Regional Convolutional Neural Network

crackhopper

2016-09-24 14:27:40

Outline

Regional Convolutional Neural Network

Intro

paper

- Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)
- Fast R-CNN
- Faster R-CNN- Towards Real-Time Object Detection with Region Proposal Networks

Focus

- localizing objects with a deep network
- training a high-capacity model with only a small quantity of annotated detection data.

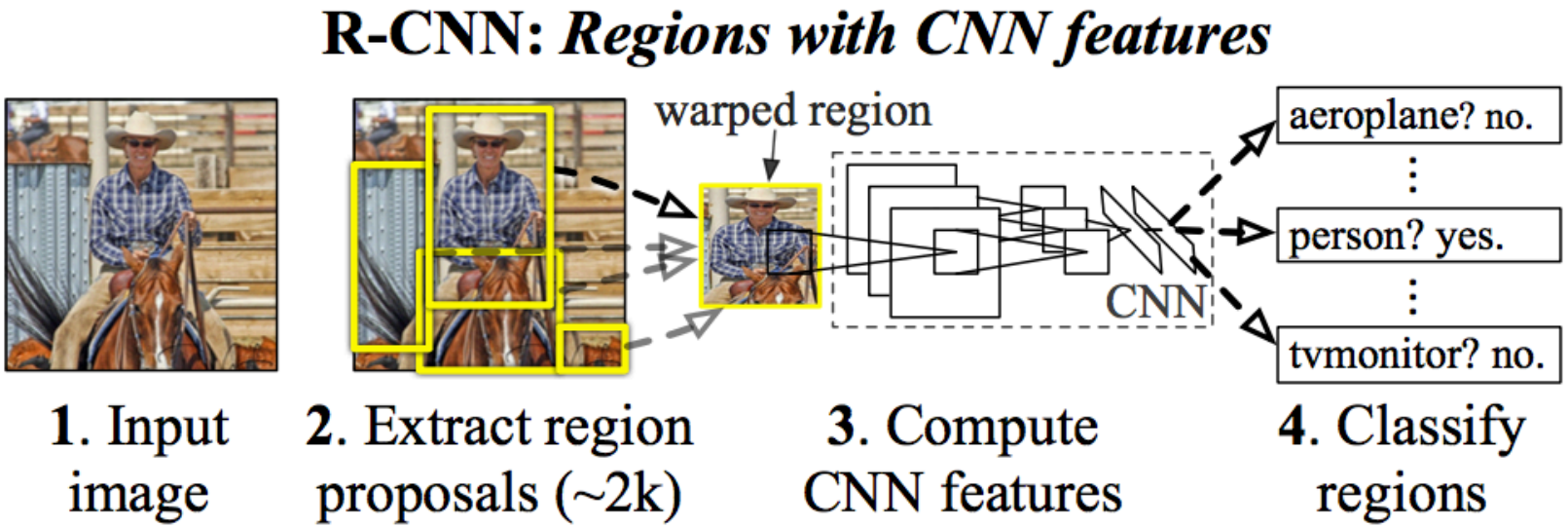
Common Methods

- frames localization as a regression problem
  - input: image
  - output: window, confidence
- build a sliding-window detector
  - input: image, window
  - output: confidence

R-CNN

- Recognition using regions
- Transfer from network training on ImageNet.
- Efficient.

R-CNN outline



- generates category-independent region proposals
- a large convolutional neural network that extracts a fixed-length feature vector from each region
- a set of class-specific linear SVMs

Regional Convolutional Neural Network

Localization

Grouping superpixels

- Objectness [1]
- Selective search [2]
- Constrained parametric min-cuts (CPMC) [3]
- Multi-scale combinatorial grouping [4]

Sliding window

- EdgeBoxes

---

[1] B.Alexe,T.Deselaers,and V.Ferrari. Measuring the objectness of image windows. TPAMI, 2012.

[2] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013.

[3] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. TPAMI, 2012.

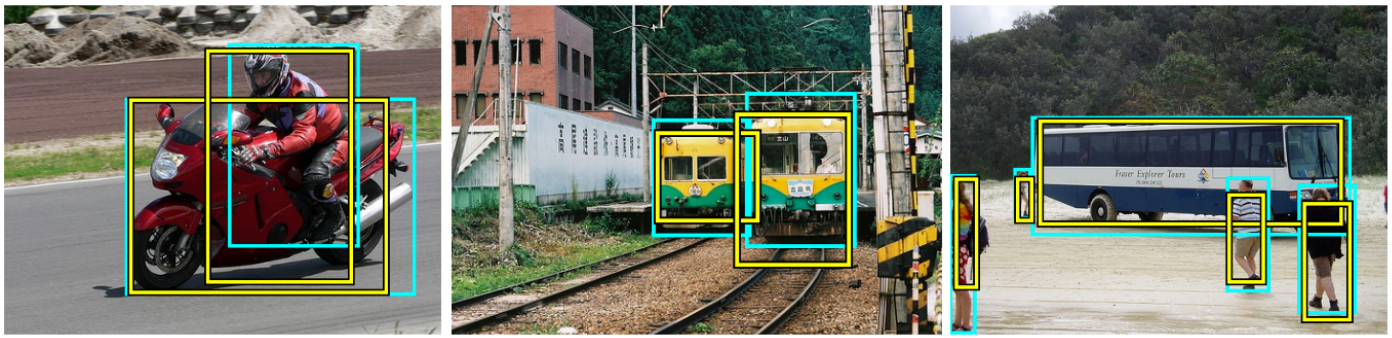
[4] P.Arbeláez,J.Pont-Tuset,J.Barron,F.Marques,and J.Malik. Multiscale combinatorial grouping. In CVPR, 2014.

