DLCV HW2

TAs (劉致廷, 吳致緯)
ntudlcv2019@gmail.com

Outline

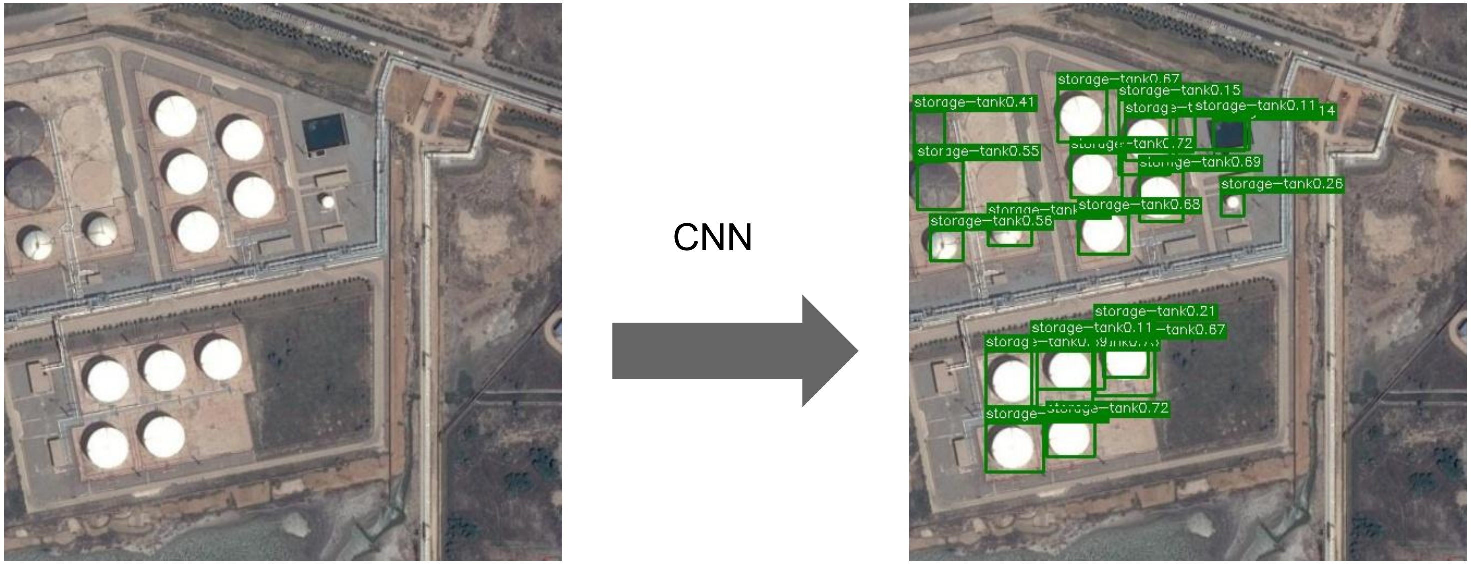
- Object Detection in Aerial Images
  - Task Definition
  - Object Detection Network (Yolo v1)
- Assignment
  - TODO
  - Implementation Details
  - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Outline

- Object Detection in Aerial Images
    - Task Definition
    - Object Detection Network (Yolo v1)
- Assignment
    - TODO
    - Implementation Details
    - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Object Detection in Aerial Images
Task Definition

A simplified real competetion! → Detect 16 kinds of objects in aerial images.
https://captain-whu.github.io/DOAI2019/dataset.html



CNN

storage−tank0.67
storage−tank0.15
storage−tank0.41
storage−tank0.11
storage−tank0.55
storage−tank0.69
storage−tank0.26
storage−tank0.68
storage−tank0.56
storage−tank0.21
storage−tank0.11
−tank0.67
storage−tank0.72

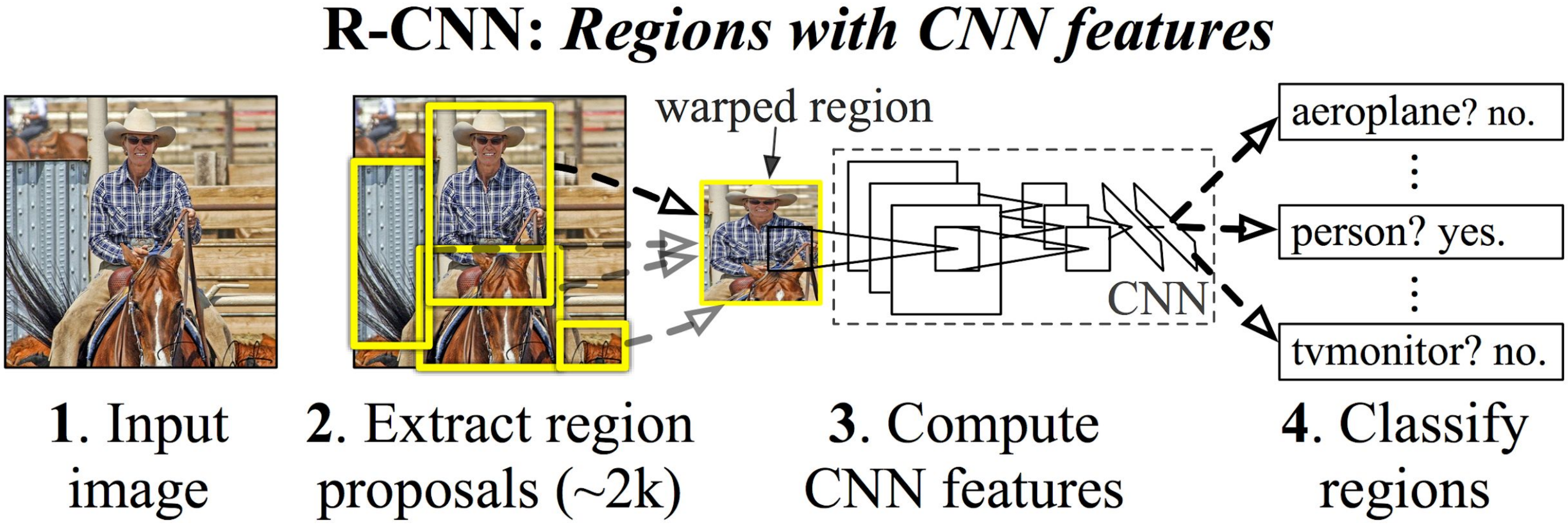
Horizontal bbox coordinates &
class & score

Outline

- Object Detection in Aerial Images
    - Task Definition
    - Object Detection Network (Yolo v1)
- Assignment
    - TODO
    - Implementation Details (Data, Loss, IoU, NMS)
    - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Object Detection Network -- Yolo V1

Region Proposal → Slow


R-CNN: *Regions with CNN features*
warped region
CNN
aeroplane? no.
person? yes.
tvmonitor? no.
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

slides are modified from https://pjreddie.com/darknet/yolo/

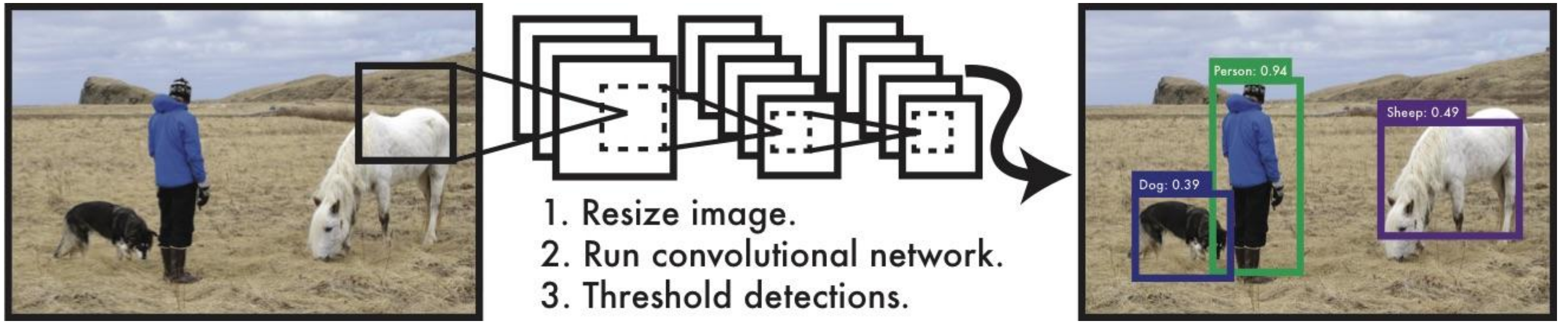
Accurate object detection is slow!

	Pascal 2007 mAP	**Speed**	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	69.0	45 FPS	22 ms/img

https://arxiv.org/pdf/1506.02640.pdf

With YOLO, you only look once at an image to perform detection

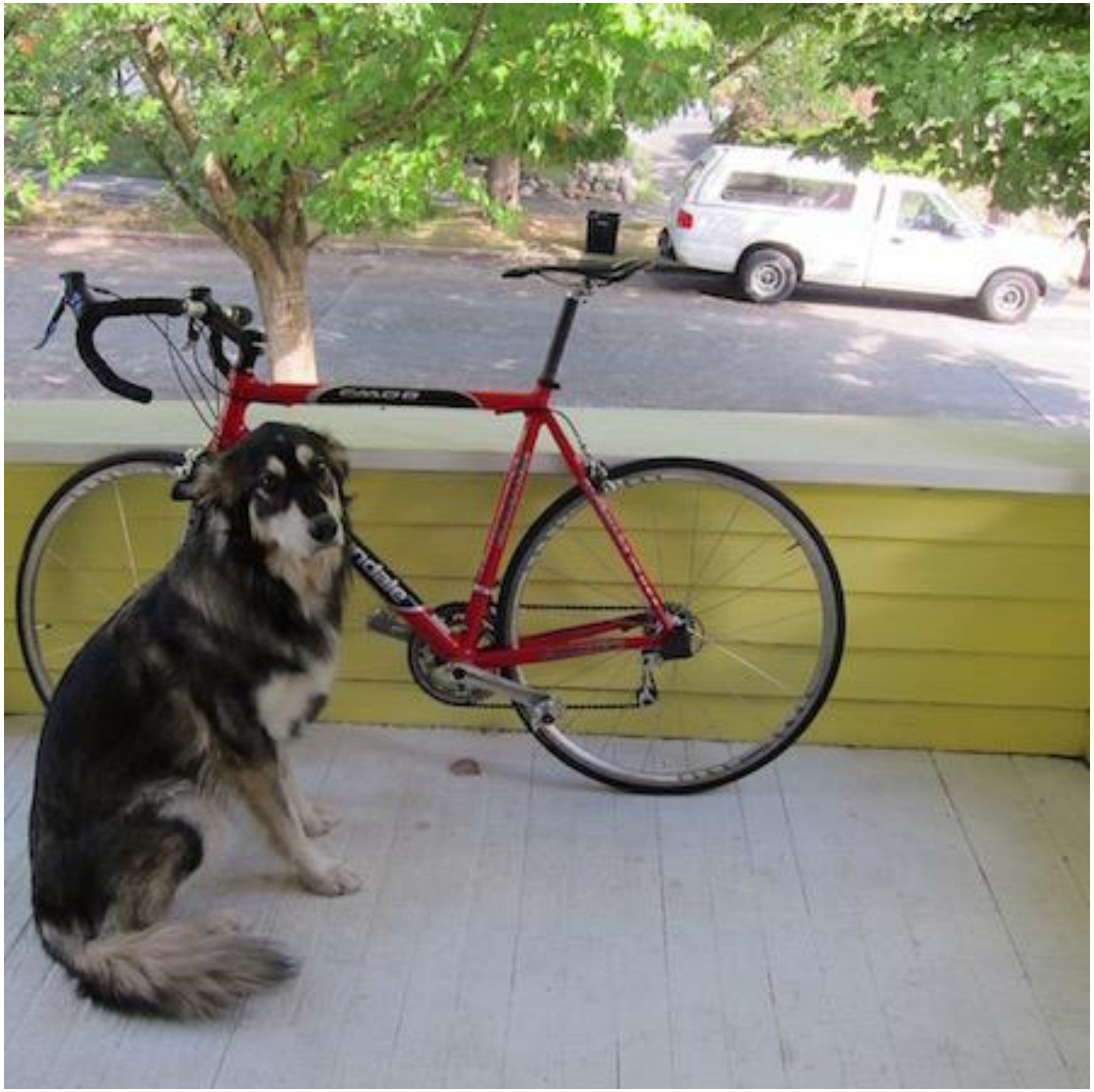
YOLO: ***You Only Look Once***



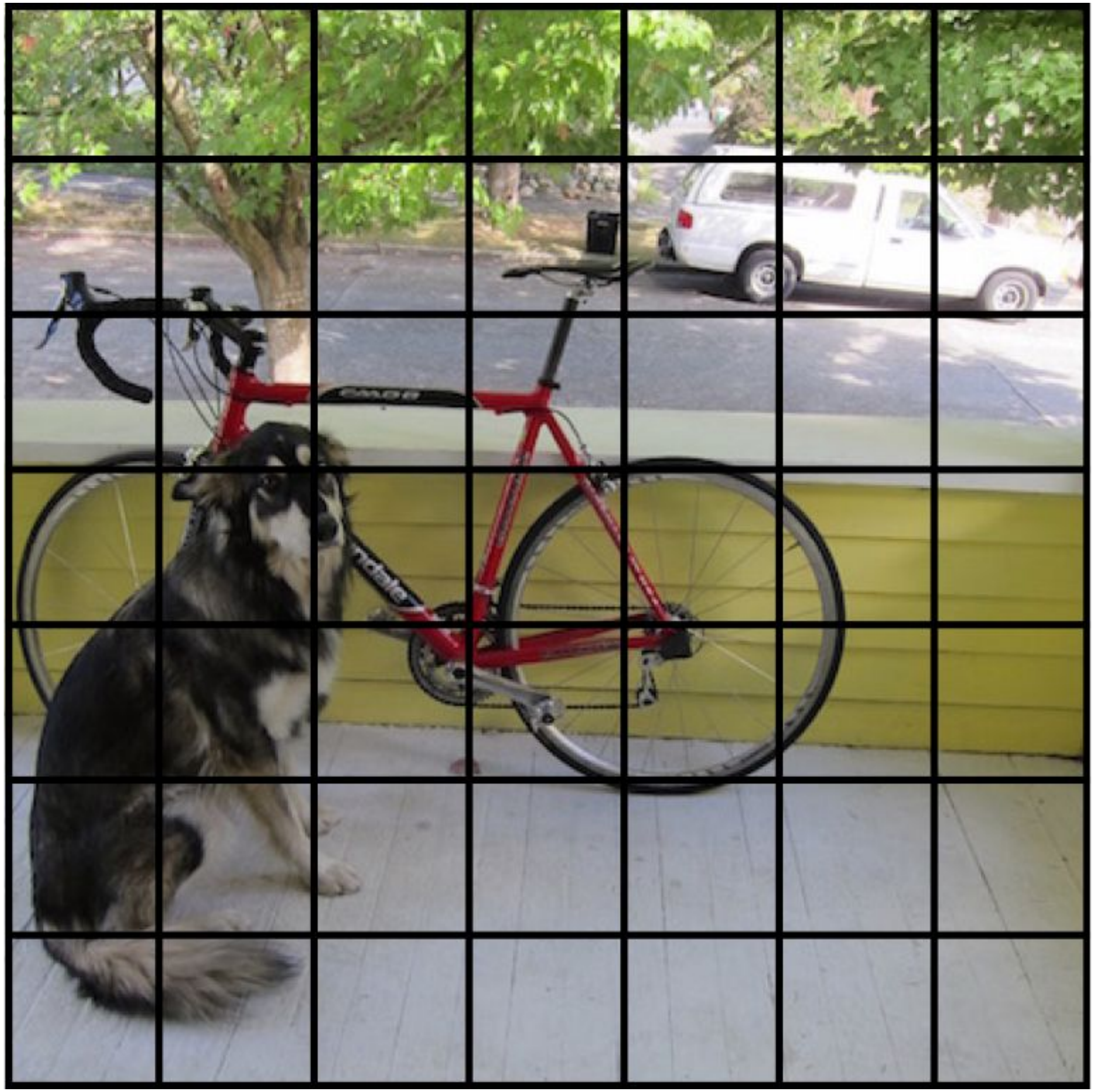
1. Resize image.
2. Run convolutional network.
3. Threshold detections.

Person: 0.94
Sheep: 0.49
Dog: 0.39

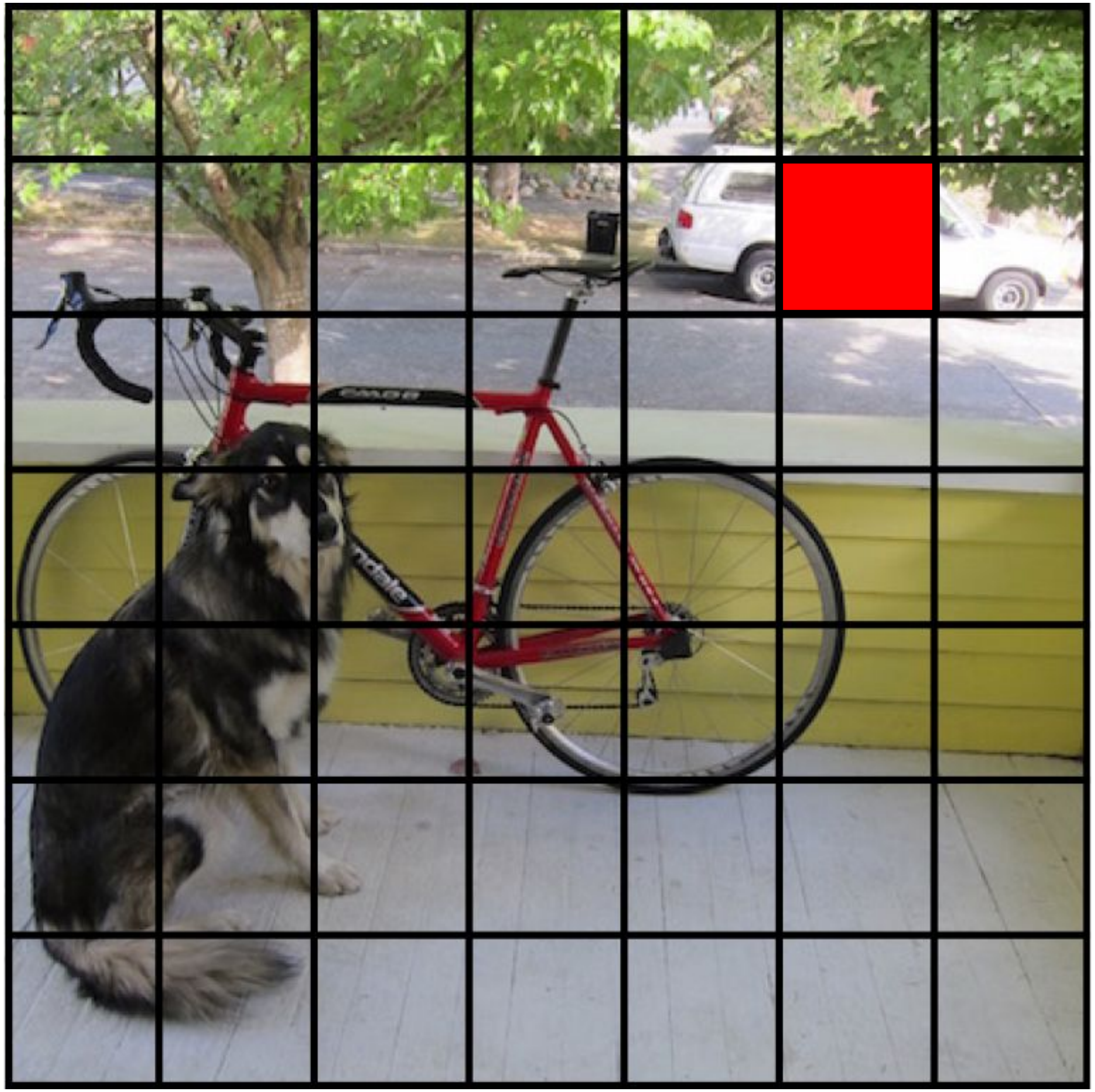
Testing an image with Yolo v1



We split the image into a grid



Each cell predicts boxes and confidences: P(Object)

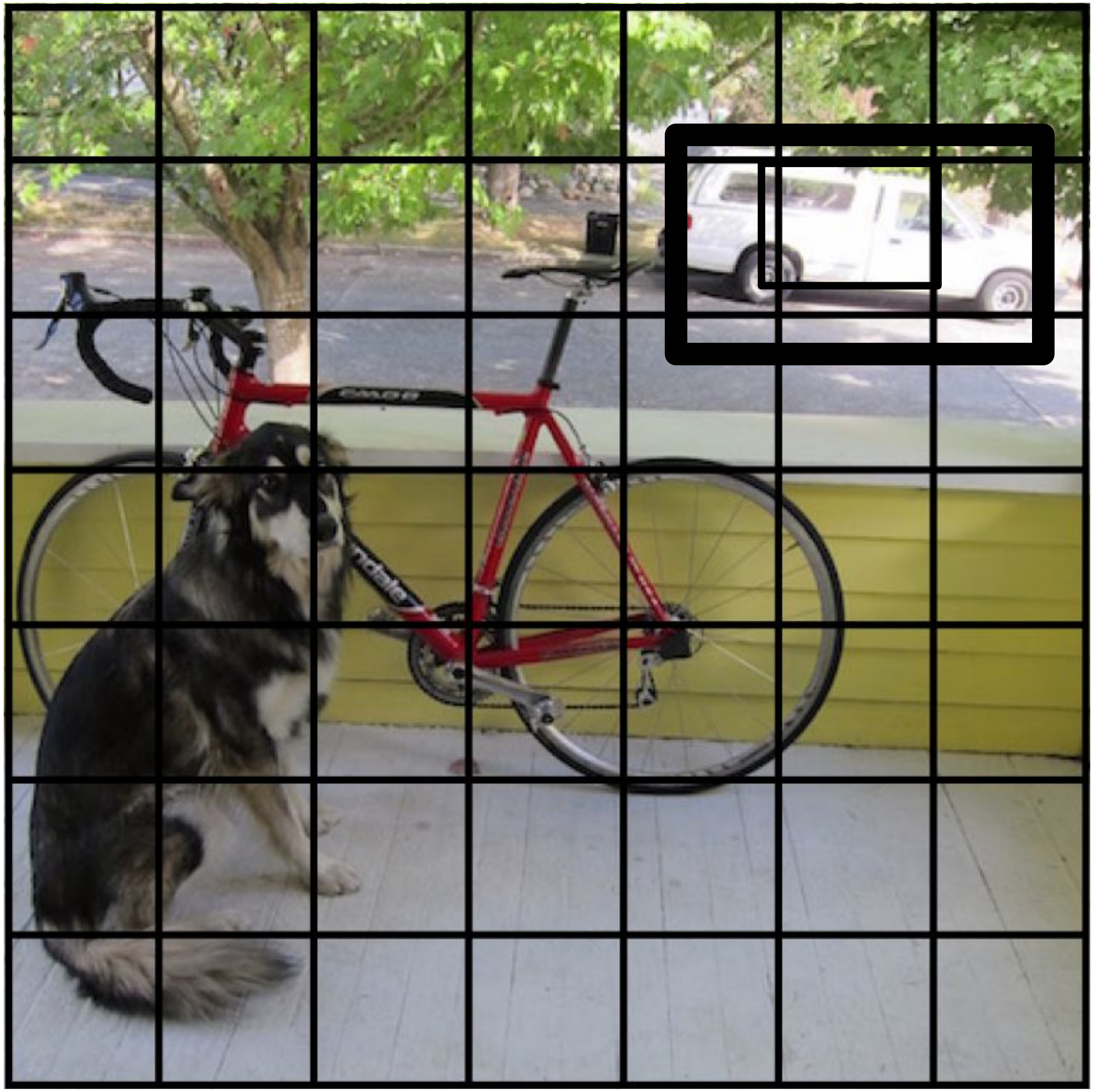


Each cell predicts boxes and confidences: P(Object)

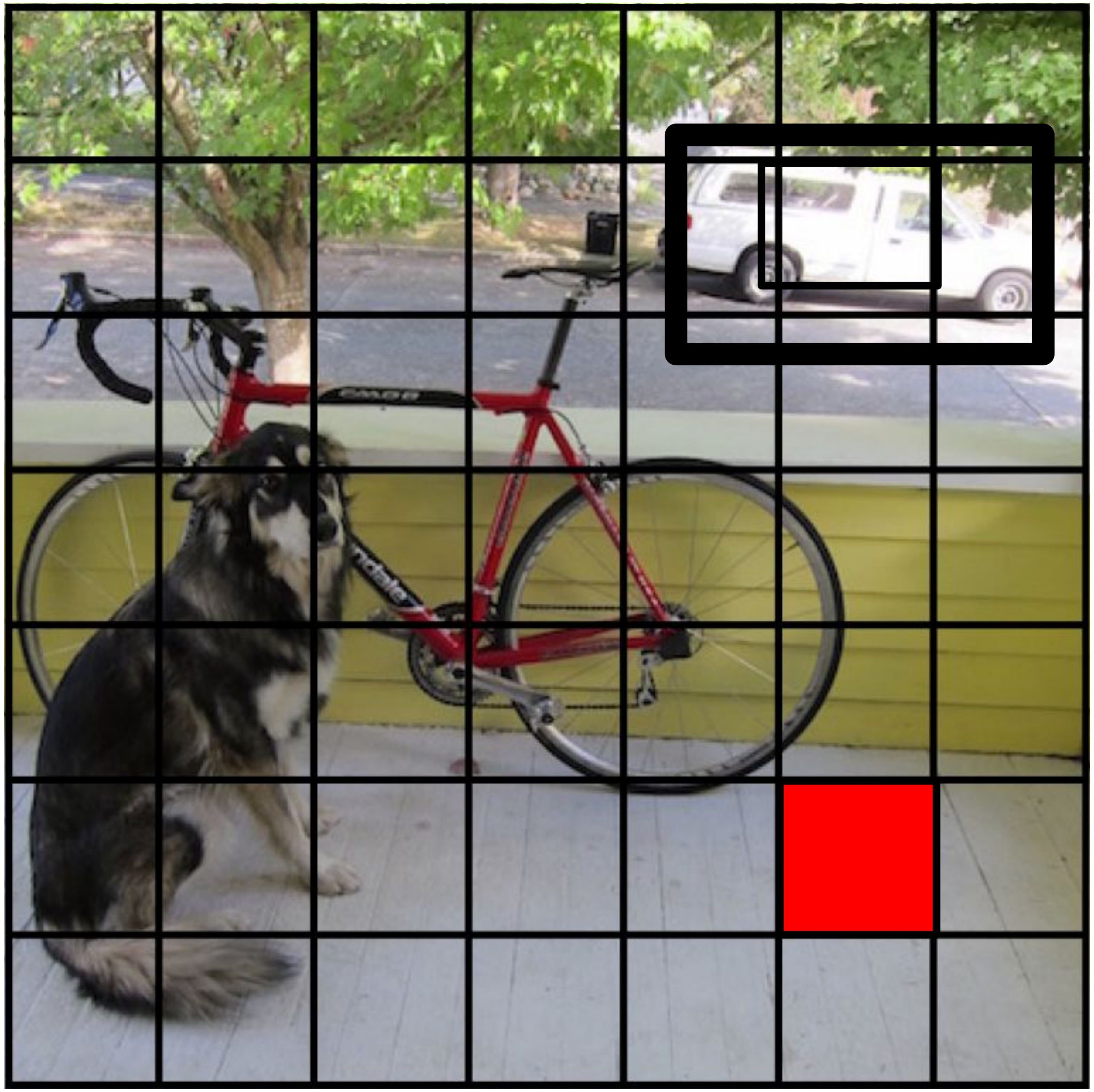
Predict **two** boxes !

