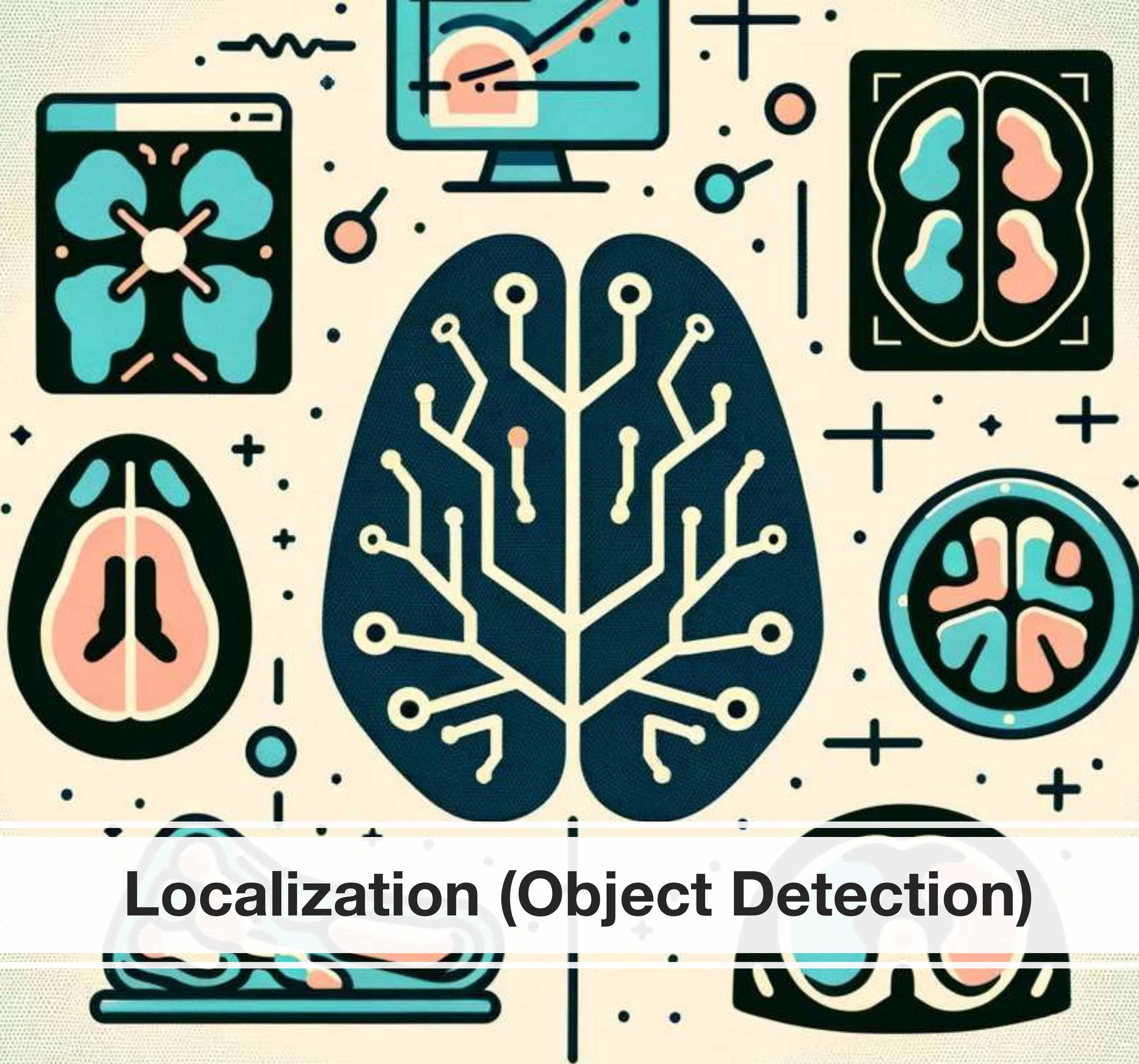


Localization (Object Detection)

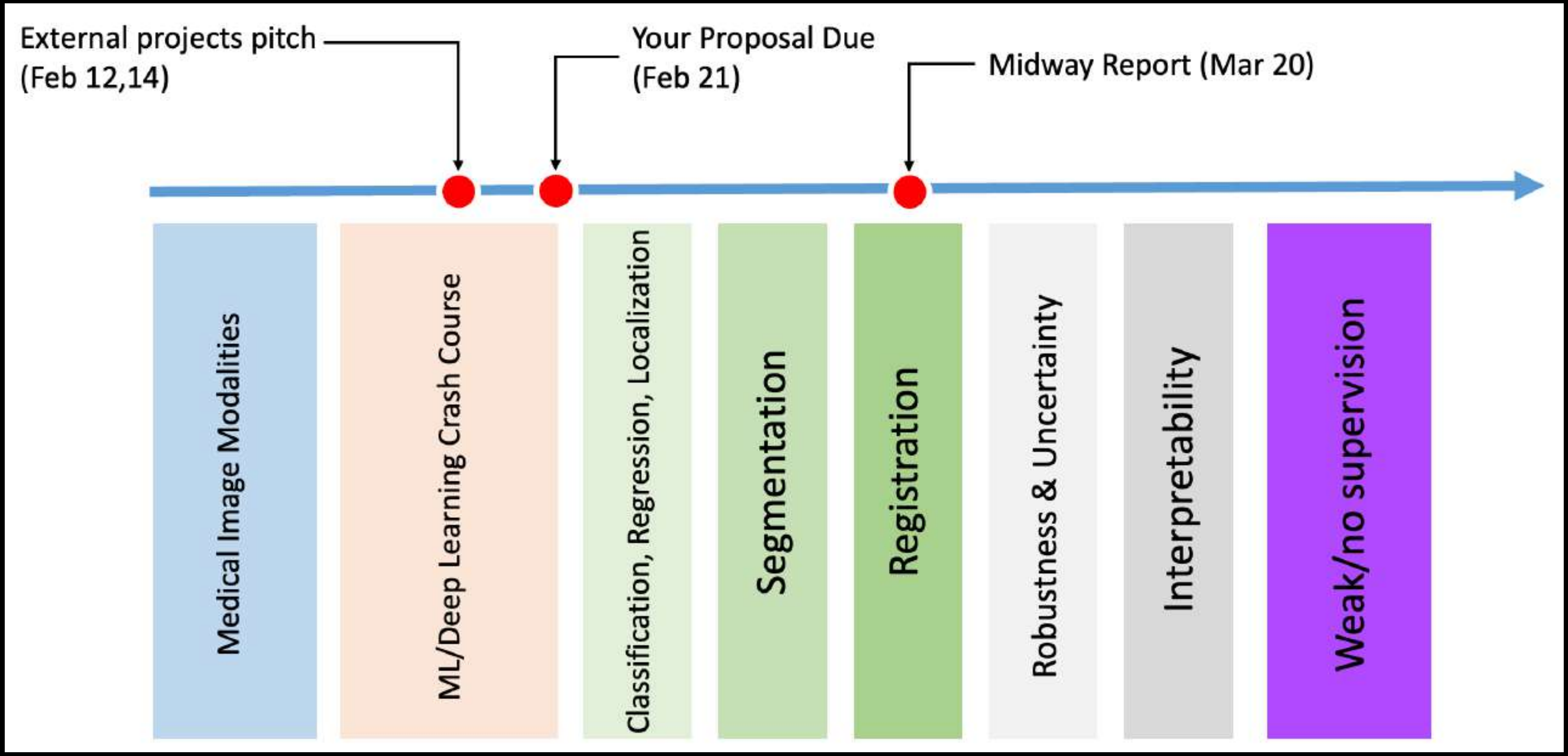
Logistics

- HW2 is due Mar 18.
- The project's progress reports are due March 20.

Agenda

- Metrics:
  - Accuracy, confusion plot, precision, recall, F1
  - Curves: ROC, PRC
- Localization (object detection):
  - Motivational Examples
  - Basic ideas
  - Two-steps methods
  - Single-step methods
  - Evaluation metrics

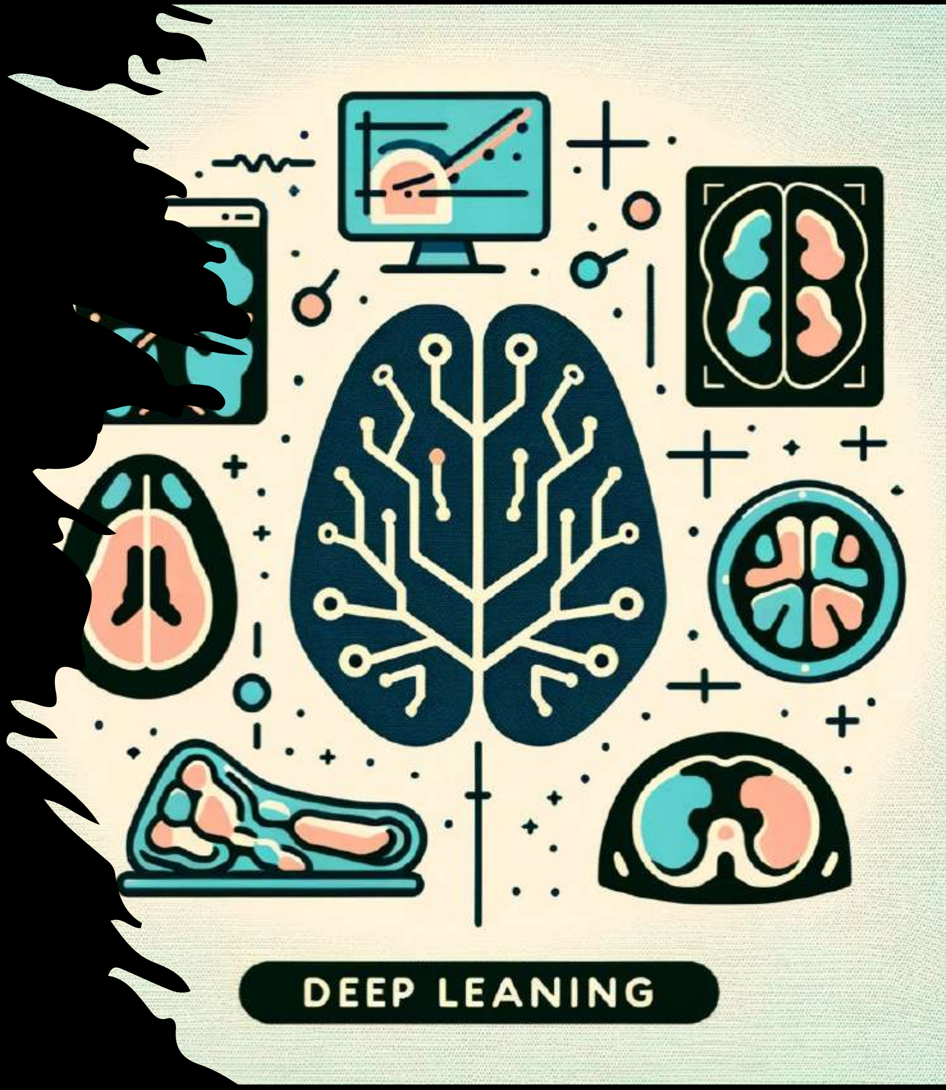
Timeline


External projects pitch (Feb 12,14)
Your Proposal Due (Feb 21)
Midway Report (Mar 20)
Medical Image Modalities
ML/Deep Learning Crash Course
Classification, Regression, Localization
Segmentation
Registration
Robustness & Uncertainty
Interpretability
Weak/no supervision

Credits

- Deep Learning for Computer Vision by Prof. Justin Johnson (University of Michigan)
  - https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/
- Introduction to Computer Vision by Prof. Kosta Derpanis (York University)
  - https://www.eecs.yorku.ca/~kosta/Courses/EECS4422/
- EC 523 (Deep Learning)

Performance Metrics



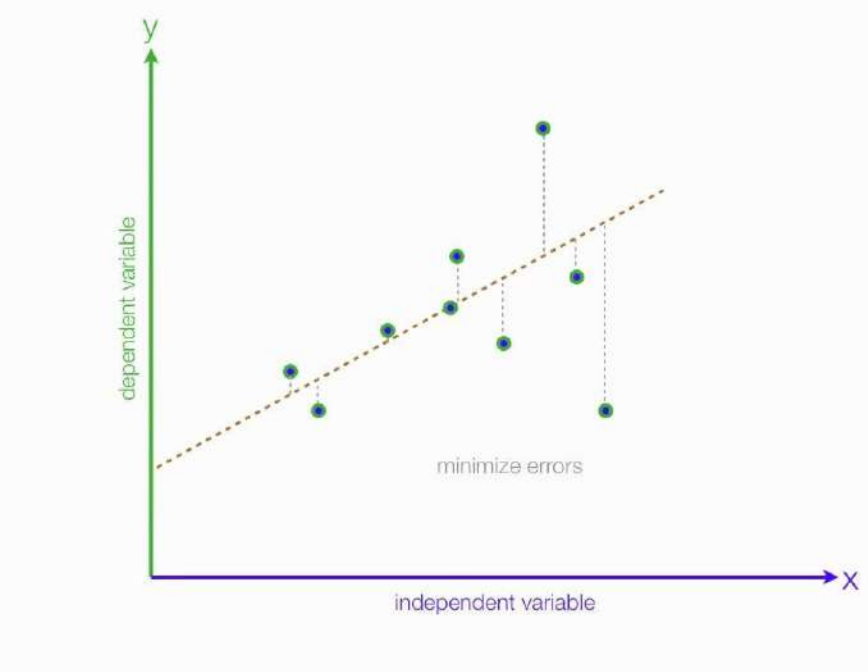
Metrics for Regression

- Suppose the true label is y_i and our model predicts $\hat{y}_i$.
- mean squared error (MSE):

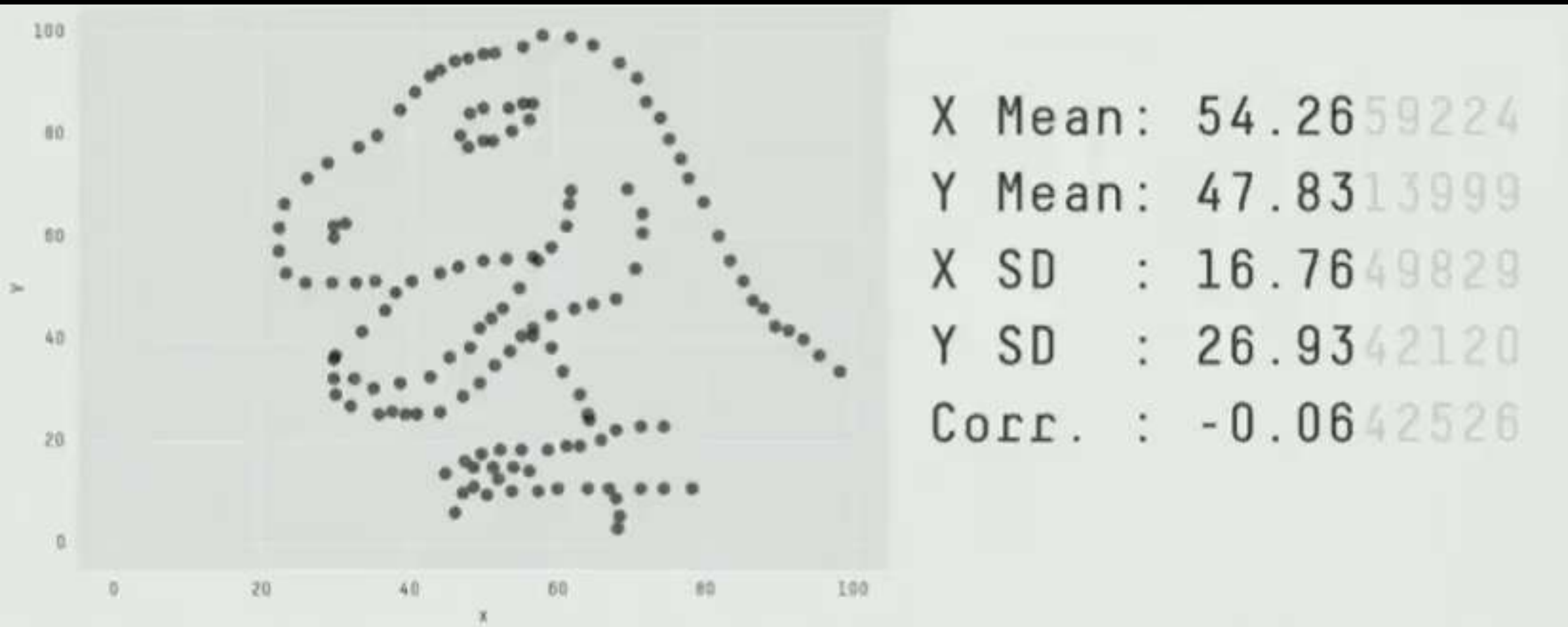
  MSE = mean of $(y_i - \hat{y}_i)^2$
- mean absolute error (MAE)

  MAE = mean of $|y_i - \hat{y}_i|$
- coefficient of determination (R^2)

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

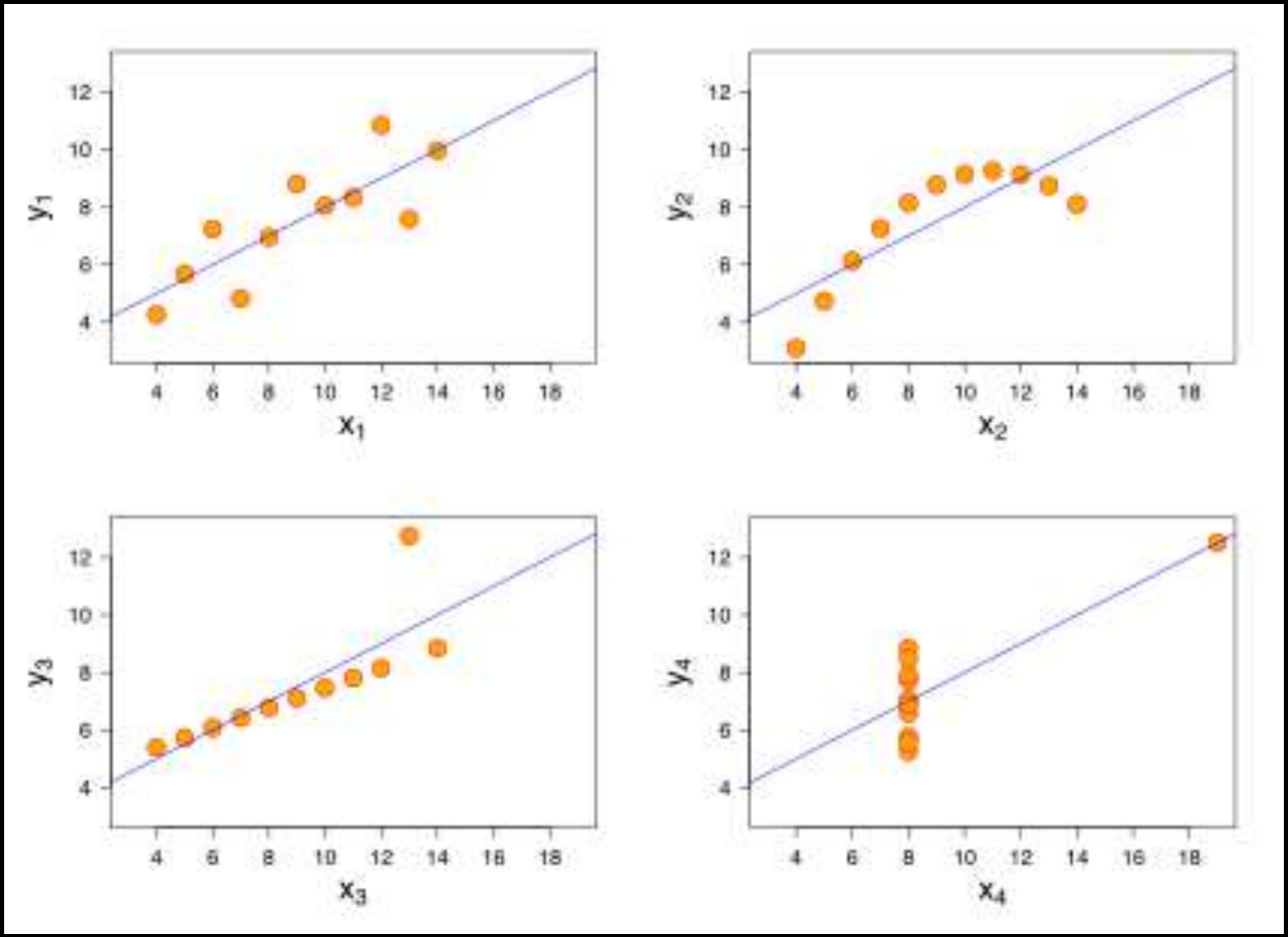


Source: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e



Source: https://twitter.com/docmilanfar/status/1400276401513648143?s=21&t=XNnoWSbNypYksQ9atT_j1w

Always plot your prediction



The same summary stat but very different behaviors

https://en.wikipedia.org/wiki/Anscombe%27s_quartet

Accuracy

- When we deal with binary classification, we often measure performance simply using Accuracy:

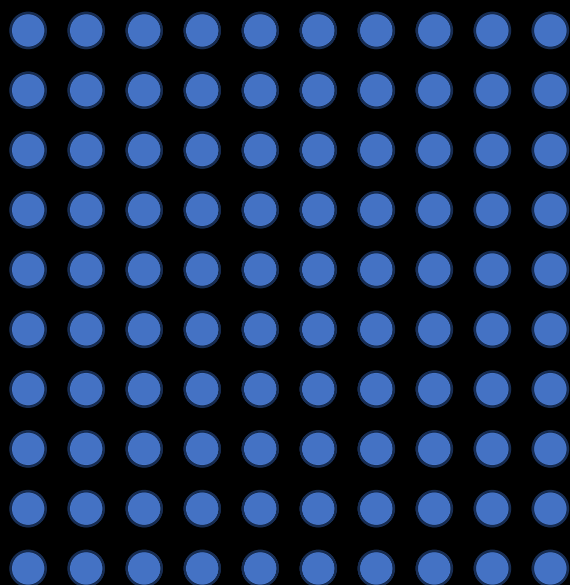
$$\text{accuracy} = \frac{\#\ \text{correct predictions}}{\#\ \text{test instances}}$$

$$\text{error} = 1 - \text{accuracy} = \frac{\#\ \text{incorrect predictions}}{\#\ \text{test instances}}$$

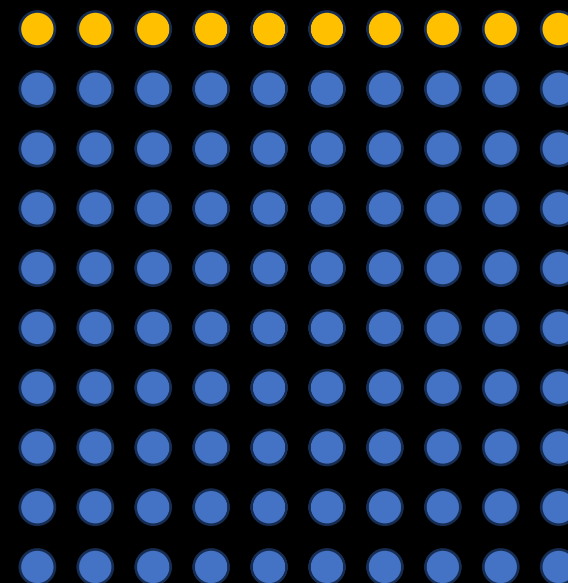
- Any possible problems with it?

