
The Ancient
Secrets
Computer Vision

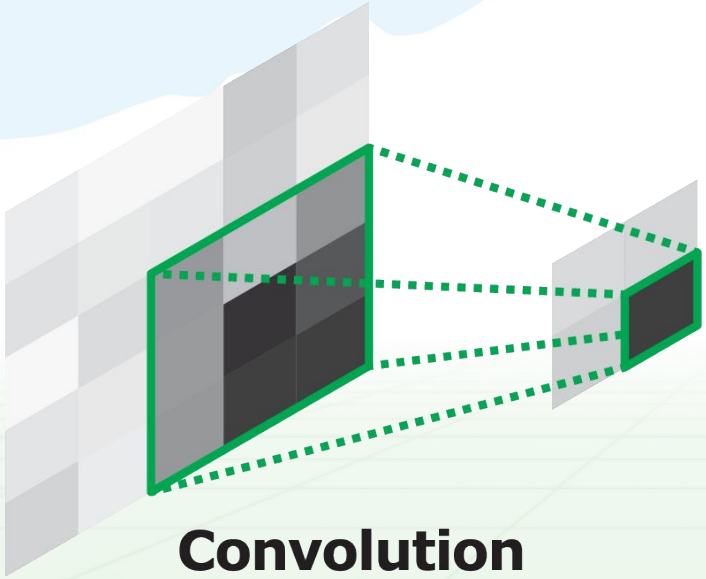
Logistics:

- You are working on your final projects (I hope!)
- Deliverables:
  - Poster session, June 6th, 4:30 - 6:30pm
  - Set up at 4pm, poster board and easels provided
  - You will be graded by at least 3 people (TAs or me or Ali)
  - Rubric is on Canvas
  - Demos are encouraged (live or recorded examples of project)
  - Poster options:
    - Make your own, print it, assemble it on poster board
    - OR
    - Use Kiana's template, send it to cse455-staff@cs.washington.edu Sunday
    - Template is on the website
    - Turn in poster on Canvas as well.