No Anchor!

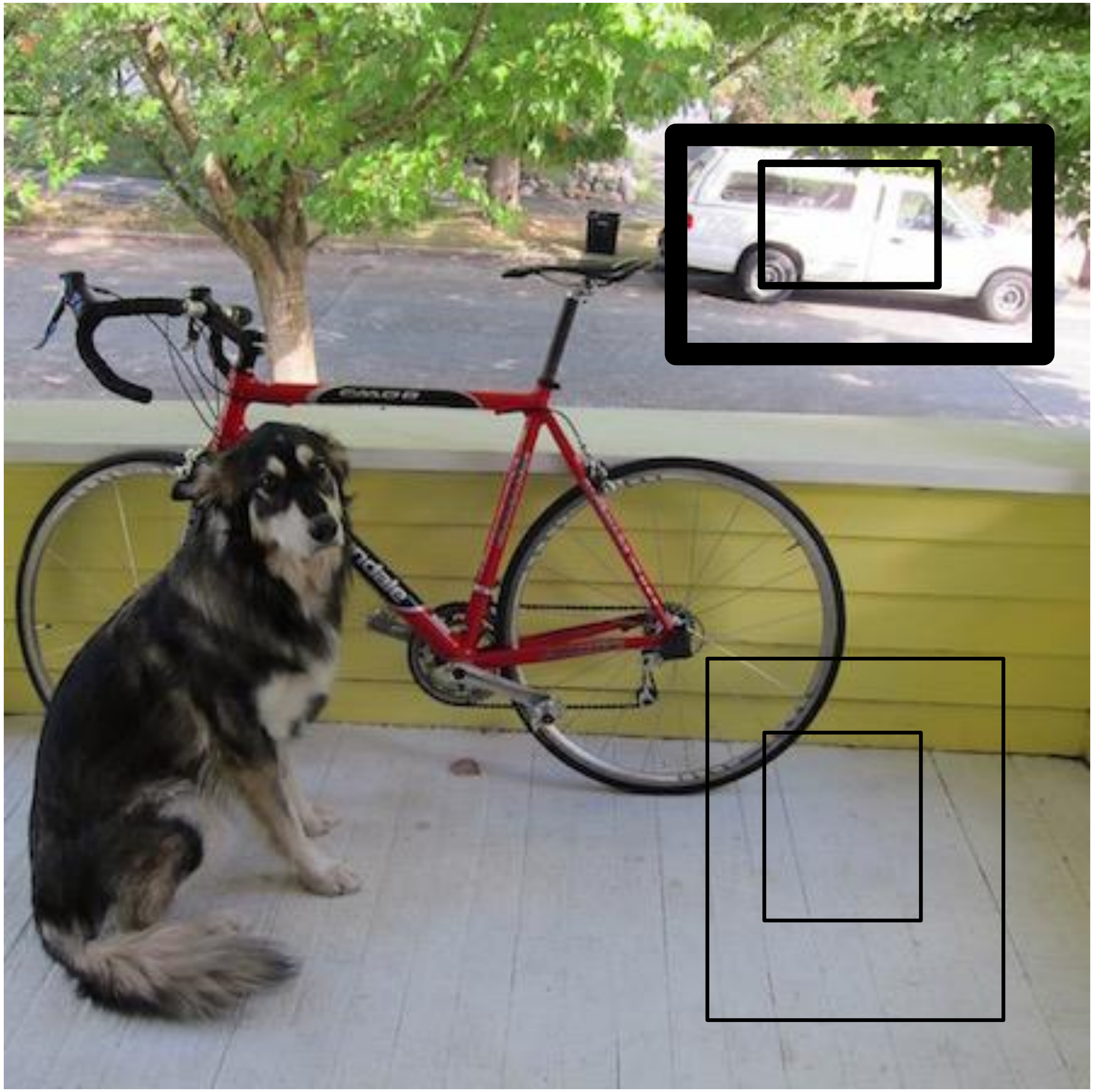
Each cell predicts boxes and confidences: P(Object)



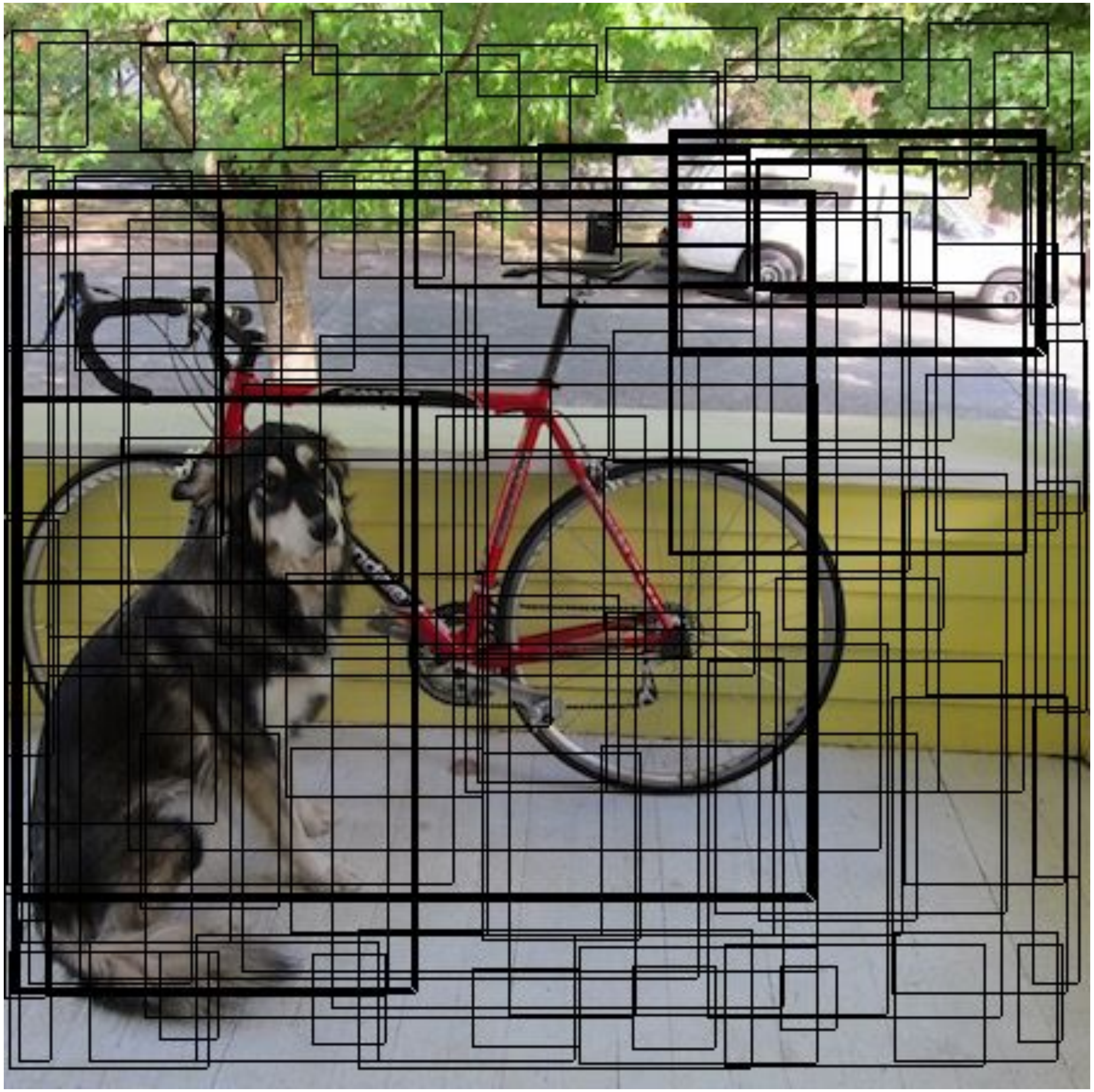
Each cell predicts boxes and confidences: P(Object)



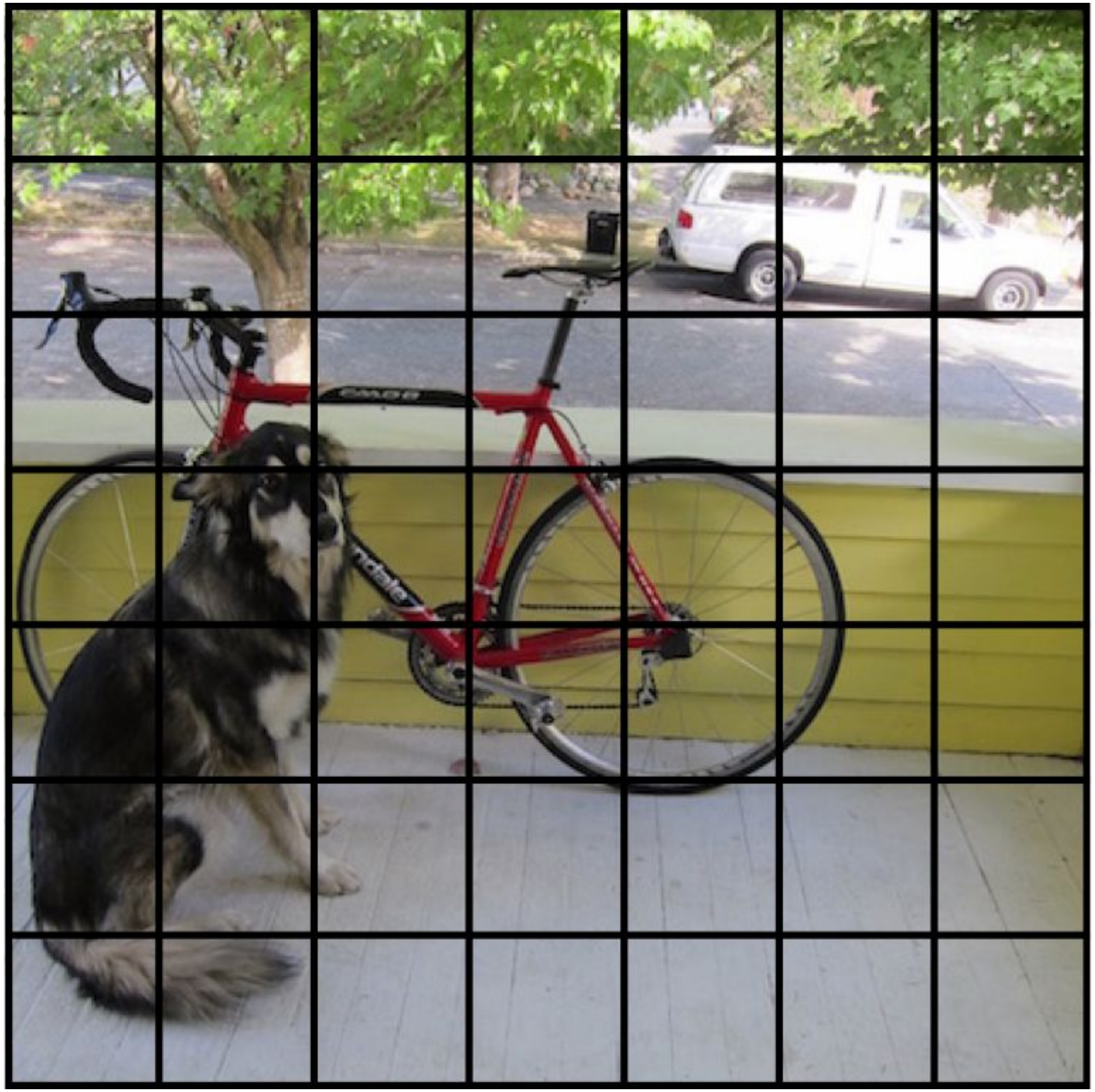
Each cell predicts boxes and confidences: P(Object)



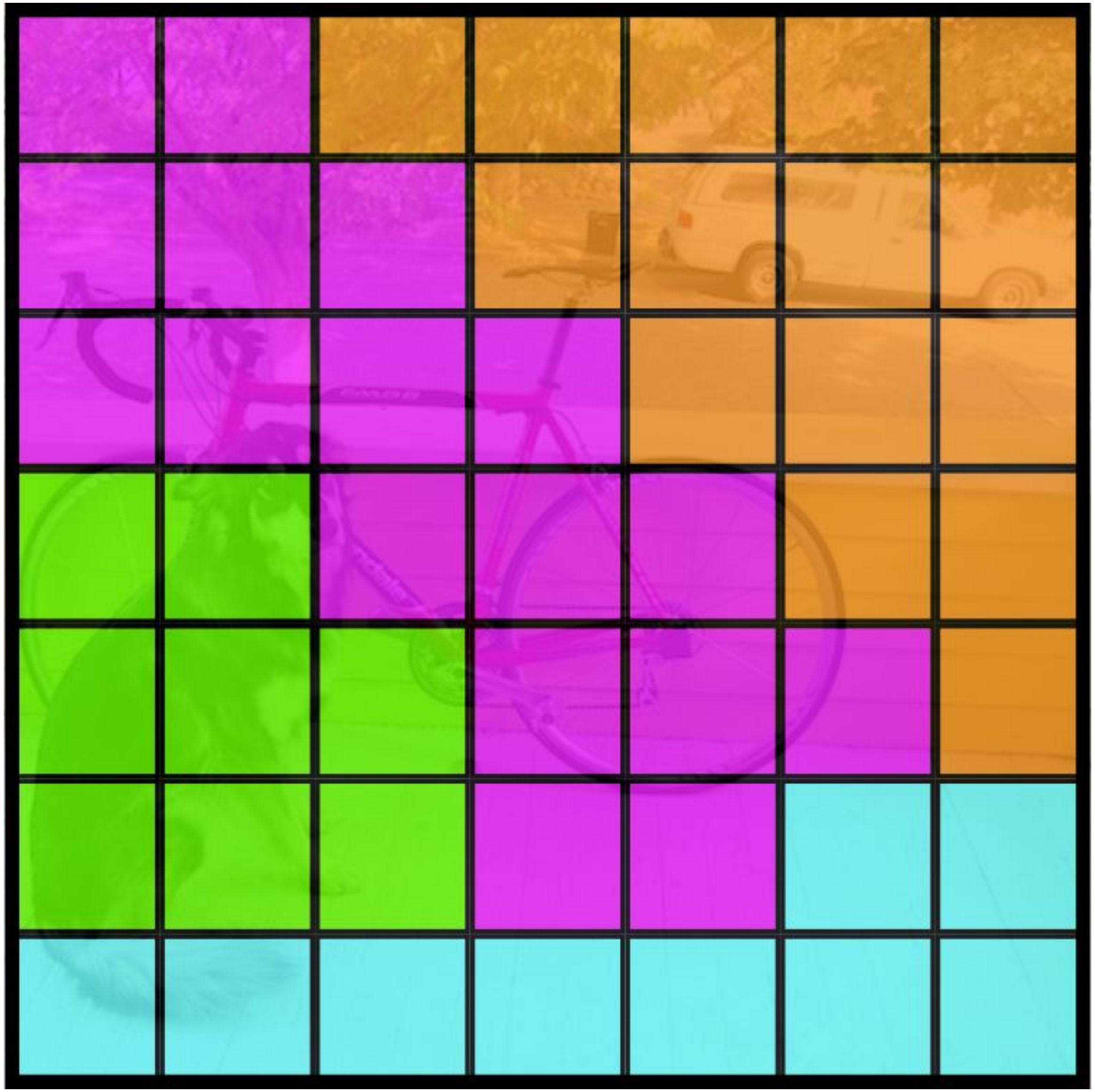
Each cell predicts boxes and confidences: P(Object)



Each cell also predicts a class probability.



Each cell also predicts a class probability.



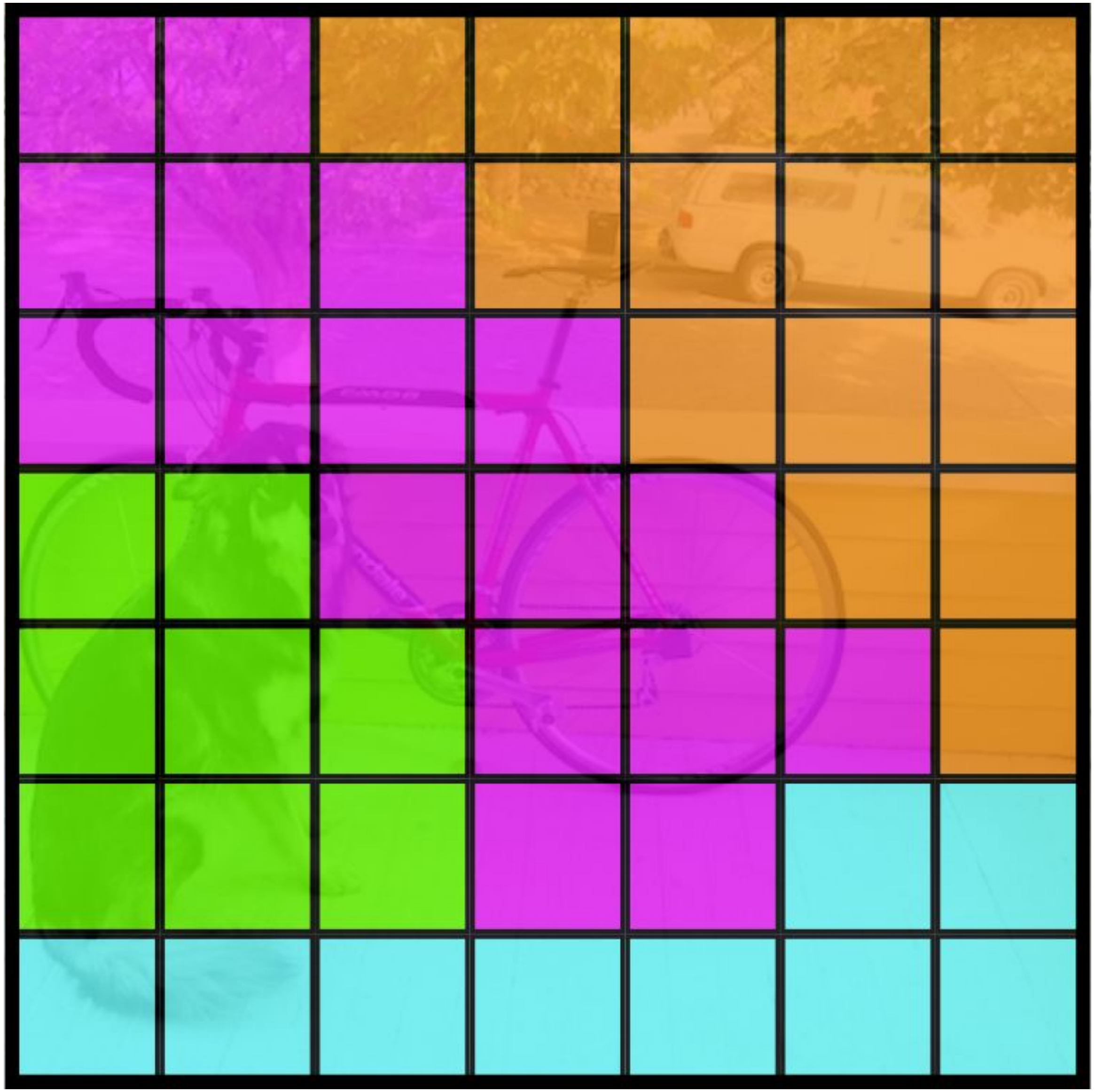
Bicycle

Car

Dog

Dining Table

Conditioned on object: P(Car | Object)



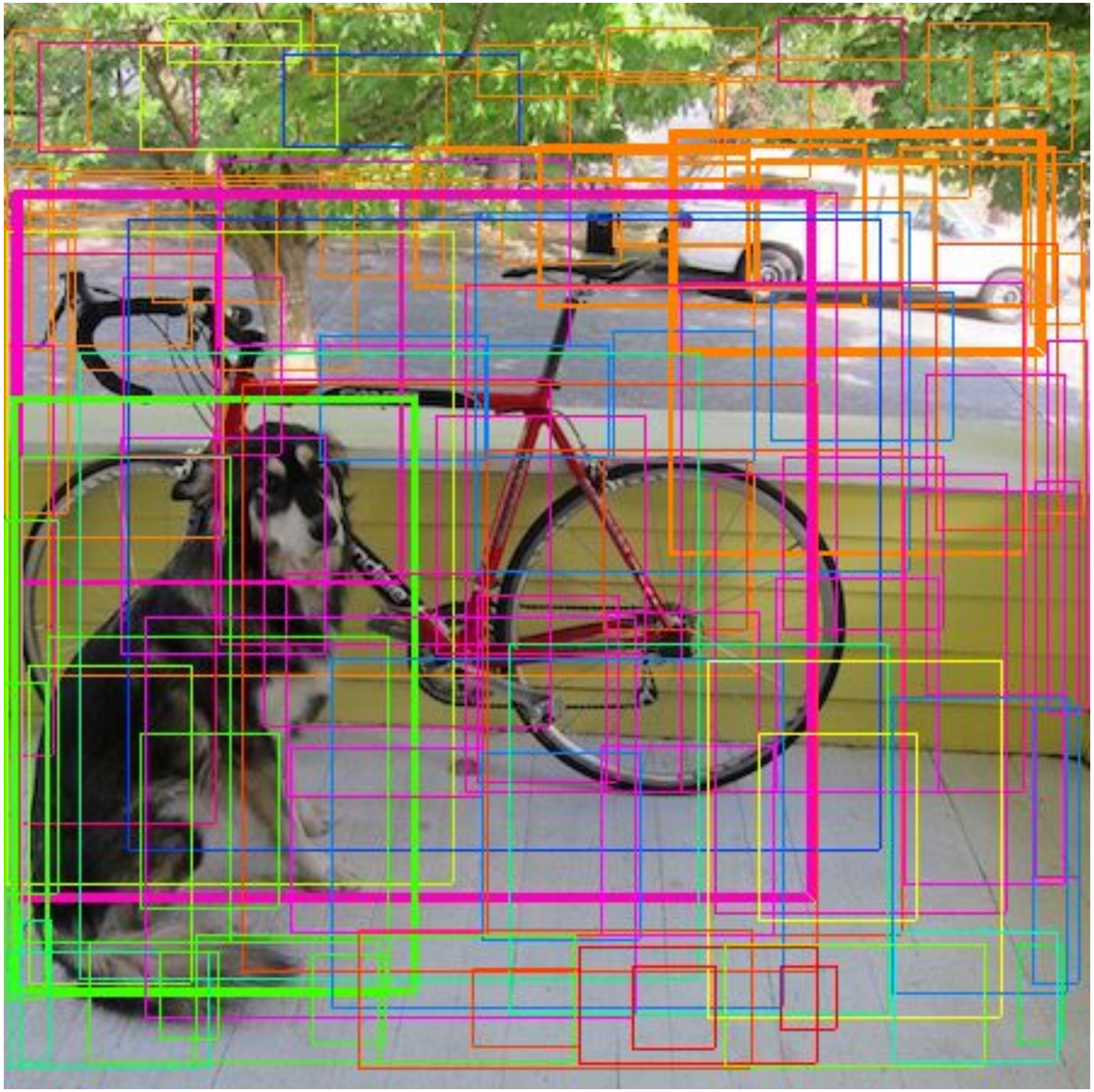
Bicycle

Car

Dog

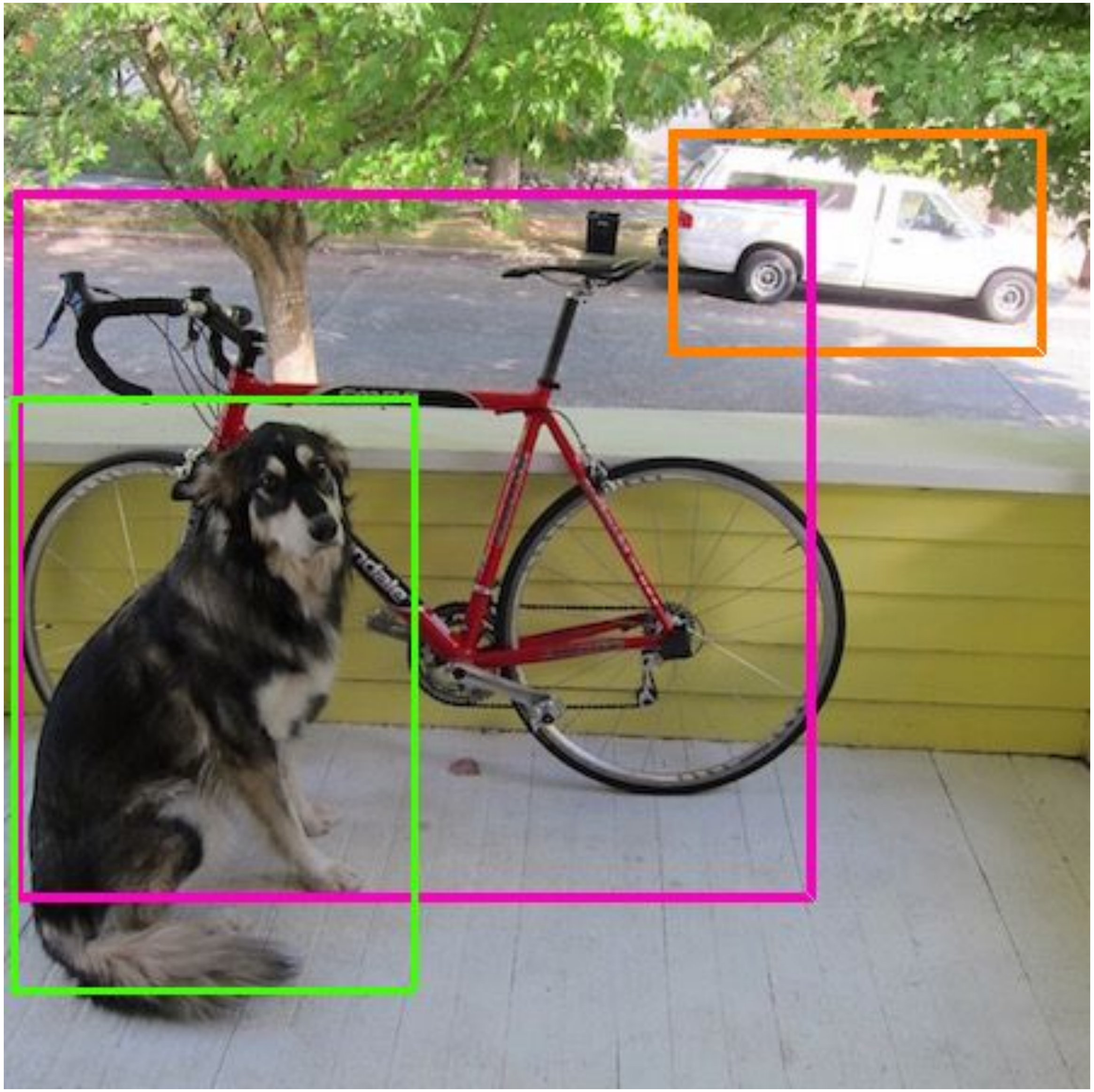
Dining
Table

Then we combine the box and class predictions.



