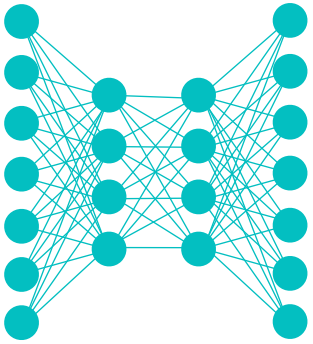
Lecture Notes for

Neural Networks and Machine Learning

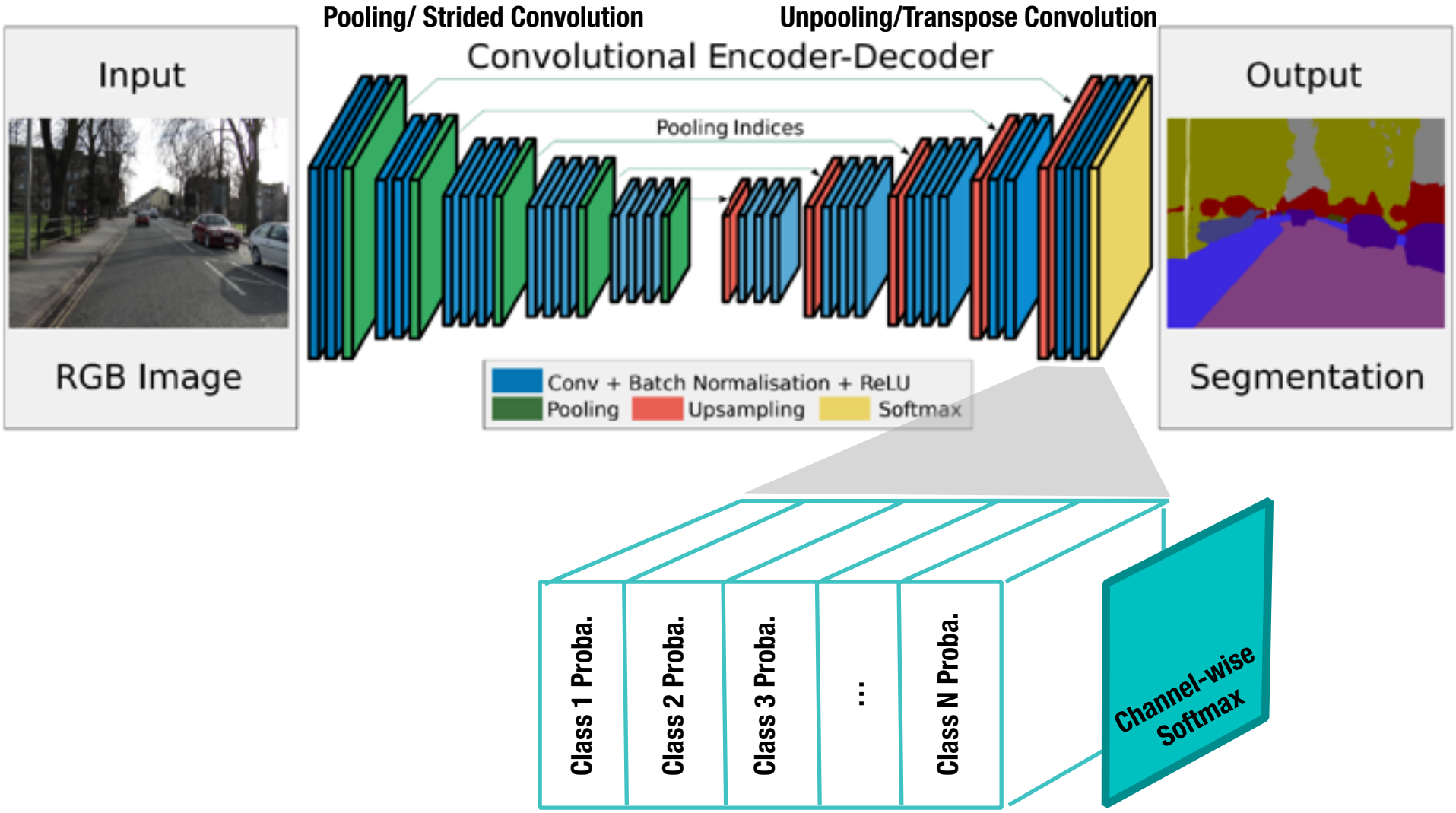
Fully Convolutional Learning II: Object Detection



