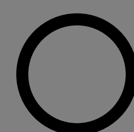
Object detection II

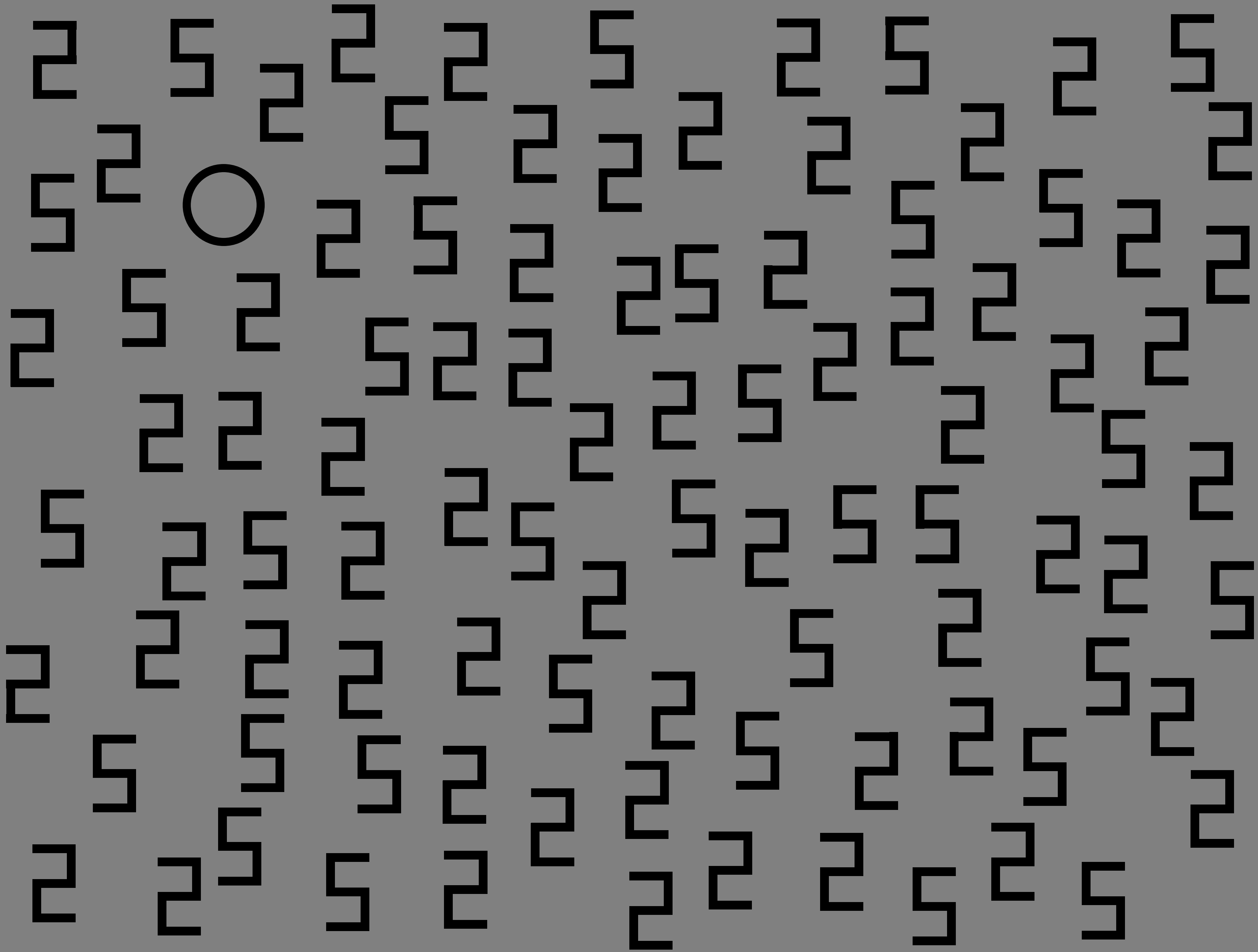
Semester 2, 2021

Kris Ehinger

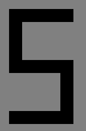
Find this:

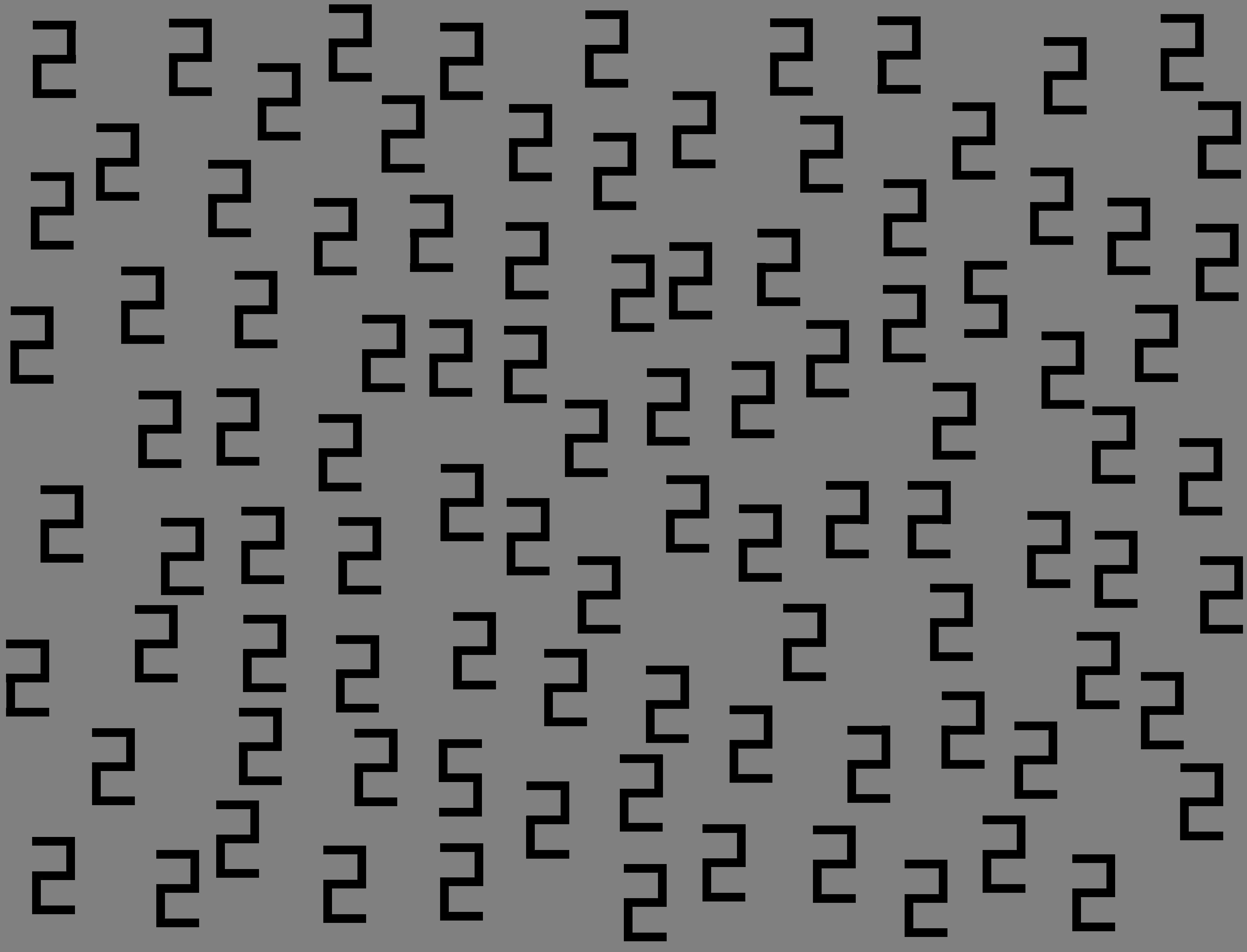
Find this:



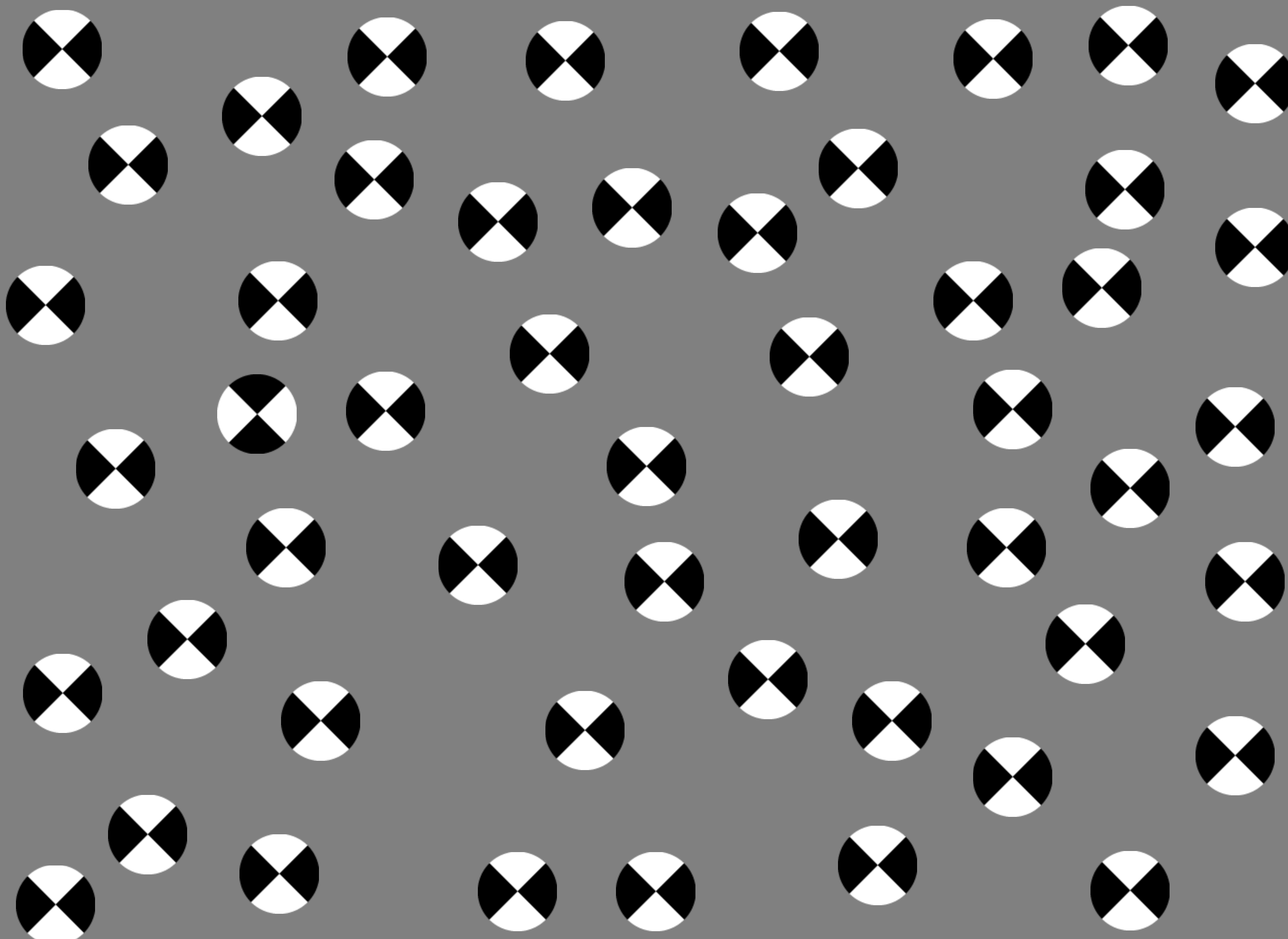


Find this:





Find this:



Outline

- Single-stage object detectors
- Instance segmentation
- Evaluating object detectors
- Beyond patches?

