

CS181: Computer Vision

Object Detection Implementation:

Inference with a pretrained Faster R-CNN

Fine-tuning a pretrained Faster R-CNN

Tuesday, October 6th, 2026



Md Alimoor Reza

Associate Professor of Computer Science

Announcements

- Quiz#1

- you should be working on it. Take it offline on Blackboard
- should be taken [within a fixed-hour window](#) (10 hours) during which you are expected to complete the quiz individually. You are free to start anytime within the seven-day period (released on [September 29th](#) and [due by October 7th](#))

What type of recognition problem?



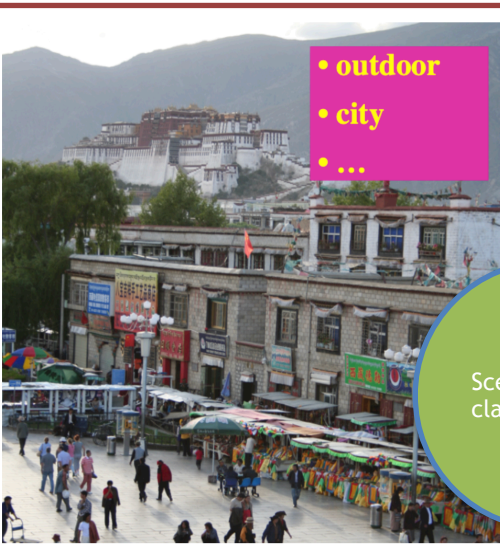
Object Recognition



What type of recognition problem?



What type of recognition problem?



What type of recognition problem?

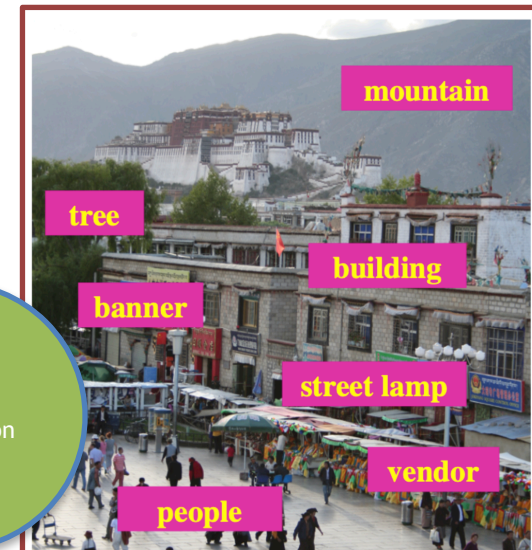


Image Classification: what are the objects in this image?



<u>Label</u>		<u>Probability</u>
Person	+	0.99
Building		0.01
Cat		0.00
Dog		0.01
...		...
...		...
...		...
...		...

Input:

1. an image



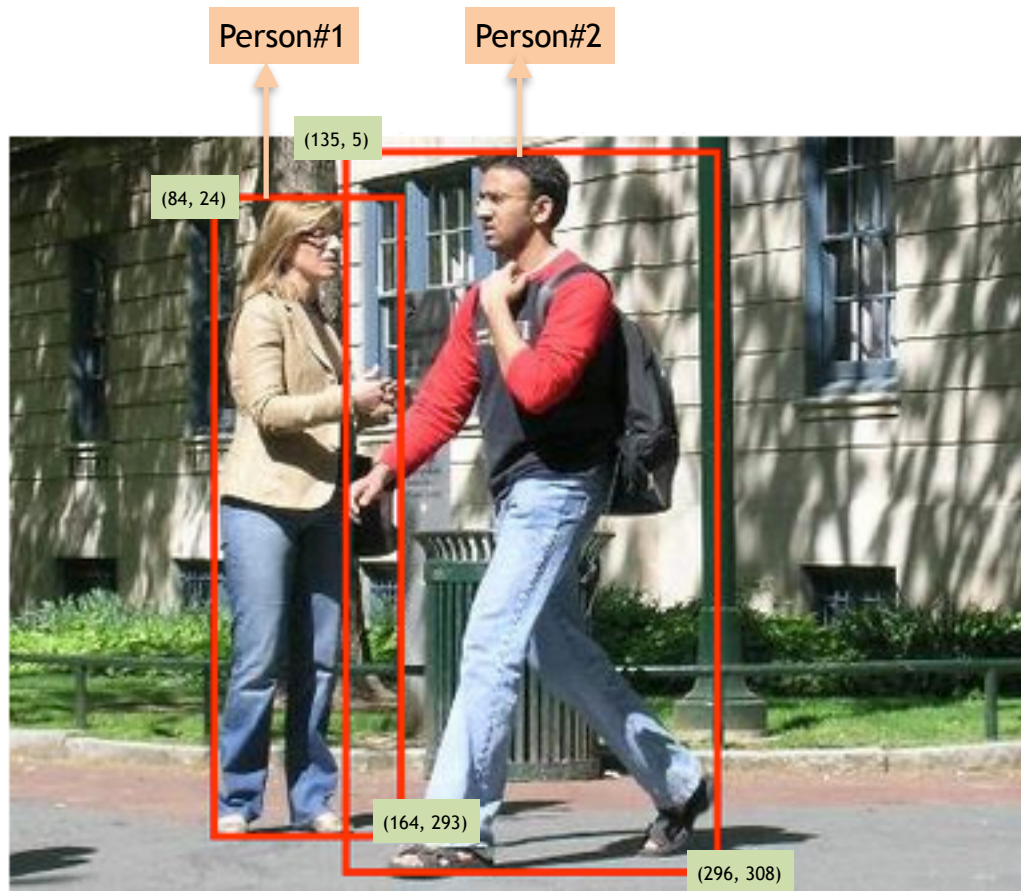
Classification model



Output:

1. Labels of prediction
2. Probability of labels

Object Detection: what are the locations of objects?



