Introduction to Deep Learning

16. Object Detection

STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

Homework 6 Winners

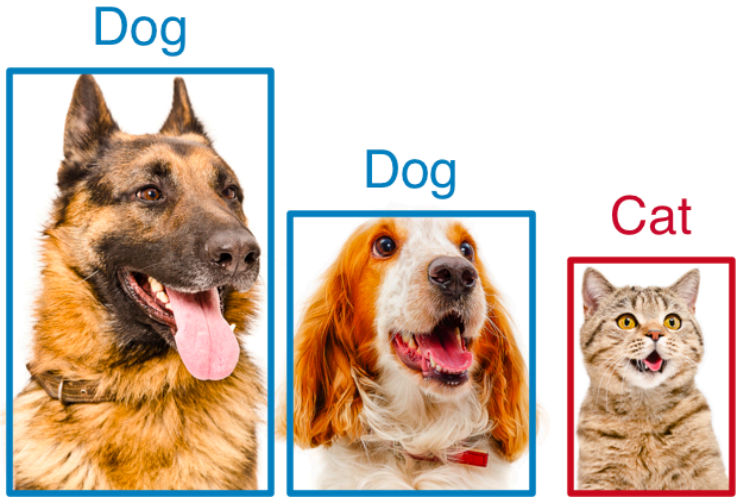
- 1st: 0.9489. Zabin Bashar, Jilin Cao, Mike Jin, Daniel Kim
  - Rank 2 on the Kaggle leaderboard!
  - Used ResNet-164
- 2nd: 0.9334. Andrew Tan, Andrew Peng, Farbod Nowzad, Ajay Shah3rd
- 3rd: 0.9264. Hanmaro Song, Minjune Hwang, Kyle Nguyen, Joanne Chen, Kyle Cho

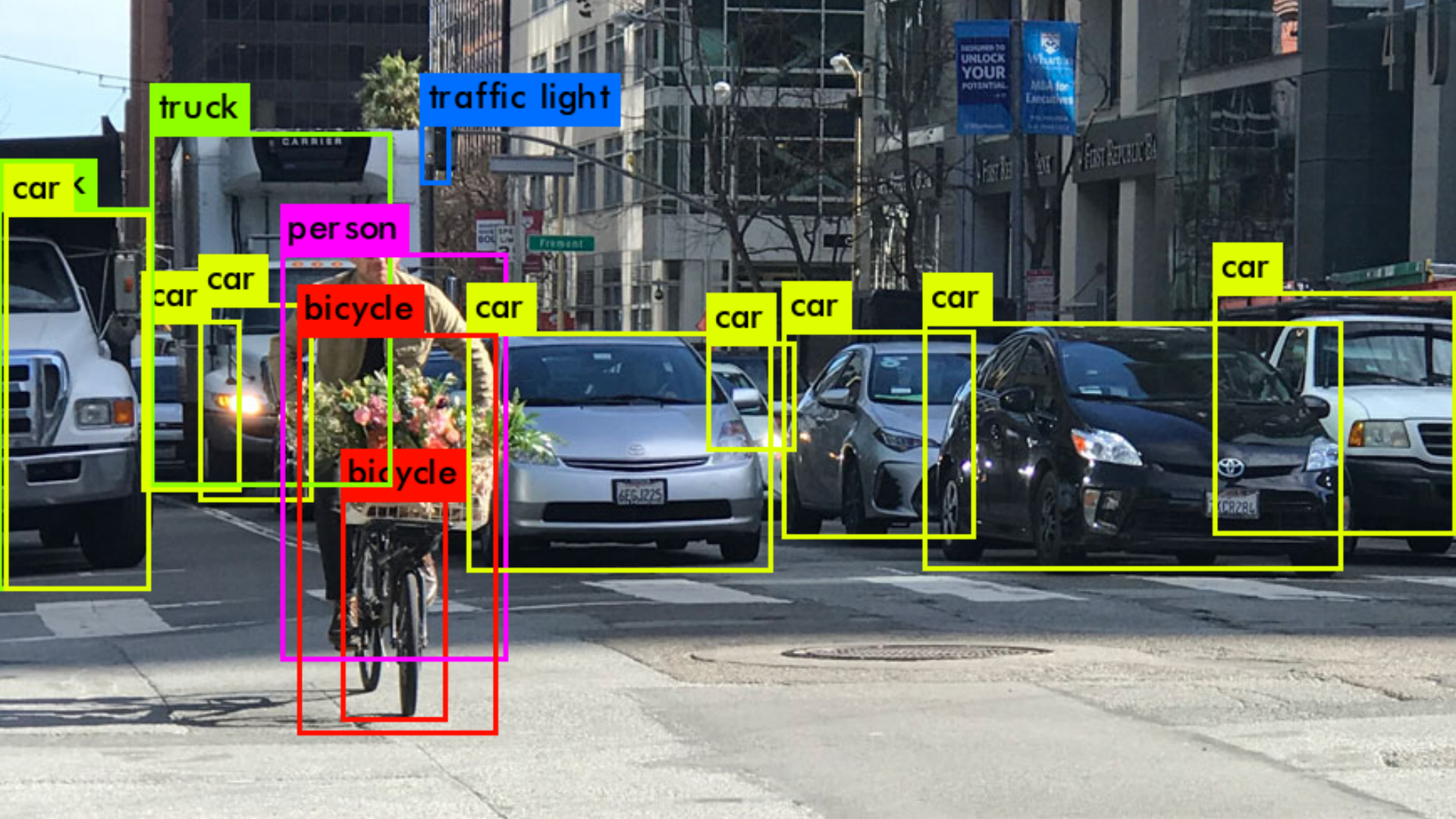
Projects

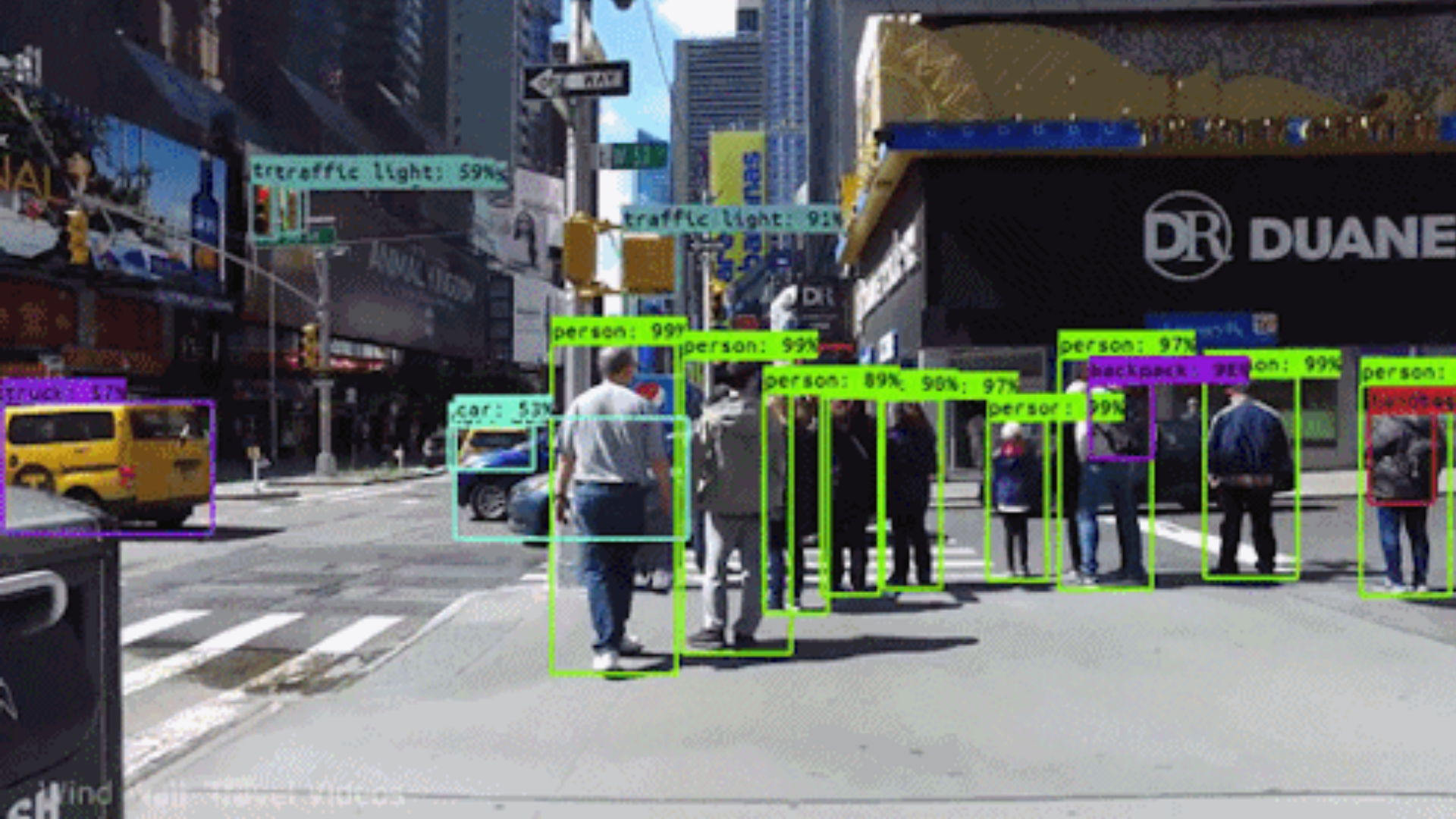
- Projects to build new applications
  - Start to collect data
  - If it's from Kaggle, check it's not too easy or too hard
  - Use fine-tuning to get results quickly
    - GluonCV, GluonNLP, or any other model zoo
- Projects to try new model architectures
  - Google if someone else already tried it
  - Get results as early as possible
    - May take too long to train, may not converge

