

Our Machine Learning algorithm is able to receive Google Street images, and return the processed results, from a user with a TCP connection through Socket Programming in the Python language.

The Client side of this TCP connection would be the user/KAB Web App team, and the Server side would be the Machine Learning algorithm itself.

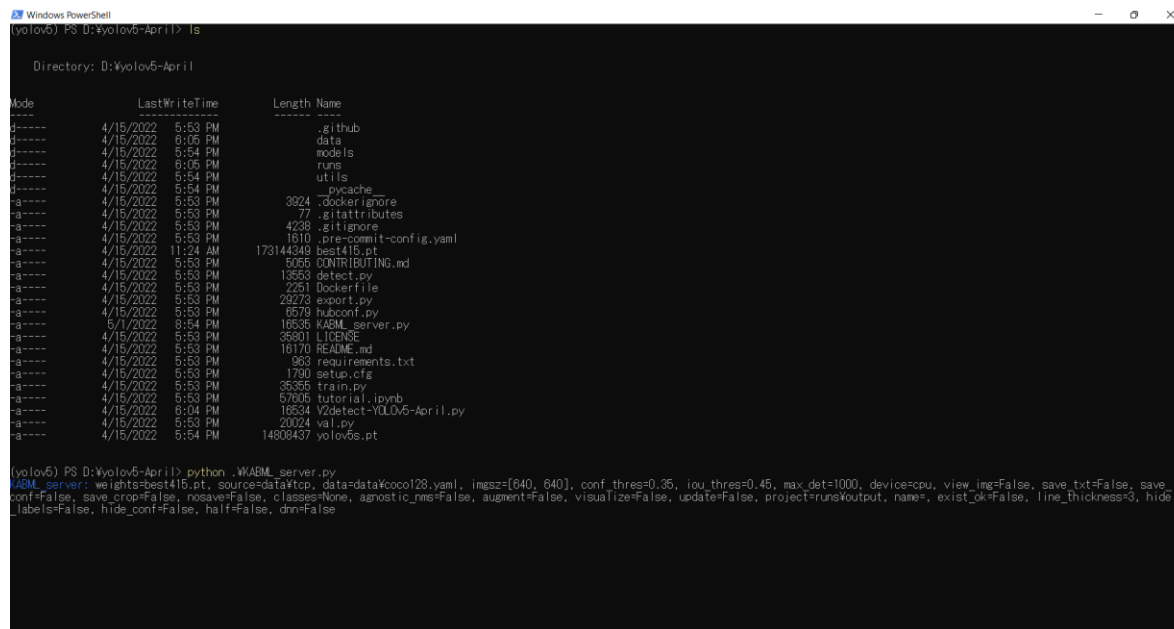
---

*(Be sure that the IP address used for the TCP connection is correct/references the same address on both sides.)*

To start off, you need to have the Server side running first. To do so, run this command in your Command Prompt/Terminal, while in the directory of the Server code:

- **python KABML\_server.py**

You should see this:



```
Windows PowerShell
(yolov5) PS D:\yolov5-April> ls

Directory: D:\yolov5-April

Mode                LastWriteTime         Length Name
----                -
d----- 4/15/2022 5:53 PM                .github
d----- 4/15/2022 6:06 PM                data
d----- 4/15/2022 5:54 PM                models
d----- 4/15/2022 6:06 PM                runs
d----- 4/15/2022 5:54 PM                utils
d----- 4/15/2022 5:54 PM                pycache
-a----- 4/15/2022 5:53 PM             3924 .dockerignore
-a----- 4/15/2022 5:53 PM              77 .gitattributes
-a----- 4/15/2022 5:53 PM             4238 .gitignore
-a----- 4/15/2022 5:53 PM             1610 .pre-commit-config.yaml
-a----- 4/15/2022 11:24 AM      173144349 best41b.pt
-a----- 4/15/2022 5:53 PM             9356 CONTRIBUTING.md
-a----- 4/15/2022 5:53 PM            13563 detect.py
-a----- 4/15/2022 5:53 PM             2251 Dockerfile
-a----- 4/15/2022 5:53 PM            29273 export.py
-a----- 4/15/2022 5:53 PM             6979 hubconf.py
-a----- 5/1/2022 8:14 PM            16936 KABML_server.py
-a----- 4/15/2022 5:53 PM            35801 LICENSE
-a----- 4/15/2022 5:53 PM            16170 README.md
-a----- 4/15/2022 5:53 PM             903 requirements.txt
-a----- 4/15/2022 5:53 PM             1790 setup.cfg
-a----- 4/15/2022 5:53 PM            35355 train.py
-a----- 4/15/2022 5:53 PM            57605 tutorial.ipynb
-a----- 4/15/2022 6:04 PM            16534 Y2detect-YOLOv5-April.py
-a----- 4/15/2022 5:53 PM             20024 val.py
-a----- 4/15/2022 5:54 PM            14808437 yolov5.pt

(yolov5) PS D:\yolov5-April> python KABML_server.py
KABML_server: weights=best41b.pt, source=data\top, data=data\coco128.yaml, imgsz=(640, 640), conf_thres=0.35, iou_thres=0.45, max_det=1000, device=cpu, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=runs\output, name=, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False
```

