AI VIETNAM
All-in-One Course
(TA Session)

Object Detection with YOLOv8

Project

Dinh-Thang Duong – TA

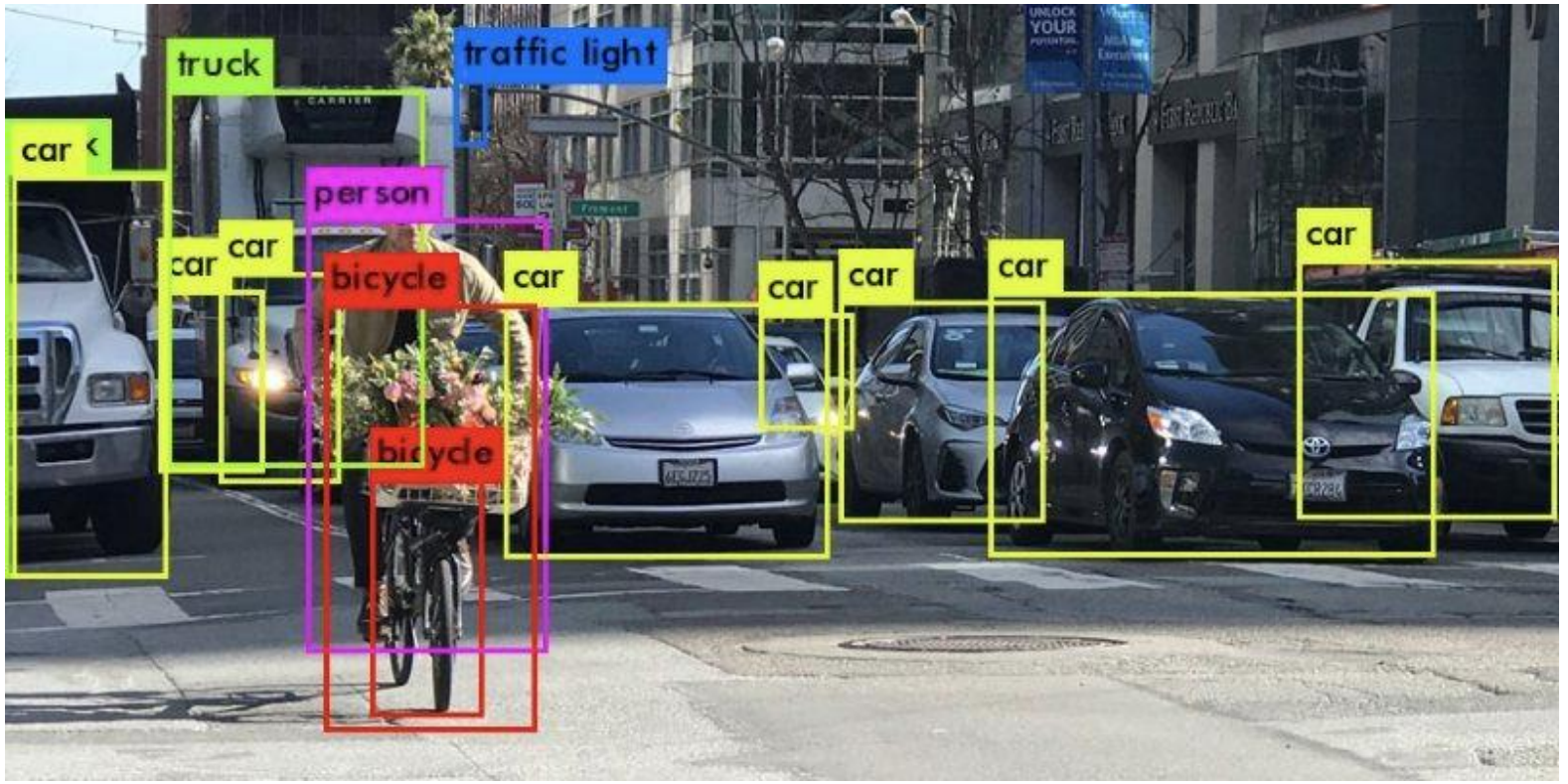
Year 2023

Outline

- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

Introduction

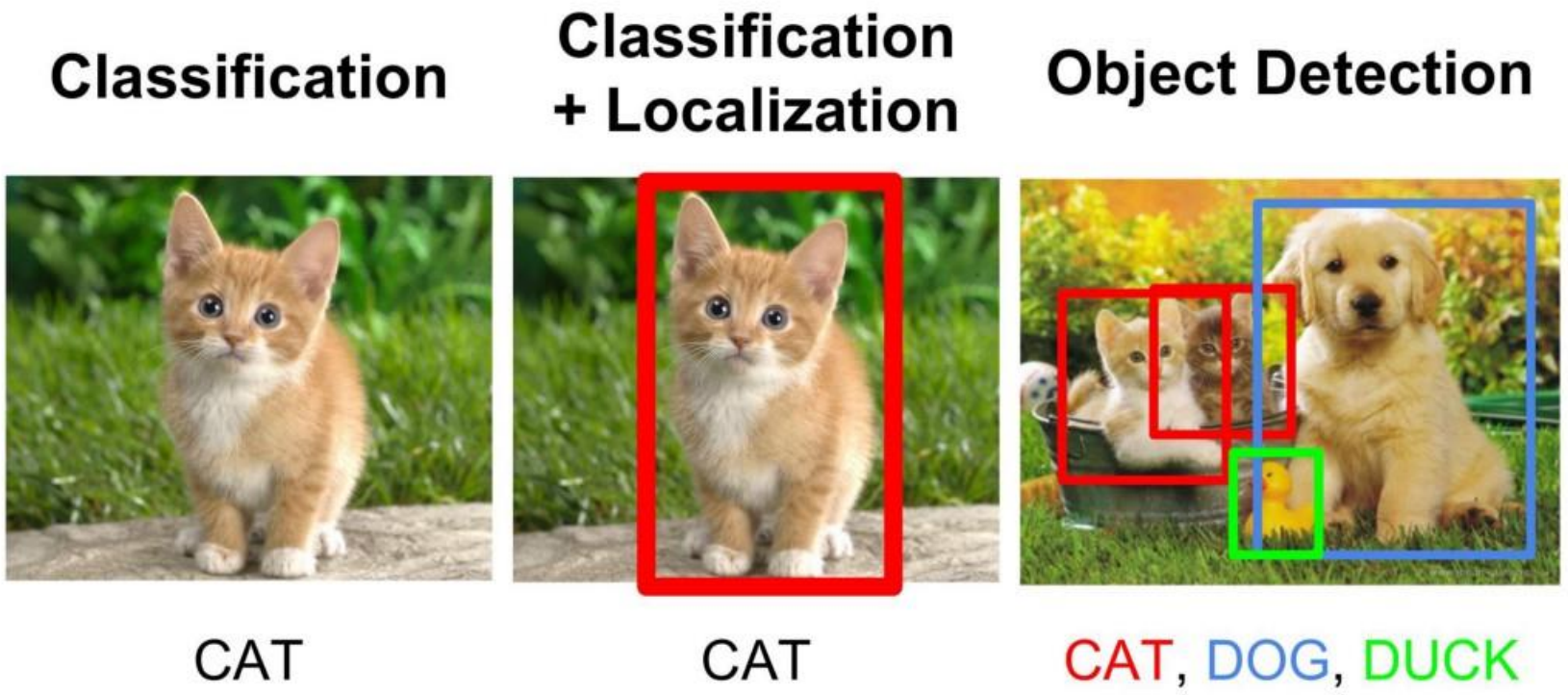
❖ Getting Started



Object Detection: a Computer Vision task for identifying objects in an image.

Introduction

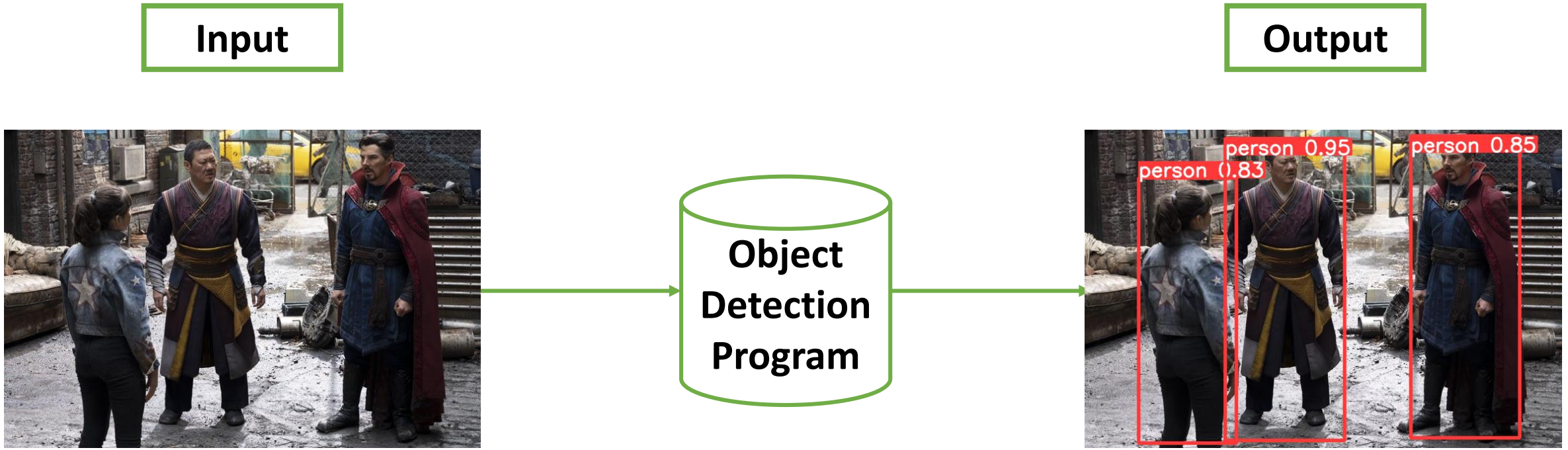
❖ What is Object Detection?



