

Project 2 - PII Detection Using YOLO

Nandini Krupa Krishnamurthy

1. Installed darknet, downloaded the pre-trained weight file and finished setting up yolo.

```
krupa@LAPTOP-7EFF46BN:~/photoprivacy/darknet$ wget https://pjreddie.com/media/files/yolov3.weights
Will not apply HSTS. The HSTS database must be a regular and non-world-writable file.
ERROR: could not open HSTS store at '/home/krupa/.wget-hsts'. HSTS will be disabled.
--2019-10-26 14:45:20-- https://pjreddie.com/media/files/yolov3.weights
Resolving pjreddie.com (pjreddie.com)... 128.208.4.108
Connecting to pjreddie.com (pjreddie.com)[128.208.4.108]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 248007048 (237M) [application/octet-stream]
Saving to: 'yolov3.weights.1'

yolov3.weights.1      100%[=====] 236.52M  803KB/s   in 4m 47s
2019-10-26 14:50:08 (844 KB/s) - 'yolov3.weights.1' saved [248007048/248007048]
```

2. Testing Yolo

```
71 res 68
72 conv 512 1 x 1 / 1 19 x 19 x1024 -> 19 x 19 x1024
73 conv 1024 3 x 3 / 1 19 x 19 x1024 -> 19 x 19 x 512 0.379 BFLOPs
74 res 71 19 x 19 x1024 -> 19 x 19 x1024
75 conv 512 1 x 1 / 1 19 x 19 x1024 -> 19 x 19 x 512 0.379 BFLOPs
76 conv 1024 3 x 3 / 1 19 x 19 x 512 -> 19 x 19 x1024 3.407 BFLOPs
77 conv 512 1 x 1 / 1 19 x 19 x1024 -> 19 x 19 x 512 0.379 BFLOPs
78 conv 1024 3 x 3 / 1 19 x 19 x 512 -> 19 x 19 x1024 3.407 BFLOPs
79 conv 512 1 x 1 / 1 19 x 19 x1024 -> 19 x 19 x 512 0.379 BFLOPs
80 conv 1024 3 x 3 / 1 19 x 19 x 512 -> 19 x 19 x1024 3.407 BFLOPs
81 conv 255 1 x 1 / 1 19 x 19 x1024 -> 19 x 19 x 255 0.189 BFLOPs
82 yolo
83 route 79
84 conv 256 1 x 1 / 1 19 x 19 x 512 -> 19 x 19 x 256 0.095 BFLOPs
85 upsample 2x 19 x 19 x 256 -> 38 x 38 x 256
86 route 85 61
87 conv 256 1 x 1 / 1 38 x 38 x 768 -> 38 x 38 x 256 0.568 BFLOPs
88 conv 512 3 x 3 / 1 38 x 38 x 256 -> 38 x 38 x 512 3.407 BFLOPs
89 conv 256 1 x 1 / 1 38 x 38 x 512 -> 38 x 38 x 256 0.379 BFLOPs
90 conv 512 3 x 3 / 1 38 x 38 x 256 -> 38 x 38 x 512 3.407 BFLOPs
91 conv 256 1 x 1 / 1 38 x 38 x 512 -> 38 x 38 x 256 0.379 BFLOPs
92 conv 512 3 x 3 / 1 38 x 38 x 256 -> 38 x 38 x 512 3.407 BFLOPs
93 conv 255 1 x 1 / 1 38 x 38 x 512 -> 38 x 38 x 255 0.377 BFLOPs
94 yolo
95 route 91
96 conv 128 1 x 1 / 1 38 x 38 x 256 -> 38 x 38 x 128 0.095 BFLOPs
97 upsample 2x 38 x 38 x 128 -> 76 x 76 x 128
98 route 97 36
99 conv 128 1 x 1 / 1 76 x 76 x 384 -> 76 x 76 x 128 0.568 BFLOPs
100 conv 256 3 x 3 / 1 76 x 76 x 128 -> 76 x 76 x 256 3.407 BFLOPs
101 conv 128 1 x 1 / 1 76 x 76 x 256 -> 76 x 76 x 128 0.379 BFLOPs
102 conv 256 3 x 3 / 1 76 x 76 x 128 -> 76 x 76 x 256 3.407 BFLOPs
103 conv 128 1 x 1 / 1 76 x 76 x 256 -> 76 x 76 x 128 0.379 BFLOPs
104 conv 256 3 x 3 / 1 76 x 76 x 128 -> 76 x 76 x 256 3.407 BFLOPs
105 conv 255 1 x 1 / 1 76 x 76 x 256 -> 76 x 76 x 255 0.754 BFLOPs
106 yolo
Loading weights from yolov3.weights...Done!
data/dog.jpg: Predicted in 46.715687 seconds.
dog: 100%
truck: 92%
bicycle: 99%
krupa@LAPTOP-7EFF46BN:~/photoprivacy/darknet$
```

