YOLO

YOLO

- You Only Look Once
- Quite similar with Faster R-CNN and very FAST!

Concepts

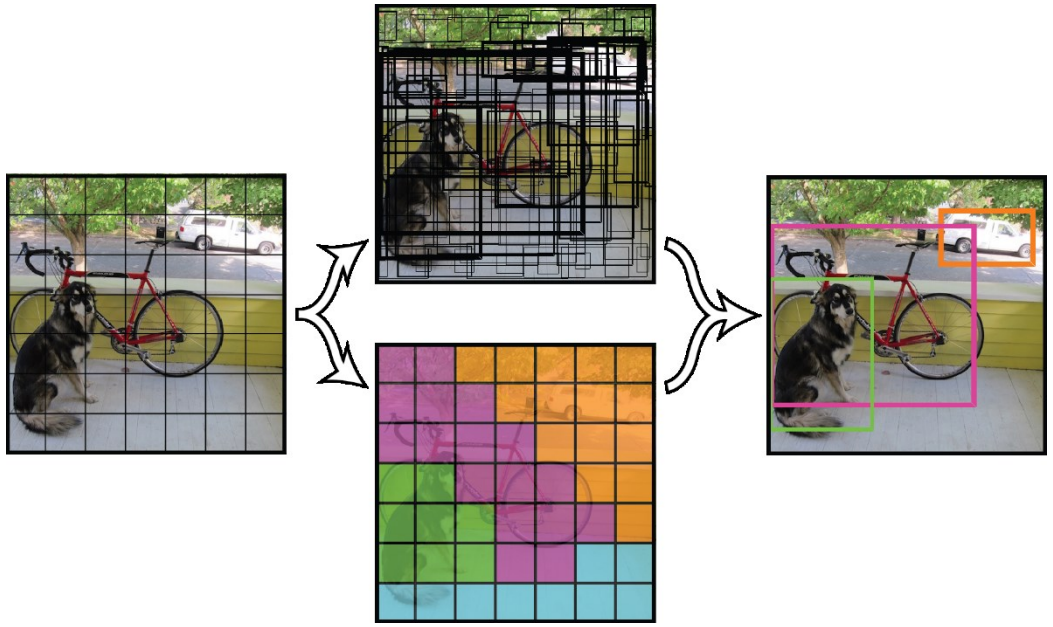
- Detection as Single Regression Problem
- Developed as Single Convolutional Network
- Reason Globally on the Entire Image
- Learns Generalizable Representations

Unified Detection

- Given an image, divide it into an SxS grid

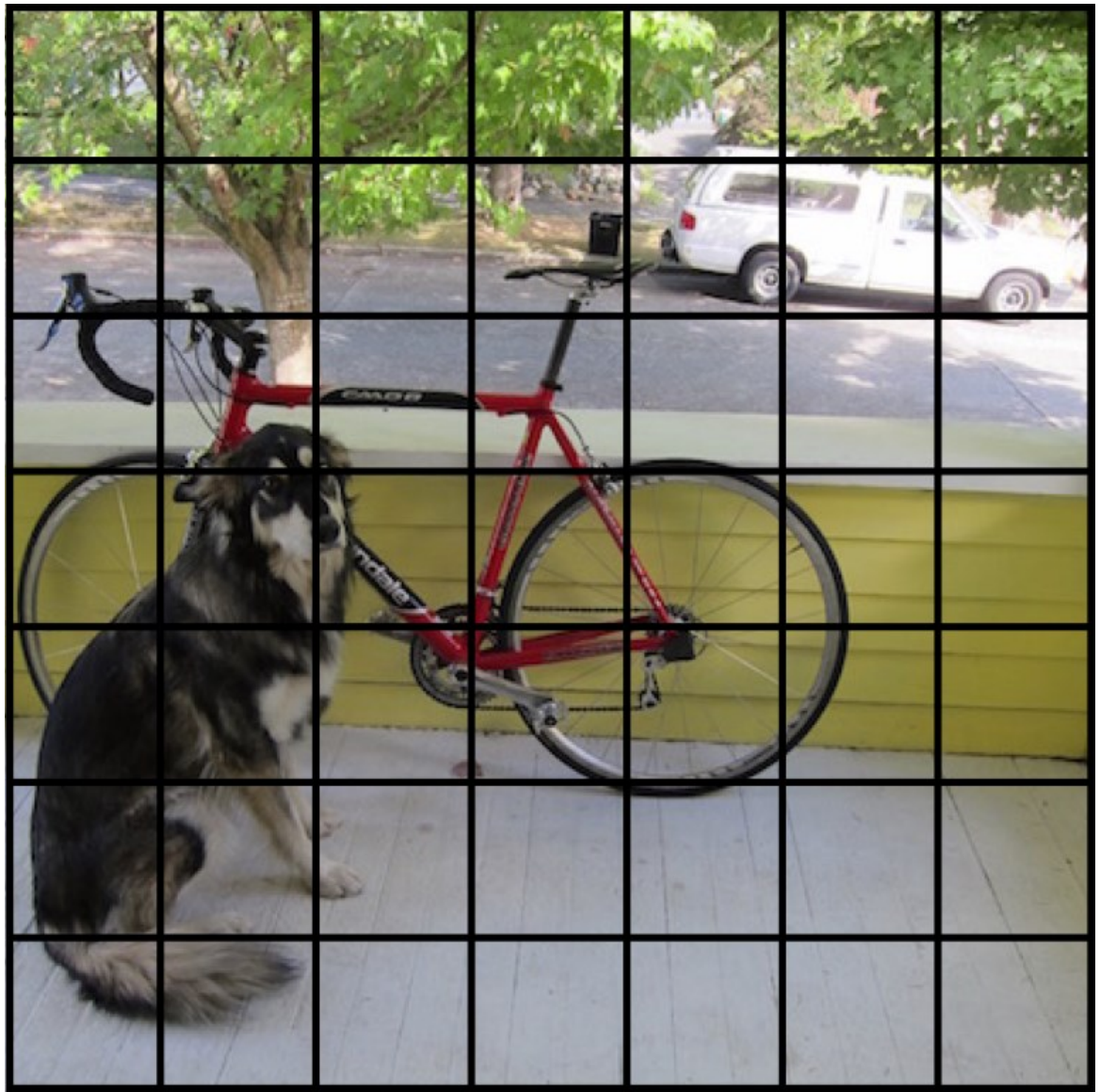
→If the center of an object falls into the grid cell, that grid cell is responsible

- Each cell predicts B bounding boxes
  - Five predictions: x, y, w, h, confidence
- Each cell predicts C class probabilities
  - Cell → C class probabilities, Bx5 bounding box informations
- One cell → One class probability
  - Low false positive!

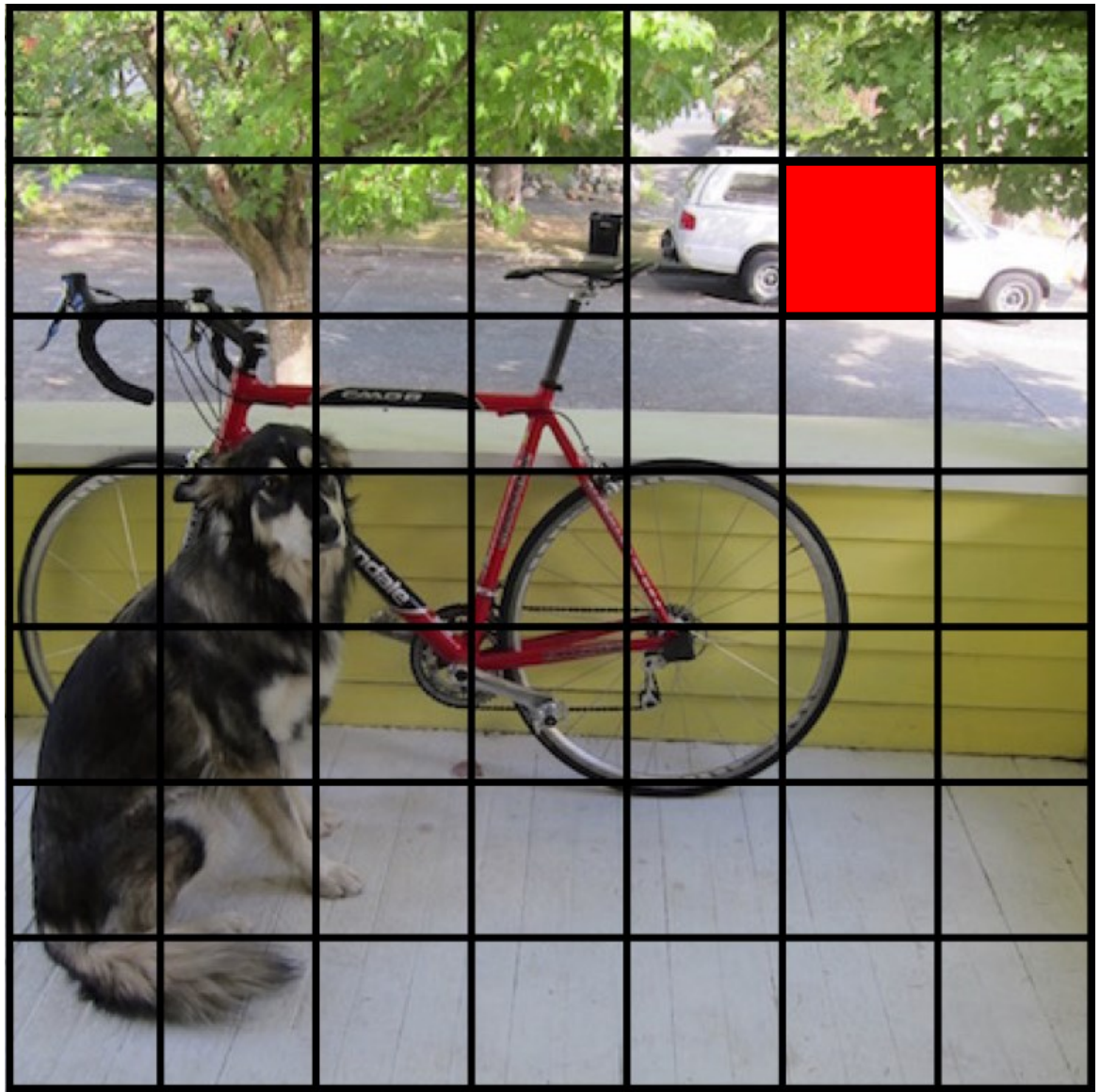


Detection Flow

Split the image into a grid(7x7)