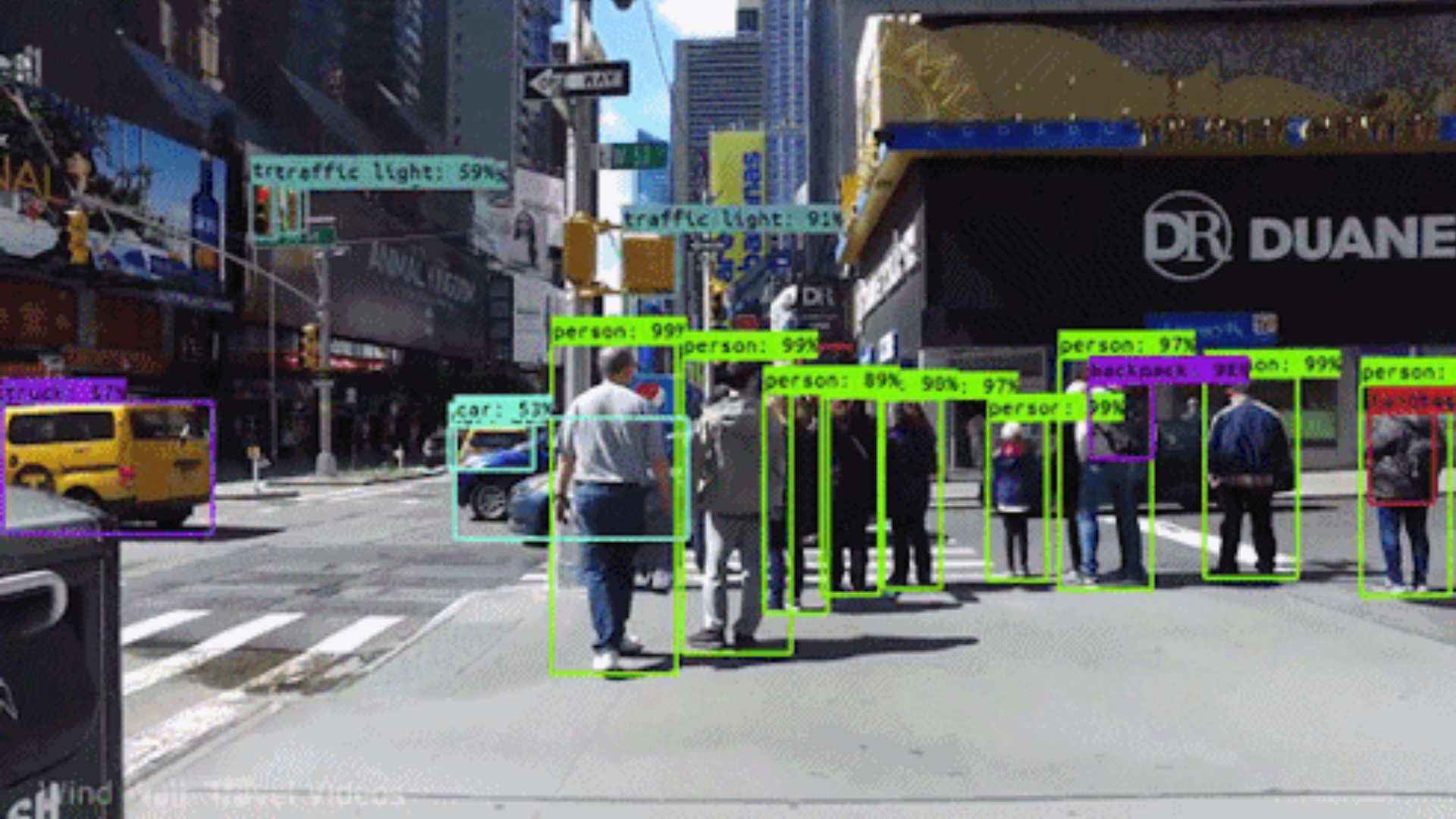
Image classification

Object Detection




truck
traffic light
car
person
car
car
bicycle
car
car
car
car
car
bicycle

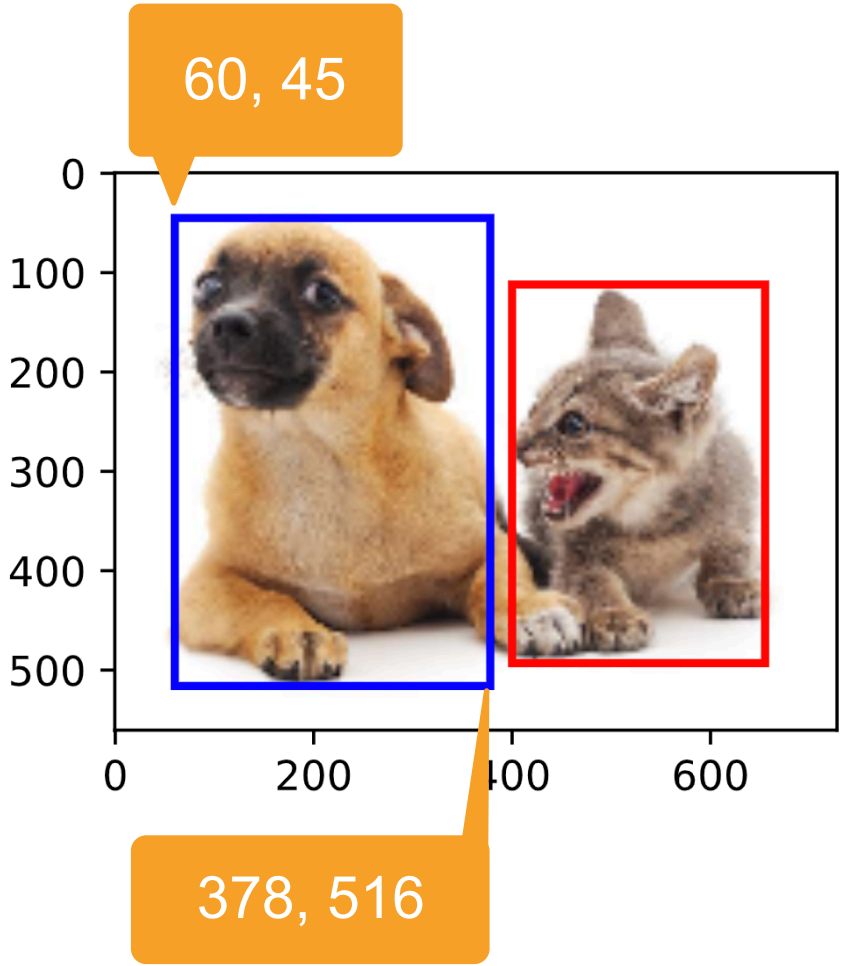

DUANE
traffic light: 59%
traffic light: 91%
person: 99%
person: 99%
person: 89%
person: 98%
person: 97%
person: 97%
backpack:
person: 99%
person:
car: 53%


DUANE
traffic light: 59%
traffic light: 91%
person: 99%
person: 99%
person: 89%
person: 98%
person: 97%
person: 97%
backpack:
person: 99%
person:
car: 53%

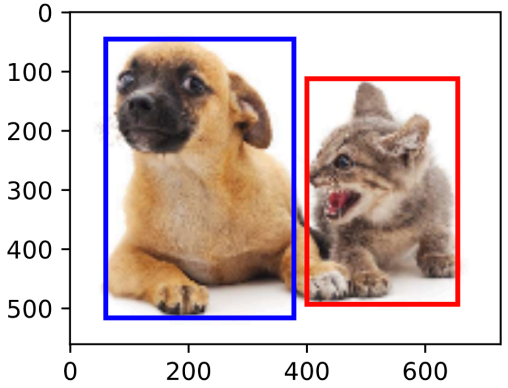
Bounding and Anchor Boxes

Bounding Box

- A bounding box can be defined by 4 numbers,
  - (top-left x, top-left y, bottom-right x, bottom-right y)
  - (top-left x, top-right y, width, height)