NMS (Non-Maximum Suppression)

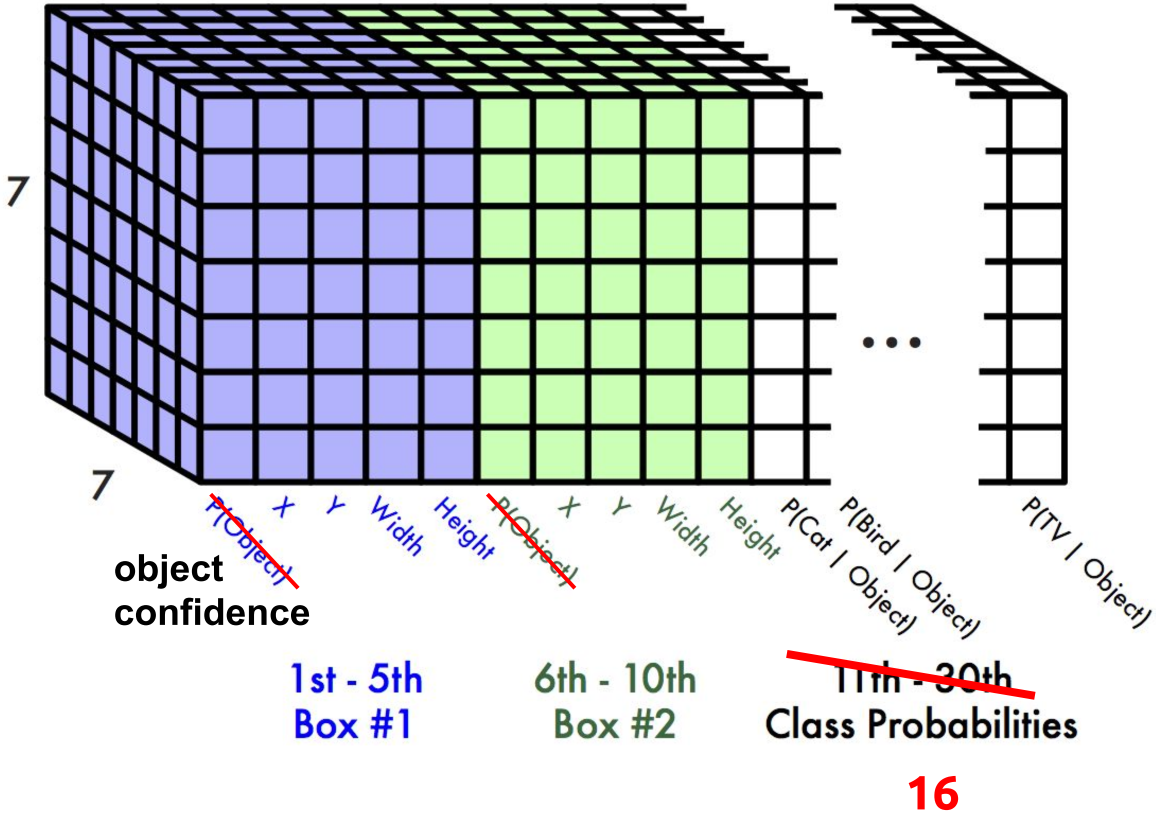
Finally we do NMS and threshold detections



This parameterization fixes the output size

Each cell predicts:

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

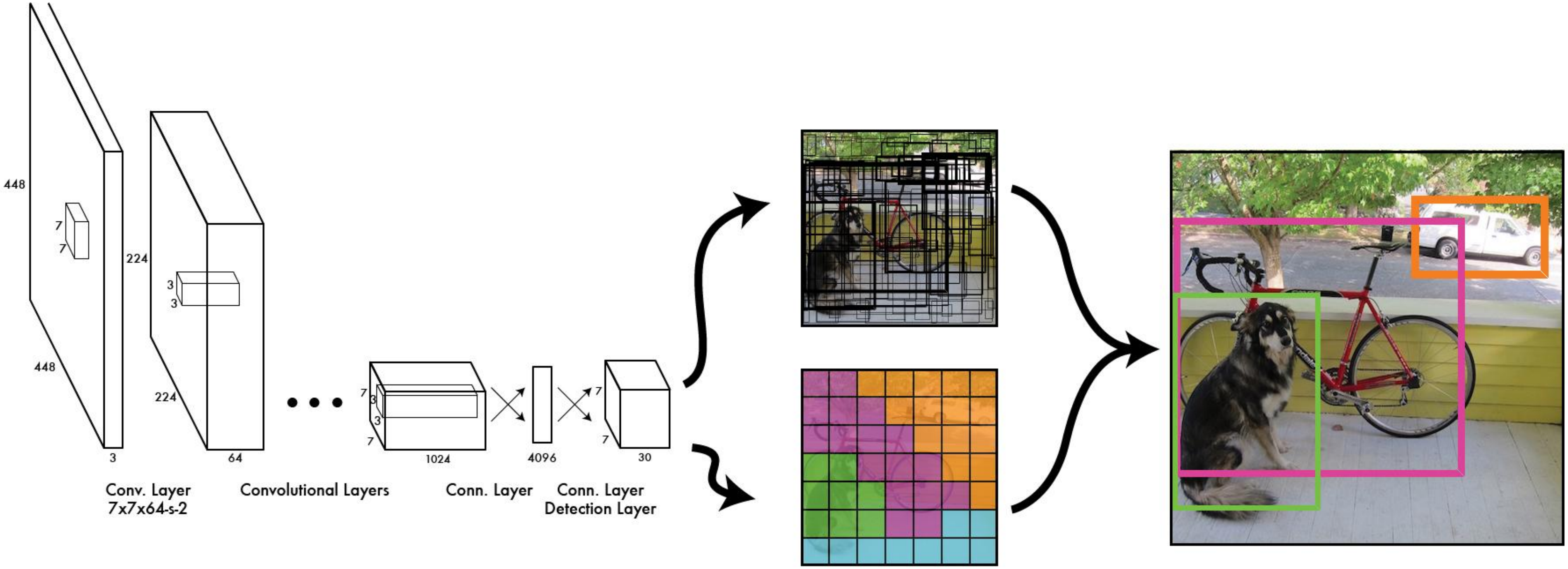

7
7
~~P(Object)~~
X
Y
Width
Height
~~P(Object)~~
X
Y
Width
Height
P(Cat | Object)
P(Bird | Object)
...
P(TV | Object)
object confidence
1st - 5th
Box #1
6th - 10th
Box #2
~~11th - 30th~~
Class Probabilities
16

For this dataset:

- 7x7 grid
- 2 bounding boxes / cell
- **16** classes

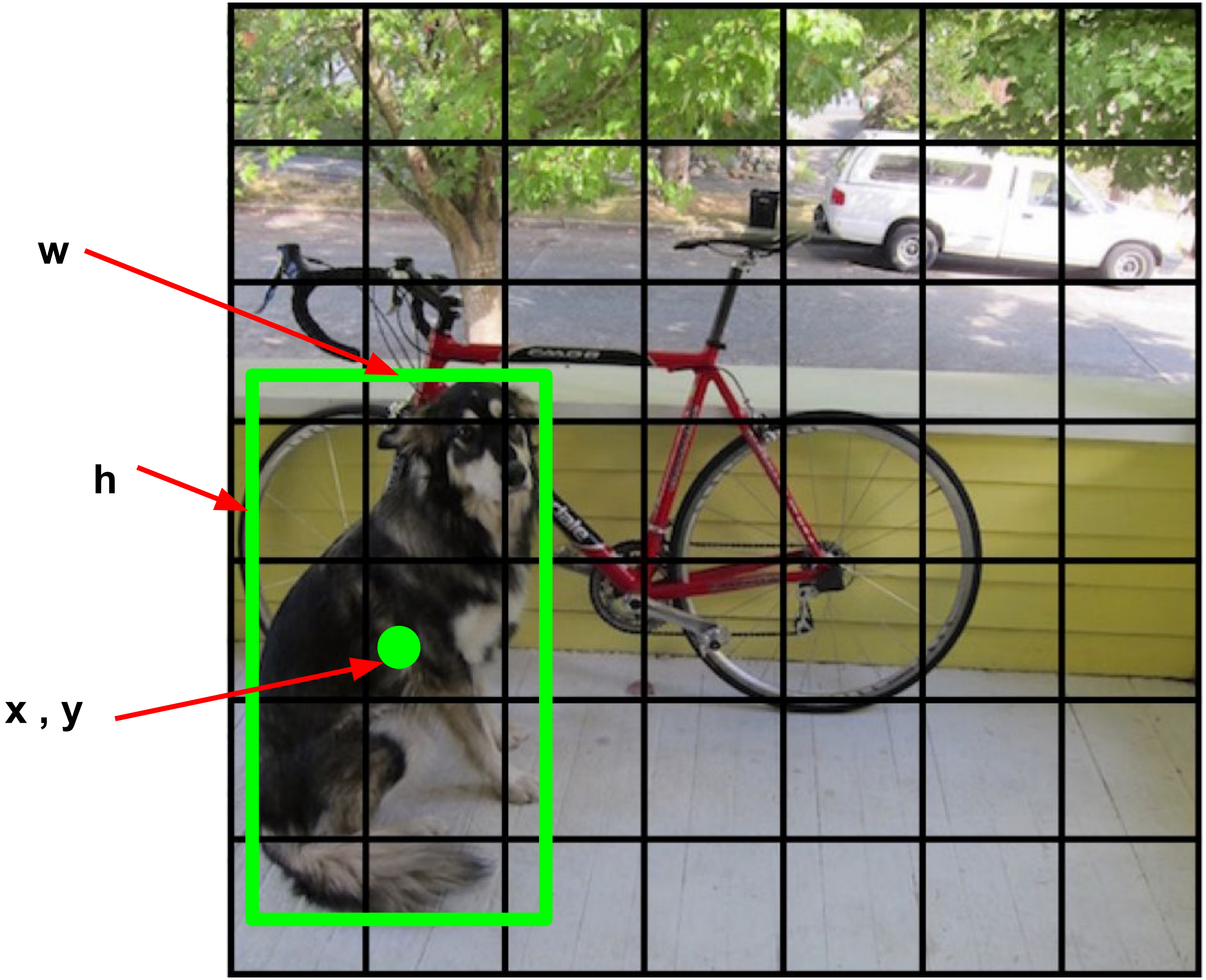
7 x 7 x (2 x 5 + 16) = 7 x 7 x 26 tensor = **1274 outputs**

Thus we can train one neural network to be a whole detection pipeline

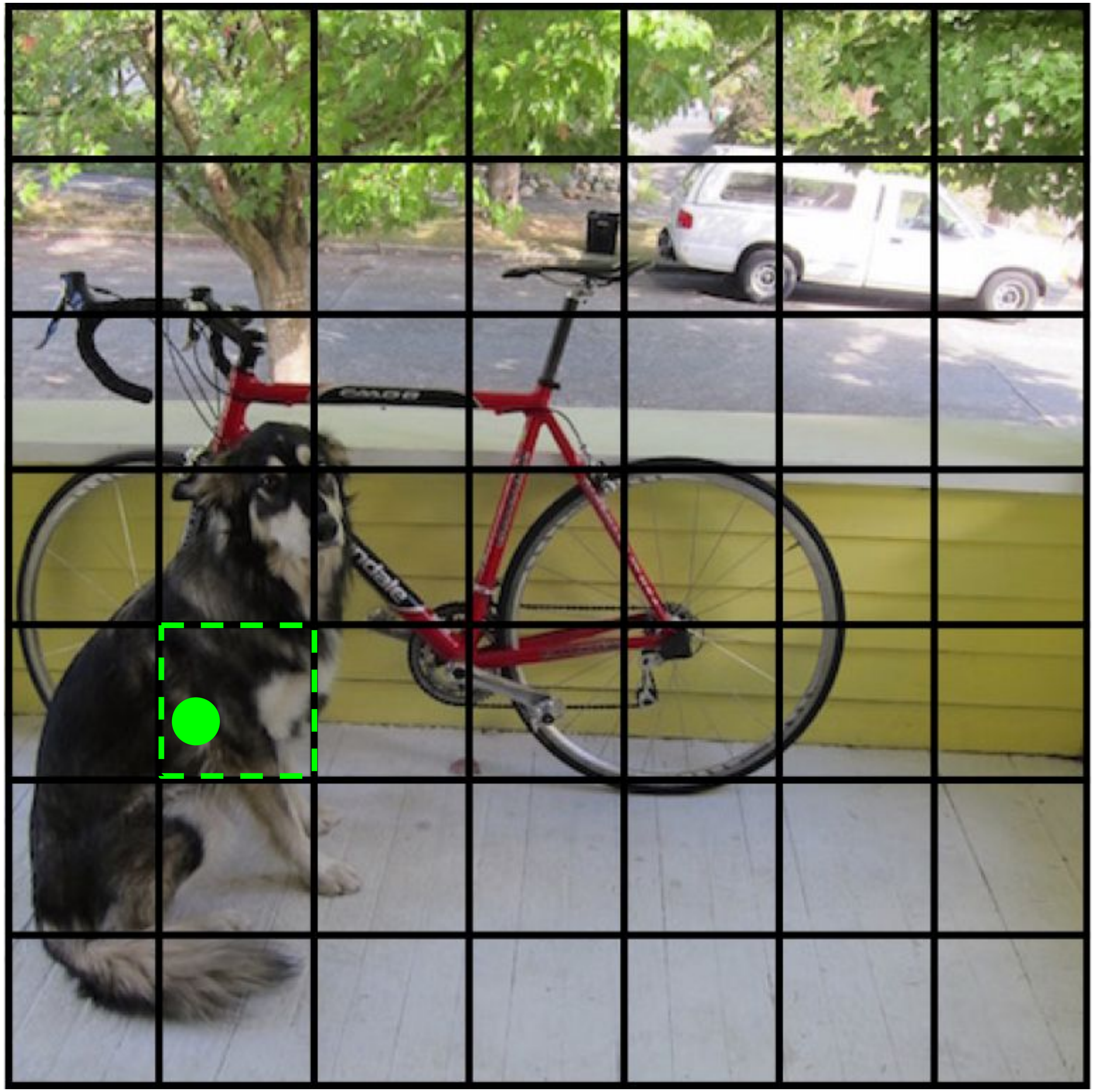

448
7
7
224
3
3
448
224
3
64
7
3
3
7
1024
4096
7
7
30
Conv. Layer
7x7x64-s-2
Convolutional Layers
Conn. Layer
Conn. Layer
Detection Layer

For convenience, we will change the backbone in our homework baseline model.

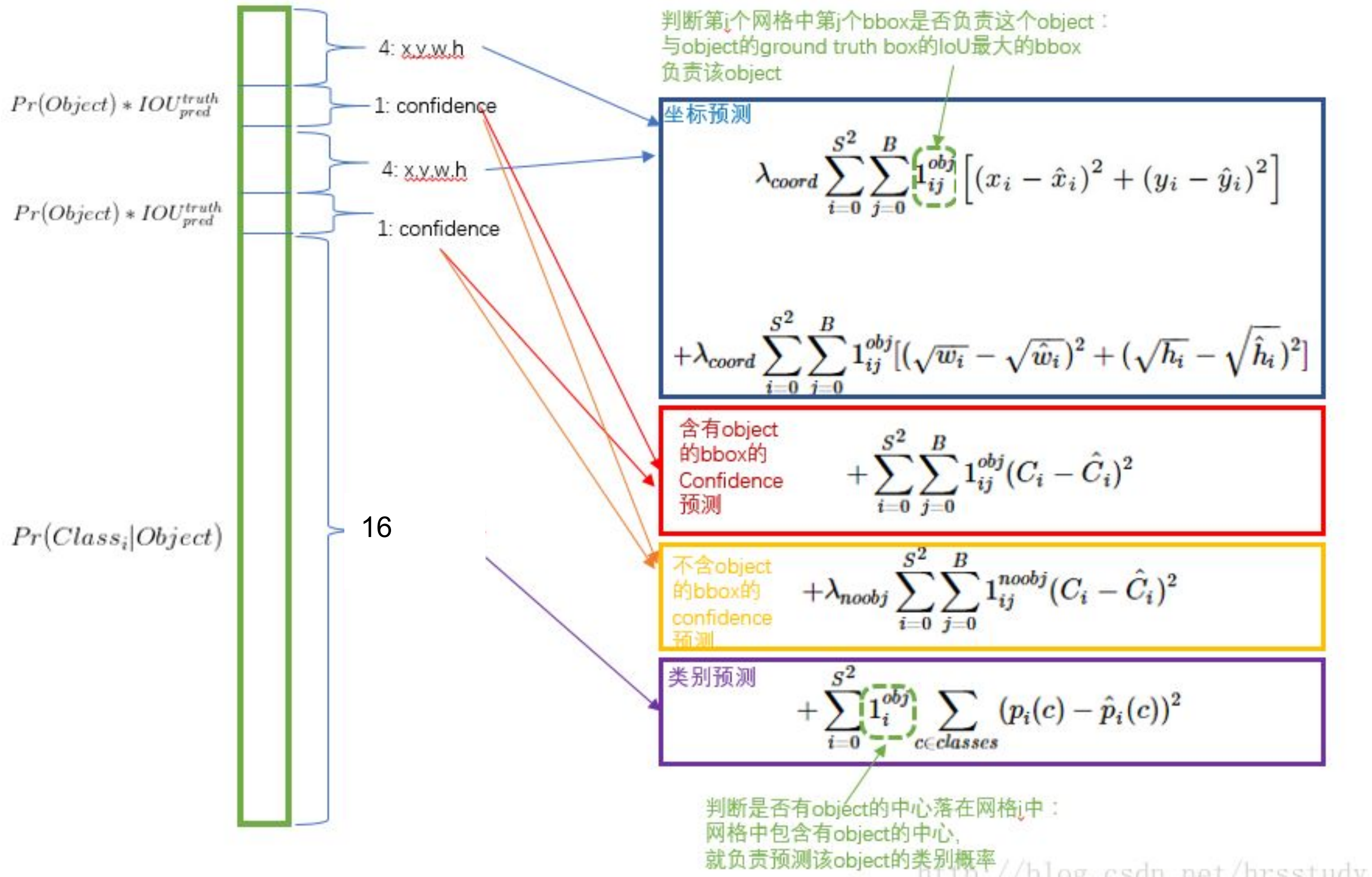
During **training**, match example to the correct cell



During **training**, match example to the correct cell



We need loss to guide the prediction


$Pr(Object) * IOU^{truth}_{pred}$
$Pr(Object) * IOU^{truth}_{pred}$
$Pr(Class_i|Object)$
4: x,y,w,h
1: confidence
4: x,y,w,h
1: confidence
16

判断第i个网格中第j个bbox是否负责这个object：与object的ground truth box的IoU最大的bbox负责该object

坐标预测

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

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

含有object的bbox的Confidence预测

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

不含object的bbox的confidence预测

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

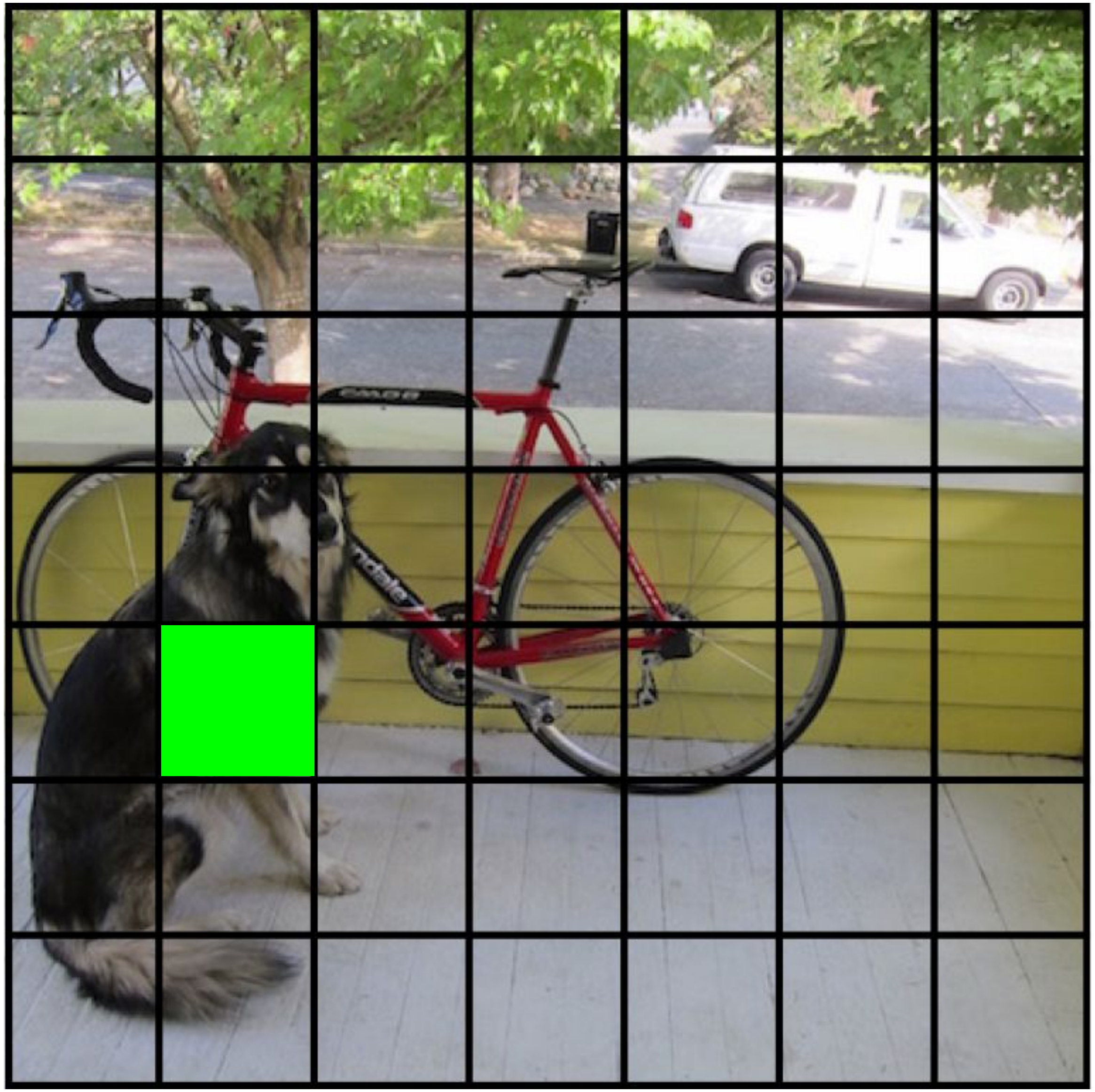
类别预测

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

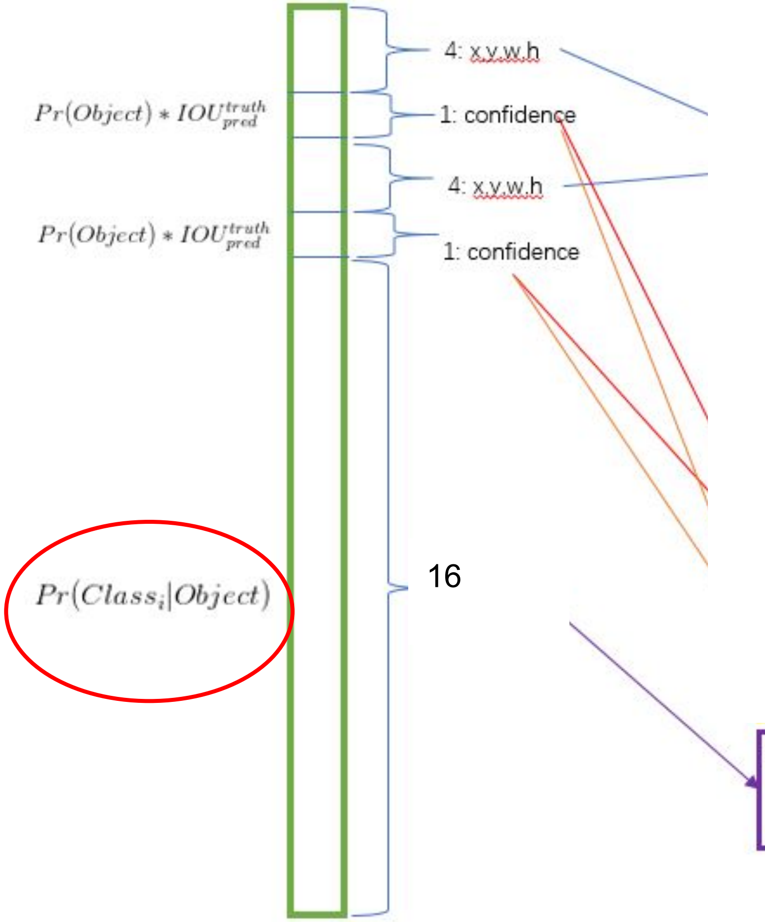
判断是否有object的中心落在网格i中：网格中包含有object的中心，就负责预测该object的类别概率

http://blog.csdn.net/hrsstudy

Adjust that cell’s class prediction



Dog = 1
Cat = 0
Bike = 0
...



