Gloabl WheatHead detection using YOLOv5

```
!pip install -q kaggle
```

```
!pip install --upgrade --force-reinstall --no-deps kaggle
```

```
Collecting kaggle
  Downloading https://files.pythonhosted.org/packages/99/33/365c0d13f07a2a54744d027fe20b60dacdfdfb33bc04746db6ad0b7934
     |████████████████████████████████| 61kB 5.4MB/s
Building wheels for collected packages: kaggle
  Building wheel for kaggle (setup.py) ... done
  Created wheel for kaggle: filename=kaggle-1.5.10-cp36-none-any.whl size=73269 sha256=213ec0f50d6754b6dfee931eb41ee96
  Stored in directory: /root/.cache/pip/wheels/3a/d1/7e/6ce09b72b770149802c653a02783821629146983ee5a360f10
Successfully built kaggle
Installing collected packages: kaggle
  Found existing installation: kaggle 1.5.9
    Uninstalling kaggle-1.5.9:
      Successfully uninstalled kaggle-1.5.9
Successfully installed kaggle-1.5.10
```

```
# To make use of kaggle dataset(API permission)
from google.colab import files
files.upload()
```

Choose Files kaggle.json

- **kaggle.json**(application/json) - 76 bytes, last modified: 12/6/2020 - 100% done

```
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"jaswanthreddysareddy","key":"6f302c732b0b0b5362c1768b5bd10631"}'}
```

```
! mkdir ~/.kaggle
```

```
! cp kaggle.json ~/.kaggle/
```

```
# Sets the permission that only owner can modify the repository
!chmod 600 ~/.kaggle/kaggle.json
```

```
!kaggle datasets list
```

```
ref                                                            title                                             size  las
-------------------------------------------------------------  ------------------------------------------------  -----  ---
alexgude/california-traffic-collision-data-from-switrs         California Traffic Collision Data from SWITRS    1GB  202
babyoda/women-entrepreneurship-and-labor-force                 Women Entrepreneurship and Labor Force           1KB  202
szymonjanowski/internet-articles-data-with-users-engagement    Internet news data with readers engagement       3MB  202
sakshigoyal7/credit-card-customers                             Credit Card customers                          379KB  202
imoore/2020-us-general-election-turnout-rates                  2020 US General Election Turnout rates           4KB  202
afrniomelo/3w-dataset                                          3W Dataset - Undesirable events in oil wells   658MB  202
patrickb1912/ipl-complete-dataset-20082020                     IPL Complete Dataset (2008-2020)                1MB  202
mrmorj/us-politicians-twitter-dataset                          US Politicians Twitter Dataset                 68KB  202
arioboo/clumps-in-vela-galaxy-images                           Clumps in VELA galaxy images                    8MB  202
shivamb/netflix-shows                                          Netflix Movies and TV Shows                    971KB  202
unanimad/us-election-2020                                      US Election 2020                               429KB  202
manchunhui/us-election-2020-tweets                             US Election 2020 Tweets                        353MB  202
sootersaalu/amazon-top-50-bestselling-books-2009-2019          Amazon Top 50 Bestselling Books 2009 - 2019     15KB  202
terenceshin/covid19s-impact-on-airport-traffic                 COVID-19's Impact on Airport Traffic           106KB  202
antgoldbloom/covid19-data-from-john-hopkins-university         COVID-19 data from John Hopkins University      3MB  202
nehaprabhavalkar/indian-food-101                               Indian Food 101                                 7KB  202
datasnaek/youtube-new                                          Trending YouTube Video Statistics              201MB  201
zynicide/wine-reviews                                          Wine Reviews                                    51MB  201