3. Downloaded and extracted dataset. The dataset has 109 images in “private” folder and 100 images in “public” folder.

```
krupa@LAPTOP-7EFF46BN:~/photoprivacy/darknet$ unzip dataset.zip
Archive:  dataset.zip
  creating: dataset/private/
  inflating: dataset/private/4330400088.jpg
  inflating: dataset/private/4331808437.jpg
  inflating: dataset/private/4333240960.jpg
  inflating: dataset/private/4333681728.jpg
  inflating: dataset/private/4333696292.jpg
  inflating: dataset/private/4336675220.jpg
  extracting: dataset/private/4337733437.jpg
  inflating: dataset/private/4338158943.jpg
  inflating: dataset/private/4338398780.jpg
  extracting: dataset/private/4338422575.jpg
  inflating: dataset/private/4338457809.jpg
  inflating: dataset/private/4338666198.jpg
  extracting: dataset/private/4339315528.jpg
  extracting: dataset/private/4339819203.jpg
  extracting: dataset/private/4340780316.jpg
  inflating: dataset/private/4342990075.jpg
  inflating: dataset/private/4343147011.jpg
  inflating: dataset/private/4343170577.jpg
  extracting: dataset/private/4344397460.jpg
  inflating: dataset/private/4344531492.jpg
  inflating: dataset/private/4344715961.jpg
  inflating: dataset/private/4345628858.jpg
  inflating: dataset/private/4347435013.jpg
  inflating: dataset/private/4347738656.jpg
  inflating: dataset/private/4348957203.jpg
  inflating: dataset/private/4349297203.jpg
  inflating: dataset/private/4350442802.jpg
  extracting: dataset/private/4350870822.jpg
  inflating: dataset/private/4351924993.jpg
  inflating: dataset/private/4352674310.jpg
  inflating: dataset/private/4353117303.jpg
  inflating: dataset/private/4353135983.jpg
  inflating: dataset/private/4353651367.jpg
  inflating: dataset/private/4353808437.jpg
  extracting: dataset/private/4353816194.jpg
  inflating: dataset/private/4353934580.jpg
  inflating: dataset/private/4354000011.jpg
  extracting: dataset/private/4354219152.jpg
  inflating: dataset/private/4355569364.jpg
```

4. Results of running detectors – “public.py” and “private.py” on public and private datasets are stored in “public.txt” and “private.txt” respectively.

krupa@LAPTOP-7EFF46BN: ~/photoprivacy/darknet

```
diningtable: 60%
hot dog: 57%

chair: 83%
person: 99%
person: 91%
person: 63%

person: 100%
person: 55%

person: 99%
person: 99%
person: 97%
person: 95%
person: 86%

chair: 72%
chair: 68%
chair: 62%
cup: 92%
bottle: 70%
person: 99%
person: 97%
person: 95%
person: 84%
person: 81%
person: 79%
person: 74%
person: 70%
person: 59%
person: 59%

person: 100%
person: 99%
person: 99%
person: 99%
person: 98%
person: 96%
person: 92%
"public.txt" 293L, 2401C
```

krupa@LAPTOP-7EFF46BN: ~/photoprivacy/darknet

```
person: 100%
person: 100%
person: 100%
person: 98%

person: 100%

person: 96%

person: 73%

person: 100%

dog: 75%
person: 92%

person: 100%
person: 99%

cup: 86%
cup: 53%
person: 100%
person: 55%
person: 55%
person: 53%

person: 100%

person: 100%

cup: 89%
cup: 62%
person: 100%

book: 65%
microwave: 52%
cell phone: 65%

person: 100%
"private.txt" 381L, 3411C
```