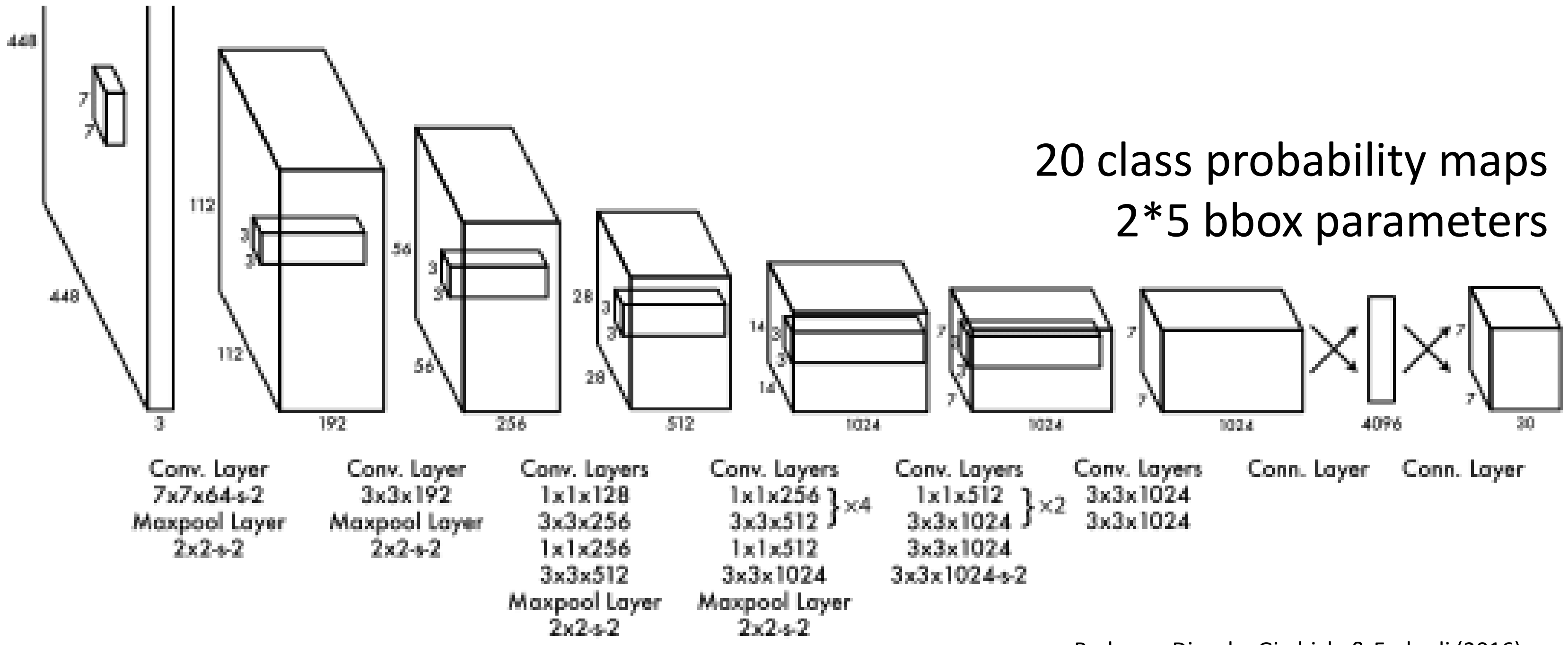
Learning outcomes

- Explain how single-stage object detectors differ from two-stage methods like Faster R-CNN
- Implement algorithms to do non-maximum suppression (NMS) and compute mAP
- Compare and contrast various approaches to object detection

Single-stage object detectors

YOLO (You Only Look Once) v1

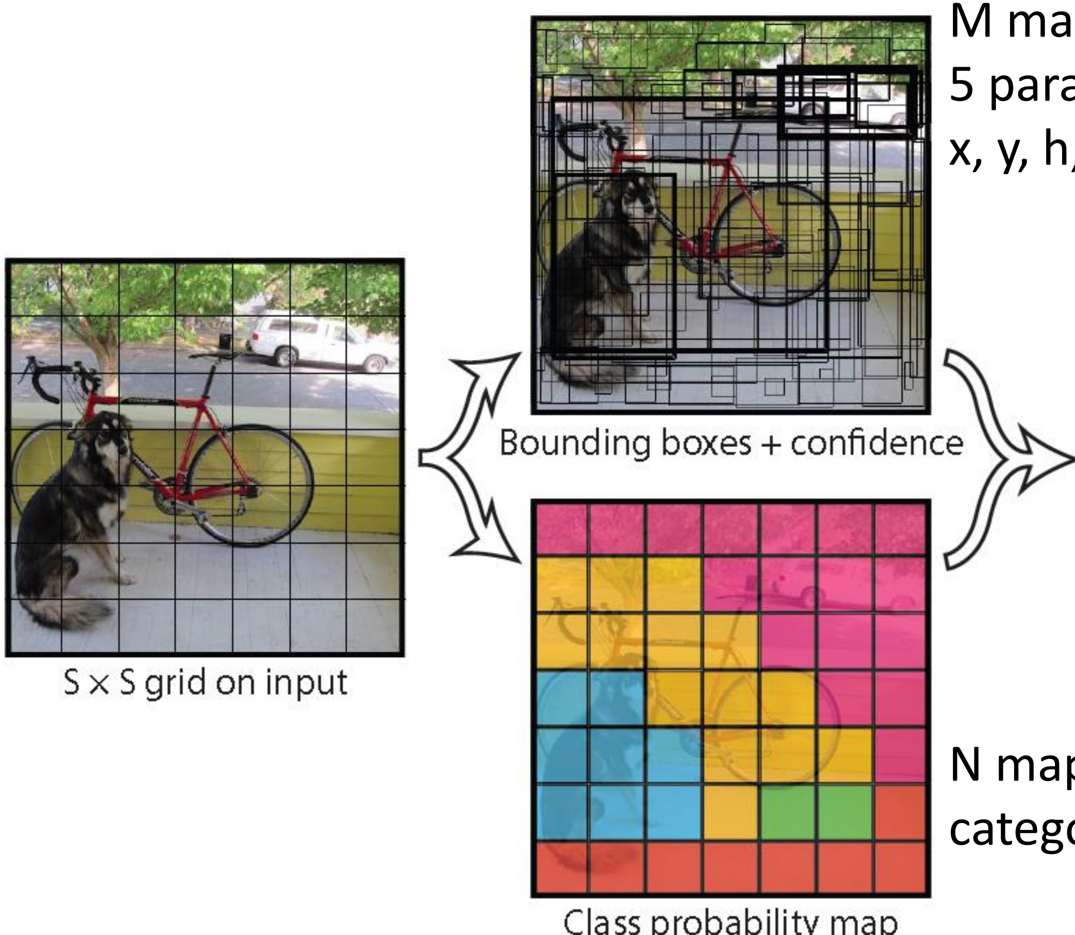
- Main idea: instead of going through multiple steps (region proposals, region classification), just predict a heatmap for each class directly in a CNN



20 class probability maps
2*5 bbox parameters

Redmon, Divvala, Girshick, & Farhadi (2016)

YOLO (You Only Look Once) v1



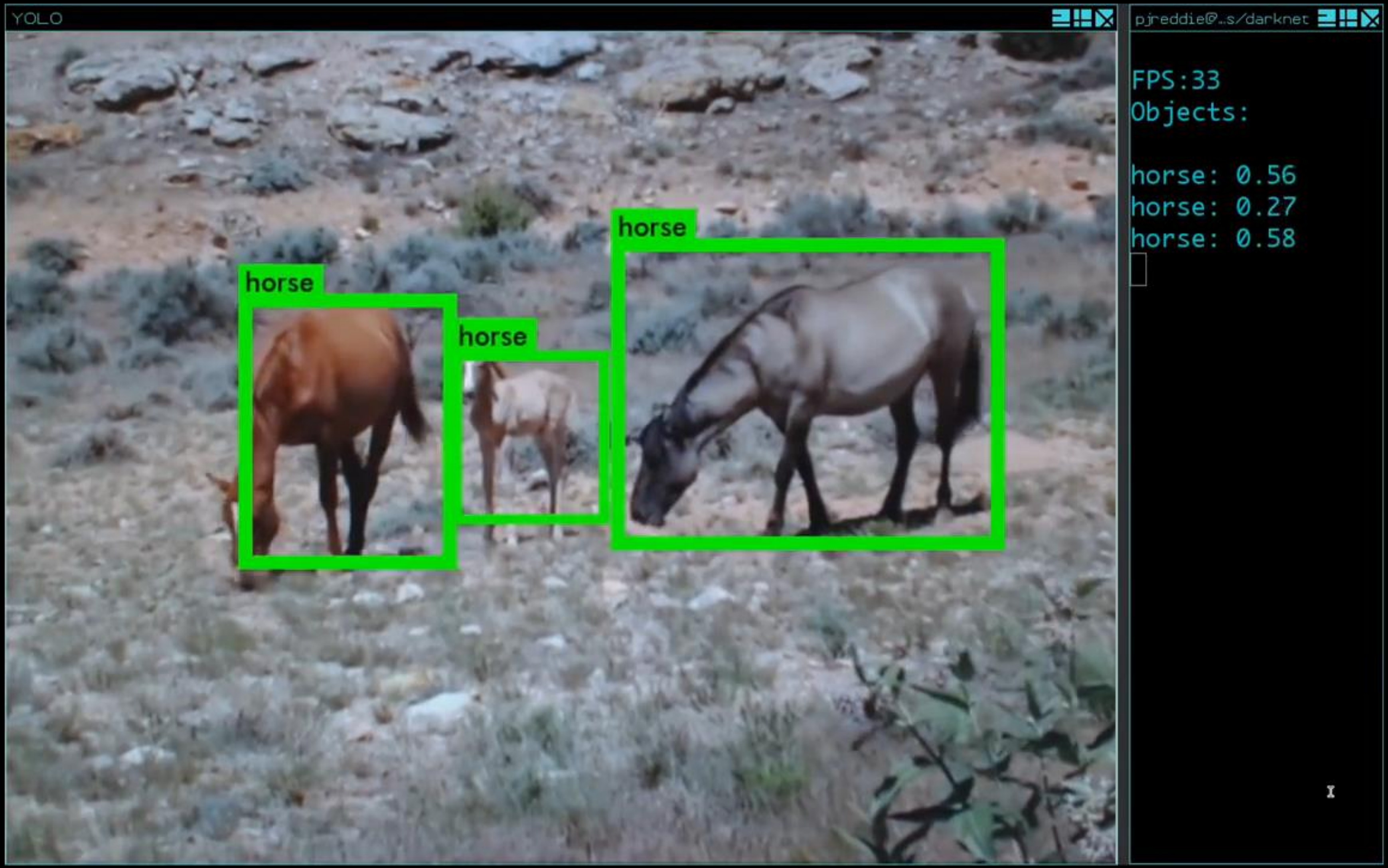
M maps for M bboxes
5 parameters per bbox:
x, y, h, w, confidence

