You Only Look

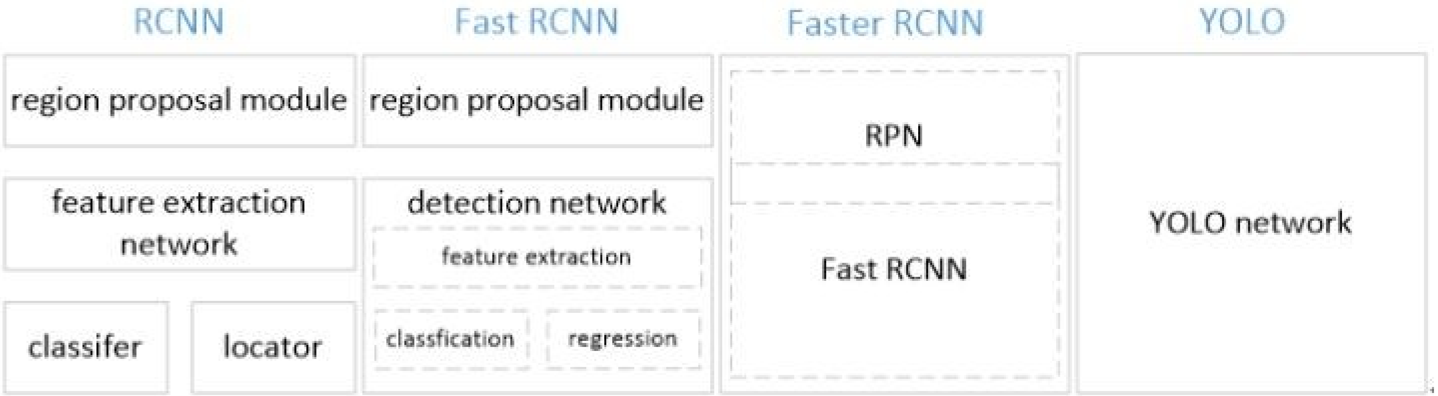
YOLO v1 [illegible]

@[illegible]

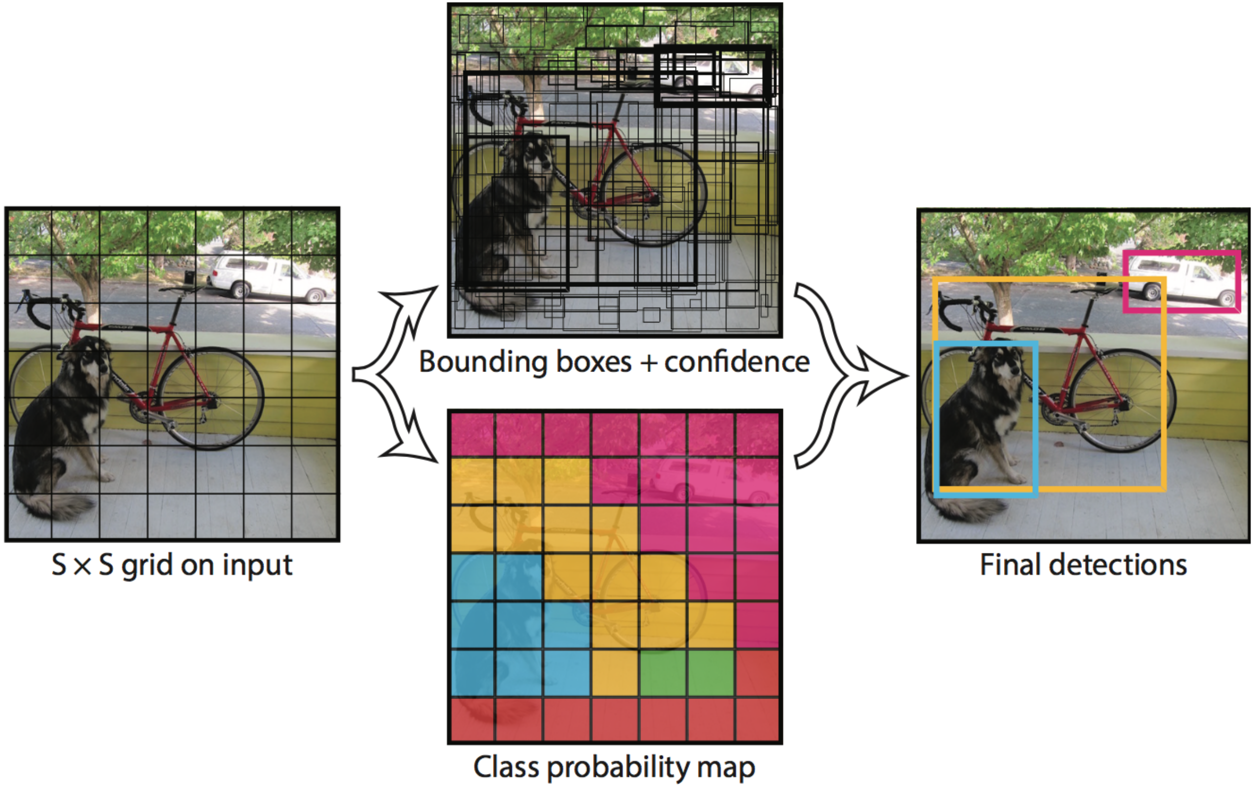
- [illegible]
- Training
- Predicting
- Non-max Suppression
- Darknet native on Windows

1.Review

□□□□



YOLO □□□□□□□□□□□□□□□□□□
YOLO □□□□□□□ region proposal □□□□


Bounding boxes + confidence
S × S grid on input
Final detections
Class probability map

2.Training

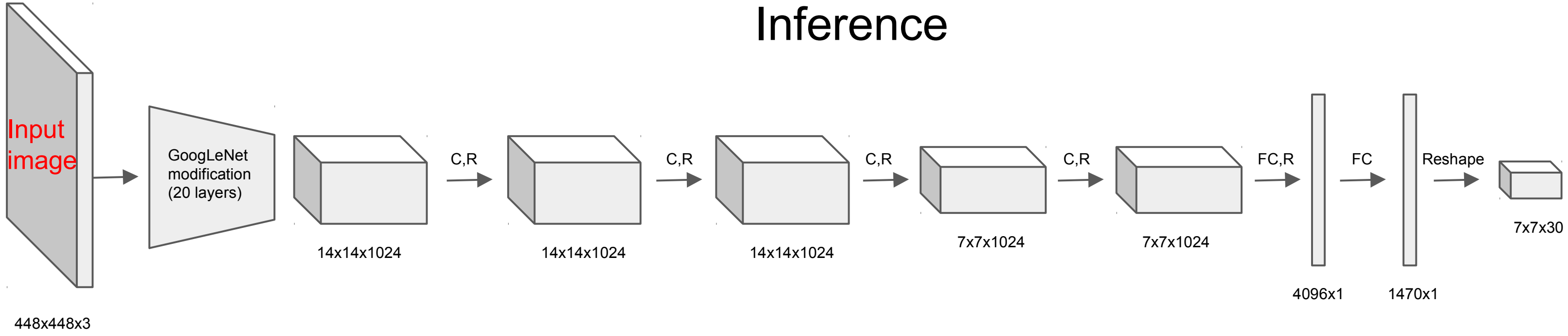
Обзор от deepsystems.io

YOLO

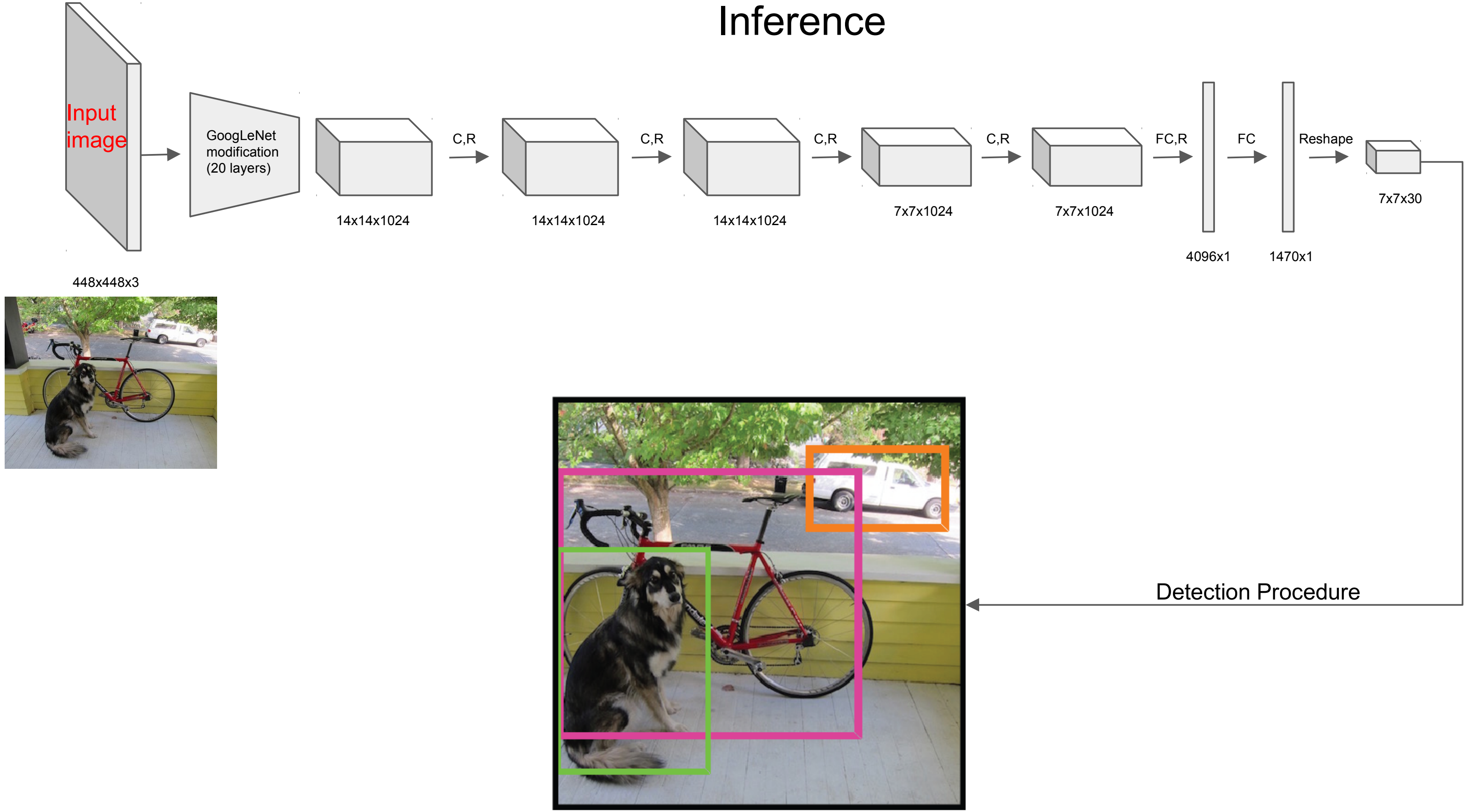
You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

Inference

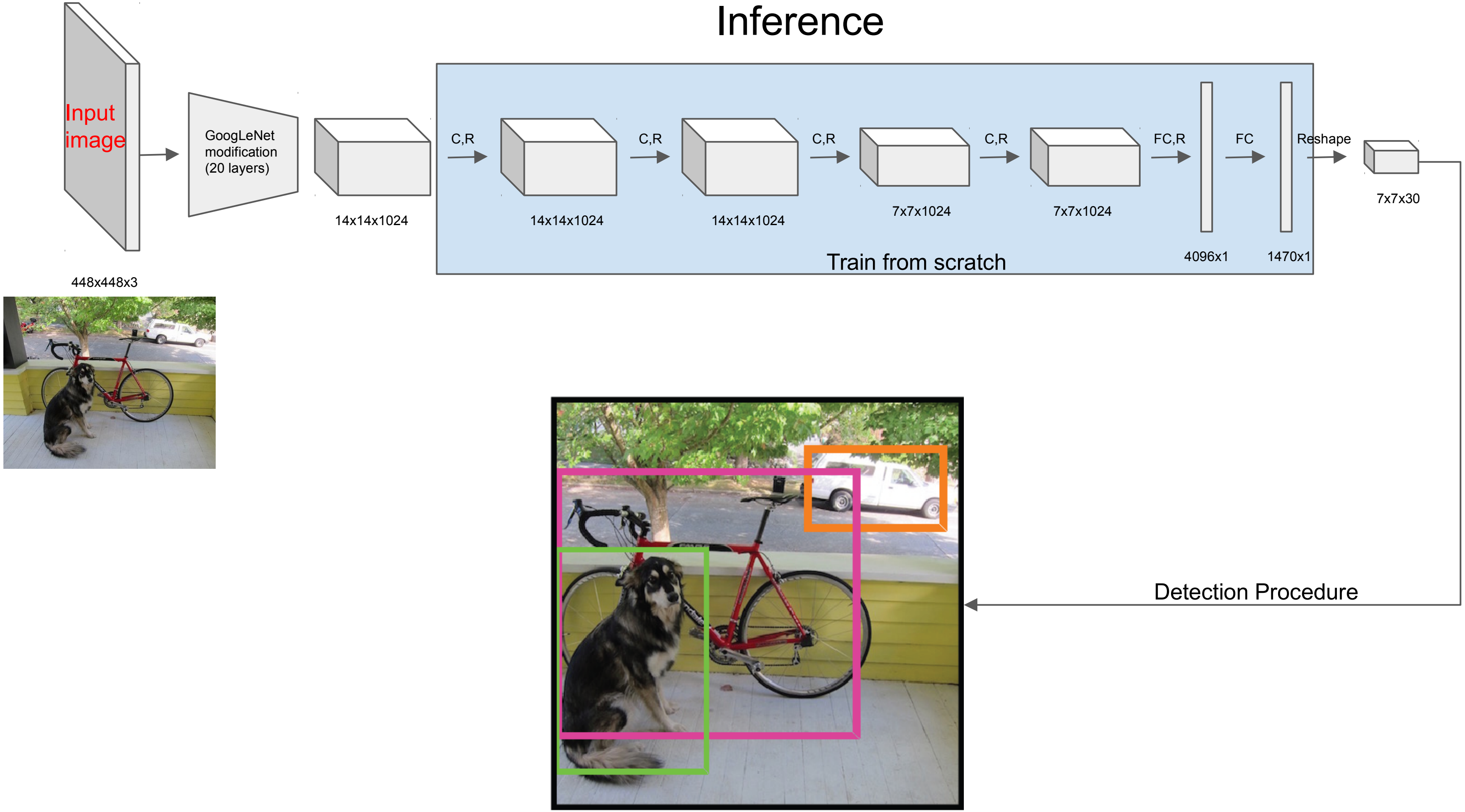

Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30

Inference



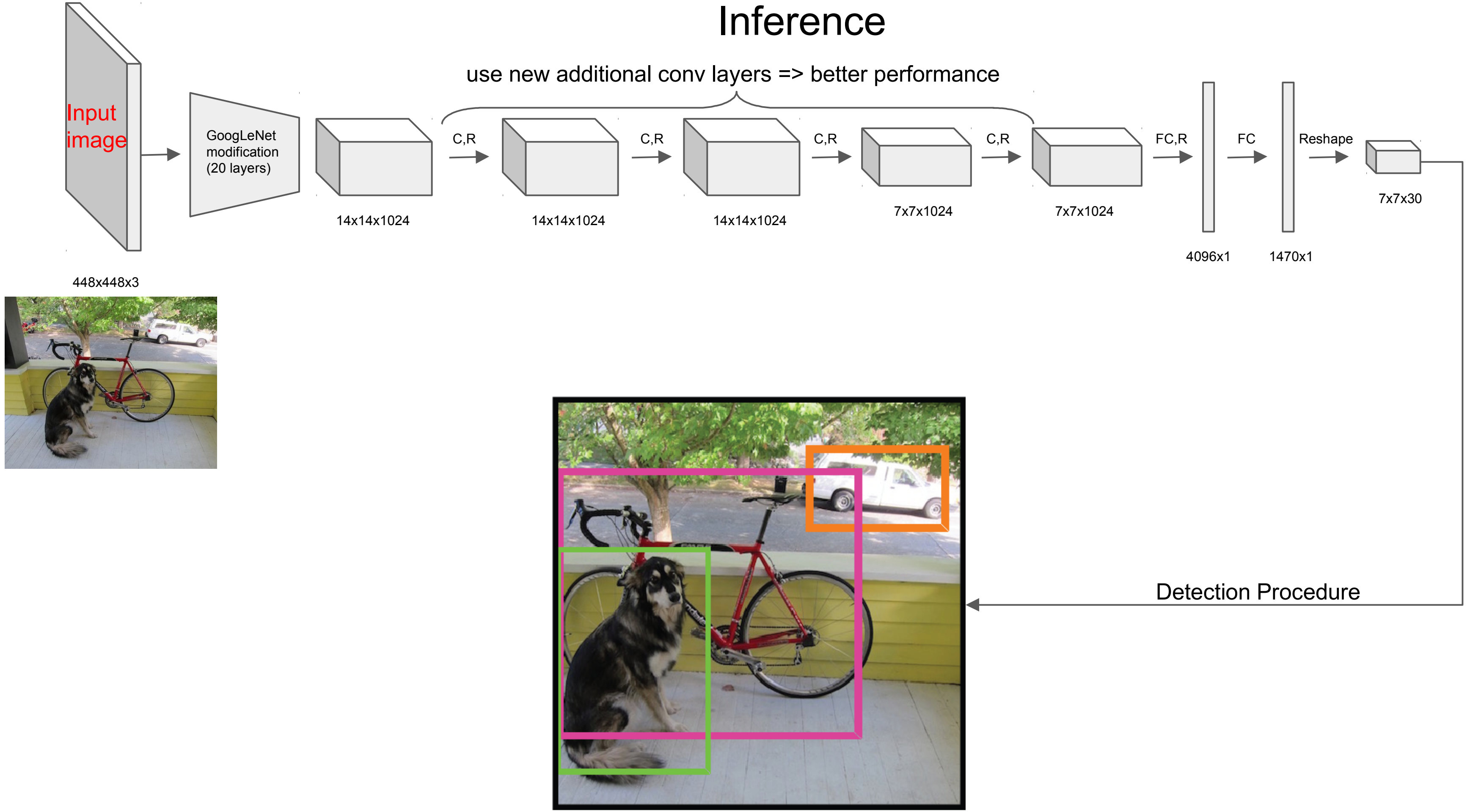
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure

Inference



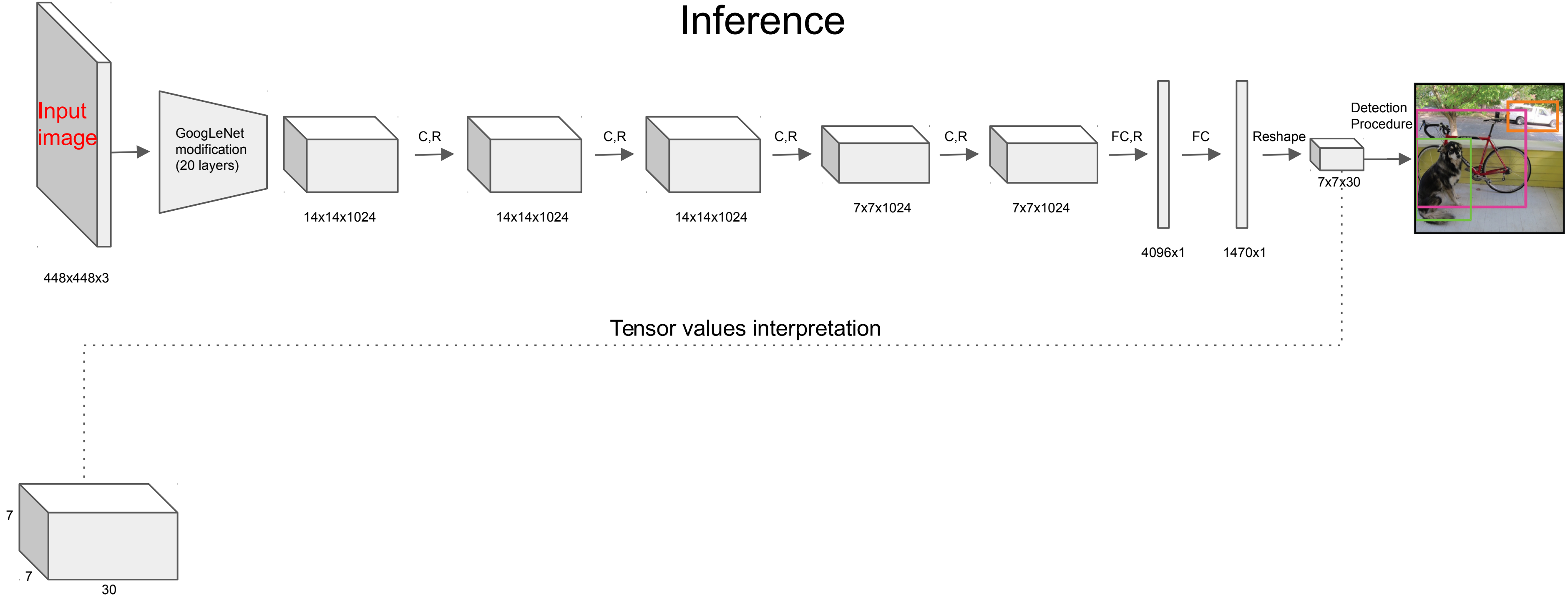
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Train from scratch
Detection Procedure