4: x,y,w,h

$Pr(Object) * IOU_{pred}^{truth}$

1: confidence

4: x,y,w,h

$Pr(Object) * IOU_{pred}^{truth}$

1: confidence

$Pr(Class_i|Object)$

16

类别预测

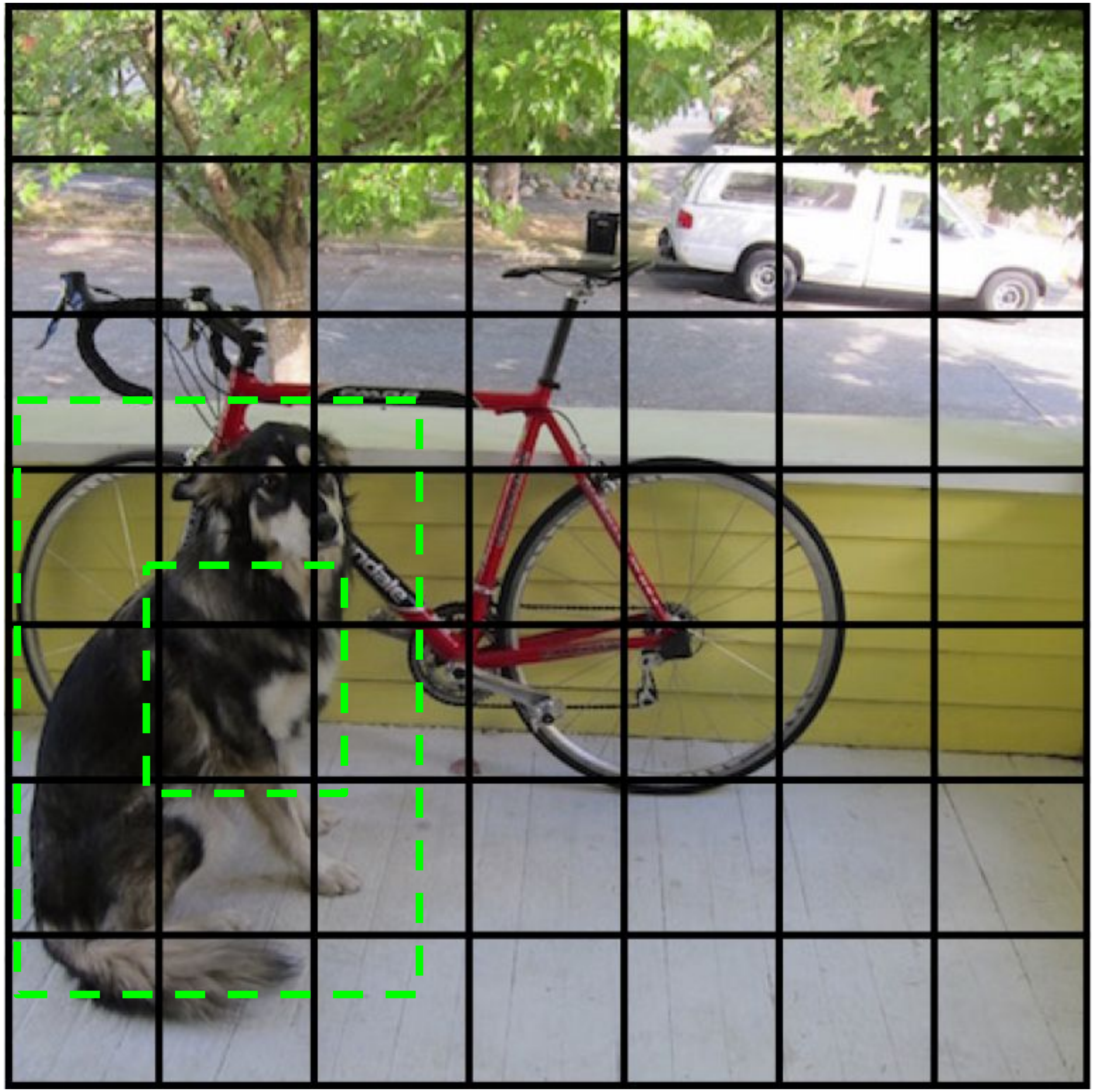
$$+\sum_{i=0}^{S^2} \boxed{1_i^{obj}} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

MSE loss

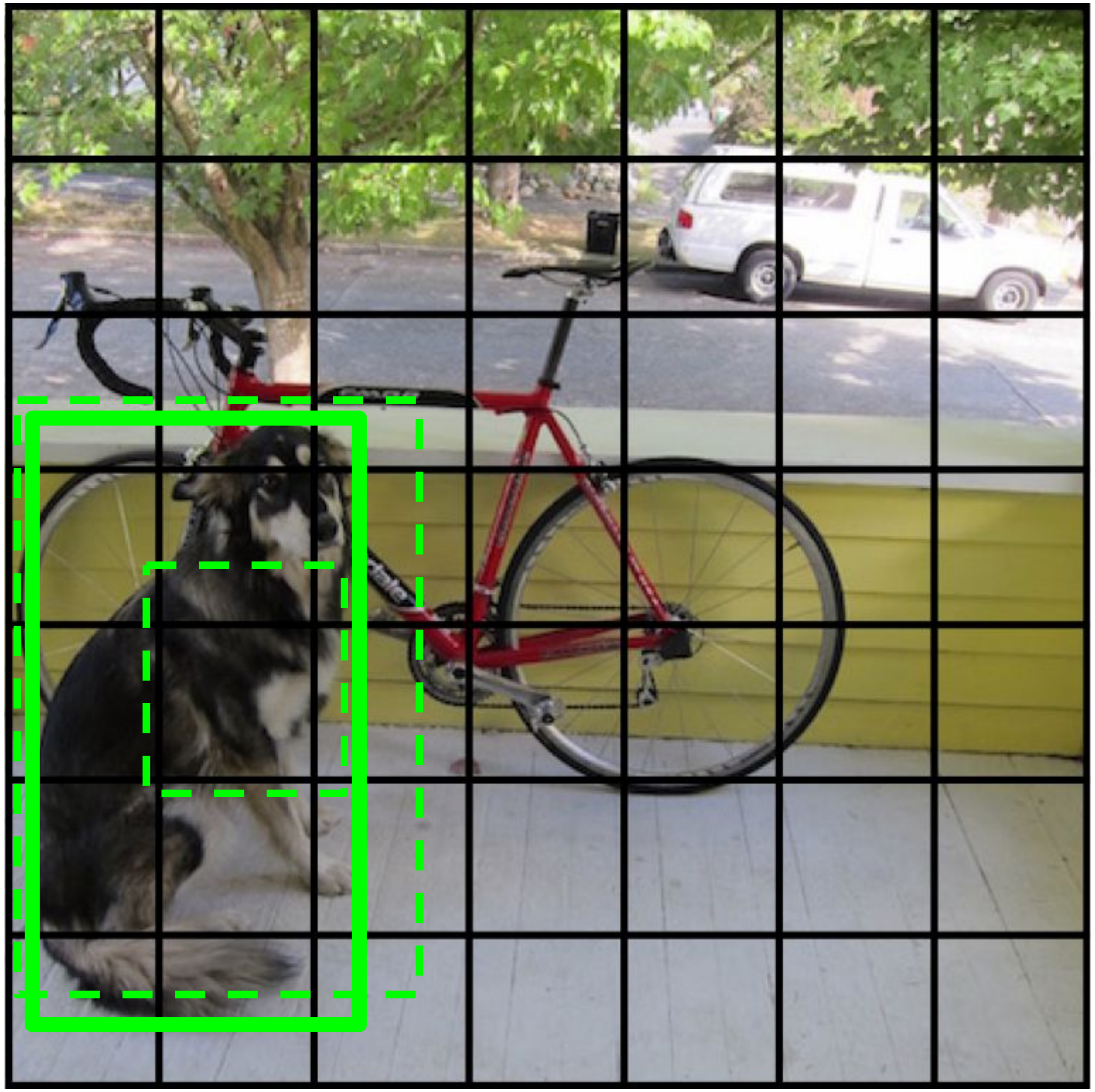
判断是否有object的中心落在网格i中：
网格中包含有object的中心，
就负责预测该object的类别概率

http://blog.csdn.net/hrsstudy

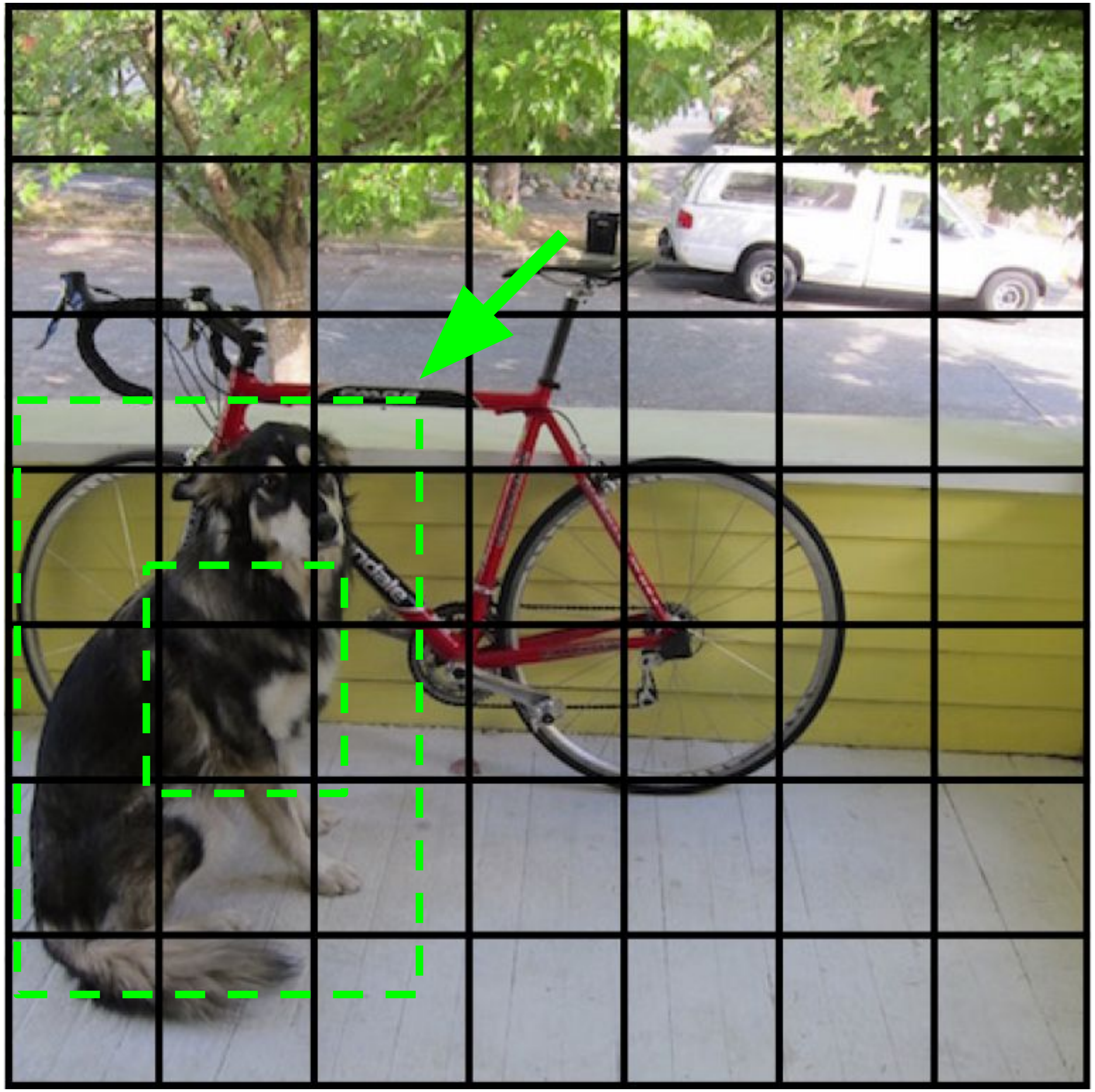
Look at that cell's predicted boxes



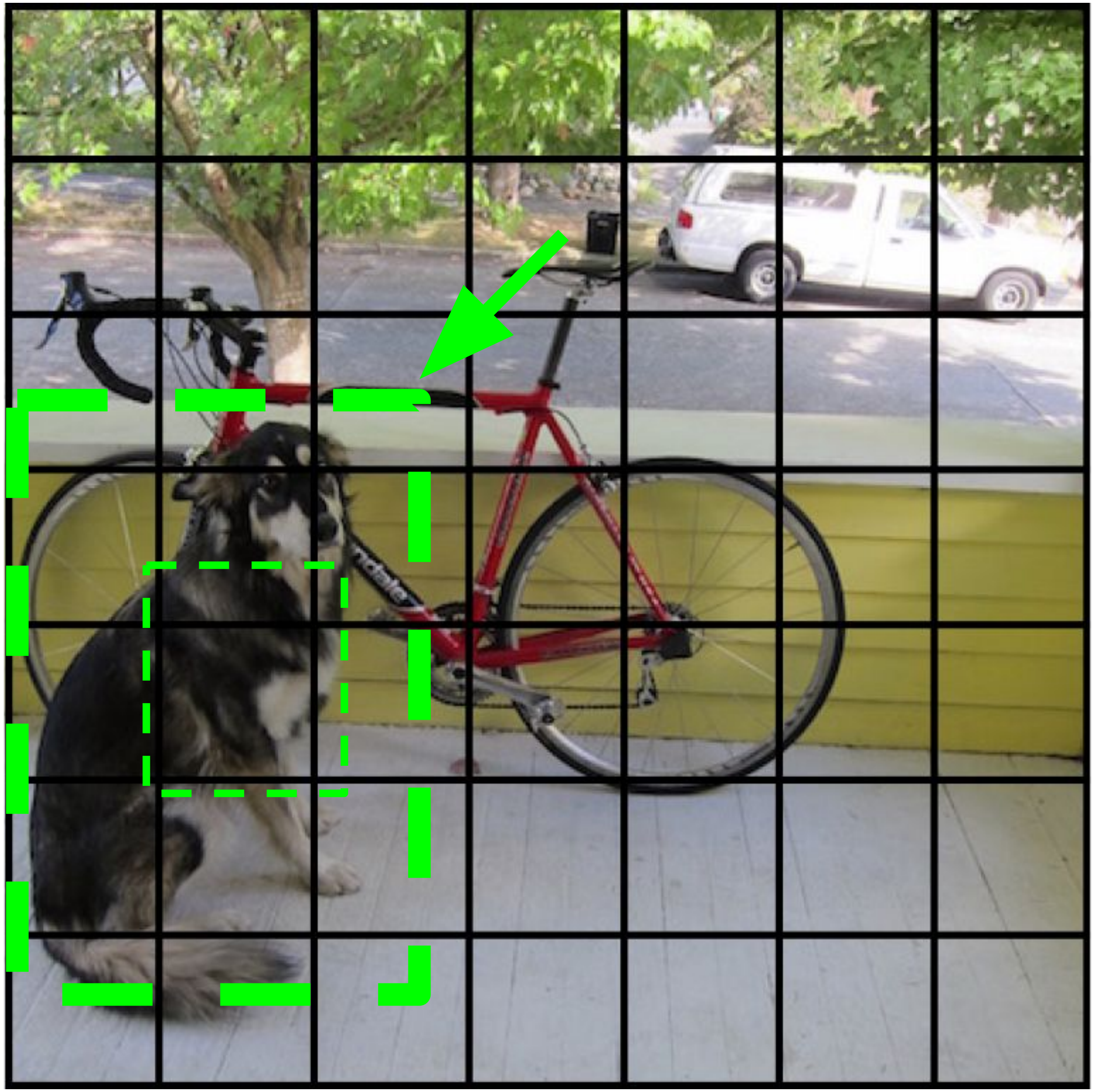
Find the best one, adjust it, increase the confidence

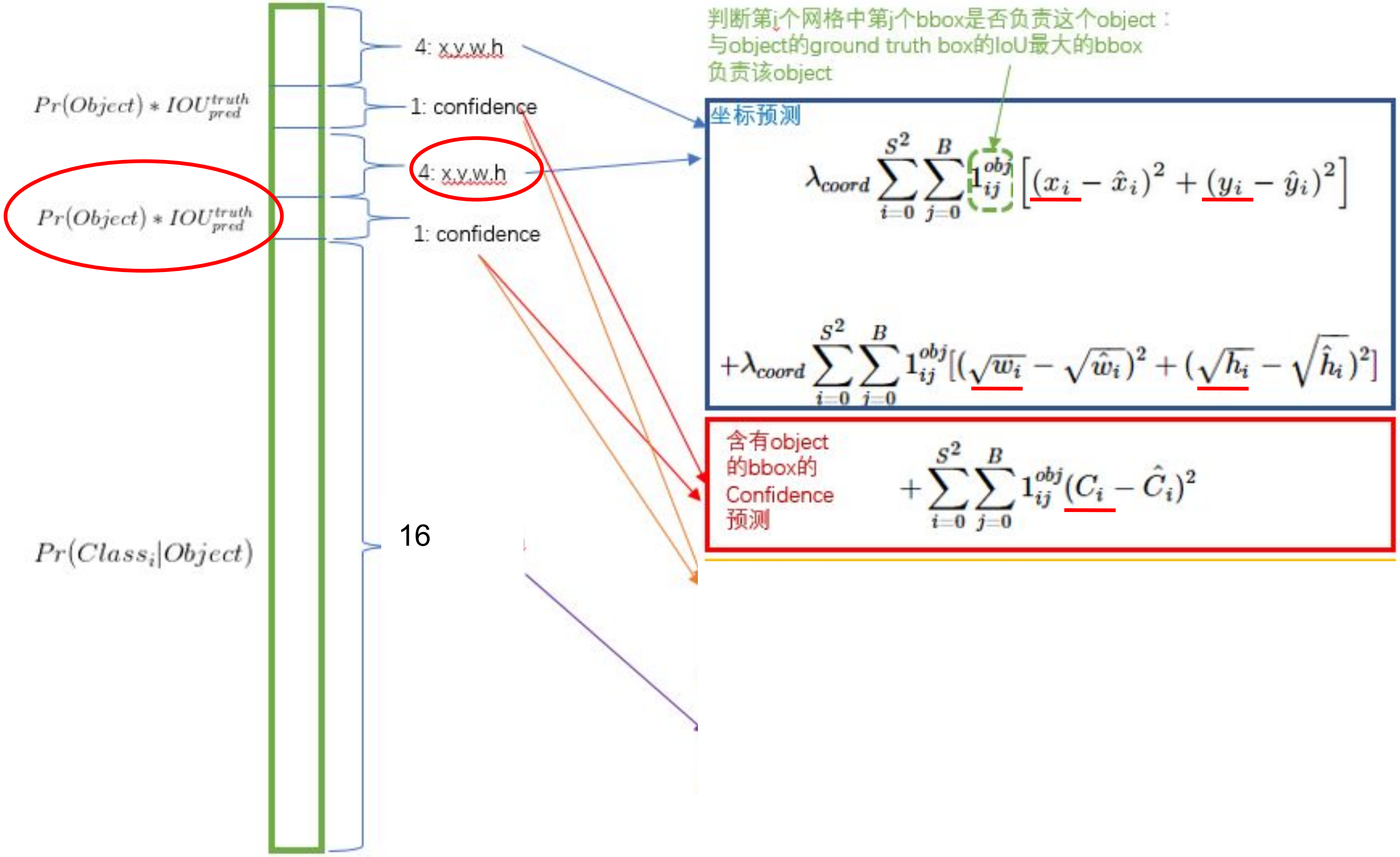


Find the best one, adjust it, increase the confidence



Find the best one, adjust it, increase the confidence





$Pr(Object) * IOU^{truth}_{pred}$

$Pr(Object) * IOU^{truth}_{pred}$

$Pr(Class_i|Object)$

4: x,y,w,h

1: confidence

4: x,y,w,h

1: confidence

16

判断第i个网格中第j个bbox是否负责这个object：
与object的ground truth box的IoU最大的bbox
负责该object

坐标预测

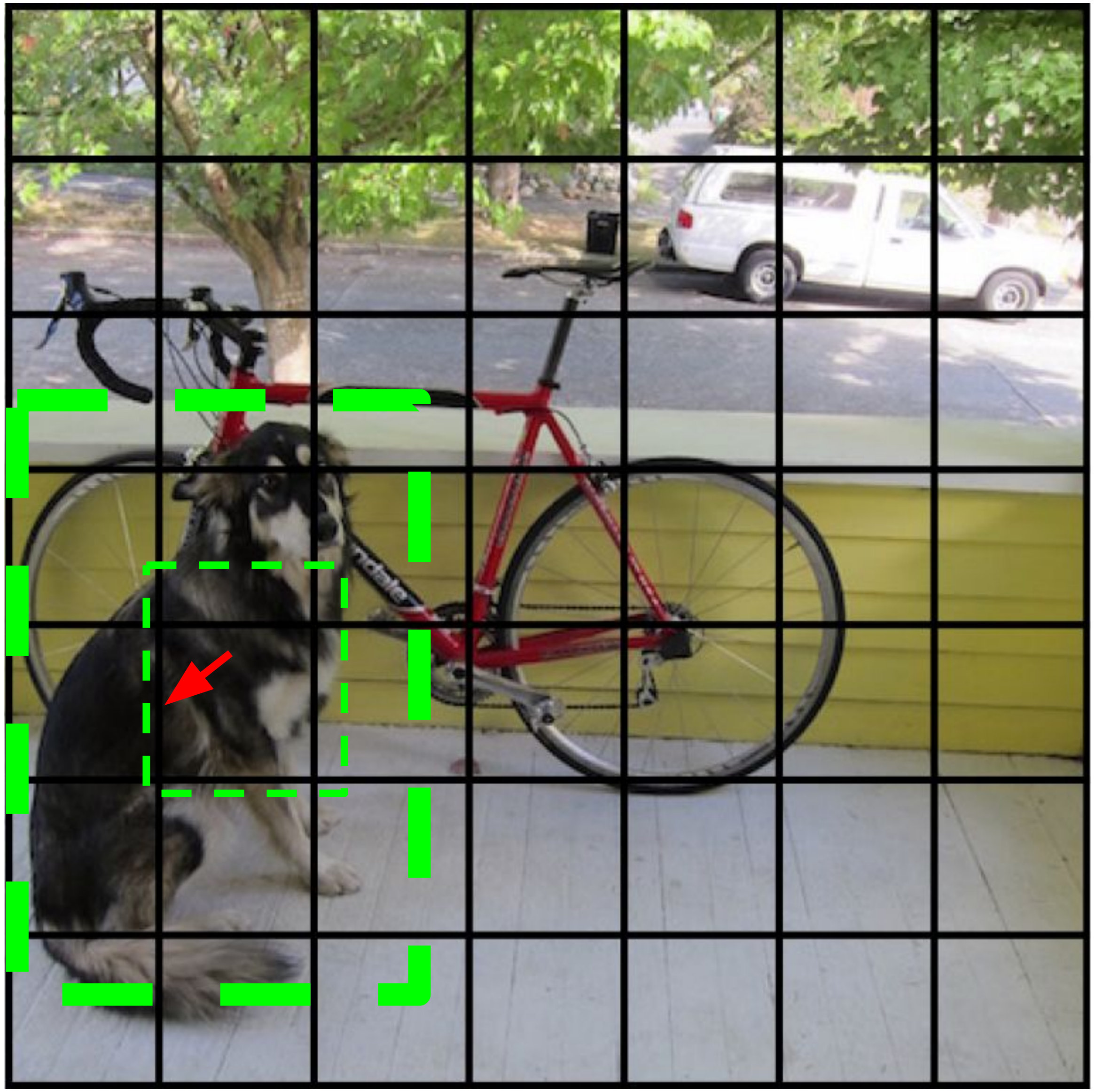
$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\boxed{1_{ij}^{obj}}\left[(\underline{x_i}-\hat{x}_i)^2+(\underline{y_i}-\hat{y}_i)^2\right]$$

