

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

Object Detection with YOLOv8

Project



AI VIET NAM
[@aivietnam.edu.vn](https://aivietnam.edu.vn)

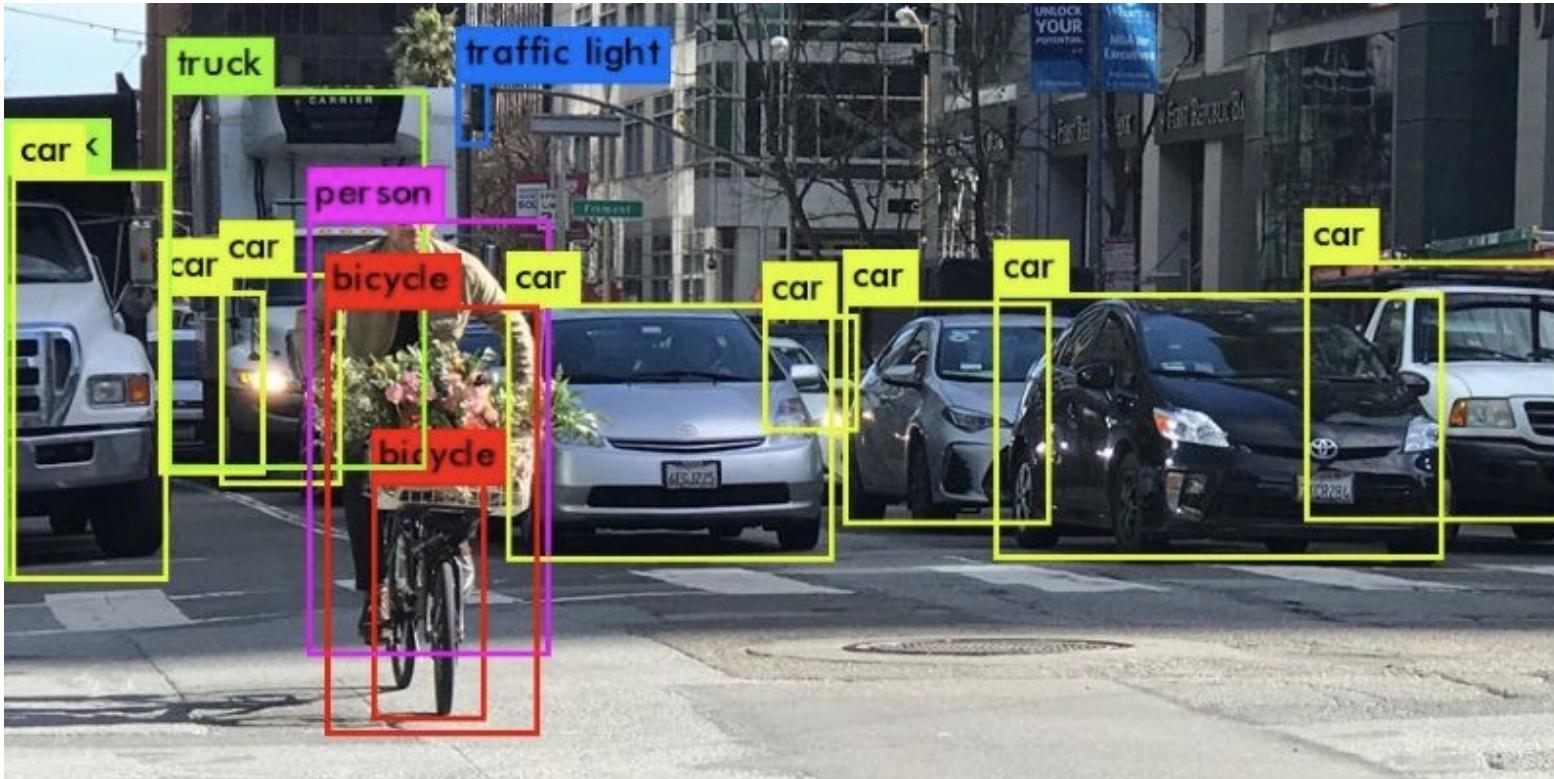
Dinh-Thang Duong – TA

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?

