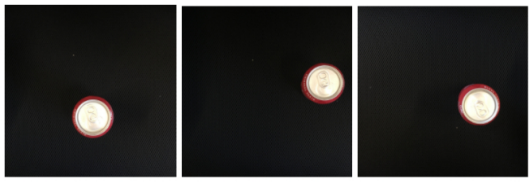
1. You are building a 3-class object classification and localization algorithm. The classes are: pedestrian (c=1), car (c=2), motorcycle (c=3). What should y be for the image below? Remember that "?" means "don't care", which means that the neural network loss function won't care what the neural network 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 point

- $y = [1, 0.22, 0.5, 0.2, 0.3, ?, ?, 1]$
- $y = [1, 0.22, 0.5, 0.2, 0.3, 0, 0, 0]$
- $y = [1, 0.22, 0.5, 0.2, 0.3, 0, 0, 1]$
- //www.pexels.com/es-es/foto/fotografia-de-motocicleta-clasica-en-carretera-995487/
- $y = [1, 0.22, 0.5, 0.2, 0.3, 1, 1, 1]$

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 the same size in the image. There is at most one soft-drink can in each image. Here are some typical images in your training set: 1 point



The most adequate output for a network to do the required task is $y = [p_c, b_x, b_y, b_h, b_w, c_1]$. (Which of the following do you agree with the most?)

- True, since this is a localization problem.
- True, p_c indicates the presence of an object of interest, b_x, b_y, b_h, b_w indicate the position of the object and its bounding box, and c_1 indicates the probability of there being a can of soft-drink.
- False, we don't need b_h, b_w since the cans are all the same size.
- False, since we only need two values c_1 for no soft-drink can and c_2 for soft-drink can.

3. When building a neural network that inputs a picture of a person's face and outputs N landmarks on the face (assume that the input image contains exactly one face), we need two coordinates for each landmark, thus we need 2N output units. True/False? 1 point

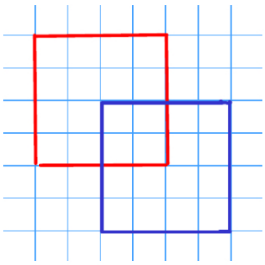
- False
- True

4. You are working to create an object detection system, like the ones described in the lectures, to locate cats in a room. To have more data with which to train, you search on the internet and find a large number of cat photos. 1 point

Which of the following is true about the system?

- We should use the internet images in the dev and test set since we don't have bounding boxes.
- We should add the internet images (without the presence of bounding boxes in them) to the train set.
- We can't use internet images because it changes the distribution of the dataset.
- We can't add the internet images unless they have bounding boxes.

5. What is the IoU between the red box and the blue box in the following figure? Assume that all the squares have the same measurements. 1 point



- 1

- 7
- $\frac{1}{8}$
- $\frac{1}{4}$
- $\frac{1}{2}$

6. Suppose you run non-max suppression on the predicted boxes nelow. The parameters you use for non-max suppression are that boxes with probability ≤ 0.7 are discarded, and the IoU threshold for deciding if two boxes overlap is 0.5. 1 point

After non-max suppression, only three boxes remain. True/False?

- False
- True

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? 1 point

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

8. What is Semantic Segmentation? 1 point

- Locating objects in an image by predicting each pixel as to which class it belongs to.
- 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.

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

(padding = 1, stride = 2)

Input: 2x2

1	3
2	4

Filter: 3x3

1	0	1
0	0	0
1	0	1

Result: 6x6

	0	0	0	0	
	0	**X**	0	7	
	0	0	0	**Y**	
	0	**Z**	0	4	

- X = 4, Y = 3, Z = 2
- X = 10, Y = 0, Z = 6
- X = 3, Y = 0, Z = 4
- X = 10, Y = 0, Z = 0

10. When using the U-Net architecture with an input $h \times w \times c$, where c denotes the number of channels, the output will always have the shape $h \times w$. True/False? 1 point

- False
- True