Object Detection = Classification + Localization

Introduction

❖ Human Detection Problem



Introduction

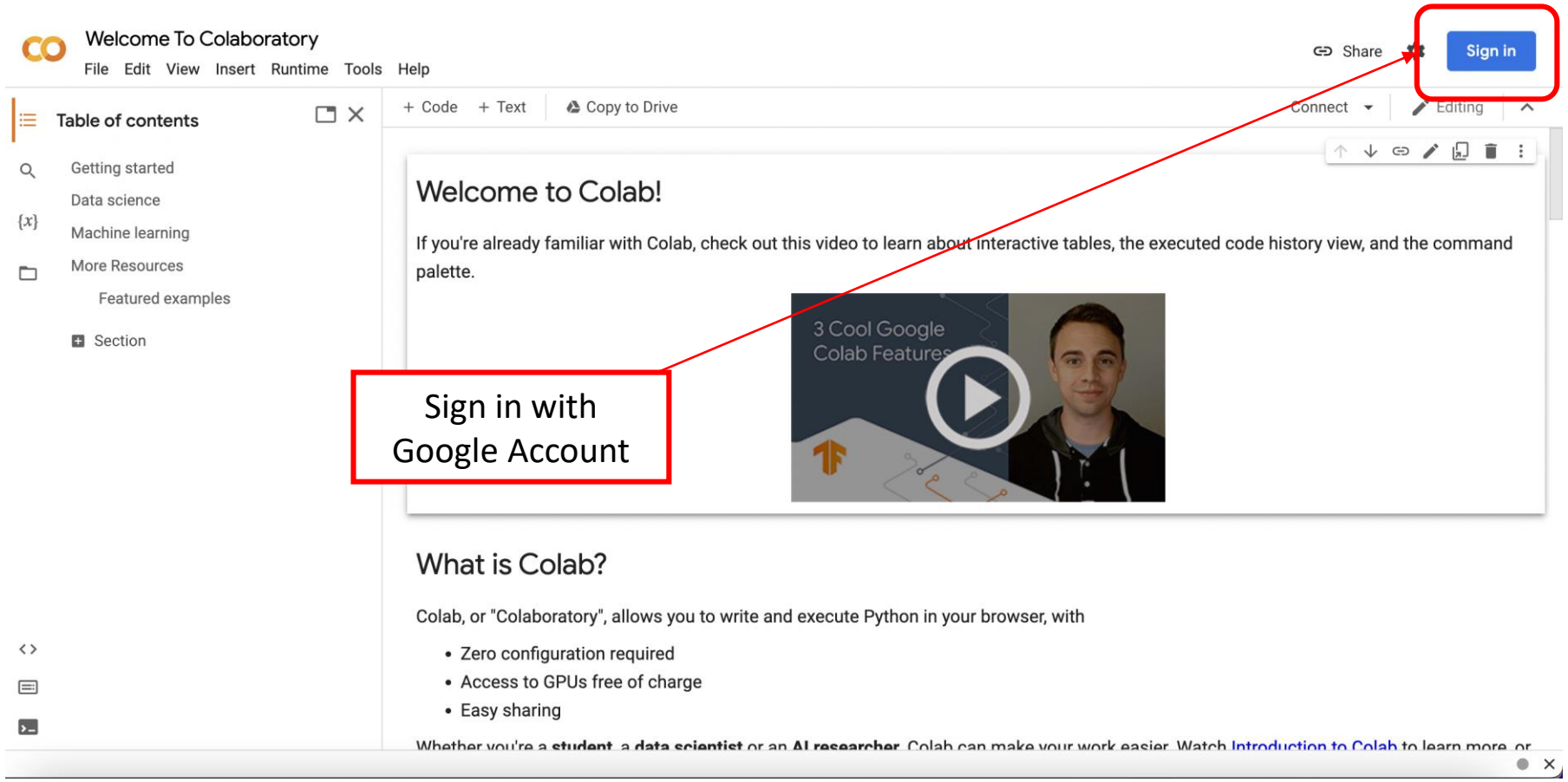
❖ Human Detection Demo

Outline

- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

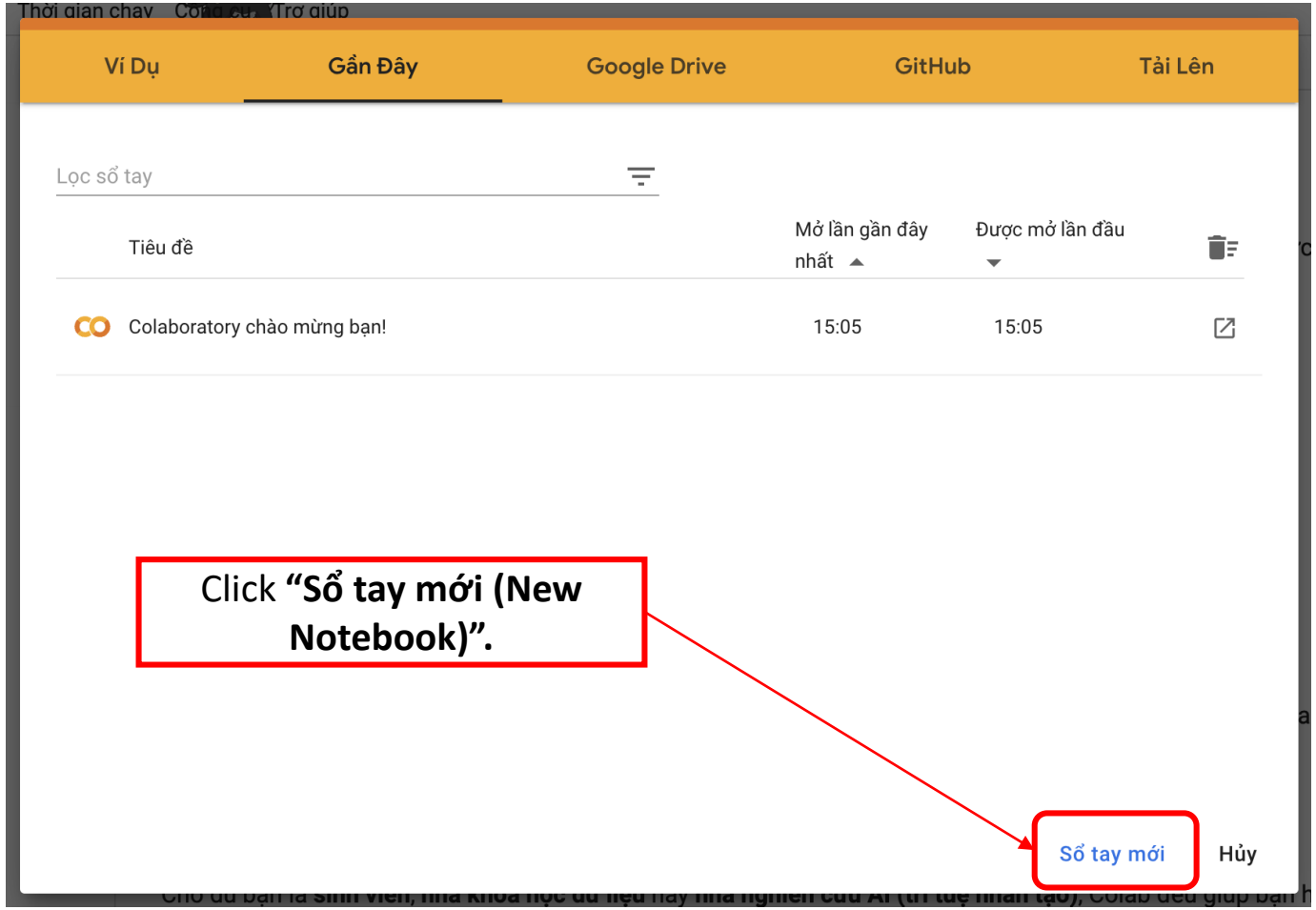
Code Environment

❖ Google Colab instruction



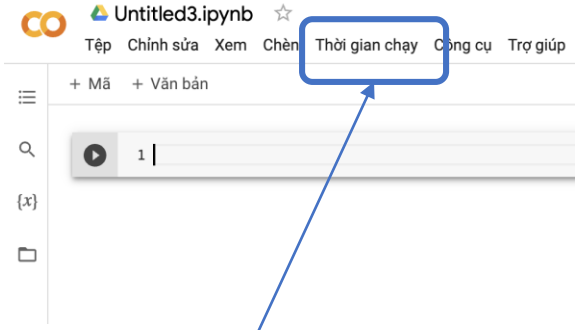
Code Environment

❖ Google Colab instruction

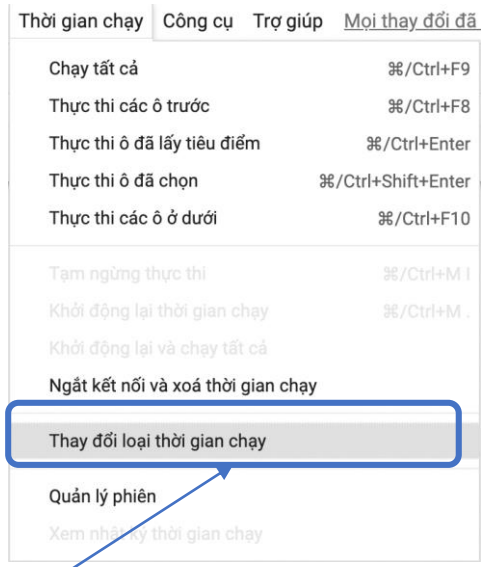


Code Environment

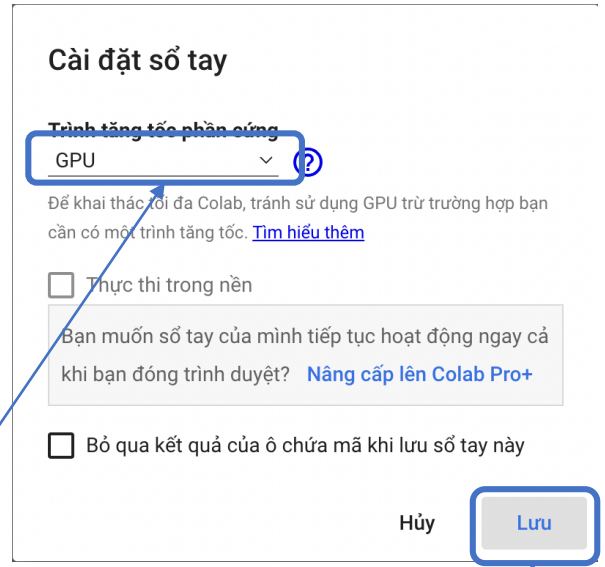
❖ Google Colab instruction



1. Click **"Thời gian chạy (Runtime)".**



2. Click **"Thay đổi loại thời gian chạy (Change Runtime Type)".**

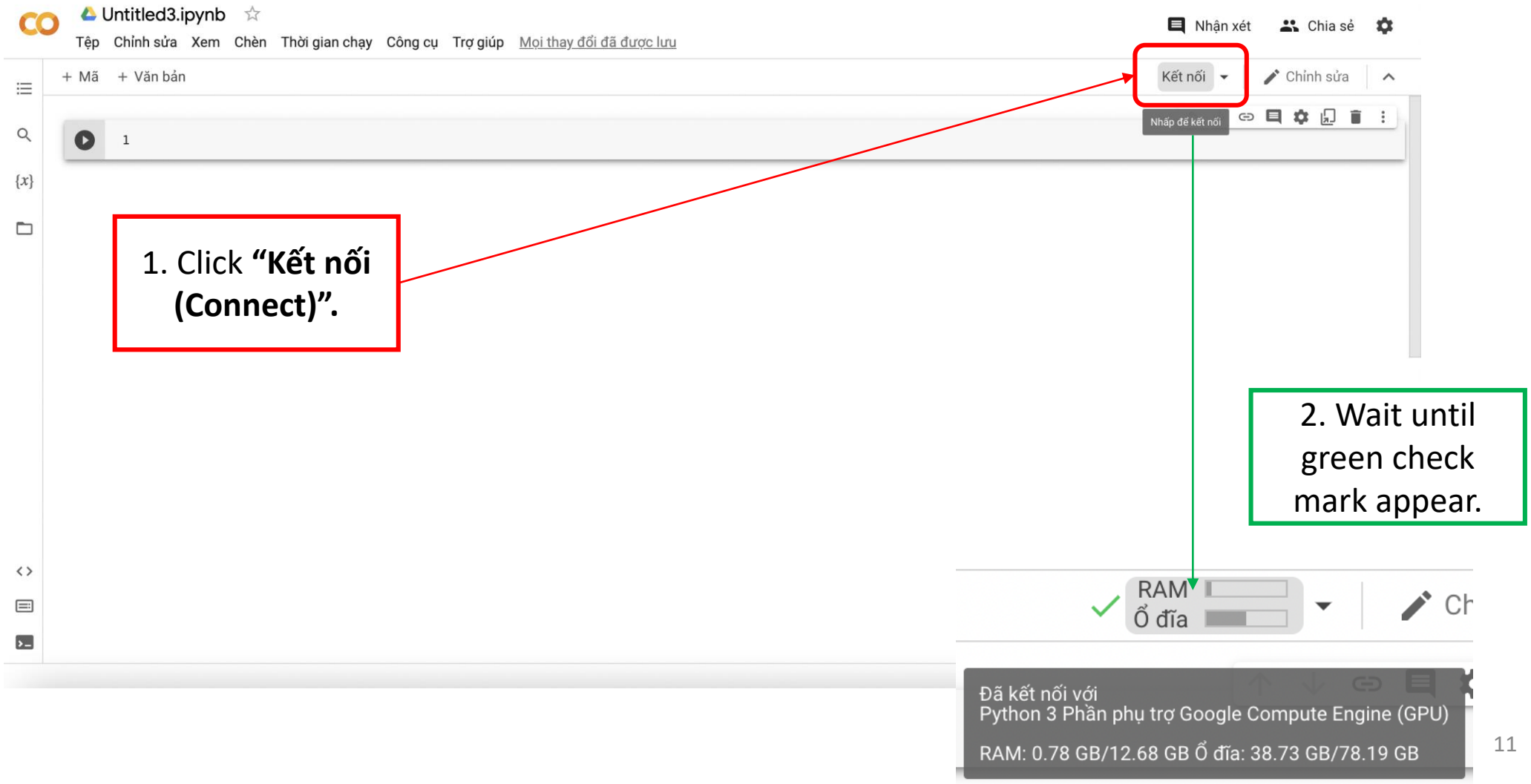


3. Click to change from **None to GPU**.

4. Click **"Lưu (Save)".**

Code Environment

❖ Google Colab instruction



Outline

- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

YOLOv8 – How to use

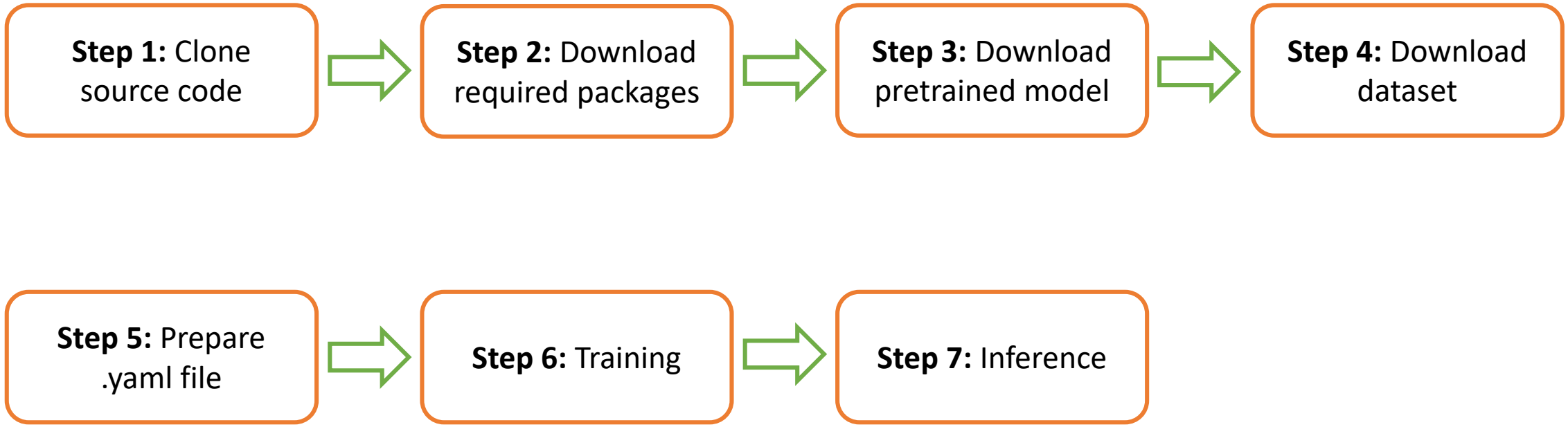
❖ Introduction

YOLOv8: A family of object detection architectures and models pretrained on the COCO dataset

source: https://github.com/ultralytics/ultralytics

YOLOv8 – How to use

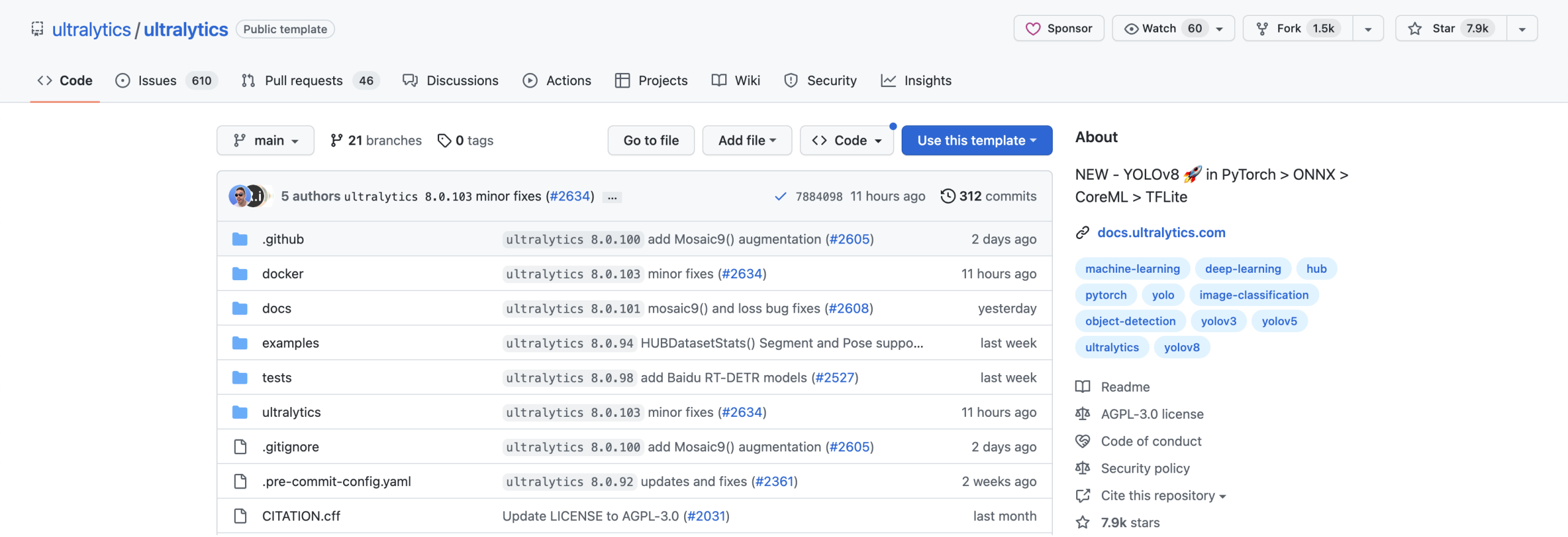
❖ YOLOv8 Installation and Usage



source: https://github.com/ultralytics/ultralytics

YOLOv8 – How to use

❖ Step 1: Clone YOLOv8 source code

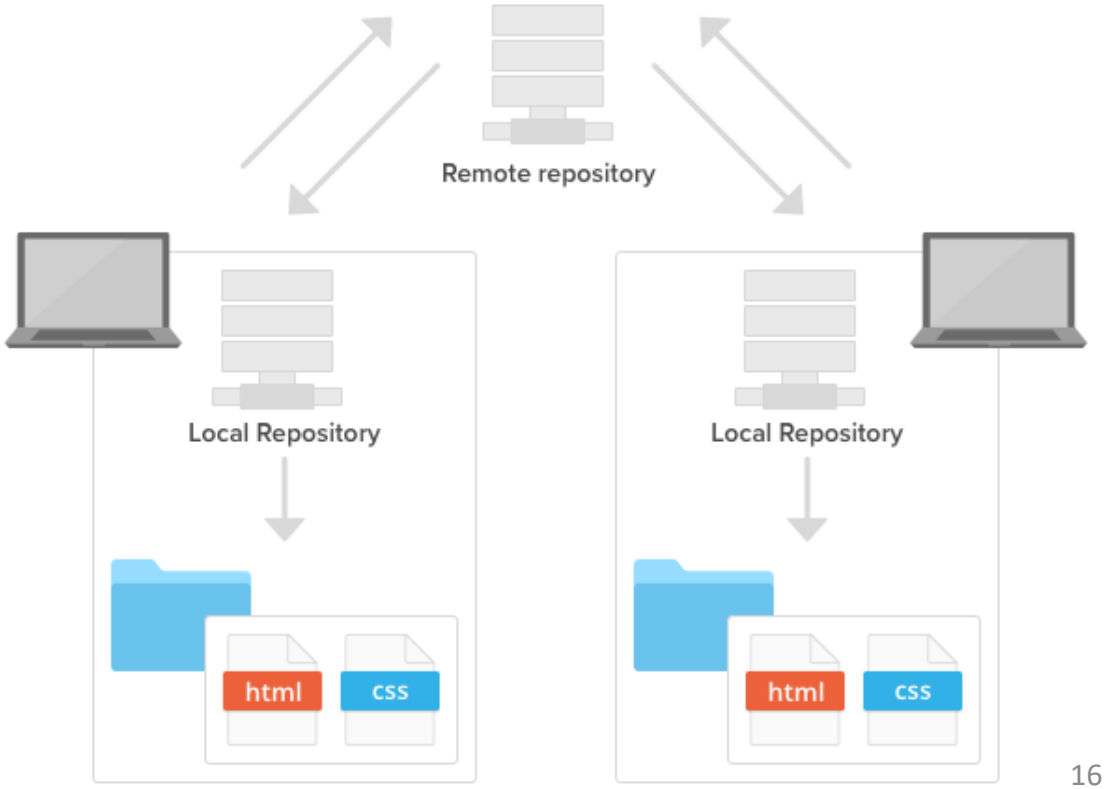


YOLOv8 GitHub Repository

source: https://github.com/ultralytics/ultralytics

YOLOv8 – How to use

❖ Step 1.1: About GitHub



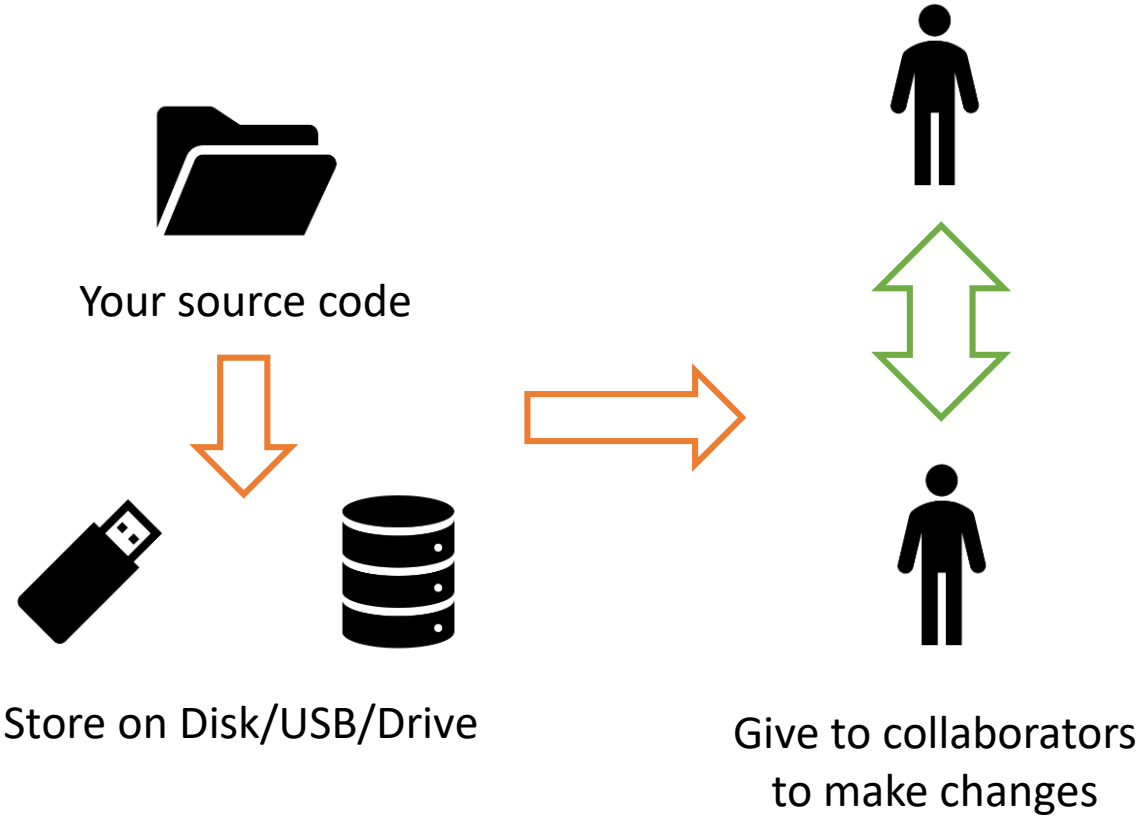
A code hosting platform for collaboration and version control