N maps for N object categories

Redmon, Divvala, Girshick, & Farhadi (2016)

YOLO (You Only Look Once) v1

- Output is a set of N class probability maps + M bounding box parameter maps
- Loss is sum-squared error between true and predicted maps, with some weighting:
  - Bbox location parameters get higher weight in the loss
  - Grid cells that don’t contain objects don’t contribute to classification loss
  - Bbox parameters are penalised based on their confidence, encouraging the M bboxes to specialise for different objects



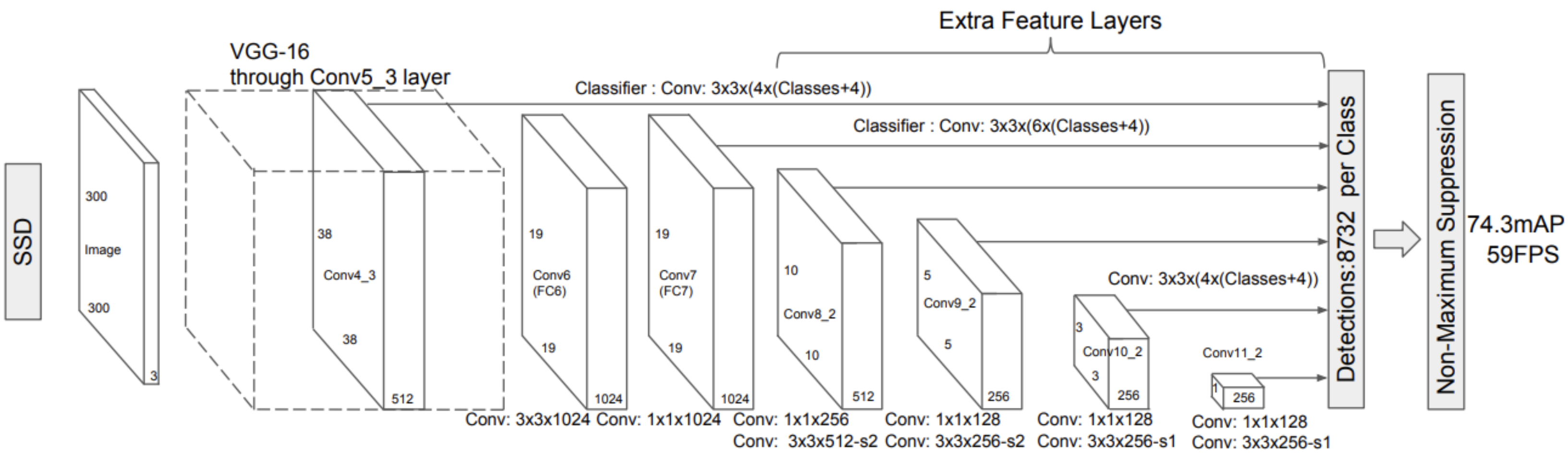
https://www.youtube.com/watch?v=K9a6mGNmhbc

YOLO (You Only Look Once) v1

- Advantages:
  - Fast
  - Accurate, for a real-time object detector
- Disadvantages:
  - Limited spatial precision
  - Generally less accurate than slower detectors
- (There have been multiple versions of this algorithm that have improved on the original method)

SSD: Single shot multibox detector

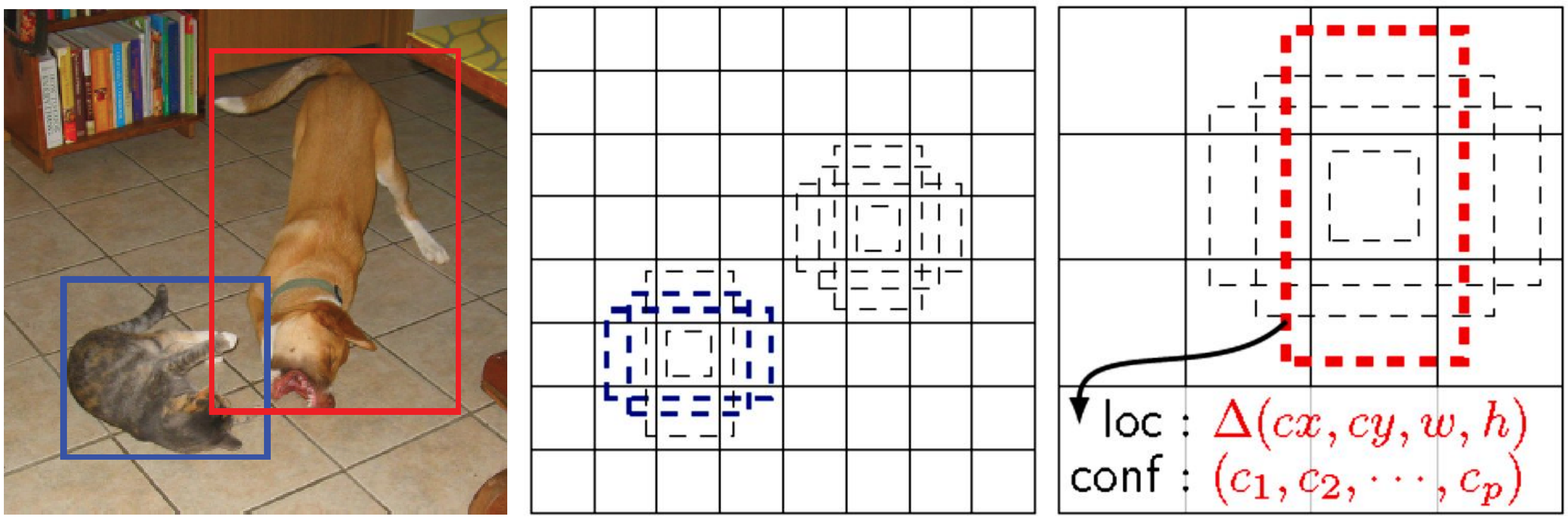
- Similar to YOLO: instead of generating region proposals, directly predict a set of class+bbox heatmaps
  - For each anchor point: k bboxes * (N class confidences * 4 bbox parameters)



Lie, Anguelov, Erhan, Szegedy, Reed, Fu, & Berg (2016)

SSD: Single shot multibox detector

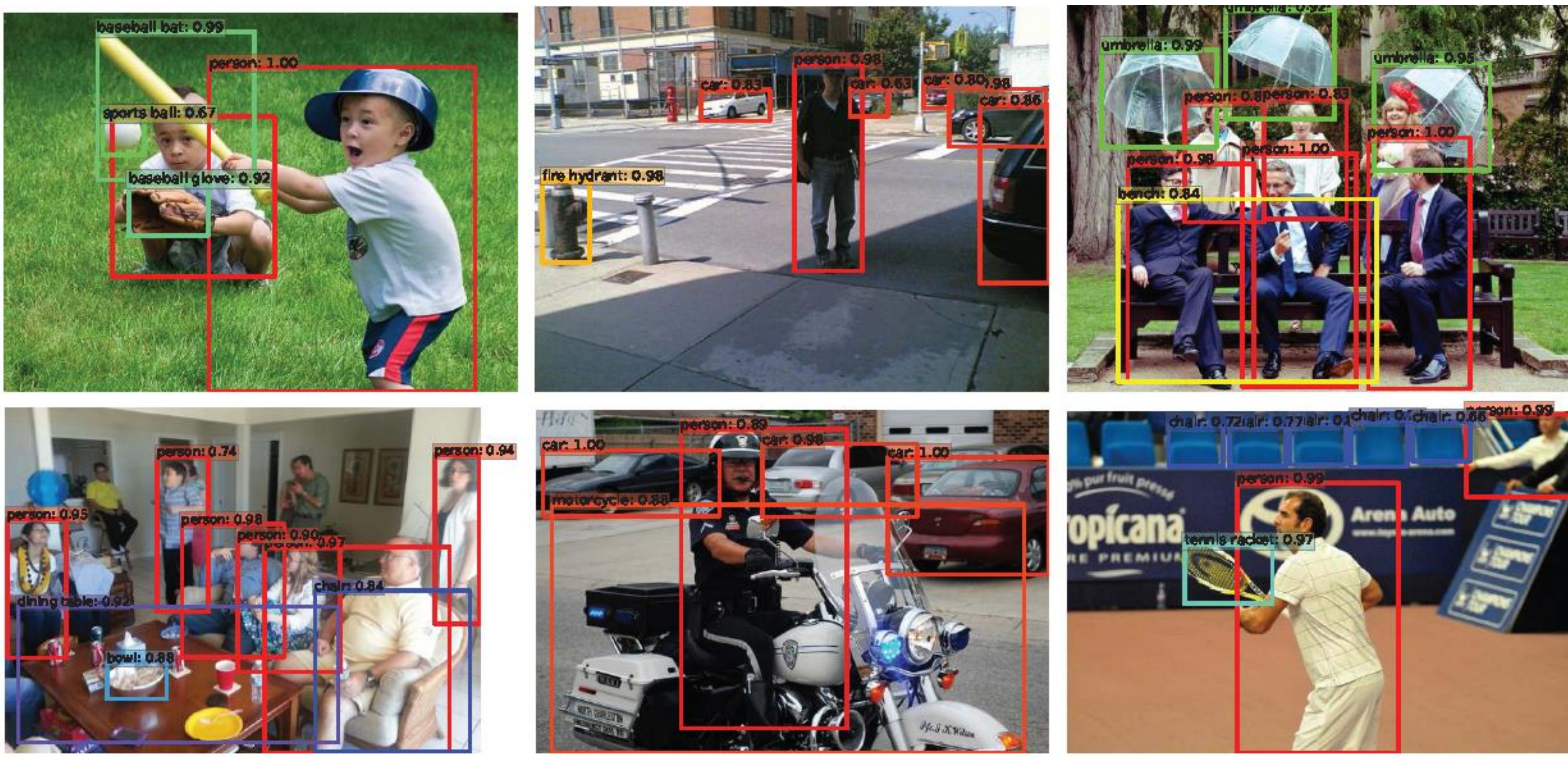
- Major change: anchor points in multiple convolutional layers, allowing for detection at different scales



