramids : images resized at multiple scale and feature maps are computed for each scale.
2. sliding windows of multiple scales (and/or aspect ratios) on feature map.

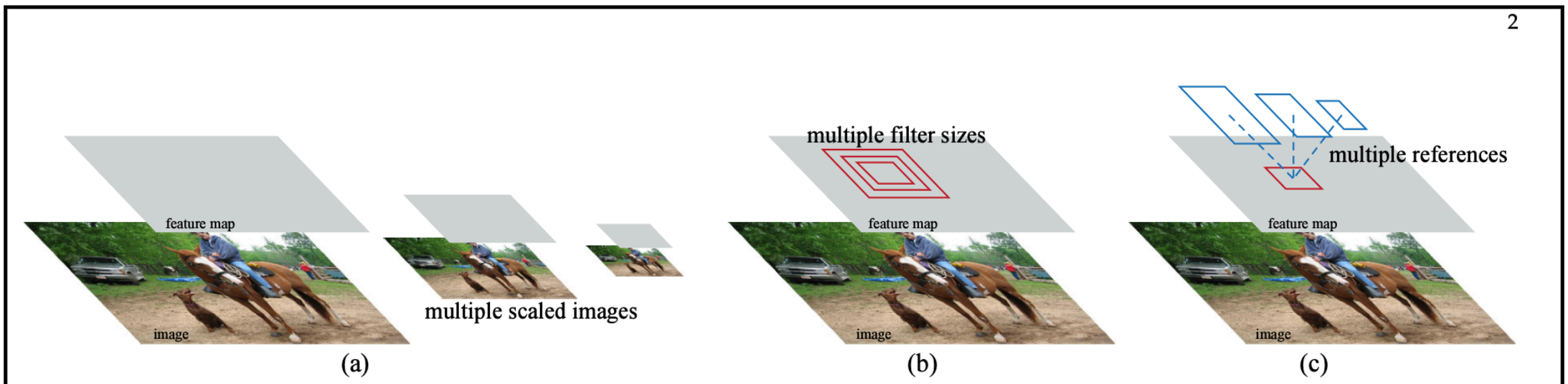
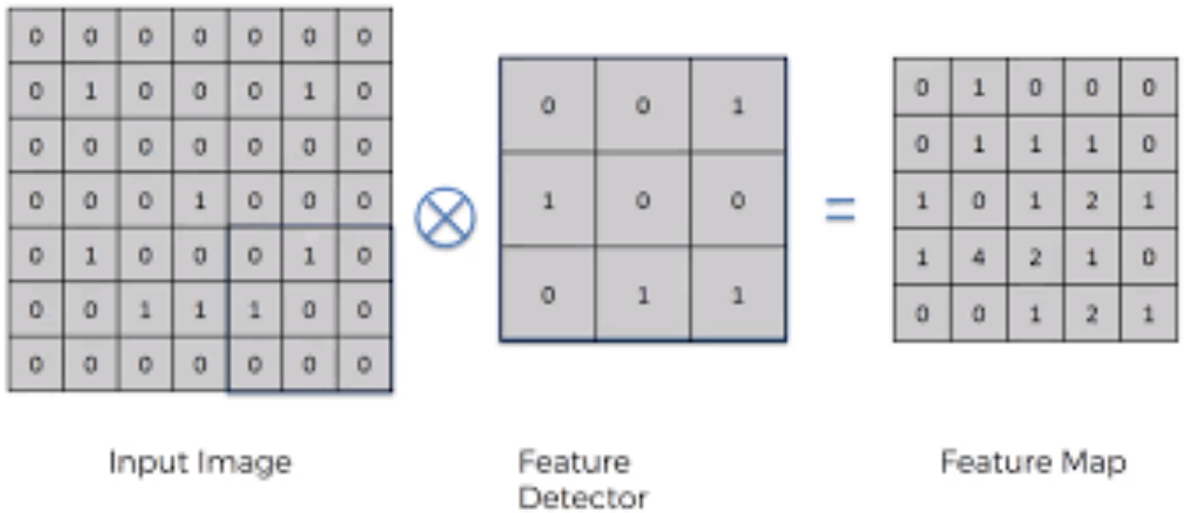
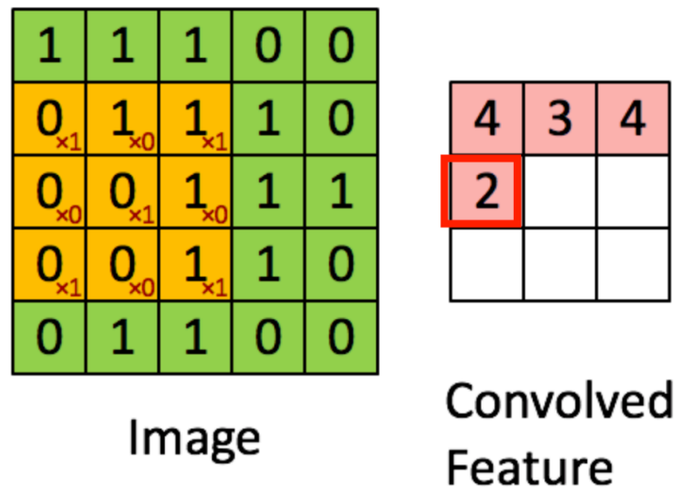
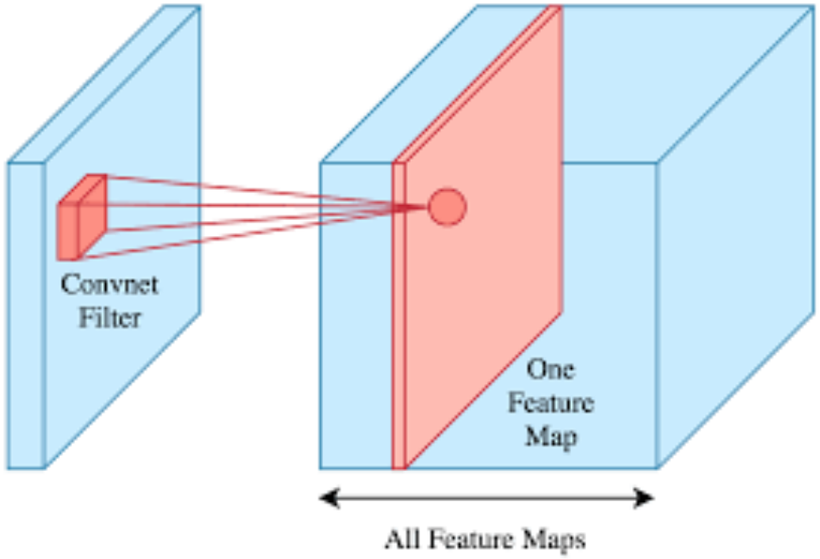


Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

Feature Map aka Activation Map & Filters







In CNN terminology, the **3×3 matrix** is called a '**filter**' or 'kernel' or '**feature detector**' and

the **matrix formed by sliding the filter over the image** and computing the dot product is called the 'Convolved Feature' or 'Activation Map' or the '**Feature Map**'.

The **feature map** is the output of one **filter** applied to the previous layer.
It's a mapping of where a certain kind of feature is found in the image. A high activation means a certain feature was found.

Anchor-based method

- More cost-efficient
- This method classifies and regresses bounding boxes with **reference to anchor boxes** of multiple scales and aspect ratios
- It only relies on images and feature maps of a **single scale**, and uses filters (sliding windows on the feature map) of a single size.

Loss Function

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*).$$

- ‘i‘is the index of an anchor in a mini-batch
- ‘pi’ is the predicted probability of anchor i being an object.
- The ground-truth label p∗i is 1 if the anchor is positive, and is 0 if the anchor is negative.
- The classification loss is log loss over two classes (object vs. not object
- regression loss is activated only for positive anchors (p∗i = 1) and is disabled otherwise (p∗i = 0).
- λ = 10 : balancing parameter

For bounding box regression, we adopt the parameterizations of the 4 coordinates following [5]:

$$\begin{aligned} t_{\mathrm{x}} &= (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}} &= \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}), \\ t_{\mathrm{x}}^* &= (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}}^* = (y^* - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}}^* &= \log(w^*/w_{\mathrm{a}}), \quad t_{\mathrm{h}}^* = \log(h^*/h_{\mathrm{a}}), \end{aligned} \tag{2}$$

- We assign a positive label to two kinds of anchors:
- (i) the anchor/anchors with the highest Intersection-over- Union (IoU) overlap with a ground-truth box, or
- (ii) an anchor that has an IoU overlap higher than 0.7 with any ground-truth box.

Training RPN

- Back- propagation
- Stochastic gradient descent (SGD)

References

- https://arxiv.org/pdf/1506.01497.pdf
- https://computersciencewiki.org/index.php/Feature_maps_(Activation_maps)
- https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
- https://www.researchgate.net/publication/332079570_Geospatial_Object_Detection_on_High_Resolution_Remote_Sensing_Imagery_Based_Double_Multi-Scale_Feature_Pyramid_Network/figures?lo=1