Inference

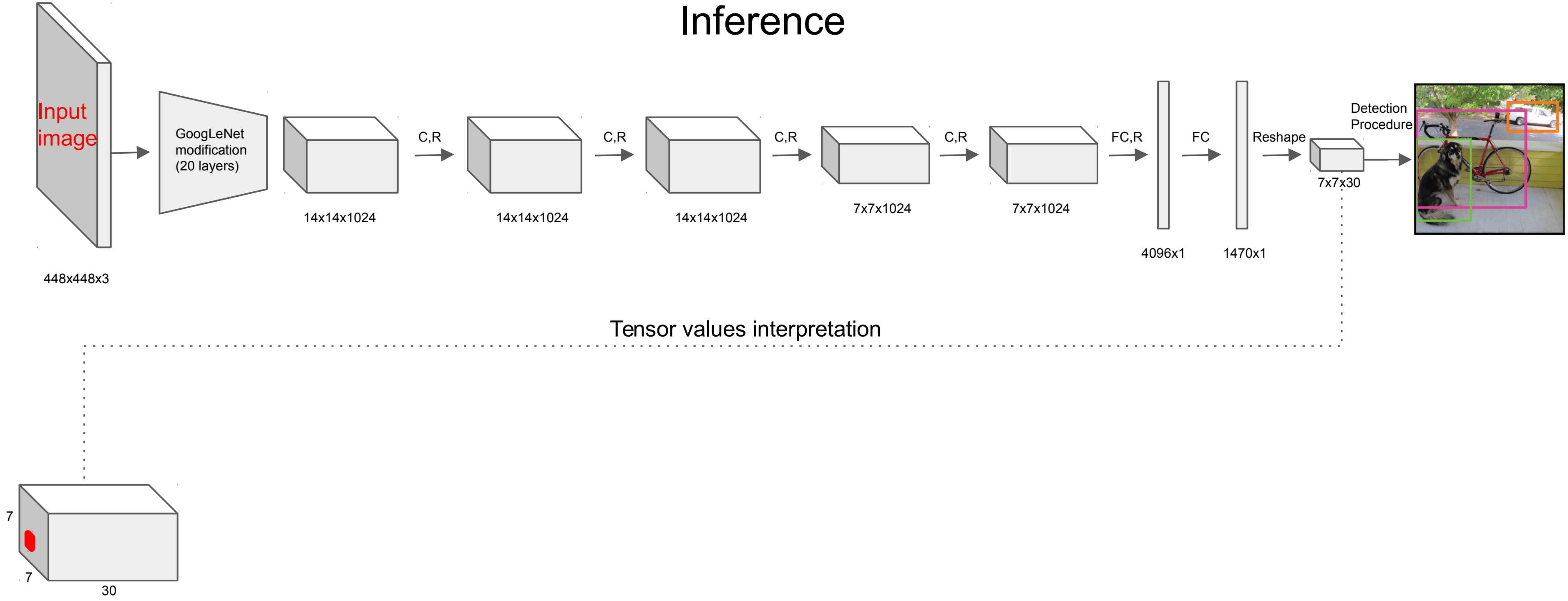


deepsystems.io

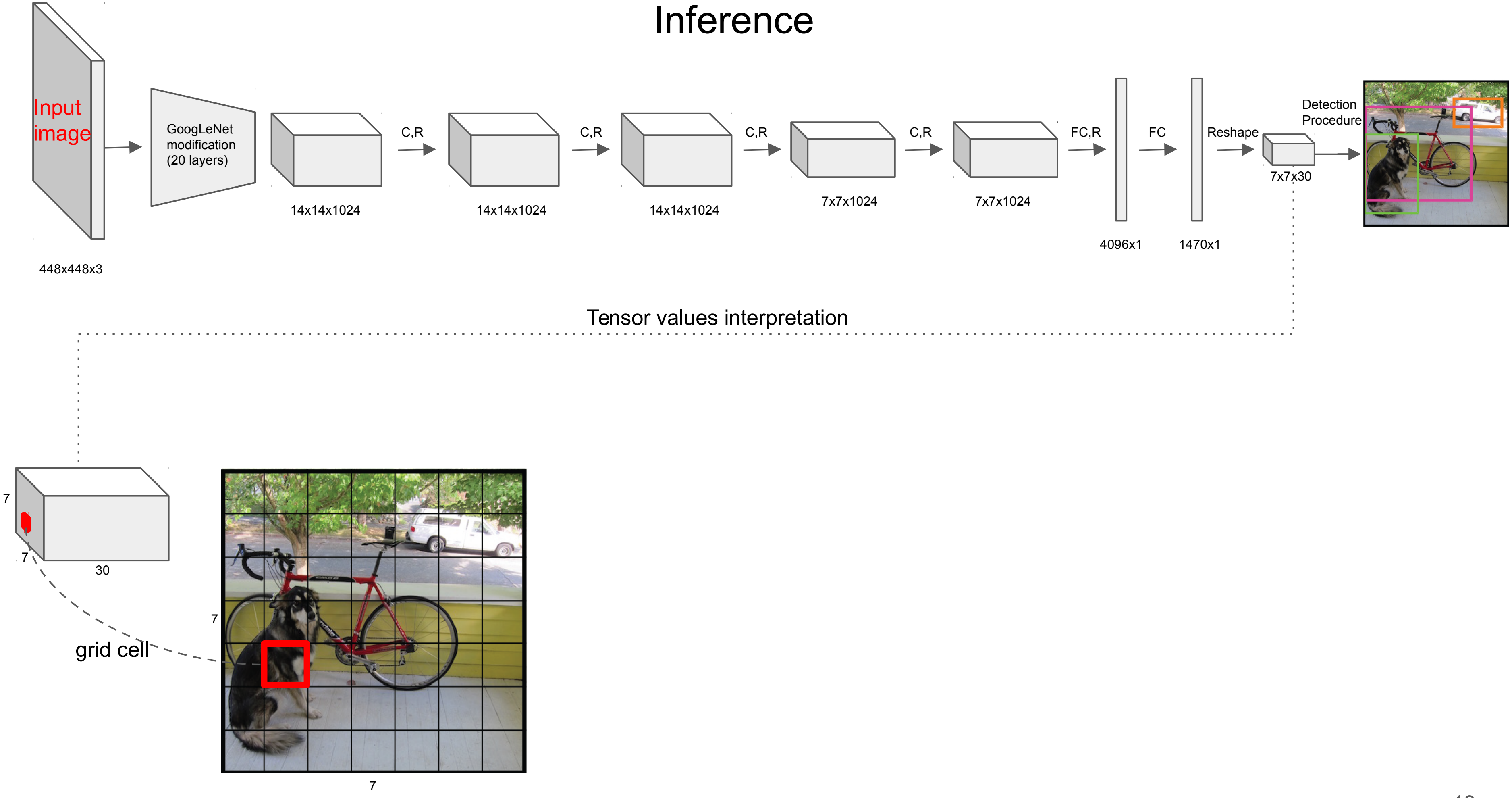
Inference


Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
7
7
30

Inference

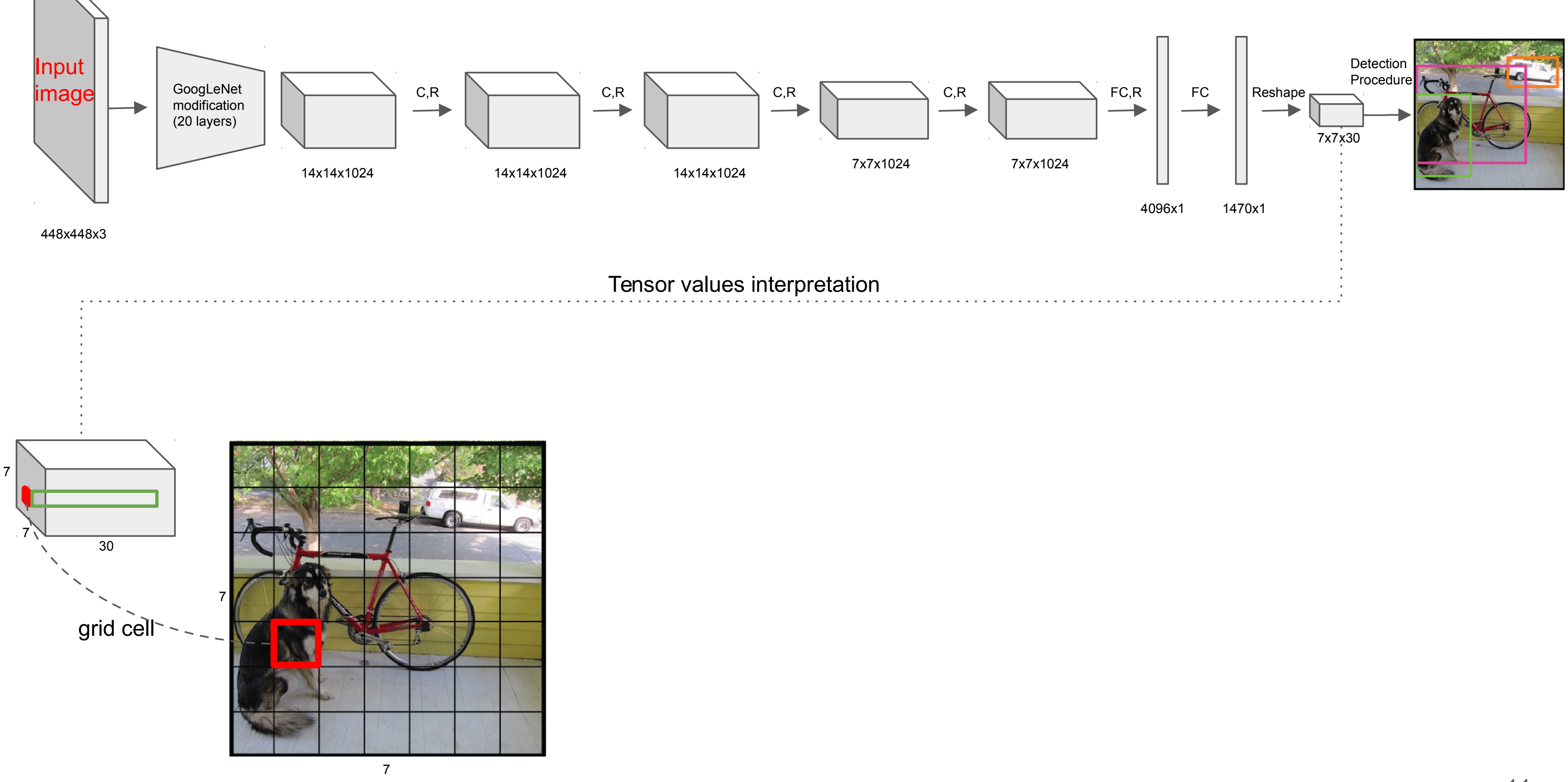

Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
7
7
30

Inference

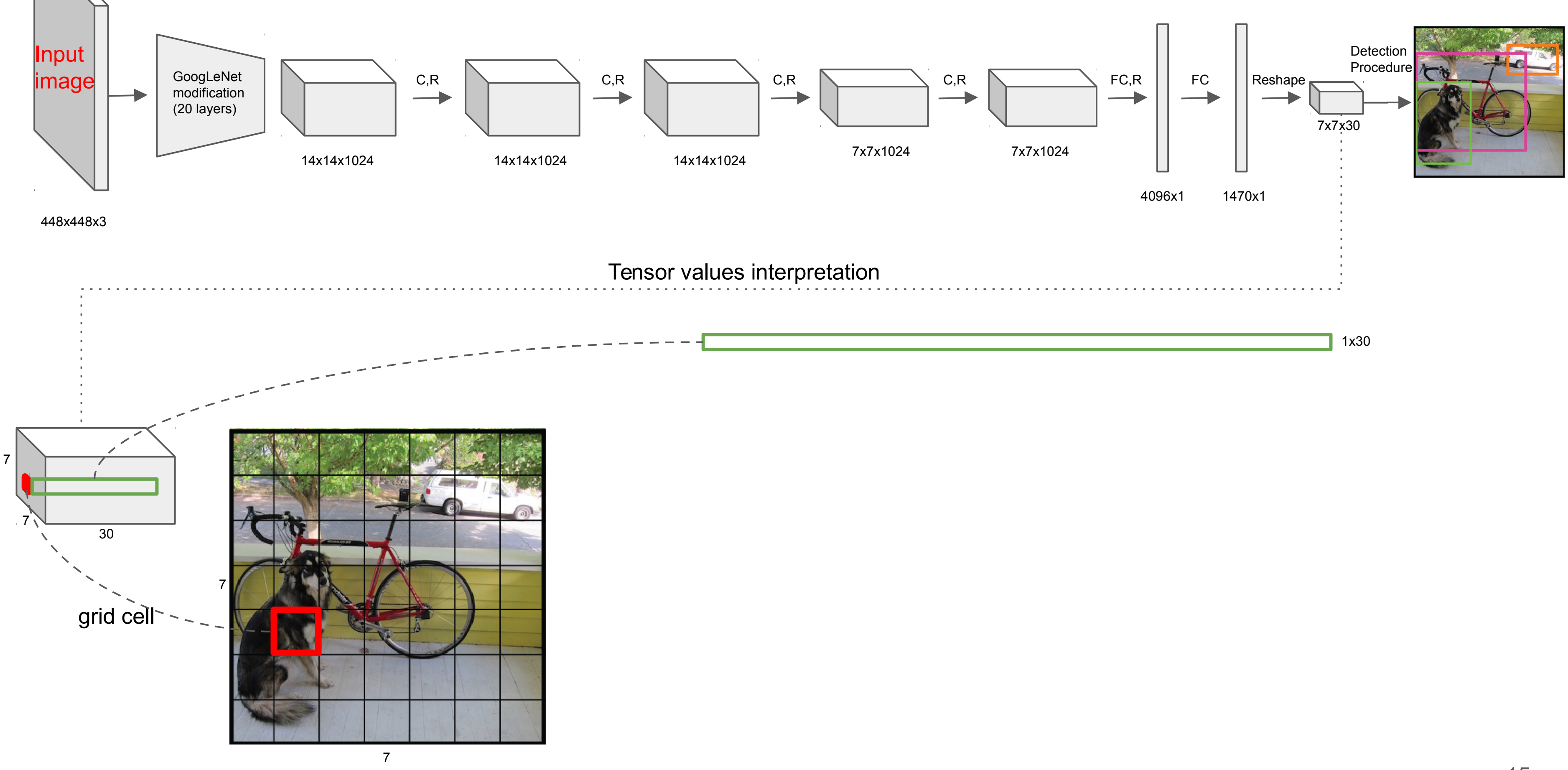


Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
7
7
30
grid cell
7
7

Inference

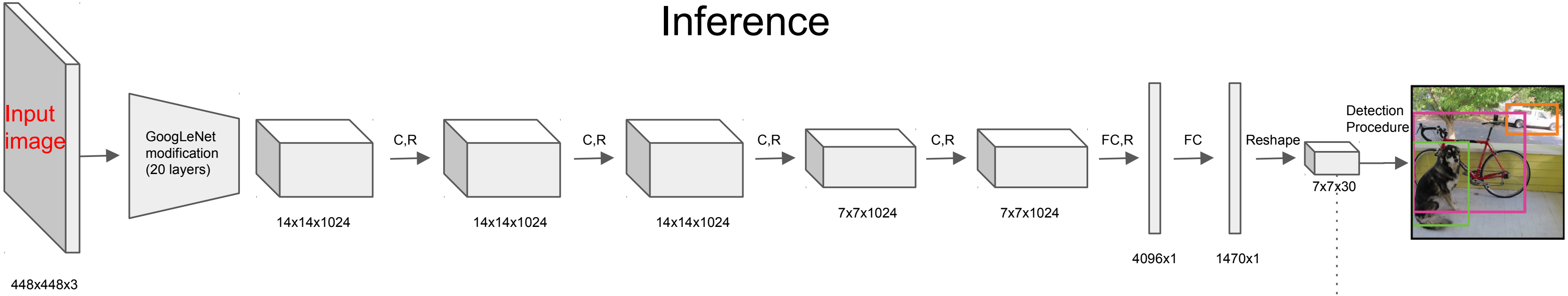

Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
7
7
30
grid cell
7
7

Inference

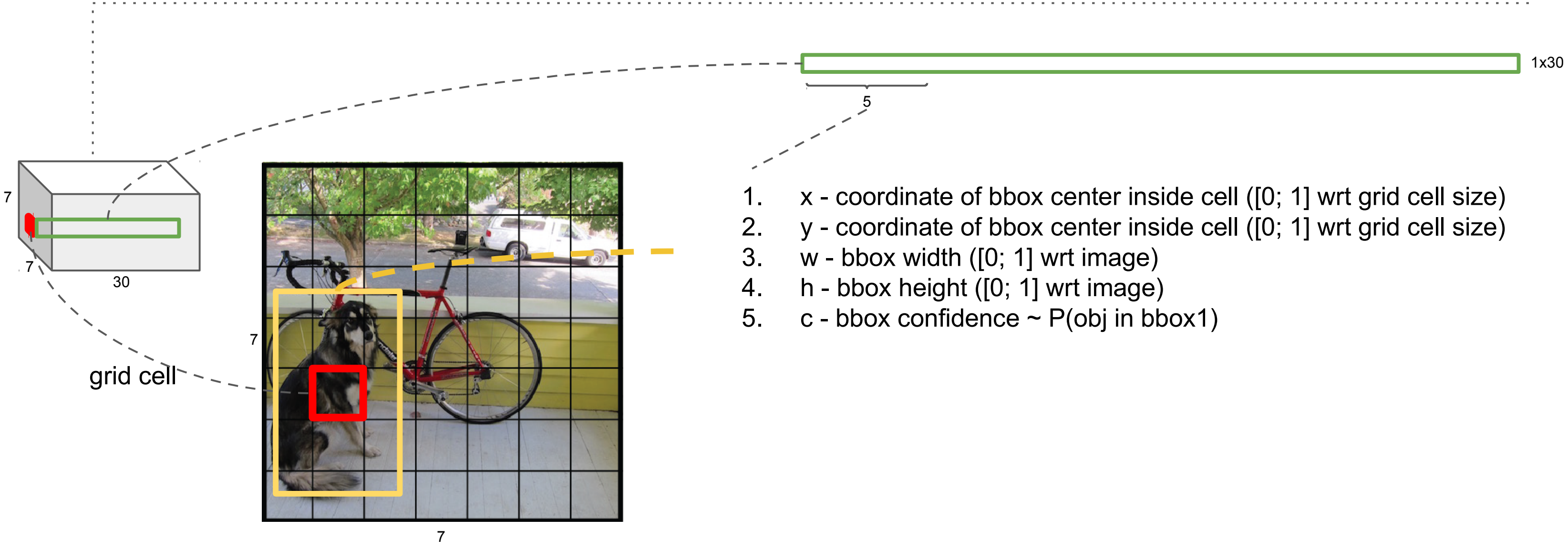


Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
1x30
7
7
30
grid cell
7
7

Inference

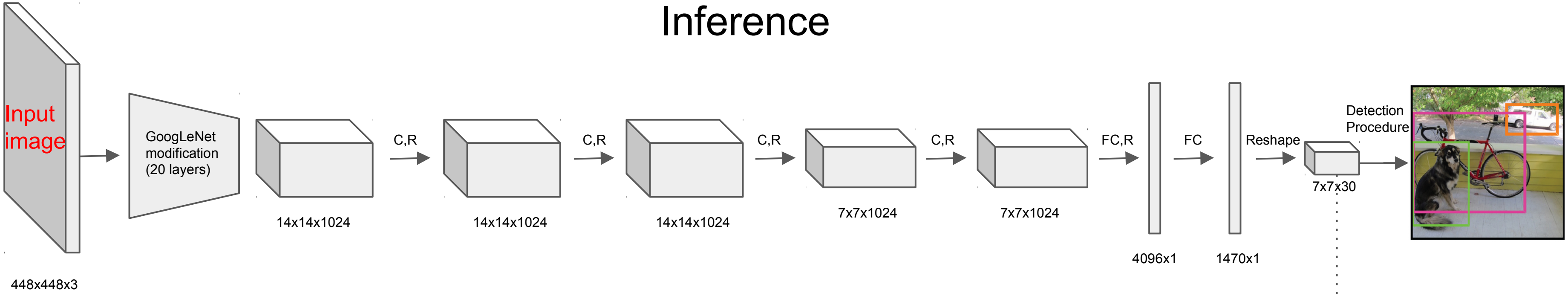


Tensor values interpretation

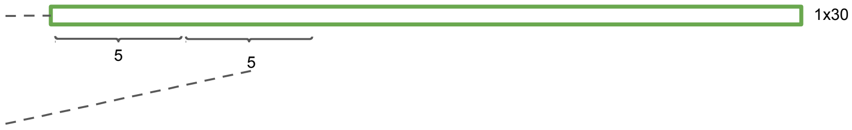


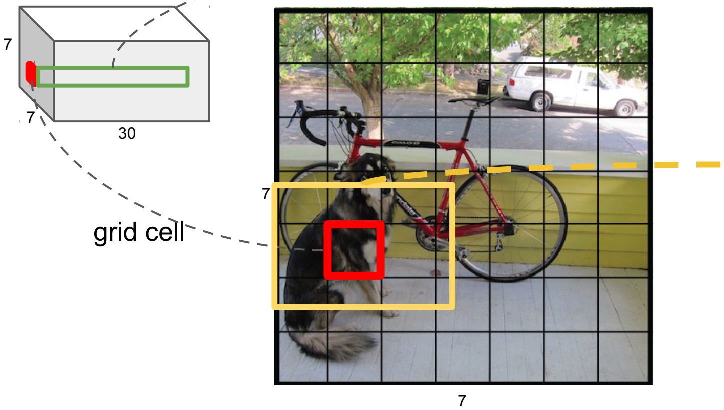
1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox confidence ~ P(obj in bbox1)

