YOLO9000: Better, Faster, Stronger

Date: January 24, 2018

Prepared by Haris Khan (University of Toronto)

Overview

1. Motivation for one-shot object detection and weakly-supervised learning
2. YOLO
3. YOLOv2 / YOLO9000
4. Future Work

One-Shot Detection

- Eliminates regional proposal steps used in R-CNN [3], Fast R-CNN [4] and Faster R-CNN [5]

Motivation:

- Develop object detection methods that predict bounding boxes and class probabilities at the same time
  - Want to achieve real-time detection speeds
  - Maintain / exceed accuracy benchmarks set by previous region proposal methods

Improving Detection Datasets

VOC 2007 / 2012:

- 20 classes
  - i.e. person, cat, dog, car, chair, bottle

MS COCO:

- 80 classes
  - i.e. book, apple, teddy bear, scissors

ImageNet1000:

- 1000 classes
  - i.e. German shepherd, golden retriever, European fire salamander

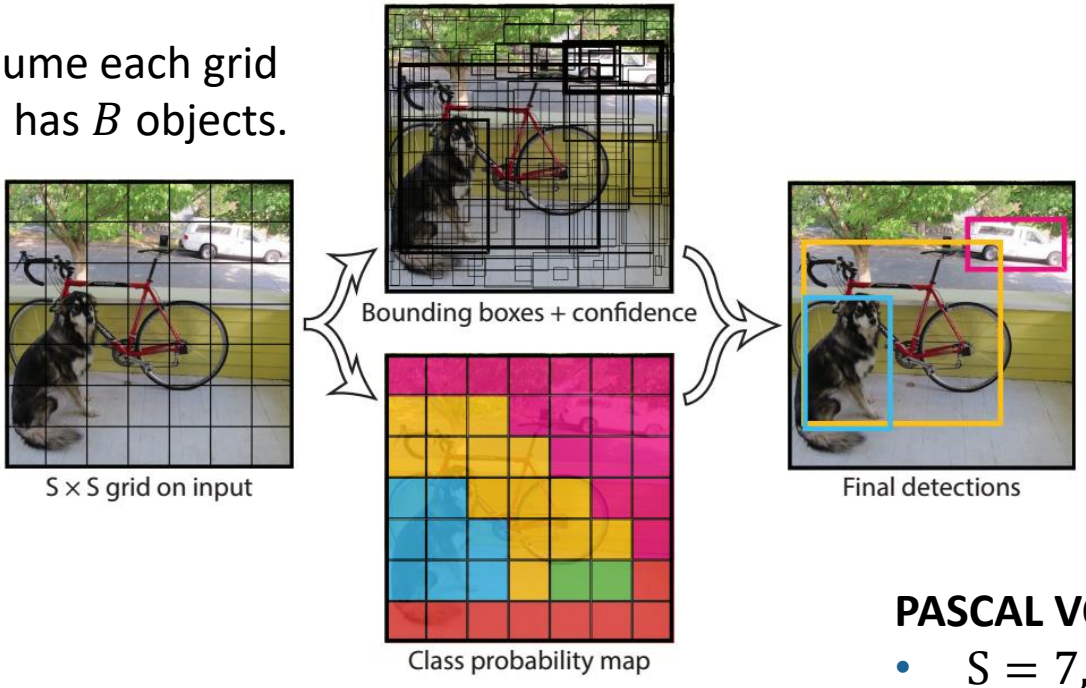
Motivation:

- Increase the number and detail of classes that can be learned during training using existing detection and classification datasets

You Only Look Once (YOLO) [1]

2. Bounding box feature vector = $[x, y, w, h, c]$

1. Assume each grid cell has B objects.



3. Merge predictions into $S \times S \times (5B + C)$ output tensor

2. C object classes

PASCAL VOC:

- $S = 7,\ B = 2,\ C = 20$
- Output tensor size = $7 \times 7 \times 30$

Image Credit: [1]