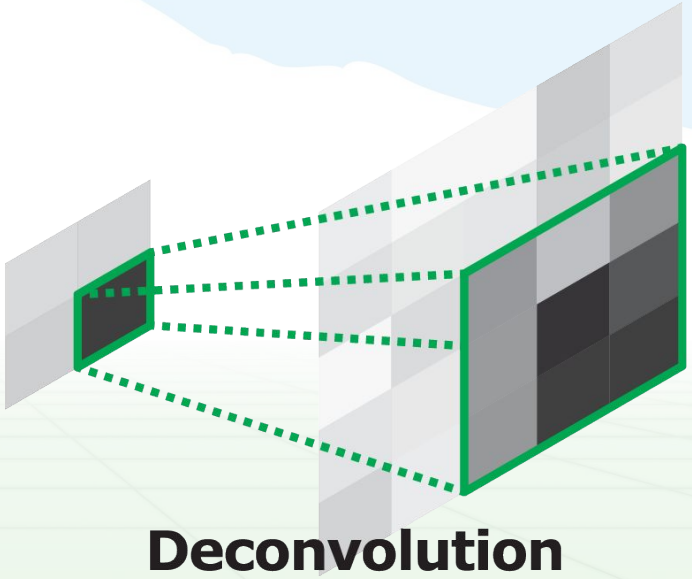

Previously
On
the
Ancient Secrets
of Computer Vision

Semantic Segmentation

Semantic Segmentation



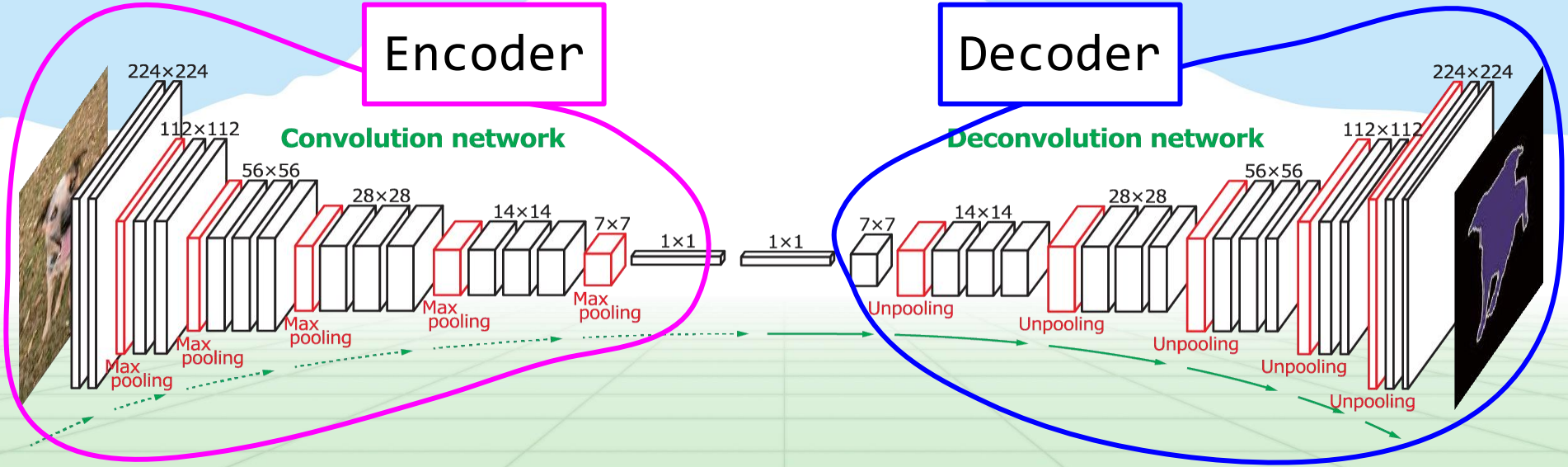
Convolution



Deconvolution

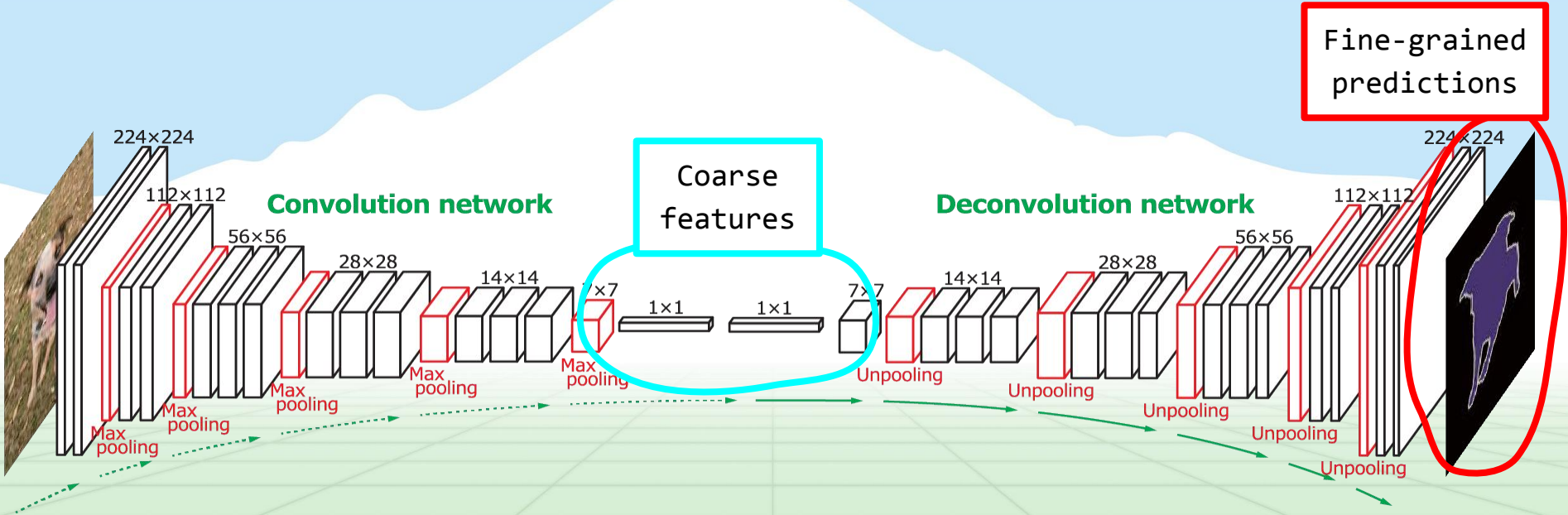
https://arxiv.org/pdf/1505.04366.pdf

Semantic Segmentation



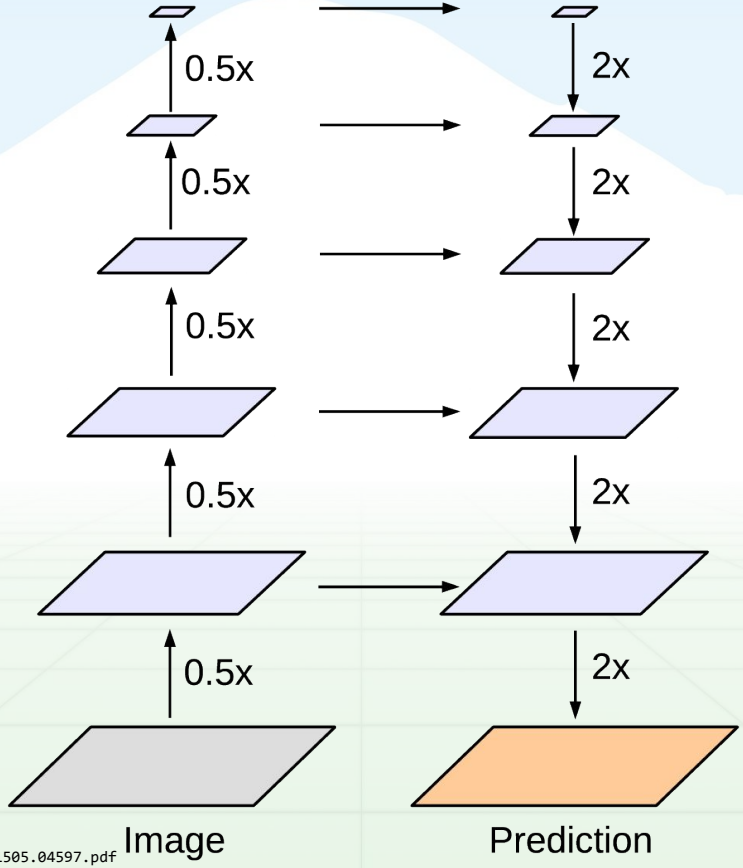
https://arxiv.org/pdf/1505.04366.pdf

Semantic Segmentation



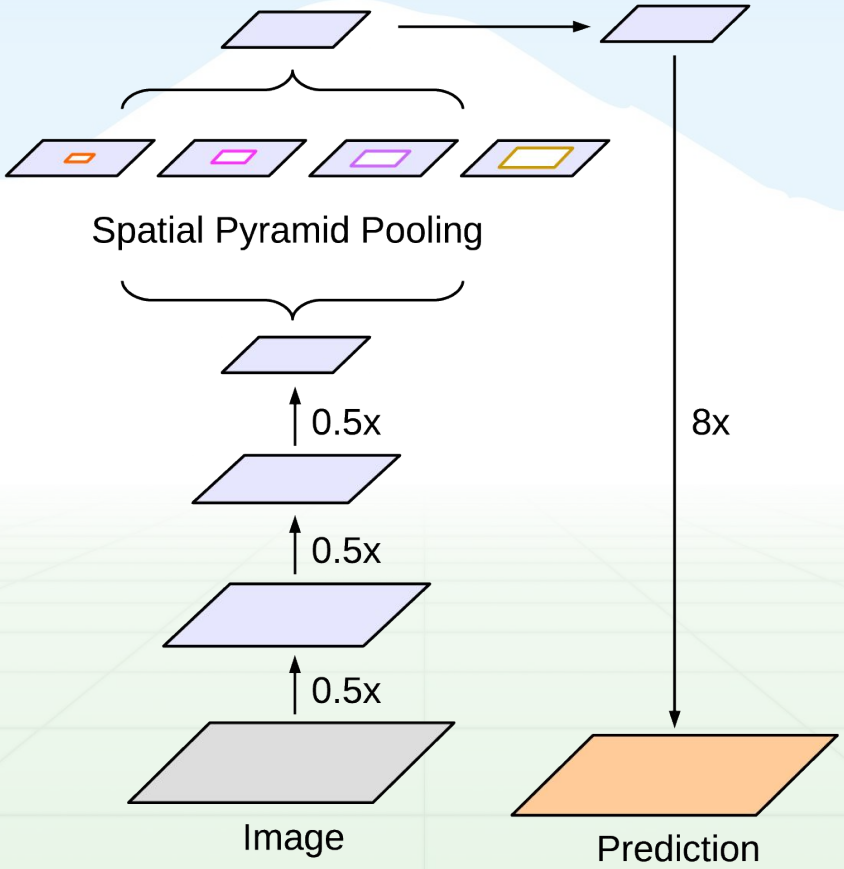
https://arxiv.org/pdf/1505.04366.pdf

U-net/Segnet



https://arxiv.org/pdf/1511.00561.pdf, https://arxiv.org/pdf/1505.04597.pdf

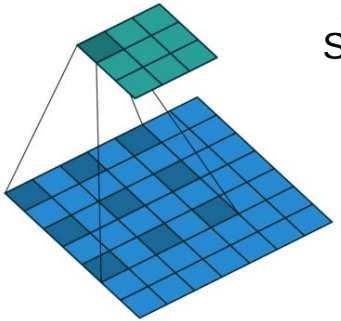
Spatial pyramid pooling



DeepLabv3+

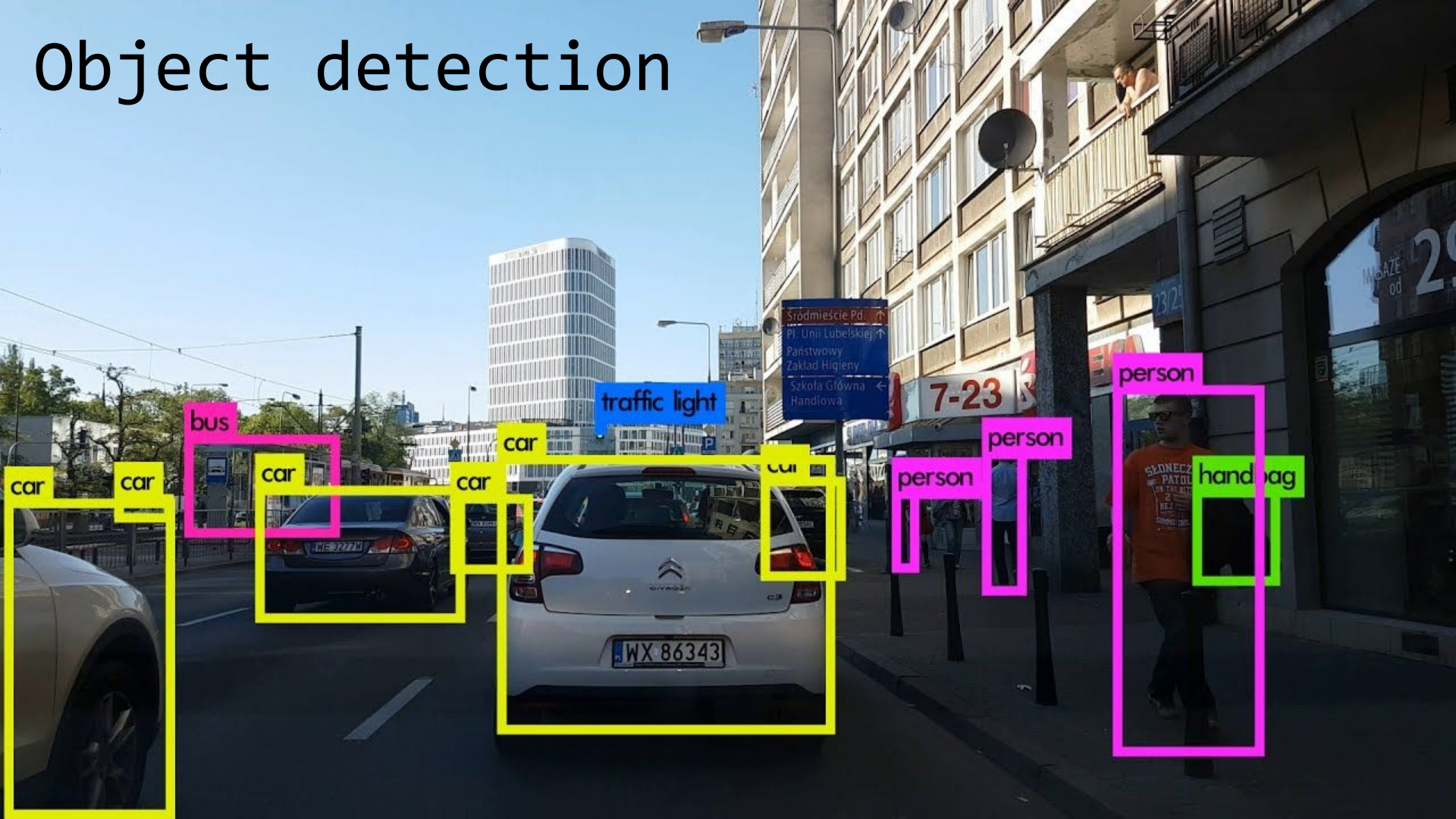
Atrous convolutions

Spaced inputs



https://arxiv.org/pdf/1802.02611.pdf

Object detection



PASCAL VOC

One of the first large detection datasets:

- 20 classes
- 11,530 training images
- 27,450 annotated objects

Also used for semantic segmentation!

http://host.robots.ox.ac.uk/pascal/VOC/

Scoring object detection

Multiple classes, multiple objects per images

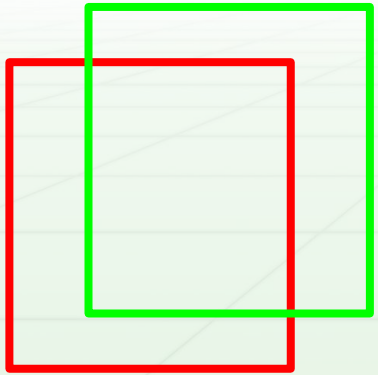
Can’t just use accuracy

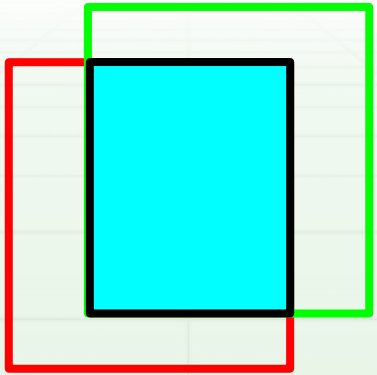
“Correct” bounding box:

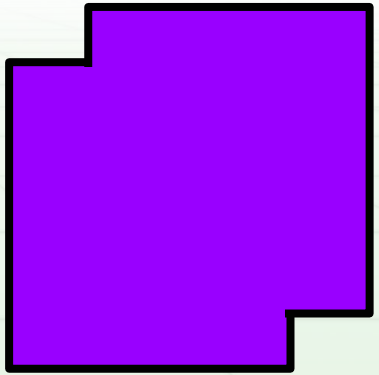
Intersection / Union > 0.5

Intersection: Ground truth ∩ prediction

Union: Ground truth ∪ prediction







Scoring object detection

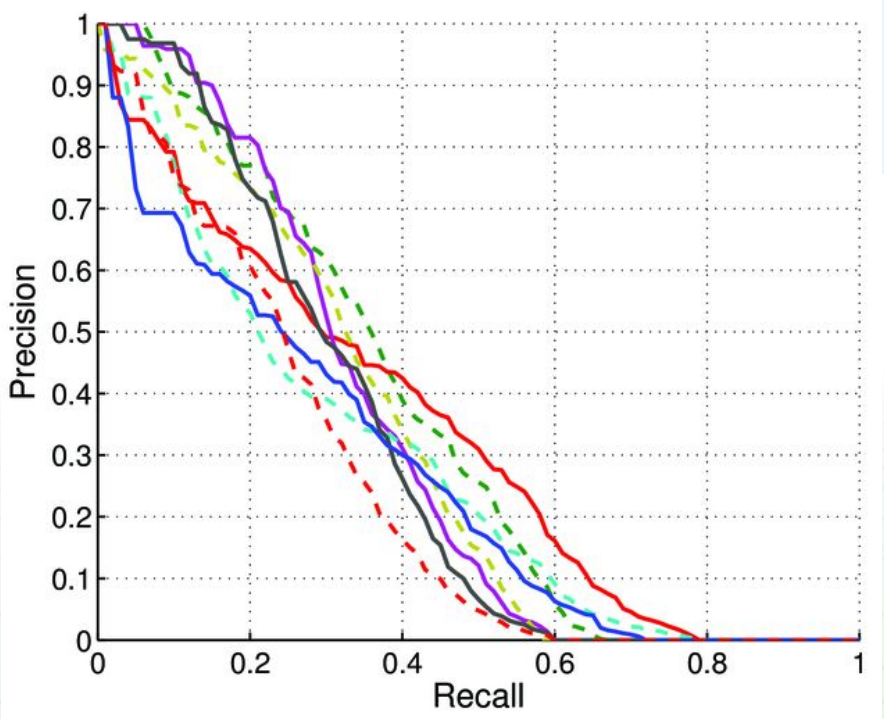
Precision-Recall curve: vary threshold, plot precision and recall

Average precision:

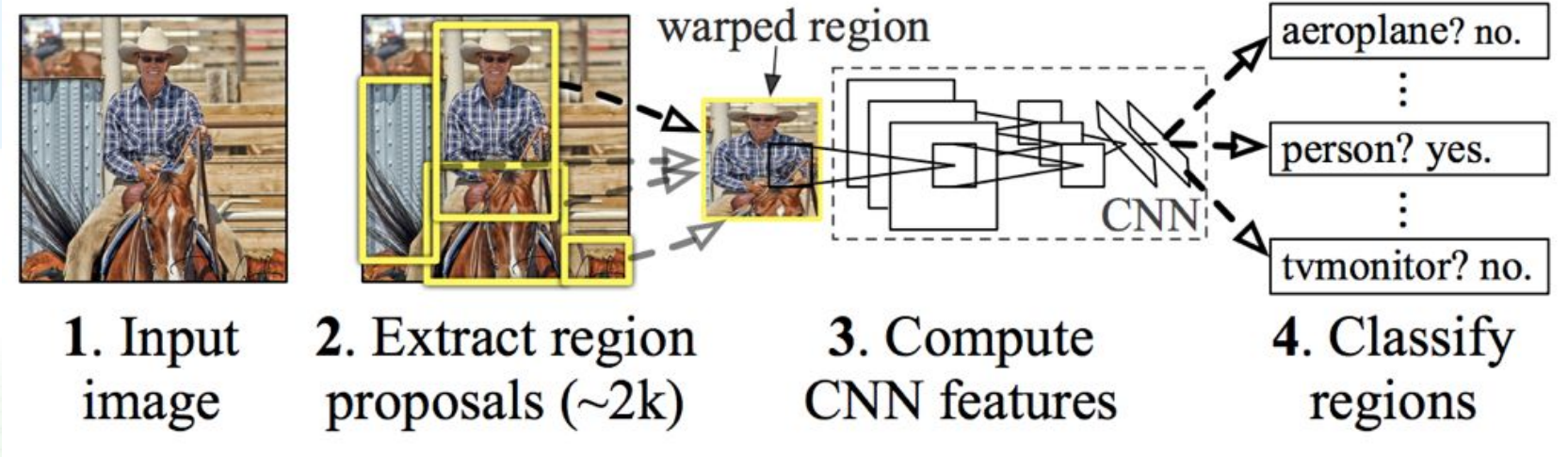
- Area under PR curve
- Only for a single class

Take mean of AP across classes:

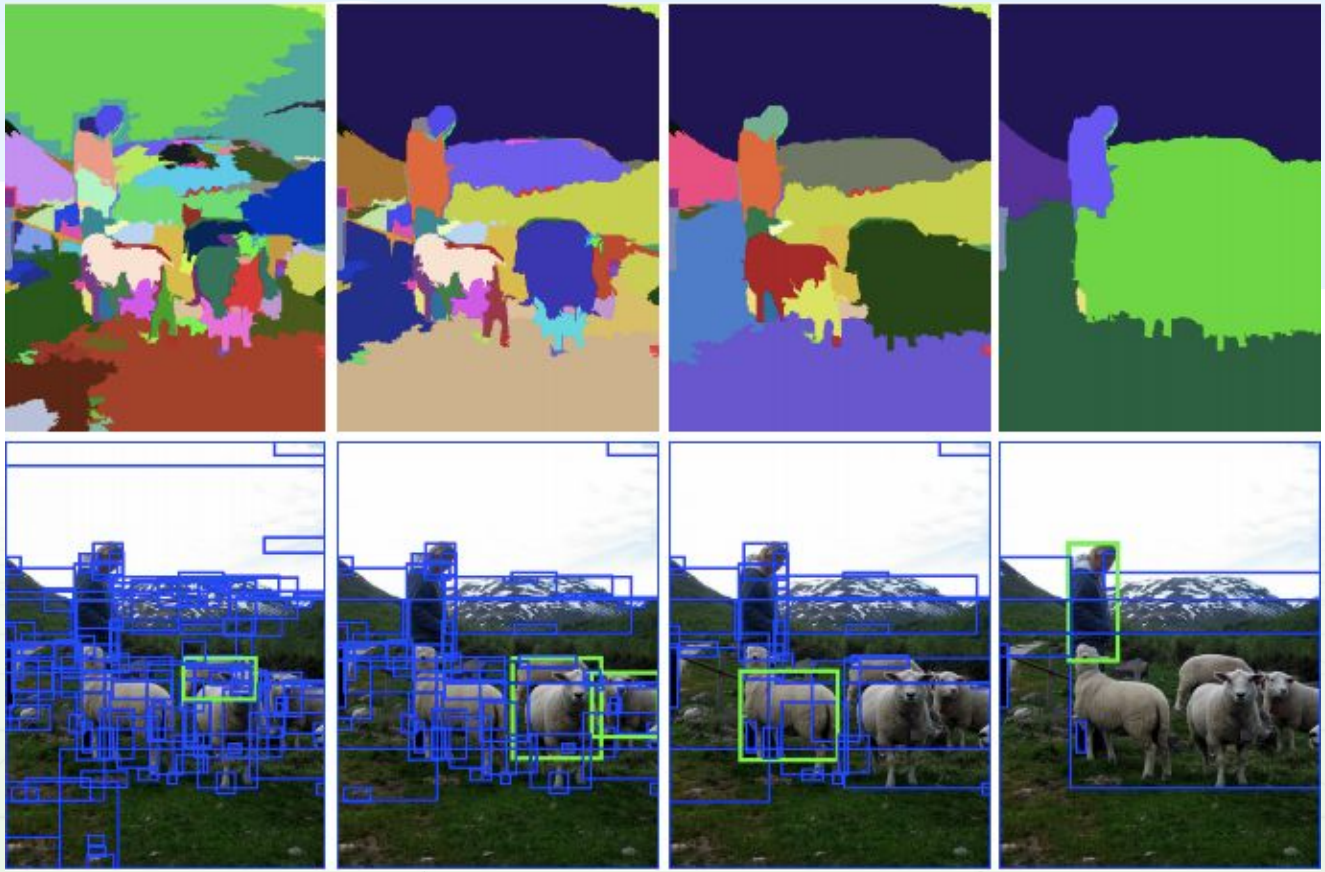
- Mean AP (mAP)
- Standard detection metric
- Sometimes at particular IOU
  - I.e. mAP@.5 or mAP@.75