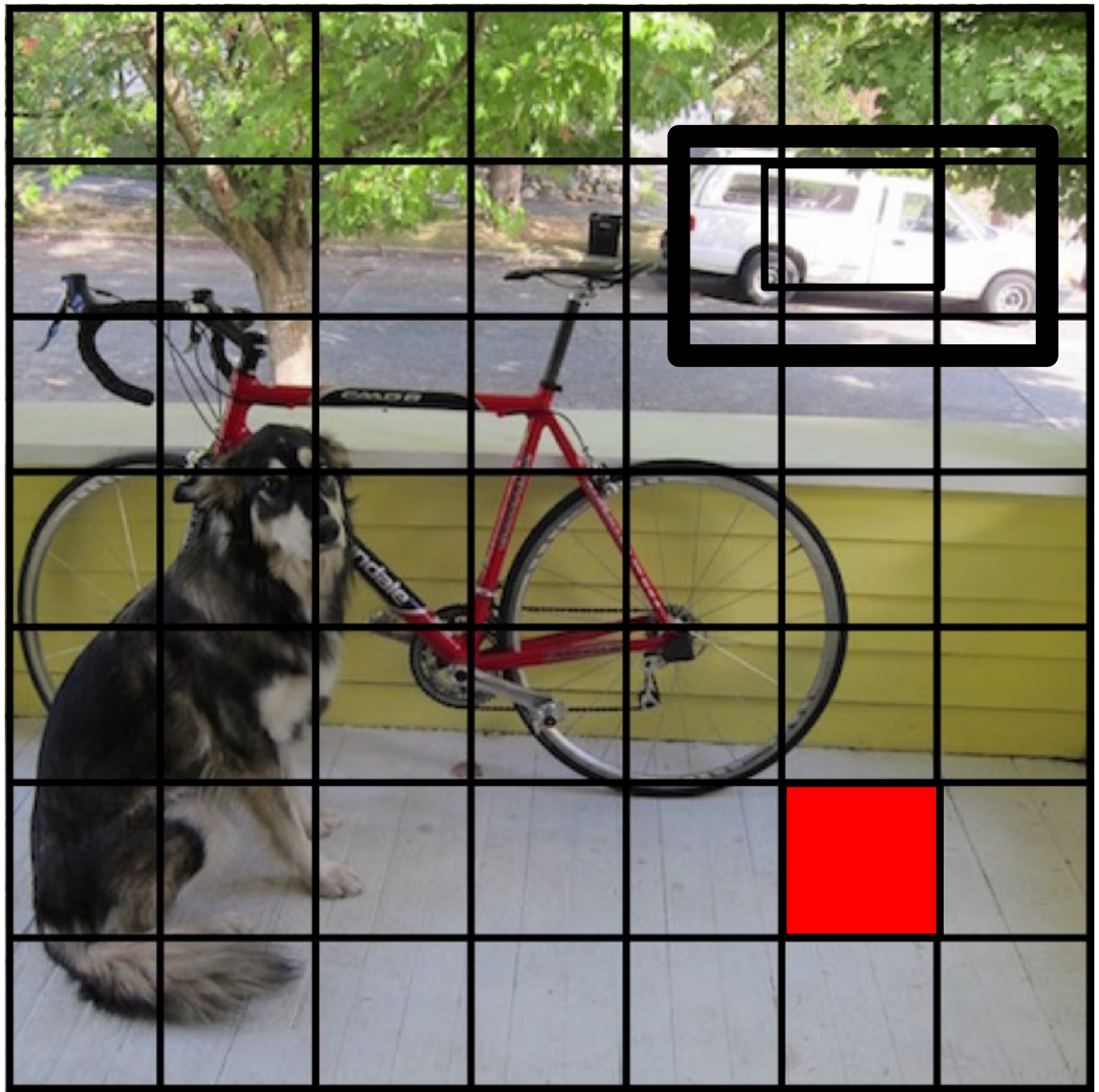


Each cell predicts boxes and confidences: P(Object)

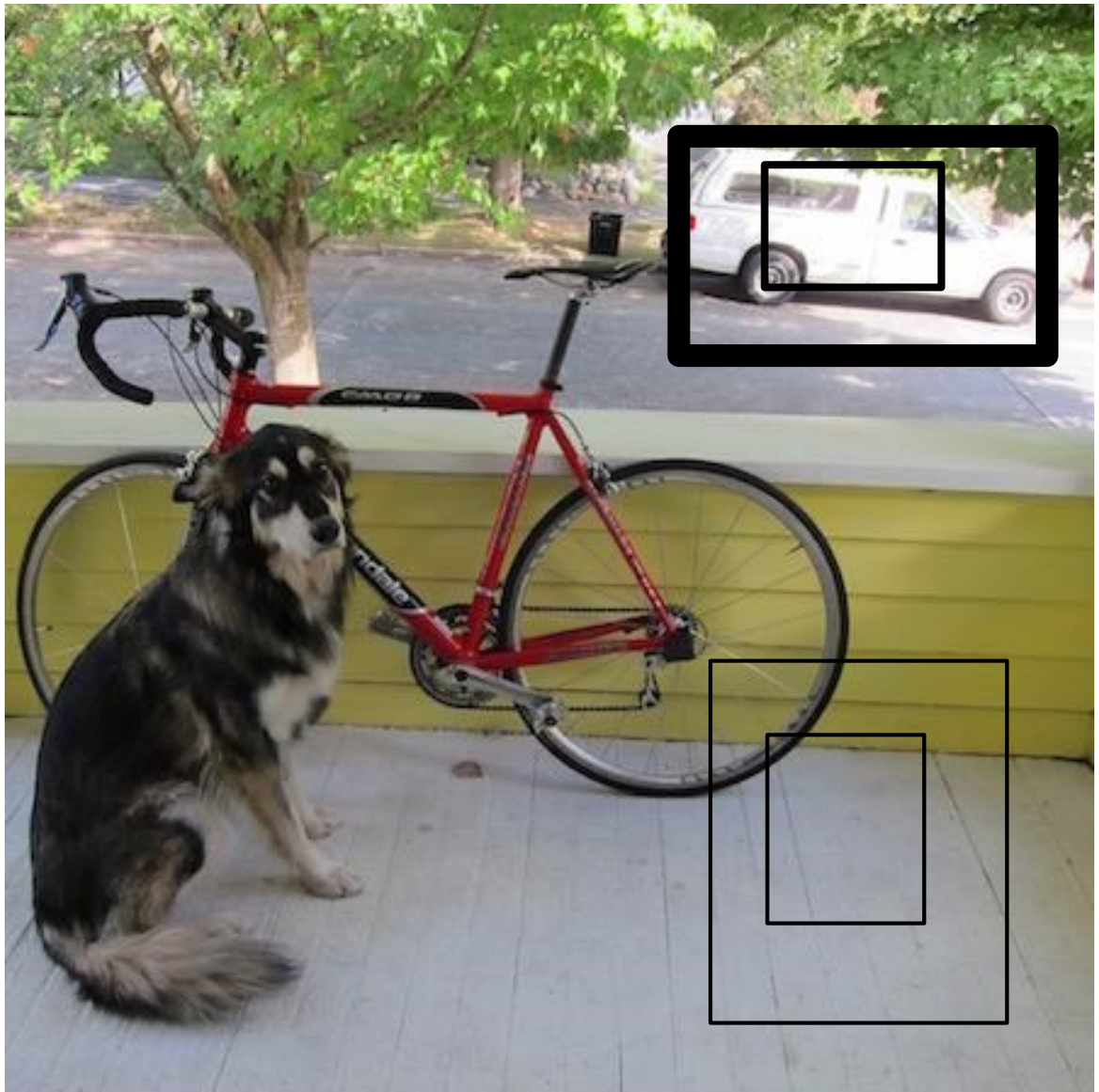


Each cell predicts boxes and confidences: P(Object)

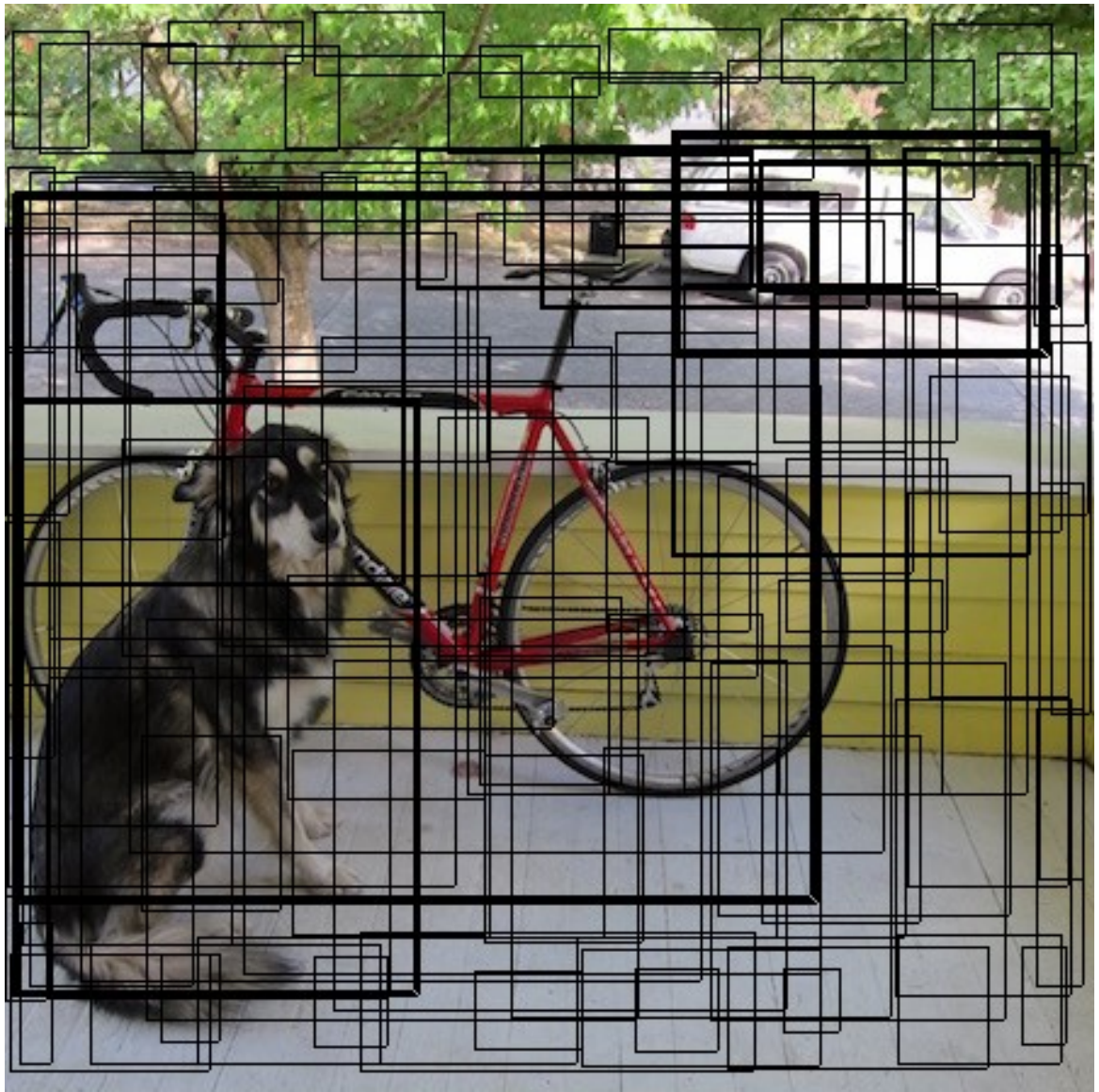
Each cell predicts boxes and confidences: P(Object)