Objectness

What is objectness?

The objectness measure acts as a class-generic object detector. It quantifies how likely it is for an image window to contain an object of any class, such as cars and dogs, as opposed to backgrounds, such as grass and water. [a]

[a] http://groups.inf.ed.ac.uk/calvin/objectness/



Objectness

- Sample Windows
- Judge whether a window contains an object

Objectness Cue

- Multi-scale Saliency (MS)
- Color Contrast (CC)
- Edge Density (ED)
- Superpixels Straddling (SS)
- Location and Size (LS)

Selective Search

- Capture All Scales.
  Objects can occur at any scale within the image.
- Diversification.
  There is no single optimal strategy to group regions together.
- Fast to Compute.
  The goal of selective search is to yield a set of possible object locations for use in a practical object recognition framework.

Selective Search–1 Hierarchical Grouping

- Efficient Graph-Based Image Segmentation, Pedro F. Felzenszwalb

Input: $G = (V, E)$, an undirected graph, where $v_i \in V$ is a pixel in the image. w_{ij} associated with $(v_i, v_j) \in E$ is measure of the dissimilarity between two pixels.
Output:$S = \{C_1, ..., C_r\}$. S is a segmentation of the graph, each component is denoted as C_i.
Core:

$$
\begin{aligned}
Int(C) &= \max_{e \in MST(C,E)} w(e) \\
MInt(C_1, C_2) &= \min(Int(C_1) + k/|C_1|, Int(C_2) + k/|C_2|) \\
Dif(C_1, C_2) &= \min_{v_i \in C_1, v_j \in C_2} w(v_i, v_j)
\end{aligned}
$$

if $Dif(C_1, C_2) < MInt(C_1, C_2)$ then join two components.

Selective Search–1 Hierarchical Grouping

Bottom-up grouping

- First the similarities between all neighbouring regions are calculated.
- The two most similar regions are grouped together, and new similarities are calculated between the resulting region and its neighbours.
- The process of grouping the most similar regions is repeated until the whole image becomes a single region.

Selective Search–2 Diversification Strategies

Diversify the sampling and create a set of complementary strategies whose locations are combined afterwards.

- Complementary Colour Spaces
- Complementary Similarity Measures (color,texture,size,fill)
- Complementary Starting Regions.

Selective Search–3 Combining Locations

Sort all proposals according some score

Given a grouping strategy j, let r_i^j be the region which is created at position i in the hierarchy, where $i = 1$ represents the top of the hierarchy.

We now calculate the position value v_i^j as $RND \times i$, where RND is a random number in range $[0, 1]$.

Feature Extractor

- CaffeNet

Convert the image data in that region into a form that is compatible with the CNN (227x227 pixel size).
Dilate the tight bounding box by p pixels. (we use p = 16).

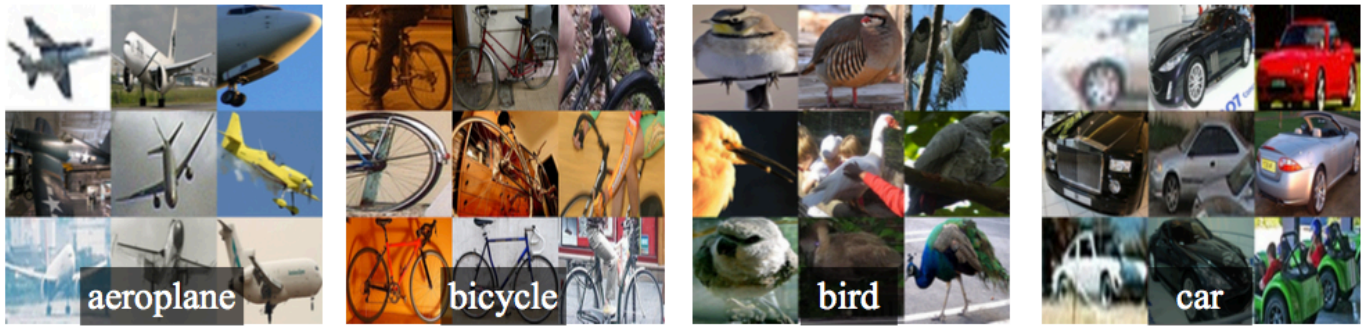


Figure 2: Warped training samples from VOC 2007 train.

Feature Extractor

Supervised pre-training

- CaffeNet
- ImageNet

Domain-specific fine-tuning

- Dataset VOC, ILSVRC2013. Less Class labels.
- Larger than 0.5 IoU (Intesection of Union) overlap with a ground-truth box as positives for that box' s class and the rest as negatives.
- Start SGD at a learning rate of 0.001 (1/10th of the initial pre-training rate), which allows fine-tuning to make progress while not clobbering the initialization.

