Lecture 12: Object detection

Contains slides from S. Lazebnik, R. Girshick, B. Hariharan, P. Isola
Announcements

- PS1 grades out
- Next two problem sets:
  - PS6: image generation
  - PS7: representation learning
Before we talk about objects:
What does it mean to understand a scene?
A view of a park on a nice spring day

Source: Antonio Torralba
PEOPLE UNDER THE SHADOW OF THE TREES

DUCKS ON TOP OF THE GRASS
Do not feed the ducks sign

PEOPLE WALKING IN THE PARK

PERSON FEEDING DUCKS IN THE PARK

DUCKS LOOKING FOR FOOD

Source: Antonio Torralba
What makes this challenging?
Why do we care about recognition?

We can perceive the 3D shape, texture, material properties, without knowing about objects. But, the concept of category encapsulates also information about what can we do with those objects.

Source: Torralba, Freeman, Isola
Object categories aren’t everything
Object categories aren’t everything

A picture is worth a 1000 words… Or just 10?

Source: A. Efros
What labels? Recognizing exact instances?

A Beijing City Transit Bus #17, serial number 43253?
Need more general (useful) information

What can we say the very first time we see this thing?

Functional:
• A large vehicle that may be moving fast, probably to the right, and will hurt you if you stand in its way.
• However, at specified places, it will allow you to enter it and transport you quickly over large distances.

Communicational:
• bus, autobus, λεωφορείο, ônibus, автобус, 公共汽车, etc.
Visual challenges with categories

- A lot of categories are functional
- Categories are 3D, but images are 2D
- World is highly varied

Source: A. Efros
Limits to direct perception
Importance of context

Source: Antonio Torralba
Today

- Introduction to scene understanding
- Object detection models
- Evaluating object detectors
- Future challenges
Today

• Introduction to scene understanding
• Object detection models
• Evaluating object detectors
• Future challenges
Previously: object recognition

Source: Torralba, Freeman, Isola
Previously: semantic segmentation

“\text{A bunch of bird stuff}”

Source: Torralba, Freeman, Isola
Object detection

Challenge: unbounded number of detections, possibly multiple detections per pixel

Source: Torralba, Freeman, Isola
Idea #1: regress bounding box
Idea #1: regress bounding box

**Outputs**

1. **Class label**
   - Ground truth: dog
   - Prediction: cat

2. **Box coords.**
   - (x, y, w, h)

**Losses**

1. **Cross entropy loss**
   \[ L_{cls} = -\log(p(y = \text{dog})) \]

2. **Squared distance**
   \[ L_{box} = \left\| \begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix} - \begin{bmatrix} x_{gt} \\ y_{gt} \\ w_{gt} \\ h_{gt} \end{bmatrix} \right\|^2 \]

Doesn’t scale well to multiple objects.
Idea #2: sliding window

Need multiple scales and aspect ratios
Idea #2: sliding window

*Bounding box $(x, y, w, h)$*
Example: histograms of oriented gradients (HOG)

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

Source: S. Lazebnik
Example: pedestrian detection with HOG

Train a pedestrian template using a linear classifier. Represent each window using HOG.

positive training examples

negative training examples

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

Source: S. Lazebnik
Pedestrian detection with HOG

For multi-scale detection, repeat over multiple levels of a HOG pyramid

HOG feature map  Template  Detector response map

Source: S. Lazebnik
Idea #3: selective search

- Problem: evaluating a detector is very expensive
- An image with $n$ pixels has $O(n^2)$ windows
- Only generate and evaluate a few hundred region proposals for regions that are “likely” to be an object of interest.

Source: S. Lazebnik
Selective search

- Example: edge boxes [Zitnick & Dollar, 2014]

- Heuristic: detect edges, group them into contours

- Rank each window based on number of contours in window

- These are the only windows our detector will see
Recall: idea #3: selective search

- Problem: evaluating a detector is very expensive
- An image with $n$ pixels has $O(n^2)$ windows
- Only generate and evaluate a few hundred region proposals for regions that are “likely” to be an object of interest.
Selective search

- Example: edge boxes [Zitnick & Dollar, 2014]
- Heuristic: detect edges, group them into contours
- Rank each window based on number of contours in window
- These are the only windows our detector will see
R-CNN: Region proposals + CNN features