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

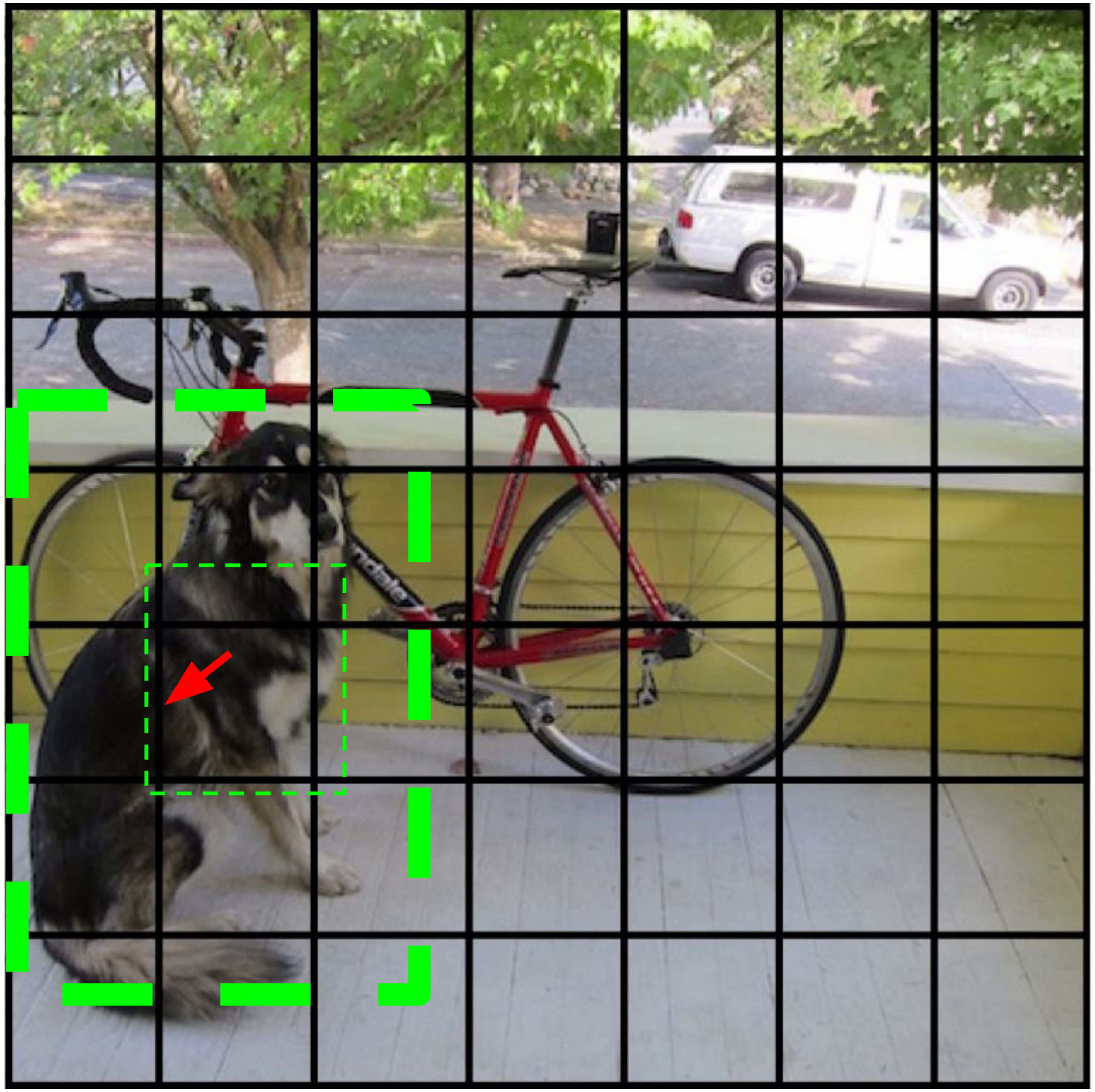
含有object
的bbox的
Confidence
预测

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

Decrease the confidence of other boxes



Decrease the confidence of other boxes



$Pr(Object) * IOU_{pred}^{truth}$

4: x,y,w,h

1: confidence

4: x,y,w,h

$Pr(Object) * IOU_{pred}^{truth}$

1: confidence

$Pr(Class_i|Object)$

16

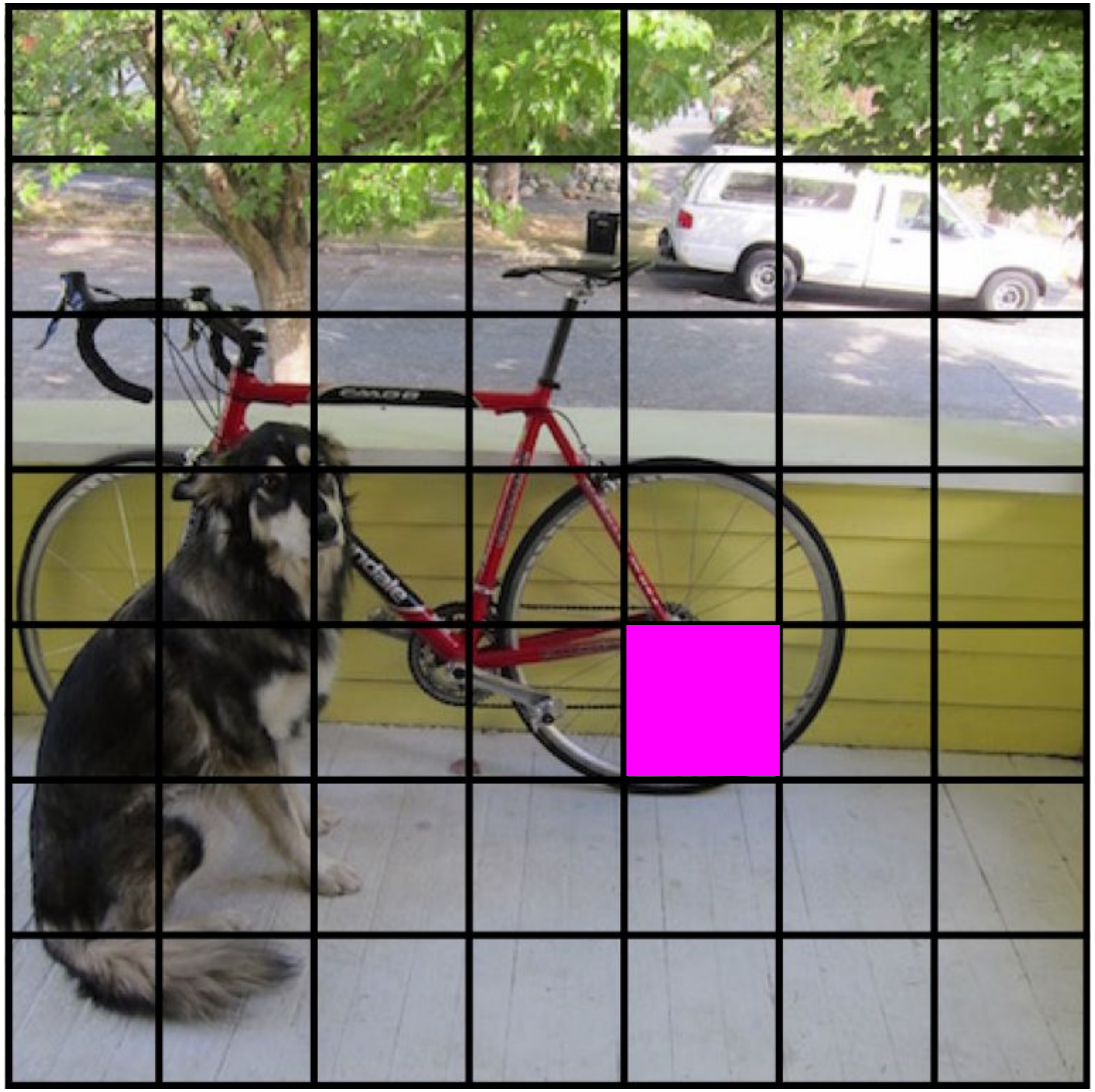
不含object的bbox的confidence预测

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

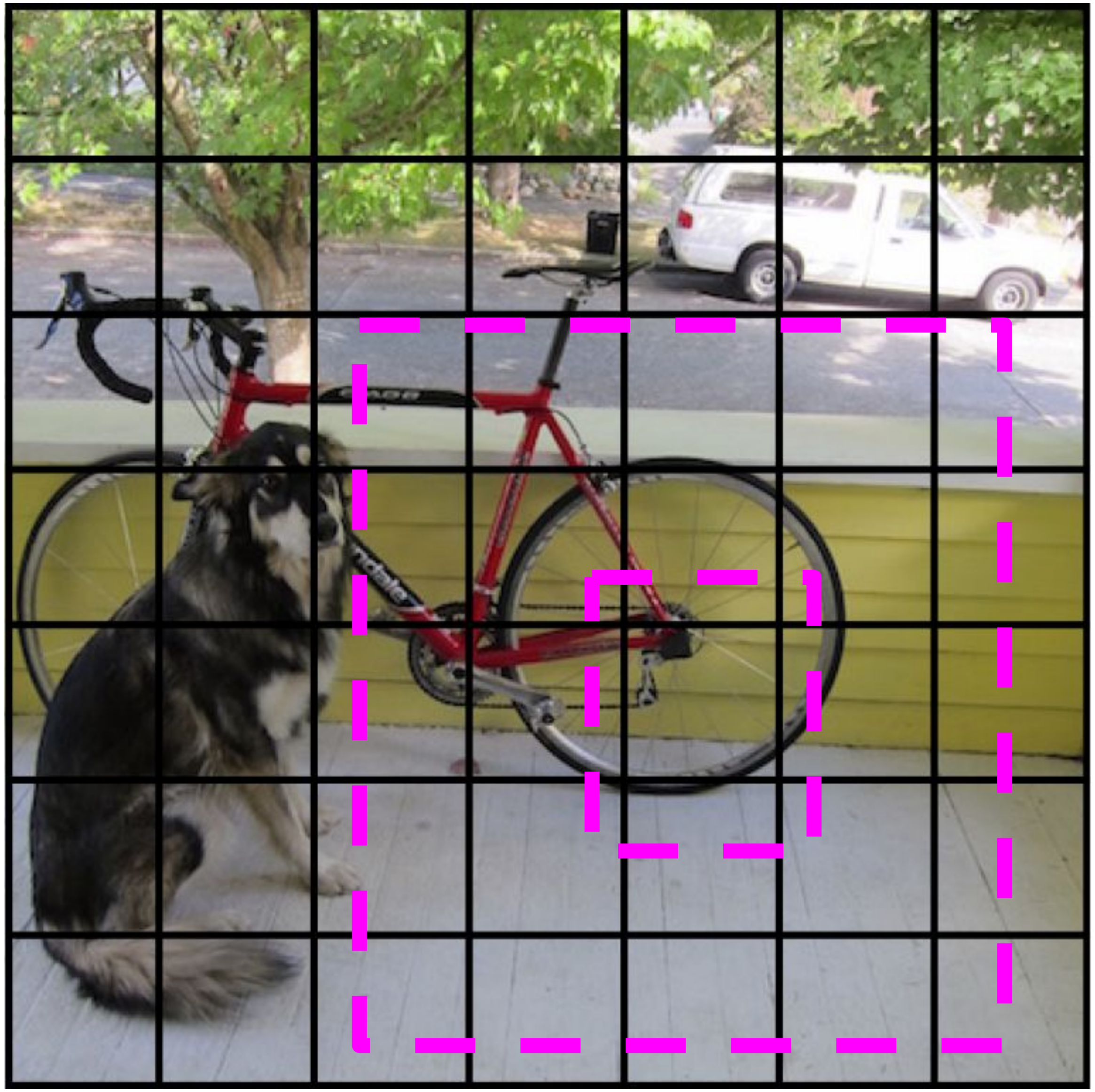
网格中包含有object的中心，
就负责预测该object的类别概率

http://blog.csdn.net/hrsstudy

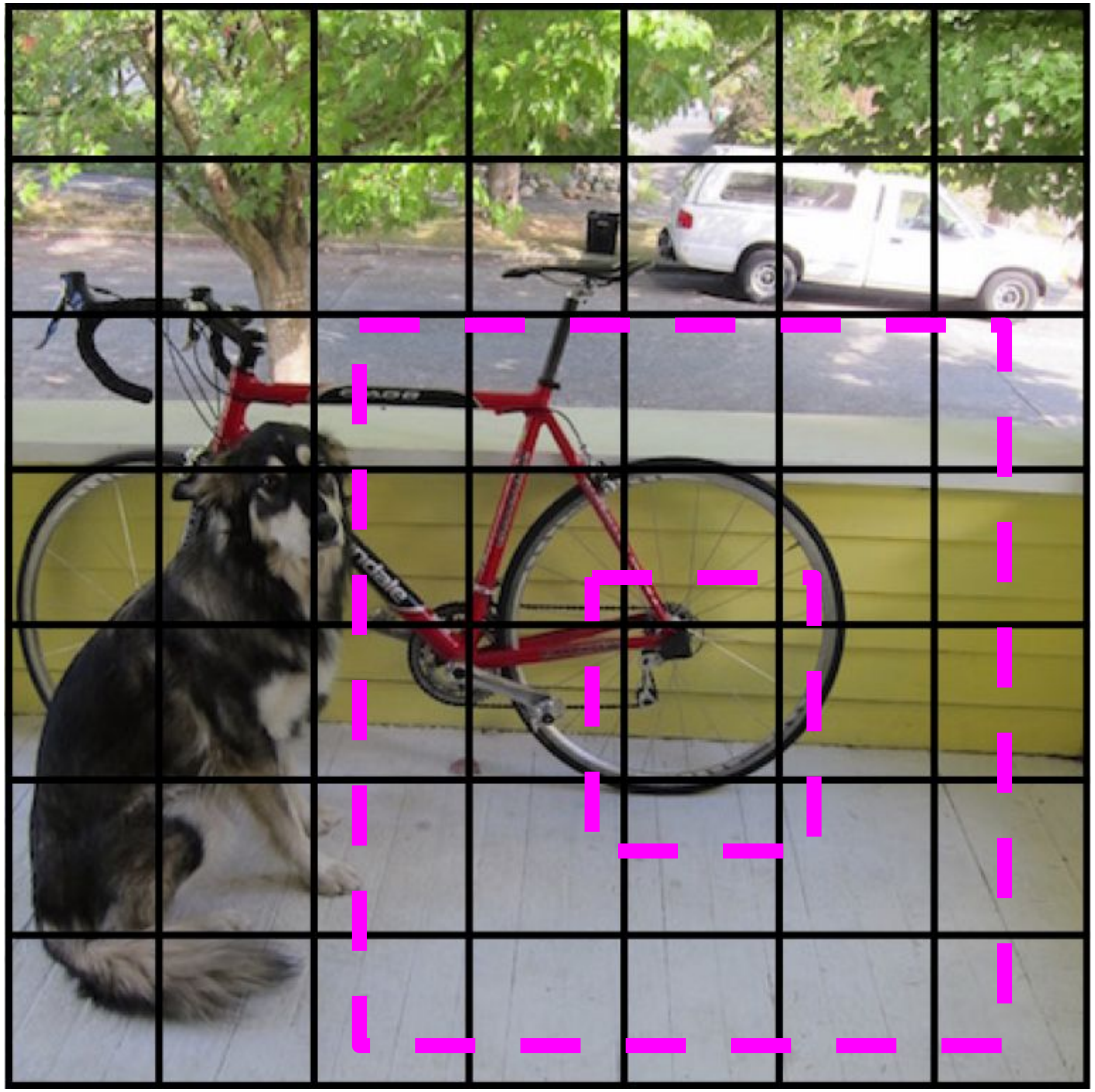
Some cells don't have any ground truth detections!



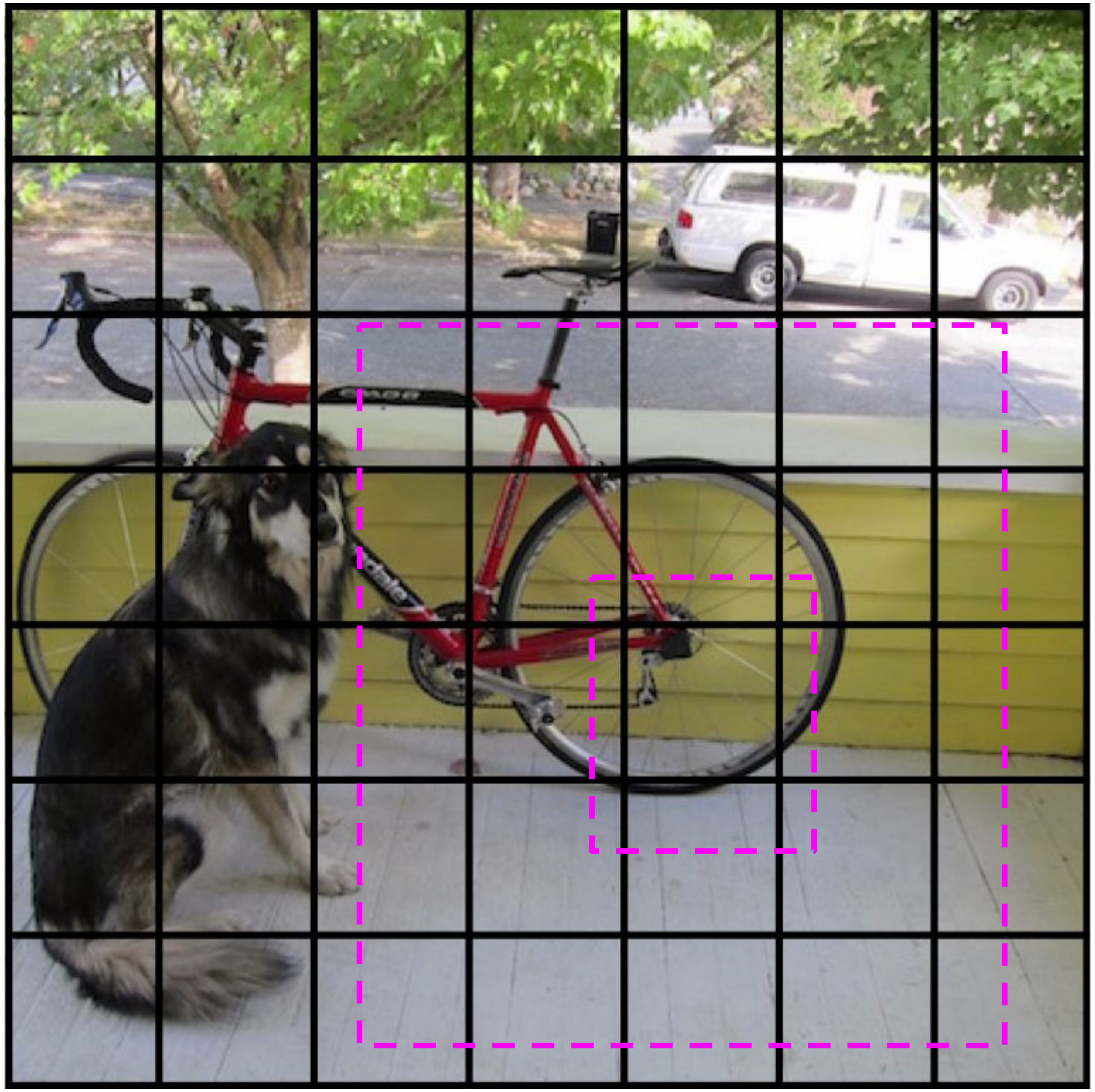
Some cells don't have any ground truth detections!