[xmin, ymin, xmax, ymax]

<u>Label</u>	<u>Probs.</u>	<u>Bounding Box Coordinates</u>
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

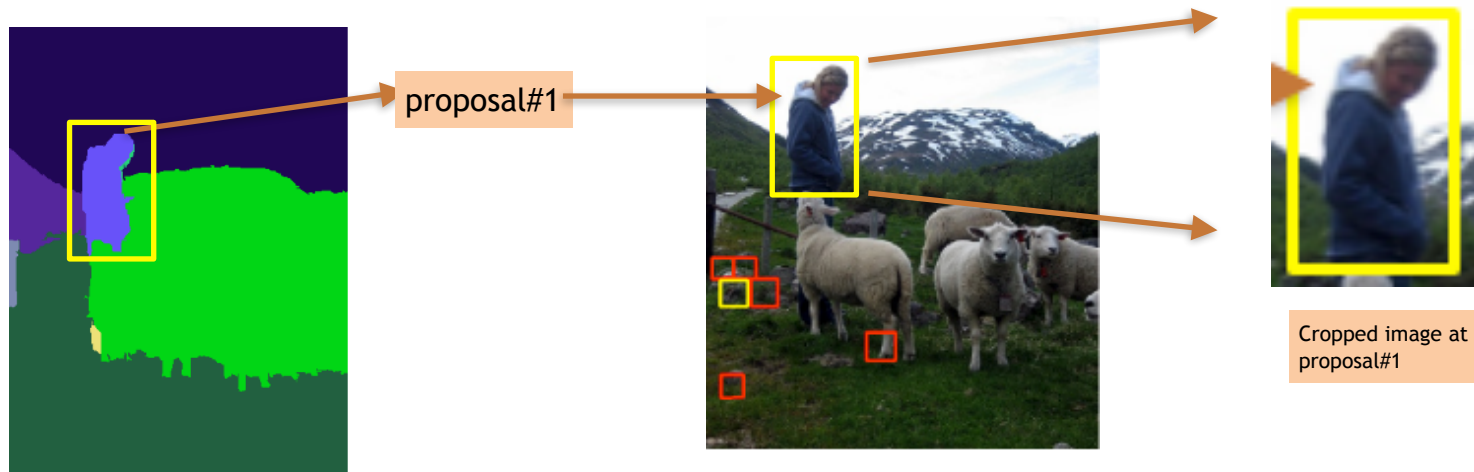
- Mask RCNN

- YOLO
- SSD

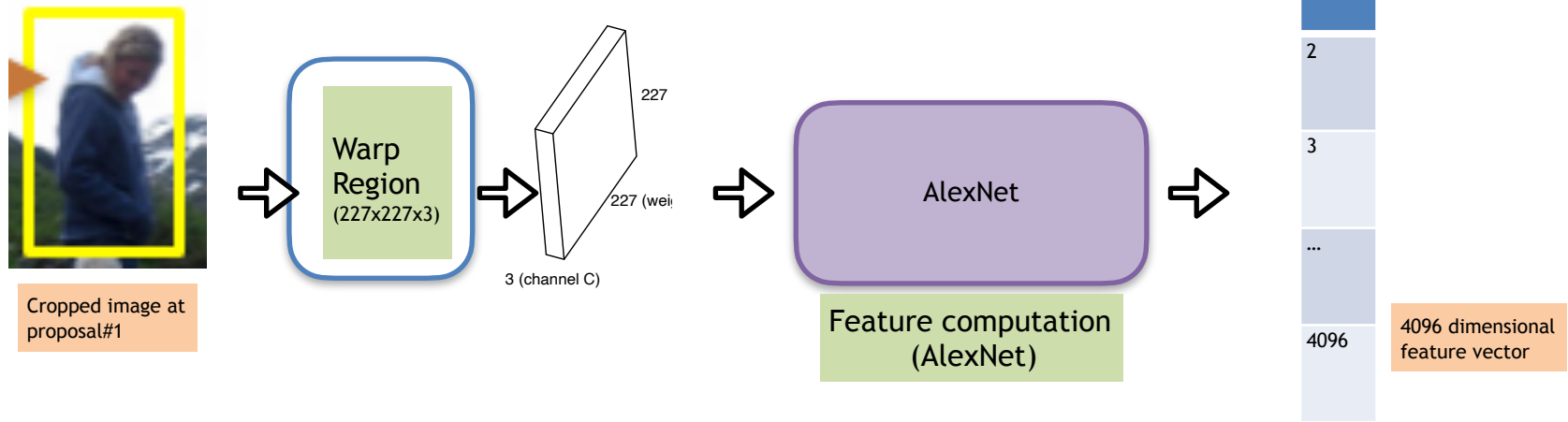
Review: Object Detector#1: R-CNN

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

Review: Object Detector#1: R-CNN



