Machine Learning Engineer Nanodegree

Capstone Proposal

Moinuddin Syed

Product Object Detection

Domain Background

Humans use their eyes and their brains to see and visually sense the world around them. Computer vision is the science that aims to give a similar, if not better, capability to a machine or computer. Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding [1]. Researchers realized that it was necessary to tackle images from the real world. Thus, much research was needed in the so called ``low-level" vision tasks such as edge detection and segmentation. Low-level image processing algorithms are applied to 2D images to obtain the ``primal sketch" (directed edge segments, etc.), from which a 2.5 D sketch of the scene is obtained using binocular stereo. Finally, high-level (structural analysis, a priori knowledge) techniques are used to get 3D model representations of the objects in the scene. This is probably the single most influential work in computer vision ever [2].

Object detection, an interesting challenge to identify the object in an image or series of frames of images, has seen a lot of its application in cutting edge technologies like self driving car, security systems etc. Many organisations have held challenges like, PASCAL Visual Objects Challenge (VOC) [3] and ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[4]

Out of all the applications, I personally was interested in using this technology of object detection on the e-commerce side, by identifying a specific object as a product.

Problem Statement

In this project I would like to apply deep learning technology (CNN), to identify an object in an image as a product for ecommerce , image search purposes. For example, given an image with multiple objects and if there is a laptop object in the image, the object identification model should identify the object in the image as a laptop, with its bounding box. Right now for this project the scope of product categories is electronic devices and accessories, I wish to later expand this to all categories.

Datasets and Inputs

The dataset I wish to use for this project is openimages dataset. I previously thought of using COCO dataset, but the labels/object classes are quite limited for the products, and PASCAL VOC has only 20 classes.

Open Images is a dataset of ~9 million URLs to images that have been annotated with image-level labels and bounding boxes spanning thousands of classes.Overall, there are 19,995 distinct classes with image-level labels (19,693 have at least one human-verified sample and 7870 have a sample in the machine-generated pool). Of these, 5000 classes are considered trainable.Overall, there are 600 distinct classes with a bounding box attached to at least one image. Of these, 545 classes are considered trainable (the intersection of the 600 boxable classes with the 5000 image-level trainable classes).[5]

Therefore, with this good number of bbox object classes, it made more sense to use openimages than other dataset for this particular problem.

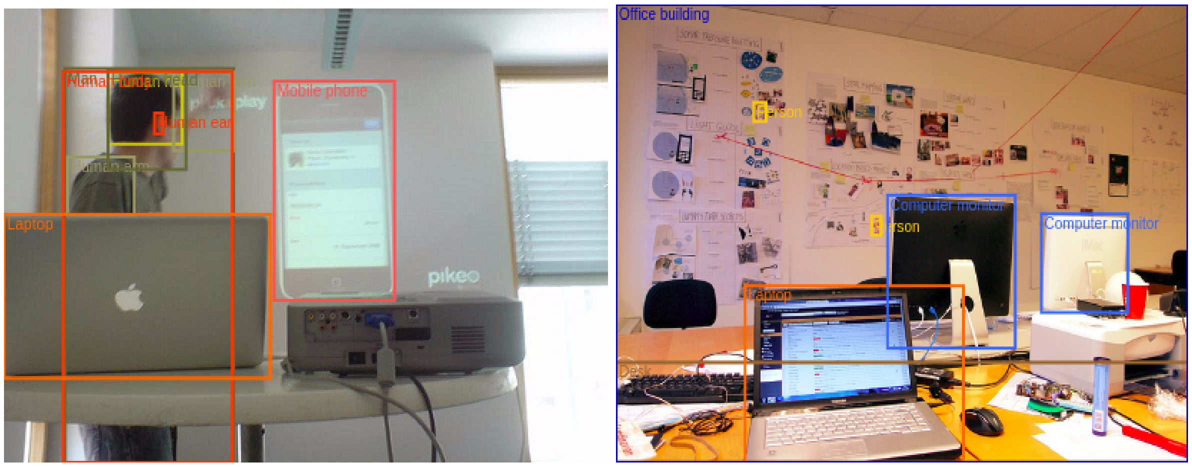

Figure 1. Images from openimages with laptop bbox

Solution Statement

CNN being the gold standard for image classification [6], something more than just classification is required for object detection.Object detection is the task of finding the different objects in an image and classifying them, this where Region based CNN (R-CNN) came into picture, to create boundary boxes or region proposals, for which a CNN generates features and then is classified and regressed to particular object class. But R-CNN also has been evolved along the time to more better and efficient versions, that is to Fast R-CNN and then to Faster R-CNN.[7]

In this project I would like to use Faster R-CNN to solve this problem, Faster R-CNN uses a Region proposal Network which is solely based on CNN. The intent is to apply this on openimages dataset and observe its performance and results.

