Object detection
The Task
Datasets

- Face detection
- One category: face
- Frontal faces
- Fairly rigid, unoccluded

Pedestrians

- One category: pedestrians
- Slight pose variations and small distortions
- Partial occlusions

Histograms of Oriented Gradients for Human Detection. N. Dalal and B. Triggs. CVPR 2005
PASCAL VOC

• 20 categories
• 10K images
• Large pose variations, heavy occlusions
• Generic scenes
• Cleaned up performance metric

1990’s
2000’s
2007 - 2012
Coco

- 80 diverse categories
- 100K images
- Heavy occlusions, many objects per image, large scale variations
Evaluation metric
Matching detections to ground truth

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$
Matching detections to ground truth

• Match detection to most similar ground truth
  • highest IoU
• If IoU > 50%, mark as correct
• If multiple detections map to same ground truth, mark only one as correct
• **Precision** = #correct detections / total detections
• **Recall** = #ground truth with matched detections / total ground truth
Tradeoff between precision and recall

• ML usually gives scores or probabilities, so threshold
• Too low threshold $\rightarrow$ too many detections $\rightarrow$ low precision, high recall
• Too high threshold $\rightarrow$ too few detections $\rightarrow$ high precision, low recall
• Right tradeoff depends on application
  • Detecting cancer cells in tissue: need high recall
  • Detecting edible mushrooms in forest: need high precision
Average precision
Average precision
Average average precision

- AP marks detections with overlap > 50% as correct
- But may need better localization
- Average AP across multiple overlap thresholds
- Confusingly, still called average precision
- Introduced in COCO
Mean and category-wise AP

• Every category evaluated independently
• Typically report mean AP averaged over all categories
• Confusingly called “mean Average Precision”, or “mAP”
Why is detection hard(er)?

- Precise localization
Why is detection hard(er)?

- Much larger impact of pose
Why is detection hard(er)?

• Occlusion makes localization difficult
Why is detection hard(er)?

- Counting
Why is detection hard(er)?

- Small objects
Detection as classification

• Run through every possible box and classify
• How many boxes?
  • Every pair of pixels = 1 box

\[ \binom{N}{2} = O(N^2) \]

• For 300 x 500 image, \( N = 150K \)
• \( 2.25 \times 10^{10} \) boxes!
Idea 1: scanning window

- Fix size
  - Can take a few different sizes
- Fixed stride
- Convolution with a filter
  - Classic: compute HOG features over entire image
Dealing with scale
Dealing with scale

- Use same window size, but run on *image pyramid*
Issues

• Classifies millions of boxes, so must be very fast
• Needs ultra-fine sampling of scales and object sizes, can still miss outlier sizes
## Scanning window results on PASCAL

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2007</th>
<th>VOC 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
<td>29.6%</td>
</tr>
</tbody>
</table>

**Reference systems**

**Slide credit**: Ross Girshick
Idea 2: Object proposals

- Use segmentation to produce ~5K candidates

Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.
Idea 2: object proposals

• Many different segmentation algorithms (k-means on color, k-means on color+position, N-cuts....)
• Many hyperparameters (number of clusters, weights on edges)
• Try everything!
  • Every cluster is a candidate object
  • Thousands of segmentations -> thousands of candidate objects
Idea 2: Object proposals

• Tens of ways of generating candidates ("proposals")
• What fraction of ground truth objects have proposals near them?

What makes for effective detection proposals? J. Hosang, R. Benenson, P. Dollar, B. Schiele. In TPAMI
What do we do with proposals?

• Each proposal is a group of pixels
• Take tight fitting box and **classify it**
• *Can leverage any image classification approach*
## Proposal methods results

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2007</th>
<th>VOC 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
<td>29.6%</td>
</tr>
<tr>
<td>UVA sel. search (Uijlings et al. 2013)</td>
<td></td>
<td>35.1%</td>
</tr>
<tr>
<td>Regionlets (Wang et al. 2013)</td>
<td></td>
<td>41.7%</td>
</tr>
<tr>
<td>SegDPM (Fidler et al. 2013)</td>
<td></td>
<td>40.4%</td>
</tr>
<tr>
<td>R-CNN</td>
<td>54.2%</td>
<td>50.2%</td>
</tr>
<tr>
<td>R-CNN + bbox regression</td>
<td>58.5%</td>
<td>53.7%</td>
</tr>
</tbody>
</table>

### Reference systems

- **VOC 2007**
- **VOC 2010**

- **Proposal methods results**

---

*Slide credit: Ross Girshick*
Proposal methods results

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2007</th>
<th>VOC 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
<td>29.6%</td>
</tr>
<tr>
<td>UVA sel. search (Uijlings et al. 2013)</td>
<td></td>
<td>35.1%</td>
</tr>
<tr>
<td>Regionlets (Wang et al. 2013)</td>
<td>41.7%</td>
<td>39.7%</td>
</tr>
<tr>
<td>SegDPM (Fidler et al. 2013)</td>
<td></td>
<td>40.4%</td>
</tr>
</tbody>
</table>

Reference systems

metric: mean average precision (higher is better)

Slide credit: Ross Girshick
R-CNN: Regions with CNN features

Input image  
Extract region proposals (~2k / image)  
Compute CNN features  
Classify regions (linear SVM)

Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation

R. Girshick, J. Donahue, T. Darrell, J. Malik
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

Slide credit: Ross Girshick
R-CNN at test time: Step 2

Input image

Extract region proposals (~2k / image)

Compute CNN features

Slide credit: Ross Girshick
R-CNN at test time: Step 2

Input image

Extract region proposals (~2k / image)