At this point, the Server side is up and running, and waiting for the Client to connect and send Google Street images.

Now you can run the Client side. To do so, run this command in your Command Prompt/Terminal, while in the directory of the Client code:

- **python KABML\_client.py *folder\_name***

The ***folder\_name*** should be the name of the folder that holds your Google Street images.

Once you run the Client command, you should see this on the Client side:

```
Windows PowerShell
(base) PS C:\Users\Ferninator\Desktop\VCIS 490\MULTI-IMAGE TESTING TOP\VS code - YOLOv5> ls

Directory: C:\Users\Ferninator\Desktop\VCIS 490\MULTI-IMAGE TESTING TOP\VS code - YOLOv5

Mode                LastWriteTime         Length Name
----                -
4/15/2022 11:29 AM                0 GS_sample
4/20/2022 2:54 PM                0 GS_sample_Output
4/15/2022 11:29 AM                0 Keith_Q3
4/15/2022 12:14 PM                0 Keith_Q3_1206_Output
4/15/2022 12:17 PM                0 Laras_GS_sample
5/1/2022 8:07 PM                0 ML-output
3/20/2022 9:36 PM                0 sample
3/20/2022 6:47 PM                0 sample2
3/20/2022 8:21 PM                0 WABM_Client.py
3/20/2022 8:17 PM                0 16388_Y2detect-YOLOv5.py

(base) PS C:\Users\Ferninator\Desktop\VCIS 490\MULTI-IMAGE TESTING TOP\VS code - YOLOv5> python .\WABM_Client.py .\GS_sample\
VS_sample\
ending folder: .\GS_sample\
ending file: cazarez_v124.jpg (22501 bytes)
ending file: Dominguez_v145.jpg (37888 bytes)
ending file: Dominguez_v235.jpg (21609 bytes)
ending file: Dominguez_v277.jpg (32759 bytes)
ending file: johnson_v193.jpg (48065 bytes)
ending file: kiefer_v1.jpg (59473 bytes)
ending file: kiefer_v130.jpg (52926 bytes)
ending file: kiefer_v70.jpg (640x640 3 litters, Done. (1.027s))
ending file: leal_v81.jpg (65161 bytes)
ending file: stone_v62.jpg (56395 bytes)
receiving folder: runs/output (11 files)
receiving file: cazarez_v124.jpg (48243 bytes)
receiving file: Dominguez_v145.jpg (78149 bytes)
receiving file: Dominguez_v235.jpg (44128 bytes)
receiving file: Dominguez_v277.jpg (67045 bytes)
receiving file: johnson_v193.jpg (48065 bytes)
receiving file: kiefer_v1.jpg (59473 bytes)
receiving file: kiefer_v130.jpg (52926 bytes)
receiving file: kiefer_v70.jpg (65109 bytes)
receiving file: leal_v81.jpg (65161 bytes)
receiving file: results.json (998 bytes)
receiving file: stone_v62.jpg (56395 bytes)
(base) PS C:\Users\Ferninator\Desktop\VCIS 490\MULTI-IMAGE TESTING TOP\VS code - YOLOv5>
```