Object Detection Dataset



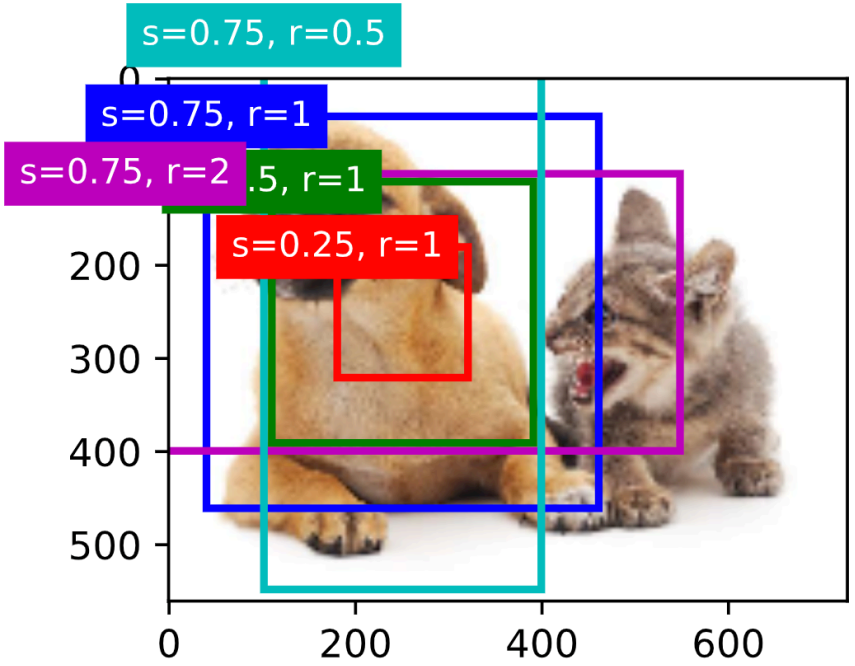
- Each row present an object
  - Image_filename, object_category, bounding box
- COCO (cocodataset.org)
  - 80 objects
  - 330K images
  - 1.5M objects

cou

aws

Anchor Boxes

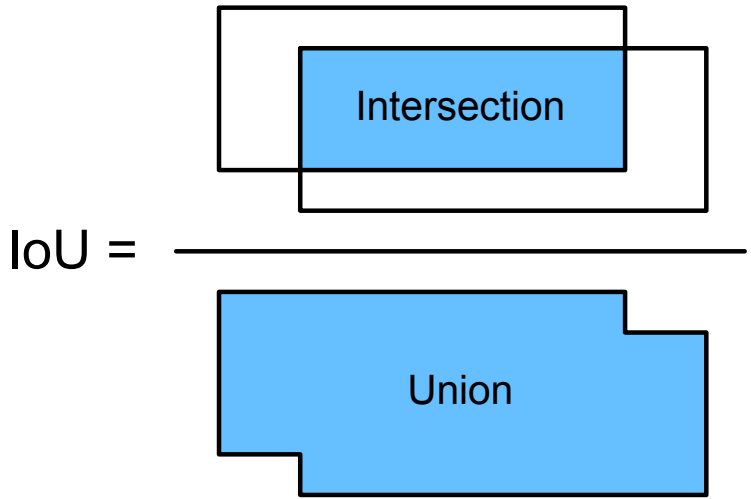
- A detection algorithm often
  - Proposes multiple regions, called anchor boxes
  - Predict if an anchor box contains an object
  - If yes, predict the offset from the anchor box to the ground truth bounding box



IoU - Intersection over Union

- IoU measures the similarity between two boxes
  - 0 means no-overlapping
  - 1 means identical
- It's an especial case of Jacquard index
  - Given sets *A* and *B*

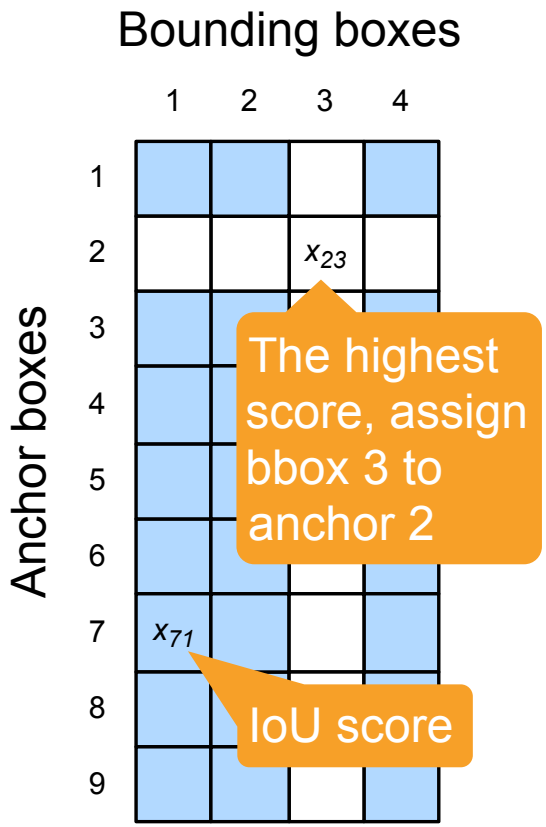
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



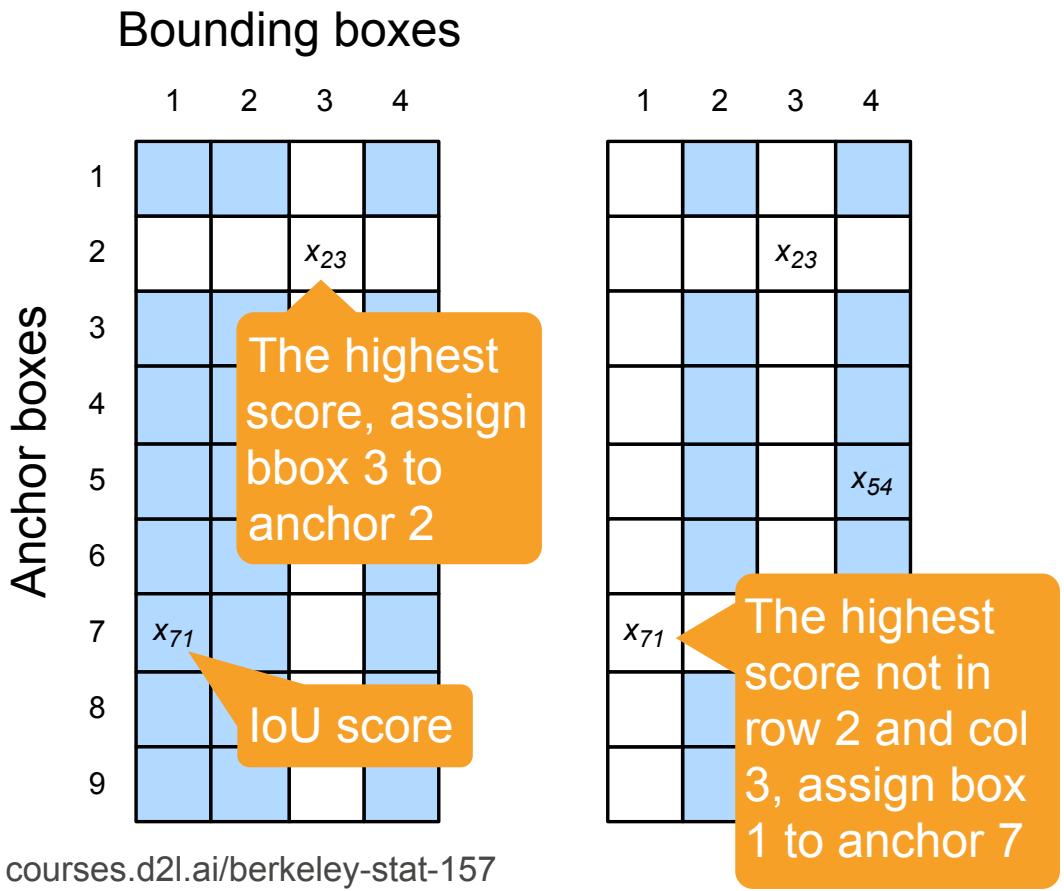
Assign Labels to Anchor Boxes

- Each anchor box is a training example
- Label each anchor box with
  - Background
  - Associate with a bounding box
- We may generate a large amount of anchor boxes
  - Leads to a large portion of negative examples