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Lie, Anguelov, Erhan, Szegedy, Reed, Fu, & Berg (2016)

SSD: Single shot multibox detector



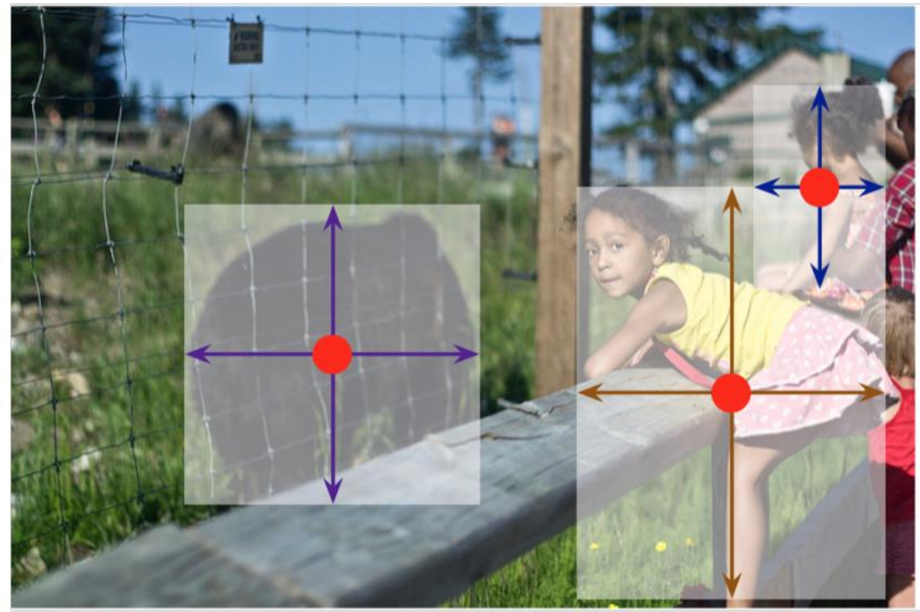
Lie, Anguelov, Erhan, Szegedy, Reed, Fu, & Berg (2016)

SSD: Single shot multibox detector

- Faster than region-proposal methods like Faster R-CNN
- Generally less accurate than region-proposal methods
- Anchor points in early layers helps with spatial prediction and detection of small objects

Alternatives to bounding boxes

CornerNet: predict 2 corner points



CenterNet: predict object's central point

Other options? Oriented bounding box, oriented ellipse, Gaussian distribution, oriented Gaussian?

Law & Deng (2018); Zhou, Wang, & Krahenbuhl (2019)

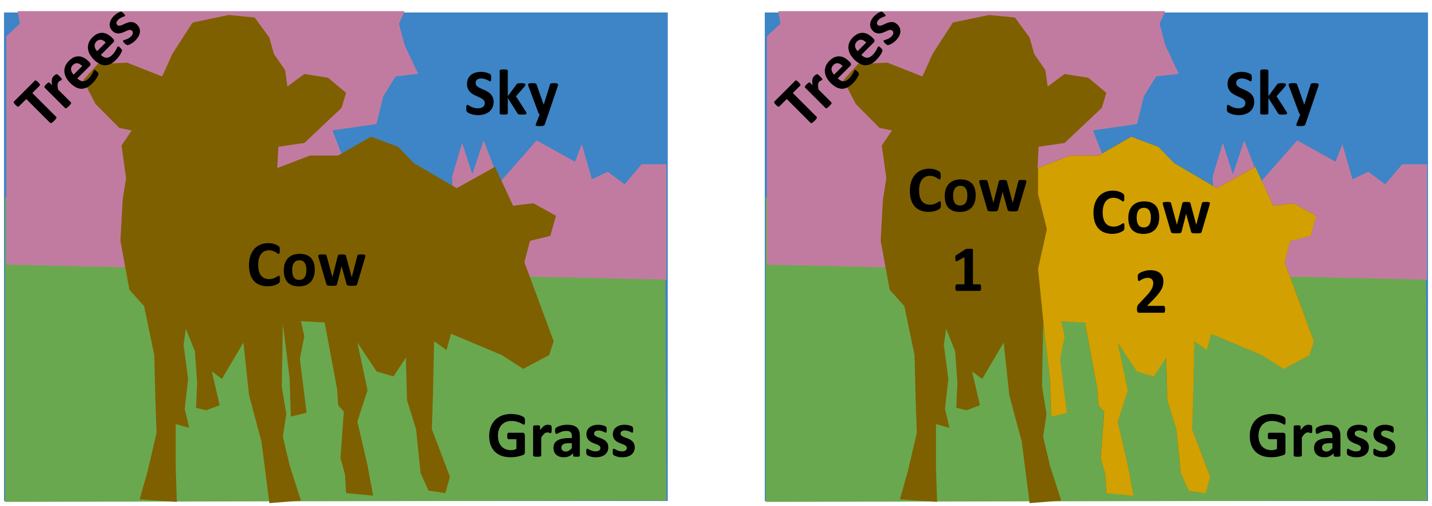
Summary

- Single-stage detectors skip the region proposal step and predict object classes/bounding boxes directly
- Single-stage methods tend to be faster but less accurate than two-stage methods like Faster R-CNN
- Some recent methods simplify the prediction by predicting single points instead of bounding boxes

Instance segmentation

Instance segmentation

- Semantic segmentation classifies pixels, doesn't distinguish between instances
- How to separate instances?



Instance segmentation

- Common method:
  - Run object detector, extract bounding boxes and labels
  - Do binary (foreground/background) segmentation within each bounding box
- Commonly-used architecture: Mask R-CNN

Faster R-CNN

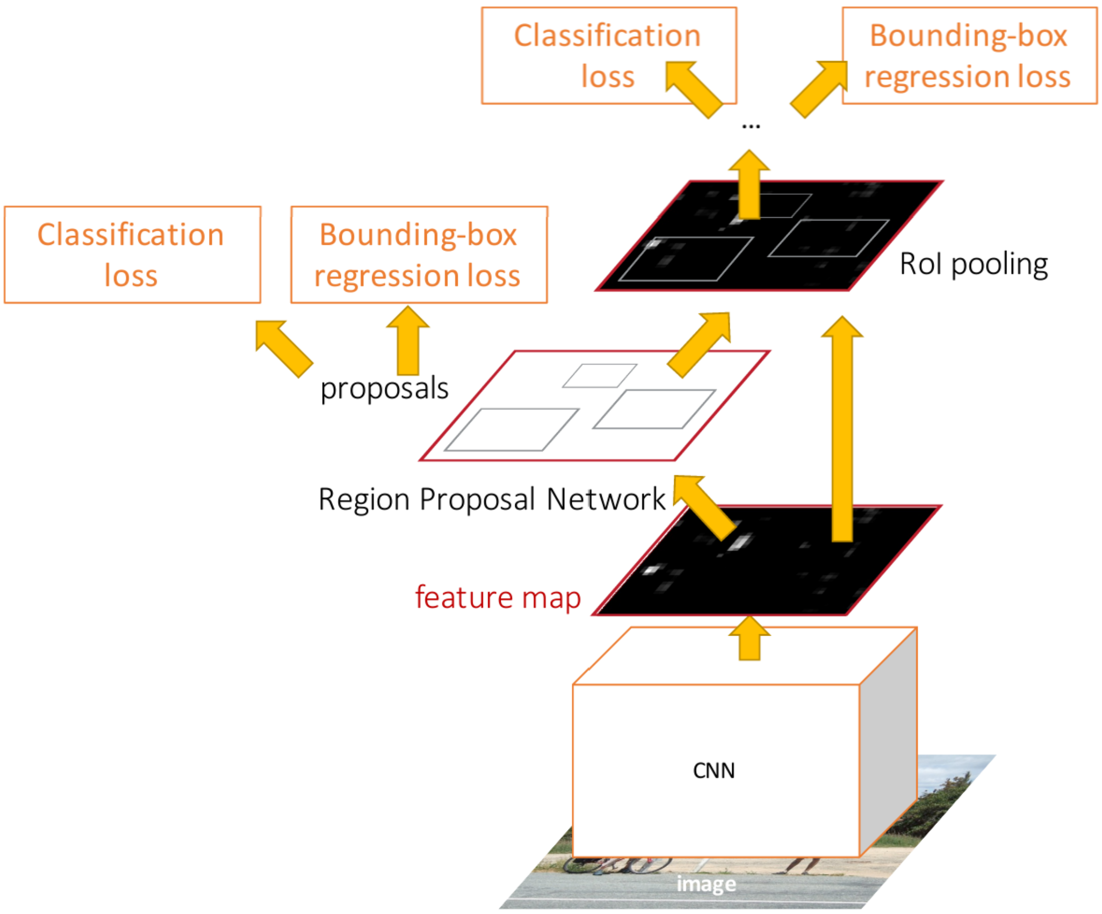
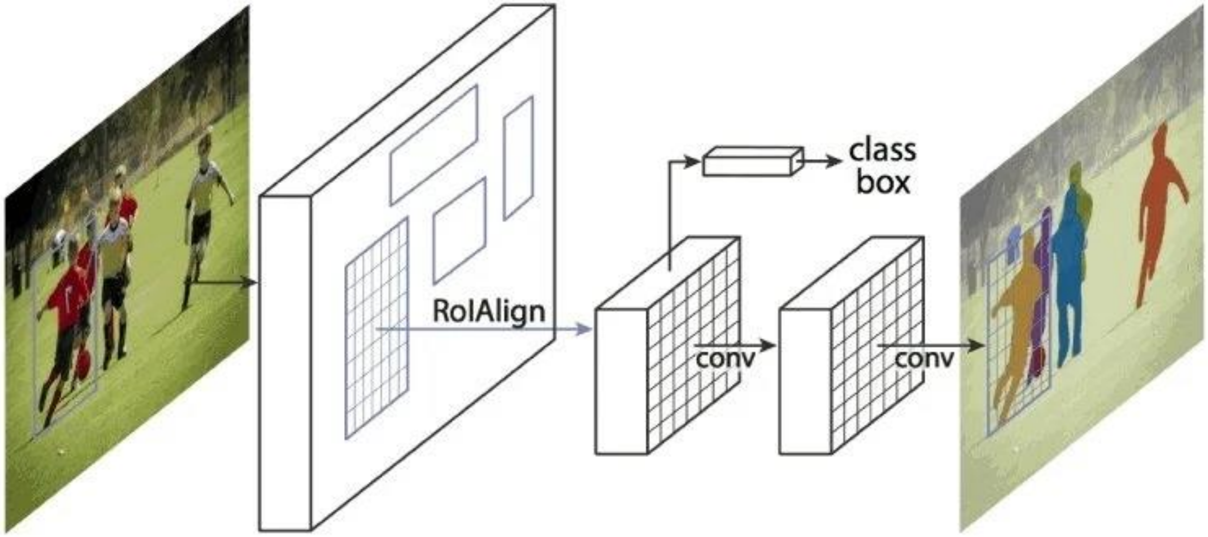


