Deep Object Detection

Ali Farhadi
Mohammad Rastegari
CSE 576
So Far

• Backpropagation

• Convolutional Neural Networks (CNN)

• AlexNet

• Deeper Architectures
Deep Learning Practical Tips

• Use off-the-shelf architectures

• Verify the correctness of your network by training over a single batch.
  – Overfit : Good to go!
  – Did not converge : Something is wrong with forward/backward functions or data!

• Use a proper learning rate regime.
Object Detection
Sliding Window
Sliding Window
Sliding Window
Object Proposal
Selective Search


Object hypotheses

Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at the same scale, whereas on the right we necessarily find the objects at different scales. On the right we necessarily find the objects at different scales. On the left we find many objects at the same scale, whereas on the right we necessarily find the objects at different scales.
Region-Based CNN (R-CNN)

Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions (linear SVM)

Grishick et al [CVPR’14]
Object Detection by R-CNN
Object Detection by R-CNN
Object Detection by R-CNN

- Depending on region proposal
- Need to apply CNN ~2K times per image
Fast R-CNN

ROI Pooling

Reshape
Fast R-CNN
Bounding Box Regression
Bounding Box Regression
Faster R-CNN

Less number of proposals compared to Selective Search
300 vs. 2000
<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>Sec/im</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-CNN</td>
<td>59.2</td>
<td>20</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>68.4</td>
<td>2</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>72.1</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Direct Regression
No Proposal

We do not know the number of objects in an image
Dog

"You Only Look Once"
Real-Time Detection
$S \times S$ grid on input

Bounding boxes + confidence

Class probability map

Final detections
YOLO

Reshape

x, y, w, h, c

c1, c2, ..., cN
<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>Sec/im</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-CNN</td>
<td>59.2</td>
<td>20</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>68.4</td>
<td>2</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>72.1</td>
<td>0.5</td>
</tr>
<tr>
<td>YOLO</td>
<td>57.9</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Source Code

• Fast R-CNN
  – https://github.com/mahyarnajibi/fast-rcnn-torch
  – https://github.com/rbgirshick/fast-rcnn

• YOLO
  – https://github.com/pjreddie/darknet/blob/master/src/yolo.c