CS 182 Lecture 8: Computer Vision 6/28

Standard CV probs: object classification, object localization (bounding box), object detection (every object), semantic segmentation (scene understanding)

Object localization: $D = \{(x_i, y_i)\}$ ← image → $(l_i, x_i, y_i, w_i, h_i)$ (bbox)

Measuring localization accuracy → matter of convention
Typical: IoU (intersection over union). Correct if IoU > 0.5 (threshold).
not a loss func! →

O.L as regression: multi-task model, cross-entropy loss for class (l_i) (two heads), regression loss for x_i, y_i, h_i, w_i
OR train classification THEN regression head

O.L using sliding window: classify every patch in image to form bounding box.
→ consider scaling w/ different aspect ratios, matching aspect ratio of object.

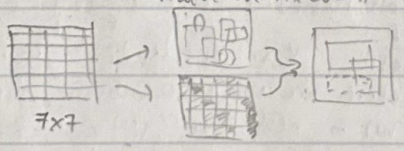
OverFeat! For every patch, output class probs AND bbox coords.
"Average" together boxes to get answer.

Save compute for sliding window thru "convolutional classification",
Reuse calculations across windows.
The intuition here is that everything is boiled down to convolutions.
The FC layers → 1×1 convolutions!

n_i objects, differs for each x_i.

Object Detection → $(x_1, C_{1,1}, x_{1,1}, y_{1,1}, w_{1,1}, h_{1,1}, \ldots, C_{1,n_1}, x_{1,n_1}, y_{1,n_1}, \ldots)$.
Non-maximal suppression: "kill" off any detections that have other high-scoring detections of the same class nearby.

2 Ideas for Model: - output test class at each window
- train on fixed # of boxes → YOLO! (You only look 49 times!)

YOLO: 

for each cell:
→ x, y, w, h → c → IoU] – B of these

implements "conv. classification"

essentially an overdusted overfeat. →

If too many objects, skip them. R-CNN? A "smarter" sliding window.
Incorporates proposal method init.
expensive!!!

Fast R-CNN: First forward image through conv.
Then process ROIs thru MaxPool + FC.
Now: ONE convolution. Can be trained end-to-end.
"R" = Region proposal. →

Use same model to train ROIs using bbox coords to be optimized.
Find places where objects are present.

Semantic segmentation: label every single pixel with its class.
Main issue: expensive, head-on.

Idea: High-res → low-res → high-res, much cheaper. "bottle-neck".
→ downsample → upsample
convolutions

Upsampling / Transpose convolution: incorporate fractional "stride" (ex. 1/2).
"Unpooling": remember the positions of max elements and place them back.

Problem: we may lose spatial information.

U-Net: implements skip connections.
concatenate down w/ up
(initial) (processed)
↓ theoretically contains spatial info