Benchmark Model

I will be using the results obtained by tensorflow model for openimages dataset using Faster R-CNN as benchmark[8]. The results obtained by tensorflow was 37 mAP (mean Average Precision). As the data I'm using for electronic products and accessories will be much smaller, hence there is difference in the data the model is trained on. Therefore, as to what I infer there will be variations in the results.

Evaluation Metrics

The primary evaluation metric used widely for object detection is mAP (mean Average Precision). To calculate it for Object Detection, you calculate the average precision for each class in your data based on your model predictions. Average precision is related to the area under the precision-recall curve for a class. Then taking the mean of these average individual-class-precision gives you the Mean Average Precision.

This metric is defined originally for evaluating detector performance on Open Images dataset and is fairly similar to the PASCAL VOC 2007 metric. It computes interpolated average precision (AP) for each class and averages it among all classes (mAP).Open Images annotations contain ‘group-of’ ground-truth boxes(boxes containing group of same objects), that are treated differently for the purpose of deciding whether detections are "true positives", "ignored", "false positives" .[9]

Precision and recall are defined as:

- Precision = number-of-true-positives/(number-of-true-positives + number-of-false-positives)
- Recall = number-of-true-positives/number-of-non-group-of-boxes

Project Design

The Faster R-CNN model replaces the Fast R-CNN’s Selective Search algorithm with a Regional Proposal Network (RPN), an additional network that shares a common set of convolutional layers with the detector network. In a Faster R-CNN a single CNN is used to both carry out region proposals and classification.

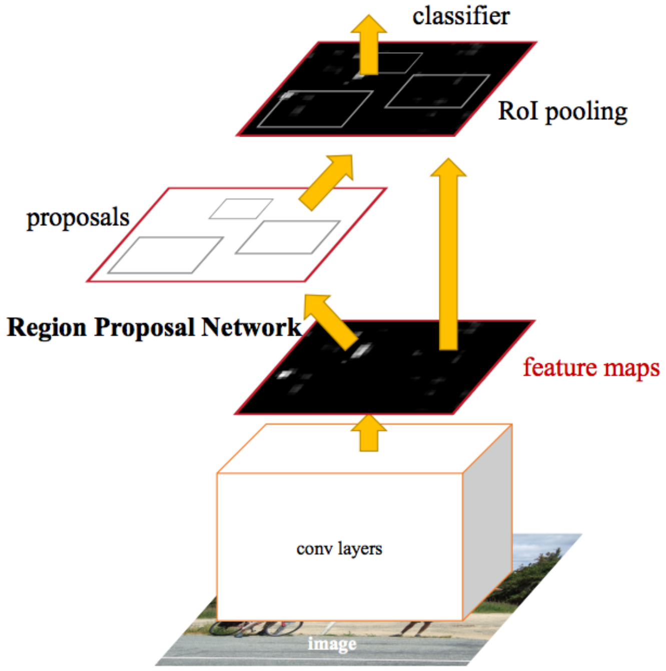

Figure 2. Workflow of Faster R-CNN

The workflow of the project is mentioned in the following steps,

1. Analyze the data to be downloaded. Collect the object classes of electronic products with trainable bboxes. Find the images containing these objects and make a record of images with their objects with bbox
2. Download these images from the source and store them with their unique object IDs , then create TFR records of these images.[11]
3. Train the data along with their bboxes, with pre trained Faster R-CNN model, trained on openimages dataset, using tensorflow object detection API[10].
4. Run the tests to obtain the mAP metrics.
5. Compare the results with the benchmarks talked about above.

Refrences

1. http://www.bmva.org/visionoverview
2. Huang, T. (1996-11-19). Vandoni, Carlo, E, ed. *Computer Vision : Evolution And Promise* (PDF).
3. PASCAL VOC ,http://host.robots.ox.ac.uk/pascal/VOC/
4. ILSVRC ,http://www.image-net.org/challenges/LSVRC/
5. openimages github, https://github.com/openimages/dataset
6. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton ,ImageNet Classification with Deep Convolutional Neural Networks
7. Object Localization and Detection, https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/object_localization_and_detection.html
8. Open Images-trained models, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md#open-images-models
9. Open Images metric, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/evaluation_protocols.md#open-images
10. "Speed/accuracy trade-offs for modern convolutional object detectors." Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K, CVPR 2017, https://arxiv.org/abs/1611.10012
11. TFR records, http://warmspringwinds.github.io/tensorflow/tf-slim/2016/12/21/tfrecords-guide/