Cropped image at **proposal#1**



Cropped image at **proposal#2**



Cropped image at **proposal#3**

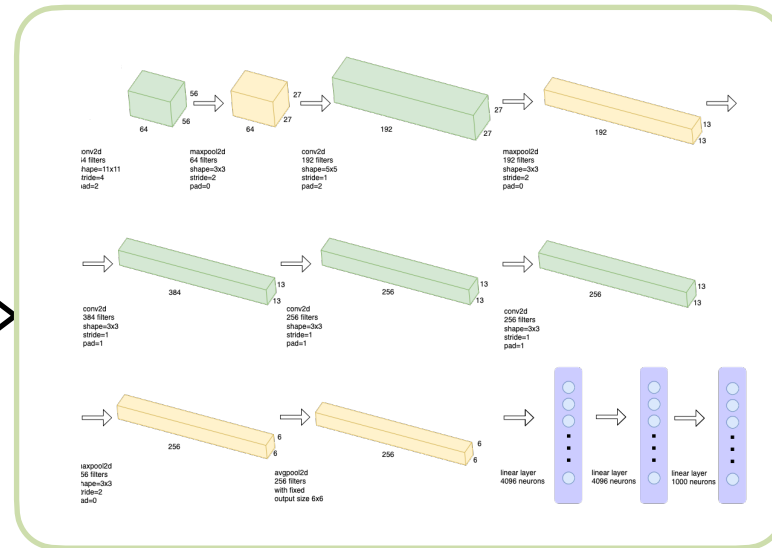
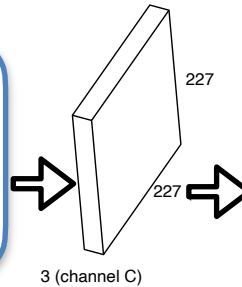


Cropped image at **proposal#4**

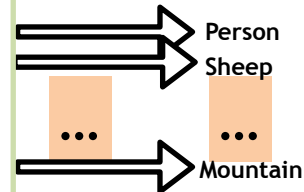


Cropped image at **proposal#2000**

Warp
Region
(227x227x3)



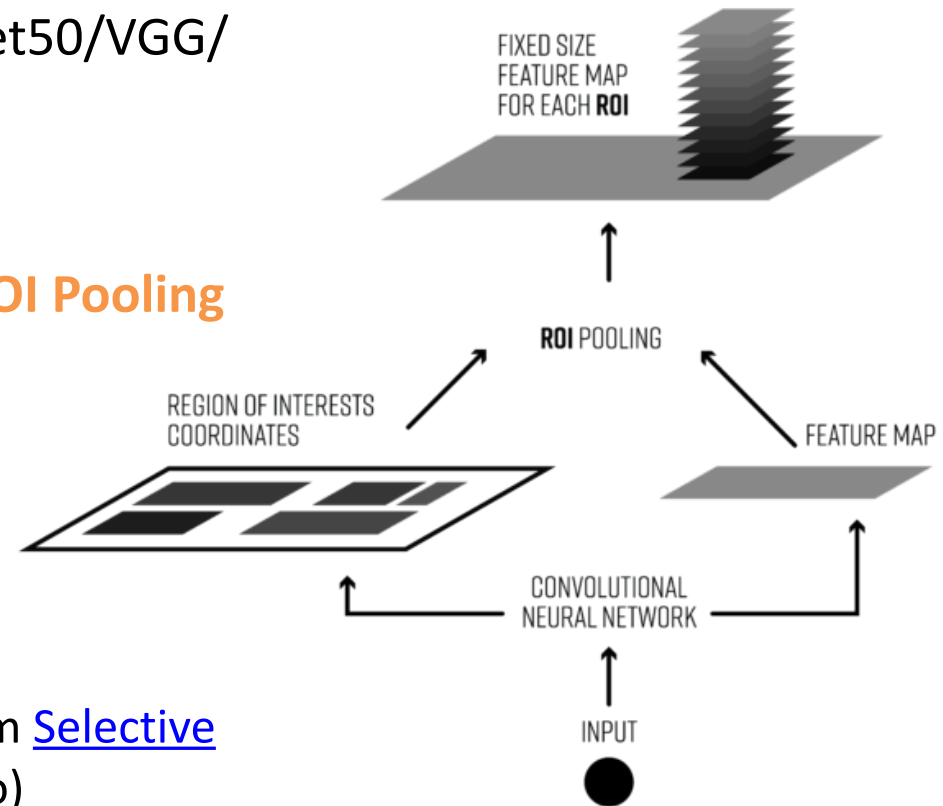
Feature computation
(AlexNet)