Classification



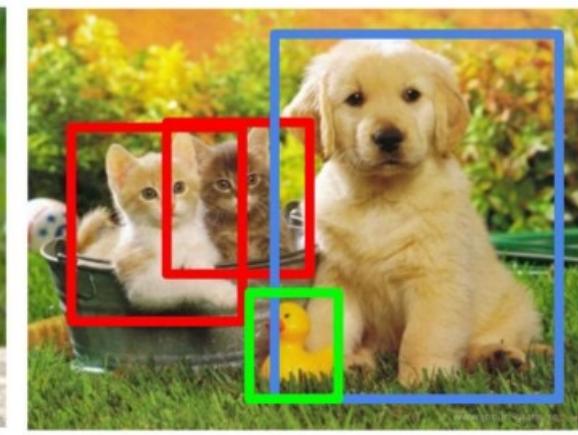
CAT

Classification + Localization



CAT

Object Detection

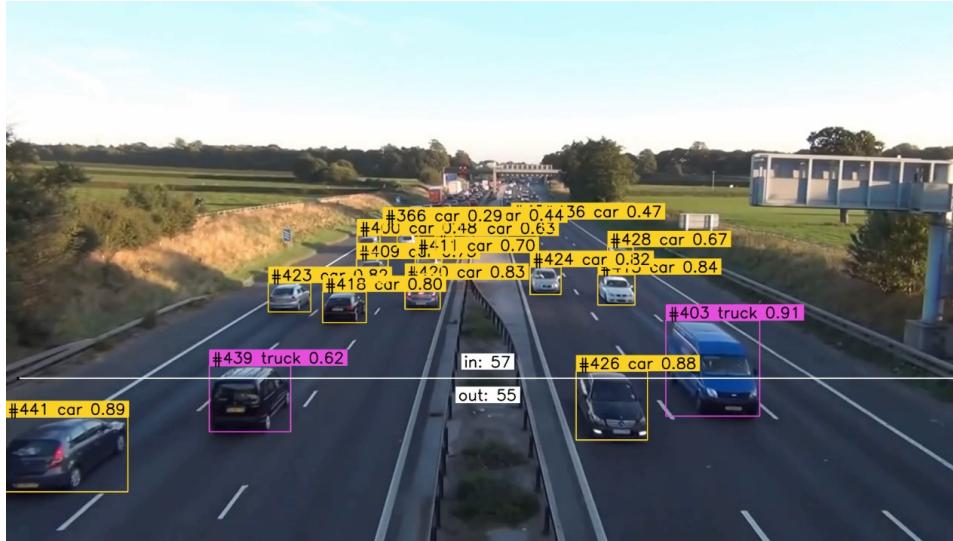
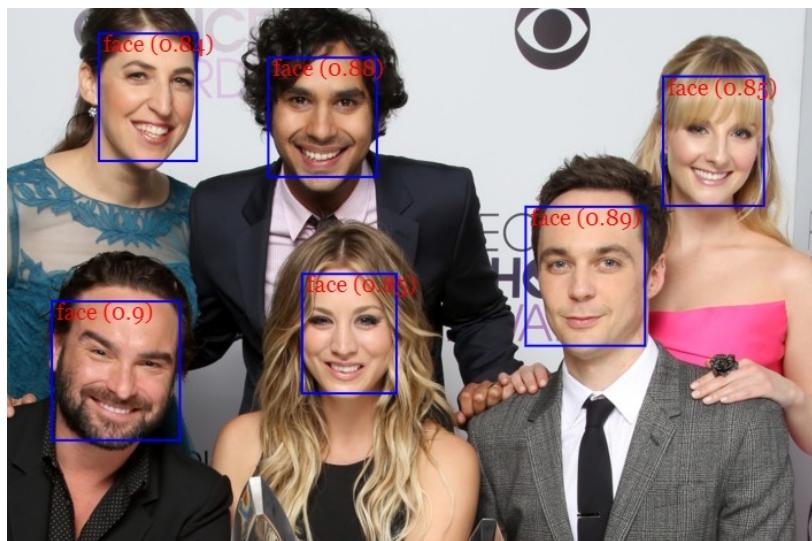
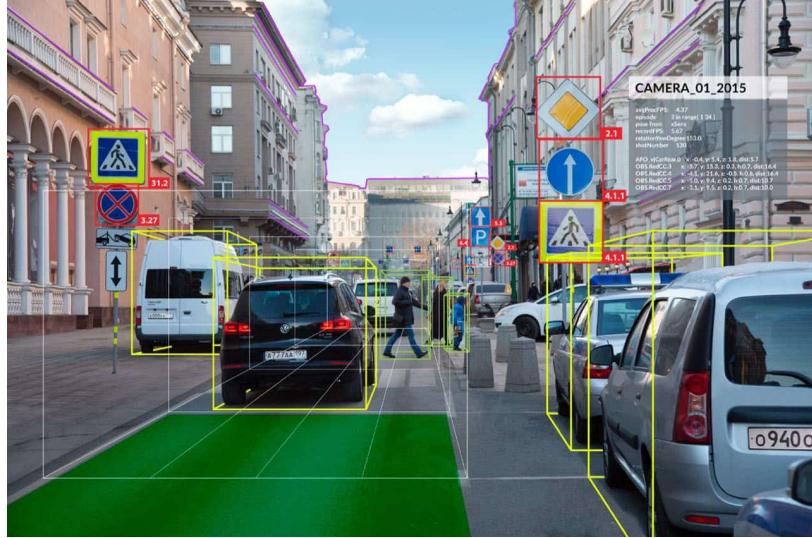


