R-CNN minus R

Karel Lenc, Andrea Vedaldi
Object detection

**Goal:** tightly enclose objects of a certain type in a bounding box

- “bikes”
- “planes”
- “horses”
- “birds”
Top performer: Region proposals + CNN

WHERE
[Image 267x568 to 354x656]
[Image 266x467 to 354x555]
[Image 267x365 to 354x453]
[Image 620x101 to 809x289]
[Image 214x102 to 402x290]

WHAT

- Chair
- Background
- Potted plant

WHERE

[Image 214x102 to 402x290]

[Proposal generation]

Top performer: Region proposals + CNN

**WHAT**

Convolutional neural networks

E.g. region classification with AlexNet, VGG VD

**WHERE**

Segmentation algorithm

E.g. region proposal from selective search
[Uijlings et al. 2013]
Can CNN understand *where* as well as *what*?

Convolutional neural network?

WHAT

WHERE
Approaches to object detection

Scanning windows

- **Sliding windows**
  - HOG detector [Dalal Triggs 2005]
  - DPM [Felzenszwalb et al. 2008]
- **Cascaded windows**
  - AdaBoost [Viola Jones 2004]
  - MKL [Vedaldi et al. 2009]
  - B and Bound [Lampert et al. 2009]
- **Jumping windows**
  - [Sivic et al. 2008]
- **Selective windows**

Hough voting

- Implicit shape models [Amit Geman 1997, Leibe et al. 2003]
- Max margin [Maji Berg 2009], Random Forests [Gall Lempitsky 2009]

Classifiers & features

- linear SVMs, kernel SVM, Fisher Vectors, … [Cinbis et al. 2013, …]
- HOG, SIFT, C-SIFT, … [van de Sande et al. 2010, …]
- Segmentation cues, … [Shotton et al. 2008, Cinbis et al. 2013, …]
Evolution of object detection

PASCAL VOC 2007 data

Year

mAP [%]


0 10 20 30 40 50 60 70

DPM [Felzenszwalb et al.]

MKL [Vedaldi et al.]

DPMv5 [Girshick et al.]

Regionlet [Wang et al.]

RCNN-Alex [Girshick et al.]

RCNN-VGG [Girshick et al.]
**Pros**: simple and effective

**Cons**: slow as the CNN is re-evaluated for each tested region

[R-CNN](Girshick et al. 2013)
**SPP R-CNN**

[He et al. 2014]

Convolutional features = local features

Region descriptor = pooled local features

- Spatial pyramid + max pooling [He et al. 2014]
- Bag of words, Fisher vector, VLAD, .... [Cimpoi et. al. 2015]

Order of magnitudes speedup
SPP-CNN results in a significant test-time speedup

However, region proposal extraction is the new bottleneck

R-CNN minus R: can we get rid of region proposal extraction?
Streamlining R-CNN and SPP-CNN

Dropping proposal generation
Streamlining R-CNN and SPP-CNN

Dropping proposal generation
1. Pre-train a large CNN (on ImageNet)

2. Extract region proposals (on PASCAL VOC)

3. Use pre-processed regions to:
   1. Fine-tune the CNN
   2. Learn an SVM to rank regions
   3. Learn a bounding-box regressor to refine localization

(SPP) R-CNN training comprises many steps
A complex learning pipeline

(SPP) R-CNN training comprises many steps

With SPP R-CNN of [He et al. 2014]
fine-tuning is limited to the fully connected layers
Streamlining R-CNN

Removing the SVM

<table>
<thead>
<tr>
<th>phase</th>
<th>score(s)</th>
<th>learning loss</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>fine tuning</td>
<td>$S_c = \exp(\langle w_c, \phi(x) \rangle + b_c)$</td>
<td>$- \log \frac{S_{c_0}}{S_0 + S_1 + S_2 + \ldots + S_C}$</td>
<td>38.1</td>
</tr>
<tr>
<td>region ranking</td>
<td>$Q_1 = \langle w_1, \phi(x) \rangle + b_1$</td>
<td>$\max{0, 1 - y Q_1}$</td>
<td>59.8</td>
</tr>
<tr>
<td></td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$Q_C = \langle w_C, \phi(x) \rangle + b_C$</td>
<td>$\max{0, 1 - y Q_C}$</td>
<td></td>
</tr>
<tr>
<td>region ranking</td>
<td>$Q_c = \log \frac{S_c}{S_0}$</td>
<td>from fine-tuning</td>
<td>58.4</td>
</tr>
</tbody>
</table>

Up to a simple transformation, softmax is just as good as hinge loss for box ranking.
SPP and bounding box regressions can be easily implemented in a CNN (with a DAG topology) and trained jointly in one step.
Streamlining R-CNN and SPP-CNN

Dropping proposal generation
A constant-time region proposal generator

Algorithm

**Preprocessing**

Collect all the training bounding boxes \((x_1, y_1, x_2, y_2)\)

Use K-means to extract K clusters in \((x_1, y_1, x_2, y_2)\) space

**Proposal generation**

Regardless of the image, return the same K cluster centers

Proposals are now very fast but very inaccurate

We let the CNN compensate with the bounding box regressor
Proposal statistics on PASCAL VOC

- **ground truth**
- **selective search**
  - 2K
- **sliding windows**
  - 7K
- **clustering**
  - 3K

Images showing distribution of proposals for each category.
Information pathways

[See also Lenc Vedaldi CPVR 2015]
CNN-based bounding box regression

**Dashed line:** proposals

**Solid line:** corrected by the CNN
Observations

- **Selective search** is much better than fixed generators
- However, **bounding box regression** almost eliminates the difference
- **Clustering** allows to use significantly less boxes than sliding windows
Finding (1) Streamlining accelerates SPP
Finding (2) Dropping selective search is a huge benefit

Average Time per Image [ms]

- **Streamlined SPP**
  - Minus R

- **SPP**
  - Sel. Search
  - Im. Prep.
  - GPU↔CPU
  - CONV Layers
  - Spat. Pooling
  - FC Layers
  - Bbox Regr.
Finding (2) Dropping selective search is a huge benefit
Timings

Test-time speedups

Times faster than R-CNN

- Minus R: 67.5 times faster
- Streamlined SPP: 5.0 times faster
- SPP: 4.5 times faster
- RCNN: 1.0 times faster
Conclusions

Current CNNs can localize objects well

- External segmentation cues bring only a minor benefit at a great expense

Benefits of CNN-only solutions

- Much faster, particularly at test time
- Much simpler and streamlined implementations

Future steps

- Eliminate the remaining accuracy gap
  - Essentially achieved in [Faster R-CNN, Ren et al. 2015]
- Beyond bounding boxes
- Beyond detection