Congratulations! You passed!
TO PASS 80% or higher

Keep Learning

GRADE
80%

Detection Algorithms

LATEST SUBMISSION GRADE
80%

1. You are building a 3-class object classification and localization algorithm. The classes are: pedestrian (c=1), car (c=2), motorcycle (c=3). What should y be for the image below? Remember that "?" means "don't care", which means that the neural network loss function won't care what the neural network gives for that component of the output. Recall $y = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$.

1 / 1 point

- $y = [?, ?, ?, ?, ?, ?, ?, ?]$
- $y = [0, ?, ?, ?, ?, 0, 0, 0]$
- $y = [1, ?, ?, ?, ?, ?, ?, ?]$
- ◉ $y = [0, ?, ?, ?, ?, ?, ?, ?]$
- $y = [1, ?, ?, ?, ?, 0, 0, 0]$

Correct
Correct.

2. You are working on a factory automation task. Your system will see a can of soft-drink coming down a conveyor belt, and you want it to take a picture and decide whether (i) there is a soft-drink can in the image, and if so (ii) its bounding box. Since the soft-drink can is round, the bounding box is always square, and the soft drink can always appear as the same size in the image. There is at most one soft drink can in each image. Here're some typical images in your training set:

0 / 1 point

What is the most appropriate set of output units for your neural network?

- Logistic unit (for classifying if there is a soft-drink can in the image)
- ◉ Logistic unit, b_x, b_y, b_h (since $b_w = b_h$)
- Logistic unit, b_x and b_y

Logistic unit, b_x, b_y, b_h, b_w

Incorrect
Do you really need b_h? given that the bounding boxes will all be the same size.

3. If you build a neural network that inputs a picture of a person's face and outputs N landmarks on the face (assume the input image always contains exactly one face), how many output units will the network have? **1 / 1 point**

- 3N
- N
- N^2
- 2N

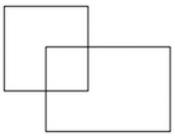
Correct
Correct

4. When training one of the object detection systems described in lecture, you need a training set that contains many pictures of the object(s) you wish to detect. However, bounding boxes do not need to be provided in the training set, since the algorithm can learn to detect the objects by itself. **1 / 1 point**

- False
- True

Correct
Correct, you need bounding boxes in the training set. Your loss function should try to match the predictions for the bounding boxes to the true bounding boxes from the training set.

5. What is the IoU between these two boxes? The upper-left box is 2x2, and the lower-right box is 2x3. The overlapping region is 1x1. **1 / 1 point**



- ⅙
- 1/10
- None of the above
- 1/9

Correct
Correct, the left box's area is 4 while the right box 's is 6. Their intersection's area is 1. So their union's area is 4 + 6 - 1 = 9 which leads to an intersection over union of 1/9.

6. Suppose you run non-max suppression on the predicted boxes above. The parameters you use for non-max suppression are that boxes with probability ≤ 0.4 are discarded, and the IoU threshold for deciding if two boxes overlap is 0.5. How many boxes will remain after non-max suppression? **1 / 1 point**

- 7
- 6
- 3
- 4
- 5 ✓

Correct
Correct!

7. Suppose you are using YOLO on a 19x19 grid, on a detection problem with 20 classes, and with 5 anchor boxes. During training, for each image you will need to construct an output volume y as the target value for the neural network; this corresponds to the last layer of the neural network. (y may include some "?", or "don't cares"). What is the dimension of this output volume? **0 / 1 point**

- 19x19x(25x20) ✓
- 19x19x(5x20)
- 19x19x(20x25)
- 19x19x(5x25)

Incorrect
How many anchor boxes are you using? How is one box defined?

8. What is Semantic Segmentation? **1 / 1 point**

- Locating objects in an image belonging to different classes by drawing bounding boxes around them.
- Locating an object in an image belonging to a certain class by drawing a bounding box around it.
- Locating objects in an image by predicting each pixel as to which class it belongs to. ✓

Correct

9. Using the concept of Transpose Convolution, fill in the values of **X**, **Y** and **Z** below. **1 / 1 point**

(*padding = 1, stride = 2*)

Input: 2x2

1	2
3	4

Filter: 3x3

1	0	-1
1	0	-1

1	0	-1

Result: 6x6

	0	1	0	-2	
	0	**X**	0	**Y**	
	0	1	0	**Z**	
	0	1	0	-4	

- (●) X = 2, Y = -6, Z = -4
- ( ) X = -2, Y = -6, Z = -4
- ( ) X = 2, Y = 6, Z = 4
- ( ) X = 2, Y = -6, Z = 4

✓ **Correct**

10. Suppose your input to an U-Net architecture is h x w x 3, where 3 denotes your number of channels (RGB). What will be the dimension of your output ? **1 / 1 point**

- ( ) D: h x w x n, where n = number of of output channels
- (●) h x w x n, where n = number of output classes
- ( ) h x w x n, where n = number of input channels
- ( ) h x w x n, where n = number of filters used in the algorithm

✓ **Correct**