Input image

ConvNet

Linear

ConvNet

Linear

ConvNet

Linear

Warped image regions

Classify regions with linear classifier

Forward each region through a CNN

Region proposals from **selective search** (~2K rectangles that are likely to contain objects)


Source: R. Girshick
R-CNN at test time

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

Source: R. Girshick
R-CNN at test time

Input image

Extract region proposals (~2k / image)

Compute CNN features

a. Crop
b. Scale

227 x 227

Source: R. Girshick
R-CNN at test time

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

1. Crop
2. Scale
3. Forward propagate
Output: “fc7” features

Source: R. Girshick
R-CNN at test time

Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions

Warped proposal

4096-dimensional fc7 feature vector

linear classifier

person? 1.6

horse? -0.3

Source: R. Girshick
Proposal refinement

Original proposal → Linear regression on CNN features → Predicted object bounding box

Bounding-box regression

Source: R. Girshick
Bounding-box regression

\[(x, y) \rightarrow (\Delta x \times w + x, \Delta y \times h + h)\]

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

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

Source: R. Girshick
Problems with R-CNN

1. Slow! Have to run CNN per window

2. Hand-crafted mechanism for region proposal might be suboptimal.
“Fast” R-CNN: reuse features between proposals

- ConvNet
  - Forward whole image through ConvNet
  - Conv5 feature map of image
  - RoI Pooling layer
  - Fully-connected layers
    - Linear
    - Linear + softmax
      - Softmax classifier
        - Bounding-box regressors
          - Region proposals

Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015
ROI Pooling

• How do we crop from a feature map?

• Step 1: Resize boxes to account for subsampling
ROI Pooling

• How do we crop from a feature map?

• Step 2: Snap to feature map grid

Source: B. Hariharan
• How do we crop from a feature map?

• Step 3: Overlay a new grid of fixed size

Source: B. Hariharan
ROI Pooling

• How do we crop from a feature map?
• Step 4: Take max in each cell
• Can improve with bilinear sampling

See more here: https://deepsense.ai/region-of-interest-pooling-explained/

Source: B. Hariharan
“Faster” R-CNN: learn region proposals

RPN: Region Proposal Network

\[ f_i = FCN(I) \]
RPN: Region Proposal Network

$ f_i = \text{FCN}(I) $ 

3x3 “sliding window” 
Scans the feature map looking for objects
RPN: Anchor Box

Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*.

$f_i = FCN(I)$

3x3 “sliding window” Scans the feature map looking for objects.

Conv feature map

Source: R. Girshick
RPN: Anchor Box

- Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*

- 3x3 “sliding window”
  - Objectness classifier [0, 1]
  - Box regressor predicting (dx, dy, dh, dw)

Source: R. Girshick
RPN: Prediction (on object)

Objectness score

\[ P(\text{object}) = 0.94 \]

3x3 “sliding window”

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

Source: R. Girshick
RPN: Prediction (on object)

- Anchor box: transformed by box regressor
- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

P(object) = 0.94

3x3 “sliding window”

Source: R. Girshick
RPN: Prediction **(off object)**

- **Objectness score**
  - 3x3 “sliding window”
  - Objectness classifier
  - Box regressor predicting \((dx, dy, dh, dw)\)

**Anchor box:** transformed by box regressor

\[ P(\text{object}) = 0.02 \]

Source: R. Girshick
RPN: Multiple Anchors

- 3x3 “sliding window”
  - $K$ objectness classifiers
  - $K$ box regressors

Anchor boxes: $K$ anchors per location with different scales and aspect ratios

Conv feature map

Source: R. Girshick
One network, four losses

Classification loss
Bounding-box regression loss

Region Proposal Network
RoI pooling

proposals
feature map

CNN

Source: R. Girshick, K. He, S. Lazebnik
Faster R-CNN results

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

<table>
<thead>
<tr>
<th>system</th>
<th>time</th>
<th>07 data</th>
<th>07+12 data</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-CNN</td>
<td>~50s</td>
<td>66.0</td>
<td>-</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>~2s</td>
<td>66.9</td>
<td>70.0</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>198ms</td>
<td>69.9</td>
<td>73.2</td>
</tr>
</tbody>
</table>

Source: S. Lazebnik
How do we deal with scale?

Idea #1: Gaussian pyramid

Source: Torralba, Freeman, Isola

[Lin et al., “Feature Pyramid Networks for Object Detection”, 2017]
Each pooling reduces the resolution by a factor of 2.