Logistics and Agenda

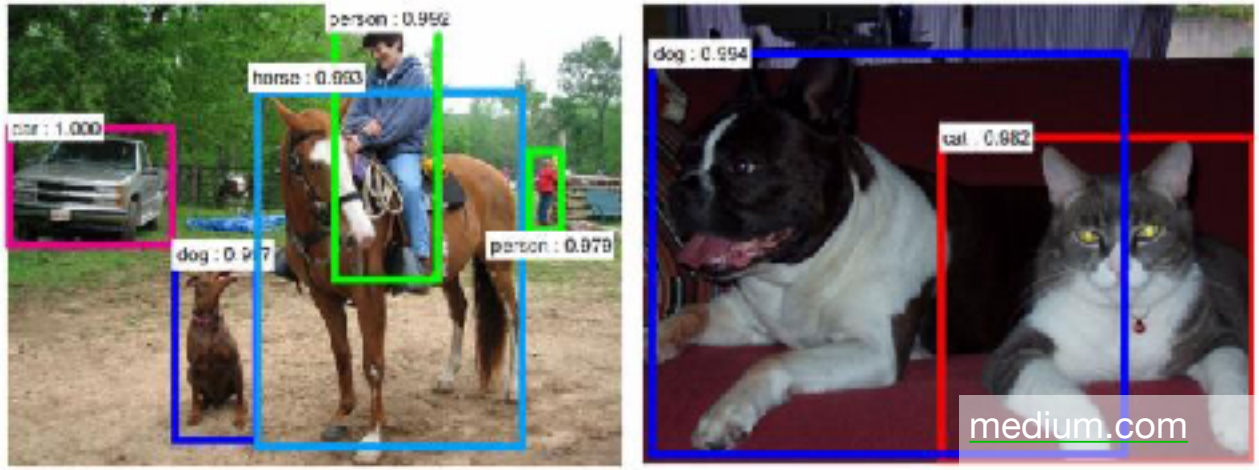
- Logistics
  - Lab due dates
  - Lab grading updates
- Agenda
  - Full Convolutional Architectures
    - Semantic Segmentation Basics (last time)
    - Object Detection (this time):
      - RCNN, YOLO
    - Instance Segmentation (next time, probably):
      - Mask-RCNN, YOLACT

Last Time


Pooling/ Strided Convolution
Unpooling/Transpose Convolution
Input
Convolutional Encoder-Decoder
Output
Pooling Indices
RGB Image
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax
Segmentation
Class 1 Proba.
Class 2 Proba.
Class 3 Proba.
...
Class N Proba.
Channel-wise Softmax

This time… Object Detection Methods

- Semantic segmentation has good mIoU values (up to 90%) but this is exaggerated by background recognition, many classes are <40%
- How to adapt these techniques to get bounding boxes, not semantic segmentations?
  - Could this be easier? More stable?
  - More consistent labeling?
  - Suitable for “higher risk” tracking applications?

