CS4670/5670: Computer Vision
Kavita Bala

Lecture 41: Recognition Beyond Classification

Thanks to Andrej Karpathy
Announcements

• Office hours till next Friday
• Drop me a note if you want to meet
Announcements

• Exam review will be announced
• Exam topics
  – Lec 18 (Cameras, Mar 9)-lec 39 (ConvNets, May 4)
• Grading: any unresolved issues drop me a note
• Final: Sunday 2pm, May 22
  – BTN100WEST, Barton Hall 100 West-Main Floor
The PASCAL Visual Object Classes Challenge 2009 (VOC2009)

Three (+2) challenges:
– Classification challenge (is there an X in this image?)
– Detection challenge (draw a box around every X)
– Segmentation challenge (which class is each pixel?)

Slides from Noah Snavely
Classification: C classes
Input: Image
Output: Class label
Evaluation metric: Accuracy

Localization:
Input: Image
Output: Box in the image (x, y, w, h)
Evaluation metric: Intersection over Union

Classification + Localization: Do both
Remember from Lec32

- Area of Overlap (AO) Measure

\[
AO(B_{gt}, B_p) = \frac{|B_{gt} \cap B_p|}{|B_{gt} \cup B_p|}
\]

- Need to define a threshold \( t \) such that \( AO(B_{gt}, B_p) \) implies a correct detection: 50%
Classification+Localization: ImageNet

1000 classes (same as classification)

Each image has 1 class, at least one bounding box

~800 training images per class

Algorithm produces 5 (class, box) guesses

Example is correct if at least one guess has correct class AND bounding box at least 0.5 intersection over union (IoU)

Krizhevsky et. al. 2012
Localization for one object

Idea #1: Localization as Regression

Input: image

Only one object, simpler than detection

Neural Net

Output:
Box coordinates (4 numbers)

Correct output:
box coordinates (4 numbers)

Loss:
L2 distance
Classification+Localization for one object

**Step 1:** Train (or download) a classification model (AlexNet, VGG, GoogLeNet)

**Step 2:** Attach new fully-connected “regression head” to the network

**Step 3:** Train the regression head only with SGD and L2 loss

**Step 4:** At test time use both heads

Localization as regression: simple but powerful
Idea #2: Sliding Window

- Run classification + regression network at multiple locations on a high-resolution image

- Convert fully-connected layers into convolutional layers for efficient computation

- Combine classifier and regressor predictions across all scales for final prediction
Network input: 3 x 221 x 221

Larger image: 3 x 257 x 257

Classification scores:
P(cat)

0.5

0.5

0.75

0.6

Classification scores:
P(cat)
Greedily merge boxes and scores (details in paper)

Network input: 3 x 221 x 221

Larger image: 3 x 257 x 257

Classification scores:
P(cat)

0.5 0.75
0.6 0.8

Classification score: P

0.8
Overfeat

In practice use many sliding window locations and multiple scales
Localization Error

**Localization Error (Top 5)**

- **AlexNet**: Localization method not published
- **Overfeat**: Multiscale convolutional regression with box merging
- **VGG**: Same as Overfeat, but fewer scales and locations; simpler method, gains all due to deeper features
- **ResNet**: Different localization method (RPN) and much deeper features
Object Detection

• Need to test many scales and positions
• Solution: only look at a small set of possible positions

• Approach: propose regions, then regressors
RCNN


Slide credit: Ross Girshick
Fast R-CNN

Fast R-CNN (test time)

Regions of Interest (RoIs) from a proposal method

Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
Fully-connected layers
“RoI Pooling” (single-level SPP) layer
“conv5” feature map of image
Forward whole image through ConvNet

ConvNet
Input image

Slide credit: Ross Girshick
<table>
<thead>
<tr>
<th></th>
<th>R-CNN</th>
<th>Fast R-CNN</th>
<th>Faster R-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test time per image (with proposals)</td>
<td>50 seconds</td>
<td>2 seconds</td>
<td>0.2 seconds</td>
</tr>
<tr>
<td>(Speedup)</td>
<td>1x</td>
<td>25x</td>
<td>250x</td>
</tr>
<tr>
<td>mAP (VOC 2007)</td>
<td>66.0</td>
<td><strong>66.9</strong></td>
<td><strong>66.9</strong></td>
</tr>
</tbody>
</table>
ImageNet Detection (mAP)