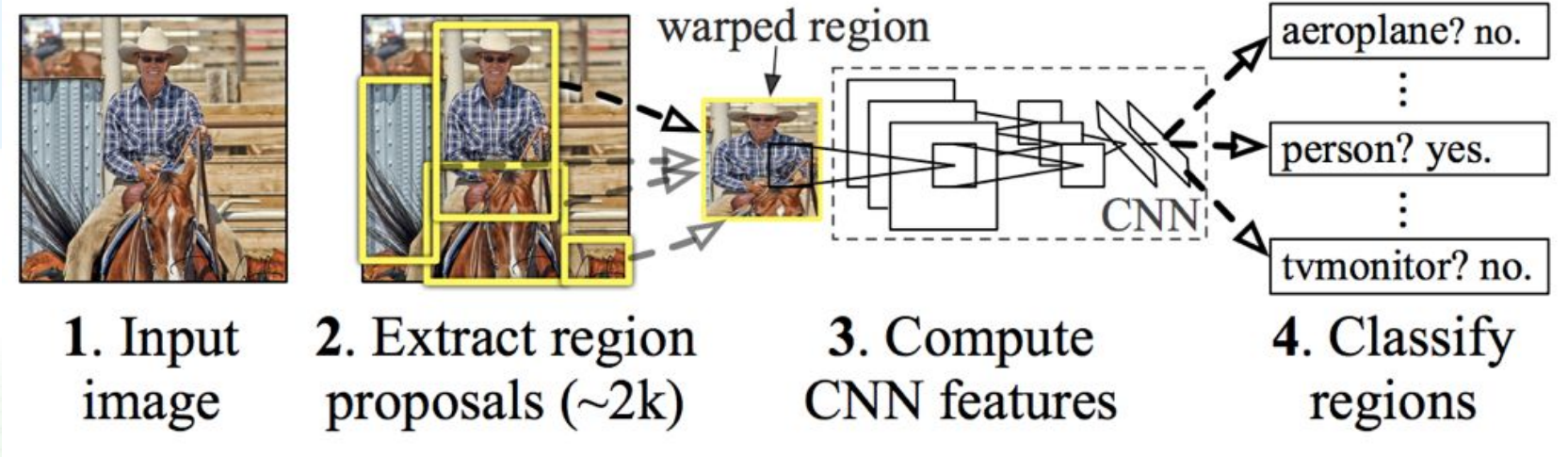
R-CNN: Regions with CNN features



Selective search: fewer proposals



R-CNN: Regions with CNN features



Lots of post processing, 20 sec/im

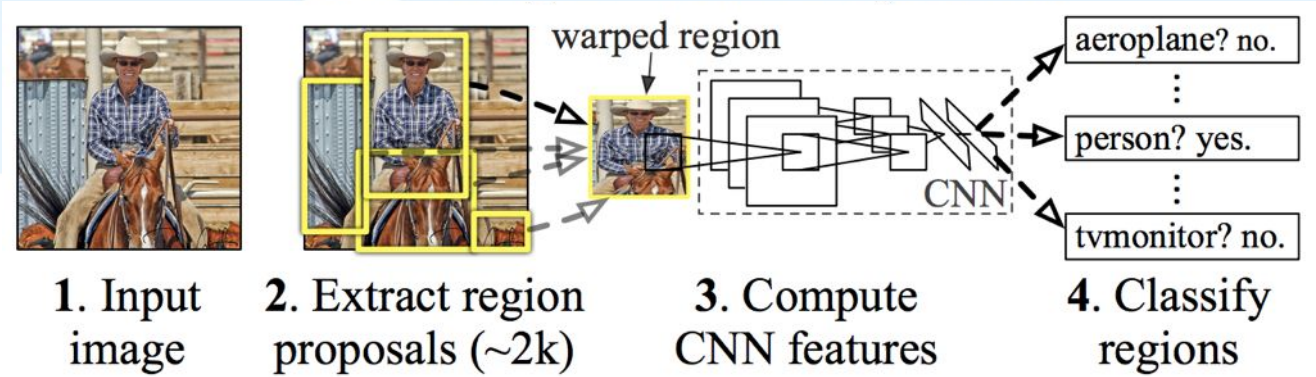
Pascal VOC:

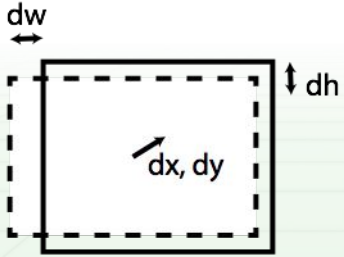
AlexNet

53.3% mAP

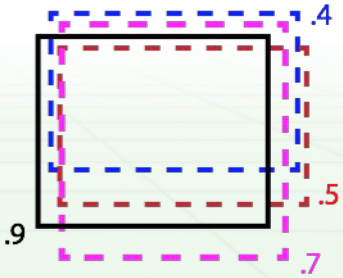
VGG-16

62.4% mAP

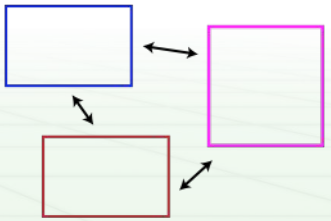




5. Bounding box regression

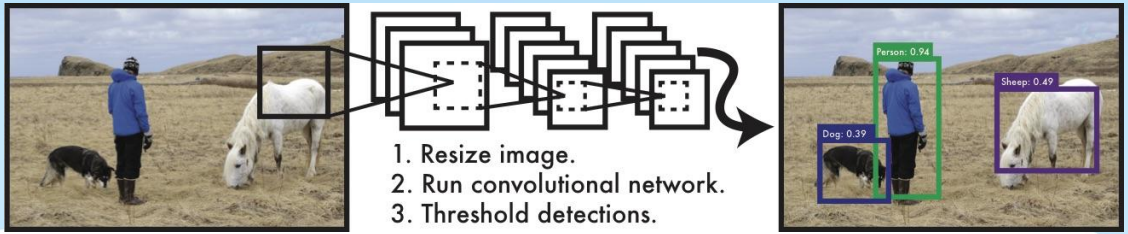


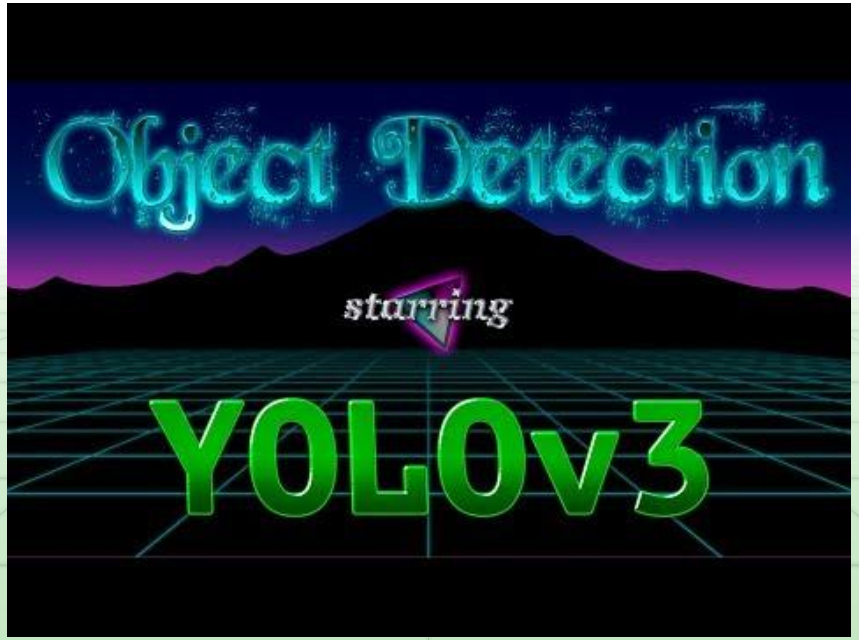
6. Non-max suppression



7. RNN rescoring??

YOLO

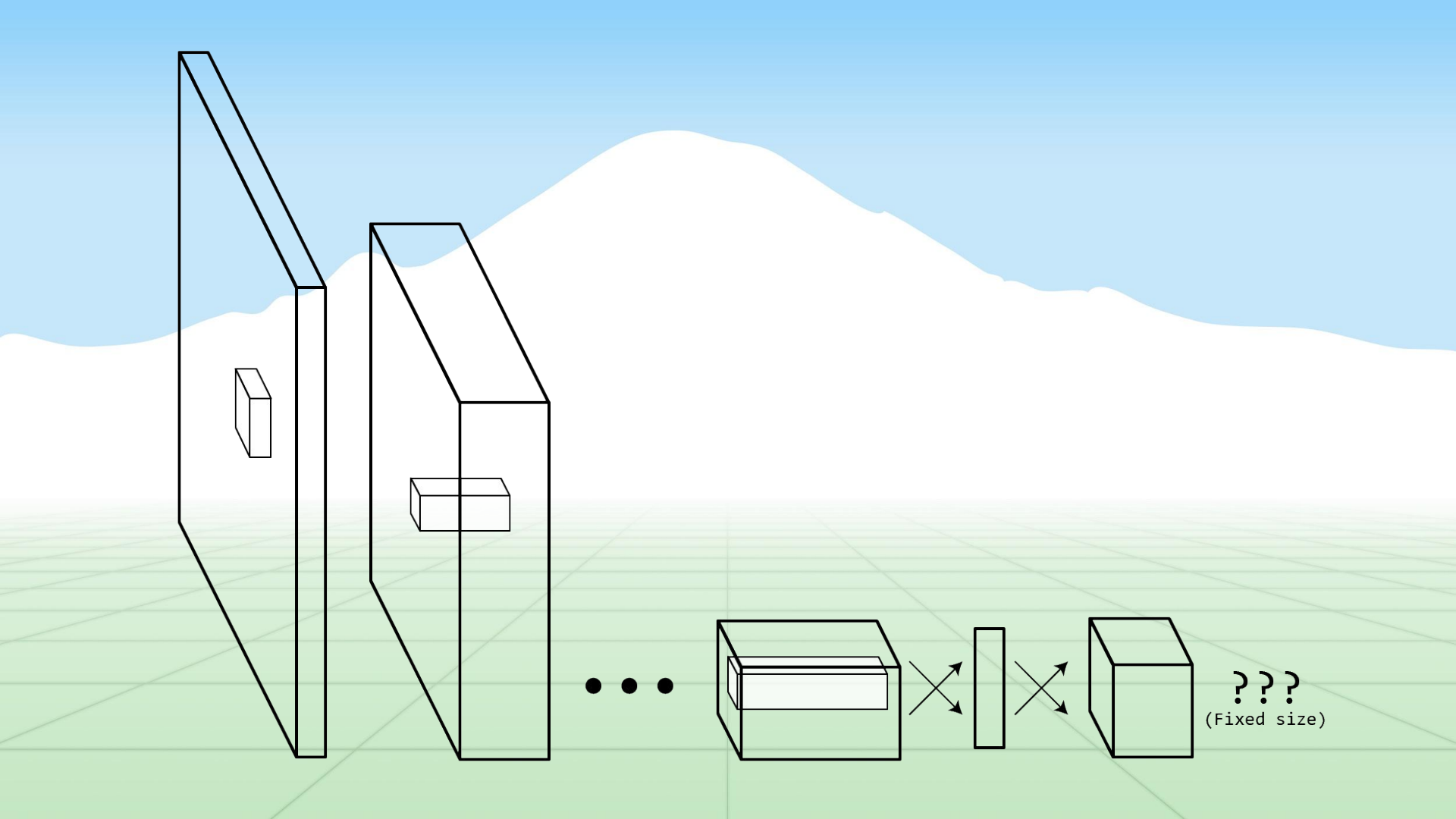

1. Resize image.
2. Run convolutional network.
3. Threshold detections.
Person: 0.94
Sheep: 0.49
Dog: 0.39


Object Detection
starring
YOLOv3


Chapter
Eighteen
Detection and
Instance Segmentation

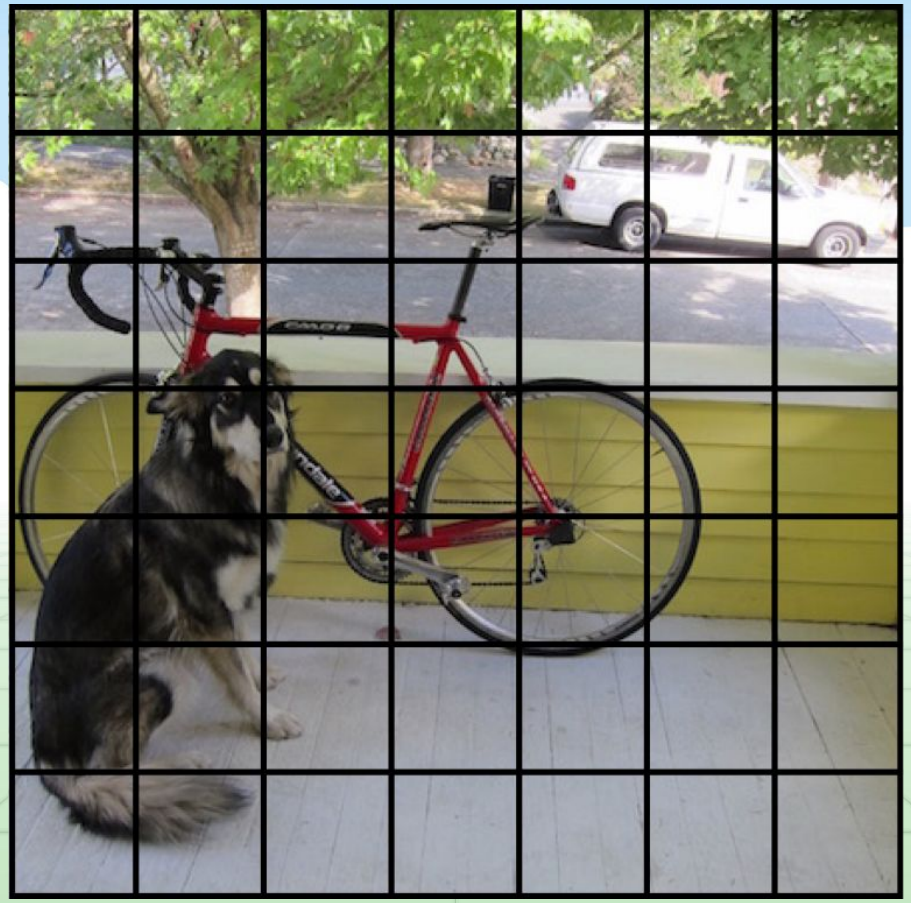
	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img

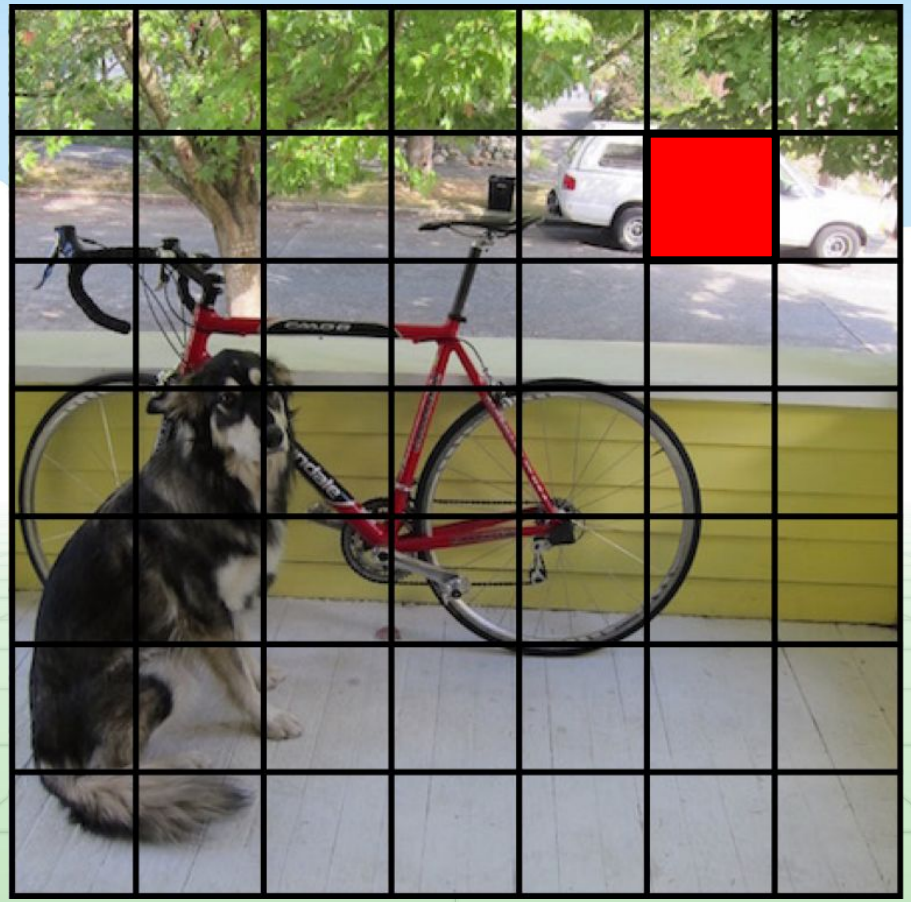

???
(Fixed size)

Say you have an image...

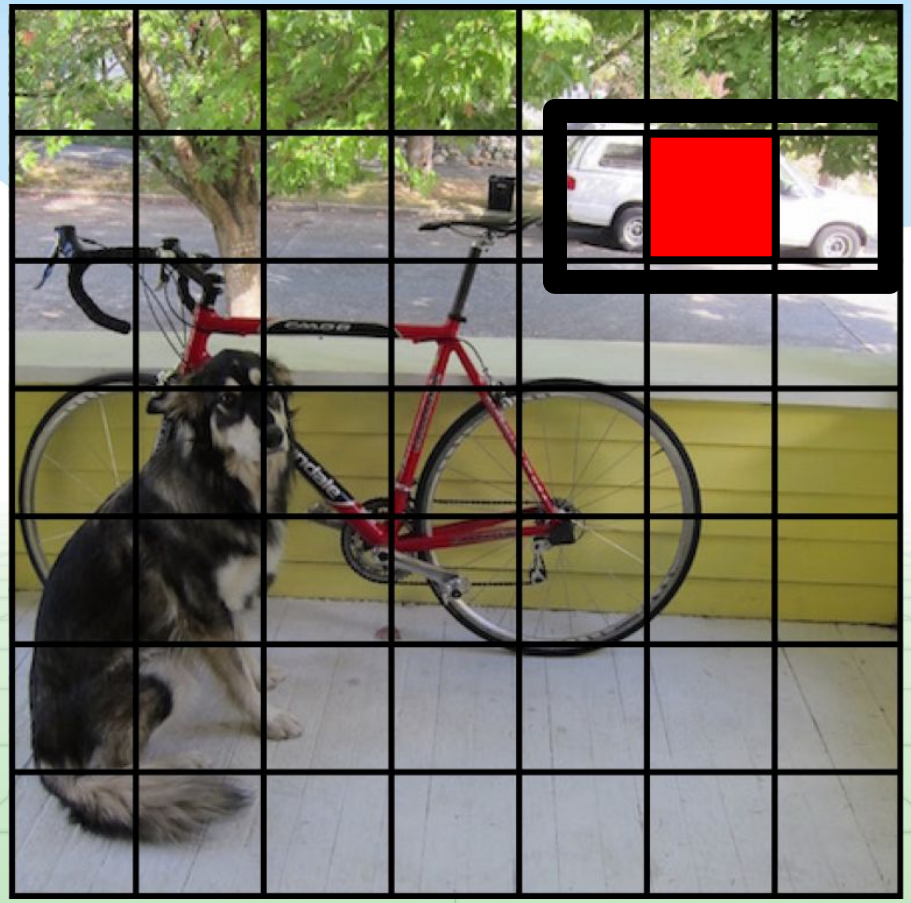
Split it into a grid



For each cell predicts P(obj)