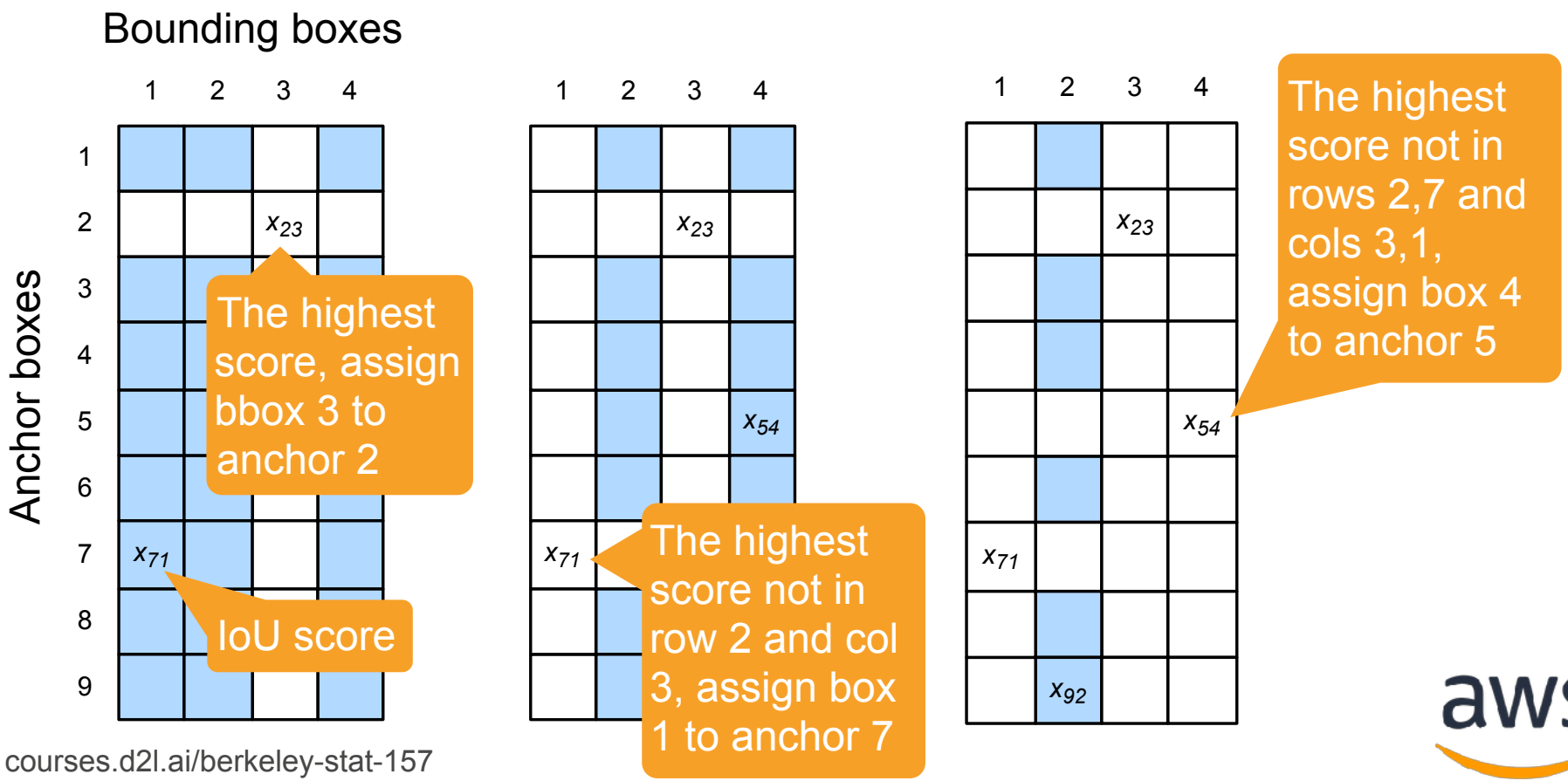
Assign Labels to Anchor Boxes



Assign Labels to Anchor Boxes

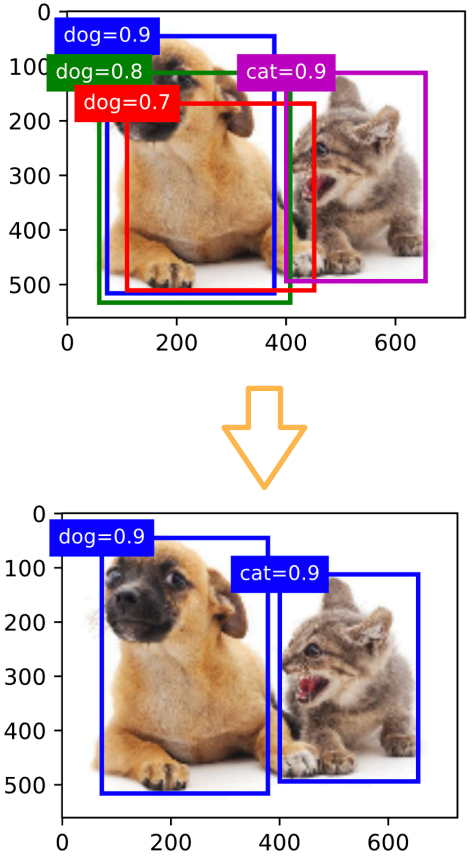


Assign Labels to Anchor Boxes

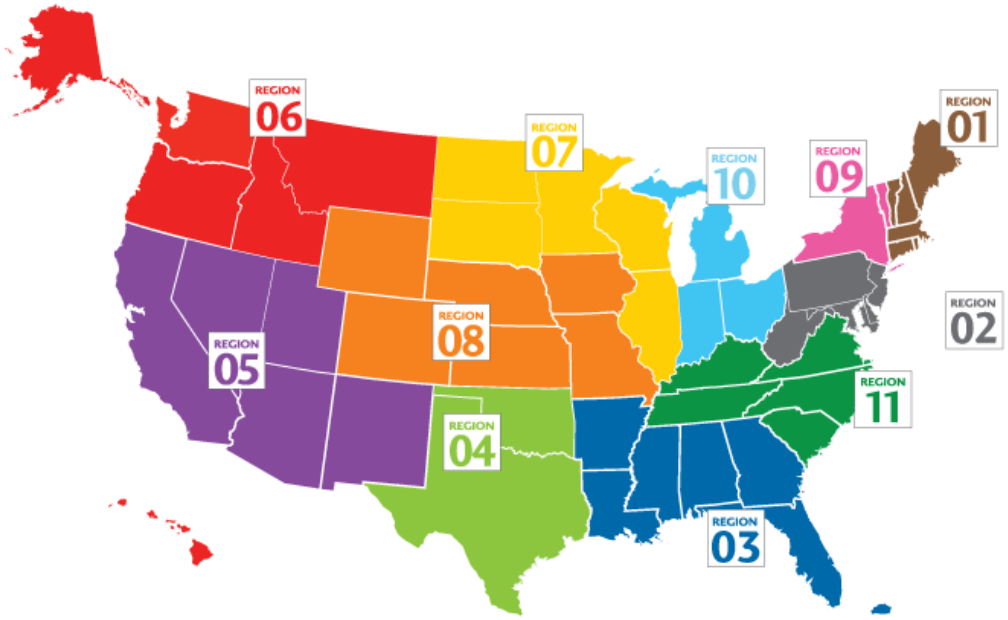


Output with non-maximum suppression (NMS)

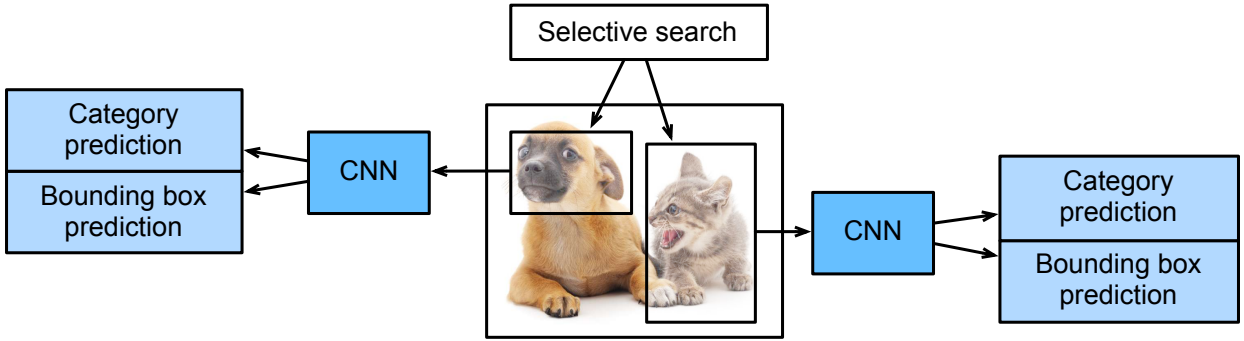
- Each anchor box generates one bounding box prediction
- Select the one with the highest score (not background)
- Remove all other predictions with IoU > θ compared to the selected one
- Repeat until all are selected or removed



Region-based CNNs

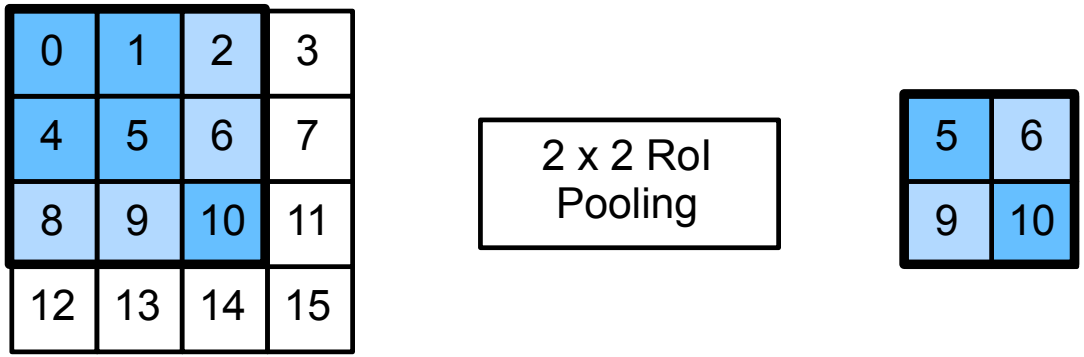


R-CNN



- Select anchor boxes with a heuristic algorithm
- Use a pre-trained networks to extract features for each anchor box
- Train a SVM to classify category
- Train a linear regression to predict bounding boxes