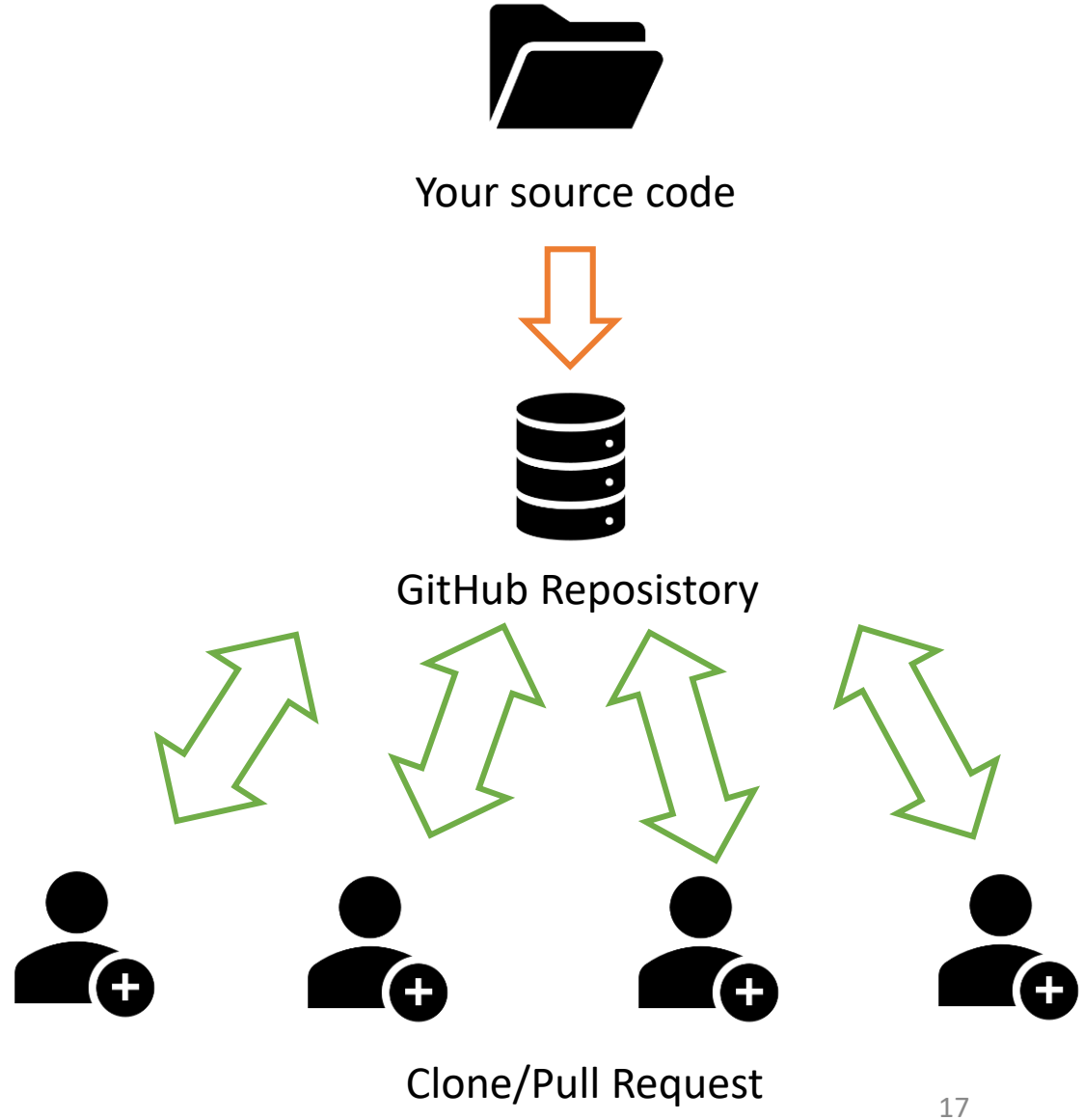
GitHub usages:

1. Store your code
2. Work together
3. Find source code from other people

YOLOv8 – How to use

❖ Step 1.1: About GitHub

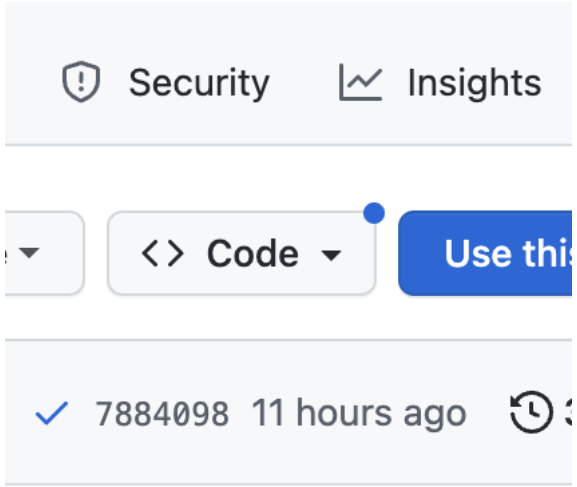




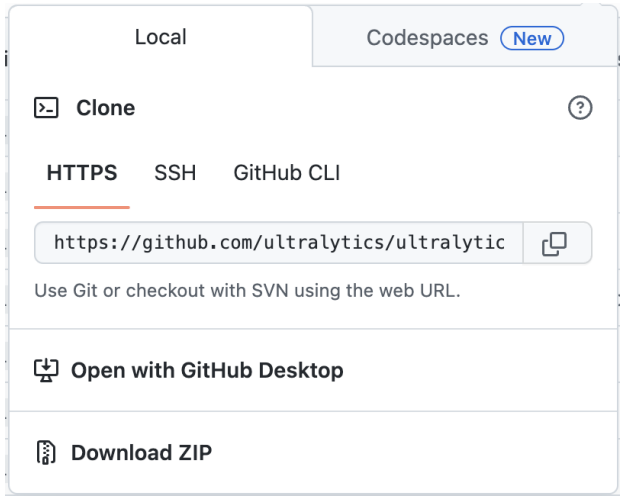
YOLOv8 – How to use

❖ Step 1: Clone YOLOv8 source code

Step 1: Click **Code**



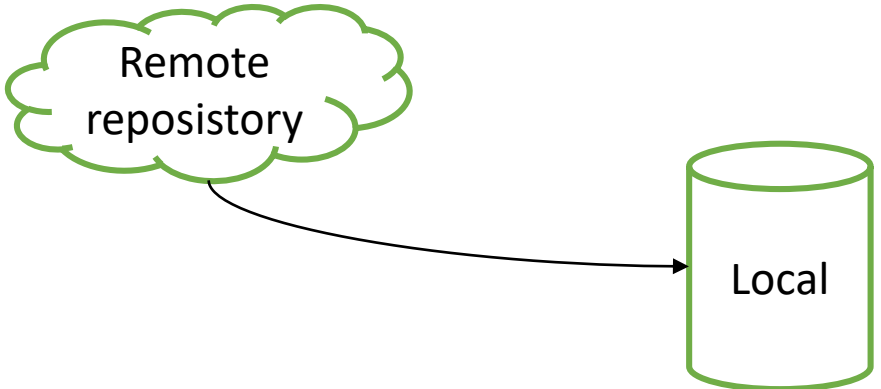
Step 2: Copy link



Step 3: Clone source code

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

```
Cloning into 'ultralytics'...
remote: Enumerating objects: 8224, done.
remote: Counting objects: 100% (337/337), done.
remote: Compressing objects: 100% (194/194), done.
remote: Total 8224 (delta 198), reused 241 (delta 143), pack-reused 7887
Receiving objects: 100% (8224/8224), 5.88 MiB | 13.14 MiB/s, done.
Resolving deltas: 100% (5516/5516), done.
```



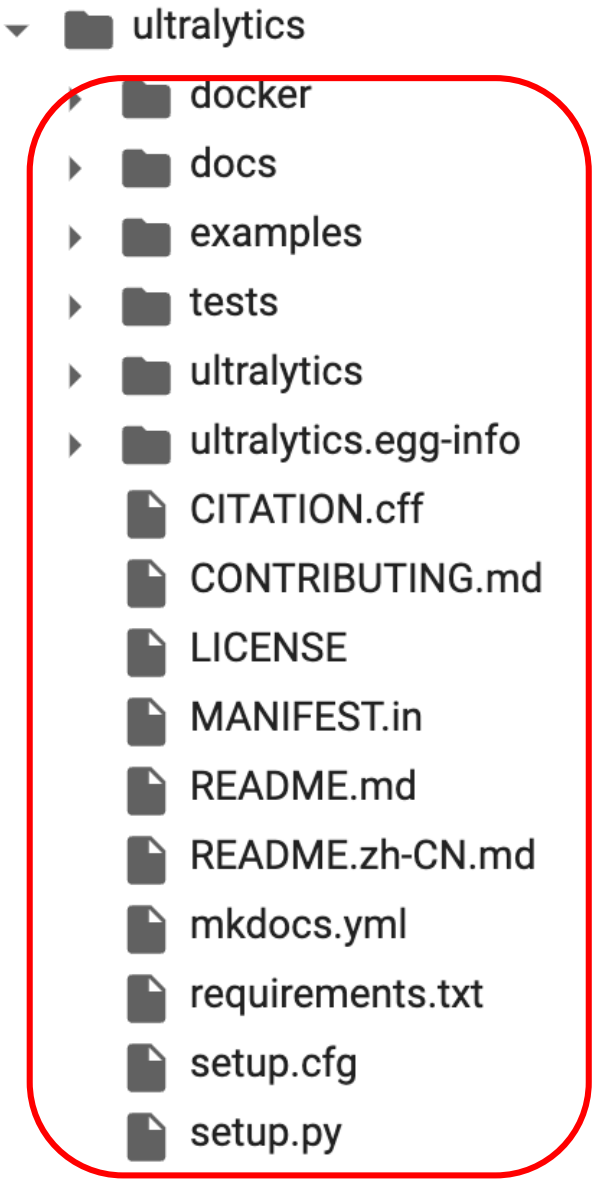
***Clone source code command:**

`git clone <link>:` Clone (copy) a repository into a new directory.

YOLOv8 – How to use

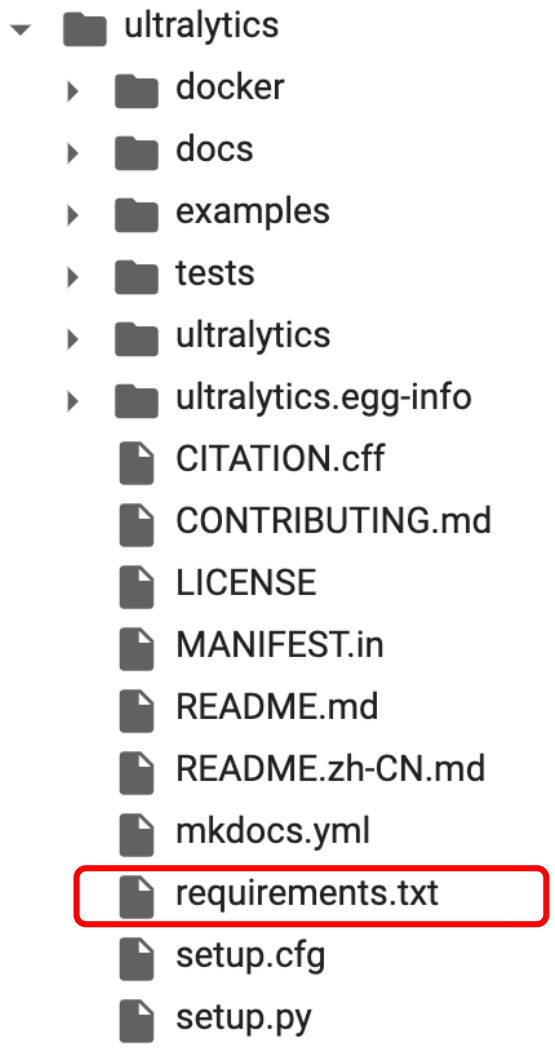
❖ Step 1: Clone YOLOv8 source code

.github	`ultralytics 8.0.100` add Mosaic9() augmentation (#2605)	2 days ago
docker	`ultralytics 8.0.103` minor fixes (#2634)	11 hours ago
docs	`ultralytics 8.0.101` mosaic9() and loss bug fixes (#2608)	yesterday
examples	`ultralytics 8.0.94` HUBDatasetStats() Segment and Pose suppo...	last week
tests	`ultralytics 8.0.98` add Baidu RT-DETR models (#2527)	last week
ultralytics	`ultralytics 8.0.103` minor fixes (#2634)	11 hours ago
.gitignore	`ultralytics 8.0.100` add Mosaic9() augmentation (#2605)	2 days ago
.pre-commit-config.yaml	`ultralytics 8.0.92` updates and fixes (#2361)	2 weeks ago
CITATION.cff	Update LICENSE to AGPL-3.0 (#2031)	last month
CONTRIBUTING.md	`ultralytics 8.0.91` tracker fix and docs comments (#2343)	2 weeks ago
LICENSE	Update LICENSE to AGPL-3.0 (#2031)	last month
MANIFEST.in	`ultralytics 8.0.51` add assets and CI actions (#1296)	2 months ago
README.md	`ultralytics 8.0.90` actions and docs improvements (#2326)	2 weeks ago
README.zh-CN.md	`ultralytics 8.0.90` actions and docs improvements (#2326)	2 weeks ago
mkdocs.yml	`ultralytics 8.0.99` HUB resume fix and Docs updates (#2567)	4 days ago
requirements.txt	`ultralytics 8.0.100` add Mosaic9() augmentation (#2605)	2 days ago
setup.cfg	`ultralytics 8.0.47` Docker and reformat updates (#1153)	3 months ago
setup.py	`ultralytics 8.0.99` HUB resume fix and Docs updates (#2567)	4 days ago



YOLOv8 – How to use

❖ Step 2: Download required packages



requirements.txt

```
# Ultralytics requirements
# Usage: pip install -r requirements.txt

# Base ----------------------------------------
matplotlib>=3.2.2
opencv-python>=4.6.0
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.64.0

# Logging -------------------------------------
# tensorboard>=2.13.0
# clearml
# comet

# Plotting ------------------------------------
pandas>=1.1.4
seaborn>=0.11.0
```

Download all thses packages to be able to run YOLOv8

YOLOv8 – How to use

❖ Step 2: Download required packages (Approach 1)

```
1 !pip install ultralytics
2 import ultralytics
3
4 ultralytics.checks()
```

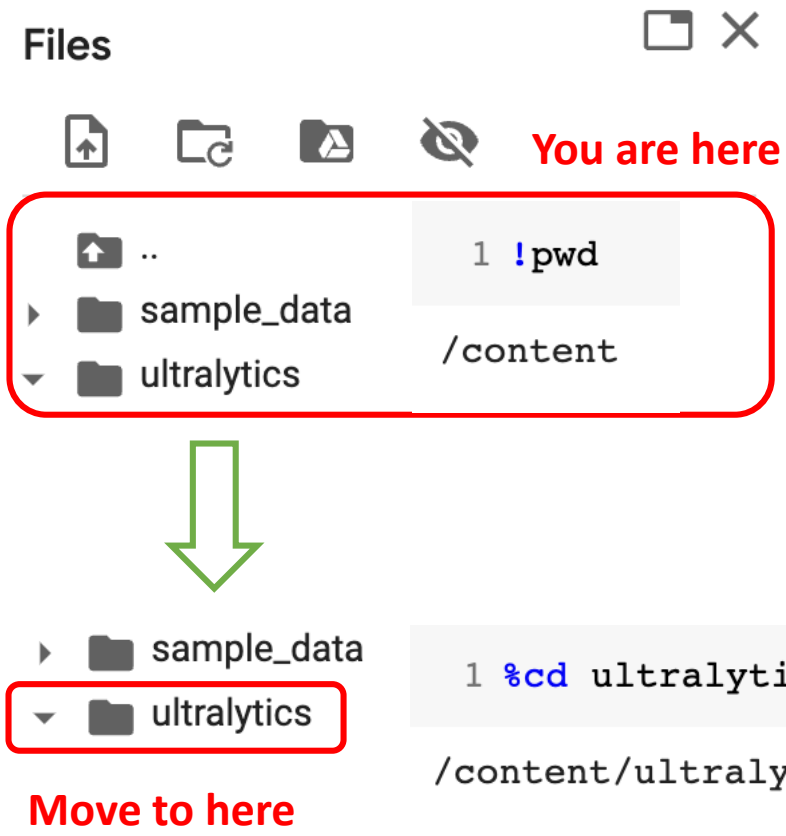
```
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 28.5/78.2 GB disk)
```

Simply install **ultralytics** packages by pip command

YOLOv8 – How to use

❖ Step 2: Download required packages (Approach 2)

Step 1: Move to YOLOv8



Step 2: Install packages

```
1 !pip install -e .
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Obtaining file:///content/ultralytics
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib>=3.2.2 in /usr/local/lib/python3.10/dist-packages (from ult
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from ultraly
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from ultraly
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from ultra
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from ultralyt
Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from ultralyt
Requirement already satisfied: torchvision>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from ul
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralyt
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from ultraly
```

Full commands:

```
1 %cd ultralytics
2 !pip install -e .
```

YOLOv8 – How to use

❖ Step 3: Download pretrained models

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Pretrained models: models that have been trained on a large dataset

YOLOv8 – How to use

❖ Step 3: Download pretrained models

Step 1: Right click and copy link address

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Step 2: Download file using wget command

```
1 !wget https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8s.pt
```

```
--2023-05-17 10:14:55--  https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8s.pt
Resolving github.com (github.com)... 192.30.255.112
Connecting to github.com (github.com)|192.30.255.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
```