Inference



Tensor values interpretation

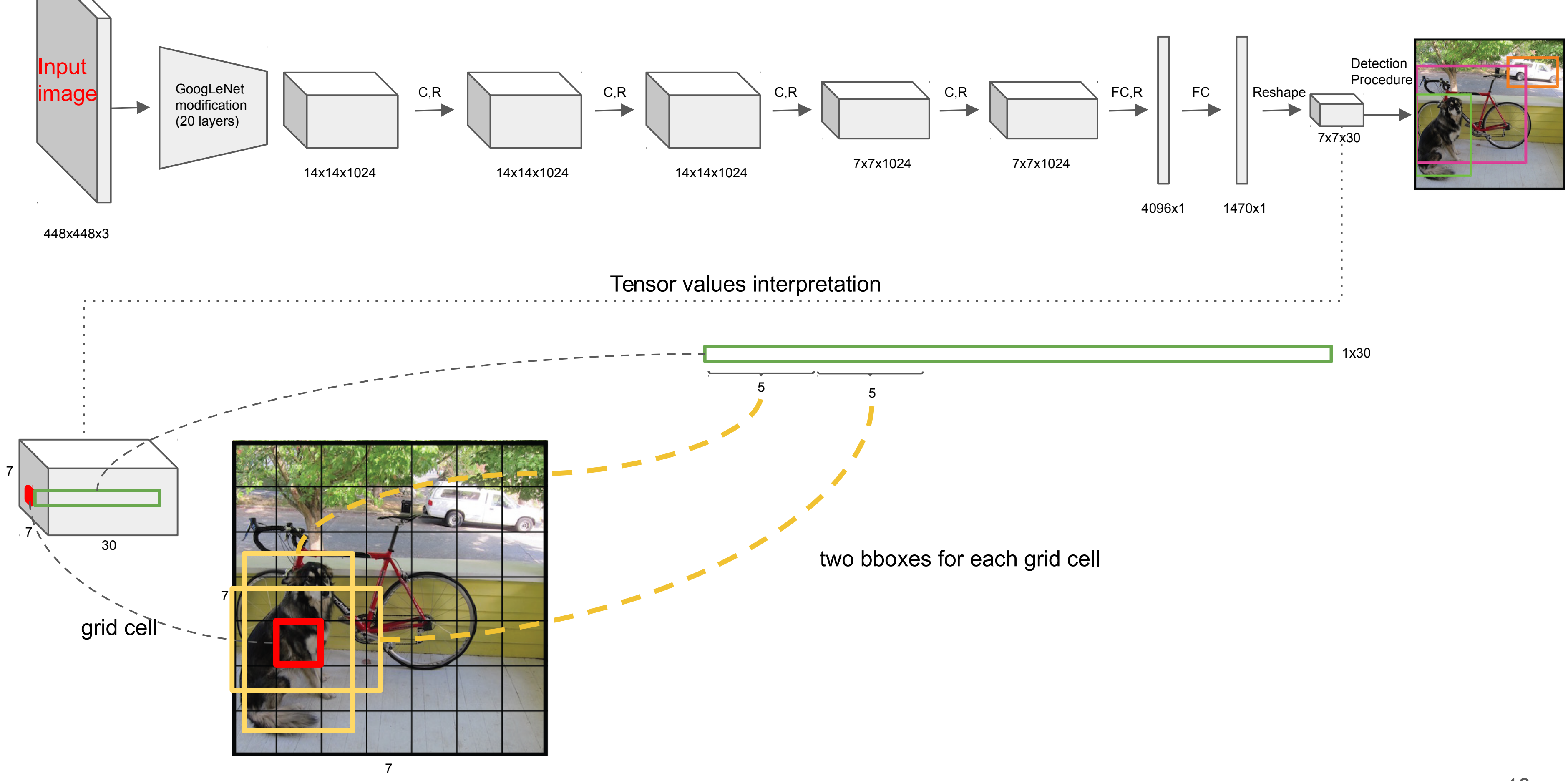




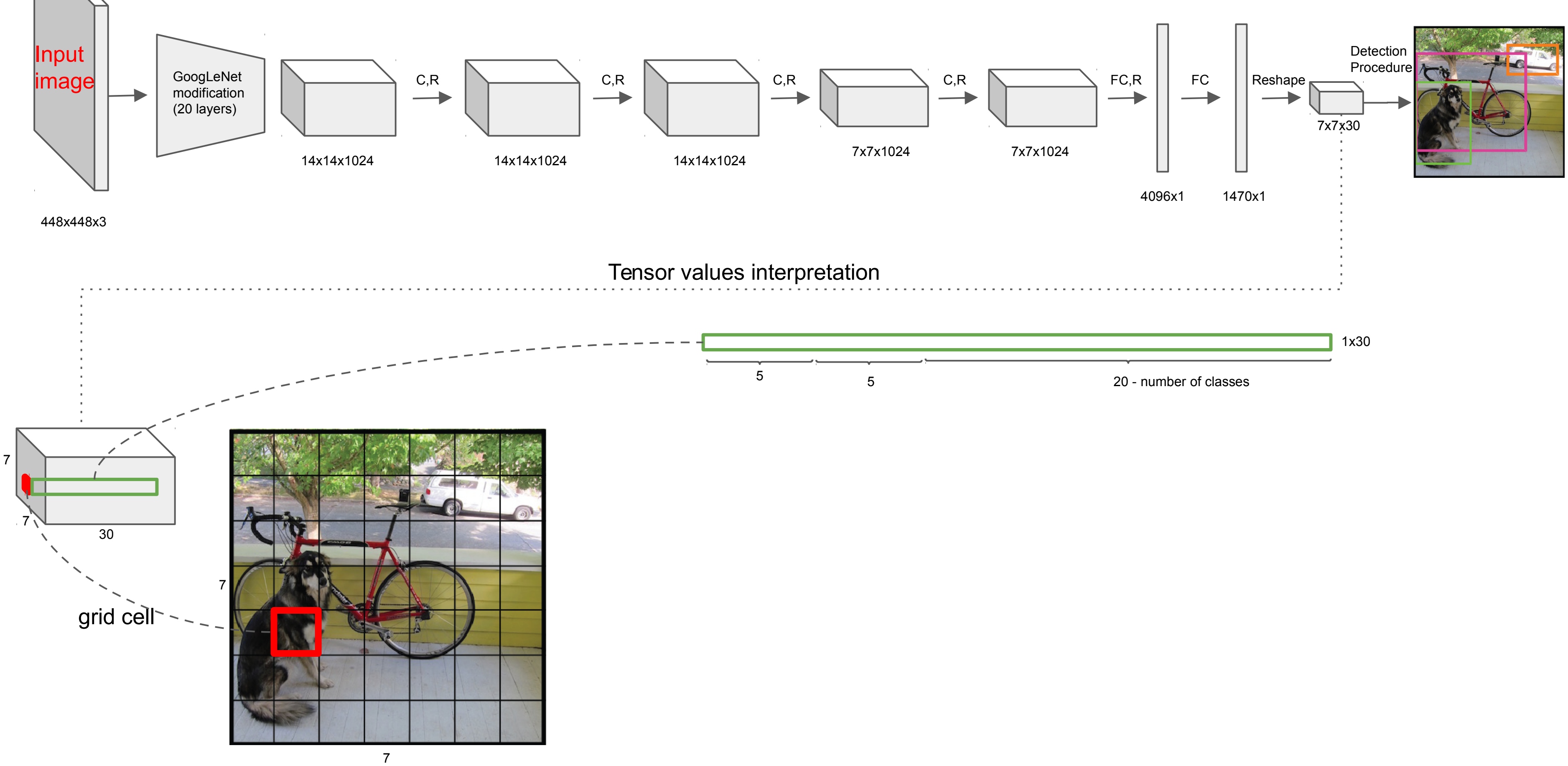
1. x - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
2. y - coordinate of bbox center inside cell ([0; 1] wrt grid cell size)
3. w - bbox width ([0; 1] wrt image)
4. h - bbox height ([0; 1] wrt image)
5. c - bbox **confidence**

$$confidence = Pr(Object) \times IoU_{pred}^{truth}$$

Inference


Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
1x30
5
5
7
7
30
grid cell
7
7
two bboxes for each grid cell

Inference

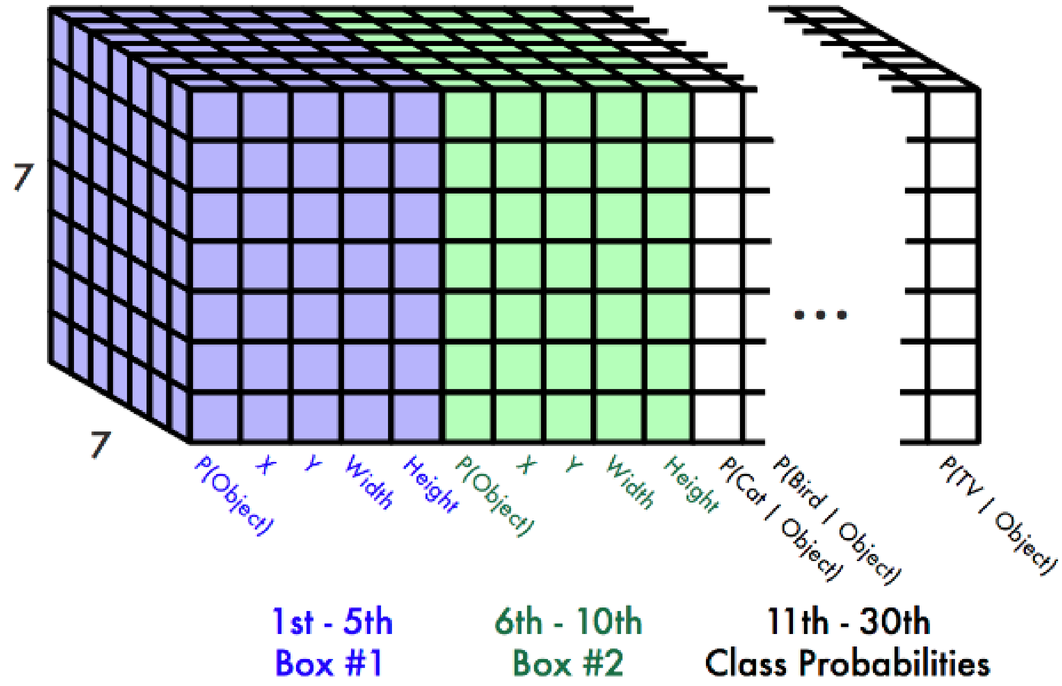


Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
1x30
5
5
20 - number of classes
7
7
30
grid cell
7
7

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



For Pascal VOC:

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

7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = **1470 outputs**

Loss

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

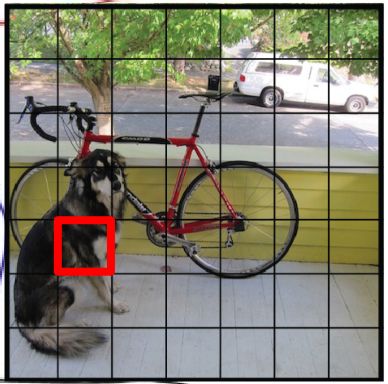
$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

坐标误差

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

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

IOU误差

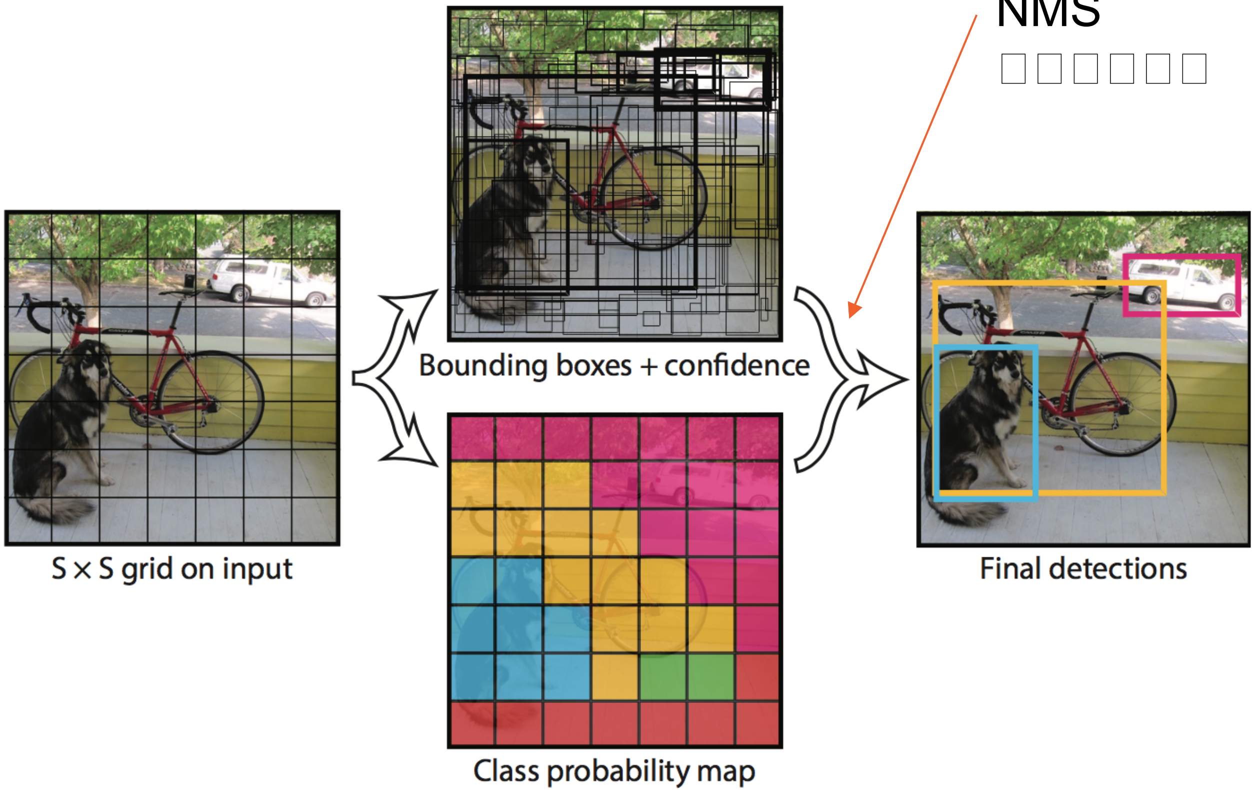


$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \textrm{classes}} (p_i(c) - \hat{p}_i(c))^2$$