google/tinyquickdraw                                           QuickDraw Sketches                              11GB  201
karangadiya/fifa19                                             FIFA 19 complete player dataset                  2MB  201
```

```
# Downloading kaggle dataset
!kaggle competitions download -c global-wheat-detection
```

```
Downloading global-wheat-detection.zip to /content
 98% 597M/607M [00:08<00:00, 93.8MB/s]
100% 607M/607M [00:09<00:00, 68.8MB/s]
```

```
# Unzipping kaggle dataset
!unzip global-wheat-detection.zip -d wheat_detection
```

```
  inflating: wheat_detection/train/fb6e8bd3f.jpg
  inflating: wheat_detection/train/fb804ab78.jpg
  inflating: wheat_detection/train/fb9e42d30.jpg
  inflating: wheat_detection/train/fbc79aea4.jpg
  inflating: wheat_detection/train/fbecfeba8.jpg
  inflating: wheat_detection/train/fc0978226.jpg
  inflating: wheat_detection/train/fc6860020.jpg
  inflating: wheat_detection/train/fc8cf268b.jpg
  inflating: wheat_detection/train/fcaaff29d.jpg
  inflating: wheat_detection/train/fcb0f108a.jpg
  inflating: wheat_detection/train/fcbc77665.jpg
  inflating: wheat_detection/train/fcf3194f7.jpg
  inflating: wheat_detection/train/fd151ae52.jpg
  inflating: wheat_detection/train/fd30a89bb.jpg
  inflating: wheat_detection/train/fd3c02c74.jpg
  inflating: wheat_detection/train/fd5624913.jpg
  inflating: wheat_detection/train/fd8b236e8.jpg
  inflating: wheat_detection/train/fd95e2677.jpg
  inflating: wheat_detection/train/fd9829305.jpg
  inflating: wheat_detection/train/fda2b546f.jpg
  inflating: wheat_detection/train/fda2e0448.jpg
  inflating: wheat_detection/train/fda86ae9a.jpg
  inflating: wheat_detection/train/fdbdb35fc.jpg
  inflating: wheat_detection/train/fde56a455.jpg
  inflating: wheat_detection/train/fe029eca4.jpg
  inflating: wheat_detection/train/fe0d6c114.jpg
  inflating: wheat_detection/train/fe125f8a6.jpg
  inflating: wheat_detection/train/fe1266f43.jpg
  inflating: wheat_detection/train/fe133ccb4.jpg
  inflating: wheat_detection/train/fe14db6a5.jpg
  inflating: wheat_detection/train/fe287c574.jpg
  inflating: wheat_detection/train/fe2a54e65.jpg
  inflating: wheat_detection/train/fe342a5bf.jpg
  inflating: wheat_detection/train/fe3d17ee2.jpg
  inflating: wheat_detection/train/fe3f956a2.jpg
  inflating: wheat_detection/train/fe5c1ea5a.jpg
  inflating: wheat_detection/train/fe5c215e2.jpg
  inflating: wheat_detection/train/fe641f02e.jpg
  inflating: wheat_detection/train/fe6e2ecbc.jpg
  inflating: wheat_detection/train/fe7ecb1f4.jpg
```

```
  inflating: wheat_detection/train/fe8876602.jpg
  inflating: wheat_detection/train/feac3a701.jpg
  inflating: wheat_detection/train/feb09b731.jpg
  inflating: wheat_detection/train/feb78a8de.jpg
  inflating: wheat_detection/train/fec6edf64.jpg
  inflating: wheat_detection/train/feda9265c.jpg
  inflating: wheat_detection/train/fefd59985.jpg
  inflating: wheat_detection/train/ff1716939.jpg
  inflating: wheat_detection/train/ff46aa8d4.jpg
  inflating: wheat_detection/train/ff5c9104a.jpg
  inflating: wheat_detection/train/ff86547fb.jpg
  inflating: wheat_detection/train/ffaa964a2.jpg
  inflating: wheat_detection/train/ffb0f6eca.jpg
  inflating: wheat_detection/train/ffb445410.jpg
  inflating: wheat_detection/train/ffbb9c623.jpg
  inflating: wheat_detection/train/ffbf75e5b.jpg
  inflating: wheat_detection/train/ffbfe7cc0.jpg
  inflating: wheat_detection/train/ffc870198.jpg
  inflating: wheat_detection/train/ffdf83e42.jpg
```

```
!pwd
```

```
/content
```

```
!git clone  https://github.com/ultralytics/yolov5
```

```
Cloning into 'yolov5'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 3212 (delta 21), reused 22 (delta 8), pack-reused 3165
Receiving objects: 100% (3212/3212), 6.49 MiB | 22.29 MiB/s, done.
Resolving deltas: 100% (2133/2133), done.