5. Results of running “results.py” to obtain top 10 images in both public and private category

```
krupa@LAPTOP-7EFF46BN:~/photoprivacy/darknet$ python3 results.py

private.txt
*****
Num.    Occurances    Object
1.      156           person
2.      26            cup
3.       8           dog
4.       8         wine glass
5.       6           chair
6.       5         cell phone
7.       5         tvmonitor
8.       5         diningtable
9.       5           bottle
10.     4            book

public.txt
*****
Num.    Occurances    Object
1.     124           person
2.     15           bottle
3.       6           chair
4.       6           knife
5.       4            car
6.       4         giraffe
7.       4            book
8.       3           train
9.       2         diningtable
10.     2            cup

krupa@LAPTOP-7EFF46BN:~/photoprivacy/darknet$
```

Observations

1. 'Person' is the top result for both private and public images dataset. This means, people tend to keep their photos as both public and private.
2. Some objects such as “book”, “cup” and “chair” fall under both the categories indicating that the environment in which these object are placed decides the category they have to fall under.
3. Usually those objects which are inaccessible by all and are confined to one place fall under private category and those which can be accessed by general public fall under public category.
4. Personal belongings – example, “wine glass”, “cell phone” and “tvmonitor” fall under private.

Appendix

CODES:

private.py

```
import os
from subprocess import Popen, PIPE
file1 = open("private.txt", "a+", encoding = "utf-8")

images_dir = '/home/krupa/photoprivacy/darknet/dataset'
setting = 'private'

for img in os.listdir(os.getcwd()+"/dataset/private"):
    count += 1

    p = Popen(['./darknet', 'detect', 'cfg/yolov3.cfg', 'yolov3.weights', images_dir + '/' + setting + '/' +
str(img)], cwd = '/home/krupa/photoprivacy/darknet', stdout = PIPE, stderr = PIPE)
    stdout, stderr = p.communicate()

    output = stdout.decode("utf-8")
    output_split = output.split("\n")

    for i in output_split:
        if "Predicted in" not in i:
            print(i)
            file1.write(i)
            file1.write("\n")
file1.close()
```

public.py

```
import os
from subprocess import Popen, PIPE
file1 = open("public.txt", "a+", encoding = "utf-8")

images_dir = '/home/krupa/photoprivacy/darknet/dataset'
setting = 'public'

for img in os.listdir(os.getcwd()+"/dataset/public"):
    count += 1

    p = Popen(['./darknet', 'detect', 'cfg/yolov3.cfg', 'yolov3.weights', images_dir + '/' + setting + '/' +
str(img)], cwd = '/home/krupa/photoprivacy/darknet', stdout = PIPE, stderr = PIPE)
    stdout, stderr = p.communicate()

    output = stdout.decode("utf-8")
    output_split = output.split("\n")

    for i in output_split:
        if "Predicted in" not in i:
            print(i)
            file1.write(i)
            file1.write("\n")
file1.close()
```

results.py

```
import string
import os
file_list = ['private.txt','public.txt']

for file_name in file_list:
    d = dict()
    contents = []
    print("\n")
    print(file_name)
    print("*****")
    file_name = open (file_name , "r", encoding = 'utf-8')
    lines_private=[line.rstrip("\n") for line in (file_name)]
    lines_private = list(filter(None, lines_private))

    for line in lines_private:
        index = line.find(":")
        value = line[0:index]
        contents.append(value)

    for line in contents:
        if line in d:
            d[line] = d[line] + 1
        else:
            d[line] = 1

    sorted_d = sorted(d, key=d.get, reverse=True)
    #print(sorted_d)

    print("{}.\t{}\t{}".format("Num", "Occurances", "Object"))
    for i in range(10):
        #print(i+1,sorted_d[i], d[sorted_d[i]])
        print("{}.\t{}\t{}".format(i+1,d[sorted_d[i]], sorted_d[i]))

    print("\n")
    file_name.close()
```