Command : `wget <link>`

- CONTRIBUTING.md
- LICENSE
- MANIFEST.in
- README.md
- README.zh-CN.md
- mkdocs.yml
- requirements.txt
- setup.cfg
- setup.py
- yolov8n.pt
- yolov8s.pt

When execution complete, pretrained file will be placed in ultralytics folder.

source: https://github.com/ultralytics/ultralytics

YOLOv8 – How to use

❖ Step 4: Download dataset

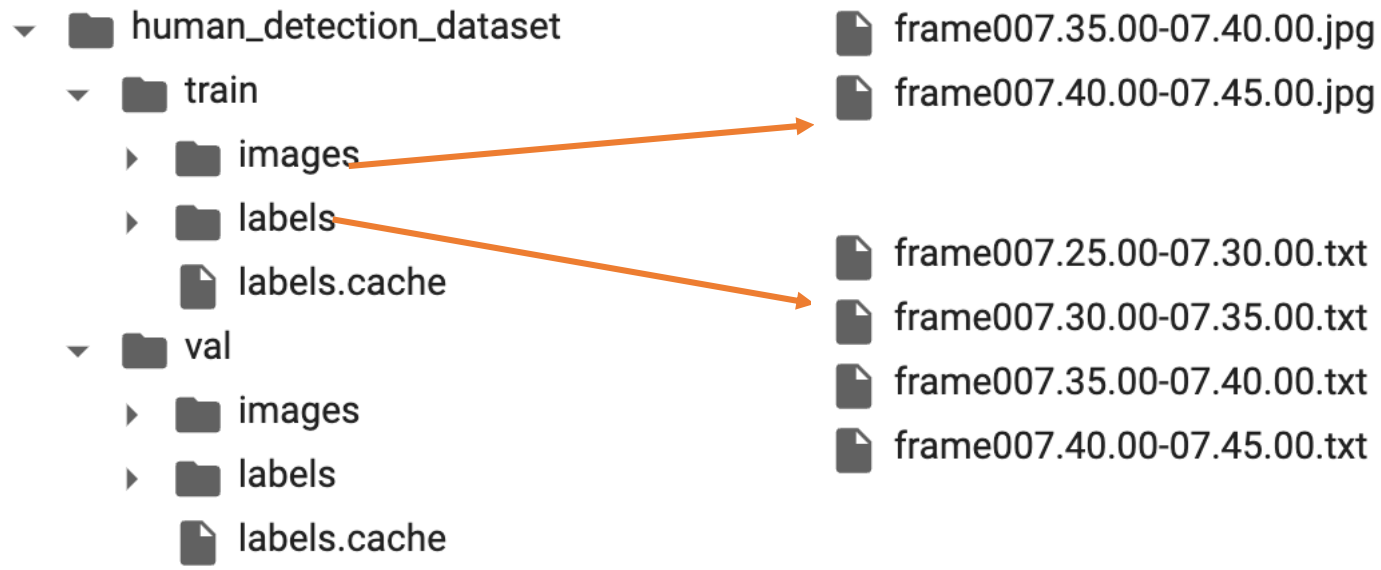
Image:

Label:

007.25.00-07.30.00.txt

```
1 0 0.789062 0.855093 0.0239583 0.075
2 0 0.835417 0.837037 0.0145833 0.0425926
3 0 0.816406 0.849537 0.0171875 0.0435185
4 0 0.965104 0.713426 0.06875 0.562037
5 0 0.448698 0.864815 0.139062 0.262963
6 0 0.283333 0.839815 0.0833333 0.312963
7 0 0.739583 0.824074 0.1 0.346296
8 0 0.588281 0.698148 0.204688 0.592593
9
```

YOLOv8 Dataset folder format



- **Train set:** 2220
- **Val set:** 1642
- **Total:** 3862

YOLOv8 – How to use

❖ Step 4: Download dataset

Human dataset: https://drive.google.com/file/d/1--0QuKMwj31K-CSvD8oq5fceFweiFPuN/view?usp=share_link

1. Download with gdown:

```
!gdown --id 1--0QuKMwj31K-CSvD8oq5fceFweiFPuN&
!unzip /content/human_detection_dataset.zip
```

```
/usr/local/lib/python3.7/dist-packages/gdown/cli.py:131: FutureWarning: Opti
  category=FutureWarning,
Access denied with the following error:

        Too many users have viewed or downloaded this file recently. Please
        try accessing the file again later. If the file you are trying to
        access is particularly large or is shared with many people, it may
        take up to 24 hours to be able to view or download the file. If you
        still can't access a file after 24 hours, contact your domain
        administrator.

You may still be able to access the file from the browser:

        https://drive.google.com/uc?id=1bYytJdFa1D_r2km6ZZ8T99TnPlvyCAhf

unzip:  cannot find or open helmet.zip, helmet.zip.zip or helmet.zip.ZIP.
```

2. Copy from google drive:

```
from google.colab import drive

drive.mount('/content/gdrive')
```

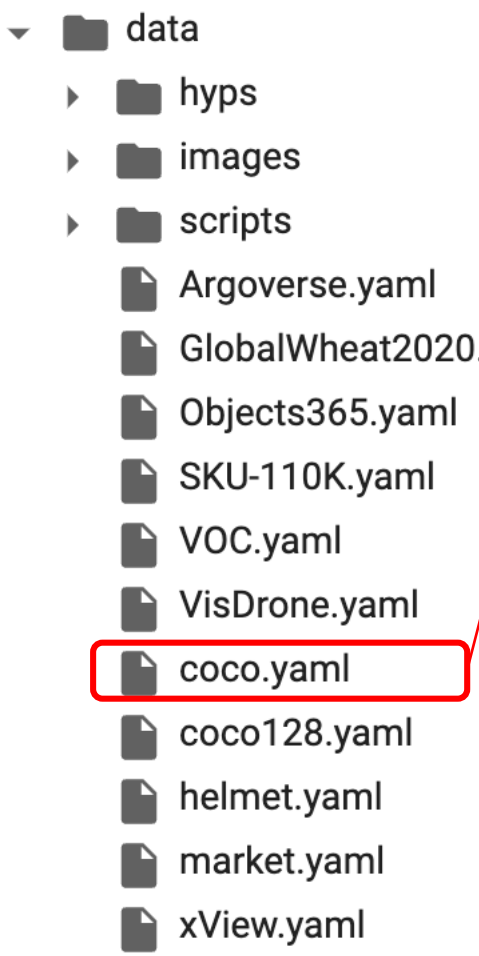
```
/content
Mounted at /content/gdrive
```

```
!cp path_to_human_dataset.zip .
```

Method 1 might not work if the number of downloads is high.

YOLOv8 – How to use

❖ Step 5: Prepare .yaml file



coco.yaml

```yaml
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# COCO 2017 dataset http://cocodataset.org by Microsoft
# Example usage: python train.py --data coco.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco  ← downloads here (20.1 GB)


# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt
path: ../datasets/coco  # dataset root dir
train: train2017.txt  # train images (relative to 'path') 118287 images
val: val2017.txt  # val images (relative to 'path') 5000 images
test: test-dev2017.txt  # 20288 of 40670 images, submit to https://compe

# Classes
nc: 80  # number of classes
names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'tr
       'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', '
       'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella',
       'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'bas
       'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife',
       'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
       'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop',
       'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book',
       'hair drier', 'toothbrush']  # class names

```

Path to root folder

Path to train images folder

Path to val images folder

Path to test images folder (optional)

Number of classes to detect

Classes's name

YOLOv8 – How to use

❖ Step 5: Prepare .yaml file

```yaml
import yaml

dataset_info = {
    'train': './train/images',
    'val': './train/images',
    'nc': 1,
    'names': ['Human']
}

yaml_savepath = './data.yaml'
with open(yaml_savepath, 'w') as f:
    doc = yaml.dump(
        dataset_info,
        f,
        default_flow_style=None,
        sort_keys=False
    )
```

Dict of required information

YOLOv8 – How to use

❖ Step 6: Training

Training command.

```
1 !yolo train model=yolov8s.pt data=../human_detection_dataset/data.yaml epochs=20 imgsz=640
```

```
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
yolo/engine/trainer: task=detect, mode=train, model=yolov8s.pt, data=../human_detection_dataset/data.yaml, epochs=20,
Downloading https://ultralytics.com/assets/Arial.ttf to /root/.config/Ultralytics/Arial.ttf...
100% 755k/755k [00:00<00:00, 22.1MB/s]
Overriding model.yaml nc=80 with nc=1
```

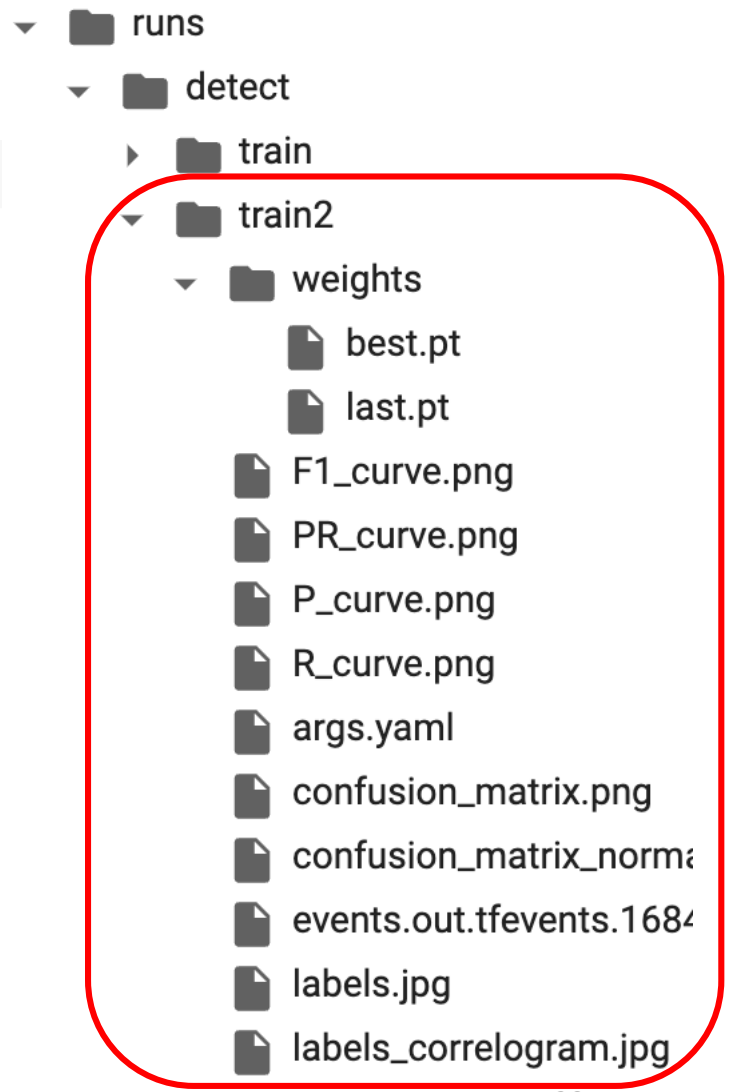
Log screen when training process is end.

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/20      4.05G     0.7462     0.4929     0.9263        158        640: 100% 139/139 [04:49<00:00,  2.08s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 52/52 [01:06<00:00,  1.28s/it]
                   all       1642      13171       0.86      0.745      0.843      0.588

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/20      4.07G      0.739     0.4844     0.9218        159        640: 100% 139/139 [04:48<00:00,  2.08s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 52/52 [01:16<00:00,  1.47s/it]
                   all       1642      13171      0.855      0.744      0.847      0.596

20 epochs completed in 2.010 hours.
Optimizer stripped from /content/ultralytics/runs/detect/train2/weights/last.pt, 22.5MB
Optimizer stripped from /content/ultralytics/runs/detect/train2/weights/best.pt, 22.5MB

Validating /content/ultralytics/runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 52/52 [01:10<00:00,  1.35s/it]
                   all       1642      13171      0.856      0.743      0.847      0.596
Speed: 0.5ms preprocess, 2.9ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to /content/ultralytics/runs/detect/train2
```



YOLOv8 – How to use

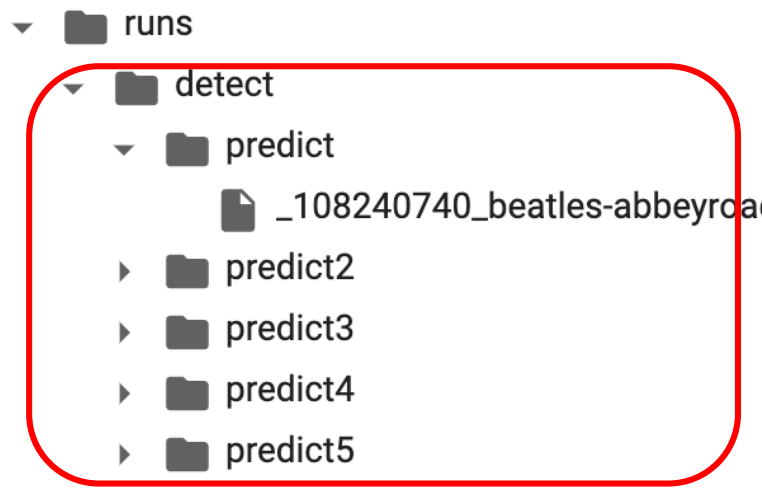
❖ Step 7: Predict

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
19/20      4.05G     0.7462     0.4929     0.9263        158        640: 100% 139/139 [04:49<00:00,  2.08s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 52/52 [01:06<00:00,  1.28s/it]
             all       1642      13171      0.86      0.745      0.843      0.588

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
20/20      4.07G      0.739     0.4844     0.9218        159        640: 100% 139/139 [04:48<00:00,  2.08s/it]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 52/52 [01:16<00:00,  1.47s/it]
             all       1642      13171      0.855      0.744      0.847      0.596

20 epochs completed in 2.010 hours.
Optimizer stripped from /content/ultralytics/runs/detect/train2/weights/last.pt, 22.5MB
Optimizer stripped from /content/ultralytics/runs/detect/train2/weights/best.pt, 22.5MB

Validating /content/ultralytics/runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 52/52 [01:10<00:00,  1.35s/it]
                   all       1642      13171      0.856      0.743      0.847      0.596
Speed: 0.5ms preprocess, 2.9ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to /content/ultralytics/runs/detect/train2
```



Path to trained weight file (.pt)

Predict command

```
# With uploaded image
!yolo predict model=./runs/detect/train2/weights/best.pt source='/content/ultralytics/frame007.25.00-07.30.00.jpg'
```

```
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients
```

YOLOv8 – How to use

❖ Step 7: Predict

source	model(arg)	type	notes
image	`'im.jpg'`	`str`, `Path`	
URL	`'https://ultralytics.com/images/bus.jpg'`	`str`	
screenshot	`'screen'`	`str`	
PIL	`Image.open('im.jpg')`	`PIL.Image`	HWC, RGB
OpenCV	`cv2.imread('im.jpg')`	`np.ndarray`	HWC, BGR
numpy	`np.zeros((640,1280,3))`	`np.ndarray`	HWC
torch	`torch.zeros(16,3,320,640)`	`torch.Tensor`	BCHW, RGB
CSV	`'sources.csv'`	`str`, `Path`	RTSP, RTMP, HTTP
video ✅	`'vid.mp4'`	`str`, `Path`	
directory ✅	`'path/'`	`str`, `Path`	
glob ✅	`'path/*.jpg'`	`str`	Use `*` operator
YouTube ✅	`'https://youtu.be/Zgi9g1ksQHc'`	`str`	
stream ✅	`'rtsp://example.com/media.mp4'`	`str`	RTSP, RTMP, HTTP

Other source options that we can input to run predict on YOLOv8.