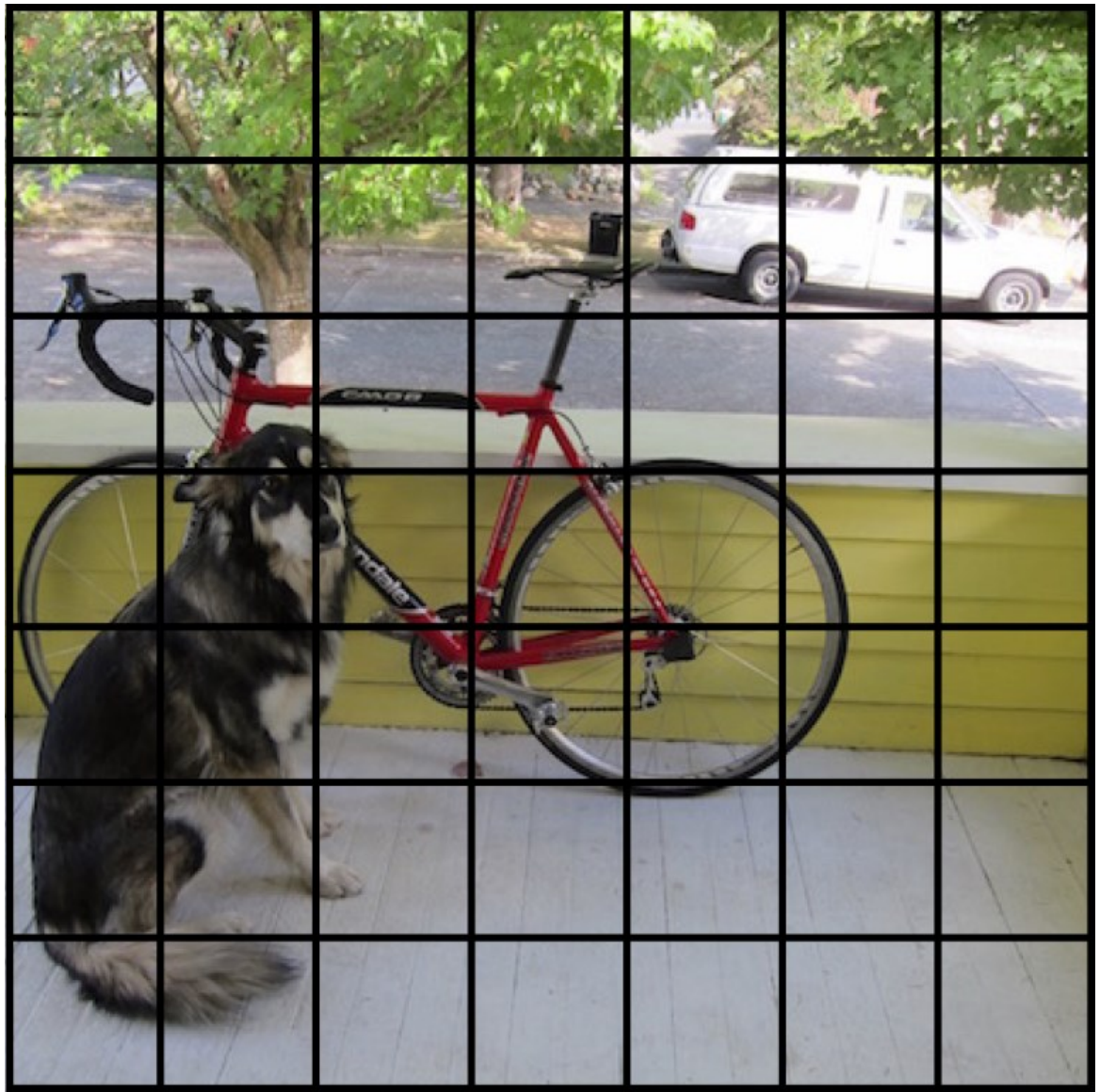
Each cell predicts boxes and confidences: P(Object)



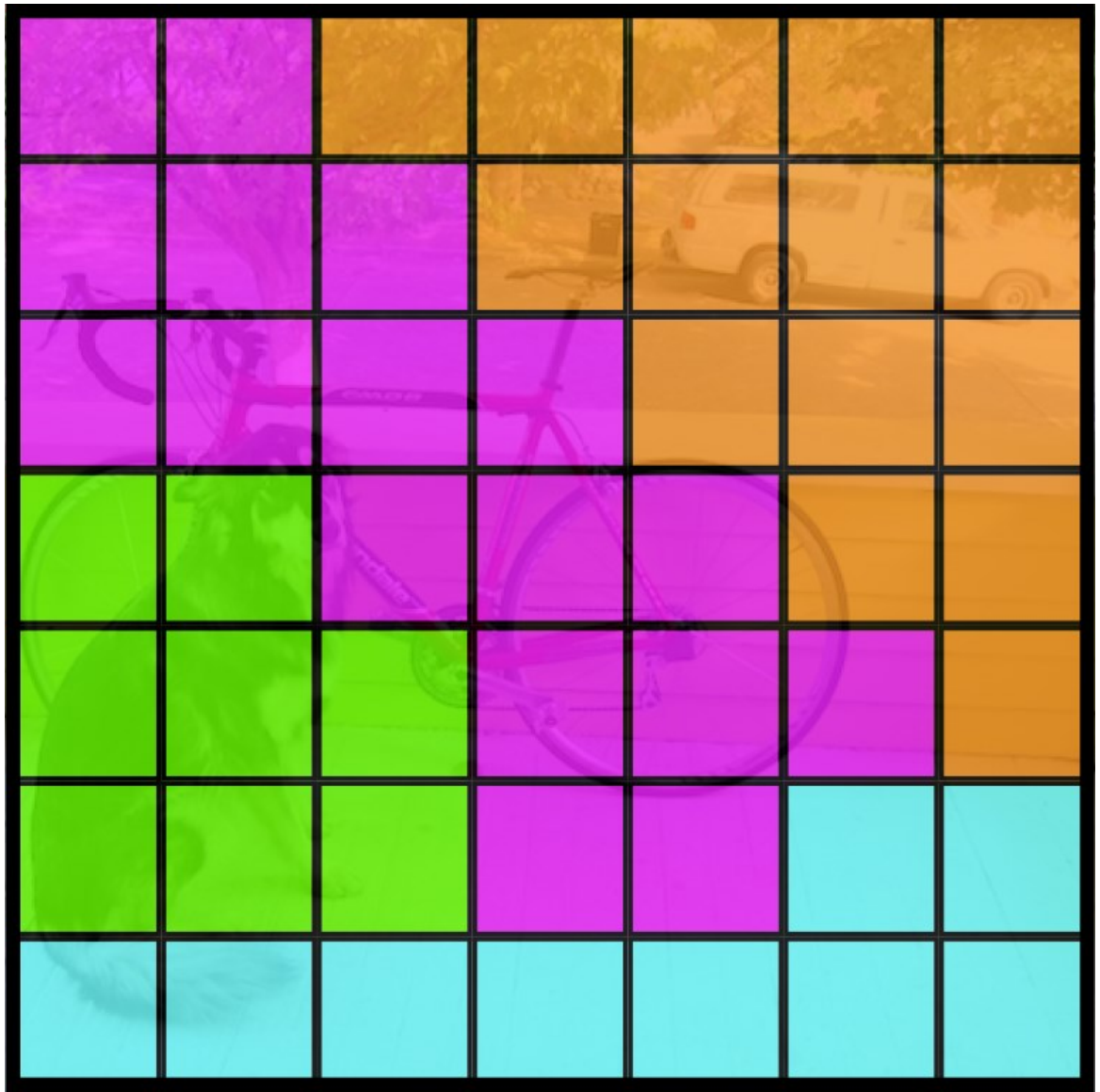
Each cell predicts boxes and confidences: P(Object)



