Tips and Tricks

students: Aslan Ubingazhibov and Alisa Kugusheva

Lecturer: Nikolay Falaleev

Data Science
Higher School of Economics
Russia, Moscow
01-04-2021

1 Models

Since the task was real time object detection, it was obvious that the best choice is yolo. We tested only 2 models: yolo3 (106 layers) and tiny yolo (23 layers).

2 Hardware

We were using 3 different GPUs:
1) Tesla v100 (our friend's gpu)
2) GTX 1050
3) Google Colab

3 Dataset

We have found several datasets, like this, this2, this3, but we have decided to use bosch small traffic lights dataset. This dataset contains 13427 camera images at a resolution of 1280x720 pixels and contains about 24000 annotated traffic lights. The annotations include bounding boxes of traffic lights as well as the current state (active light) of each traffic light.

We have also split videos provided by the lecturer to around 1000 frames and annotated them using supervisely.

4 Experiments

4.1 First experiment

We used Bosch Small Traffic Lights Dataset to train YOLOv3-tiny (23 layers) using transfer learning, where we used the pre-trained VOC data and just changed the end of the deep-neural-network. We

1. Prepared the dataset (conversion of annotation format to suitable format, creating some Files that training requires and etc);
2. trained the model. We could not train 'big' YOLO since Google Colab was turning off too early;
3. Tested the model by visualization (didn't use the metric).

We think that the result wasn't good enough – the model was detecting not only traffic lights, but also sometimes just green, yellow and red objects. Also bounding boxes were too big, or too small. Check the videos: 1, 2.

It's been a headache to use this framework since it has some problems with OpenCV and problems with GPU implementation. So we have decided to choose different approach. The author himself has noted the problems with OenCV as you can see from Fig. 1. Train: resize the image (416, 416), leanring rate = 0.01, decay=0.0005.

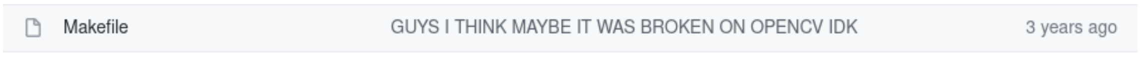
Makefile GUYS I THINK MAYBE IT WAS BROKEN ON OPENCV IDK 3 years ago

Figure 1: Darknet problem

The whole training has been provided using this github.

4.2 Second experiment

The second approach was to use a popular framework (mmdetection). MMDetection is an open-source object detection toolbox based on PyTorch. We have decided to use it because of its very good documentation and easiness in use. The main document we were using is a config file, where you basically set parameters of your model as well as parameters of training and testing pipeline. We have trained there 2 models: one just for 1000 images (frames from videos, which were provided by the lecturer) and one for the bosch dataset + around 800 images of that 1000 images. However, we had problems with speed in inference - we were able to reach maximum of 25 FPS using Tesla v100 GPU which is not enough (for GTX 1050 it was 10 fps). The problem was that the Docker didn't utilize all GPU (only 8-9%) and we couldn't fix it. I think it is important to note that here we haven't used transfer learning, but just trained whole pre-trained model. According to the authors: 'The results show that freezing speeds up training, but reduces final accuracy slightly. A full W&B Report of the runs can be found at this link.' Some augmentations we used - mmultiscale, padding (so the size of the image is divisible by 32), randomflip.

4.3 Third experiment

Finally, we used pytorch implemetation of YOLOv3. We trained again 2 models: tiny YOLOv3 and YOLOv3:
augmentations:
augmentation type: (probability) - translation: 0.1, scale: 0.8 - 1.5, horizontal flip: p = 0.5, mosaic 1.0

yolov3-tiny: mAP - 49.6 (70 fps on GTX 1050)

mosaic augmentation seems to work very well. (Mosaic augmentation - 4 random images are stitched together). Also, apart from improved accuracy, mosaic augmentation fascilitates fast convergence (8th epoch compared to 27th epoch for the experiments below)
no mosaic augmentation yolov3: mAP - 52.8
with mosaic augmentation yolov3: mAP - 61

5 Post processing - Real time Tracker

We used the tracker from Sort, which improved the metric little bit.