YOLO - Architecture

- Inspired by GoogLeNet
- 24 convolutional layers + 2 FC layers

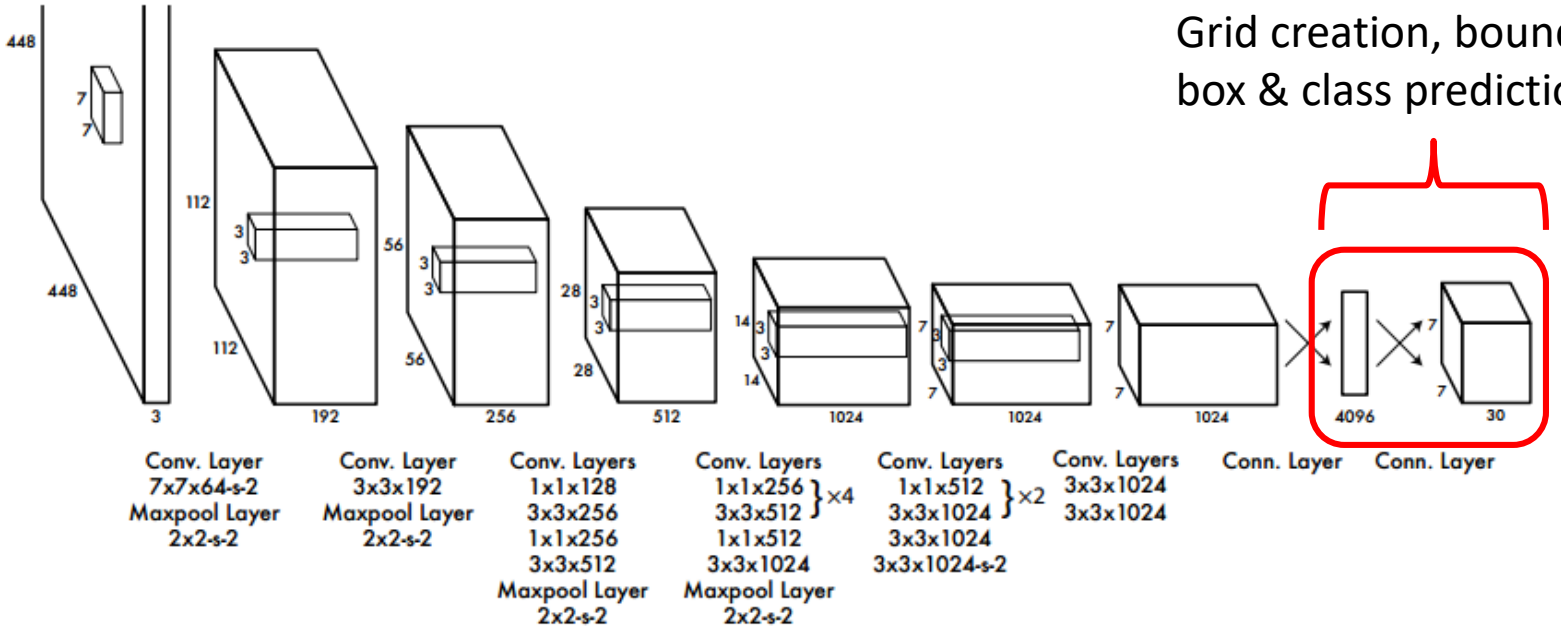


Image Credit: [1]

YOLO - Training Loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

Image Credit: [1]

- Only back-propagate loss if object is present

YOLO - Test Results

- Primary evaluation done on VOC 2007 & 2012 test sets

VOC 2007 Test Results

	Real-Time Detectors	Train	mAP	FPS
	100Hz DPM [31]	2007	16.0	100
	30Hz DPM [31]	2007	26.1	30
	Fast YOLO	2007+2012	52.7	**155**
*	YOLO	2007+2012	**63.4**	45
	Less Than Real-Time			
	Fastest DPM [38]	2007	30.4	15
	R-CNN Minus R [20]	2007	53.5	6
	Fast R-CNN [14]	2007+2012	70.0	0.5
	Faster R-CNN VGG-16[28]	2007+2012	73.2	7
	Faster R-CNN ZF [28]	2007+2012	62.1	18
	YOLO VGG-16	2007+2012	66.4	21

VOC 2012 Test Results

VOC 2012 test	mAP
MR_CNN_MORE_DATA [11]	**73.9**
HyperNet_VGG	71.4
HyperNet_SP	71.3
Fast R-CNN + YOLO	70.7
MR_CNN_S_CNN [11]	70.7
Faster R-CNN [28]	70.4
DEEP_ENS_COCO	70.1
NoC [29]	68.8
Fast R-CNN [14]	68.4
UMICH_FGS_STRUCT	66.4
NUS_NIN_C2000 [7]	63.8
BabyLearning [7]	63.2
NUS_NIN	62.4
R-CNN VGG BB [13]	62.4
R-CNN VGG [13]	59.2
YOLO	57.9
Feature Edit [33]	56.3
R-CNN BB [13]	53.3
SDS [16]	50.7
R-CNN [13]	49.6

Table Credits: [1]

*Speed measured on Titan X GPU

YOLO - Limitations

- Produces more localization errors than Fast R-CNN
  - Struggles to detect small, repeated objects (i.e. flocks of birds)
  - Bounding box priors not used during training