Object category classifiers

For each class, we have a SVM

- Score all regions
- Perform an non-maximum suppression

non-maximum suppression

If two proposals have an overlap, we use the larger score one, and remove the other one.

Object category classifiers

For a car classifier:

Clear Points

- an image region tightly enclosing a car should be a positive example
- a background region, which has nothing to do with cars, should be a negative example.

Question

how to label a region that partially overlaps a car?

- Test on different IoU overlap rate threshold!

Bounding-box regression

After scoring each selective search proposal with a class-specific detection SVM, we predict a new bounding box for the detection using a class-specific bounding-box regressor.

Method:

- Simaliar with DPM(deformable part models) using CNN features

Problem Definition:
INPUT: (P^i, G^i), where P^i is the proposed region, and G^i is the ground truth.
OUTPUT a transformation that maps a proposed box P to a ground-truth box G.

Bounding-box regression

to be extended, (together with DPM)

We parameterize the transformation in terms of four functions $d_x(P)$, $d_y(P)$, $d_w(P)$, and $d_h(P)$. The first two specify a scale-invariant translation of the center of P's bounding box, while the second two specify log-space translations of the width and height of P's bounding box. After learning these functions, we can transform an input proposal P into a predicted ground-truth box $\hat{G}$ by applying the transformation

$$\hat{G}_x = P_w d_x(P) + P_x \tag{1}$$

$$\hat{G}_y = P_h d_y(P) + P_y \tag{2}$$

$$\hat{G}_w = P_w \exp(d_w(P)) \tag{3}$$

$$\hat{G}_h = P_h \exp(d_h(P)). \tag{4}$$

Regional Convolutional Neural Network

SPPnet

SPPnet [5]

Motive

Existing deep convolutional neural networks (CNNs) require a fixed-size (e.g., 224×224) input image.
The new network structure, called SPP-net, using "spatial pyramid pooling", can generate a fixed-length representation regardless of image size/scale.

Method

Using SPP-net, we compute the feature maps from the entire image only once, and then pool features in arbitrary regions (sub-images) to generate fixed-length representations for training the detectors. This method avoids repeatedly computing the convolutional features.

[5] K.He,X.Zhang,S.Ren,and J.Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV,2014. Recognition,

SPPnet

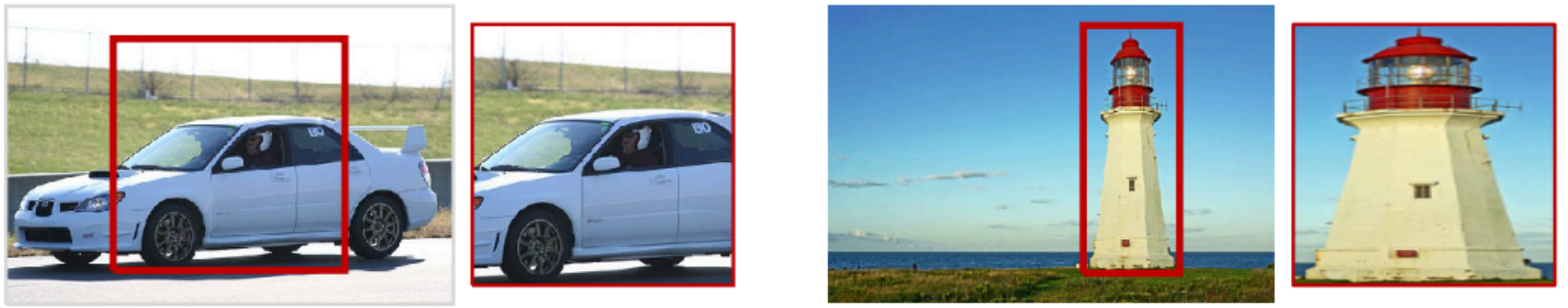


Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure.

Spatial pyramid pooling

Popularly known as spatial pyramid matching or SPM, as an extension of the Bag-of-Words (BoW) model, which partitions the image into divisions from finer to coarser levels, and aggregates local features in them.

Bag-of-Words Example

- For an article
- HoG (Histgram of Gradient)

