Object detection
The Task

person 1

person 2

horse 1

horse 2
Datasets

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

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

Histories 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
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

Precision

Recall
Average precision

![Graph showing average precision with x-axis labeled 'Recall' and y-axis labeled '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
  • Well-localized object of class k or not?

• 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!

• Related challenge: almost all boxes are negative!
Idea 1: scanning window

• Fix size
• Fix stride
• Crop boxes and classify

Alternatively
  • Compute collection of feature maps
  • Convolve with filter
Multiple object sizes

• Objects can appear at any size

• *Discretize* set of sizes into a few different sizes
  • Sometimes called “anchors”

• Train separate classifier for each size
Dealing with large scale changes
Dealing with large scale changes

- Use an image pyramid
- Run same detector at multiple scales
- Take union of results
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.
Object proposals

• Basic idea: use grouping cues to identify segments that are likely to be objects

• Multiple versions
  • Do graph cuts with different seeds
  • Oversegment and then combinatorially group nearby objects
Two classes of object detection approaches

- Object proposal-based
  - Also called two-stage detectors
  - Canonical examples
    - R-CNN family
  - Pros:
    - Smaller number of candidates to classify
    - Less class imbalance
    - “Cascade” approach
  - Cons:
    - More complex, slower
    - Can miss due to missed proposals

- Scanning window-based
  - Also called single-stage detectors
  - Canonical examples
    - SSD family
  - Pros
    - Simpler
    - Faster
  - Cons
    - Larger number of candidates, more class imbalance
    - Can miss due to mismatched size
ConvNet-based object detection
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
Step 1: 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.
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)

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

1. Crop
2. Scale (anisotropic)
3. Forward propagate

Output: “fc7” features

Input image
Extract region proposals (~2k / image)
Compute CNN features
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
Bounding-box regression

\[(x, y)\]

original

predicted

\[\Delta w \times w + w\]

\[\Delta h \times h + h\]

\[(\Delta x \times w + x, \Delta y \times h + h)\]
# 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>
</tbody>
</table>

**Reference systems**

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><strong>R-CNN</strong></td>
<td>54.2%</td>
<td>50.2%</td>
</tr>
<tr>
<td><strong>R-CNN + bbox regression</strong></td>
<td>58.5%</td>
<td>53.7%</td>
</tr>
</tbody>
</table>

The metric is mean average precision (higher is better).
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
Speeding up R-CNN

• Each box requires a ConvNet run
• 2k boxes $\rightarrow$ 2000 times slower than classification!
• Can we share feature computation between the boxes?
Speeding up R-CNN

• Each box requires a ConvNet run
• 2k boxes $\rightarrow$ 2000 times slower than classification!
• Can we share feature computation between the boxes?
ROI Pooling

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

Layer 1
Layer 2
Layer 3

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: Overlay a new 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

• *Cascade*

Faster R-CNN
Faster R-CNN

• At each location, consider boxes of many different sizes and aspect ratios
• If k such sizes, use simple convolutional layer to output k ”objectness scores”
Faster R-CNN

• At each location, consider boxes of many different sizes and aspect ratios

• If k such sizes, use simple convolutional layer to output k "objectness scores"
Faster R-CNN

- At each location, consider boxes of many different sizes and aspect ratios
- Produce scores for each box using a convolution
- Also produce regressed coordinates using another convolution
Faster R-CNN

• $s$ scales $\times a$ aspect ratios = $sa$ anchor boxes
• Use convolutional layer on top of filter map to produce $sa$ scores
• Another convolution to produce $4sa$ bounding box offsets
• 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
SSD (Single Shot Detector)

- Why go through separate proposals?
- Directly produce class-specific scores at each location for every scale and aspect ratio
  - \( s \) scales * \( a \) aspects * \( c \) classes = \( sac \) scores per location

A comprehensive evaluation

Speed and accuracy trade-offs for modern convolutional object detectors
Alireza Fathi, Anoop Korattikara, Chen Sun, Ian Fischer, Jonathan Huang, Kevin Murphy, Menglong Zhu, Sergio Guadarrama, Vivek Rathod, Yang Song, Zbigniew Wojna
CVPR 2017
Detecting small objects

- Small objects get low resolution features
Feature pyramid networks

Feature pyramid networks

<table>
<thead>
<tr>
<th>Faster R-CNN</th>
<th>proposals</th>
<th>feature</th>
<th>head</th>
<th>lateral?</th>
<th>top-down?</th>
<th>AP@0.5</th>
<th>AP</th>
<th>AP_s</th>
<th>AP_m</th>
<th>AP_l</th>
</tr>
</thead>
<tbody>
<tr>
<td>(*) baseline from He et al. [16]†</td>
<td>RPN, $C_4$</td>
<td>$C_4$</td>
<td>conv5</td>
<td></td>
<td></td>
<td>47.3</td>
<td>26.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) baseline on conv4</td>
<td>RPN, $C_4$</td>
<td>$C_4$</td>
<td>conv5</td>
<td></td>
<td></td>
<td>53.1</td>
<td>31.6</td>
<td>13.2</td>
<td>35.6</td>
<td>47.1</td>
</tr>
<tr>
<td>(b) baseline on conv5</td>
<td>RPN, $C_5$</td>
<td>$C_5$</td>
<td>2fc</td>
<td></td>
<td></td>
<td>51.7</td>
<td>28.0</td>
<td>9.6</td>
<td>31.9</td>
<td>43.1</td>
</tr>
<tr>
<td>(c) FPN</td>
<td>RPN, ${P_k}$</td>
<td>${P_k}$</td>
<td>2fc</td>
<td>√</td>
<td>√</td>
<td>56.9</td>
<td>33.9</td>
<td>17.8</td>
<td>37.7</td>
<td>45.8</td>
</tr>
</tbody>
</table>

Other details - Non-max suppression
Non-max suppression

• Might find the same object with different sized-boxes and different scales
• But must fire exactly once on each object
• Idea: if two detections overlap significantly (>50% IoU), drop lower scoring one
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
Other details - ROI Align

• Snapping box to grid introduces quantization artifacts
• Instead, use bilinear interpolation