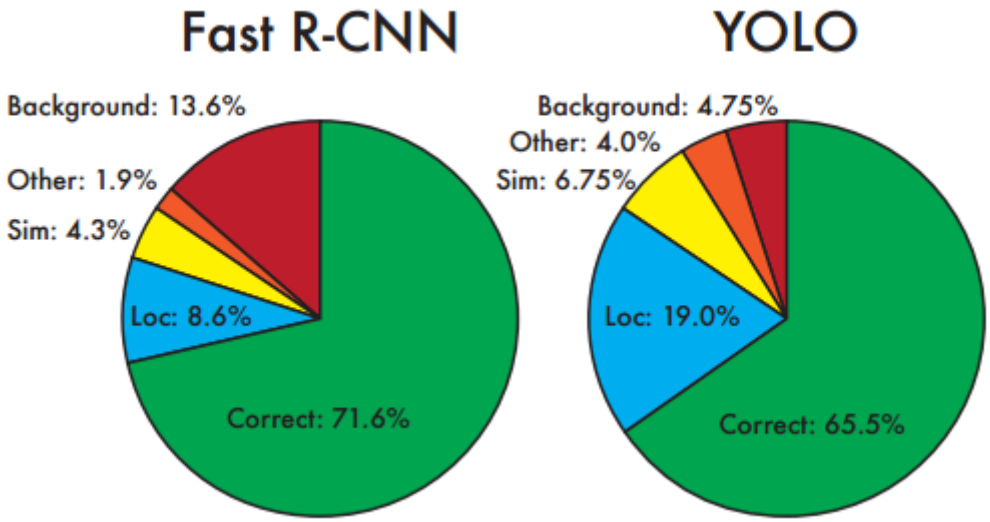


Image Credit: [1]

YOLO9000 - Paper Overview

YOLOv2 [2]:

- Modified version of original YOLO that increases detection speed and accuracy

YOLO9000 [2]:

- Training method that increases the number of classes a detection network can learn by using weakly-supervised training on the union of detection (i.e. VOC, COCO) and classification (i.e. ImageNet) datasets

YOLOv2 - Modifications

	Modification	Effect
Bounding Boxes	Anchor Boxes	7% recall increase
	Dimension clusters + new bounding box parameterization	4.8% mAP increase
Architecture	New Darknet-19 replaces GoogLeNet	33% computation decrease, 0.4% mAP increase
	Convolutional prediction layer	0.3% mAP increase
Training	Batch normalization	2% mAP increase
	High resolution fine-tuning of weights	4% mAP increase
	Multi-scale images	1.1% mAP increase
	Passthrough for fine-grained features	1% mAP increase

YOLOv2 - Bounding Boxes

- Anchor boxes allow multiple objects of various aspect ratio to be detected in a single grid cell
- Anchor boxes sizes determined by k-means clustering of VOC 2007 training set
  - **k = 5** provides best trade-off between average IOU / model complexity
  - Average IOU = 61.0%
- Feature vector parameterization directly predicts bounding box centre point, width and height

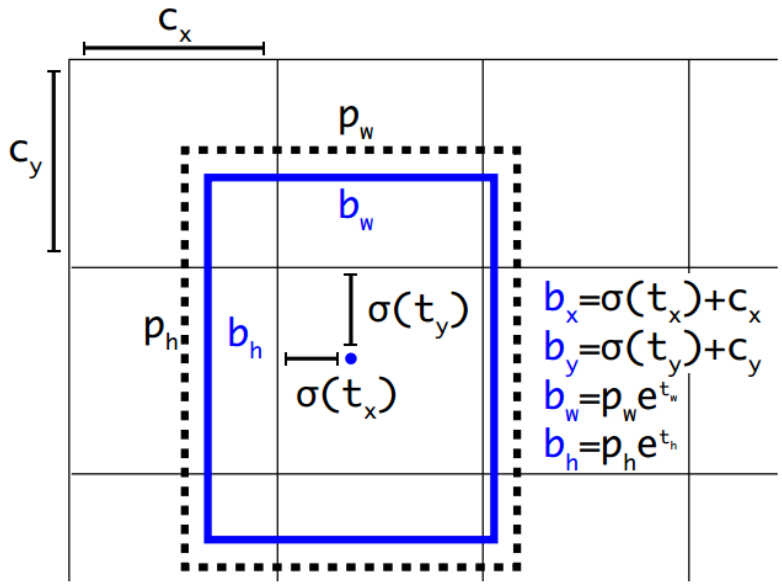


Image Credit: [2]

YOLOv2 - DarkNet-19

- 19 convolutional layers and 5 max-pooling layers
- Reduced number of FLOPs
  - VGG-16 -> 30.67 billion
  - YOLO -> 8.52 billion
  - **YOLOv2 -> 5.58 billion**

DarkNet-19 for Image Classification

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Table Credit: [2]