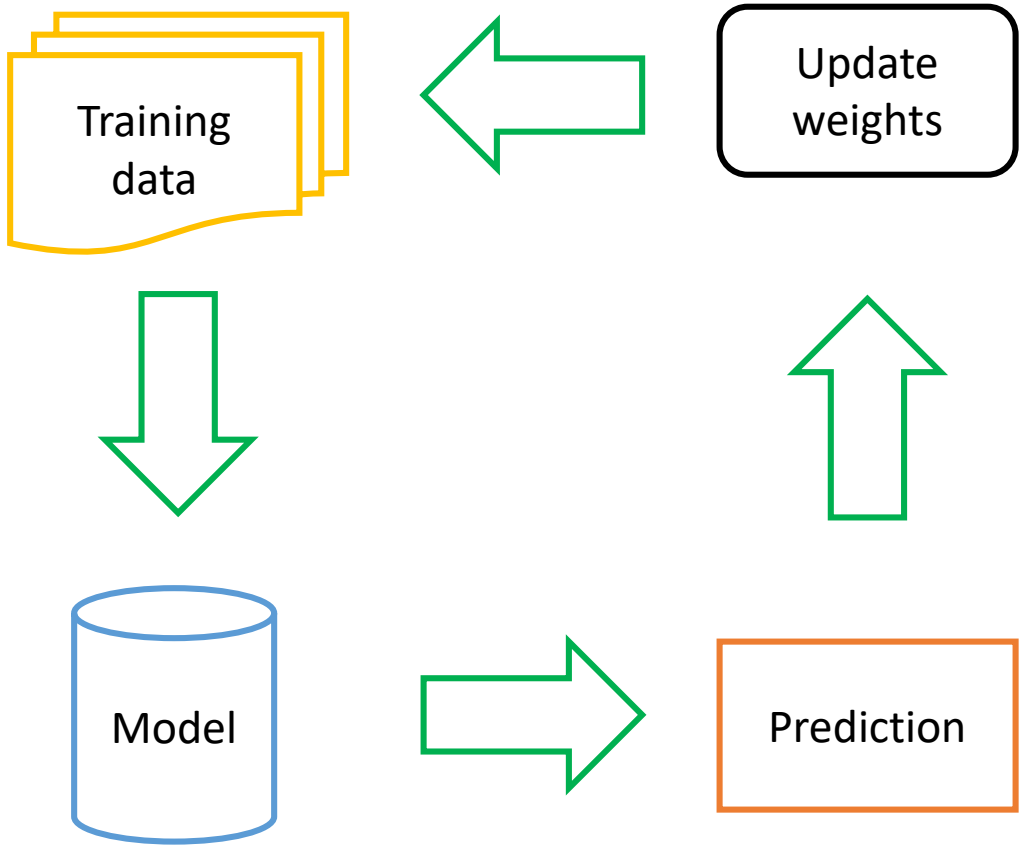
```
# With uploaded image
!yolo predict model=yolov8n.pt source=filepath
```

```
# With online image
!yolo predict model=yolov8n.pt source=online_img_url
```

```
# With youtube video
!yolo predict model=yolov8n.pt source=youtube_video_url
```

YOLOv8 – How to use

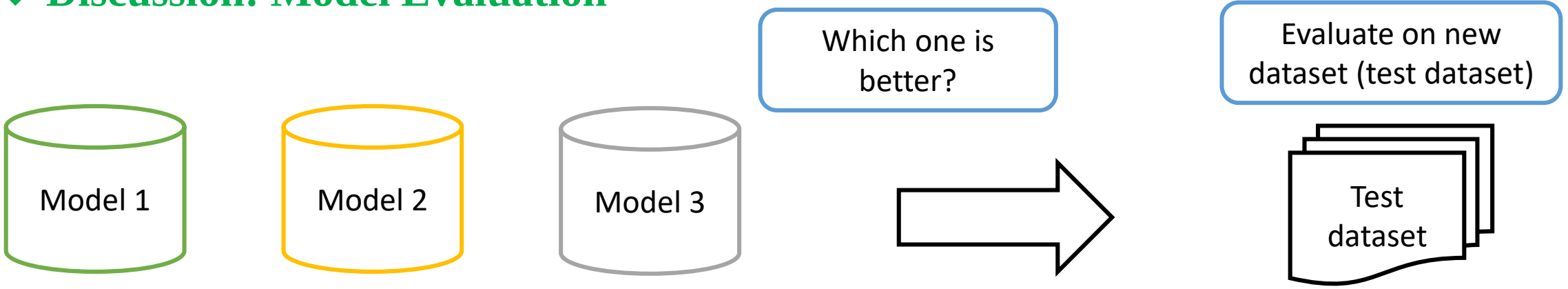
❖ Discussion: Training command flags meaning



Flag	Meaning
img	The size of training image
batch	Number of data samples to read in 1 epoch
epochs	Number of iteration (number of times model "see" training data)
data	Path to .yaml file
weights	Path to pretrained model

YOLOv8 – How to use

❖ Discussion: Model Evaluation




```
1 !yolo val model=./runs/detect/train2/weights/best.pt data=../human_detection_dataset/data.yaml
```

```
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients
val: Scanning /content/human_detection_dataset/val/labels.cache... 1642 images, 0 backgrounds, 0 corrupt: 100%
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 103/103 [01:19<
                   all       1642      13171      0.857      0.743      0.847      0.598
Speed: 0.4ms preprocess, 5.6ms inference, 0.0ms loss, 2.7ms postprocess per image
Results saved to /content/ultralytics/runs/detect/val
```

Outline

- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

Linux Commands (Colab)

❖ Introduction

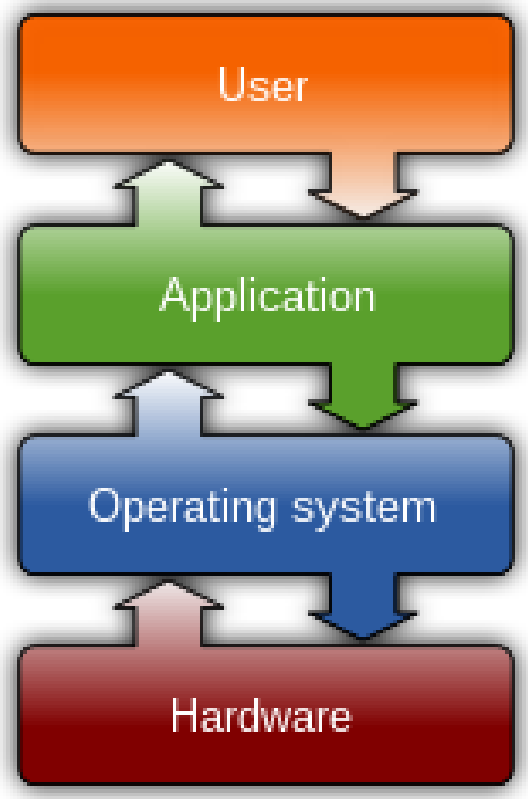
Microsoft Windows

macOS

These are called Operating System (OS)

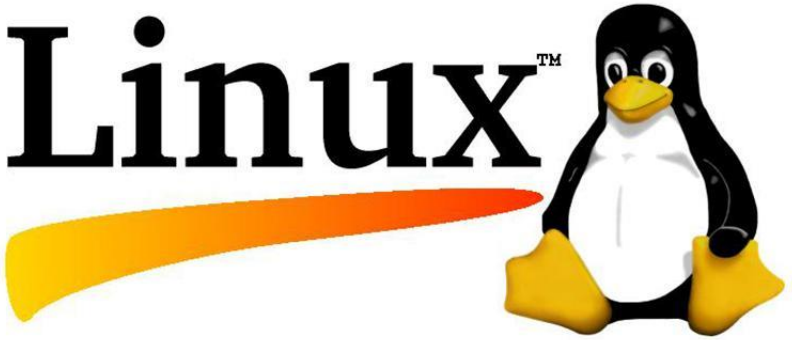
Linux Commands (Colab)

❖ Introduction



Operating System: A system software that manages computer hardware, software resources…

source: https://en.wikipedia.org/wiki/Operating_system



Some famous operating systems

Linux Commands (Colab)

❖ Linux

Linux OS: A family of open-source Unix-like OSs based on the Linux Kernel

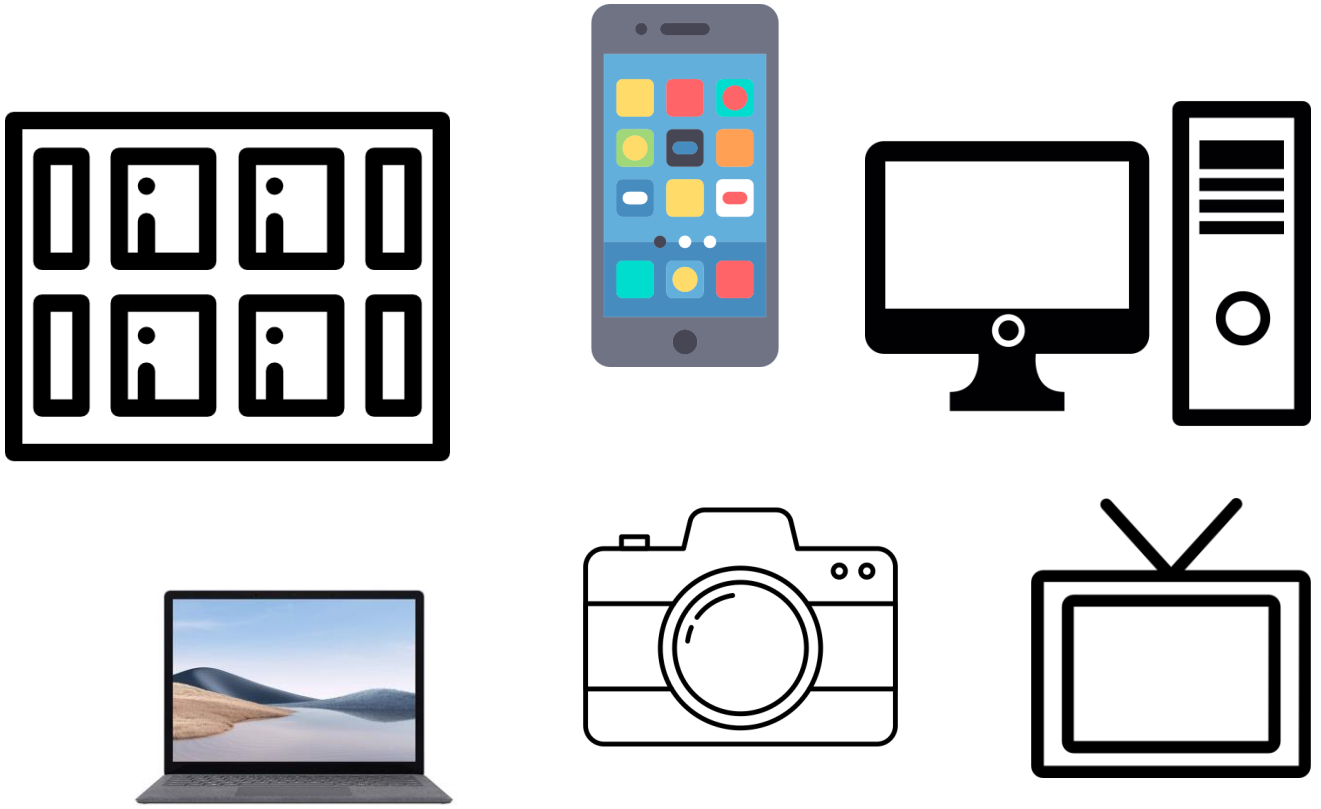
Linux Distributions: Operating System that based on Linux

Linux Commands (Colab)

❖ Linux

Which systems use Linux?

- 100% of the top 500 supercomputers
- 36% of the top 10 million webservers
- 97% of the top 1 million webservers
- Nearly 100% of all Android devices
- 1.8% of desktops/laptops
- 15% of desktops/laptops that aren't running a Microsoft OS
- Roku STBs



Lots of machines use Linux

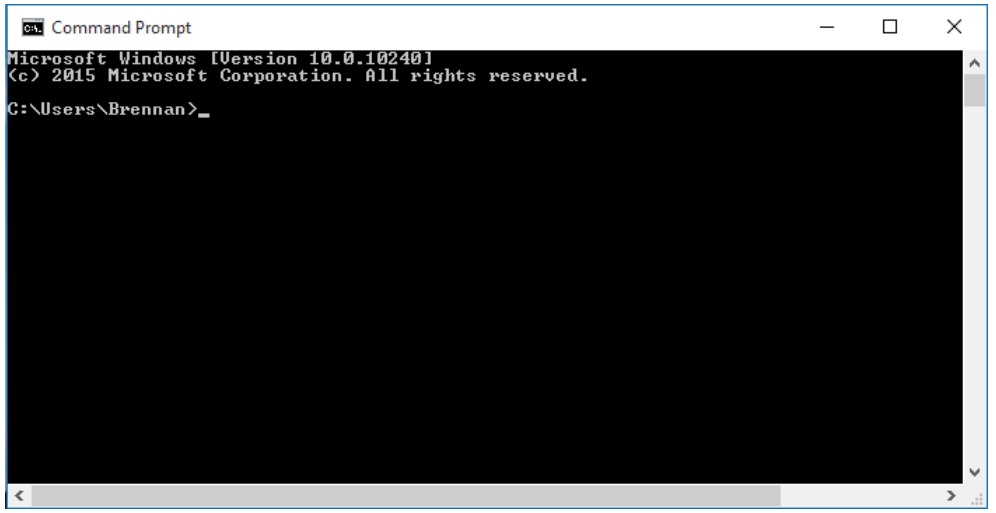
source: https://www.quora.com/Which-computers-are-using-Linux

Linux Commands (Colab)

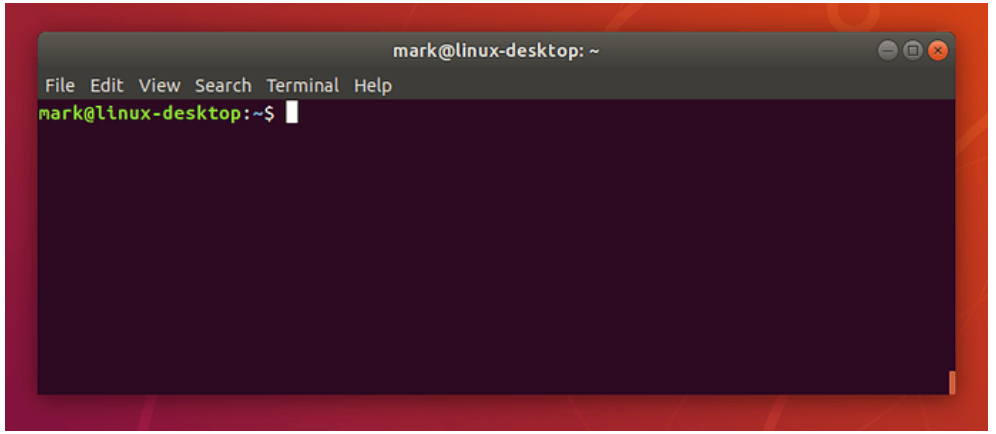
❖ Command Line Interface (CLI)

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x.  2 root root 4096 May 18 16:03 local
lrwxrwxrwx.  1 root root   11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx.  1 root root   10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x.  2 root root 4096 May 18 16:03 nis
drwxr-xr-x.  2 root root 4096 May 18 16:03 opt
drwxr-xr-x.  2 root root 4096 May 18 16:03 preserve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 report
lrwxrwxrwx.  1 root root    6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x.  2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmfusion-free-updates                                          | 2.7 kB     00:00
rpmfusion-free-updates/primary_db                               | 206 kB     00:04
rpmfusion-nonfree-updates                                       | 2.7 kB     00:00
updates/metalink                                                | 5.9 kB     00:00
updates                                                         | 4.7 kB     00:00
Updates/primary_db                      73% [=================       ]  62 kB/s | 2.6 MB     00:15 ETA
```

Command-line interface (CLI): a text-based user interface used to run programs, interact with computer.



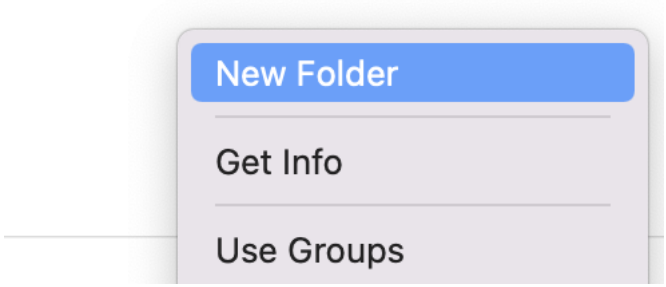
Windows Command Line (cmd)

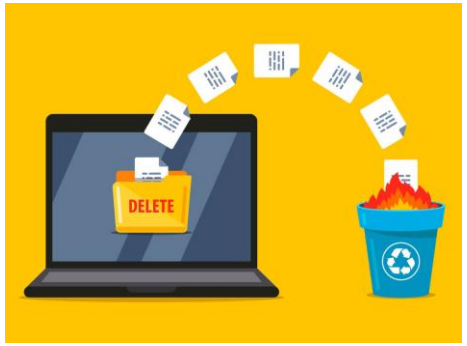


Ubuntu Command Line (Terminal)

Linux Commands (Colab)

❖ Why CLI?





1. Create folder in traditional way

```
thangduong@Duongs-MacBook-Pro Desktop % mkdir new_folder
thangduong@Duongs-MacBook-Pro Desktop %
```

2. Create folder using command line

Can do many process conveniently

Linux Commands (Colab)

❖ Linux Command Line

```
thangduong@Duongs-MacBook-Pro Desktop % mkdir -h
mkdir: illegal option -- h
usage: mkdir [-pv] [-m mode] directory_name ...
thangduong@Duongs-MacBook-Pro Desktop %
```

```
thangduong@Duongs-MacBook-Pro Desktop % mkdir new_folder
thangduong@Duongs-MacBook-Pro Desktop %
```

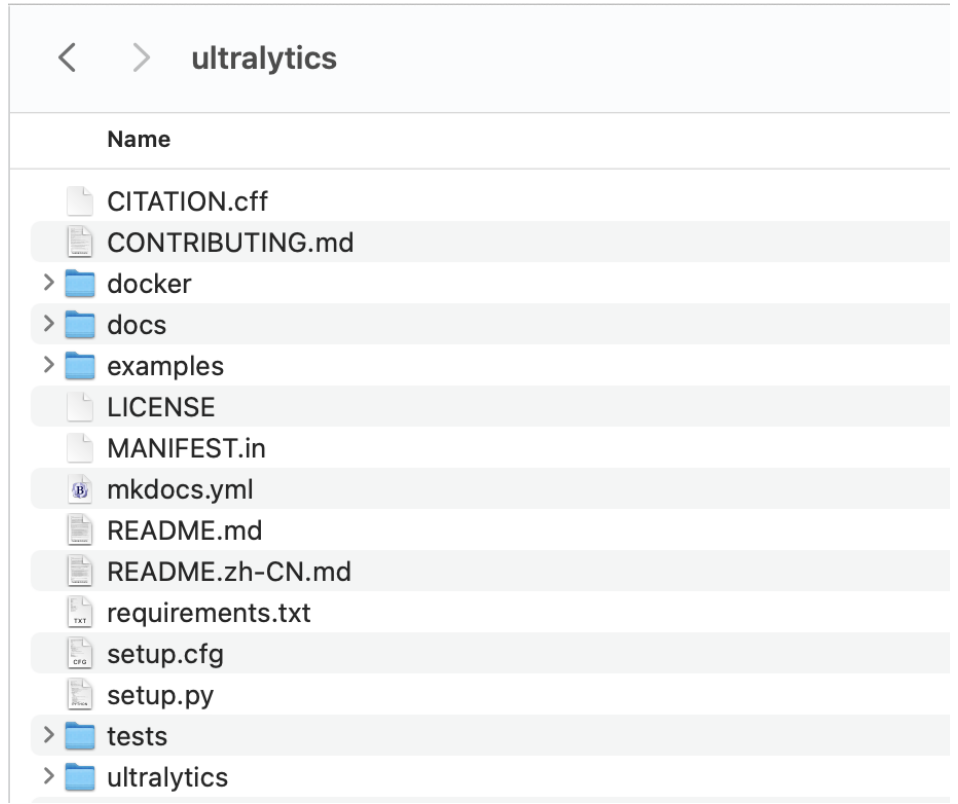
flag: An option for the command, can be started with (-) or (--)

argument: Input given to a command line

Linux Commands (Colab)

❖ Common Linux Commands: ls

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff                examples
CONTRIBUTING.md             mkdocs.yml
LICENSE                     requirements.txt
MANIFEST.in                 setup.cfg
README.md                   setup.py
README.zh-CN.md             tests
docker                      ultralytics
docs
```

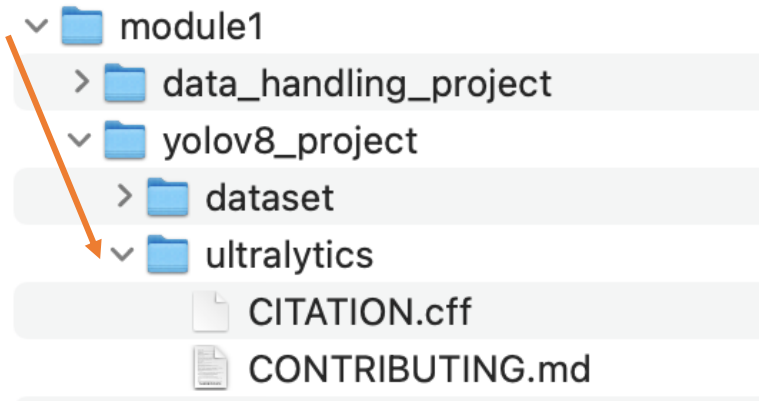


- **Command:** `ls`
- **Usage:** To list all file/folder names in current directory.