```

```
# To support YAML tags
!pip install -U PyYAML
```

```
Collecting PyYAML
  Downloading https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d32151
```

```
     |████████████████████████████████| 276kB 8.7MB/s
Building wheels for collected packages: PyYAML
  Building wheel for PyYAML (setup.py) ... done
  Created wheel for PyYAML: filename=PyYAML-5.3.1-cp36-cp36m-linux_x86_64.whl size=44619 sha256=c60fddc72248b91240971d
  Stored in directory: /root/.cache/pip/wheels/a7/c1/ea/cf5bd31012e735dc1dfea3131a2d5eae7978b251083d6247bd
Successfully built PyYAML
Installing collected packages: PyYAML
  Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed PyYAML-5.3.1
```

```
import pandas as pd
import tensorflow as tf
import yaml
import os
import ast #for converting "[]" to []
from sklearn import model_selection
import numpy as np
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt
import shutil #used for file copying
from tqdm import tqdm #Tqdm package is one of the more comprehensive packages for progress bars with python and is handy for
```

```
!mkdir wheat_data
```

```
!mkdir /content/wheat_data/images
```

```
!mkdir /content/wheat_data/labels
```

```
!mkdir /content/wheat_data/images/train
!mkdir /content/wheat_data/images/validation
```

```
!mkdir /content/wheat_data/labels/train
!mkdir /content/wheat_data/labels/validation


input_path = '/content/wheat_detection'
output_path = '/content/wheat_data'

def process_data(data, data_type = "train"):
    for _, row in tqdm (data.iterrows(), total = len(data)):
        image_name = row["image_id"]
        bounding_boxes = row["bboxes"]
        yolo_data = []
        for bbox in bounding_boxes:
            x = bbox[0]
            y = bbox[1]
            w = bbox[2]
            h = bbox[3]
            x_center = x + w /2
            y_center = y + h /2
            x_center /= 1024.0
            y_center /= 1024.0
            w /= 1024.0
            h /= 1024.0
            yolo_data.append([0, x_center, y_center, w, h]) #first we append the class to which bounding box belongs to
        yolo_data = np.array(yolo_data)
        np.savetxt(
            os.path.join(output_path, f"labels/{data_type}/{image_name}.txt"),
            yolo_data,
            fmt = ["%d", "%f", "%f", "%f", "%f"]
        )
        shutil.copyfile(
            os.path.join(input_path, f"train/{image_name}.jpg"),
            os.path.join(output_path, f"images/{data_type}/{image_name}.jpg")
        )

if __name__ == "__main__":
  df = pd.read_csv(os.path.join(input_path, 'train.csv' ))
  df.bbox = df.bbox.apply(ast.literal_eval)
```

```
  df = df.groupby('image_id')['bbox'].apply(list).reset_index(name = 'bboxes')

  df_train,df_validate = model_selection.train_test_split(df,test_size = 0.1, random_state=42, shuffle = True)

  df_train = df_train.reset_index(drop=True)
  df_validate = df_validate.reset_index(drop=True)

  process_data(df_train, data_type='train')
  process_data(df_validate, data_type='validation')
```

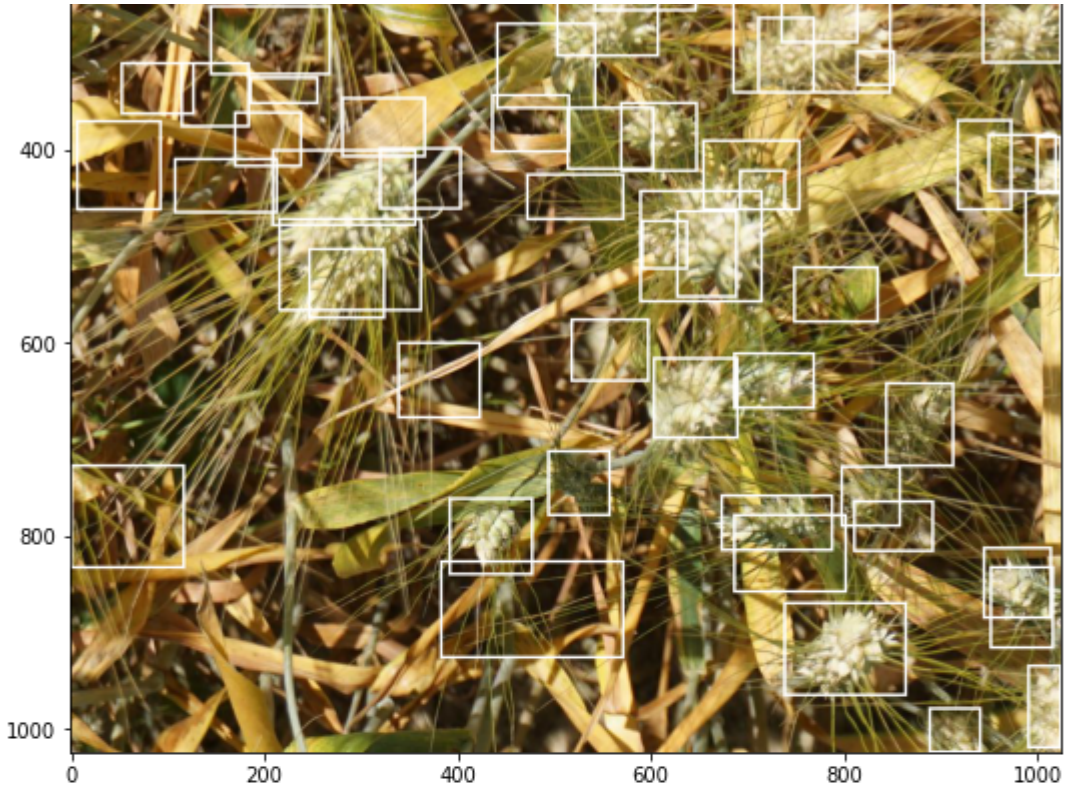
```
100%|██████████| 3035/3035 [00:03<00:00, 878.98it/s]
100%|██████████| 338/338 [00:00<00:00, 1006.90it/s]
```

```
# Displaying first 10 images bounding box
df_train.head(10)
```

	image_id	bboxes
0	6d34eb3b2	[[391, 760, 87, 80], [208, 400, 149, 79], [215...
1	c7c00c1a3	[[91, 10, 91, 66], [753, 0, 58, 38], [570, 0, ...
2	09b4141a6	[[245.0, 240.0, 113.0, 110.0], [296.0, 709.0, ...
3	e837f4e67	[[491.0, 473.0, 159.0, 87.0], [558.0, 225.0, 8...
4	f1d20d58a	[[459.0, 917.0, 153.0, 103.0], [684.0, 5.0, 60...
5	894baab2e	[[0, 124, 31, 87], [105, 102, 57, 61], [186, 1...
6	a28dcf45d	[[0, 403, 26, 94], [150, 154, 125, 68], [378, ...
7	7ff95f9ec	[[614, 239, 143, 139], [905, 90, 111, 130], [8...
8	8dd8f0ca6	[[50.0, 80.0, 56.0, 93.0], [73.0, 289.0, 64.0,...