YOLOv2 - Example

Video link: https://youtu.be/Cgxsv1riJhI?t=290

YOLO9000 - Concept

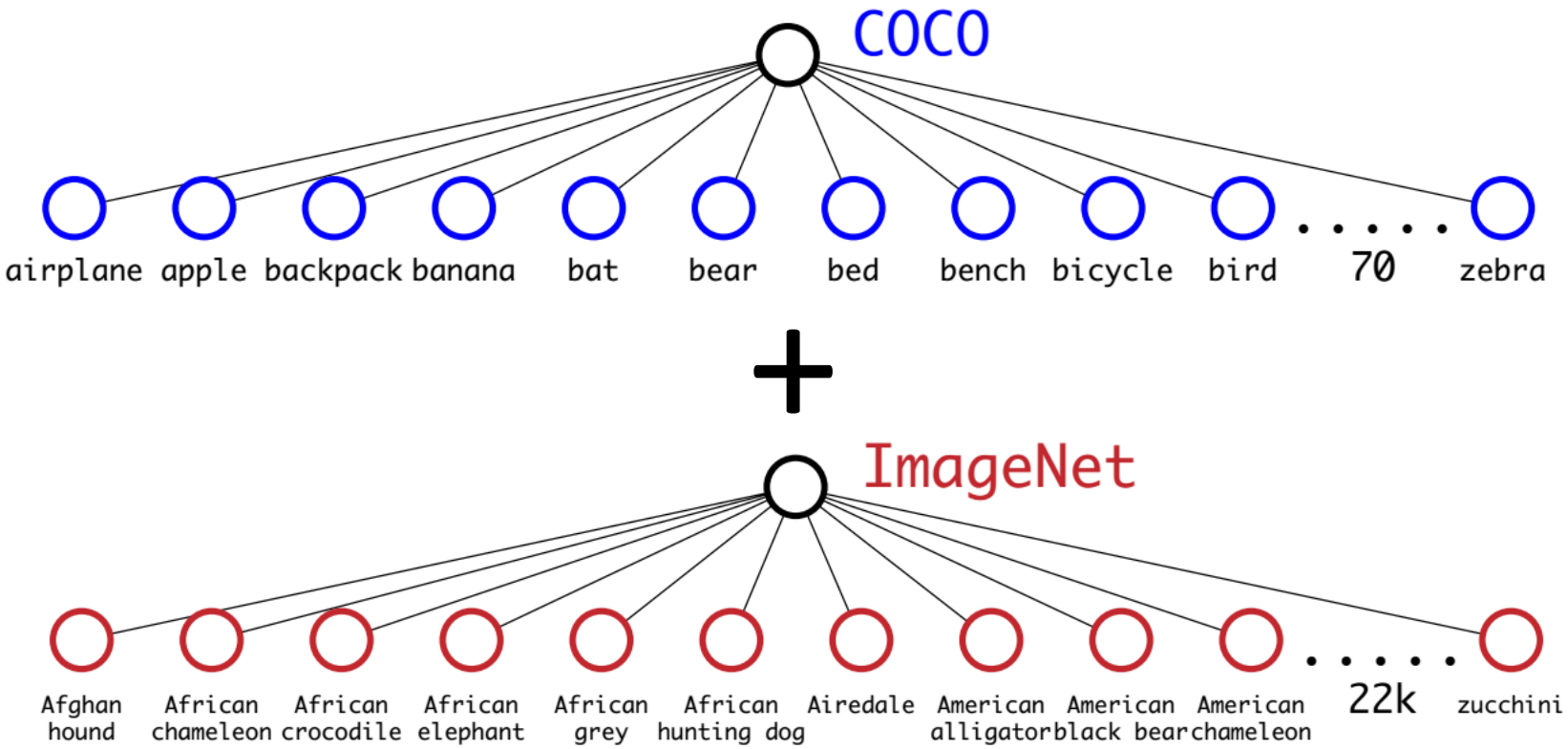
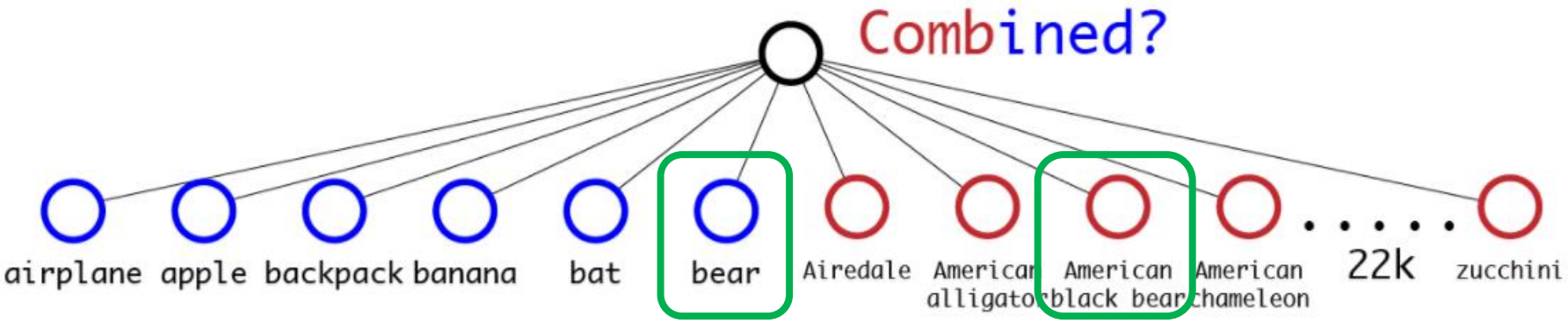


Image Credits: [2]

Can’t just mash classes together...