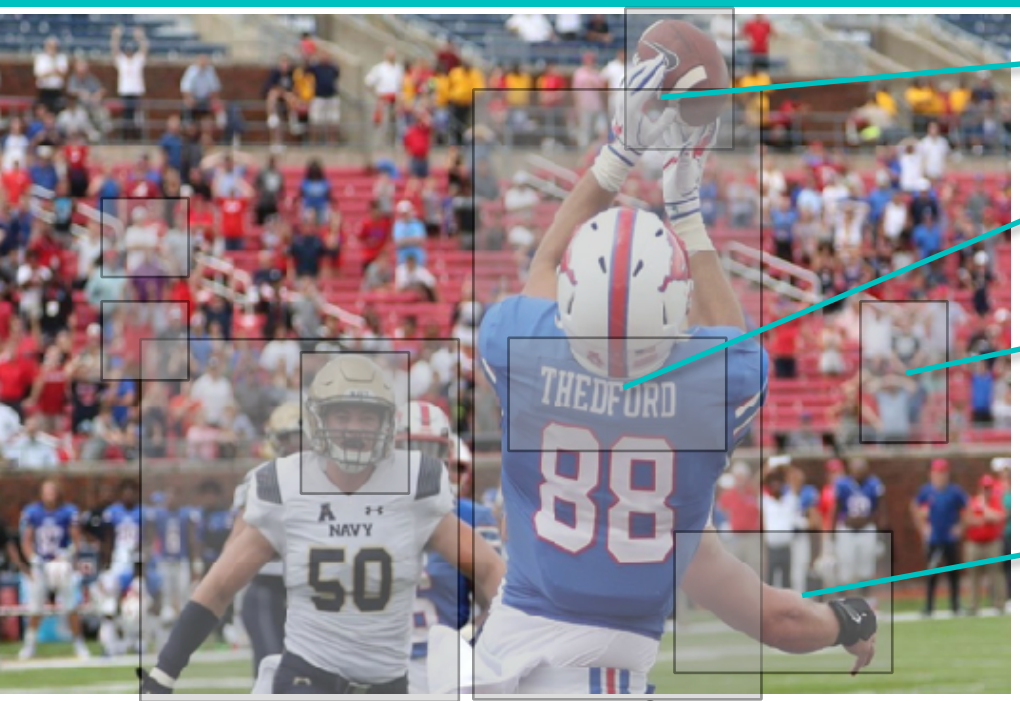


Object Detection with RCNN

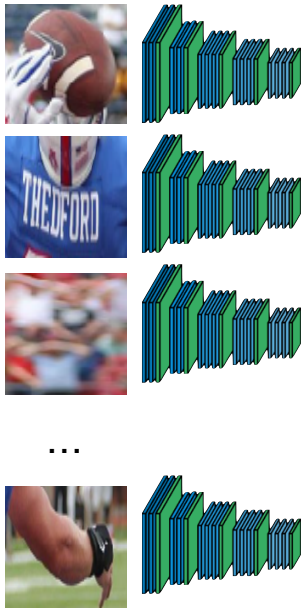
A history in naming one network five different times
with five different papers
each time changing one thing about the architecture

Research!

2014: R-CNN



Propose BBox ROIs from Image



Warp

Features

Classification (SVM)
BBox Offset (Linear Regression)

Classification (SVM)
BBox Offset (Linear Regression)

Classification (SVM)
BBox Offset (Linear Regression)

...

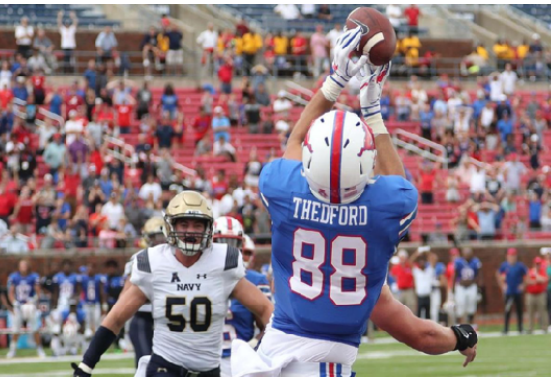
...