Imbalanced Classification

Predicted Labels



True Labels: 10% prevalence



Accuracy of dummy classifier = 90% (!!!)

		Predicted condition	
	Total population $= P + N$	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P) [a]	True positive (TP), hit[b]	False negative (FN), type II error, miss, underestimation[c]
	Negative (N) [d]	False positive (FP), type I error, false alarm, overestimation[e]	True negative (TN), correct rejection[f]

Source: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

		Predicted condition		
	Total population $= \mathrm{P} + \mathrm{N}$	Predicted Positive (PP)	Predicted Negative (PN)	
Actual condition	Positive (P) [a]	True positive (TP), hit[b]	False negative (FN), type II error, miss, underestimation[c]	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{\mathrm{TP}}{\mathrm{P}} = 1 - \mathrm{FNR}$
	Negative (N) [d]	False positive (FP), type I error, false alarm, overestimation[e]	True negative (TN), correct rejection[f]	False positive rate (FPR), probability of false alarm, fall-out $= \frac{\mathrm{FP}}{\mathrm{N}} = 1 - \mathrm{TNR}$

Source: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

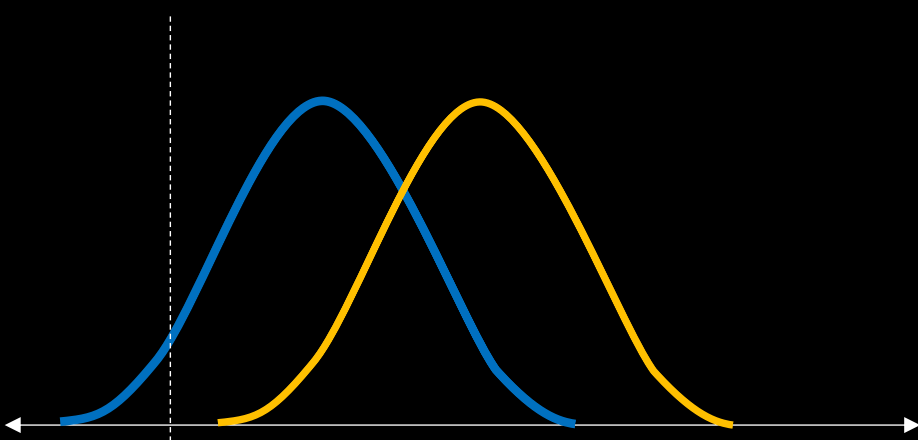
Receiver operating characteristic (ROC) Curve

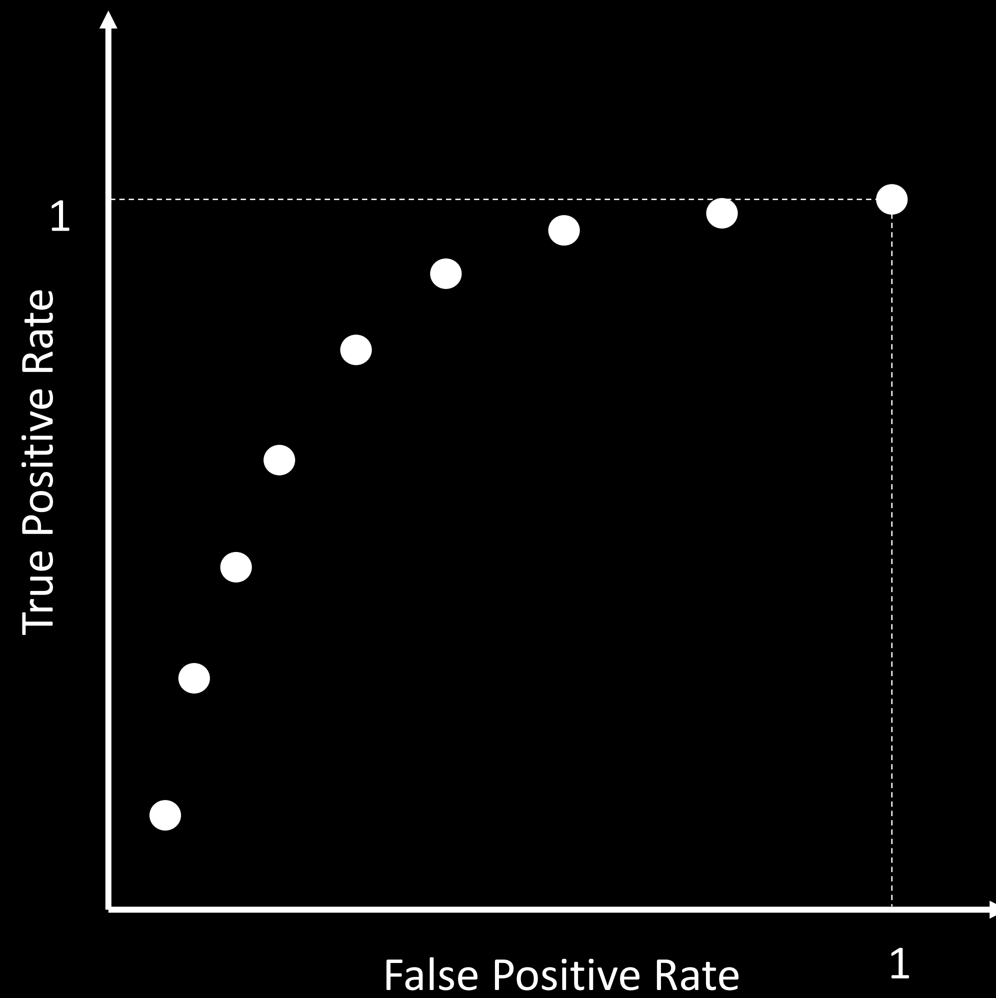
Consider Binary Classification:

True Labels

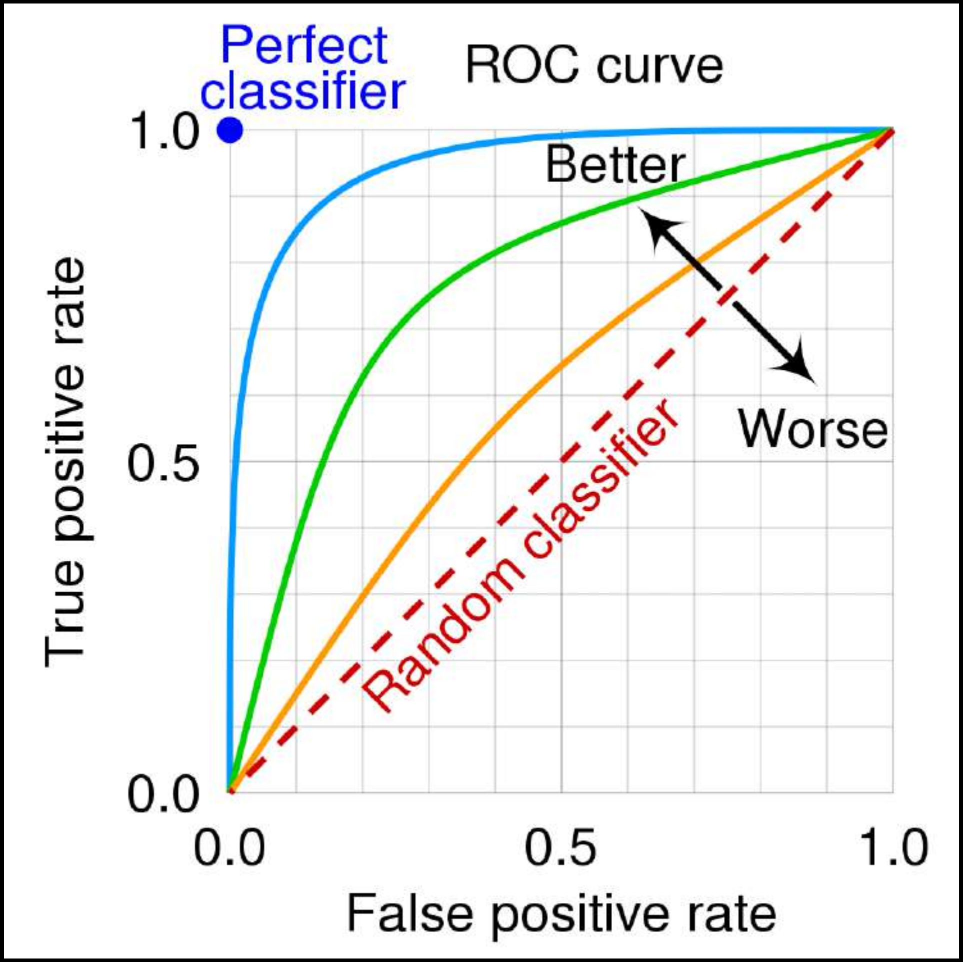
.1, .5, .5, .8, .7, .2, .9, .8, .1, .5

Predicted Probabilities





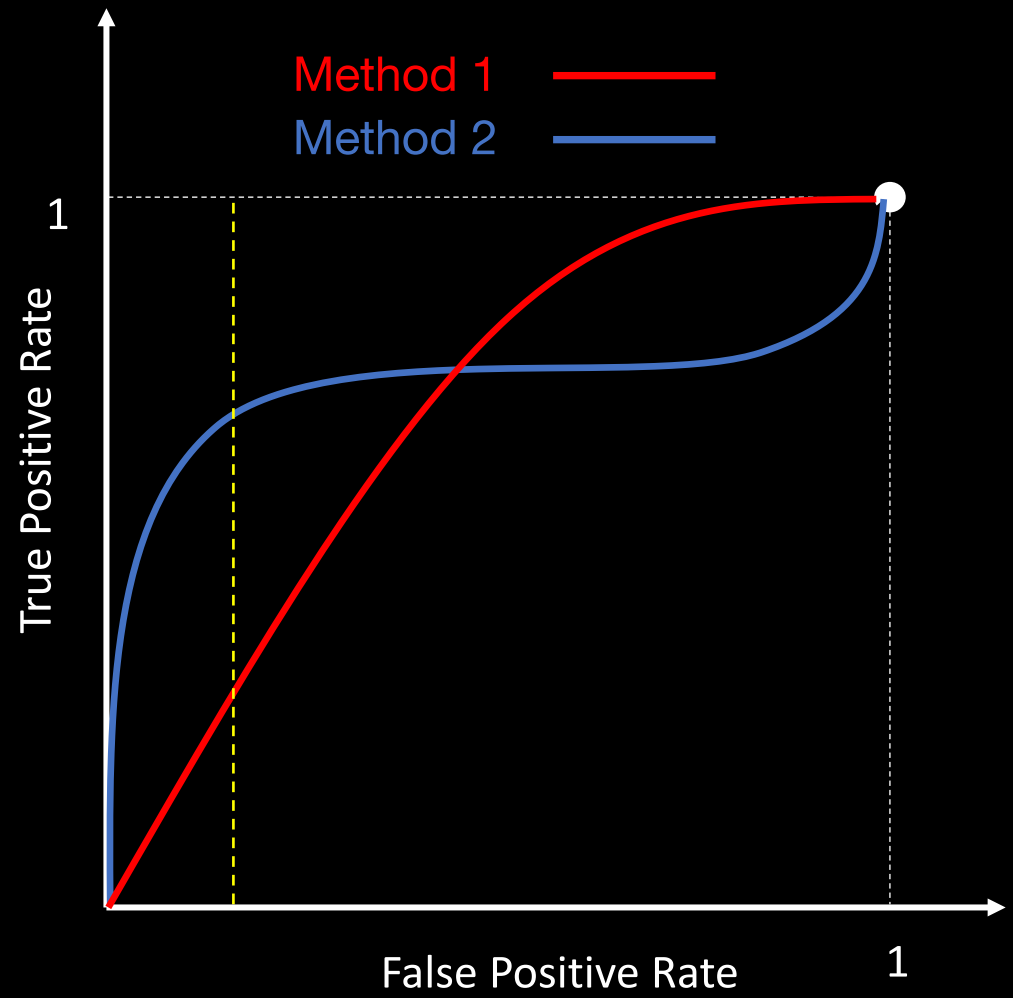
Receiver operating characteristic (ROC) Curve



Source: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Receiver operating characteristic (ROC) Curve

They both have the same Area Under the Curve. Which method is better?



Issues with ROC Curve

- No sensitive to imbalanced data
- Difficult to extend to multi-class setting

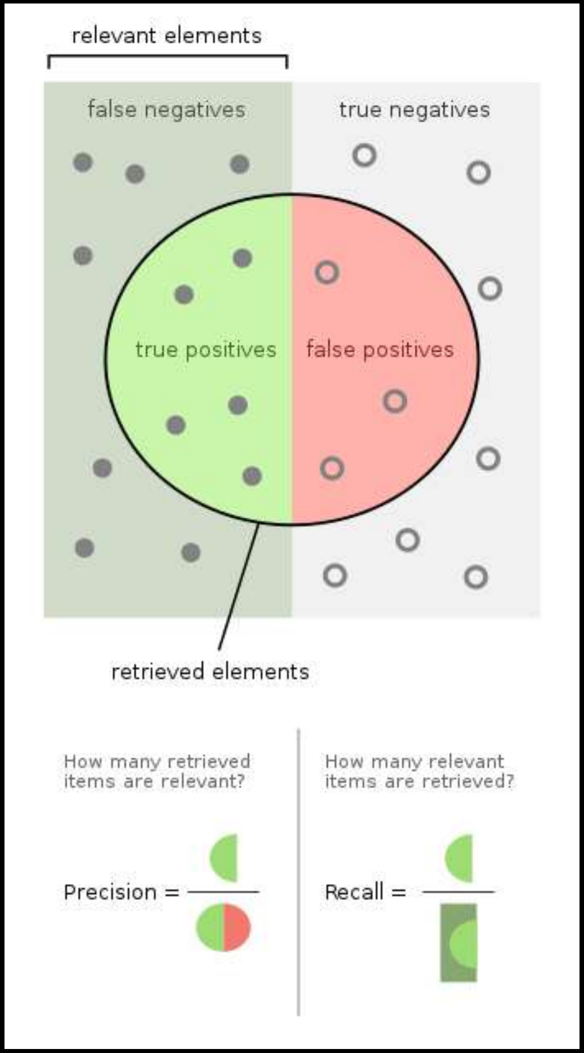
Precision and Recall

Precision is the fraction of relevant instances among the retrieved instances.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Predicted Positive}}$$

Recall is the fraction of relevant instances that were retrieved.

$$\text{Recall} = \frac{\text{True Positive}}{\text{Total Number of Positives}}$$



Source: https://en.wikipedia.org/wiki/Precision_and_recall

Precision and Recall

Consider Binary Classification ( is positive class):

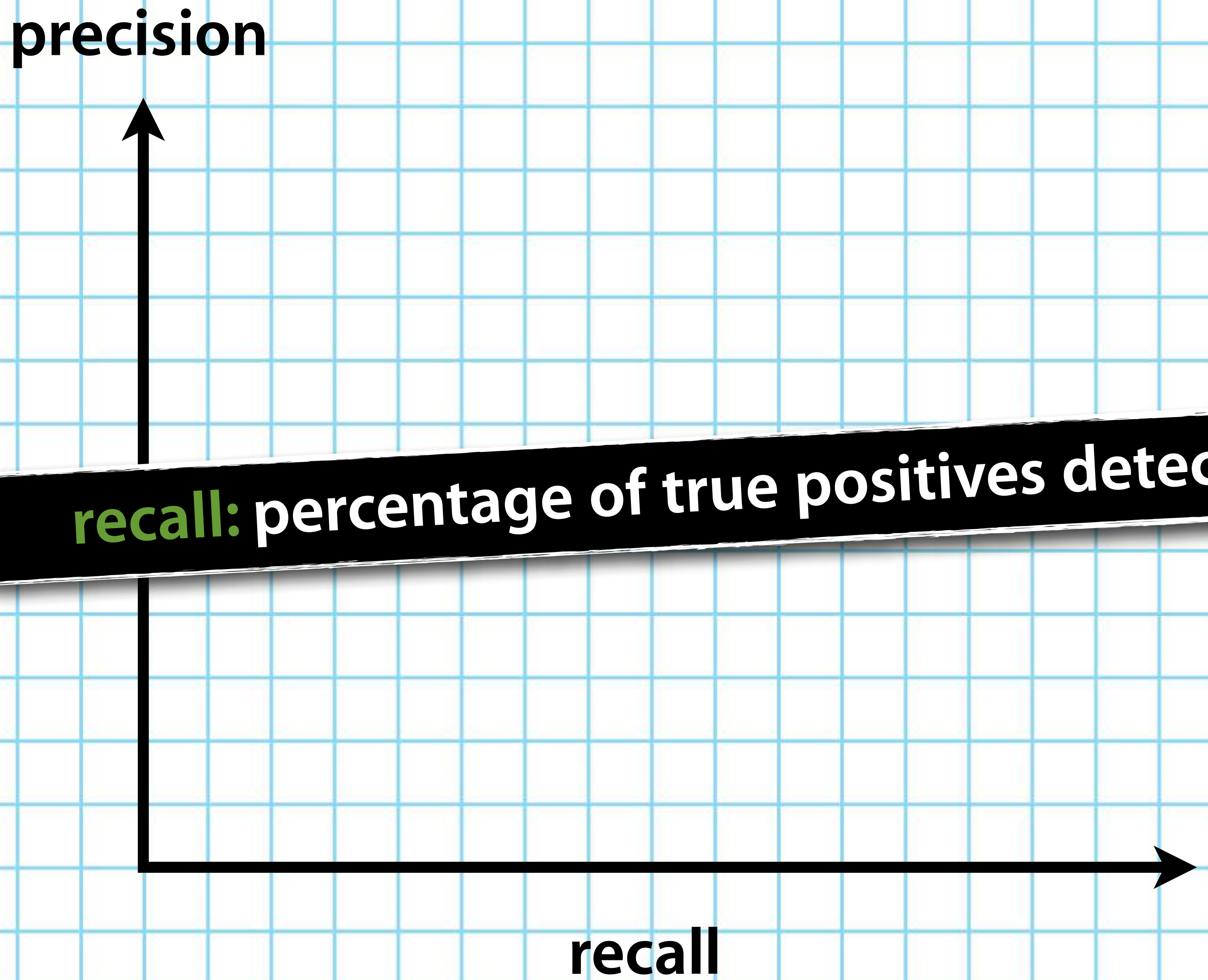
True Labels

Predicted Labels (small threshold):

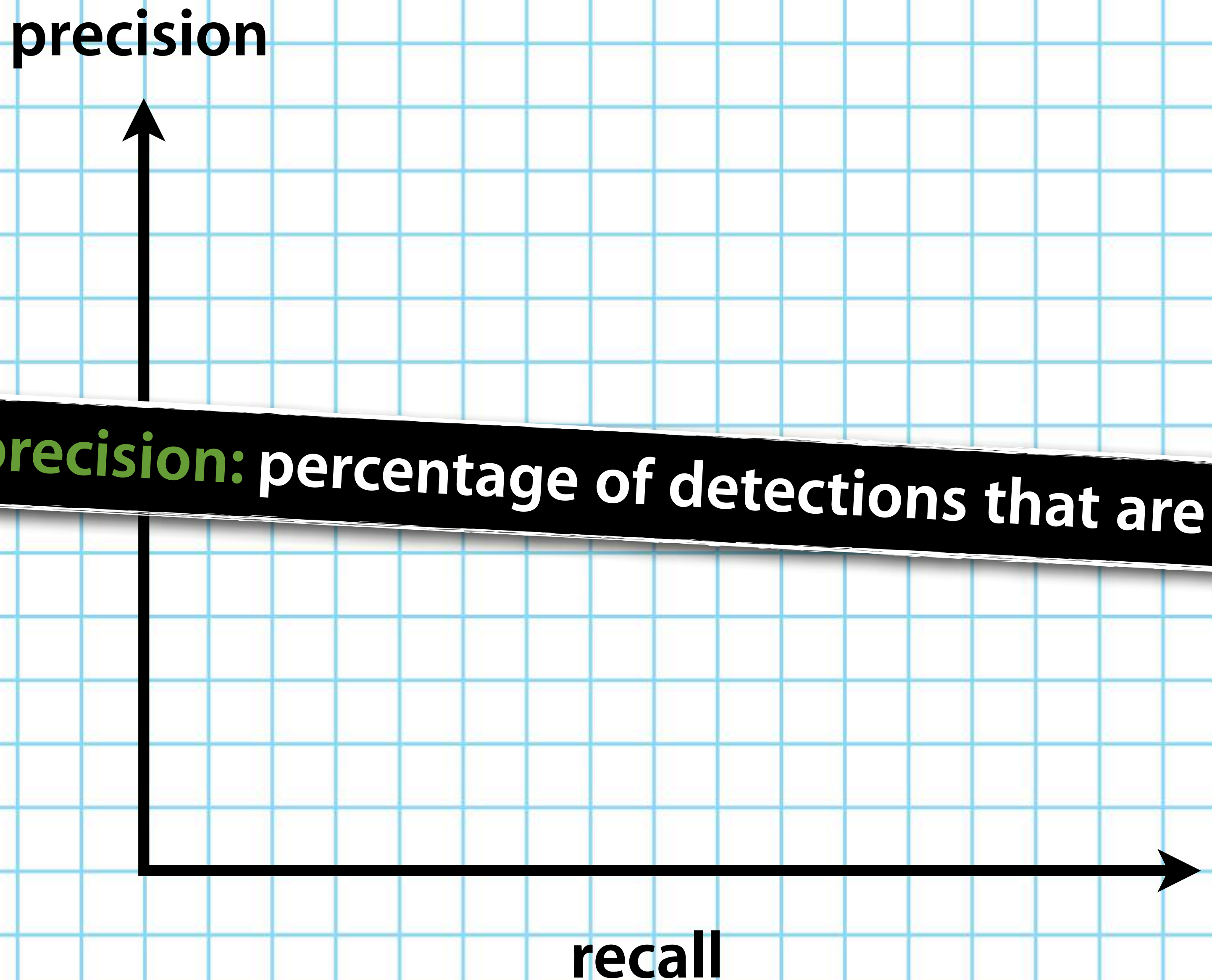
Recall (↓), Precision (↑)

