Video Streaming and tracking

0712534 陳永承

- **Experiment Setup**
  envoriment: google colab
  model: YOLOv4
- **Briefly explain your code**
  YOLOv4: https://github.com/AlexeyAB/darknet
  Clone model from GitHub.

```
%cd '/content/gdrive/My Drive/VSAT_HW'
!git clone https://github.com/AlexeyAB/darknet
```

更改makefile內的參數後執行make編譯

```
%cd /content/gdrive/MyDrive/VSAT_HW/darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
```

在darknet/data中存入obj.name與obj.data，obj.name只要寫car，obj.data則寫入檔案位置

obj.names obj.data

```
1 classes=1
2 train=data/train.txt
3 valid=data/valid.txt
4 test=data/test.txt
5 names=data/obj.names
6 backup=backup/
```

更改yolov4的cfg，最後3層的yolo層class改為1，yolo前一層conv中的filter改成18

yolov4-obj.cfg

```
133 activation=leaky
134
135 [convolutional]
136 size=1
137 stride=1
138 pad=1
139 filters=18
140 activation=linear
141
142
143 [yolo]
144 mask = 6,7,8
145 anchors = 12, 16, 19, 36
146 classes=1
```

將data分別寫成train.txt/validation.txt/test.txt

```
import glob
import pandas as pd
import numpy as np
json_file_pathlist=glob.glob("data/training_dataset/*.jpg")
```

```
[ ] df=pd.DataFrame({"image_path":json_file_pathlist})
    df.head()
```

	image_path
0	data/training_dataset/1_000008.jpg
1	data/training_dataset/1_000010.jpg
2	data/training_dataset/1_000007.jpg
3	data/training_dataset/1_000009.jpg
4	data/training_dataset/1_000004.jpg

```
training_data=df.iloc[:]
print(training_data)
```

```
                                image_path
0      data/training_dataset/1_000008.jpg
1      data/training_dataset/1_000010.jpg
2      data/training_dataset/1_000007.jpg
3      data/training_dataset/1_000009.jpg
4      data/training_dataset/1_000004.jpg
```

```
training_data.to_csv("data/train.txt", index=None, header=None)
```

訓練yolov4

```
!chmod 777 darknet
!./darknet detector train data/obj.data cfg/yolov4-obj.cfg yolov4.conv.137 -dont_show
```

測試validation data

```
!chmod 777 darknet
!./darknet detector map data/obj.data cfg/yolov4-obj.cfg backup/yolov4-obj_best.weights
```

將testing data寫成result

```
!chmod 777 ./darknet
!./darknet detector test data/obj.data cfg/yolov4-obj.cfg backup/yolov4-obj_best.weights -dont_show -ext_output <data/test.txt> ./result
```

SE module part

YOLO v4: https://github.com/WongKinYiu/PyTorch_YOLOv4/

將model/models.py定義SE layer

models.py

```python
class SELayer(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SELayer, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.fc = nn.Sequential(nn.Linear(channel, channel // reduction),
                                nn.ReLU(inplace=True),
                                nn.Linear(channel // reduction, channel),
                                nn.Sigmoid())

    def forward(self, x):
        b, c, _, _ = x.size()
        y = self.avg_pool(x).view(b, c)
        y = self.fc(y).view(b, c, 1, 1)
        return x * y

```

更動cfg, 在兩conv間加上SE層

yolov4-obj_SE.cfg

```
activation=leaky

[SE]
reduction=16

##############################

[convolutional]
```

training yolo-v4 with SE layer

```
!python train.py --device 0 --batch-size 4 --data car.yaml --cfg cfg/yolov4.cfg --weights 'yolov4.conv.137' --name 'yolov4_SE'
```

- **Your validation results on your two model**

  Yolo v4 with SE module

  mAP@0.5: 0.904

```
Scanning labels data/valid_dataset.cache3 (240 found, 0 missing, 0 empty, 0 duplicate, for 240 images): 240it [00:00, 8414.34it/s]
               Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 30/30 [00:54<00:00,  1.83s/it]
                 all         240    1.92e+03       0.434       0.982       0.904       0.508
Speed: 92.9/2.3/95.2 ms inference/NMS/total per 640x640 image at batch-size 8
```

  Yolo v4 original module

  mAP@0.5: 0.945

```
IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.945043, or 94.50 %
```

- **Discussion**

  增加SE layer後的計算量似乎增加不少, 導致batch size要減少非常多, 否則GPU的記憶體空間會不夠用。

  增加SE layer後會讓運算時間增加, 在Colab的環境中不能訓練太久, 所以增加SE module後的epoch數少很多。

比對2種model的predict後發現，即使2 model的mAP沒有差太多，但增加SE module會預測多出很多物體，雖然ground truth沒有車子的confidient數值比有車子的少很多。推測是因為沒有加SE module的YOLO訓練時epoch多，且每次batch size也比較大，所以訓練比較多次，讓沒有物體的confidient大幅下降。

YOLO v4 with SE module

YOLO v4

- **reference**
  YOLOv4 darknet: https://github.com/AlexeyAB/darknet
  YOLOv4: https://github.com/WongKinYiu/PyTorch_YOLOv4/