CS195: Computer Vision

Object Detection Implementation:

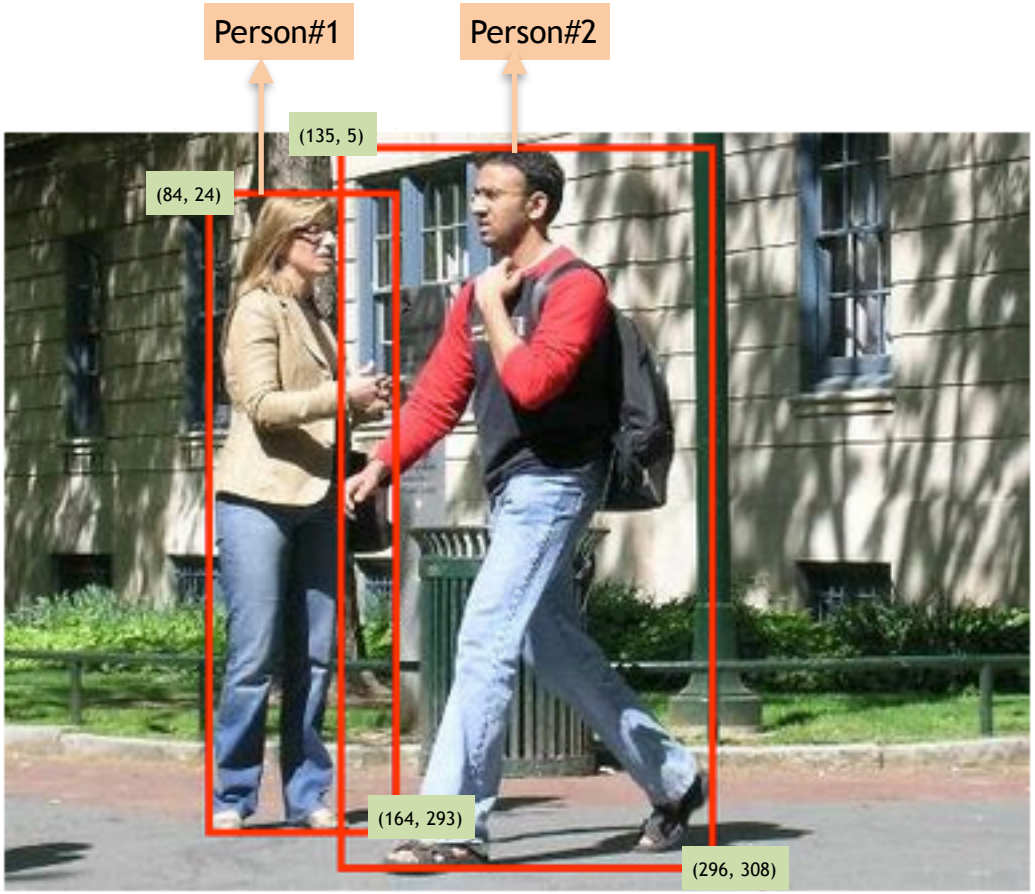
Inference with a pretrained Faster R-CNN

Fine-tuning a pretrained Faster R-CNN

Monday, October 7th, 2024

Drake UNIVERSITY

Object Detection: what are the locations of objects?



[xmin, ymin, xmax, ymax]

Label	Probs.	Bounding Box Coordinates
Person#1	0.99	[84, 24, 164, 293]
Person#2	0.99	[139, 5, 296, 308]
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…

Input:
1. an image

Detection model

Output:
1. Labels of prediction
2. Probabilities of labels
3. Locations of rectangular bounding boxes associated with each label