Slide Credit: Joseph Redmon [3]

YOLO9000 - WordTree

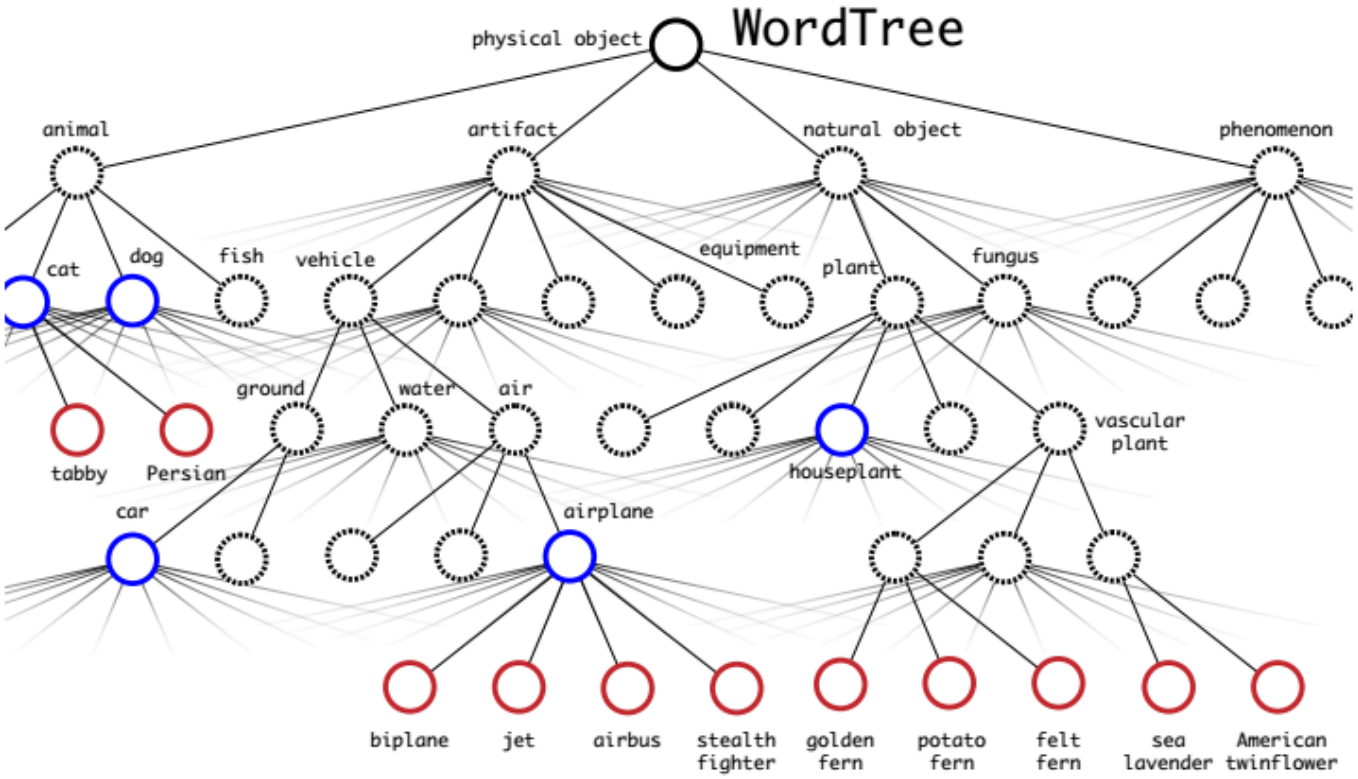
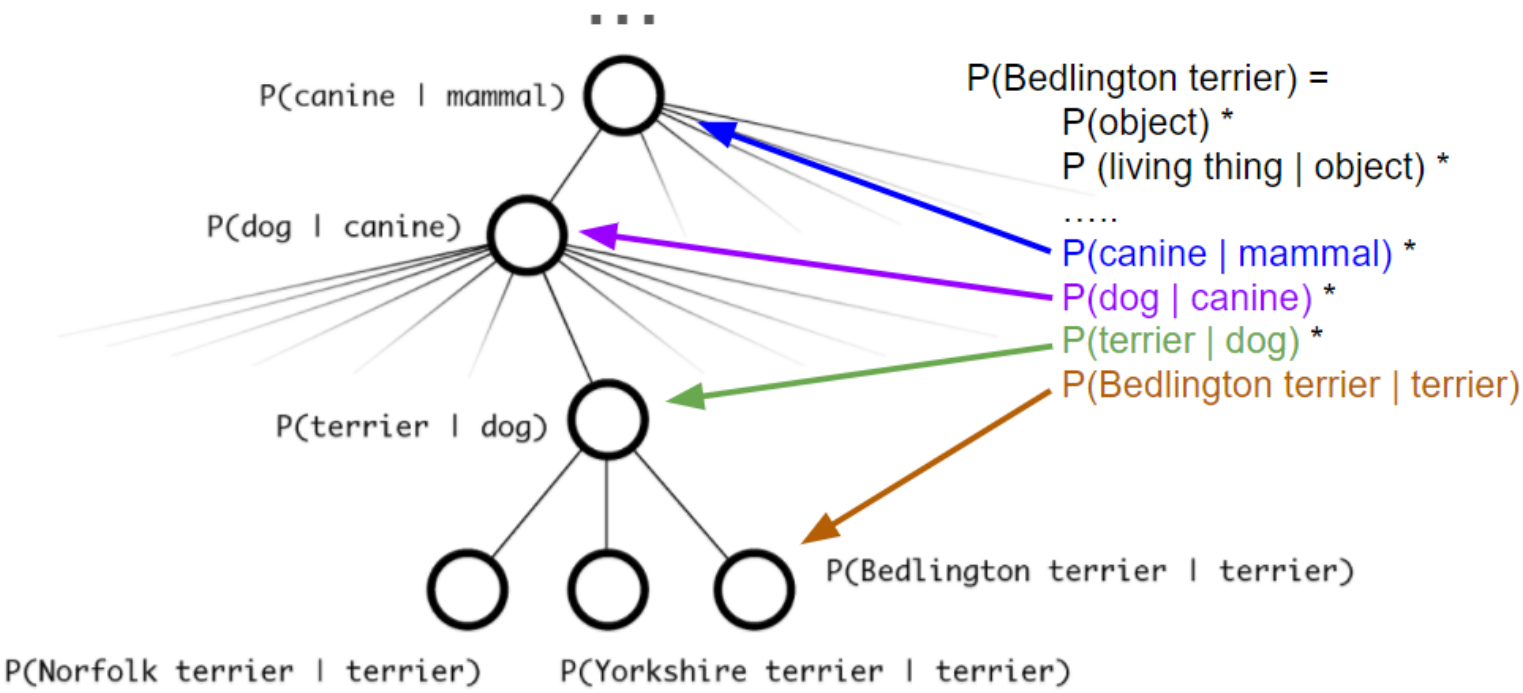