Each cell also predicts a class probability



Each cell also predicts a class probability



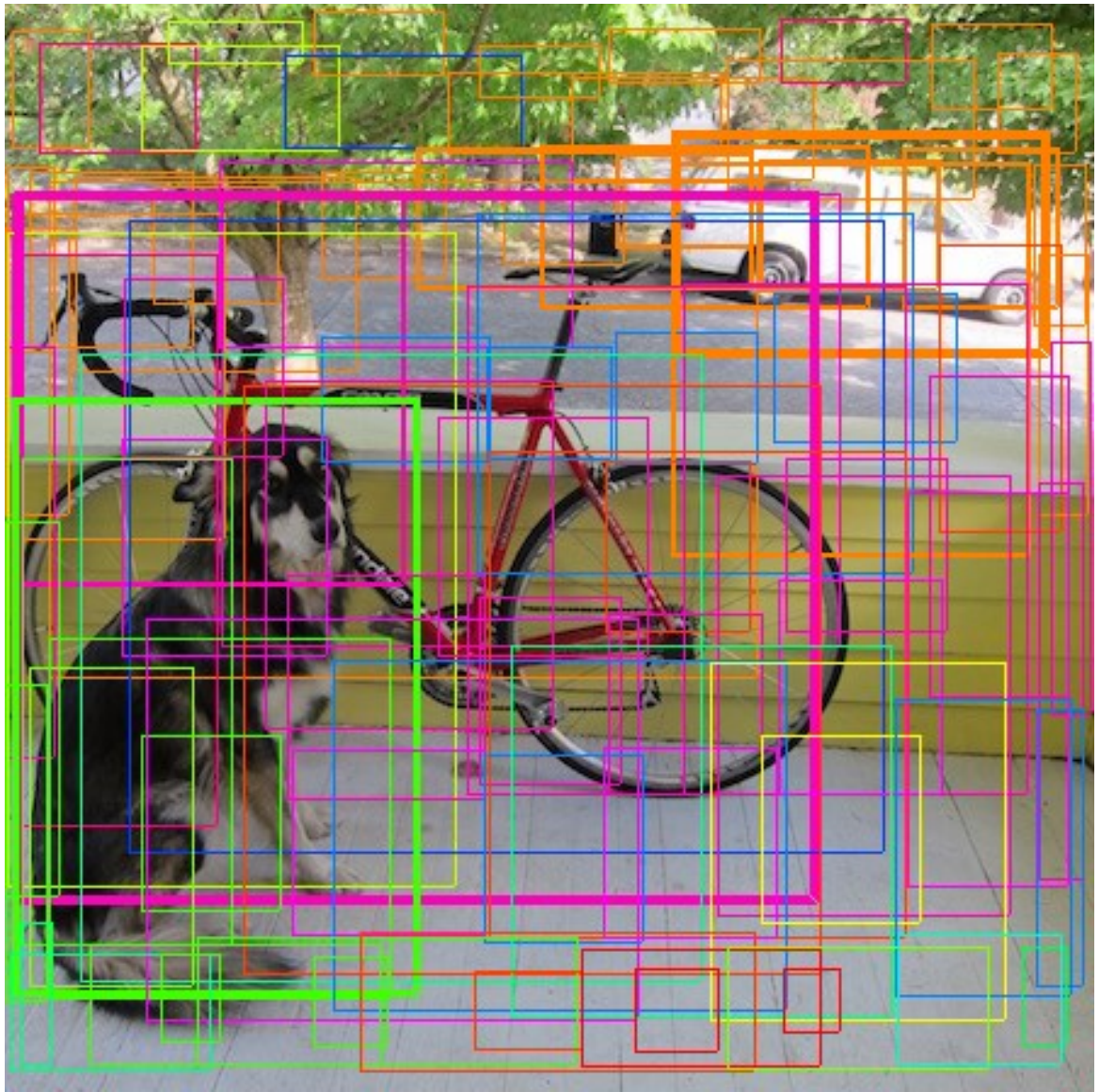
Bicycle

Car

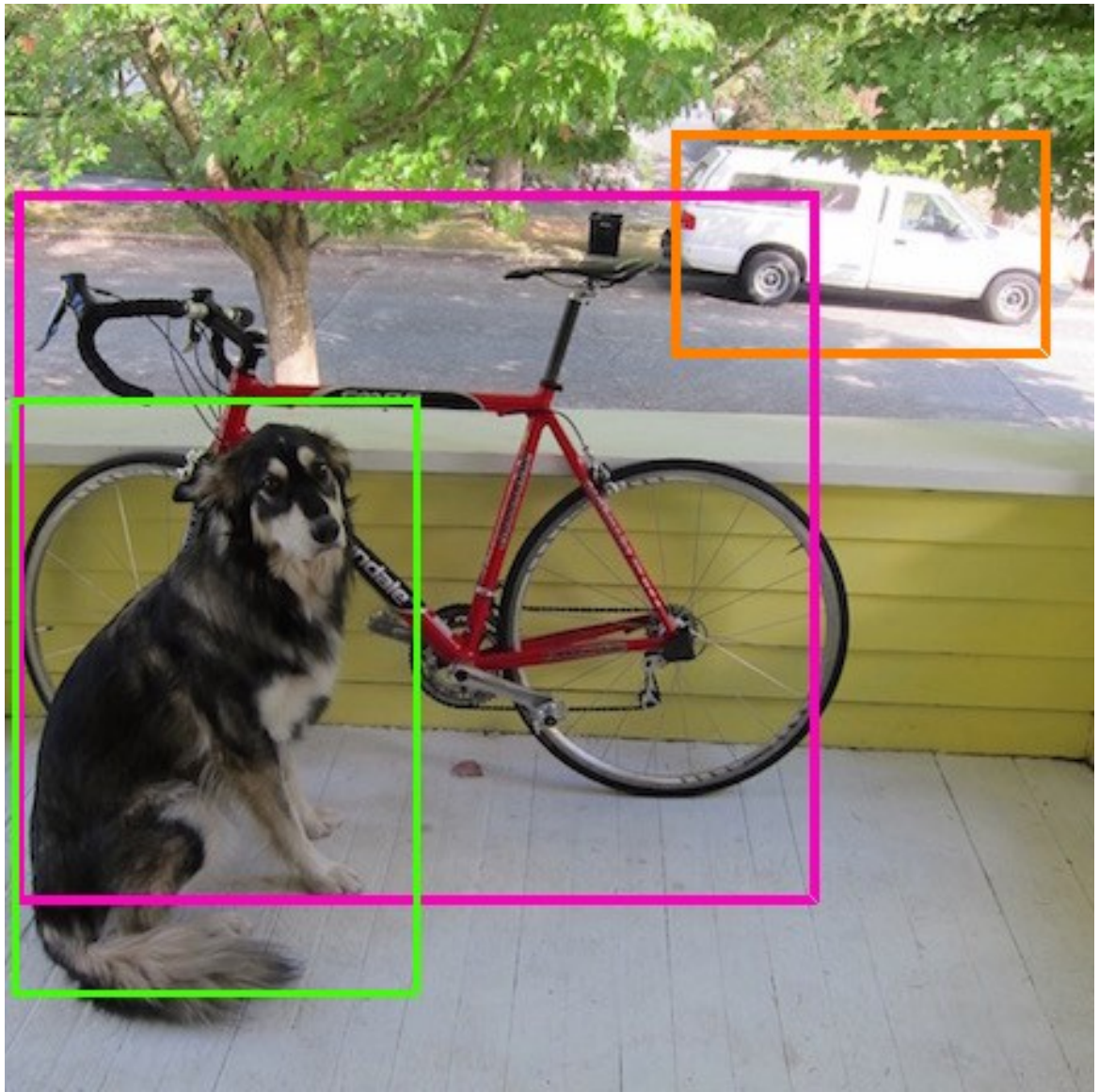
Dog

Dining Table

Then we combine the box and class predictions.



Finally we do NMS and threshold detections



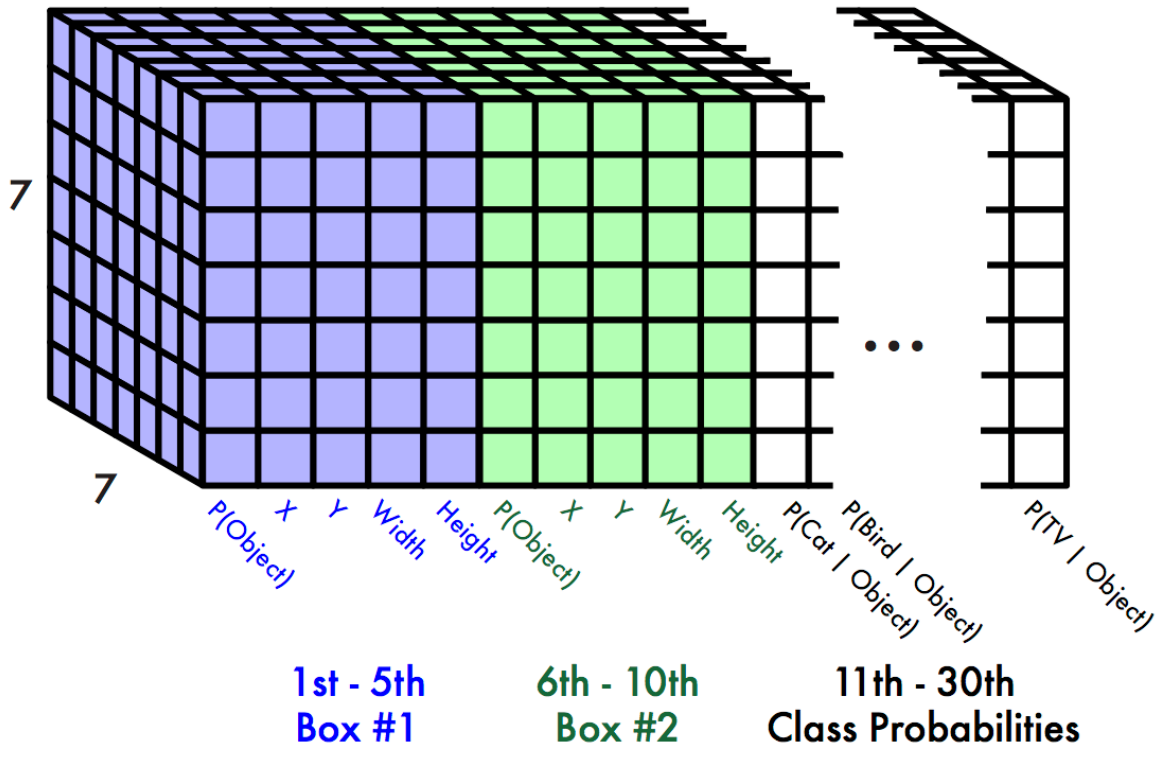
Outputs

Each cell predicts:

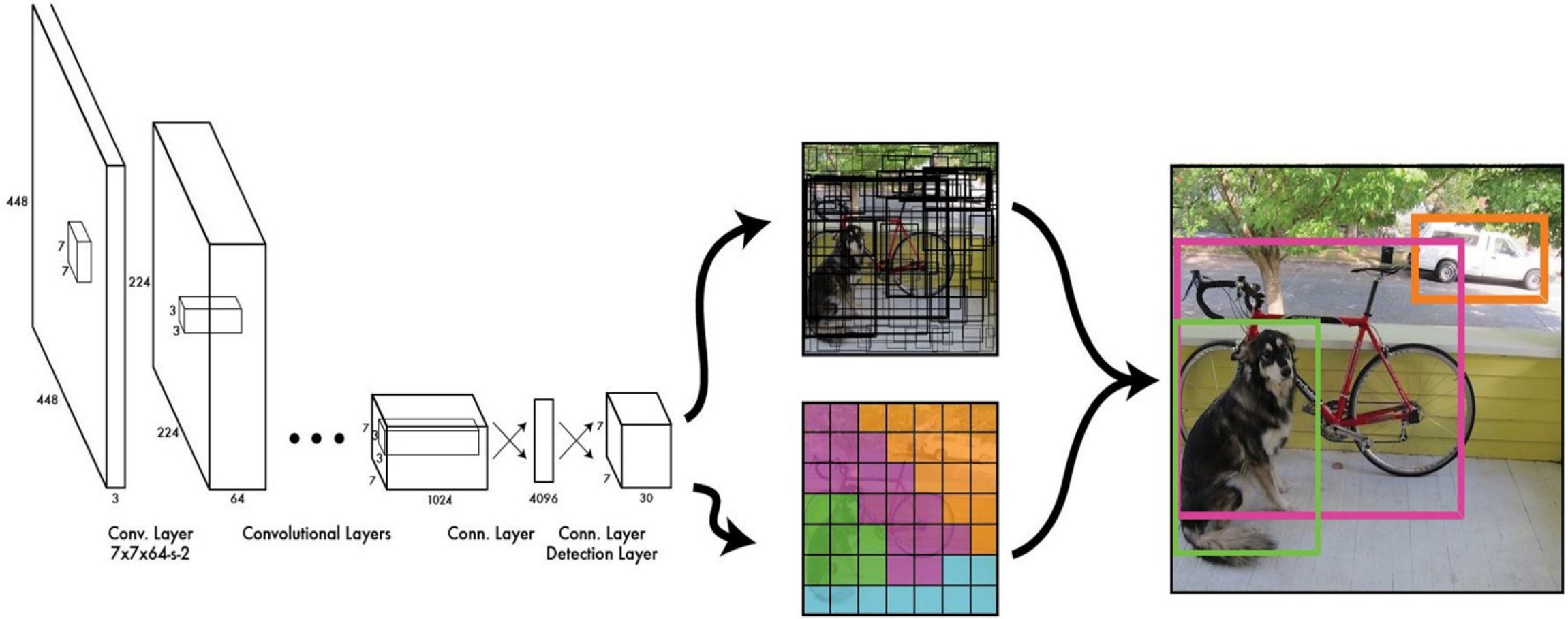
- For each bounding box:
  - 4 coordinates (x, y, w, h)
  - 1 confidence value
- Some number of class probabilities

For Pascal VOC:

- 7x7 grid
- 2 bounding boxes / cell
- 20 classes



YOLO


448
448
7
7
224
224
3
3
3
64
7
7
3
3
7
1024
4096
7
7
30
Conv. Layer
7x7x64-s-2
Convolutional Layers
Conn. Layer
Conn. Layer
Detection Layer

Loss Function

- In training, one predictor which has the highest IoU with the ground truth is responsible

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

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

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

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

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 \qquad (3)$$

$\mathbb{1}_{ij}^{obj}$ — **The *j*th bbox predictor** in ***cell i*** is "responsible" for that prediction

$\mathbb{1}_{ij}^{noobj}$

$\mathbb{1}_{i}^{obj}$ — If object appears in ***cell i***

Note that the loss function only penalizes classification error if an object is present in that grid cell (hence the conditional class probability discussed earlier). It also only penalizes bounding box coordinate error if that predictor is "responsible" for the ground truth box (i.e. has the highest IOU of any predictor in that grid cell).

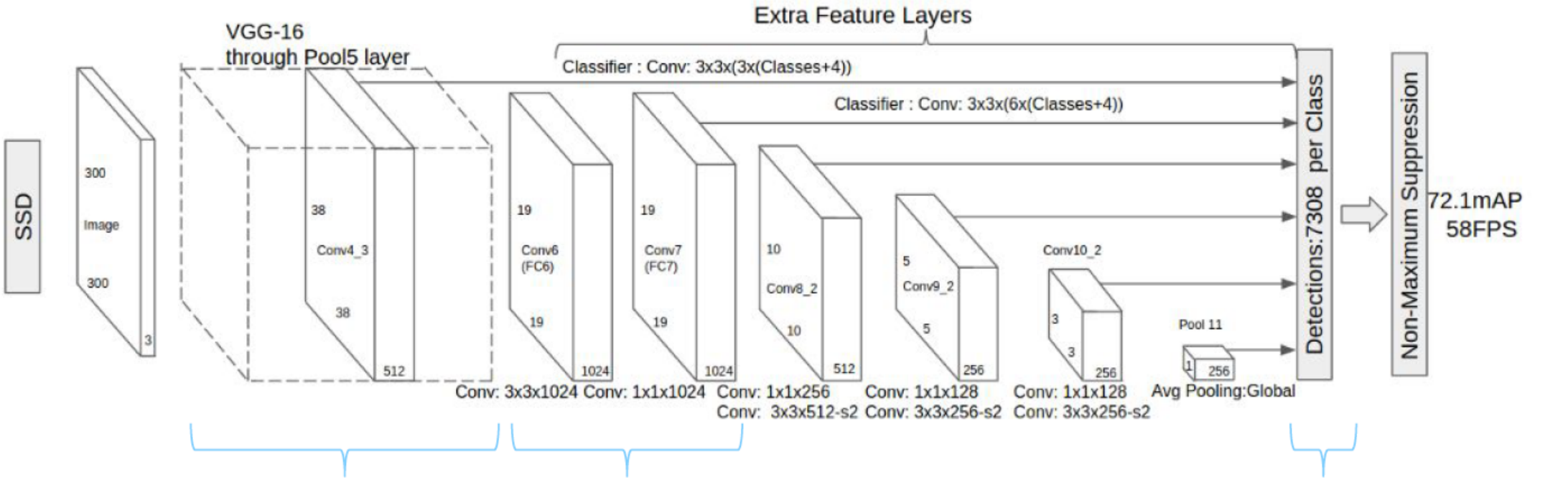
Problems

- Each grid cell can predict only B(=2) bounding boxes and one class probability
  - Not good for small objects that flock together
- Uses relatively coarse features
  - Locations of bboxes are inaccurate
- Loss function treats errors in small bbox and big bbox equally
  - Not good for scoring

SSD

- The core of SSD is
  - Predicting category scores and bbox offsets for a fixed set of default boxes
  - From each cell of multiple convolutional feature maps

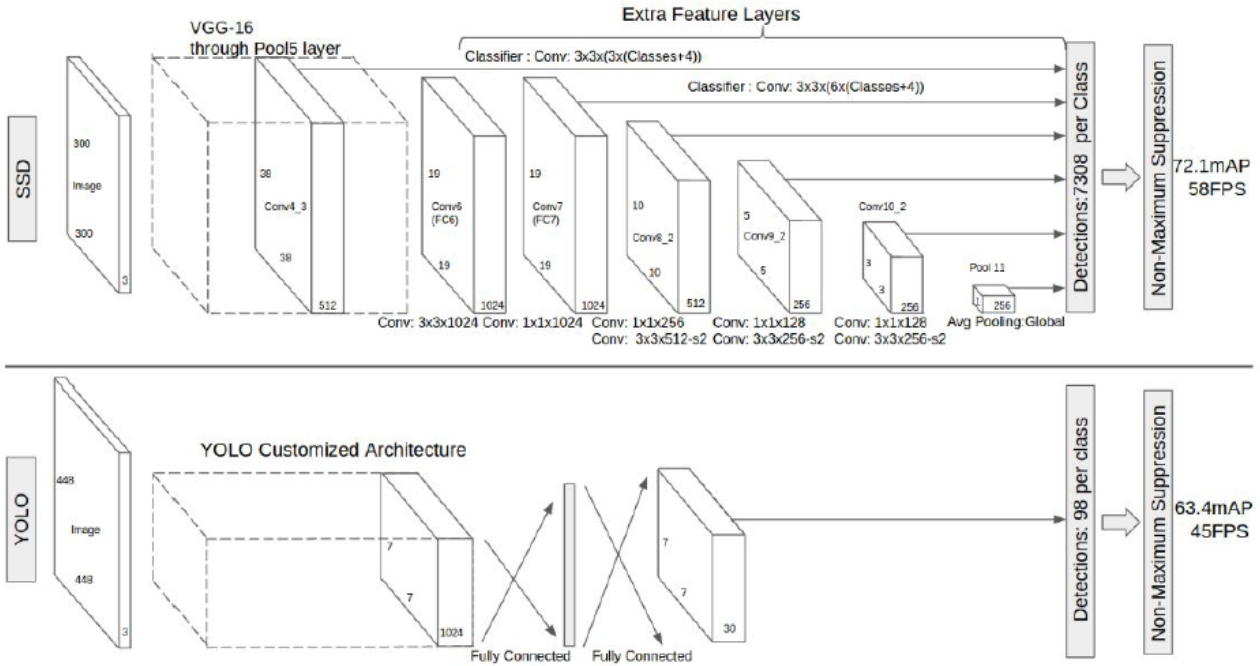
Overall Structure



Multi-scale feature maps for detection: observe how conv feature maps decrease in size and allow predictions at multiple scales

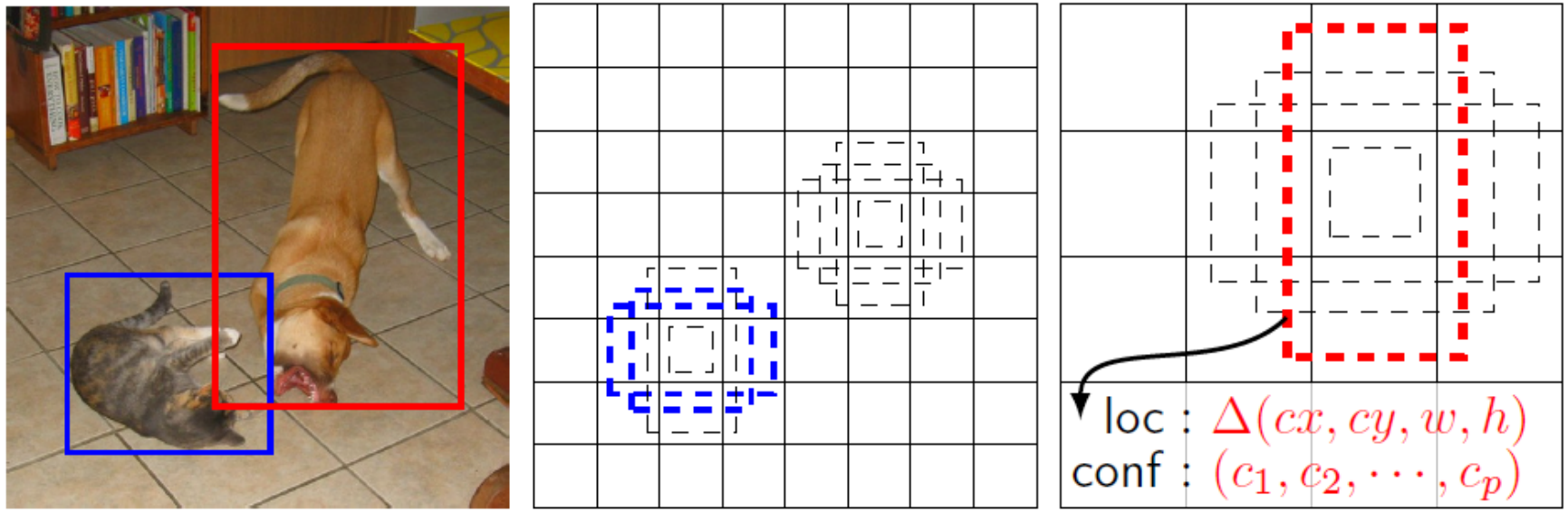
Comparison to YOLO

- The size of an input image is different
- SSD uses multiple feature maps
- While YOLO outputs are box positions, SSD outputs are box offsets of six default boxes



SSD

- Default boxes and aspect rations
  - Similar to the anchors of Faster R-CNN, with the difference that SSD applies them on several feature maps of different resolutions
  - There are matched two default boxes with the cat and one with the dog at the below example, which are treated as positives and the rest as negatives.



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

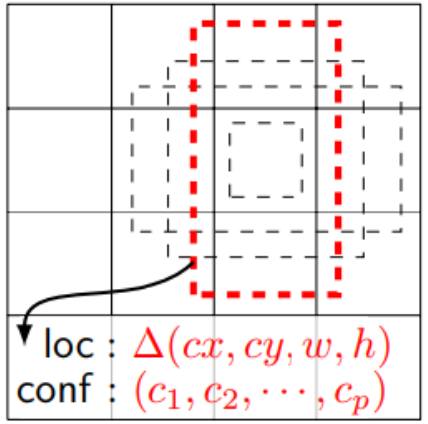
Default Box

Default box scale

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}(k - 1), k \in [1, m]$$

$$s_{min} = 0.2, s_{max} = 0.9$$

각 level에서 상대적인 scale의 크기 설정.
SSD300의 경우 6개의 level로,
[0.2, 0.32, 0.44, 0.56, 0.68, 0.8, 0.9]



Default box aspect ratios

$$a_r \in [1, 2, 3, \frac{1}{2}, \frac{1}{3}]$$

$$w_k^a = s_k\sqrt{a_r}$$

$$h_k^a = s_k/\sqrt{a_r}$$

# of Default box: 6
1, 2, 3, 1/2, 1/3 5개의 비율로 width, height 설정.

# of Default box: 4
1, 3, 1/3 3개의 비율로 width, height 설정.

추가로, ratio가 1이고 $s'_k = \sqrt{s_k s_{k+1}}$인 box를 둠.

Matching Strategy

- Default box와 ground-truth의 IoU가 가장 높은 default box는 positive example로 학습
- IoU가 0.5가 넘는 경우도 모두 positive로 취급
  → YOLO는 1개의 물체에 대하여 1개의 box만 positive

Detector

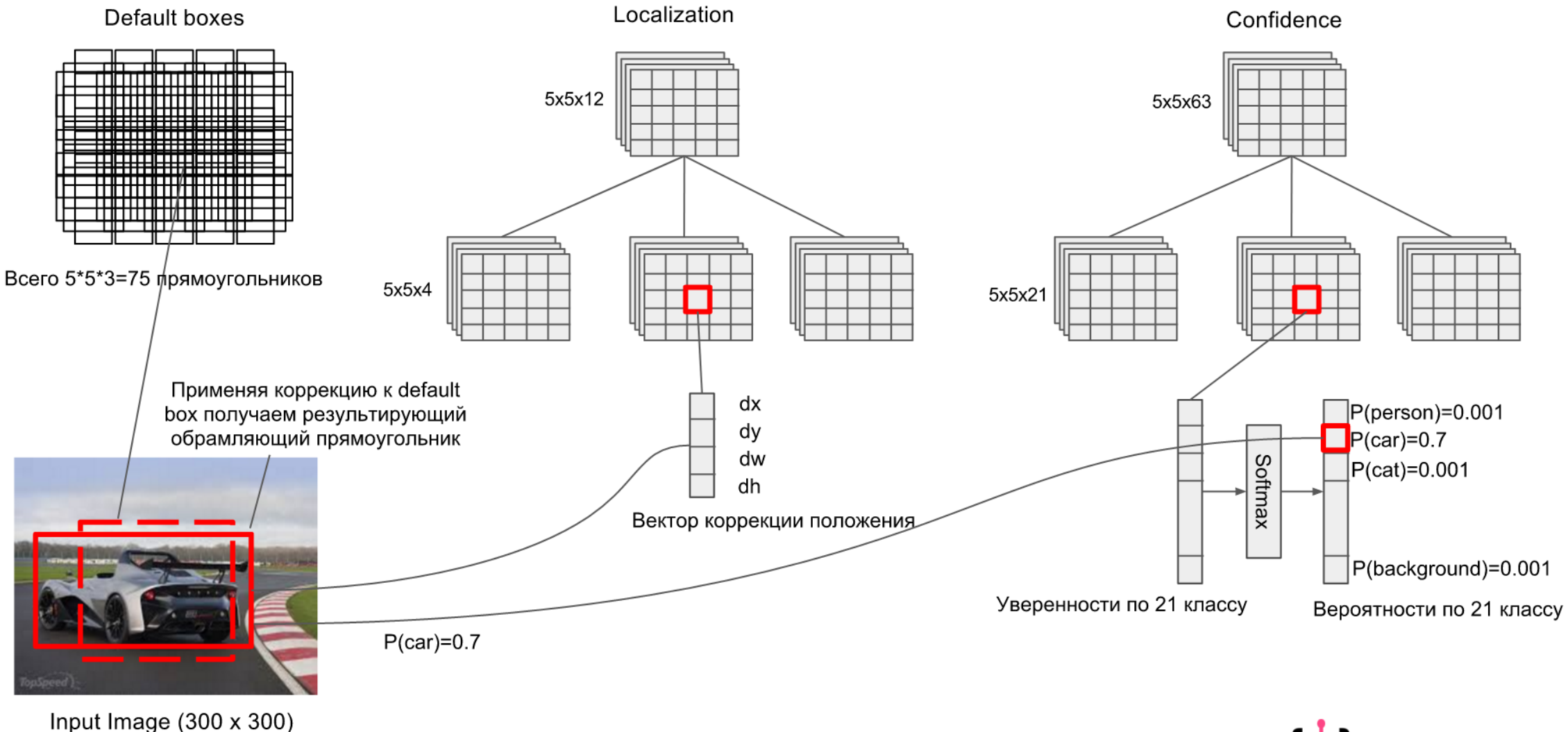
- Each detector will output a single value, so we need