Spatial pyramid pooling

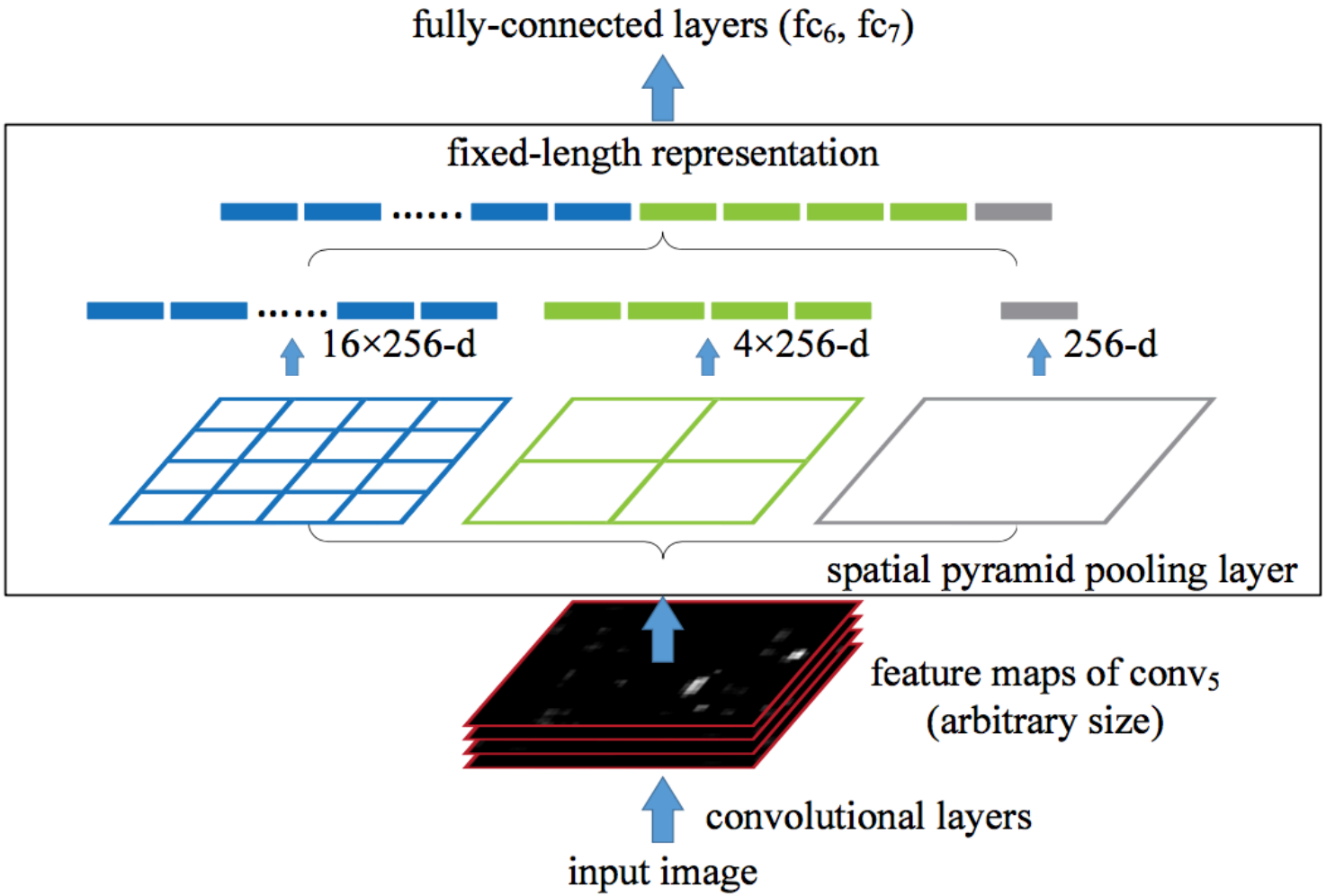


Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the

SPPnet for detection

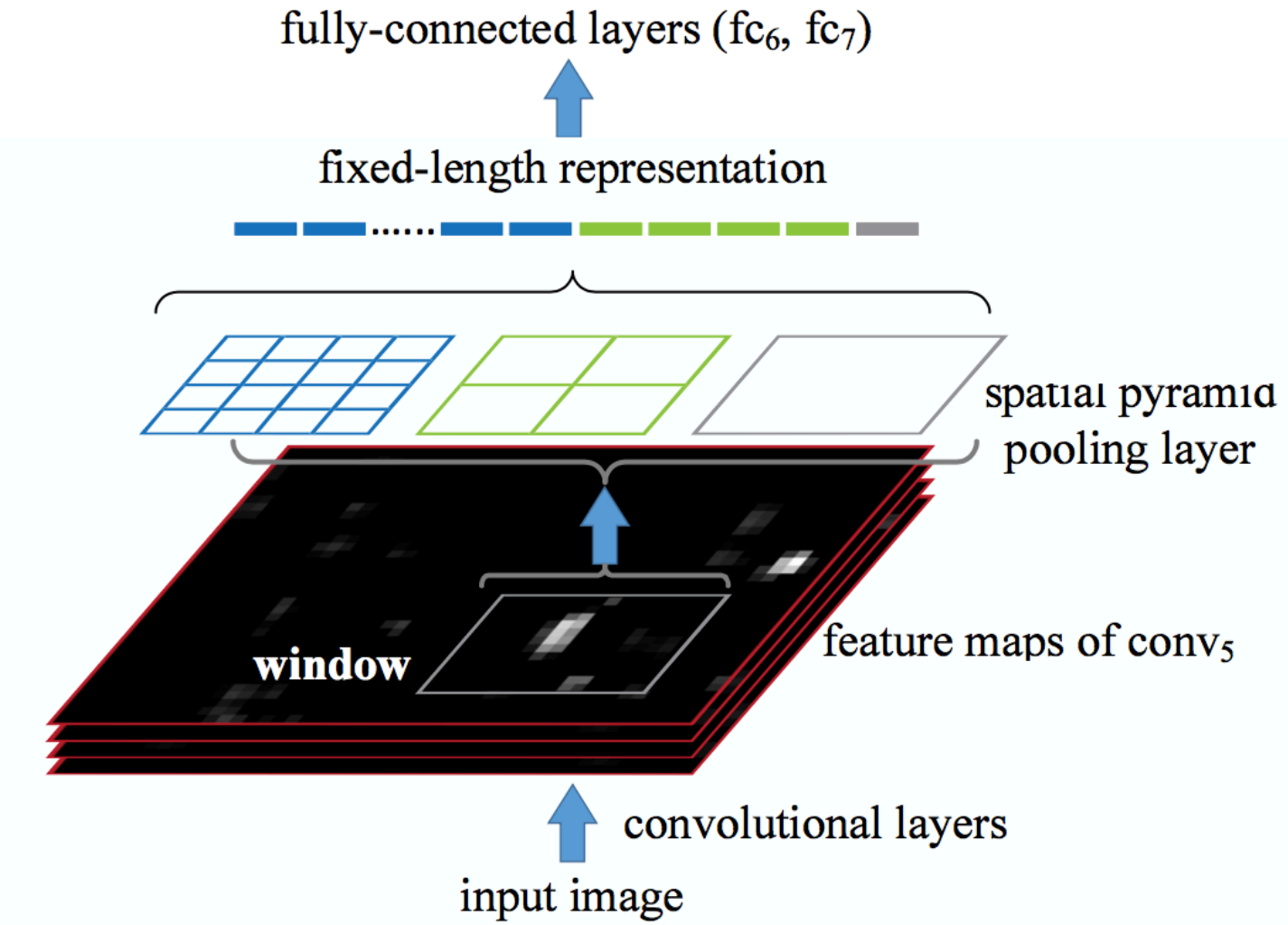


Figure 5: Pooling features from arbitrary windows

Regional Convolutional Neural Network

Fast R-CNN

R-CNN Drawbacks

- Training is a multi-stage pipeline (CNN extractor, SVM classifier, Bounding-box regression)
- Training is expensive in space and time.
- Object detection is slow.

Spatial pyramid pooling networks (SPPnets) were proposed to speed up R-CNN by sharing computation. The SPPnet method computes a convolutional feature map for the entire input image and then classifies each object proposal using a feature vector extracted from the shared feature map. (Deep, extract features in one pass)

Fast R-CNN architecture