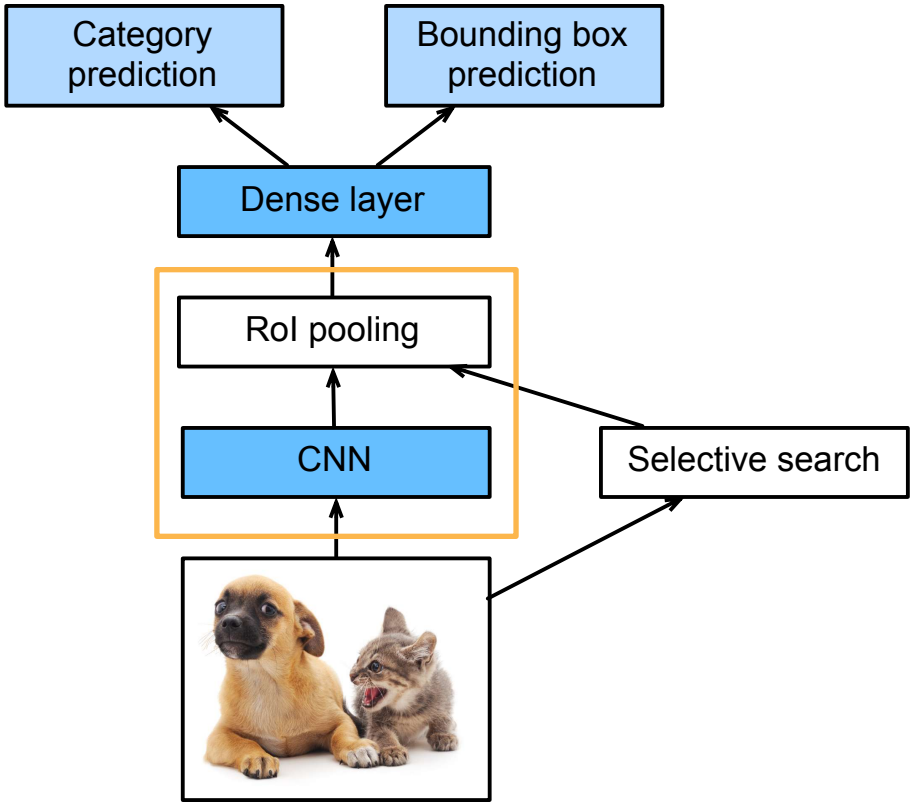
Region of Interest (RoI) Pooling



- Given an anchor box, uniformly cuts it into *n x m* blocks, output the maximal value in each block
- Returns *nm* values for each anchor box

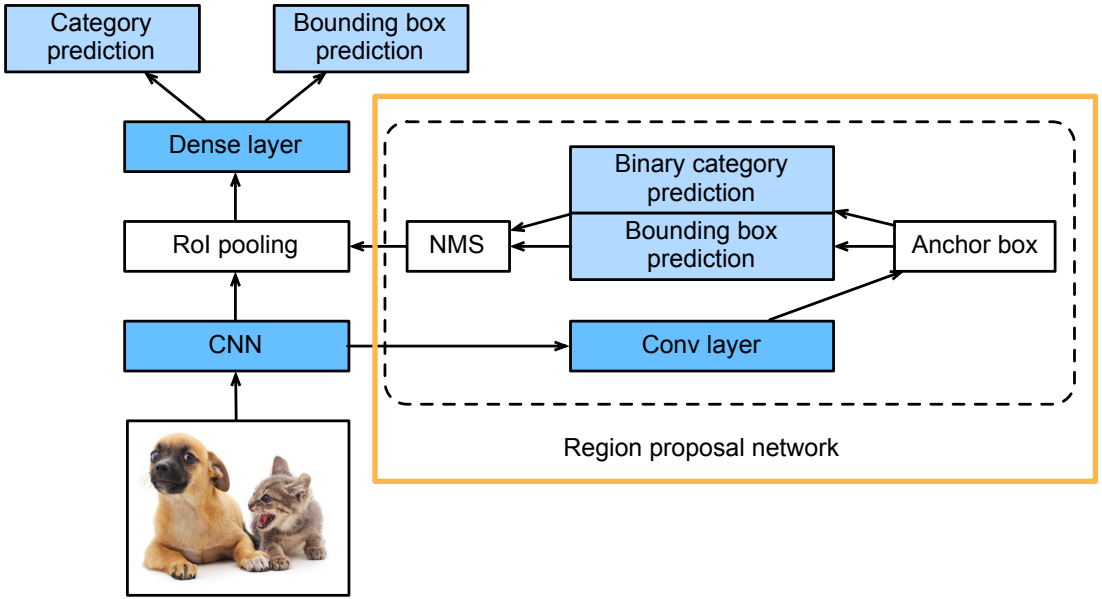
Fast RCNN

- A CNN to extract features (fast)
- RoI pooling returns fixed length feature for each anchor box



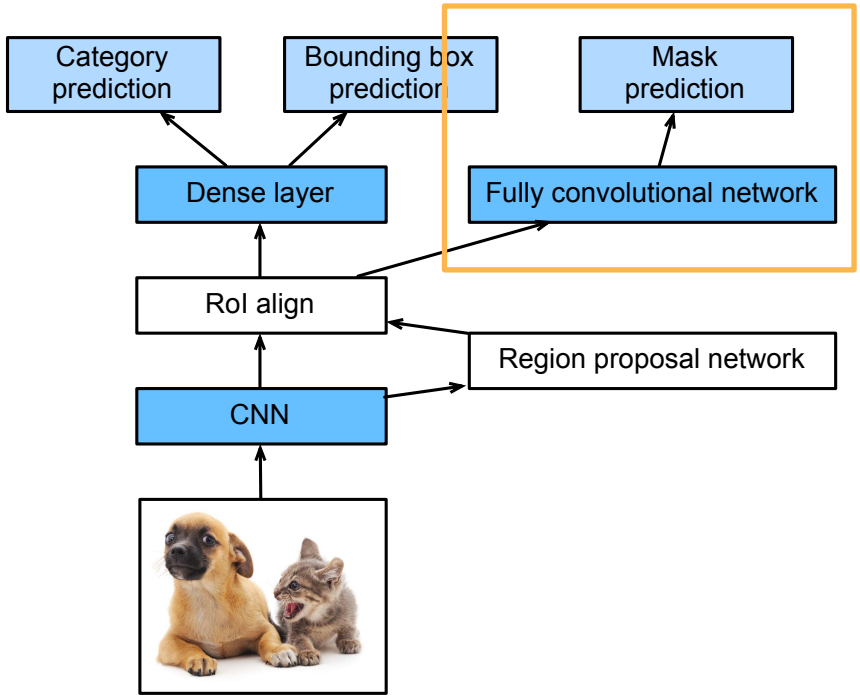
Faster R-CNN

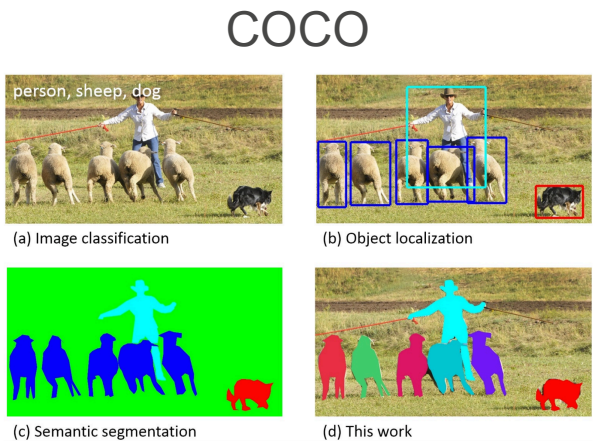
- Use a region proposal network to replace select search for high quality anchor boxes

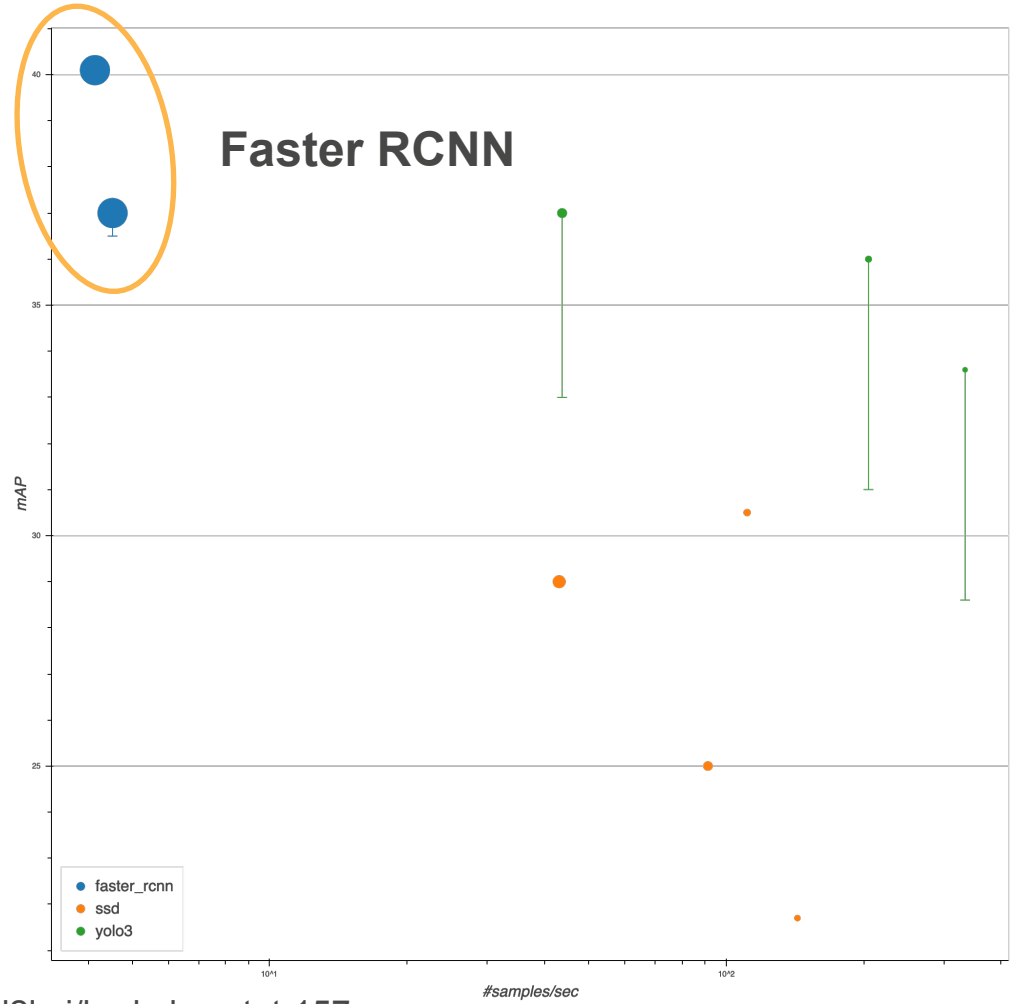


Mask R-CNN

- If pixel-level labels are available, add an additional loss (FCN) to take into account them