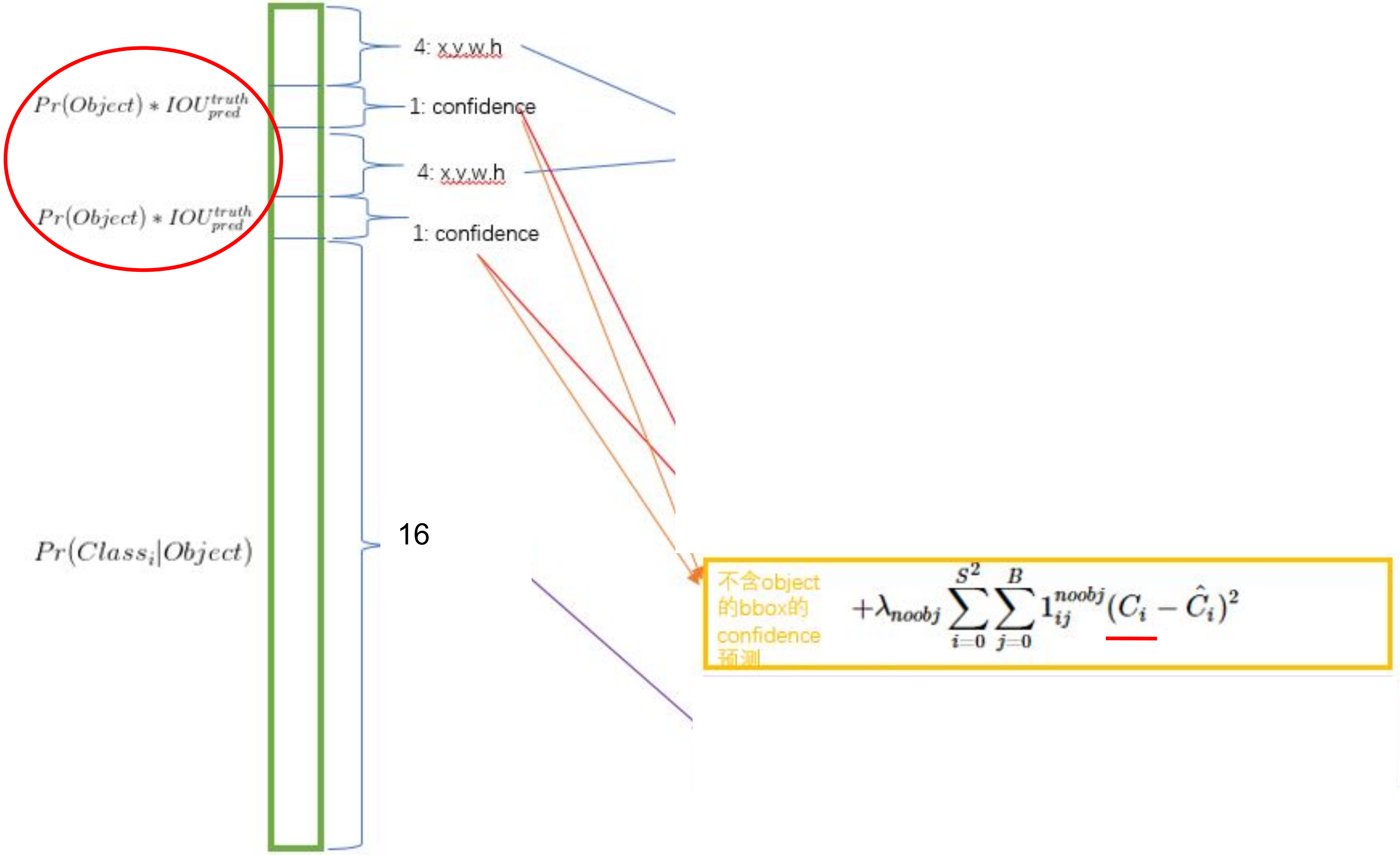


Decrease the confidence of these boxes



Decrease the confidence of these boxes





$Pr(Object) * IOU_{pred}^{truth}$

$Pr(Object) * IOU_{pred}^{truth}$

$Pr(Class_i|Object)$

4: x,y,w,h

1: confidence

4: x,y,w,h

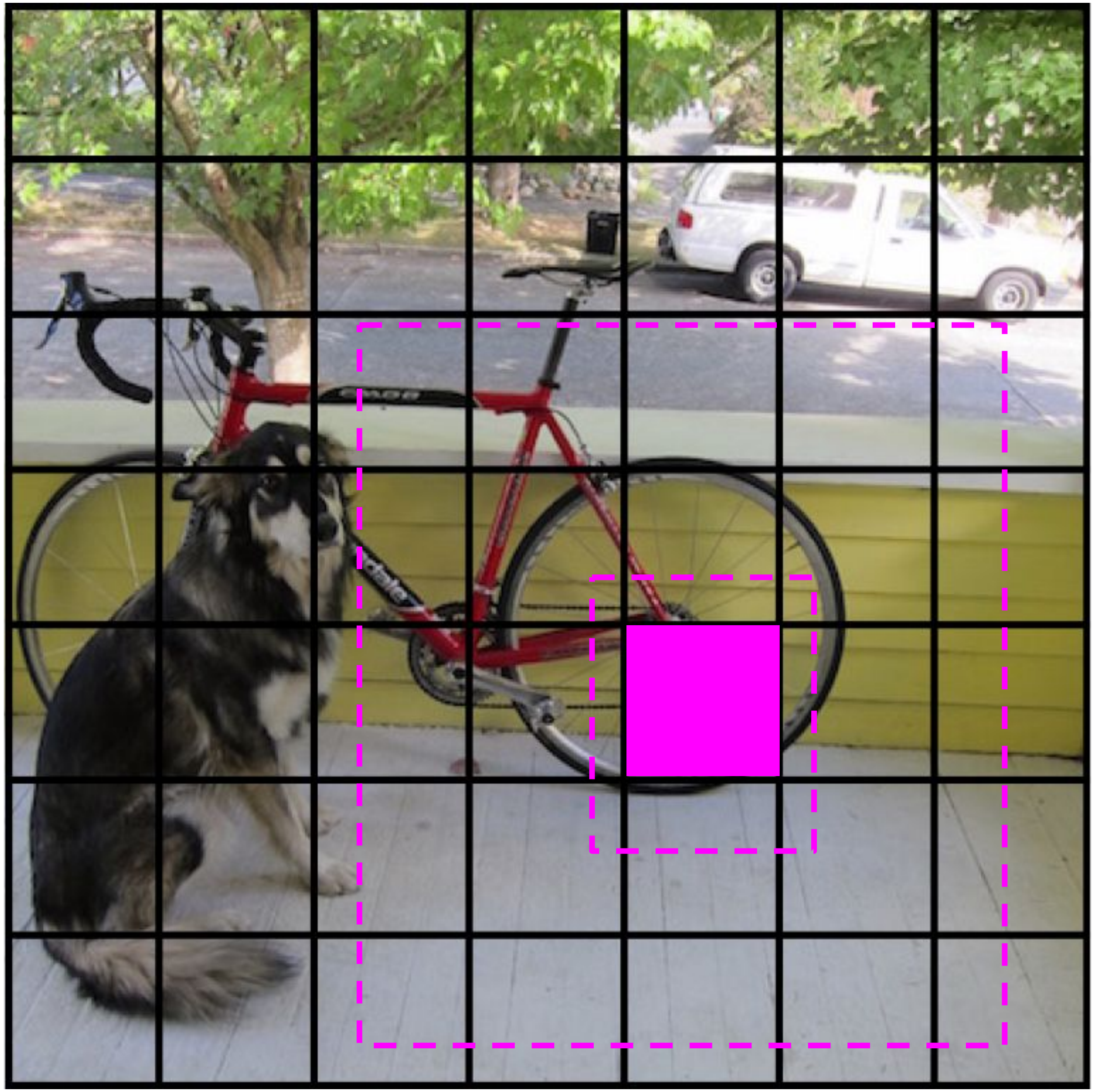
1: confidence

16

不含object的bbox的confidence预测

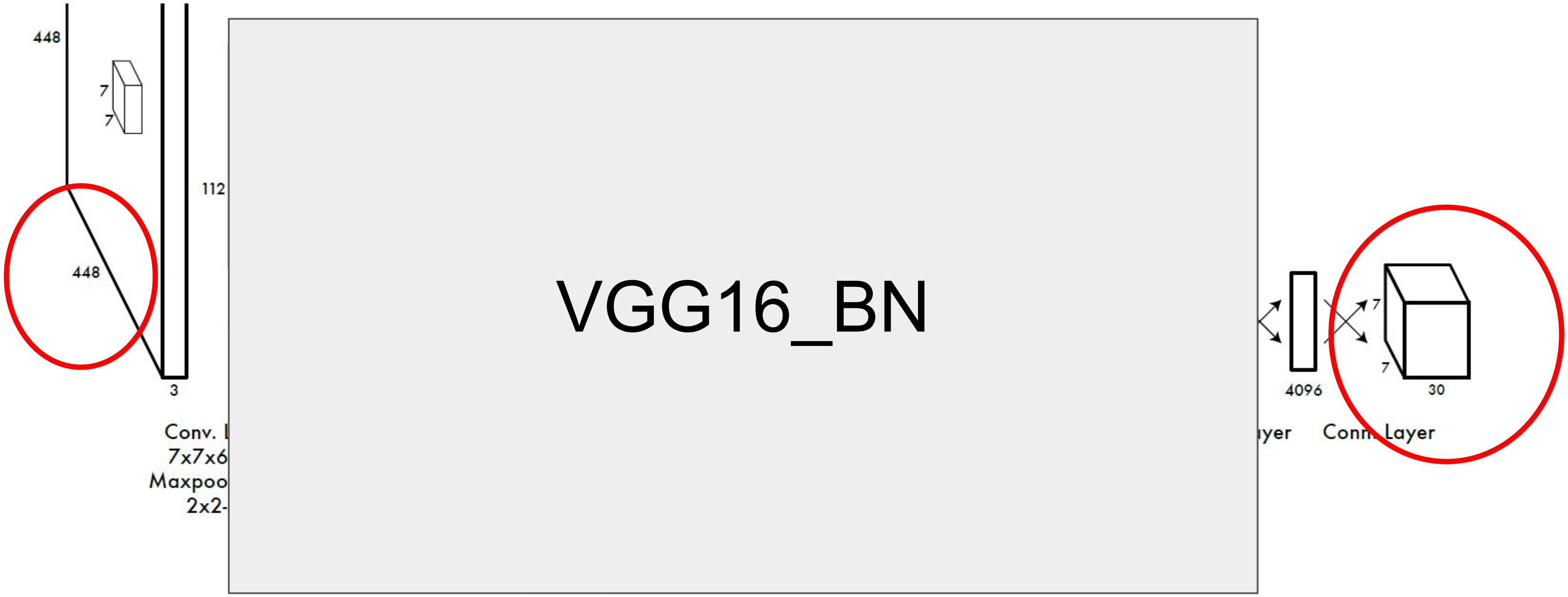
$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}(\underline{C_i} - \hat{C}_i)^2$$

Don’t adjust the class probabilities or coordinates



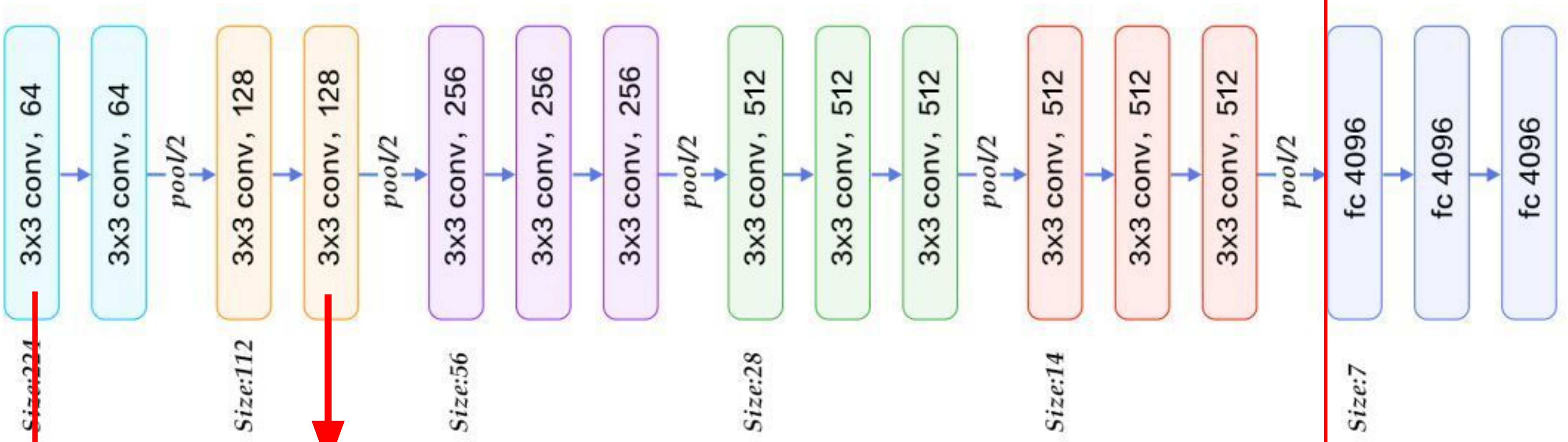
Backbone model

The input & output are the same as Yolov1.
But we will use VGG16_bn as out baseline backbone model.


448
7
7
448
112
3
Conv. L
7x7x6
Maxpoo
2x2-
VGG16_BN
4096
7
7
30
ayer
Conn. Layer

Therefore, we can use Imagenet pretrain model.

What is VGG16_bn


3x3 conv, 64
3x3 conv, 64
pool/2
3x3 conv, 128
3x3 conv, 128
pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
fc 4096
fc 4096
fc 4096
Size:224
Size:112
Size:56
Size:28
Size:14
Size:7

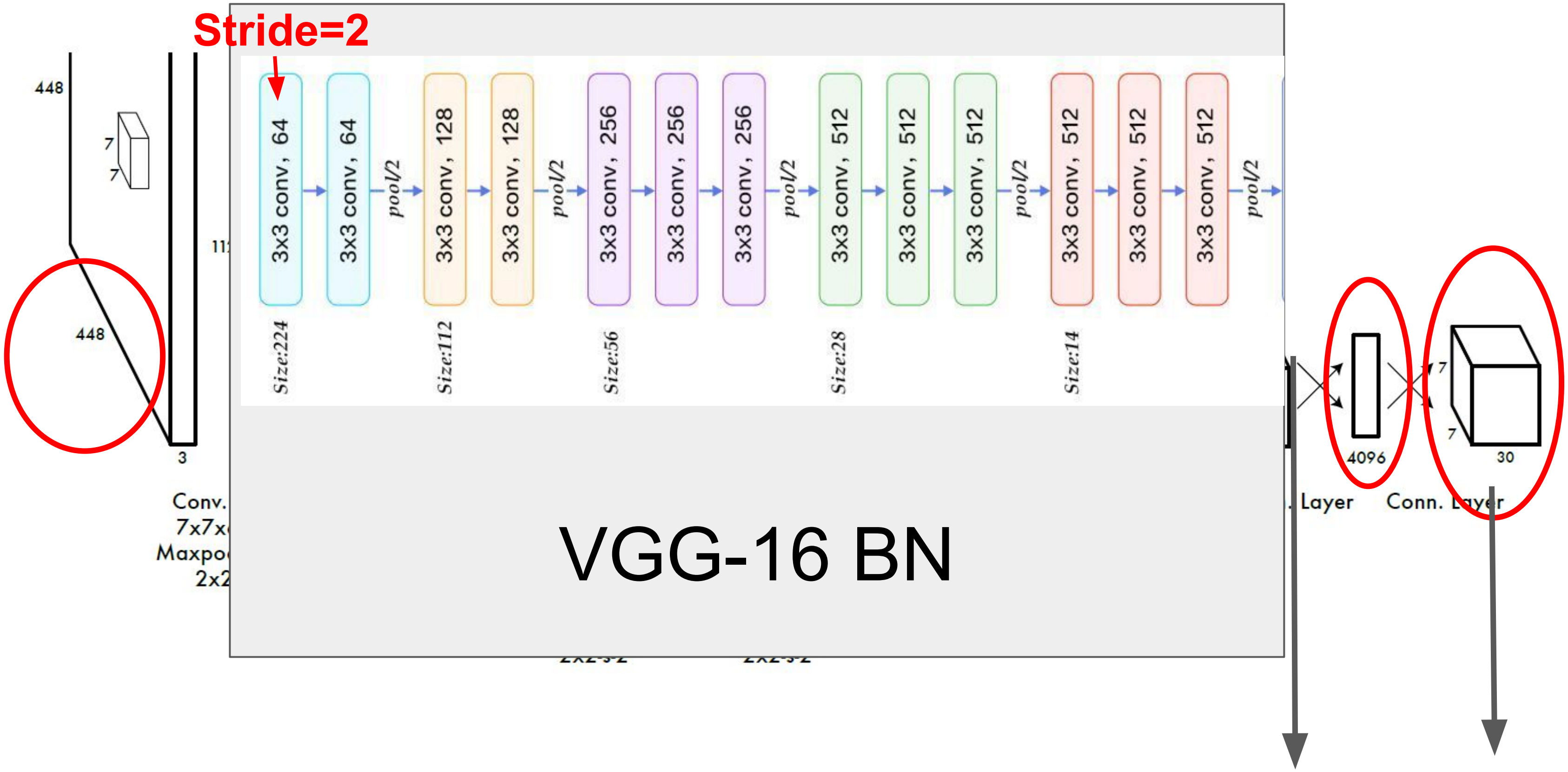
Inside : Conv + bn +relu

→ We only need this part!

Note ! We made some modification.

Stride=2 for 1st Convolution. (for reduce feature map size)

Summary of our baseline model


448
7
7
448
3
112
Conv.
7x7x
Maxpo
2x2
3x3 conv, 64
3x3 conv, 64
pool/2
3x3 conv, 128
3x3 conv, 128
pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
Size:224
Size:112
Size:56
Size:28
Size:14
4096
7
7
30
Layer
Conn. Layer

Stride=2

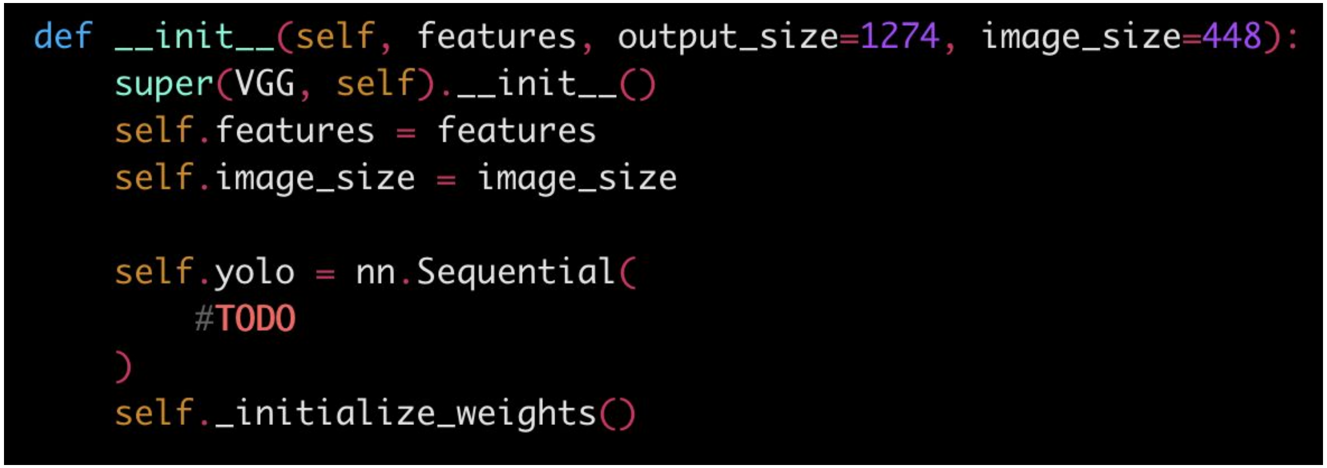
VGG-16 BN

512x7x7

7x7x26

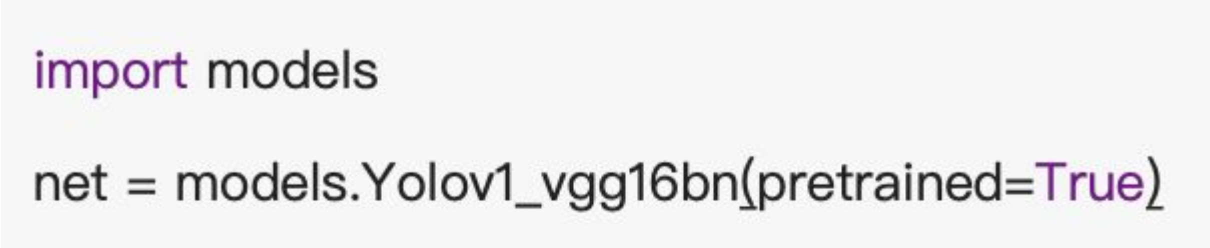
Don’t Worry ~~

We provide the model code (models.py) with some “TODO” 🙃



```
def __init__(self, features, output_size=1274, image_size=448):
    super(VGG, self).__init__()
    self.features = features
    self.image_size = image_size

    self.yolo = nn.Sequential(
        #TODO
    )
    self._initialize_weights()
```

How to use ? Put the code in the same directory, then...



```
import models

net = models.Yolov1_vgg16bn(pretrained=True)
```

Details of training class & confidence

$\Pr(\text{Class}_i|\text{Object})$

Use one-hot vector to train MSE loss

$\Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}}$

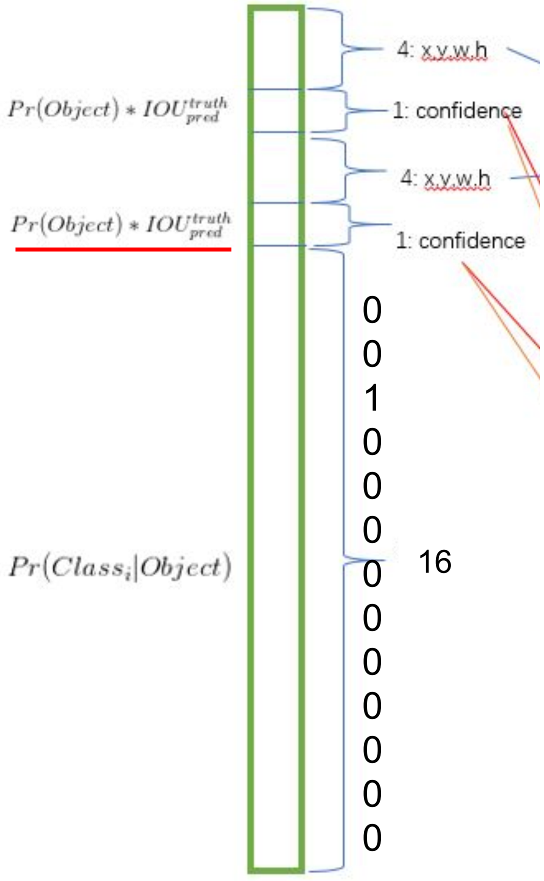
confidence score

MSE loss

GT are not just 1 or 0

Ground Truth Label:

- If it is an object → the label is **1 * IoU**.
- If it is not an object → the label is **0**.



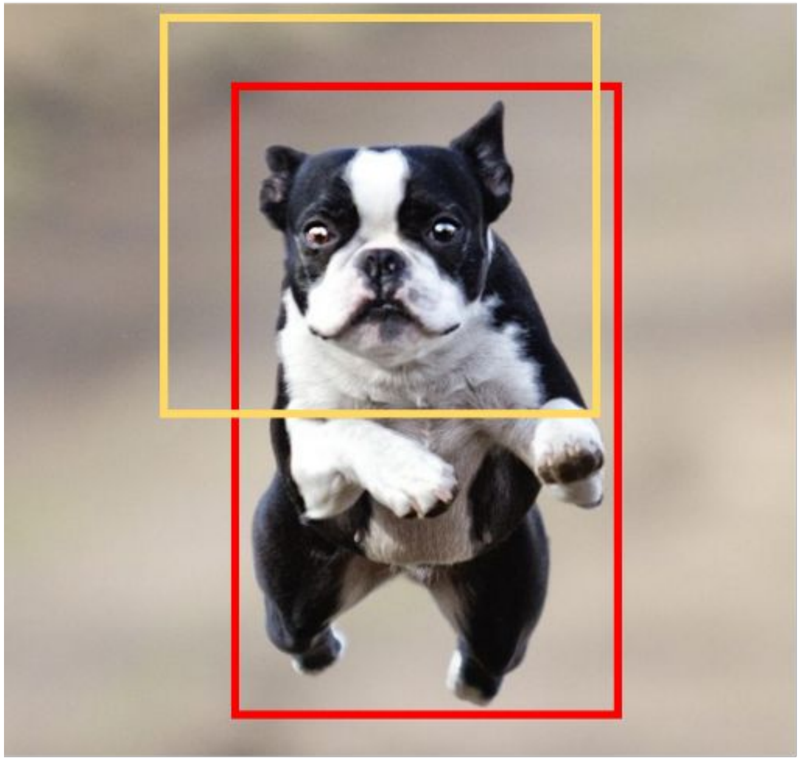
4: x,y,w,h
$Pr(Object) * IOU^{truth}_{pred}$
1: confidence
4: x,y,w,h
$Pr(Object) * IOU^{truth}_{pred}$
1: confidence
0
0
1
0
0
0
0
0
0
0
0
0
0
16
$Pr(Class_i|Object)$

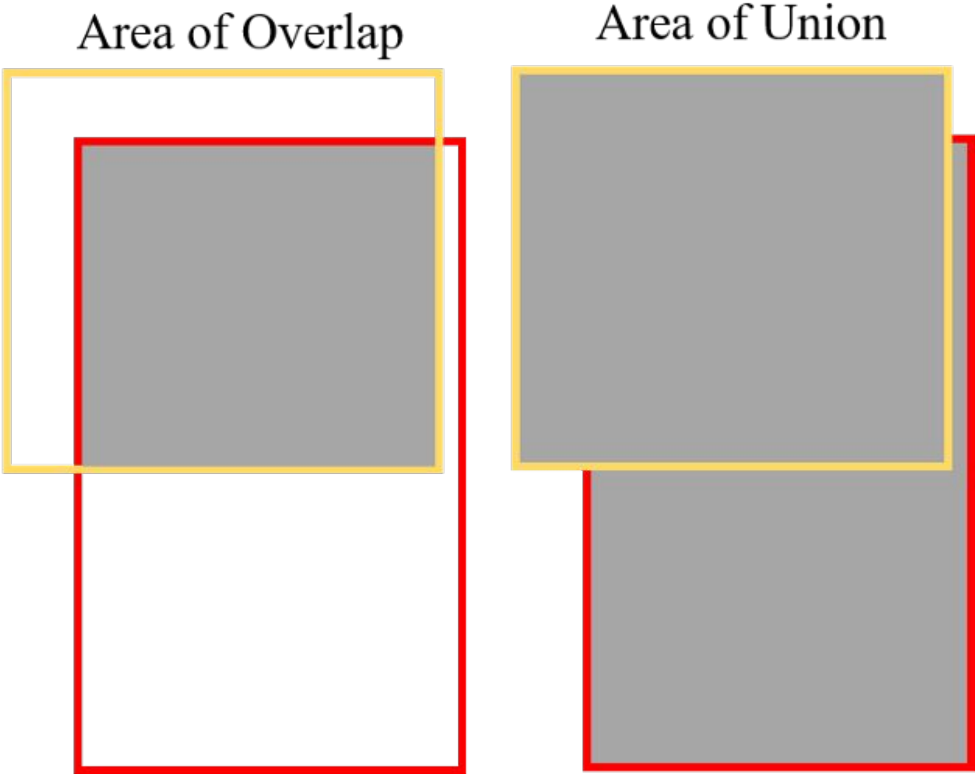
Implementation Details

IoU

intersection over union

$$\text{IoU}=\frac{\text{Area of Overlap}}{\text{Area of Union}}$$




Area of Overlap
Area of Union

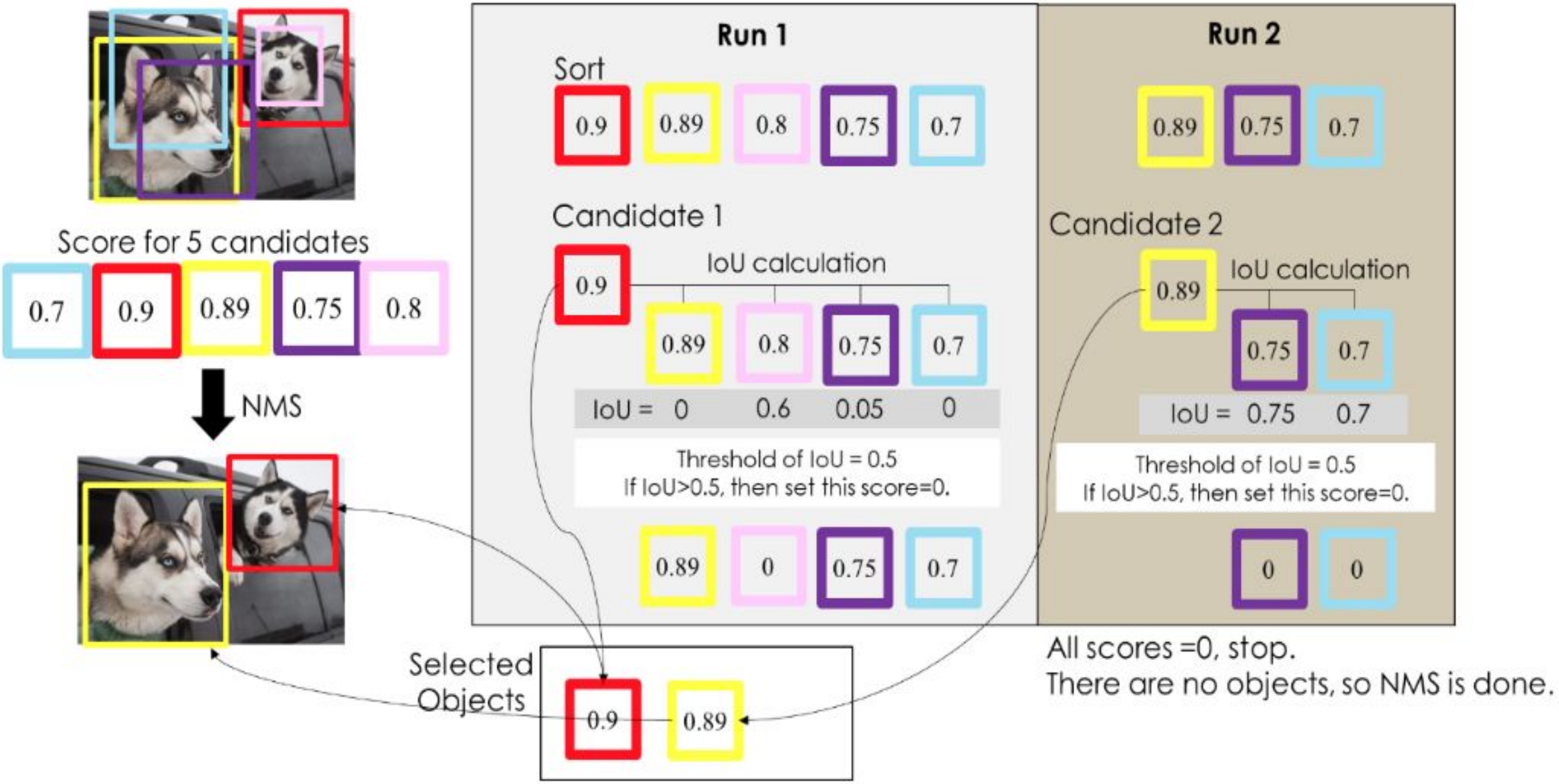
Details of Inference

1. For every bbox, calculate its **class-specific confidence score.**

$$\underbrace{\boxed{\Pr(\text{Class}_i|\text{Object})}}_{\textbf{max class score}} * \underbrace{\boxed{\Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}}}}_{\textbf{confidence score}} = \underbrace{\boxed{\Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}}}}_{\textbf{class-specific confidence score}}$$

2. We will have 7*7*2 = **98** scores.
3. Use one threshold (maybe 0.1) to filter out the low score detection.
4. Execute Non-maximum Suppression (NMS).

Non-maximum Suppression


Score for 5 candidates
0.7
0.9
0.89
0.75
0.8
NMS
Run 1
Sort
0.9
0.89
0.8
0.75
0.7
Candidate 1
0.9
IoU calculation
0.89
0.8
0.75
0.7
IoU = 0
0.6
0.05
0
0.89
0
0.75
0.7
Run 2
0.89
0.75
0.7
Candidate 2
0.89
IoU calculation
0.75
0.7
IoU = 0.75
0.7
0
0
Selected Objects
0.9
0.89

Threshold of IoU = 0.5
If IoU>0.5, then set this score=0.

