Object detection with CNNs

PASCAL VOC

Before CNNs

After CNNs

mean Average Precision (mAP)

year

0% 10% 20% 30% 40% 50% 60% 70% 80%

Region proposals

- As an alternative to sliding window search, evaluate a few hundred region proposals
  - Can use slower but more powerful features and classifiers
  - Take advantage of low-level perceptual organization cues
  - Proposal mechanism can be category-independent
  - Proposal mechanism can be trained
Selective search

- Use hierarchical segmentation: start with small superpixels and merge based on diverse cues

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013
Evaluation of region proposals

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders,
Selective Search for Object Recognition, IJCV 2013
Selective search detection pipeline

- Feature extraction: color SIFT, codebook of size 4K, spatial pyramid with four levels = 360K dimensions

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013
Another proposal method: EdgeBoxes

- Box score: number of edges in the box minus number of edges that overlap the box boundary
- Uses a trained edge detector
- Uses efficient data structures (incl. integral images) for fast evaluation
- Gets 75% recall with 800 boxes (vs. 1400 for Selective Search), is 40 times faster

R-CNN: Region proposals + CNN features

Source: R. Girshick

R-CNN details

- **Regions**: ~2000 Selective Search proposals
- **Network**: AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- **Bounding box regression** to refine box locations
- **Performance**: mAP of 53.7% on PASCAL 2010 (vs. 35.1% for Selective Search and 33.4% for DPM).

R-CNN pros and cons

• Pros
  • Accurate!
  • Any deep architecture can immediately be “plugged in”

• Cons
  • Ad hoc training objectives
    • Fine-tune network with softmax classifier (log loss)
    • Train post-hoc linear SVMs (hinge loss)
    • Train post-hoc bounding-box regressions (least squares)
  • Training is slow (84h), takes a lot of disk space
    • 2000 CNN passes per image
  • Inference (detection) is slow (47s / image with VGG16)
Fast R-CNN

Forward whole image through ConvNet

“conv5” feature map of image

“RoI Pooling” layer

Fully-connected layers

Bounding-box regressors

Linear

Softmax classifier

Linear + softmax

Region proposals

FCs

Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015
Fast R-CNN training

Log loss + smooth L1 loss

Linear + softmax

Linear

FCs

ConvNet

Multi-task loss

Trainable

Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015
Fast R-CNN: Another view

R. Girshick, Fast R-CNN, ICCV 2015
ROI pooling closeup

(actual output feature map size: 7 x 7)
Fast R-CNN results

<table>
<thead>
<tr>
<th></th>
<th>Fast R-CNN</th>
<th>R-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train time (h)</td>
<td>9.5</td>
<td>84</td>
</tr>
<tr>
<td>- Speedup</td>
<td>8.8x</td>
<td>1x</td>
</tr>
<tr>
<td>Test time / image</td>
<td>0.32s</td>
<td>47.0s</td>
</tr>
<tr>
<td>Test speedup</td>
<td>146x</td>
<td>1x</td>
</tr>
<tr>
<td>mAP</td>
<td>66.9%</td>
<td>66.0%</td>
</tr>
</tbody>
</table>

Timings exclude object proposal time, which is equal for all methods. All methods use VGG16 from Simonyan and Zisserman.

Source: R. Girshick
Faster R-CNN

S. Ren, K. He, R. Girshick, and J. Sun,
Region proposal network (RPN)

• Slide a small window over the conv5 layer
  • Predict object/no object
  • Regress bounding box coordinates
  • Box regression is with reference to anchors (3 scales x 3 aspect ratios)
Faster R-CNN results

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet
Object detection progress

![Graph showing the progress of object detection with different methods over years. The x-axis represents the year from 2006 to 2016, and the y-axis represents the mean Average Precision (mAP) ranging from 0% to 80%. The graph highlights different phases: Before deep convnets and Using deep convnets. The methods R-CNNv1, Fast R-CNN, and Faster R-CNN are marked with symbols on the graph.](image-url)
YOLO

1. Take conv feature maps at 7x7 resolution
2. Add two FC layers to predict, at each location, a score for each class and 2 bboxes w/ confidences (output is 7x7x30 for PASCAL)

- 7x speedup over Faster RCNN (45-155 FPS vs. 7-18 FPS)
- Some loss of accuracy due to lower recall, poor localization

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi,
YOLO v2

- Batch normalization
- Pre-train on higher resolution ImageNet
- Use and improve anchor box idea from Faster RCNN
- Train at multiple resolutions
- Very good accuracy, very fast

SSD

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, 
Feature pyramid networks

• Improve predictive power of lower-level feature maps by adding contextual information from higher-level feature maps

• Predict different sizes of bounding boxes from different levels of the pyramid (but share parameters of predictors)

New detection benchmark: COCO (2014)

- 80 categories instead of PASCAL’s 20
- Current best mAP: 52%

http://cocodataset.org/#home
New detection benchmark: COCO (2014)

J. Huang et al.,
**Speed/accuracy trade-offs for modern convolutional object detectors**, CVPR 2017
Review: Object detection with CNNs
Review: R-CNN

R. Girshick, J. Donahue, T. Darrell, and J. Malik,
Review: Fast R-CNN

- Forward whole image through ConvNet
- “conv5” feature map of image
- “RoI Pooling” layer
- Fully-connected layers
- Bounding-box regressors
- Linear + softmax
- FCs
- Softmax classifier
- Region proposals

R. Girshick, Fast R-CNN, ICCV 2015
Review: Faster R-CNN

S. Ren, K. He, R. Girshick, and J. Sun,
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015
Summary: Object detection with CNNs

- R-CNN: region proposals + CNN on cropped, resampled regions
- Fast R-CNN: region proposals + RoI pooling on top of a conv feature map
- Faster R-CNN: RPN + RoI pooling
- Next generation of detectors
  - Direct prediction of BB offsets, class scores on top of conv feature maps
  - Get better context by combining feature maps at multiple resolutions