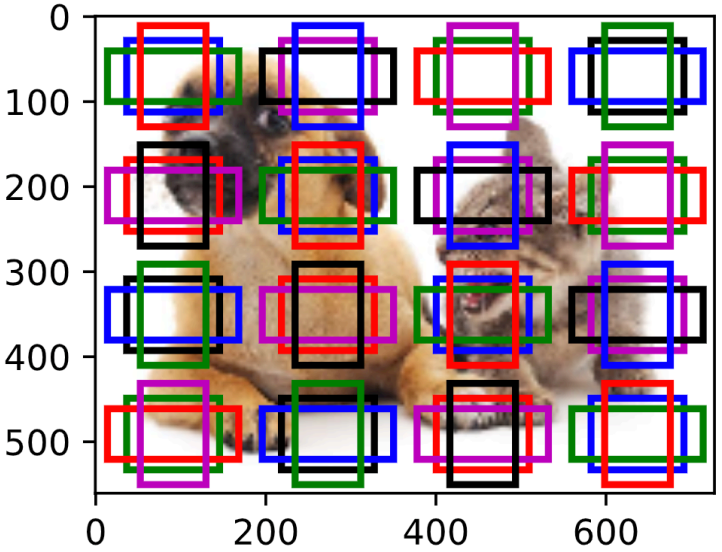


https://gluon-cv.mxnet.io/model_zoo/detection.html

Single Shot Multibox Detection (SSD)

Generate Anchor Boxes

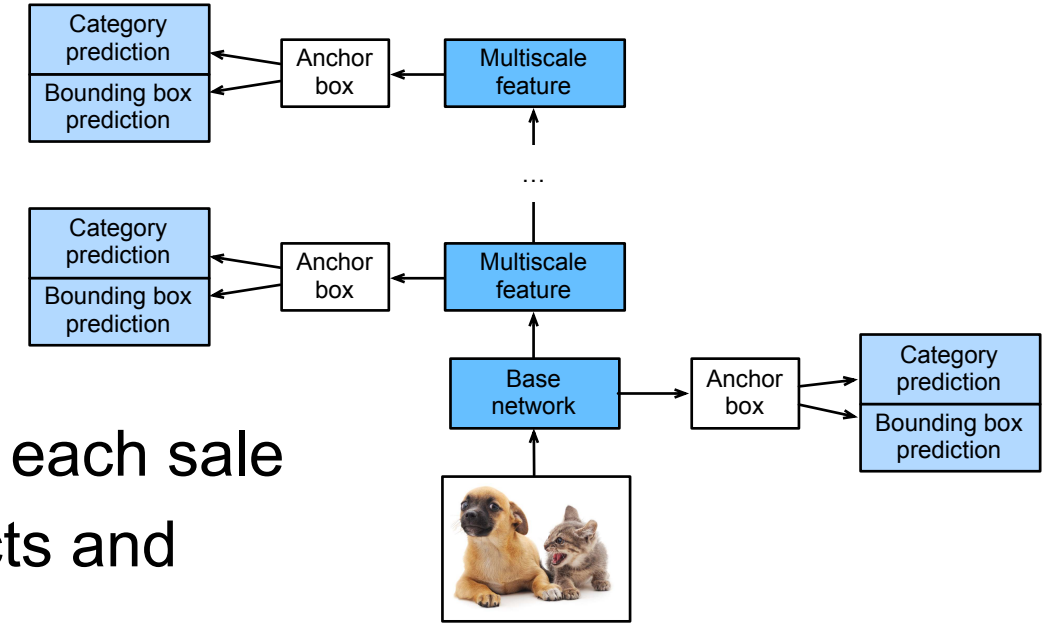
- For each pixel, generate multiple anchor boxes centered at this pixel
- Given *n* sizes $s_1, \ldots, s_n$ and *m* ratios $r_1, \ldots, r_m$, generate *n+m-1* anchor boxes

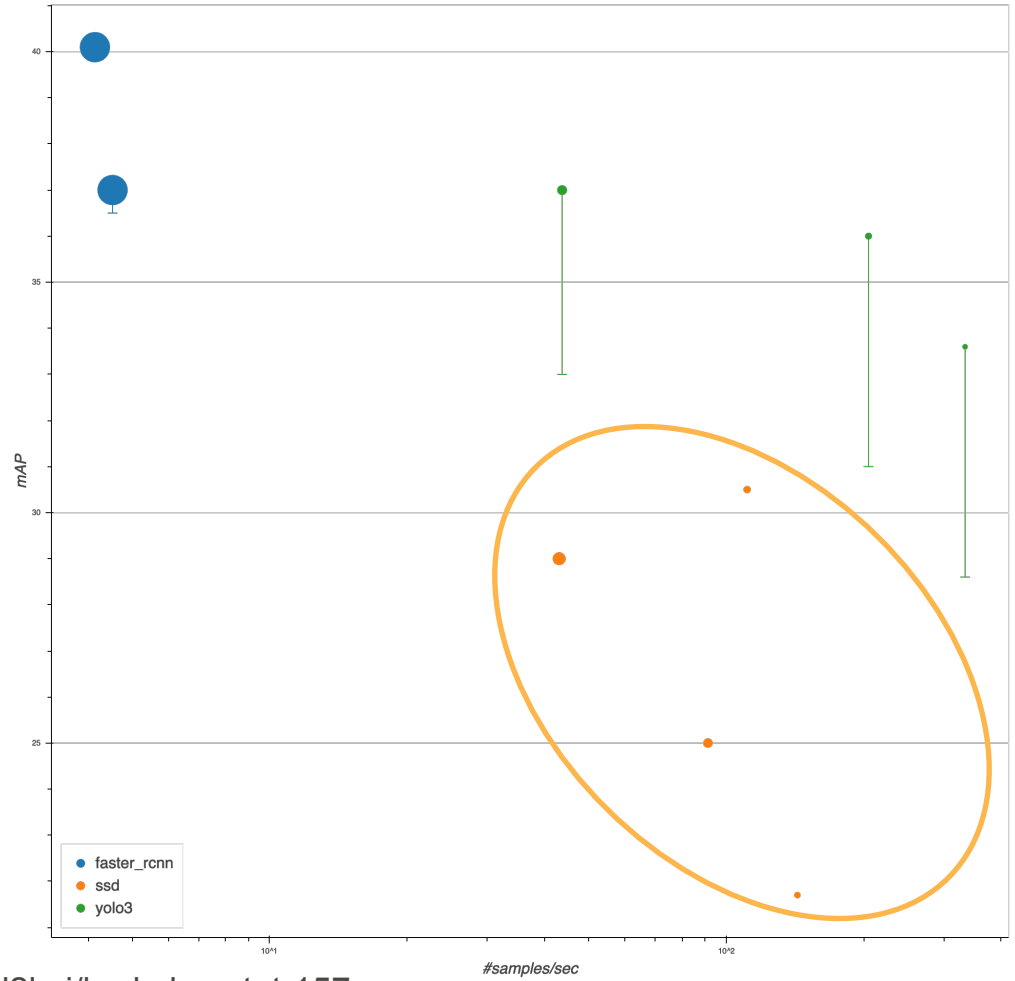


$$(s_1, r_1), (s_2, r_1), \ldots, (s_n, r_1), (s_1, r_2), \ldots, (s_1, r_m)$$

SSD Model

- A base network to extract feature, followed by conv-blocks to halve width and height
- Generate anchor boxes at each sale
  - Bottom for small objects and top for large objects
- Predict class and bounding box for each anchor box





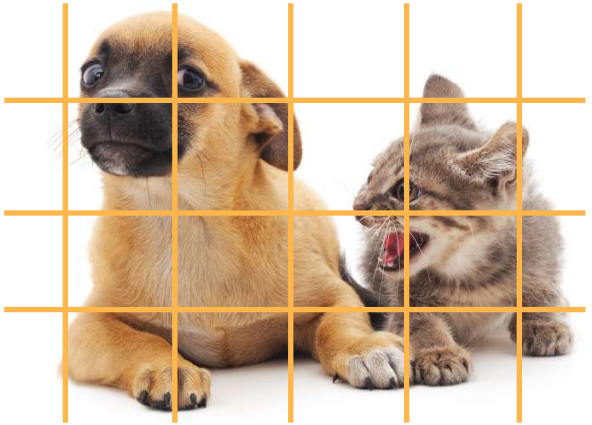
https://gluon-cv.mxnet.io/model_zoo/detection.html