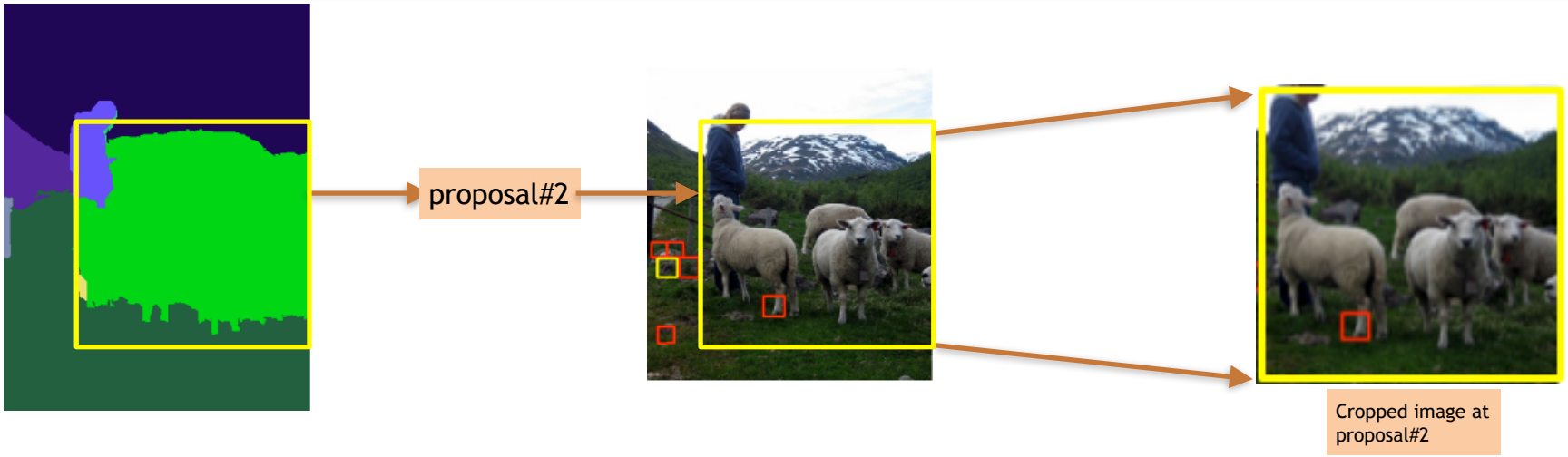
Deep learning-based object detections

- R-CNN
- Fast RCNN
- **Faster RCNN**
  - Implementation in PyTorch

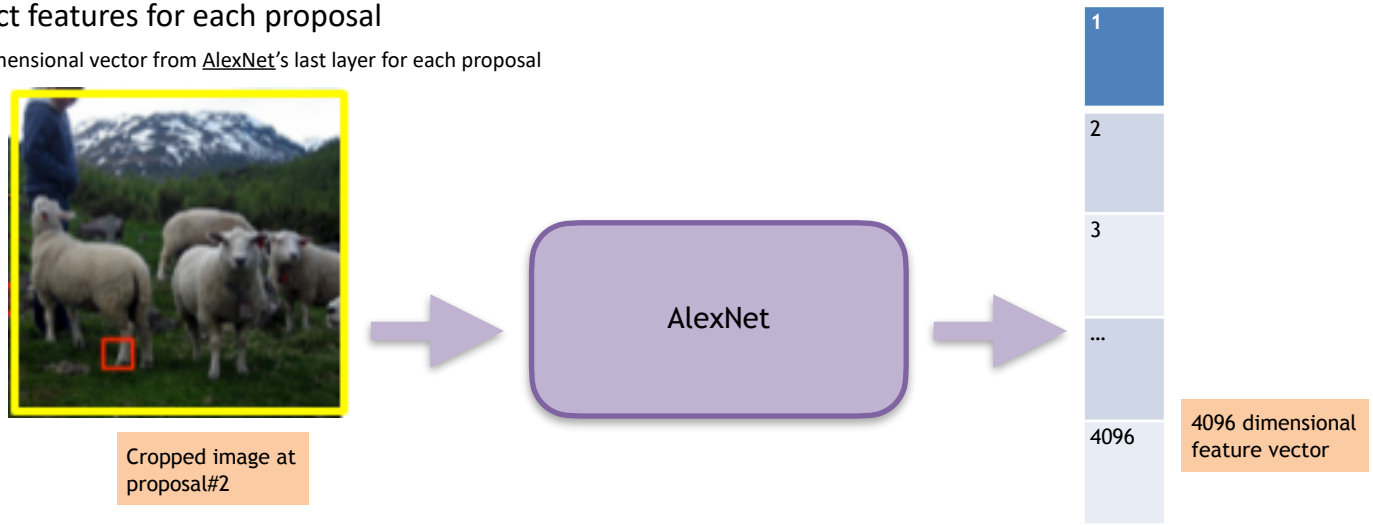
R-CNN

Rich feature hierarchies for accurate object detection - Girshick et al. CVPR'14

- Step 1: Generate proposal
  - 2000 proposals of various sizes



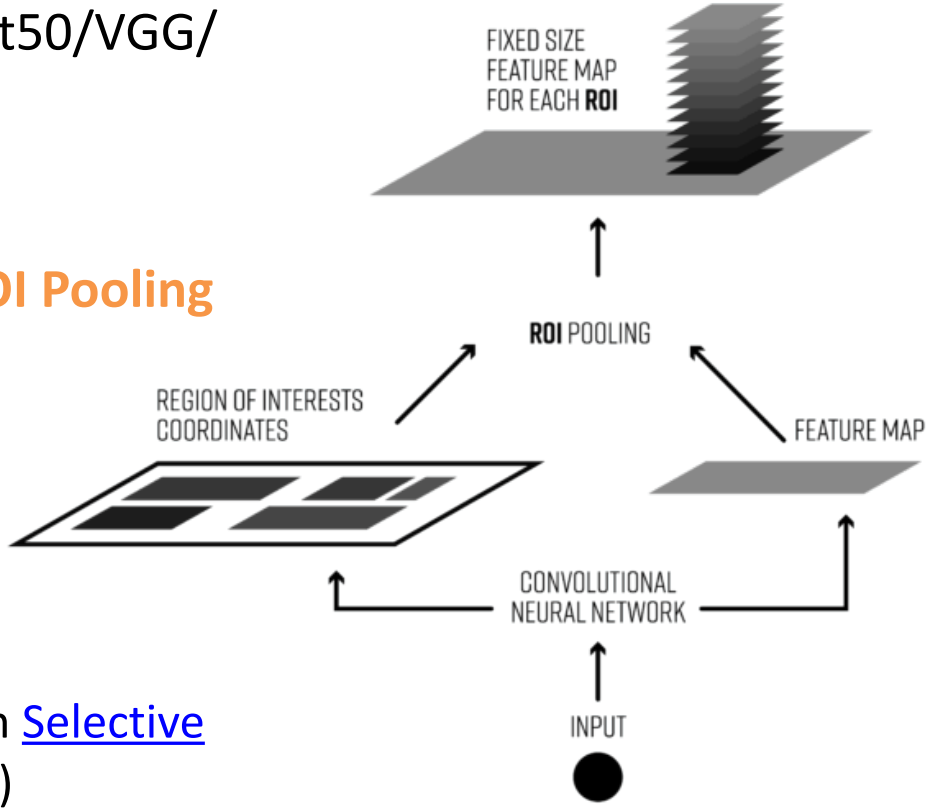
- Step 2: Extract features for each proposal
  - 4096 dimensional vector from AlexNet's last layer for each proposal



- Step 3: Classify each proposal it into one of the 1000 classes using SVM

Fast R-CNN

- Feed the image through a CNN (ResNet50/VGG/ AlexNet)
- Extract fixed-size feature map using **ROI Pooling**
  - for different Region Of Interests (ROI)
- End-to-end trainable:
  - except proposals generation part from Selective Search - IJCV'13 (R-CNN does that too)

