Project Report

Toy Buses Object Detection Faster-RCNN with InceptionV2,MS-COCO Transfer Learning

Dean Shabi, Shani Ben Baruch

introduction

Object Detection is a computer vision problem that has been discussed for many years; the problem combines the classification and localization(location) of objects in different images/videos. This problem has many uses, such as facial recognition and autonomous driving.

The Deep Learning algorithms that have evolved over the past five years have resulted in a leap forward in the accuracy and quality of the solution to the problem and today enable the reach of the ability to identify similar to humans in Real Time.

There are currently a number of algorithms for object localization:Faster-RCNN(BASEDRCNN),YOLO,SSD. These algorithms use different methods to extract areas of interest (RPN per parable) from the image, and inject them into a classification network such as AlexNet, ResNet, VGG, Dense, to identify if the desired object exists in that area of interest.

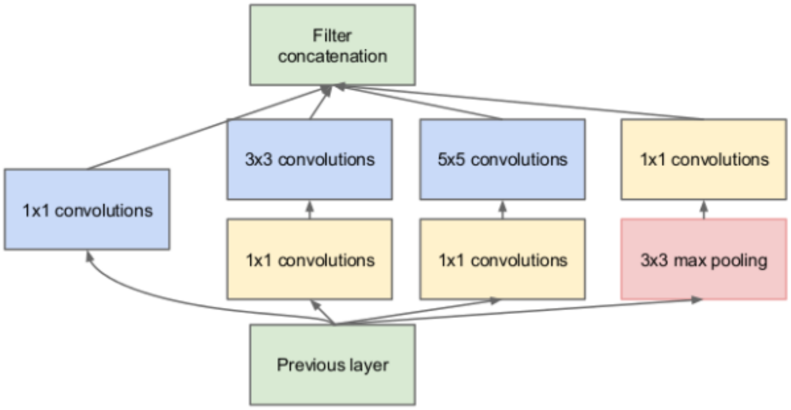
Challenge

This project received a database of 60 images of toy buses (below) tagged according to their color and placed in the image by The Boxing Boxes.

Developing the algorithm

1. We decided to use Deep Learning methods to solve the problem, both to learn the field and to achieve as good results as possible.
2. The size of the Dataset was a serious problem; We have very few photos and tags to get to the best classifications as the one that AlexNet received on ImageNet in 2012, by training from scratch. That's why we decided to use Transfer Learning.
3. Transfer Learning is weight sharing between deep neural networks.
    a. Instead of rebooting the CNN weights and re-training them (using a cost optimization and backpropagation function), you transfer weights (filters in CNN's case) from a network of the same type trained on a labeled image pool, and use them as Initial Weights in an identity network that will be trained on different Dataset.
    b. This is particularly effective because it has been found that taking the weights from the deep layers of a network that has been trained on any image pool, and resetting the top layers (which are problem-specific), followed by the tuning of the entire network, allows for training with a smaller amount of information and greater accuracy.
    c. The bottom layers tend to focus on the generic characteristics of images such as Edges, Blobs andCho, so taking weights that have "specialised" allows you to shorten training time and reduce the chances of overfitting to a small extent and the amount of new images is small.
4. As implied in the introduction, there are a large number of combinations and solutions to the problem. It is necessary to examine them all in aspects of accuracy, running time and complexity and decide on:

   a. **Detection Algorithm** - There are a number of good algorithms, with a number of good algorithms, with runtime, accuracy and network complexity.
   b. **Classification Network** - Various developments in the field of classification have made **ResNet** a priority over the VGG network, while Alexnet is a network that has been studied by many and there is a lot of documentation on the subject.
   c. **Weights for Transfer Learning**–online are shared by weights trained on different image databases,such as COCO,Kitti,Open Image Dataset. These weights can be taken and used for transfer learning.
   d. The configuration that was eventually selected is Faster-RCNN, which combines the InceptionV2 algorithm, with Transfer Learning from the COCO dataset.
5. Faster-RCNN is an algorithm built from multiple layers:
   a. Run the image into a CNN to get vector features/properties.
   b. InceptionV2-filters of CNN networks can only learn linear characteristics from the Inputs. Therefore, Google has created a new network architecture by creating more complex filters. instead of linear convolutions, connect POOL,1x1,3x3,5x5 combinations in a serial and parallel form to create a complex property vector, yet a relatively low amount of parameters across the entire network.