Image Credit: [2]

Each node is a conditional probability



Slide Credit: Joseph Redmon [3]

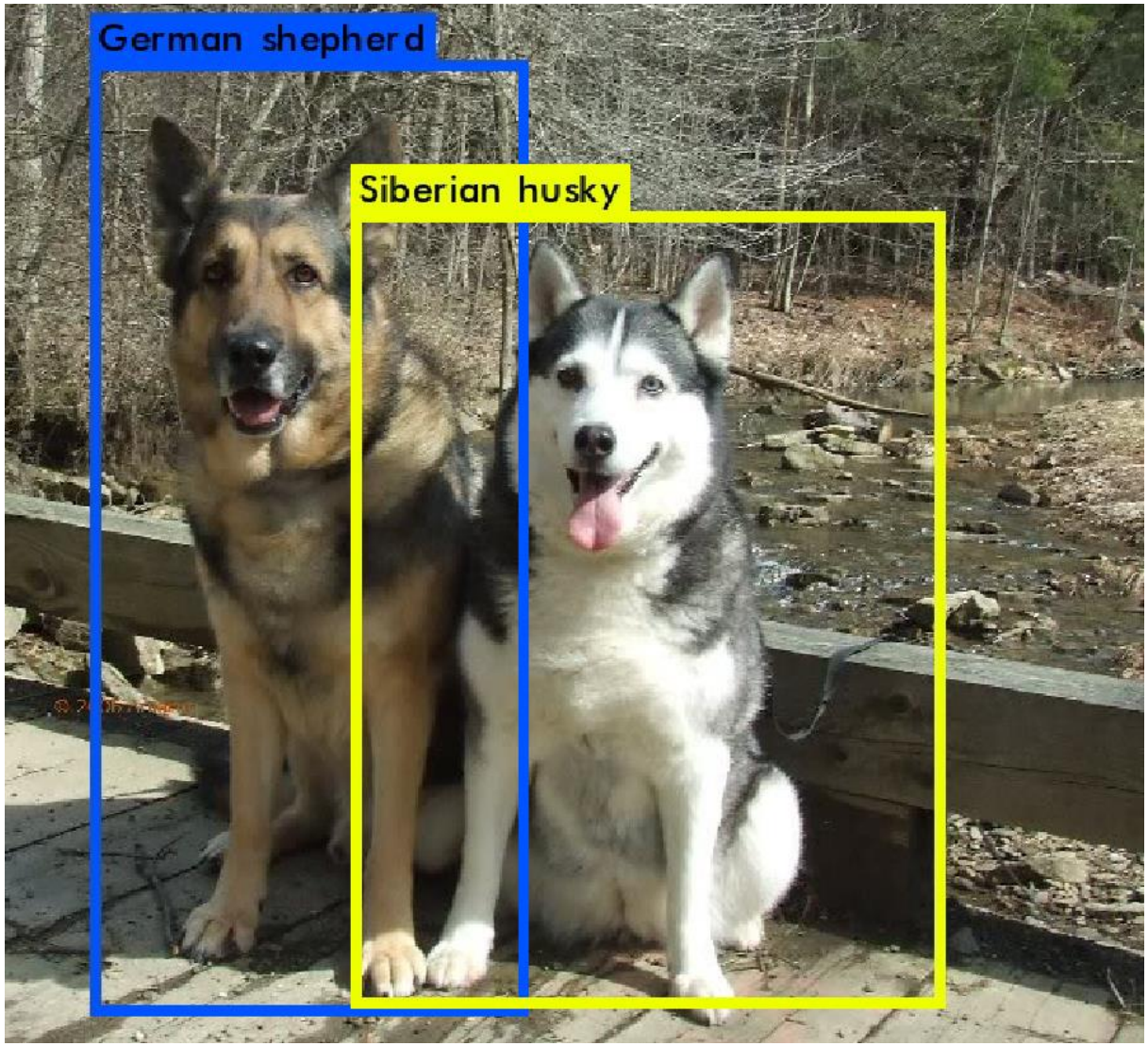


Image Credit: Joseph Redmon [3]