分类误差

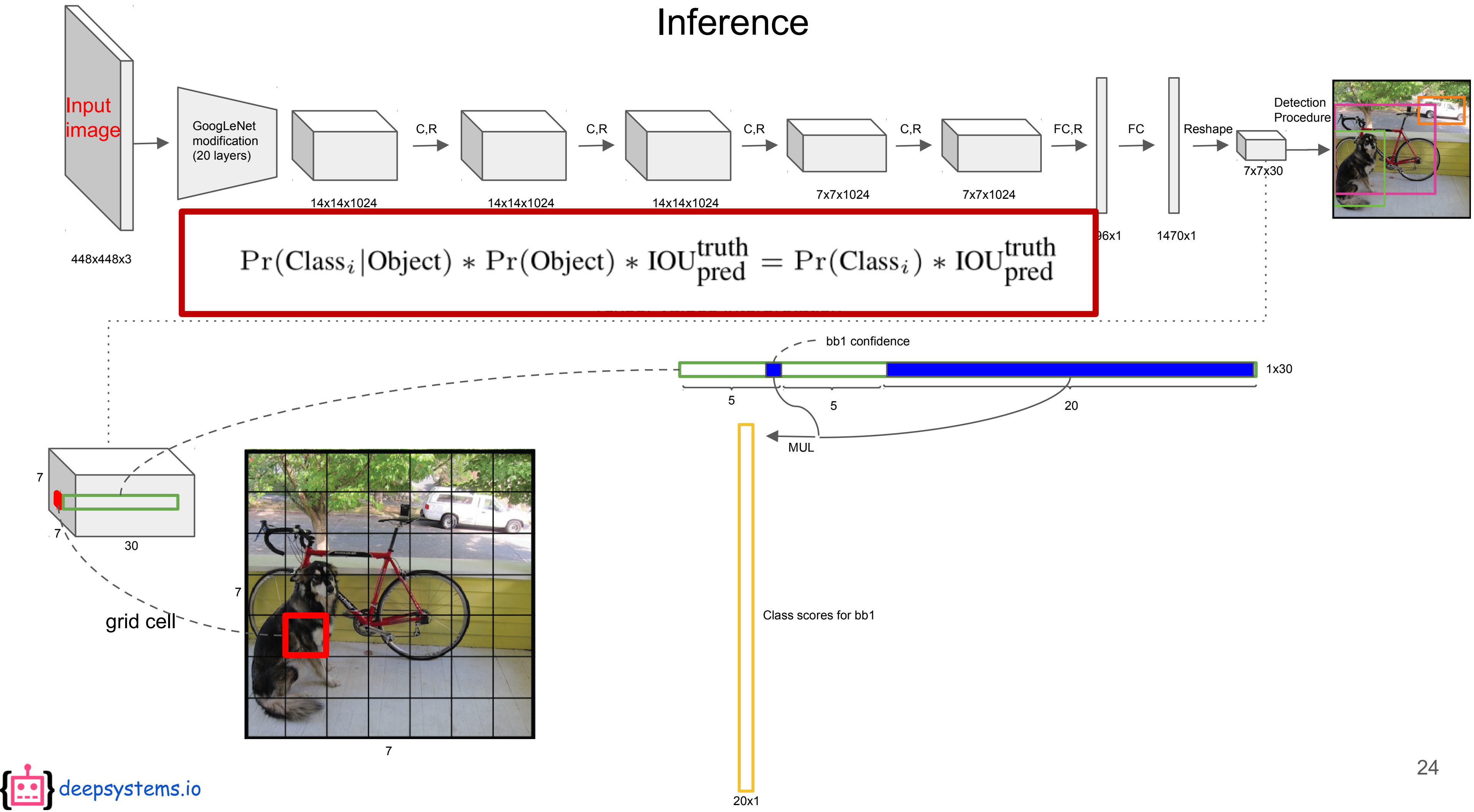
3.Testing

Non-max suppression
NMS

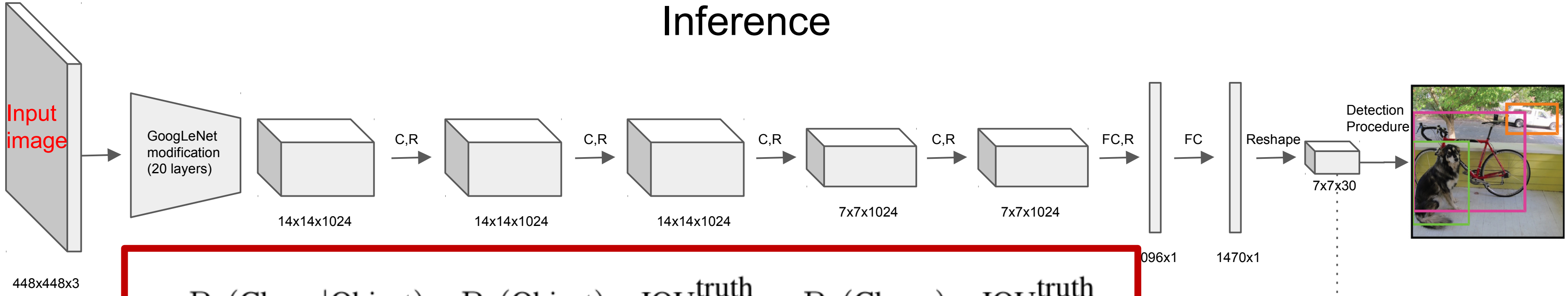


Inference

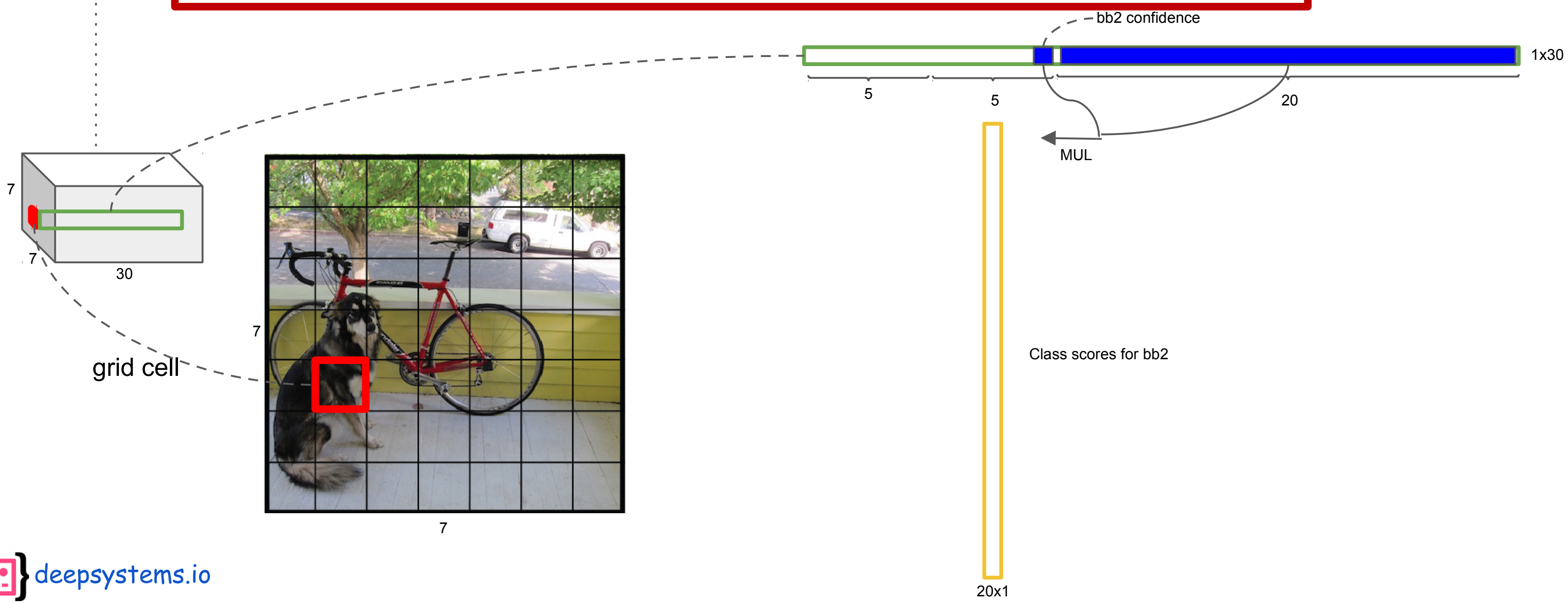
$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$



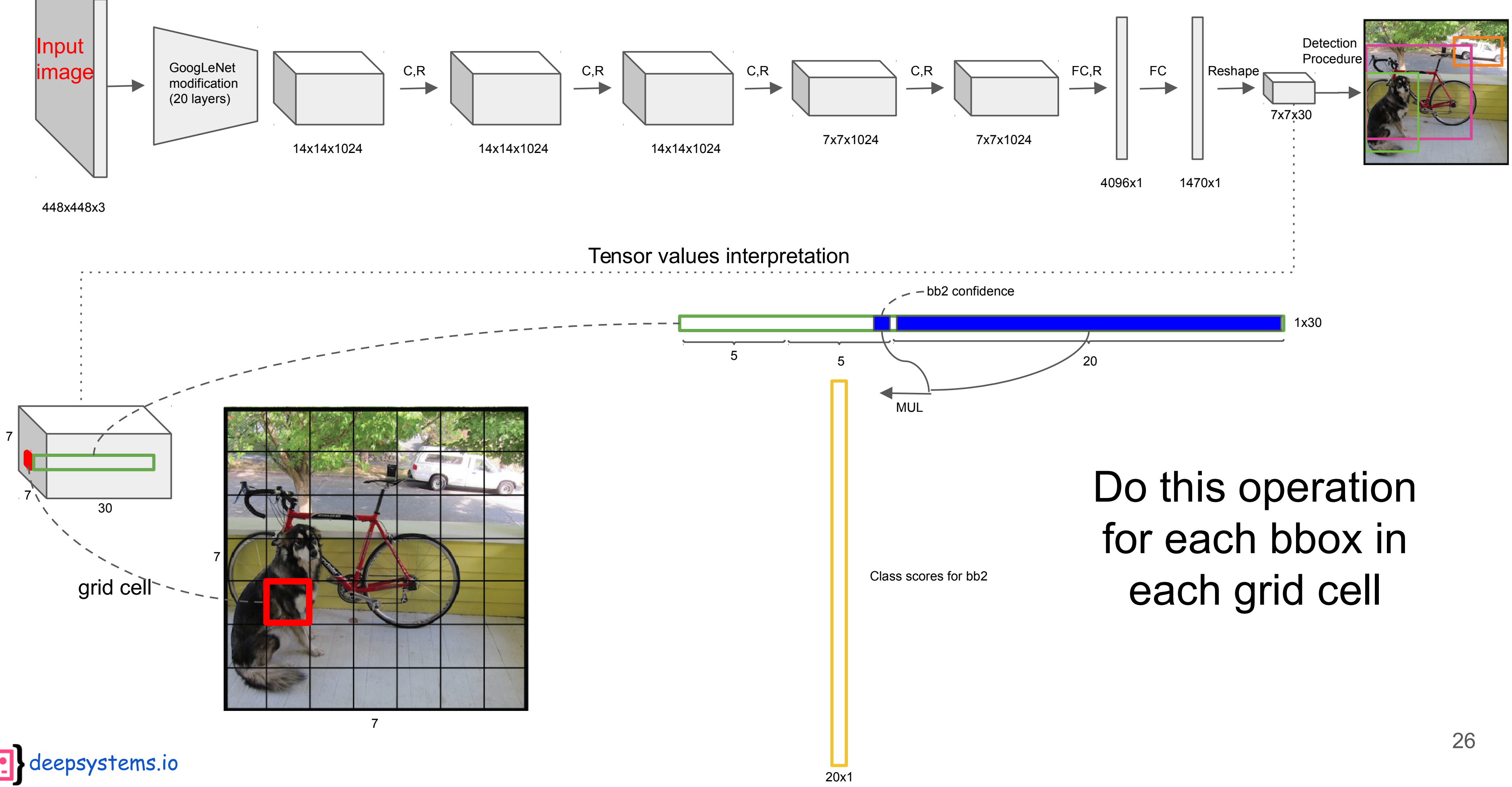
Inference



$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}}$$



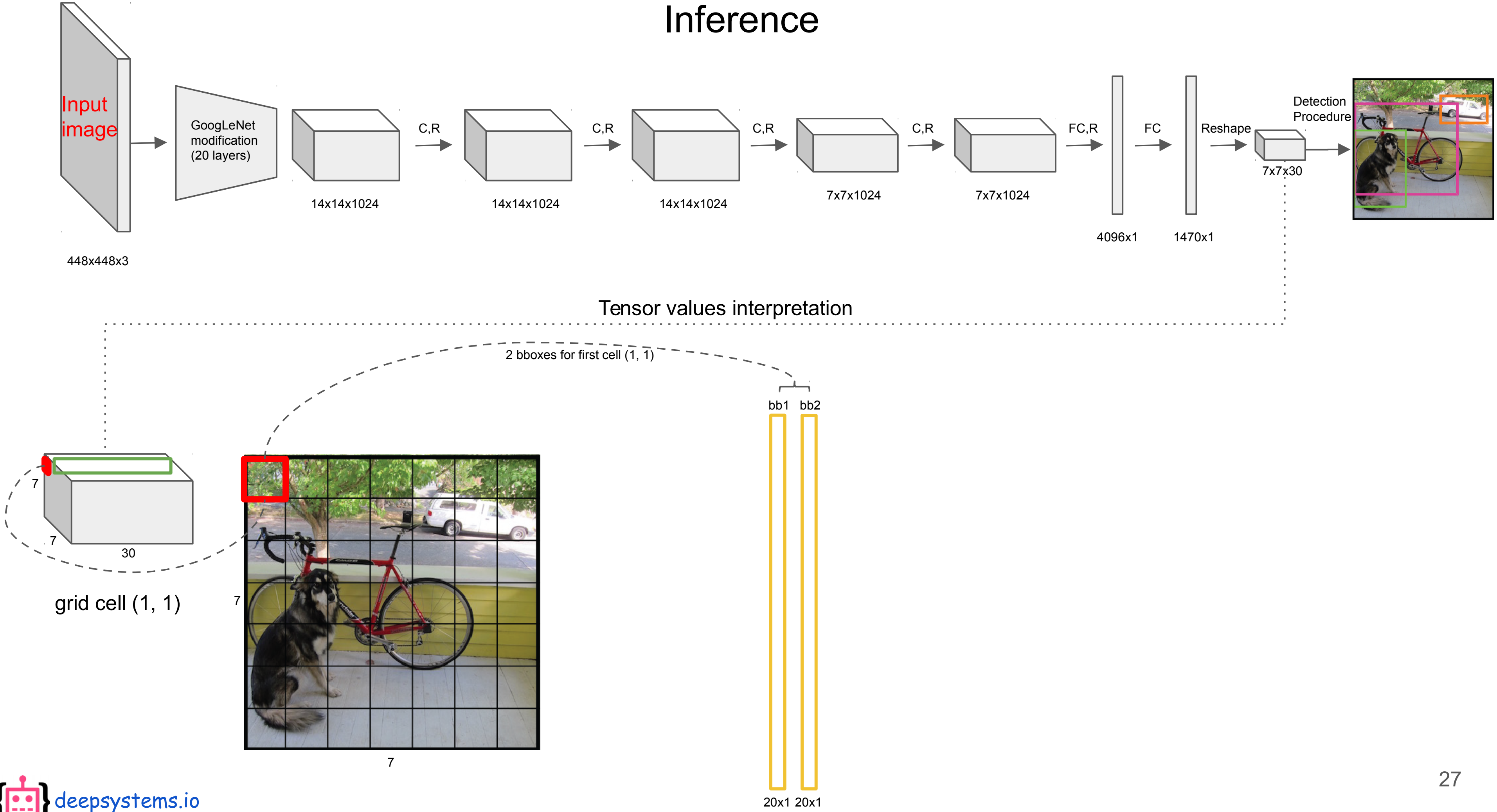
Inference



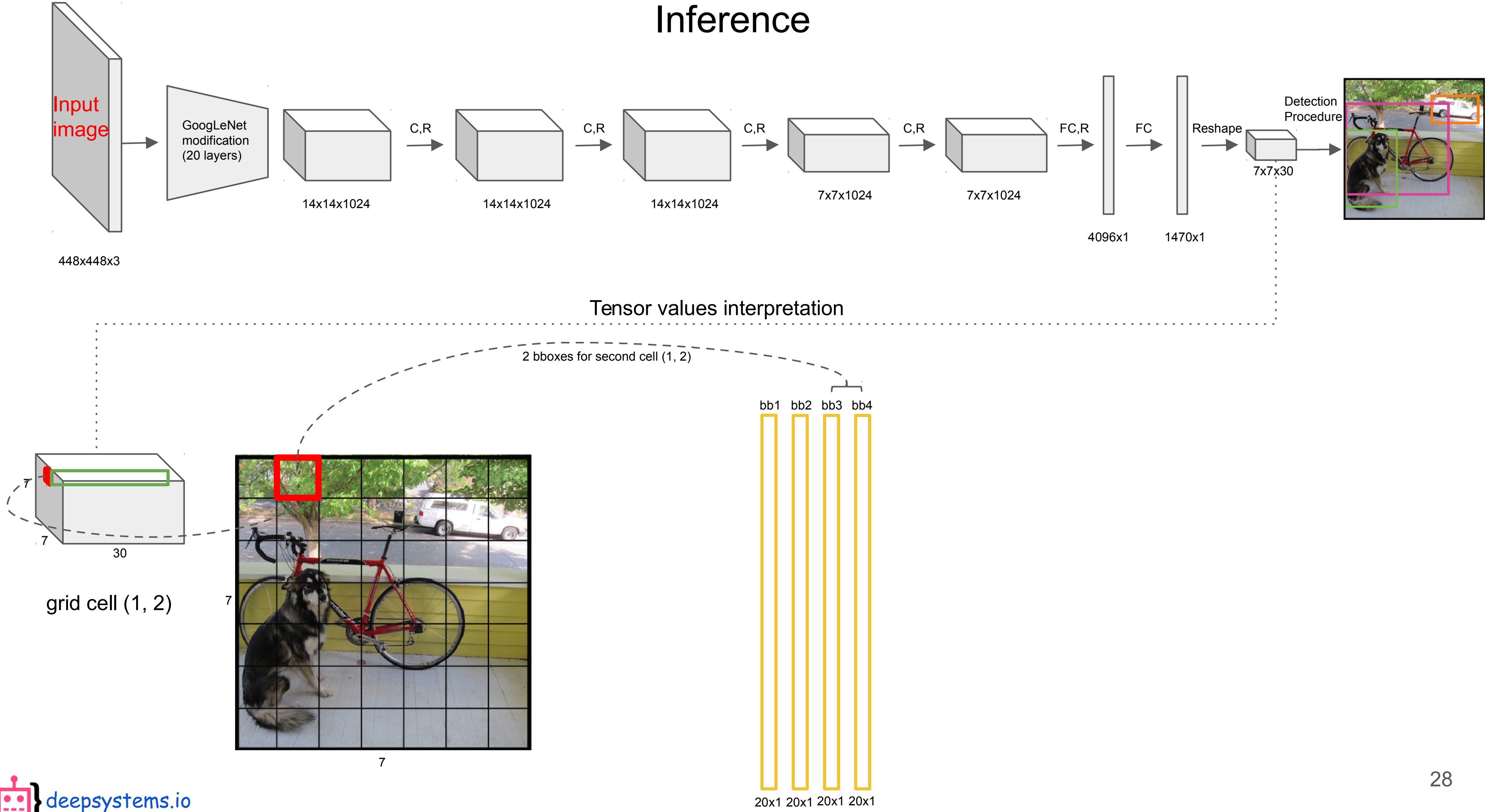
Do this operation for each bbox in each grid cell

deepsystems.io

Inference

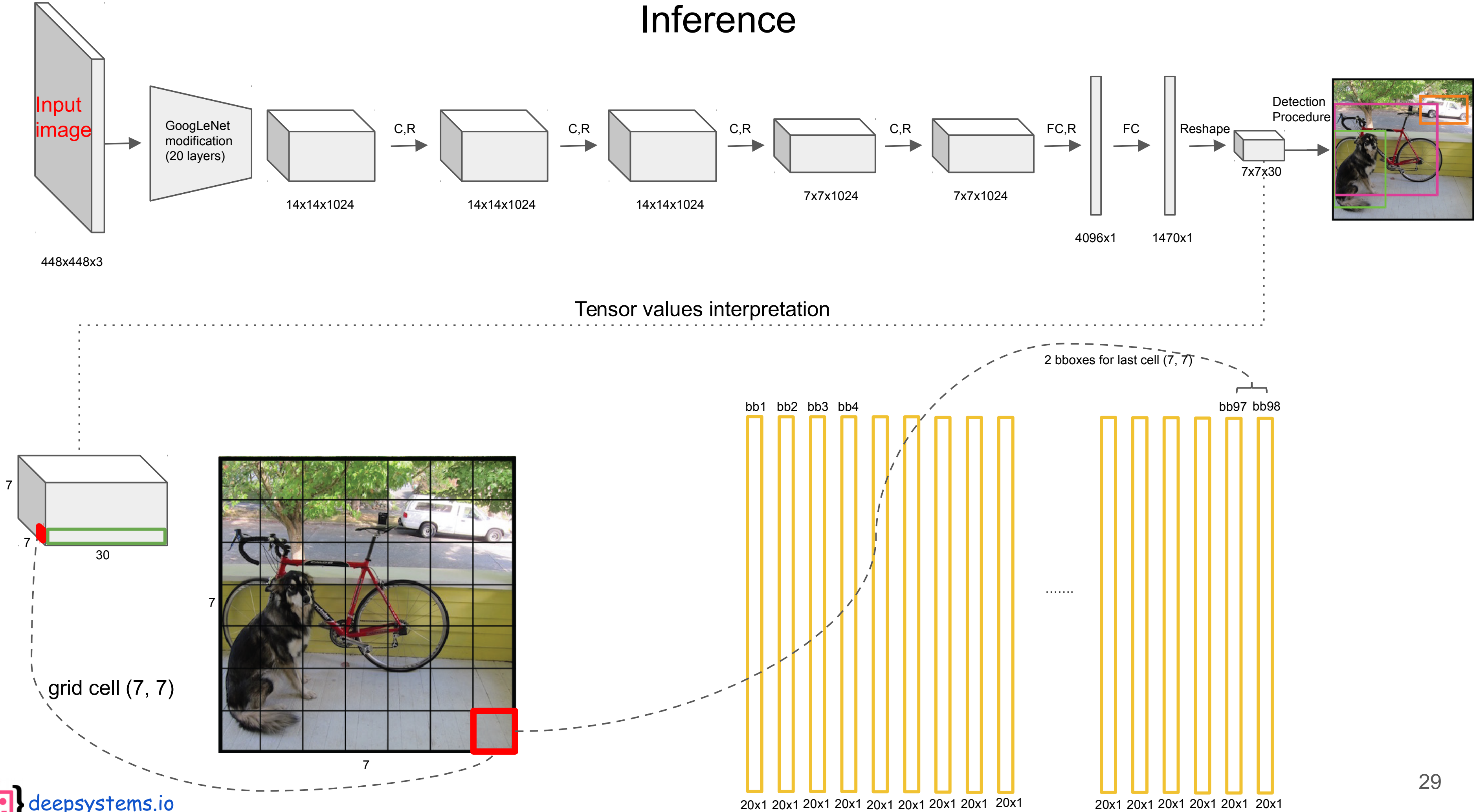

Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
2 bboxes for first cell (1, 1)
bb1
bb2
7
7
30
grid cell (1, 1)
7
7
20x1
20x1