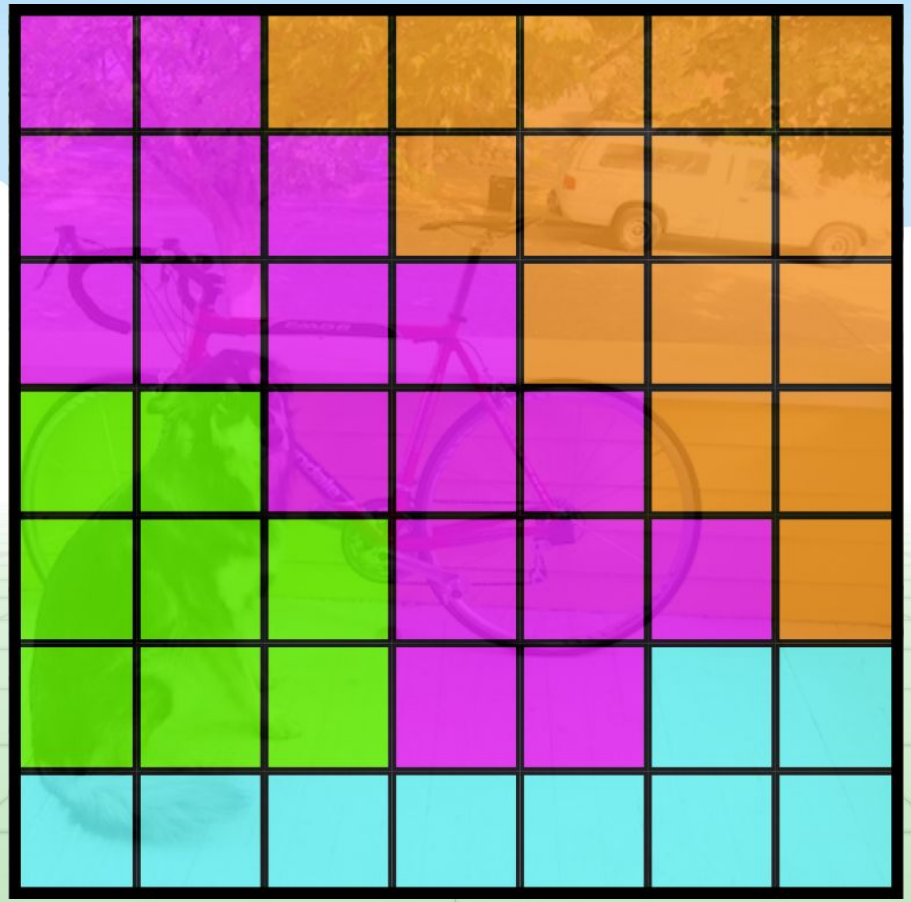


Also predict a bounding box

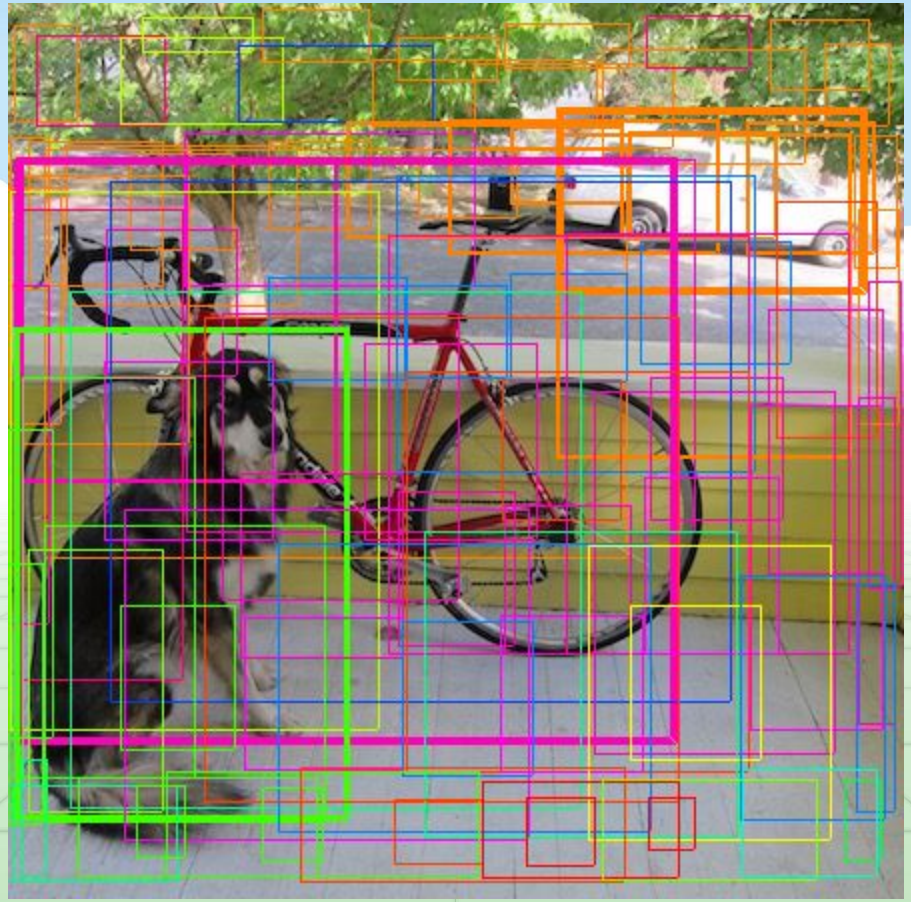


Also predict a bounding box

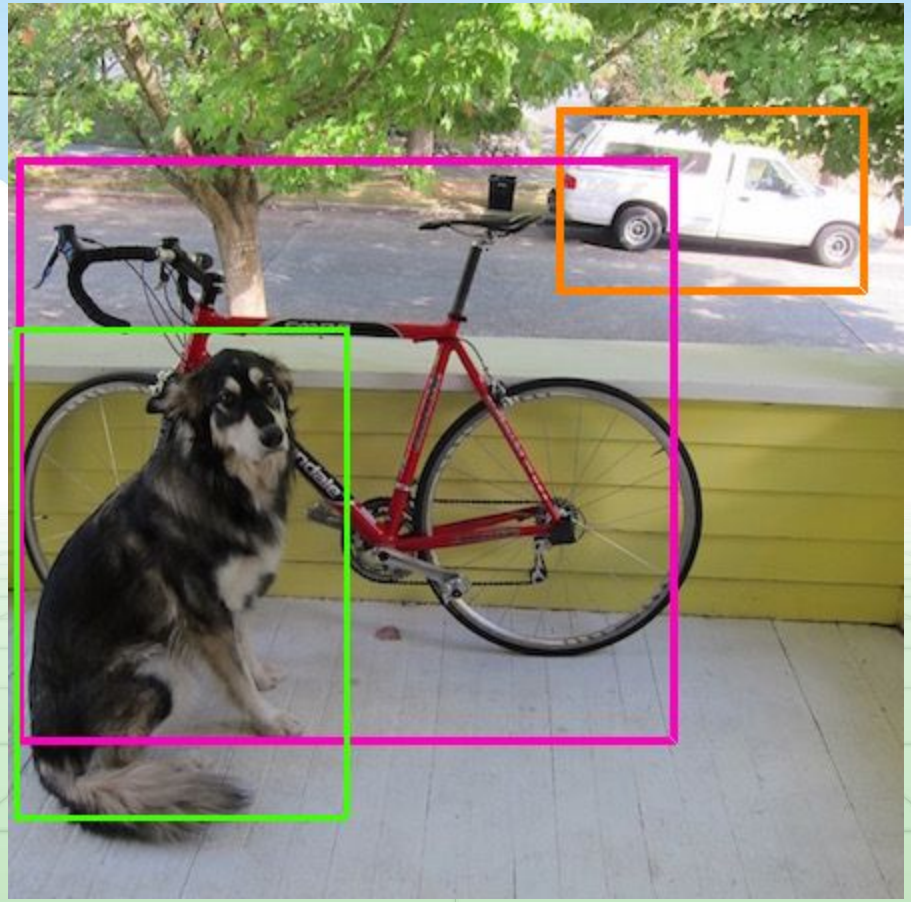
Also class probabilities



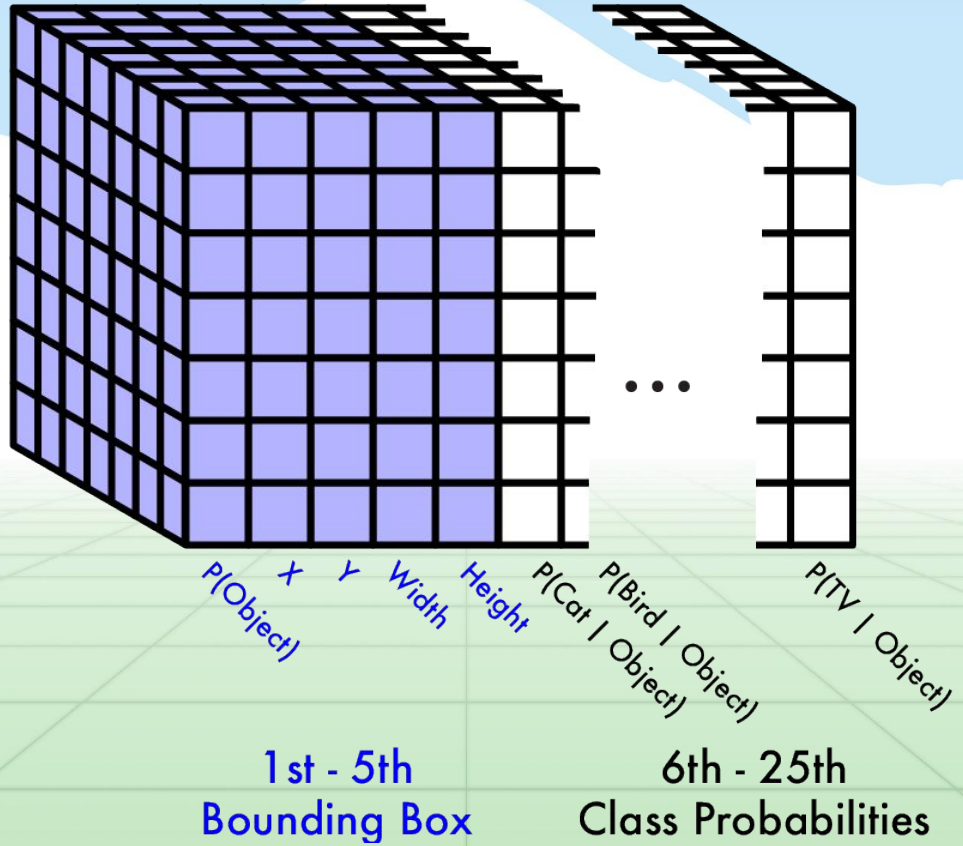
Also class probabilities



Threshold and non-max suppression

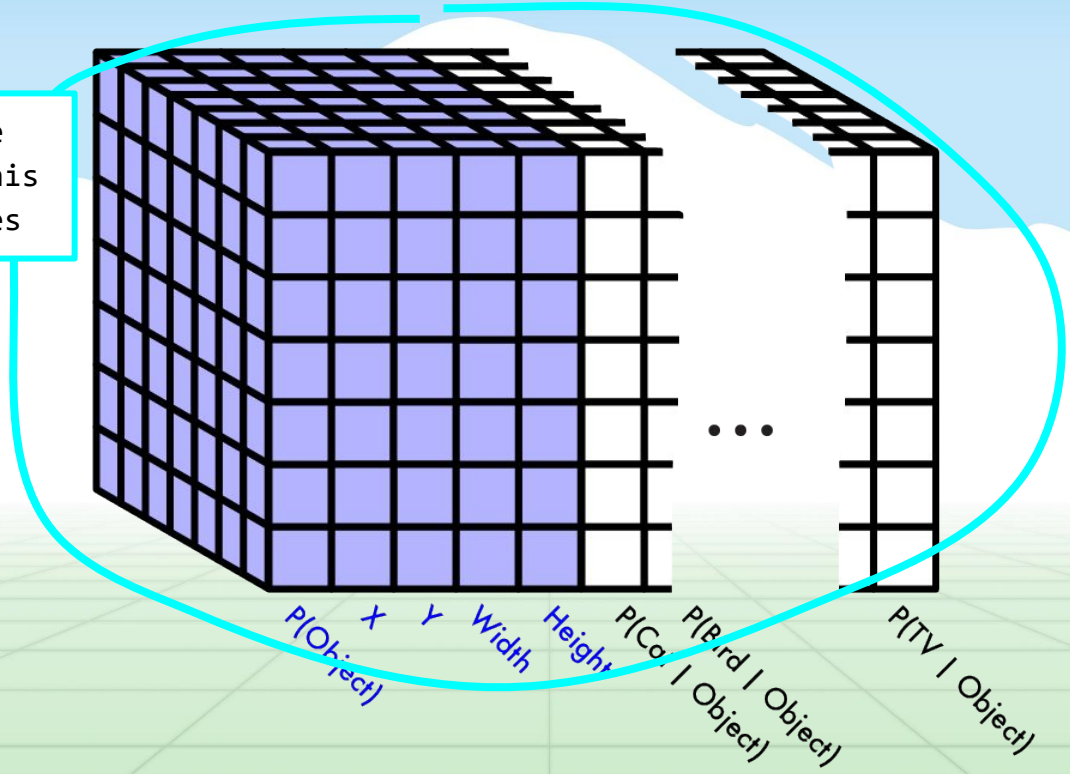


Tensor encoding detection



Tensor encoding detection

Can predict multiple boxes, just stack this tensor multiple times



Multiple bounding boxes per cell

Going to predict 5 boxes per cell

Want spatial diversity between boxes

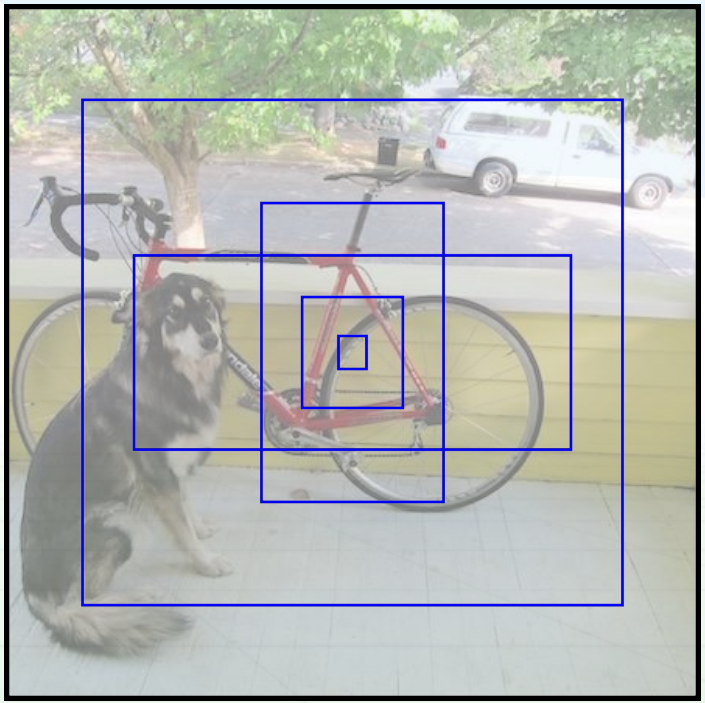
    Some small, medium, large

Don’t predict boxes directly

Predict offsets from priors

Priors should be sensible

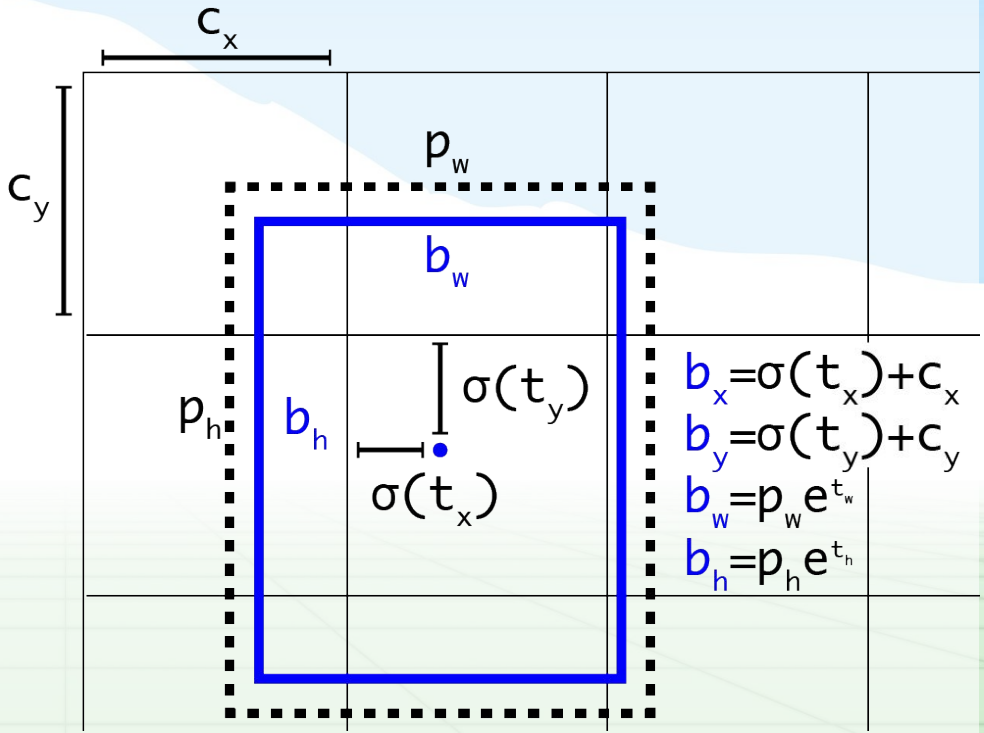
    YOLO uses k-means clustering of training data bounding box (w,h)



Bounding box encoding (YOLOv2 and v3)

Given cell offset c_x and c_y
Prior box p_w and p_h

Our predicted bounding box is given by:



YOLO loss function

$L(YOLO) = \alpha_1 L(confidence) + \alpha_2 L(localization) + \alpha_3 L(classification)$