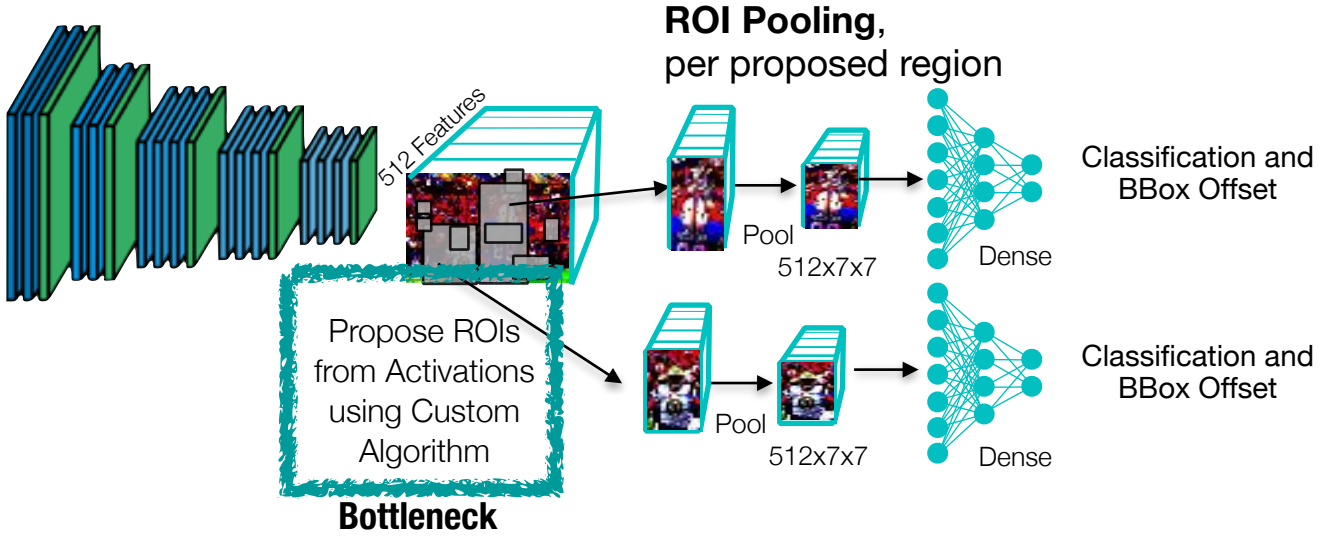
Classification (SVM)
BBox Offset (Linear Regression)

- Too Slow to Be Useful
- SVM and BBox Regression Trained Separately
- Fine Tuned Existing ConvNet (for Warped Images)
- ~50 Seconds per Image when Deployed