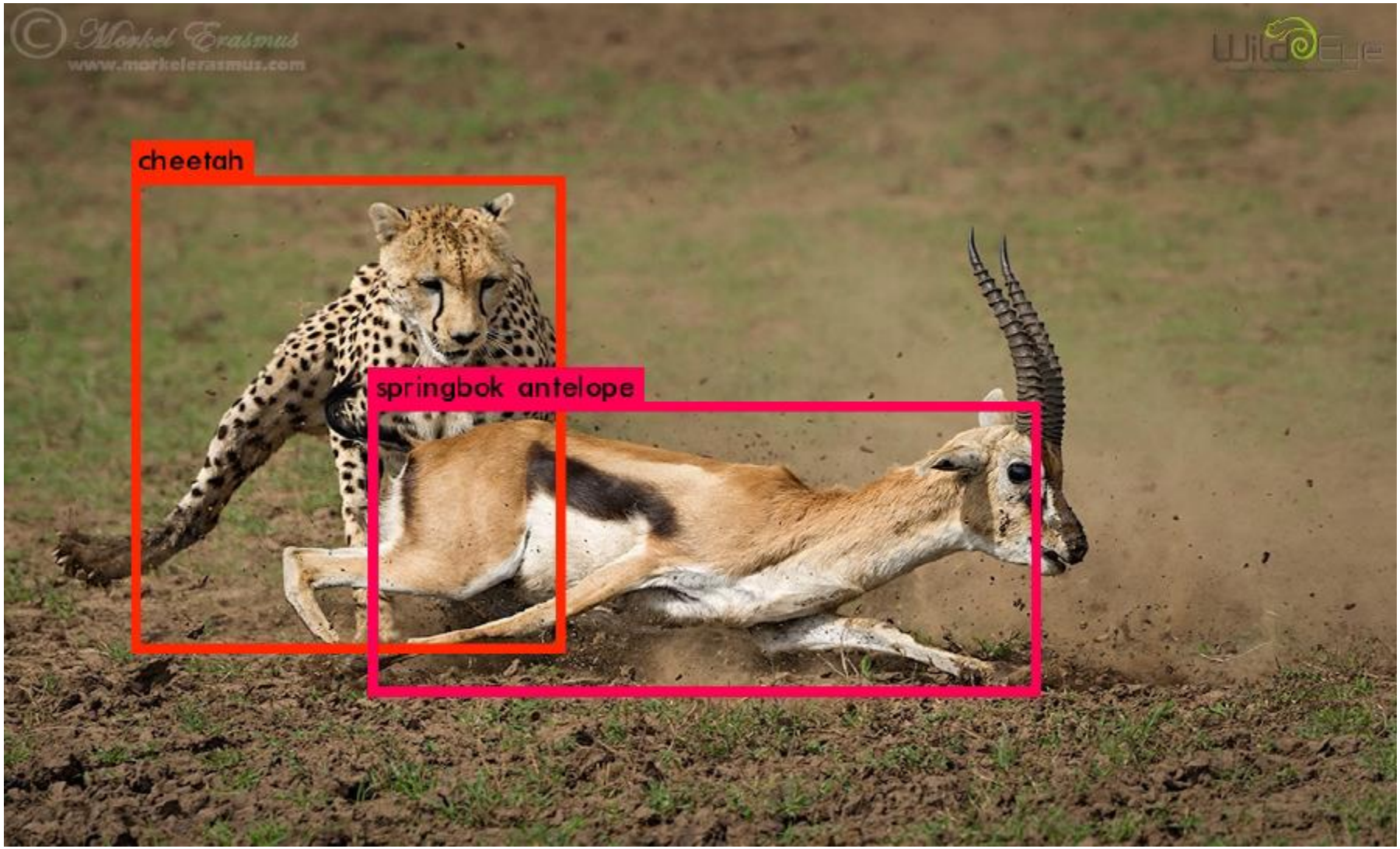


Image Credit: Joseph Redmon [3]

YOLOv2 - Detection Training

Datasets:

- VOC 2007+2012, COCO trainval35k

Data Augmentation:

- Random crops, colour shifting

Hyperparameters:

- # of epochs = 160
- Learning rate = 0.001
- Weight decay = 0.0005
- Momentum = 0.9

Training Enhancements:

- Batch normalization
- High resolution fine-tuning
- Multi-scale images
- Three 3x3 & 1x1 convolutional layers replace last convolutional layer of DarkNet-19 base model
- Passthrough connection between 3x3x512 and second-to-last convolutional layers, adding fine-grained features to prediction layer

YOLO9000 - Detection Training

Datasets:

- 9418 classes
  - ImageNet (top 9000 classes)
  - COCO detection dataset
  - ImageNet detection challenge

Bounding Boxes:

- Minimum IOU threshold = 0.3
- # of dimension clusters =3

Backpropagating Loss:

- For detection images, backpropagate as in YOLOv2
- For unsupervised classification images, only backpropagate classification loss, while finding best matching bounding box from WordTree

