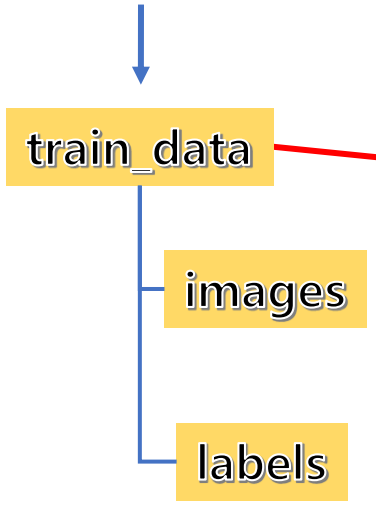
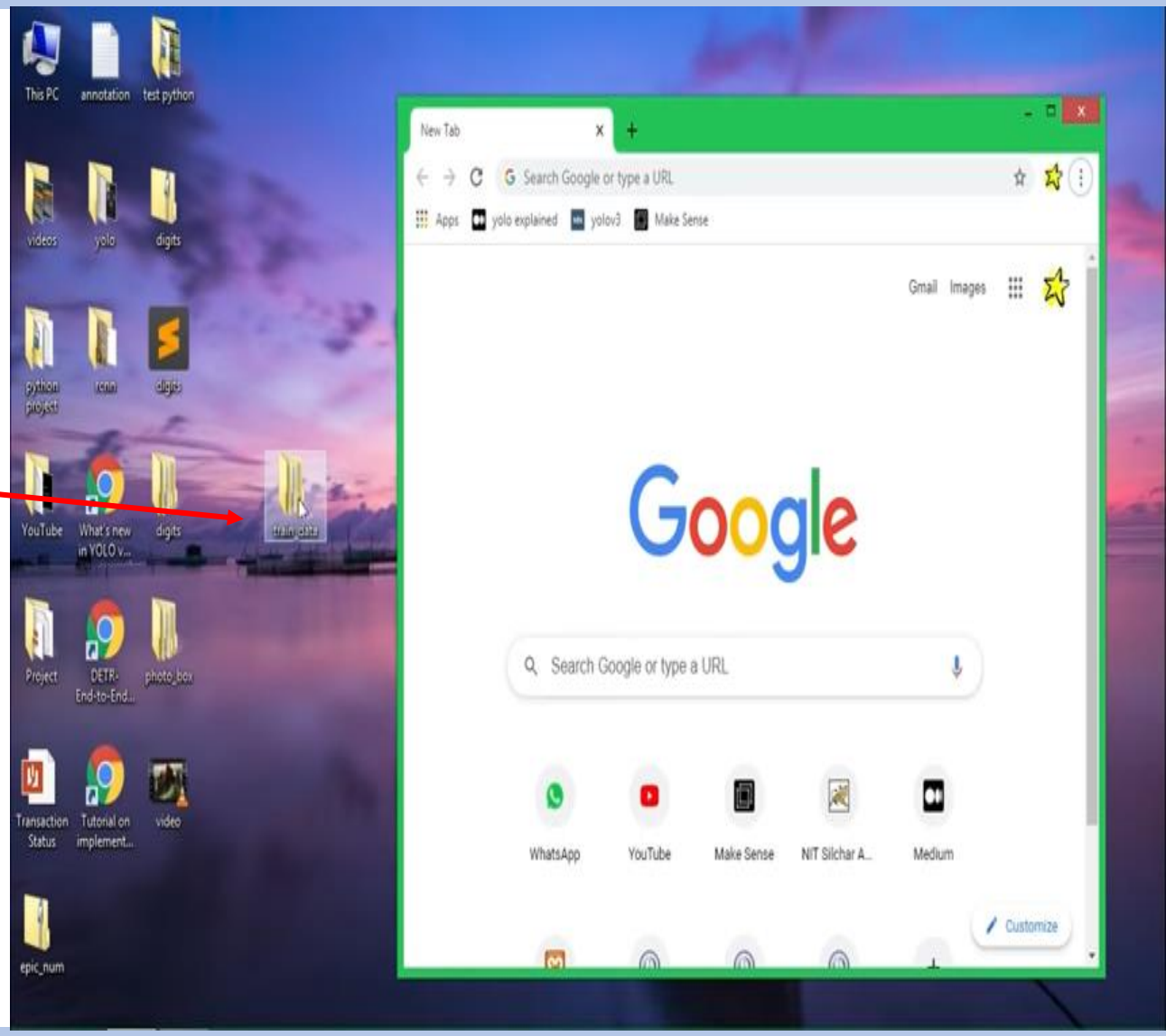
How to run YOLO

Data preparation : Make Your Own Dataset

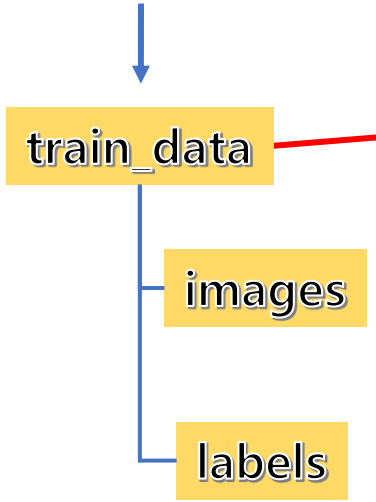
Create Data Folder

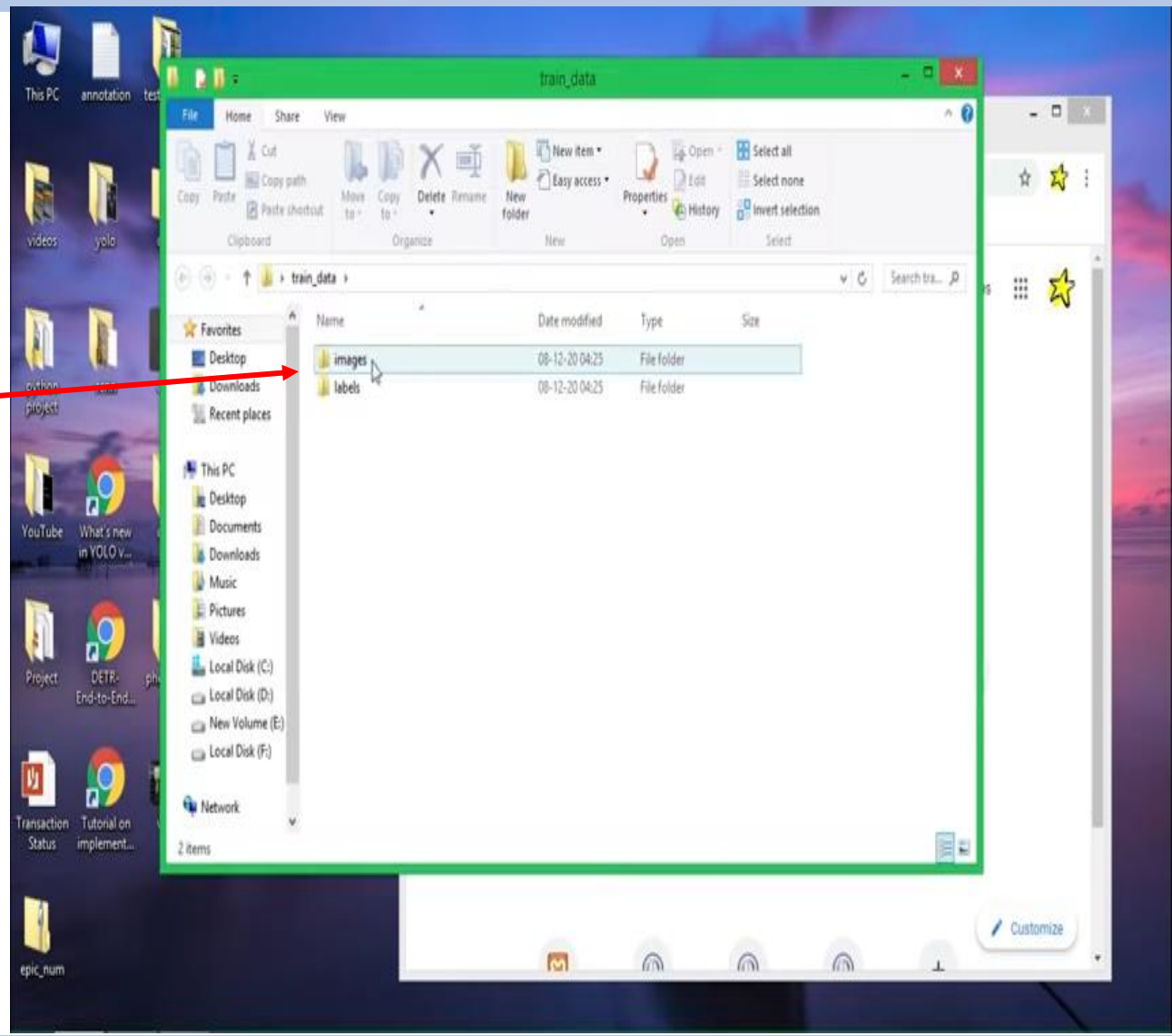




Data preparation : Make Your Own Dataset

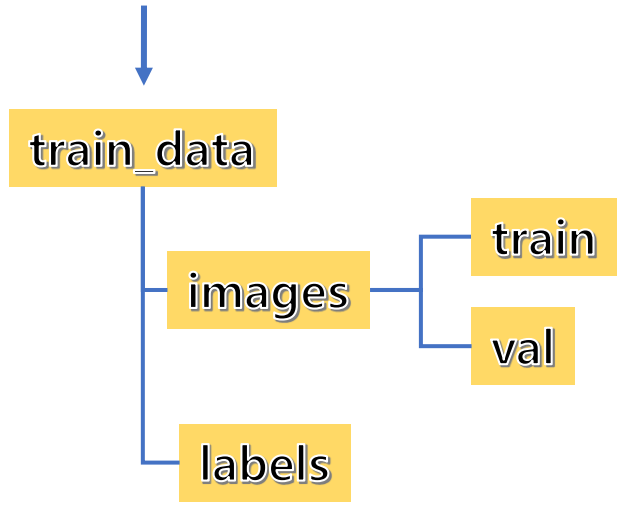
Create Data Folder

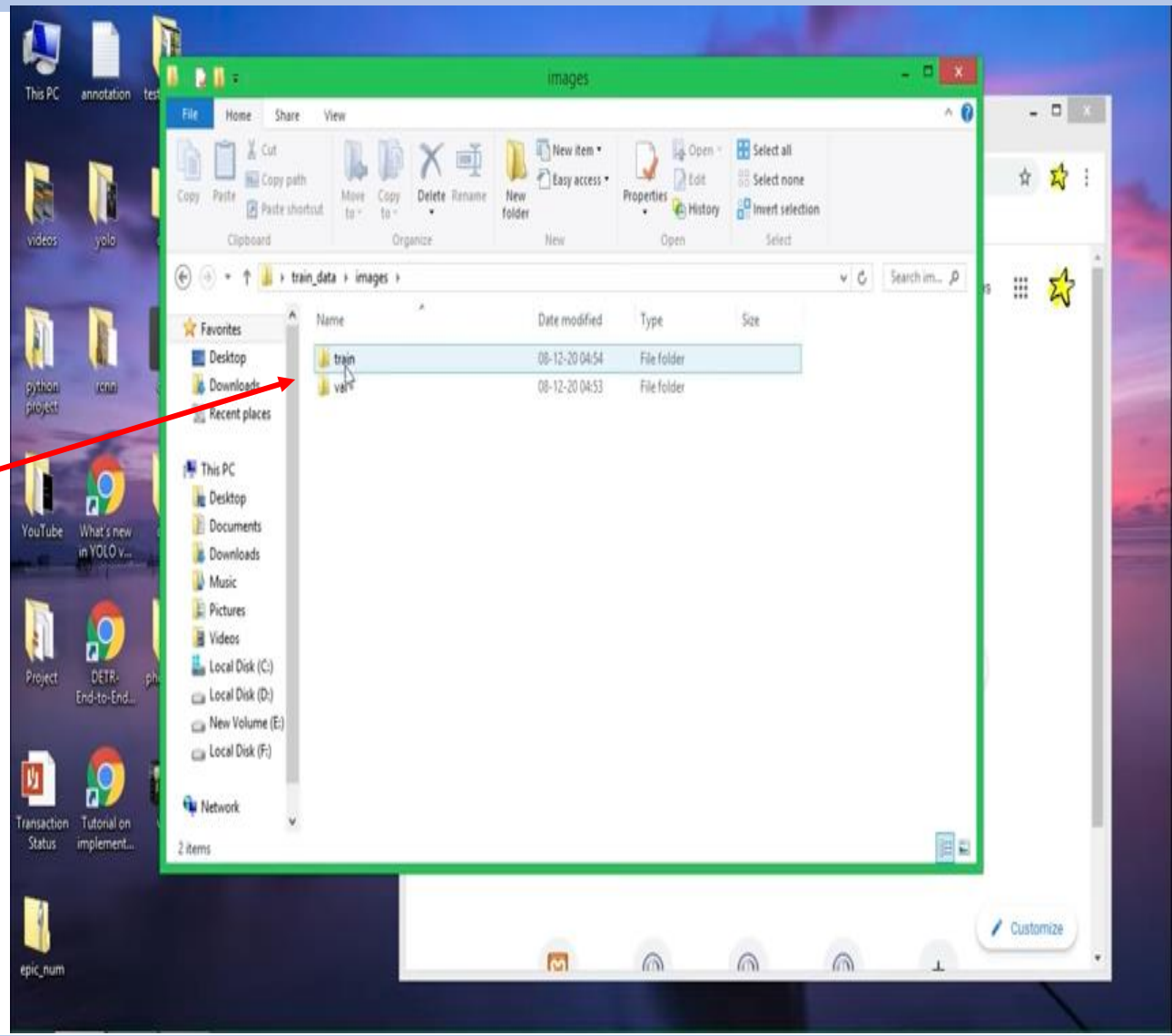




Data preparation : Make Your Own Dataset

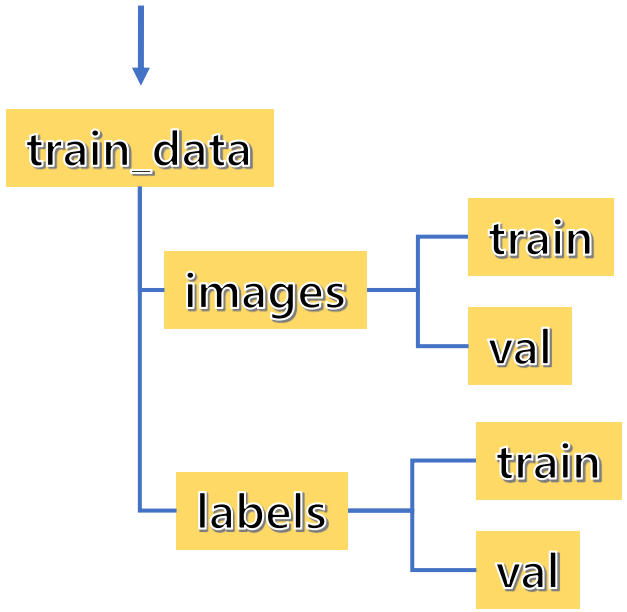
Create Data Folder

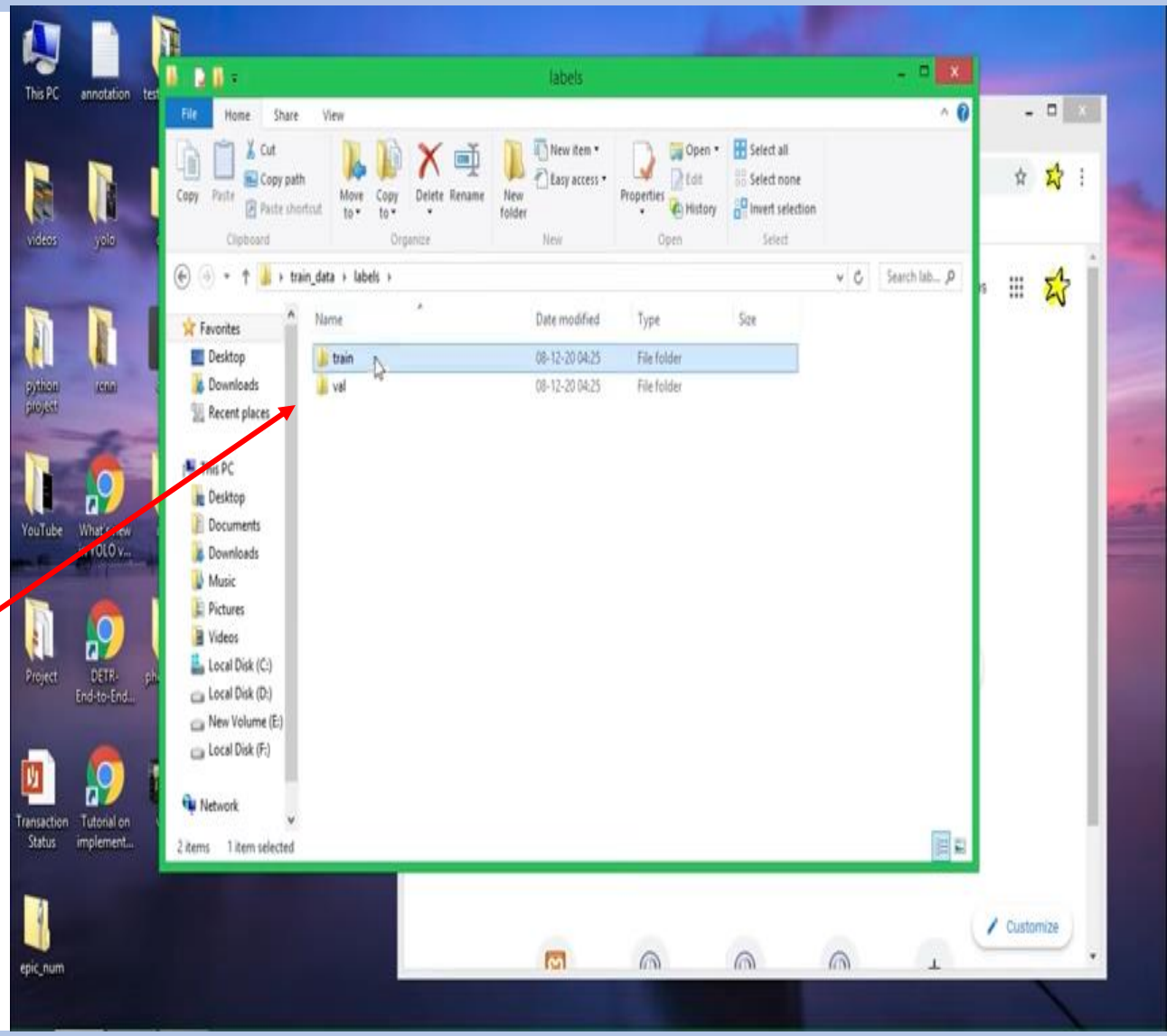




Data preparation : Make Your Own Dataset

Create Data Folder

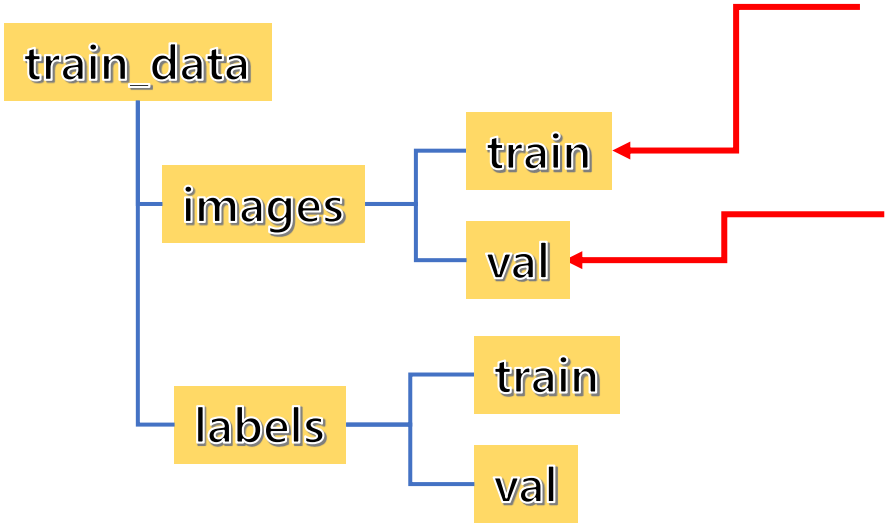




Data preparation : Make Your Own Dataset

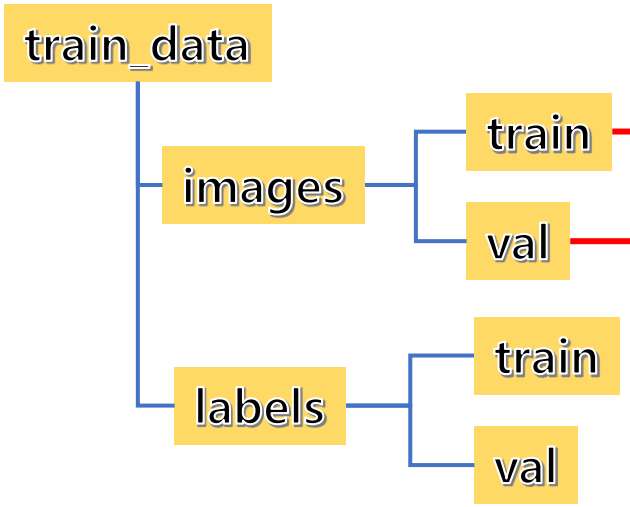
숭실대학교 Soongsil University

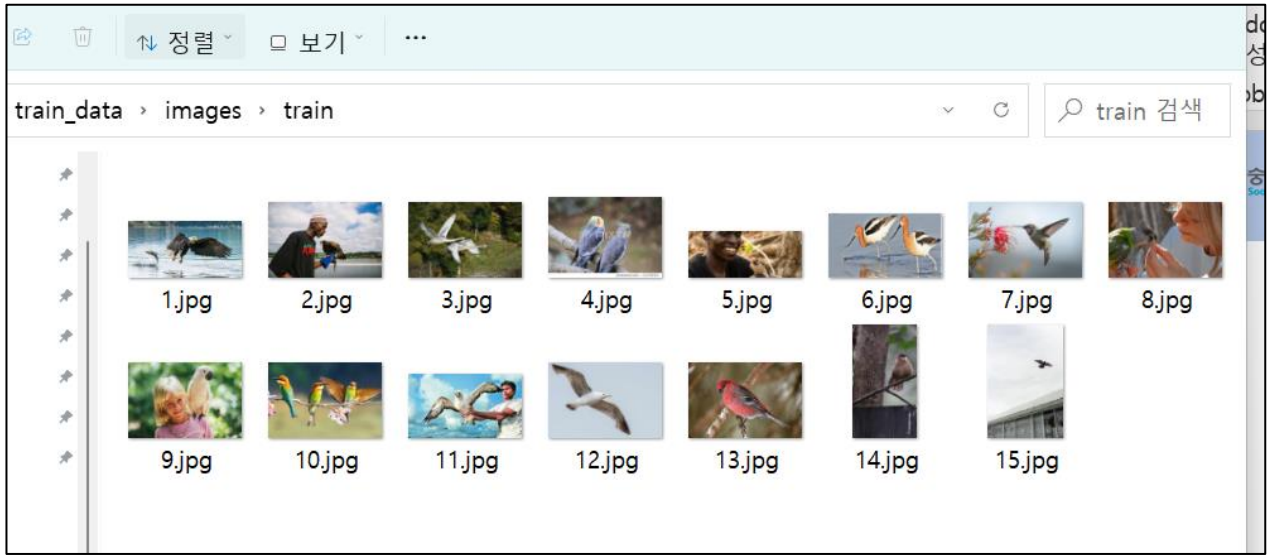
Get Image Data

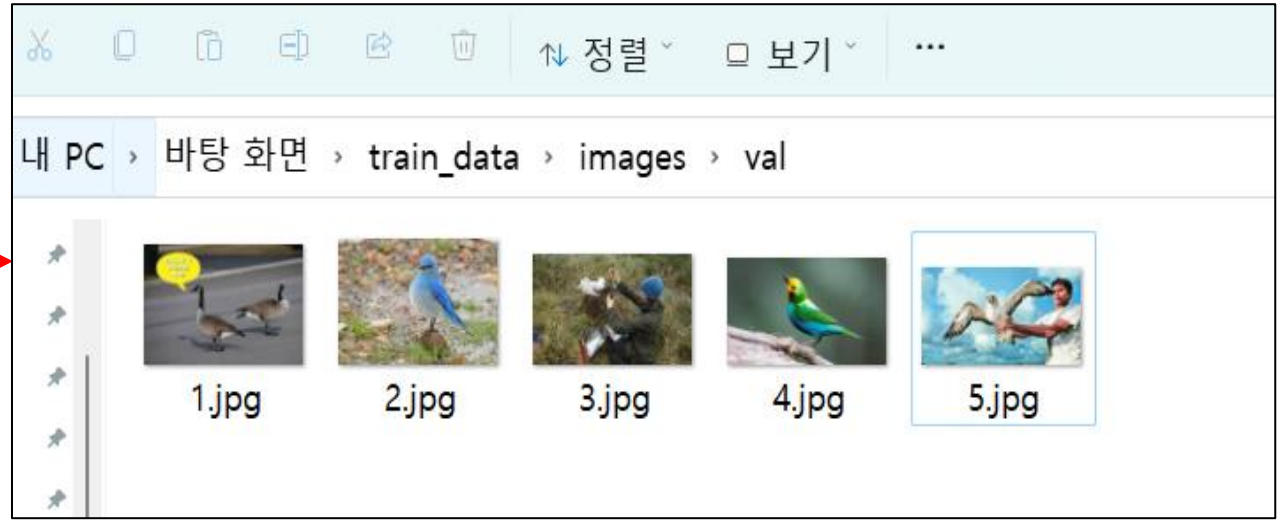


Data preparation : Make Your Own Dataset

Get Image Data







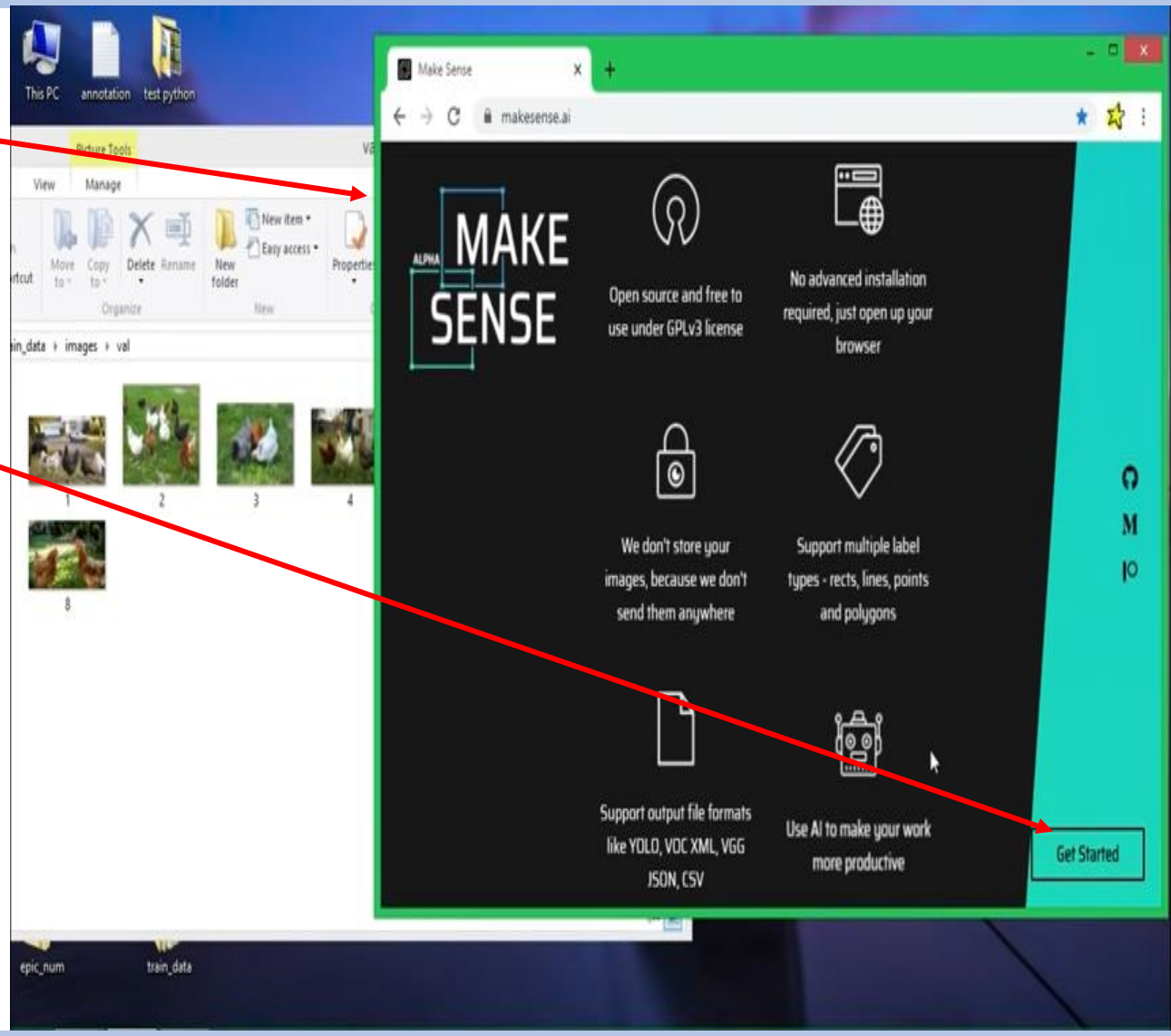
Data preparation : Make Your Own Dataset

숭실대학교 Soongsil University

Go to makesense.ai **for data labeling**

↓

Get Started



https://www.makesense.ai/

Data preparation : Make Your Own Dataset

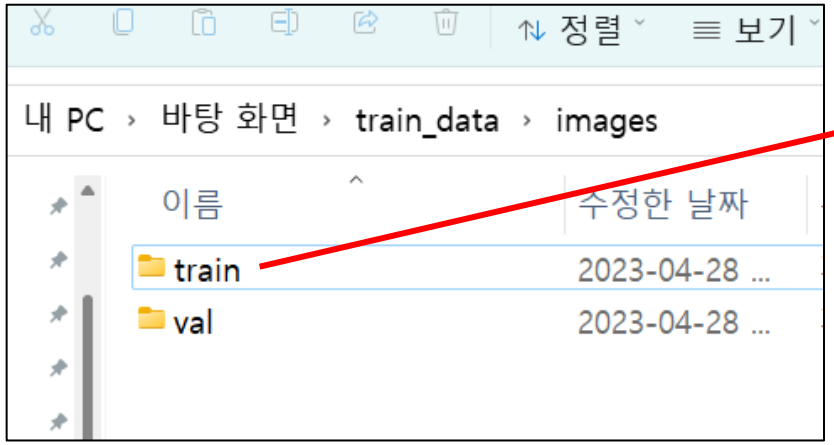
숭실대학교 Soongsil University

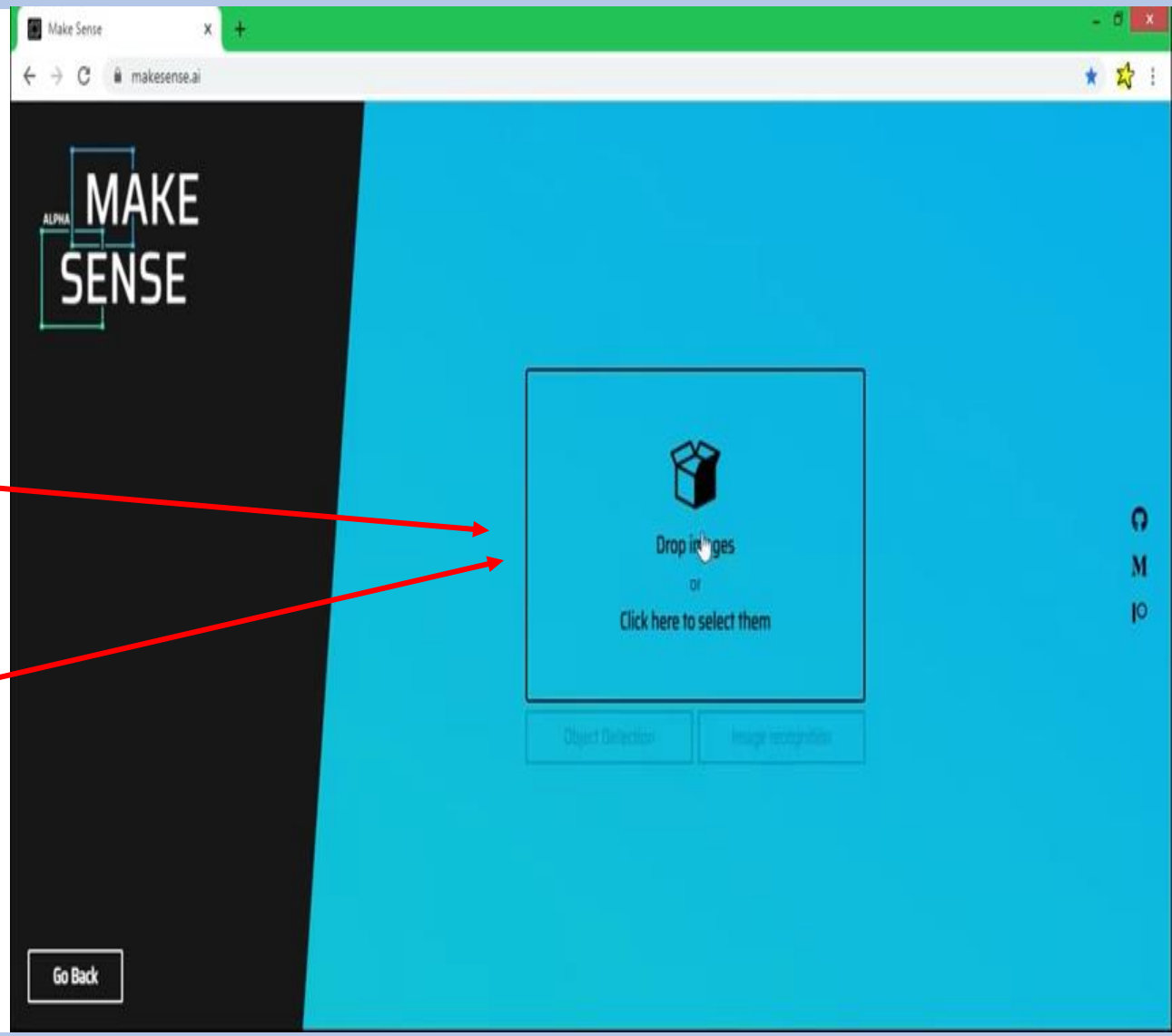
Go to makesense.ai **for data labeling**

Get Started

Drag Drop Images

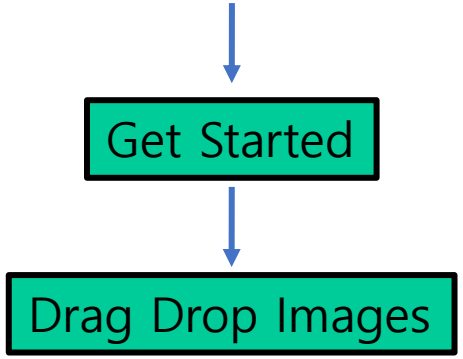
Drag train folder in images folder





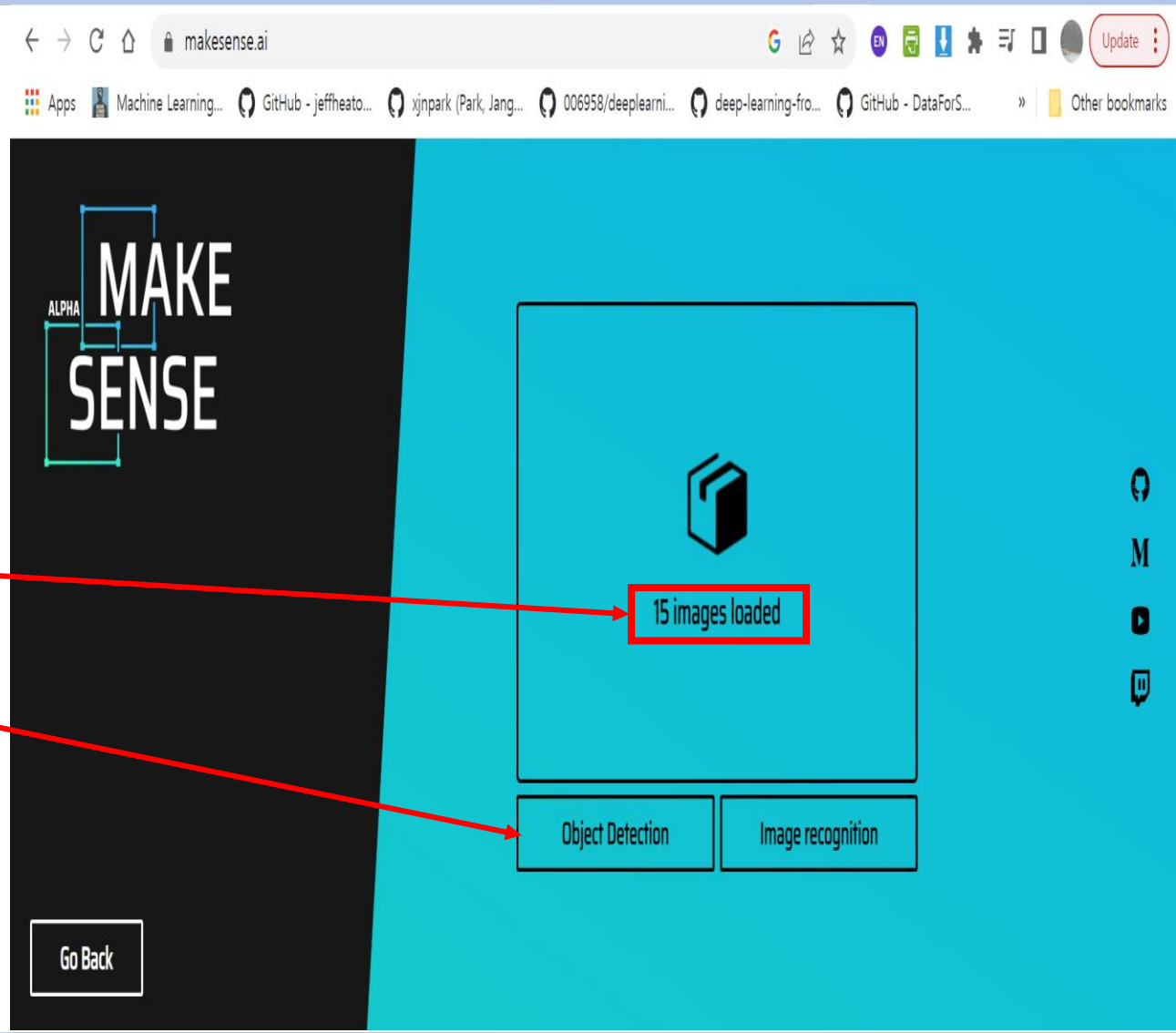
Data preparation : Make Your Own Dataset

Go to makesense.ai **for data labeling**



Drag train folder in images folder

Select : Object Detection



Data preparation : Make Your Own Dataset

Go to makesense.ai **for data labeling**

Get Started

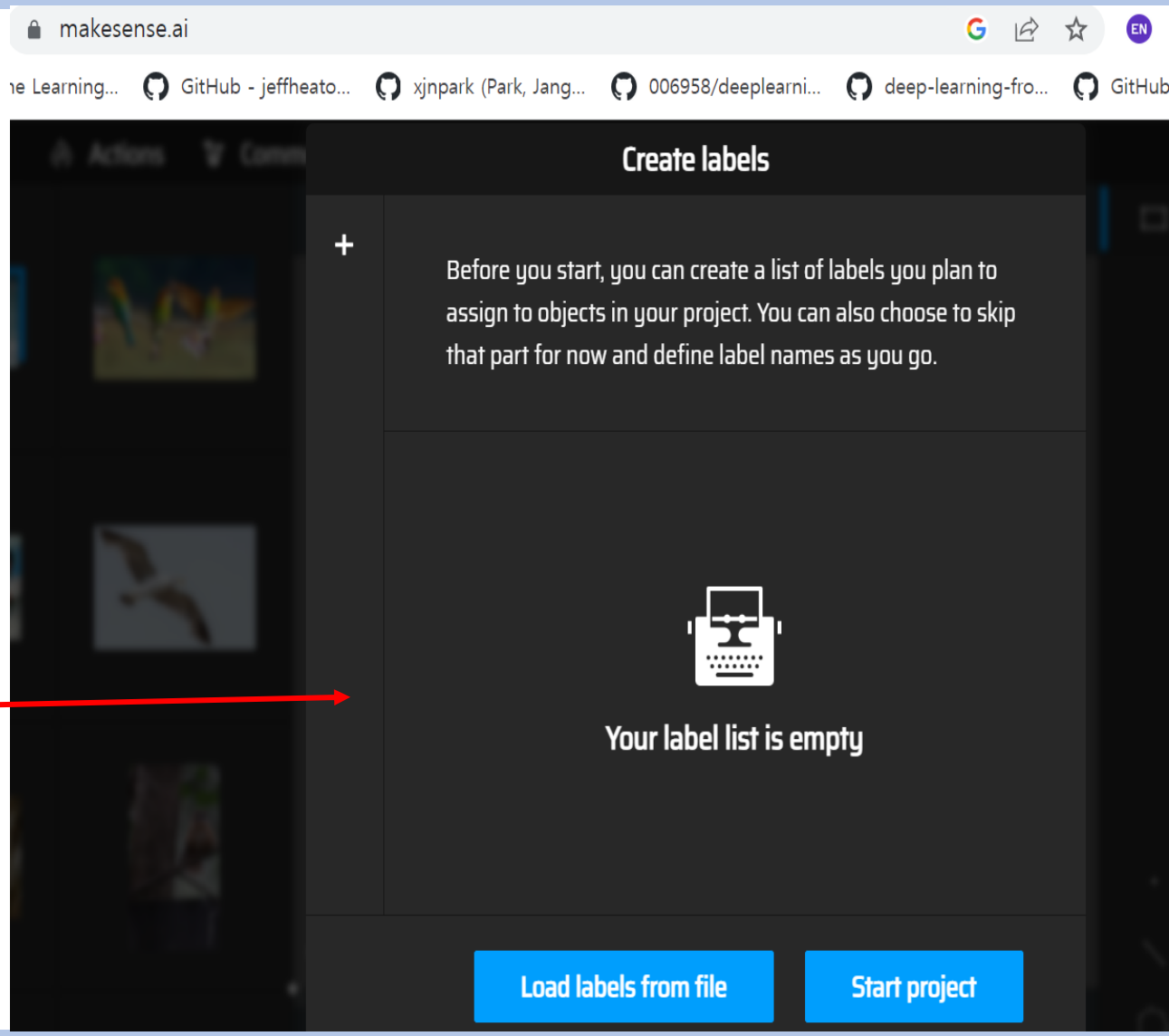
Drag Drop Images

Drag train folder in images/train

Create Labels

Click + or enter, class1 name : bird

Click + or enter, class2 name : person



Data preparation : Make Your Own Dataset

Go to makesense.ai **for data labeling**

↓

Get Started

↓

Drag Drop Images

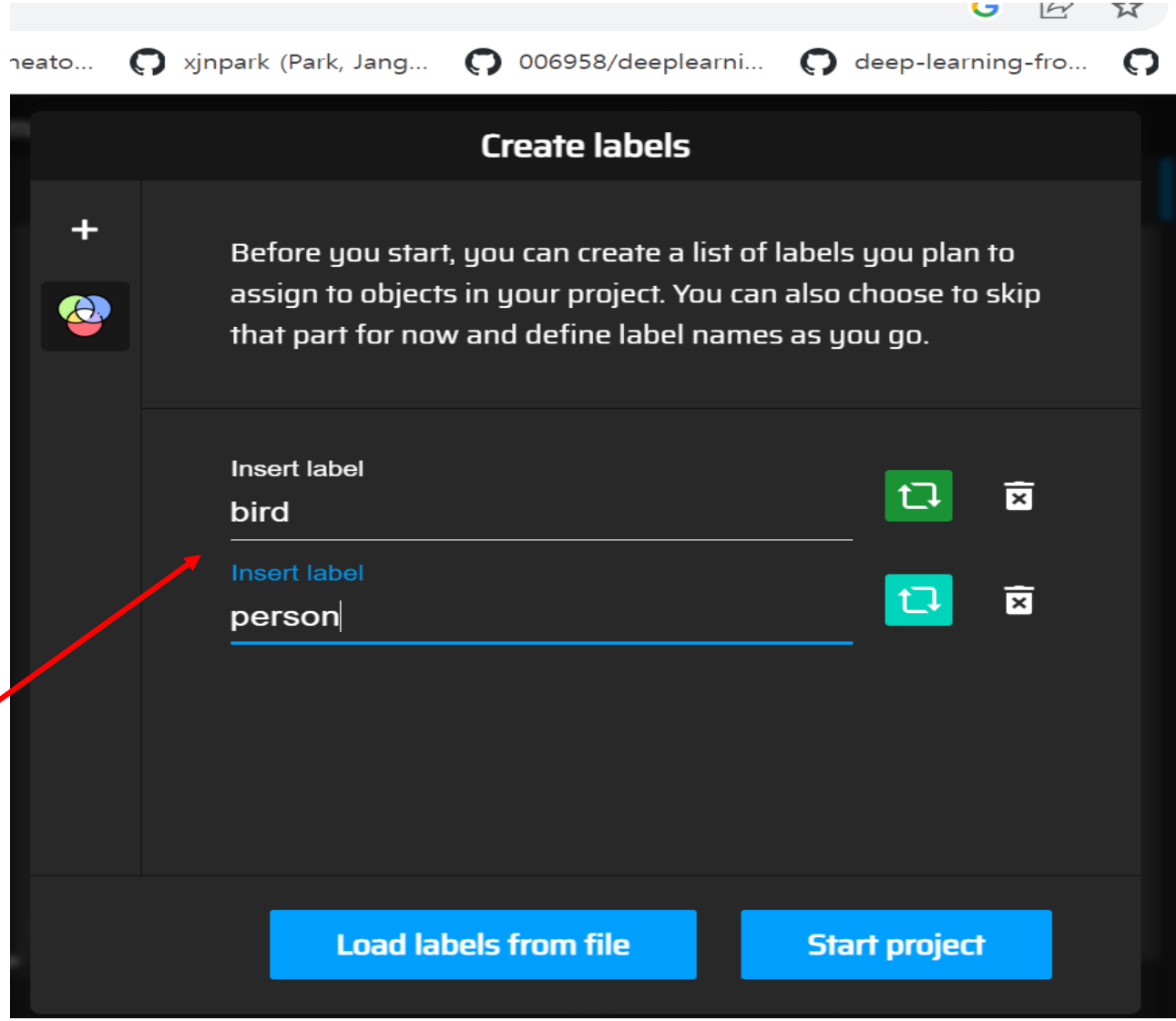
Drag train folder in images/train

↓

Create Labels

Click + or enter, class1 name : bird

Click + or enter, class2 name : person



Data preparation : Make Your Own Dataset

Go to makesense.ai **for data labeling**

↓

Get Started

↓

Drag Drop Images

Drag train folder in images/train

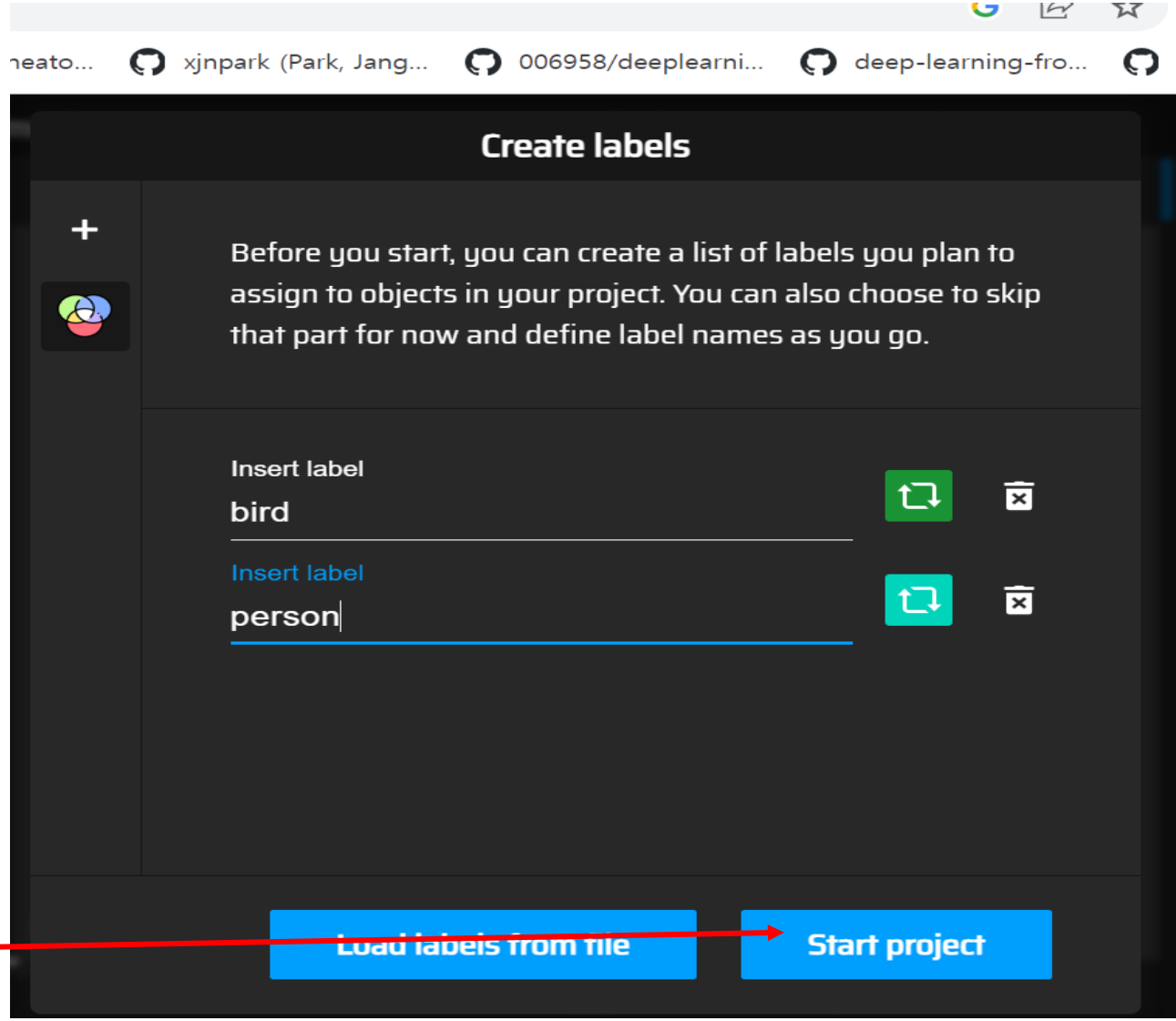
↓

Create Labels

Click + or enter, class1 name : bird

Click + or enter, class2 name : person

Start project



Data preparation : Make Your Own Dataset

숭실대학교 Soongsil University

Apps | Machine Learning... | GitHub - jeffheato... | xjnpark (Park, Jang... | 006958/deeplearni... | deep-learning-fro... | GitHub - DataForS... | Othe

Make Sense | Actions | Community | Project Name: my-project-name

Images








1.jpg

Rect

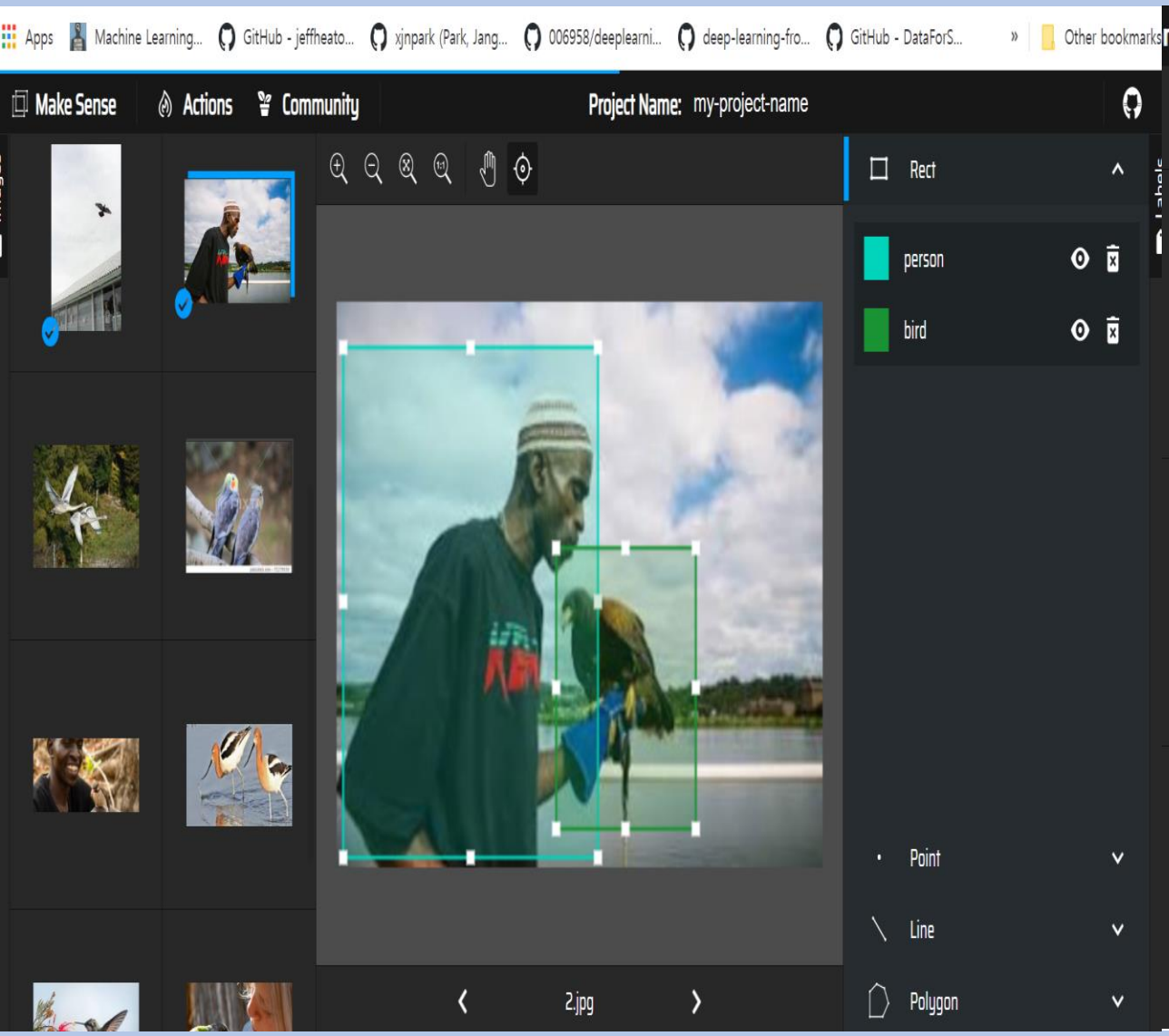
draw your first bounding box

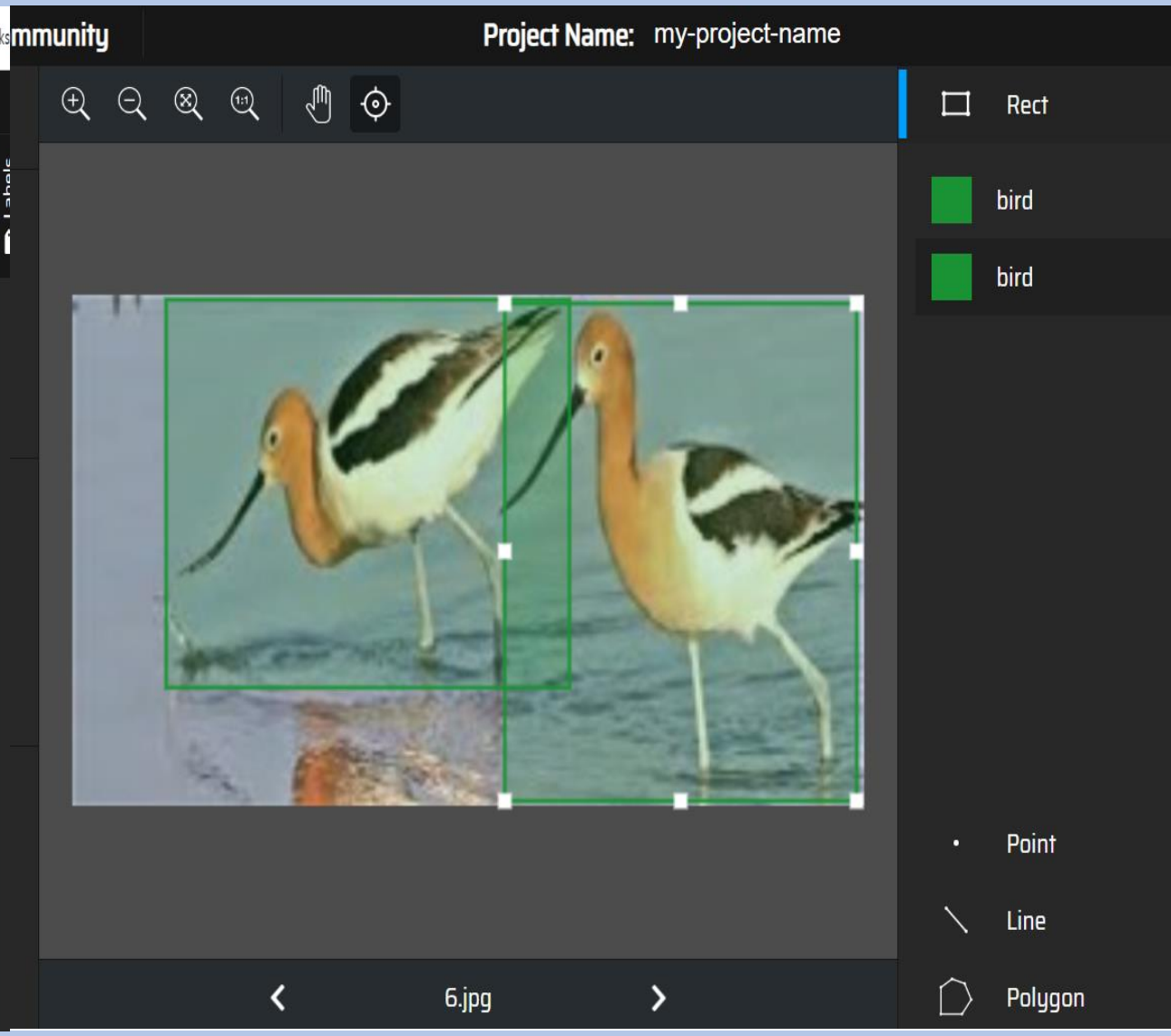
Point

Line

Polygon

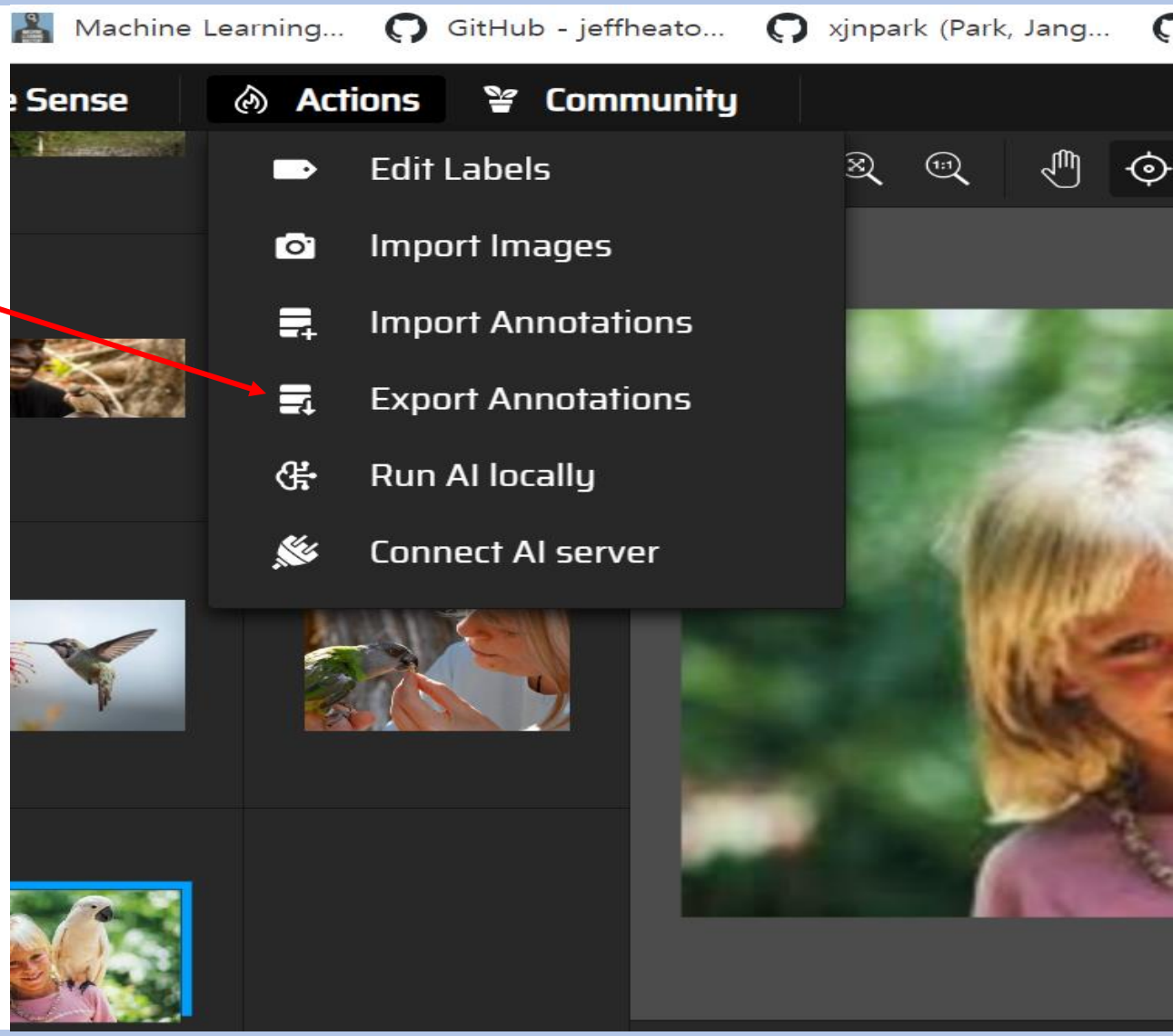
Data preparation : Make Your Own Dataset



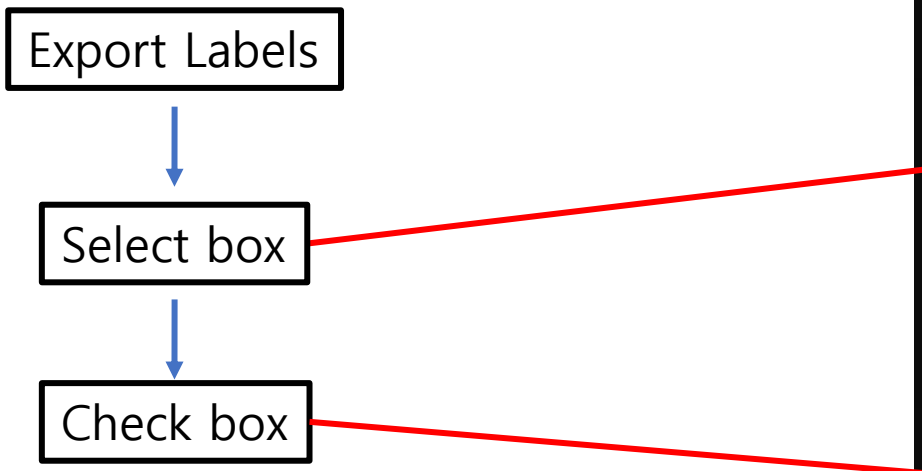


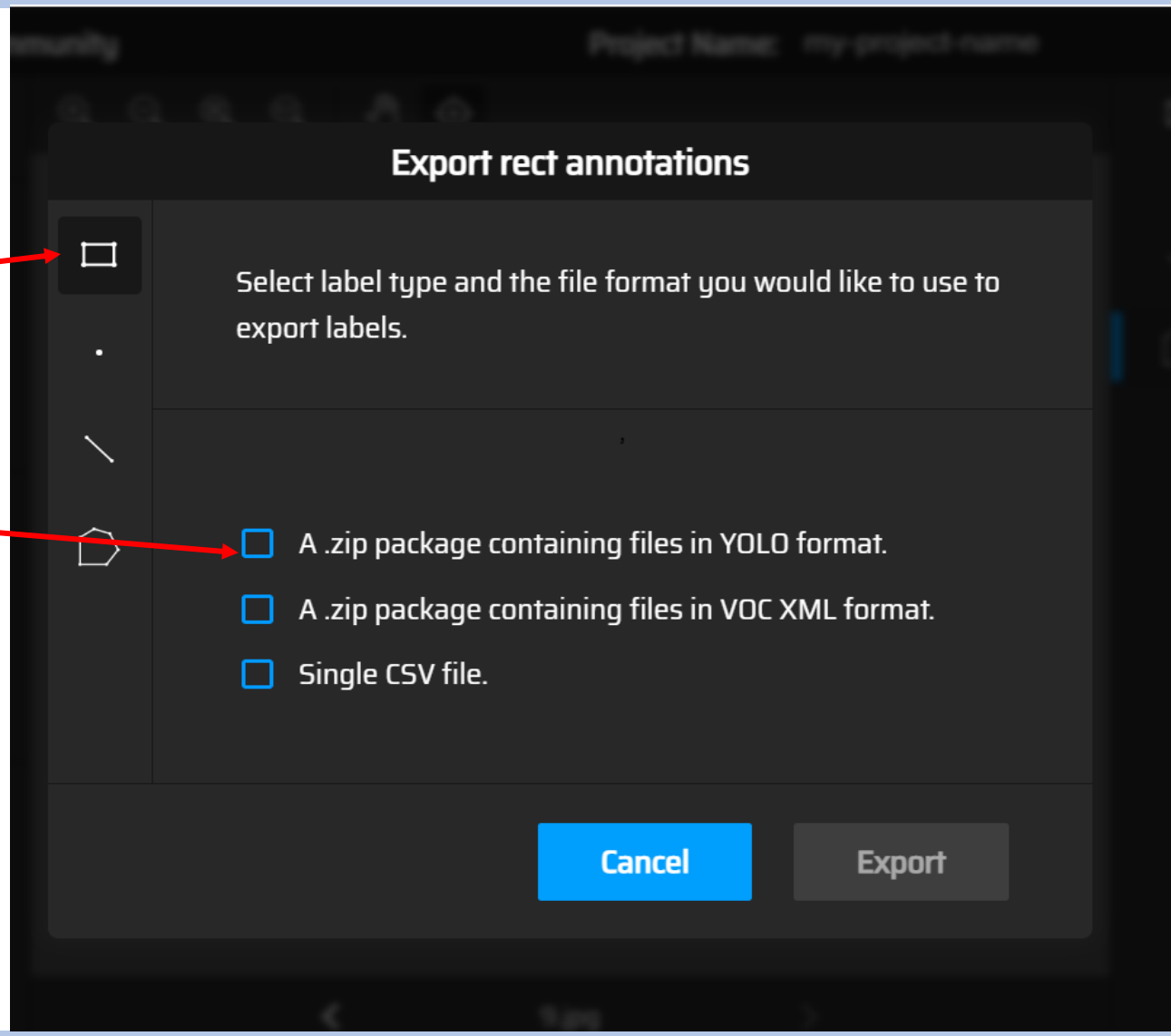
Data preparation : Make Your Own Dataset

Export Labels

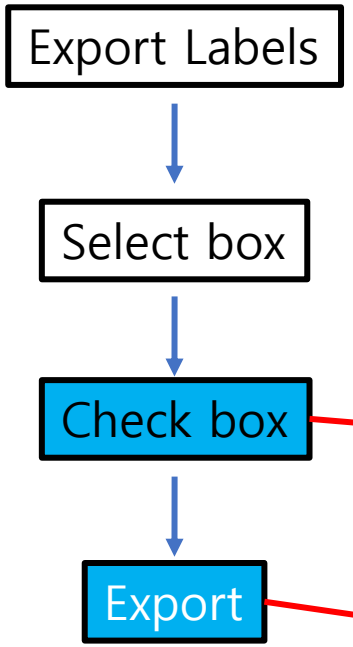


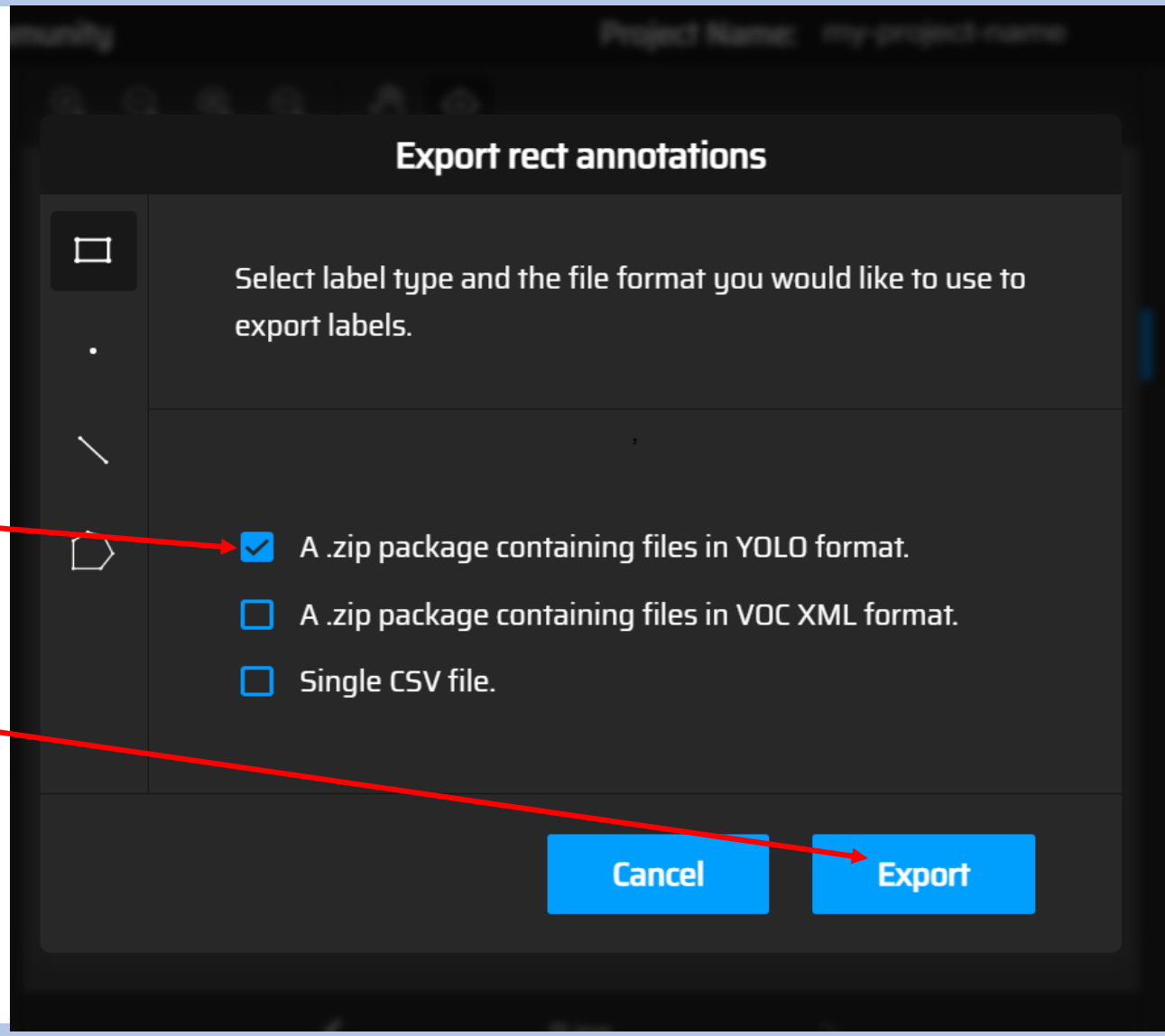
Data preparation : Make Your Own Dataset



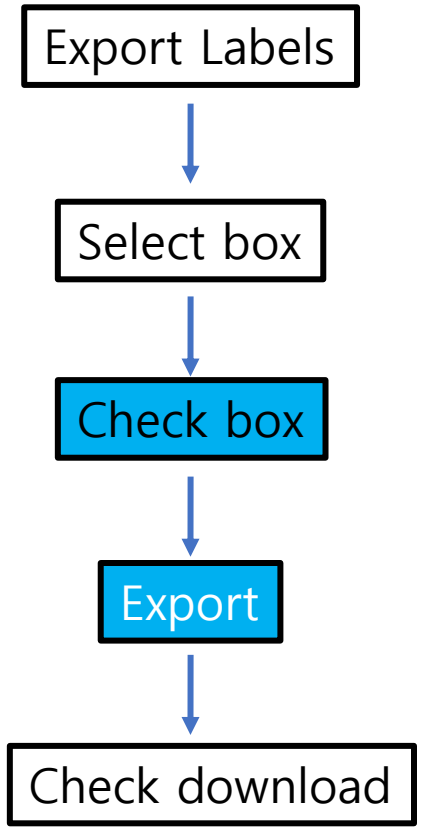


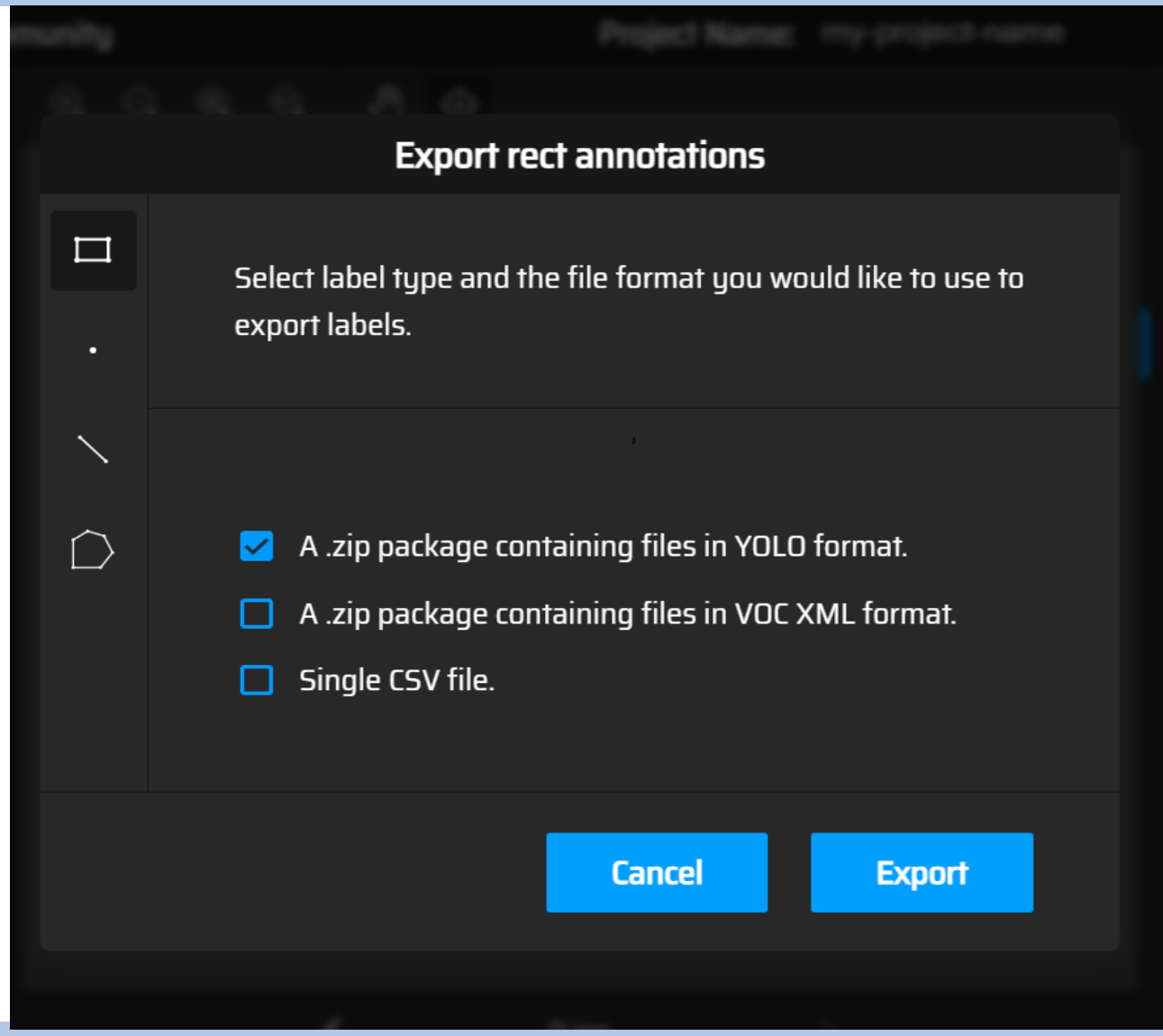
Data preparation : Make Your Own Dataset



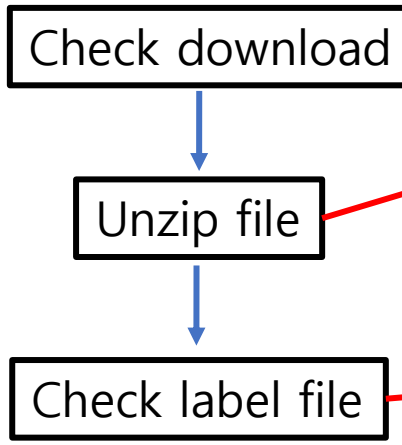


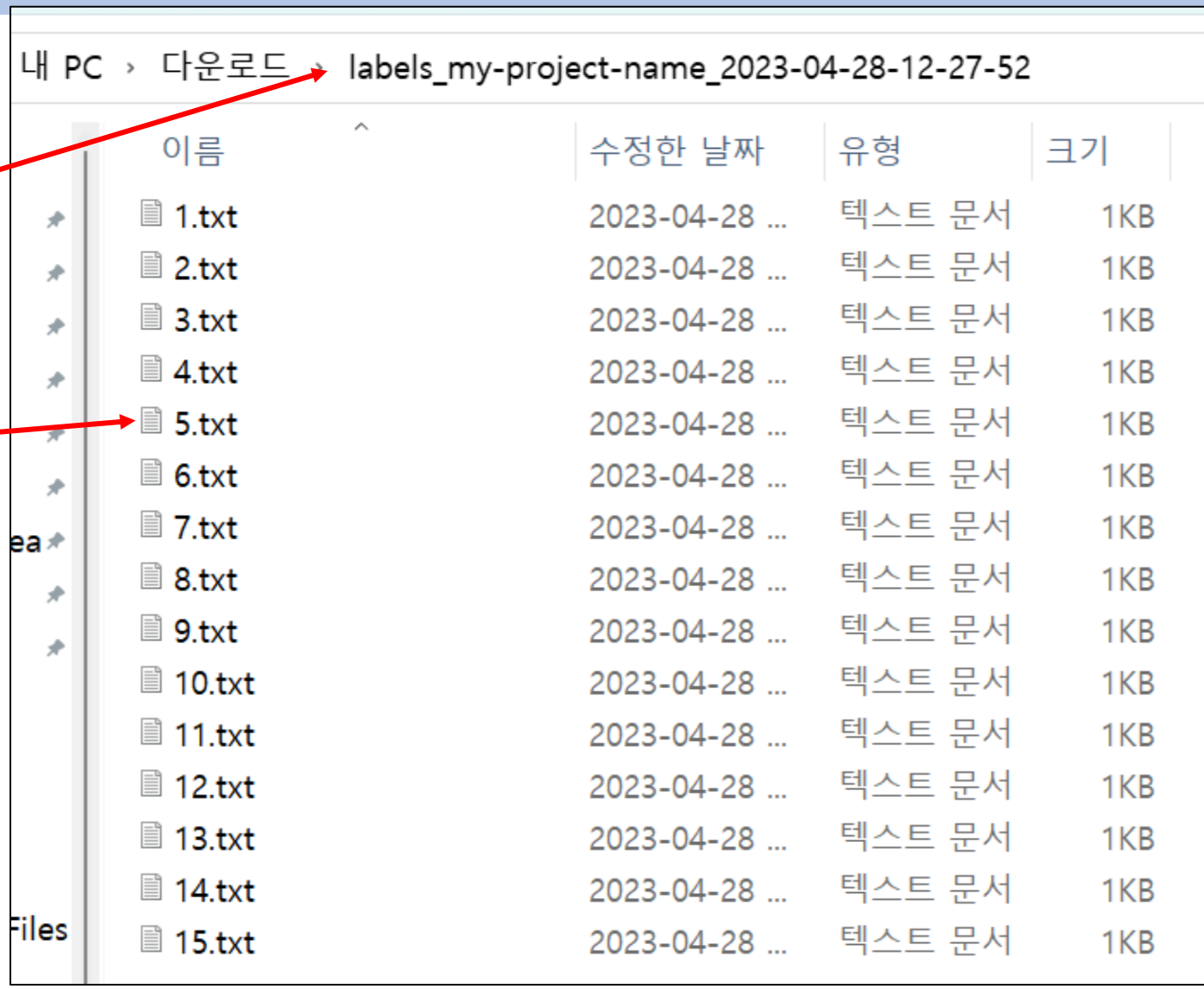
Data preparation : Make Your Own Dataset





Data preparation : Make Your Own Dataset





Data preparation : Make Your Own Dataset

Check download

Unzip file

Check label file

5.txt

편집 보기

```
1 0.331030 0.500044 0.429769 0.999912
0 0.648641 0.726765 0.264151 0.535989
```

내 PC › 다운로드 › labels_my-project-name_2023-04-28-12-27-52

이름	수정한 날짜	유형	크기
1.txt	2023-04-28 ...	텍스트 문서	1KB
2.txt	2023-04-28 ...	텍스트 문서	1KB
3.txt	2023-04-28 ...	텍스트 문서	1KB
4.txt	2023-04-28 ...	텍스트 문서	1KB
5.txt	2023-04-28 ...	텍스트 문서	1KB
6.txt	2023-04-28 ...	텍스트 문서	1KB
7.txt	2023-04-28 ...	텍스트 문서	1KB
8.txt	2023-04-28 ...	텍스트 문서	1KB
9.txt	2023-04-28 ...	텍스트 문서	1KB
10.txt	2023-04-28 ...	텍스트 문서	1KB
11.txt	2023-04-28 ...	텍스트 문서	1KB
12.txt	2023-04-28 ...	텍스트 문서	1KB
13.txt	2023-04-28 ...	텍스트 문서	1KB
14.txt	2023-04-28 ...	텍스트 문서	1KB
15.txt	2023-04-28 ...	텍스트 문서	1KB

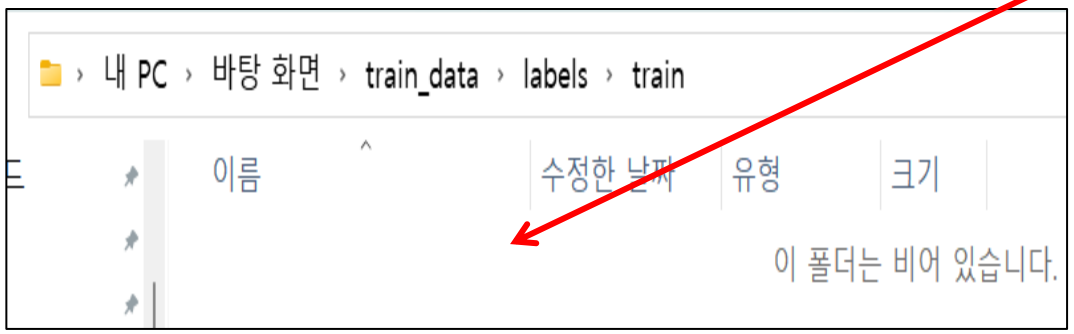
Data preparation : Make Your Own Dataset

Check download

Unzip file

Check label file

Copy all to the label/train folder



내 PC › 다운로드 › labels_my-project-name_2023-04-28-12-27-52

이름	수정한 날짜	유형	크기
1.txt	2023-04-28 ...	텍스트 문서	1KB
2.txt	2023-04-28 ...	텍스트 문서	1KB
3.txt	2023-04-28 ...	텍스트 문서	1KB
4.txt	2023-04-28 ...	텍스트 문서	1KB
5.txt	2023-04-28 ...	텍스트 문서	1KB
6.txt	2023-04-28 ...	텍스트 문서	1KB
7.txt	2023-04-28 ...	텍스트 문서	1KB
8.txt	2023-04-28 ...	텍스트 문서	1KB
9.txt	2023-04-28 ...	텍스트 문서	1KB
10.txt	2023-04-28 ...	텍스트 문서	1KB
11.txt	2023-04-28 ...	텍스트 문서	1KB
12.txt	2023-04-28 ...	텍스트 문서	1KB
13.txt	2023-04-28 ...	텍스트 문서	1KB
14.txt	2023-04-28 ...	텍스트 문서	1KB
15.txt	2023-04-28 ...	텍스트 문서	1KB

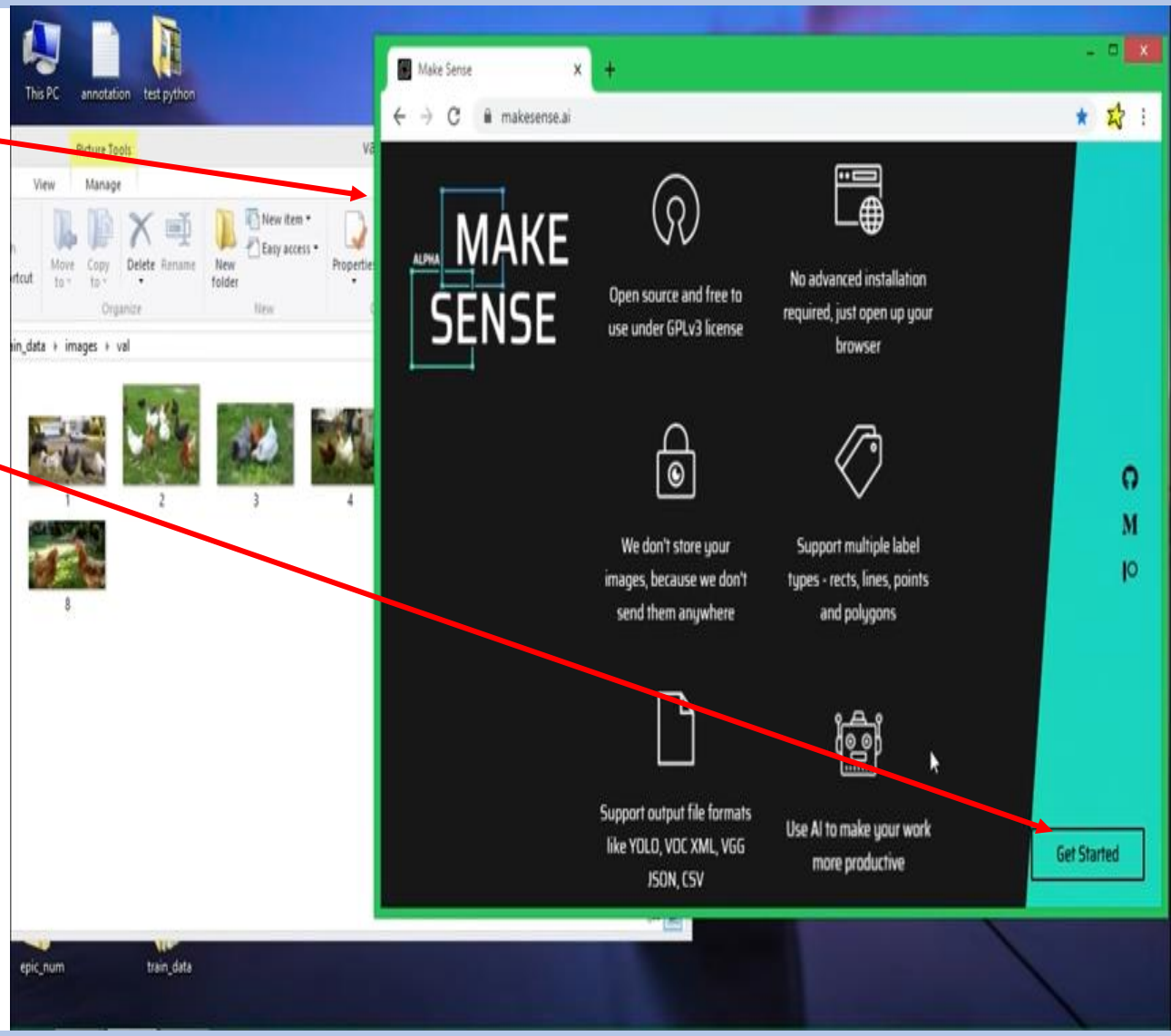
Data preparation : Make Your Own Dataset

Do same proc for val

Go to makesense.ai **for data labeling**

↓

Get Started



https://github.com/SkalskiP/make-sense

Data preparation : Make Your Own Dataset

Go to makesense.ai **for data labeling**

↓

Get Started

↓

Drag Drop Images

Drag val folder in images/train

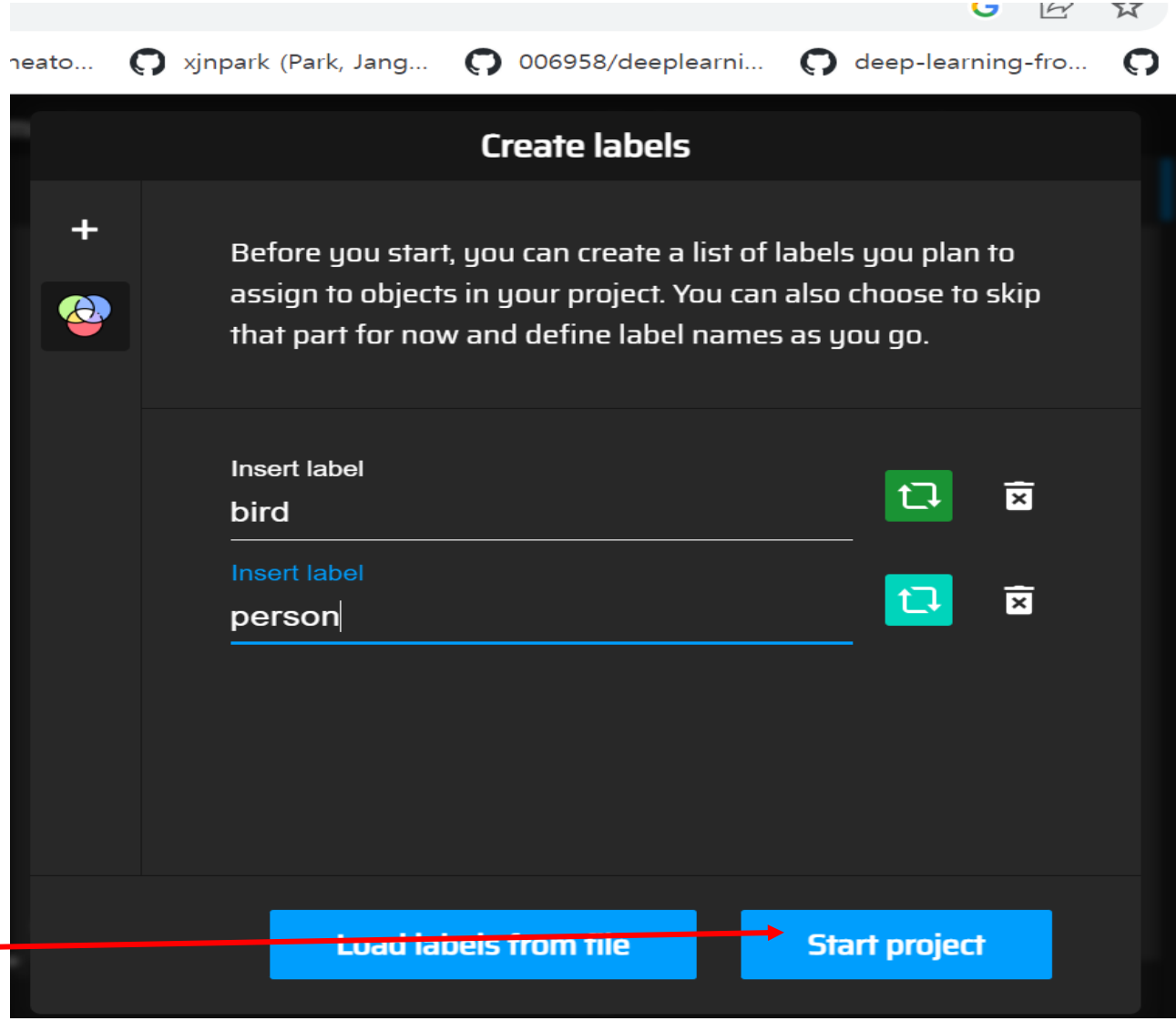
↓

Create Labels

Click + or enter, class1 name : bird

Click + or enter, class2 name : person

Start project



Data preparation : Make Your Own Dataset

Check download

↓

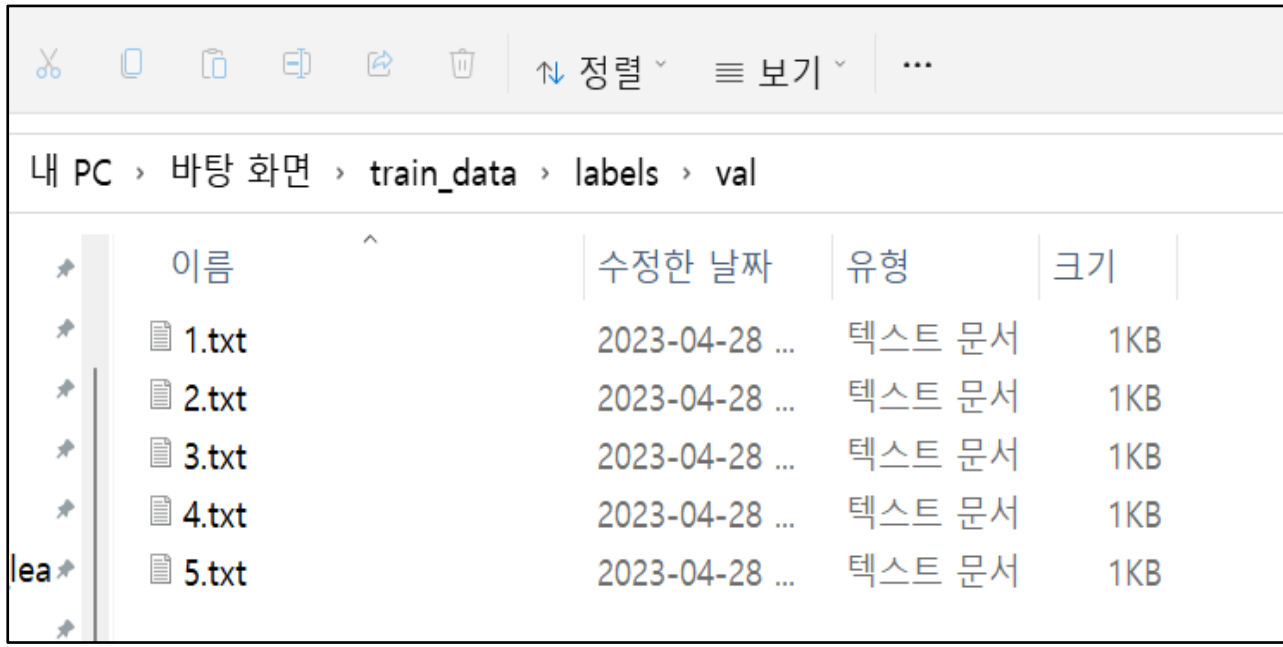
Unzip file

↓

Check label file

↓

Copy all to the label/val folder



Now, we ready for training

Run YOLOv5 in Colab

숭실대학교 Soongsil University

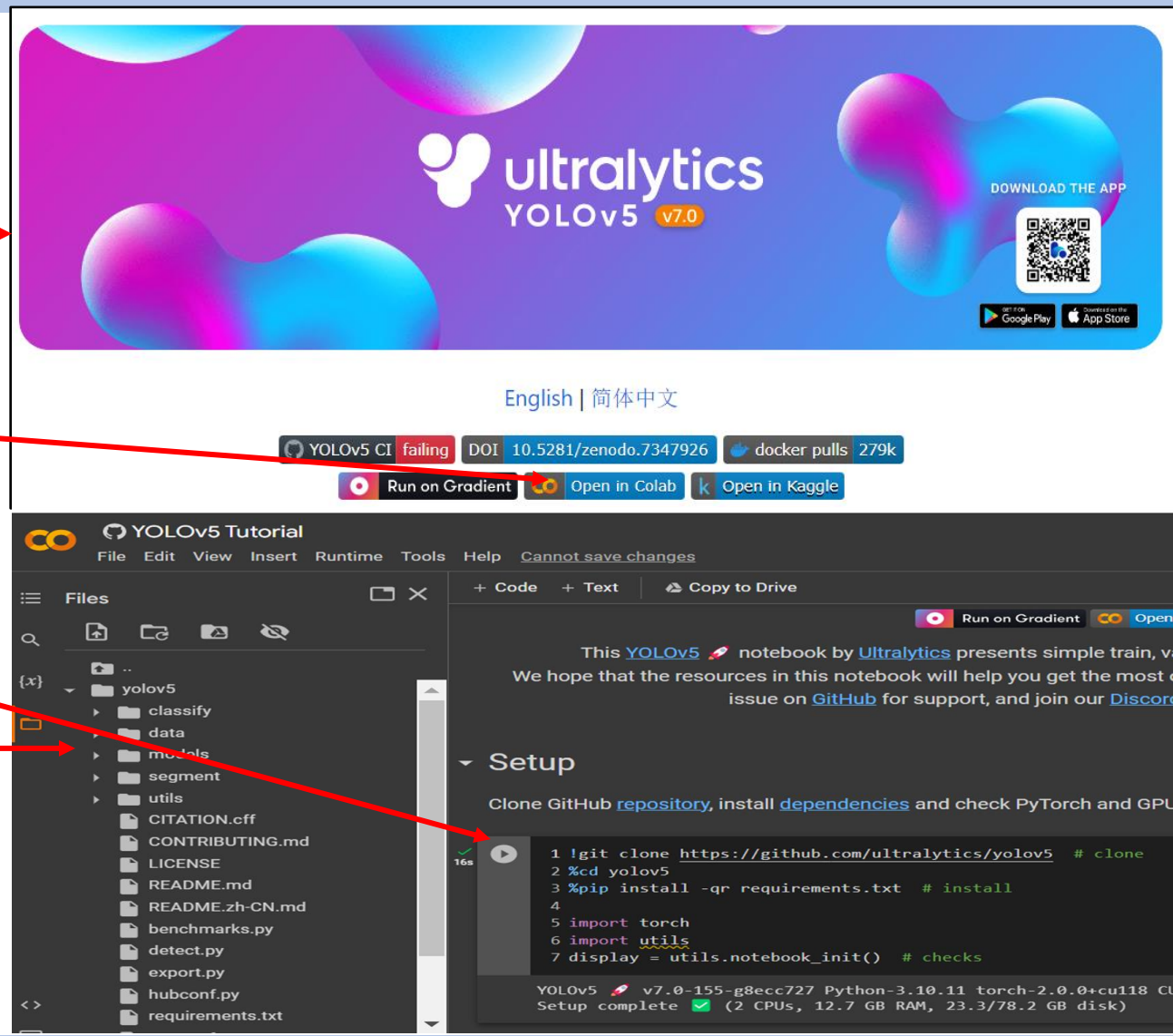
Training

Go to : https://github.com/ultralytics/yolov5

Open in Colab

Run 1st cell

Check folders



Run YOLOv5 in Colab

Training

Go to : https://github.com/ultralytics/yolov5

↓

Open in Colab

↓

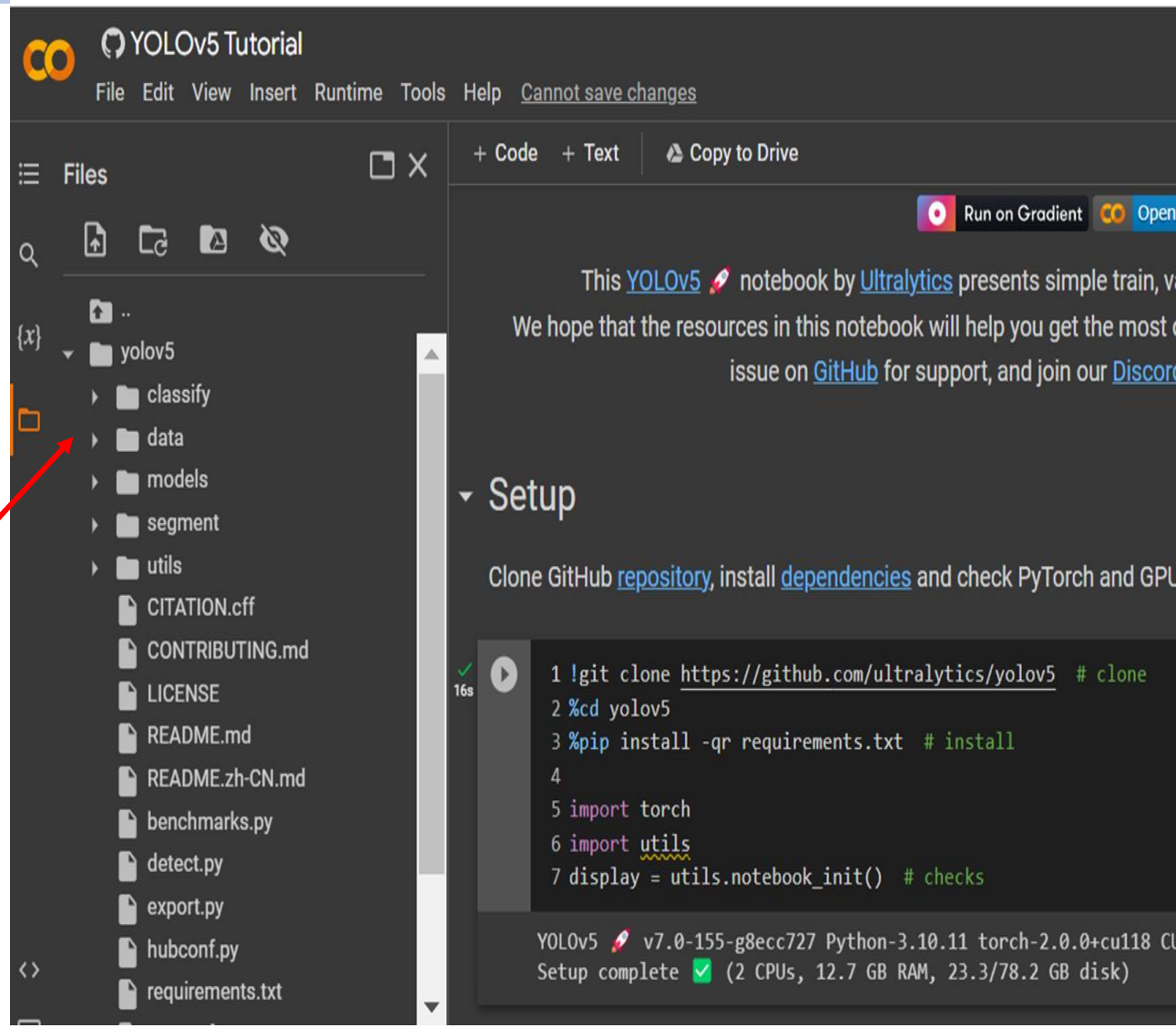
Run 1st cell

↓

Check folders

↓

Data/coco128.yaml



Run YOLOv5 in Colab

Training

Go to : https://github.com/ultralytics/yolov5

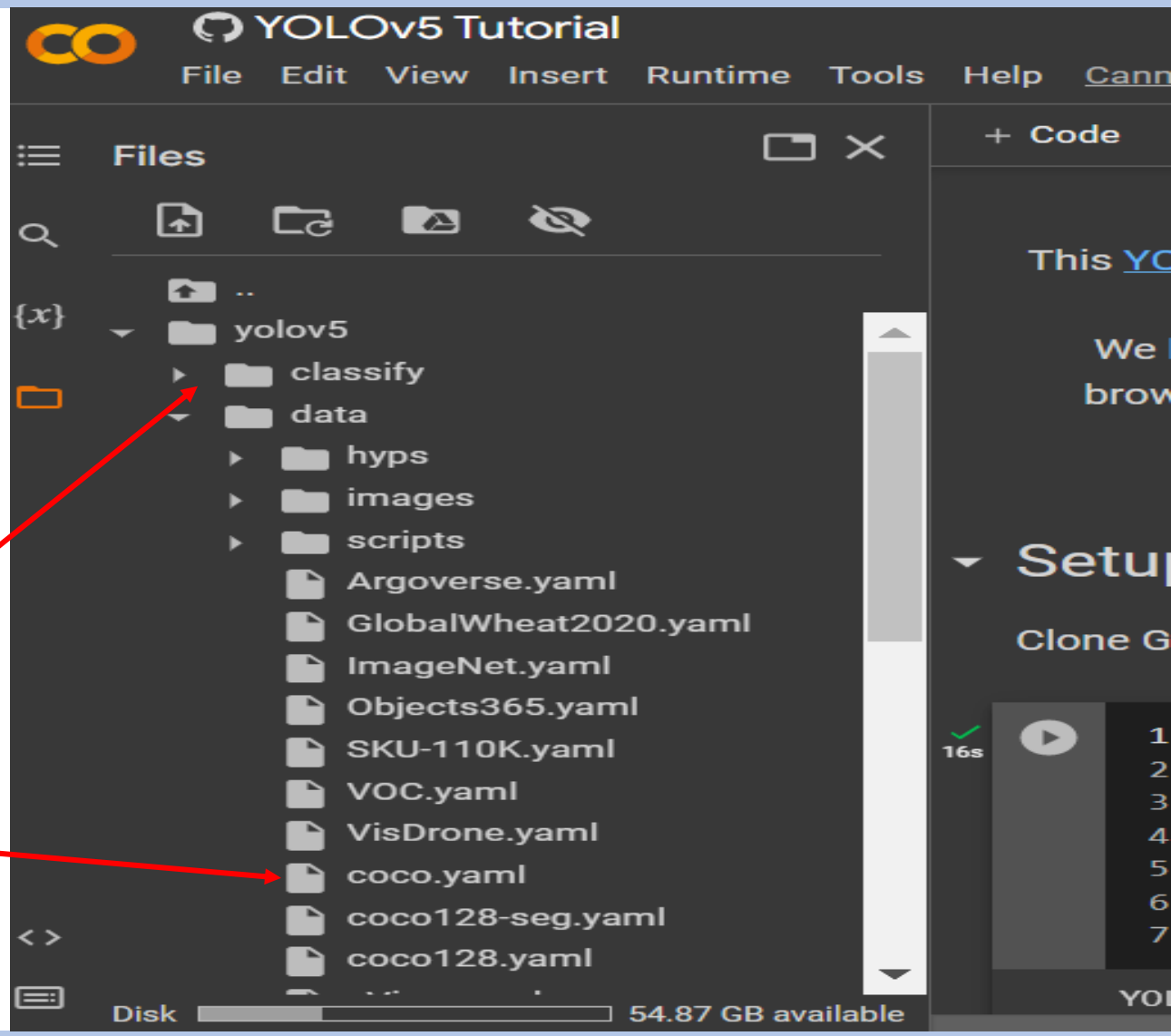
Open in Colab

Run 1st cell

Check folders

data/coco128.yaml

data/coco128.yaml file download



Run YOLOv5 in Colab

Training

Go to : https://github.com/ultralytics/yolov5

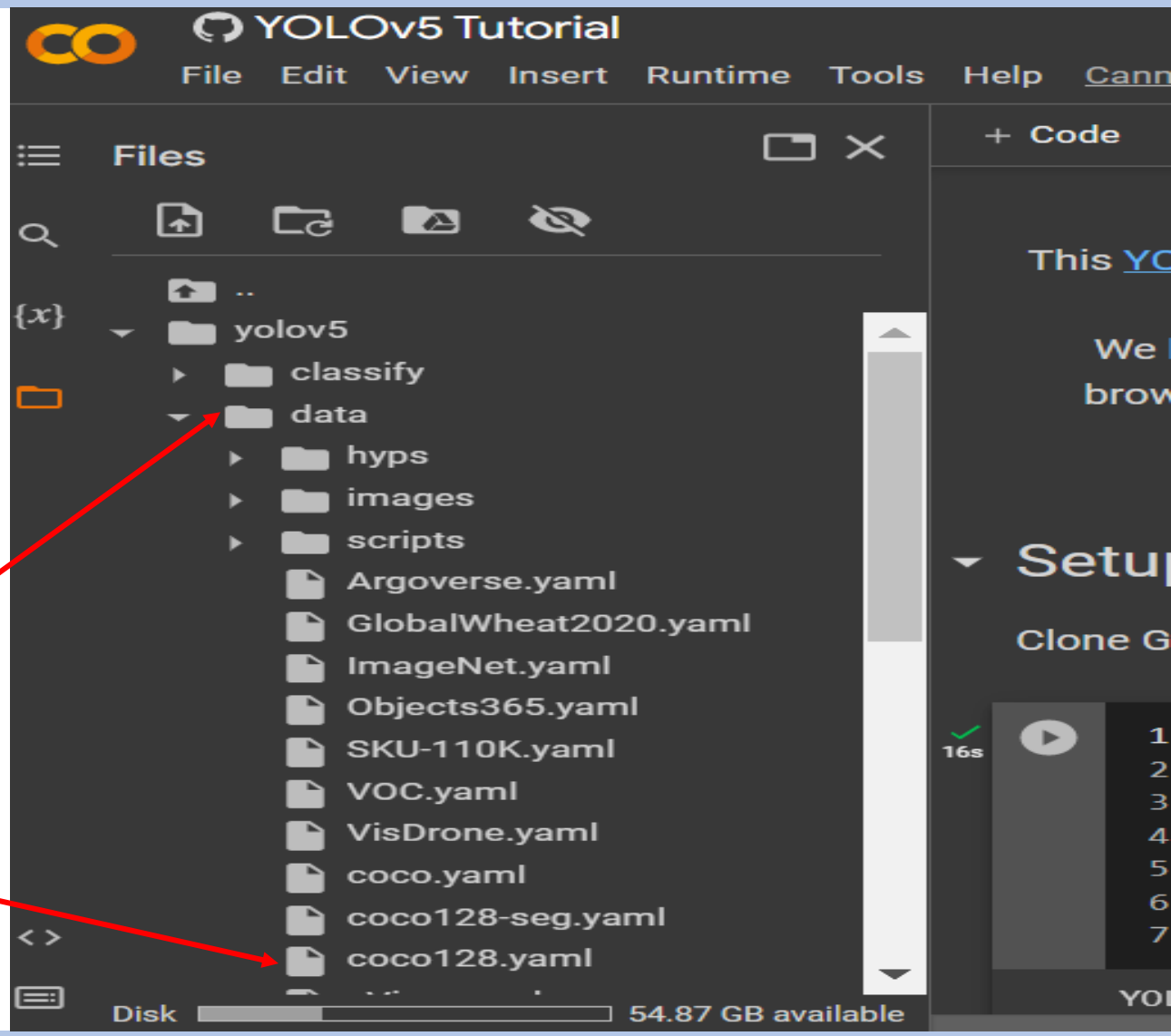
Open in Colab

Run 1st cell

Check folders

data/coco128.yaml

data/coco128.yaml file download



Run YOLOv5 in Colab

Edit coco128.yaml file and save as your own name

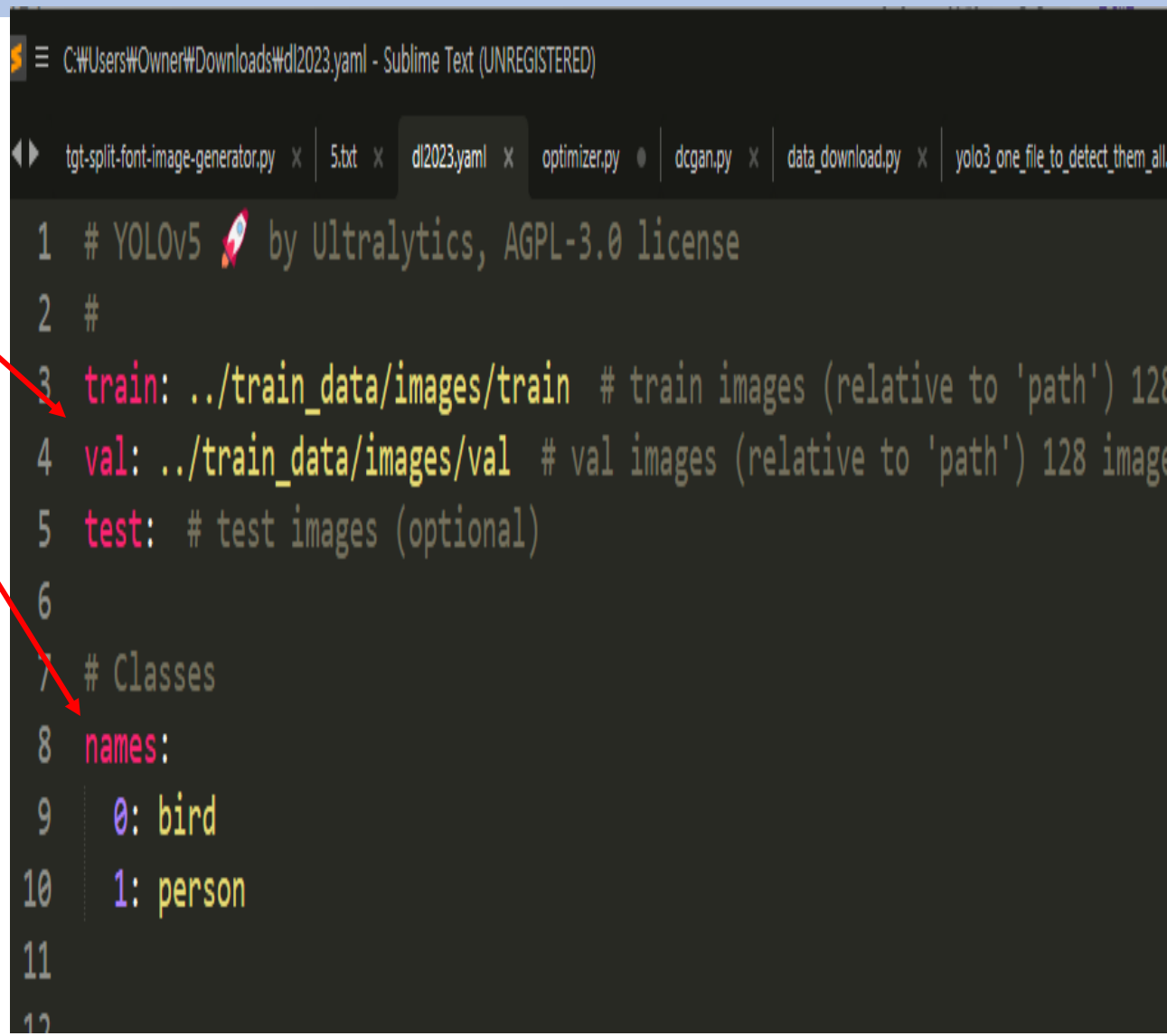
C:₩Users₩Owner₩Downloads₩coco128.yaml - Sublime Text (UNREGISTERED)

tgt-split-font-image-generator.py | 5.txt | coco128.yaml | optimizer.py | dcgan.py | data_download.py

```yaml
# YOLOv5 🚀 by Ultralytics, AGPL-3.0 license
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128
# Example usage: python train.py --data coco128.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco128  ← downloads here (7 MB)


# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/
path: ../datasets/coco128  # dataset root dir
train: images/train2017  # train images (relative to 'path')
val: images/train2017  # val images (relative to 'path') 128
test:  # test images (optional)

# Classes
names:
  0: person
  1: bicycle
  2: car
  3: motorcycle
  4: airplane
  5: bus
  6: train
  7: truck
  8: boat
  9: traffic light
  10: fire hydrant
  11: stop sign
  12: parking meter
  13: bench
```

Run YOLOv5 in Colab

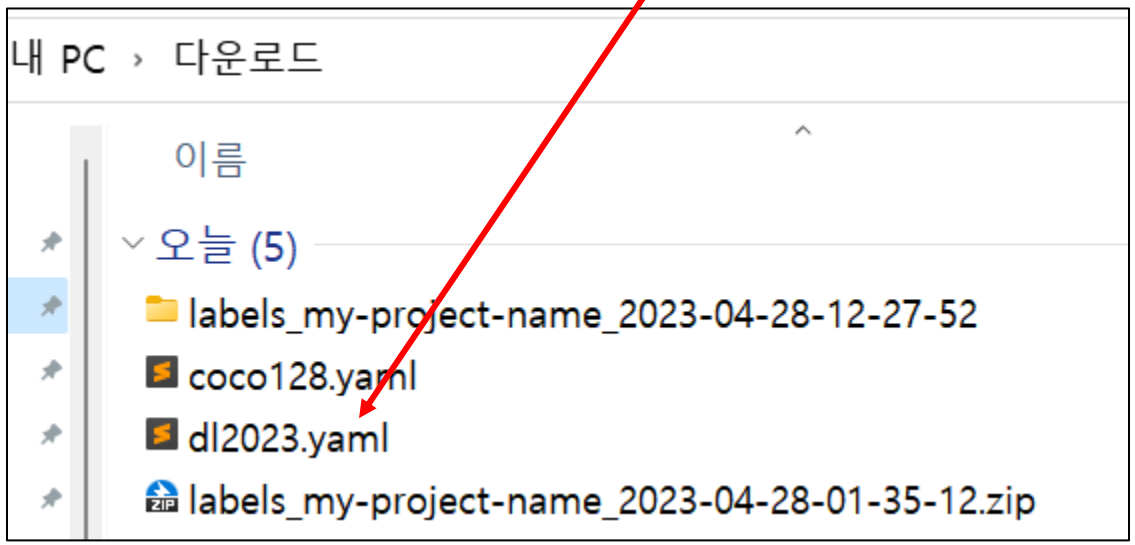
Edit coco128.yaml file and save it as your own name



Run YOLOv5 in Colab

Edit coco128.yaml file and save it as your own name

save it as 'dl2023'



C:₩Users₩Owner₩Downloads₩dl2023.yaml - Sublime Text (UNREGISTERED)

tgt-split-font-image-generator.py | 5.txt | dl2023.yaml | optimizer.py | dcgan.py | data_download.py | yolo3_one_file_to_detect_them_all

```yaml
# YOLOv5 🚀 by Ultralytics, AGPL-3.0 license
#
train: ../train_data/images/train  # train images (relative to 'path') 128
val: ../train_data/images/val  # val images (relative to 'path') 128 image
test:  # test images (optional)

# Classes
names:
  0: bird
  1: person
```

Run YOLOv5 in Colab

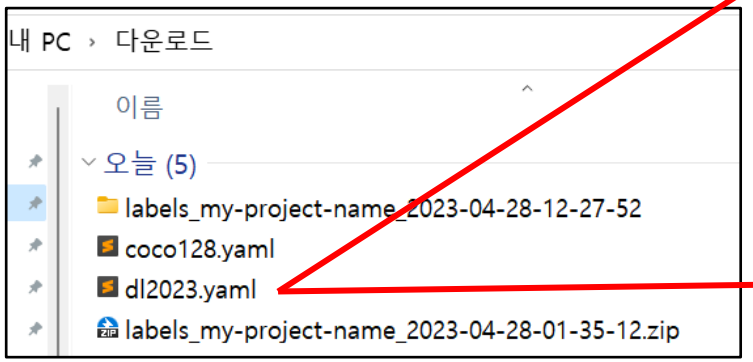
Edit coco128.yaml file and save it as your own name

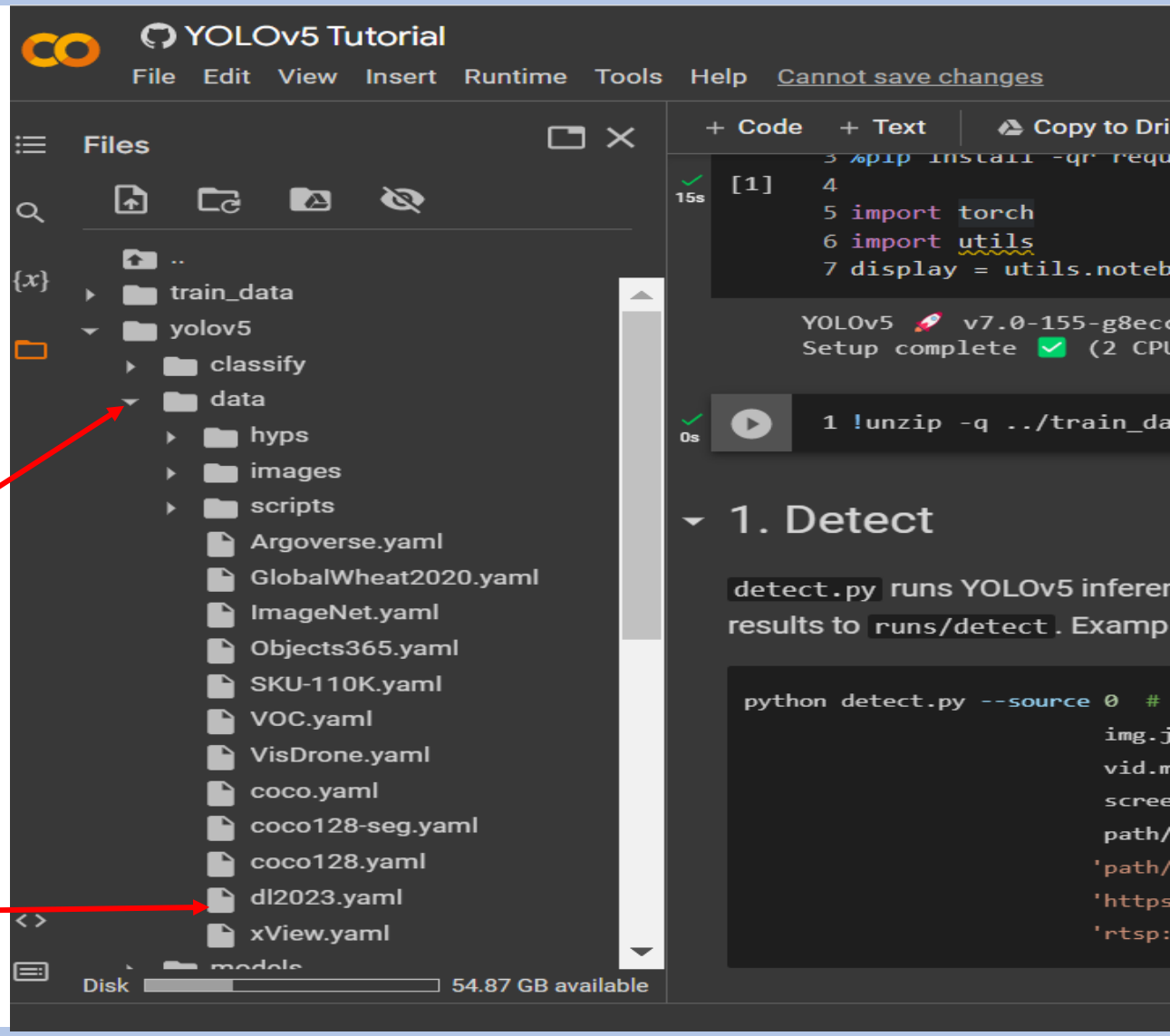
↓

save it as 'dl2023'

↓

Drag and Upload 'dl2023.yaml' At data folder





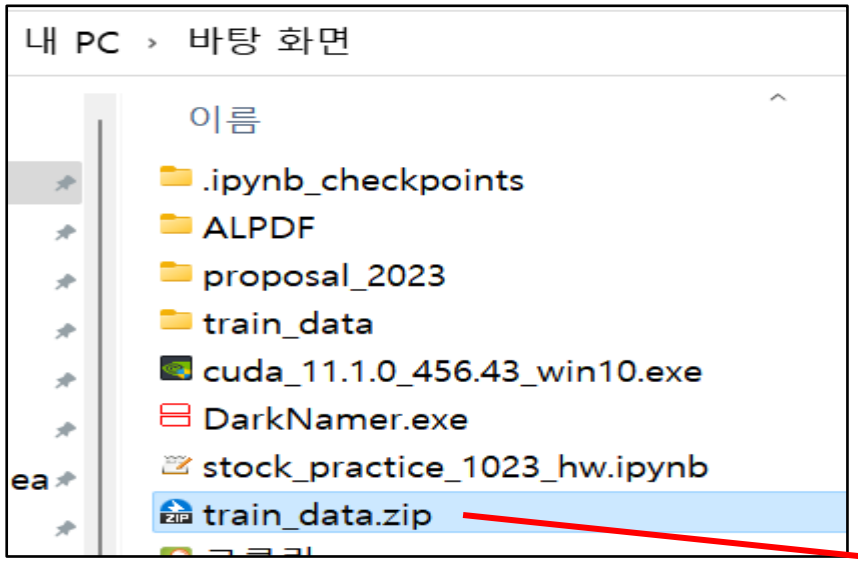
Almost done !!!

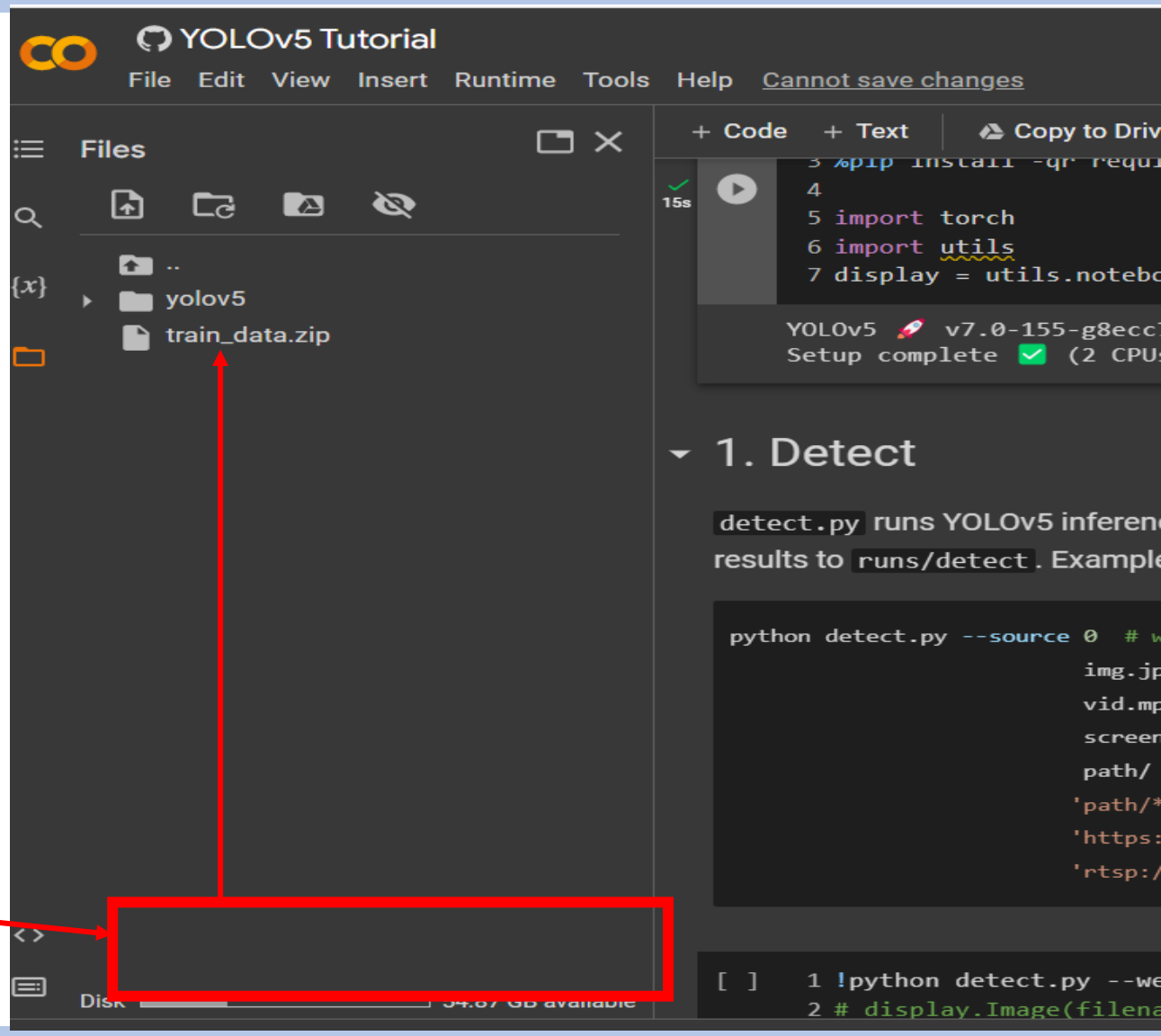
Run YOLOv5 in Colab

Make train_data fold to zip file

↓

Drag and in the Colab folder





Run YOLOv5 in Colab

Make train_data fold to zip file

↓

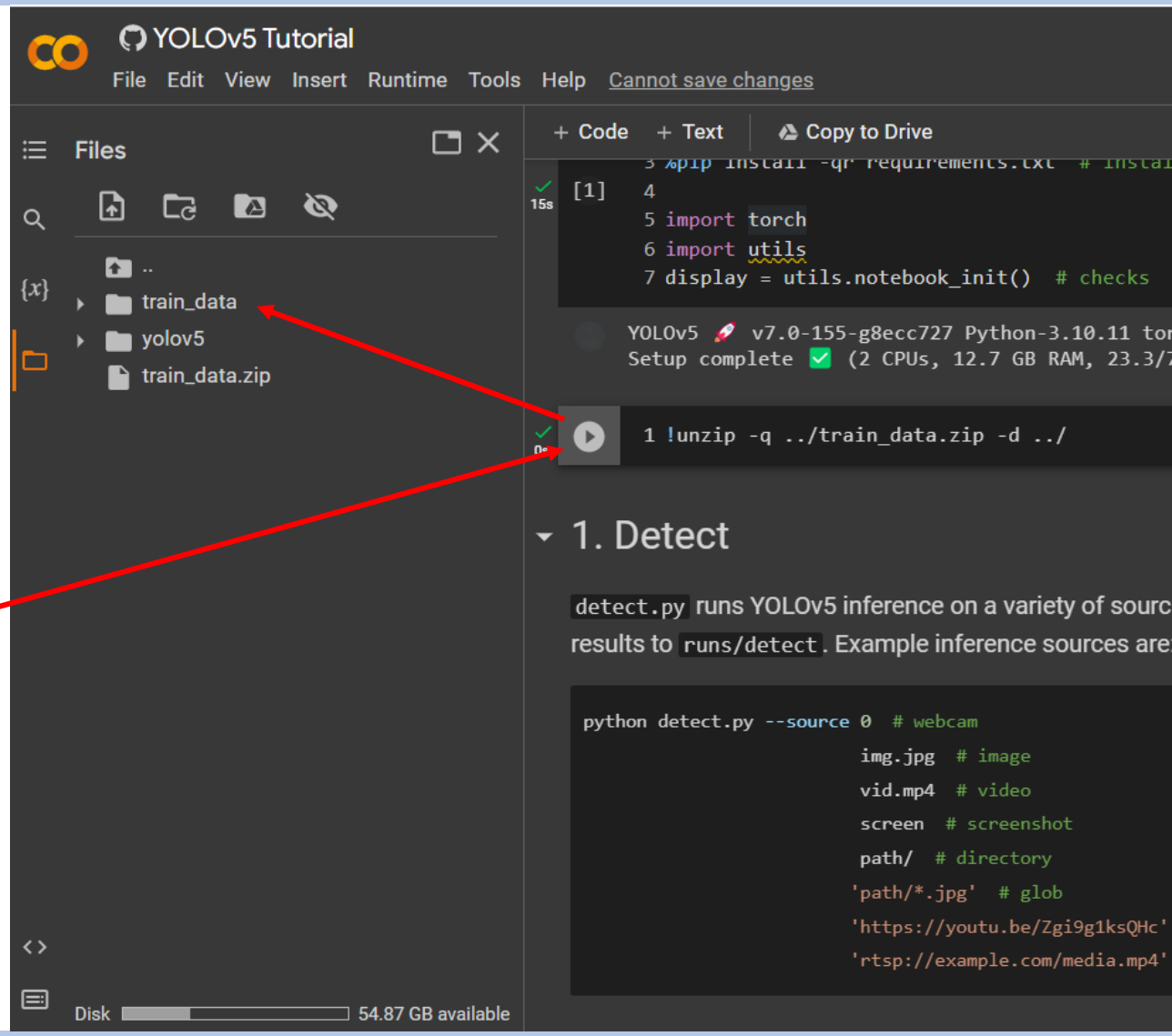
Drag and in the Colab folder

↓

unzip train_data zip file

```
1 !unzip -q ../train_data.zip -d ../
```

0s



Run YOLOv5 in Colab

Make train_data fold to zip file

↓

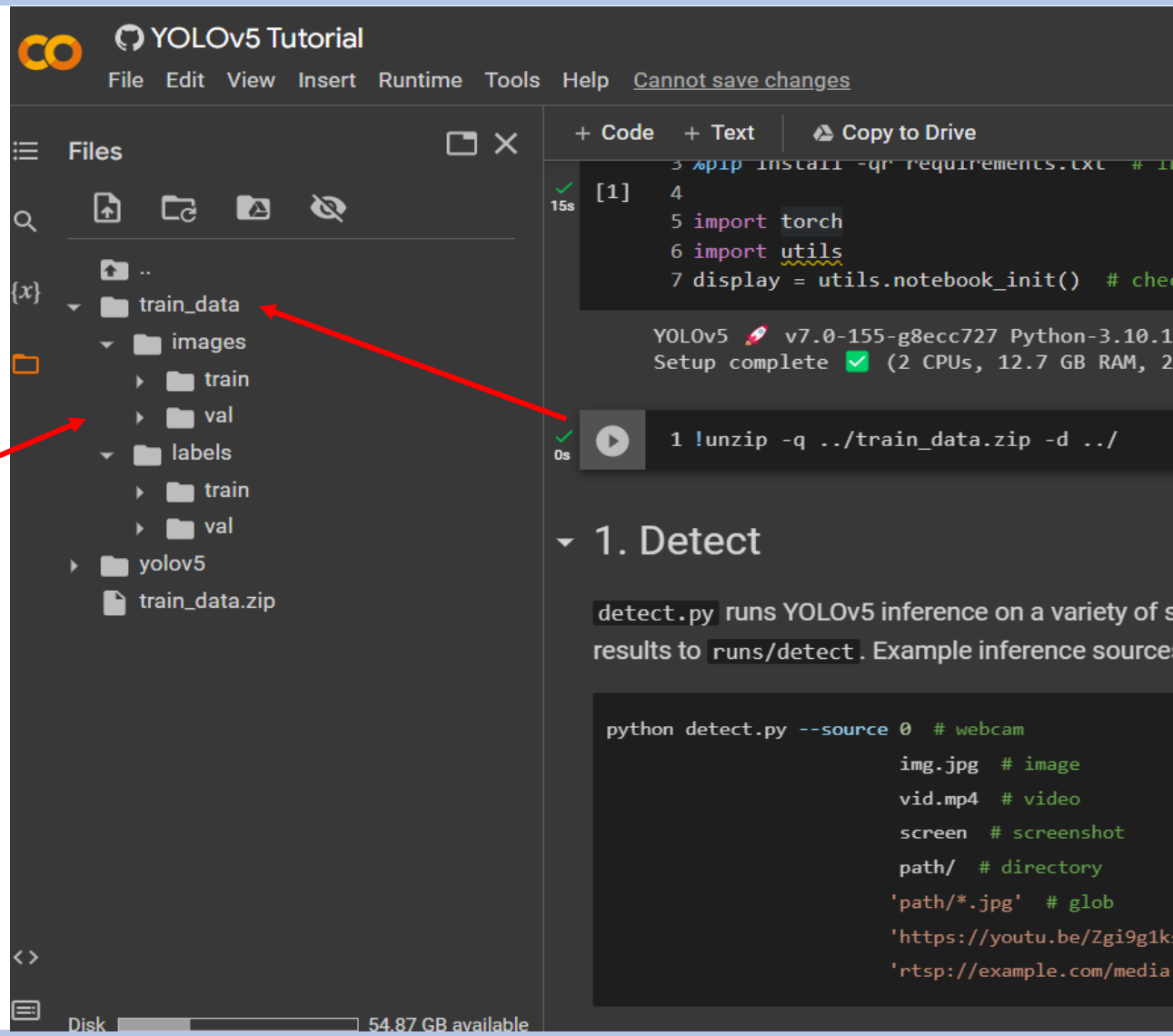
Drag and in the Colab folder

↓

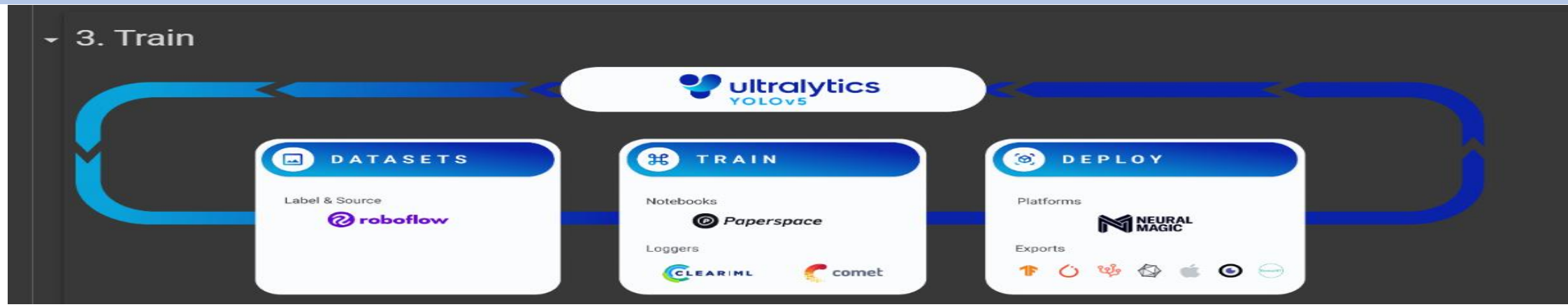
unzip train_data zip file

↓

check train_data



Run YOLOv5 in Colab

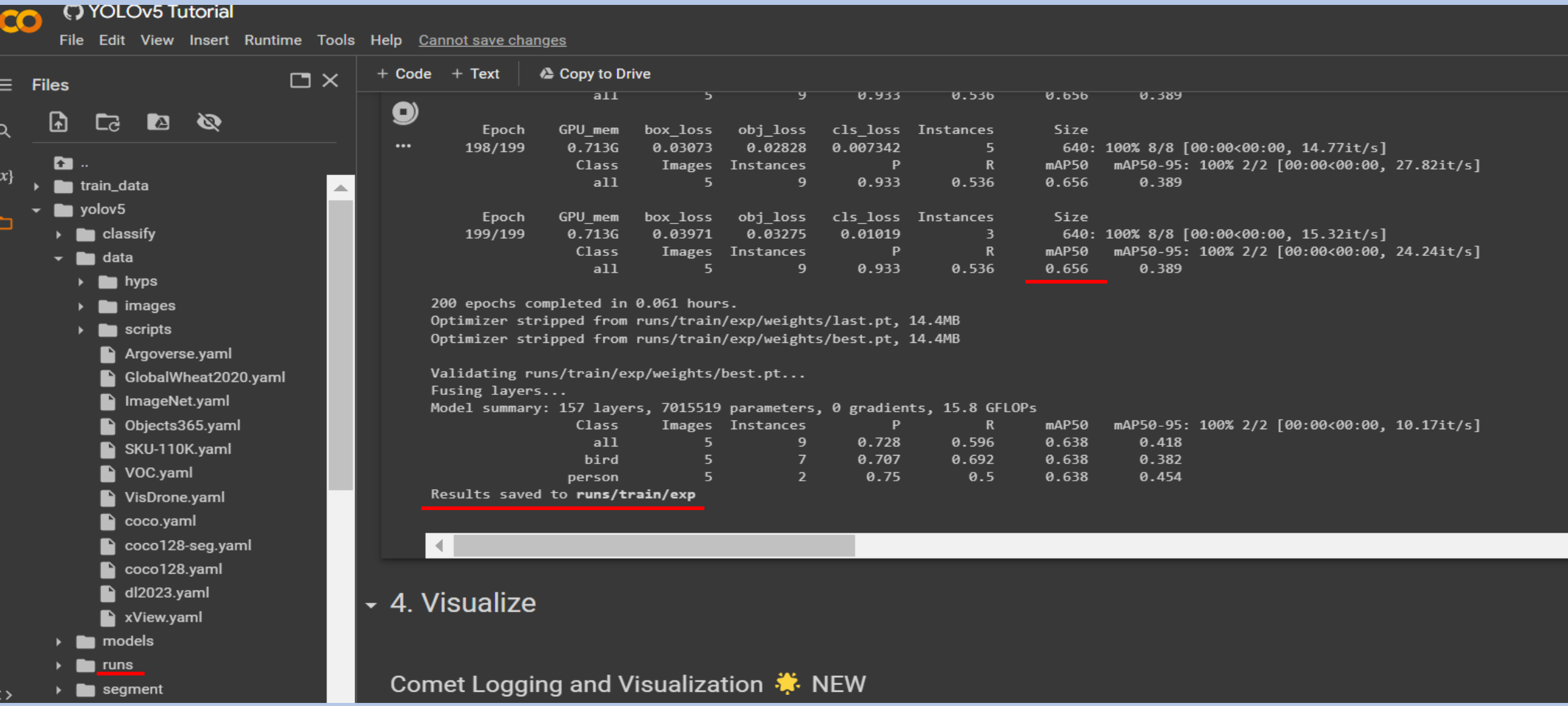



```
[ ]   1 # Train YOLOv5s on COCO128 for 3 epochs
      2 !python train.py --img 640 --batch 16 --epochs 3 --data coco128.yaml --weights yolov5s.pt --cache
```

Go to Train part → Change command

```
1 # Train YOLOv5s on COCO128 for 3 epochs
2 !python train.py --img 640 --batch 1 --epochs 200 --data dl2023.yaml --weights yolov5s.pt --cache
```

Run YOLOv5 in Colab

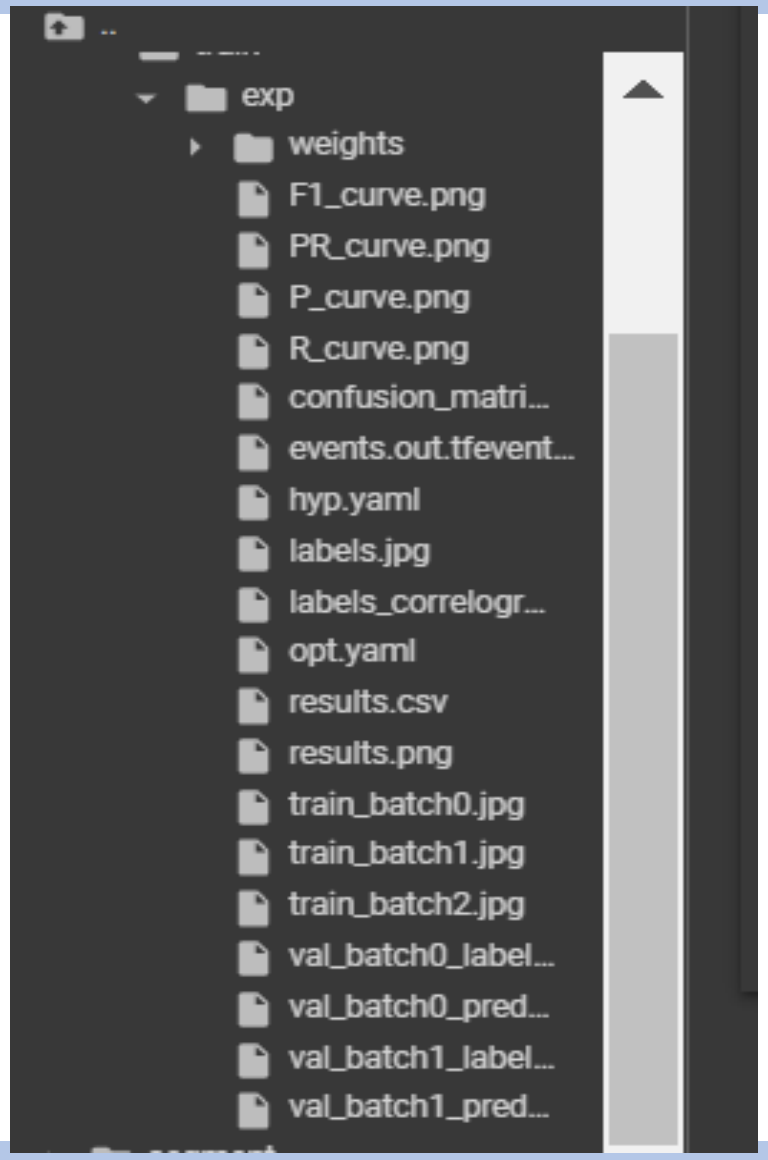


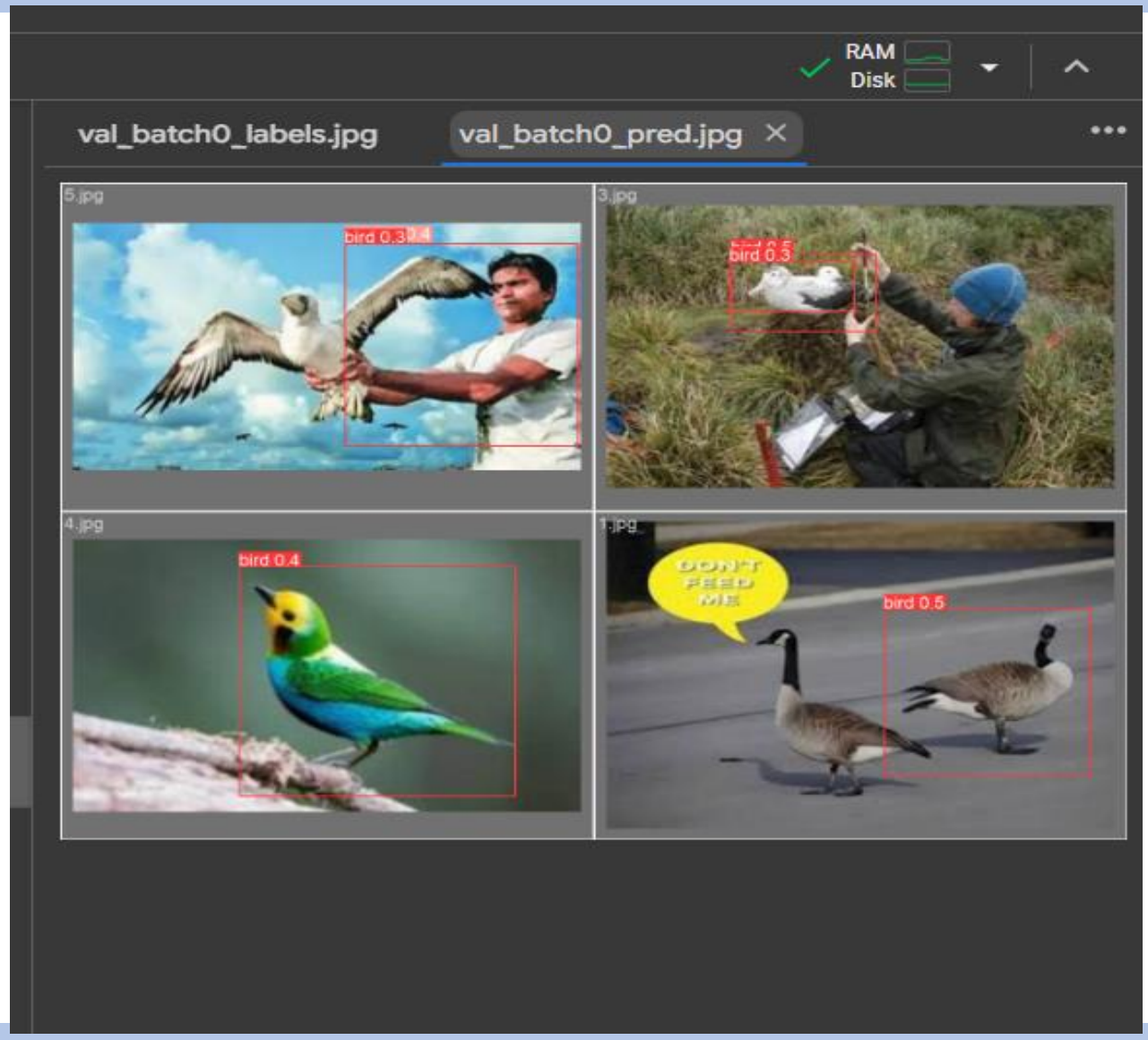
Check result

Run YOLOv5 in Colab

Check result

Download result





Run YOLOv5 in Colab

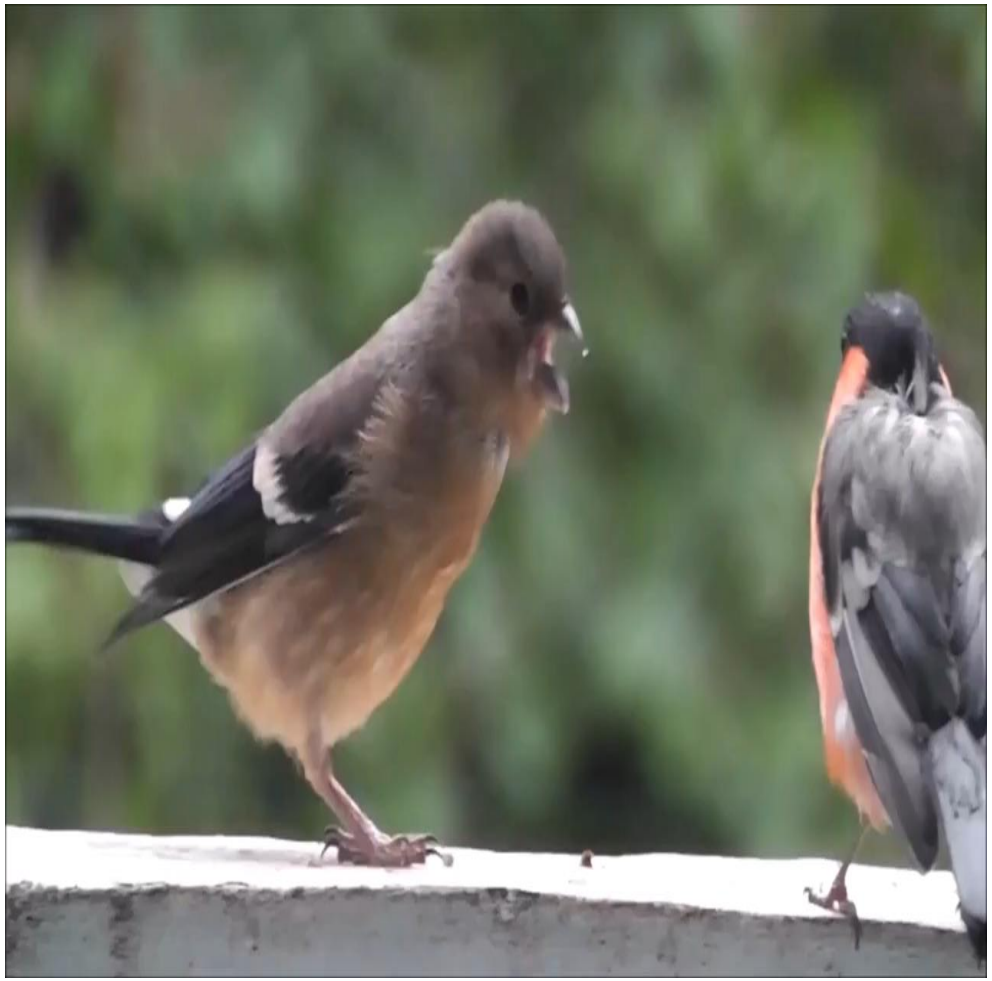
Check video clip

Check Video Clip

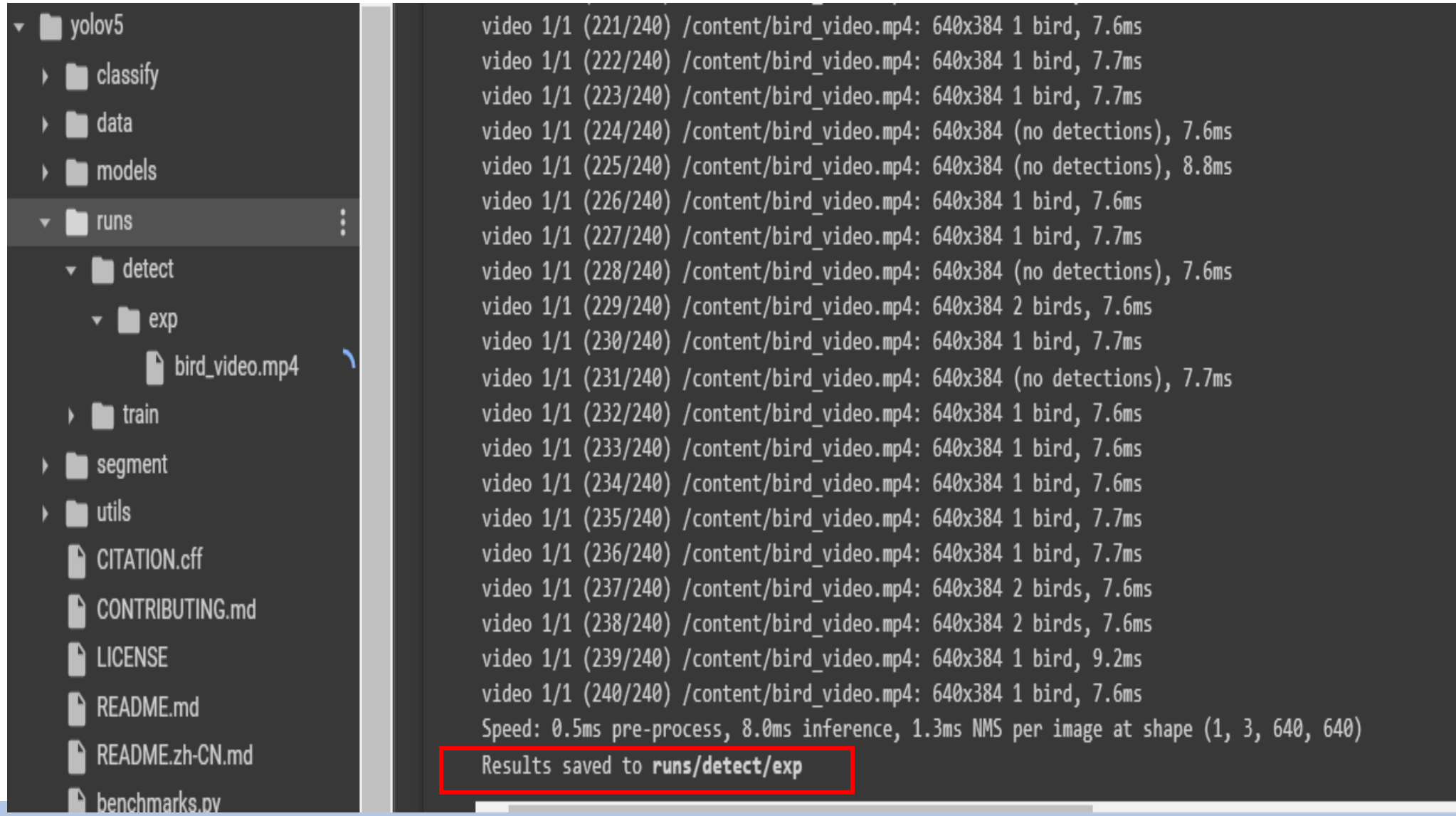
```
[18]  1 !python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images
      2 # display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

```
1 !python detect.py --weights runs/train/exp/weights/best.pt --img 640 --conf 0.25 --source ../bird_video.mp4
2 # display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

```
Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7225885 parameters, 0 gradients, 16.4 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 4/4 [00:06<00:00,  1.56s/it]
                   all        128        929      0.759      0.646      0.734       0.49
                person        128        254      0.857      0.706      0.805      0.525
               bicycle        128          6      0.773      0.577      0.725      0.414
                   car        128         46      0.664      0.435      0.551       0.24
```



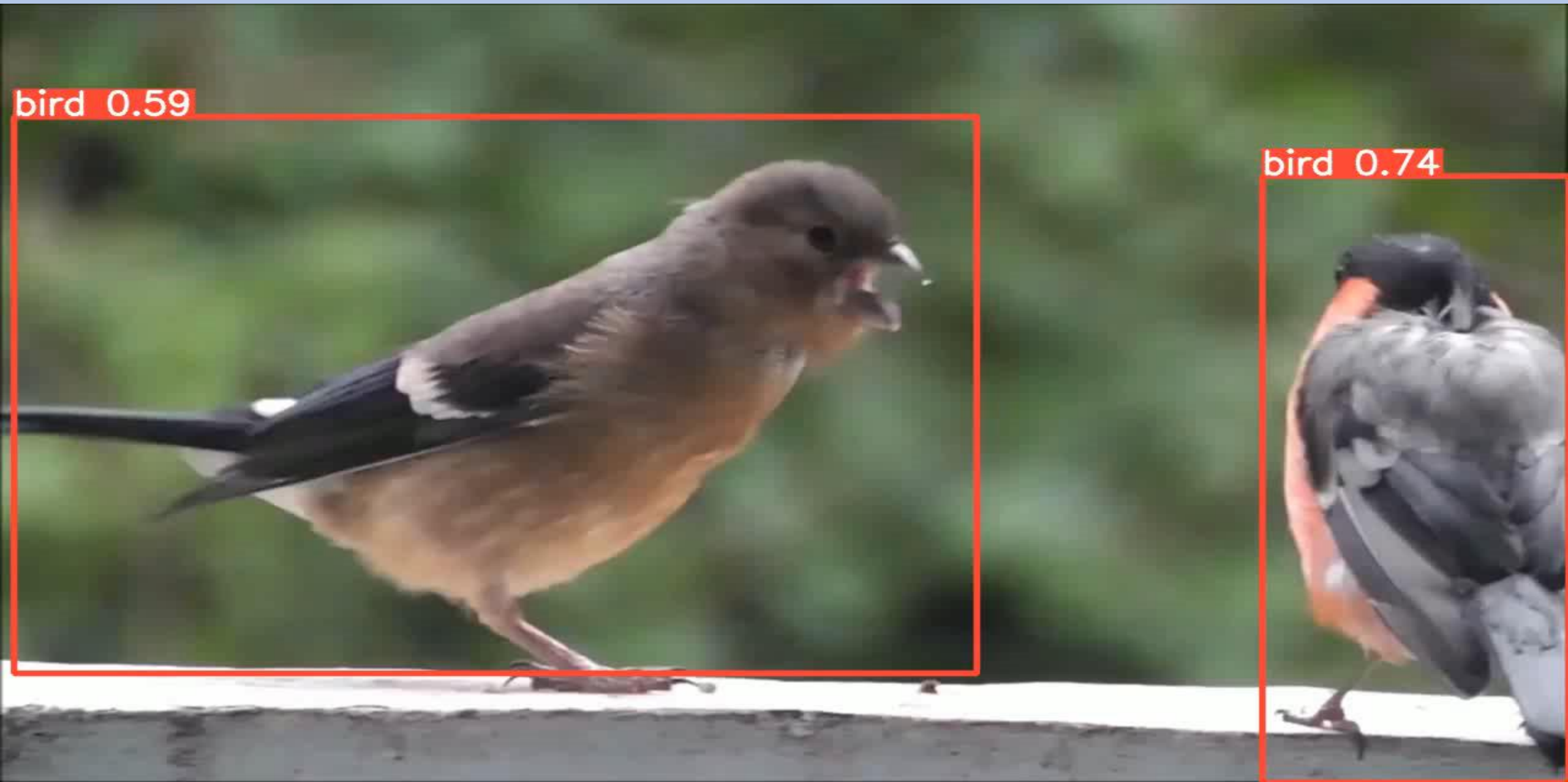
Check Video Clip



Download result

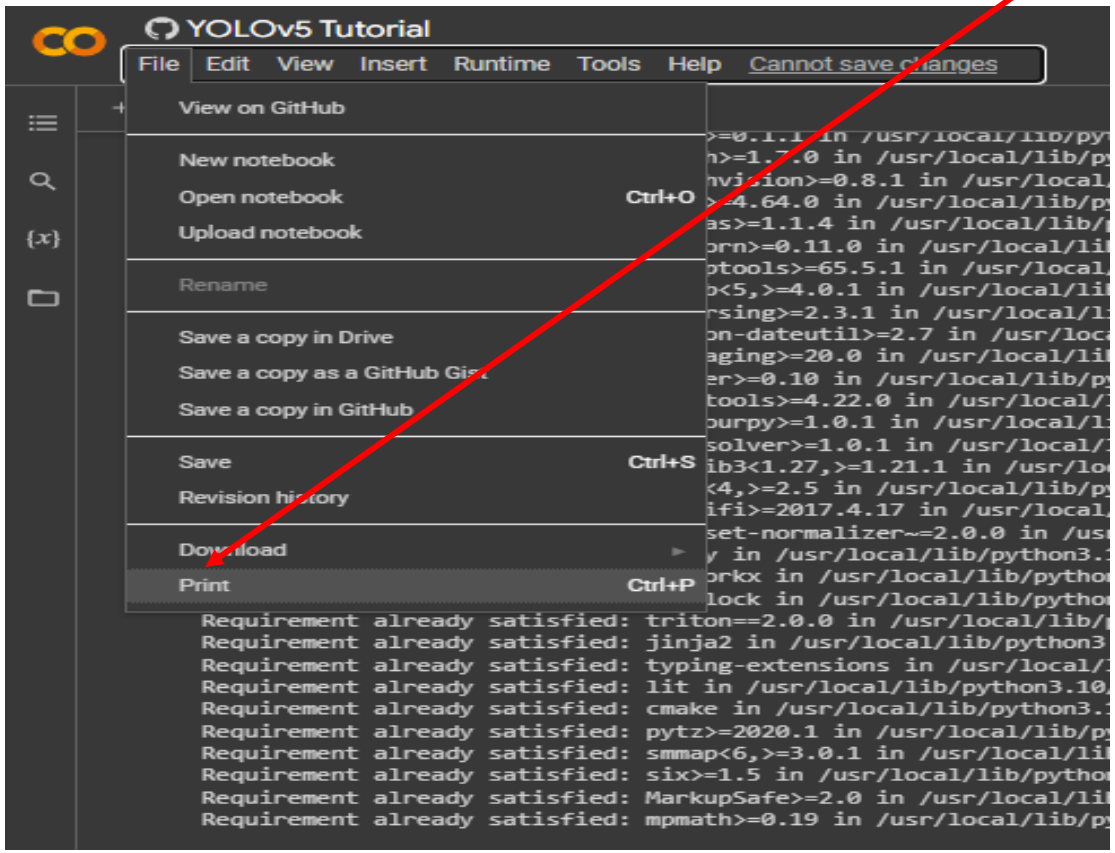
Check Video Clip

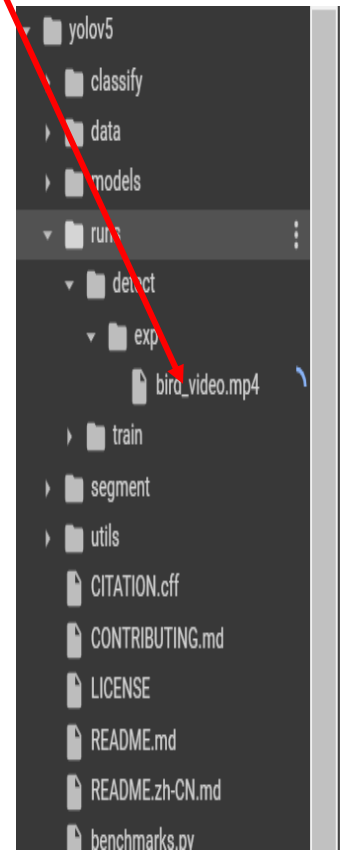
숭실대학교 Soongsil University

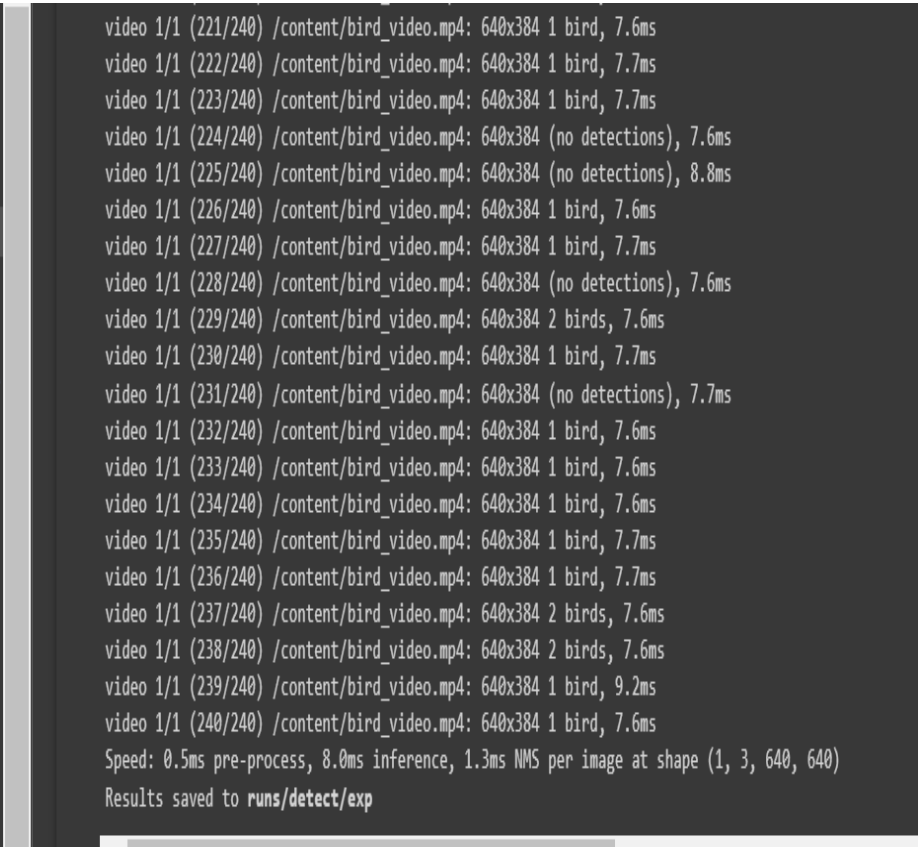


Upload PDF and Video Clip .mp4 file

Print as PDF and upload it
And also upload video clip result you downloaded

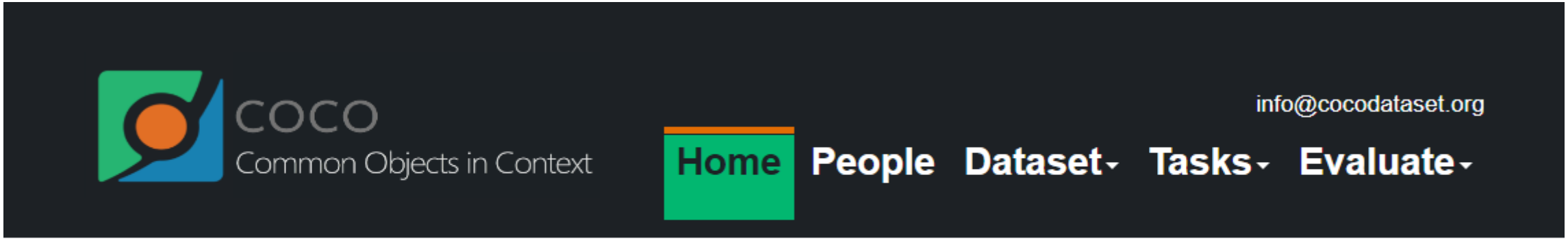






Data Preparation : Using Dataset from sites

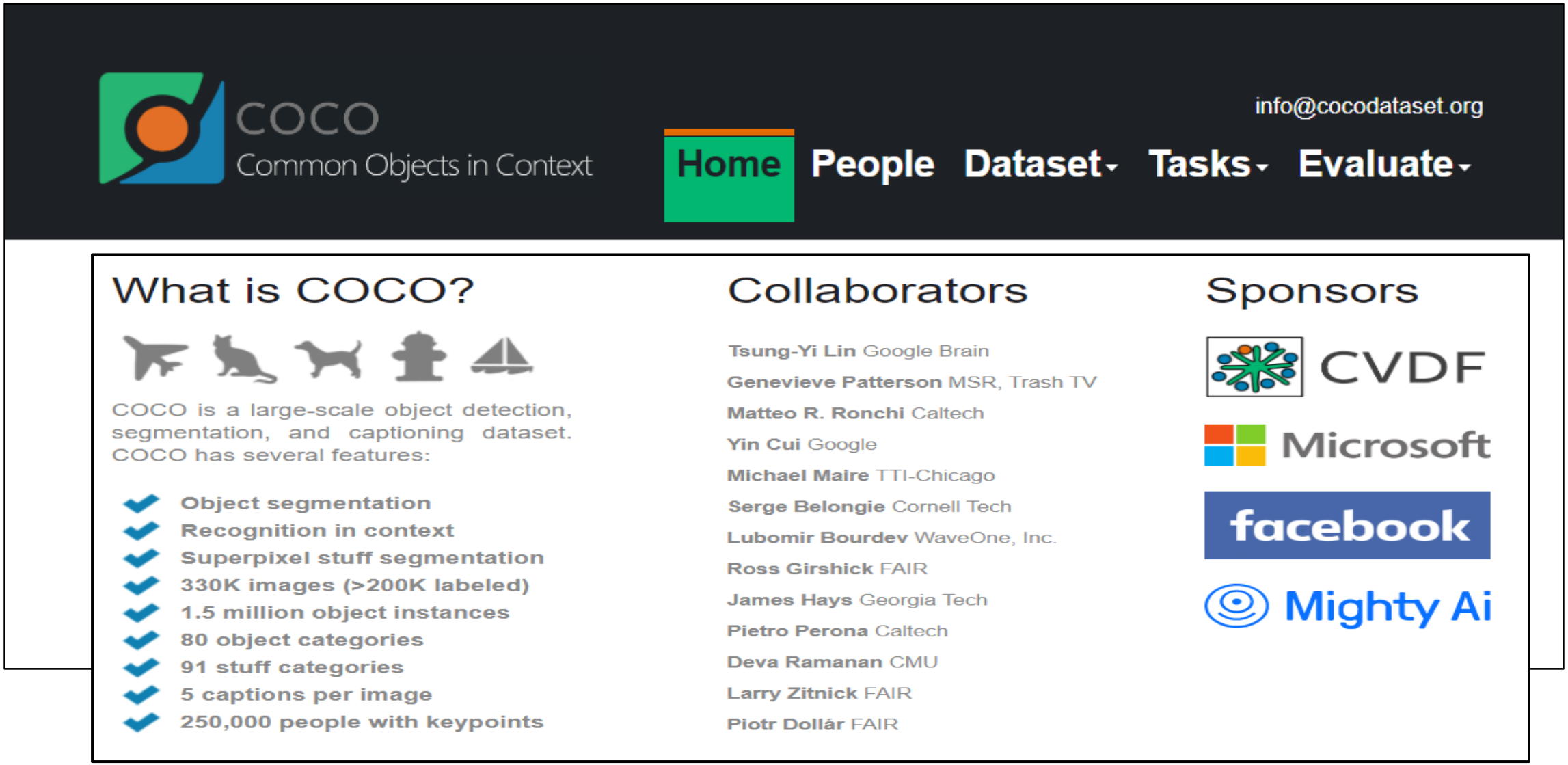
Data Preparation : Get Dataset from sites



News

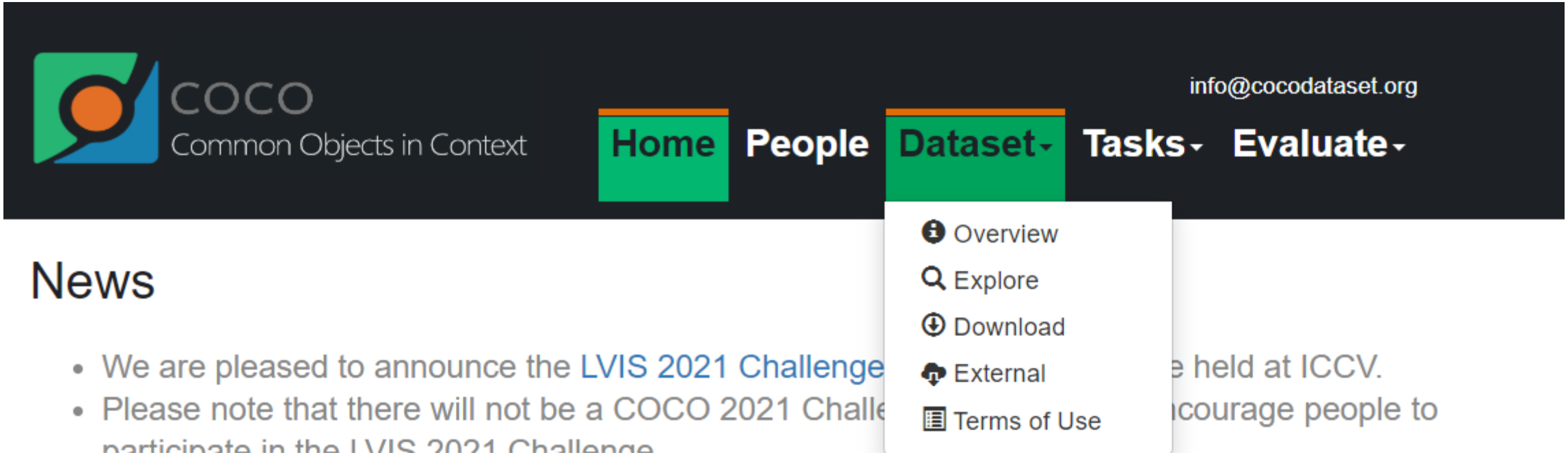
- We are pleased to announce the LVIS 2021 Challenge and Workshop to be held at ICCV.
- Please note that there will not be a COCO 2021 Challenge, instead, we encourage people to participate in the LVIS 2021 Challenge.
- We have partnered with the team behind the open-source tool FiftyOne to make it easier to download, visualize, and evaluate COCO
- FiftyOne is an open-source tool facilitating visualization and access to COCO data resources and serves as an evaluation tool for model analysis on COCO.

Data Preparation : Get Dataset from sites



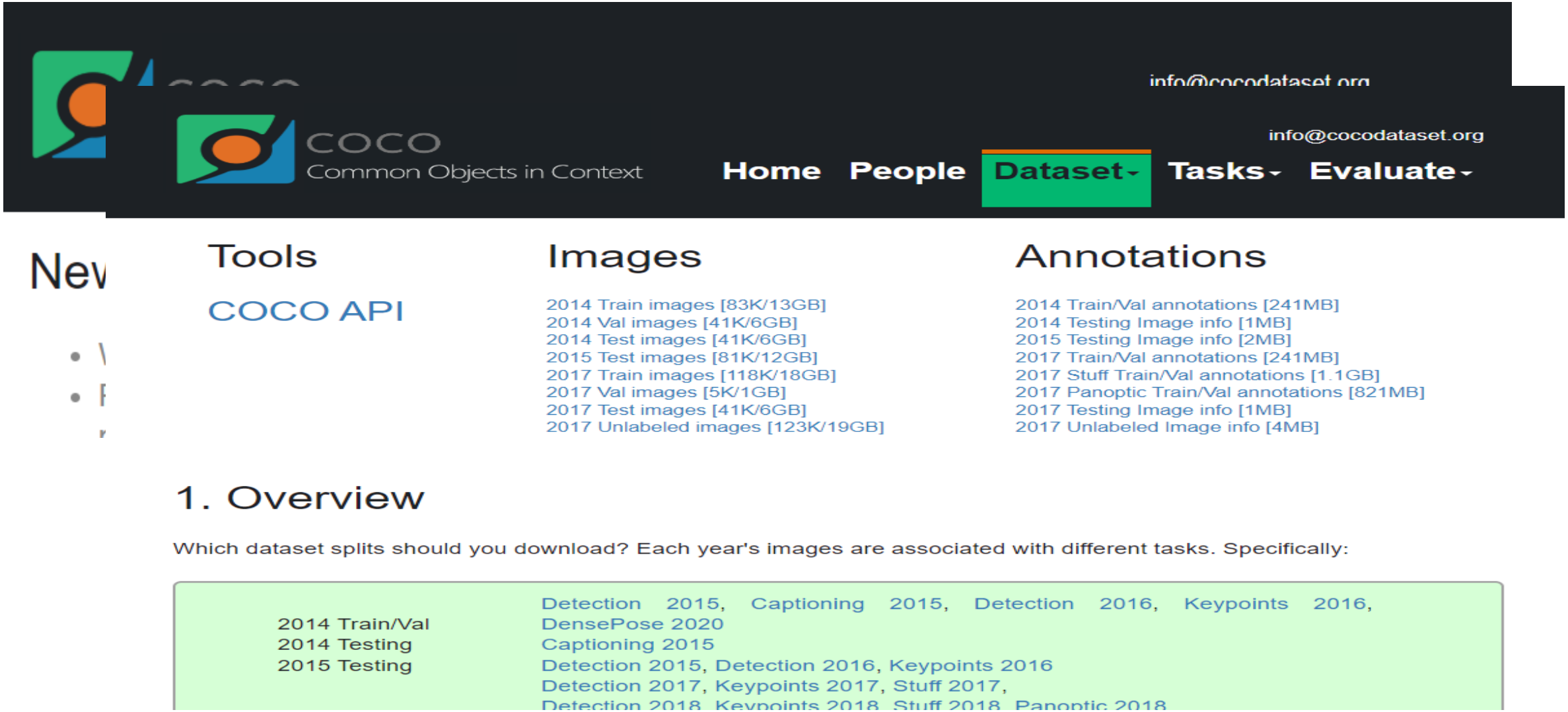
https://cocodataset.org/#home

Data Preparation : Get Dataset from sites



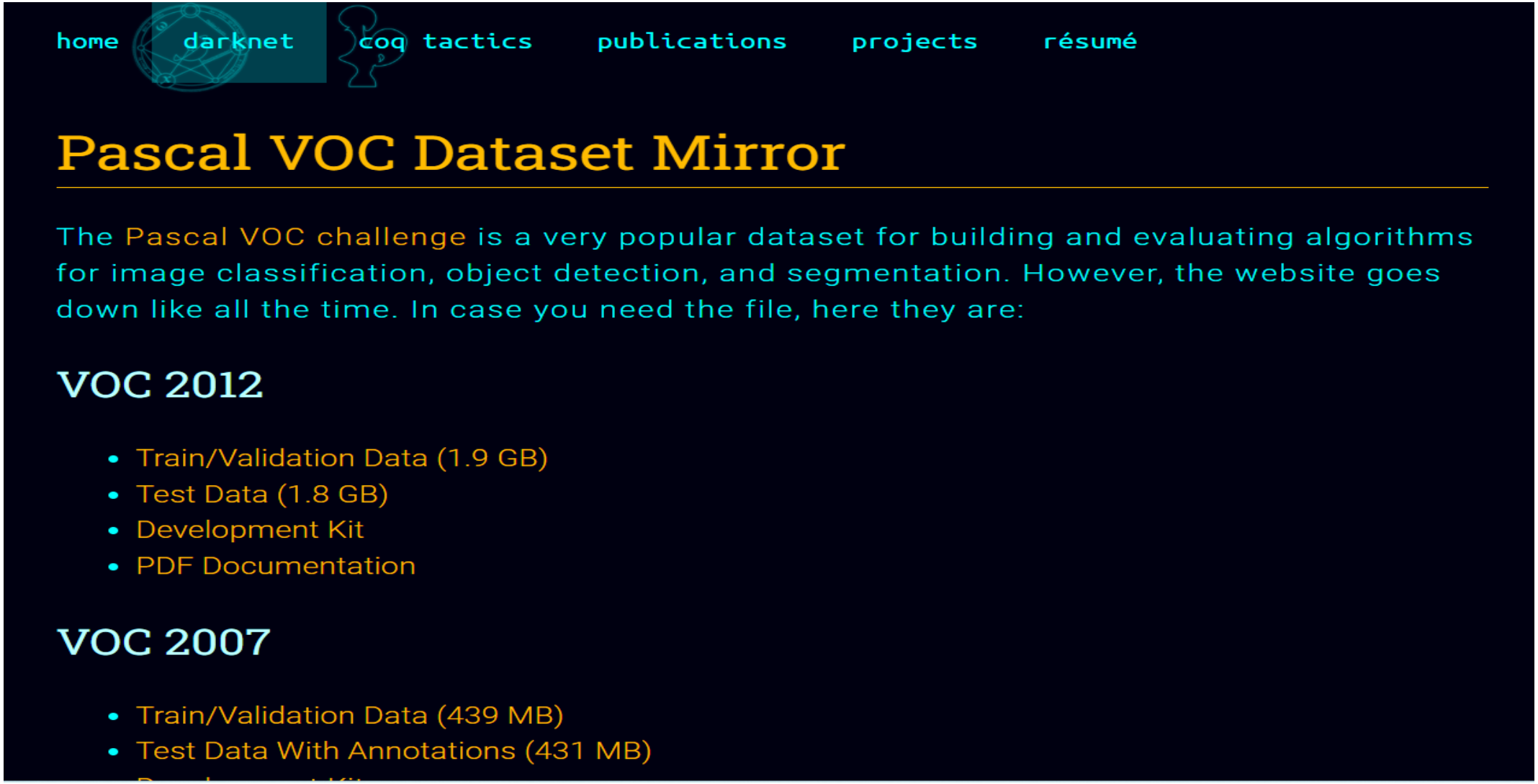
https://cocodataset.org/#home

Data Preparation : Get Dataset from sites



https://cocodataset.org/#home

Data Preparation : Get Dataset from sites



https://pjreddie.com/projects/pascal-voc-dataset-mirror/

GENERAL

The 10 Best Public Datasets for Object Detection in 2022

by Clemens Viernickel on September 14th, 2022

[The 10 Best Public Datasets for Object Detection in 2022 | Blog | Scale AI]

Data Preparation : Get Dataset from sites

숭실대학교 Soongsil University

TOWARDS AI

AI Community

COMPUTER VISION

50+ Object Detection Datasets from different industry domains

October 10, 2020

Last Updated on October 10, 2020 by Editorial Team

[50+ Object Detection Datasets from different industry domains – Towards AI]

Data Preparation : Get Dataset from sites

숭실대학교 Soongsil University

TOWARDS AI

AI Community

Computer Vision Datasets

xView: xView is one of the most massive publicly available datasets of overhead imagery. It contains images from complex scenes around the world, annotated using bounding boxes.

ImageNet: The largest image dataset for computer vision. It provides an accessible image database that is organized hierarchically, according to WordNet.

Kinetics-700: A large-scale dataset of video URLs from Youtube. Including human-centered actions. It contains over 700,000 videos.

Google's Open Images: A vast dataset from Google AI containing over 10 million images.

50+ Object Detection Datasets from different industry domains – Towards AI

Data Preparation : Get Dataset from sites

숭실대학교 Soongsil University

TOWARDS AI

AI Community

Computer Vision Datasets

xView: xView is one of the most massive publicly available datasets of overhead i
world, ann

ImageNet:
accessible
WordNet.

Kinetics-7
human-cer

Google's O
million ima

open-source datasets for face images with labeled gender and age. The images are collected from IMDB and Wikipedia. It has five million-plus labeled images.

Color Detection Dataset: The dataset contains a CSV file that has 865 color names with their corresponding RGB(red, green, and blue) values of the color. It also has the hexadecimal value of the color.

Stanford Dogs Dataset: It contains 20,580 images and 120 different dog breed categories.

Data Preparation : Get Dataset from sites

Stanford Dogs Dataset

Summary:

- 120 dog breeds
- ~150 images per class
- Total images: 20,580

Download dataset

Affenpinscher
(150 images)

ImageNet synset: n02110627

Afghan hound
(239 images)

ImageNet synset: n02088094

African hunting dog
(169 images)

Stanford Dogs Dataset

Aditya Khosla Nityananda Jayadevaprakash Bangpeng Yao Li Fei-Fei

Stanford University

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. Contents of this dataset:

- **Number of categories:** 120
- **Number of images:** 20,580
- **Annotations:** Class labels, Bounding boxes

Download

You can download the dataset using the links below:

- Images (757MB)
- Annotations (21MB)
- Lists, with train/test splits (0.5MB)
- Train Features (1.2GB), Test Features (850MB)
- README

Data Preparation : Get Dataset from sites

Stanford Dogs Dataset

Summary:

- 120 dog breeds
- ~150 images per class
- Total images: 20,580

Download dataset

ImageNet synset: n02113023

Pomeranian
(219 images)

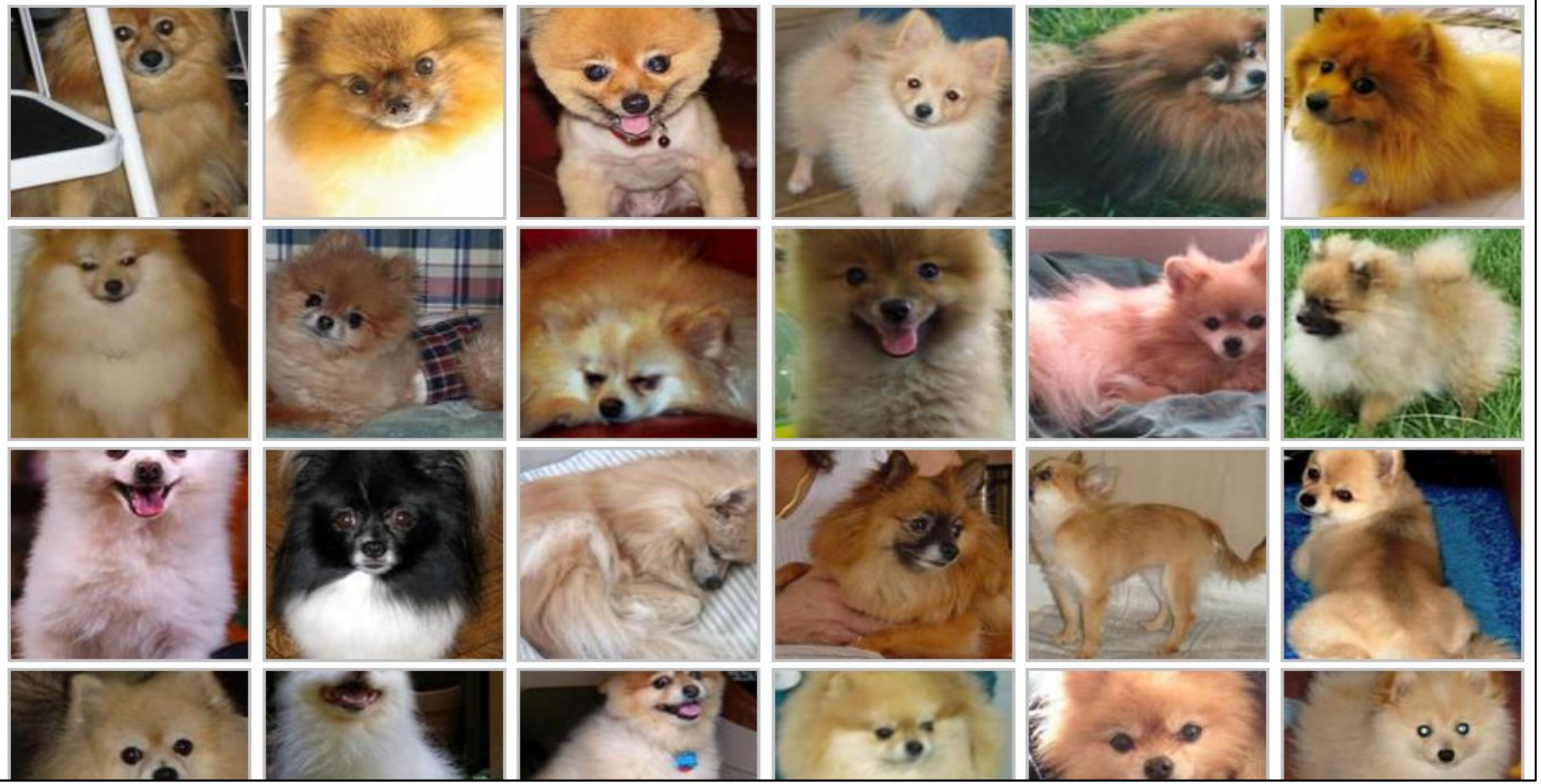
ImageNet synset: n02112018

Pug
(200 images)

ImageNet synset: n02110958

Redbone
(148 images)

Pomeranian (219 images)



Data Preparation : Get Dataset from sites

숭실대학교 Soongsil University

Stanford Dogs Dataset

Summary:

- 120 dog breeds
- ~150 images per class
- Total images: 20,580

Download dataset
(149 images)

ImageNet synset: n02086079

Pembroke
(181 images)

ImageNet synset: n02113023

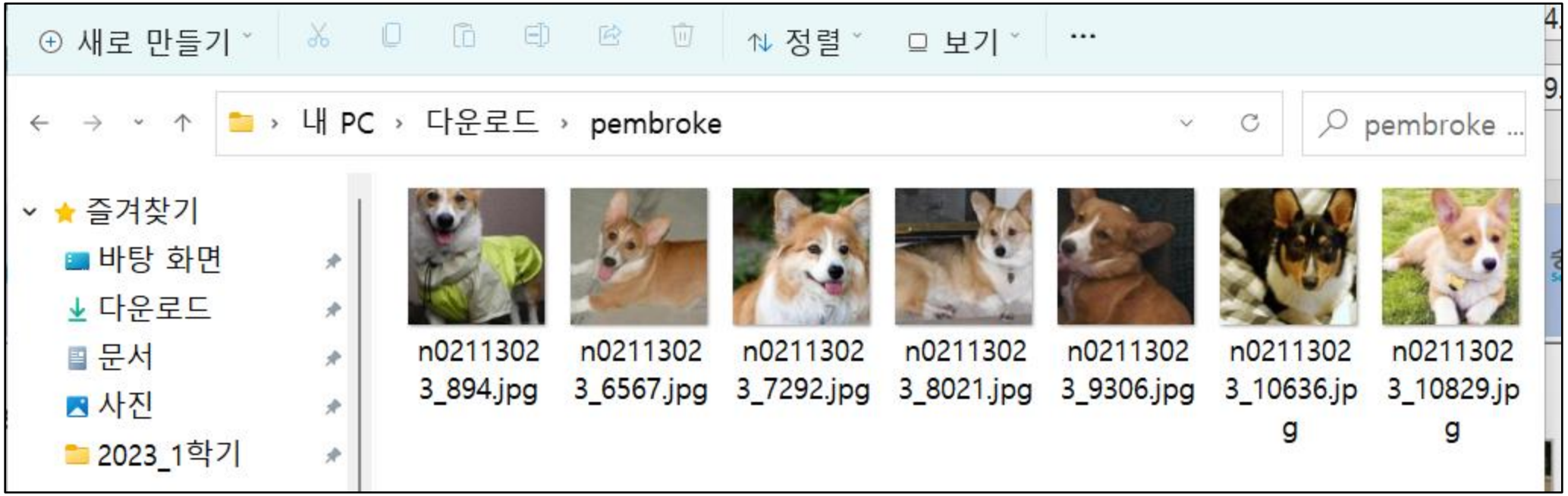
Pomeranian
(219 images)

ImageNet synset: n02112018

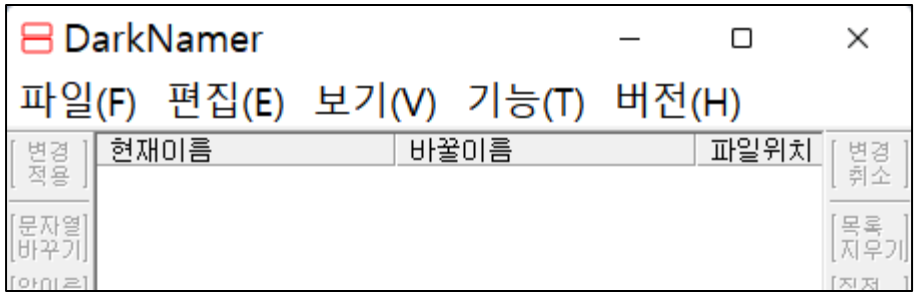
Pug
(200 images)

Pembroke (181 images)

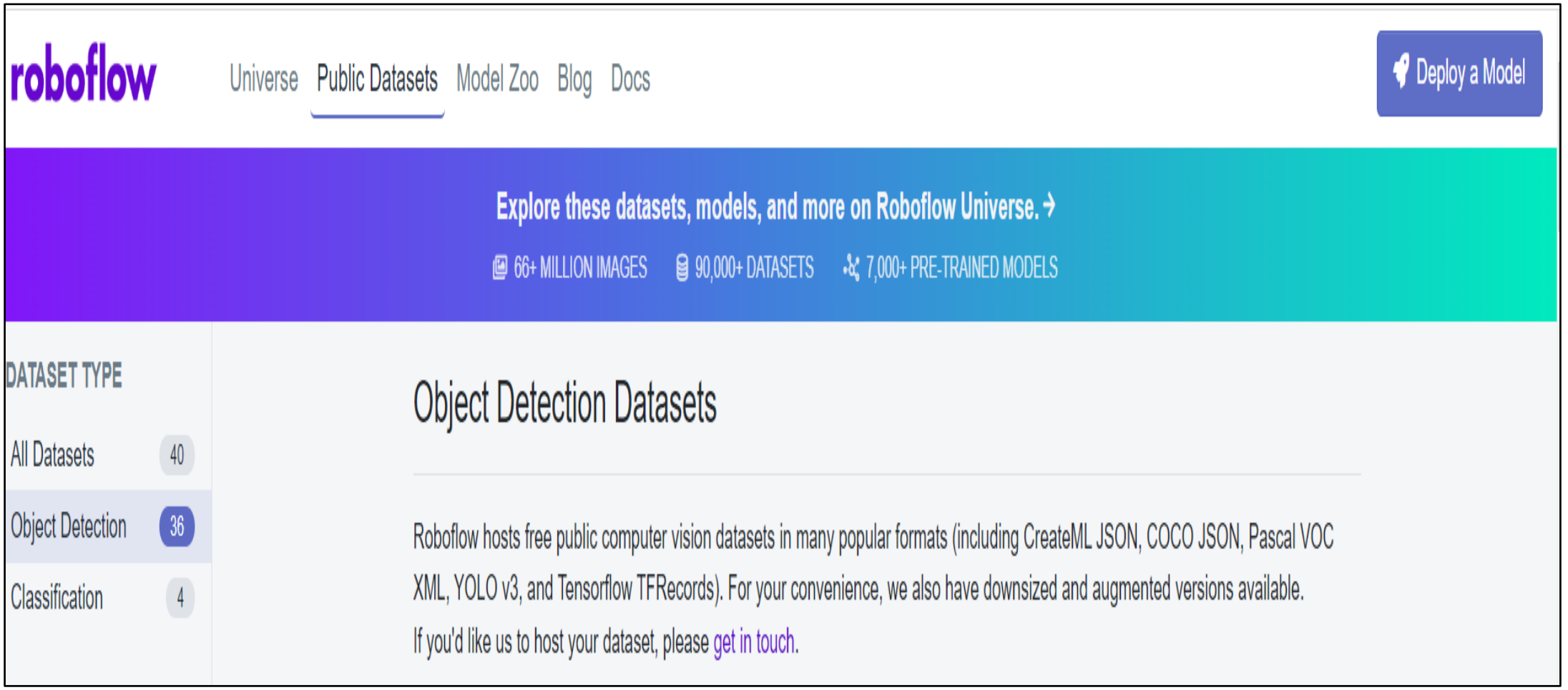
Data Preparation : Get Dataset from sites



✓ **Change file name at one time ?**

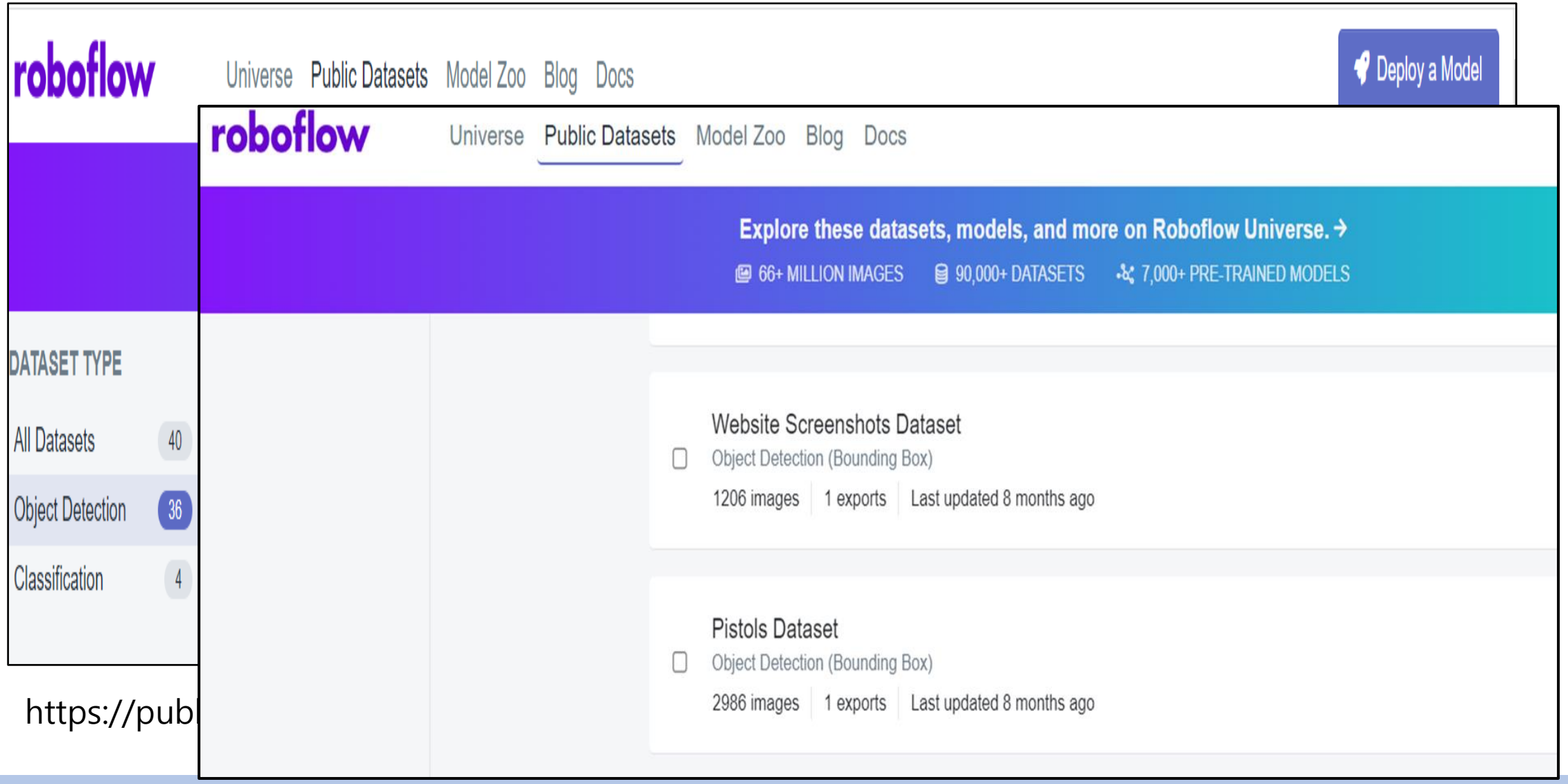


Data Preparation : Get Dataset from sites



https://public.roboflow.com/object-detection

Data Preparation : Get Dataset from sites



https://publ

Data Preparation : Get Dataset from sites

숭실대학교 Soongsil University

roboflow Universe Public Datasets Model Zoo Blog Docs

Explore these datasets, mod

66+ MILLION IMAGES 90,000

Dataset Summary
Dataset Health Check
DOWNLOADS
resize-416x416 2973

Pistols Dataset

Shared By
University of Grenada
February 2020

License
Public Domain
More Info

Annotations
Guns
Object Detection

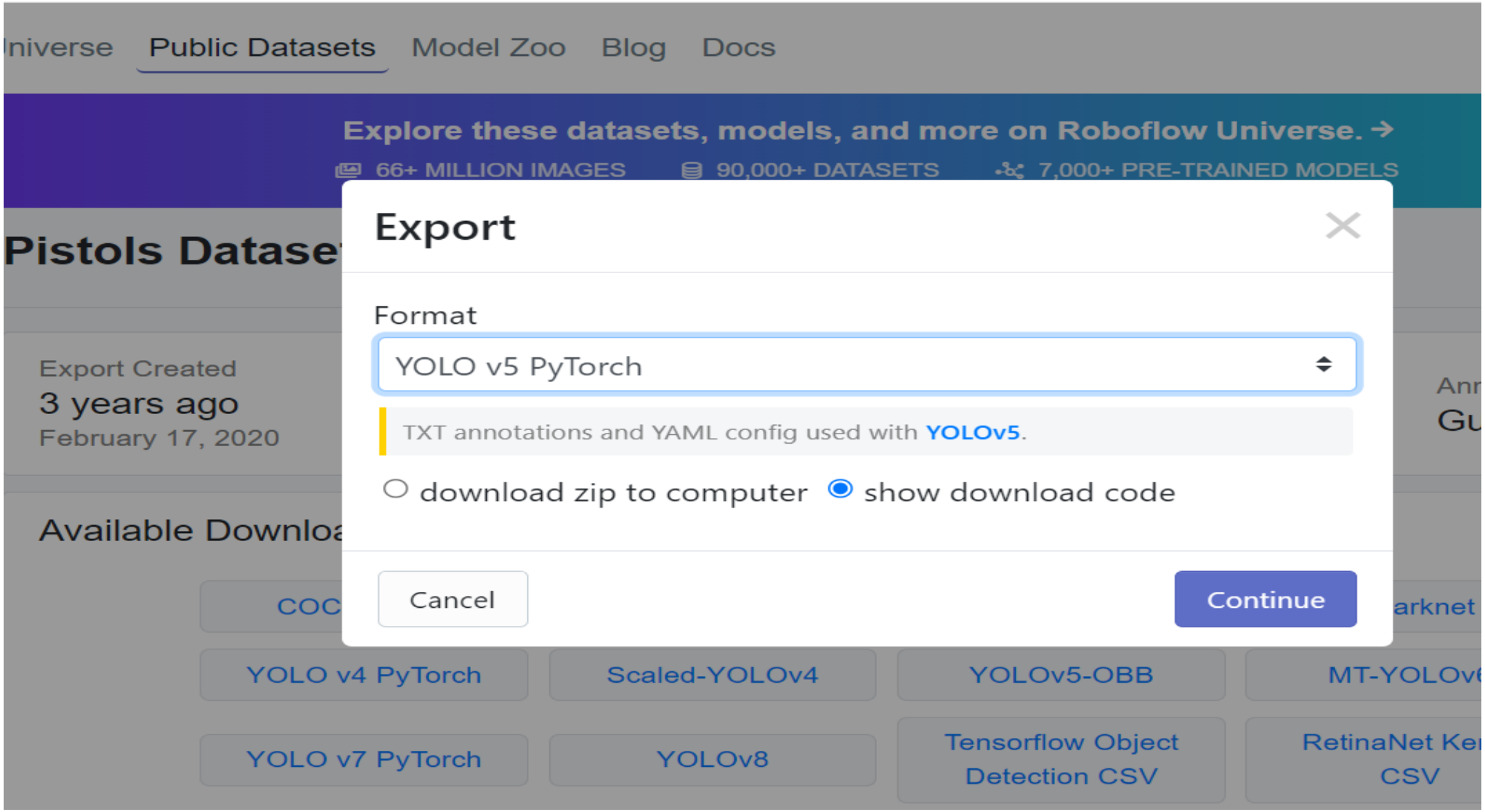
Downloads

resize-416x416 2973 Images

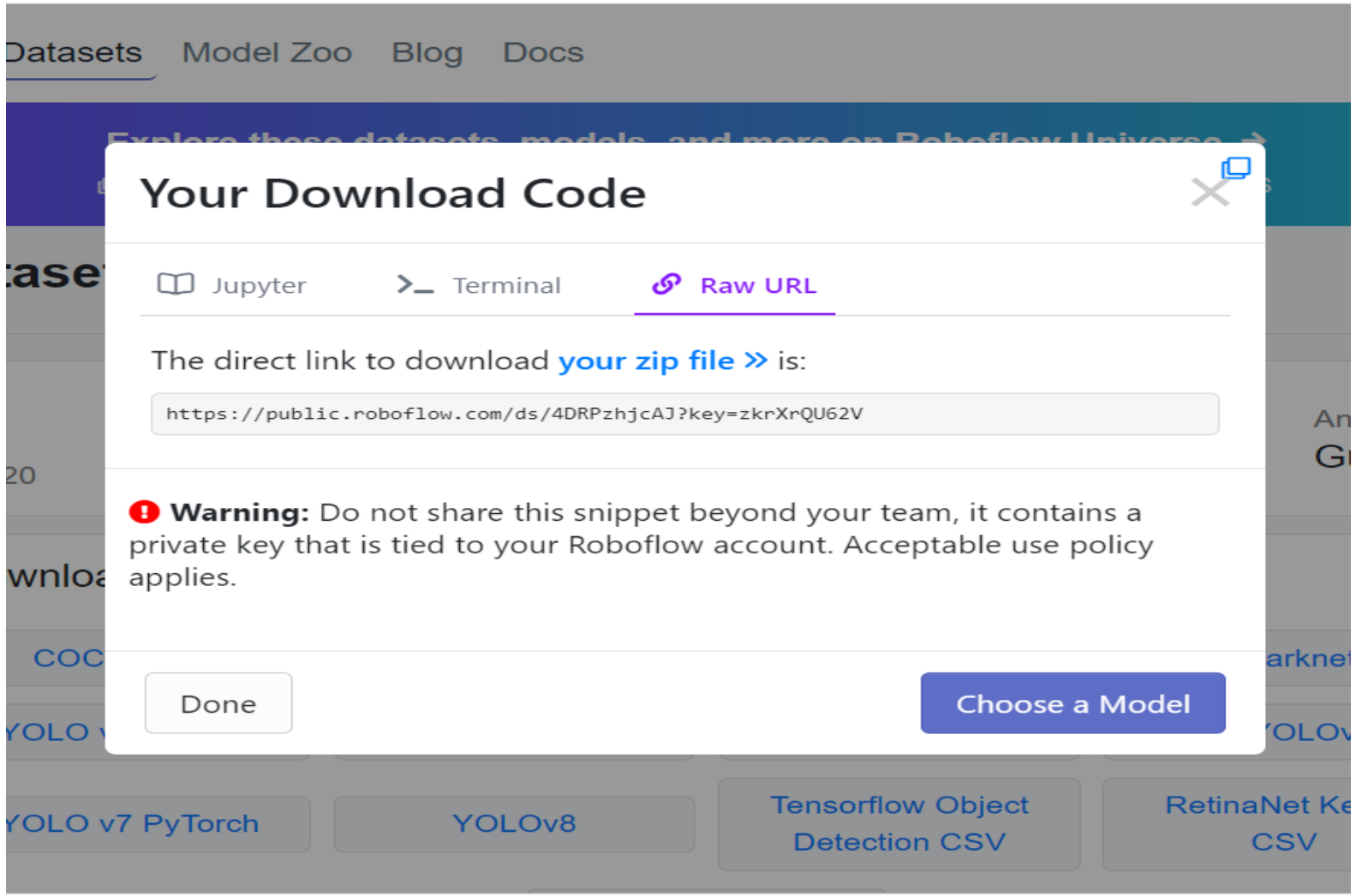
Overview

This dataset contains 2986 images and 3448 labels across a single annotation class: pistols. Images are wide-ranging: pistols in-hand, cartoons, and staged studio quality images of guns.
The dataset was originally released by the University of Grenada , duplicates removed, and rehosted by a Roboflow user.

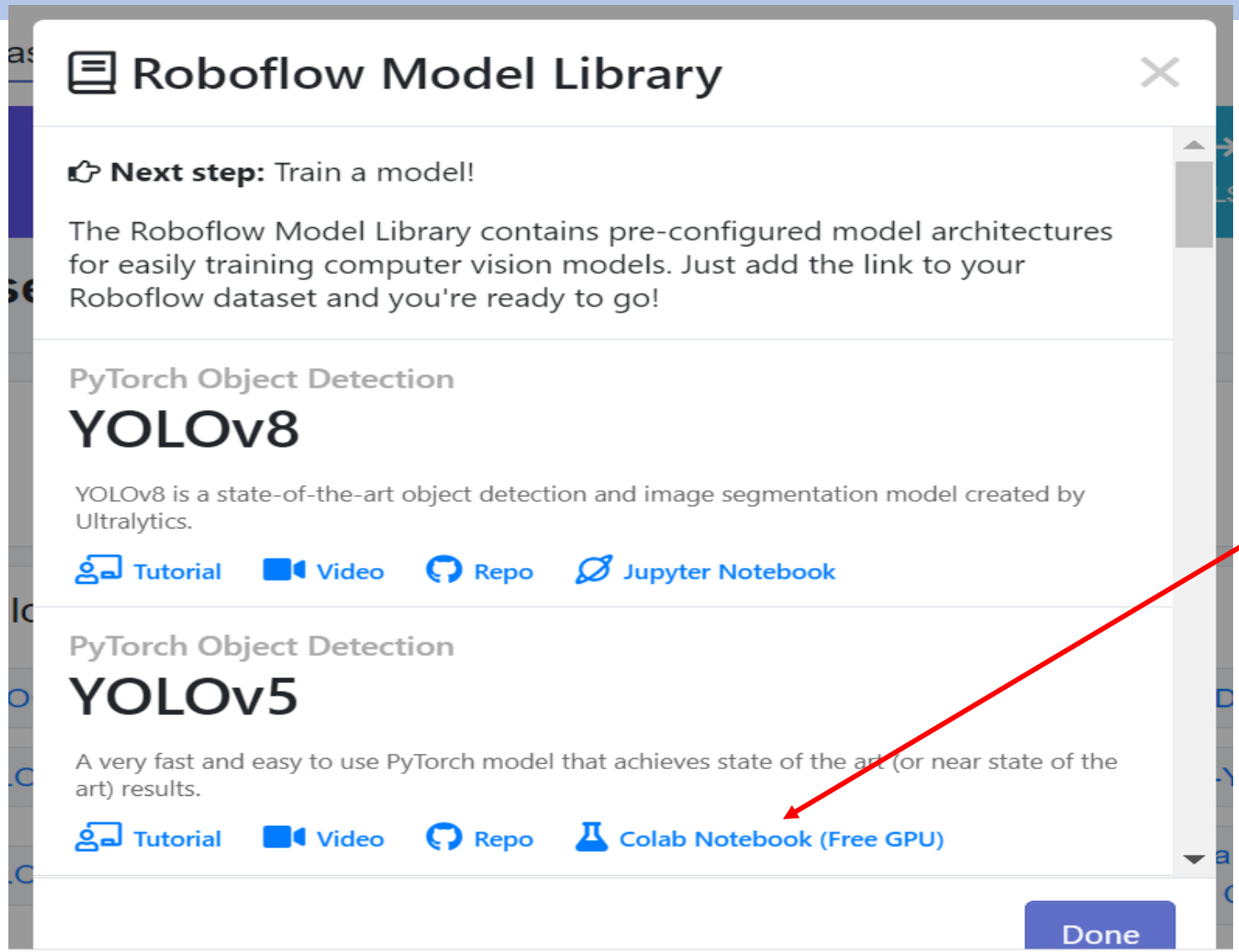
Data Preparation : Get Dataset from sites



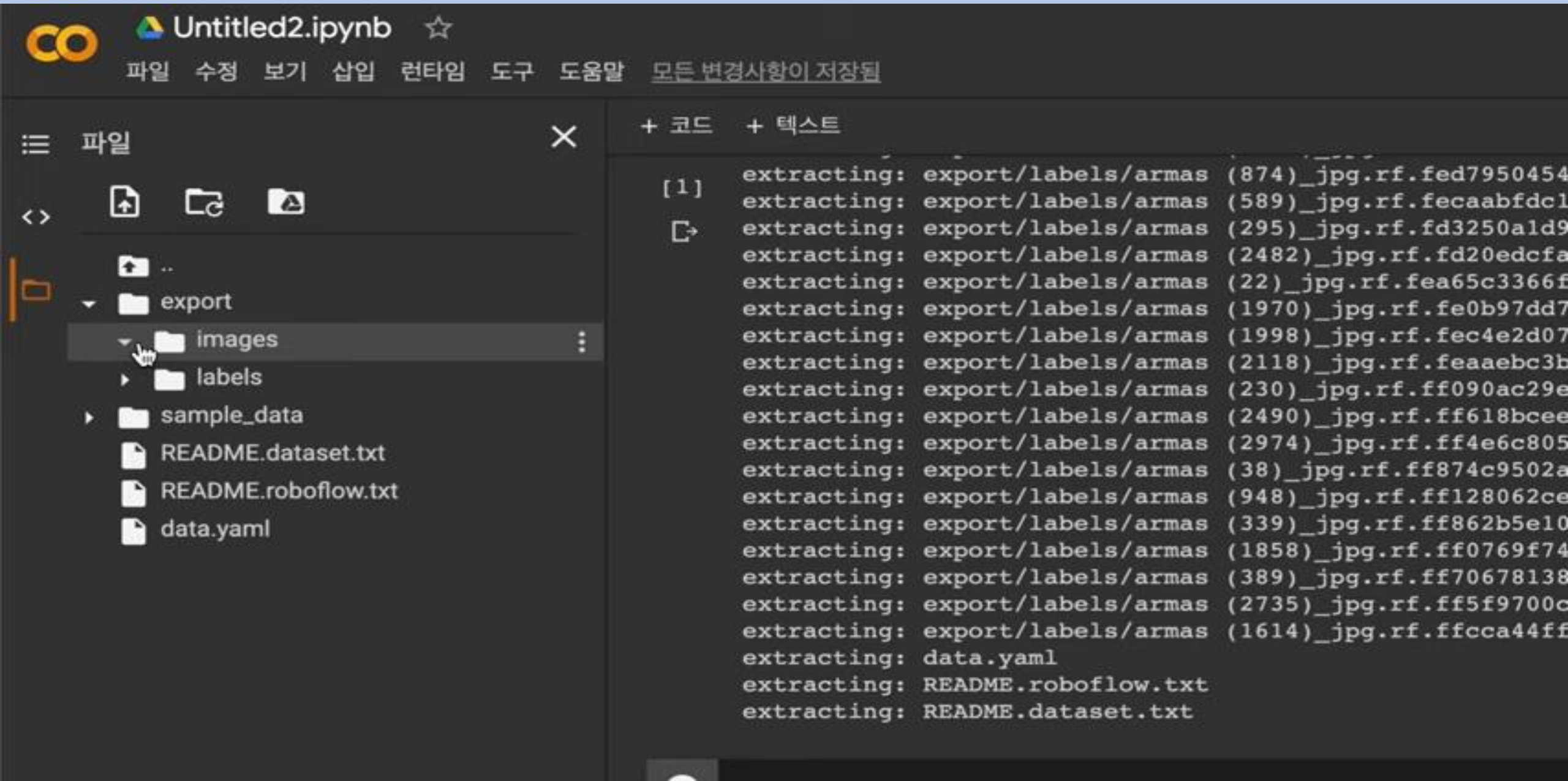
Data Preparation : Get Dataset from sites



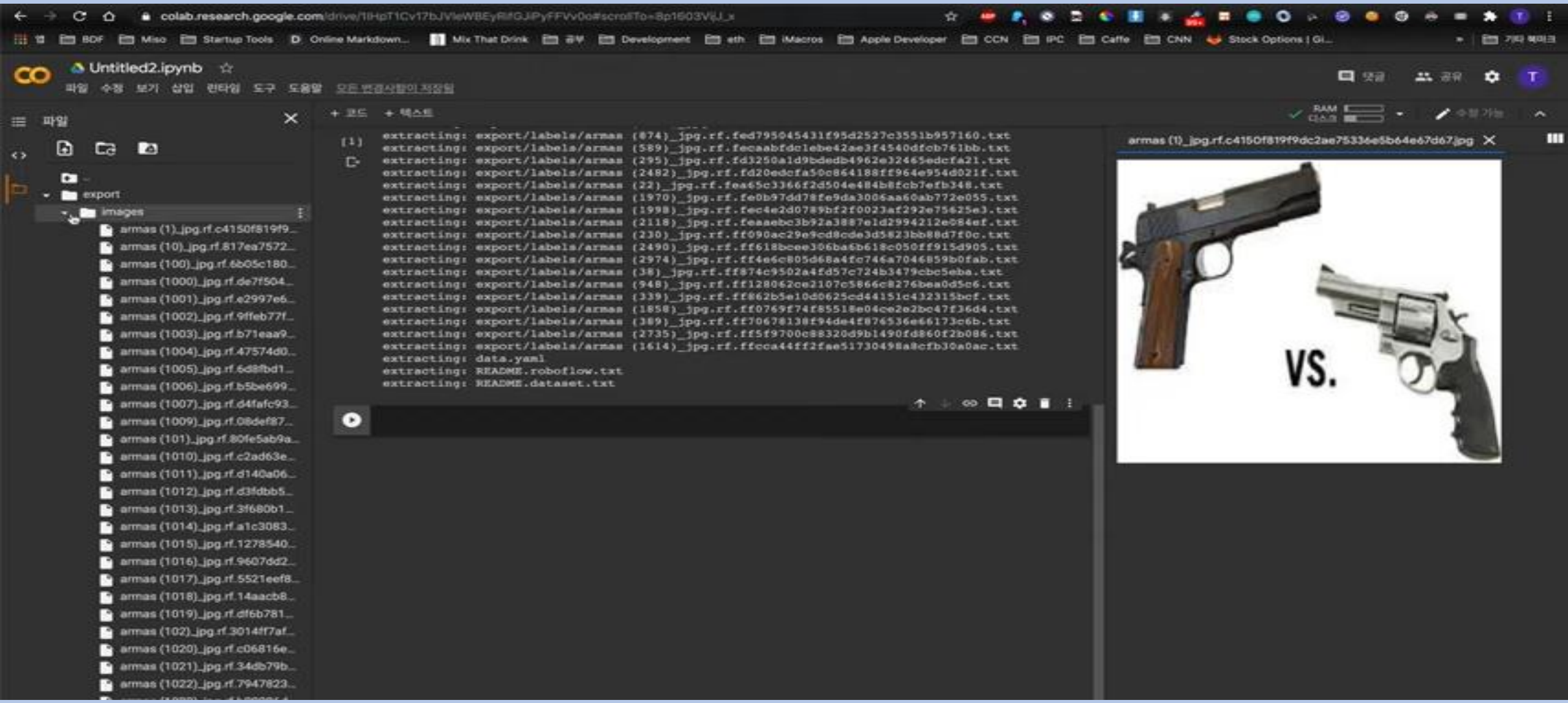
Data Preparation : Get Dataset from sites



Run YOLOv5 in Colab



Run YOLOv5 in Colab

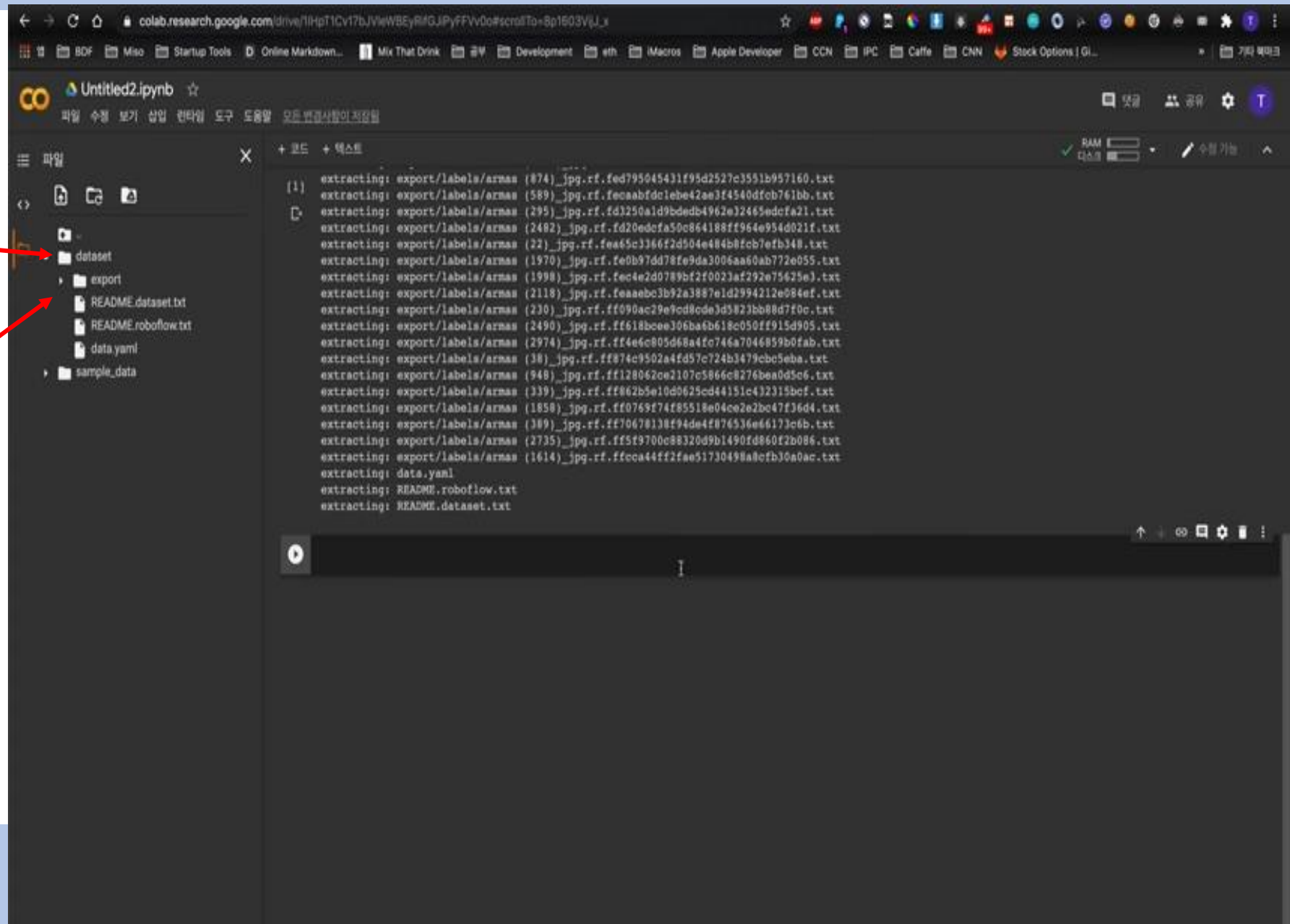


Run YOLOv5 in Colab

숭실대학교 Soongsil University

Make Dataset fold

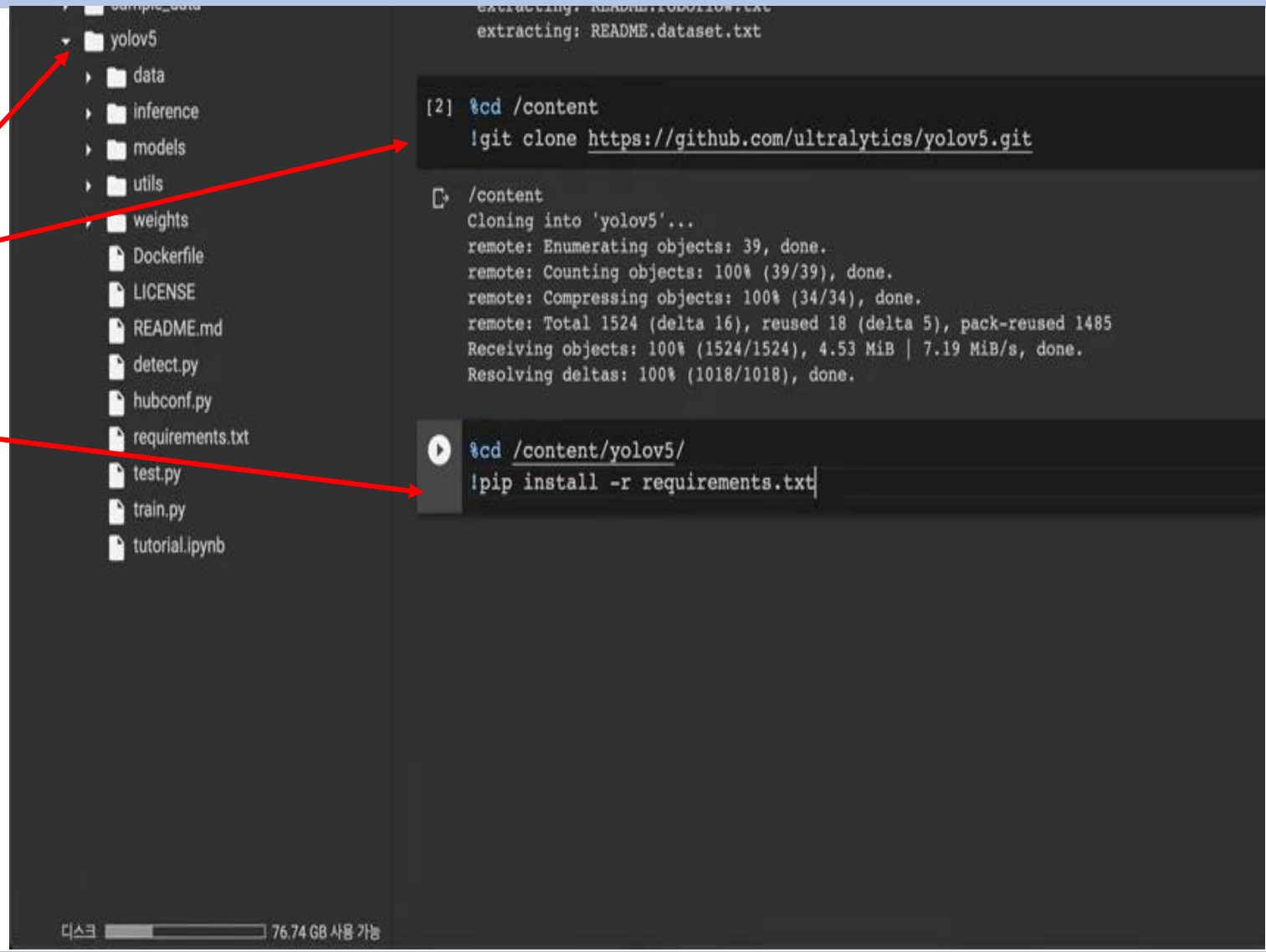
Move export fold to dataset



Run YOLOv5 in Colab

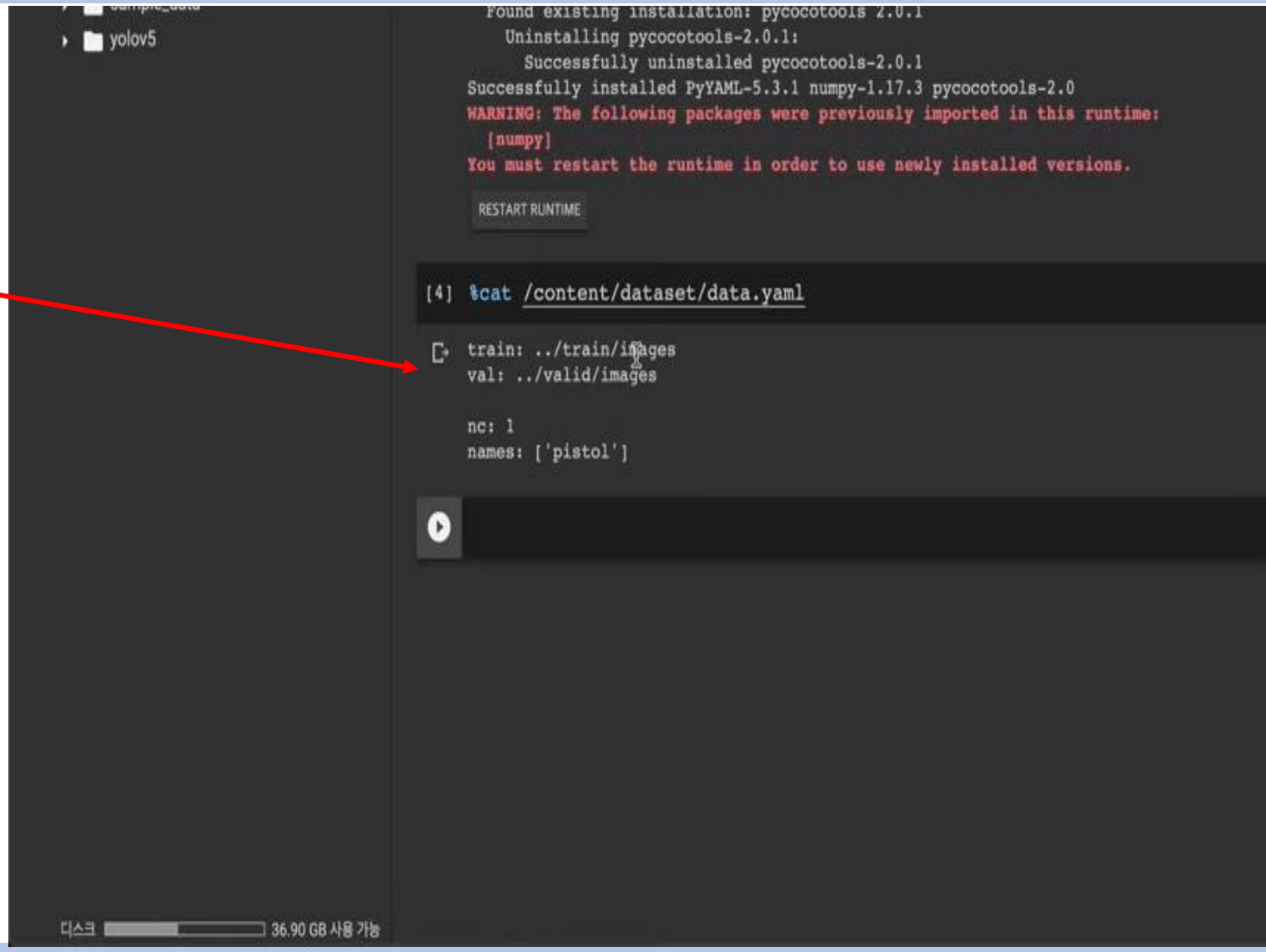
Download YOLOv5 from github

Download env from requirements



Run YOLOv5 in Colab

Check data.yaml file and edit



Run YOLOv5 in Colab

Check data.yaml file and edit

```
[7] with open('/content/dataset/train.txt', 'w') as f:
      f.write('\n'.join(train_img_list) + '\n')

    with open('/content/dataset/val.txt', 'w') as f:
      f.write('\n'.join(val_img_list) + '\n')
```

```
import yaml

with open('/content/dataset/data.yaml', 'r') as f:
  data = yaml.load(f)

print(data)

data['train'] = '/content/dataset/train.txt'
data['val'] = '/content/dataset/val.txt'

with open('/content/dataset/data.yaml', 'w') as f:
  yaml.dump(data, f)

print(data)
```

Run YOLOv5 in Colab

Read img data file and split data

```
val: ../valid/images

nc: 1
names: ['pistol']
```

```
[5] %cd /
from glob import glob

img_list = glob('/content/dataset/export/images/*.jpg')

print(len(img_list))
```

```
/
2971
```

```
from sklearn.model_selection import train_test_split

train_img_list, val_img_list = train_test_split(img_list, test_size=0.2, random_state=2000)

print()
```

Run YOLOv5 in Colab

Read img data file and split data

```
val: ../valid/images

nc: 1
names: ['pistol']
```

```
[5] %cd /
from glob import glob

img_list = glob('/content/dataset/export/images/*.jpg')

print(len(img_list))
```

```
/
2971
```

```
[6] from sklearn.model_selection import train_test_split

train_img_list, val_img_list = train_test_split(img_list, test_size=0.2, random_state=2000)

print(len(train_img_list), len(val_img_list))
```

```
2376 595
```

Run YOLOv5 in Colab

Create train, val data txt file

```
[6] from sklearn.model_selection import train_test_split

    train_img_list, val_img_list = train_test_split(img_list, test_size=0.2, random_state=2000)

    print(len(train_img_list), len(val_img_list))
```

```
2376 595
```

```
with open('/content/dataset/train.txt', 'w') as f:
    f.write('\n'.join(train_img_list) + '\n')

with open('/content/dataset/val.txt', 'w') as f:
    f.write('\n'.join(val_img_list) + '\n')
```

Run YOLOv5 in Colab

숭실대학교 Soongsil University

Train

```
%cd /content/yolov5/

!python train.py --img 416 --batch 16 --epochs 50 --data /content/dataset/data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name gun_yolov5s_results
```

Run YOLOv5 in Colab

숭실대학교 Soongsil University

Train

```
%cd /content/yolov5/

!python train.py --img 416 --batch 16 --epochs 50 --data /content/dataset/data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name gun_yolov5s_results
```

```
/content/yolov5
Apex recommended for faster mixed precision training: https://github.com/NVIDIA/apex
Namespace(batch_size=16, bucket='', cache_images=False, cfg='./models/yolov5s.yaml', data='/content/dataset/data.yaml', device
Using CUDA device0 _CudaDeviceProperties(name='Tesla T4', total_memory=15079MB)

2020-07-15 03:55:39.065777: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic librar
Start Tensorboard with "tensorboard --logdir=runs", view at http://localhost:6006/
Hyperparameters {'optimizer': 'SGD', 'lr0': 0.01, 'momentum': 0.937, 'weight_decay': 0.0005, 'giou': 0.05, 'cls': 0.58, 'cls_p
Overriding ./models/yolov5s.yaml nc=80 with nc=1

                 from  n    params  module                                  arguments
  0                -1  1      3520  models.common.Focus                     [3, 32, 3]
  1                -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
  2                -1  1     19904  models.common.BottleneckCSP             [64, 64, 1]
  3                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  4                -1  1    161152  models.common.BottleneckCSP             [128, 128, 3]
  5                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  6                -1  1    641792  models.common.BottleneckCSP             [256, 256, 3]
  7                -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
  8                -1  1    656896  models.common.SPP                       [512, 512, [5, 9, 13]]
  9                -1  1   1248768  models.common.BottleneckCSP             [512, 512, 1, False]
 10                -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12          [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  1    378624  models.common.BottleneckCSP             [512, 256, 1, False]
 14                -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16          [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  1     95104  models.common.BottleneckCSP             [256, 128, 1, False]
 18                -1  1      2322  torch.nn.modules.conv.Conv2d            [128, 18, 1, 1]
 19                -2  1    147712  models.common.Conv                      [128, 128, 3, 2]
 20         [-1, 14]  1         0  models.common.Concat                    [1]
 21                -1  1    313088  models.common.BottleneckCSP             [256, 256, 1, False]
 22                -1  1      4626  torch.nn.modules.conv.Conv2d            [256, 18, 1, 1]
 23                -2  1    590336  models.common.Conv                      [256, 256, 3, 2]
 24         [-1, 10]  1         0  models.common.Concat                    [1]
```

Run YOLOv5 in Colab

숭실대학교 Soongsil University

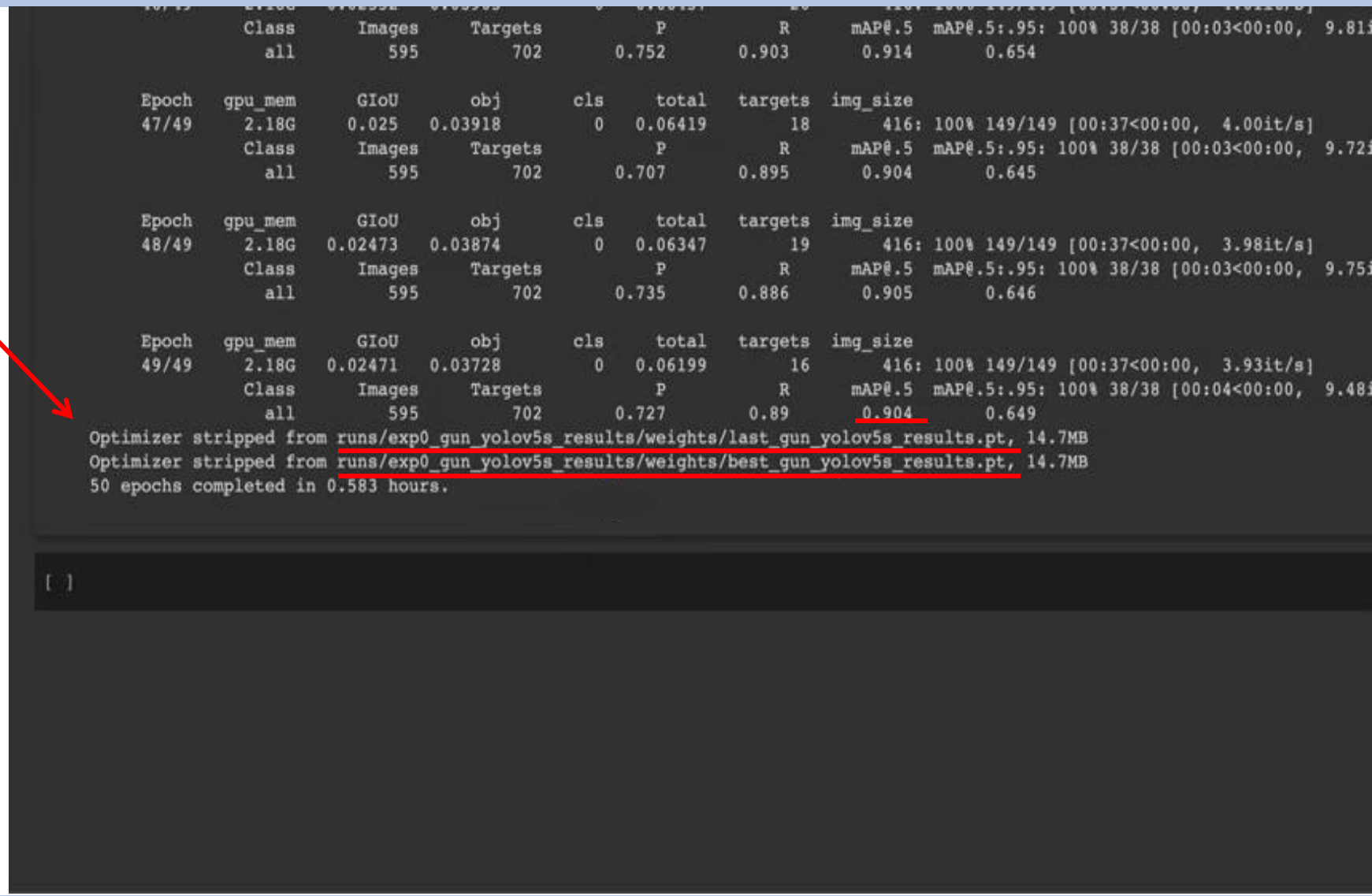
Train

```
100 14.4M    0 14.4M    0     0  7862k      0 --:--:--  0:00:01 --:--:--  261M
Downloading https://drive.google.com/uc?export=download&id=1R5T6rIyy3lLwgFXNms8whc-387H0tMQO as yolov5s.pt... Done (4.8s)
Scanning images: 100% 2376/2376 [00:00<00:00, 7681.61it/s]
Scanning labels /content/dataset/export/labels.cache (2376 found, 0 missing, 0 empty, 0 duplicate, for 2376 images): 100% 237
Scanning images: 100% 595/595 [00:00<00:00, 8824.77it/s]
Scanning labels /content/dataset/export/labels.cache (595 found, 0 missing, 0 empty, 0 duplicate, for 595 images): 100% 595/5

Analyzing anchors... Best Possible Recall (BPR) = 1.0000
Image sizes 416 train, 416 test
Using 2 dataloader workers
Starting training for 50 epochs...

     Epoch   gpu_mem       GIoU       obj       cls     total   targets  img_size
      0/49     2.18G     0.0871   0.06698         0    0.1541        15       416: 100% 149/149 [00:40<00:00,  3.63it/s]
               Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 38/38 [00:06<00:00,  6.13i
                 all         595         702       0.173       0.821       0.629       0.348

     Epoch   gpu_mem       GIoU       obj       cls     total   targets  img_size
      1/49     2.18G    0.05653   0.06028         0    0.1168        20       416: 100% 149/149 [00:38<00:00,  3.90it/s]
               Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 38/38 [00:04<00:00,  9.16i
                 all         595         702       0.348       0.806       0.589       0.329

     Epoch   gpu_mem       GIoU       obj       cls     total   targets  img_size
      2/49     2.18G    0.05033    0.0584         0    0.1087        33       416:   1% 2/149 [00:00<01:16,  1.91it/s]
```

```
[ ]
```