Figure: Ren, He, Girshick, & Sun (2015)

Mask R-CNN

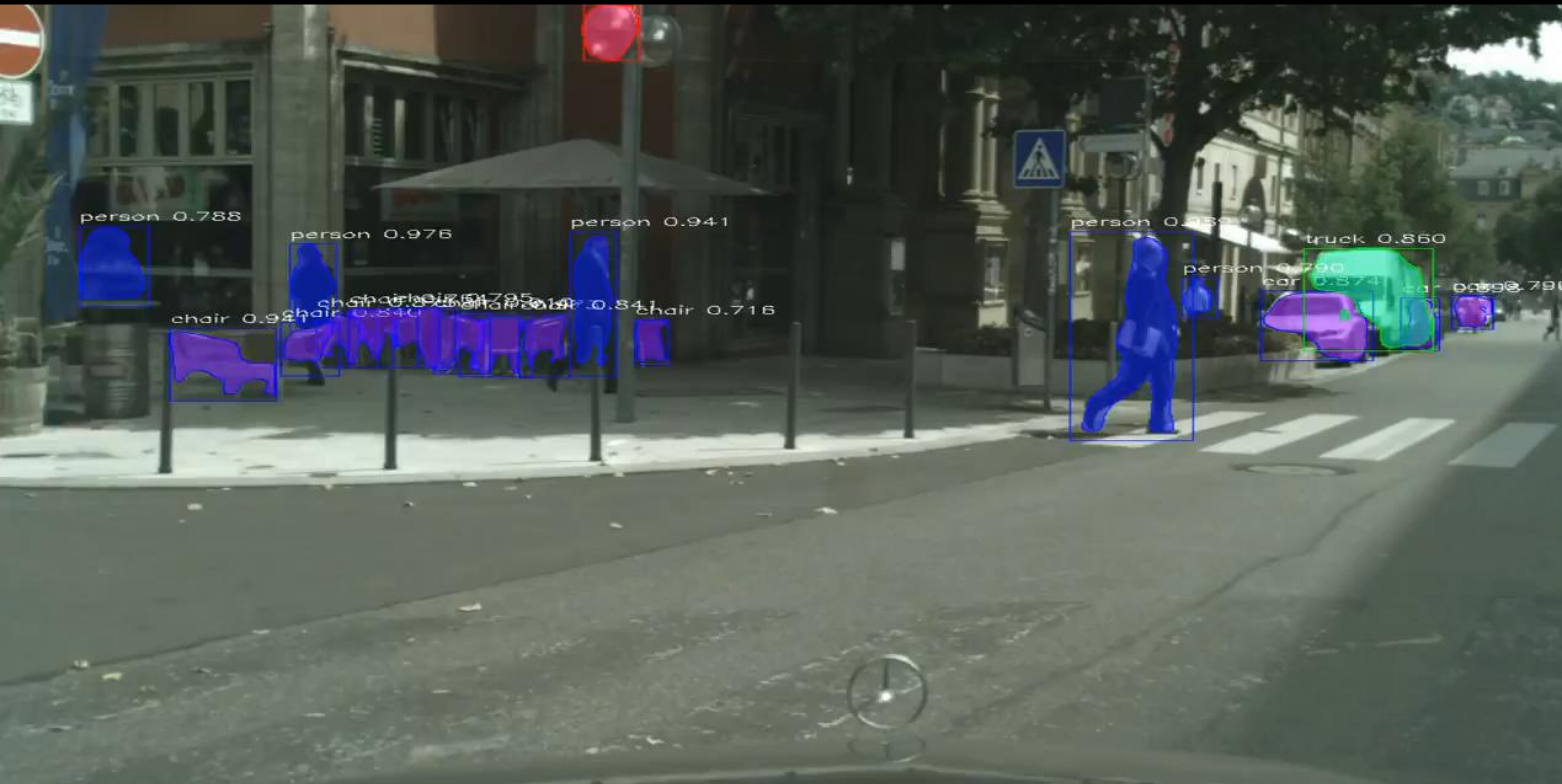
- Basically just an extra step on Faster R-CNN – each patch runs through a fully-convolutional network that predicts a binary segmentation mask
- Patch loss becomes: $L = L_{cls} + L_{box} + L_{mask}$



He, Gkioxari, Dollár, & Girshick (2017)

Mask R-CNN results

He, Gkioxari, Dollár, & Girshick (2017)



https://www.youtube.com/watch?v=EfaCtOEEmLA

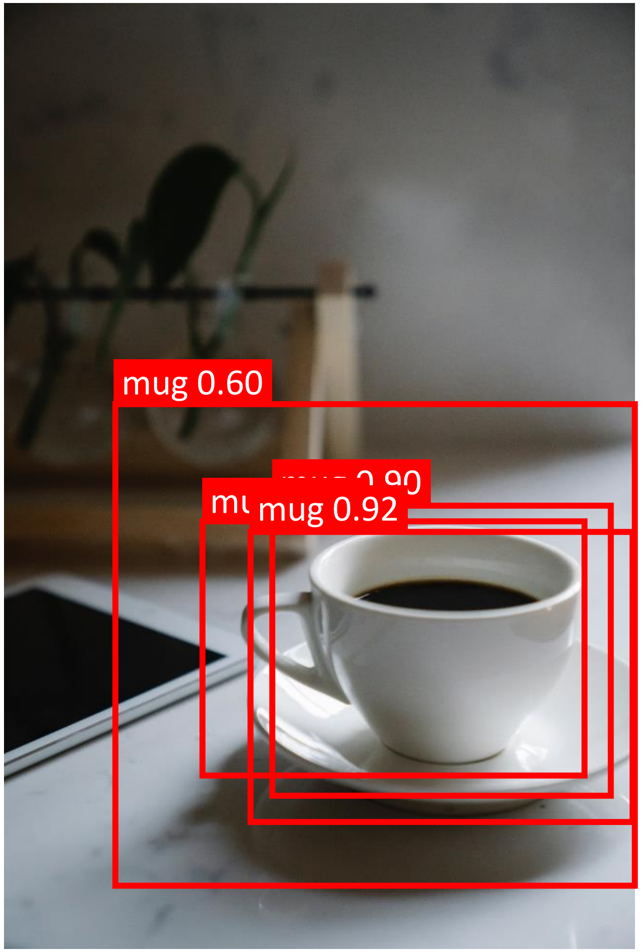
Summary

- Instance segmentation can be modelled as object detection followed by binary segmentation (foreground/background)
- Common architecture is Mask R-CNN, which is a modification of Faster R-CNN

Evaluating object detectors

Object detection result

- Typically, object detectors will return many overlapping detections
  - Can be different objects, or the same object detected at multiple scales / positions
- Treat as multiple detections? Or select one as the final prediction?



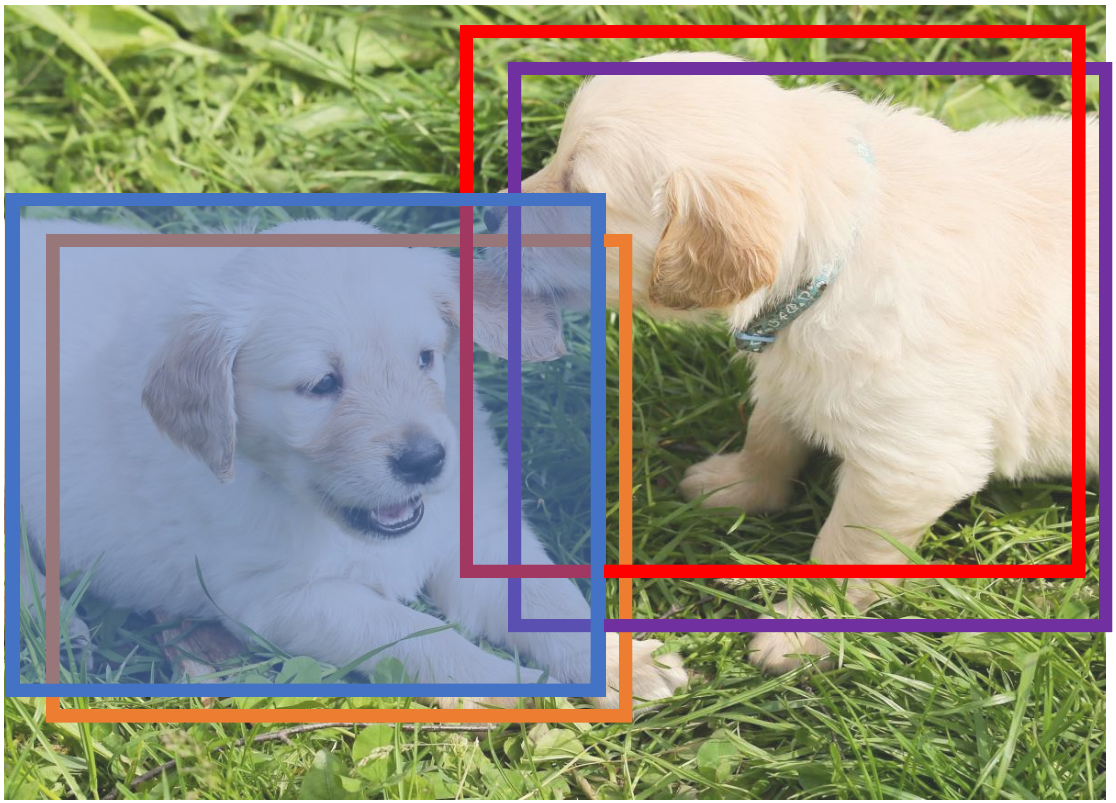
Non-max suppression (NMS)