Linux Commands (Colab)

❖ Common Linux Commands: cd

```
(base) thangduong@Duongs-MacBook-Pro module1 % ls
data_handling_project   yolov8_project
(base) thangduong@Duongs-MacBook-Pro module1 % cd yolov8_project
(base) thangduong@Duongs-MacBook-Pro yolov8_project % ls
dataset         ultralytics
(base) thangduong@Duongs-MacBook-Pro yolov8_project % cd ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            README.zh-CN.md         requirements.txt
CONTRIBUTING.md         docker                  setup.cfg
LICENSE                 docs                    setup.py
MANIFEST.in             examples                tests
README.md               mkdocs.yml              ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```



E.g: Move from module1 to ultralytics

- **Command:** `cd`
- **Usage:** "change directory", move to the specified directory.

```
thangduong@Duongs-MacBook-Pro data % ls
Argoverse.yaml          VOC.yaml                hyps
GlobalWheat2020.yaml    VisDrone.yaml           images
Objects365.yaml         coco.yaml               scripts
SKU-110K.yaml           coco128.yaml            xView.yaml
thangduong@Duongs-MacBook-Pro data % cd ..
thangduong@Duongs-MacBook-Pro yolov5 % ls
CONTRIBUTING.md         export.py               train.py
LICENSE                 hubconf.py              tutorial.ipynb
README.md               models                  utils
data                    requirements.txt        val.py
detect.py               setup.cfg
thangduong@Duongs-MacBook-Pro yolov5 %
```

Move to "1 level up" directory

```
thangduong@Duongs-MacBook-Pro yolov5 % cd
thangduong@Duongs-MacBook-Pro ~ % pwd
/Users/thangduong
thangduong@Duongs-MacBook-Pro ~ %
```

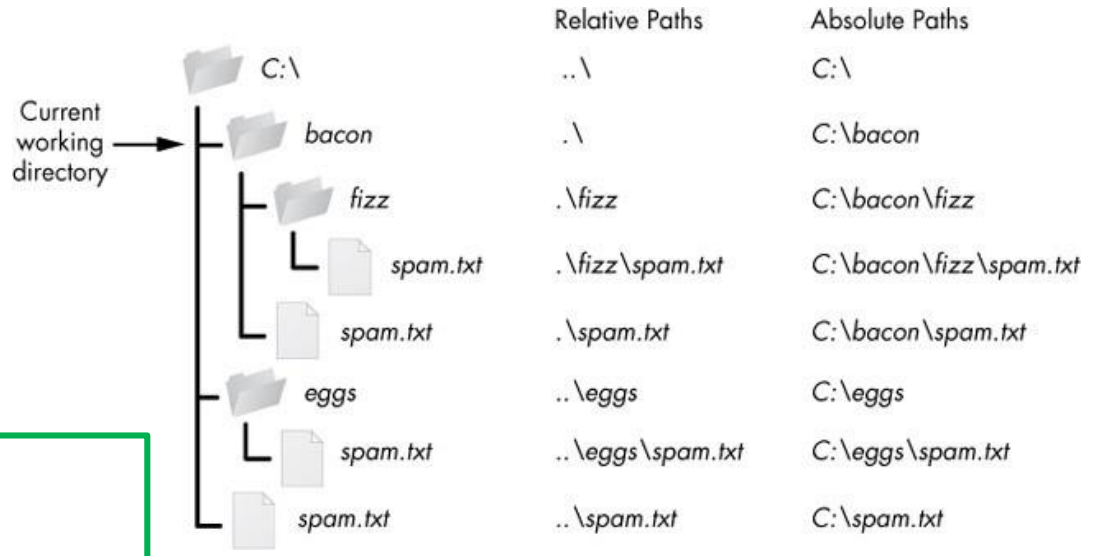
Move to home directory

Linux Commands (Colab)

❖ Common Linux Commands: pwd

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % pwd
/Users/thangduong/Desktop/aio2023/module1/yolov8_project/ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics % cd ..
(base) thangduong@Duongs-MacBook-Pro yolov8_project % pwd
/Users/thangduong/Desktop/aio2023/module1/yolov8_project
(base) thangduong@Duongs-MacBook-Pro yolov8_project %
```

- **Command:** `pwd`
- **Usage:** "Print Working Directory", output the absolute path of current directory.



Relative Path vs Absolute Path

Linux Commands (Colab)

❖ Common Linux Commands: cp and rm

- ❖ **Command:** `cp`
- ❖ **Usage:** copy file/folder to directory.

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls examples
README.md                       hub.ipynb
YOLOv8-CPP-Inference            tutorial.ipynb
YOLOv8-OpenCV-ONNX-Python
(base) thangduong@Duongs-MacBook-Pro ultralytics % cp ./examples/tutorial.ipynb .
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            docker                  setup.py
CONTRIBUTING.md         docs                    tests
LICENSE                 examples                tutorial.ipynb
MANIFEST.in             mkdocs.yml              ultralytics
README.md               requirements.txt
README.zh-CN.md         setup.cfg
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

E.g: Copy tutorial.ipynb from ./examples to current directory.

- ❖ **Command:** `rm`
- ❖ **Usage:** delete file/folder of specified directory.

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            docker                  setup.py
CONTRIBUTING.md         docs                    tests
LICENSE                 examples                tutorial.ipynb
MANIFEST.in             mkdocs.yml              ultralytics
README.md               requirements.txt
README.zh-CN.md         setup.cfg
(base) thangduong@Duongs-MacBook-Pro ultralytics % rm tutorial.ipynb
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            README.zh-CN.md         requirements.txt
CONTRIBUTING.md         docker                  setup.cfg
LICENSE                 docs                    setup.py
MANIFEST.in             examples                tests
README.md               mkdocs.yml              ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

E.g: Delete tutorial.ipynb incurrent directory.

Linux Commands (Colab)

❖ Common Linux Commands: mv and mkdir

- ❖ **Command:** `mv`
- ❖ **Usage:** Move a file/directory to the other directory.

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls ./docs
CNAME                   hub                     robots.txt
README.md               index.md                stylesheets
SECURITY.md             models                  tasks
assets                  modes                   usage
build_reference.py      overrides               yolov5
datasets                quickstart.md
help                    reference
(base) thangduong@Duongs-MacBook-Pro ultralytics % mv ./docs/yolov5 .
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            docker                  setup.py
CONTRIBUTING.md         docs                    tests
LICENSE                 examples                ultralytics
MANIFEST.in             mkdocs.yml              yolov5
README.md               requirements.txt
README.zh-CN.md         setup.cfg
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls ./docs
CNAME                   help                    quickstart.md
README.md               hub                     reference
SECURITY.md             index.md                robots.txt
assets                  models                  stylesheets
build_reference.py      modes                   tasks
datasets                overrides               usage
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

- ❖ **Command:** `mkdir`
- ❖ **Usage:** Create a folder.

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            README.zh-CN.md         requirements.txt
CONTRIBUTING.md         docker                  setup.cfg
LICENSE                 docs                    setup.py
MANIFEST.in             examples                tests
README.md               mkdocs.yml              ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics % mkdir abc
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            abc                     setup.cfg
CONTRIBUTING.md         docker                  setup.py
LICENSE                 docs                    tests
MANIFEST.in             examples                ultralytics
README.md               mkdocs.yml
README.zh-CN.md         requirements.txt
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

Linux Commands (Colab)

❖ Common Linux Commands: touch and cat

- ❖ **Command:** `touch`
- ❖ **Usage:** Create a file.

- ❖ **Command:** `cat`
- ❖ **Usage:** Display, create file.

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            README.zh-CN.md         requirements.txt
CONTRIBUTING.md         docker                  setup.cfg
LICENSE                 docs                    setup.py
MANIFEST.in             examples                tests
README.md               mkdocs.yml              ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics % touch abc.txt
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            abc.txt                 setup.cfg
CONTRIBUTING.md         docker                  setup.py
LICENSE                 docs                    tests
MANIFEST.in             examples                ultralytics
README.md               mkdocs.yml
README.zh-CN.md         requirements.txt
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

```
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            README.zh-CN.md         requirements.txt
CONTRIBUTING.md         docker                  setup.cfg
LICENSE                 docs                    setup.py
MANIFEST.in             examples                tests
README.md               mkdocs.yml              ultralytics
(base) thangduong@Duongs-MacBook-Pro ultralytics % cat > abc.txt
Hello World
^C
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff            abc.txt                 setup.cfg
CONTRIBUTING.md         docker                  setup.py
LICENSE                 docs                    tests
MANIFEST.in             examples                ultralytics
README.md               mkdocs.yml
README.zh-CN.md         requirements.txt
(base) thangduong@Duongs-MacBook-Pro ultralytics % cat abc.txt
Hello World
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

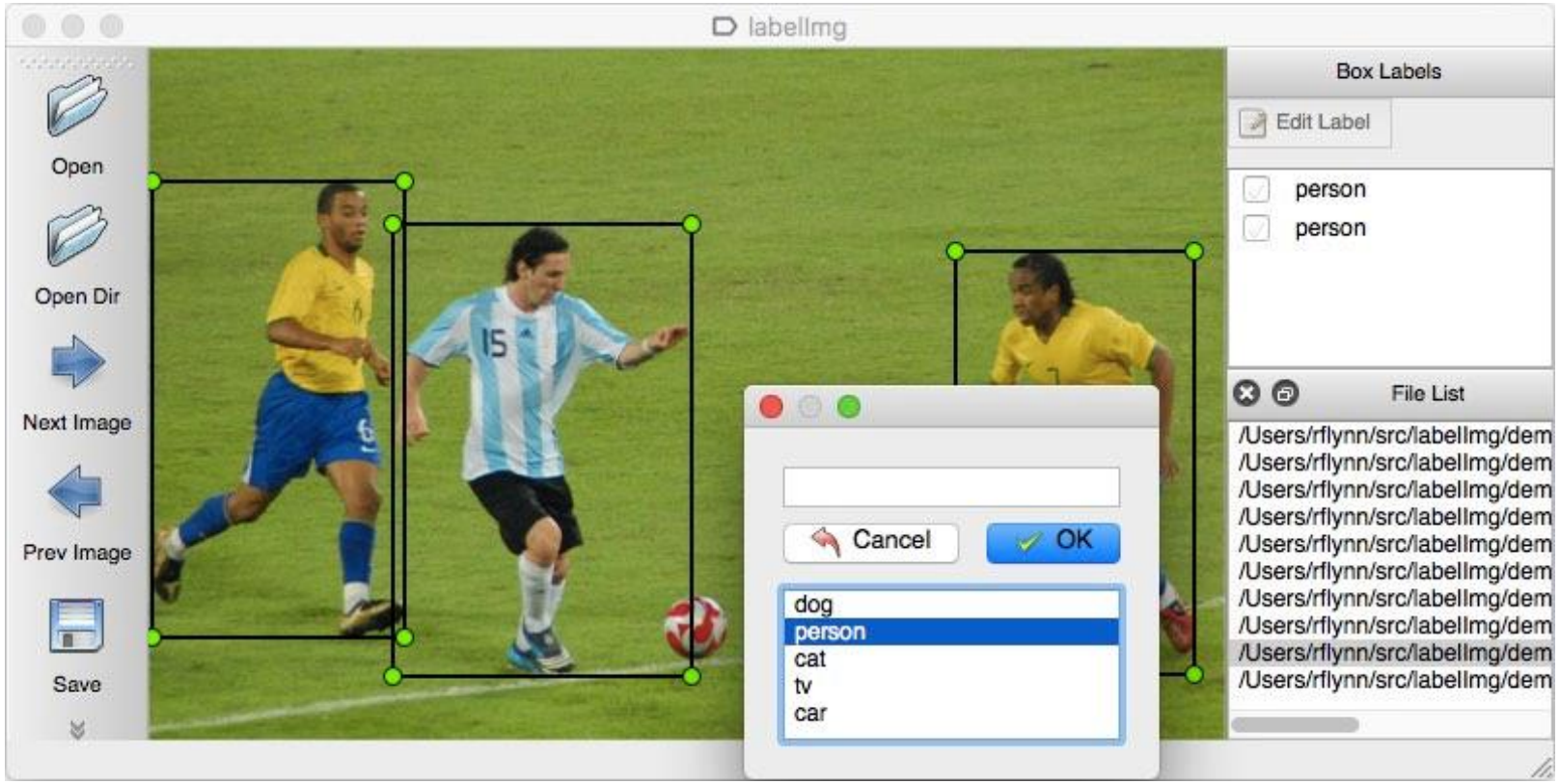
Outline

- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

labelImg

❖ Introduction

A graphical image annotation tool



source: https://github.com/tzutalin/labelImg

labelImg

❖ labelImg Installation

Installation

Get from PyPI but only python3.0 or above

This is the simplest (one-command) install method on modern Linux distributions such as Ubuntu and Fedora.

```
pip3 install labelImg
labelImg
labelImg [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

Build from source

Linux/Ubuntu/Mac requires at least Python 2.6 and has been tested with PyQt 4.8. However, Python 3 or above and PyQt5 are strongly recommended.