Inference



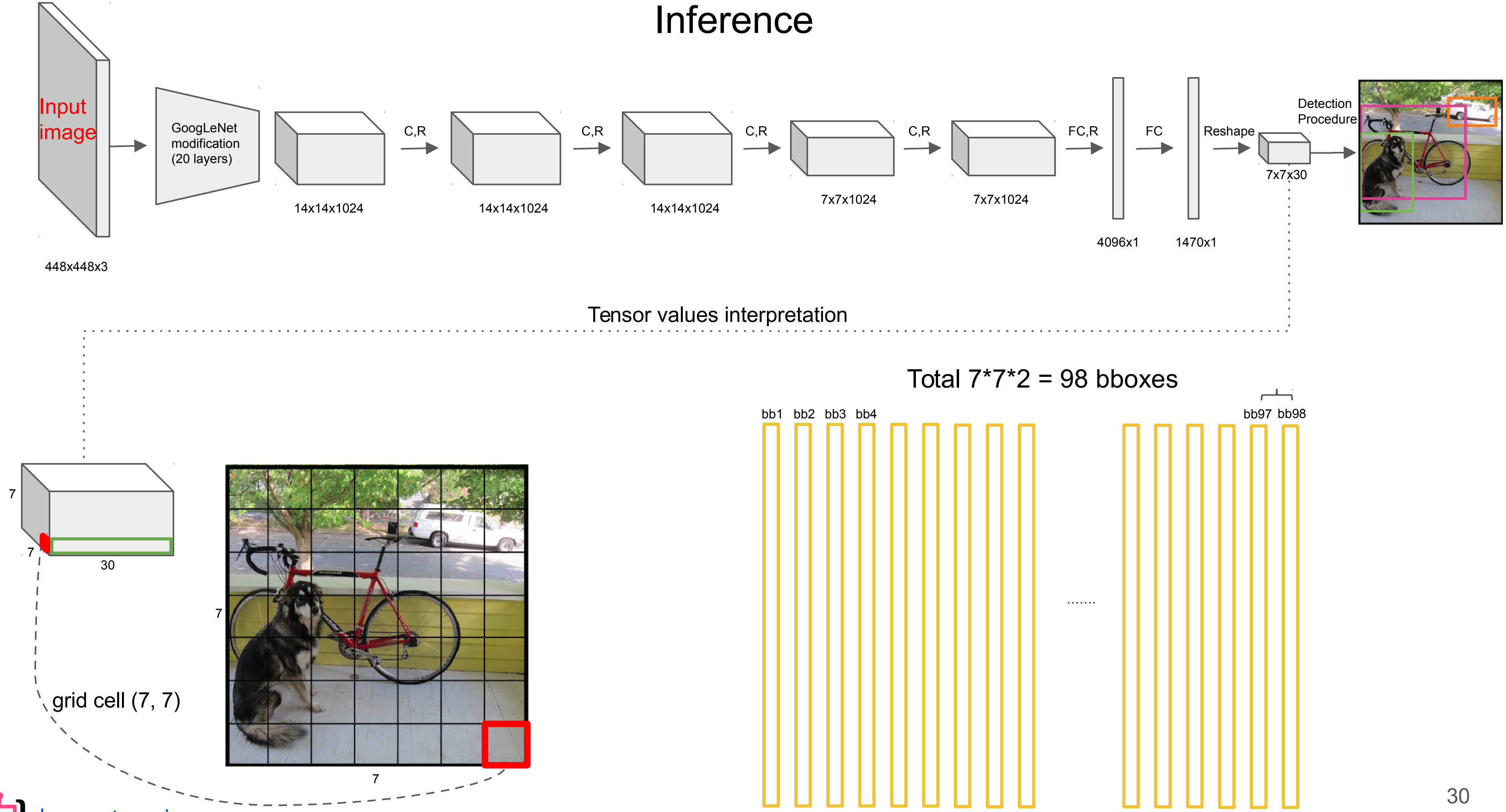
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
2 bboxes for second cell (1, 2)
7
7
30
grid cell (1, 2)
7
7
bb1 bb2 bb3 bb4
20x1 20x1 20x1 20x1

Inference

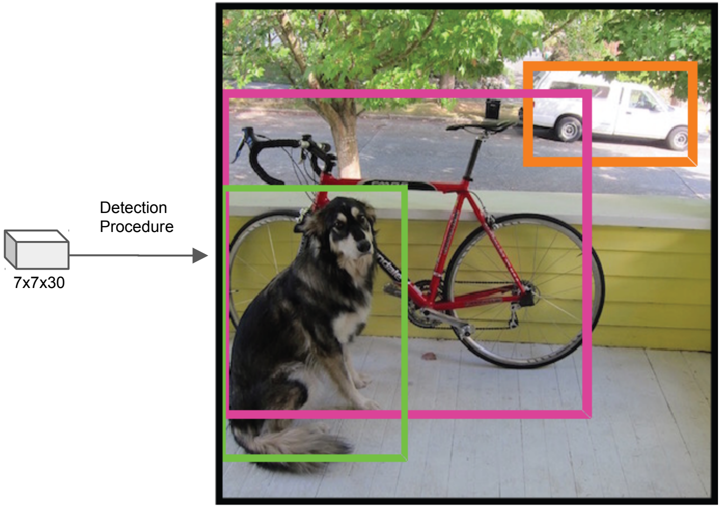


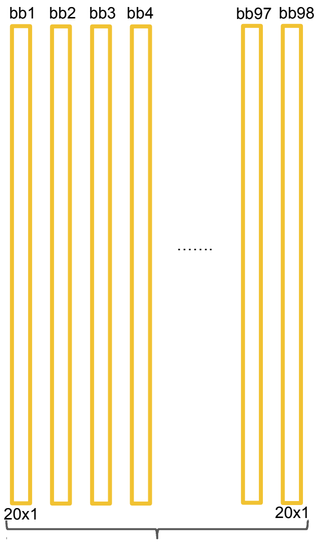
deepsystems.io

Inference

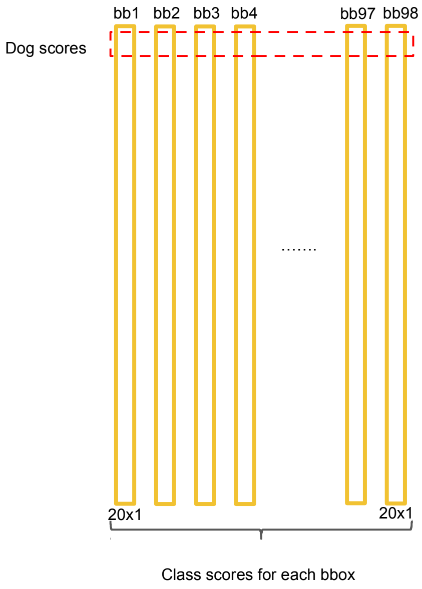

Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
7
7
30
grid cell (7, 7)
7
7
Total 7*7*2 = 98 bboxes
bb1 bb2 bb3 bb4
....
bb97 bb98

Look at detection procedure

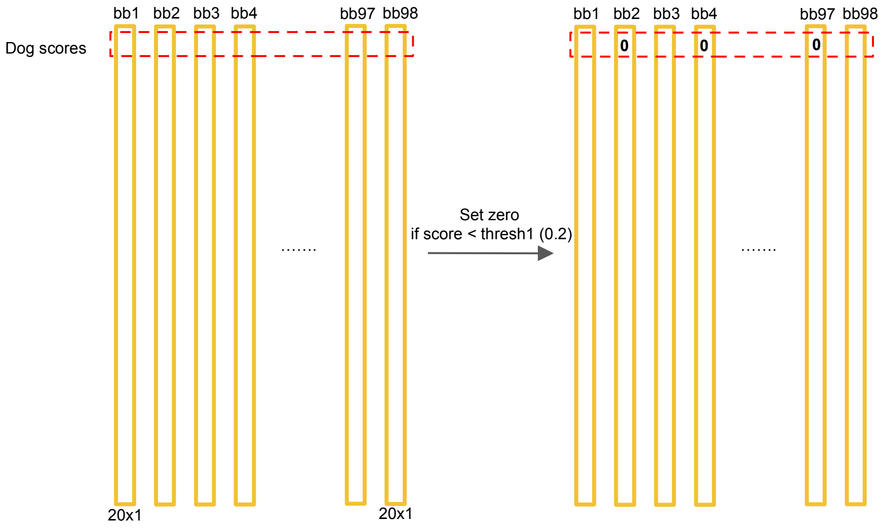


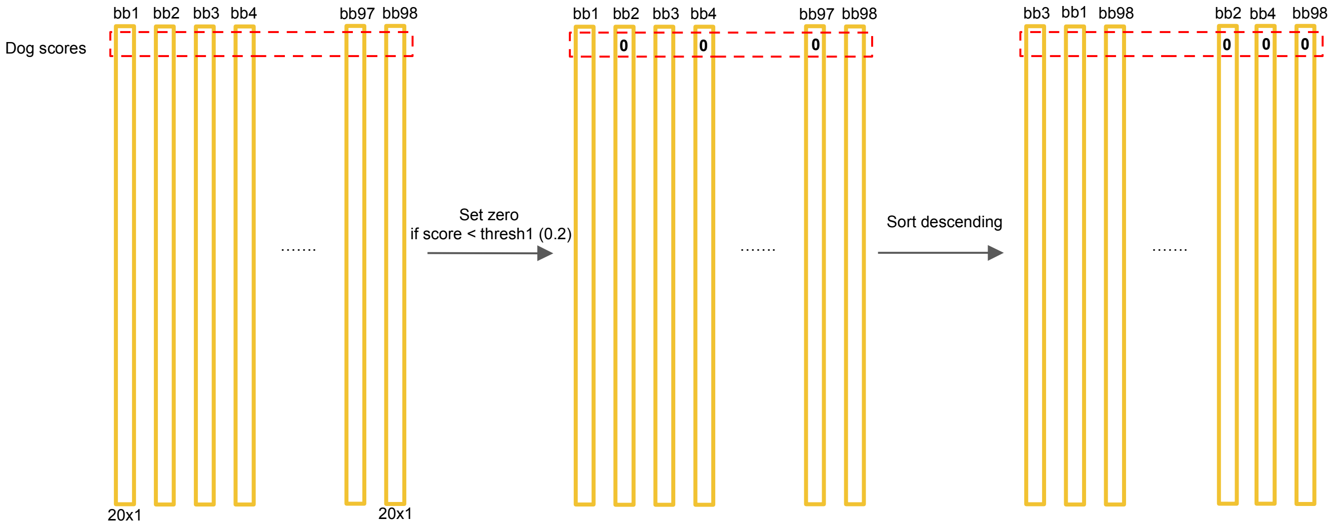


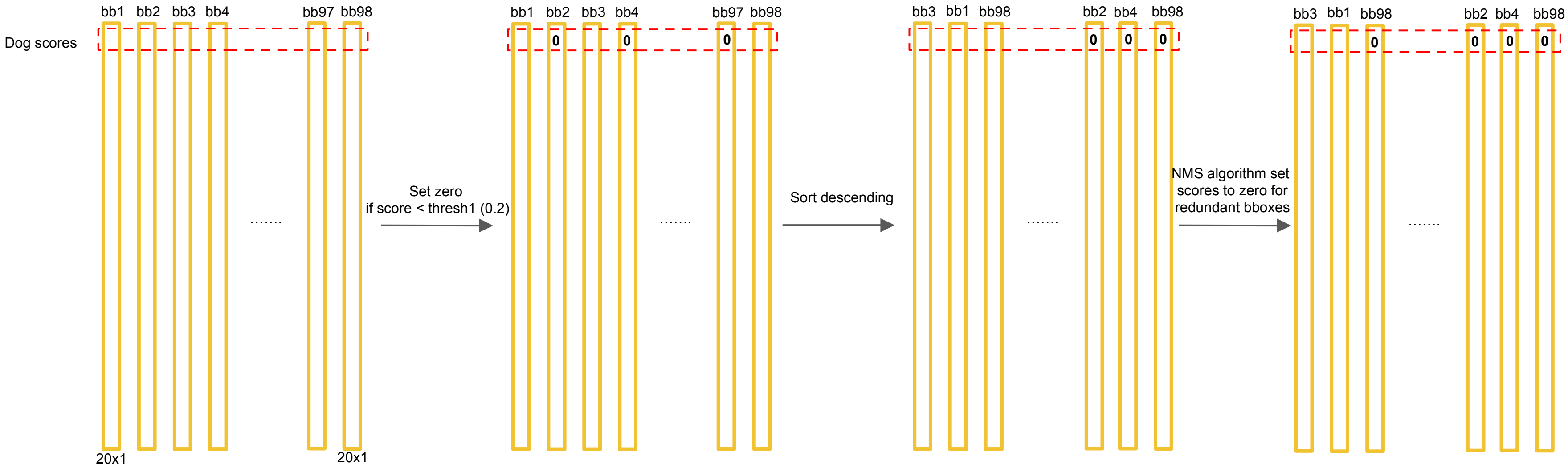
Class scores for each bbox

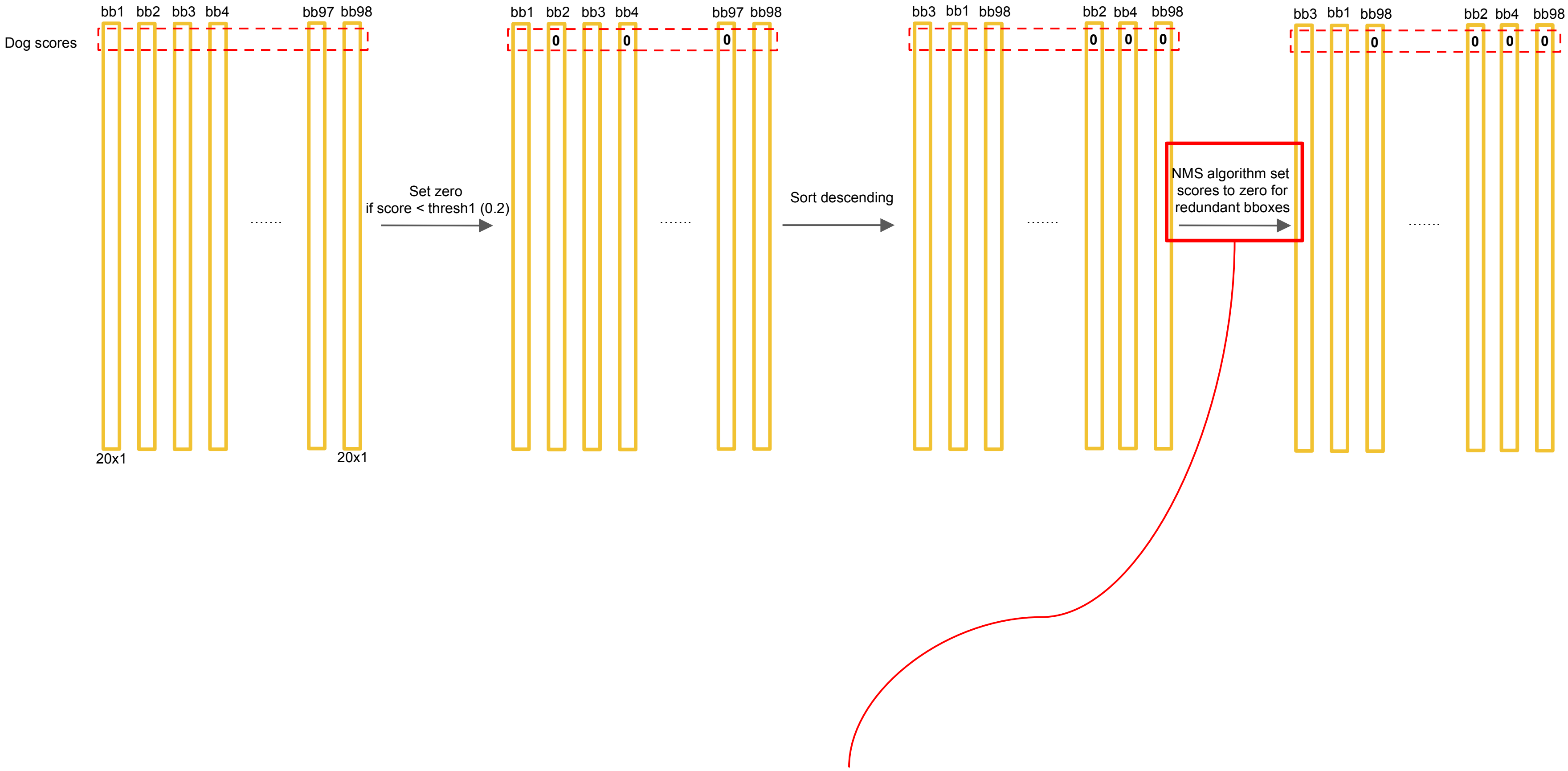


Get first class scores for each bbox


bb1 bb2 bb3 bb4 bb97 bb98
Dog scores
20x1
20x1
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4 bb97 bb98
0
0
0


Dog scores
bb1
bb2
bb3
bb4
bb97
bb98
20x1
20x1
....
Set zero
if score < thresh1 (0.2)
bb1
bb2
bb3
bb4
bb97
bb98
0
0
0
....
Sort descending
bb3
bb1
bb98
bb2
bb4
bb98
0
0
0
....


Dog scores
bb1 bb2 bb3 bb4 bb97 bb98
20x1
20x1
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4 bb97 bb98
0 0 0
Sort descending
bb3 bb1 bb98 bb2 bb4 bb98
0 0 0
NMS algorithm set scores to zero for redundant bboxes
bb3 bb1 bb98 bb2 bb4 bb98
0 0 0 0
....


Dog scores
bb1 bb2 bb3 bb4 bb97 bb98
20x1
20x1
....
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4 bb97 bb98
0 0 0
....
Sort descending
bb3 bb1 bb98 bb2 bb4 bb98
0 0 0
....
NMS algorithm set scores to zero for redundant bboxes
bb3 bb1 bb98 bb2 bb4 bb98
0 0 0 0
....

How it works

Non-Maximum Suppression: intuition

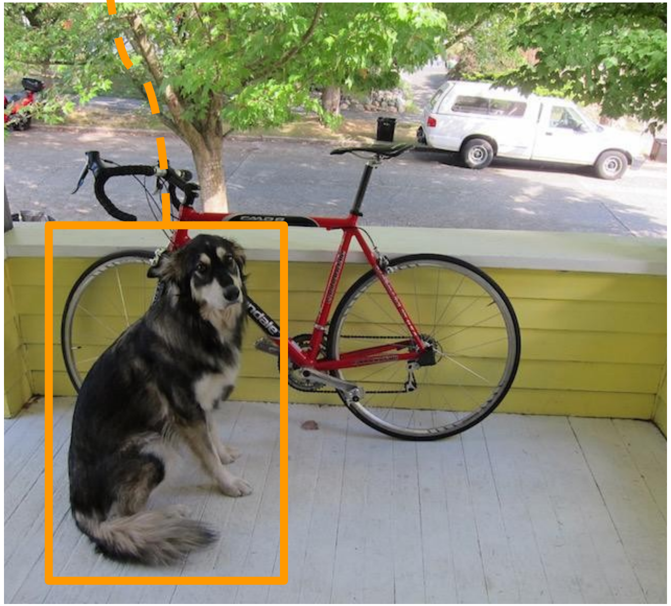
Non-Maximum Suppression: intuition

class (dog) scores for each bbox

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2							0	0	0	0	1x98

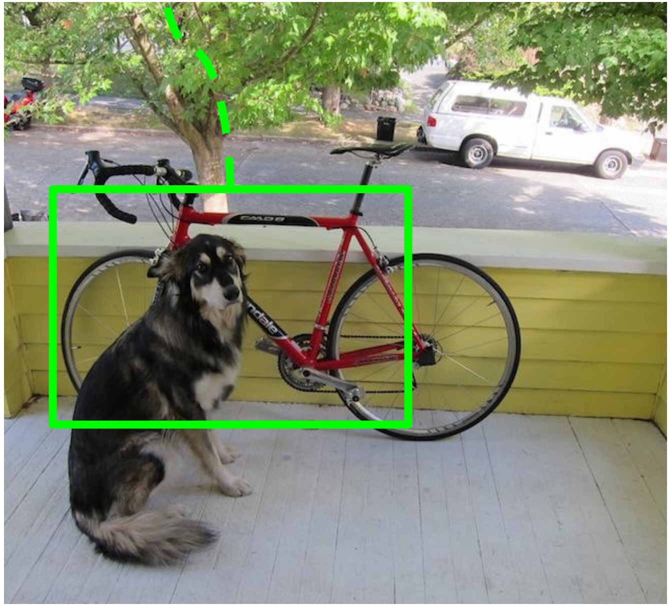
Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2							0	0	0	0	1x98



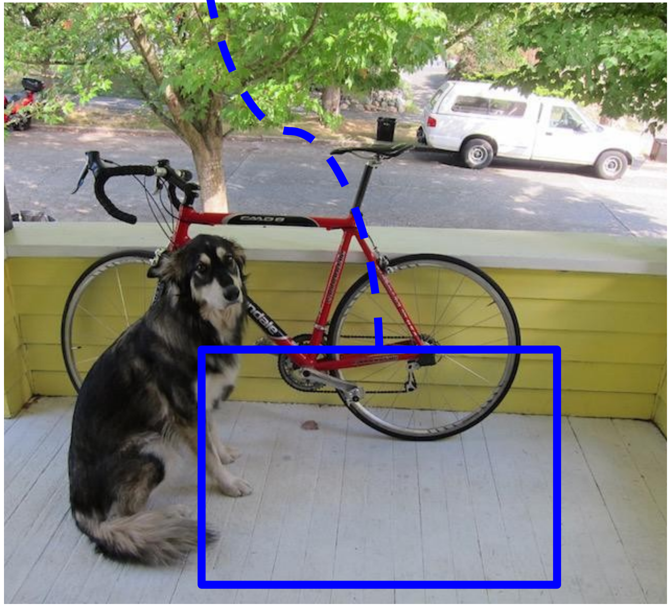
Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2							0	0	0	0	1x98



Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2							0	0	0	0	1x98

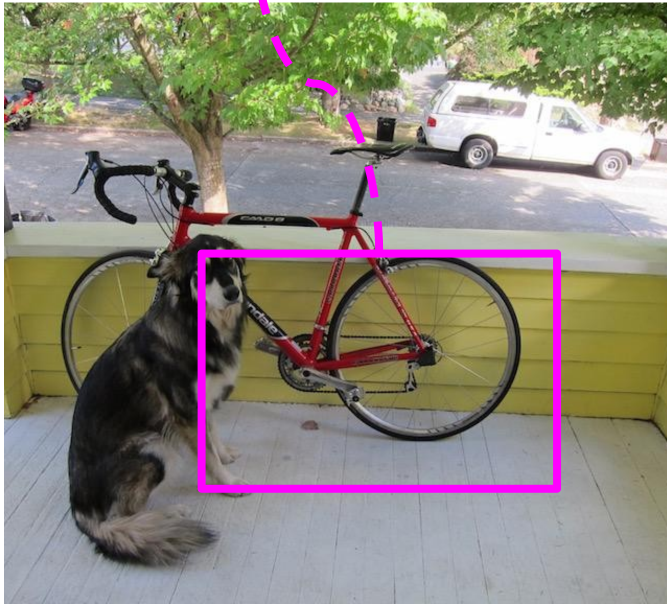


Non-Maximum Suppression: intuition

class: dog

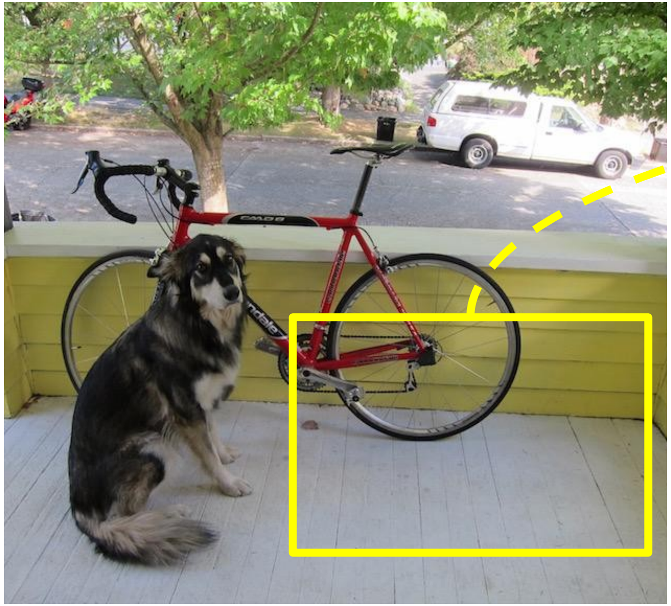
bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
0.8	0.5	0.3	0.2							0	0	0	0

1x98



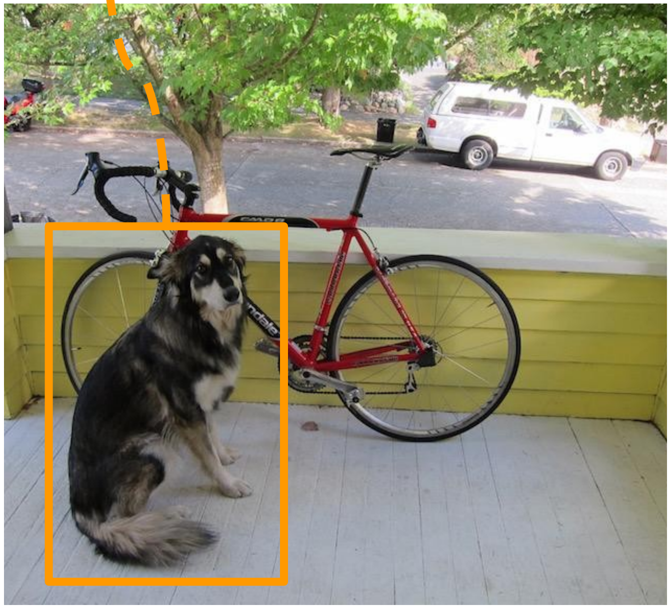
Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2							0	0	0	0	1x98



Non-Maximum Suppression: intuition

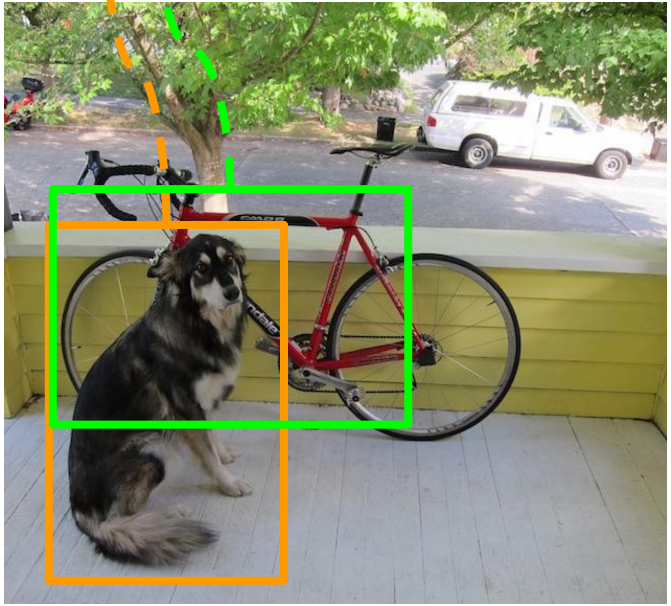
	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2							0	0	0	0	1x98



Get bbox with max score. Let’s denote it “bbox_max”

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7		bb1	bb4	bb8	bb98	
class: dog	0.8	0.5	0.3	0.2		0	0	0	0	1x98



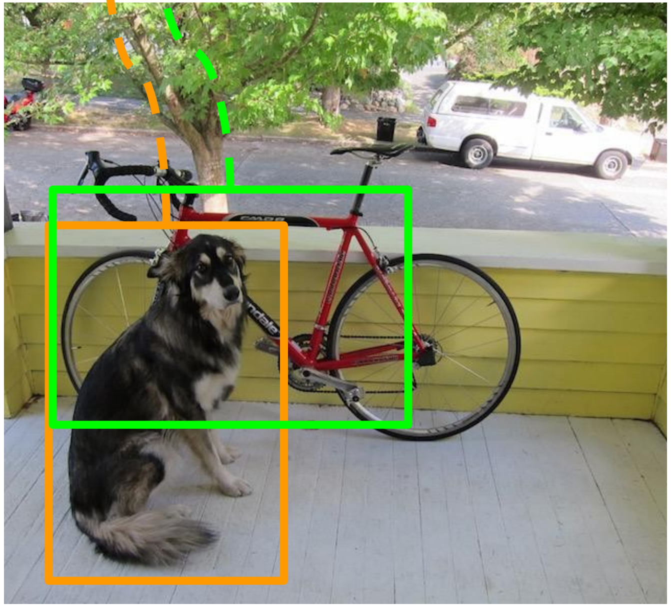
Compare "bbox_max" with others less score (non-zero!) bboxes. Let's denote it "bbox_cur"

Non-Maximum Suppression: intuition

class: dog

bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
0.8	0.5	0.3	0.2							0	0	0	0

1x98



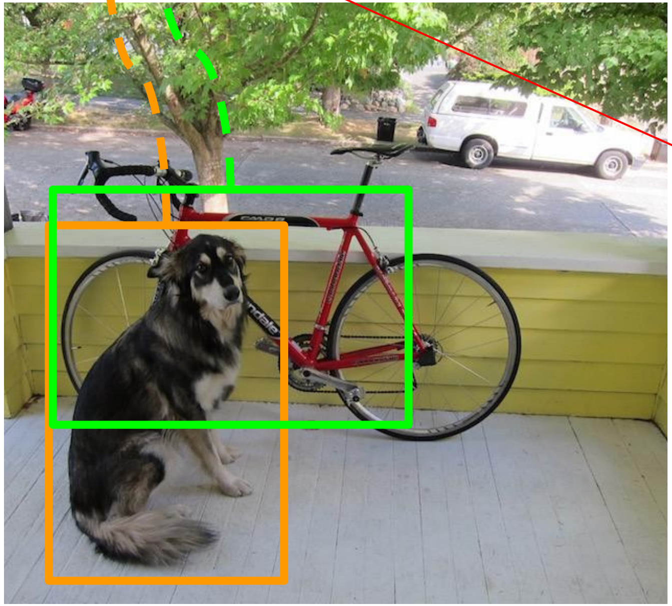
If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

Non-Maximum Suppression: intuition

class: dog

bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
0.8	**0**	0.3	0.2							0	0	0	0

1x98

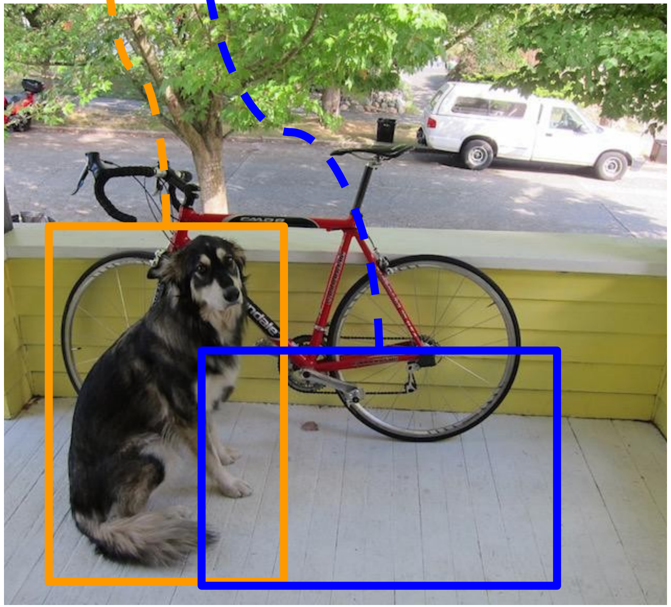


If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

In this case: set to 0.

Non-Maximum Suppression: intuition

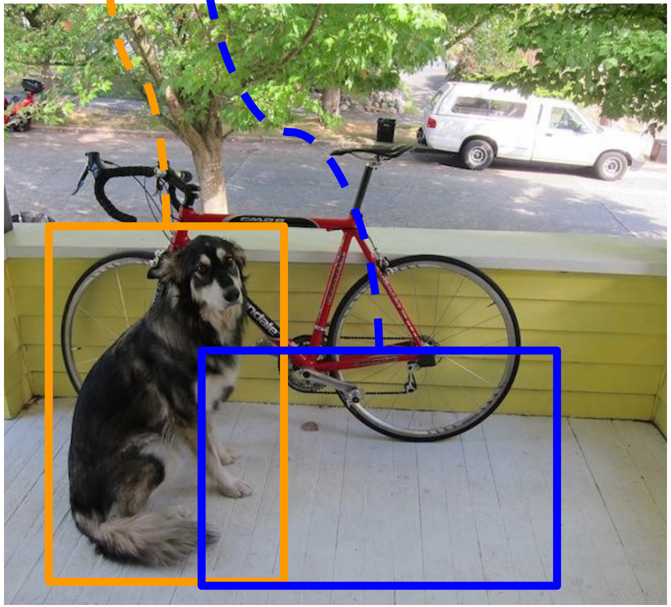
	bb47	bb20	bb15	bb7						bb1	bb4	bb8	bb98	
class: dog	0.8	0	0.3	0.2						0	0	0	0	1x98



Go to next bbox_cur.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0	0.3	0.2							0	0	0	0	1x98

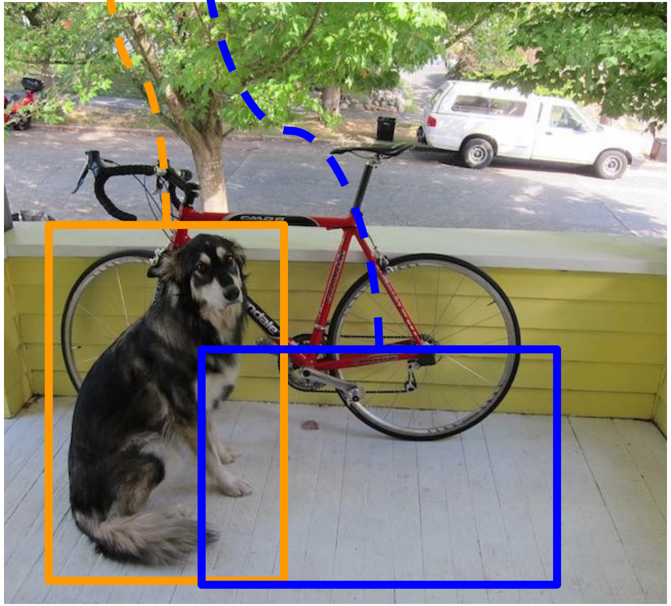


Go to next bbox_cur.

If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0	0.3	0.2							0	0	0	0	1x98



Go to next bbox_cur.

If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

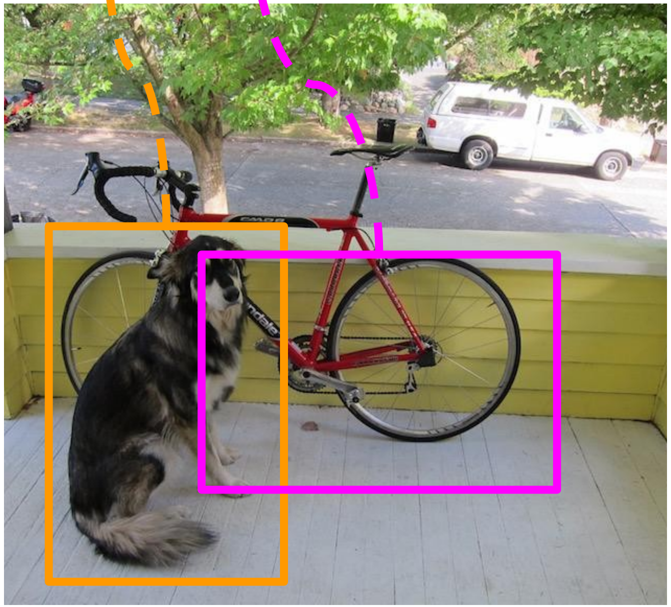
In this case: continue.

Non-Maximum Suppression: intuition

class: dog

bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
0.8	0	0.3	0.2							0	0	0	0

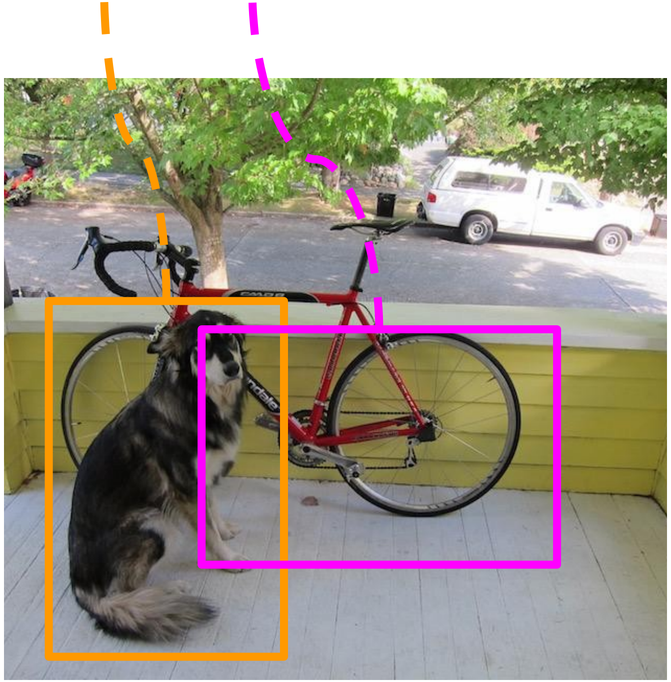
1x98



Go to next bbox_cur.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7						bb1	bb4	bb8	bb98	
class: dog	0.8	**0**	0.3	0.2						0	0	0	0	1x98

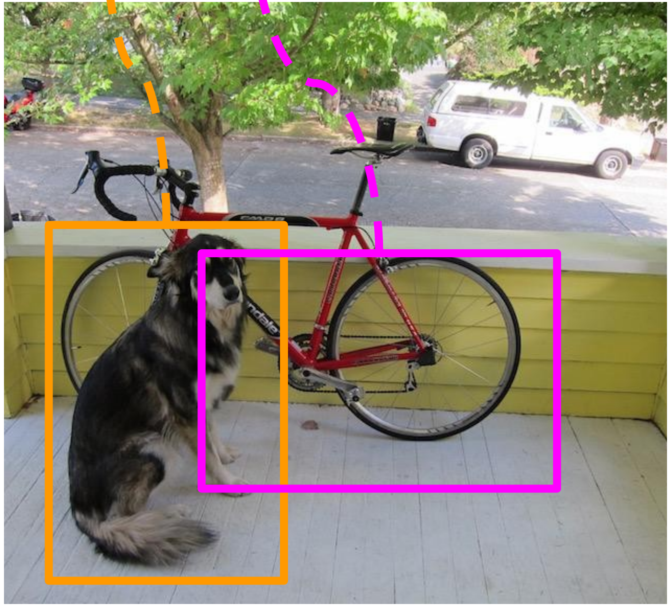


Go to next bbox_cur.

If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0	0.3	0.2							0	0	0	0	1x98



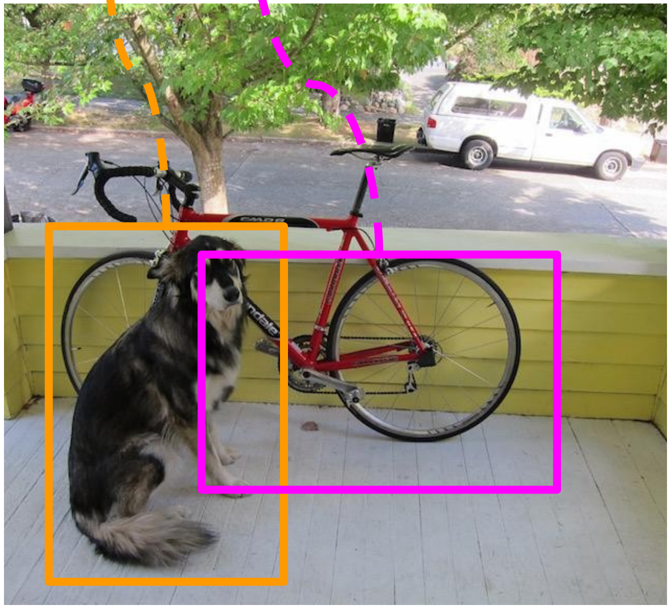
Go to next bbox_cur.

If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

In this case: continue.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	**0**	0.3	0.2							0	0	0	0	1x98



Go to next bbox_cur.

If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

In this case: continue.

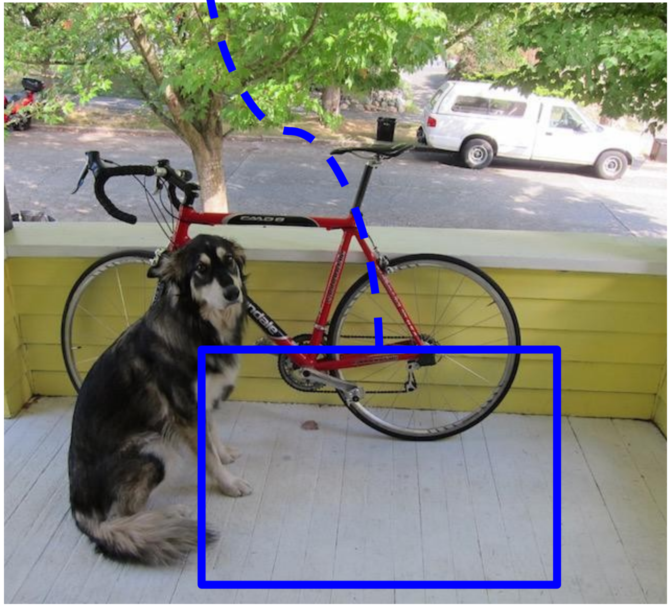
Do this procedure for other “bbox_cur”. After that ...

