Object Localization

Image classification takes an image and classify it in a given set of classes. Classification with localization also pinpoint the bound box of the localization of the object in the image. The object detection usually works out with multiple tasks at the same time.

Localization can be taught to the network with the output parameters being set as b_x, b_y for the bounding box middle point and b_w, b_h for the width and height respectively. Therefore the output of the network now has 4 extra outputs which determine the position of the object in the given image. To help the learning the values are normalized using one of the corners (as the top left) as $(0,0)$ and the opposite corner as $(1,1)$.

The loss here is only calculated when an object is detected. When no object is detected we don't care for the bounding box positions calculated and those are not taken into the loss of that particular step.

Landmark Detection

Landmark detection basically is the subset of localization where you pinpoint the localization of several points in the image at the same time, like in the Snapchat filter where it pinpoints the faces and uses that input to apply a filter.

Object Detection

Using slide window detection you can build a ConvNet that detects a given object using a small sample of image and use a sliding window to classify over a bigger image. Given different sizes and strides of sliding windows you can detect the position of objects at the cost of high computational cost if the windows are sequentially processed.

Convolutional Implementation of Sliding Windows

If instead of using fully connected layers you had the same filters from the last convolution but with the number of channels set to the number of neurons you would like to have in the next fully connected layer the output would be a $1 \times 1 \times n_c$ where n_c = number of neurons you would have in the fully connected layer. Using this same technique you can follow up using 1×1 filters and setting the following n_c as the number of neurons you would have in the subsequent layers until the last layer where n_c would be the same size as your outputs.

Now if we had an input which is bigger than our expected network would output we could simply use this bigger input and get a bigger output which would correspond exactly to the computation of all subsamples in the given input. If our input would have been processed 4 times to fit the original input that means the output would be 4 times bigger than a single output. As the convolution operations share a lot of operations is much faster to compute all of them together than separately.

Paper: https://arxiv.org/pdf/1312.6229.pdf

Bounding Box Predictions

The YOLO algorithm divides the image in a given grid and classify each cell of the grid separately. If the image is divided in a 3×3 grid and the regular output had 8 values then the new output would be a $3 \times 3 \times 8$ volume. Remember that we only care for the bounding box prediction on the cells that have been classified as some object and all the others are don't cares.

As we're now classifying inside the defined bounding cells we expect our b_x, b_y values to fit inside our cells, therefore varying between $(0,0)$ and $(1,1)$ corresponding to the corners of the bounding box. The values of b_w, b_w although can vary on values bigger than 1 as the bounding box can be over the selected grid.

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

Intersection Over Union

Intersection over union is a metric to measure the performance of bounding box prediction tasks. Given the ground truth bounding box and the predicted bounding box it computes the intersection area over union of these two bounding boxes. By convention if the IoU is greater than 0.5 the bounding box is correctly being classified.

Non-max Suppression

As the grid gets finer it's possible that multiple cells detect the object on them and end up firing the detection of the bounding box on mutiple places. The non-max supression technique chooses only the highest bounding box in the classification to output as result.

Suppose a 19×19 grid where there are 391 possible outputs. The algorithm proceeds as follows:

```
discard all the boxes with p_c ≤ 0.6

While there are remaining box

   Pick the box with the largest p_c and output that as prediction

   Discard any remaining box with IoU ≤ 0.5 with the box out within the last
step
```

Anchor Boxes

Define some anchor boxes, for example, two: one wider and one taller. Now your output will double as you want to predict each anchor box separately. The previous output was $y = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$ for a given set of 3 classes. The new output will be $y = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3, p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$ where each half will be interpreted as separates bounding boxes.

YOLO Algorithm

y has shape $n_{\text{cells-width}} \times n_{\text{cells-height}} \times n_{\text{bounding-boxes}} \times (n_{\text{classes}} + 5)$. The 5 in the number of classes comes from the p_c, b_x, b_y, b_h, b_w terms

Region Proposals

It tries to pick a few regions that makes sense to run your continent crossfire. It performs a segmentation algorithm to gain better insights of where objects could be and then run the windows on those regions.

Paper: https://arxiv.org/pdf/1311.2524.pdf

Quiz

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 would be the label for the following image? Recall $y = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$

- [x] y=[1,0.3,0.7,0.3,0.3,0,1,0]
- [ ] y=[1,0.7,0.5,0.3,0.3,0,1,0]
- [ ] y=[1,0.3,0.7,0.5,0.5,0,1,0]
- [ ] y=[1,0.3,0.7,0.5,0.5,1,0,0]
- [ ] y=[0,0.2,0.4,0.5,0.5,0,1,0]

2.Continuing from the previous problem, 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.

- [ ] y=[?,?,?,?,?,?,?,?]
- [ ] y=[0,?,?,?,?,0,0,0]
- [ ] y=[1,?,?,?,?,?,?,?]
- [x] y=[0,?,?,?,?,?,?,?]
- [ ] y=[1,?,?,?,?,0,0,0]

3.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 appears as 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:(**Note** As the cans all have the same size there's no gain in learning this parameter as well, therefore b_h , b_w don't have much use in this case)

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

4.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? (**Note**: It's 3N on the output because you'll need to output the x, y and probability)

- [ ] N
- [x] 2N
- [ ] 3N
- [ ] N^2

5.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.

- ☐ True
- ☑ False

6.Suppose you are applying a sliding windows classifier (non-convolutional implementation). Increasing the stride would tend to increase accuracy, but decrease computational cost.

- ☐ True
- ☑ False

7.In the YOLO algorithm, at training time, only one cell —the one containing the center/midpoint of an object— is responsible for detecting this object.(**Note** Even thought the object might be detected in more than one bounding box the 'winner' will always be the one with its midpoint since all the other boxes would lead to a smaller IoU and be cut out in the process)

- ☑ True
- ☐ False

8.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.

- ☐ 1/6
- ☑ 1/9
- ☐ 1/10
- ☐ None of the above

9.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?

- ☐ 3
- ☐ 4
- ☑ 5
- ☐ 6
- ☐ 7

10.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 yy as the target value for the neural network; this corresponds to the last layer of the neural network. (yy may include some "?", or "don't cares"). What is the dimension of this output volume?

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