Can tune all the alphas

L(confidence): binary cross-entropy

L(localization): RMSE

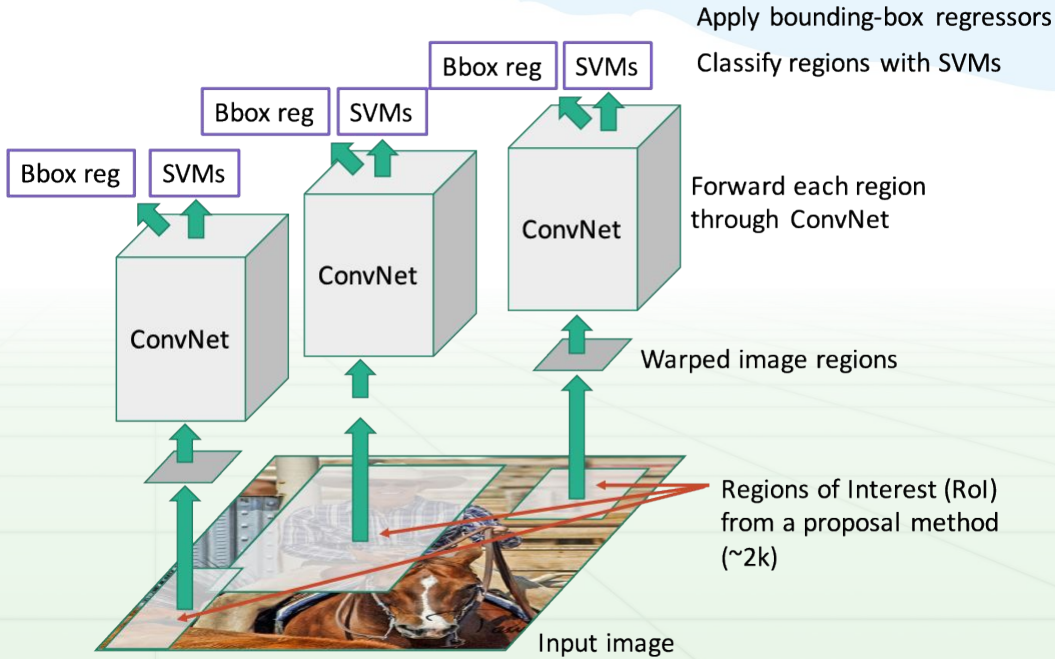
L(classification): multi-class cross-entropy or
1 v all binary cross-entropy

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img
YOLOv2	78.6	40 FPS	25 ms/img
YOLOv3	83.1	30 FPS	33 ms/img

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img
YOLOv2	78.6	40 FPS	25 ms/img
YOLOv3	83.1	30 FPS	33 ms/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
Resnet FRCNN	76.4	??	??
ResNet + COCO data	83.8	??	??
R-FCN	83.6	6 FPS	170 ms/img

R-CNN is slow

Run convnet independently over every ROI



https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

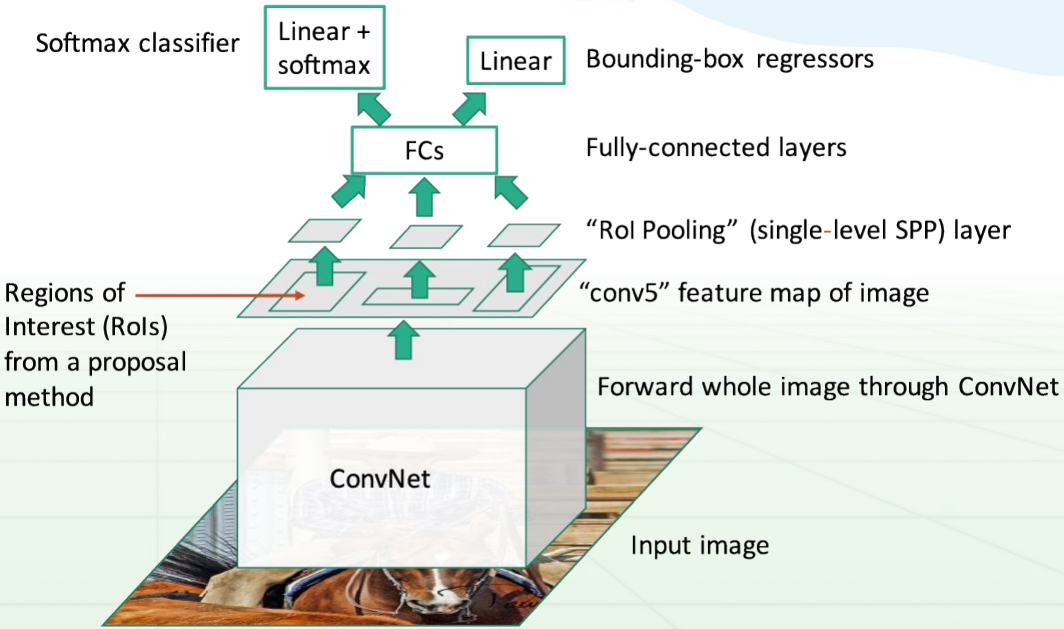
Fast R-CNN

Run convnet once, extract features using ROI pooling

ROI Pool:

Convert variable sized

ROI to fixed size output



https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

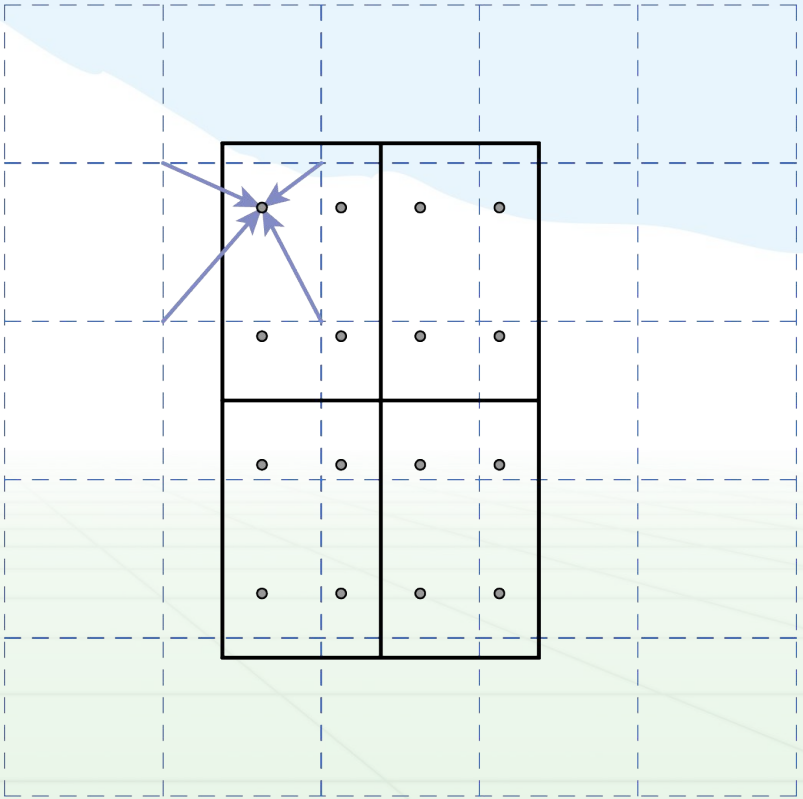
ROI Align

Better than ROI Pool so we'll talk about it instead

Split ROI into fixed size (say 2x2)

Sample image at multiple points for each cell in ROI (bilinear interp.)

Pool over these samples (max, avg…)



https://arxiv.org/pdf/1703.06870.pdf

Fast R-CNN

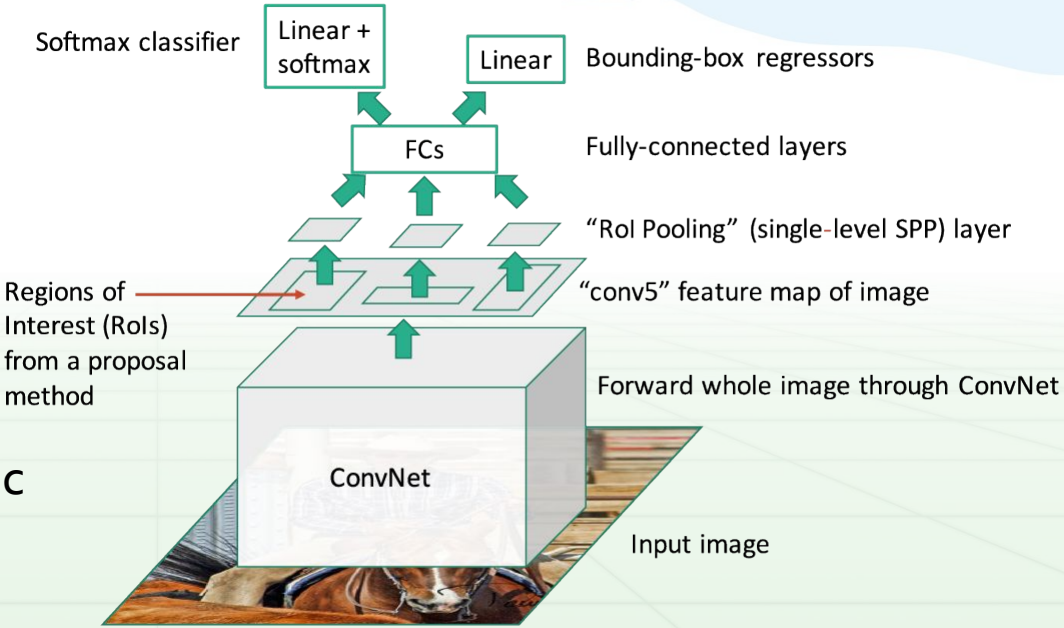
Run convnet once, extract features using ROI pooling

ROI Pool:

Convert variable sized ROI to fixed size output

Much faster, no independent network evals (except last linear layer)

Still slow region proposer, selective search takes ~2 sec



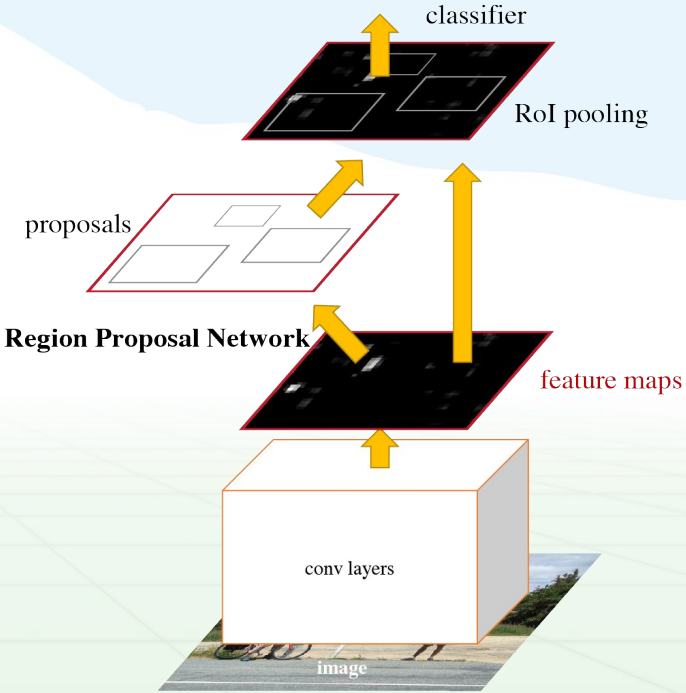
https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

Faster R-CNN

Use Convnet to propose regions *and* generate features

ROI Pool to fix size of ROI features

Additional layers to classify and predict bbox for ROIs



https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

Saturating Pascal VOC, need new data

Average Precision (AP %)

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
FOCAL_DRFCN(VOC+COCO, single model) [?]	**88.8**	95.0	93.3	**91.8**	**82.9**	**81.9**	91.6	93.0	97.1	**76.7**	92.5	71.7	96.2	94.9	**94.2**	**93.7**	75.3	**93.3**	80.0	**94.7**	85.4	01-Mar-2018
R4D_faster_rcnn [?]	88.6	94.6	92.3	91.3	82.3	79.4	**91.8**	91.8	97.4	76.6	93.6	**75.3**	97.0	94.6	93.5	92.6	75.1	92.0	80.9	94.4	**86.5**	20-Nov-2016
R-FCN, ResNet Ensemble(VOC+COCO) [?]	88.4	94.8	92.9	90.6	82.4	81.8	89.9	91.7	97.1	76.0	93.4	71.9	96.6	94.3	93.9	92.8	**75.7**	91.9	80.8	93.6	86.4	09-Oct-2016
HIK_FRCN [?]	87.9	95.0	93.2	91.3	80.3	77.7	90.6	89.9	**97.8**	72.8	**93.7**	70.7	**97.2**	**95.4**	94.0	91.8	72.7	92.8	81.1	94.1	86.2	19-Sep-2016
** VIM_SSD ** [?]	87.6	**95.3**	92.0	88.7	81.6	78.5	91.4	**93.2**	95.7	74.9	91.6	73.5	94.2	93.0	93.2	93.0	70.5	93.0	79.1	94.3	85.0	11-May-2018
** Deformable R-FCN, ResNet-101 (VOC+COCO) ** [?]	87.1	94.0	91.7	88.5	79.4	78.0	89.7	90.8	96.9	74.2	93.1	71.3	95.9	94.8	93.2	92.5	71.7	91.8	78.3	93.2	83.3	23-Mar-2017
RefineDet (VOC+COCO,single model,VGG16,one-stage) [?]	86.8	94.7	91.5	88.8	80.4	77.6	90.4	92.3	95.6	72.5	91.6	69.9	93.9	93.5	92.4	92.6	68.8	92.4	78.5	93.6	85.2	16-Mar-2018
FasterRcnn-ResNeXt101(COCO+07++12, single model) [?]	86.8	93.9	**93.4**	88.3	80.2	72.6	89.4	89.3	96.8	73.0	91.5	72.3	95.4	94.5	93.8	91.7	70.7	90.6	**81.2**	92.6	83.9	04-May-2017
** PSSNet(VOC+COCO) ** [?]	85.5	92.4	91.4	85.9	78.6	75.8	88.0	89.8	95.2	72.4	87.8	72.2	94.0	92.7	93.2	92.3	70.7	88.8	76.1	92.1	81.2	30-Mar-2018
R-FCN, ResNet (VOC+COCO) [?]	85.0	92.3	89.9	86.7	74.7	75.2	86.7	89.0	95.8	70.2	90.4	66.5	95.0	93.2	92.1	91.1	71.0	89.7	76.0	92.0	83.4	09-Oct-2016
** MONet(VOC+COCO) ** [?]	84.3	92.4	90.5	84.7	75.4	71.6	87.2	88.9	94.6	70.5	86.9	71.0	92.3	91.8	90.8	91.7	69.8	89.1	75.1	91.3	79.6	01-Apr-2018
PVANet+ [?]	84.2	93.5	89.8	84.1	75.6	69.7	88.2	87.9	93.4	70.0	87.7	**75.3**	92.9	90.5	90.9	90.2	67.3	86.4	80.3	92.0	78.8	26-Oct-2016
FSSD512 [?]	84.2	92.8	90.0	86.2	75.9	67.7	88.9	89.0	95.0	68.8	90.9	68.7	92.8	92.1	91.4	90.2	63.1	90.1	76.9	91.5	82.7	07-Nov-2017
BlitzNet512 [?]	83.8	93.1	89.4	84.7	75.5	65.0	86.6	87.4	94.5	69.9	88.8	71.7	92.5	91.6	91.1	88.9	61.2	90.4	79.2	91.8	83.0	19-Jul-2017
PFPNet512 VGG16 07++12+COCO [?]	83.8	93.0	89.9	85.1	75.8	66.4	88.4	88.3	94.0	67.9	89.5	69.7	92.0	91.8	91.6	88.7	61.1	89.1	78.4	90.5	84.3	18-Oct-2017
Faster RCNN, ResNet (VOC+COCO) [?]	83.8	92.1	88.4	84.8	75.9	71.4	86.3	87.8	94.2	66.8	89.4	69.2	93.9	91.9	90.9	89.6	67.9	88.2	76.8	90.3	80.0	10-Dec-2015

Common Objects in COntext (COCO)

80 objects

117,261 train/val images

902,435 object instances

New detection metric, mAP averaged over IOU [.5 - .95]

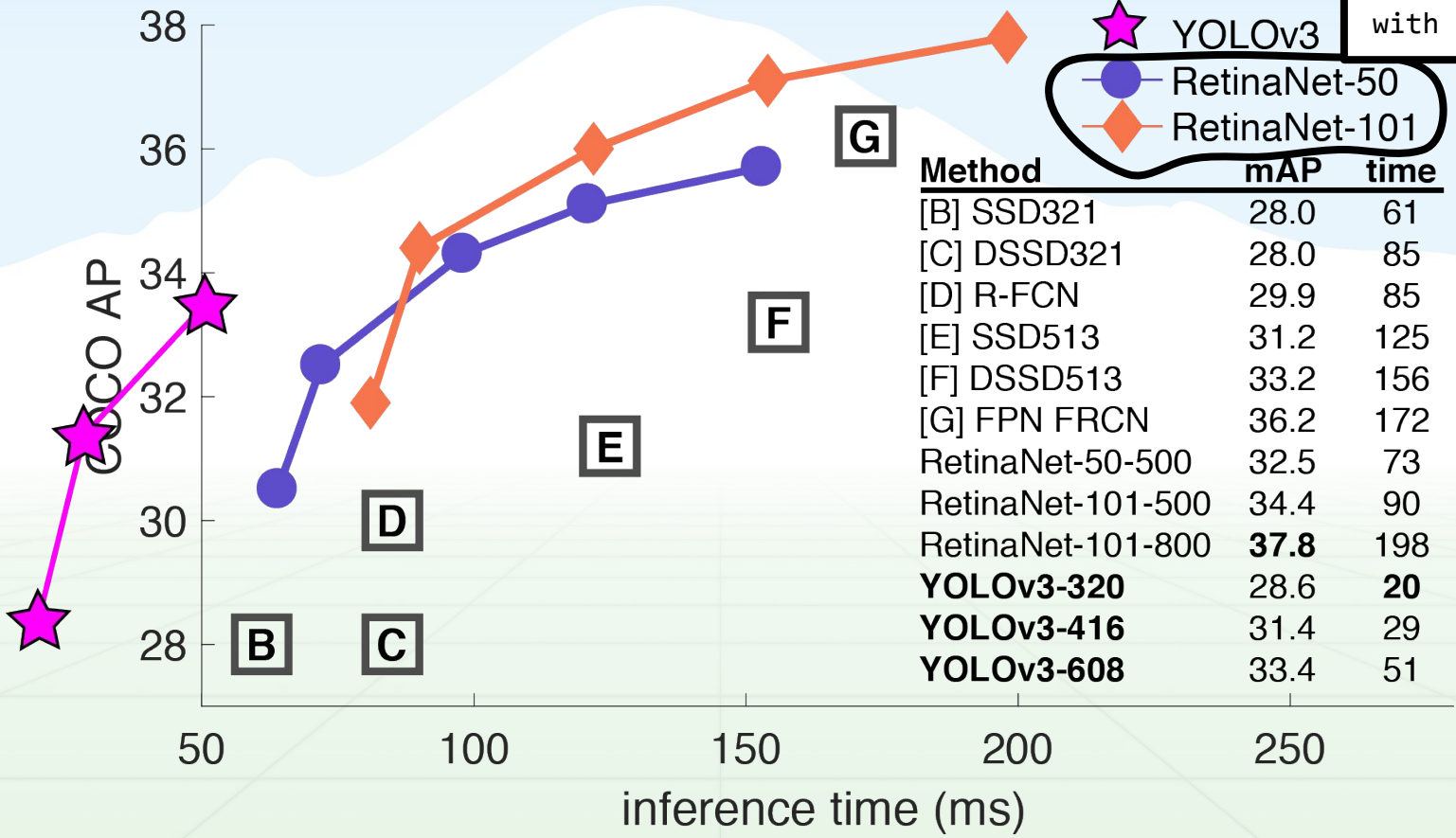
Segmentation masks for each instance

Originally by Microsoft but they were scared of copyright something something so they spun it off

http://cocodataset.org/#home

Performance on COCO

These are Faster R-CNN with new loss

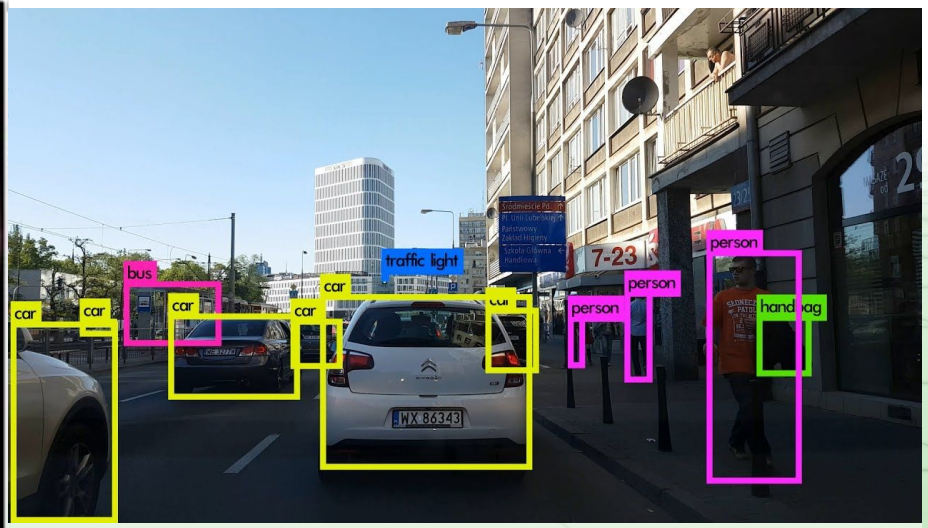


Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.6	**20**
YOLOv3-416	31.4	29
YOLOv3-608	33.4	51