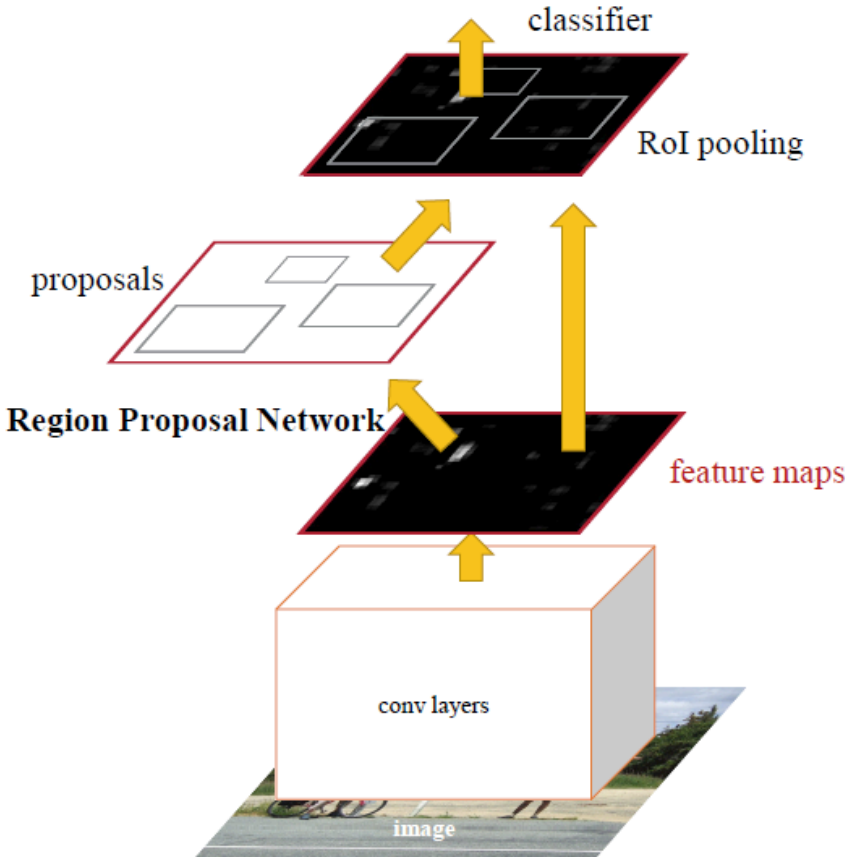


Fast R-CNN - Girshick et al. ICCV'15

Faster R-CNN

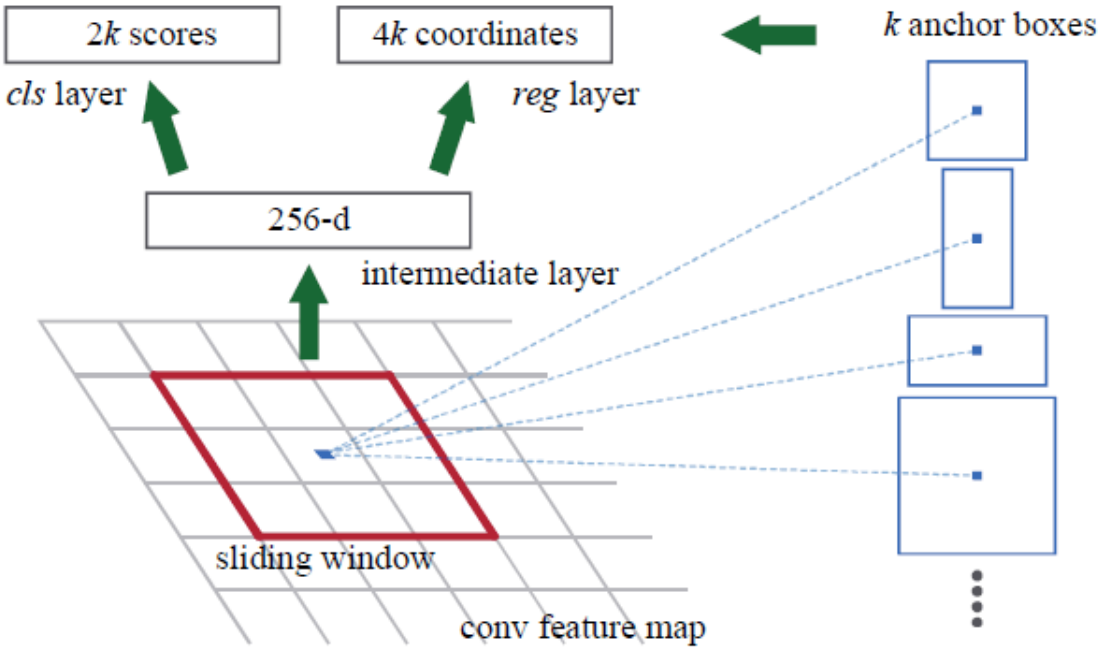
- Introduce a new network which will learn to generate good proposals. It is called **Region Proposal Network (RPN)** which will generate proposals.
- For each proposal, extract fixed-size feature map using **ROI Pooling**
- The network is end-to-end trainable:
  - Unlike R-CNN where different components are trained separately

[Faster R-CNN - Girshick et al. NIPS'15]



Faster R-CNN

- **Region Proposal Network (RPN)**



Faster R-CNN - Girshick et al. NIPS'15

PyTorch's Detectors

- The summary of the performances for the models trained on the instances set of Microsoft COCO train2017 and evaluated on Microsoft COCO val2017:

Network	box AP
Faster R-CNN ResNet-50 FPN	37.0
Faster R-CNN MobileNetV3-Large FPN	32.8
Faster R-CNN MobileNetV3-Large 320 FPN	22.8
RetinaNet ResNet-50 FPN	36.4
Mask R-CNN ResNet-50 FPN	37.9

Coding Activity: Faster R-CNN

- Let's take a pretrained Faster R-CNN model which was trained on Microsoft COCO (Common Object in COntext) Dataset with 91 categories
  - First, let's run the inference on an arbitrary dataset eg, Pedestrian Dataset
  - Then, fine-tune the pretrained model using the Pedestrian Dataset