9	e935b9eea	[[6.0, 0.0, 75.0, 86.0], [27.0, 109.0, 63.0, 8...

```
# Building bounding box for the image

def display_images(sample):
  f, ax = plt.subplots(len(sample), figsize = (20,20))
```

```
    for i, image_id in enumerate(sample['image_id']):
      image_path = os.path.join('/content/wheat_detection/train', f'{image_id}.jpg')
      image = Image.open(image_path)
      draw = ImageDraw.Draw(image)
      for bbox1 in sample['bboxes']:
        for bbox2 in bbox1:
          draw.rectangle([bbox2[0], bbox2[1], bbox2[0] + bbox2[2], bbox2[1] + bbox2[3]], width = 3)
      ax[i].imshow(image)

display_images(df_train.head(2))
```




```
!ls /content/wheat_data/images/train
```

```
32a203fe5.jpg  64aff66ec.jpg  96e14c7cd.jpg  c7c00c1a3.jpg  fb2eae229.jpg
32f716667.jpg  64b352e9a.jpg  96efbda67.jpg  c7c9b4e72.jpg  fb41c061c.jpg
331a697b2.jpg  64c39c542.jpg  96fa00d50.jpg  c7d3b408d.jpg  fb456d16f.jpg
331c69189.jpg  64ee2e175.jpg  97316e61d.jpg  c7df66915.jpg  fb52ca9b2.jpg
```

```
331ce1ee1.jpg  652963694.jpg  97372d371.jpg  c80802313.jpg  fb5807958.jpg
333d4affe.jpg  654a741f5.jpg  974b78929.jpg  c8237c2e6.jpg  fb6e04c0c.jpg
336d2e4ce.jpg  655532cc7.jpg  975c97e30.jpg  c83c24607.jpg  fb6e8bd3f.jpg
33b939a6a.jpg  655c1ffdd.jpg  9764d9f54.jpg  c83d339fe.jpg  fb804ab78.jpg
33c69a58a.jpg  655ef2d6b.jpg  976746c7d.jpg  c85eedb4c.jpg  fb9e42d30.jpg
33ea56e1c.jpg  657f6bf98.jpg  97788b3ae.jpg  c86621747.jpg  fbc79aea4.jpg
33f469471.jpg  65cb7e88b.jpg  9780d64f5.jpg  c87c9e9b2.jpg  fbecfeba8.jpg
340ac8d52.jpg  65d6480b2.jpg  97851b9c6.jpg  c8851847c.jpg  fc0978226.jpg
342af9602.jpg  65f13d74f.jpg  9790095bb.jpg  c88635e09.jpg  fc6860020.jpg
344bdc291.jpg  65f5afe35.jpg  97b19f207.jpg  c88836133.jpg  fc8cf268b.jpg

3455cbc17.jpg  6639b23df.jpg  97ba6330a.jpg  c88f6408f.jpg  fcaaff29d.jpg
346efc9f4.jpg  6651c6f2f.jpg  97f0e1800.jpg  c8909d7a1.jpg  fcb0f108a.jpg
347ecba39.jpg  666ca30b3.jpg  980c88da5.jpg  c8927c5e1.jpg  fcbc77665.jpg
348059f04.jpg  666e0a853.jpg  9832f05e1.jpg  c89aed93e.jpg  fcf3194f7.jpg
348f3c34b.jpg  667b4a999.jpg  9858d67dc.jpg  c8b81529c.jpg  fd151ae52.jpg
34b5b6acb.jpg  66adaf07a.jpg  988a60e6e.jpg  c8c449bf0.jpg  fd30a89bb.jpg
34b96b38e.jpg  66c96dce0.jpg  98ce36a93.jpg  c8defc0d1.jpg  fd3c02c74.jpg
34c5e70d5.jpg  66d517e44.jpg  98db75bde.jpg  c8e5e50ec.jpg  fd5624913.jpg
34d3f0351.jpg  66d91891d.jpg  990188afc.jpg  c9175cfdd.jpg  fd8b236e8.jpg
34df9ca13.jpg  66ee3048c.jpg  990c1777d.jpg  c925fda69.jpg  fd95e2677.jpg
34f2e3b92.jpg  671fb8771.jpg  994ec0471.jpg  c936df99a.jpg  fd9829305.jpg
34f7f196b.jpg  673b64e17.jpg  997a7cf9b.jpg  c94e8a60c.jpg  fda2b546f.jpg
350530720.jpg  677da7510.jpg  9983d297c.jpg  c94f8dd92.jpg  fda2e0448.jpg
352c3a397.jpg  678011fc2.jpg  99886a700.jpg  c97adf022.jpg  fda86ae9a.jpg
352e238a9.jpg  67a70d2d7.jpg  9988ee066.jpg  c988834c9.jpg  fdbdb35fc.jpg
3542457e2.jpg  67bc08594.jpg  999645d8a.jpg  c9a573b93.jpg  fde56a455.jpg
35af6e100.jpg  67e5c5d41.jpg  999d9d27d.jpg  c9ade75be.jpg  fe029eca4.jpg
35b59e0a0.jpg  6811a6ee9.jpg  99d221ffe.jpg  c9b23ec87.jpg  fe0d6c114.jpg