Predicted Labels (small threshold):

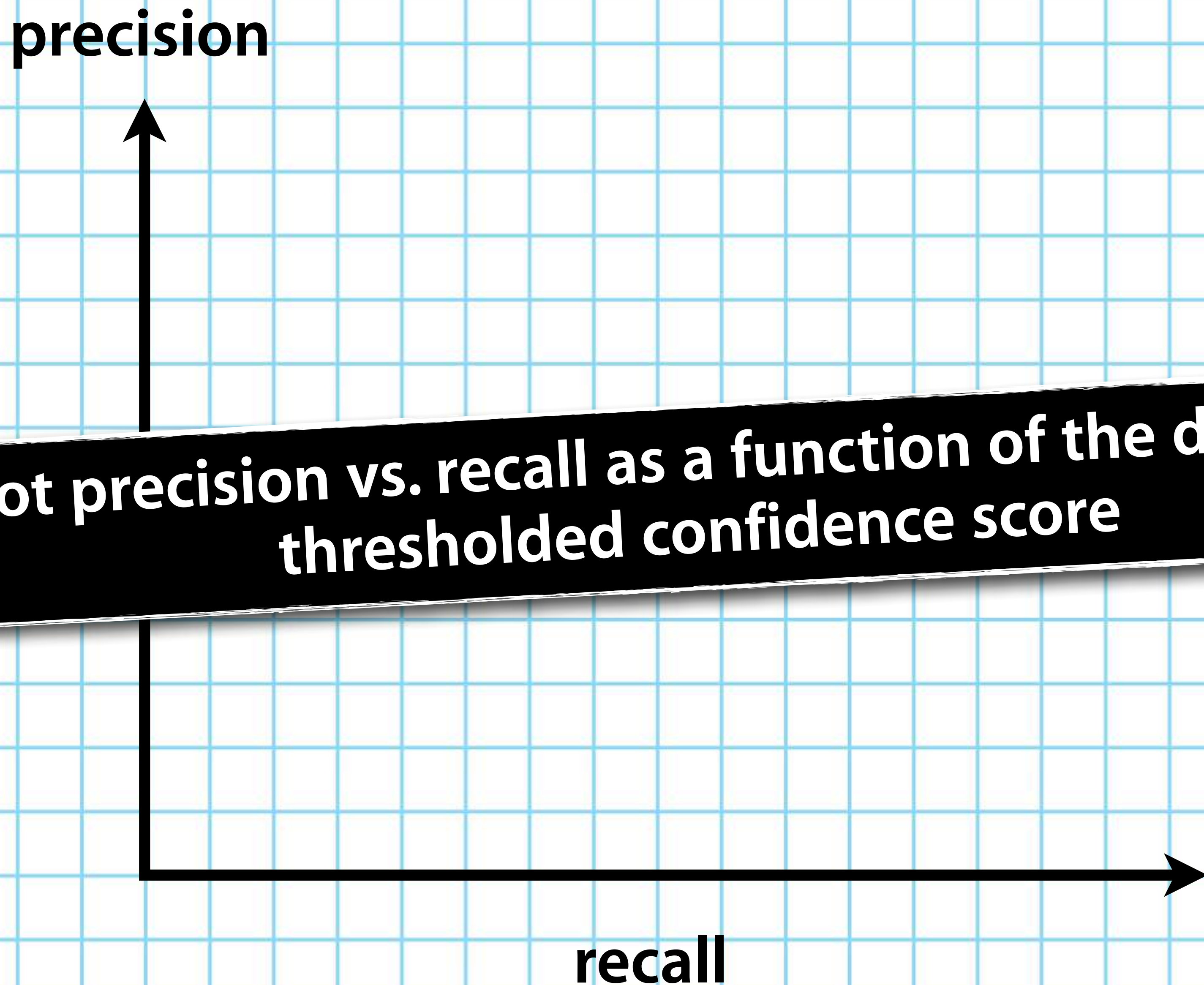
Recall (↑), Precision (↓)



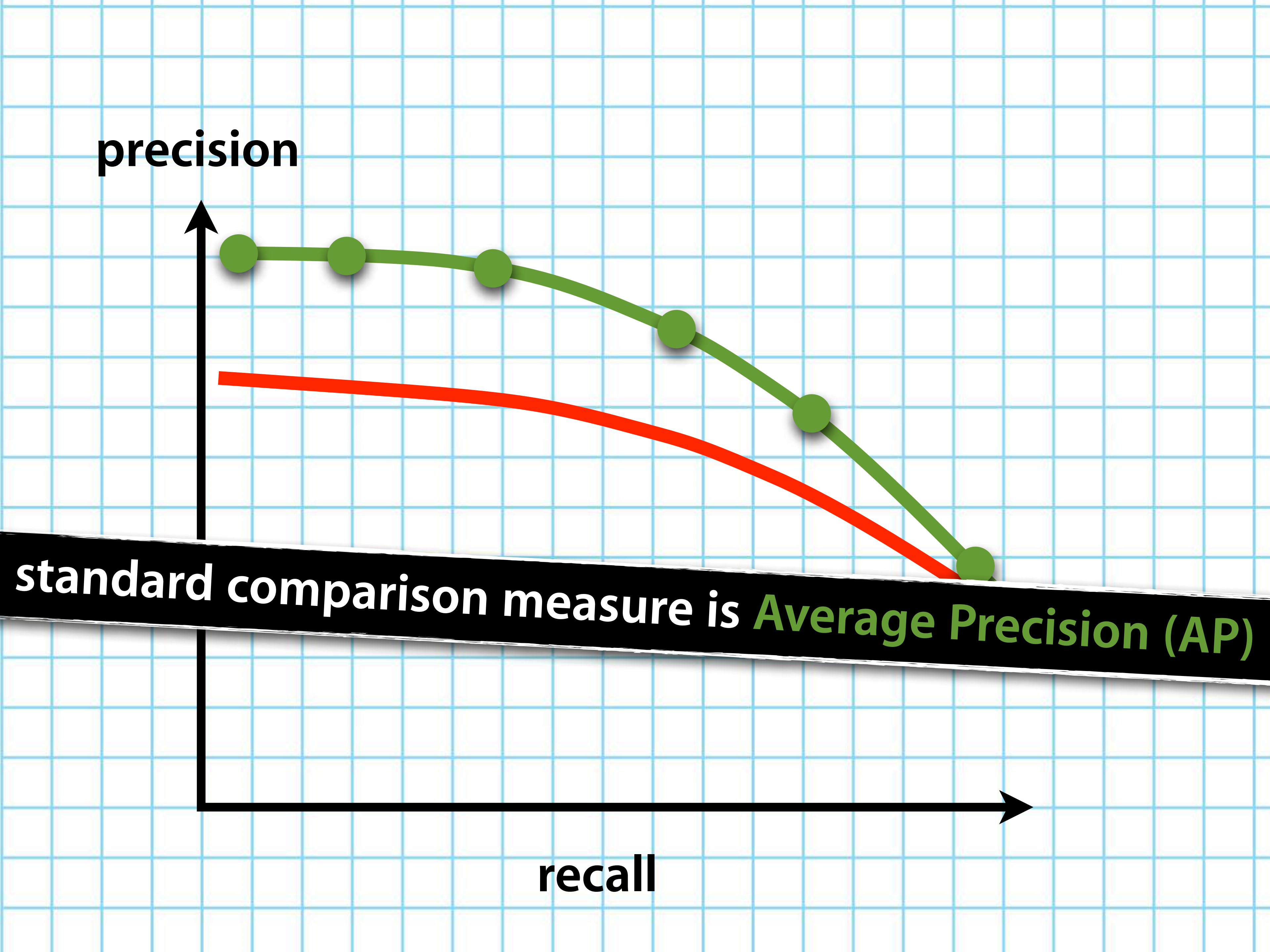
recall: percentage of true positives detected



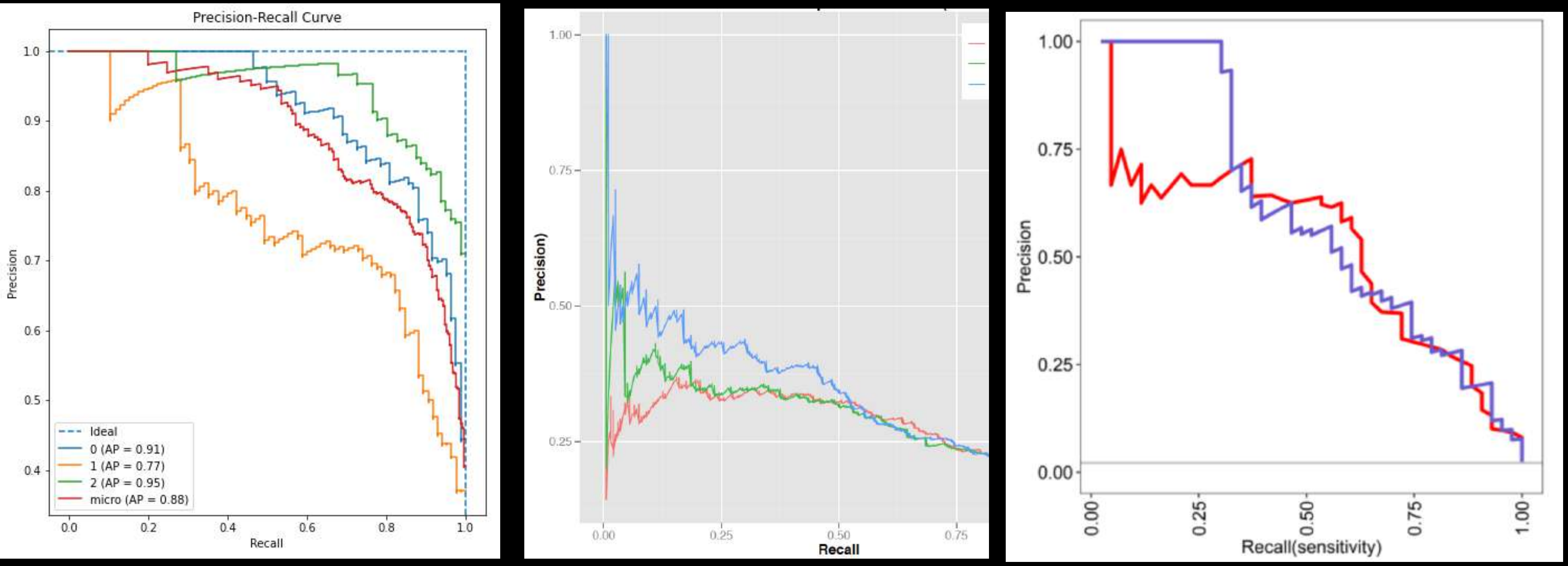
precision: percentage of detections that are correct



plot precision vs. recall as a function of the detector's thresholded confidence score


precision
recall
standard comparison measure is Average Precision (AP)

Examples of PR curves

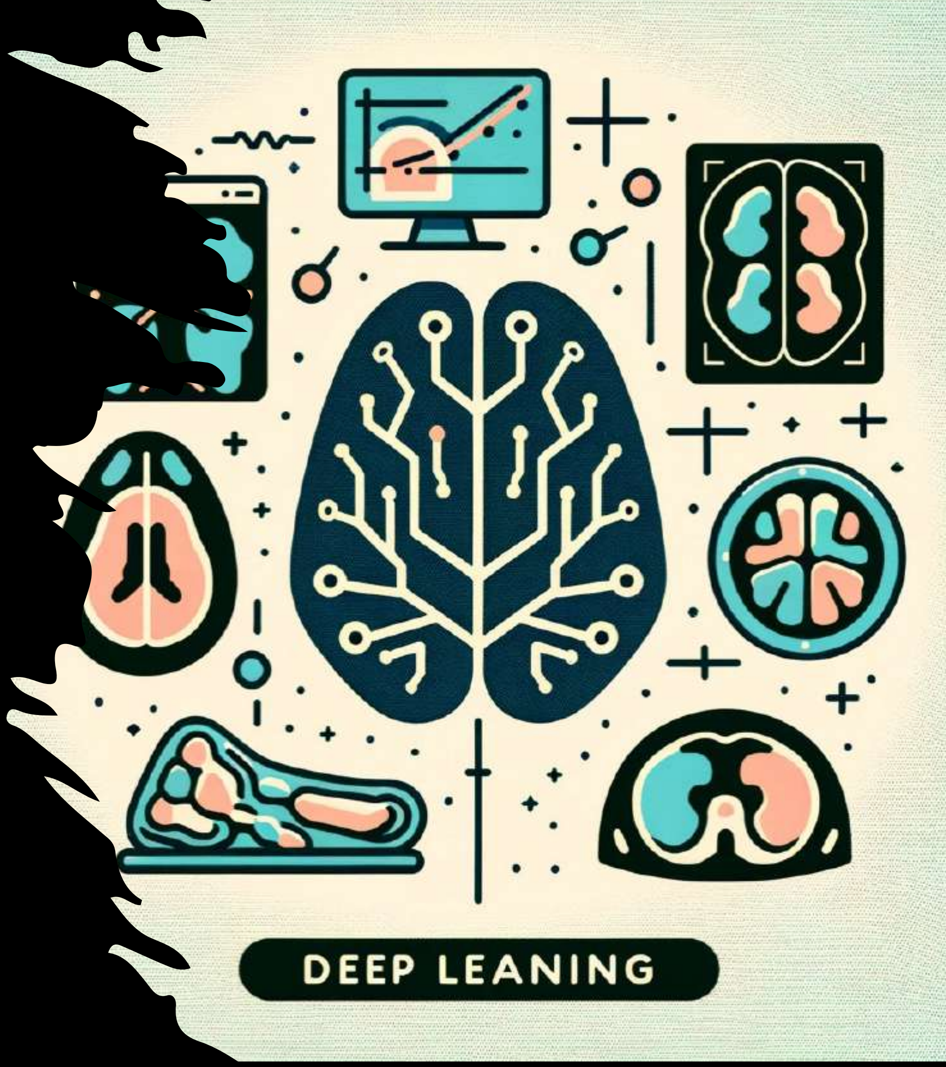


Yes, they sometimes look strange!

Summary

- In regression, always visualize your prediction; don't trust a single metric.
- In classification, the accuracy metric is barely useful.
- Similar to the regression problems, a single metric may provide a myopic view of the performance.
- AUROC value can be misleading. Look at the entire curve.
- For an imbalanced classification problem, use AUPRC.

Localization

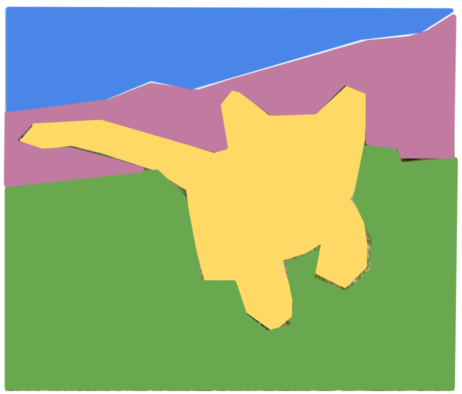


Classification

CAT

No spatial extent

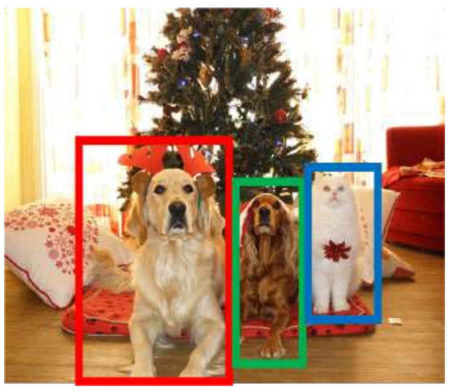
Semantic Segmentation



GRASS, CAT, TREE, SKY

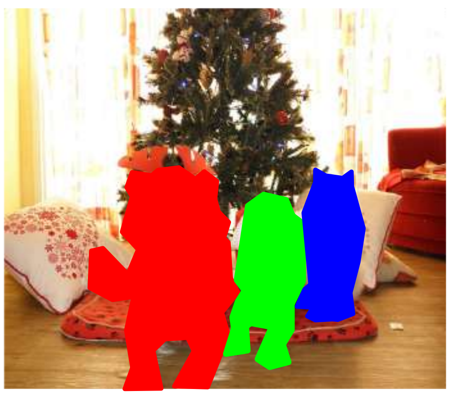
No objects, just pixels

Object Detection



DOG, DOG, CAT

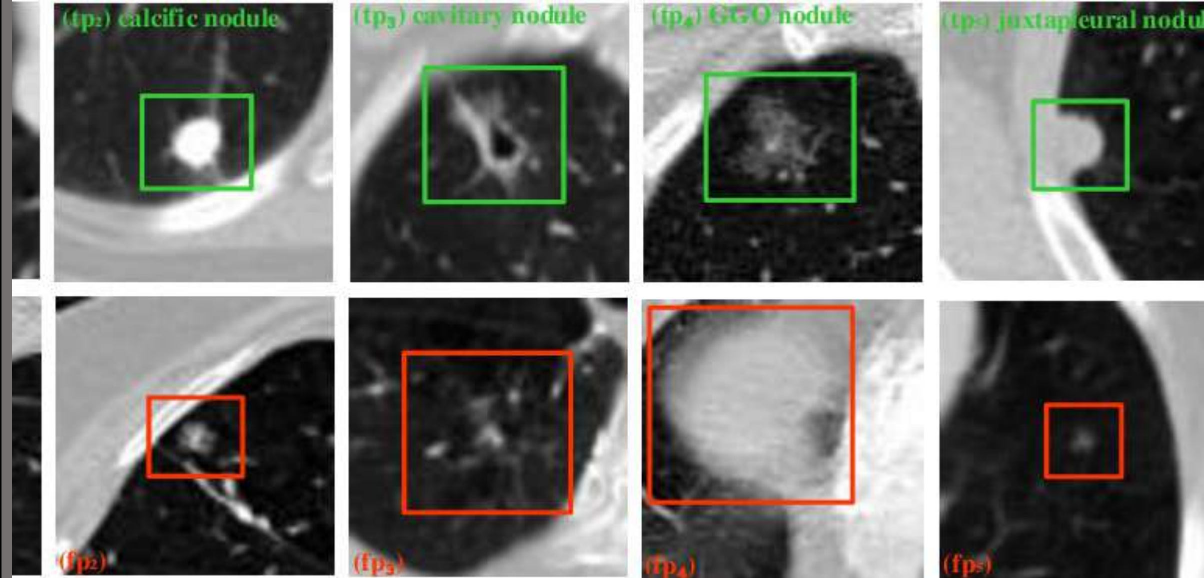
Instance Segmentation

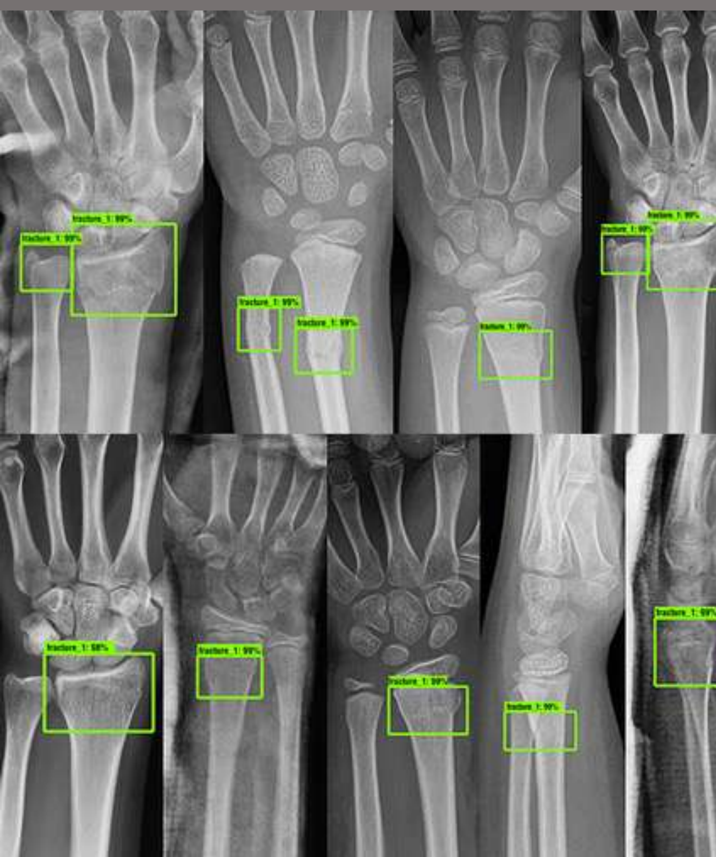


DOG, DOG, CAT

Multiple Objects

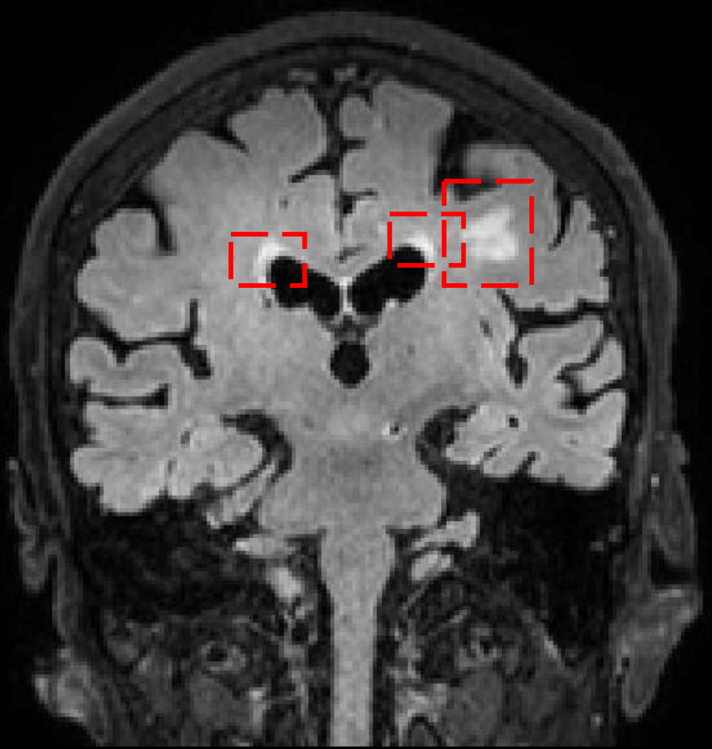
Examples: Localization in Medical Images