  (classes + 4) detectors for a detection

- As we have #default boxes, we need

  (classes + 4) x #default boxes detectors

Detector


Default boxes
Всего 5*5*3=75 прямоугольников
Localization
5x5x12
5x5x4
dx
dy
dw
dh
Вектор коррекции положения
Confidence
5x5x63
5x5x21
Softmax
P(person)=0.001
P(car)=0.7
P(cat)=0.001
P(background)=0.001
Уверенности по 21 классу
Вероятности по 21 классу
Применяя коррекцию к default box получаем результирующий обрамляющий прямоугольник
P(car)=0.7
Input Image (300 x 300)

Loss Function

- Localization loss의 경우 YOLO와 같이 직접 좌표값을 사용하는 것이 아니라, RCNN 계열과 같이 delta(Bbox를 얼마나 움직일지의 값)으로 loss를 계산함

$$L(x, c, l, g) = \frac{1}{N}(\underbrace{L_{conf}(x, c)}_{\text{confidence loss}} + \underbrace{\alpha}_{\substack{\text{is 1 by} \\ \text{cross-validation}}} \underbrace{L_{loc}(x, l, g)}_{\text{localization loss}})$$

confidence loss
softmax loss

localization loss
Smooth L1 loss

is 1 by
cross-validation

N: number of default matched BBs
x: is 1 if the default box is matched to a determined ground truth box, and 0 otherwise
l: predicted bb parameters
g: ground truth bb parameters
c: class

Hard Negative Mining

- No. of default boxes(SSD300) : 8,732
- 거의 대부분의 default box는 background(negative example)로 학습
  → class imbalance problem
- YOLO에서는 confidence로 해결 (confidence = 0이 되도록)
- Hard Negative Mining
  - Negative sample을 confidence loss 순으로 정렬하여 높은 순서대로 sorting 후 positive vs negative 비율을 1:3으로 구성하여 학습
  - 즉 background를 뽑은 대부분의 default box들 중에서 backgroun일 확률이 가장 낮게 나온 순서대로 positive의 3배만큼만 뽑아서 사용

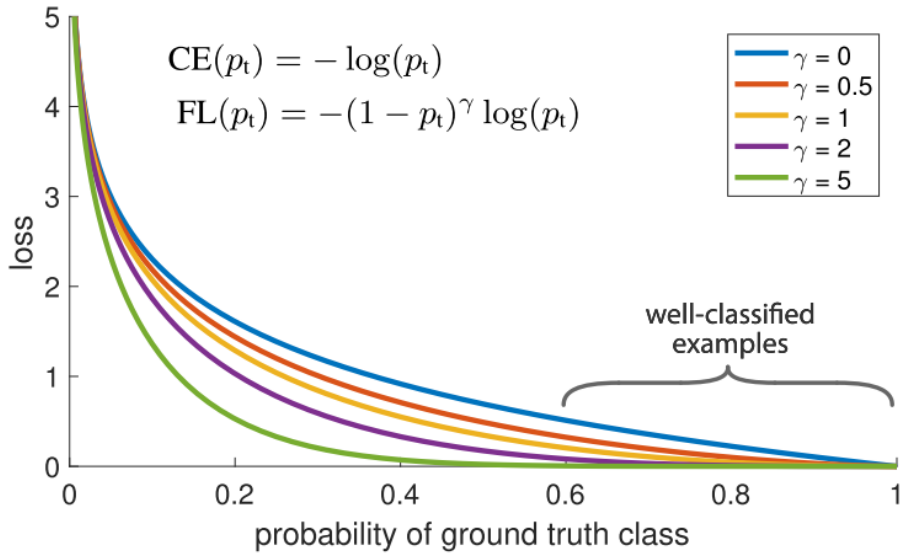
Class Imbalance Problem

- Class 간의 data가 균형이 안맞는 경우
  - Weighted cross-entropy
    - Cross entropy 식에 class 별로 weight를 다르게 주어서 사용

$$L = -\sum_{C} w_c \cdot \log p$$

  - Focal Loss
    - Cross entropy에 비해 높은 확률의 경우에는 상대적으로 작은 loss를 나은 확률의 경우에는 높은 loss를 주도록 $(1\text{-}p)^{\gamma}$를 곱해줌

$$L = -\sum (1-p)^{\gamma} \cdot \log p$$



Results

- PASCAL '07

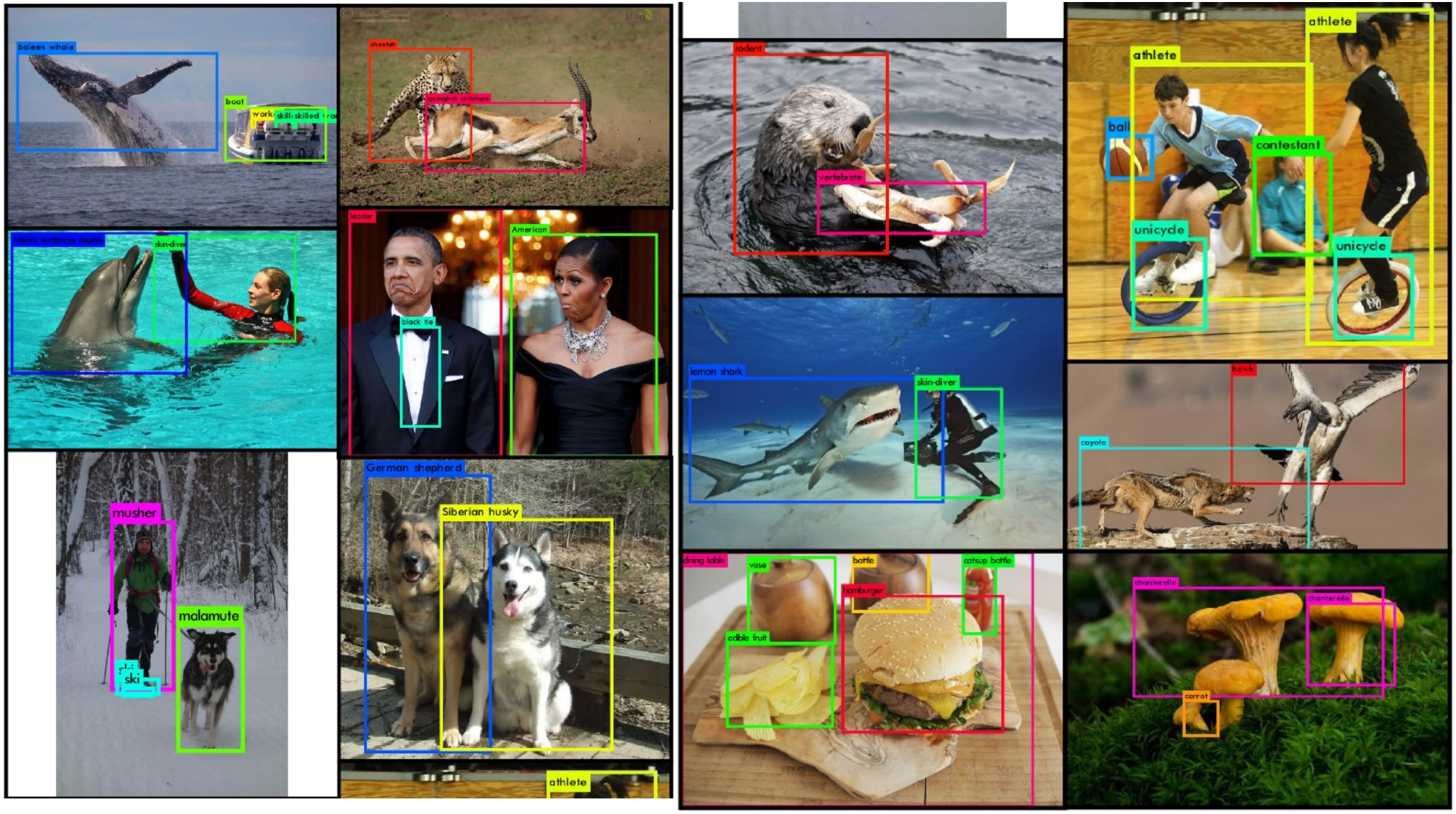
Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	**74.8**	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	**65.5**	**52.1**	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	**79.8**	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	**72.3**	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	**75.1**	**79.8**	79.5	**74.5**	63.4	51.9	**84.9**	**85.6**	**87.2**	**56.6**	80.1	70.0	**85.4**	**84.9**	**80.9**	**78.2**	**49.0**	**78.4**	72.4	**84.6**	**75.5**

PASCAL '12

Method	*mAP*	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	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 [2]	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 [5]	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	70.3	84.2	76.3	69.6	53.2	40.8	78.5	73.6	88.0	50.5	73.5	**61.7**	85.8	80.6	81.2	77.5	44.3	73.2	**66.7**	81.1	65.8
SSD500	**73.1**	**84.9**	**82.6**	**74.4**	**55.8**	**50.0**	**80.3**	**78.9**	88.8	**53.7**	76.8	59.4	**87.6**	**83.7**	**82.6**	**81.4**	**47.2**	**75.5**	65.6	**84.3**	**68.1**

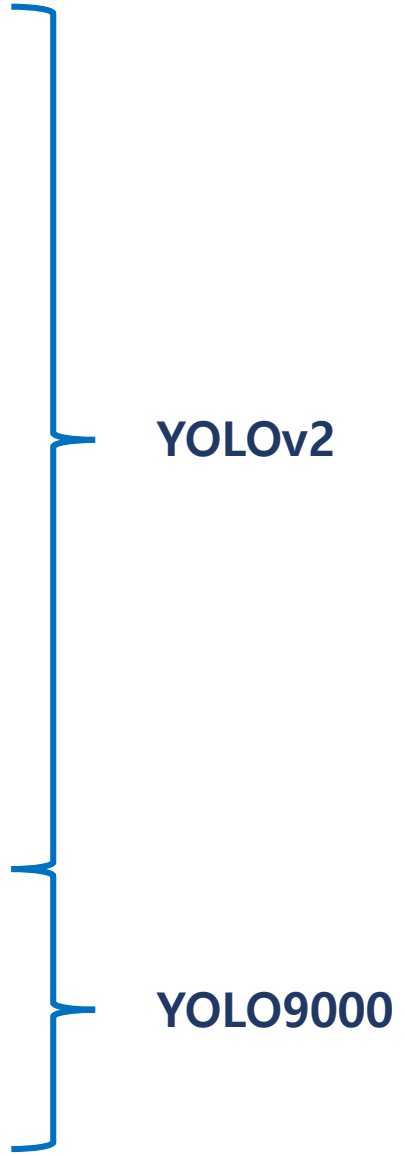
Let's See the Result First

Let's See the Result First



YOLO9000 - Better, Faster, and Stronger

- Better
  - Batch normalization
  - High resolution classifier
  - Convolution with anchor boxes
  - Dimension clusters
  - Direct location prediction
  - Fine-grained features
  - Multi-scale training
- Faster
  - Darknet-19
  - Training for classification
  - Training for detection
- Stronger
  - Hierarchical classification
  - Dataset combination with Word-tree
  - Joint classification and detection



Introduction & Motivation

- Detection frameworks have become increasingly fast and accurate → However, most detection methods are still constrained to a small set of objects.