Compute CNN features

a. Crop

b. Scale (anisotropic)

227 x 227

Slide credit: Ross Girshick
R-CNN at test time: Step 2

1. Crop
2. Scale (anisotropic)

Input image → Extract region proposals (~2k / image) → Compute CNN features

Output: “fc7” features

Slide credit: Ross Girshick
R-CNN at test time: Step 3

Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions

Warped proposal

4096-dimensional fc7 feature vector

linear classifiers (SVM or softmax)

person? 1.6

horse? -0.3

Slide credit: Ross Girshick
Step 4: Object proposal refinement

Original proposal

Linear regression on CNN features

Predicted object bounding box

Bounding-box regression

Slide credit: Ross Girshick
## R-CNN results on PASCAL

<table>
<thead>
<tr>
<th>Reference systems</th>
<th>VOC 2007</th>
<th>VOC 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
<td>29.6%</td>
</tr>
<tr>
<td>UVA sel. search (Uijlings et al. 2013)</td>
<td></td>
<td>35.1%</td>
</tr>
<tr>
<td>Regionlets (Wang et al. 2013)</td>
<td>41.7%</td>
<td>39.7%</td>
</tr>
<tr>
<td>SegDPM (Fidler et al. 2013)</td>
<td></td>
<td>40.4%</td>
</tr>
</tbody>
</table>

Metric: mean average precision (higher is better)

Slide credit: Ross Girshick
## R-CNN results on PASCAL

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2007</th>
<th>VOC 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
<td>29.6%</td>
</tr>
<tr>
<td>UVA sel. search (Uijlings et al. 2013)</td>
<td></td>
<td>35.1%</td>
</tr>
<tr>
<td>Regionlets (Wang et al. 2013)</td>
<td>41.7%</td>
<td>39.7%</td>
</tr>
<tr>
<td>SegDPM (Fidler et al. 2013)</td>
<td></td>
<td>40.4%</td>
</tr>
<tr>
<td>R-CNN</td>
<td>54.2%</td>
<td>50.2%</td>
</tr>
<tr>
<td>R-CNN + bbox regression</td>
<td>58.5%</td>
<td>53.7%</td>
</tr>
</tbody>
</table>

*metric: mean average precision (higher is better)*

Slide credit: Ross Girshick
Training R-CNN

• Train convolutional network on ImageNet classification

• *Finetune* on detection
  • Classification problem!
  • Proposals with IoU > 50% are positives
  • Sample fixed proportion of positives in each batch because of imbalance
Other details - Non-max suppression

How do we deal with multiple detections on the same object?
Other details - Non-max suppression

• Go down the list of detections starting from highest scoring
• Eliminate any detection that overlaps highly with a higher scoring detection
• Separate, heuristic step
Speeding up R-CNN
Speeding up R-CNN
ROI Pooling

- How do we crop from a feature map?
- Step 1: Resize boxes to account for subsampling

Fast R-CNN. Ross Girshick. In ICCV 2015
ROI Pooling

- How do we crop from a feature map?
- Step 2: Snap to feature map grid
ROI Pooling

• How do we crop from a feature map?
• Step 3: Place a grid of fixed size
ROI Pooling

• How do we crop from a feature map?
• Step 4: Take max in each cell
## Fast R-CNN

<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>Speedup</td>
<td>146x</td>
<td>1x</td>
</tr>
<tr>
<td>mean AP</td>
<td>66.9</td>
<td>66.0</td>
</tr>
</tbody>
</table>
Fast R-CNN

- Bottleneck remaining (not included in time):
  - Object proposal generation

- Slow
  - Requires segmentation
  - $O(1s)$ per image
Faster R-CNN

• Can we produce *object proposals* from convolutional networks?
• A change in intuition
  • Instead of using grouping
  • Recognize likely objects?
• For every possible box, score if it is likely to correspond to an object

Faster R-CNN
Faster R-CNN

• At each location, consider boxes of many different sizes and aspect ratios
Faster R-CNN

• At each location, consider boxes of many different sizes and aspect ratios
Faster R-CNN

- At each location, consider boxes of many different sizes.
Faster R-CNN

• $s$ scales * $a$ aspect ratios = $sa$ anchor boxes
• Use convolutional layer on top of filter map to produce $sa$ scores
• Pick top few boxes as proposals
## Faster R-CNN

<table>
<thead>
<tr>
<th>Method</th>
<th>mean AP (PASCAL VOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast R-CNN</td>
<td>65.7</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>67.0</td>
</tr>
</tbody>
</table>
### Impact of Feature Extractors

<table>
<thead>
<tr>
<th>ConvNet</th>
<th>mean AP (PASCAL VOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG</td>
<td>70.4</td>
</tr>
<tr>
<td>ResNet 101</td>
<td>73.8</td>
</tr>
</tbody>
</table>
# Impact of Additional Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Training data</th>
<th>mean AP (PASCAL VOC 2012 Test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast R-CNN</td>
<td>VOC 12 Train (10K)</td>
<td>65.7</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>VOC07 Trainval + VOC 12 Train</td>
<td>68.4</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>VOC 12 Train (10K)</td>
<td>67.0</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>VOC07 Trainval + VOC 12 Train</td>
<td>70.4</td>
</tr>
</tbody>
</table>
The R-CNN family of detectors

Mean AP

- R-CNN/10K/VGG
- Fast R-CNN/10K/VGG
- Fast R-CNN/20K/VGG
- Faster R-CNN/20K/VGG
- Faster R-CNN/20K/ResNet101

Mean AP