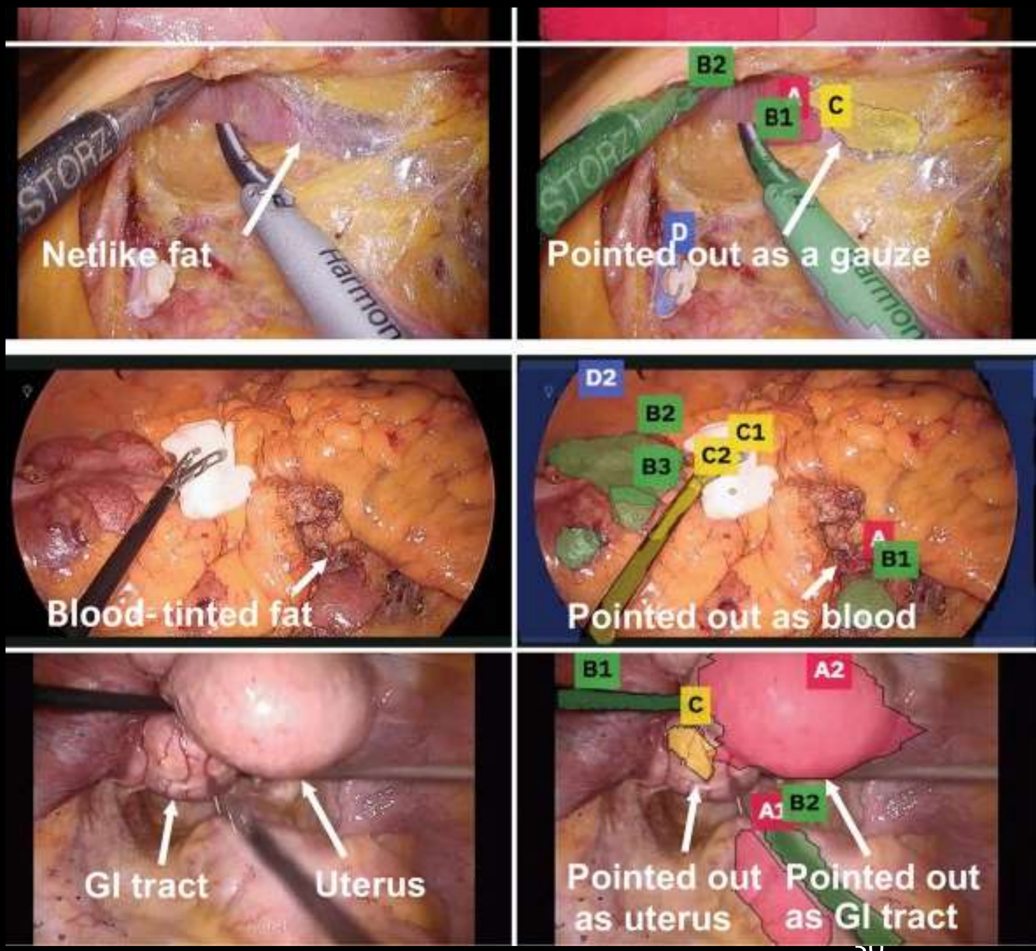




- Source: https://www.semanticscholar.org/paper/Two-Stage-Convolutional-Neural-Network-Architecture-Cao-Liu/cb02fbd6dbf2323d69af0c9521c9e8ae1ebe0577

Examples: Localization in Medical Images



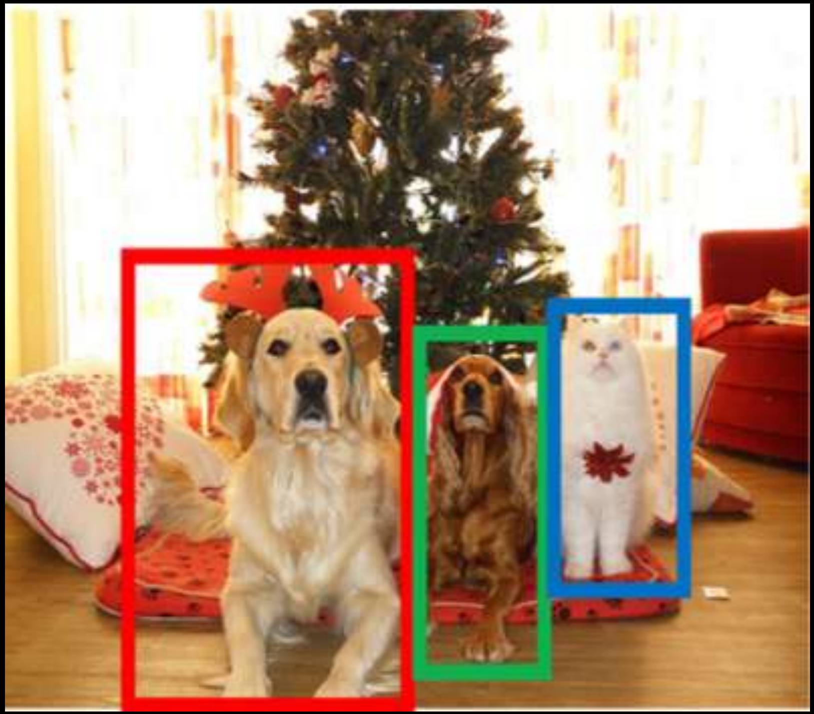


Source: https://link.springer.com/article/10.1007/s11548-021-02434-w

General Setup

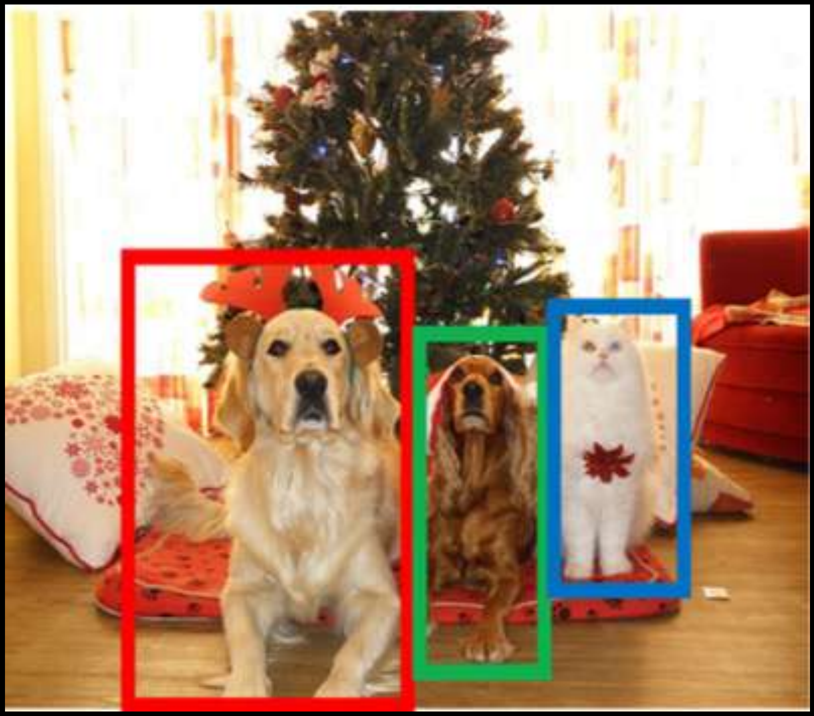
Input: Single Image

Output: A set of detected objects;
For each object, predict:
1. Category label(from fixed, known set of categories)
2. Bounding box (four numbers: x, y, width, height)



Why is it challenging?

Multiple outputs: Needtooutput variable numbers of objects per image
- **Multiple types of output:** Need to predict ”what” (category label) as well as “where” (bounding box)
- **Large images:** Classification works at 224x224; need higher resolution for detection, often ~800x600

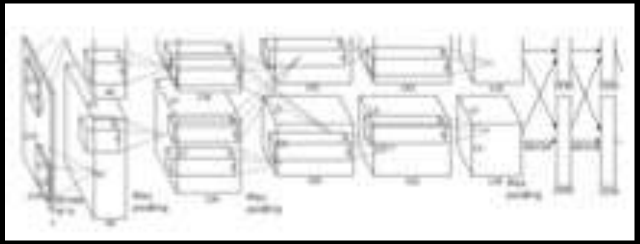


Detecting Single Object

"What"

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss



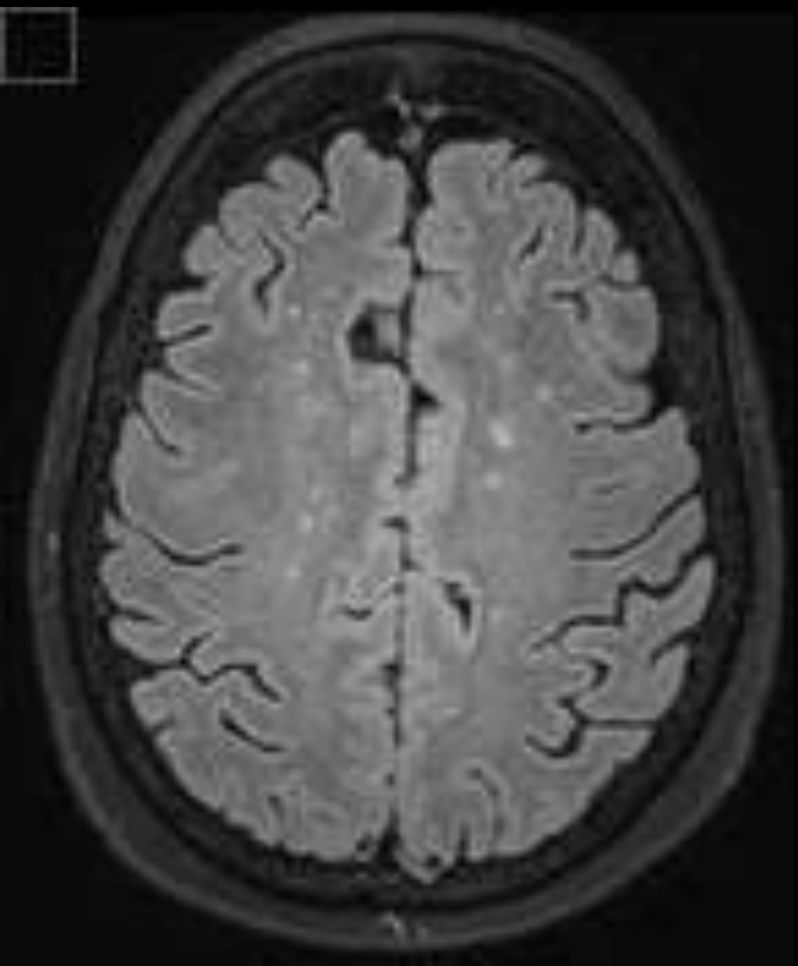
Vector
4096

Box Coordinates
(x,y,w,h)

L2 Loss

"Where"

Classical approach

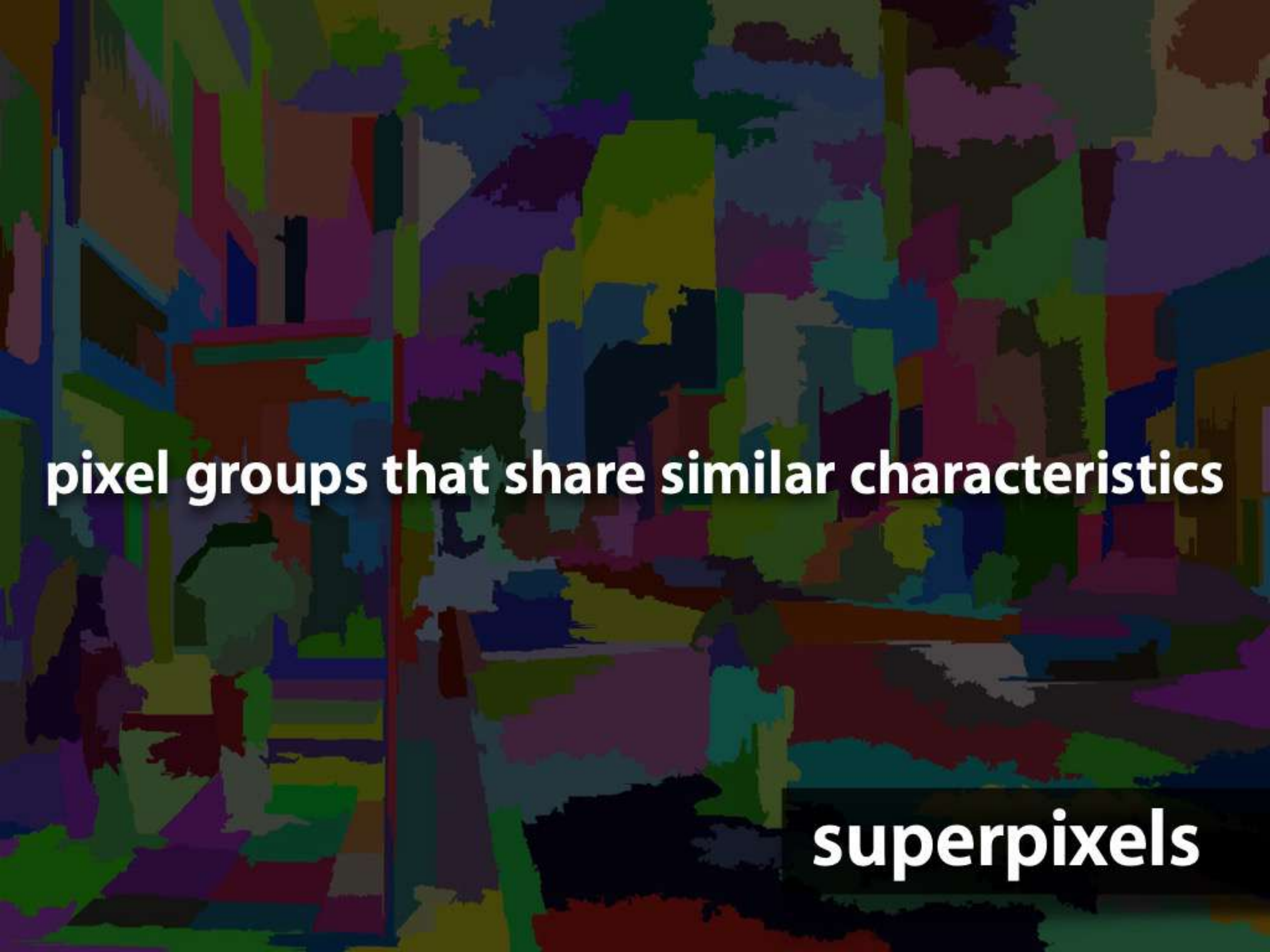


- The sliding window across the image
  - Use CCN to extract features
  - Classify each box
  - Change the resolution of image for multiple scales

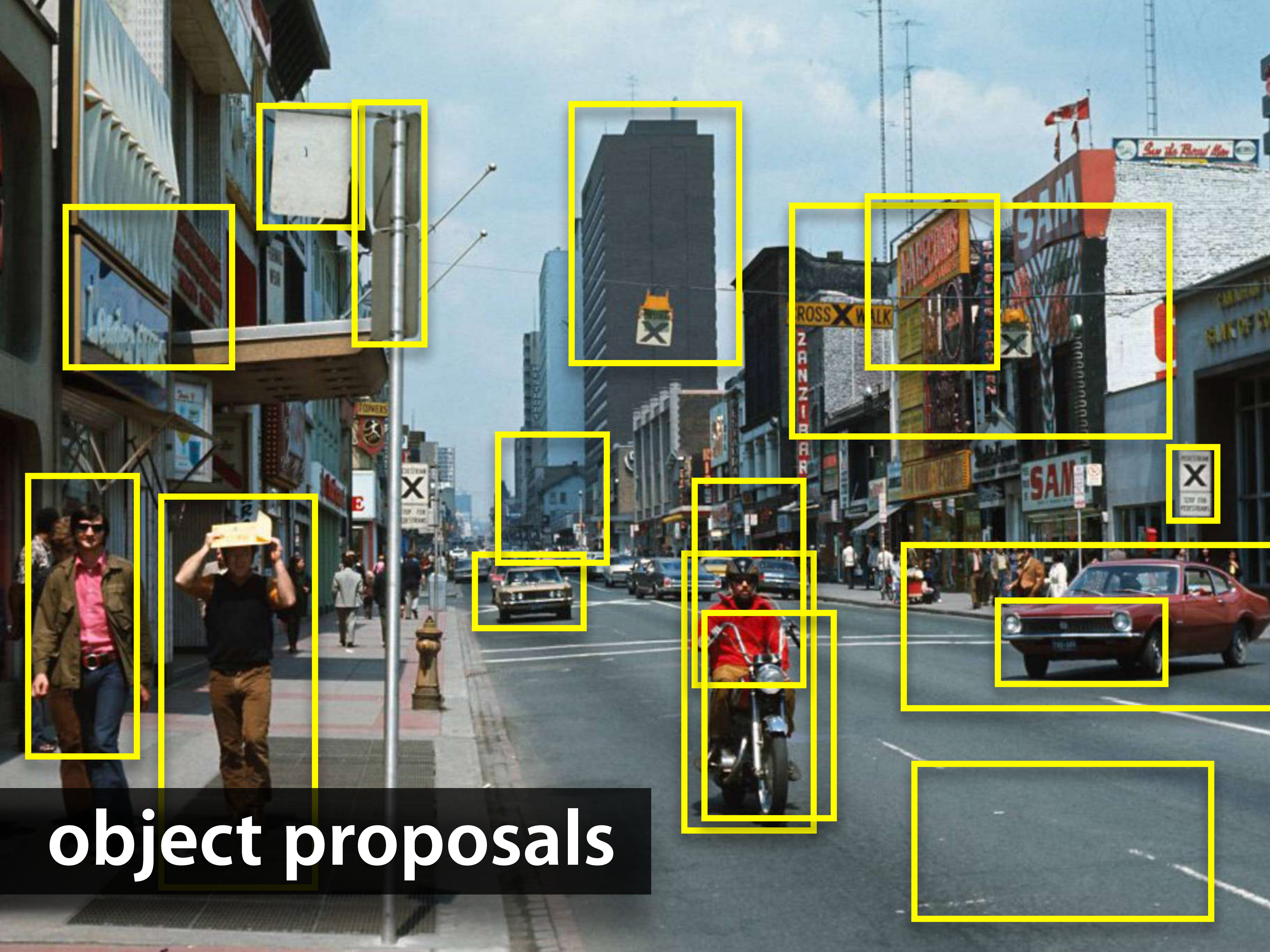
- Very Slow
- What about non-squared boxes
- What about overlapping windows