Classify

Review: Object Detector#2: Fast R-CNN

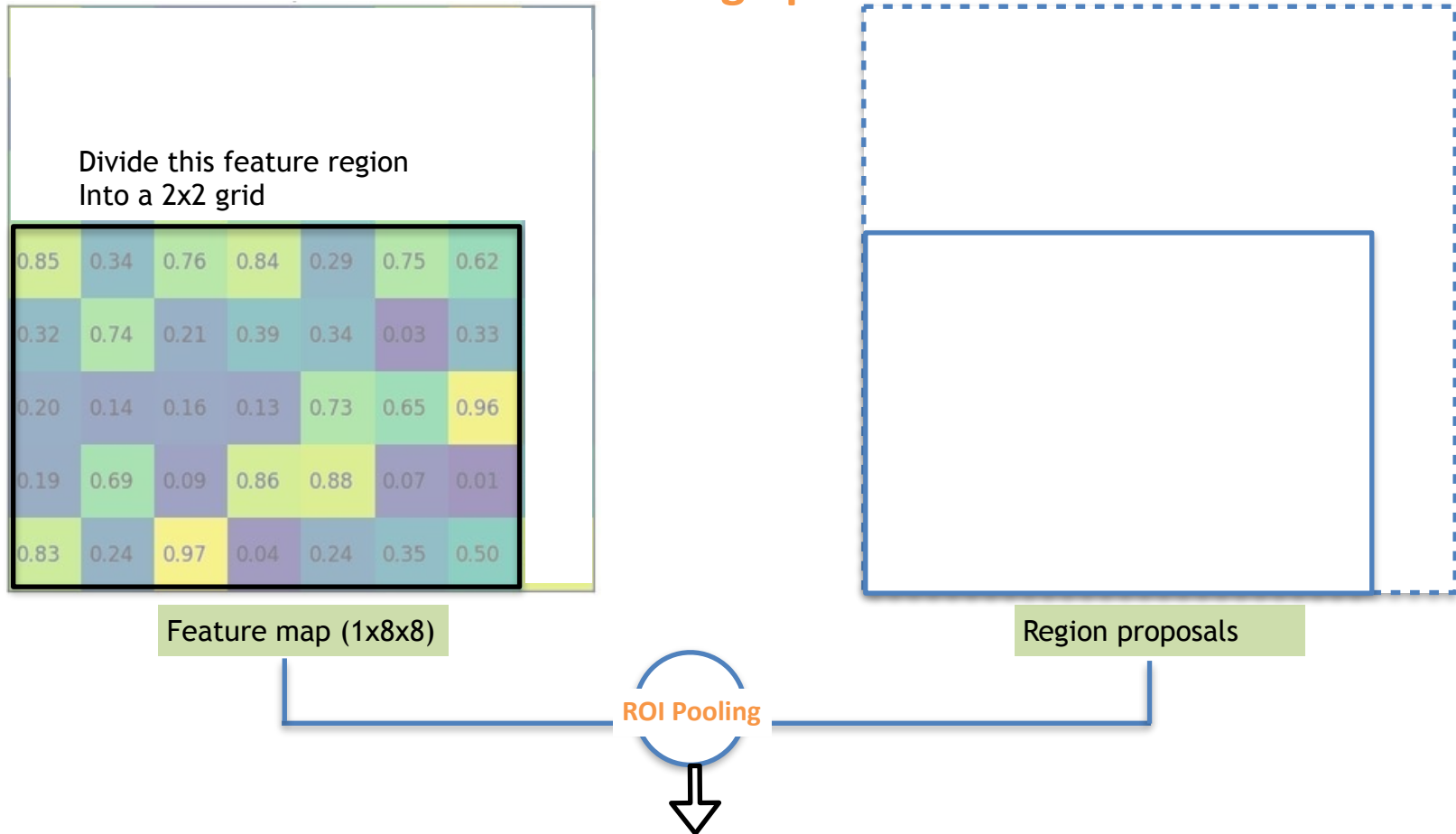
- Feed the image through a CNN (ResNet50/VGG/AlexNet)
- Extract fixed-size feature map using **ROI Pooling**
 - for different **Region Of Interests (ROI)**
- End-to-end trainable:
 - except proposals generation part from [Selective Search - IJCV'13](#) (R-CNN does that too)



[Fast R-CNN - Girshick et al. ICCV'15](#)