- We would like detection to scale to level of object classification → However, labelling images for detection is far more expensive than labelling for classification.

- Maybe we cannot see detection datasets on the same scale as classification datasets in the near future.

Introduction

- Authors propose a new method to expand the scope of current detection systems by using classification dataset we already have.

- Joint training algorithm is also proposed that allows to train object detectors on both detection and classification data

- First, improving upon the base YOLO detection system to produce YOLOv2 (SOTA real-time detector)
- Then, using dataset combination method and joint training algorithm to train model on more than 9000 classes!

To Improve YOLO

- YOLO's shortcomings relative to SOTA detection systems
  - YOLO makes a significant number of localization errors
  - YOLO has relatively low recall compared to region proposal based method

- Focus mainly on improving recall and localization while maintain classification accuracy.

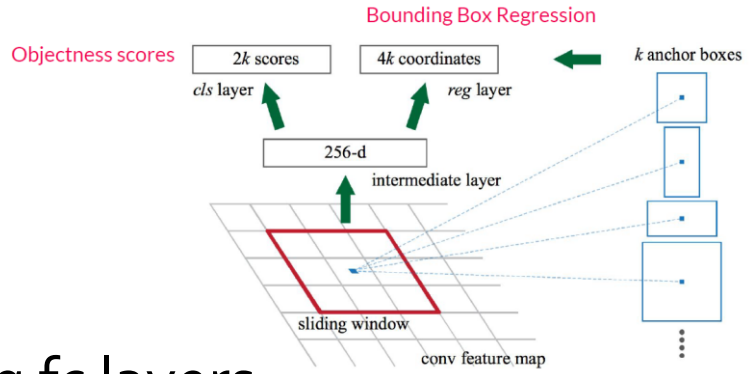
- It is still very important to detect FAST!

Better

- Batch Normalization
  - Adding BN on all of conv layers in YOLO – 2% improvement in mAP
  - Removing dropout without overfitting

- High Resolution Classifier
  - YOLO trains the classifier network @224x224 and increase resolution to 448 for detection → The network has to simultaneously switch to learning object detection and adjust to the new input resolution
  - Fine tuning the classifier network @448x448 resolution for 10 epochs on ImageNet – almost 4% improvement in mAP

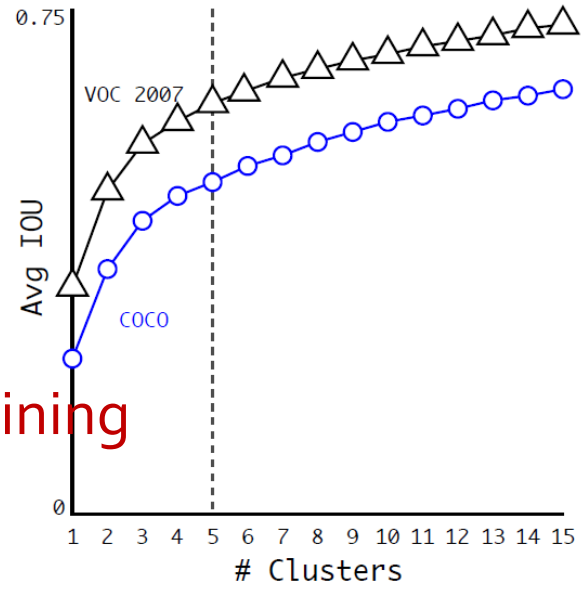
Better

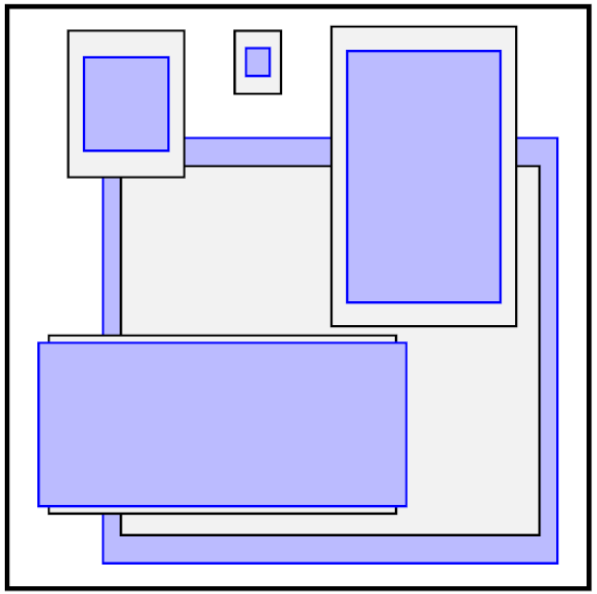


- Convolution with Anchor Boxes
  - YOLO predicts the coordinates of b-boxes directly using fc layers.
  - Faster R-CNN predicts b-boxes using hand-picked priors(anchor boxes).
  - YOLOv2 removes the fc layers and use anchor boxes to predict bounding boxes
  - Shrinking the network to operate on 416x416 input images instead of 448x448
    - YOLO's CNN downsample the image by a factor of 32
    - Objects, especially large objects, tend to occupy the center of the image, so it is better to have one cell in the middle than to have four cells. → odd number of grid cells are required(416x416 → 13x13)
  - Predicting class and objectness for every anchor box
  - Using anchor boxes makes a small decrease in accuracy
    - 69.5mAP to 69.2mAP but 81% recall to 88% recall → high recall means there is more room to improve

Better

- Dimension Clusters
  - The problem with anchor boxes is that they are hand-picked
  - Running k-means clustering on the training set b-boxes to find good priors
  - To prevent larger boxes from making more error, they use a different distance metric(not Euclidian distance)
    - d(box, centroid) = 1 – IOU(box, centroid)
  - k=5 is chosen as a good tradeoff between complexity and high recall





Box Generation	#	Avg IOU
Cluster SSE	5	58.7
Cluster IOU	5	61.0
Anchor Boxes [15]	9	60.9
Cluster IOU	9	67.2

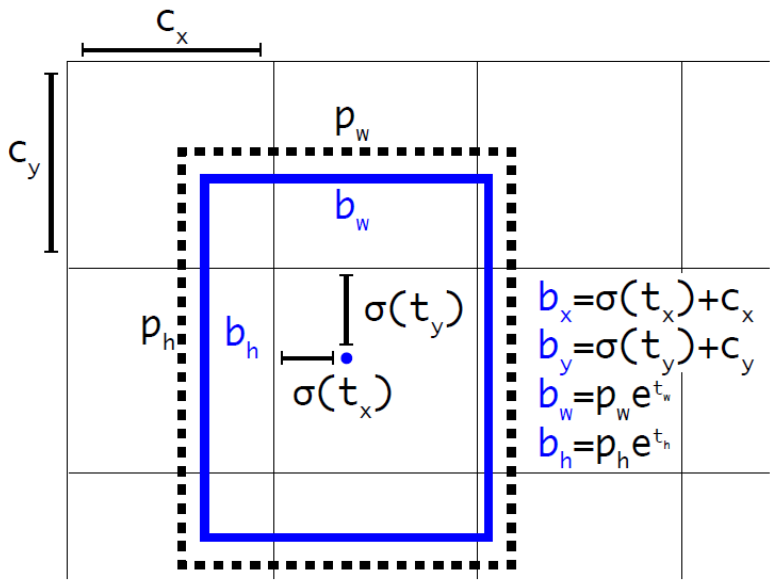
Table 1: Average IOU of boxes to closest priors on VOC 2007.

Better

- Direct Location Prediction
  - Another problem with anchor boxes is instability, in RPNs the anchor box can be anywhere in the image, regardless of what location predicted the box
  - Instead of predicting offsets, YOLOv2 predicts locations relative to the location of the grid cells
  - 5 bounding boxes for each cell, and 5 values for each bounding box → almost 5% improvement over the version with anchor boxes

RCNN's b-box regression

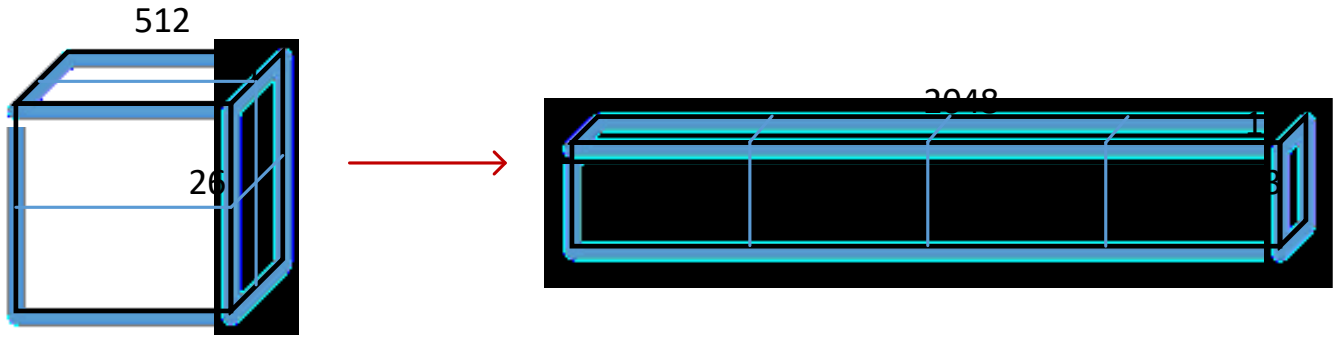
$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$



Different equations – direct prediction

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$
$$Pr(\text{object}) * IOU(b, \text{object}) = \sigma(t_o)$$

Better



- Fine-Grained Features
  - Modified YOLO uses 13x13 feature maps to predict detections
    - ➢ Good at large object, finer grained features are required for small objects
  - Simply adding a passthrough layer that brings features from an earlier layer at 26x26 resolution
  - Turn 26x26x512 feature map into a 13x13x2048 feature map which can be concatenated with original features
  - Detector runs on top of this expanded features
  - 1% performance increase

Better

- Multi-Scale Training

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	**78.6**	40

  - Removing fc layers → can be resized on the fly
  - Every 10 batches YOLOv2 randomly choose a new image dimension size in {320, 352, …, 608}
  - Effect of the input size
    - At low resolution, fair detection performance but runs fast
    - At high resolution, SOTA detection and slower but still above real time speeds

Further Experiments

- Pascal VOC2012 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

- COCO2015 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

Faster

- Darknet-19
  - mostly 3x3 filters (similar to VGG)
  - Following the work on Network in Network (NIN), use global average pooling to make predictions as well as 1x1 filters to compress the feature representation between 3x3 convolutions
  - Darknet-19: 19 convolutional layers & 5 max-pooling layers
  - 5.58 billion operations: 72.9% top-1 & 91.2% top-5 accuracy (30.69 billion operations & 90.0% top-5 accuracy for VGG-16)

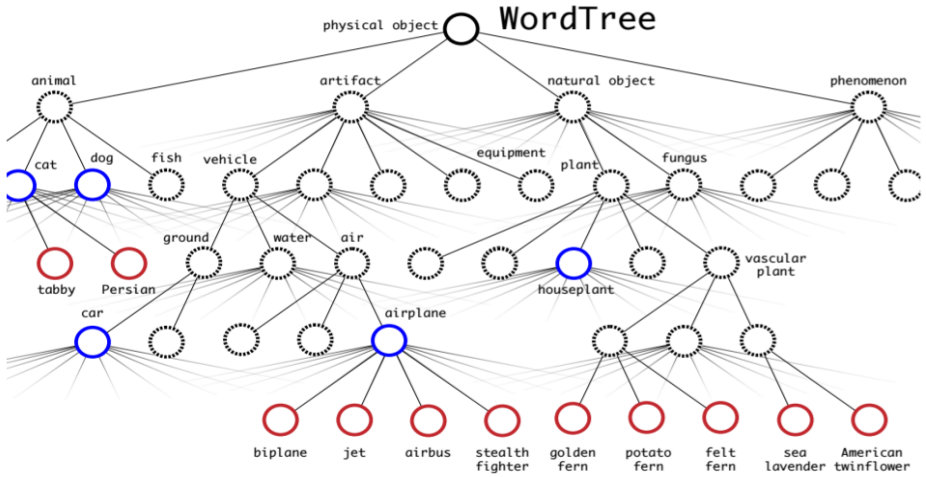
Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	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			