Run YOLOv5 in Colab

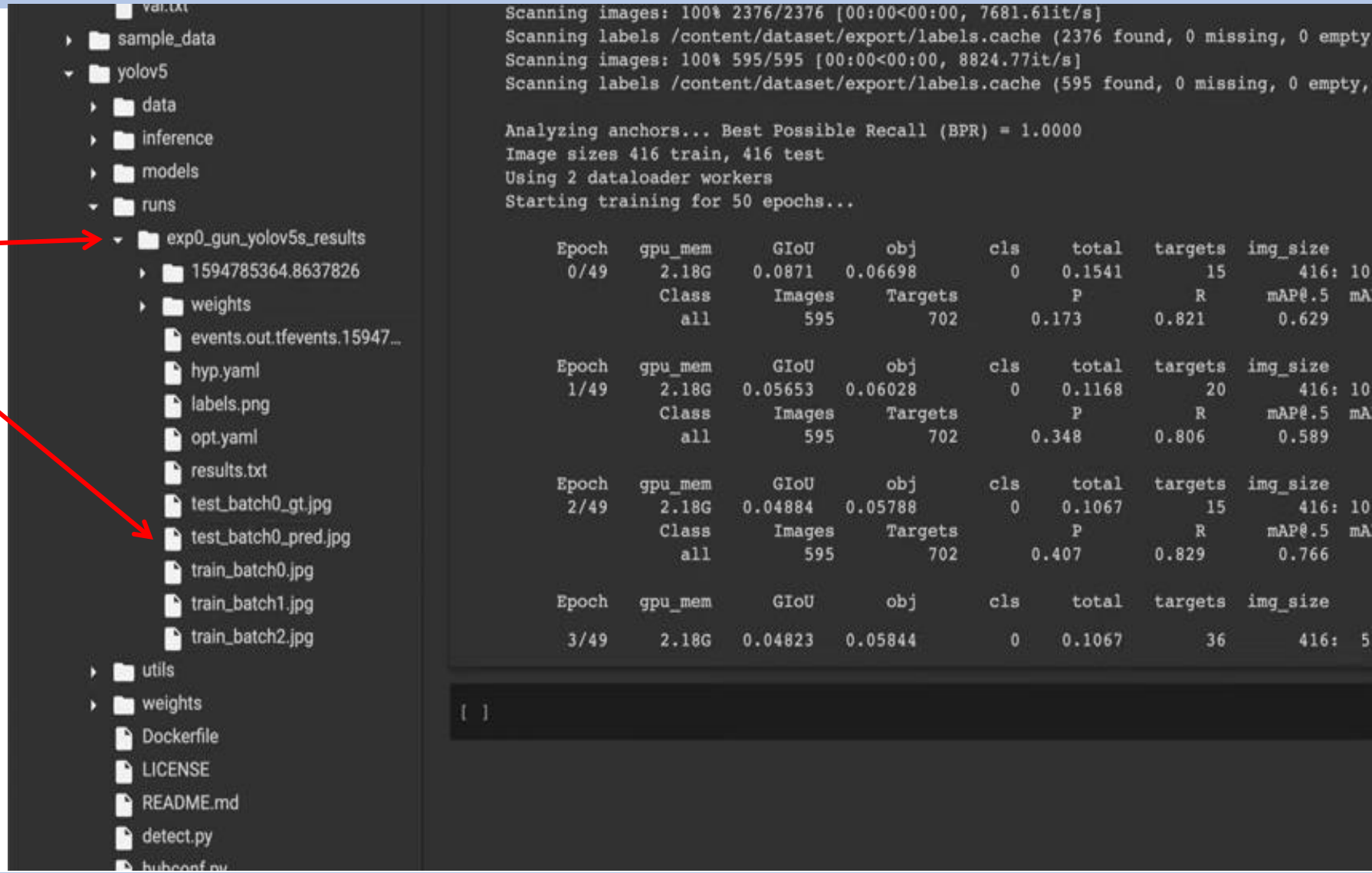
Train



Run YOLOv5 in Colab

Train

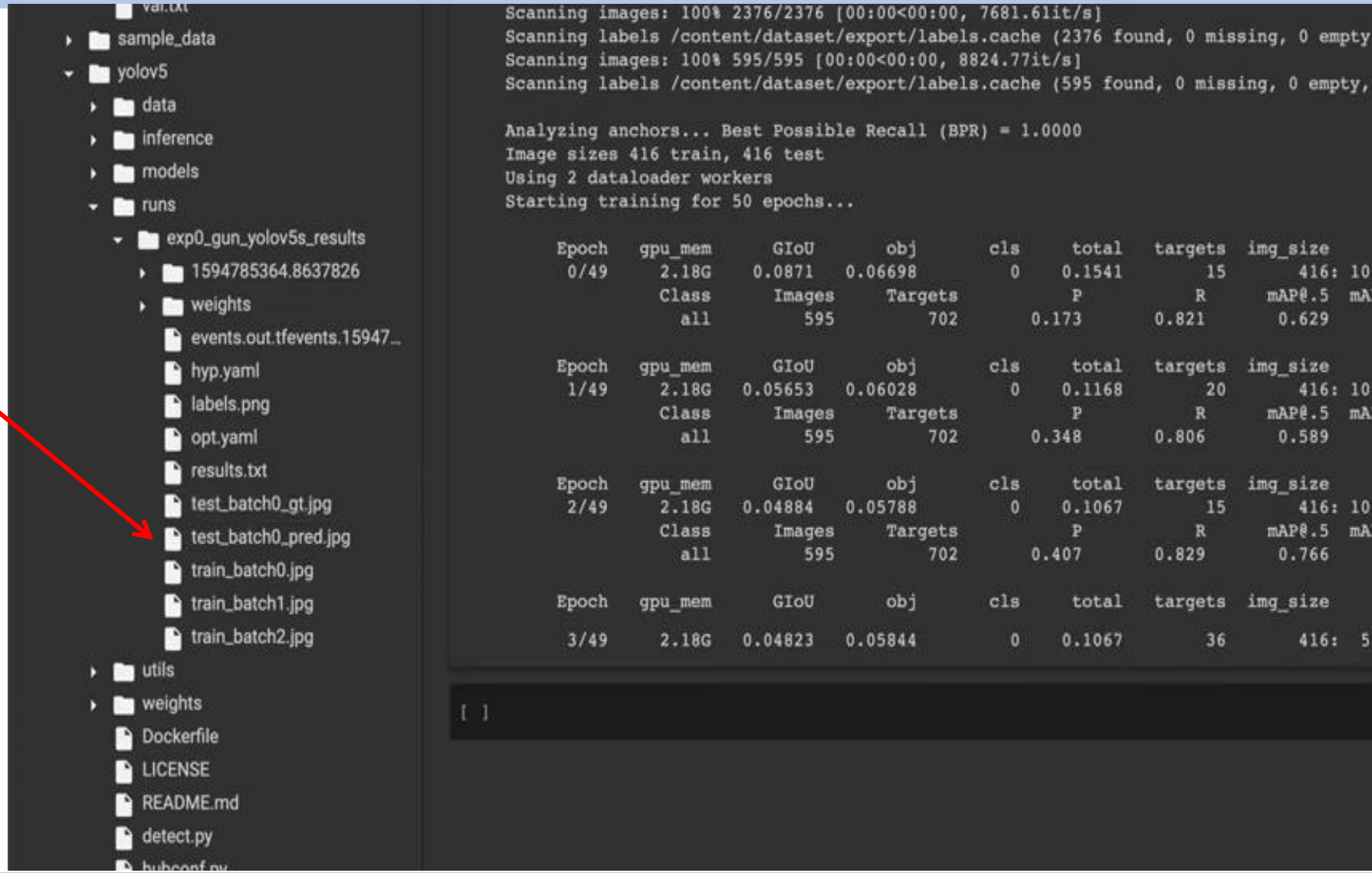
Check output



Run YOLOv5 in Colab

Train

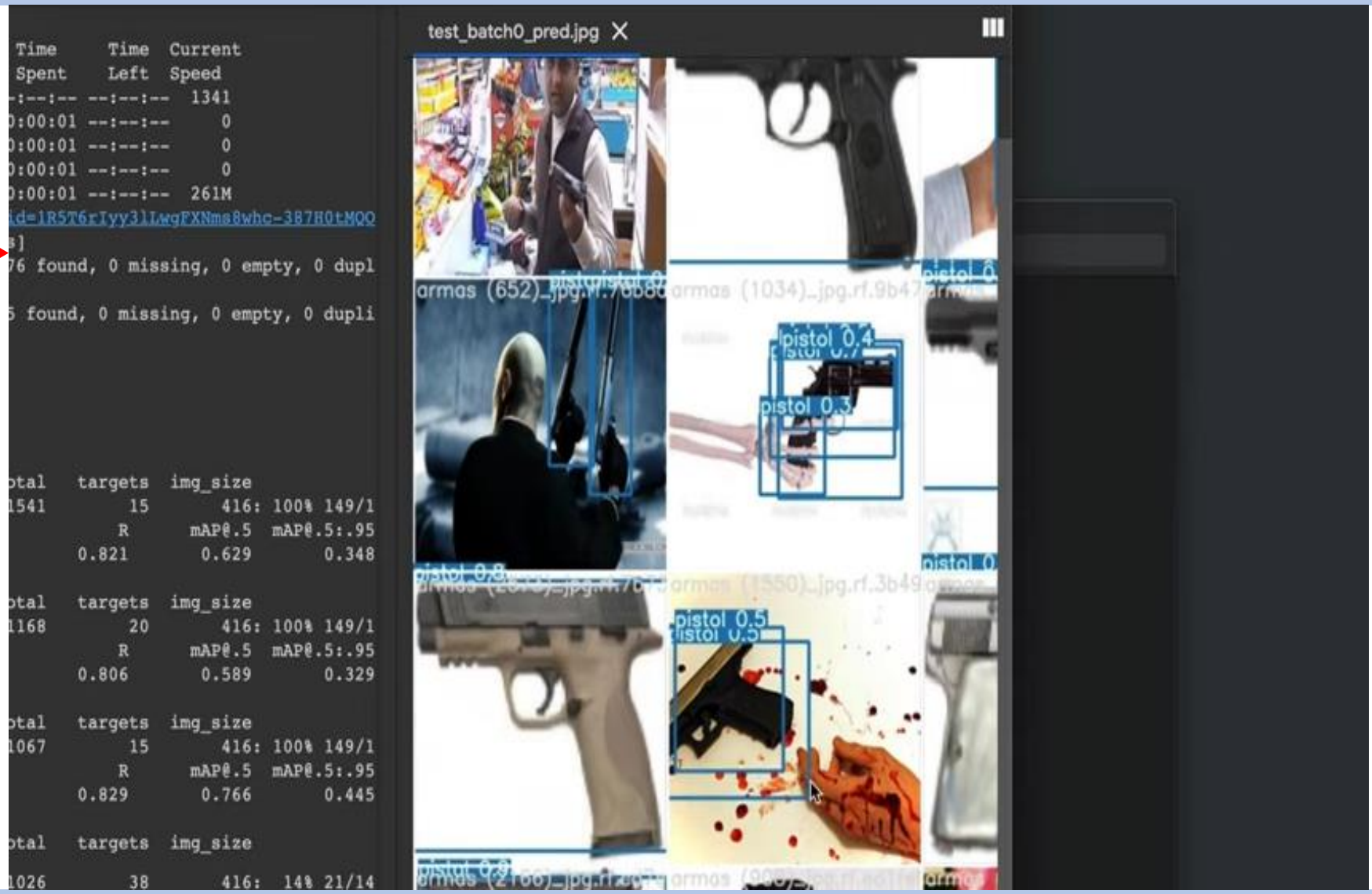
Check output



Run YOLOv5 in Colab

Train

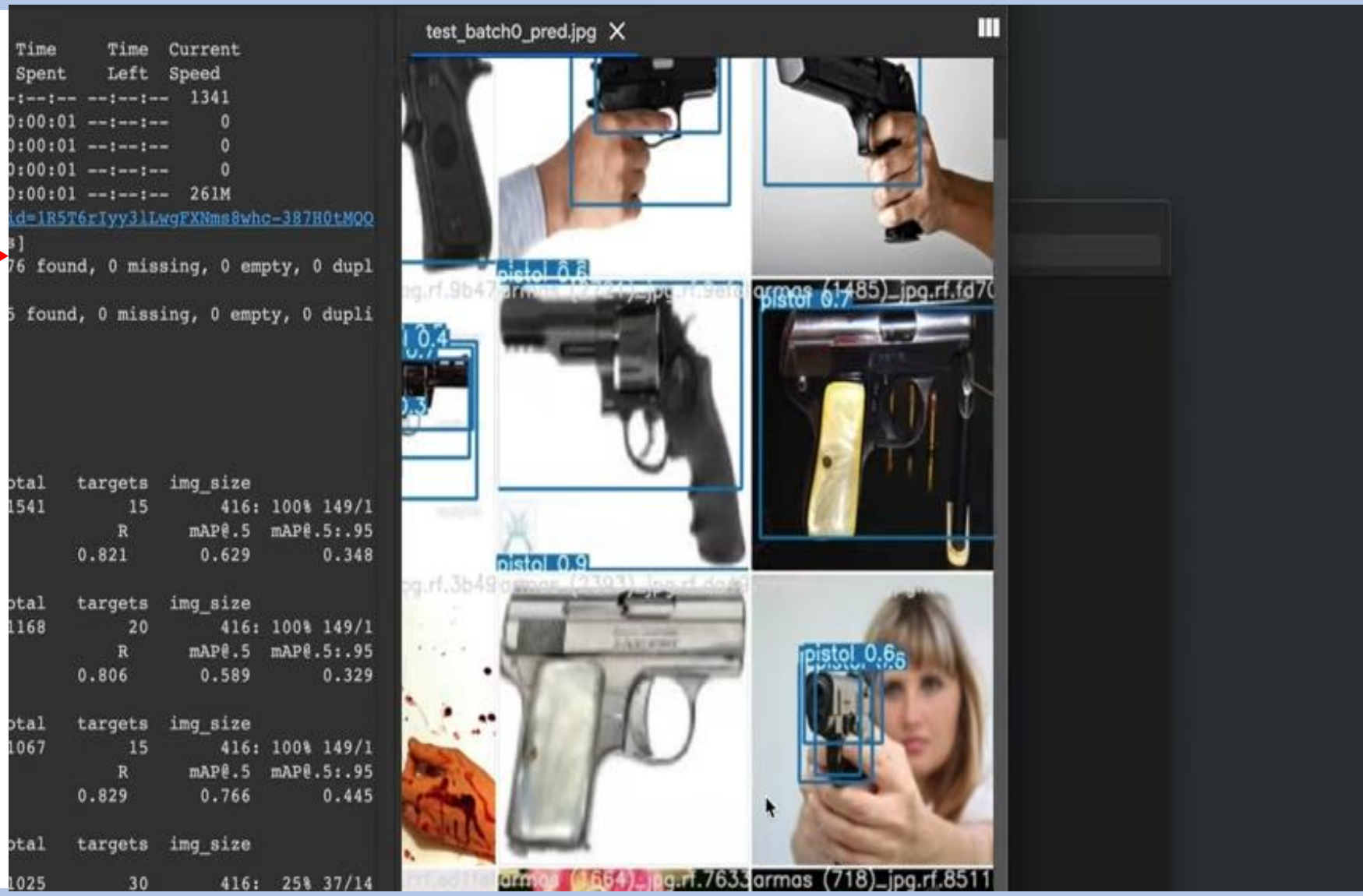
Check output



Run YOLOv5 in Colab

Train

Check output

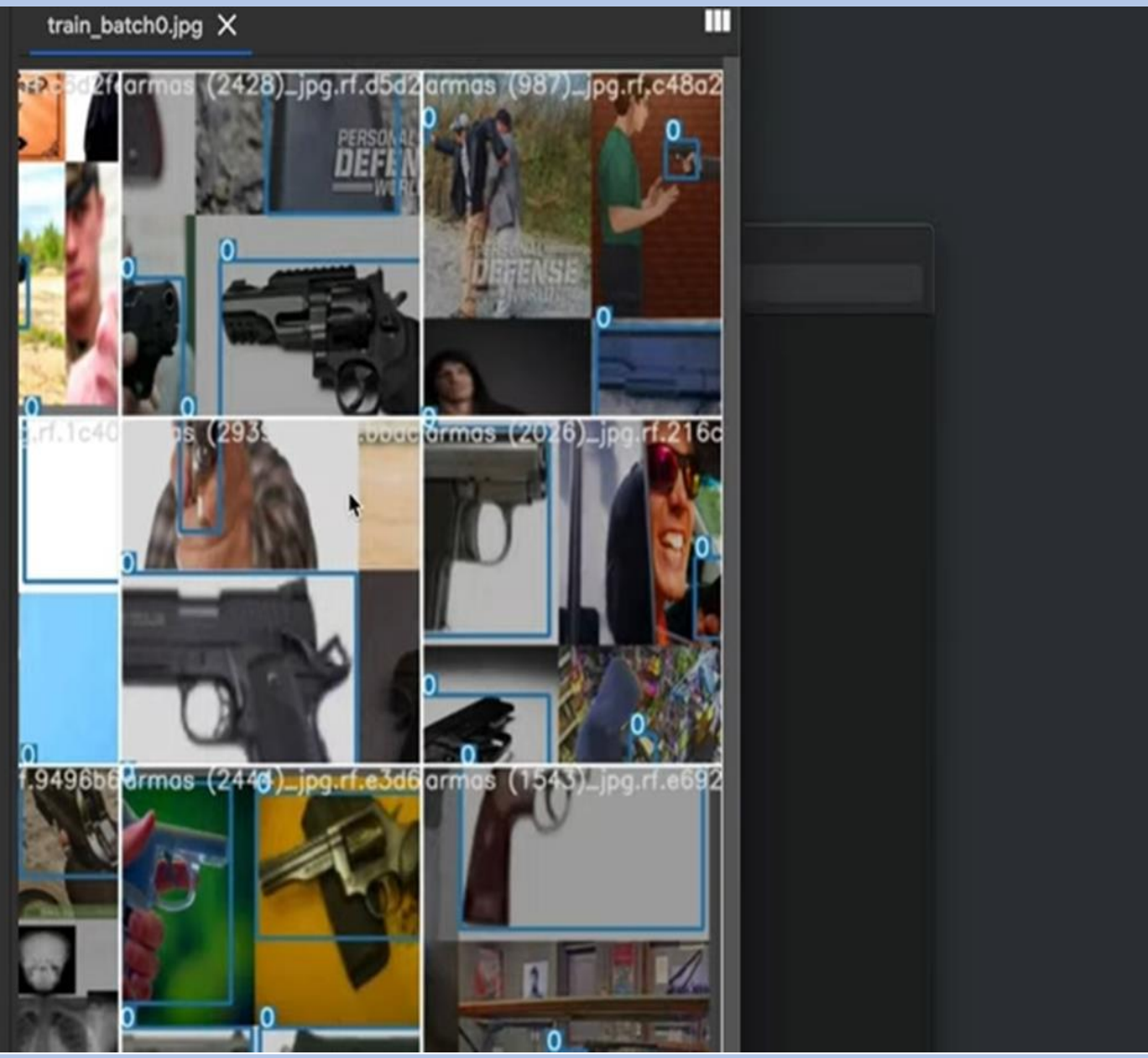


Run YOLOv5 in Colab

Train

Check output

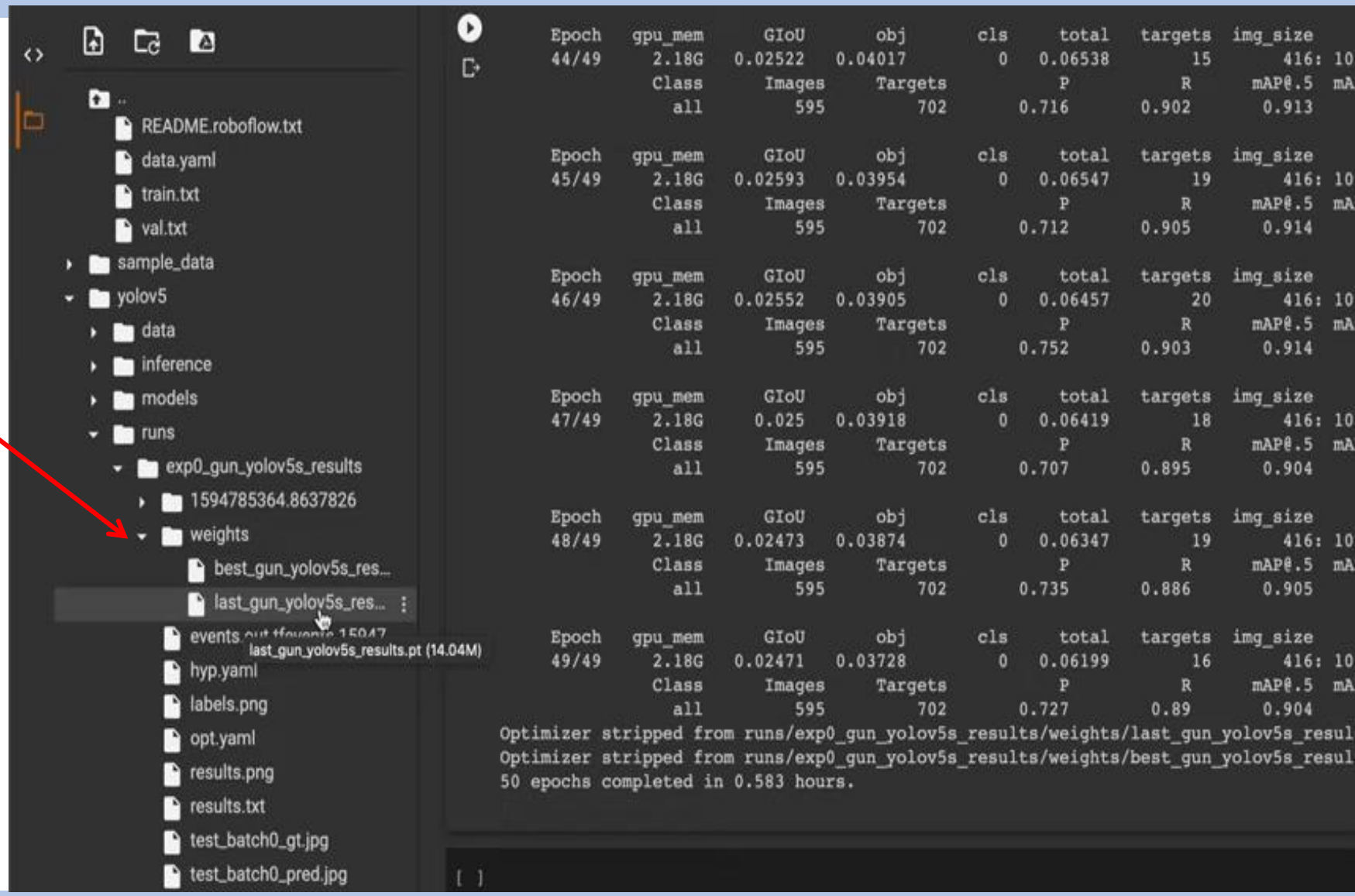
```
  Time      Time  Current
  Spent     Left  Speed
--:--:-- --:--:--  1341
 0:00:01 --:--:--     0
 0:00:01 --:--:--     0
 0:00:01 --:--:--     0
 0:00:01 --:--:--  261M
&id=1R5T6rIyy3lLwgFXNms8whc-387H0tMQO
/s]
376 found, 0 missing, 0 empty, 0 dupl
]
95 found, 0 missing, 0 empty, 0 dupli

0



total   targets  img_size
.1541        15       416: 100% 149/1
P           R      mAP@.5  mAP@.5:.95
3       0.821       0.629       0.348

total   targets  img_size
.1168        20       416: 100% 149/1
P           R      mAP@.5  mAP@.5:.95
8       0.806       0.589       0.329

total   targets  img_size
.1067        15       416: 100% 149/1
P           R      mAP@.5  mAP@.5:.95
7       0.829       0.766       0.445

total   targets  img_size

.1088        28       416:  87% 129/1
```



Run YOLOv5 in Colab

Train

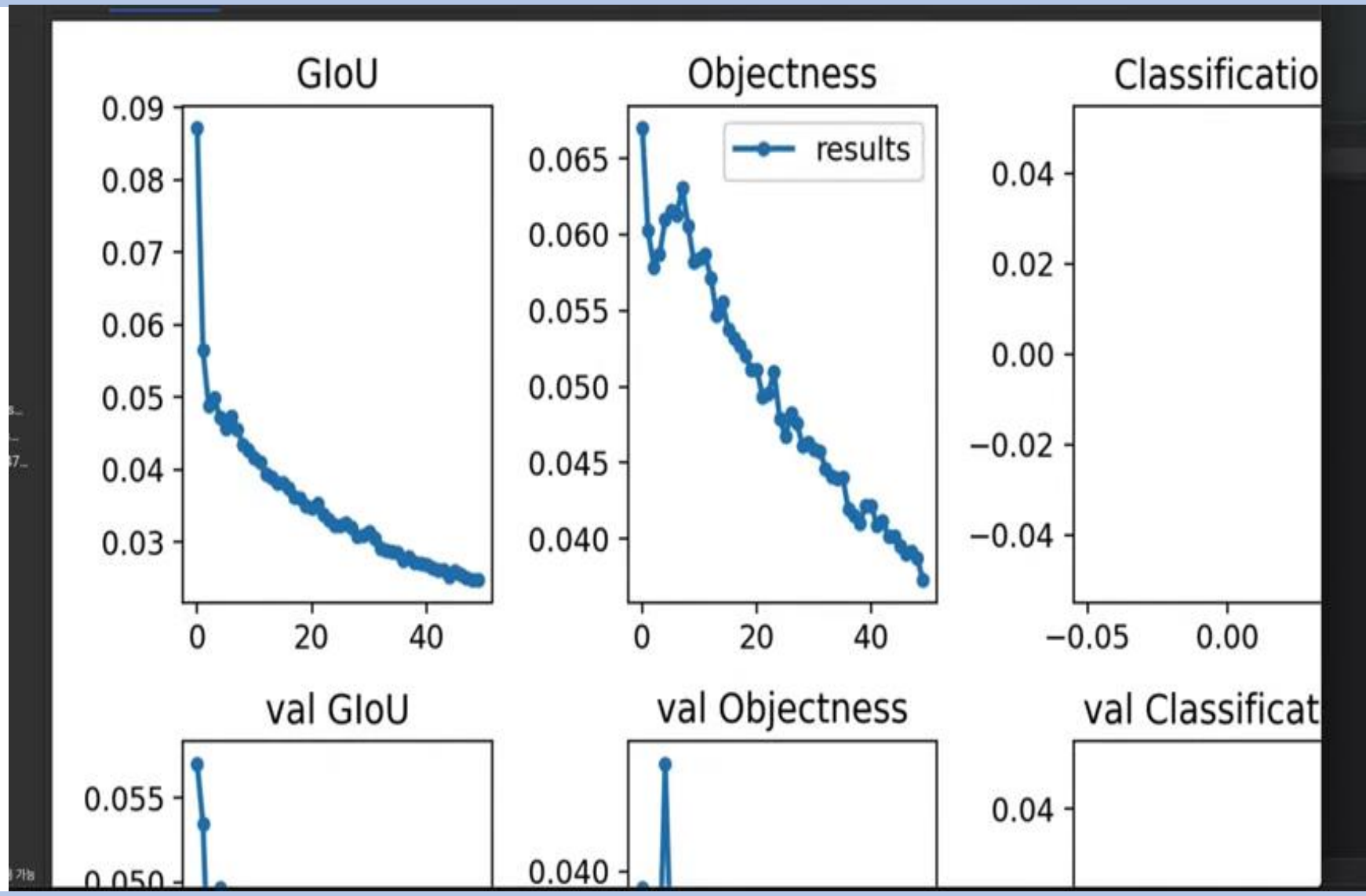
Check output



Run YOLOv5 in Colab

Train

Check output →



Run YOLOv5 in Colab

Train

Check output

Run YOLOv5 in Colab

Use tensorboard for checking the data

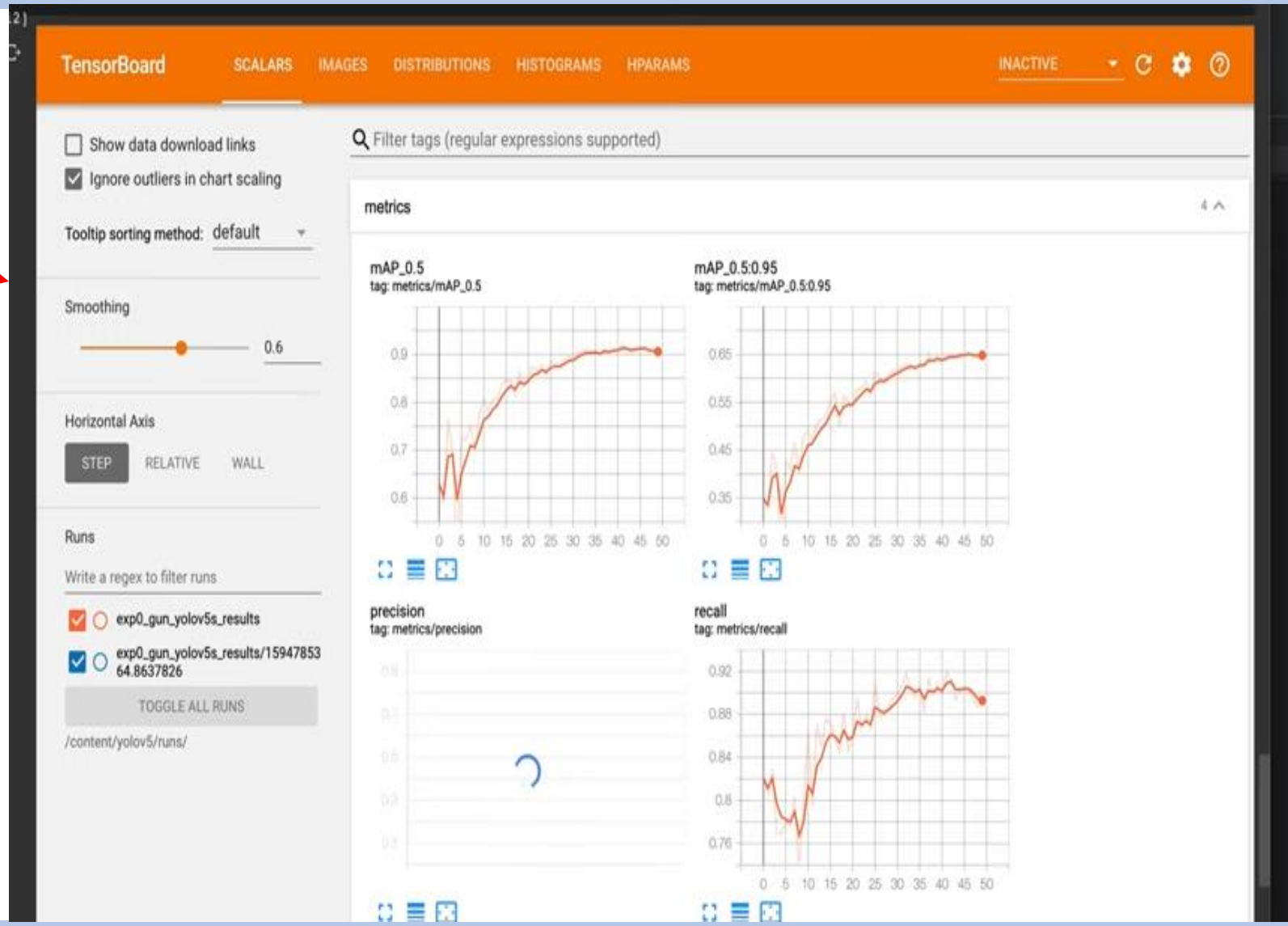
```
      Epoch   gpu_mem       GIoU       obj       cls     total   targets  img_size
      49/49     2.18G    0.02471   0.03728         0   0.06199        16       416: 100% 149/149 [00:37<00:00,  3.93it/s]
                Class      Images     Targets           P           R      mAP@.5  mAP@.5:.95: 100% 38/38 [00:04<00:00,  9.48i
                  all         595         702       0.727        0.89       0.904       0.649
Optimizer stripped from runs/exp0_gun_yolov5s_results/weights/last_gun_yolov5s_results.pt, 14.7MB
Optimizer stripped from runs/exp0_gun_yolov5s_results/weights/best_gun_yolov5s_results.pt, 14.7MB
50 epochs completed in 0.583 hours.
```

```
%load_ext tensorboard
%tensorboard --logdir /content/yolov5/runs/
```

```
... Launching TensorBoard...
```

Run YOLOv5 in Colab

Use tensorboard for checking the data



Done !!!

Assignment_6 This is the LAST assignment

숭실대학교 Soongsil University

- **Make your own YOLO model !**
  - ✓ Any YOLO
  - ✓ Due to **1000am, May 9 2023** **(before the class)**
  - ✓ **Upload as PDF file and video clip on LMS only** (don't need hardcopy)

You can't learn anything if you don't do it yourself !!!