HOMEWORK 2

OBJECT DETECTION

Le Van Hung (黎文雄)

Student ID: 0860831

1. Install darknet on ubuntu

unzip HW2.zip

cd HW2/

git clone https://github.com/pjreddie/darknet

cd darknet/

make

2. Copy data, weights and config file to darknet folder

cd ..

cp -r data darknet/

cp yolo-acv.cfg darknet/

cp yolo-acv.weights darknet/

3. Create bash script file

touch run_hw2.sh

vim run_hw2.sh

cat run_hw2.sh

#!/bin/bash

#create folder for save image

mkdir result_hw

#cd to darknet folder

```
cd darknet
#first image
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/1.jpg -thresh 0.02
cp predictions.jpg ../result_hw/image1_thresh002.jpg
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/1.jpg -thresh 0.4
cp predictions.jpg ../result_hw/image1_thresh04.jpg
#second image
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/2.jpg -thresh 0.02
cp predictions.jpg ../result_hw/image2_thresh002.jpg
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/2.jpg -thresh 0.4
cp predictions.jpg ../result_hw/image2_thresh04.jpg
#thrid image
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/3.jpg -thresh 0.02
cp predictions.jpg ../result_hw/image3_thresh002.jpg
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/3.jpg -thresh 0.4
cp predictions.jpg ../result_hw/image3_thresh04.jpg
#forth image
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/4.jpg -thresh 0.02
cp predictions.jpg ../result_hw/image4_thresh002.jpg
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/4.jpg -thresh 0.4
cp predictions.jpg ../result_hw/image4_thresh04.jpg
#fifth image
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/5.jpg -thresh 0.02
cp predictions.jpg ../result_hw/image5_thresh002.jpg
./darknet detector test data/obj.data yolo-acv.cfg yolo-acv.weights data/5.jpg -thresh 0.4
cp predictions.jpg ../result_hw/image5_thresh04.jpg
```
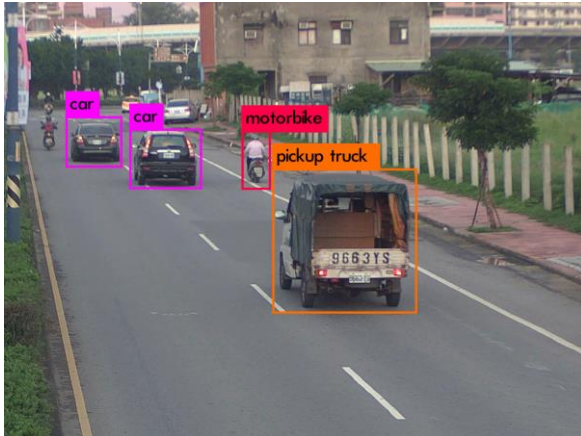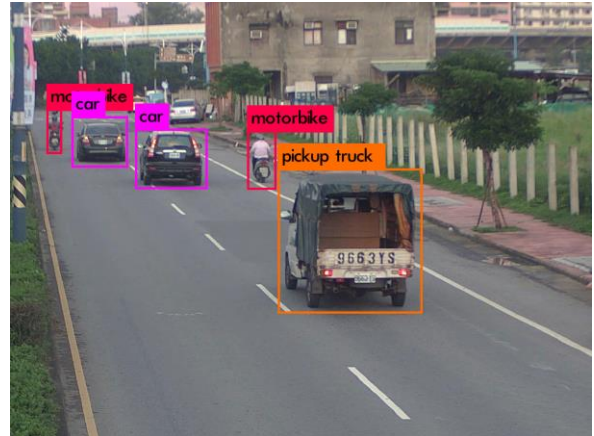
chmod 777 run_hw2.sh