Ubuntu Linux

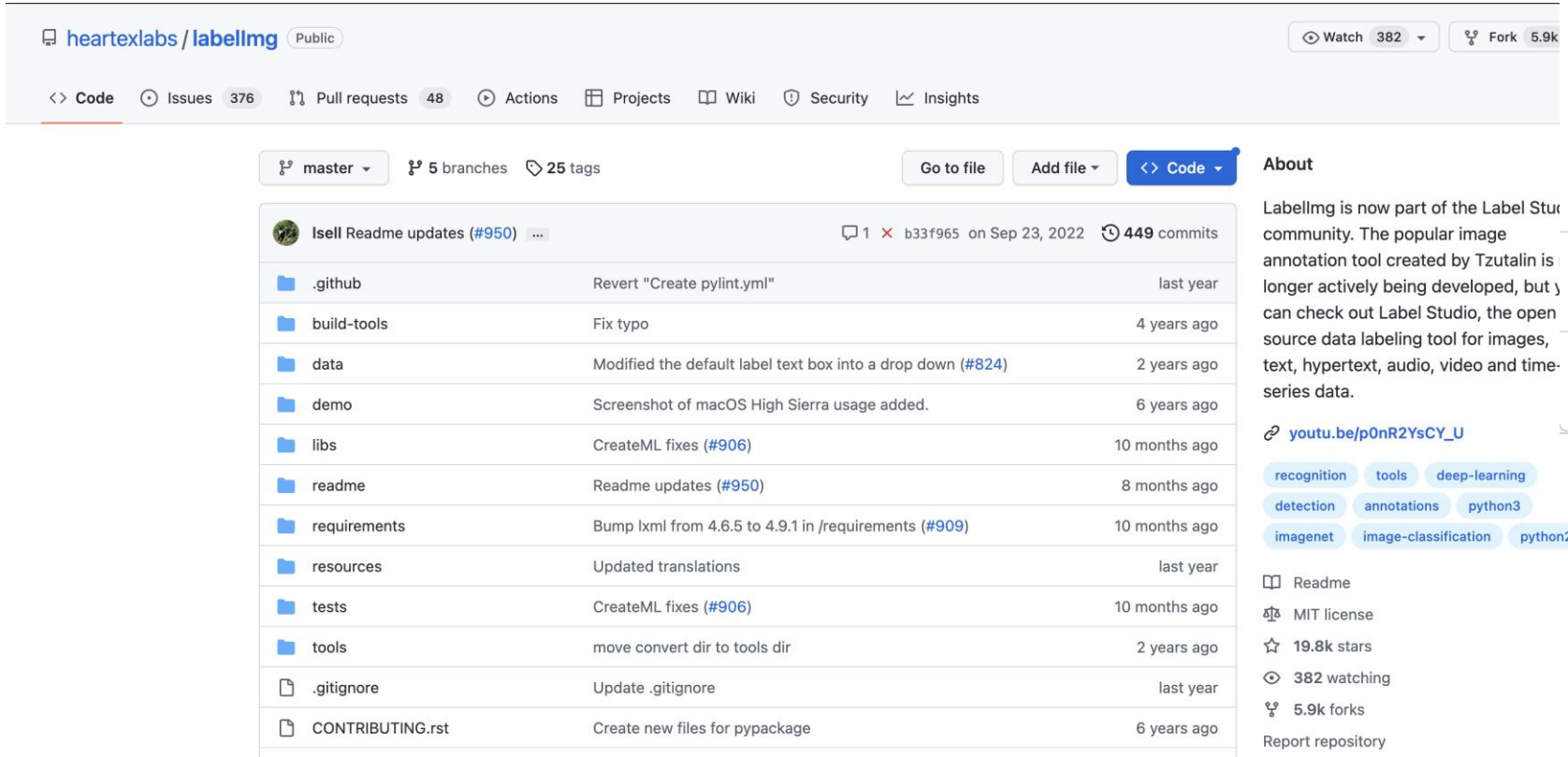
Python 3 + Qt5

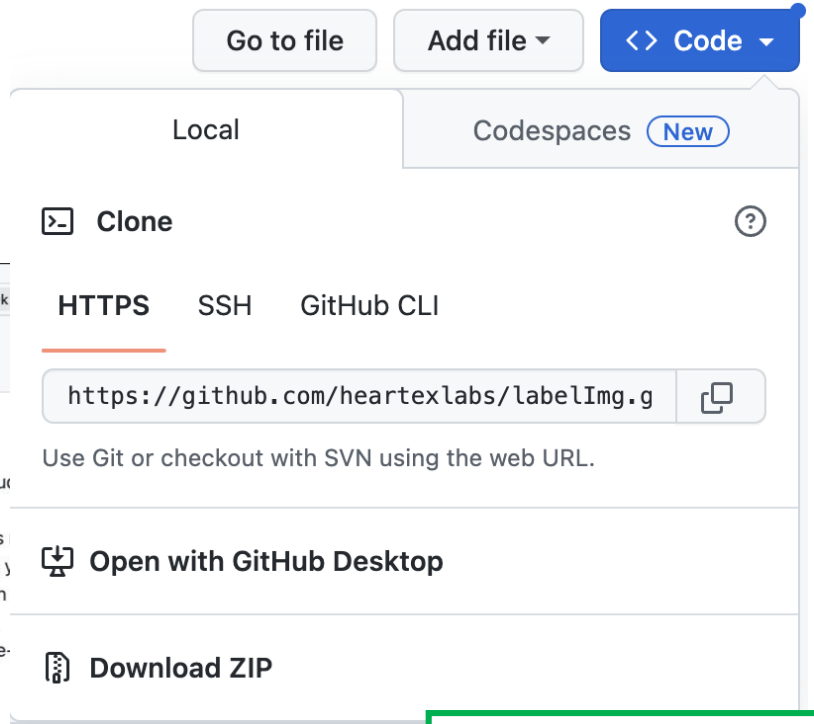
```
sudo apt-get install pyqt5-dev-tools
sudo pip3 install -r requirements/requirements-linux-python3.txt
make qt5py3
python3 labelImg.py
python3 labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

labelImg

❖ labelImg Installation

Step 1: Clone source code





Step 2: Copy link and do git clone

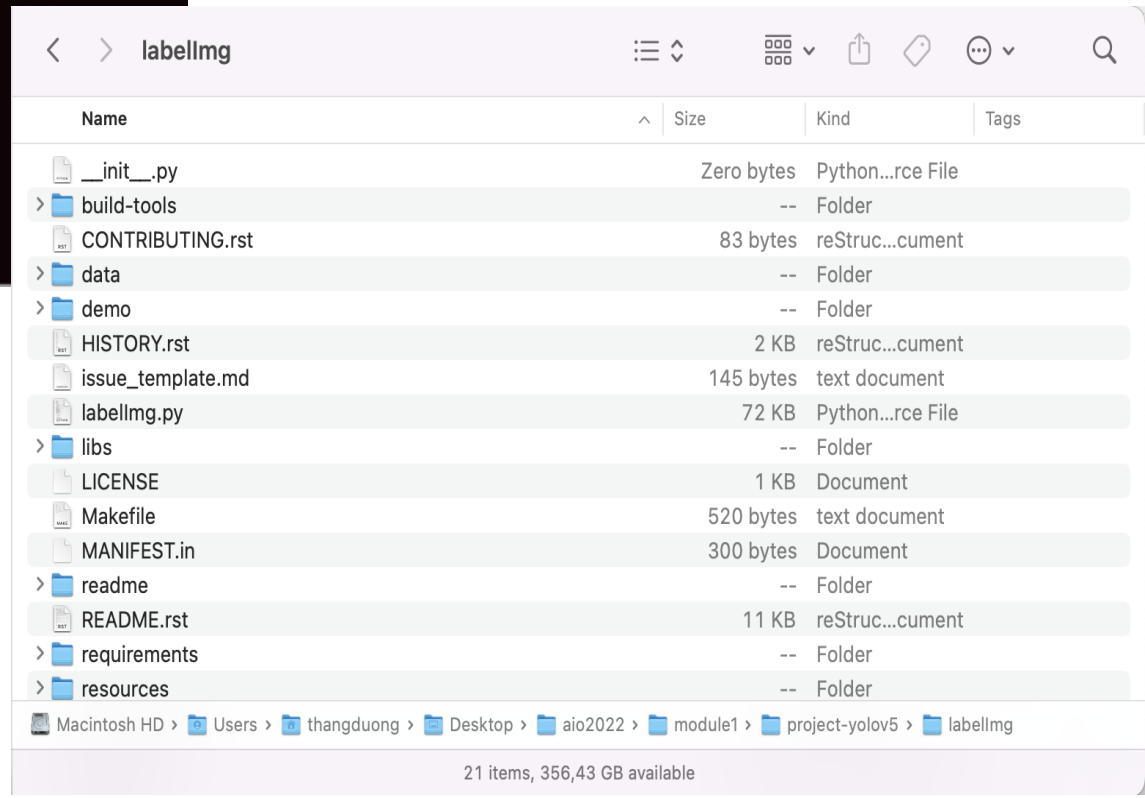
labelImg

❖ labelImg Installation

```
thangduong@Duongs-MacBook-Pro project-yolov5 % git clone https://github.com/tzut
alin/labelImg.git
Cloning into 'labelImg'...
remote: Enumerating objects: 2090, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 2090 (delta 43), reused 74 (delta 35), pack-reused 1986
Receiving objects: 100% (2090/2090), 232.91 MiB | 3.63 MiB/s, done.
Resolving deltas: 100% (1230/1230), done.
thangduong@Duongs-MacBook-Pro project-yolov5 %
```

Use git clone command

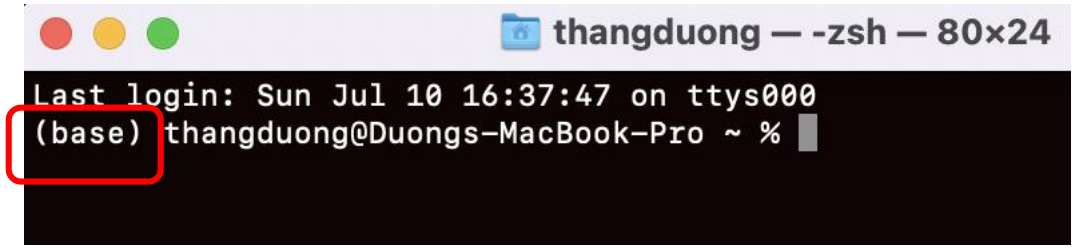
```
thangduong@Duongs-MacBook-Pro labelImg % ls
CONTRIBUTING.rst        build-tools             requirements
HISTORY.rst             data                    resources
LICENSE                 demo                    resources.qrc
MANIFEST.in             issue_template.md       setup.cfg
Makefile                labelImg.py             setup.py
README.rst              libs                    tests
__init__.py             readme                  tools
thangduong@Duongs-MacBook-Pro labelImg %
```



labelImg

❖ labelImg Installation

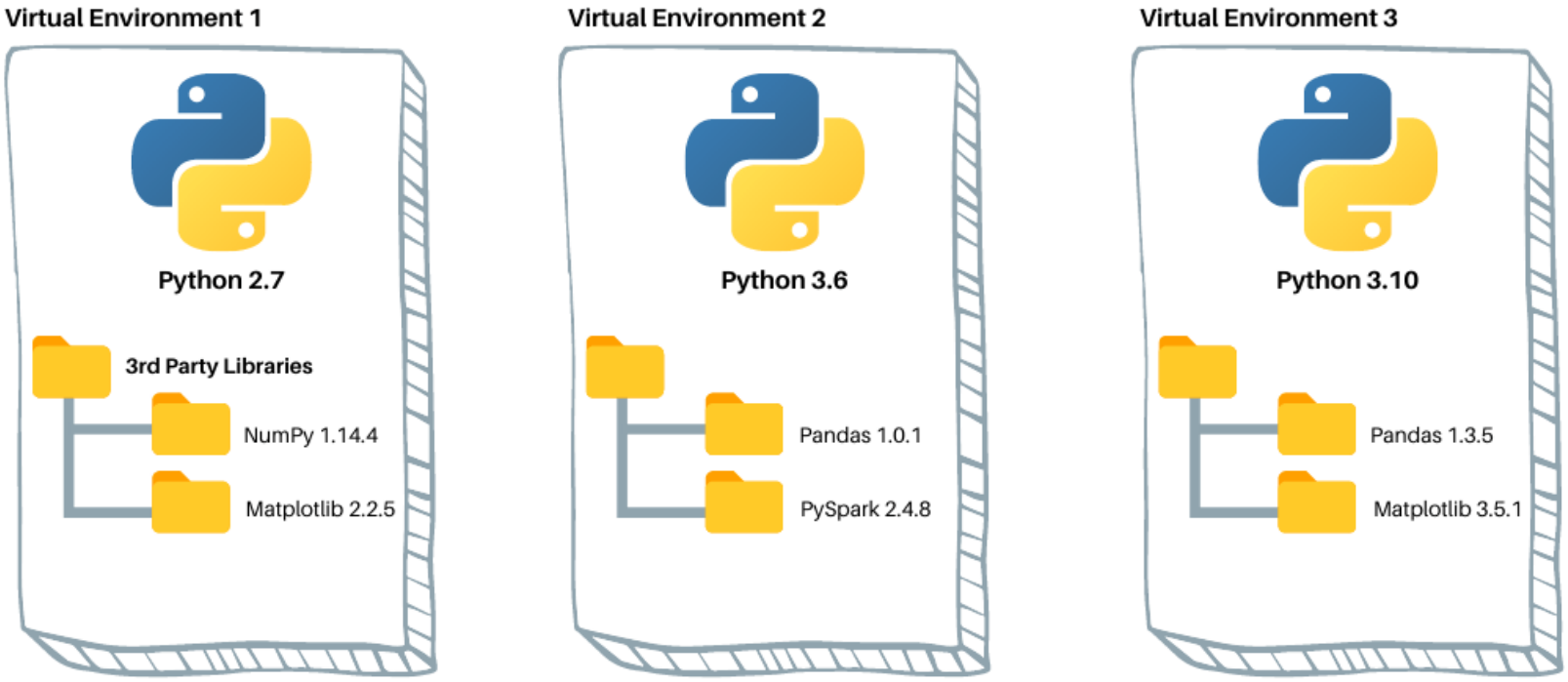
Step 2: Download Anaconda



source: https://docs.anaconda.com/anaconda/user-guide/getting-started/

labelImg

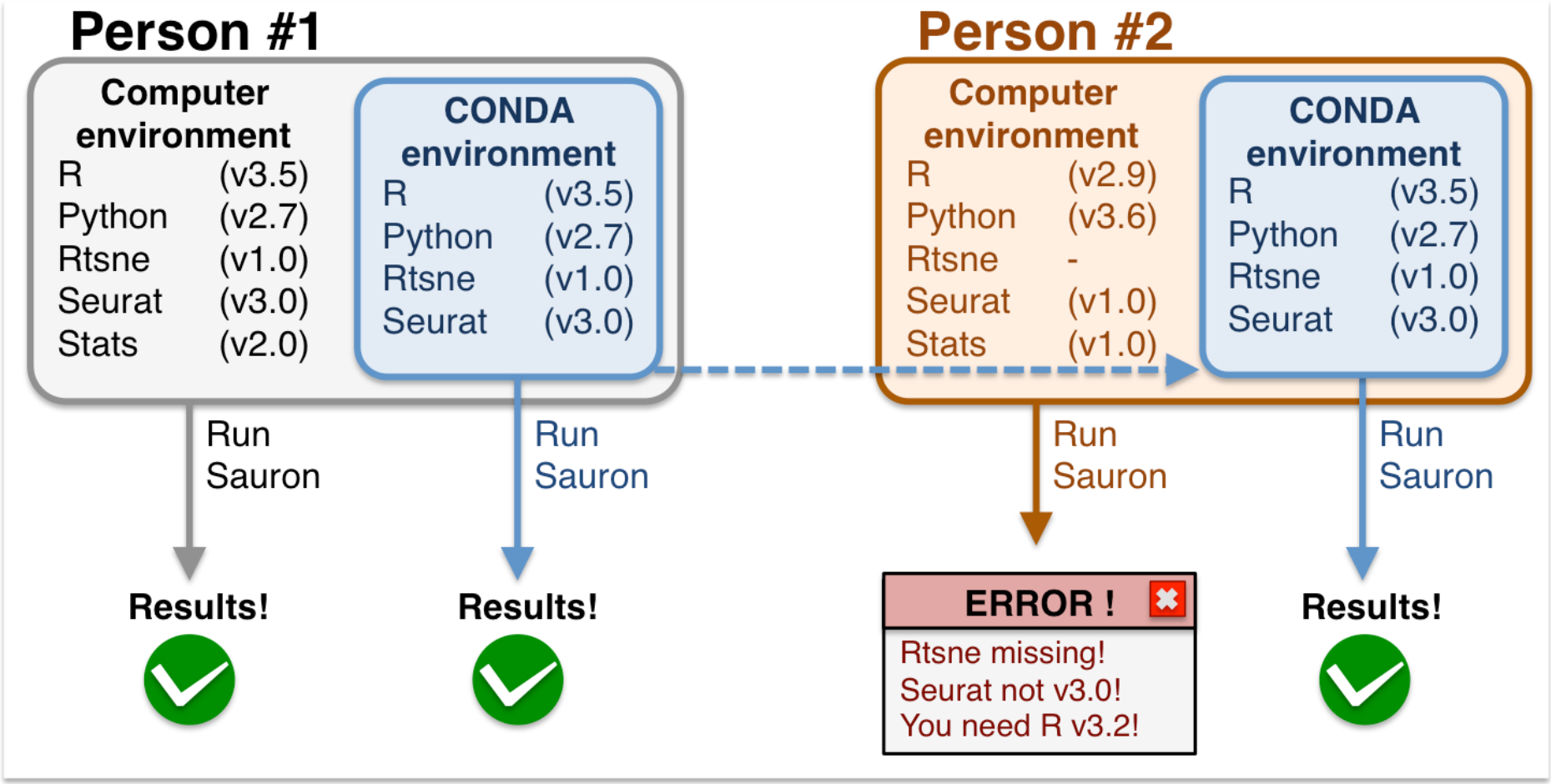
❖ About virtual environment



dataquest.io

labelImg

❖ About conda



labelImg

❖ About conda: Different requirements

master yolov5 / requirements.txt

glenn-jocher Add thop>=0.1.0 (#8558)

21 contributors +9

42 lines (37 sloc) | 1.2 KB

```
# YOLOv5 requirements
# Usage: pip install -r requirements.txt

# Base ----------------------------------------
matplotlib>=3.2.2
numpy>=1.18.5
opencv-python>=4.1.1
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1  # Google Colab version
torch>=1.7.0,!=1.12.0  # https://github.com/ultralytics/yolov5/issues/8395
torchvision>=0.8.1,!=0.13.0 # https://github.com/ultralytics/yolov5/issues/8395
tqdm>=4.41.0
protobuf<4.21.3  # https://github.com/ultralytics/yolov5/issues/8012

```

master PyTorch_YOLOv4 / requirements.txt

WongKinYiu Update requirements.txt

1 contributor

9 lines (9 sloc) | 115 Bytes

```
numpy == 1.17
opencv-python >= 4.1
torch == 1.6
torchvision
matplotlib
pycocotools
tqdm
pillow
tensorboard >= 1.14
```

labelImg

❖ labelImg Installation

Step 3: Create new conda environment

