DATS6203 Individual report
Hao Heng

Introduction

Our project is an object detection using Yolove3 pre-trained model and it is implemented in darknet framework. We will train our own model to predict seven classes: Car, Bicycle wheel, Bus, Traffic light, Jeans, Laptop, Person.

My shared work mainly includes downloading all images from those seven classes, one thousand instances for each category and converting the annotation of these images from Google Open Image dataset format to Yolov3 format, then creating all needed files for training the network.

Individual work

Since Yolov3 network and darknet is brand new to me, I have to start to learn from the beginning. I browsed some Github code and some tutorial from Youtube at the beginning.

After I have a basic understanding of Yolov3 network architecture, I start to prepare the dataset for training the model.

Darknet framework has a very specific requirement of input files, including file names, file contents, the format of the contents.

First, I use the OIDv4_ToolKit to download the images directly to our cloud instance. After installed the toolkit, go to the directory and run the command in terminal. The command line is shown as below:

```
python3 main.py downloader
--classes Car Bicycle_wheel Bus Traffic_light Jeans Laptop Person
--type_csv train
-- multiclasses 1
--limit 1000
```

All this command should be in one line during typing. The “classes” should include all categories names that you want to train on. If there are two words in this category such as “Bicycle wheel”, you need to add an underline between “Bicycle” and “wheel”. The “type_csv” specify the usage of these images. The “limit” implies the number of instances of each category. So for our project ,there are total 7000 images in our dataset, 1000 for each class.

Once the command is executed successfully, it will generate the folder which has two sub directories. One of these two contains all the images, the other contains two csv files that store encrypted string for all class names in Google Open Image dataset and coordinate of the ground truth bounding box of the object in the image. The two csv files are called “class-descriptions-boxable.csv” and “train-annotations-bbox.csv” accordingly.

What we need to now is get encrypted string of each class name since the Google Open Image dataset uses an encrypted string to store all class names in “class-descriptions-boxable.csv”. The encrypted string should look like this: “/m/0k4j ”. This is the encrypted string for class “Car”.

Then we need to locate all the images and get there coordinate of ground truth box of the object in each images. Because the coordinate in Google Open Image is recorded like “XMax, XMin, YMax, YMin”, I need to convert it to “Center x, Center y, width, height” for darknet framework.

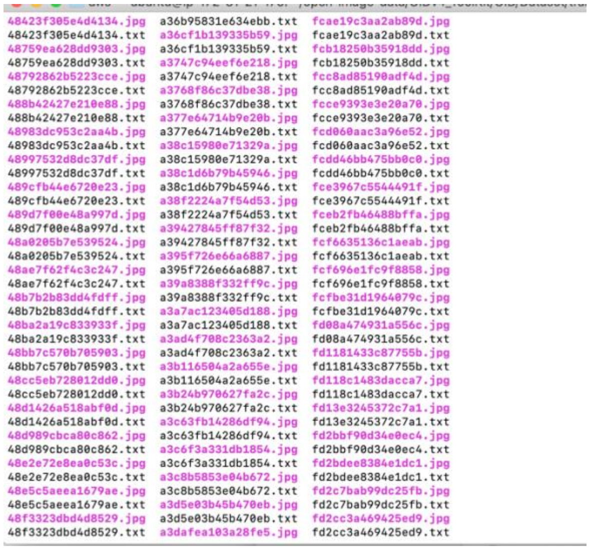
The calculation is shown as below:

Center x= (XMax +XMin)/2

Center y= (YMax +YMin)/2

width= XMax -XMin
height= YMax -YMin

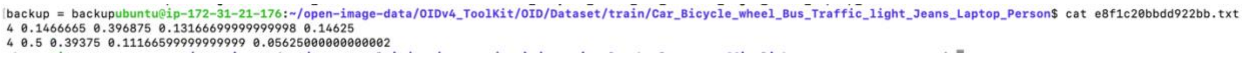
So generally, the algorithm is change the current directory to the folder of these two csv files that I mentioned earlier and read the encrypted string of each class in our custom dataset from “class-descriptions-boxable.csv” and from “train-annotations-bbox.csv”, loading needed columns for calculating the coordinate of Yolov3 after we filter out unrelated row according to our encrypted strings. Then we write the information of each row from these columns to a .txt file for all images. The file name of .txt file should be exact the same with the image file except the extension file name. These files are stored with images in the same directory.



```
48423f305e4d4134.jpg  a36b95831e634ebb.txt  fcae19c3aa2ab89d.jpg
48423f305e4d4134.txt  a36cf1b139335b59.jpg  fcae19c3aa2ab89d.txt
48759ea628dd9303.jpg  a36cf1b139335b59.txt  fcb18250b35918dd.jpg
48759ea628dd9303.txt  a3747c94eef6e218.jpg  fcb18250b35918dd.txt
48792862b5223cce.jpg  a3747c94eef6e218.txt  fcc8ad85190adf4d.jpg
48792862b5223cce.txt  a3768f86c37dbe38.jpg  fcc8ad85190adf4d.txt
488b42427e210e88.jpg  a3768f86c37dbe38.txt  fcce9393e3e20a70.jpg
488b42427e210e88.txt  a377e64714b9e20b.jpg  fcce9393e3e20a70.txt
48983dc953c2aa4b.jpg  a377e64714b9e20b.txt  fcd060aac3a96e52.jpg
48983dc953c2aa4b.txt  a38c15980e71329a.jpg  fcd060aac3a96e52.txt
48997532d8dc37df.jpg  a38c15980e71329a.txt  fcdd46bb475bb0c0.jpg
48997532d8dc37df.txt  a38c1d6b79b45946.jpg  fcdd46bb475bb0c0.txt
489cfb44e6720e23.jpg  a38c1d6b79b45946.txt  fce3967c5544491f.jpg
489cfb44e6720e23.txt  a38f2224a7f54d53.jpg  fce3967c5544491f.txt
489d7f00e48a997d.jpg  a38f2224a7f54d53.txt  fceb2fb46488bffa.jpg
489d7f00e48a997d.txt  a39427845ff87f32.jpg  fceb2fb46488bffa.txt
48a0205b7e539524.jpg  a39427845ff87f32.txt  fcf6635136c1aeab.jpg
48a0205b7e539524.txt  a395f726e66a6887.jpg  fcf6635136c1aeab.txt
48ae7f62f4c3c247.jpg  a395f726e66a6887.txt  fcf696e1fc9f8858.jpg
48ae7f62f4c3c247.txt  a39a8388f332ff9c.jpg  fcf696e1fc9f8858.txt
48b7b2b83dd4fdff.jpg  a39a8388f332ff9c.txt  fcfbe31d1964079c.jpg
48b7b2b83dd4fdff.txt  a3a7ac123405d188.jpg  fcfbe31d1964079c.txt
48ba2a19c833933f.jpg  a3a7ac123405d188.txt  fd08a474931a556c.jpg
48ba2a19c833933f.txt  a3ad4f708c2363a2.jpg  fd08a474931a556c.txt
48bb7c570b705903.jpg  a3ad4f708c2363a2.txt  fd1181433c87755b.jpg
48bb7c570b705903.txt  a3b116504a2a655e.jpg  fd1181433c87755b.txt
48cc5eb728012dd0.jpg  a3b116504a2a655e.txt  fd118c1483dacca7.jpg
48cc5eb728012dd0.txt  a3b24b970627fa2c.jpg  fd118c1483dacca7.txt
48d1426a518abf0d.jpg  a3b24b970627fa2c.txt  fd13e3245372c7a1.jpg
48d1426a518abf0d.txt  a3c63fb14286df94.jpg  fd13e3245372c7a1.txt
48d989cbca80c862.jpg  a3c63fb14286df94.txt  fd2bbf90d34e0ec4.jpg
48d989cbca80c862.txt  a3c6f3a331db1854.jpg  fd2bbf90d34e0ec4.txt
48e2e72e8ea0c53c.jpg  a3c6f3a331db1854.txt  fd2bdee8384e1dc1.jpg
48e2e72e8ea0c53c.txt  a3c8b5853e04b672.jpg  fd2bdee8384e1dc1.txt
48e5c5aeea1679ae.jpg  a3c8b5853e04b672.txt  fd2c7bab99dc25fb.jpg
48e5c5aeea1679ae.txt  a3d5e03b45b470eb.jpg  fd2c7bab99dc25fb.txt
48f3323dbd4d8529.jpg  a3d5e03b45b470eb.txt  fd2cc3a469425ed9.jpg
48f3323dbd4d8529.txt  a3dafea103a28fe5.jpg  fd2cc3a469425ed9.txt
```

Figure1. images and annotations pair

This figure shows the result of this step, as you can see, every image is followed by an annotation file that has the same file names.