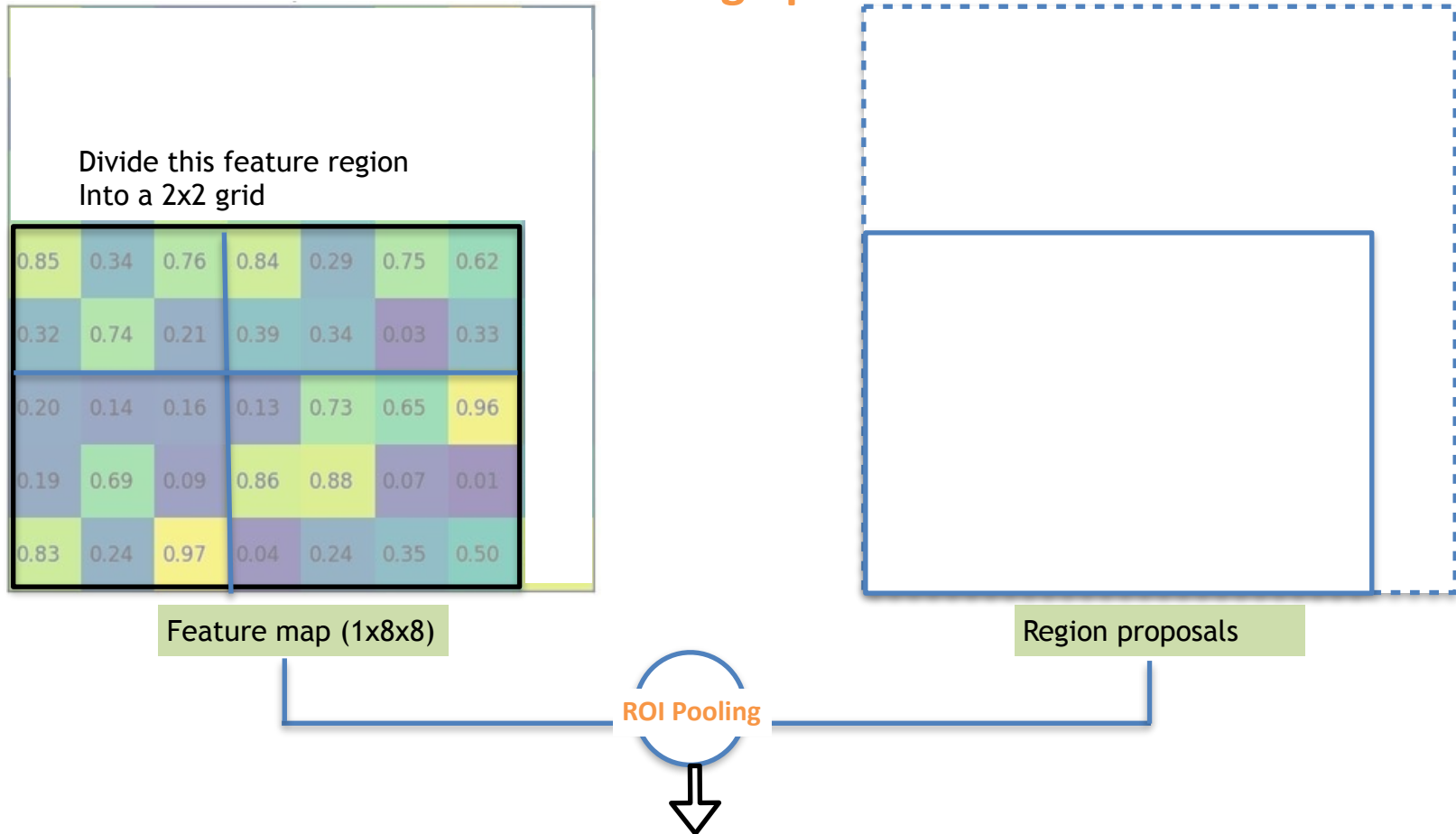
Review:Object Detector#2: Fast R-CNN

ROI Pooling operation



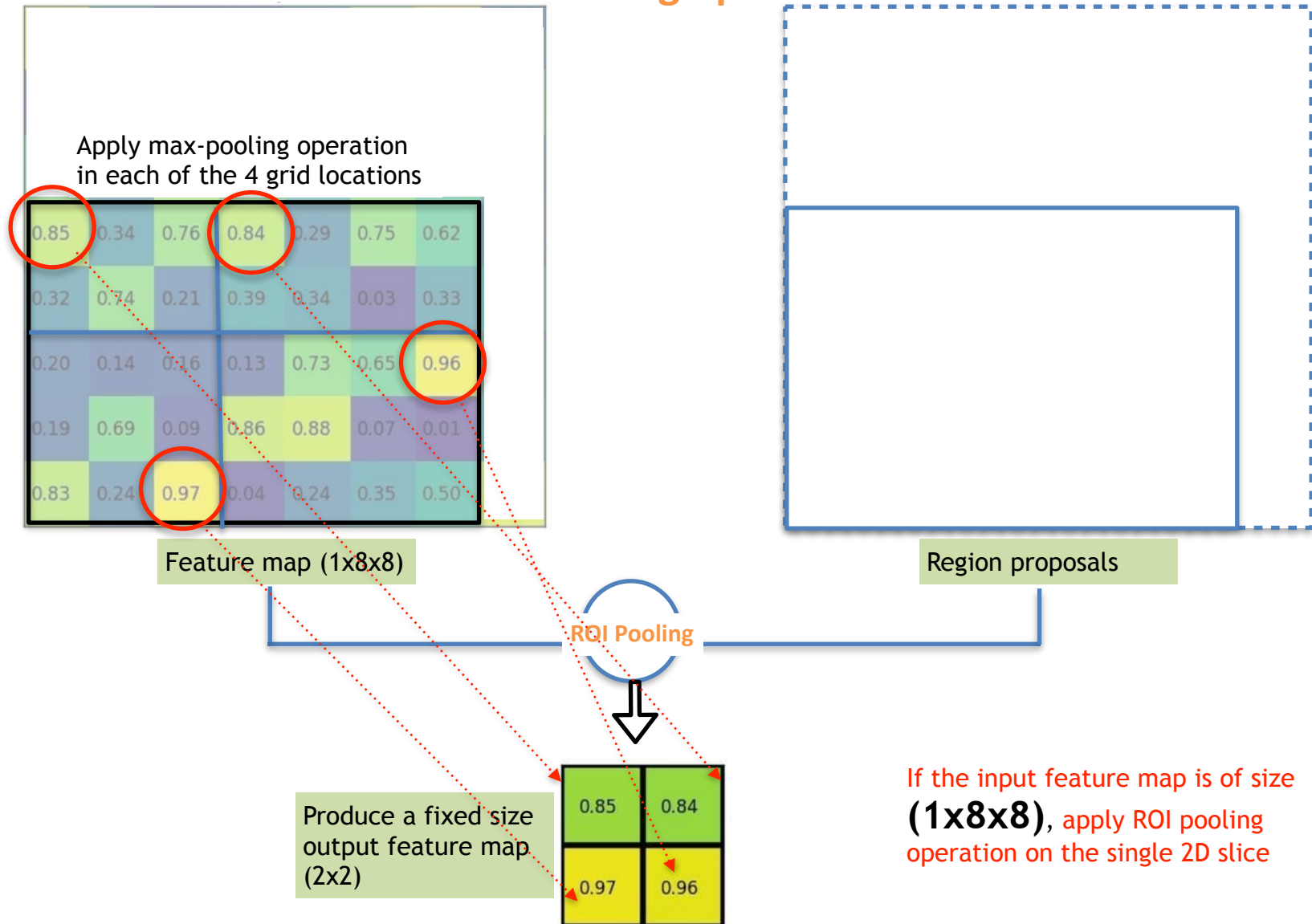
Review:Object Detector#2: Fast R-CNN

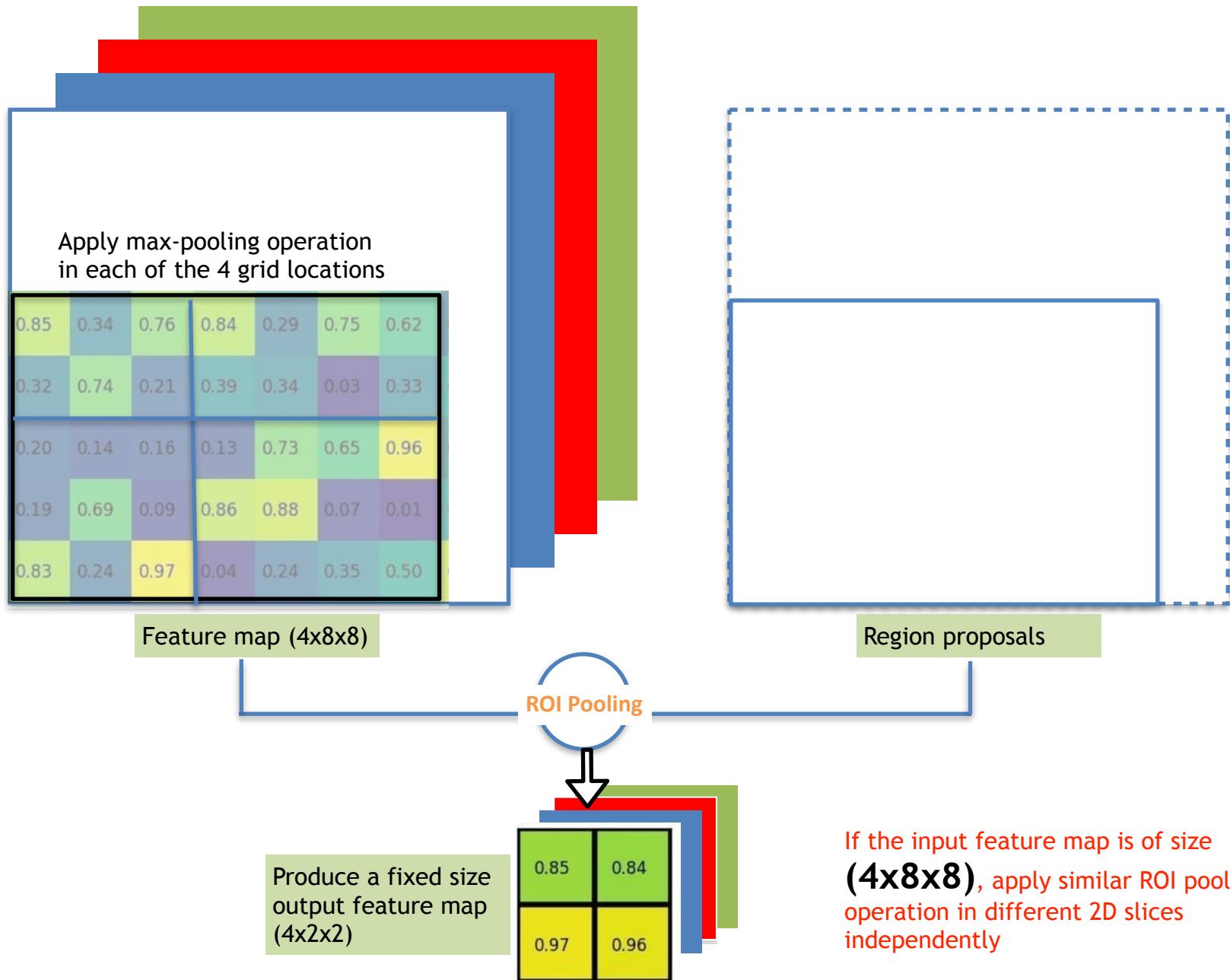
ROI Pooling operation



Review:Object Detector#2: Fast R-CNN

ROI Pooling operation





Deep learning-based object detections

- R-CNN
- Fast RCNN
- **Faster RCNN**

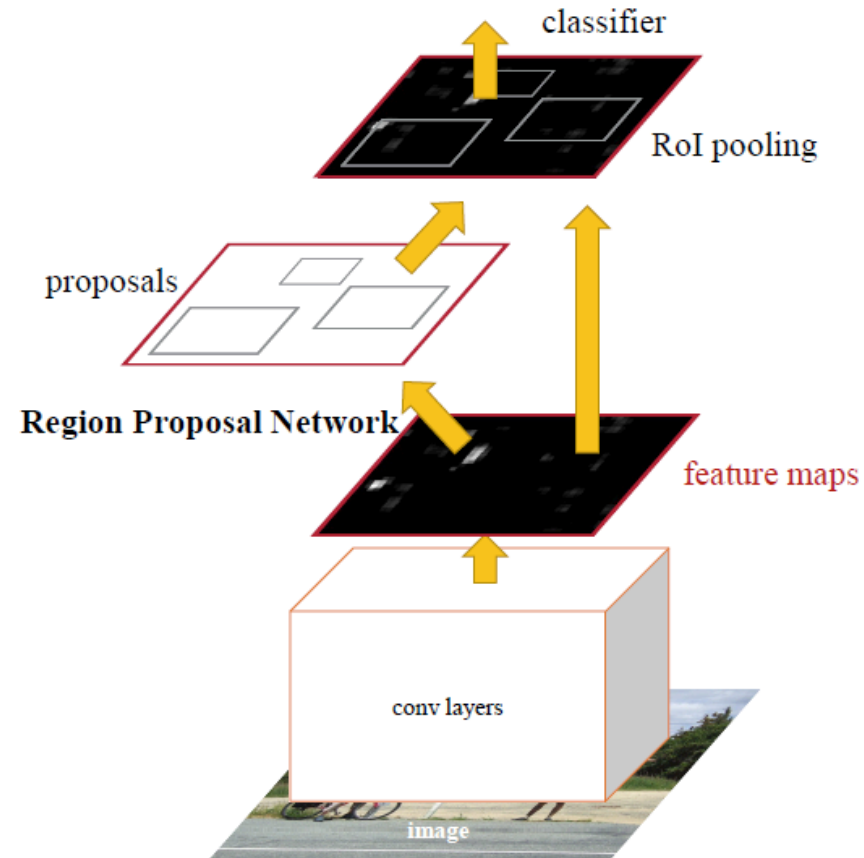
- Mask RCNN

- YOLO
- SSD

Faster R-CNN

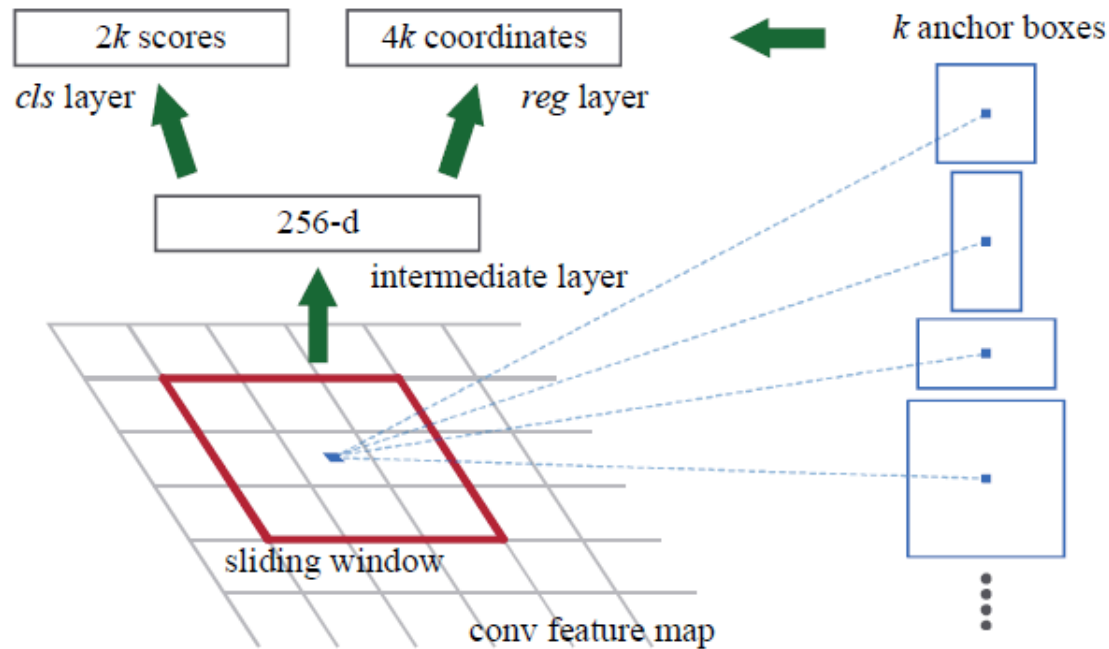
- Introduce a new network which will learn to generate good proposals. It is called **Region Proposal Network (RPN)** which will generate proposals.
- For each proposal, extract fixed-size feature map using **ROI Pooling**
- The network is end-to-end trainable:
 - Unlike R-CNN where different components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