CAT, DOG, DUCK

Object Detection = Classification + Localization

Introduction

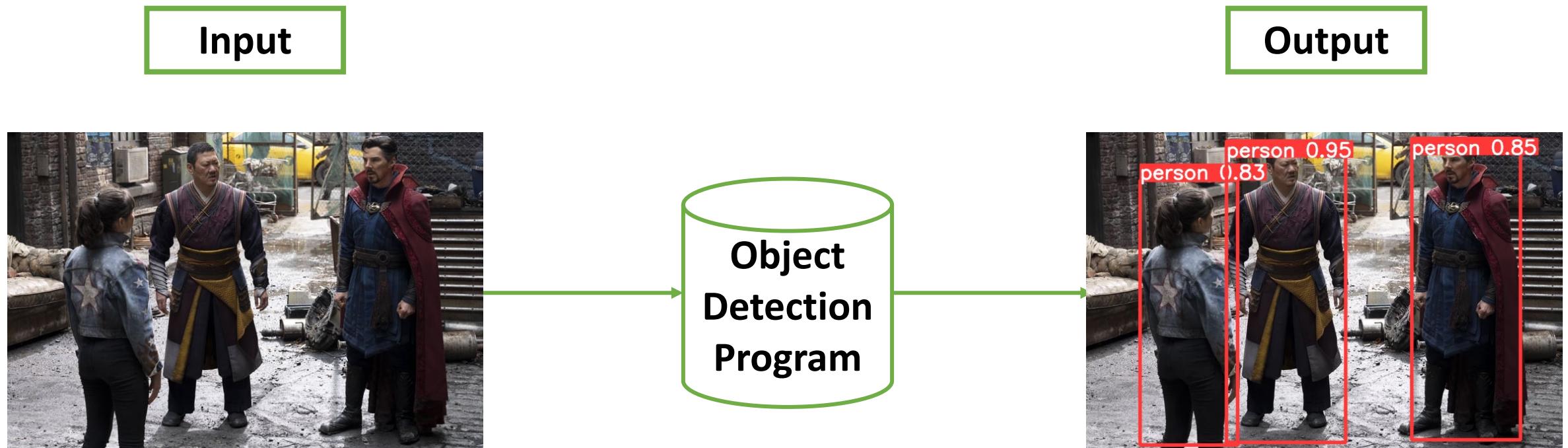
❖ Object Detection Applications



**Some Object
Detection
Applications**

Introduction

❖ Human Detection Problem



Introduction

❖ Human Detection Demo



Outline

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

Code Environment

