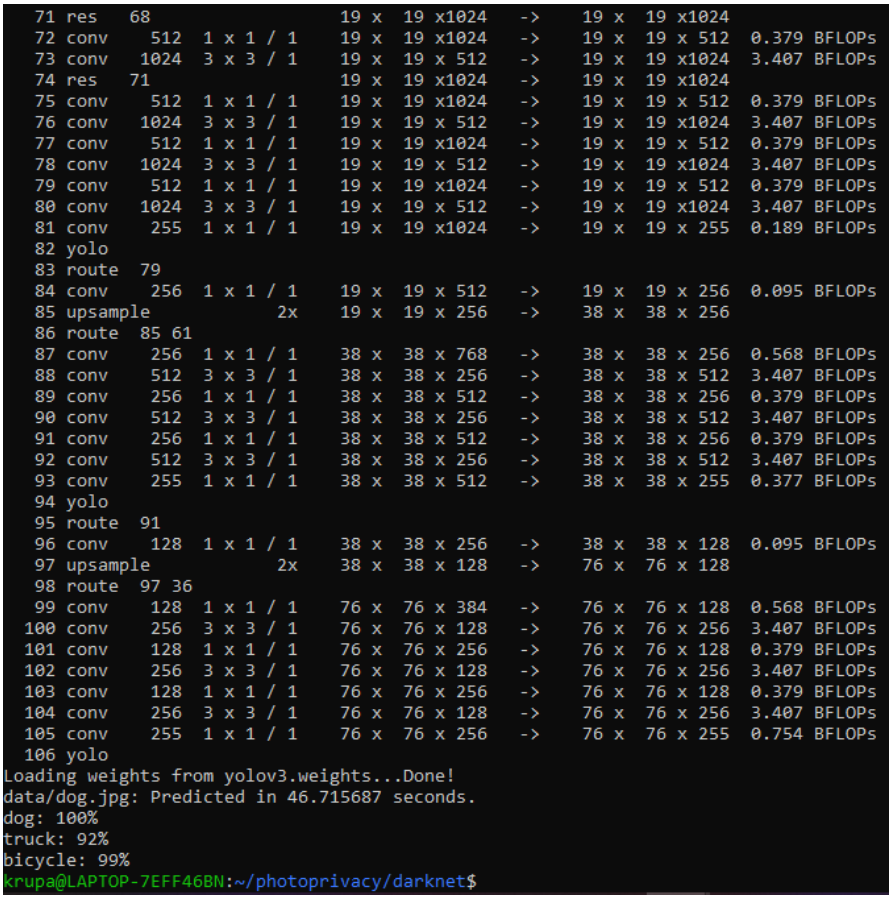
# Project 2 - PII Detection Using YOLO

Nandini Krupa Krishnamurthy

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

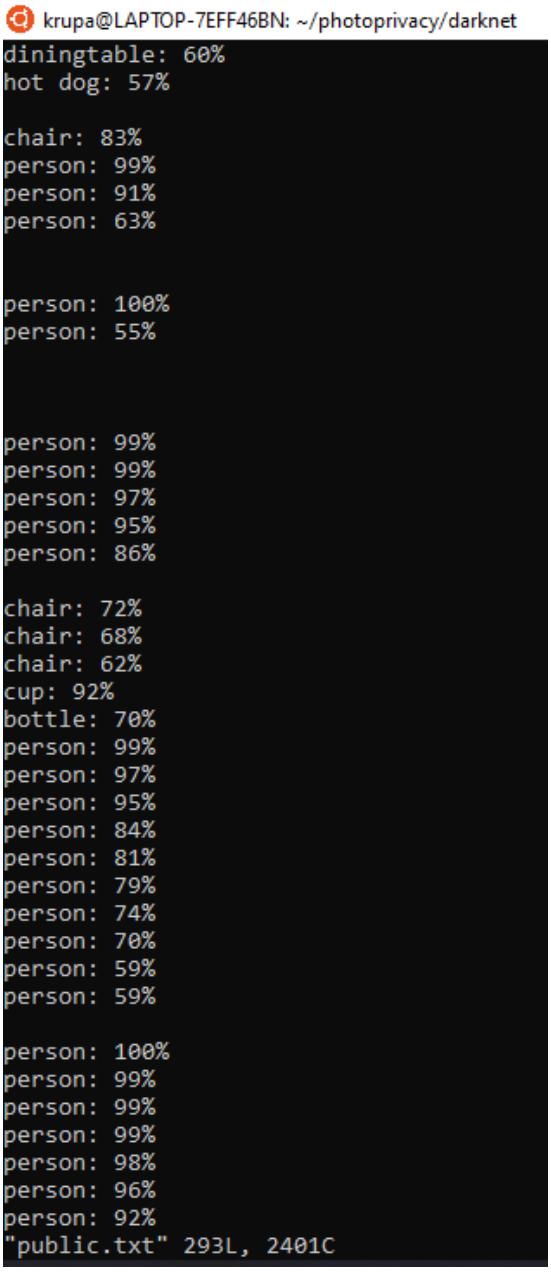

2. Testing Yolo

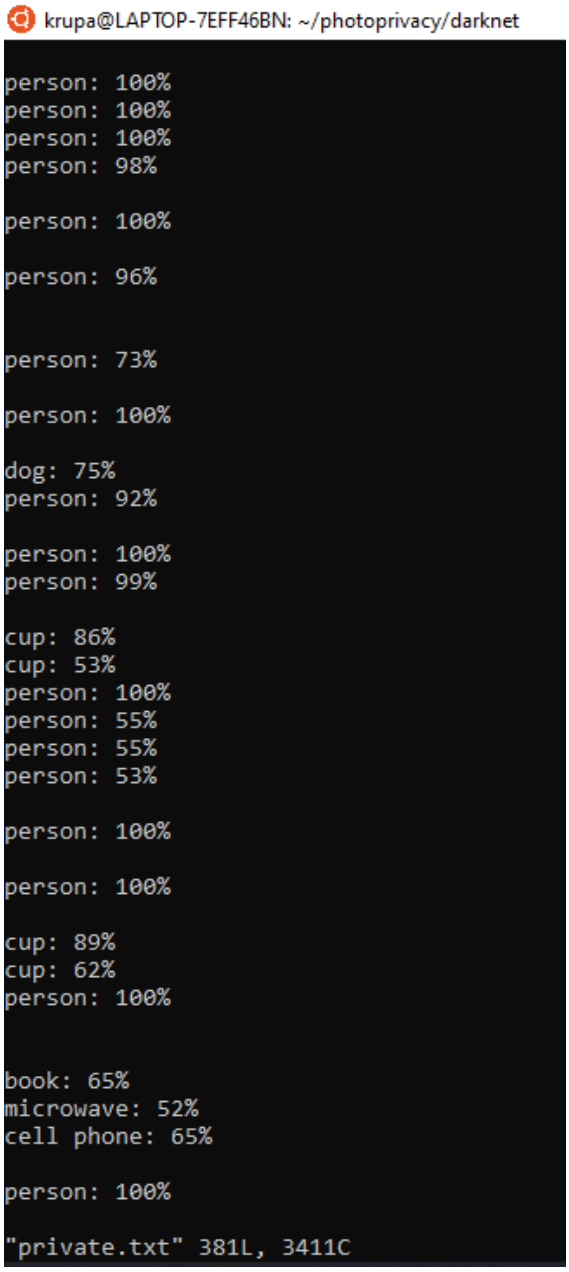


3. Downloaded and extracted dataset. The dataset has 109 images in “private” folder and 100 images in “public” folder.

```
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.



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 =0

   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 =0

   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\t{}".format(i+1,d[sorted_d[i]], sorted_d[i]))

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