35b935b6c.jpg  681baf2e2.jpg  99d5eb61c.jpg  c9d1bc268.jpg  fe125f8a6.jpg
35d6a4509.jpg  682f78993.jpg  99e092761.jpg  c9d60e075.jpg  fe1266f43.jpg
35f84d33a.jpg  684a2c62a.jpg  99f9c7b01.jpg  c9f534ca0.jpg  fe133ccb4.jpg
366187e59.jpg  684dfcf04.jpg  9a11f9b1d.jpg  ca08214f1.jpg  fe14db6a5.jpg
3684c0fae.jpg  68561559e.jpg  9a1947520.jpg  ca2b57db7.jpg  fe287c574.jpg
36d1cc44b.jpg  687972820.jpg  9a30dd802.jpg  ca3101e66.jpg  fe2a54e65.jpg
36e469dbf.jpg  68b603861.jpg  9a34996be.jpg  ca4769387.jpg  fe342a5bf.jpg
36fb90706.jpg  68fa4a555.jpg  9a3fe7557.jpg  ca4cb597b.jpg  fe3d17ee2.jpg
37073b5c1.jpg  68fedffd4.jpg  9a50eab86.jpg  ca5e51e59.jpg  fe3f956a2.jpg
3712db803.jpg  6902bbaf2.jpg  9a591b677.jpg  ca664da7d.jpg  fe5c1ea5a.jpg
37191de77.jpg  69041dcff.jpg  9a87c92e7.jpg  ca7d49213.jpg  fe5c215e2.jpg
372d12c55.jpg  690c7d8f0.jpg  9a9f42302.jpg  caa7d043e.jpg  fe641f02e.jpg
3733d0e53.jpg  69286a6cc.jpg  9aa0487b0.jpg  cab50874d.jpg  fe7ecb1f4.jpg
3738685a3.jpg  6939a6e17.jpg  9aa5dec56.jpg  cabe03050.jpg  fe8876602.jpg
3741873d5.jpg  69595016d.jpg  9ad34dab3.jpg  cacd820db.jpg  feac3a701.jpg
3744f2482.jpg  695ae4a12.jpg  9adbfe503.jpg  cadb3740d.jpg  feb09b731.jpg
```

```
3744f2482.jpg  695ae4a12.jpg  9adbfe565.jpg  caab5746d.jpg  feb69b751.jpg
3753c58f7.jpg  695cd49dd.jpg  9adca8ff8.jpg  cb0a32cbb.jpg  feb78a8de.jpg
375a4795b.jpg  696530284.jpg  9ae3752f9.jpg  cb256df87.jpg  fec6edf64.jpg
376689e0c.jpg  69662aa1c.jpg  9aea92dce.jpg  cb34f7509.jpg  fefd59985.jpg
3782f5786.jpg  69809f5c0.jpg  9b260d538.jpg  cb4d2d66b.jpg  ff46aa8d4.jpg
378966871.jpg  69855d3de.jpg  9b4ad2587.jpg  cb7a2a690.jpg  ff5c9104a.jpg
379ef727d.jpg  69913c545.jpg  9b58d4c4e.jpg  cbab952fe.jpg  ffb0f6eca.jpg
37a28463a.jpg  6994e1821.jpg  9b783b776.jpg  cbbc58a4c.jpg  ffbf75e5b.jpg
37b0d5ea3.jpg  699810258.jpg  9ba7810c6.jpg  cbfaa3ed9.jpg  ffbfe7cc0.jpg

37bcaaeb6.jpg  69a211ccf.jpg  9bbee18cc.jpg  cbfd50139.jpg  ffc870198.jpg
```

```
!ls /content/wheat_data/labels/train
```

```
328dde875.txt  64a883d10.txt  96db737c8.txt  c7ba07095.txt  fb2b30bc9.txt
32a203fe5.txt  64aff66ec.txt  96e14c7cd.txt  c7c00c1a3.txt  fb2eae229.txt
32f716667.txt  64b352e9a.txt  96efbda67.txt  c7c9b4e72.txt  fb41c061c.txt
331a697b2.txt  64c39c542.txt  96fa00d50.txt  c7d3b408d.txt  fb456d16f.txt
331c69189.txt  64ee2e175.txt  97316e61d.txt  c7df66915.txt  fb52ca9b2.txt
331cefee1.txt  652903094.txt  97372d371.txt  c808023f3.txt  fb5867958.txt
333d4affe.txt  654a741f5.txt  974b78929.txt  c8237c2e6.txt  fb6e04c0c.txt
336d2e4ce.txt  655532cc7.txt  975c97e30.txt  c83c24607.txt  fb6e8bd3f.txt
33b939a6a.txt  655c1ffdd.txt  9764d9f54.txt  c83d339fe.txt  fb804ab78.txt
33c69a58a.txt  655ef2d6b.txt  976746c7d.txt  c85eedb4c.txt  fb9e42d30.txt
33ea56e1c.txt  657f6bf98.txt  97788b3ae.txt  c86621747.txt  fbc79aea4.txt
33f469471.txt  65cb7e88b.txt  9780d64f5.txt  c87c9e9b2.txt  fbecfeba8.txt
340ac8d52.txt  65d6480b2.txt  97851b9c6.txt  c8851847c.txt  fc0978226.txt
342af9602.txt  65f13d74f.txt  9790095bb.txt  c88635e09.txt  fc6860020.txt