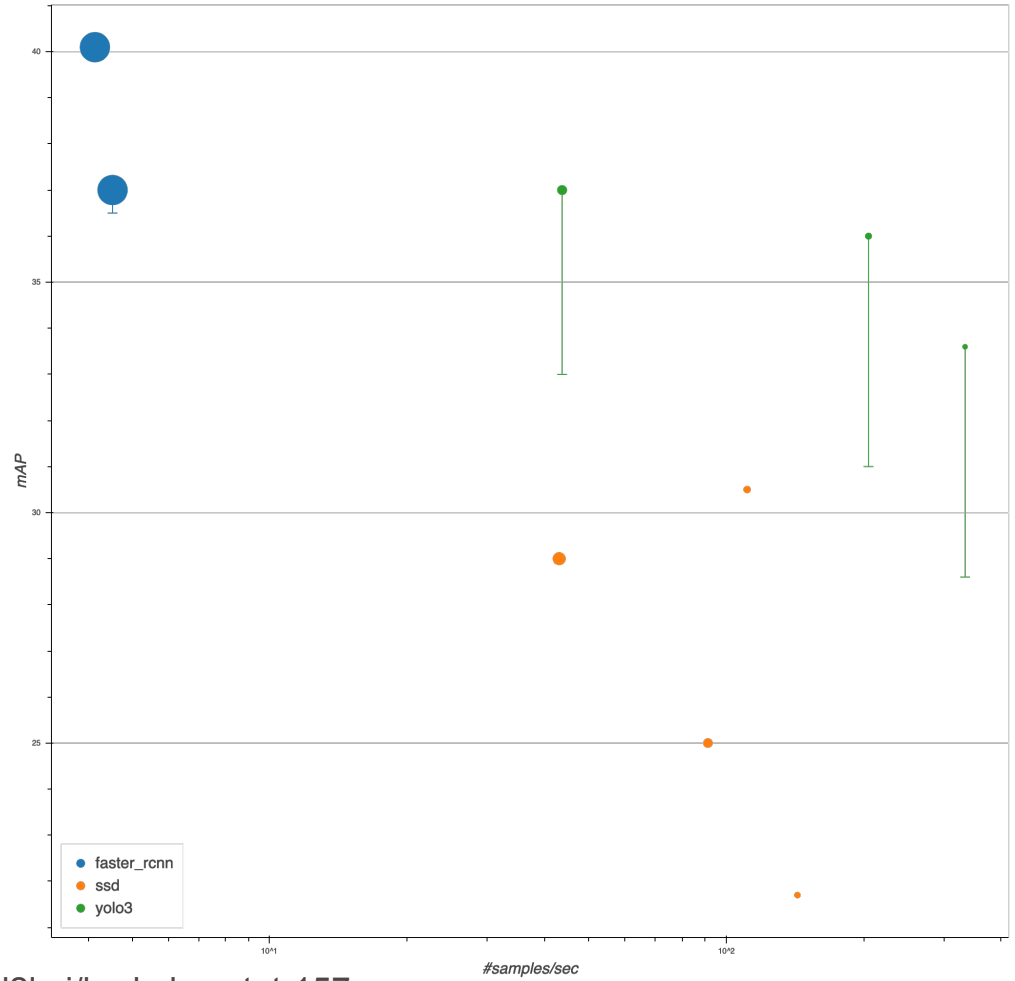
SSD

You Only Look Once (YOLO)

YOLO

- Anchor boxes are highly overlapped in SSD
- YOLO cuts the input image uniformly into *S x S* anchor boxes
- Each anchor box predicts *B* bounding boxes
- V2 and V3 add more improvements



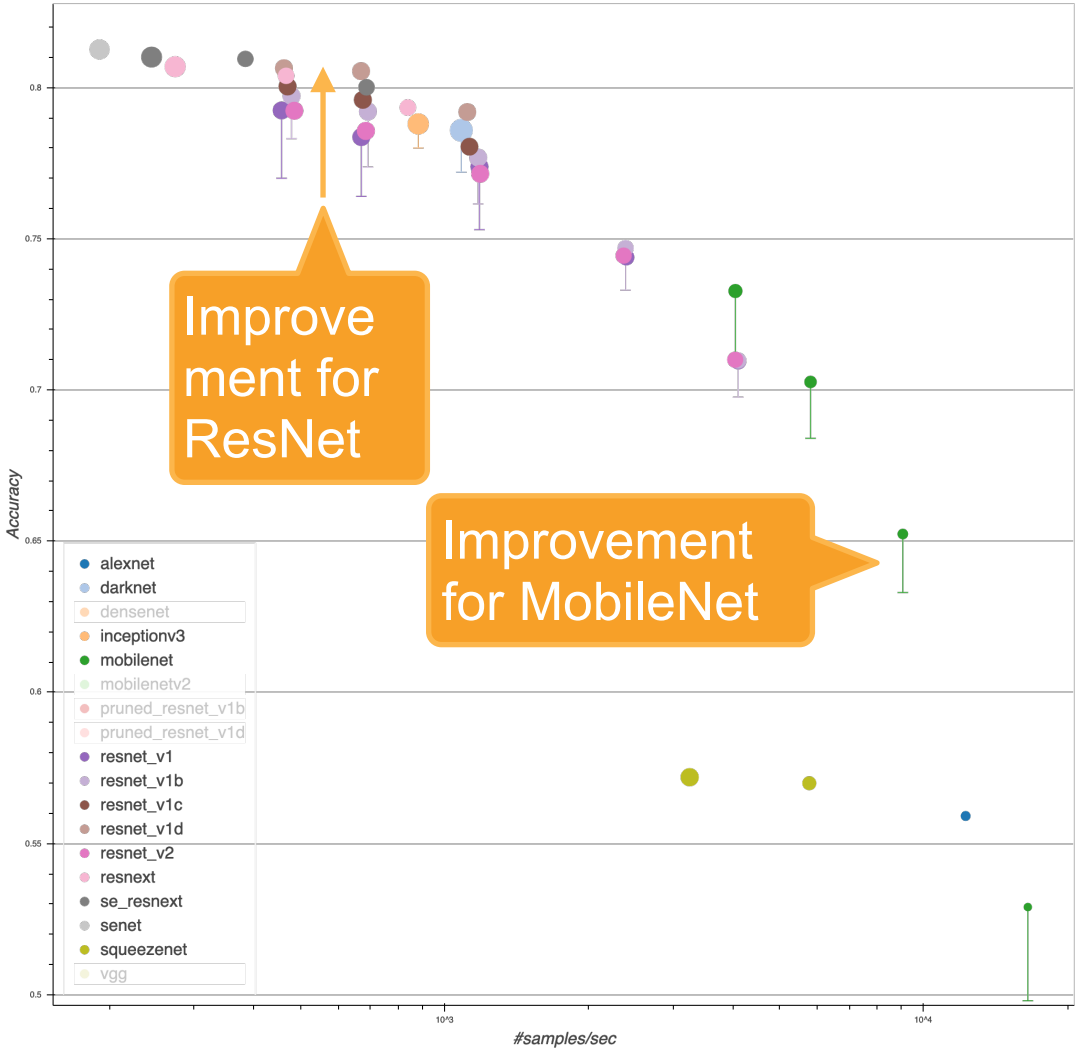


https://gluon-cv.mxnet.io/model_zoo/detection.html

Yolo V3

Tricks for Training

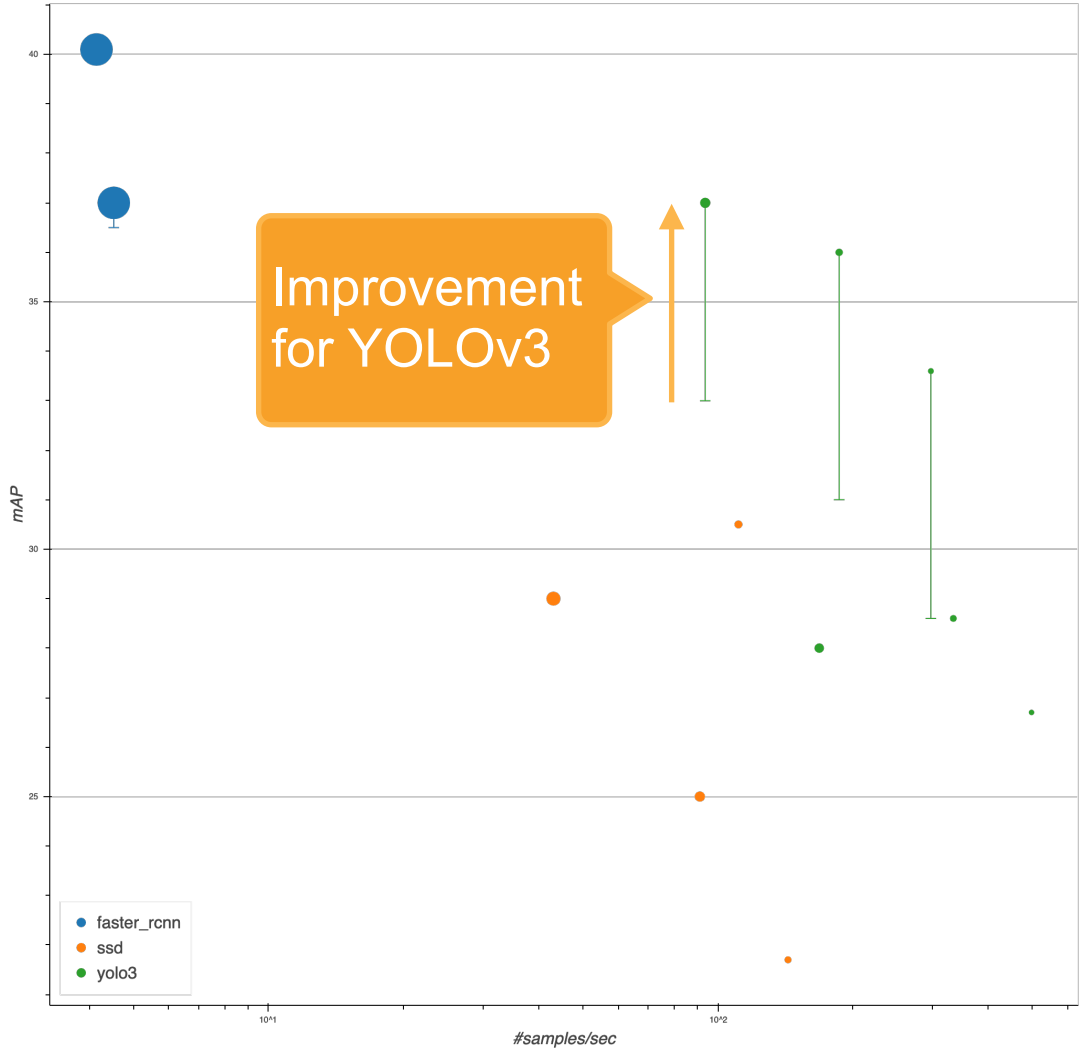
Various training tricks greatly improve image classification model accuracy