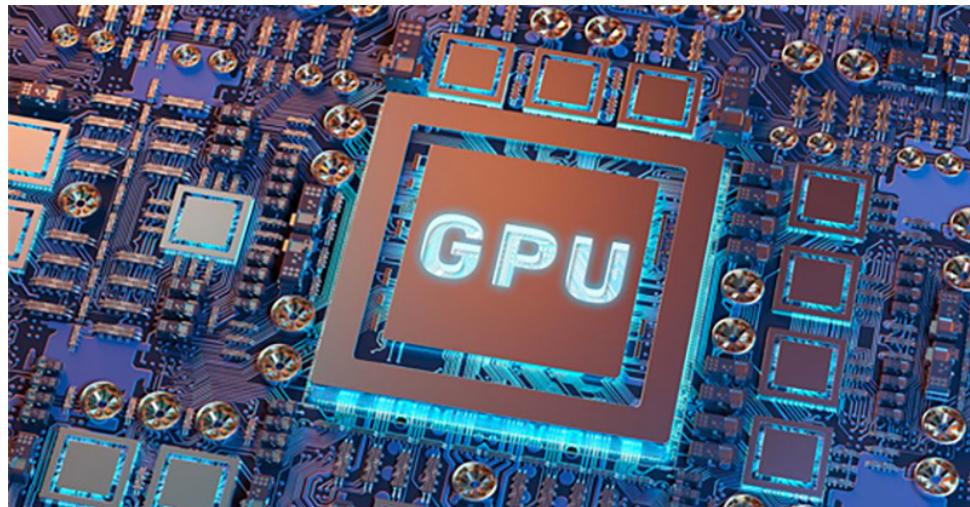
❖ Google Colab Introduction



Google Colaboratory: A data analysis and machine learning tool that allows us to combine executable Python code and rich text.

Code Environment

❖ Google Colab Advantages



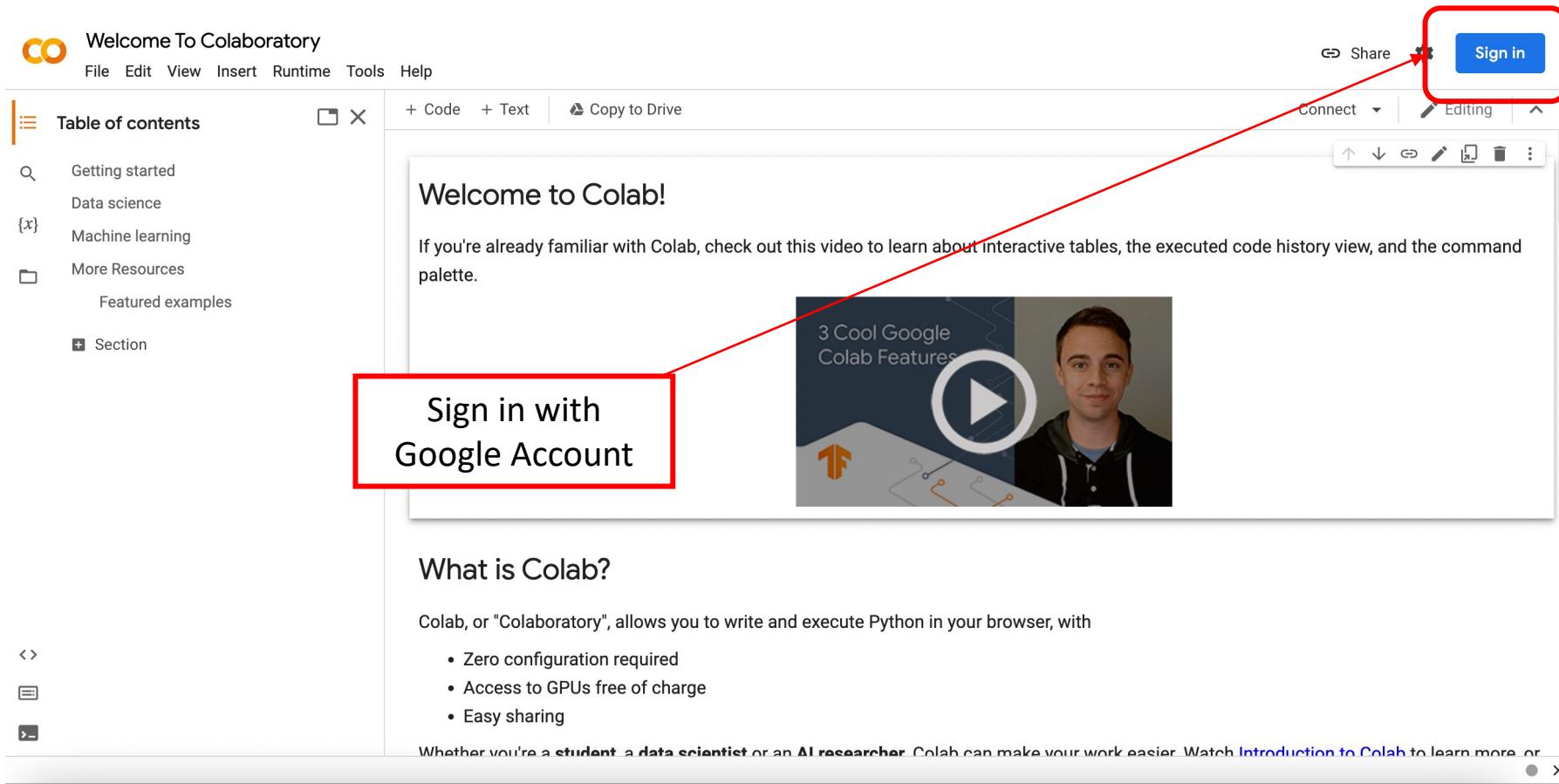
- **Free GPU.**
- **Pre-installed popular ML packages.**
- **Notebook interface.**

```
1 !pip list
```