   c. Region Proposal Network (RPN)–A separate network that removes areas of interest from the image by guessing whether a particular "anchor" in the image is a "background" or "object" (an anchor is a certain size, there are 9 anchors of different sizes around each point in the image). In order not to receive duplicate IDs, theboxes/ areas of interest received from operators cann-Max Suppression to keep only the most certain offers.
   d. Bounding Box Regressor–For anchors labeled as an "object" according to theRPN, classicification + regression for the LOCATION of the BBV is a price function that includes the classification price and regression price.

The training process

1. Increase dataset:
   a. As mentioned the amount of photos we received was very small. Therefore, in order to increase the image pool and consequently obtain better identifications, we used different types of ogmentations such as: rotation, transalation and added gaussie noise.
   b. In order to obtain different types of augmentations, we used ImgAug (which enabled you to receive diverse images) and LabalImg (which allowed the resulting images to be tagged).
   c. Other augmentations we tried were clarifications and blackouts, "hiding" some of the images and "hiding" some of the images and shear to the image; these augmentations damaged the accuracy of the algorithm on the set.
   d. The final product was 607 different images (excluding the source images) and XML files that contained the locations of the various buses and their color.

2. Splitting all the images into train and test. Our network was trained on 450 images in total.
3. XML to CSV:
    a. In order to use the information contained in the modified XML files, we required one of the same twoCSV files. One for the train and one for the test.
    b. Each line in the CSV file contained the image name, the location of the bus in the image, and its color.
4. Creates TFRecords for each CSV file.
5. Preparingconfiguration file for the model we used (Faster RCNN Inception V2 using Coco dataset).
6. Training with The Tensorflow Object Detection API.
    a. The training was carried out in the fine-tune method for all parameters, instead of freezing the weights from coco and training only to the upper layers.
    b. This method is preferable in the case of AnB Transfer learning, a network that has been trained on images of a specific type in the original and is required to identify images of a different type.
7. Export inference graph from the training stage we reached and the config file we have prepared.
8. Check the test images.

Problems discovered during training

1. RAM issues: To resolve this issue we used
    a. Batch size is small (and more specifically the batch size was equal to one).
    b. Lines of code that have been added to the config file aimed at limiting runtime and memory resources.
2. Identification issues for the yellow buses:
    a. Our network has difficulty recognizing the yellow color and sometimes it is classified as red/green.
    b. Theoretically, this problem can be overcome if you trust additional images with a varying level of brightness.
    c. In practice, when we created a wider image pool, we severely impaired the identification capabilities of the network, so we decided not to use the enlarged repository.

Create the Test File for the Inference Test

1. We wanted to simplify the test file so that it was simple to understand and to run as fast as possible.
2. We created a new file by using the original test file provided in the classroom and using the test file provided by the API.
3. Algorithm steps:
    a. Defines the inference graph with which we want to work.
    b. Go over each image from the relevant route and find the buses by orders from Thesorflow and the graph we used.
    c. Screening of low-grade buses – we ignored buses that scored below 50%.
    d. Filtering buses of the same color – there are six different colors in total and therefore an image with two red buses for example makes no sense. In such a situation we left the identification with the highest score of the two and ignored the second identification.
    e. Screening buses with a large overlapping ratio(IOU owners > 0.6)– the buses take up space in the space, so if we have identified two BUSES with a large overlap ratio, we can conclude that this is a double detection of the same bus.
    f. The remaining identifications we wrote in the format required for the filetxt we created.

Points for improvement in follow-up work:

1. Using one network to identify a bus, and another network to identify the color of the object– since we encountered cases of incorrect color recognition (especially the yellow),and yet the Boxes were relatively accurate, it is possible to train a separate network on the characteristic vector that will learn to recognize color, at the same time as identifying a "bus" by Faster RCNN.
2. Using a different type of network to Detection/Classification can improve performance in terms of running time and accuracy.
3. Training on weights from another image pool is used by ImageNet, or Kitti trained to detect vehicles.