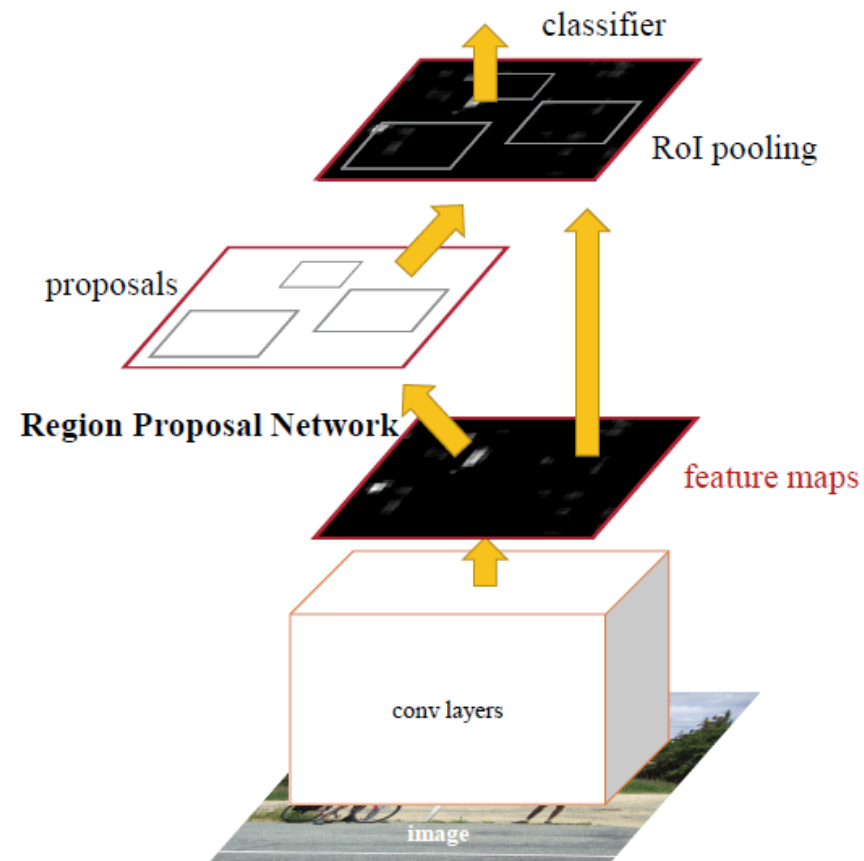
PyTorch's Detectors

- 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

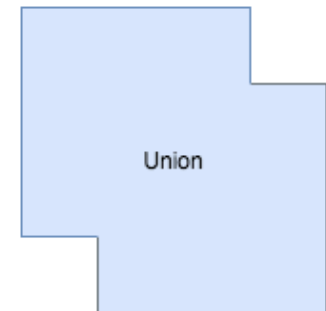
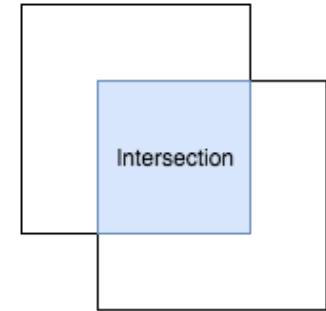
- Let's take a pretrained Faster R-CNN model which was trained on Microsoft COCO (Common Object in COntext) Dataset with 91 categories
 - First, let's [run the inference](#) on an arbitrary dataset eg, Pedestrian Dataset
 - Then, [fine-tune the pretrained model](#) using the Pedestrian Dataset



How do you evaluate your detection?

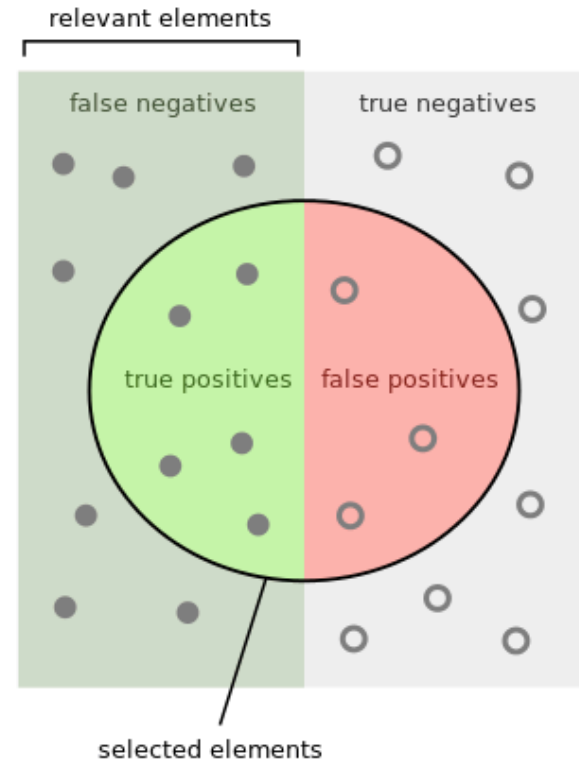
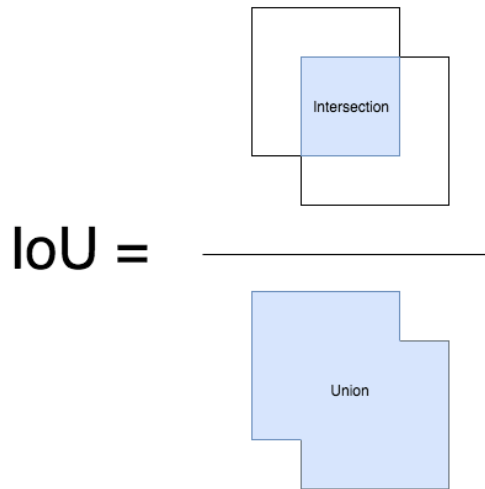
- Intersection over Union (IoU):
 - evaluation metric to assess the detection performance. It requires computation of two entities
 - **Numerator:** Intersection between the predicted bounding box and the ground-truth bounding box
 - **Denominator:** Union between the predicted bounding box and the ground-truth bounding box

$\text{IoU} =$

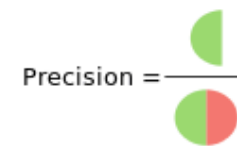


How do you evaluate your detection?

- $\text{IoU} > 0.5$ is True Positive
- $\text{IoU} < 0.5$ is False Positive

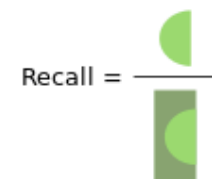


How many selected items are relevant?



Precision =

How many relevant items are selected?



Recall =