INPUT an entire image, a set of object proposals.
STEP

- Processes the whole image with CNN to produce a conv feature map.
- For each object proposal a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map.

Each feature vector is fed into a sequence of fully connected (fc) layers that finally branch into two sibling output layers:

- one that produces softmax probability estimates over K object classes plus a catch-all "background" class
- another layer that outputs four real-valued numbers for each of the K object classes. Each set of 4 values encodes refined bounding-box positions for one of the K classes.

Fast R-CNN architecture

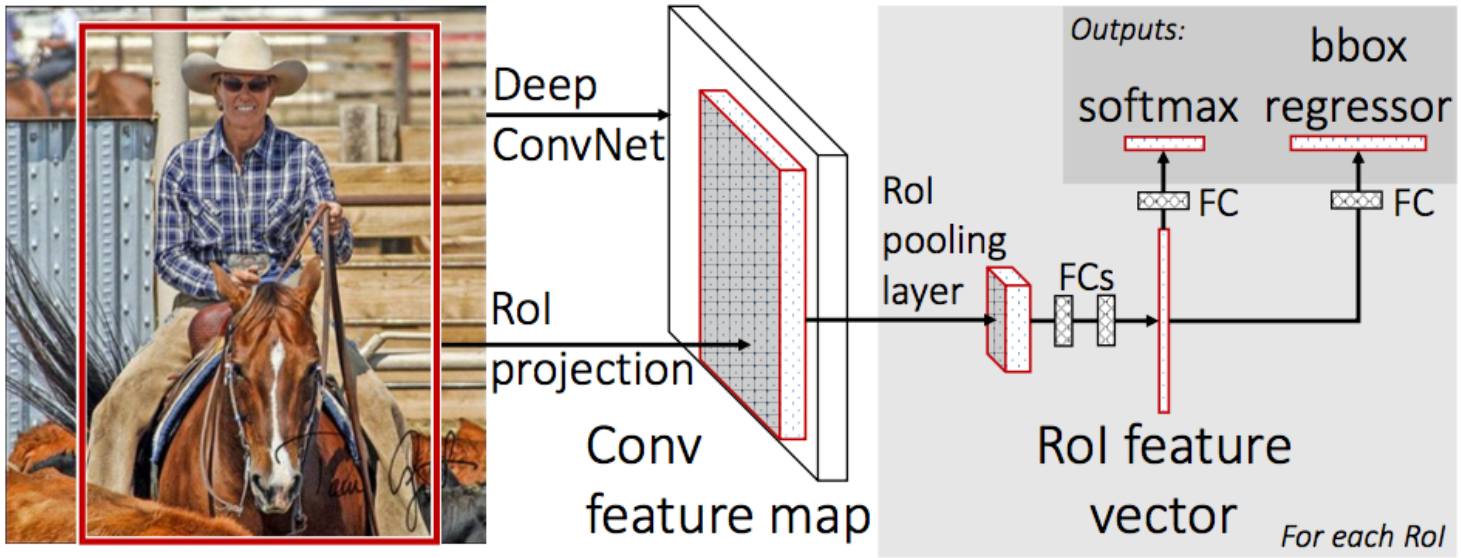


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

The RoI pooling layer

INPUT Feature map of different size
OUTPUT Fixsize map
STEP

- max pool (i.e. only one pyramid level)