YOLOv2 - Test Results

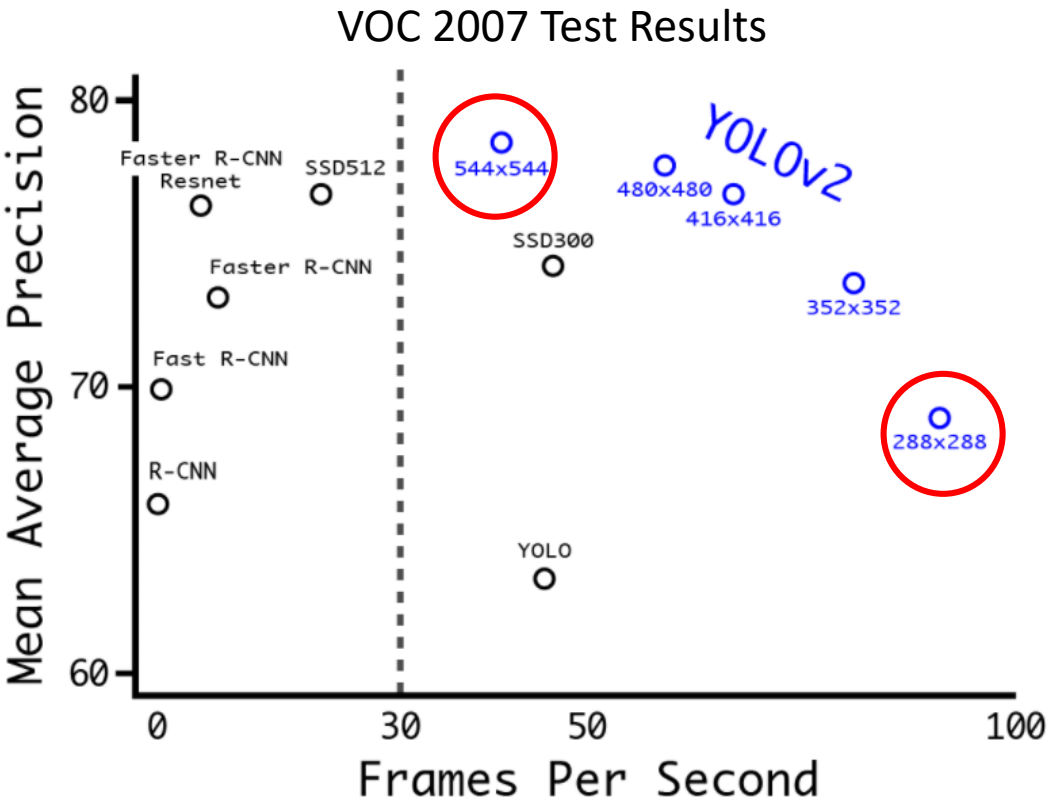


Image Credit: Joseph Redmon [3]

VOC 2012 Test Results

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN [5]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster R-CNN [15]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
YOLO [14]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300 [11]	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD512 [11]	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
ResNet [6]	07++12	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
YOLOv2 544	07++12	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7

COCO Test-Dev 2015 Results

		0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast R-CNN [5]	train	19.7	35.9	-	-	-	-	-	-	-	-	-	-
Fast R-CNN[1]	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster R-CNN[15]	trainval	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster R-CNN[10]	trainval	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300 [11]	trainval35k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512 [11]	trainval35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**
YOLOv2 [11]	trainval35k	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4

Table Credits: [2]

YOLO9000 - Test Results

- Evaluated on ImageNet detection task
  - 200 classes total
    - 44 detection labelled classes shared between ImageNet and COCO
    - 156 unsupervised classes
- Overall detection accuracy = 19.7% mAP
  - 16.0% mAP achieved on unsupervised classes

Best and Worst Classes on ImageNet

diaper	0.0
horizontal bar	0.0
rubber eraser	0.0
sunglasses	0.0
swimming trunks	0.0
...	
red panda	50.7
fox	52.1
koala bear	54.3
tiger	61.0
armadillo	61.7

Table Credit: [2]

YOLO9000 - Paper Evaluation

Strengths:

- Speed performance of YOLOv2 far exceeds competitors (i.e. SSD)
- Anchor box priors via clustering allow detector to learn ideal aspect ratios from training data
- WordTree method increases the number of learnable classes using existing datasets

Weaknesses:

- Detection performance of YOLOv2 on COCO is well below state-of-the-art
- Description of how loss function uses unsupervised training examples is vague
- Results from YOLO9000 tests are inconclusive
  - Does not compare method with alternative weakly-supervised techniques

Future Work

- Improve the accuracy of one-shot detectors in dense object scenes
  - RetinaNet [7]
- Investigate the transferability of weakly-supervised training to other domains, such as image segmentation or dense captioning

Questions?

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[2] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” *arXiv preprint. ArXiv161208242*, 2016.

[3] J. Redmon, “YOLO9000 Better, Faster, Stronger,” presented at the CVPR, 2017.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” arXiv preprint. ArXiv170802002, 2017.