Girshick et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, 2014

2015: Fast R-CNN

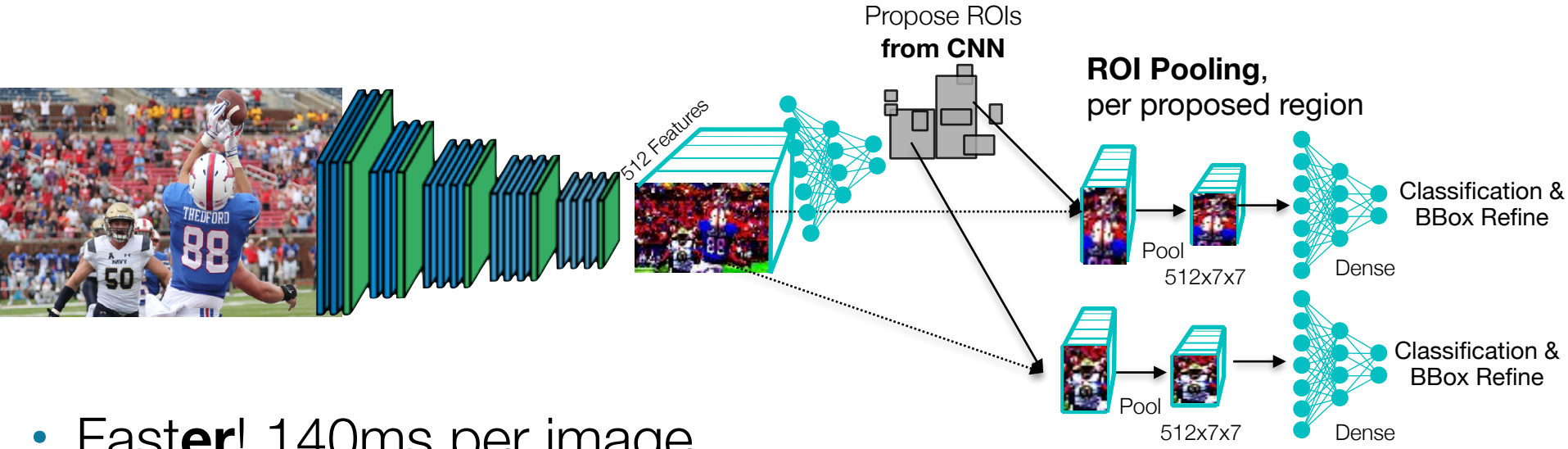




- Fast! 2.3 seconds per image (not ~50)
- But still not real time…

Girshick et al. Fast R-CNN, 2015, February

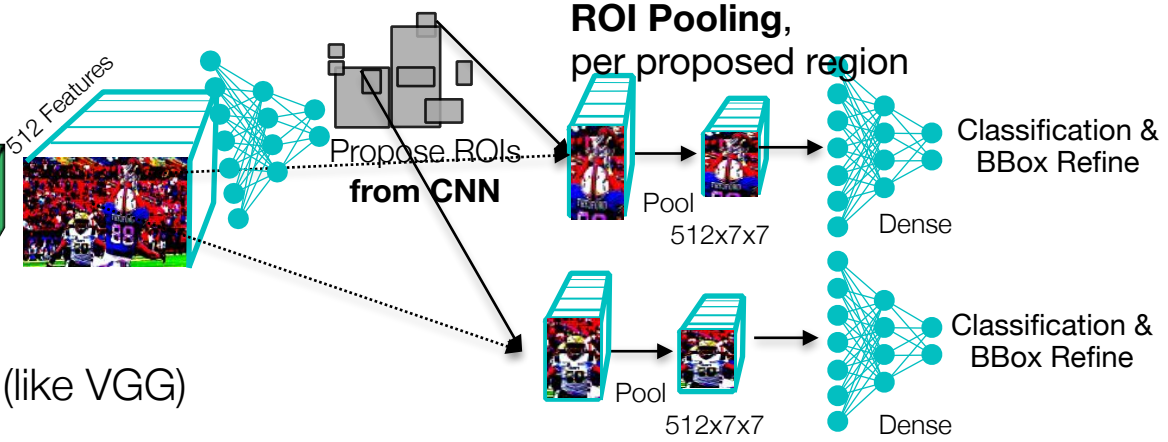
2015: Faster R-CNN



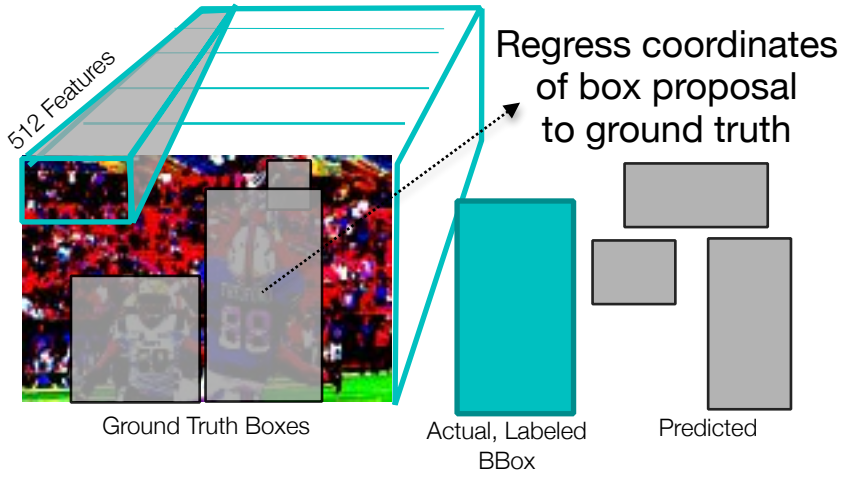
- Fast**er**! 140ms per image (7 FPS)
- Highly Accurate

Ren et al. Faster R-CNN, 2015, November

2015: Faster R-CNN, Training



- Pre-train image classifier backbone (like VGG)
- Train region proposal network
  - Sliding window (1x1 conv) across CNN activations from backbone classifier
  - Regress multiple bounding boxes (usually k proposals)
  - Regress "object-ness" of each box
- Train Fast R-CNN on generated ROI proposals
- Fix weights of classifier pipeline, fine tune RPN
- Fix RPN and fine tune classifier
- Rinse, repeat fine tuning



$$l_{box} = \sum_i \hat{p}_i \left[(x - \hat{x}_i)^2 + (y - \hat{y}_i)^2 + (\log w - \log \hat{w}_i)^2 + (\log h - \log \hat{h}_i)^2 \right]$$

$$l_{class} = \sum_c CE(c, \hat{c})$$

$$l_{obj} = \sum_i CE(p_i, \hat{p}_i)$$

Ren et al. Faster R-CNN, 2015, November

Lecture Notes for

Neural Networks and Machine Learning

FCN Learning: Detection

Next Time:
Instance Segmentation

Reading: None