Segmentation vs Detection

Pixel-level labels
Category only

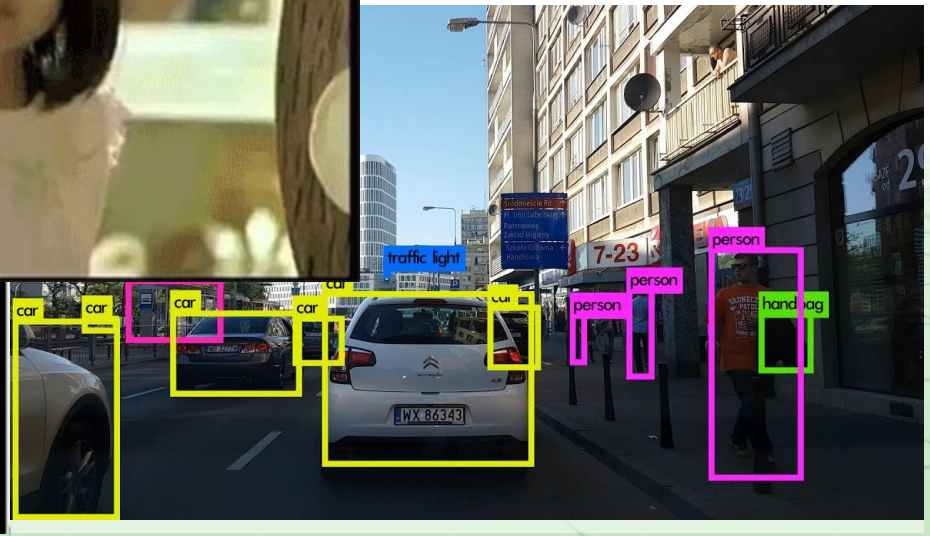
Bounding box labels
Category + instance



Segmentation vs Detection

Pixel-level labels
Category only

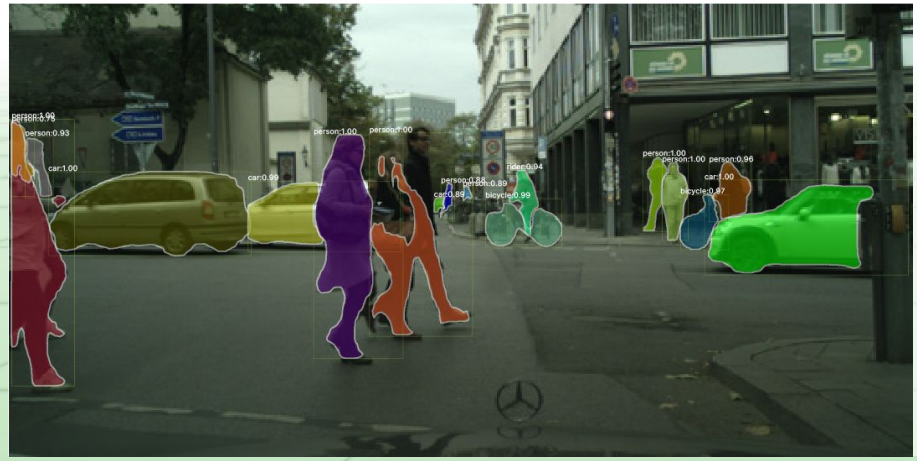
Bounding box labels
Category + instance

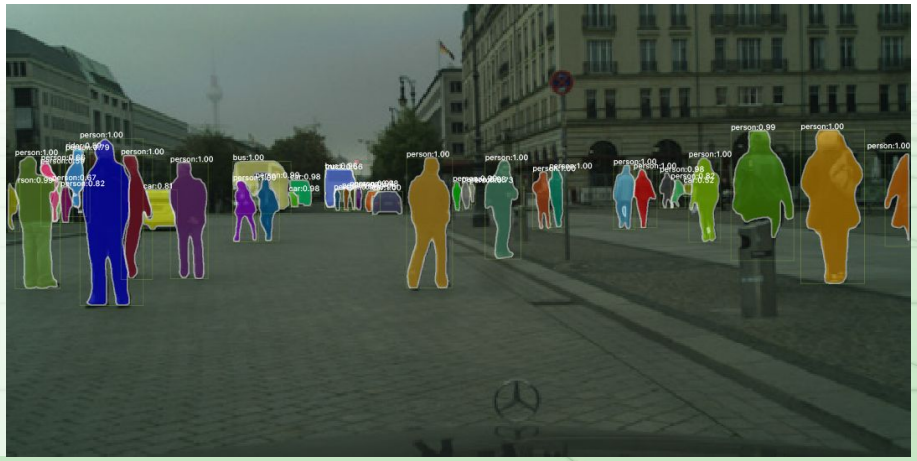


Instance Segmentation

Given an image produce instance-level segmentation

- Which class does each pixel belong to
- Also which instance

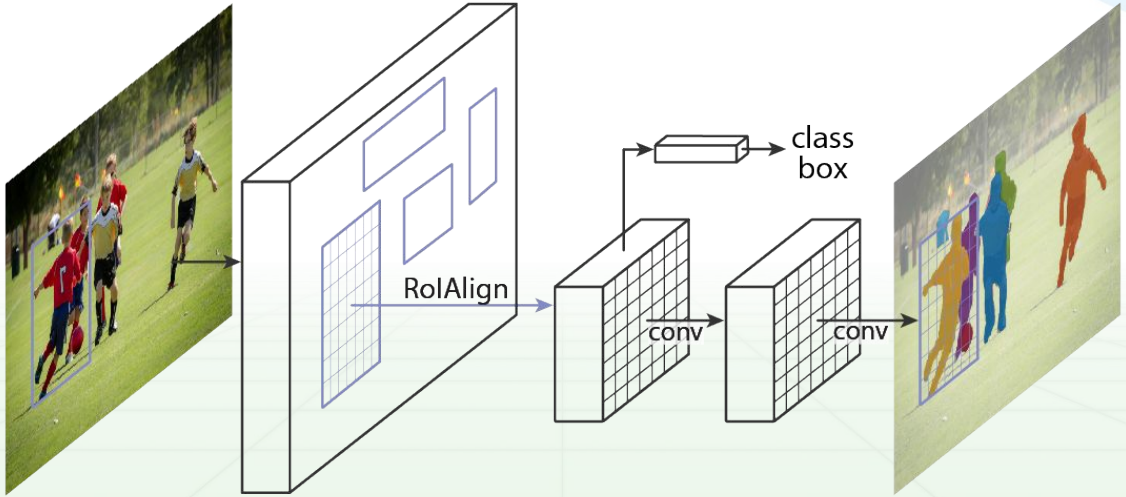




Mask R-CNN Demo

Mask R-CNN

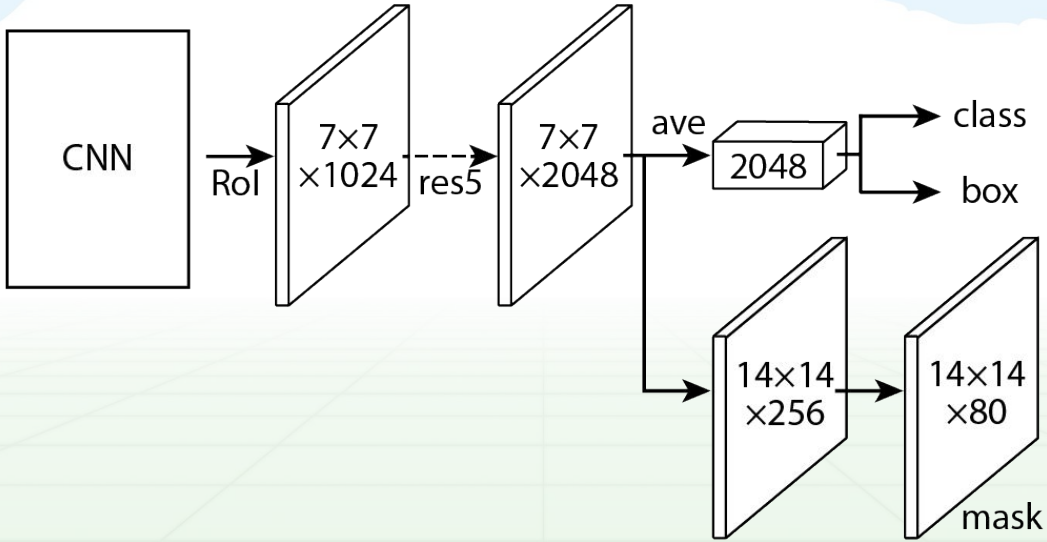
Similar to R-CNN but predict mask instead of box



https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN

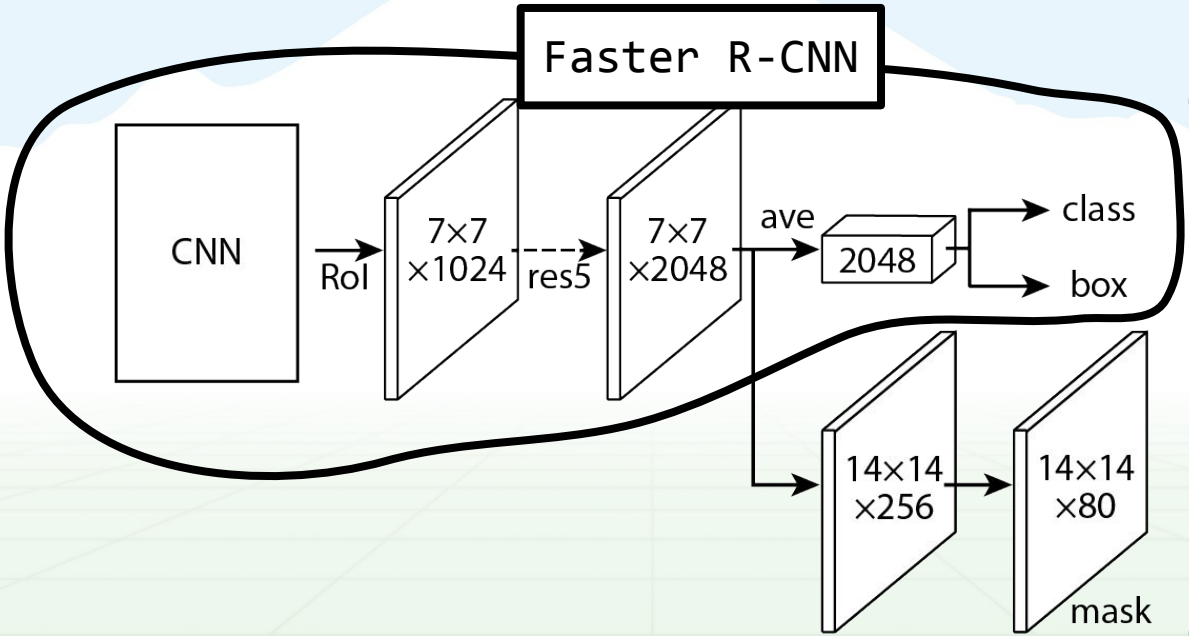
Similar to R-CNN but predict mask *as well as* box



https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN

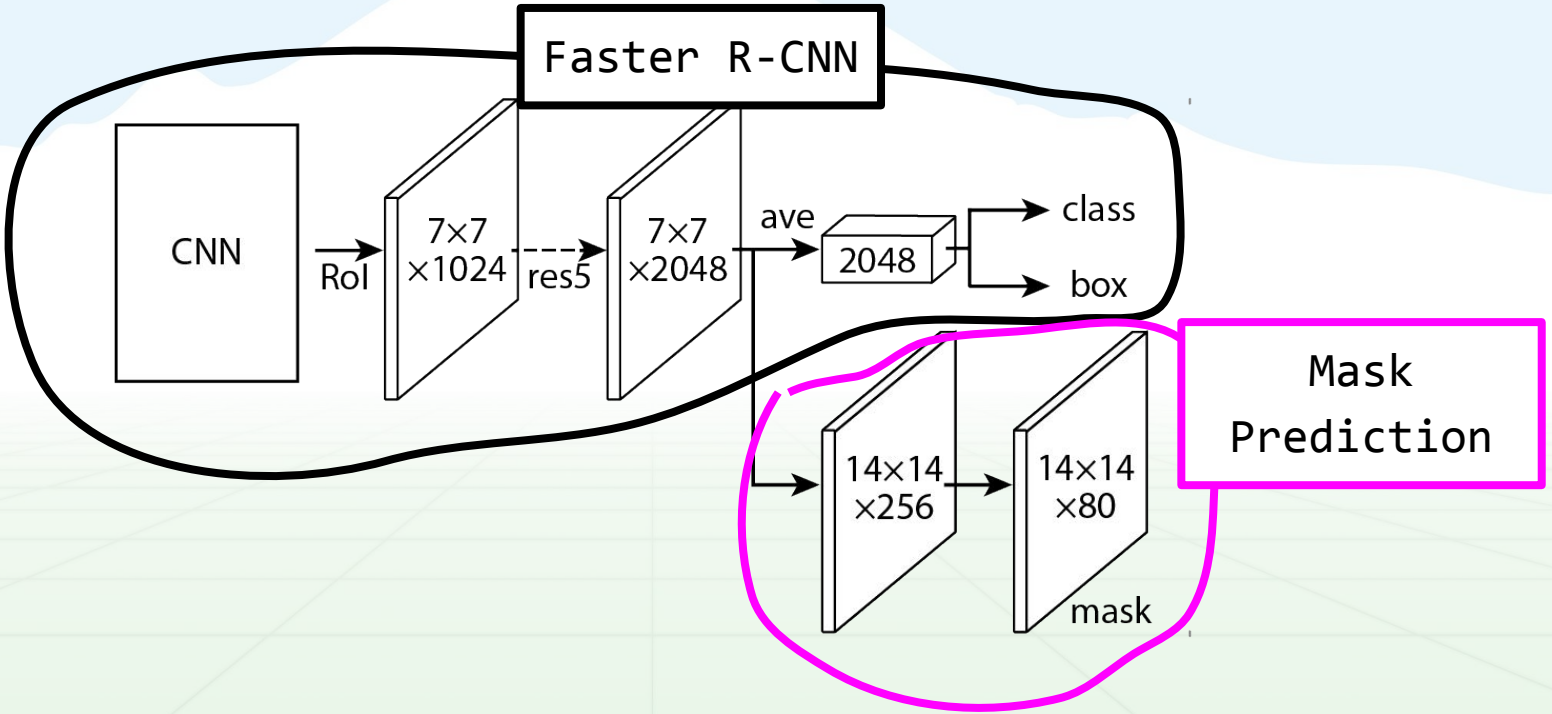
Similar to R-CNN but predict mask *as well as* box



https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN

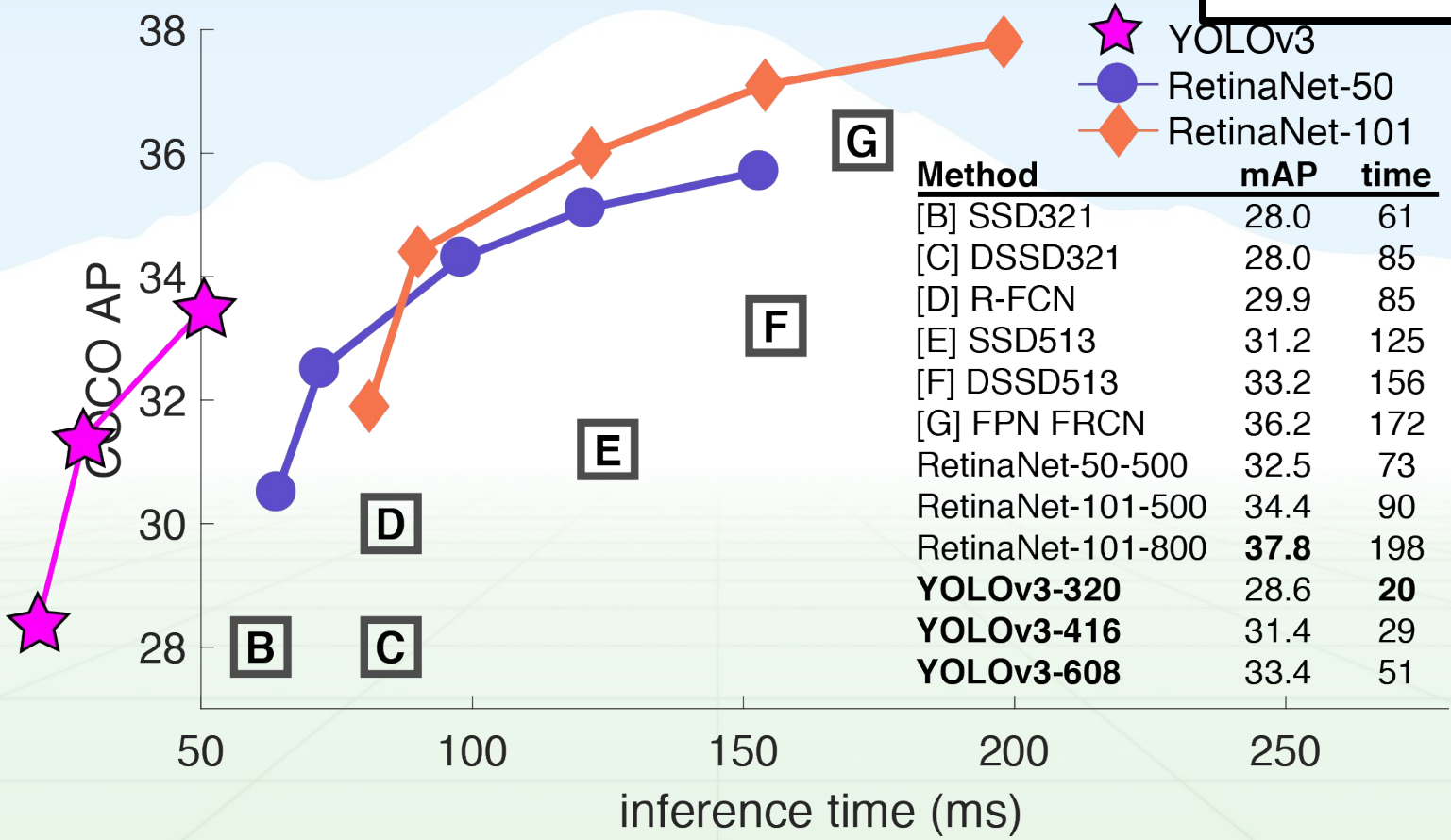
Similar to R-CNN but predict mask *as well as* box



https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN on COCO: 41 mAP

Mask R-CNN is over here somewhere



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.6	**20**
YOLOv3-416	31.4	29
YOLOv3-608	33.4	51

Segmentation, Detection, I. Seg.

COCO dataset also has captions!

5 captions per image

Detection/segmentation is (maybe) just pattern matching

To caption an image maybe you really have to *understand* it

Need to model both visual information and *language*

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

A horse carrying a large load of hay and two people sitting on it.

Bunk bed with a narrow shelf sitting underneath it.

Natural Language Processing using neural networks

Images are static, language has a component of time

Characters/words appear in sequence, need to read previous words to understand subsequent ones (kind of like frames in a video)

How to we process time-series data using neural networks?

Next time! Recurrent neural networks