344bdc291.txt  65f5afe35.txt  97b19f207.txt  c88836133.txt  fc8cf268b.txt
3455cbc17.txt  6639b23df.txt  97ba6330a.txt  c88f6408f.txt  fcaaff29d.txt
346efc9f4.txt  6651c6f2f.txt  97f0e1800.txt  c8909d7a1.txt  fcb0f108a.txt
347ecba39.txt  666ca30b3.txt  980c88da5.txt  c8927c5e1.txt  fcbc77665.txt
348059f04.txt  666e0a853.txt  9832f05e1.txt  c89aed93e.txt  fcf3194f7.txt
348f3c34b.txt  667b4a999.txt  9858d67dc.txt  c8b81529c.txt  fd151ae52.txt
34b5b6acb.txt  66adaf07a.txt  988a60e6e.txt  c8c449bf0.txt  fd30a89bb.txt
34b96b38e.txt  66c96dce0.txt  98ce36a93.txt  c8defc0d1.txt  fd3c02c74.txt
34c5e70d5.txt  66d517e44.txt  98db75bde.txt  c8e5e50ec.txt  fd5624913.txt
34d3f0351.txt  66d91891d.txt  990188afc.txt  c9175cfdd.txt  fd8b236e8.txt
34df9ca13.txt  66ee3048c.txt  990c1777d.txt  c925fda69.txt  fd95e2677.txt
34f2e3b92.txt  671fb8771.txt  994ec0471.txt  c936df99a.txt  fd9829305.txt
34f7f196b.txt  673b64e17.txt  997a7cf9b.txt  c94e8a60c.txt  fda2b546f.txt
350530720.txt  677da7510.txt  9983d297c.txt  c94f8dd92.txt  fda2e0448.txt
```

```
352c3a397.txt  67811fc2.txt  99886a700.txt  c97adf022.txt  fda86ae9a.txt
352e238a9.txt  67a70d2d7.txt  9988ee066.txt  c988834c9.txt  fdbdb35fc.txt
3542457e2.txt  67bc08594.txt  999645d8a.txt  c9a573b93.txt  fde56a455.txt
35af6e100.txt  67e5c5d41.txt  999d9d27d.txt  c9ade75be.txt  fe029eca4.txt
35b59e0a0.txt  6811a6ee9.txt  99d221ffe.txt  c9b23ec87.txt  fe0d6c114.txt
35b935b6c.txt  681baf2e2.txt  99d5eb61c.txt  c9d1bc268.txt  fe125f8a6.txt
35d6a4509.txt  682f78993.txt  99e092761.txt  c9d60e075.txt  fe1266f43.txt
35f84d33a.txt  684a2c62a.txt  99f9c7b01.txt  c9f534ca0.txt  fe133ccb4.txt
366187e59.txt  684dfcf04.txt  9a11f9b1d.txt  ca08214f1.txt  fe14db6a5.txt
3684c0fae.txt  68561559e.txt  9a1947520.txt  ca2b57db7.txt  fe287c574.txt
36d1cc44b.txt  687972820.txt  9a30dd802.txt  ca3101e66.txt  fe2a54e65.txt
36e469dbf.txt  68b603861.txt  9a34996be.txt  ca4769387.txt  fe342a5bf.txt
36fb90706.txt  68fa4a555.txt  9a3fe7557.txt  ca4cb597b.txt  fe3d17ee2.txt
37073b5c1.txt  68fedffd4.txt  9a50eab86.txt  ca5e51e59.txt  fe3f956a2.txt
3712db803.txt  6902bbaf2.txt  9a591b677.txt  ca664da7d.txt  fe5c1ea5a.txt
37191de77.txt  69041dcff.txt  9a87c92e7.txt  ca7d49213.txt  fe5c215e2.txt
372d12c55.txt  690c7d8f0.txt  9a9f42302.txt  caa7d043e.txt  fe641f02e.txt
3733d0e53.txt  69286a6cc.txt  9aa0487b0.txt  cab50874d.txt  fe7ecb1f4.txt
3738685a3.txt  6939a6e17.txt  9aa5dec56.txt  cabe03050.txt  fe8876602.txt
3741873d5.txt  69595016d.txt  9ad34dab3.txt  cacd820db.txt  feac3a701.txt
3744f2482.txt  695ae4a12.txt  9adbfe503.txt  cadb3740d.txt  feb09b731.txt
3753c58f7.txt  695cd49dd.txt  9adca8ff8.txt  cb0a32cbb.txt  feb78a8de.txt
375a4795b.txt  696530284.txt  9ae3752f9.txt  cb256df87.txt  fec6edf64.txt
376689e0c.txt  69662aa1c.txt  9aea92dce.txt  cb34f7509.txt  fefd59985.txt
3782f5786.txt  69809f5c0.txt  9b260d538.txt  cb4d2d66b.txt  ff46aa8d4.txt
378966871.txt  69855d3de.txt  9b4ad2587.txt  cb7a2a690.txt  ff5c9104a.txt
379ef727d.txt  69913c545.txt  9b58d4c4e.txt  cbab952fe.txt  ffb0f6eca.txt