Initializing from pre-trained networks

Experiment with three pre-trained ImageNet networks, each with five max pooling layers and between five and thirteen conv layers.
When a pre-trained network initializes a Fast R-CNN network, it undergoes three transformations.

- the last max pooling layer is replaced by a RoI pooling layer
- the last fully connected layer and softmax are replaced with the two sibling layers.
- take two data inputs: a list of images and a list of RoIs in those images.

Fine-tuning for detection

In Fast R-CNN training, stochastic gradient descent (SGD) mini-batches are sampled hierarchically, first by sampling N images and then by sampling R/N RoIs from each image.
Making N small decreases mini-batch computation.

experiment

When using N $= 2$ and R $= 128$, the proposed training scheme is roughly $64\times$ faster than sampling one RoI from 128 different images (i.e., the R-CNN and SPPnet strategy).

Multi-task loss

A Fast R-CNN network has two sibling output layers.

- $p = (p_0, ..., p_K)$ computed by a softmax over the $K + 1$ outputs of a fully connected layer.
- $t^k = (t^k_x, t^k_y, t^k_w, t^k_h)$ for each of the K object classes, indexed by k. (where t^k specifies a scale-invariant translation and log-space height/width shift relative to an object proposal.)

Each training RoI is labeled with a ground-truth class u and a ground-truth bounding-box regression target v.

multi-task loss:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

Multi-task loss

multi-task loss:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

The first iterm:

$$L_{cls}(p, u) = -\log p_u$$

$[u \geq 1]$ is 1 when $u \geq 1$, means it's a target.
And the last item:

$$L_{loc}(t^u, v) = \sum_i \text{smooth}_{L_1}(t_i^u - v_i)$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if} |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Truncated SVD for faster detection

Large fully connected layers are easily accelerated by compressing them with truncated SVD.
In this technique, a layer parameterized by the $u \times v$ weight matrix W is approximately factorized as

$$W \approx U \Sigma_t V^T$$

Truncated SVD reduces the parameter count from uv to $t(u + v)$, which can be significant if t is much smaller than $\min(u, v)$.

To compress a network, the single fully connected layer corresponding to W is replaced by two fully connected layers, without a non-linearity between them.

Regional Convolutional Neural Network

Intro

Introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. [6]

Motive

- Selective Search is an order of magnitude slower (2 seconds per image)
- Its accuracy depends on the performance of the region proposal module.[a]

[a] J.Hosang,R.Benenson,P.Dollar,and B.Schiele, "What makes for effective detection proposals?" IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015.

[6] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.

Other Proposed Network for Localization

- OverFeat [7]
  - a fully-connected layer is trained to predict the box coordinates for the localization task that assumes a single object.
  - The fully-connected layer is then turned into a convolutional layer for detecting multiple class-specific objects.
- MultiBox methods [8]
  - last fully-connected layer simultaneously predicts multiple class-agnostic boxes, generalizing the "single-box" fashion of OverFeat.
  - does not share features between the proposal and detection networks.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in International Conference on Learning Representations (ICLR), 2014.

[8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

Region Proposal Networks

INPUT: an image (of any size) as input and
OUTPUT a set of rectangular object proposals, each with an objectness score.
FEATURE integrate with the detection network, by using 'attention' [9] mechanisms to tell the Fast R-CNN module where to look.
STEP

- slide a small network over the convolutional feature map output by the last shared convolutional layer (input 3×3), mapping to a lower-dimensional feature
- fed into two sibling fully-connected layers: a box-regression layer (reg) and a box-classification layer (cls). Each sliding window is mapped to.