Non-Maximum Suppression: intuition

class: dog

bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
0.8	0	0.3	0.2							0	0	0	0

1x98



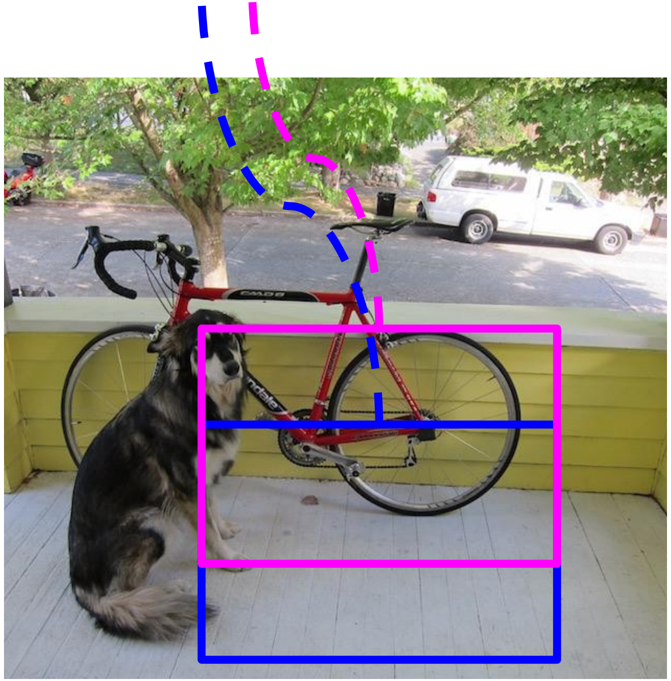
Go to next bbox with big score. Let's denote it "bbox_max"

Non-Maximum Suppression: intuition

class: dog

bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
0.8	**0**	0.3	0.2							0	0	0	0

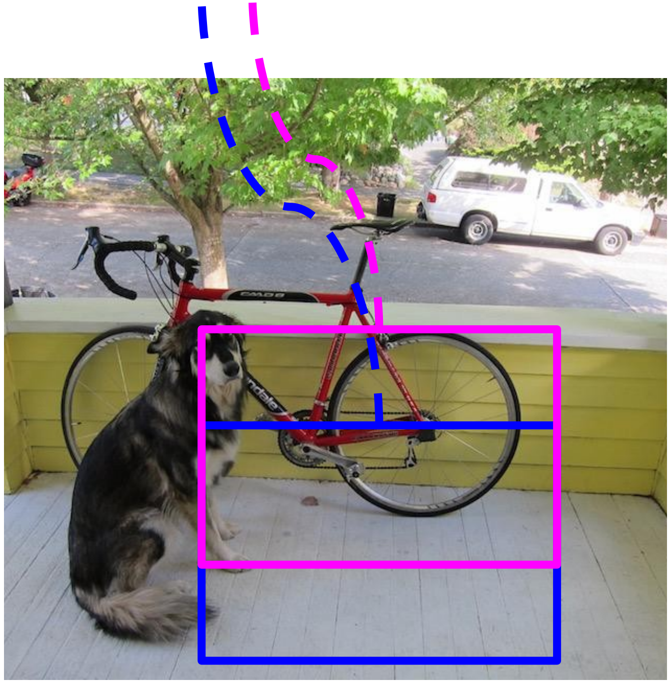
1x98



Go to next bbox_cur.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0	0.3	0.2							0	0	0	0	1x98

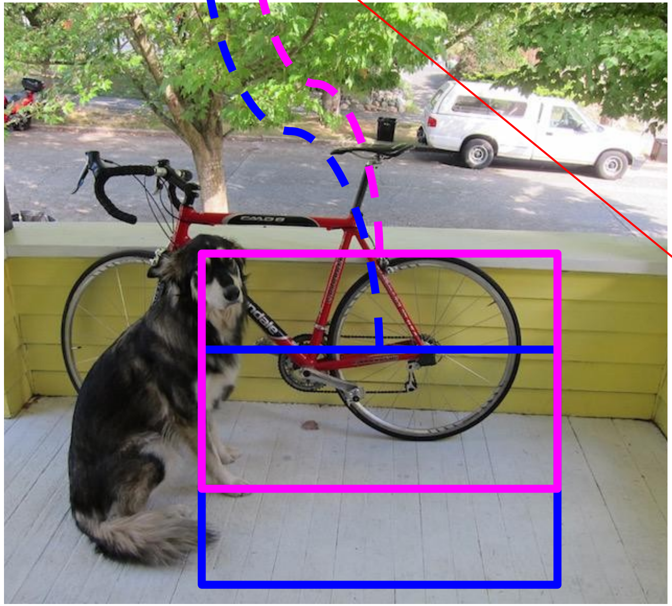


Go to next bbox_cur.

If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98	
class: dog	0.8	0	0.3	0							0	0	0	0	1x98



Go to next bbox_cur.

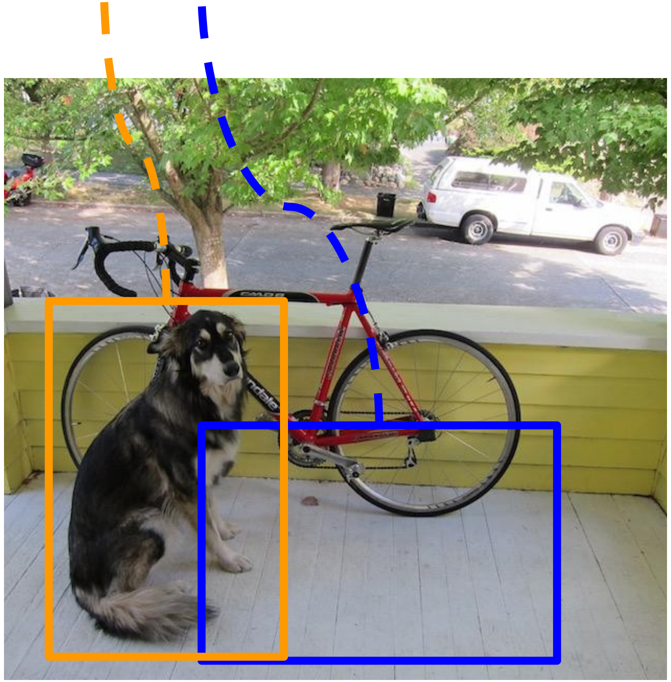
If IoU(bbox_max, bbox_cur) > 0.5 then set 0 score to bbox_cur.

In this case: set to 0.

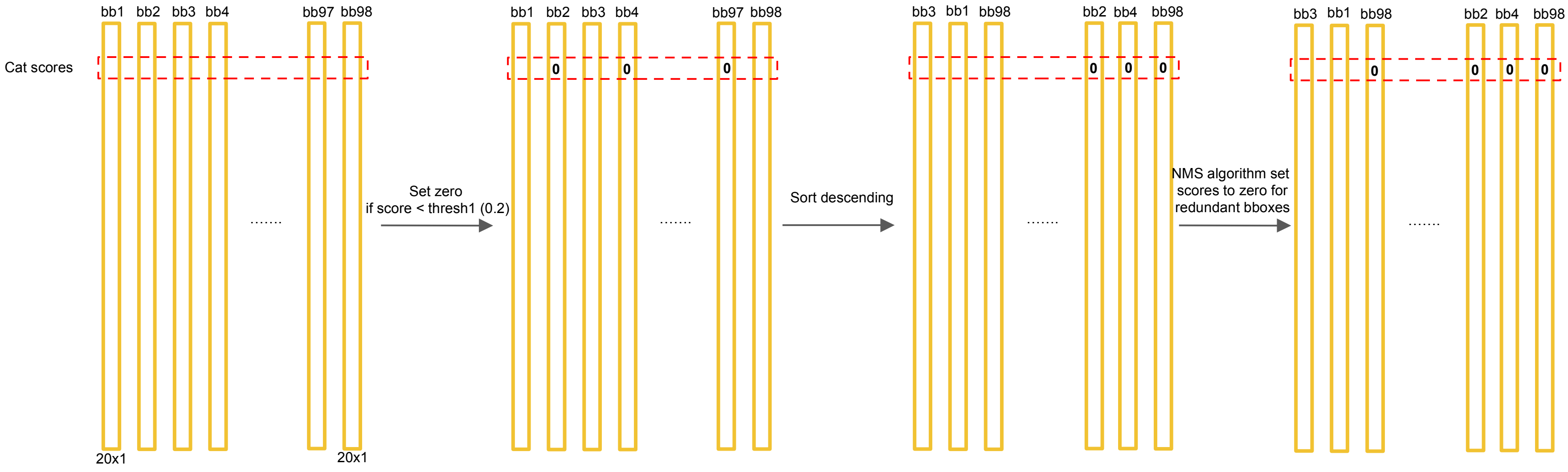
Do this procedure for other “bbox_max” and for other corresponding “bbox_cur”.

Non-Maximum Suppression: intuition

	bb47	bb20	bb15	bb7							bb1	bb4	bb8	bb98
class: dog	0.8	0	0.3	0							0	0	0	0

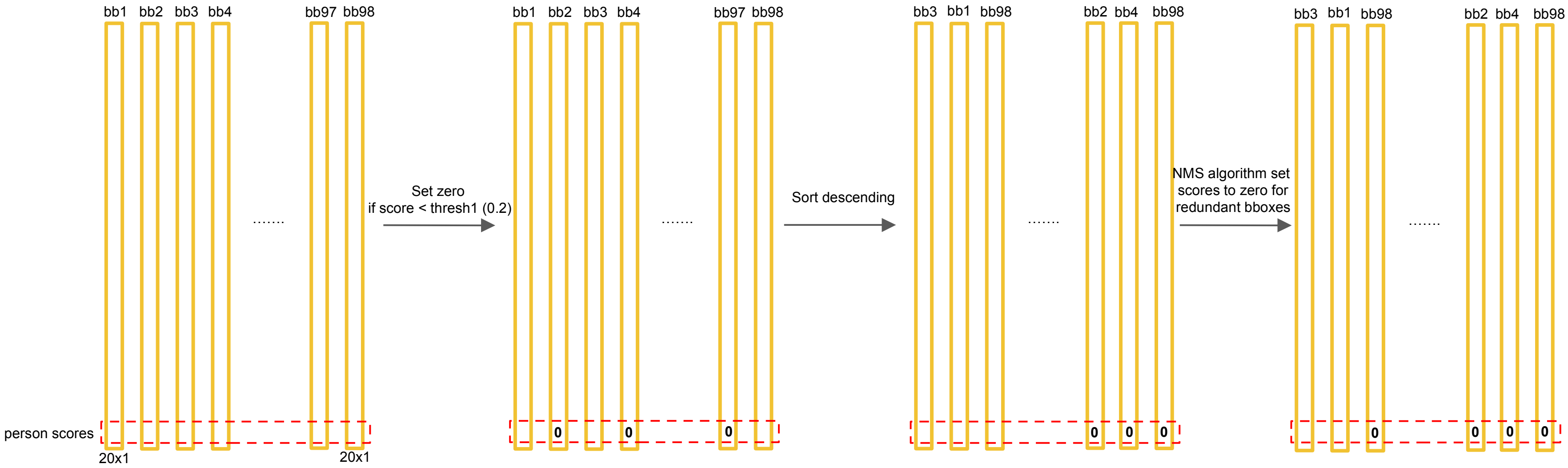


After comparison almost all pairs of bboxes the only two bboxes left with non-zero class score value.