- Typical approach: non-maximum suppression (NMS)
- Algorithm:
  - Starting with the highest-scoring bounding box…
  - Drop bounding boxes with lower score that overlap with this box above some IoU threshold (e.g., 0.7)
  - Repeat with next highest-scoring bounding box
- Often done separately within each object class

Non-max suppression (NMS)

P(dog) = 0.9

P(dog) = 0.8



IoU(■, ■) = 0.78

IoU(■, ■) = 0.05

IoU(■, ■) = 0.07

P(dog) = 0.7

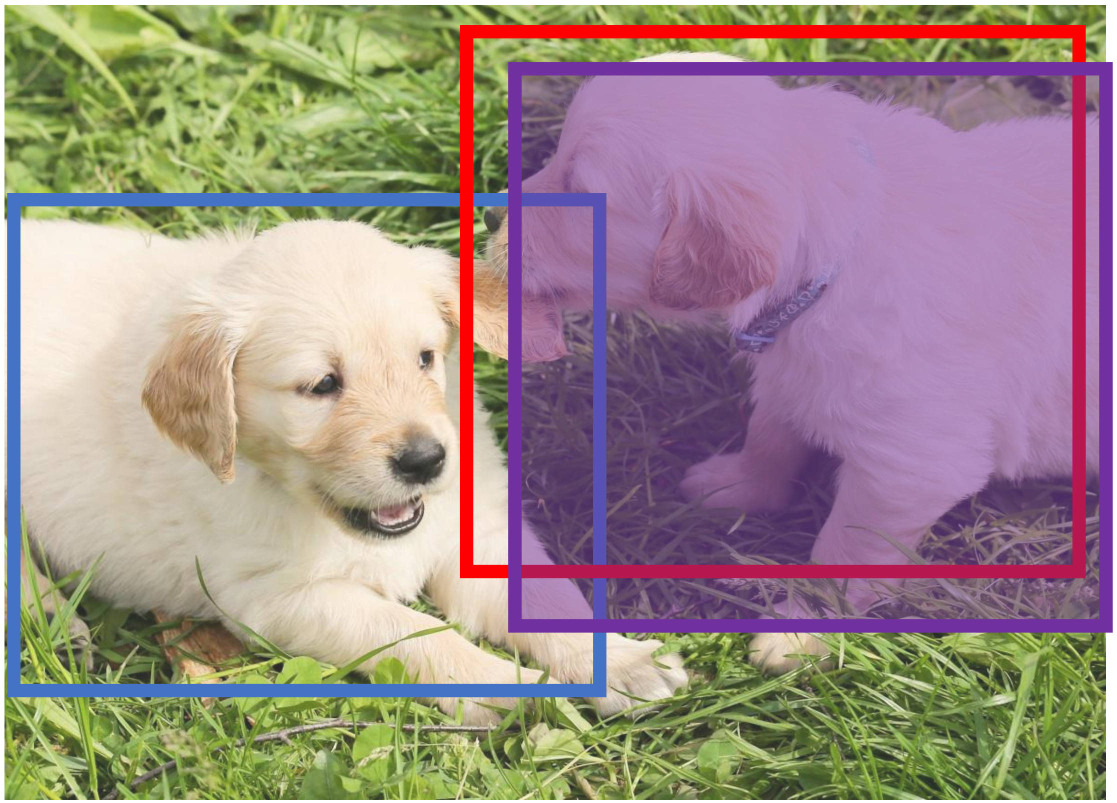
P(dog) = 0.75

Figure: J. Johnson

Non-max suppression (NMS)

IoU(■, ■) = 0.74

P(dog) = 0.9



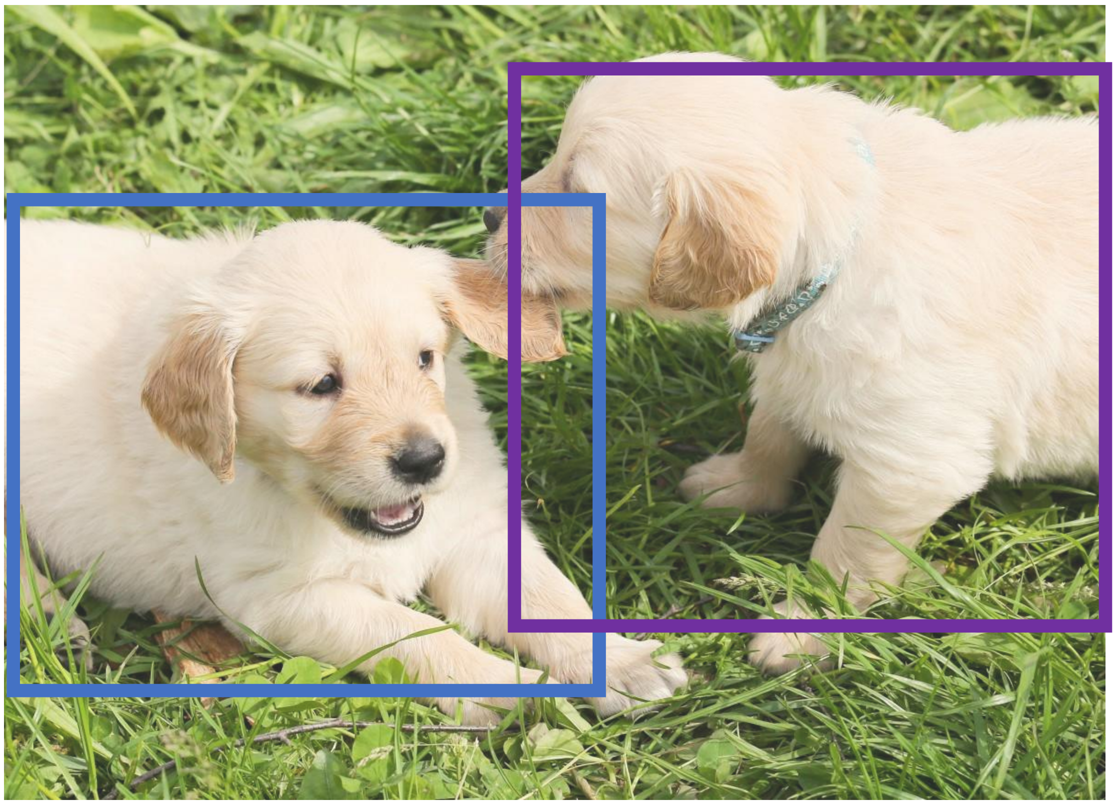
P(dog) = 0.7

P(dog) = 0.75

Figure: J. Johnson

Non-max suppression (NMS)

P(dog) = 0.9



P(dog) = 0.75

Figure: J. Johnson

Non-max suppression (NMS)

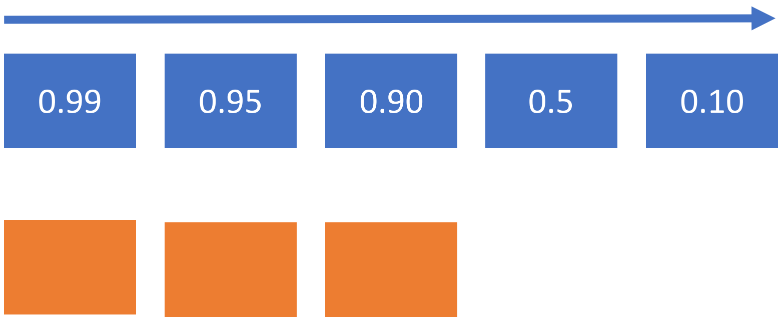
- NMS can drop some correct detections when objects are highly overlapping
- But generally this is preferable to counting the same object many times

Evaluation

- How to evaluate, given that there may be multiple objects/detections per image?
- Commonly-used method:
  - Run detection on entire test set
  - Run NMS to remove overlapping detections
  - For each object category, compute Average Precision (AP) = area under precision-recall (P-R) curve

Evaluation

All dog detections sorted by score



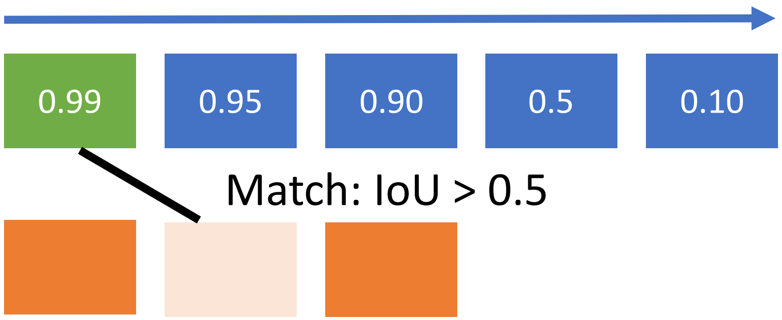
All ground-truth (GT) dog boxes

Sort detections from highest to lowest score

For each detection:

Evaluation

All dog detections sorted by score



All ground-truth (GT) dog boxes

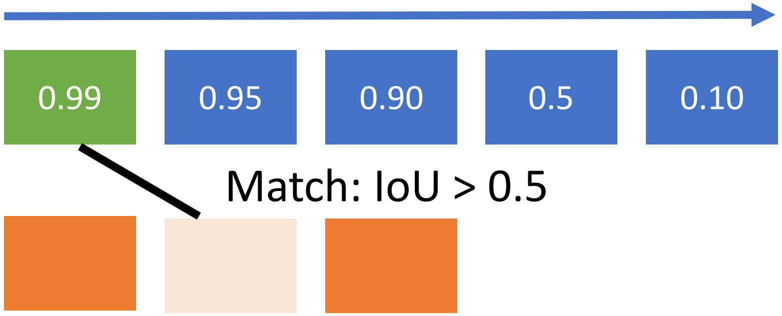
Sort detections from highest to lowest score

For each detection:

- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative

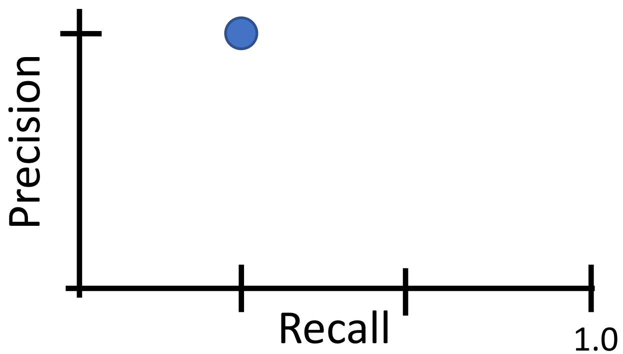
Evaluation

All dog detections sorted by score



All ground-truth (GT) dog boxes

Precision = 1/1 = 1.0
Recall = 1/3 = 0.33

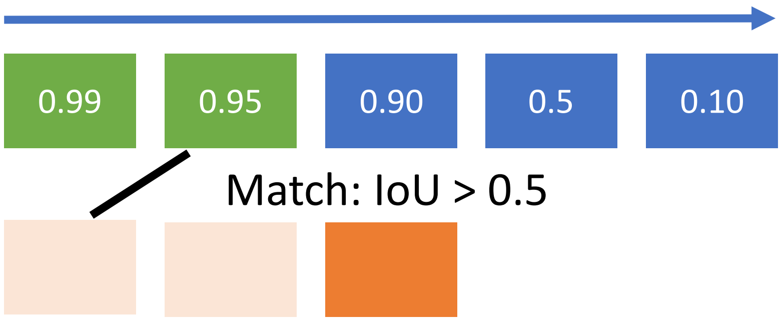


Sort detections from highest to lowest score
For each detection:

- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

Evaluation

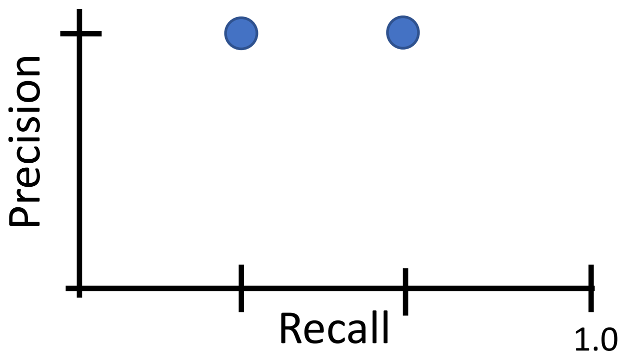
All dog detections sorted by score



All ground-truth (GT) dog boxes

Precision = 2/2 = 1.0

Recall = 2/3 = 0.67



Sort detections from highest to lowest score

For each detection:

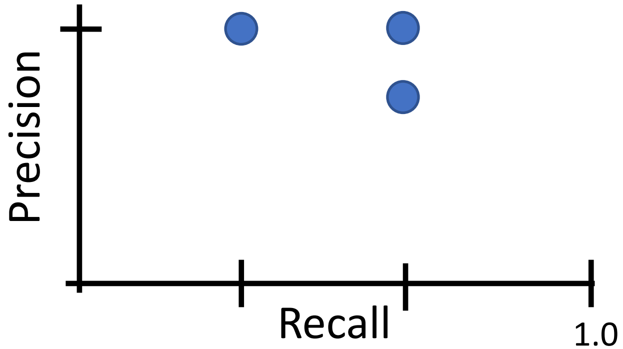
- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

Evaluation

All dog detections sorted by score

All ground-truth (GT) dog boxes

Precision = 2/3 = 0.67
Recall = 2/3 = 0.67



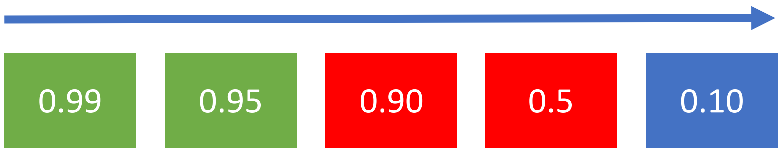
Sort detections from highest to lowest score

For each detection:

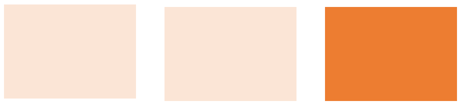
- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

Evaluation

All dog detections sorted by score

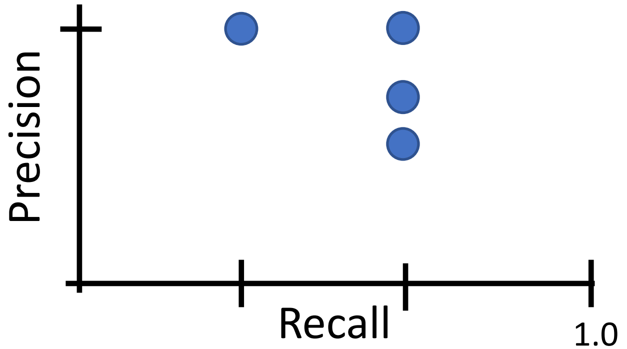


No match with IoU > 0.5



All ground-truth (GT) dog boxes

Precision = 2/4 = 0.5
Recall = 2/3 = 0.67



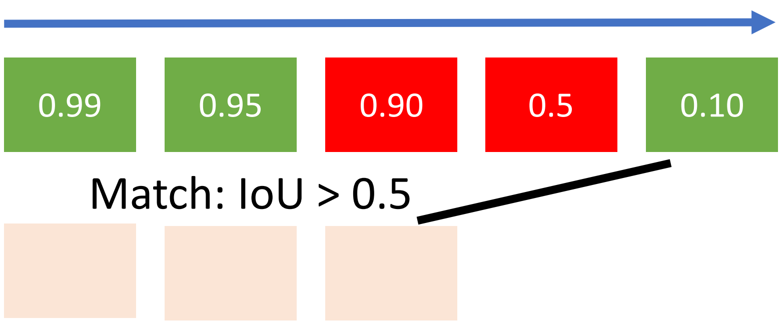
Sort detections from highest to lowest score

For each detection:

- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

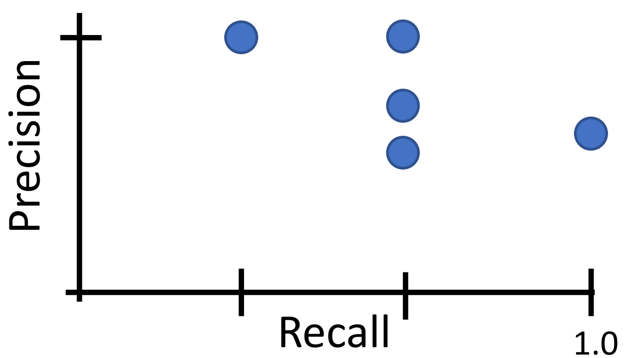
Evaluation

All dog detections sorted by score



All ground-truth (GT) dog boxes

Precision = 3/5 = 0.6
Recall = 3/3 = 1.0



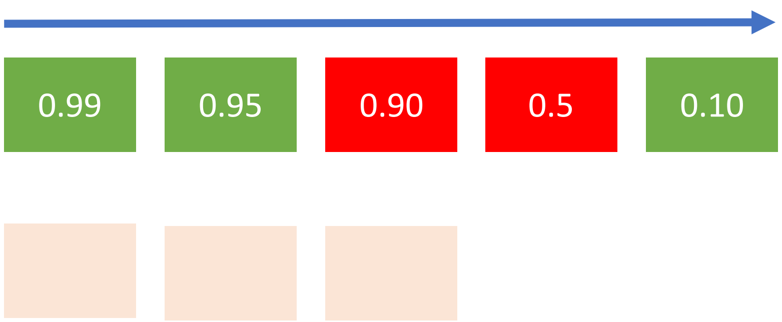
Sort detections from highest to lowest score

For each detection:

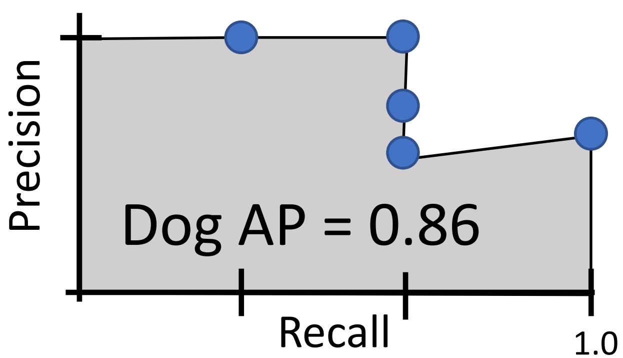
- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

Evaluation

All dog detections sorted by score



All ground-truth (GT) dog boxes



Sort detections from highest to lowest score

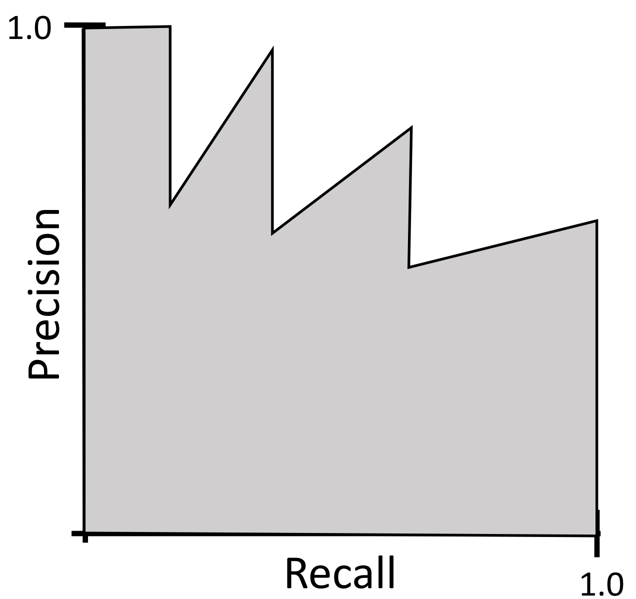
For each detection:

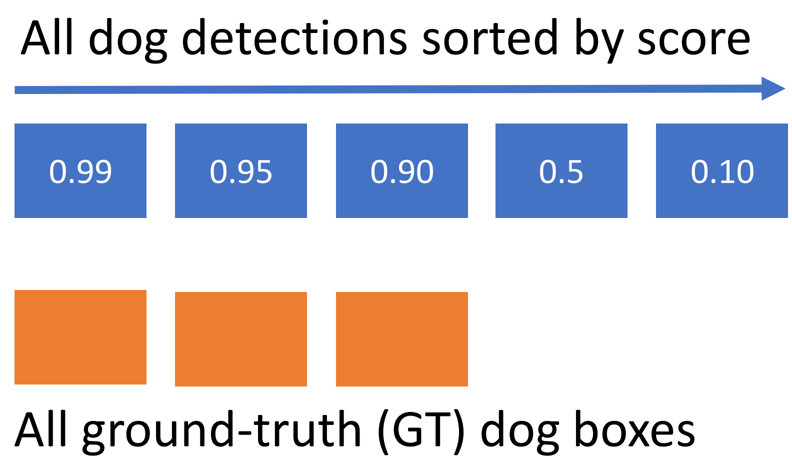
- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

Average Precision (AP) = area under PR curve

Properties of P-R curve

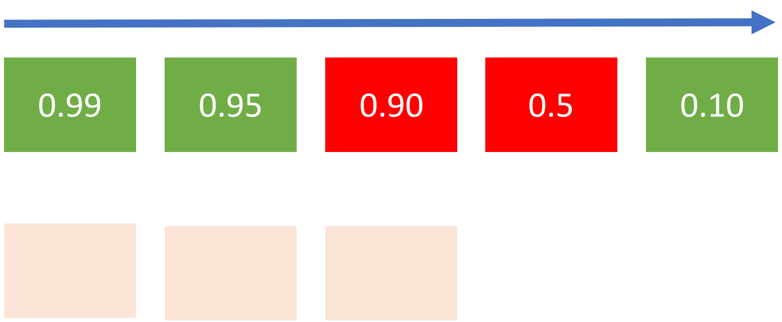
- What is the best possible AP (area under P-R curve)?
- How would you accomplish this?