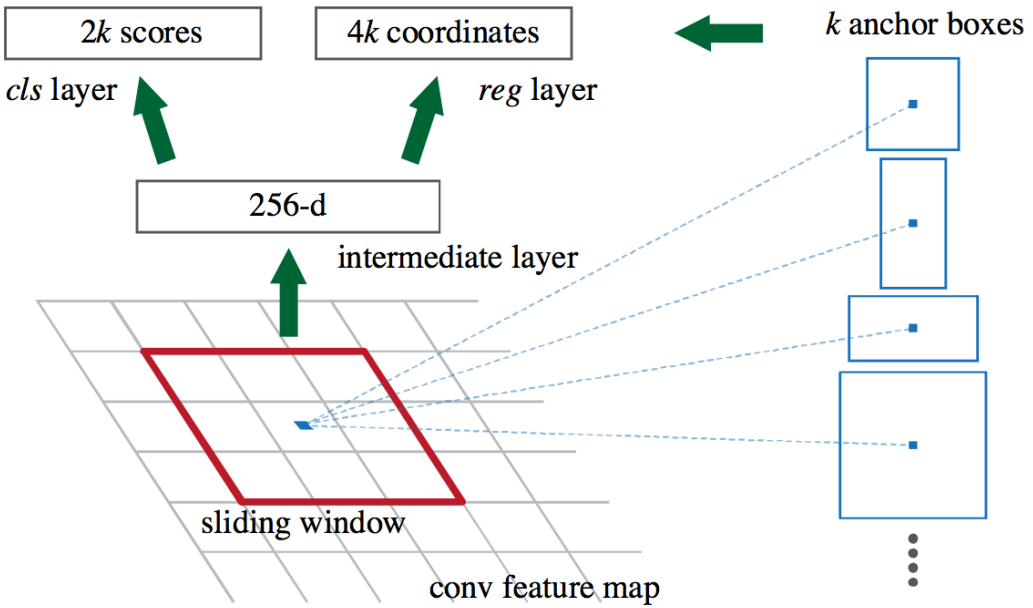
[9] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in Neural Information Processing Systems (NIPS), 2015.

Region Proposal Networks–Anchors

Anchors : At each sliding-window location, we simultaneously predict k region proposals, called Anchors.

- centered at the sliding window
- associated with a scale and aspect ratio (128,256,512; 1:1,2:1,1:2)

For a convolutional feature map of a size $W \times H$ (typically $\sim$ 2,400), there are WHk anchors in total. (Finally select some highest scores)



RPN–Loss Function

For training RPNs, we assign a binary class label (of being an object or not) to each anchor.

- positive anchors:
  - an anchor that has an IoU overlap higher than 0.7 with any ground-truth box. (sometimes, there's none)
  - the anchor/anchors with the highest Intersection-over-Union (IoU) overlap with a ground-truth box.
- non-positive anchors:
  - IoU ratio is lower than 0.3 for all ground-truth boxes.

Anchors that are neither positive nor negative do not contribute to the training objective.

RPN–Loss Function

Multi-task Loss Function

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

- The classification loss L_{cls} is log loss over two classes (object vs. not object)
- The regression loss L_{reg} also use smooth-L1

Training RPNs

- resample to a ratio of 1:1.

Sharing Features for RPN and Fast R-CNN

Both RPN and Fast R-CNN, trained independently, will modify their convolutional layers in different ways.
We discuss three ways for training networks with features shared:

- Alternating training
  - The network tuned by Fast R-CNN is then used to initialize RPN, and this process is iterated.
- Approximate joint training
  - both the RPN loss and the Fast R-CNN loss are combined.
- Non-approximate joint training
  - A theoretically valid backpropagation solver should also involve gradients w.r.t. the box coordinates. (These gradients are ignored in the above approximate joint training.)
  - need an RoI pooling layer that is differentiable w.r.t. the box coordinates.