- NeoNet ensemble (2015): 58.85
- Faster R-CNN ensemble (2015): 53.57
- GoogleNet ensemble (2014): 42.94
- NUS ensemble (2014): 43.93
- SPP ensemble (2014): 37.21
- UvA-Euvison (2013): 35.11
- Overfeat (2013): 22.58
- Overfeat (2013): 19.4
Semantic Segmentation

Classification

Classification + Localization

Object Detection

Instance Segmentation

Single object

Multiple objects
Deep learning for materials

- Train at different scales
Deep learning to predict materials
Material Predictions
Semantic segmentation

CRF Runtime: ~1s for 640x480 image

\[ E(x|I, \theta) = \sum_i \psi_i(x_i|\theta) + \sum_{i<j} \psi_{ij}(x_i, x_j|\theta) \]
Semantic segmentation

Mean class accuracy: 84.95% out of 20 categories
Prior work: 41% out of 10 categories
Results
Summary

• Localization
  – Find fixed number of objects
  – L2 regression from CNN features to box coordinates

• Detection
  – Find variable number of objects
  – Sliding window, too dense
  – Use region proposals: R-CNN and variants

• Segmentation
  – Couple with dense CRF formulations for boundaries
Other Innovations

• Recurrent Neural Nets (RNN)
  – Memory

• Residual Nets (ResNet)
  – Deep

• ...

Data Sets

• Critical to the success of deep learning
  – Object classification and segmentation
  – Scene classification
  – Materials

• Examples
  – PASCAL VOC
    • Not Crowdsourced, bounding boxes, 20 categories
  – ImageNet
    • Huge, Crowdsourced, Hierarchical, Iconic objects
  – SUN Scene Database
    • Not Crowdsourced, 397 (or 720) scene categories
  – Microsoft COCO
    • Crowdsourced, large
  – Material Database: OpenSurfaces
PASCAL VOC 2005-2012

20 object classes

Classification: person, motorcycle

Detection

Action: riding bicycle

22,591 images

Segmentation

Large Scale Visual Recognition Challenge (ILSVRC) 2010-2012

20 object classes 22,591 images
1000 object classes 1,431,167 images

How do we classify scenes?

Different objects, different spatial layout
Which are the important elements?

<table>
<thead>
<tr>
<th>cabinets</th>
<th>ceiling</th>
<th>cabinets</th>
</tr>
</thead>
<tbody>
<tr>
<td>window</td>
<td>seat</td>
<td>seat</td>
</tr>
<tr>
<td>seat</td>
<td>seat</td>
<td>seat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cabinets</th>
<th>ceiling</th>
<th>cabinets</th>
</tr>
</thead>
<tbody>
<tr>
<td>window</td>
<td>seat</td>
<td>seat</td>
</tr>
<tr>
<td>seat</td>
<td>seat</td>
<td>seat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cabinets</th>
<th>ceiling</th>
<th>cabinets</th>
</tr>
</thead>
<tbody>
<tr>
<td>window</td>
<td>seat</td>
<td>seat</td>
</tr>
<tr>
<td>seat</td>
<td>seat</td>
<td>seat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cabinet</th>
<th>ceiling</th>
</tr>
</thead>
<tbody>
<tr>
<td>wall</td>
<td>column</td>
</tr>
<tr>
<td>screen</td>
<td>seat</td>
</tr>
<tr>
<td>seat</td>
<td>seat</td>
</tr>
</tbody>
</table>

Similar objects, and similar spatial layout

Different lighting, different materials, different “stuff”
397 Well-sampled Categories
• Places Database
• ~7 million images from 476 scene categories
ImageNet-CNN and Places-CNN

- Same structure as AlexNet, but trained on different databases.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>SUN397</th>
<th>MIT Indoor67</th>
<th>Scene15</th>
<th>SUN Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Places-CNN feature</td>
<td>54.32±0.14</td>
<td>68.24</td>
<td>90.19±0.34</td>
<td>91.29</td>
</tr>
<tr>
<td>ImageNet-CNN feature</td>
<td>42.61±0.16</td>
<td>56.79</td>
<td>84.23±0.37</td>
<td>89.85</td>
</tr>
<tr>
<td>Caltech101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Places-CNN feature</td>
<td>65.18±0.88</td>
<td>45.59±0.31</td>
<td>42.86±0.25</td>
<td>94.12±0.99</td>
</tr>
<tr>
<td>ImageNet-CNN feature</td>
<td>87.22±0.92</td>
<td>67.23±0.27</td>
<td>54.92±0.33</td>
<td>94.42±0.76</td>
</tr>
<tr>
<td>Caltech256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Places-CNN feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageNet-CNN feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.