```
(base) thangduong@Duongs-MacBook-Pro ~ % conda create -n labelimg_env python=3.9
 -y
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate labelimg_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) thangduong@Duongs-MacBook-Pro ~ %
```

Step 4: Activate the environment

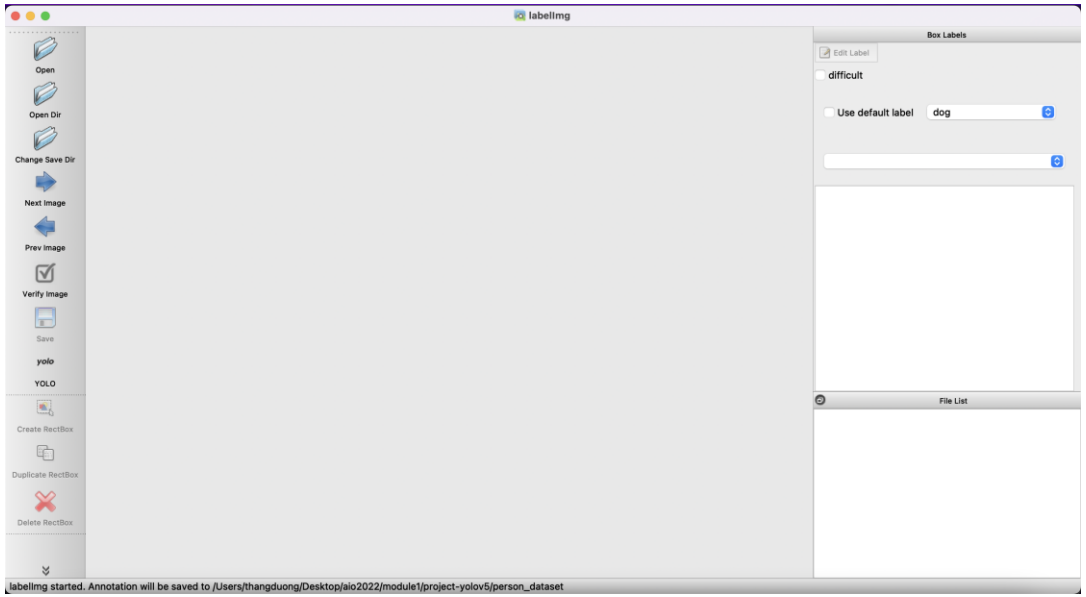
```
(base) thangduong@Duongs-MacBook-Pro ~ % conda activate labelimg_env
(labelimg_env) thangduong@Duongs-MacBook-Pro ~ %
```

labelImg

❖ labelImg Installation

Step 5, 6: Install required packages for labelImg

```
conda install pyqt=5
conda install -c anaconda lxml
pyrcc5 -o libs/resources.py resources.qrc
```

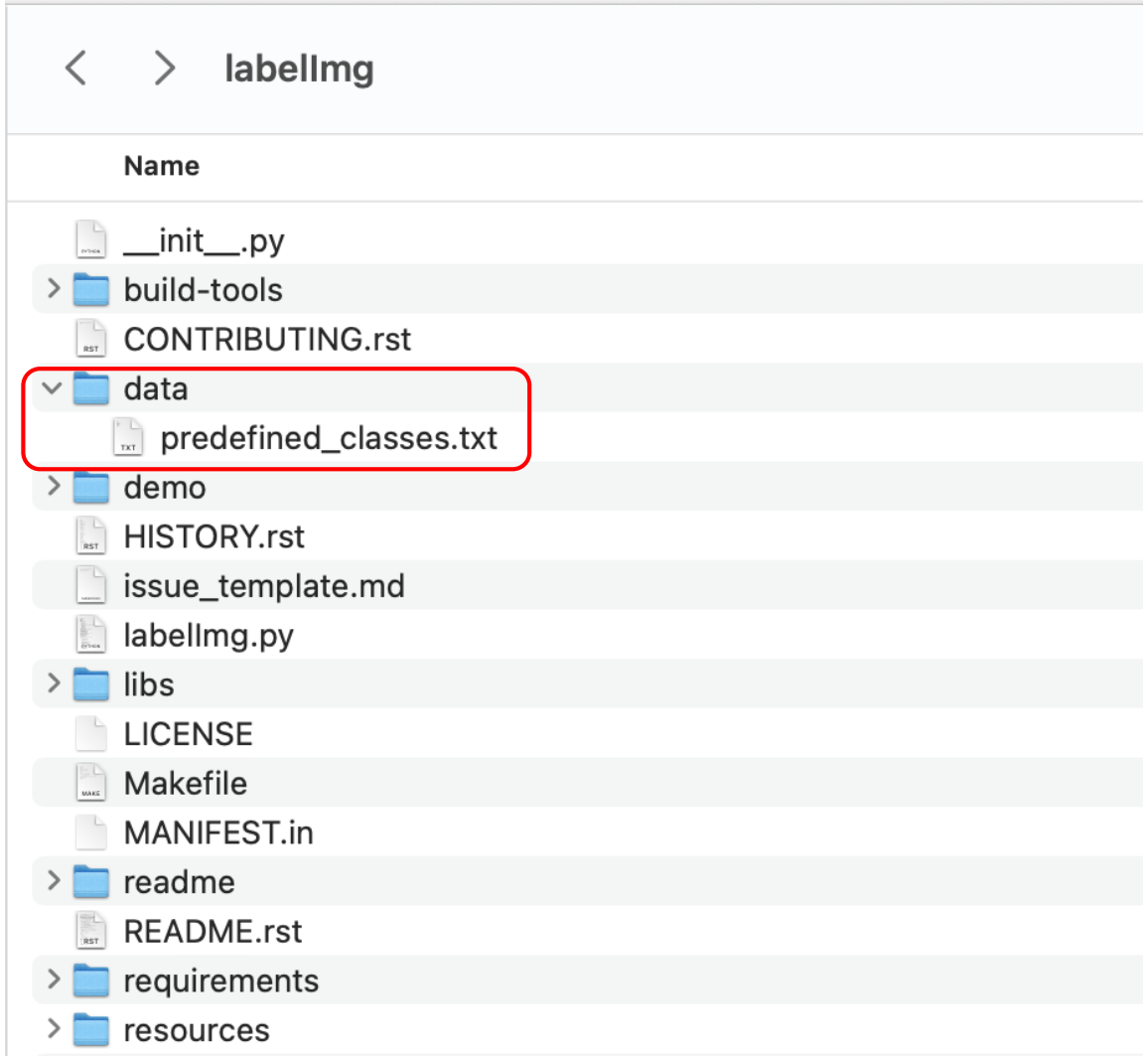


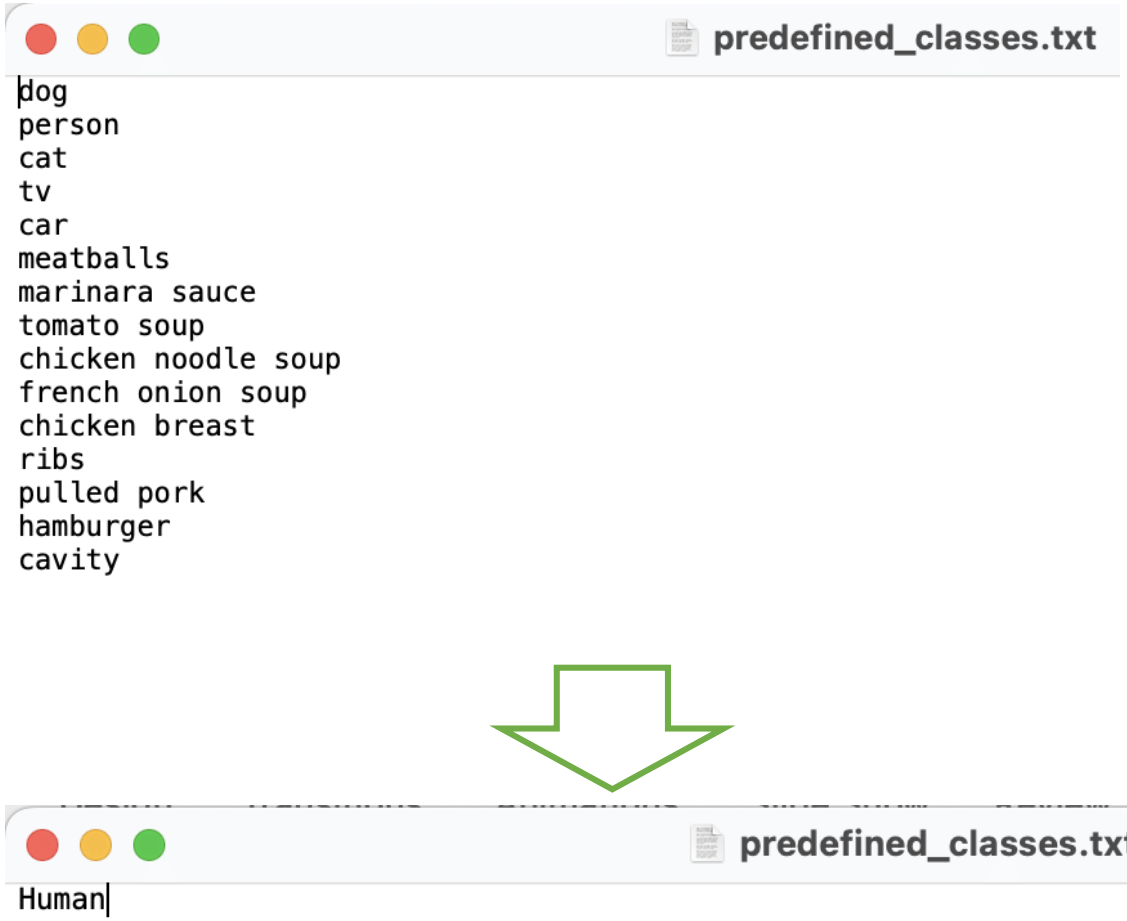
Step 7: Run labelImg

```
(labelimg_env) thangduong@Duongs-MacBook-Pro labelImg % python labelImg.py
/Users/thangduong/Desktop/aio2022/module1/project-yolov5/labelImg/labelImg.py:21
3: DeprecationWarning: an integer is required (got type DockWidgetFeatures).  Im
plicit conversion to integers using __int__ is deprecated, and may be removed in
 a future version of Python.
  self.dock.setFeatures(self.dock.features() ^ self.dock_features)
```

labelImg

❖ How to use?

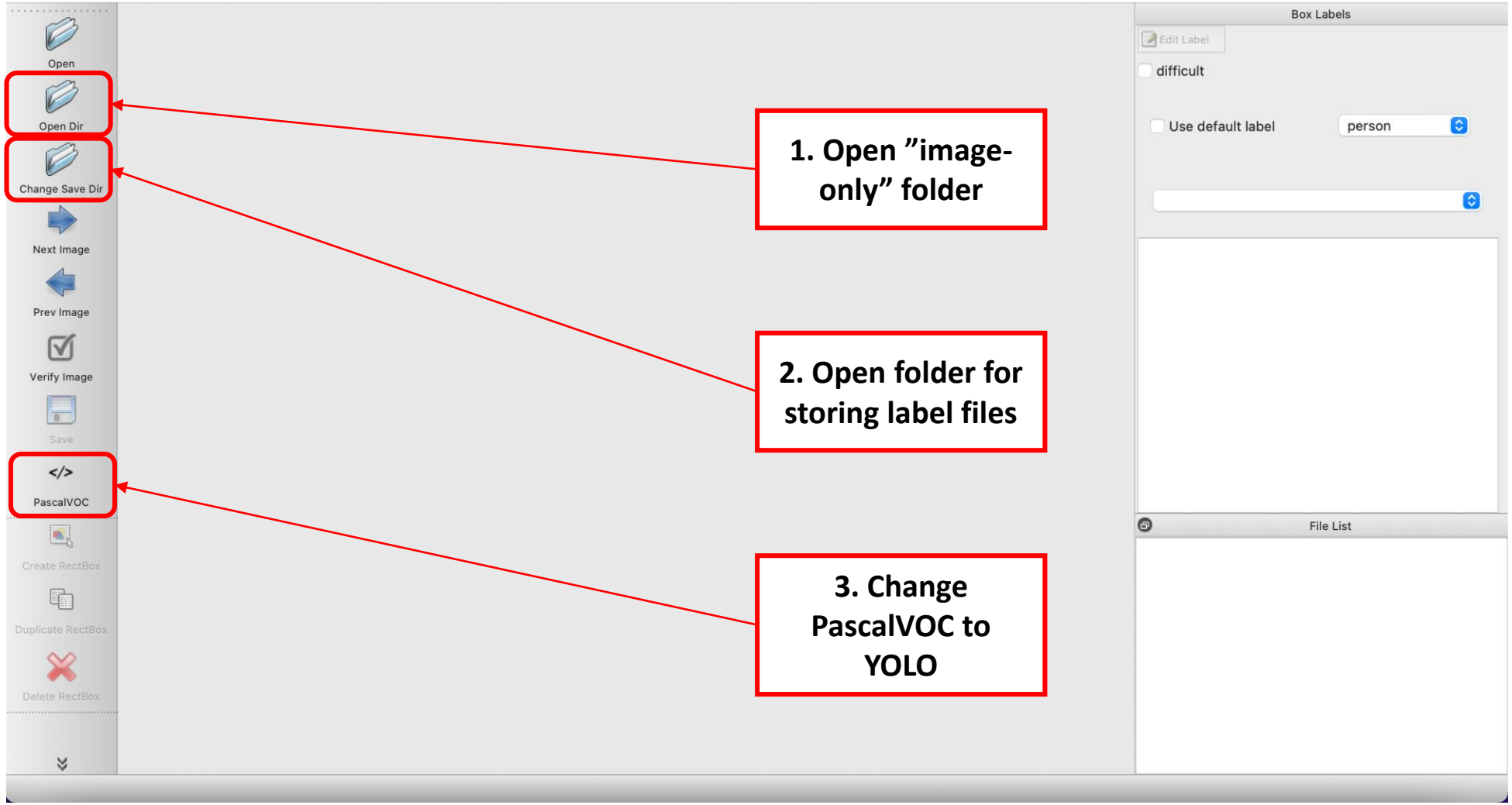




Change to classnames we work with

labelImg

❖ How to use?



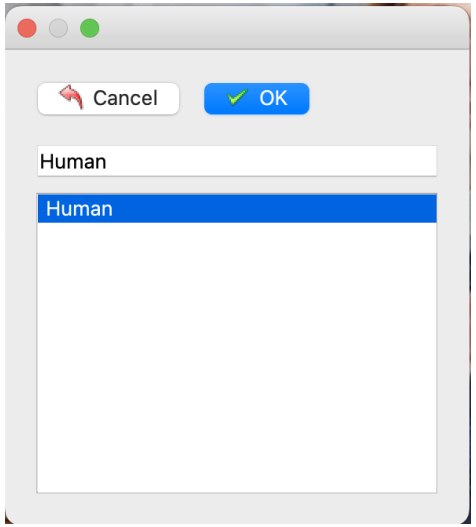
labelImg

❖ How to use?

1. Click "RectBox"

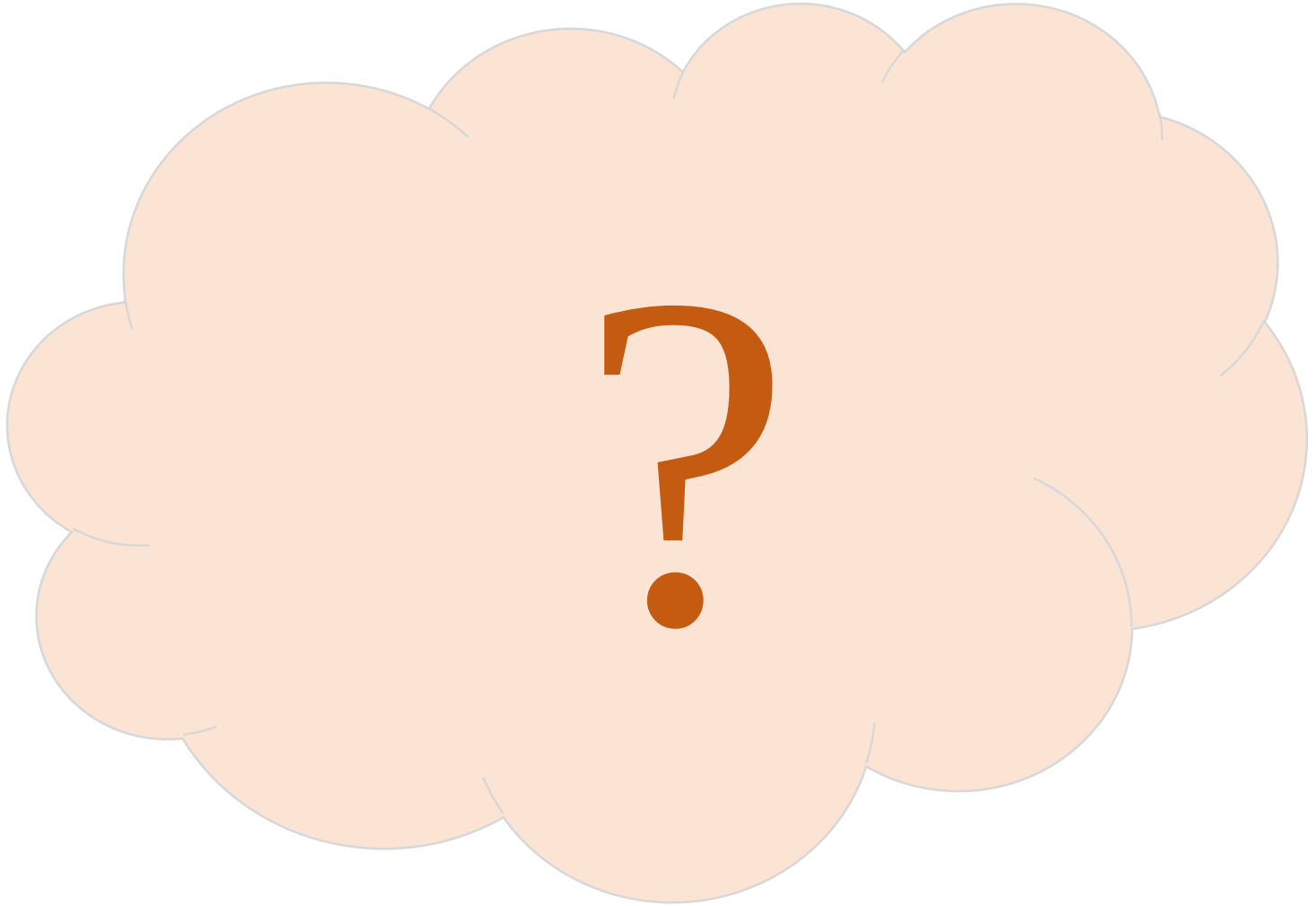
2. Draw the rectangle so that it fits the object

3. Choose the right class's name for the object



Iterate the process with other objects

Question




Thank you!