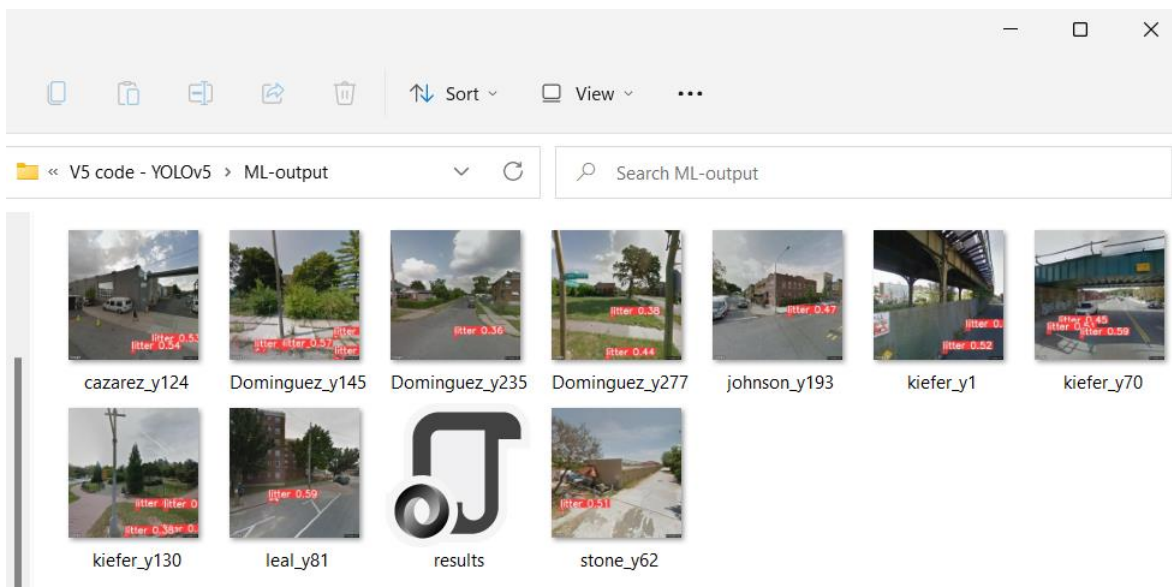
And see this on the Server side:

```
Windows PowerShell
a---- 4/15/2022 5:53 PM 20024 val.py
a---- 4/15/2022 5:54 PM 14808437 yolov5s.pt

(yolov5) PS D:\yolov5-April> python .\WABM_server.py
WABM_server: weights=best41b.pt, source=data\top, data=data\koco128.yaml, imgsz=[640, 640], conf_thres=0.35, iou_thres=0.45, max_det=1000, device=cpu, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_rms=False, augment=False, visualize=False, update=False, project=runs\output, name=, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False
[127.0.0.1]:52644 connected
Receiving folder: .\GS_sample\ (10 files)
Receiving file: cazarez_v124.jpg (22501 bytes)
Receiving file: Dominguez_v145.jpg (37888 bytes)
Receiving file: Dominguez_v235.jpg (21609 bytes)
Receiving file: Dominguez_v277.jpg (32759 bytes)
Receiving file: johnson_v193.jpg (48065 bytes)
Receiving file: kiefer_v1.jpg (59473 bytes)
Receiving file: kiefer_v130.jpg (52926 bytes)
Receiving file: kiefer_v70.jpg (640x640 3 litters, Done. (1.027s))
Receiving file: leal_v81.jpg (65161 bytes)
Receiving file: stone_v62.jpg (56395 bytes)
YOLOv5 v6.1-133-g3eeafab1 torch 1.9.0+cu111 CPU

Fusing layers...
YOLOv5 summary: 444 layers, 86172414 parameters, 0 gradients, 204.0 GFLOPs
image 1/10 D:\yolov5-April\data\top\Dominguez_v145.jpg: 640x640 5 litters, Done. (1.098s)
image 2/10 D:\yolov5-April\data\top\Dominguez_v235.jpg: 640x640 1 litter, Done. (1.119s)
image 3/10 D:\yolov5-April\data\top\Dominguez_v277.jpg: 640x640 2 litters, Done. (1.004s)
image 4/10 D:\yolov5-April\data\top\cazarev_v124.jpg: 640x640 2 litters, Done. (1.066s)
image 5/10 D:\yolov5-April\data\top\johnson_v193.jpg: 640x640 1 litter, Done. (1.118s)
image 6/10 D:\yolov5-April\data\top\kiefer_v1.jpg: 640x640 2 litters, Done. (1.071s)
image 7/10 D:\yolov5-April\data\top\kiefer_v130.jpg: 640x640 4 litters, Done. (1.130s)
image 8/10 D:\yolov5-April\data\top\kiefer_v70.jpg: 640x640 3 litters, Done. (1.027s)
image 9/10 D:\yolov5-April\data\top\leal_v81.jpg: 640x640 1 litter, Done. (1.188s)
image 10/10 D:\yolov5-April\data\top\stone_v62.jpg: 640x640 1 litter, Done. (1.097s)
Speed: 1.7ms pre-process, 1091.2ms inference, 1.9ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/output
ending folder: runs/output
Sending file: cazarez_v124.jpg (48243 bytes)
Sending file: Dominguez_v145.jpg (78149 bytes)
Sending file: Dominguez_v235.jpg (44128 bytes)
Sending file: Dominguez_v277.jpg (67045 bytes)
Sending file: johnson_v193.jpg (48065 bytes)
Sending file: kiefer_v1.jpg (59473 bytes)
Sending file: kiefer_v130.jpg (52926 bytes)
Sending file: kiefer_v70.jpg (65109 bytes)
Sending file: leal_v81.jpg (65161 bytes)
Sending file: results.json (998 bytes)
Sending file: stone_v62.jpg (56395 bytes)
```