input image
CROSS X WALK
ZANZIBAR
SAM


superpixels

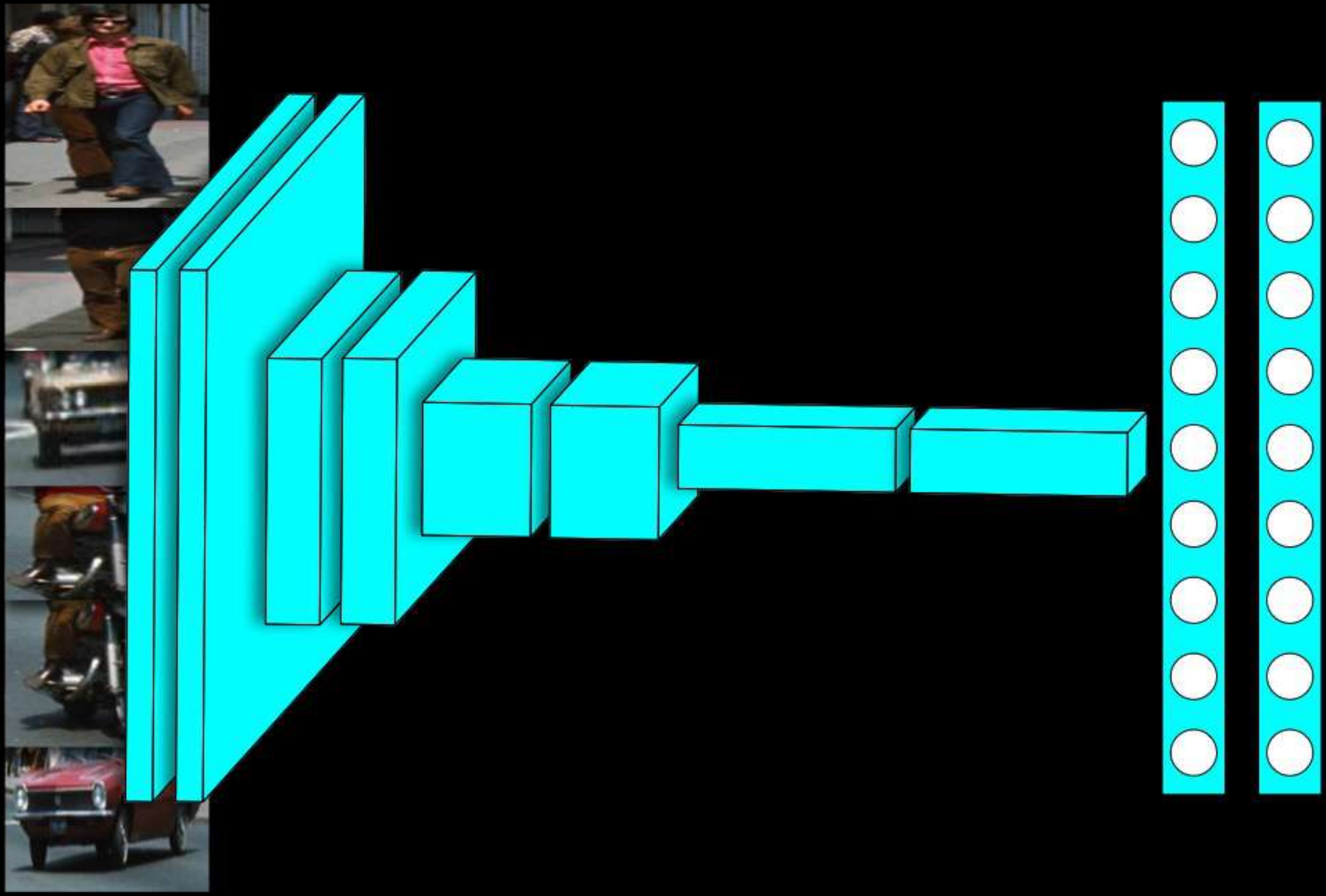

pixel groups that share similar characteristics
superpixels

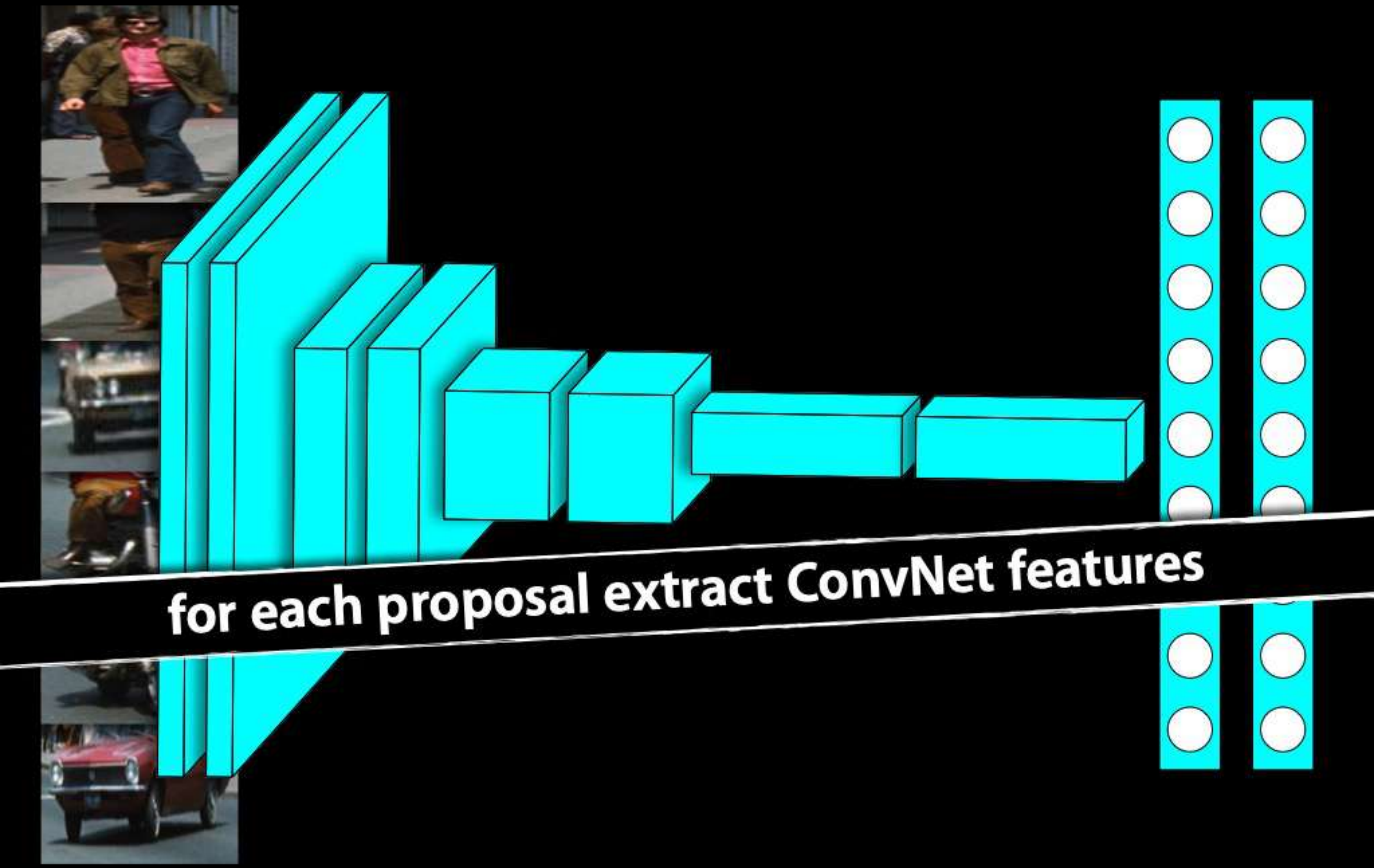

object proposals

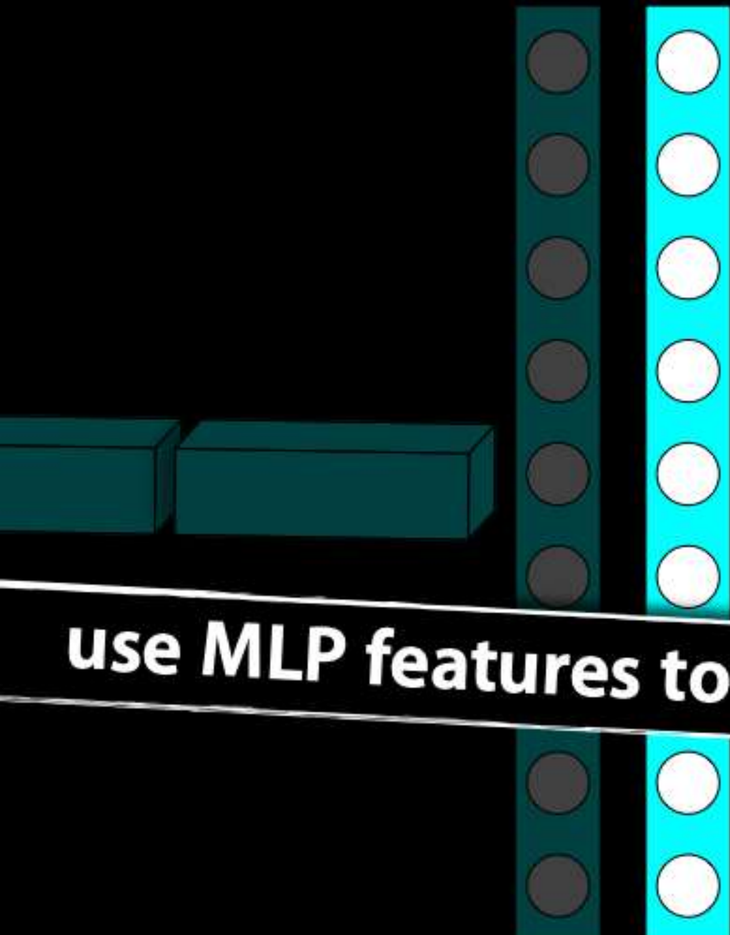

object proposals


CROSS WALK
SAM
extract object region proposals

warp each region to a canonical size




for each proposal extract ConvNet features



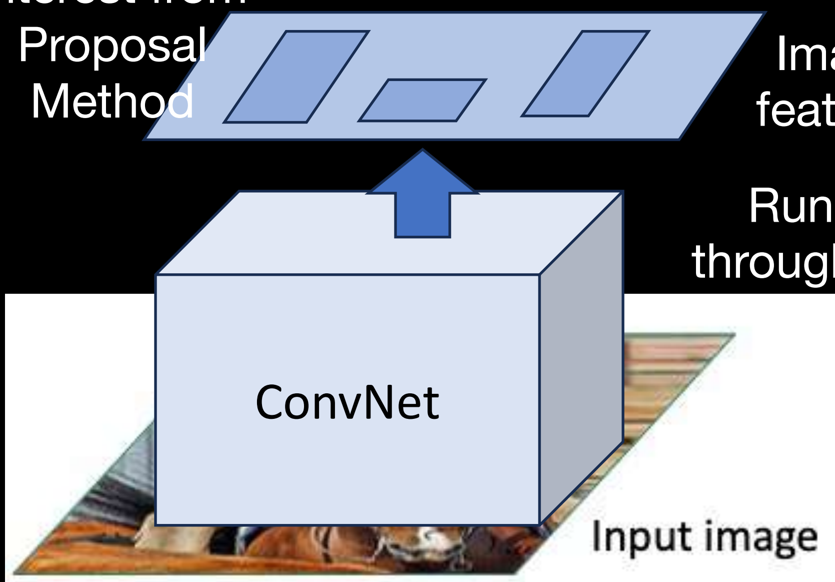
use MLP features to classify and refine bounding box forward pass for each proposal yields

slow processing

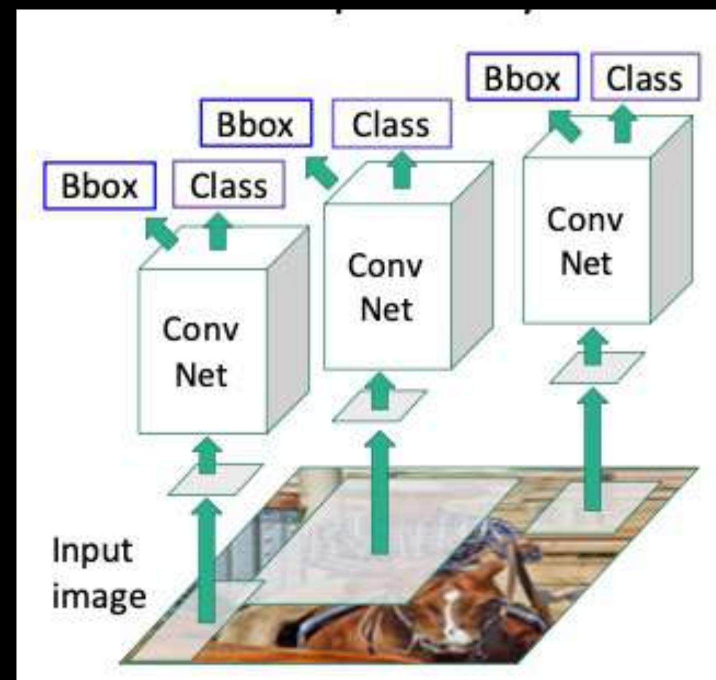
Region of Interest from Proposal Method

Image features

Run image through ConNet

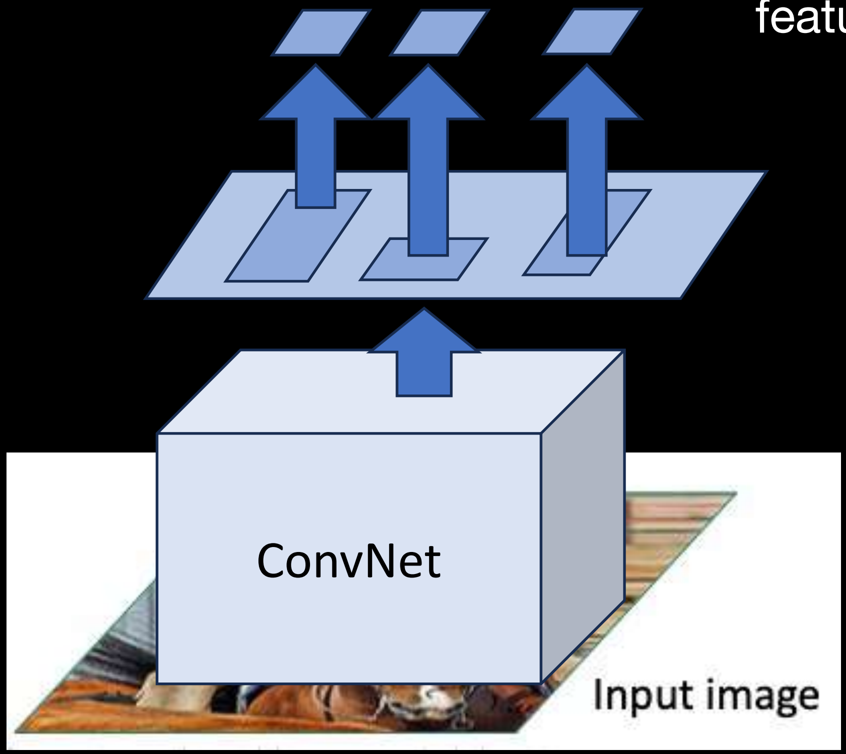


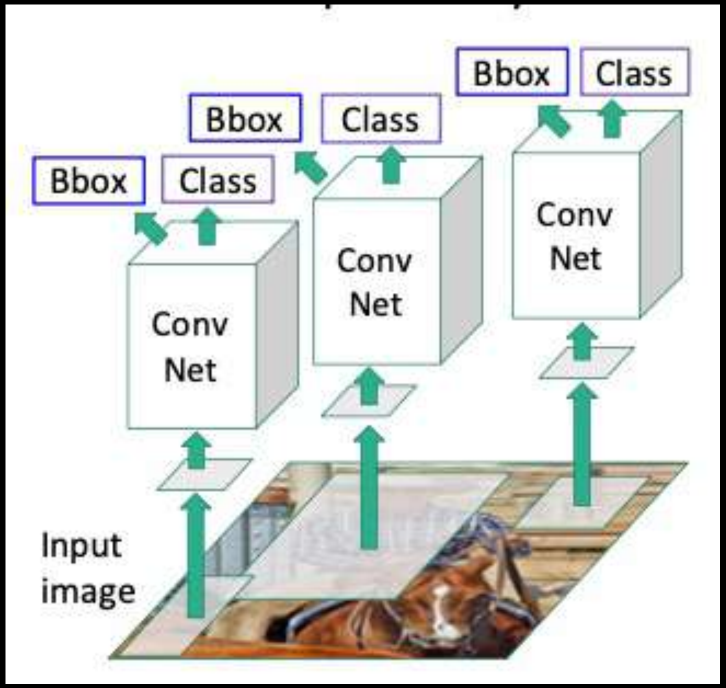
<u>"Slow" R-CNN Process</u>
each region independently