VGG network

CNN architectures build:
- Multiscale feature hierarchies, but
- each layer builds a different representation
- first layers are low level, while
- last layers are high level.

A feature pyramid requires a uniform representations across scales.

Source: Torralba, Freeman, Isola
Idea #2: Feature pyramid network

Image pyramid

Encoder-decoder architecture (U-Net)

Feature pyramid

Source: Torralba, Freeman, Isola
Object detection progress

Performance on PASCAL VOC

Source: S. Lazebnik
Streamlined detection architectures

- The Faster R-CNN pipeline separates proposal generation and region classification:
  - Conv feature map of the entire image
  - Region Proposals
  - RoI features
  - Classification + Regression
  - Detections

- Is it possible to do detection in one shot?
Single-stage object detector

- Divide the image into a coarse grid using a fully convolutional net

- Directly predict class label, confidence, and a few candidate boxes for each grid cell.

Source: S. Lazebnik
1. Take convolutional feature maps at 7x7 resolution

2. Predict, at each location, a score for each class and 2 bounding boxes (w/ confidence)
   - E.g. for 20 classes, output is 7x7x30 (30 = 20 + 2*(4+1))
   - 7x speedup over Faster R-CNN (45-155 FPS vs. 7-18 FPS) but less accurate (e.g. 65% vs. 72 mAP%)
   - Extension: use anchor boxes in last layer to try a few possible aspect ratios

YOLO detector

Source: S. Lazebnik

Today

• Introduction to scene understanding
• Object detection models
• Evaluating object detectors
• Future challenges
Evaluating an object detector

- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a true or false positive

**Intersection over union (IoU):** \( \frac{\text{Area}(\text{GT} \cap \text{Det})}{\text{Area}(\text{GT} \cup \text{Det})} > 0.5 \)

Source: S. Lazebnik
Evaluating an object detector

Intersection over union (also known as Jaccard similarity)

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Source: B. Hariharan
Evaluating an object detector

- For each class, plot Precision-Recall curve and compute Average Precision (area under the curve)
- Take mean of AP over classes to get mAP

**Precision:**
true positive detections / total detections

**Recall:**
true positive detections / total positive test instances

Source: S. Lazebnik
Average precision

true pos. / total detections

true positive detections / total positive test instances
Average precision

Precision

Recall

true pos. /
total detections

true positive detections /
total positive test instances

Source: B. Hariharan
Non-maximum suppression

- Subtlety: we predict a bounding box for every sliding window. Which ones should we keep?
- Keep only “peaks” in detector response.
- Discard low-prob boxes near high-prob ones
- Often use a simple greedy algorithm
Non-maximum suppression

Greedy algorithm, run on each class independently

let $A$ be the set of all bounding boxes
let $D$ be the set of detections we’ll keep, $D = \emptyset$

while $A \neq \emptyset$:
    remove $x$ the box with highest probability from $A$
    if $x$ doesn’t significantly overlap with an existing box in $D$ (e.g. IoU > 0.5):
        $D = D \cup \{x\}$

return $D$
Today

- Introduction to scene understanding
- Object detection models
- Evaluating object detectors
- **Future challenges**
Beyond bounding boxes: instance segmentation

Predict segmentation mask for each object
From COCO [Lin et al., 2014]

Source: B. Hariharan
Instance segmentation

Faster R-CNN

Extra "head" on network predicts binary mask

[He et al., "Mask R-CNN", 2017]
Example Mask Training Targets

Image with training proposal  28x28 mask target

Source: R. Girshick
Example Mask Training Targets

Image with training proposal

28x28 mask target

Source: R. Girshick
Example Mask Training Targets

Image with training proposal  28x28 mask target  Image with training proposal  28x28 mask target

Source: R. Girshick
Example Mask Training Targets

Image with training proposal 28x28 mask target Image with training proposal 28x28 mask target

Image with training proposal 28x28 mask target Image with training proposal 28x28 mask target
➢ Add keypoint head (28x28x17)

➢ Predict one “mask” for each keypoint

➢ Softmax over spatial locations (encodes one keypoint per mask “prior”)
We still need *lots* of labeled examples.
Handle the long tail of the distribution

Person, dog, table, …

Teacup, wreath, birdfeeder, …
Handle the “long tail” of the distribution

From COCO (80 categories) [Lin et al., 2014]

LVIS dataset (1000+ categories) “Few shot” (e.g. < 20 examples) [Gupta et al., 2019]
Next time: video