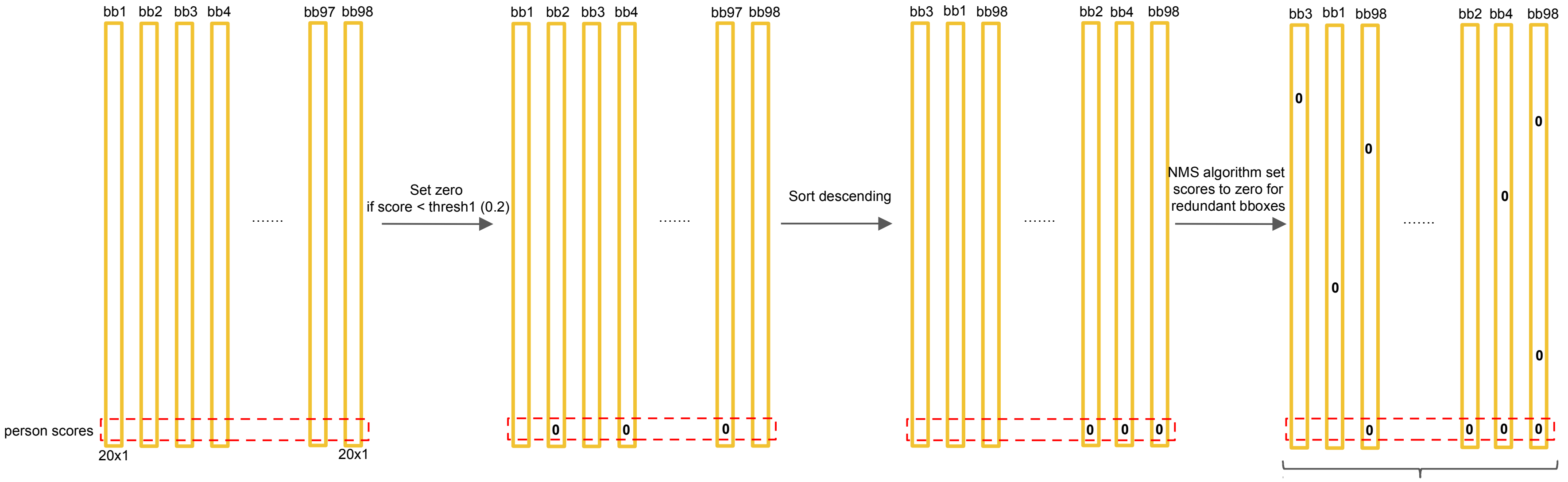


Do this procedure for next class

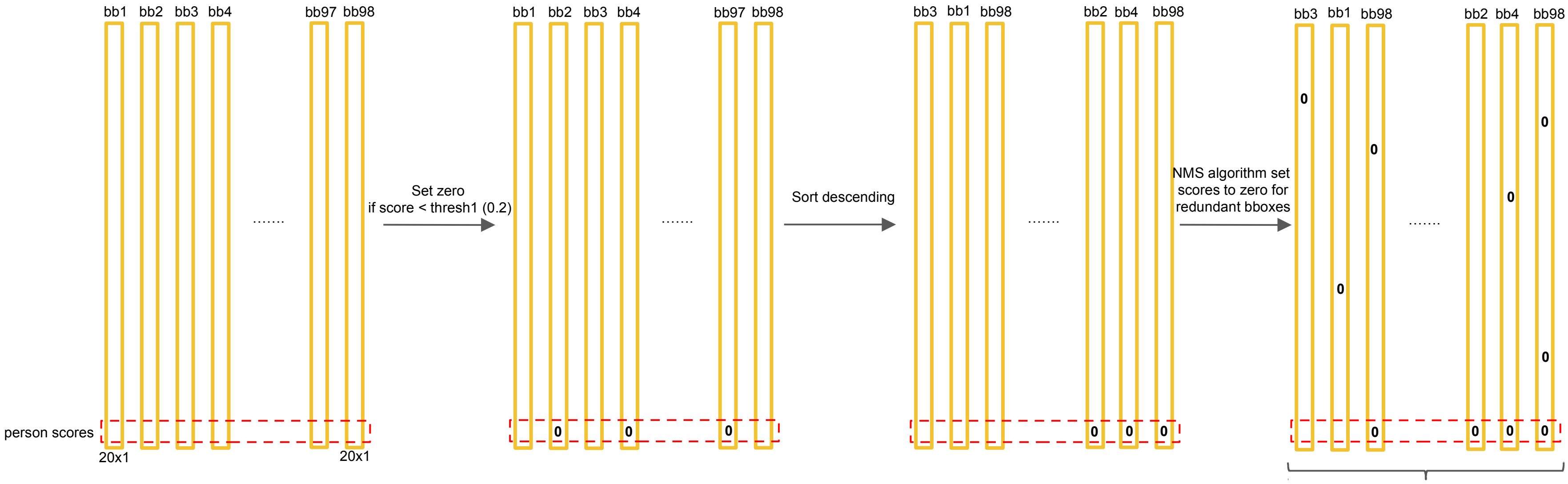
deepsystems.io



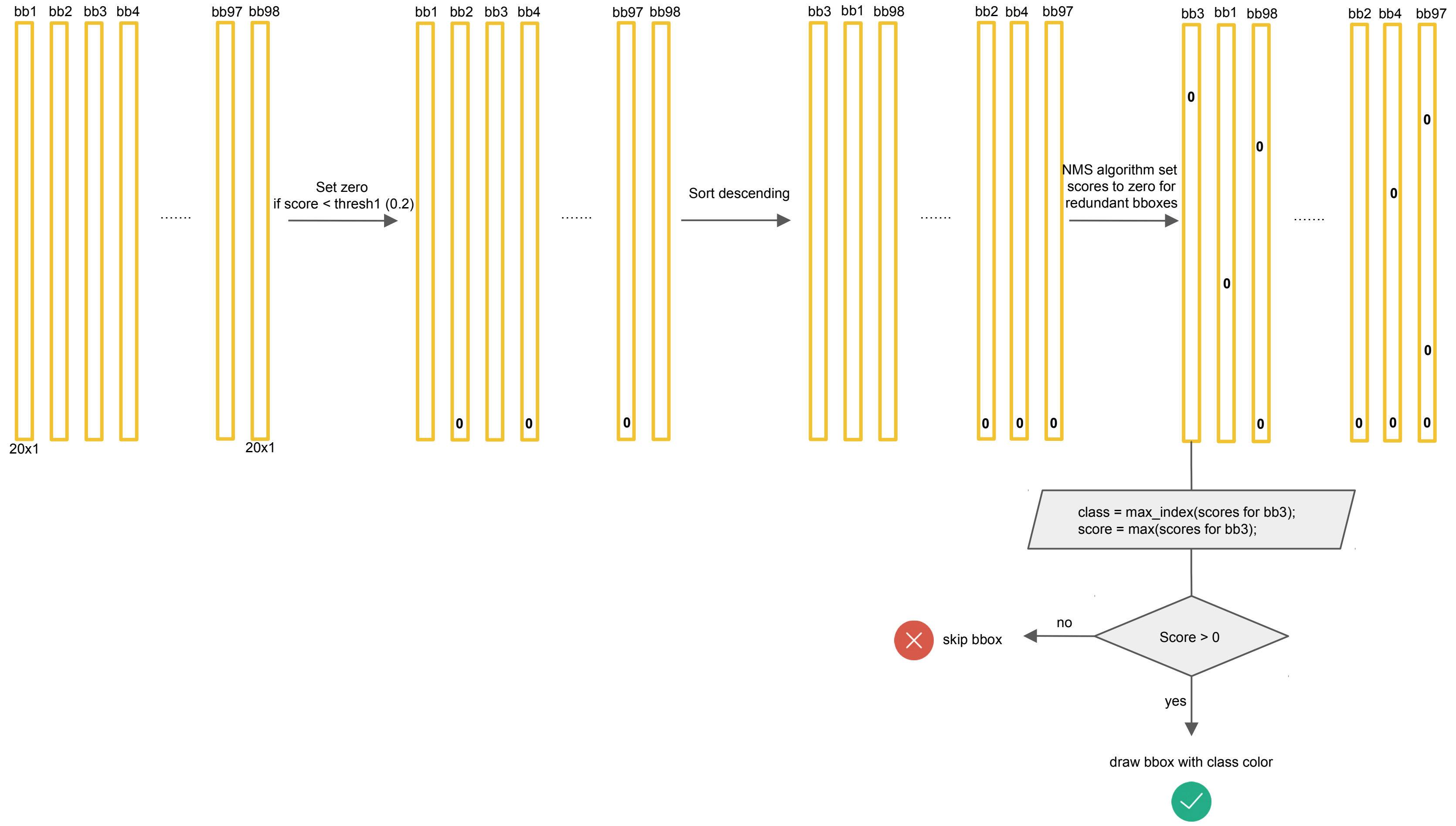
Do this procedure for all classes

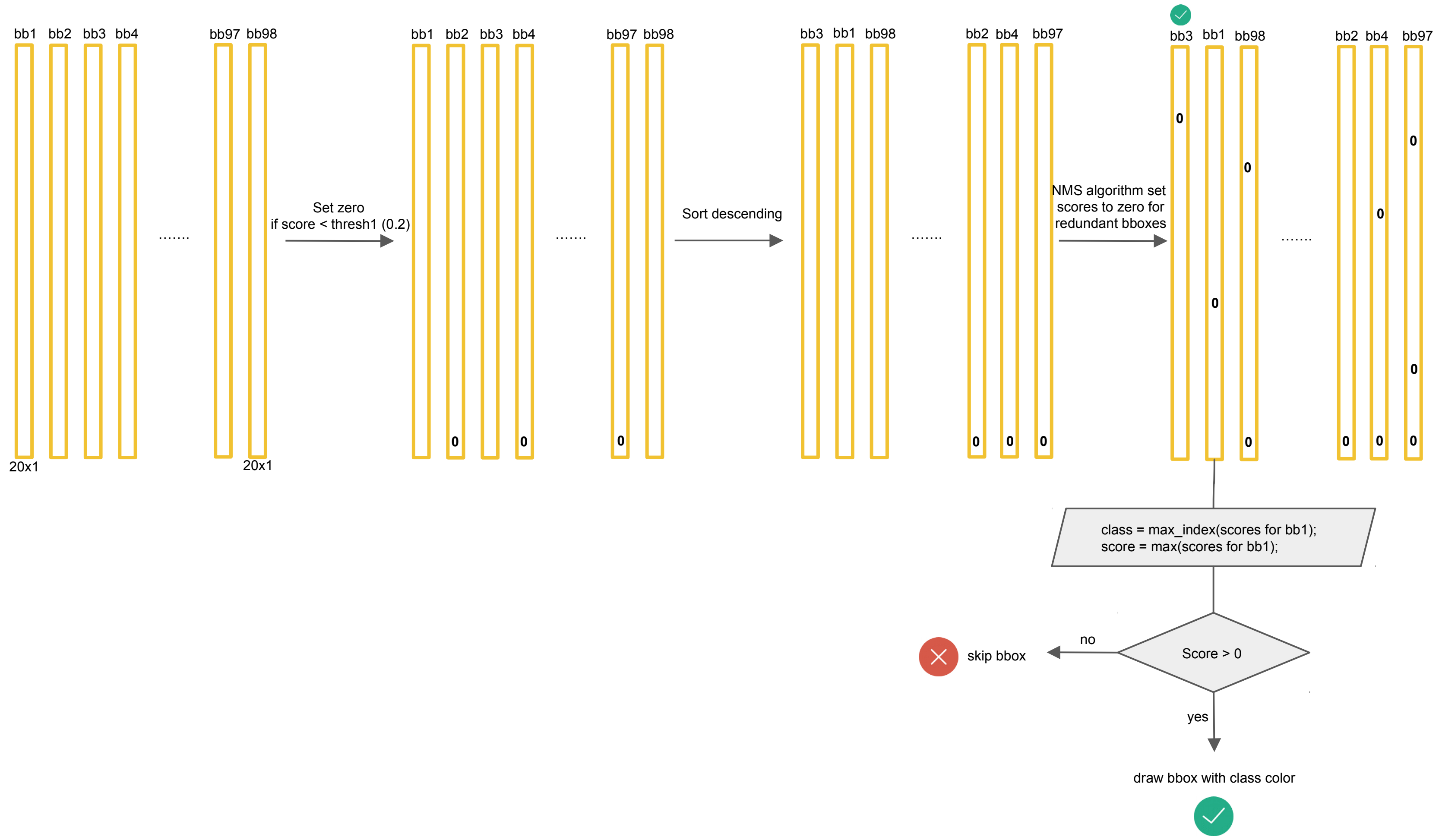


After this procedure - a lot of zeros

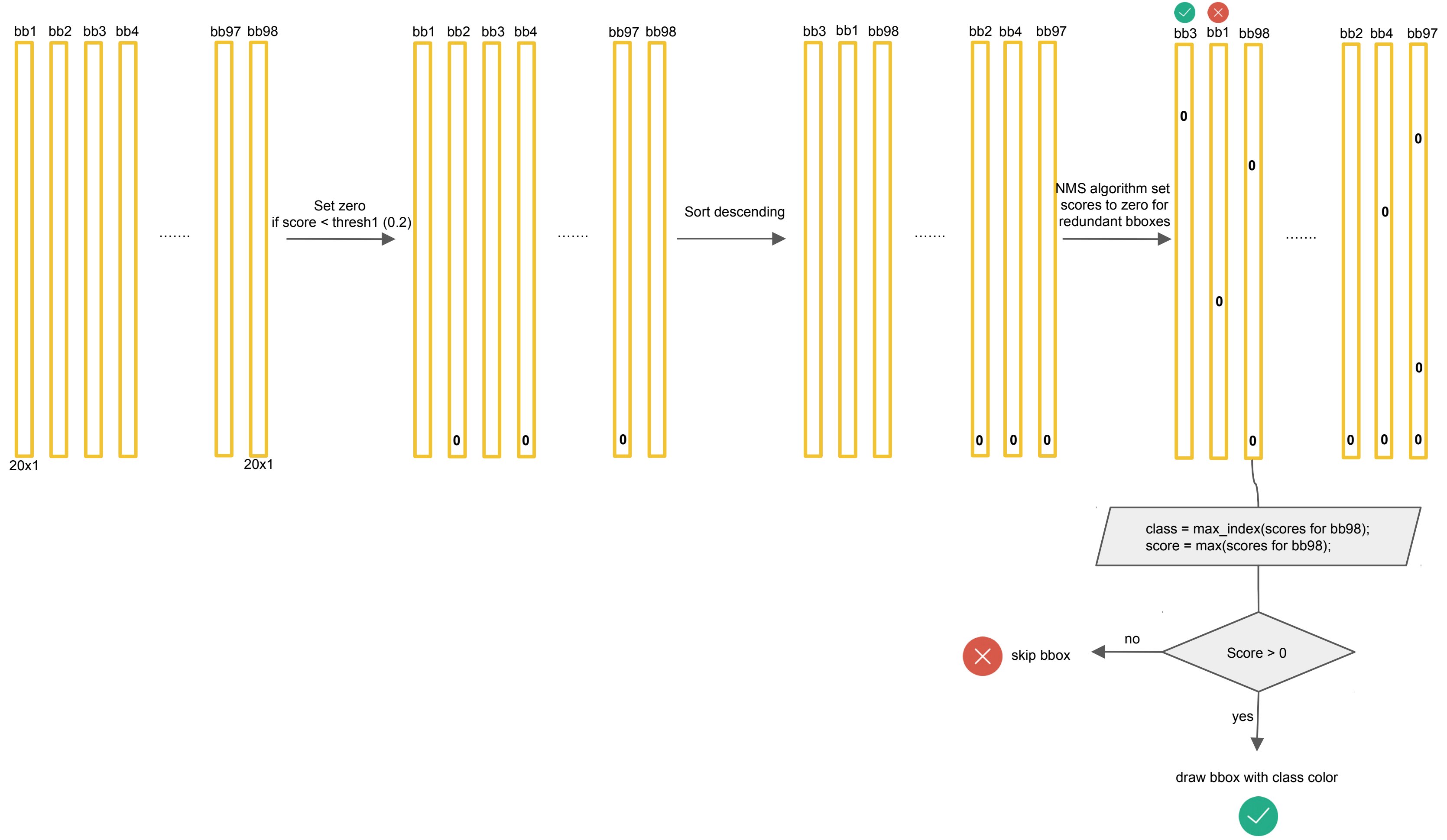


Select bboxes to draw by class score values

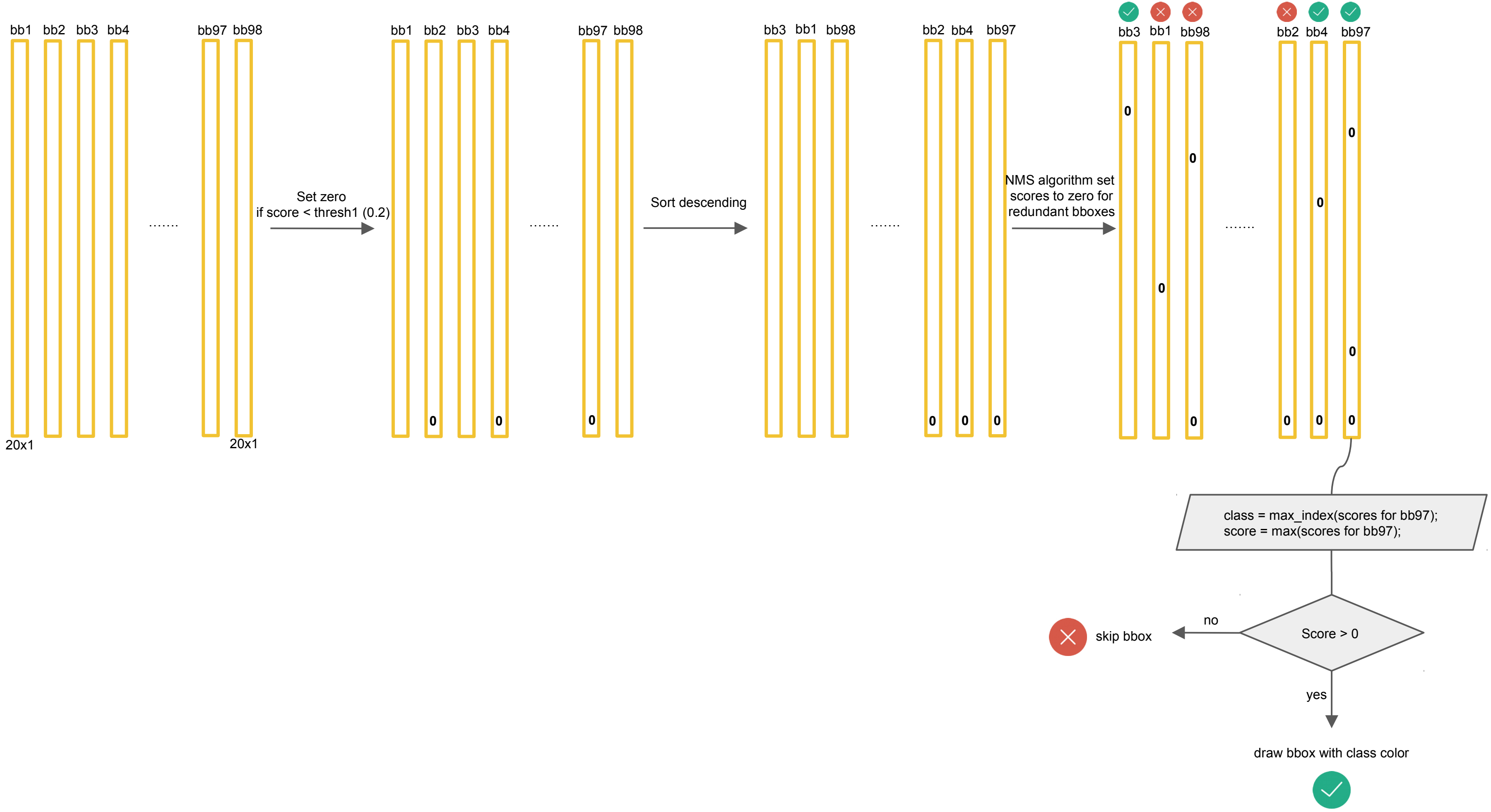

bb1 bb2 bb3 bb4
bb97 bb98
20x1
20x1
....
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4
bb97 bb98
0
0
0
....
Sort descending
bb3 bb1 bb98
bb2 bb4 bb97
0
0
0
....
NMS algorithm set
scores to zero for
redundant bboxes
bb3 bb1 bb98
bb2 bb4 bb97
0
0
0
0
0
0
0
0
0
0
....
class = max_index(scores for bb3);
score = max(scores for bb3);
Score > 0
no
skip bbox
yes
draw bbox with class color



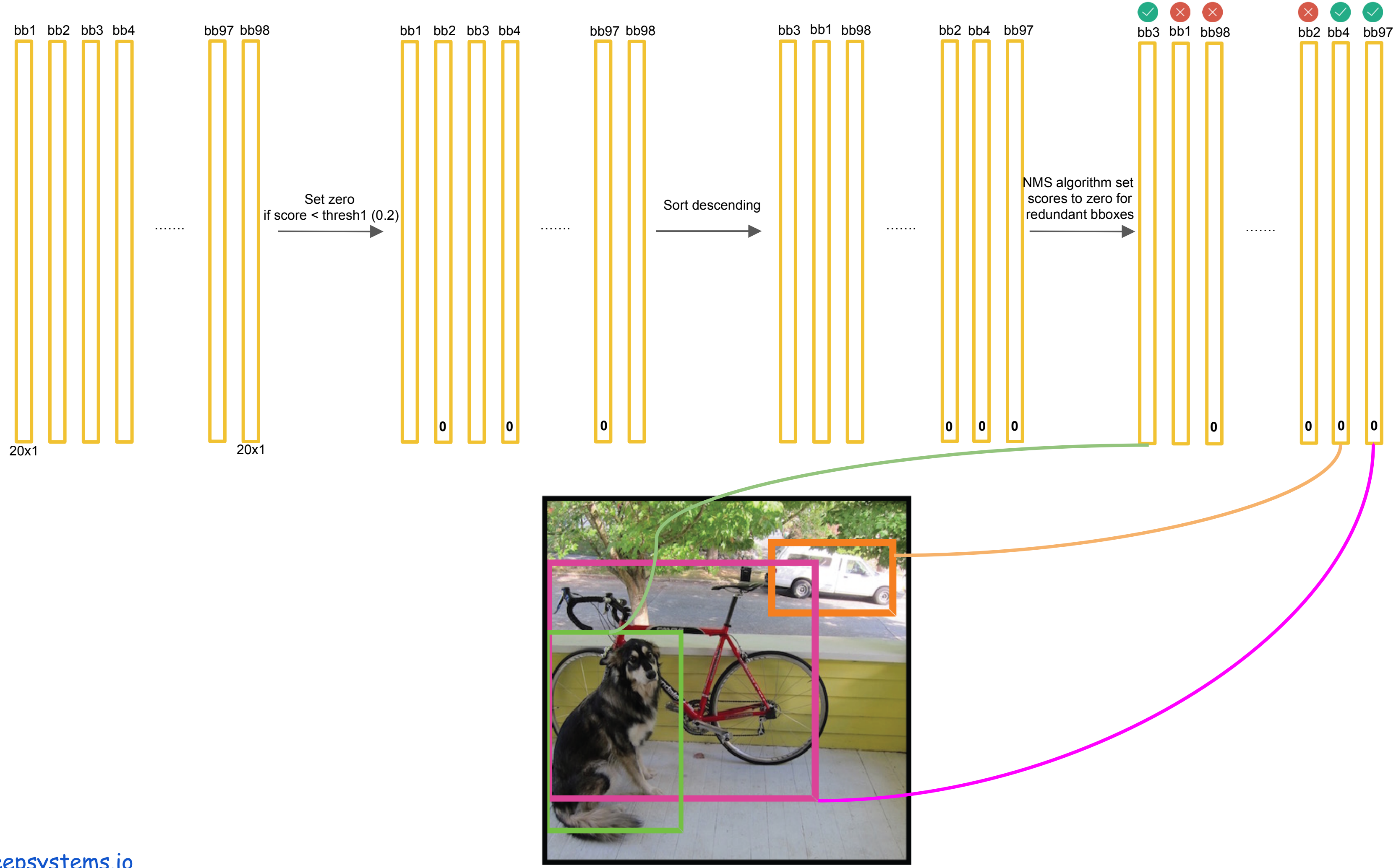
bb1 bb2 bb3 bb4
bb97 bb98
20x1
20x1
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4
bb97 bb98
0
0
0
Sort descending
bb3 bb1 bb98
bb2 bb4 bb97
0
0
0
NMS algorithm set
scores to zero for
redundant bboxes
bb3 bb1 bb98
bb2 bb4 bb97
0
0
0
0
0
0
0
0
0
0
class = max_index(scores for bb1);
score = max(scores for bb1);
Score > 0
no
skip bbox
yes
draw bbox with class color



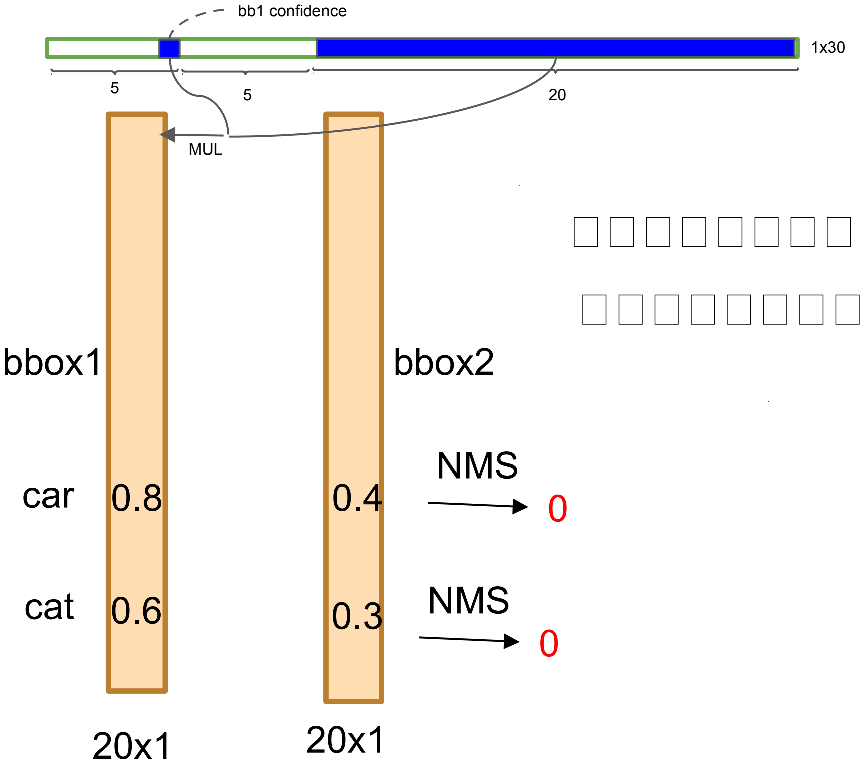
deepsystems.io



bb1 bb2 bb3 bb4
bb97 bb98
20x1
20x1
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4
bb97 bb98
Sort descending
bb3 bb1 bb98
bb2 bb4 bb97
NMS algorithm set scores to zero for redundant bboxes
bb3 bb1 bb98
bb2 bb4 bb97
class = max_index(scores for bb97);
score = max(scores for bb97);
Score > 0
no
skip bbox
yes
draw bbox with class color



deepsystems.io



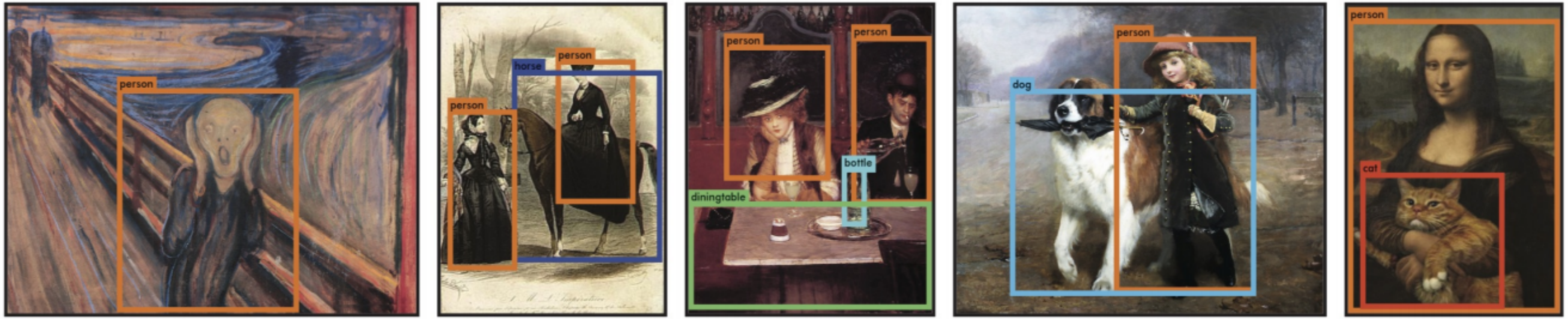
□□□□□□□□□ grid cell □□□□□□□ class □ bbox □

□□□□□□□□□ grid cell □□□□□□□ class □ bbox □

□□□□ , □ IOU □□

Key Points

1. Fast: YOLO - 45 fps, YOLO-tiny - 155 fps.
2. End-to-end training.
3. Makes more localization errors but is less likely to predict false positives on background
4. Performance is lower than the current state of the art.
5. Combined Fast R-CNN + YOLO model is one of the highest performing detection methods.
6. Learns very general representations of objects: it outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.



Darknet on Windows

Darknet https://github.com/pjreddie/darknet

Darknet for Windows https://github.com/AlexeyAB/yolo-windows

Darknet for Win tricks https://github.com/pjreddie/darknet/issues/721

Darknet install https://pjreddie.com/darknet/install/

Yolo v1 https://pjreddie.com/darknet/yolov1/

Yolo v2 https://pjreddie.com/darknet/yolov2/

Yolo v3 https://pjreddie.com/darknet/yolo/