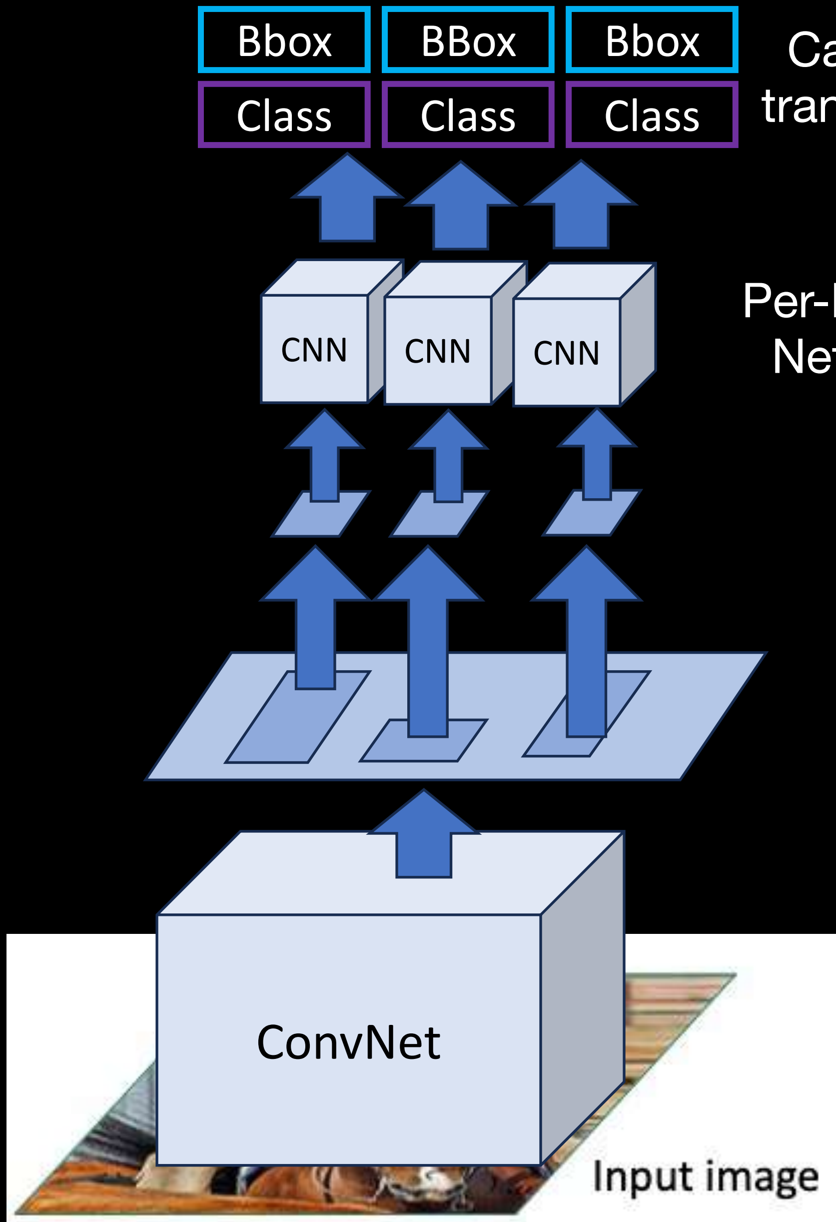


"Slow" R-CNN Process
each region
independently

Crop + Resize
features



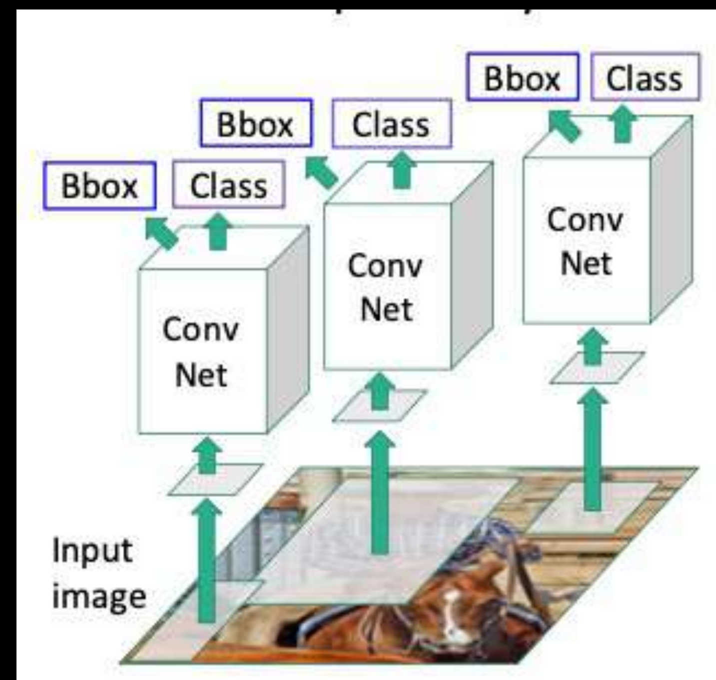


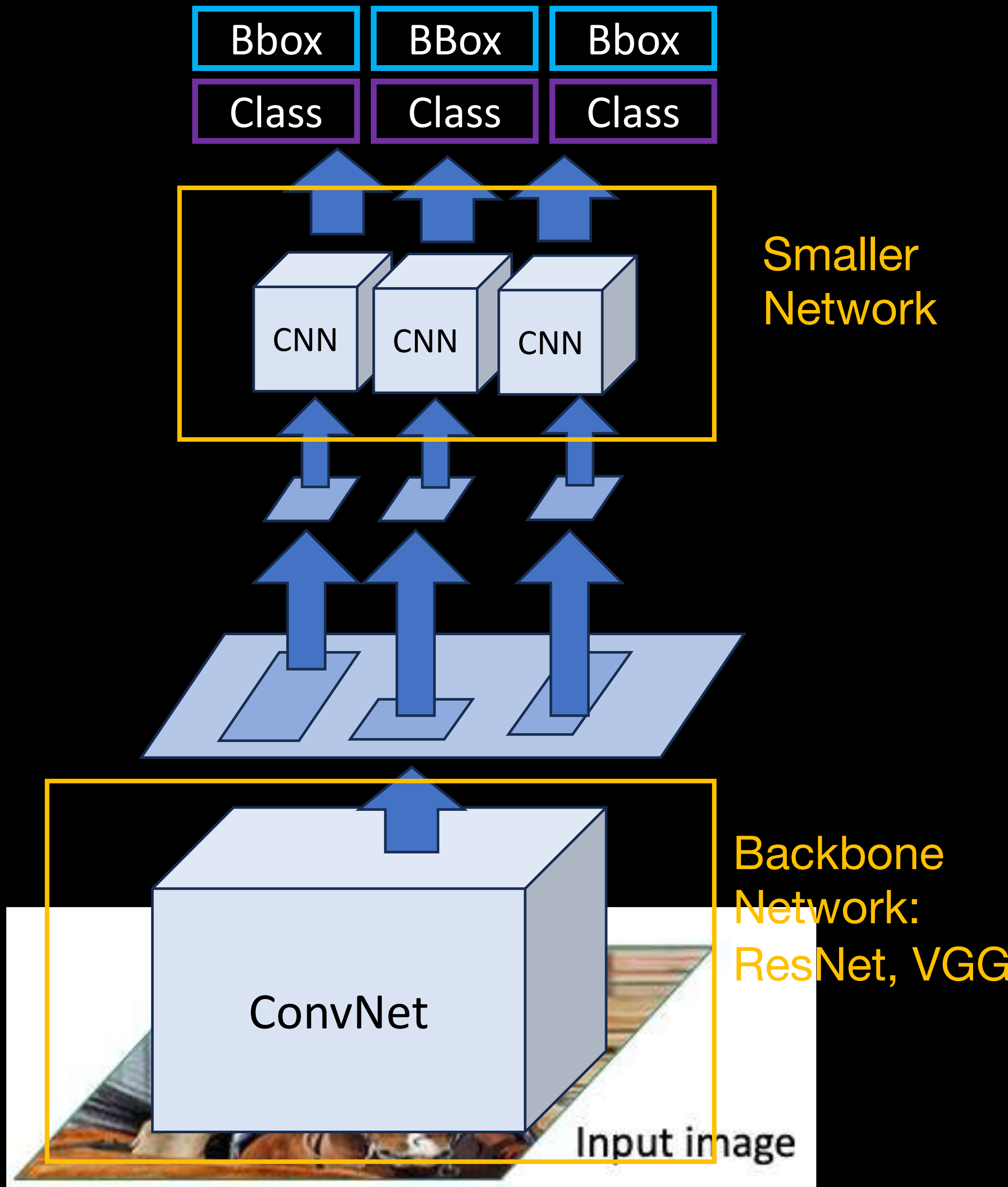


Category and box transform per region

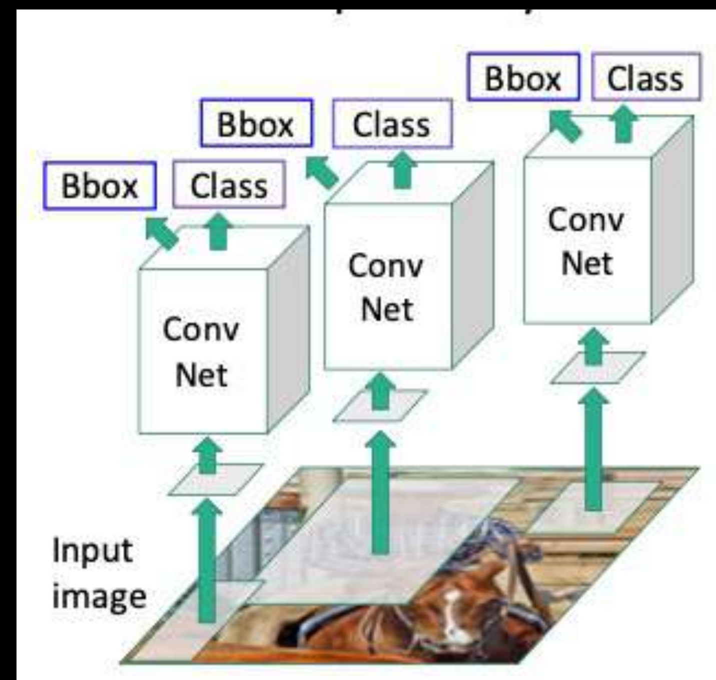
Per-Region Network

"Slow" R-CNN Process
each region independently

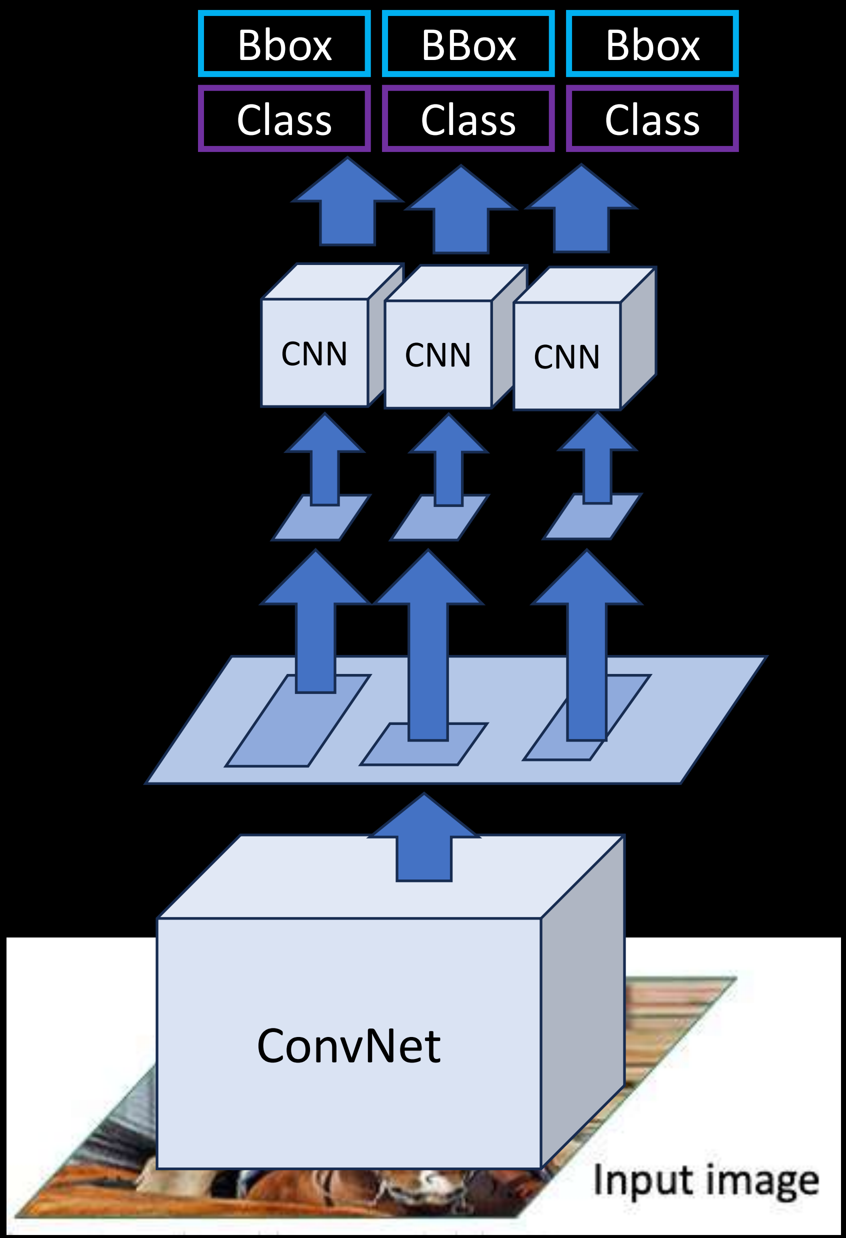




“Slow” R-CNN Process
each region independently

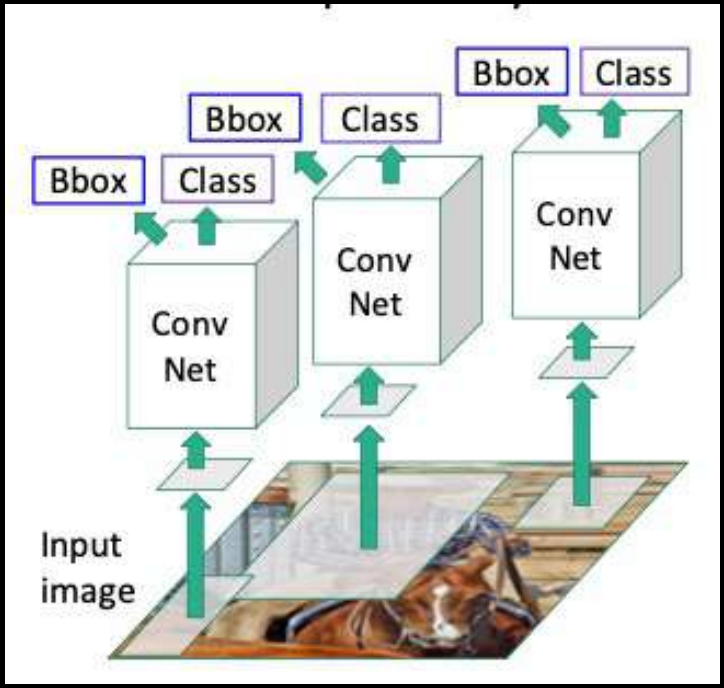


Fast R-CNN Process



"Slow" R-CNN Process

each region
independently



handcrafted region proposals are a

performance bottleneck


Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

[illegible]
Microsoft Research
{v-shren, kahe, rbg, jiansun}@microsoft.com

[illegible]

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [7] and Fast R-CNN [5] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we in[illegible] a *Region Pro-*[illegible] jectness scores at each position. RPNs are trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. With a simple alternating optimization, RPN and Fast R-CNN can be trained to share convolutional features. For the very deep VGG-16 model [19], our detection system has a frame rate of 5fps (*including all steps*) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2% mAP) and 2012 (70.4% mAP) using 300 proposals per image. Code is available at https://github.com/ShaoqingRen/faster_rcnn.

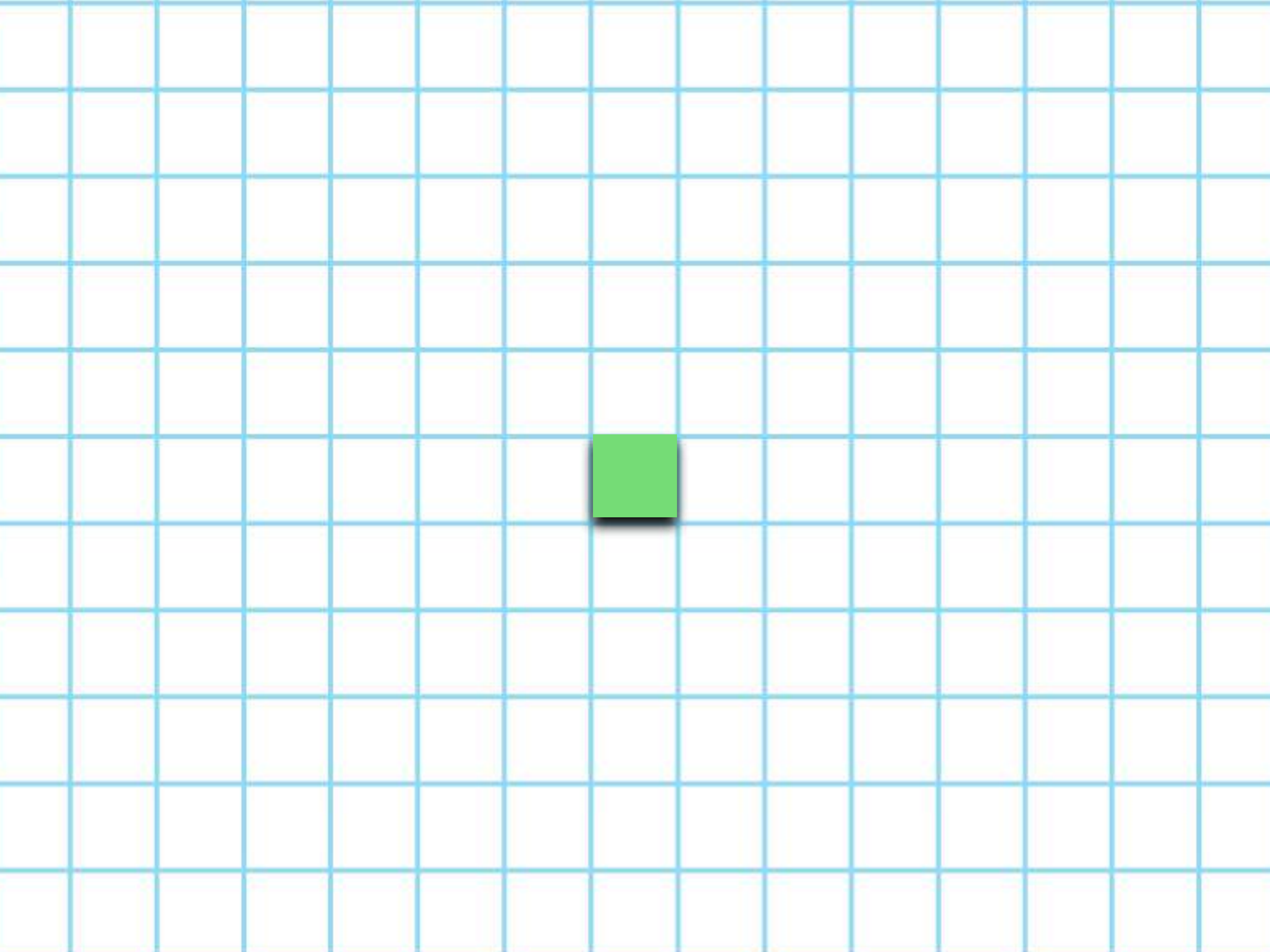
Fast R-CNN

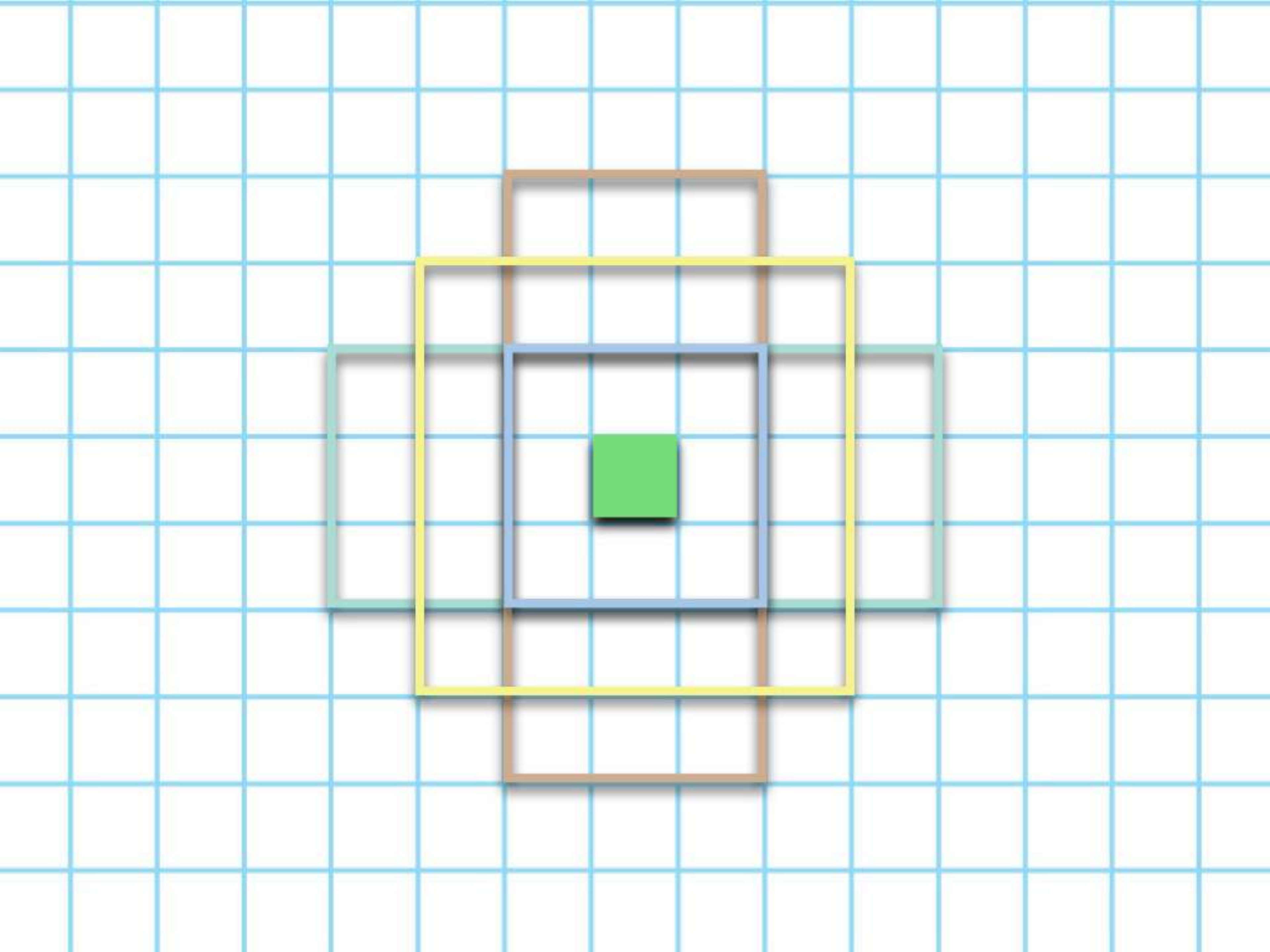
+

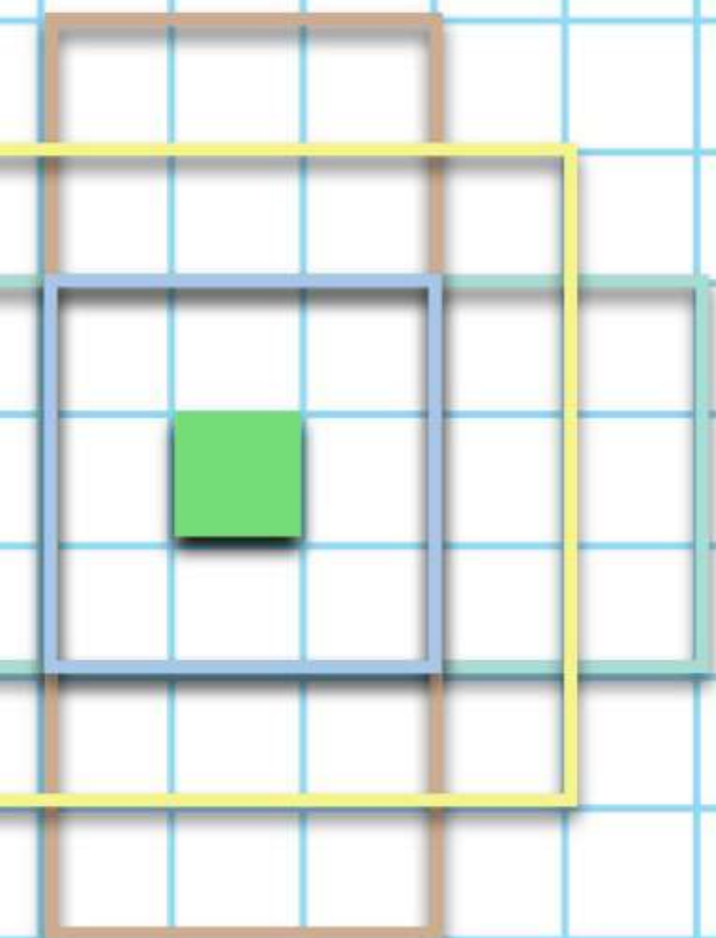
Region Proposal Network

Anchor Boxes

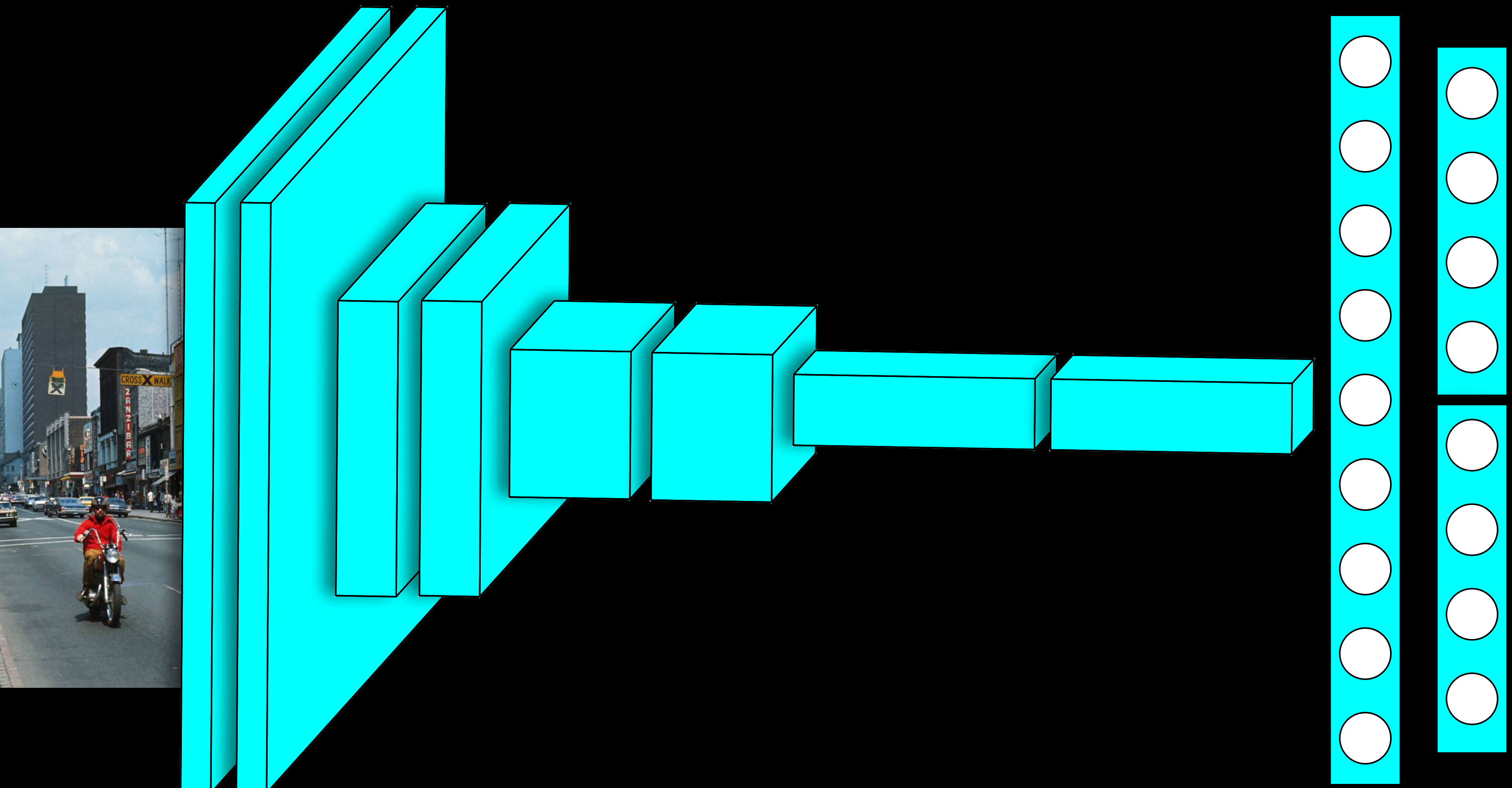
region proposals

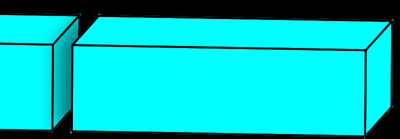




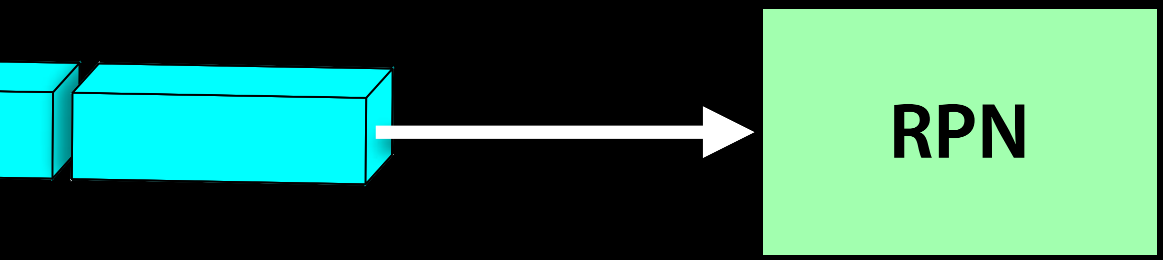


network predicts valid proposals


CROSS WALK
ZANZIBAR

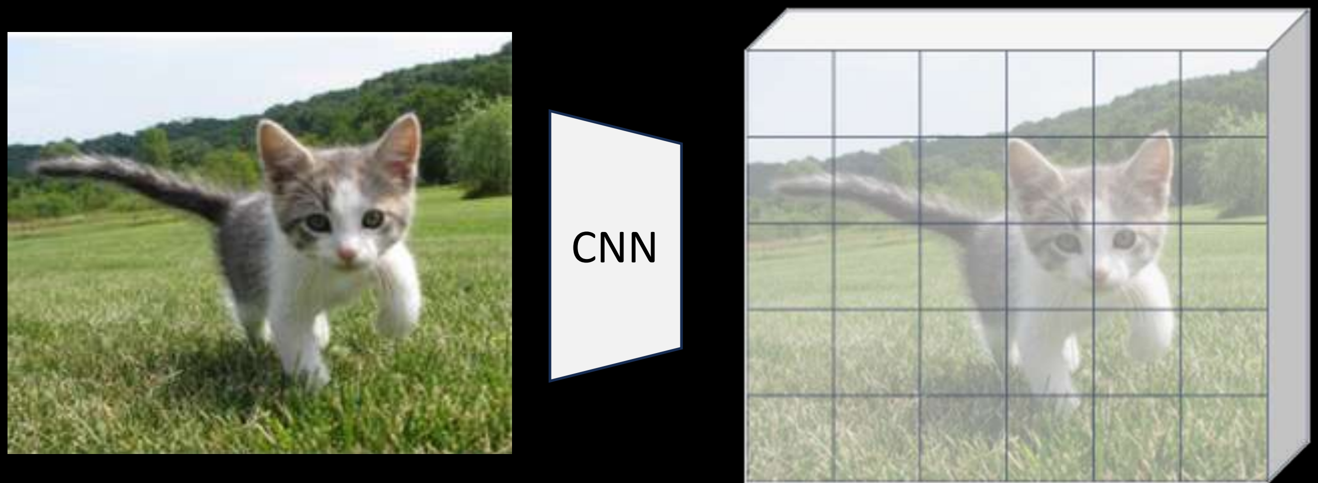


Region Proposal Network



Region Proposal Network (RPN)