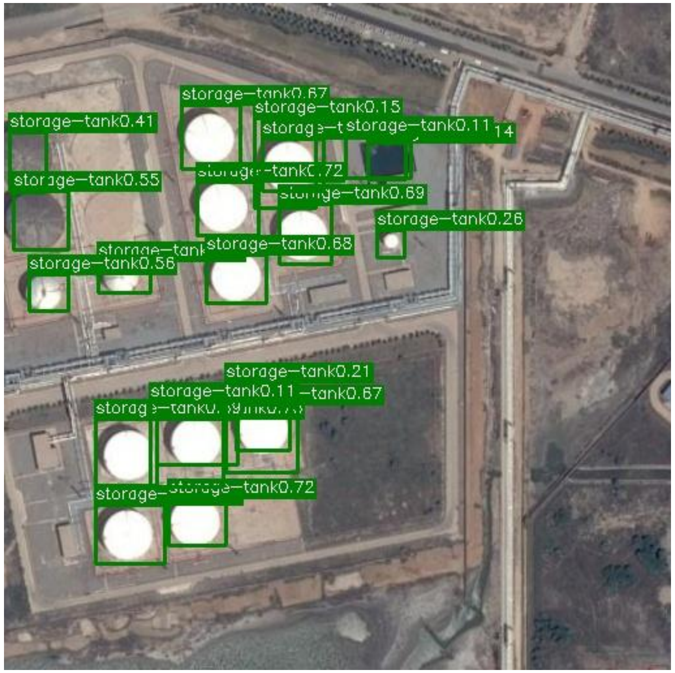
Threshold of IoU = 0.5
If IoU>0.5, then set this score=0.

All scores =0, stop.
There are no objects, so NMS is done.

Non-maximum Suppression

Why we still keep two anchor boxes of one grid for NMS?
→ One grid may predict **two same class** bboxs with **low IoU**.

Cons : YoloV1 cannot handle two bboxes of one grid with different classes.


storage–tank0.67
storage–tank0.15
storage–tank0.41
storage–tank0.11
14
storage–tank0.55
–tank0.72
storage–tank0.69
storage–tank0.26
storage–tank0.68
storage–tank0.56
storage–tank0.21
storage–tank0.11
–tank0.67
storage–tank0.72

Outline

- Object Detection in Aerial Images
    - Task Definition
    - Object Detection Network (Yolo v1)
- Assignment
    - TODO
    - Implementation Details
    - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

TODO

- In this assignment, you will need to implement **two** detection models and provide discussions.
  1. **YoloV1 with VGG16_bn backbone (baseline model)**
     Implement YoloV1-vgg16bn model to perform object detection. The results of this model should **pass** the baseline performance.
  2. **An improved model**
     Implement an improved model. The performance of this model should be better than that of the baseline model. You can change the backbone (Resnet), increase the grid number or choose any model different from YoloV1-vgg16bn, such as Yolov2, Yolov3, SSD, faster-rcnn, etc.
     (You can ask TA if you have some idea but you are not sure whether it is an improved model.)

Outline

- Object Detection in Aerial Images
    - Task Definition
    - Object Detection Network (Yolo v1)
- Assignment
    - TODO
    - Implementation Details
    - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Implementation Details

Provided Data

hw2_train_val.zip (1.5GB)

hw2_evaluation_task2.py

visualize_bbox.py

models.py

The .py files above will be included in your default GitHub repository.
To download the dataset (hw2_train_val.zip), follow the instructions in the **README** file in your GitHub repository.

Implementation Details

Dataset Description

```
hw2_train_val/
 |____ train15000/
 |__________ images/
 |__________ labelTxt_hbb/
 |____ val1500/
 |__________ images/
 |__________ labelTxt_hbb/
```

- train15000/
  - images/ Contains 15000 aerial images. ex. 13000.jpg
  - labelTxt_hbb/ Contains 15000 label txt file. ex. 13000.txt
- val1500/
  - images/ Contains 1500 aerial images. ex. 0150.jpg
  - labelTxt_hbb/ Contains 1500 label txt file. ex. 0150.txt

- We will have a hidden test set which contains 1500 images, and the directory format is same as “val1500/”. (We will call it test1500)

Implementation Details

Annotation / Result format

We don’t need it while training. Only need it in evaluation.

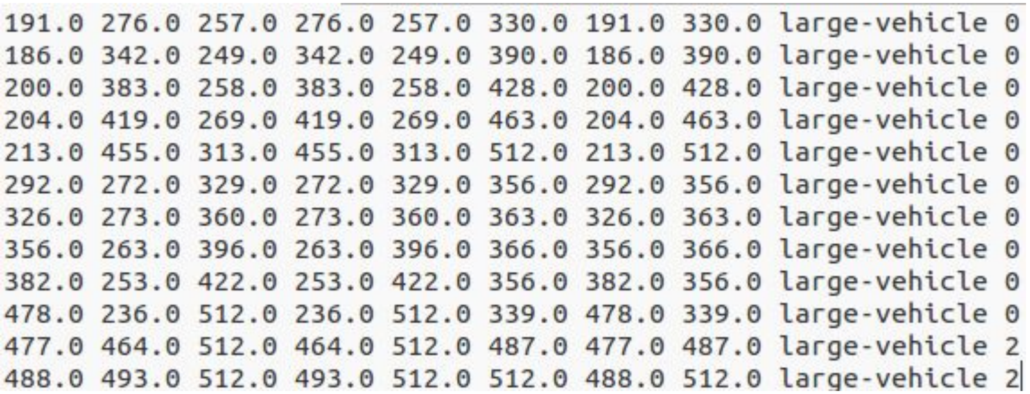
Each row represents a detection:

Annotation file: xmin ymin xmax ymin xmax ymax xmin ymax class difficulty

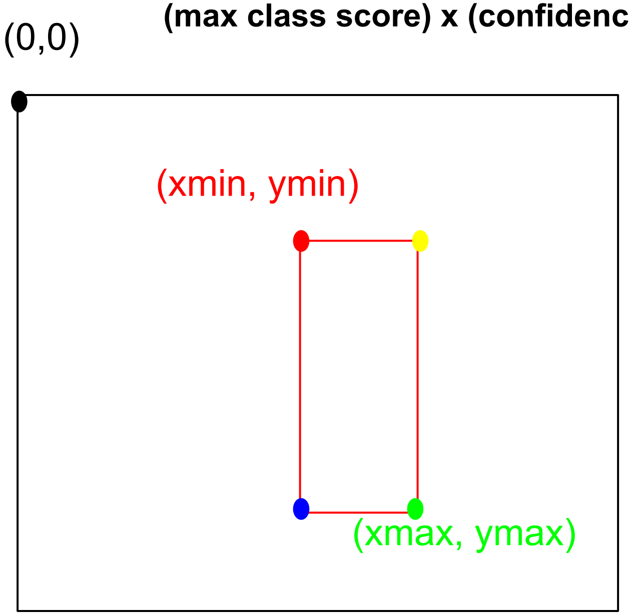
0,1,2

Prediction file: xmin ymin xmax ymin xmax ymax xmin ymax class **class-specific confidence score**

(max class score) x (confidence)



```
191.0 276.0 257.0 276.0 257.0 330.0 191.0 330.0 large-vehicle 0
186.0 342.0 249.0 342.0 249.0 390.0 186.0 390.0 large-vehicle 0
200.0 383.0 258.0 383.0 258.0 428.0 200.0 428.0 large-vehicle 0
204.0 419.0 269.0 419.0 269.0 463.0 204.0 463.0 large-vehicle 0
213.0 455.0 313.0 455.0 313.0 512.0 213.0 512.0 large-vehicle 0
292.0 272.0 329.0 272.0 329.0 356.0 292.0 356.0 large-vehicle 0
326.0 273.0 360.0 273.0 360.0 363.0 326.0 363.0 large-vehicle 0
356.0 263.0 396.0 263.0 396.0 366.0 356.0 366.0 large-vehicle 0
382.0 253.0 422.0 253.0 422.0 356.0 382.0 356.0 large-vehicle 0
478.0 236.0 512.0 236.0 512.0 339.0 478.0 339.0 large-vehicle 0
477.0 464.0 512.0 464.0 512.0 487.0 477.0 487.0 large-vehicle 2
488.0 493.0 512.0 493.0 512.0 512.0 488.0 512.0 large-vehicle 2
```



Implementation Details

Class Format

There are 16 classes:

plane, ship, storage-tank, baseball-diamond, tennis-court, basketball-court,

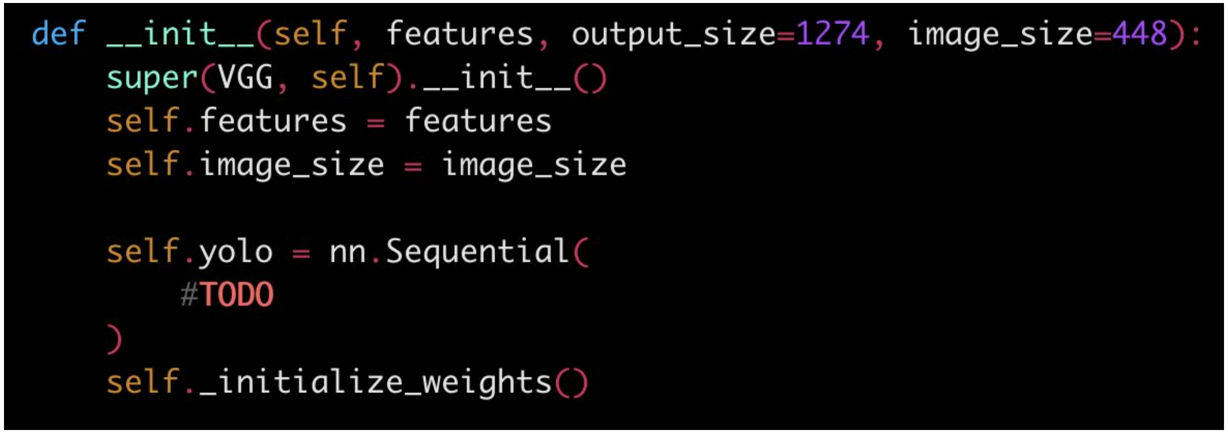
ground-track-field, harbor, bridge, small-vehicle, large-vehicle, helicopter,

roundabout, soccer-ball-field, swimming-pool, container-crane

Implementation Details

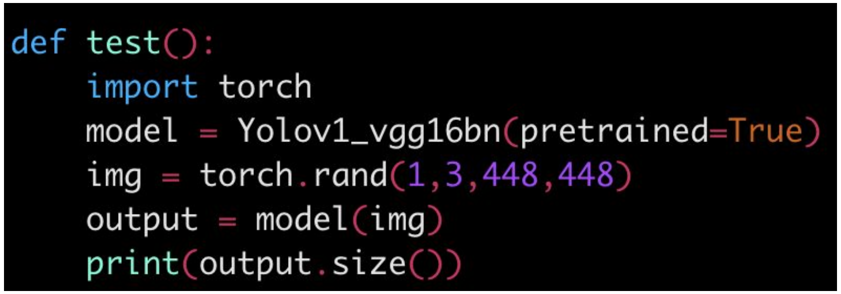
Backbone Model -- VGG16_bn

1. **We have to resize the image to 448 x 448**
2. **You have to complete the TODO (or you can just write your own one)**



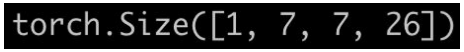
```
def __init__(self, features, output_size=1274, image_size=448):
    super(VGG, self).__init__()
    self.features = features
    self.image_size = image_size

    self.yolo = nn.Sequential(
        #TODO
    )
    self._initialize_weights()
```

3. **I have a test example of the model forwarding in my code.**



```
def test():
    import torch
    model = Yolov1_vgg16bn(pretrained=True)
    img = torch.rand(1,3,448,448)
    output = model(img)
    print(output.size())
```

→ The output should be

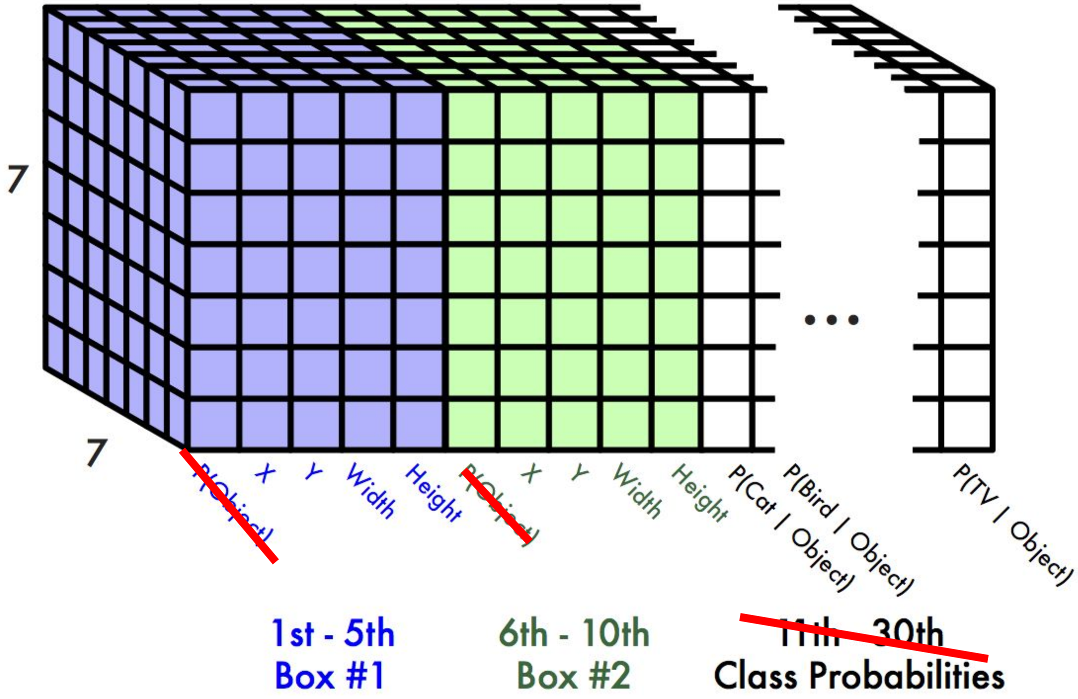


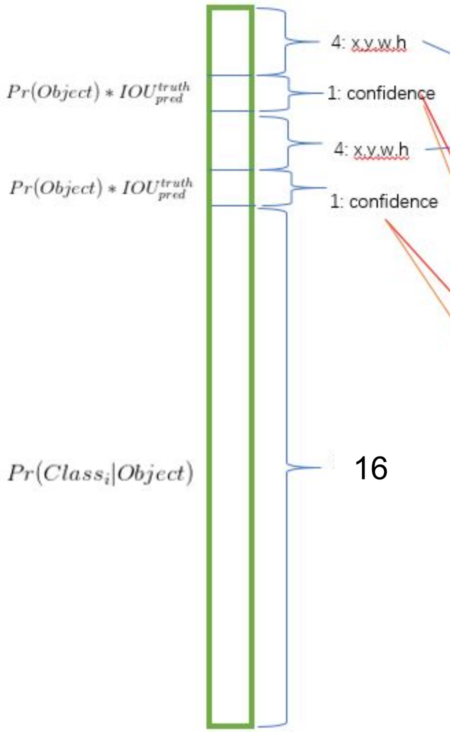
```
torch.Size([1, 7, 7, 26])
```

Implementation Details

Model Prediction

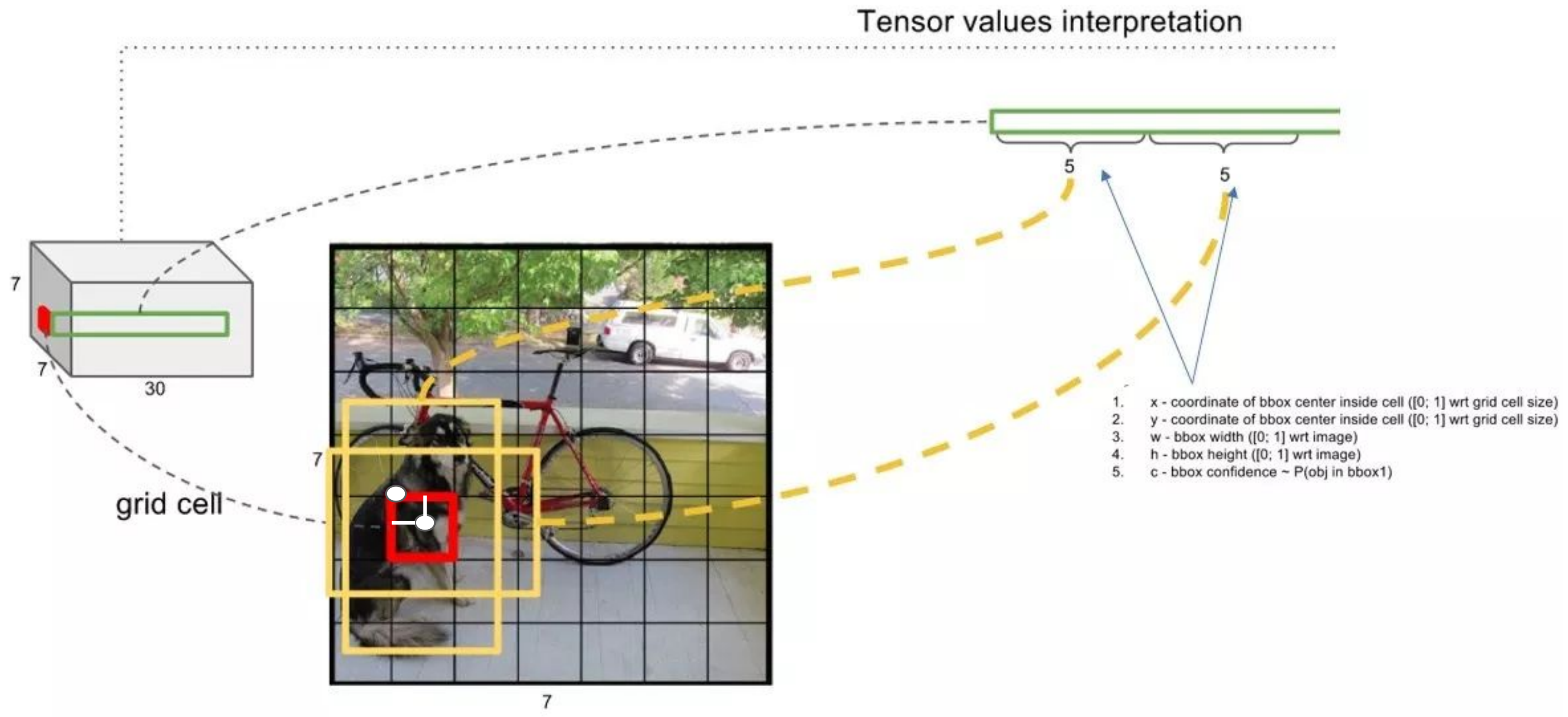
You have to parse the annotation file to meet the format of Yolo
→ (x ,y , w, h , conf. , cls (16-dim))


7
7
~~P(Object)~~
X
Y
Width
Height
~~P(Object)~~
X
Y
Width
Height
P(Cat | Object)
P(Bird | Object)
...
P(TV | Object)
1st - 5th
Box #1
6th - 10th
Box #2
~~11th - 30th~~
Class Probabilities


4: x,y,w,h
$Pr(Object) * IOU_{pred}^{truth}$
1: confidence
4: x,y,w,h
$Pr(Object) * IOU_{pred}^{truth}$
1: confidence
$Pr(Class_i|Object)$
16

Implementation Details

Model Prediction -- x & y


Tensor values interpretation
7
7
30
grid cell
7
7
5
5

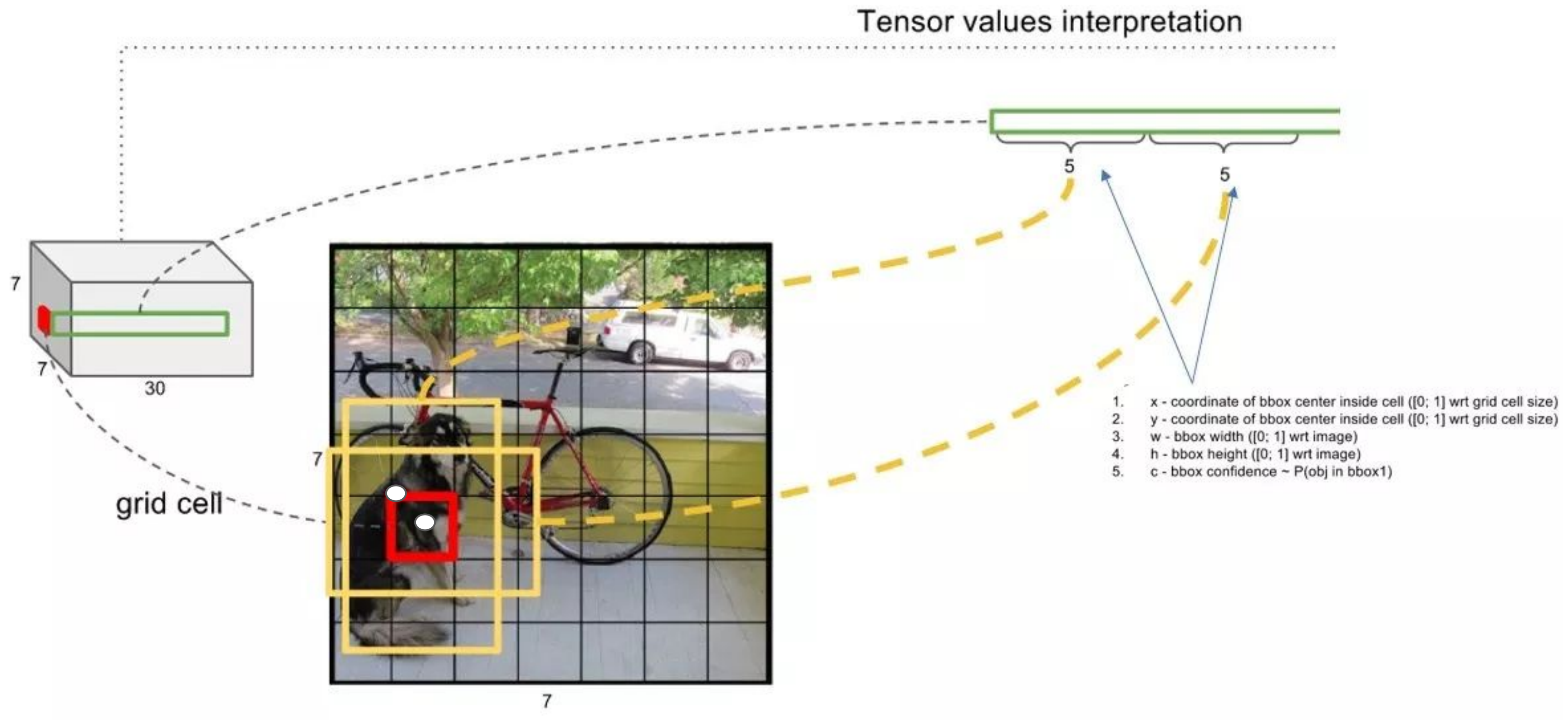
1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox confidence ~ P(obj in bbox1)

x : x-coordinate of bbox **center** inside cell ([0:1] with regard to **grid cell size**)

y : y-coordinate of bbox **center** inside cell ([0:1] with regard to **grid cell size**)

Implementation Details

Model Prediction -- w & h

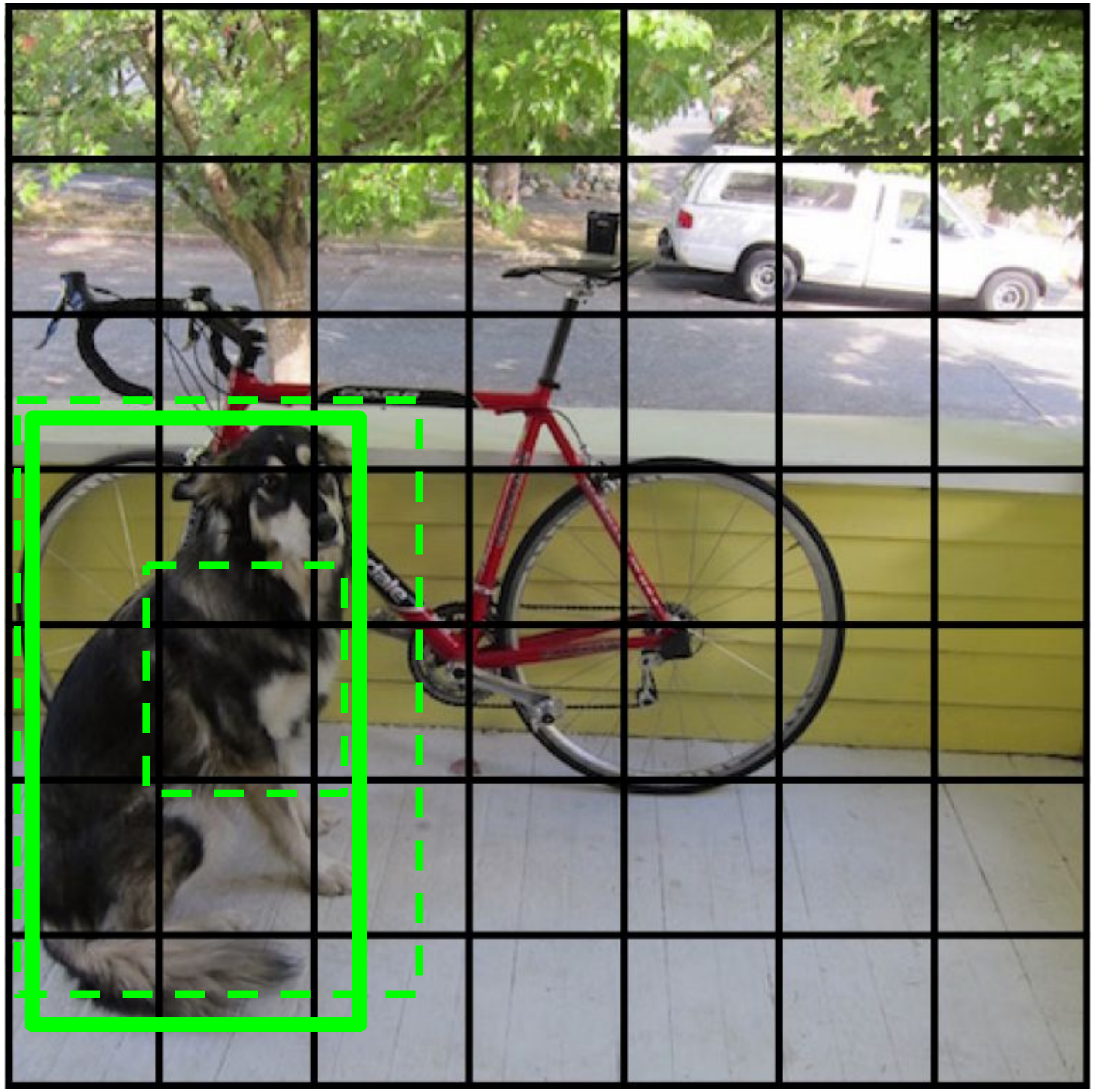

Tensor values interpretation
7
7
30
grid cell
7
7
5
5

1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox confidence ~ P(obj in bbox1)

w : bbox width ([0:1] with regard to **image size**)

h : bbox height ([0:1] with regard to **image size**)

