QPIAI Assignment

Problem:

8k resolution image captured from a drone contains a few pieces of litter (~20x20 or 30x30 pixels in size).
An example of the image:
https://www.shutterstock.com/image-photo/aerial-top-view-photo-taken-by-1358859473
Please note that the number of pieces of litter per image may be 3 to 5 on average.
The task is to draw bounding boxes on the litter pieces.

Solution

Overall

This problem is difficult to solve because the objects are really small in size as compared to the image.

There are a few things which can be done:

- Image tiling while both training and inference.
- Increasing the image width and height.
- Modifying the anchor box sizes according to small sized ground truth boxes
- Using different image augmentation techniques to increase the amount of data.

Let’s deep dive into details!

Basics for Object Detection

Object detection involves classifying localized bounding boxes in the image, that is classifying the objects and searching on the position for the bounding box. Through object detection mechanisms and algorithms, we are able to understand what's in an image, while being able to describe both what is in an image and the locations of those objects in the image.

On a predefined set of class labels (e.g. people and cars), object detection helps us describe the locations of each detected object in the image using a bounding box. The training data for object detection models include images and corresponding bounding box coordinates.

Data Preparation:

Input data for Pascal VOC is an **XML file**, whereas COCO dataset uses a **JSON file**.

In the data preparation steps we can do the following things:

- Dividing the image values by 255.0 (image normalization to reduce the computation complexity)
- Image tiling ie splitting the image into x*x grid
- Splitting images into train, test and val
- Resize the image and bounding box accordingly
- Perform data augmentation ie Mosaic, cut out, cut mix, etc to increase the amount of training data

Model:

There are mainly two types of models in object detection : One stage and two stage detectors.

Framework for two stage object detection networks (RCNN, Fast RCNN or Faster RCNN) is as follows:

1. Extracting regions of interest which are then warped to a fix size image, as is required by the CNN (with or without the RoI pooling layer)
2. Feature extraction is extracted by running a pretrained convolutional network on top of the region proposals.
3. A classifier such as SVM makes classification decisions based on extracted features.
4. Bounding box regression to predict location and size of the bounding box surrounding the object (using coordinates for box origin with dimensions of the bounding boxes)

Issues with the two stage detectors:

1. Slow inference and expensive training policy
2. Multi-stage pipeline training (CNN, classifier and regressor)

Thus, these days we are using one stage detector like Yolo, SSD, retina net.

SSD:

Working of the Single Shot Multibox Detection networks is based on these components:

1. Feature extractor convolutional network
2. Multi scaled feature layers which decrease in size progressively to allow for prediction of detections at multiple scales.
3. Non Maximum Suppression for elimination of overlapping bounding boxes, keeping only one box per each object detected.
4. In an effort to reduce objects belonging to the background class, we use hard negative mining which helps filter out anchor boxes that do not contain an object.

YOLO family:

You Only Look Once (YOLO) family of detection frameworks aim to build a real time object detector, which what they lack in small differences of accuracy when compared to the two stage detectors, are able to provide faster inferences.

YOLO does not go through a regional proposal phase (as was the case with two stage detectors), instead predicts over limited bounding boxes generated by splitting image into a grid of cells, with each cell being responsible for classification and generation of bounding boxes, which are then consolidated by NMS.

Steps:

1. Prediction of bounding box coordinates (cell location offsets: [x, y] and dimensions of bounding box: [width, height])
2. Objectness score which indicates the probability of the cell contains an object. (probability that box contains an object x IoU of prediction and ground truth)
3. Class Prediction using sigmoid/softmax: if bounding box contains an object- network predicts probability for K number of classes.

Steps to be followed whilst training the model to increase mAP(Mean Average Precision) for small sized objects:
- Change shape of anchor boxes according to the dataset.
- Using different loss functions (other than categorical cross entropy) - focal loss and poly loss

Metric

The **metric** used to determine model performance is **Mean Average Precision**. **Precision**, which is the measure of the percentage of correctly predicted labels, and **recall**, which is the measure of how well the model was able to fit the datapoints corresponding to the positive class, are along with IoU (Intersection over Union) which is the area of the overlap between our predictions and ground truth. A threshold is usually chosen to classify whether the prediction is a true positive or a false negative. **Average precision** is the area under the precision-recall curve and follows precision and recall in having a value between 0 and 1. Interpolation of the precision value for a recall by the maximum precision which makes the curve between precision and recall be less susceptible to small changes in ranking of the points. **Mean Average Precision (or mAP)** is calculated by average precision values for each class label.