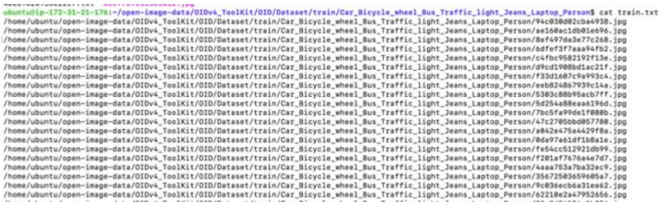


```
backup = backupubuntu@ip-172-31-21-176:~/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person$ cat e8f1c20bbdd922bb.txt
4 0.1466665 0.396875 0.13166699999999998 0.14625
4 0.5 0.39375 0.11166599999999999 0.05625000000000002
```

Figure2. Annotation content

This figure shows that each annotation file contains five columns corresponding to class name, center x, center y, width, height. One line means one ground truth bounding box. Multiple lines means there is more than one object in this image.

The next step is going to image folder, read all path of image and store them in a list. Then split the list into training set and test set and write two part into train.txt and test.txt. In our project we use 85% of the dataset for training.



```
ubuntu@ip-172-31-21-176:~/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person$ cat train.txt
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/94c030d02cba4938.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/ae160ac1db01e696.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/8ef497da3e77c268.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/6dfef3f7aaa94fb2.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/c4fbc9582192f13e.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/d9cd1908bd1ac21f.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/f33d1607c9a993c4.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/eeb824867939c14a.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/5303c88b95acb7ff.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/5d254a88eaa6196d.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/7bc5fa99de1f080b.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/47c2705bbd057780.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/a042e475a4429f8a.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/0da97a61df1b8a1e.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/fe54cc512921db99.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/f201af7676a4e7d7.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/4aaa753a7ba32ec9.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/35672503659605a7.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/9c036ecb6a31ee62.jpg
/home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/62210e2a47952656.jpg
```

Figure3. train.txt and test.txt

This figure shows the content of train.txt and test.txt

The final step of creating our custom dataset is generating .data file and .names file. The contents of .data file and .names file are shown as below:

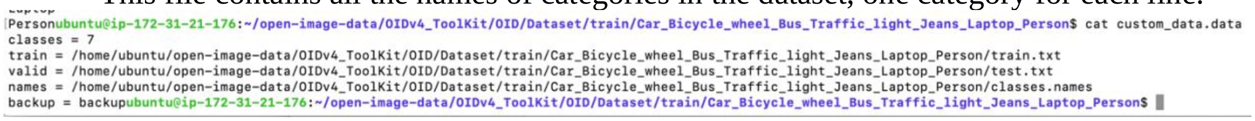
data.data file

This file only contains five fixed lines. It should be looked like this:

classes=the number of classes in the dataset
train=full path to train.txt file
valid=full path to test.txt file
names=full path to classes.names file
backup=backup

classes.names file

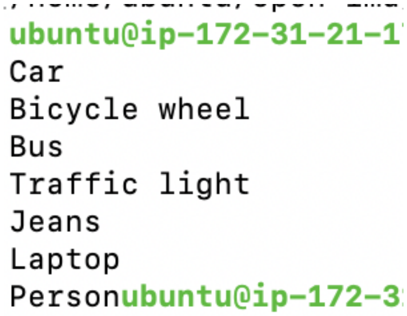
This file contains all the names of categories in the dataset, one category for each line.



```
Personubuntu@ip-172-31-21-176:~/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person$ cat custom_data.data
classes = 7
train = /home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/train.txt
valid = /home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/test.txt
names = /home/ubuntu/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person/classes.names
backup = backupubuntu@ip-172-31-21-176:~/open-image-data/OIDv4_ToolKit/OID/Dataset/train/Car_Bicycle_wheel_Bus_Traffic_light_Jeans_Laptop_Person$
```

Figure4. content of .data file

This figure shows the content of .data file.



```
ubuntu@ip-172-31-21-1
Car
Bicycle wheel
Bus
Traffic light
Jeans
Laptop
Personubuntu@ip-172-3
```

Figure5. content of .names file

This figure shows the content of .names file.

Conclusion

By far, I have created .data file, .names file, train.txt, test.txt and covert the coordinate of ground truth bounding box all annotation of each images. This is all the process required for creating a custom dataset to train on.

Percentage of code from Internet

40%

References

[1]Joseph Redmon, Santosh Divvala, Ross Girshick, ALi Farhadi “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2]Open Images Dataset V6 + Extensions. (n.d.). Retrieved December 08, 2020, from https://storage.googleapis.com/openimages/web/index.html

[3]Joseph R. (2013-2016). Darknet: Open Source Neural Networks in C. Retrieved from http://pjreddie.com/darknet/