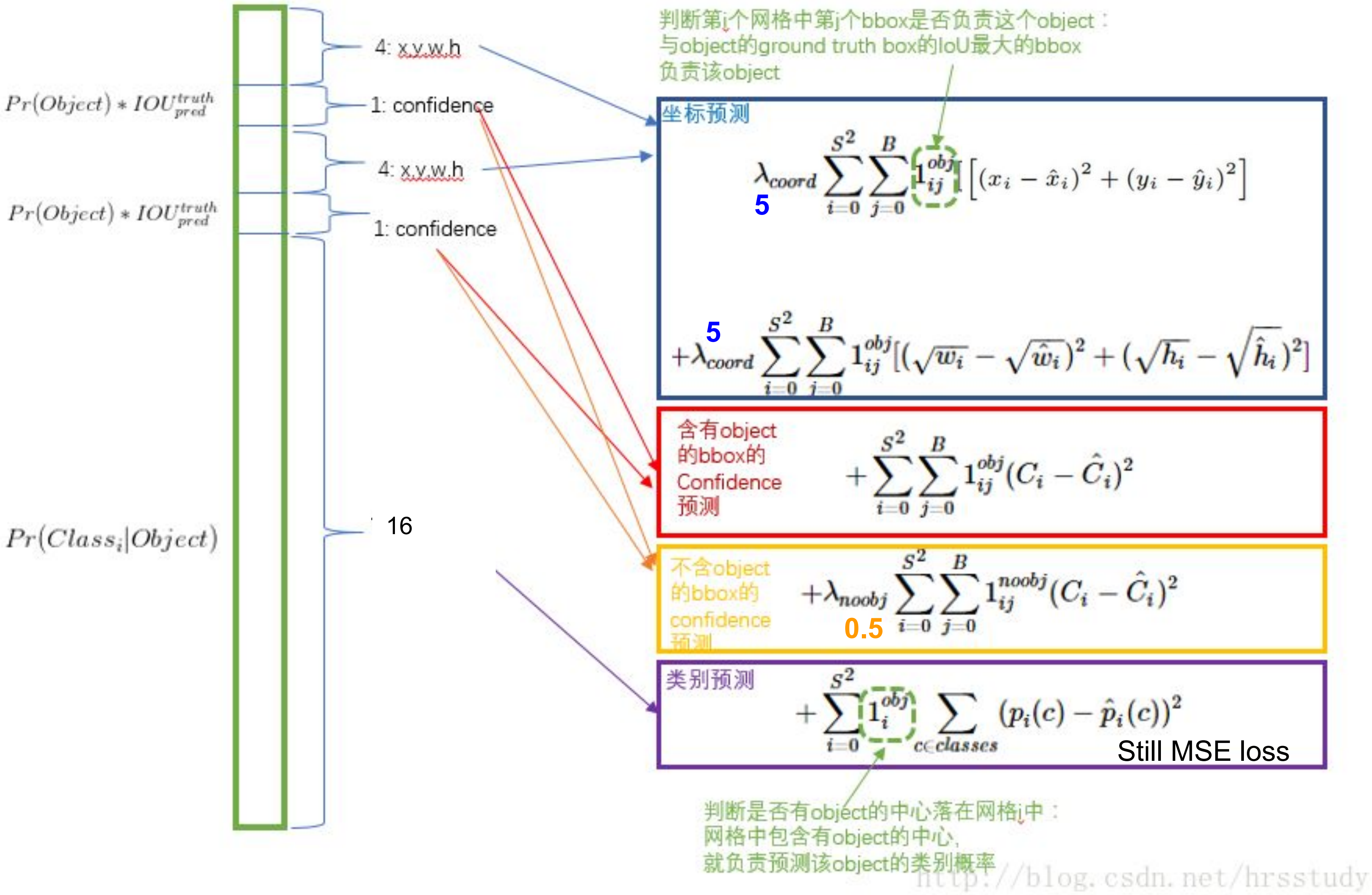
Find the best one, adjust it, increase the confidence



How to find?

Use IoU to find

Loss Hyperparameters



$Pr(Object) * IOU_{pred}^{truth}$

4: x,y,w,h

1: confidence

$Pr(Object) * IOU_{pred}^{truth}$

4: x,y,w,h

1: confidence

$Pr(Class_i|Object)$

16

判断第i个网格中第j个bbox是否负责这个object：与object的ground truth box的IoU最大的bbox负责该object

坐标预测

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

5

5

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

含有object的bbox的Confidence预测

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

不含object的bbox的confidence预测

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

0.5

类别预测

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

Still MSE loss

判断是否有object的中心落在网格i中：网格中包含有object的中心，就负责预测该object的类别概率

http://blog.csdn.net/hrsstudy

Outline

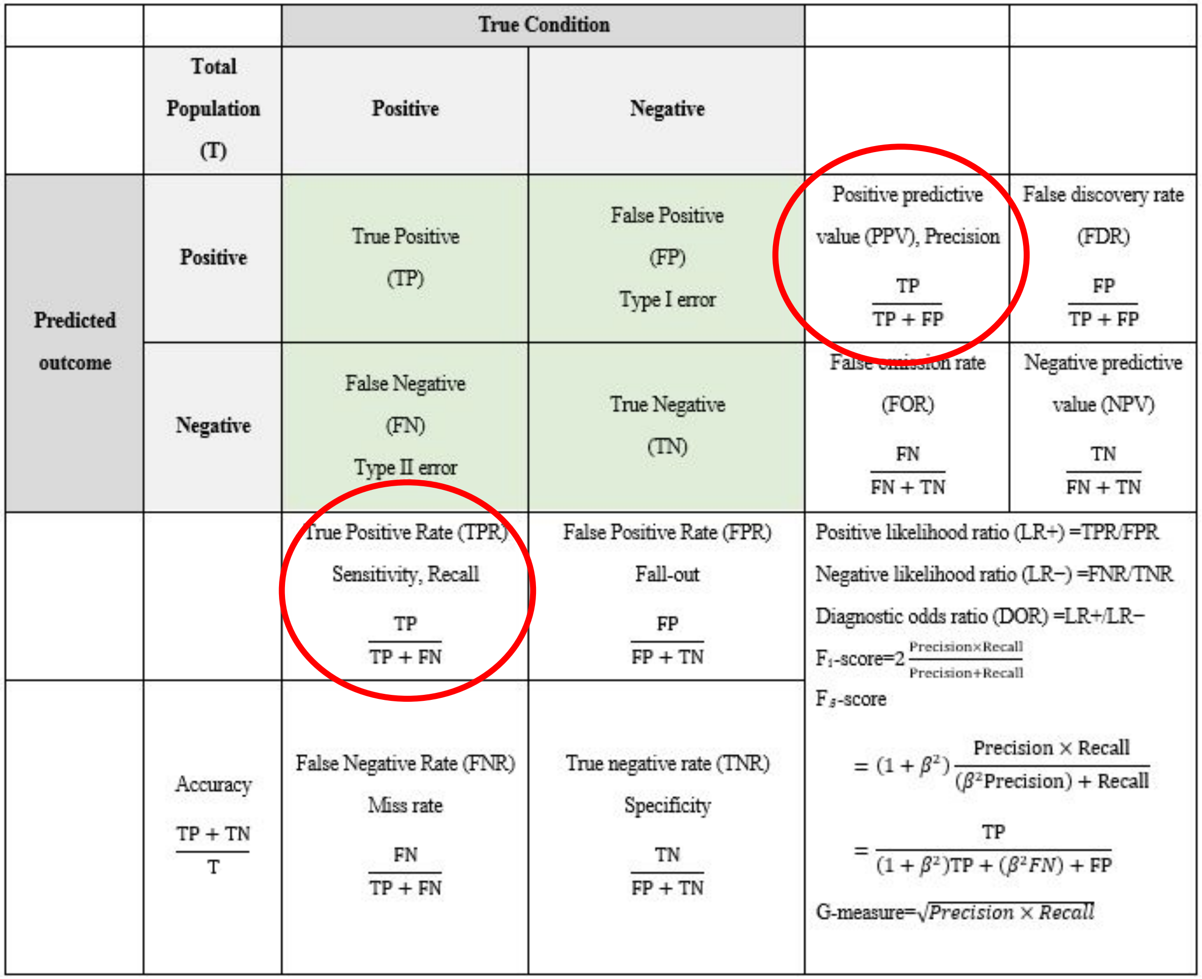
- Object Detection in Aerial Images
  - Task Definition
  - Object Detection Network (Yolo v1)
- Assignment
  - TODO
  - Implementation Details (Data, Loss, IoU, NMS)
  - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Model Evaluation

- Evaluation metric: Choose GT with difficulty = 0 to calculate **mean Average Precision (mAP)**

Precision?

Recall ?

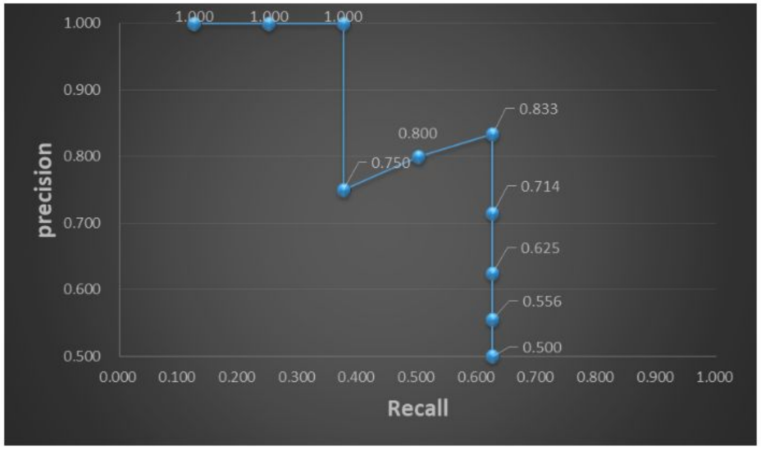


		True Condition			
	Total Population (T)	Positive	Negative		
Predicted outcome	Positive	True Positive (TP)	False Positive (FP) Type I error	Positive predictive value (PPV), Precision $\frac{TP}{TP + FP}$	False discovery rate (FDR) $\frac{FP}{TP + FP}$
	Negative	False Negative (FN) Type II error	True Negative (TN)	False omission rate (FOR) $\frac{FN}{FN + TN}$	Negative predictive value (NPV) $\frac{TN}{FN + TN}$
		True Positive Rate (TPR) Sensitivity, Recall $\frac{TP}{TP + FN}$	False Positive Rate (FPR) Fall-out $\frac{FP}{FP + TN}$	Positive likelihood ratio (LR+) =TPR/FPR Negative likelihood ratio (LR−) =FNR/TNR Diagnostic odds ratio (DOR) =LR+/LR− F_1-score=$2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ F_β-score $= (1+\beta^2)\frac{\text{Precision} \times \text{Recall}}{(\beta^2\text{Precision}) + \text{Recall}}$ $= \frac{TP}{(1+\beta^2)TP + (\beta^2 FN) + FP}$ G-measure=$\sqrt{Precision \times Recall}$	
	Accuracy $\frac{TP + TN}{T}$	False Negative Rate (FNR) Miss rate $\frac{FN}{TP + FN}$	True negative rate (TNR) Specificity $\frac{TN}{FP + TN}$		

Model Evaluation

For a class (ex. Dog), we can calculate its **Precision/Recall** curve with different **confidence score** threshold.

The area under the curve is its **AP** (Average Precision)



precision-recall graph

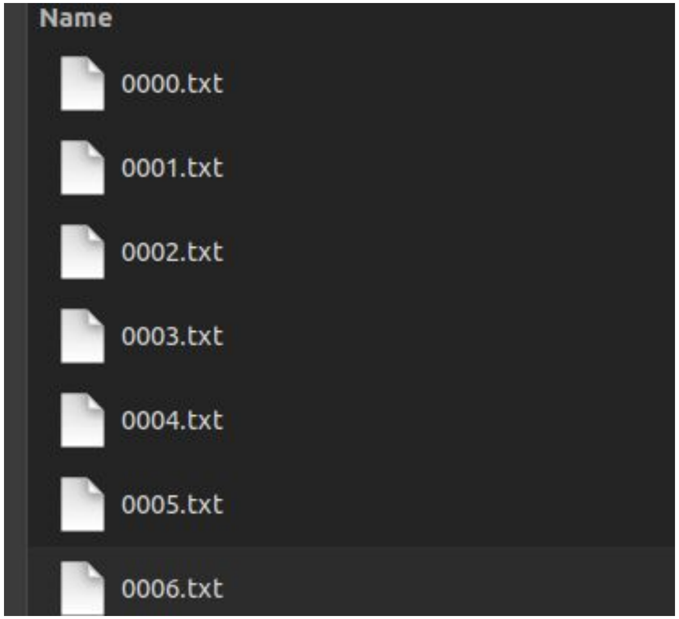
The average AP of **all classes** is the **mean Average Precision (mAP)** !!

Model Evaluation

- Evaluation : hw2_evaluation_task2.py
    - **We have provided the evaluation script for you**
    - Usage:

```
python3 hw2_evaluation_task2.py <PredictionDir>  <AnnotationDir>
```

    - AnnotationDir should be the directory of ground truth dir. (ex. val1500/)
    - PredictionDir should be the directory to put your prediction files.
    - What to put in <PredictionDir>?   Same file name as the GT annotation file !!



Name
0000.txt
0001.txt
0002.txt
0003.txt
0004.txt
0005.txt
0006.txt

Grading Policy

- Baseline (50%)
  - validation set (20%)
  - testing set (30%)
- Report (55%)

Grading Policy

Baseline (50%)

- Implement baseline model **YoloV1-vgg16bn**
- **mAP** score should be above the baseline score

  (valid: 0.086 (8.6%) / test: 0.086 (8.6%))

  - validation set (20%) / test set (30%, only TAs have the test set)
  - Note: baseline credit depends **only on YoloV1-vgg16bn** model.
  - ~~We will announce the baseline score on Fri. (03/29)~~

Grading Policy

Report (55%) (report template is provided in your GitHub repository)

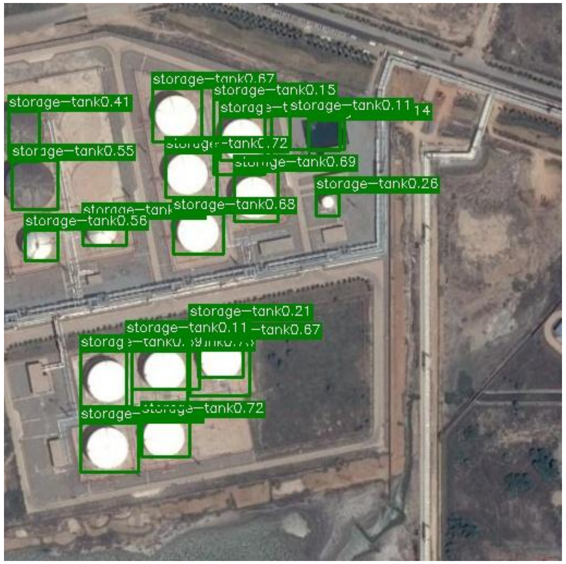
1. (5%) Print the network architecture of your YoloV1-vgg16bn model and describe your training config. (optimizer,batch size….and so on)
2. (10%) Show the predicted bbox image of “val1500/0076.jpg”, “val1500/0086.jpg”, “val1500/0907.jpg” during the early, middle, and the final stage during the training stage. (For example, results of 1st, 10th, 20th epoch)
3. (10%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model and describe it.
4. (10%) Show the predicted bbox image of “val1500/0076.jpg”, “val1500/0086.jpg”, “val1500/0907.jpg” during the early, middle, and the final stage during the training process of this improved model.
5. (15%) Report mAP score of both models on the validation set. Discuss the reason why the improved model performs better than the baseline one. You may conduct some experiments and show some evidences to support your reasoning.
6. **bonus (5%)** Which classes prediction perform worse than others? Why? You should describe and analyze it.

Don’t Worry~

We provide the visualization code ><

How to run:

```
$  python3 visualize_bbox.py <image.jpg> <label.txt>
```


storage−tank0.67
storage−tank0.15
storage−tank0.41
storage−tank0.11
14
storage−tank0.72
storage−tank0.55
storage−tank0.69
storage−tank0.26
storage−tank0.68
storage−tank0.56
storage−tank0.21
storage−tank0.11
−tank0.67
storage−tank0.72

Outline

- Object Detection in Aerial Images
    - Task Definition
    - Object Detection Network (Yolo v1)
- Assignment
    - TODO
    - Implementation Details (Data, Loss, IoU, NMS)
    - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Deadline & Homework Policy

- Report and source code deadline : **2019/04/17 (Wed.) 01:00 AM (GMT+8)**

- Late policy : Up to **5** free late days in a semester. After that, late homework will be deducted 30% each day.

- Taking any unfair advantages over other class members (or letting anyone do so) is strictly prohibited. Violating university policy would result in F for this course.
- If you refer to some parts of the public code, you have to put your reference in your report.

- Students are encouraged to discuss the homework assignments, but you must complete the assignment by yourself. TA will compare the similarity of everyone’s homework. Any form of cheating or plagiarism will not be tolerated, which will also result in F for students with such misconduct.

Homework Policy - Submission

- Click the following link and sign in to your GitHub account to get your submission repository (if you don't have an account, you need to create one first):

  https://classroom.github.com/a/EA-0ohqz

  When asked, make sure that you select the correct name/student ID. If you don't find your name/student ID in the list, please contact TAs.

- After getting your GitHub repository (which should be named "hw2-<username>"), be sure to **read the README carefully** before starting your work.

Homework Policy - Submission

- Your GitHub repository should include the following files:
    - hw2_<studentID>.pdf
    - hw2.sh (for running YoloV1-vgg16bn model)
    - hw2_best.sh (for running improved model)
    - your python files (train.py and others)
    - your model files (can be loaded by your python file)
- **Don’t upload your dataset.**
- If any of the file format is wrong, you will get zero point.
- If your model is larger than GitHub’s maximum capacity (100MB), you can upload your model to another cloud service (e.g., Dropbox). However, you script file should be able to download the model automatically. (Dropbox tutorial: link)

Homework Policy - Bash Script (We only run testing)

- TA will run your code as shown below:
  - bash hw2.sh $1 $2
  - bash hw2_best.sh $1 $2
    - $1: testing images directory (ex. ../test1500/images/)
    - $2: output prediction directory (ex. ./Test_hbb/)
- You **should** generate the prediction files under $2 with the same names of its corresponding images but the extension is “.txt”.
- Note that you should **NOT** hard code any path in your file or script
- Your testing code have to be finished in **10 mins.**

Homework Policy - Packages

- Python : 3.5+
- Tensorflow : 1.13
- Keras : 2.2+
- Pytorch : 1.0
- h5py : 2.9.0
- Numpy : 1.16.2
- Pandas : 0.24.0
- torchvision, open-cv, Matplotlib, Scikit-image, Pillow, Scipy, Python standard Lib.
- **E-mail or ask TA first if you want to import other packages.**

Outline

- Object Detection in Aerial Images
    - Task Definition
    - Object Detection Network (Yolo v1)
- Assignment
    - TODO
    - Implementation Details (Data, Loss, IoU, NMS)
    - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

Tutorial & Documentation

- Yolo v1 paper

  https://pjreddie.com/media/files/papers/yolo_1.pdf

- Yolo 詳解

  https://blog.csdn.net/c20081052/article/details/80236015

- NMS

  https://medium.com/@chih.sheng.huang821/機器-深度學習-物件偵測-non-maximum-suppression-nms-aa70c45adffa

- AP, IoU

  https://medium.com/@chih.sheng.huang821/深度學習系列-什麼是ap-map-aaf089920848

Outline

- Object Detection in Aerial Images
  - Task Definition
  - Object Detection Network (Yolo v1)
- Assignment
  - TODO
  - Implementation Details (Data, Loss, IoU, NMS)
  - Model Evaluation & Grading Policy
- Deadline & Homework Policy
- Tutorial & Documentation
- Q & A

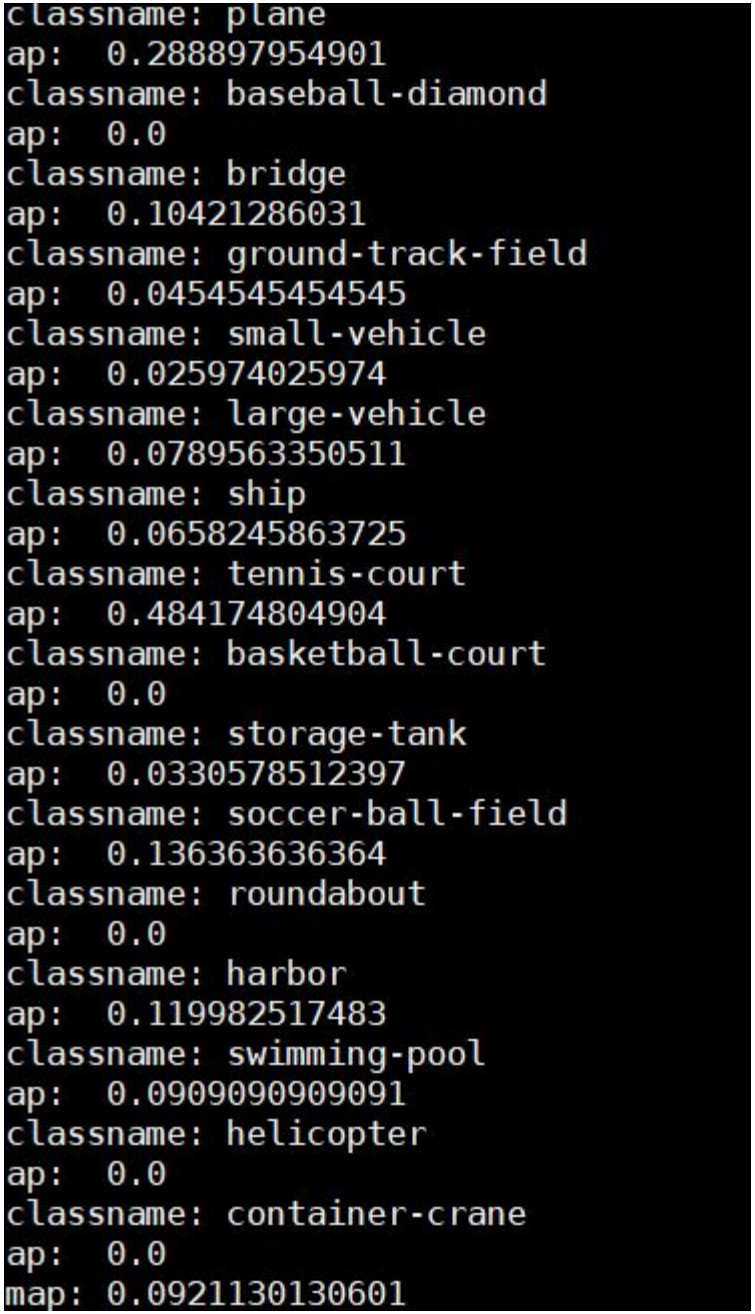
Q & A

- If you have any question, you can:
    - Use TA hours (please check course website for time/location)
    - Contact TAs by e-mail (ntudlcvta2019@gmail.com)
    - Post your question under hw2 FAQ section in FB group
    - Useful website: link
    - **DO NOT** directly message TAs (we will ignore any direct message)

Note

1. Be careful about DataLoader
   → You should visualize the bbox and **class** after you parse your data.
2. The decreasing loss may not guarantee that you train it successfully.
3. TA’s baseline config.
   a. 1080Ti, Batchsize=16 → 4~6 hours.
   b. GPU Memory : 6GB
   c. training loss : 1~3
   d. Drouput after 4096 FC
   e. lr = 0.001 → after 30 epochs = 0.0001 → after 40 epochs = 0.00001
4. Check your NMS. (visualize before and after)
5. Late policy : Up to **5** free late days in a semester. After that, late homework will be deducted 30% each day.
6. AP of each classes are in next page.

AP of each classes from the baseline model trained by TA.



```
classname: plane
ap:  0.288897954901
classname: baseball-diamond
ap:  0.0
classname: bridge
ap:  0.10421286031
classname: ground-track-field
ap:  0.0454545454545
classname: small-vehicle
ap:  0.025974025974
classname: large-vehicle
ap:  0.0789563350511
classname: ship
ap:  0.0658245863725
classname: tennis-court
ap:  0.484174804904
classname: basketball-court
ap:  0.0
classname: storage-tank
ap:  0.0330578512397
classname: soccer-ball-field
ap:  0.136363636364
classname: roundabout
ap:  0.0
classname: harbor
ap:  0.119982517483
classname: swimming-pool
ap:  0.0909090909091
classname: helicopter
ap:  0.0
classname: container-crane
ap:  0.0
map: 0.0921130130601
```