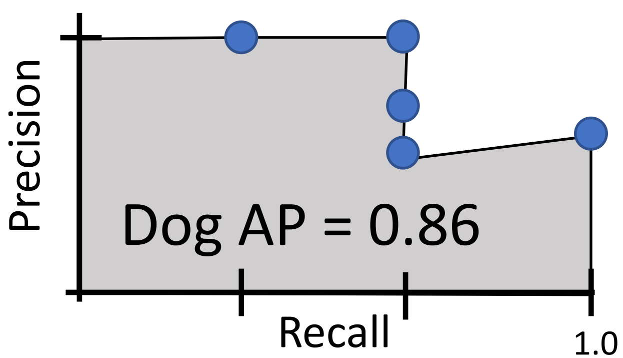


Evaluation

All dog detections sorted by score



All ground-truth (GT) dog boxes



Sort detections from highest to lowest score

For each detection:

- If it matches a GT box with IoU > 0.5, mark it as positive and remove that GT box
- Otherwise, mark it negative
- Plot a point on the P-R curve

Average Precision (AP) = area under PR curve

Mean AP = Average AP over all object classes

Mean Average Precision (mAP)

- Example:
  - Bird AP = 0.65
  - Cat AP = 0.80
  - Dog AP = 0.86
  - mAP@0.5 = 0.77
- "COCO mAP": Compute mAP for multiple IoU thresholds (0.5, 0.55. 0.6, … 0.95) and average
  - Example: mAP@0.5 = 0.77, mAP@0.55 = 0.72, … mAP@0.95 = 0.19
  - COCO mAP = 0.45

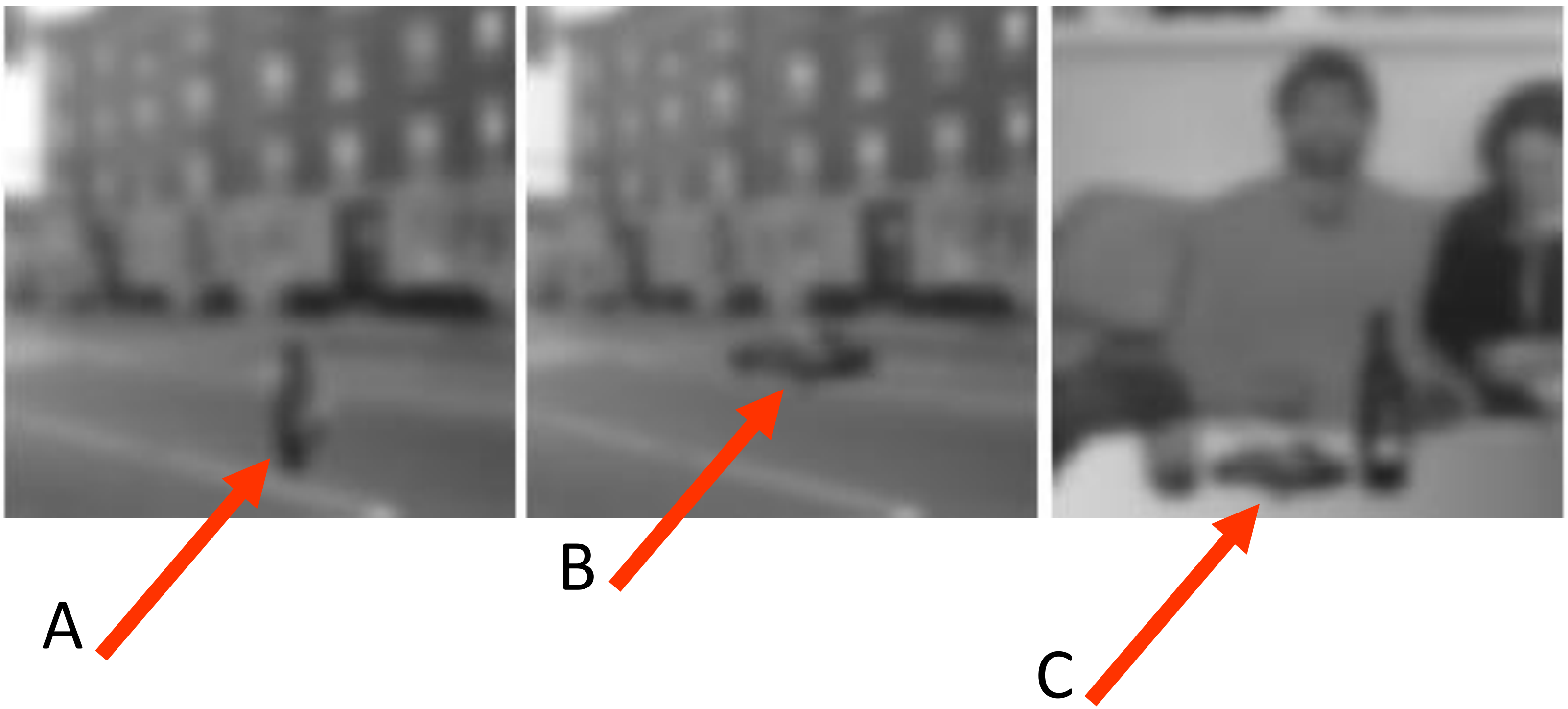
Evaluation summary

- Common metric for evaluating object detectors is mAP (or COCO mAP)
- Both NMS and P-R steps require IoU thresholds; different thresholds can change results
- Object detection is complex – one number is not very informative
  - How accurate is the object classification?
  - How accurate are the bounding boxes?
  - What kinds of errors is the model making (misses, false alarms)?

Beyond patches?

Scene priors

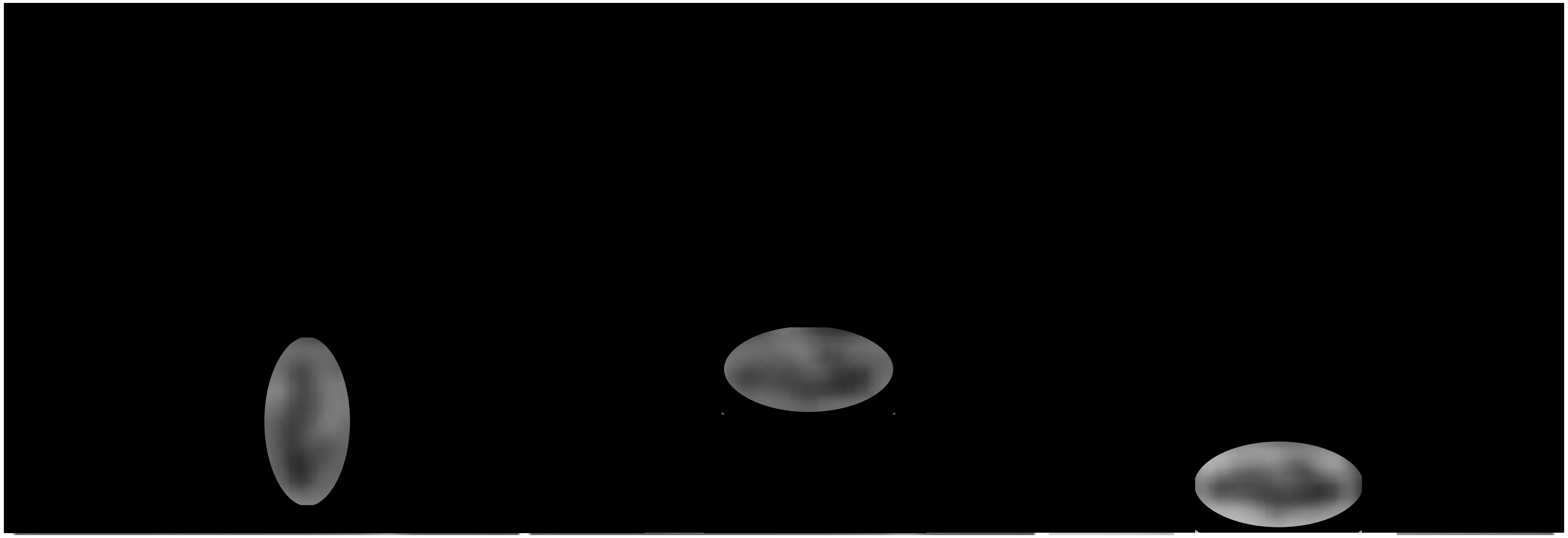
Scene priors



What are these objects?

Image: A. Torralba

Scene priors



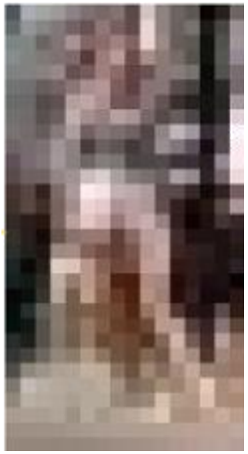
The same pixels (originally from a car)

Image: A. Torralba

Scene context

A

B



C

D

Image: D. Hoiem

Object detection in context

- Scene context provides both global and local priors:
    - Global prior: likelihood of the object appearing at all
    - Local: likelihood of the object in given location

- Including these priors can help reduce false detections
- Is there a downside to including these priors?

Image: A. Torralba

Summary

- Object detection is typically modelled as a patch classification problem
- Various ways to approach the classification problem: two-stage region proposal detectors, single-stage detectors
- However, this is not the only way to approach object detection – information outside the patch (scene context) can also be used to predict object presence / location