./run_hw2.sh

4. Result

Threshold = 0.4   Threshold = 0.02
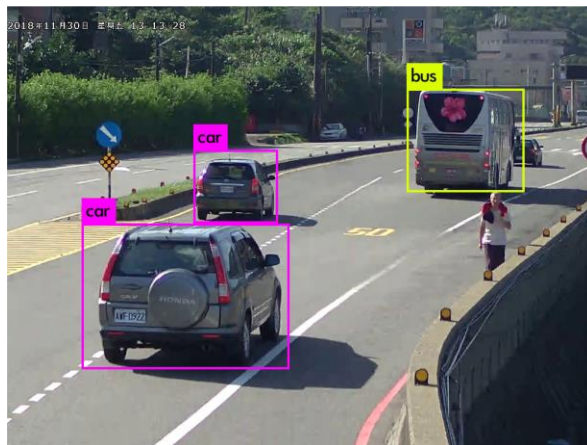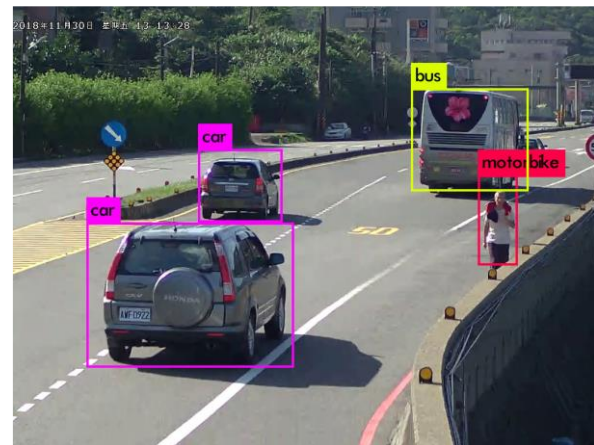

1.jpg

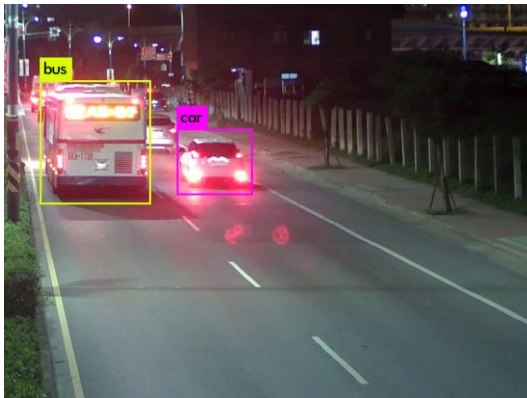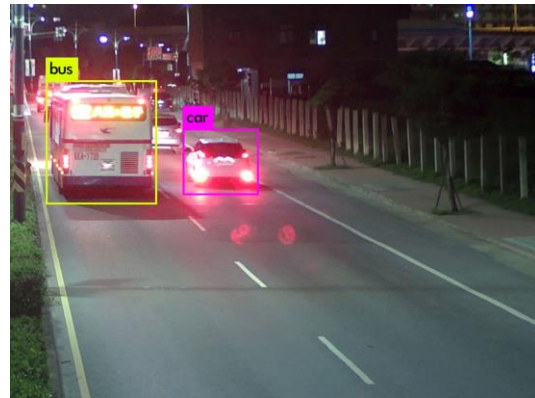Threshold = 0.4   Threshold = 0.02

2.jpg

Threshold = 0.4   **Threshold =0.02**

3.jpg

Threshold = 0.4   **Threshold =0.02**
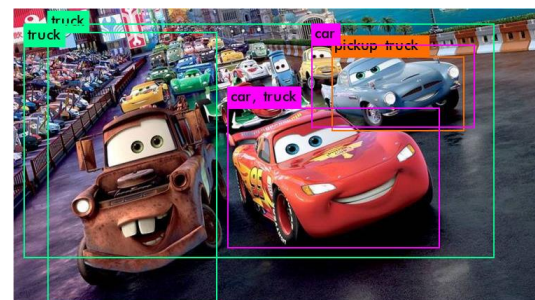
4.jpg

Threshold = 0.4   **Threshold = 0.02**

5.jpg

5. Comparison of the threshold 0.02 and 0.4

a. **Threshold value:** For each predicted box, the class confidence score indicate how many percentage of the predicted box is an object from a certain class. Threshold value determines what is minimum value of the class confidence score can be accepted.

b. **Metrics:** In order to determine the quality of the model and compare the results, I use these below metrics:

> ➢ **True Positive (TP):** object correctly detected
> ➢ **False Positive (FP):** object detected but misclassified
> ➢ **False Negative (FN):** object could not detected

c. **Compare result**

	Threshold = 0.4	Threshold = 0.02
True Positive (TP)	**Low**	**High** One more motorbike in 1.jpg and more car in 5.jpg can detected correctly.
False Positive (FP)	**Low**	**High** Misclassify in 2.jpg and 5.jpg
False Negative (FN)	**High** In 1.jpg, a small motorbike could not detected In 2.jpg, one man could not detected In 4.jpg, many cars could not detected.	**Low**

Limitation of yolov3: how close object can be? In 4.jpg, yolov3 can not detect all of the car because cars are very close to each other.