Faster

- Training for classification
  - ImageNet 1000 classes for 160 epochs
  - Standard data augmentation: random crops, rotations, hue, saturation, and exposure shifts
  - Initial training : 224x224 → 448x448 fine-tuning for 10 epochs
    - Higher resolution achieves a top-5 accuracy of 93.3%
- Training for detection
  - Adding 3x3 conv layers with 1024 filters each followed by a final 1x1 conv layer
  - For VOC, predicting 5 boxes with 5 coordinates each and 20 classes per box, so 125 filters
  - 160 epochs with a start learning rate of 10^{-3} dividing it by 10 at 60 and 90 epochs

Stronger

- Hierarchical classification
  - ImageNet labels are pulled from WordNet, a language database that structures concepts and how they relate
    - "Norfolk terrier" and "Yorkshire terrier" are both hyponyms of "terrier" which is a type of "hunting dog", which is a type of "dog", which is a "canine", etc.
  - WordNet is structured as a directed graph, not a tree, because language is complex
    - To make a tree not a graph, If a concept has tow paths to the root, choose the shorter path
  - Root note is a "Physical Object"



Stronger

- Hierarchical classification
  - To perform classification with WordTree, predicting conditional probabilities at every node for the probability of each hyponym of that synset given that synset

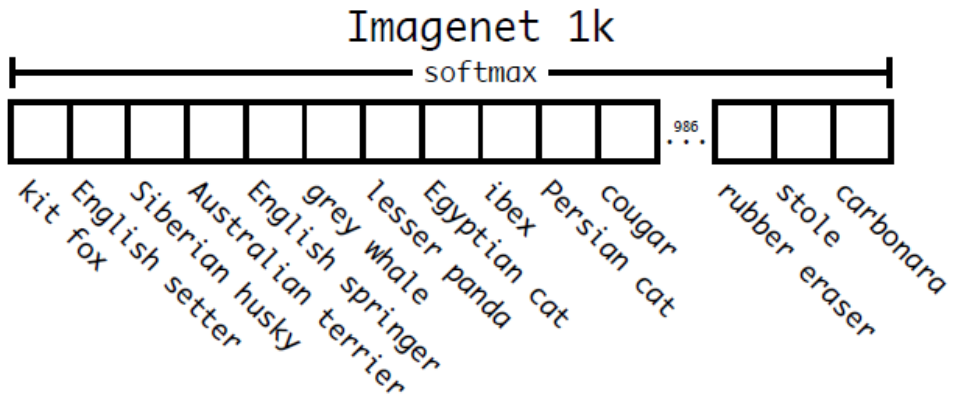
$$Pr(\text{Norfolk terrier}|\text{terrier})$$
$$Pr(\text{Yorkshire terrier}|\text{terrier})$$
$$Pr(\text{Bedlington terrier}|\text{terrier})$$
$$\ldots$$

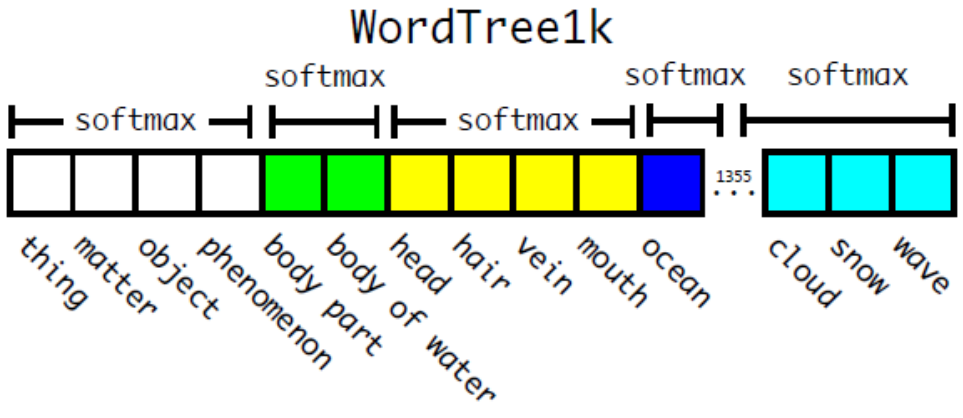
  - Then if a picture of a Norfolk terrier is encountered, its probability is calculated as below

$$Pr(\text{Norfolk terrier}) = Pr(\text{Norfolk terrier}|\text{terrier})$$
$$*Pr(\text{terrier}|\text{hunting dog})$$
$$*\ldots*$$
$$*Pr(\text{mammal}|Pr(\text{animal})$$
$$*Pr(\text{animal}|\text{physical object})$$

Stronger

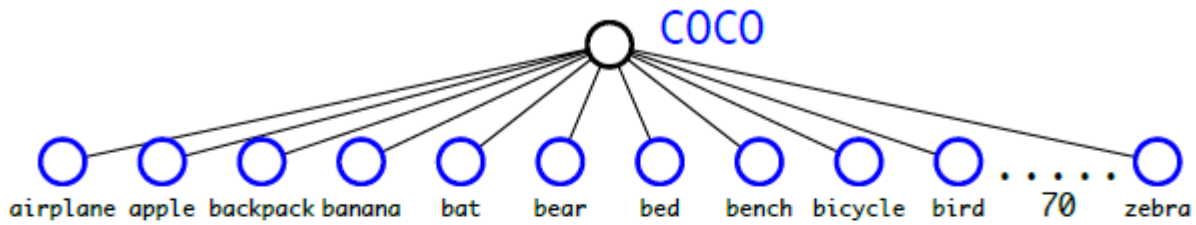
- Hierarchical classification
  - To build WordTree1k, adding in all of the intermediate nodes which expands the label space from 1000 to 1369
  - During training, ground truth labels are propagated up
    - If an image is labelled as a "Norfolk terrier" it also gets labelled as a "dog" and a "mammal", etc.
  - To compute the conditional probabilities the model predicts a vector of 1369 values and computes the softmax over all synsets that are hyponyms of the same concept → 90.4% top-5 accuracy

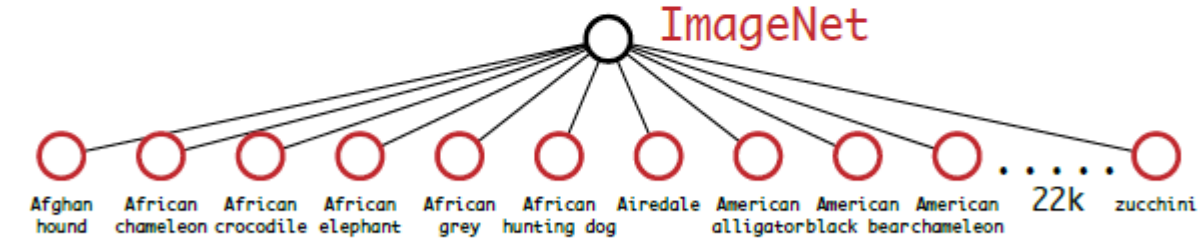


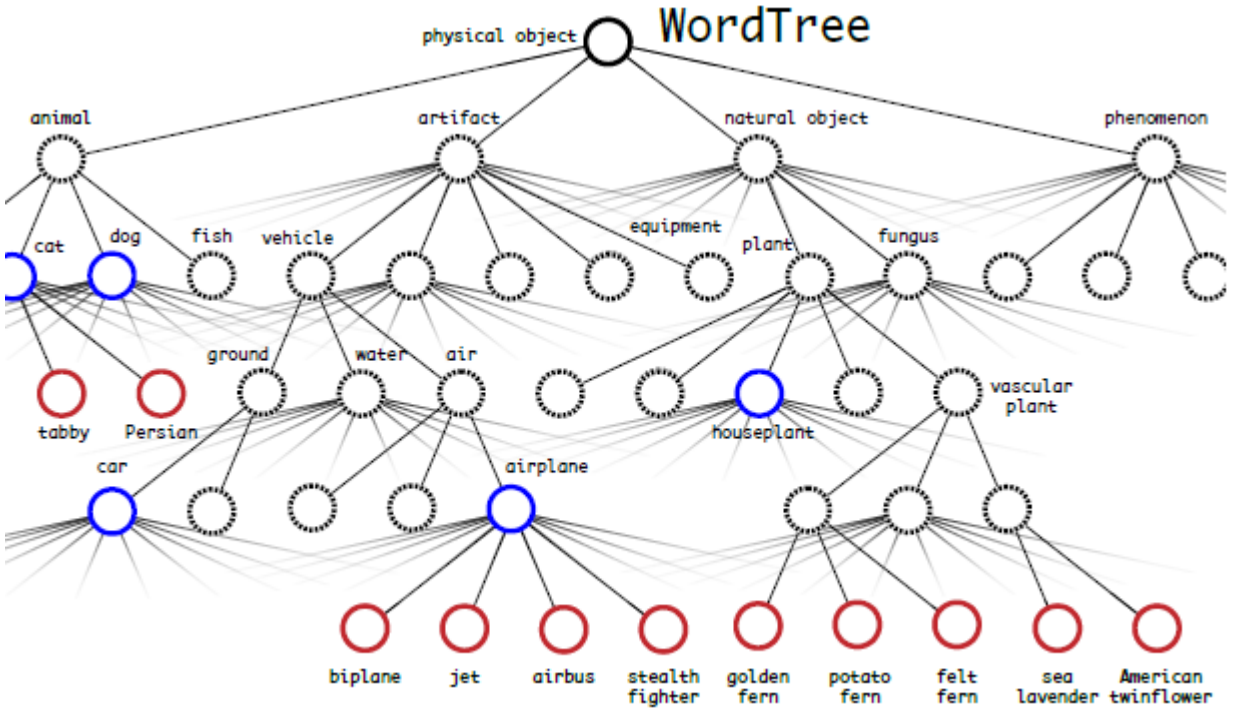


Stronger

- Dataset Combination with WordTree
  - Example of using WordTree to combine the labels from ImageNet and COCO.
    - COCO – general concepts(higher nodes)
    - ImageNet – specific concepts(lower nodes and leaves)







Stronger

- Joint Classification and Detection
  - New dataset created using the COCO detection dataset and the top 9000 classes from the full ImageNet release. ImageNet detection challenge dataset is also added for evaluation
  - Using YOLOv2 but only 3 priors instead of 5 to limit output size
  - The network sees a detection image, backpropagate loss as normal. For classification loss, only backpropagate loss at or above the corresponding level and only backpropagate classification loss
    - Simply find the b-box that predicts the highest probability for that class and compute the loss on just its predicted tree

Stronger

- Joint Classification and Detection
  - Evaluating YOLO9000 on the ImageNet detection task
    - ImageNet only share 44 object categories with COCO
    - 19.7mAP overall with 16.0mAP on the disjoint 156 object classes that it has never seen
  - YOLO9000 learns new species of animals well but struggles with learning categories like clothing and equipment

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

Conclusion

- YOLOv2 is fast and accurate
- YOLO9000 is a strong step towards closing the dataset size gap between detection and classification
- Dataset combination using hierarchical classification would be useful in the classification and segmentation domains