Selected Topics in Visual Recognition using Deep Learning
HW3 Street View House Numbers Detection Report

0856049 吳毓軒

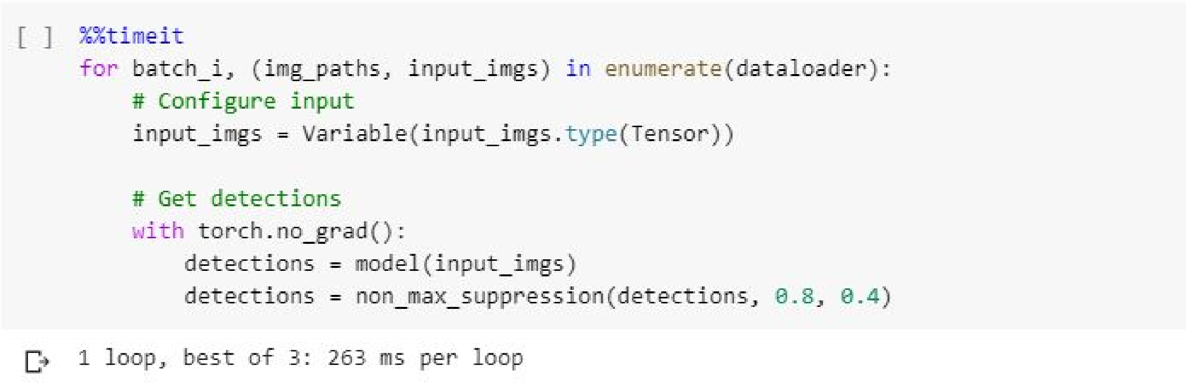
GitHub repository link

Link: https://github.com/yuhsuan1203/VRDL/tree/master/HW3

Reference if you used code from GitHub

Link: https://github.com/eriklindernoren/PyTorch-YOLOv3

Speed benchmark: **263ms**



```
[ ]  %%timeit
     for batch_i, (img_paths, input_imgs) in enumerate(dataloader):
         # Configure input
         input_imgs = Variable(input_imgs.type(Tensor))

         # Get detections
         with torch.no_grad():
             detections = model(input_imgs)
             detections = non_max_suppression(detections, 0.8, 0.4)
```

```
1 loop, best of 3: 263 ms per loop
```

Introduction

Object detection is an important task in many applications. In this homework, we need to train a convolutional neural network to detect the Street View House Numbers.

Methodology

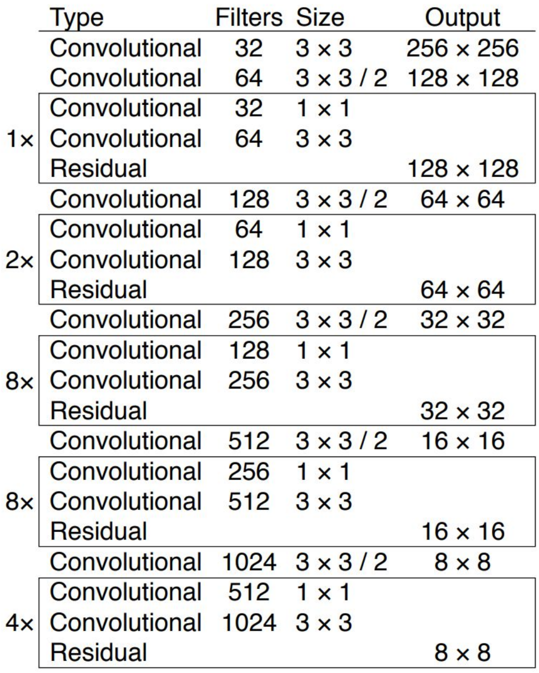
- Data Preprocess
  - Add padding and resize to the same size

    Each input image is added padding to become a square and then resized to 416 x 416.
  - Data augmentation

    Although some digits, such as 2, 3, and so on, look quite different if flipped horizontally, I still applied the augmentation for this problem. And in fact, the result does not change a lot if not applying this technique.

- Model Architecture

The model is based on YOLOv3. YOLOv3 makes detections at three different scales.In YOLOv3, Darknet-53 is used for feature extraction. After the feature extractor, several convolutional layers are added in order to predict the bounding box, objectness and the class. The model architecture of Darknet-53 is shown below.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8

Darknet-53

- Hyperparameters
  - epochs: 45 (20 ~ 50 is also good)
  - batch size: 8
  - Optimizer: Adam with learning rate=0.001

Summary & Findings

The result mAP is shown below. It is quite good but still has a lot of improvement.

mAP_0.45774_0856049.json

Findings

I submitted many different result files in this homework. In my final submission file, the object confidence threshold is 0.8 when inferencing. But in some other submitted files, the mAP of setting the threshold to 0.1 is slightly better than 0.8. Therefore, I think the confidence threshold is also an important parameter to be tuned.