As you can see, all the images from the *folder\_name* have been sent over to the Server side. Once the Server side has received all the images, it starts processing through them all, one at a time, to detect any instances of litter in the Google Street images. Once it is done going through all the images, it sends the results back to the Client, in which the Client side will save the results into a folder called **ML-output**.



At this point the Client side has finished its purpose. However, the Server side will continue running until the Server code's process is terminated. (*Pressing Ctrl-C*)

Until the Server code's process is terminated, the Server side will be ready for any new runs on the Client side. (*Repeat the same steps from the Client side.*)

```

Windows PowerShell
PS C:\Users\Terminator\Desktop\VCIS 400\MLTI-IMAGE TESTING TOPV5 code - YOLOv5> python .\WAEML_client.py .\WS_sample\
WS_sample\
Sending folder: .\WS_sample\
Sending file: cazarez_y124.jpg (22501 bytes)
Sending file: Dominguez_y145.jpg (37888 bytes)
Sending file: Dominguez_y235.jpg (21888 bytes)
Sending file: Dominguez_y277.jpg (32755 bytes)
Sending file: johnson_y193.jpg (24149 bytes)
Sending file: kiefer_y1.jpg (28042 bytes)
Sending file: kiefer_y130.jpg (24545 bytes)
Sending file: kiefer_y70.jpg (24248 bytes)
Sending file: leal_y81.jpg (22098 bytes)
Sending file: stone_y62.jpg (28450 bytes)
Sending folder: runs\output (11 files)
Receiving file: cazarez_y124.jpg (45243 bytes)
Receiving file: Dominguez_y145.jpg (78149 bytes)
Receiving file: Dominguez_y235.jpg (44126 bytes)
Receiving file: Dominguez_y277.jpg (67045 bytes)
Receiving file: johnson_y193.jpg (45005 bytes)
Receiving file: kiefer_y1.jpg (59472 bytes)
Receiving file: kiefer_y130.jpg (52929 bytes)
Receiving file: kiefer_y70.jpg (65109 bytes)
Receiving file: leal_y81.jpg (55181 bytes)
Receiving file: results.json (988 bytes)
Receiving file: stone_y62.jpg (55595 bytes)
(base) PS C:\Users\Terminator\Desktop\VCIS 400\MLTI-IMAGE TESTING TOPV5 code - YOLOv5> python .\WAEML_client.py .\Waith_QSY\
Waith_QSY
Sending file: image3.jpg (118915 bytes)
Sending file: image4 - Copy.jpg (51626 bytes)
Sending file: image4.jpg (246349 bytes)
Sending file: image5.jpg (184389 bytes)
Sending file: image6.jpg (134892 bytes)
Sending file: image8.jpg (234537 bytes)
Sending file: image9.jpg (341413 bytes)
Sending file: image10.jpg (378837 bytes)
Sending file: image11.jpg (350815 bytes)
Sending file: image12.jpg (335842 bytes)
Sending file: image13.jpg (349492 bytes)
Sending folder: runs\output (12 files)
Receiving file: image3.jpg (124758 bytes)
Receiving file: image4 - Copy.jpg (62287 bytes)
Receiving file: image4.jpg (451250 bytes)
Receiving file: image5.jpg (340093 bytes)
Receiving file: image6.jpg (251023 bytes)
Receiving file: image7.jpg (437000 bytes)
Receiving file: image8.jpg (606039 bytes)
Receiving file: image9.jpg (600026 bytes)
Receiving file: image10.jpg (624543 bytes)
Receiving file: image11.jpg (713452 bytes)
Receiving file: image12.jpg (680634 bytes)
Receiving file: results.json (1029 bytes)
(base) PS C:\Users\Terminator\Desktop\VCIS 400\MLTI-IMAGE TESTING TOPV5 code - YOLOv5>