Run backbone CNN to get
features aligned to input image



Input Image
(3x640x480)

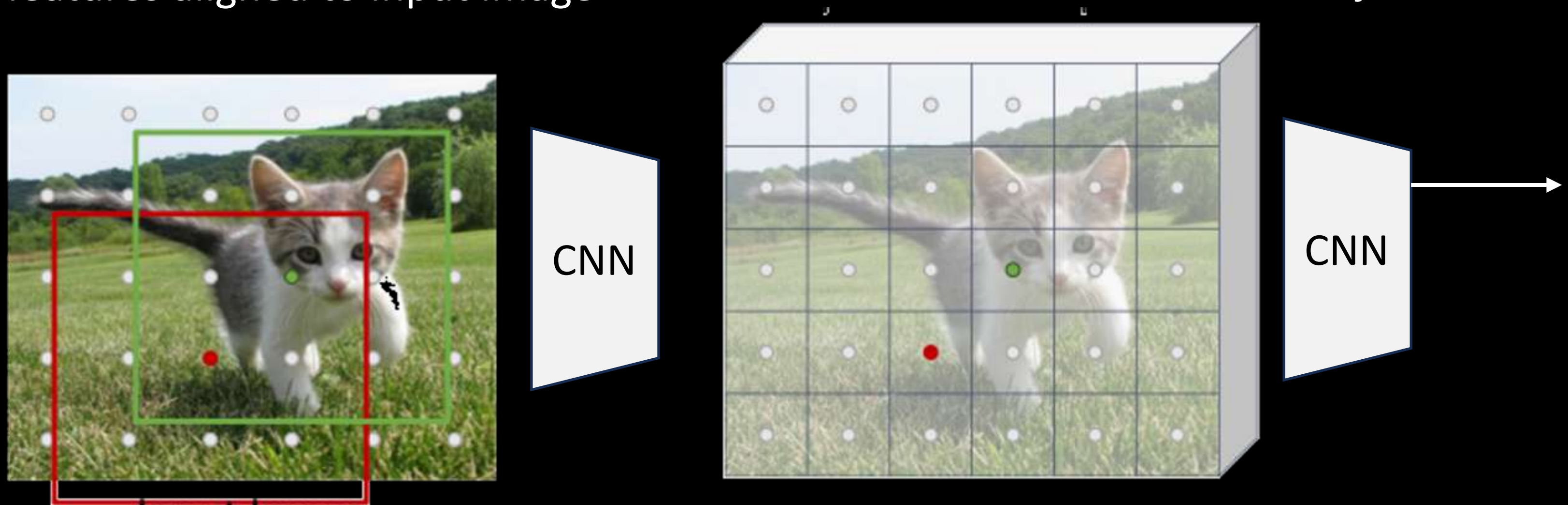
Image Features
(512x5x6)

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Predict **object** vs **not object** scores for all anchors with a conv layer



Anchor is object?
5 x 6

Input Image
(3x640x480)

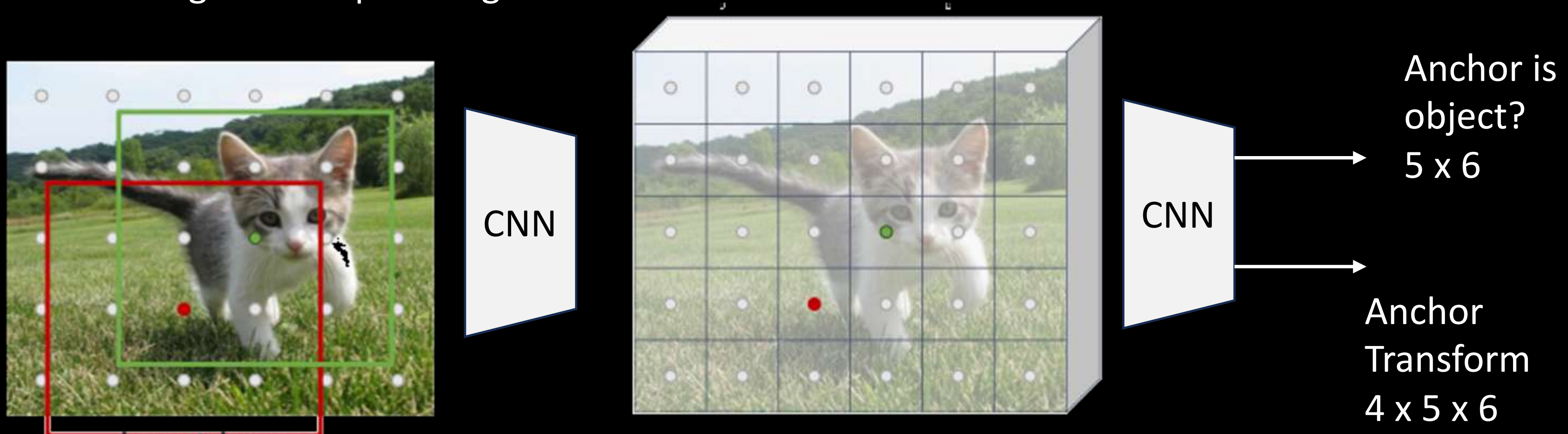
Image Features
(512x5x6)

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Predict **object** vs **not object** scores for all anchors with a conv layer

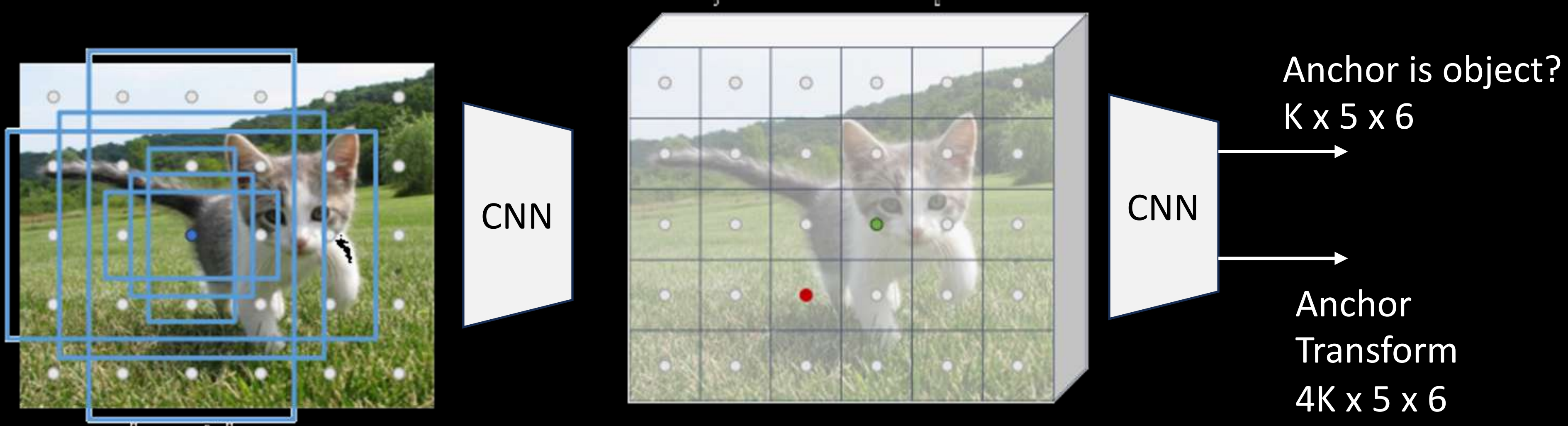


Input Image (3x640x480)

Image Features (512x5x6)

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different sizes and scale (here K=6)



Input Image
(3x640x480)

Image Features
(512x5x6)

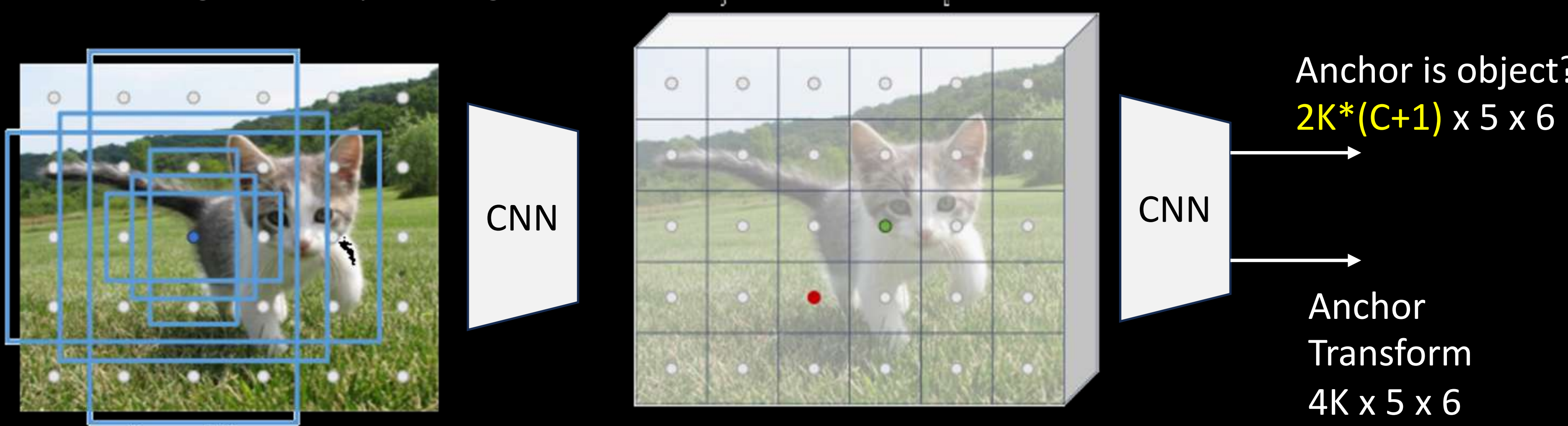
Two Stage

detectors

Single-Stage Detector: RetinaNet

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(3x640x480)

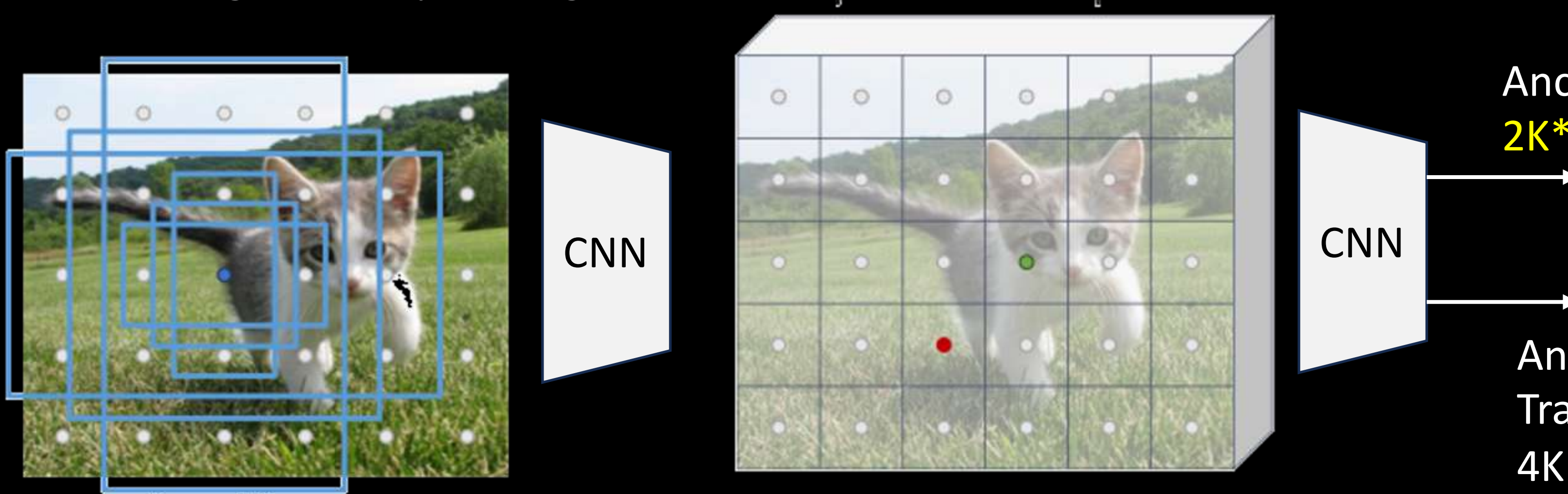
Image Features
(512x5x6)

Similar to RPN – but rather than classify anchors as object/no object, directly predict object category (among C categories) or background

Single-Stage Detector: RetinaNet

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Anchor is object?
2K*(C+1) x 5 x 6

Anchor Transform
4K x 5 x 6

Input Image
(3x640x480)

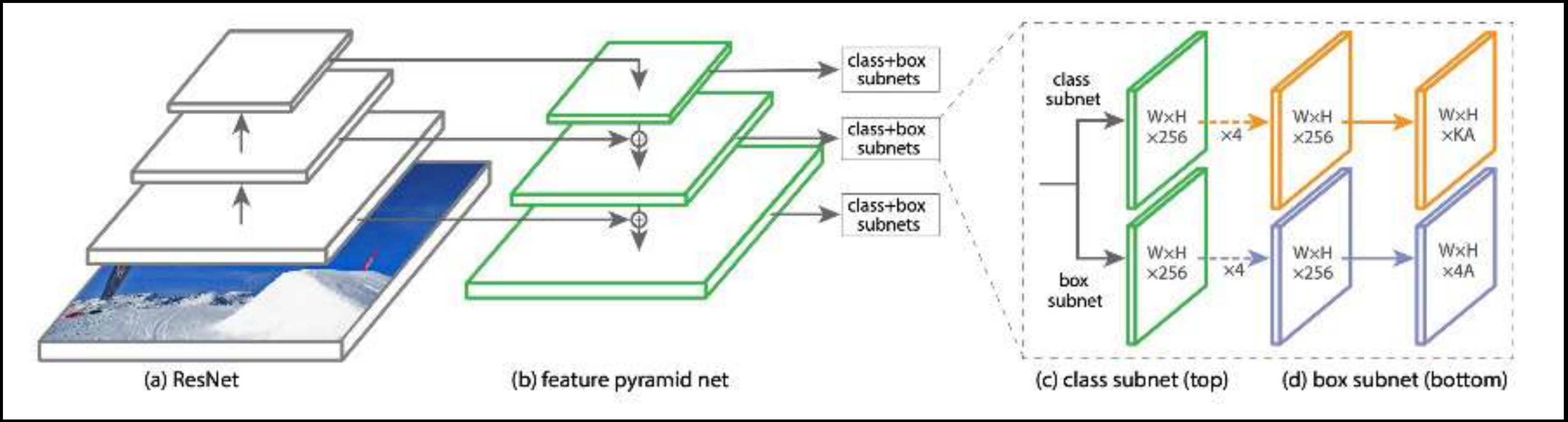
Image Features
(512x5x6)

Problem: class imbalance – many more background anchors vs non-background

Solution: new loss function (Focal Loss); see paper

RetinaNet in Practice

In practice, RetinaNet also uses Feature Pyramid Network to handle multiscale



SSD: Single Shot MultiBox Detector

Wei Liu[1], Dragomir Anguelov[2], Dumitru Erhan[3], Christian Szegedy[3], Scott Reed[4], Cheng-Yang Fu[1], Alexander C. Berg[1]

[1]UNC Chapel Hill [2]Zoox Inc. [3]Google Inc. [4]University of Michigan, Ann-Arbor
wliu@cs.unc.edu, [2]drago@zoox.com, [3]{dumitru,szegedy}@google.com,

CornerNet: Detecting Objects as Paired Keypoints

Hei Law · Jia Deng

...ct We propose CornerNet, a new approach to ...detection where we detect an object bounding ... pair of keypoints, the top-left corner and the ...right corner, using a single convolution neural

more efficient. One-stage detectors place anchor ... densely over an image and generate final box pr... tions by scoring anchor boxes and refining their co... nates through regression.

Focal Loss for Dense Object Detection

Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollár
Facebook AI Research (FAIR)

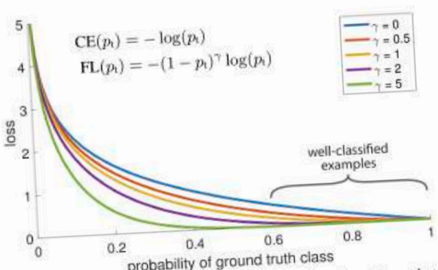


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1-p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclas... examples. As our experiments will demonstrate, the propos... training highly accurate dense object detecto... vast numbers of easy background examples.

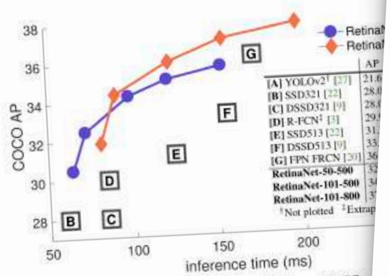


Figure 2. Speed (ms) versus accuracy (AP) on COCO t... Enabled by the focal loss, our simple one-stage Retin... tor outperforms all previous one-stage and two-stage d... cluding the best reported Faster R-CNN [28] system...

Abstract

The highest accuracy object detector... on a two-stage approach popularized b... classifier is applied to a sparse set of c... cations. In contrast, one-stage detecto... over a regular, dense sampling of possi... have the potential to be faster and simp... the accuracy of two-stage detectors thu... we investigate why this is the case. We... treme foreground-background class im... during training of dense detectors is th... propose to address this class imbalan... standard cross entropy loss such that... loss assigned to well-classified examp... Loss focuses training on a sparse set... prevents the vast number of easy negat... ing the detector during training. To e... ness of our loss, we design and train a... we call RetinaNet. Our results show t... the focal loss, RetinaNet is able to m...

Bottom-up Object Detection by Grouping Extreme and Center Points

Xingyi Zhou
UT Austin
zhouxy@cs.utexas.edu

Jiacheng Zhuo
UT Austin
jzhuo@cs.utexas.edu

Philipp Krähenbühl
UT Austin
philkr@cs.utexas.edu

Abstract

With the advent of deep learning, object detection drifted from a bottom-up to a top-down recognition problem. State of the art algorithms enumerate a near-exhaustive list of object locations and classify each into...

...1: We propose to detect objects by finding their ex... points. They directly form a bounding box , but also much tighter octagonal approximation of the object.

...his paper, we propose ExtremeNet, a bottom-up ob... tection framework that detects four extreme points ...ost, left-most, bottom-most, right-most) of an ob... We use a state-of-the-art keypoint estimation frame... [..., 5, 30, 31, 49] to find extreme points, by predicting ...ulti-peak heatmaps for each object category. In ad... we use one heatmap per category predicting the ob... ter, as the average of two bounding box edges in x and y dimension. We group extreme points into ... with a purely geometry-based approach. We group ...eme points, one from each map, if and only if their ...ic center is predicted in the center heatmap...

Objects as Points

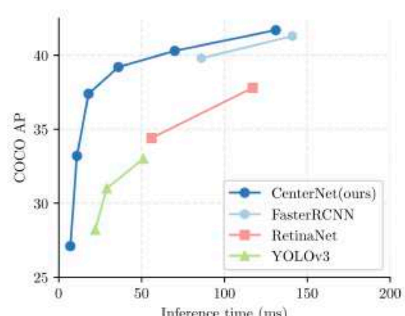
Xingyi Zhou
UT Austin
zhouxy@cs.utexas.edu

Dequan Wang
UC Berkeley
dqwang@cs.berkeley.edu

Philipp Krähenbühl
UT Austin
philkr@cs.utexas.edu

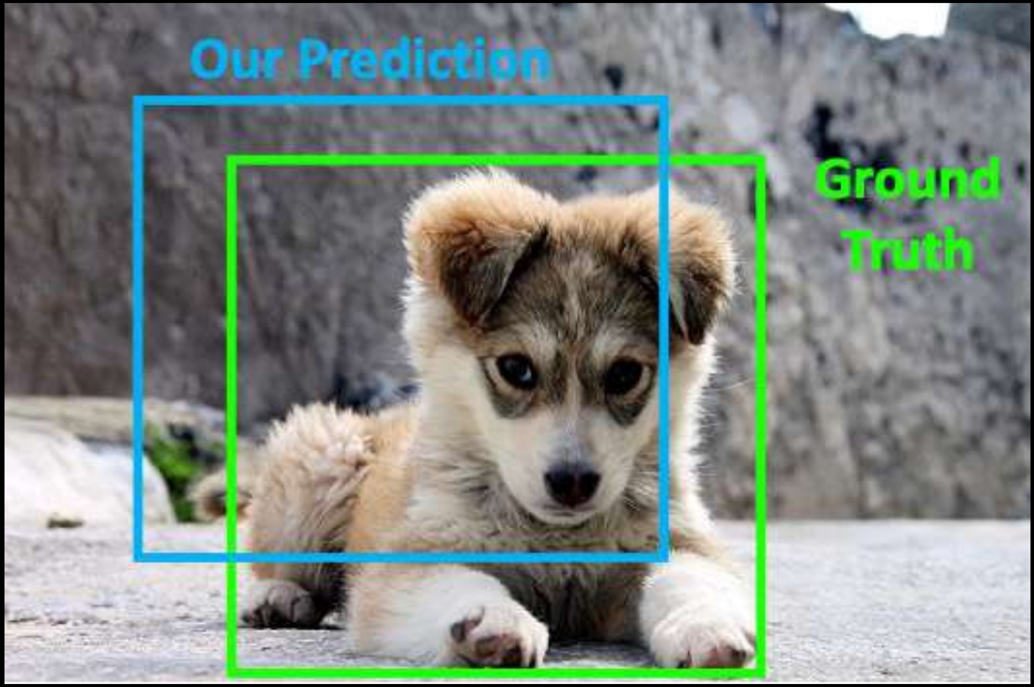
Abstract

Detection identifies objects as axis-aligned boxes in an image. Most successful object detectors enumerate a nearly exhaustive list of potential object locations and classify each. This is wasteful, inefficient, and requires additional post-processing. In this paper, we take a different approach. We model an object as a single point — the center point of its bounding box. Our detector uses keypoint estimation to find center points and regresses to all other object properties, such as size, 3D location, orientation, and even pose. Our center point based approach, CenterNet, is end-to-end differentiable, simpler, faster, and more accurate than corresponding bounding box based detectors...



Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?