37a28463a.txt  6994e1821.txt  9b783b776.txt  cbbc58a4c.txt  ffbf75e5b.txt
37b0d5ea3.txt  699810258.txt  9ba7810c6.txt  cbfaa3ed9.txt  ffbfe7cc0.txt
37bcaaeb6.txt  69a211ccf.txt  9bbee18cc.txt  cbfd50139.txt  ffc870198.txt
37e705df9.txt  69b3074c4.txt  9bc09acf1.txt  cc1a14dfc.txt  ffdf83e42.txt
```

```
!cat /content/wheat_data/labels/train/fda86ae9a.txt
```

```
0 0.059082 0.784180 0.118164 0.099609
0 0.151855 0.640625 0.106445 0.164062
0 0.294922 0.727539 0.080078 0.082031
0 0.220508 0.915039 0.117188 0.132812
0 0.370117 0.828125 0.123047 0.083984
0 0.557617 0.777344 0.128906 0.082031
0 0.767578 0.812012 0.128906 0.086914
0 0.660156 0.933594 0.074219 0.126953
```

```
0 0.870605 0.770996 0.088867 0.084961
0 0.557129 0.581055 0.249023 0.228516
0 0.482422 0.575195 0.126953 0.121094
0 0.326172 0.488770 0.130859 0.192383
0 0.065918 0.313965 0.073242 0.067383
0 0.068359 0.172852 0.111328 0.115234
0 0.014160 0.031738 0.028320 0.055664
0 0.119141 0.086426 0.107422 0.081055
0 0.333496 0.195312 0.084961 0.144531
0 0.655273 0.308594 0.207031 0.115234
0 0.895508 0.404297 0.142578 0.101562
0 0.965820 0.532715 0.068359 0.143555
0 0.926367 0.100977 0.147266 0.181641
0 0.625488 0.158691 0.108398 0.094727
0 0.352539 0.054688 0.279297 0.109375
0 0.026367 0.457031 0.050781 0.074219
0 0.020020 0.394043 0.040039 0.063477
0 0.020508 0.552734 0.041016 0.101562
0 0.154297 0.451660 0.171875 0.217773
0 0.302246 0.552246 0.067383 0.075195
0 0.539062 0.980469 0.119141 0.039062
0 0.970215 0.969727 0.055664 0.060547
```

```
!ls
```

```
drive                        kaggle.json  wheat_data       yolov5
global-wheat-detection.zip   sample_data  wheat_detection
```

```
#YAML is a human-readable data-serialization language. It is commonly used for configuration files and in applications where
!cat wheat.yaml
```

```
cat: wheat.yaml: No such file or directory
```

```
!cat /content/drive/MyDrive/wheat.yaml
```

```
# COCO 2017 dataset http://cocodataset.org - first 128 training images
# Download command:  python -c "from yolov5.utils.google_utils import gdrive_download; gdrive_download('1n_oKgR81BJtqk
# Train command: python train.py --data ./data/coco128.yaml
# Dataset should be placed next to yolov5 folder:
#   /parent_folder
```

```
#     /coco128
#     /yolov5


# train and val datasets (image directory or *.txt file with image paths)
train:  /content/wheat_data/images/train
val: /content/wheat_data/images/validation

# number of classes
nc: 1

# class names
names: ['wheat']
```

```
!ls /content/yolov5/models
```

```
common.py        export.py  __init__.py  yolov5l.yaml  yolov5s.yaml
experimental.py  hub        yolo.py      yolov5m.yaml  yolov5x.yaml
```

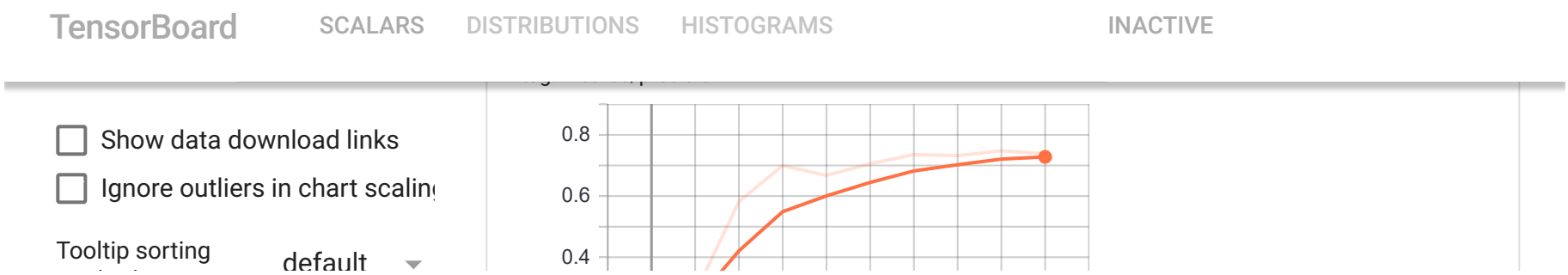
```
%load_ext tensorboard
```

```
# Training the model
!python3 /content/yolov5/train.py --img 1024 --batch 8 --epochs 10 --data /content/drive/MyDrive/wheat.yaml --cfg /content/y
```

```
Using 2 dataloader workers
Logging results to runs/train/wheat_detector_model
Starting training for 10 epochs...