Faster R-CNN–Graph

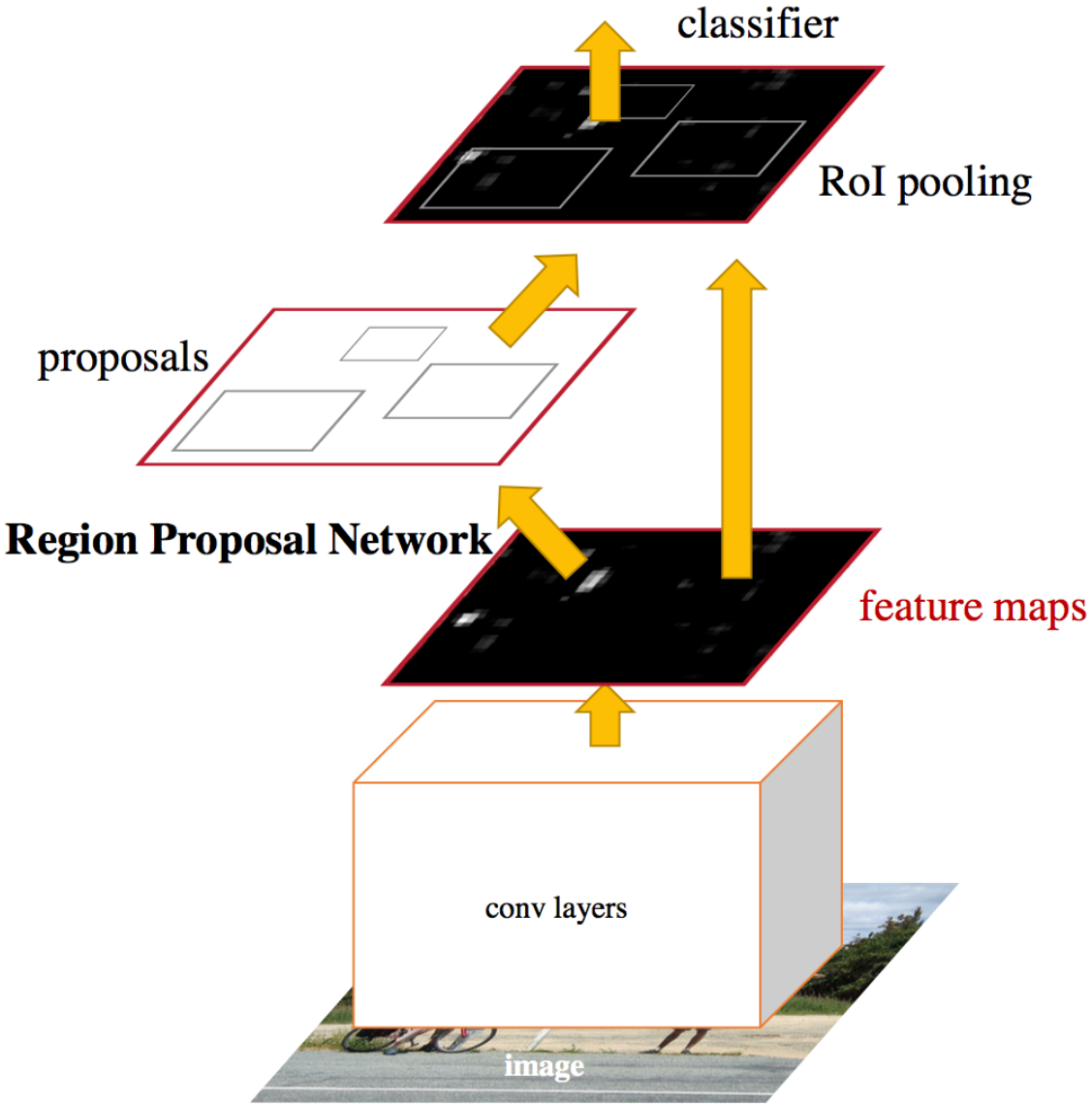
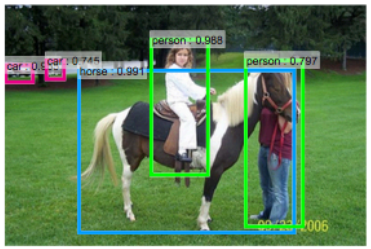
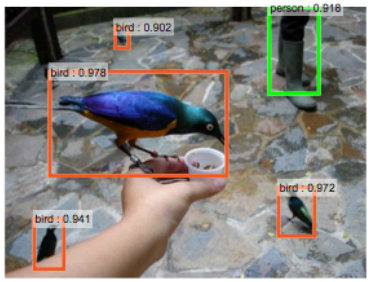
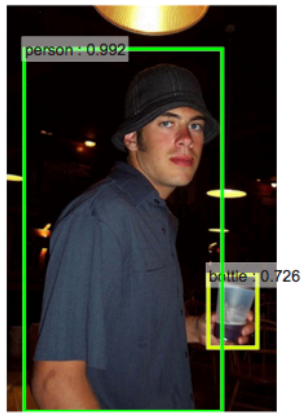


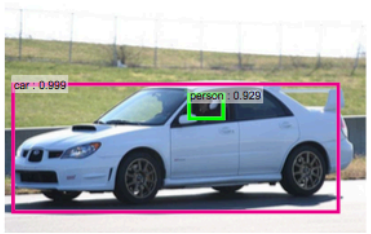
Figure 2: Faster R-CNN is a single, unified network

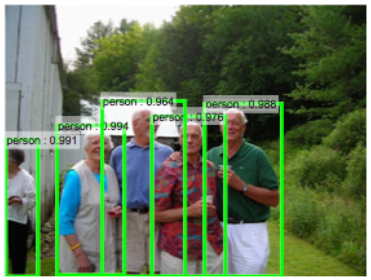
Faster R-CNN–Performance

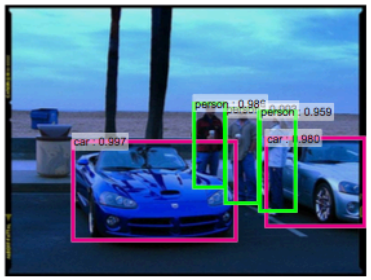


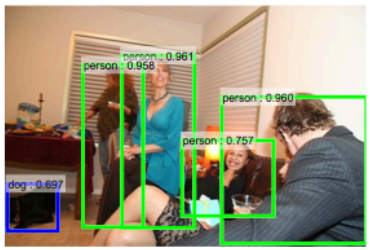


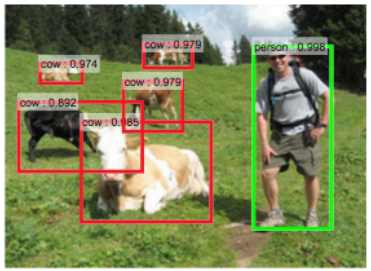












End

Thanks

Further Reading:

- Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." arXiv preprint 1506.02640 (2015). (R.B.G)

main reference:

SS	Selective search for object recognition.
R-CNN	Rich feature hierarchies for accurate object detection and semantic segmentation Tech report
SPPnet	Spatial pyramid pooling in deep convolutional networks for visual recognition.
Fast R-CNN	Fast R-CNN
Faster R-CNN	Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks