Overview:

The given problem was to train and use an object detection model for detecting the number of products present in a shelf image.

After running the trained model on all test images, I have modified and stored 50 test images in the predictions folder for manual review.

Reproduction instructions:

The complete process takes about ~2 hours and the results can be reproduced on a system with >8GB RAM and a decent GPU with the following steps:

- Open cmd and move to the project directory (which contains the scripts`)
- Use `python data_prep.py`
- Open the research directory with `cd models/research`
- Now, use `cp object_detection/packages/tf2/setup.py .` and then `pip install . ` to install the object detection library
- Install protoc on your OS
- Use `protoc object_detection/protos/*.proto --python_out=.`
- `cd ../..` for heading back to the project directory
- Now, execute `python train_inference.py` to train the model and make predictions
- We can see the predicted test images (with bounding boxes) in the `predictions` folder.

Dataset preparation:

The dataset preparation as outlined in the `data_prep.py` includes:

- Downloading the required images, dataset and the necessary libraries
- Making Pandas csv data for training and test images with filename and annotation details
- Converting the csv files to TFRecord for training with Tensorflow model

Detection Network used:

For the given scenario, I used an EfficientDet D0 baseline model with a SSD (Single Shot Detector) framework with Tensorflow. The model performed pretty well with an mAP of 0.79.

Hyper-parameters used:

The complete config used for training the network can be found in the `training/effdet.config` file. The major hyper-parameters used included:

- IoU: An IoU of 0.25 was found be be the optimal value with this network
- Confidence Threshold: I used a threshold of 0.5 for filtering out any unwanted bounding boxes
- Num_iterations: 2000 epochs
- Batch size: 2
- Max_boxes: 100

Anchor Box tuning:

The model uses a multi-scale anchor generator, which can take into account the different shapes of products present in some images (some are close shot pictures, some are far away) and generate the anchor box accordingly.
After the model training, an average aspect ratio of 0.1/0.1 - 0.1/0.2 was found for the bounding boxes used by the model

The following results were achieved by the model after half the training (i.e. 1000 iterations) with an mAP of 0.513. Further training upto 2000-3000 iterations yields an mAP of around 0.79: