Assignment-3 Object Detection

Abhishek Kumar 18111002
Hemant Parihar 18111021

March 2019

Abstract

Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the advent of deep learning techniques, the accuracy for object detection has increased drastically. The network is trained on the most challenging publicly available dataset (PASCAL VOC 2007). As mentioned in the assignment problem statement we use resnet 18 to detect bounding boxes around the object. Then use non-maximum supression to remove those boxes which are detecting the same object with less probability. We also use two layers detection (multi-scale feature maps) to detect objects independently as in SSD.

1 Introduction

In Image classification, we need to predict the class of the image. A slightly complicated problem is that of image localization, where the image contains a single object and the system should predict the class of the object as well as location of the object in the image (a bounding box around the object). In object detection we need to correctly classify the image as well as localize the objects present in an image (multiple object can be there).

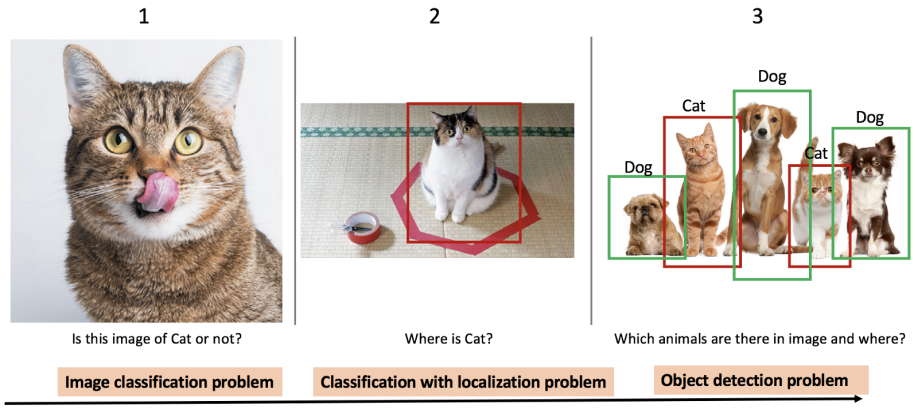


Figure 1: Source: Google Image

2 Challenges faced

2.1 Background

What is the right way to take the background (hard negative class) ? How many images we need to take for the background?
We tried different-different number of images for the background and observe the result vary with that. Also we tried different-different ways to take the background images.

2.2 Sliding Window

Test data contains object with very different sizes. Some object cover the entire image, some objects are very tiny (like bottles). How many windows we need to take and the size of those window? Also, more the number of windows more time will be consumed. So we tried different different aspect ratio of window and see on which we are getting good results. As there are different scale and orientation of objects fixing the number of sliding windows was a trade off we had to make.

2.3 Classification Using ResNet

As it seems the sliding window approach highly depends on classification accuracy and if the model is very highly accurate then it will work like wonders or else it won't but accuracy seems to be low around 89 percent which leads to many wrong predictions with high confidence and leads to poor performance.

2.4 Hyperparameters

Setting the hyperparameters for getting good boxes is very difficult as there is tradeoff between the stride of sliding window and number of correct bounding boxes obtained. Also setting the thresholds for classification and non maxima suppression also is very classifier dependent so optimizing models for these things were a bit challenging.

2.5 2nd detection (before the last layer)

From where we need to take the output of the middle layer. So we tried different different middle layer to see what changes are coming in the result.

Also when we wanted to delete a layer if it is a sequential block, then the entire layers present in that sequential block is deleted. But we wanted to delete only few layer inside the sequential block.

3 Detail Steps

3.1 Build Data Function

In build Data function we build our train and test data set. Using the annotation of the object we crop the object from the image.

For Background class we crop the random portion of the image that doesn't contain any of the objects.

3.1.1 Train the network

Then we train the Model_1 which is resnet 18 (through the last layer) with some fine turning, For model_2 we remove the few last layer from the resnet 18 so that we can take the output from the middle layer.

3.1.2 Sliding Window

After our models are trained on the training data. We take the images from the test data and slide the different different window over the images and got a patch. Then we transform the patch according to the model inputs. And give the patch to the model for the prediction. If our model predict that patch to be one of the 3 classes with probability greater than some threshold. Then we accept that patch as one of the candidate where object can be present. Then we

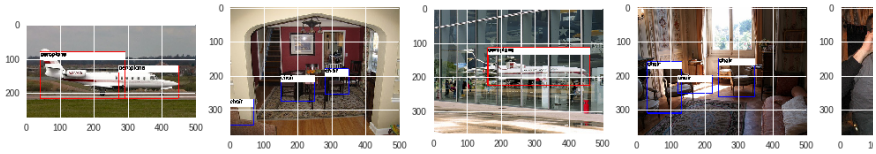
use non-maximum supression to make sure that for an object only one bounding box will be there.

3.1.3 non-max supression

In non-maximum supression, we take the box that have highest probability. Then we find other boxes which have Intersection over union with that box greater than some threshold, then we supress those boxes. Then from the remaining boxes we again take the box with the high probability and repeat the process.

4 Results

Here we have shown some bounding boxes but required ones can be located in jupyter file.



Map's obtained :

mAPs	Model Single layer	Model Two layers
Airplane	0.125	0.2083
Chair	0.2	0.11035
Bottle	0.0312	0.0471232
Moving Average	0.1168	0.12015

Conclusion

Two layer detection may help a bit but in this sliding window method since we are using alot of approximate hyperparameters (unoptimized), so it does not help also bounding boxes obtained are not very good as compared what we can if we do regression.

5 References

1. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173
2. https://www.pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/