GluonCV model zoo
https://gluon-cv.mxnet.io/model_zoo/classification.html

aws

Apply to object detection models as well



GluonCV model zoo https://gluon-cv.mxnet.io/model_zoo/detection.html

Mixup Training Example

- Randomly select two examples *i* and *j,* sample a random number $\lambda \in [0,1]$
- Compute the mixed new example

$$x = \lambda x_i + (1-\lambda)x_j \quad y = \lambda y_i + (1-\lambda)y_j$$

- Train on mixed examples

bittern	0
...	0
otter	0
...	0
analog_clock	1

* 0.9 +

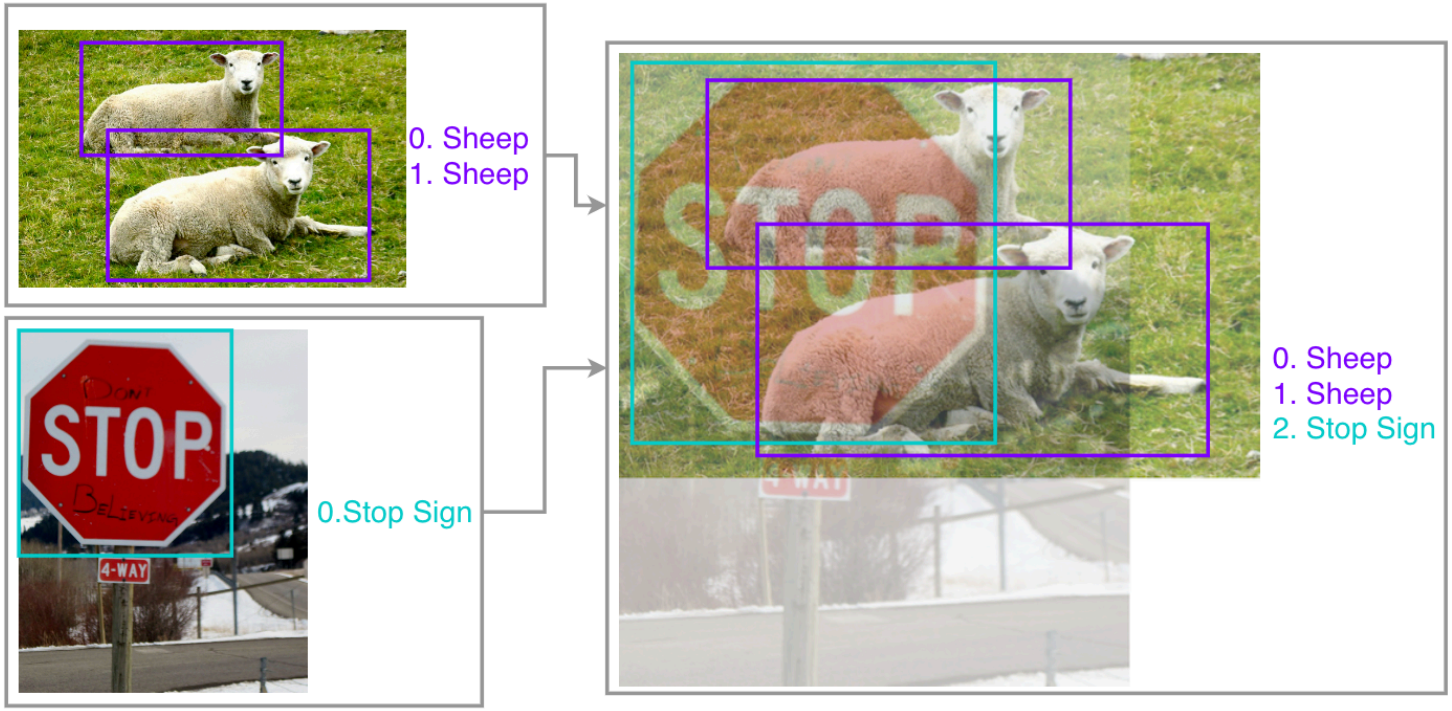
bittern	1
...	0
otter	0
...	0
analog_clock	0

* 0.1 =

bittern	0.1
...	0
otter	0
...	0
analog_clock	0.9

Mixup Training Example

- Apply to object detection as well



Label Smoothing

- Assume $\mathbf{y} \in \mathbb{R}^n$ is the one-hot encoding of label

$$y_i = \begin{cases} 1 & \text{if belongs to class } i \\ 0 & \text{otherwise} \end{cases}$$

- Approximating 0/1 values with softmax is hard
- The smoothed version

$$y_i = \begin{cases} 1 - \epsilon & \text{if belongs to class } i \\ \epsilon/(n-1) & \text{otherwise} \end{cases}$$

  - Commonly use $\epsilon = 0.1$

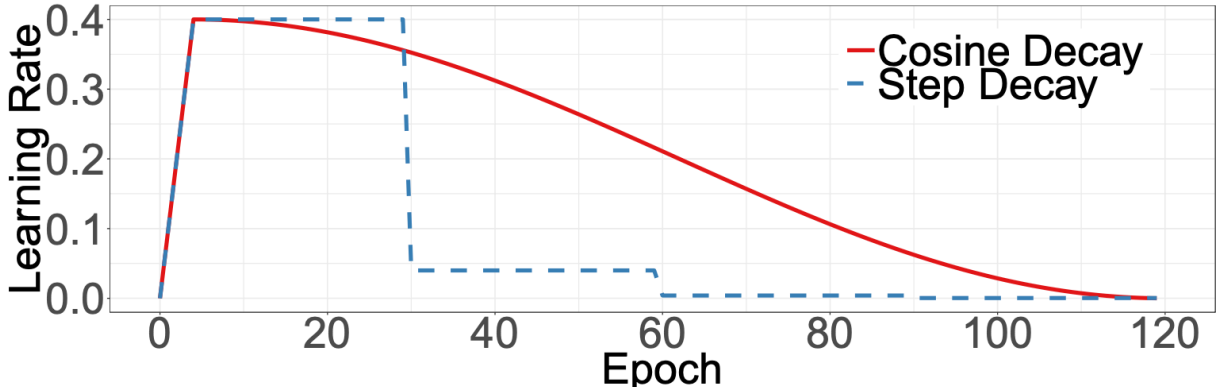
Learning Rate Warmup

- A large learning rate for randomly initialized parameters may cause numerical issue
- The warmup trick uses a small learning rate at beginning and then increases it to the initial value. For example:
  - If we choose the initial learning rate to be 0.1 and use 5 epochs for warmup
  - Start the learning rate with 0, linearly increases it to 0.1 in the first 5 epochs

Cosine Decay

- We need to decrease learning rate for SGD to converge
  - E.g. decreasing by 10x at epoch 30, 60, and 90
- Assume in total *T* iterations (batches), the cosine decay computes learning rate at iteration *t* by

$$\eta_t = 1/2 \left(1 + \cos\left(t\pi/T\right)\right) \eta$$



Synchronized Batch Normalization

- BatchNorm needs a large batch size for reliable statistics
- Object detection tasks may allow a small batch size due to GPU memory constraints, e.g. 1 image per GPU
- In multi-GPU training, each GPU computes mean/ variance separately
- Synchronized BatchNorm computes statistics over all GPUs

Random Batch Shapes

- Images are resized to same shape in a batch, e.g. 224 width and 224 height
- We can vary this shape:
  - For each batch, choose a random width/height from 224 (7x32), 256 (8x32), 228 (9x32), …
  - Resize all images into this shape

Image Classification

Refinements	ResNet-50-D		Inception-V3		MobileNet	
	Top-1	Δ	Top-1	Δ	Top-1	Δ
Efficient	77.16		77.50		71.90	
+ cosine decay	77.91	+0.75	78.19	+0.69	72.83	+0.93
+ label smoothing	78.31	+0.4	78.40	+0.21	72.93	+0.1
+ mixup	**79 . 15**	+0.84	**78.77**	+0.37	**73.28**	+0.35

Hang et.al *Bag of Tricks for Image Classification with Convolutional Neural Networks*

Results for YOLOv3

Incremental Tricks	mAP	Δ	Cumu Δ
- data augmentation	64.26	-15.99	-15.99
baseline	80.25	0	0
+ synchronize BN	80.81	+0.56	+0.56
+ random training shapes	81.23	+0.42	+0.98
+ cosine lr schedule	81.69	+0.46	+1.44
+ class label smoothing	82.14	+0.45	+1.89
+ mixup	**83.68**	**+1.54**	**+3.43**

Zhi et al, *Bag of Freebies for Training Object Detection Neural Networks*