IOTIOT

Co-Op Internship Report

Grocery Product Recognition

NOVEMBER- DECEMBER 2020

Rishav Dash

EXTC, RAIT

Table of content

Data Preprocessing

Dataset Used - Mvtec d2s: Densely segmented supermarket dataset

Annotation format - Bounding box object detection, TFRecords

The following experiments were conducted on the Mvtec d2s dataset. Due to the large size of images present in the original dataset, which is mostly because of the various augmentations and settings of the objects, we have utilized only subsets in our experiments.

In an attempt to focus on bounding box detection of the objects only, the d2s segmentation annotations were converted to bounding box annotations using **Roboflow**.

The images were in the format of COCO JSON which was needed to be converted **TFRecords** as we were using Tensorflow Object Detection API. Using Roboflow it was easily converted from COCO JSON to TFRecords format.

The TFRecords format included images with .tfrecord extension and label_map with .pbtxt extension

The data generated can be used by a link provided by Roboflow.

The Dataset consisted of 7980 images. It is split into 4380 training images, 1800 validation, and 1800 test images. The training set was not including any occluded images.

Model Used

Model Chosen - Faster RCNN with Inception Resnet v2

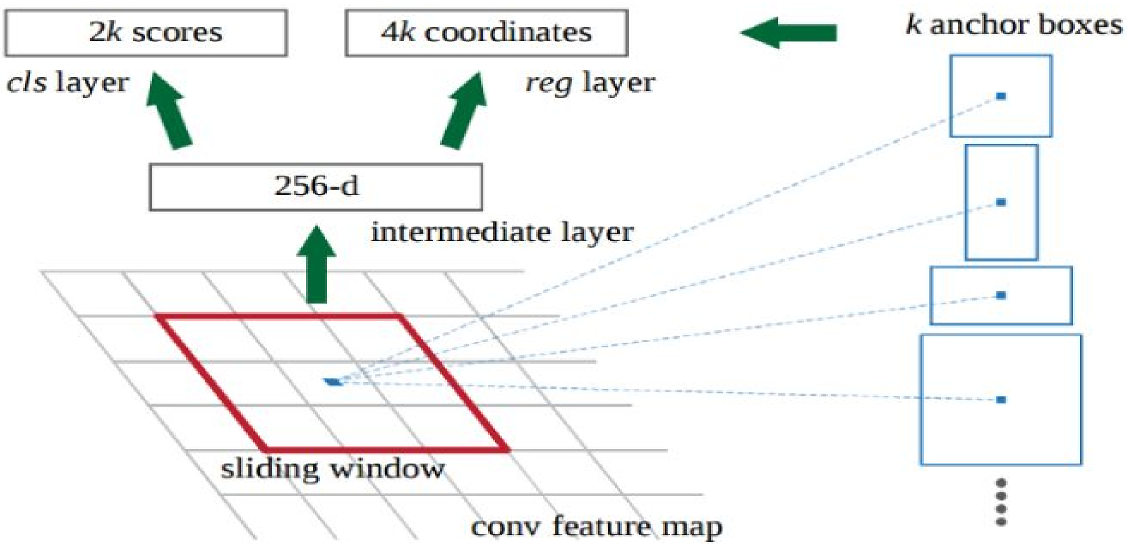
To train our model we were using TensorFlow Object Detection API. The model architecture is one of many available via TensorFlow's model zoo.

Faster RCNN working-

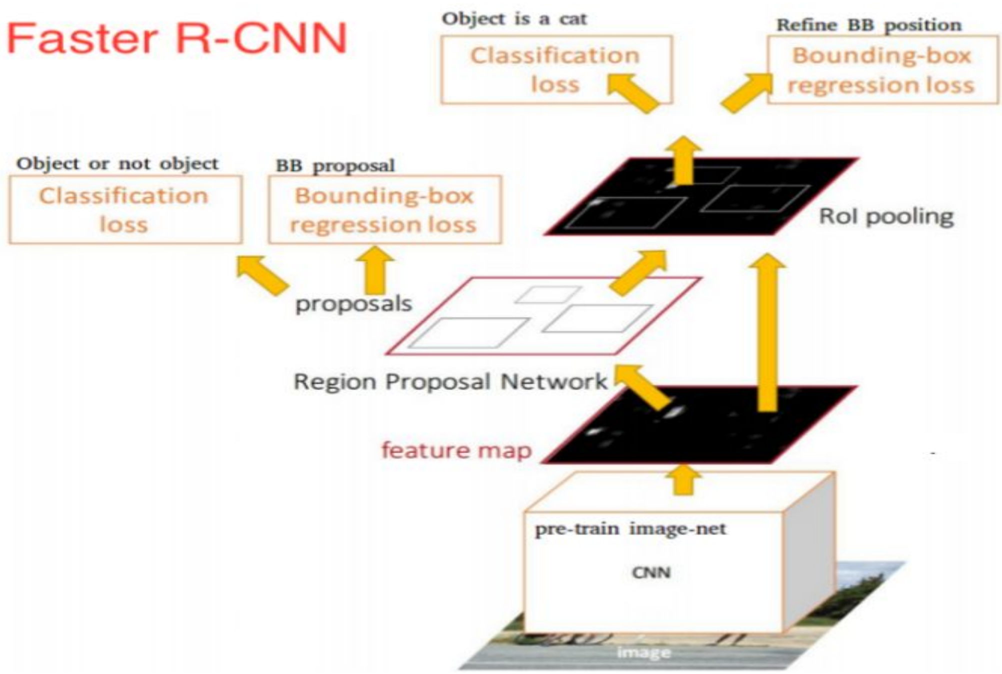
The changes that Faster RCNN had from the previous RCNNs were the region proposal network (RPN).

- At the last layer of an initial CNN, a 3x3 sliding window moves across the feature map and maps it to a lower dimension (e.g. 256-d)
- For each sliding-window location, it generates multiple possible regions based on k fixed-ratio anchor boxes (default bounding boxes)
- Each region proposal consists of a) an “objectness” score for that region and b) 4 coordinates representing the bounding box of the region

In other words, we look at each location in our last feature map and consider k different boxes centered around it: a tall box, a wide box, a large box, etc. For each of those boxes, we output whether or not we think it contains an object, and what the coordinates for that box are. This is what it looks like at one sliding window location:



In a sense, Faster R-CNN = RPN + Fast R-CNN



Training and Testing

Table 1. a) For 1st subset

Iterations - **5000**

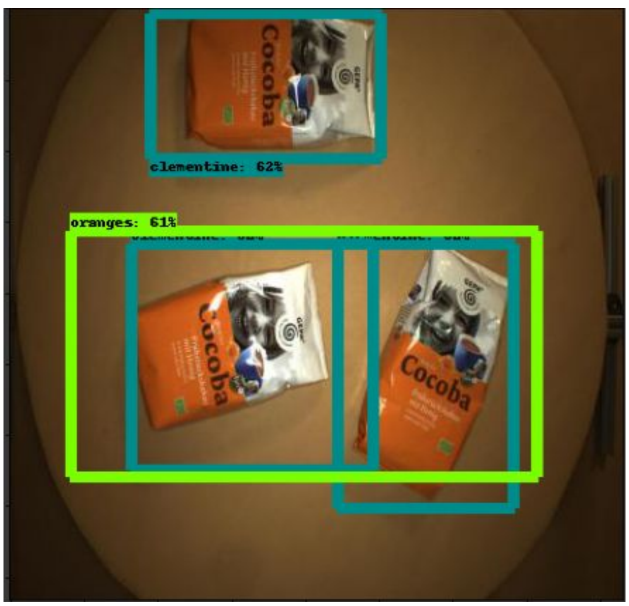
Average Precision (AP)	area	maxDets	**mAP**
IoU=0.50:0.95	all	100	0.094
IoU=0.50	all	100	**0.224**
IoU=0.75	all	100	0.040
IoU=0.50:0.95	small	100	0.000
IoU=0.50:0.95	medium	100	0.000
IoU=0.50:0.95	large	100	0.136

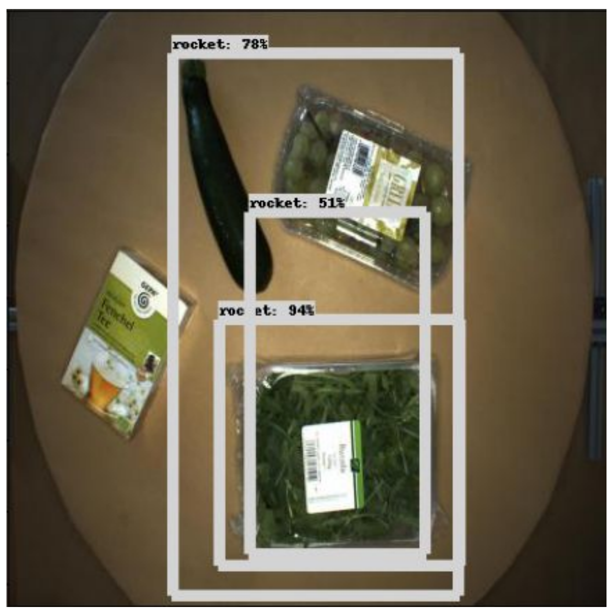
Output on test images-

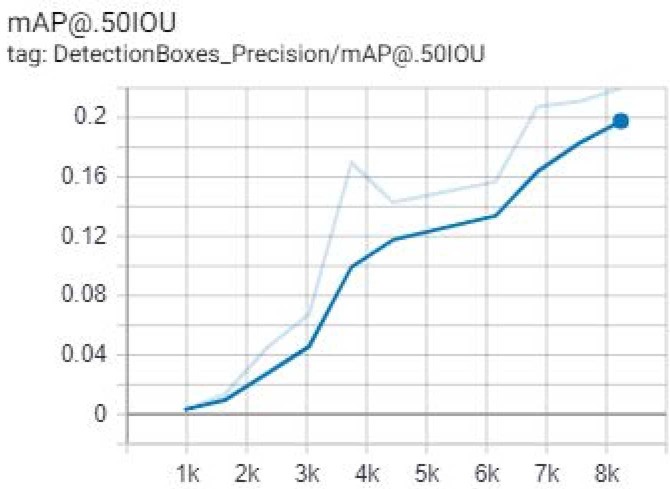
Iterations – **10000**

Average Precision (AP)	Area	maxDets	**mAP**
IoU=0.50:0.95	all	100	0.102
IoU=0.50	all	100	0.244
IoU=0.75	all	100	0.070
IoU=0.50:0.95	small	100	0.000
IoU=0.50:0.95	medium	100	0.000
IoU=0.50:0.95	large	100	0.147

Output on test images-







Conclusion

Looking at the output and the performance matrix result it can be told that the model is not trained properly. Although, after training it for 10000 epochs its gave an mAP value of 24.4% which is quite low for a bounding box detection model. The main challenge was to increase the mAP by setting up all the hyperparameters in the config file. Although, Faster RCNNs is slow but after heavy training the accuracy increase.

References

1. MVTec D2S paper: https://arxiv.org/pdf/1804.08292
2. Roboflow: https://roboflow.com/
3. RoboflowArticle:https://blog.roboflow.com/training-a-tensorflow-faster-r-cnn-object-detection-model-on-you-own-dataset/
4. RCNNs:https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9