CS6501: Deep Learning for Visual Recognition
Detection, Segmentation
Overview
Object Detection

deer

cat
Object Detection as Classification

CNN

deer?
cat?
background?
Object Detection as Classification

CNN

deer?
cat?
background?
Object Detection as Classification

- deer?
- cat?
- background?
Object Detection as Classification with Sliding Window
Object Detection as Classification with Box Proposals
Box Proposal Method – SS: Selective Search

Segmentation As Selective Search for Object Recognition. van de Sande et al. ICCV 2011
Fast-RCNN

Idea: No need to recompute features for every box independently, Regress refined bounding box coordinates.


Fast R-CNN. Girshick. ICCV 2015.

https://github.com/sunshineatnoon/Paper-Collection/blob/master/Fast-RCNN.md
Faster-RCNN

Idea: Integrate the Bounding Box Proposals as part of the CNN predictions

https://arxiv.org/abs/1506.01497
Ren et al. NIPS 2015.
Single-shot Object Detectors

• No two-steps of box proposals + Classification
• Anchor Points for predicting boxes
YOLO - You Only Look Once

Idea: No bounding box proposals.
Predict a class and a box for every location in a grid.

https://arxiv.org/abs/1506.02640

Redmon et al. CVPR 2016.
YOLO - You Only Look Once

Divide the image into 7x7 cells.
Each cell trains a detector.
The detector needs to predict the object’s class distributions.
The detector has 2 bounding-box predictors to predict bounding-boxes and confidence scores.

https://arxiv.org/abs/1506.02640

Redmon et al. CVPR 2016.
SSD: Single Shot Detector

Idea: Similar to YOLO, but denser grid map, multiscale grid maps. + Data augmentation + Hard negative mining + Other design choices in the network.

Liu et al. ECCV 2016.
Semantic Segmentation / Image Parsing

deer

cat

trees

grass
Idea 1: Convolutionalization

However resolution of the segmentation map is low.

https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf
Alexnet

https://www.saagie.com/fr/blog/object-detection-part1
Idea 1: Convolutionalization

\[
nn.Linear(n_{\text{inputs}}, n_{\text{outputs}}) \equiv nn.SpatialConvolution(n_{\text{inputs}}, n_{\text{outputs}}, 1, 1, 1, 1)
\]
Fully Convolutional Networks for Semantic Segmentation

Jonathan Long*  Evan Shelhamer*  Trevor Darrell
UC Berkeley

{jonlong, shelhamer, trevor}@cs.berkeley.edu
Idea 2: Up-sampling Convolutions or "Deconvolutions"

http://cvlab.postech.ac.kr/research/deconvnet/
Idea 2: Up-sampling Convolutions or "Deconvolutions"

https://github.com/vdumoulin/conv_arithmetic
Idea 2: Up-sampling Convolutions or ”Deconvolutions”

https://github.com/vdumoulin/conv_arithmetic
Idea 2: Up-sampling Convolutions or ”Deconvolutions”

- Deconvolutional Layers
- Upconvolutional Layers
- Backwards Strided Convolutional Layers
- Fractionally Strided Convolutional Layers
- Transposed Convolutional Layers
- Spatial Full Convolutional Layers
Idea 3: Dilated Convolutions

MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS

Fisher Yu
Princeton University

Vladlen Koltun
Intel Labs

ICLR 2016
Idea 3: Dilated Convolutions
Convolutional Layer in pytorch

```python
class torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True)  
```

**Input**
- `in_channels` (e.g. 3 for RGB inputs)

**Output**
- `out_channels` (equals the number of convolutional filters for this layer)
Questions?