Comparing Boxes: Intersection over Union (IoU)

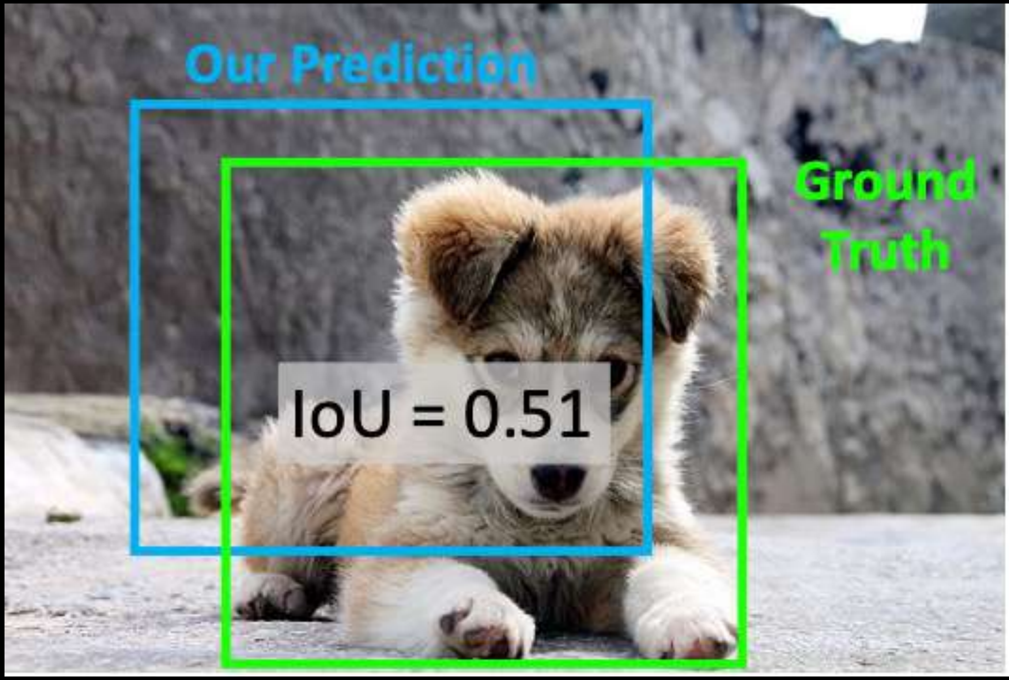
How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?



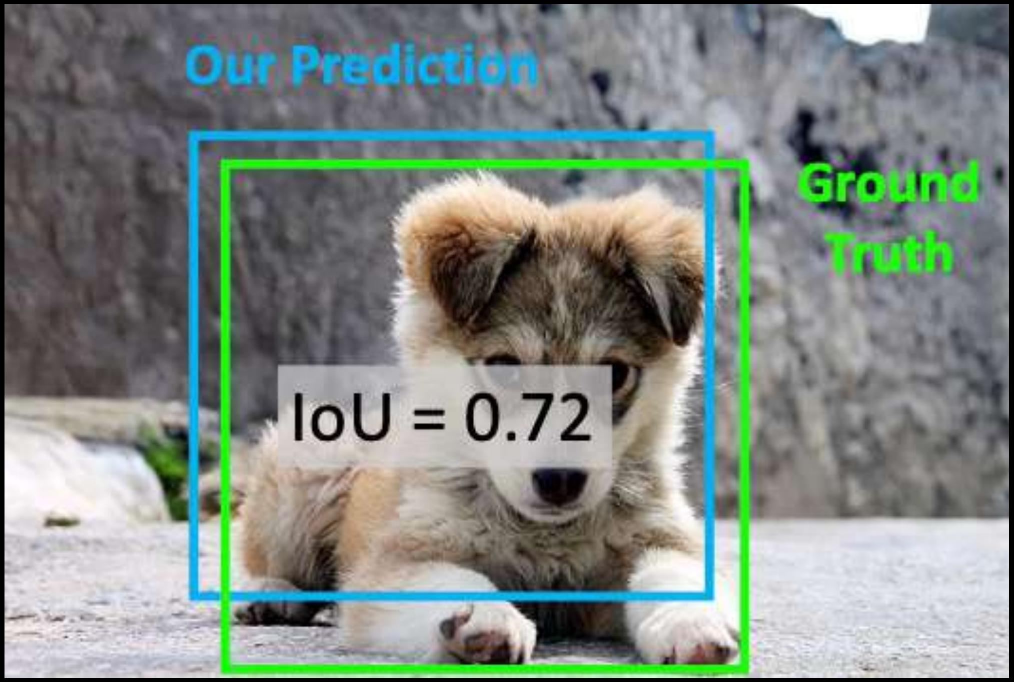
Intersection over Union (IoU) (Also called “Jaccard similarity” or “Jaccard index”):

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

IoU > 0.5 is “decent”

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?



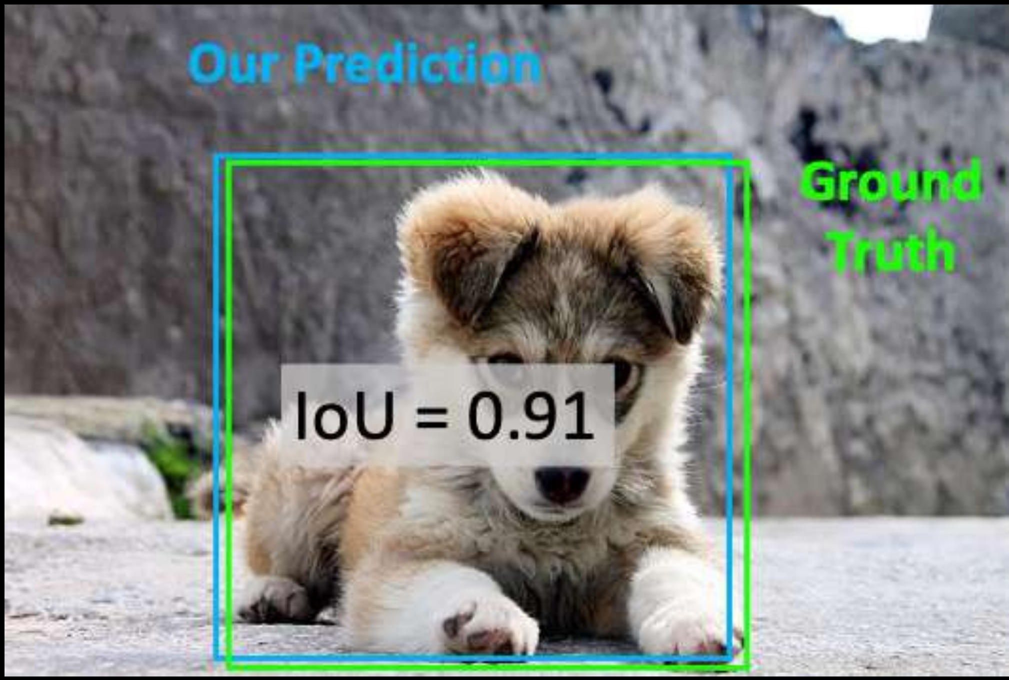
Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

IoU > 0.5 is "decent"
IoU > 0.7 is "pretty good",

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?



Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

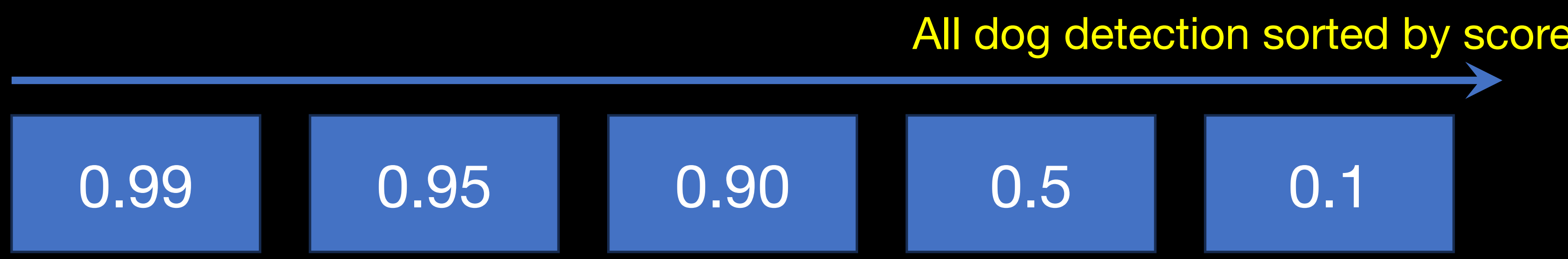
$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

IoU > 0.5 is "decent"
IoU > 0.7 is "pretty good",
IoU > 0.9 is "almost perfect"

Evaluating Object Detectors: Mean Average Precision (mAP)

1. Run object detector on all test images
2. For **each category**, compute Average Precision (AP)= area under Precision vs Recall Curve

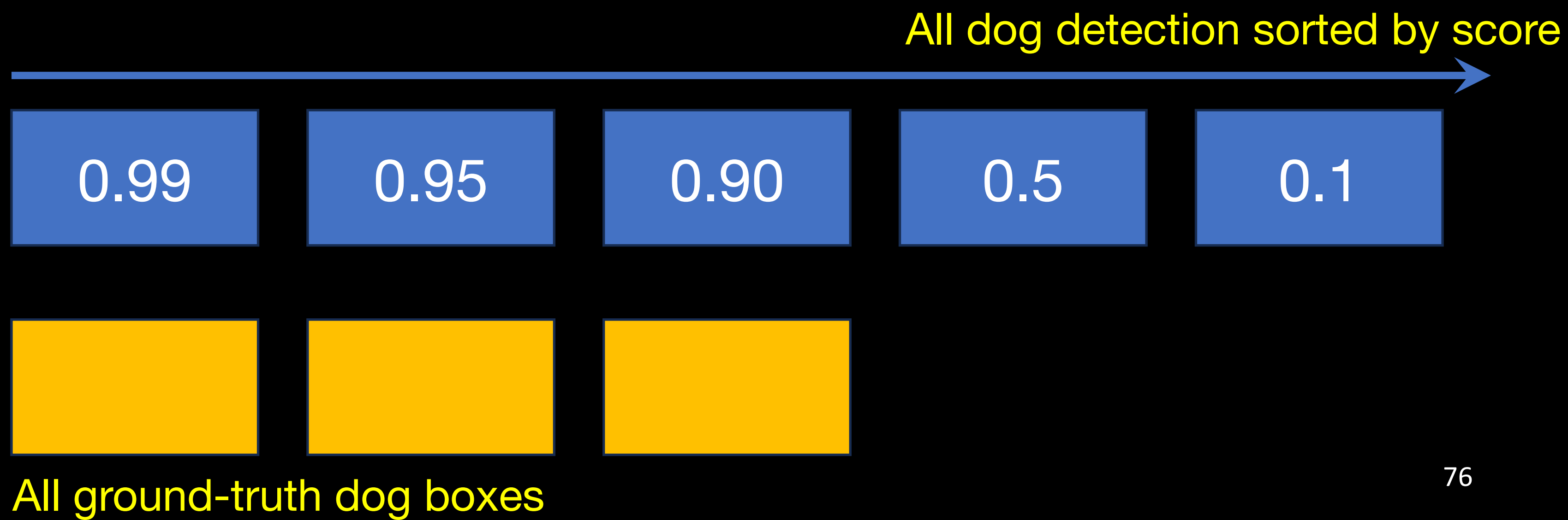
   - For each detection (highest score to lowest score)



Evaluating Object Detectors: Mean Average Precision (mAP)

1. Run object detector on all test images
2. For **each category**, compute Average Precision (AP)= area under Precision vs Recall Curve

   - For each detection (highest score to lowest score)

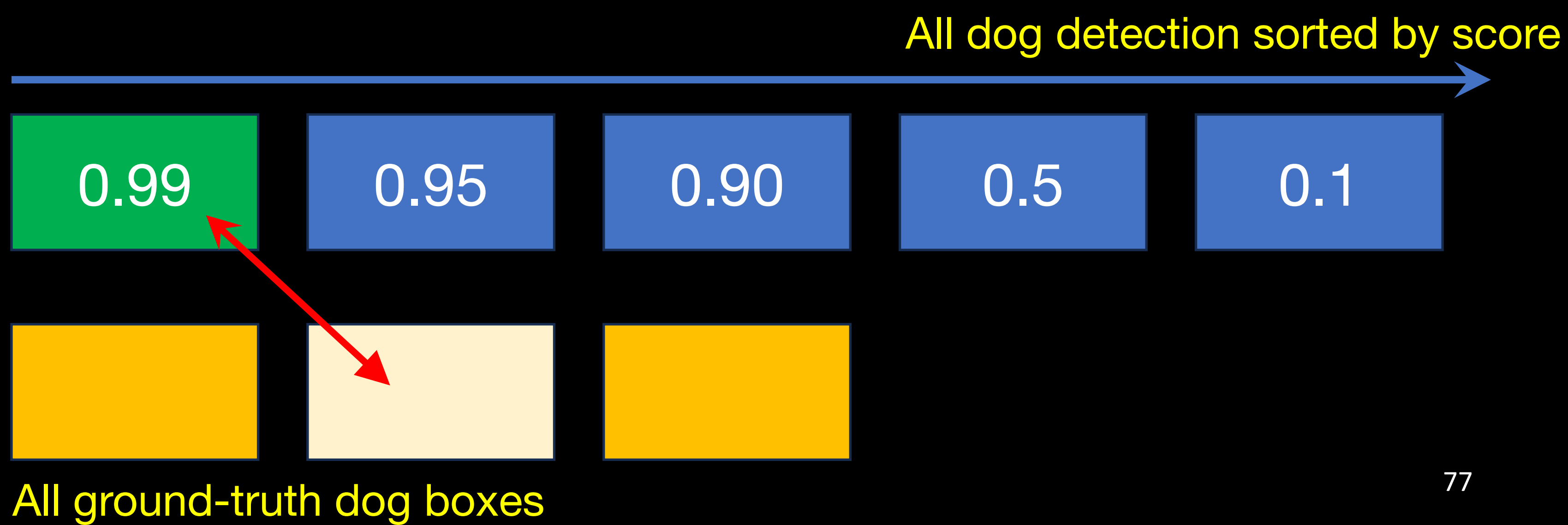


Evaluating Object Detectors: Mean Average Precision (mAP)

1. Run object detector on all test images
2. For **each category**, compute Average Precision (AP)= area under Precision vs Recall Curve

   - For each detection (highest score to lowest score)

     - If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT



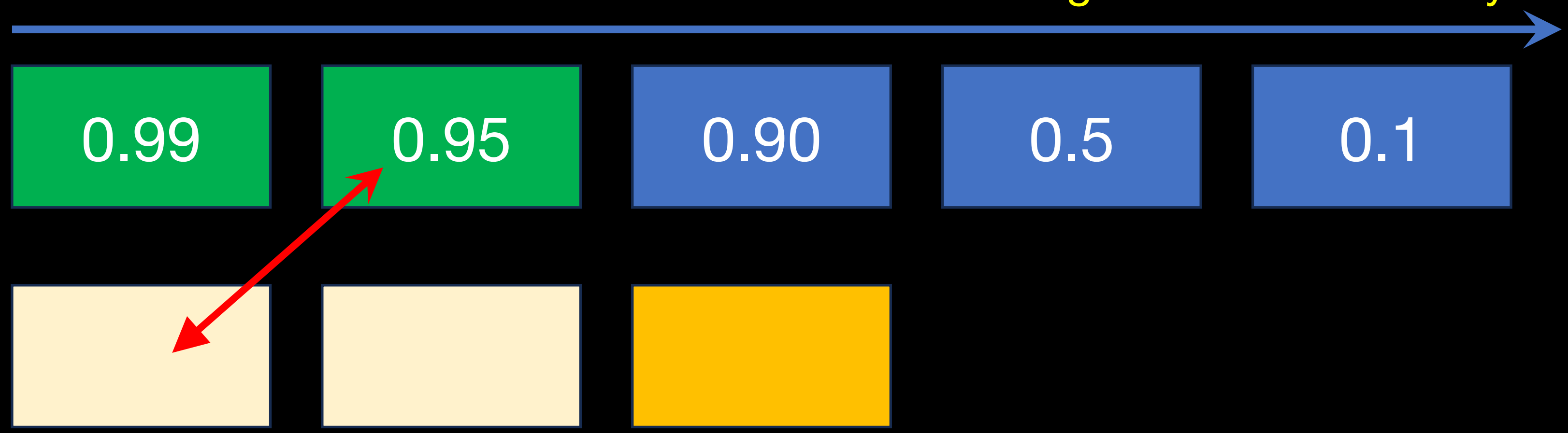
Evaluating Object Detectors: Mean Average Precision (mAP)

1. Run object detector on all test images
2. For **each category**, compute Average Precision (AP)= area under Precision vs Recall Curve

   - For each detection (highest score to lowest score)

     - If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT

All dog detection sorted by score



All ground-truth dog boxes

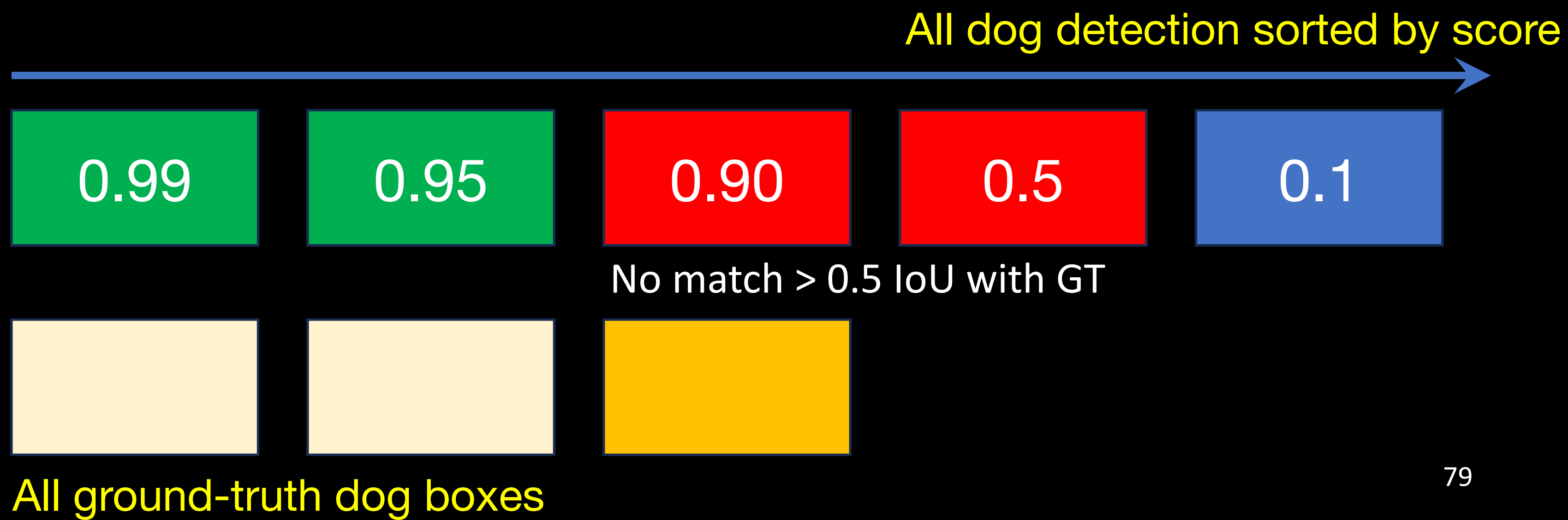
Evaluating Object Detectors: Mean Average Precision (mAP)

1. Run object detector on all test images
2. For **each category**, compute Average Precision (AP)= area under Precision vs Recall Curve

   - For each detection (highest score to lowest score)

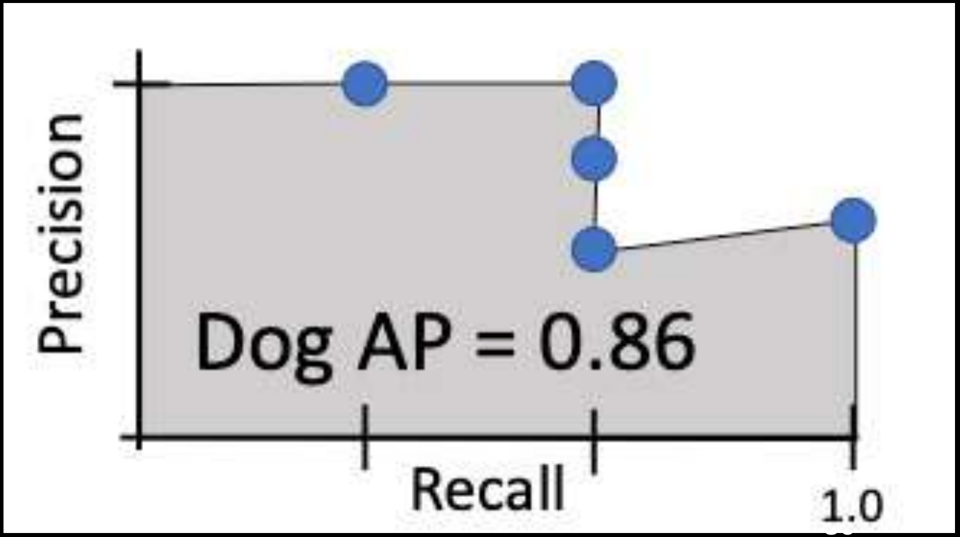
     - If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT

     - Otherwise, mark it as negative



Evaluating Object Detectors: Mean Average Precision (mAP)

1. Run object detector on all test images
2. For **each category**, compute Average Precision (AP)= area under Precision vs Recall Curve
   - For each detection (highest score to lowest score)
     - If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
     - Otherwise, mark it as negative
     - Plot a point on PR Curve
   - Average Precision (AP)



This is what is called mAP@0.5

Current State of the Art in Computer Vision

Network	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster RCNN+ FPN	ResNet-101	36.2	59.1	39.0	18.2	39.0	48.2
Mask RCNN	ResNet-101	38.2	60.3	41.7	20.1	41.1	50.2
Cascade RCNN	ResNet-101	**42.8**	**62.1**	**46.3**	23.7	45.5	55.2
YOLO v2	DarkNet	21.6	44.0	19.2	5.0	22.4	35.5
SSD	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
Retina Net	ResNet-101	40.1	59.6	43.5	23.4	42.7	50.2
Corner net	Hourglass	40.5	56.5	43.1	19.4	42.7	53.9
Fovea box	ResNetXt-101	42.1	61.9	45.2	**24.9**	**46.8**	**55.6**
FCOS		38.6	57.4	41.4	22.3	42.5	49.8

Summary

- Single-Step approaches are competitive with two-step these days
- Object Detection is difficult. You don’t need to implement it yourself
  - **Detectron2 (PyTorch):** https://github.com/facebookresearch/detectron2
  - ….