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
  0% 0/380 [00:00<?, ?it/s]NumExpr defaulting to 2 threads.
      0/9     4.36G   0.08321    0.1468         0    0.2301       138      1024: 100% 380/380 [04:07<00:00,  1.54it

               Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:18<00:0
                 all         338    1.51e+04       0.204       0.904        0.64       0.205

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
      1/9     4.32G   0.05912    0.1335         0    0.1926        89      1024: 100% 380/380 [03:51<00:00,  1.64it

               Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:12<00:0
                 all         338    1.51e+04       0.283       0.948       0.838       0.392
```

```
      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        2/9     4.32G   0.05018    0.1312         0    0.1813       175      1024: 100% 380/380 [03:49<00:00,  1.66it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:11<00:0
                  all         338    1.51e+04       0.582       0.939       0.919       0.462

      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        3/9     4.32G   0.04421    0.1269         0    0.1711       129      1024: 100% 380/380 [03:51<00:00,  1.64it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:11<00:0
                  all         338    1.51e+04         0.7       0.937       0.931        0.48

      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        4/9     4.32G   0.04148    0.1247         0    0.1662       112      1024: 100% 380/380 [03:53<00:00,  1.63it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:11<00:0
                  all         338    1.51e+04       0.667       0.938        0.93       0.483

      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        5/9     4.32G   0.03928    0.1239         0    0.1632       157      1024: 100% 380/380 [03:54<00:00,  1.62it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:11<00:0
                  all         338    1.51e+04       0.706       0.944       0.942       0.525

      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        6/9     4.32G   0.03787    0.1214         0    0.1592       211      1024: 100% 380/380 [03:53<00:00,  1.63it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:11<00:0
                  all         338    1.51e+04       0.736       0.944       0.946       0.536

      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        7/9     4.32G     0.037    0.1208         0    0.1578       149      1024: 100% 380/380 [03:55<00:00,  1.62it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:11<00:0
                  all         338    1.51e+04       0.732        0.95       0.949       0.544

      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        8/9     4.32G    0.0365    0.1189         0    0.1554       149      1024: 100% 380/380 [03:57<00:00,  1.60it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:12<00:0
                  all         338    1.51e+04       0.748       0.947        0.95       0.549


      Epoch   gpu_mem       box       obj       cls     total   targets  img_size
        9/9     4.32G   0.03632    0.1182         0    0.1545       157      1024: 100% 380/380 [04:06<00:00,  1.54it
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 43/43 [00:16<00:0
                  all         338    1.51e+04       0.739        0.95       0.951       0.551
Optimizer stripped from runs/train/wheat_detector_model/weights/last.pt, 14.9MB
Optimizer stripped from runs/train/wheat_detector_model/weights/best.pt, 14.9MB
10 epochs completed in 0.694 hours.
```

```
# To see graphically using tensorboard
tensorboard --logdir runs
```




```
!python3 /content/yolov5/detect.py --source /content/wheat_detection/test --weights /content/runs/train/wheat_detector_model
```

```
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.25, device='', exist_ok=False, img_size=640, i
Using torch 1.7.0+cu101 CUDA:0 (Tesla T4, 15079MB)

Fusing layers...
Model Summary: 232 layers, 7246518 parameters, 0 gradients
image 1/10 /content/wheat_detection/test/2fd875eaa.jpg: 640x640 30 wheats, Done. (0.013s)
image 2/10 /content/wheat_detection/test/348a992bb.jpg: 640x640 36 wheats, Done. (0.012s)
image 3/10 /content/wheat_detection/test/51b3e36ab.jpg: 640x640 26 wheats, Done. (0.013s)
image 4/10 /content/wheat_detection/test/51f1be19e.jpg: 640x640 20 wheats, Done. (0.012s)
image 5/10 /content/wheat_detection/test/53f253011.jpg: 640x640 33 wheats, Done. (0.012s)
image 6/10 /content/wheat_detection/test/796707dd7.jpg: 640x640 23 wheats, Done. (0.012s)
image 7/10 /content/wheat_detection/test/aac893a91.jpg: 640x640 22 wheats, Done. (0.012s)
image 8/10 /content/wheat_detection/test/cb8d261a3.jpg: 640x640 25 wheats, Done. (0.012s)
image 9/10 /content/wheat_detection/test/cc3532ff6.jpg: 640x640 28 wheats, Done. (0.012s)
image 10/10 /content/wheat_detection/test/f5a1f0358.jpg: 640x640 30 wheats, Done. (0.012s)
Results saved to runs/detect/exp10
Done. (0.612s)
```


```
fig, ax = plt.subplots(figsize = (10, 10))
image = Image.open('/content/runs/detect/exp10/2fd875eaa.jpg')
ax.imshow(image)
```

```
<matplotlib.image.AxesImage at 0x7f2390583a90>
```


```
fig, ax = plt.subplots(figsize = (10, 10))
image = Image.open('/content/runs/detect/exp10/796707dd7.jpg')
ax.imshow(image)
```

```
<matplotlib.image.AxesImage at 0x7f23905e4780>
```