```

```
Windows PowerShell
Sending file: kiefer_y10.jpg (52426 bytes)
Sending file: kiefer_y70.jpg (55109 bytes)
Sending file: leal_y81.jpg (55161 bytes)
Sending file: results.json (998 bytes)
Sending file: store_y62.jpg (58390 bytes)
(1127.0.0.1 - 52647) connected
Receiving folder: .\Keith.GS% (11 files)
Receiving file: Image3.jpg (118915 bytes)
Receiving file: Image4 - Copy.jpg (51826 bytes)
Receiving file: Image4.jpg (245340 bytes)
Receiving file: Image6.jpg (184389 bytes)
Receiving file: Image7.jpg (134482 bytes)
Receiving file: Image8.jpg (234537 bytes)
Receiving file: img10.jpg (341413 bytes)
Receiving file: img15.jpg (376837 bytes)
Receiving file: img6.jpg (350815 bytes)
Receiving file: img8.jpg (393542 bytes)
Receiving file: img9.jpg (349492 bytes)
YOLOv5 v6.1-133-g3ee1ab1 torch 1.9.0+cu111 CPU

Fusing layers... 444 layers, 96173414 parameters, 0 gradients, 204.0 GFLOPs
image 1/11 D:\yolov5-April\data\tcp\Image3.jpg: 640x640 2 litters, Done. (1.031s)
image 2/11 D:\yolov5-April\data\tcp\Image4 - Copy.jpg: 384x640 3 litters, Done. (0.630s)
image 3/11 D:\yolov5-April\data\tcp\Image4.jpg: 384x640 1 litter, Done. (0.566s)
image 4/11 D:\yolov5-April\data\tcp\Image5.jpg: 416x640 Done. (0.724s)
image 5/11 D:\yolov5-April\data\tcp\Image7.jpg: 544x640 2 litters, Done. (0.886s)
image 6/11 D:\yolov5-April\data\tcp\Image9.jpg: 416x640 1 litter, Done. (0.673s)
image 7/11 D:\yolov5-April\data\tcp\img10.jpg: 384x640 3 litters, Done. (0.635s)
image 8/11 D:\yolov5-April\data\tcp\img15.jpg: 384x640 2 litters, Done. (0.702s)
image 9/11 D:\yolov5-April\data\tcp\img6.jpg: 384x640 3 litters, Done. (0.582s)
image 10/11 D:\yolov5-April\data\tcp\img8.jpg: 384x640 2 litters, Done. (0.652s)
image 11/11 D:\yolov5-April\data\tcp\img9.jpg: 384x640 1 litter, Done. (0.644s)
Speed: 0.4ms pre-process, 703.7ms inference, 0.4ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/output
Sending folder: runs/output
Sending file: Image3.jpg (224799 bytes)
Sending file: Image4 - Copy.jpg (62867 bytes)
Sending file: Image4.jpg (342053 bytes)
Sending file: Image6.jpg (251933 bytes)
Sending file: Image7.jpg (437040 bytes)
Sending file: img10.jpg (600039 bytes)
Sending file: img15.jpg (680020 bytes)
Sending file: img6.jpg (624543 bytes)
Sending file: img8.jpg (713452 bytes)
Sending file: img9.jpg (630934 bytes)
Sending file: results.json (1028 bytes)
```

Due note though that for every new run on the Client side, the **ML-output** folder's contents will be replaced with the current run's Litter Detection results. If you wish to change this, feel free. However, be mindful of what changes you implement, and how they may affect the TCP connection between the Client and Server.