tensorflow	2.12.0
tensorflow-datasets	4.9.2
tensorflow-estimator	2.12.0
tensorflow-gcs-config	2.12.0
tensorflow-hub	0.13.0
tensorflow-io-gcs-filesystem	0.32.0
tensorflow-metadata	1.13.1
tensorflow-probability	0.20.1
tensorstore	0.1.36
termcolor	2.3.0
terminado	0.17.1
text-unidecode	1.3
textblob	0.17.1
tf-slim	1.1.0
thinc	8.1.9
threadpoolctl	3.1.0
tifffile	2023.4.12
tinyccss2	1.2.1
toml	0.10.2
tomli	2.0.1
toolz	0.12.0
torch	2.0.1+cull18
torchaudio	2.0.2+cull18
torchdata	0.6.1
torchsummary	1.5.1
torchtext	0.15.2
torchvision	0.15.2+cull18

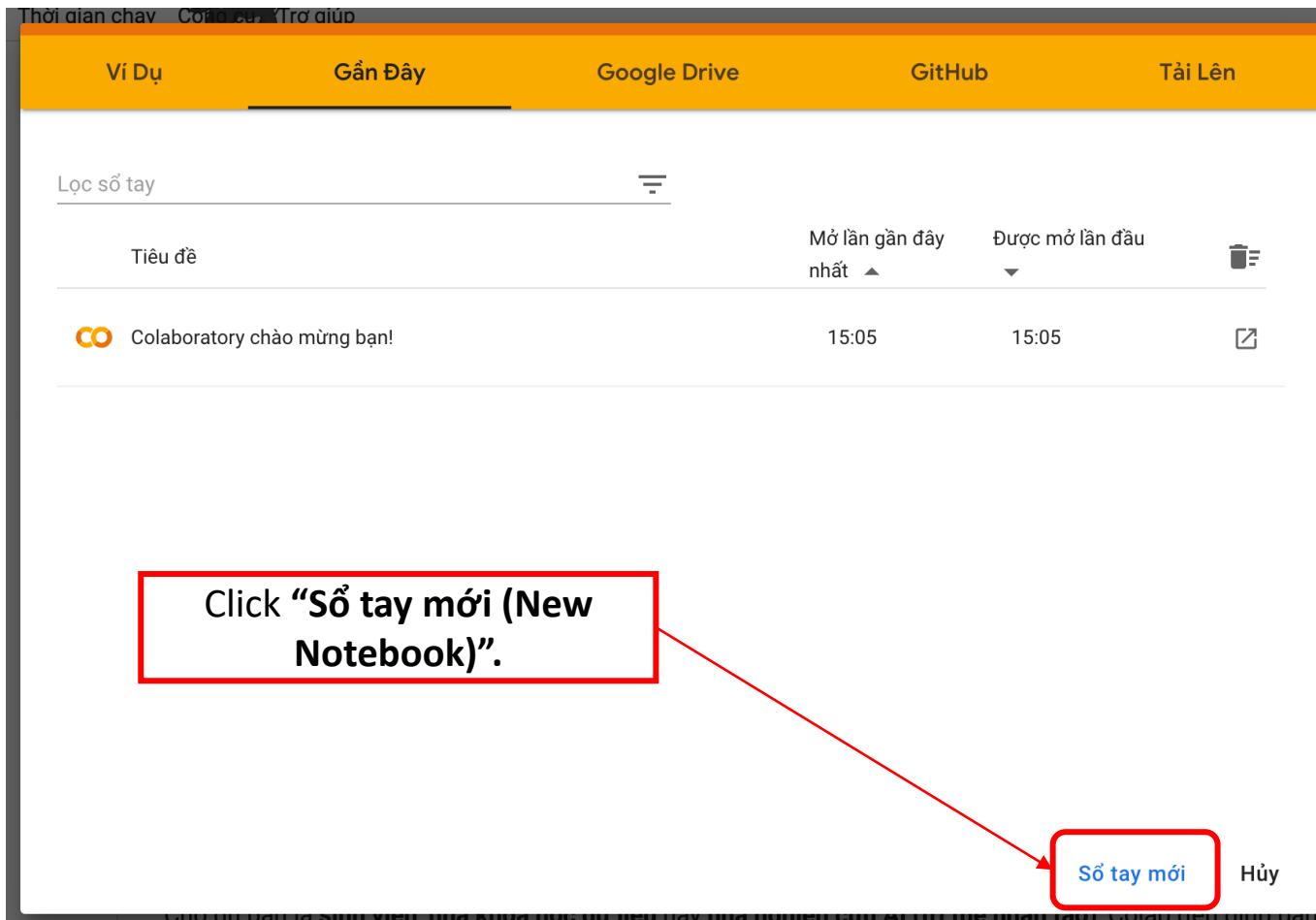
Code Environment

❖ Google Colab Instruction



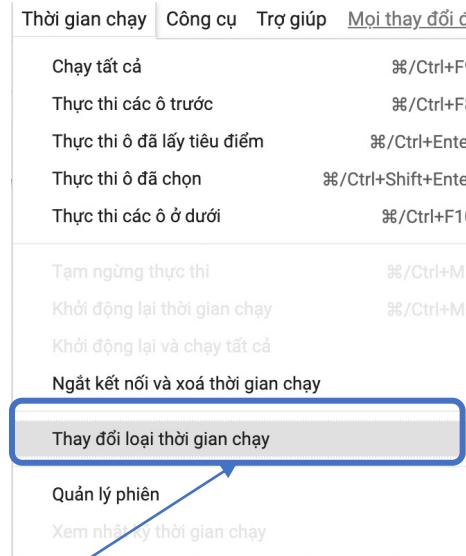
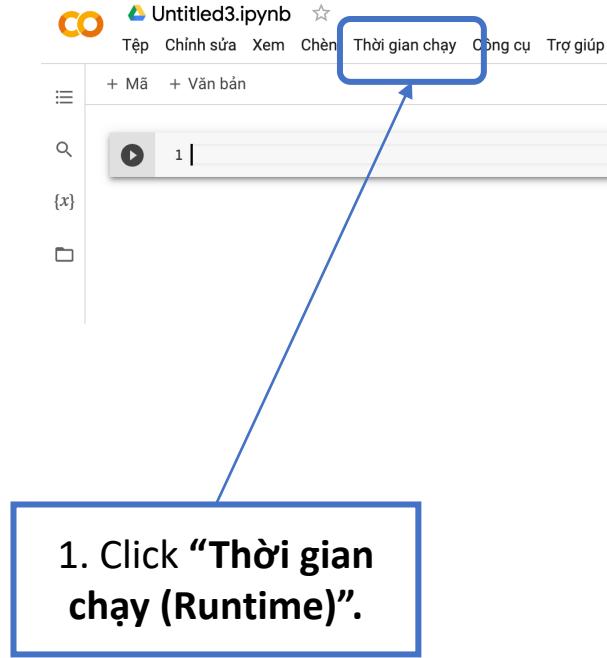
Code Environment

❖ Google Colab Instruction

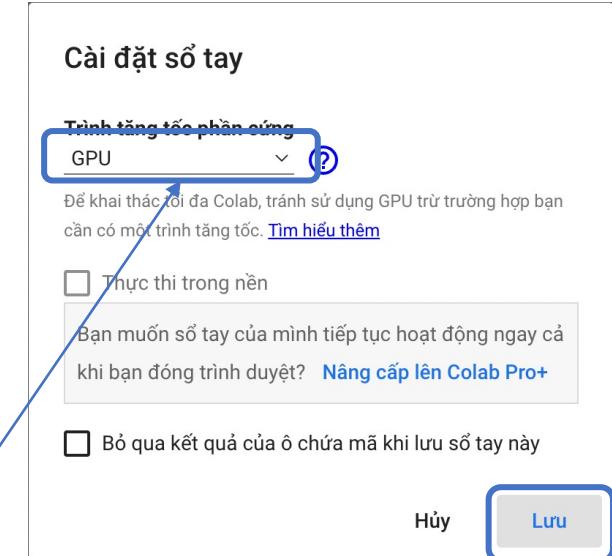


Code Environment

❖ Google Colab Instruction

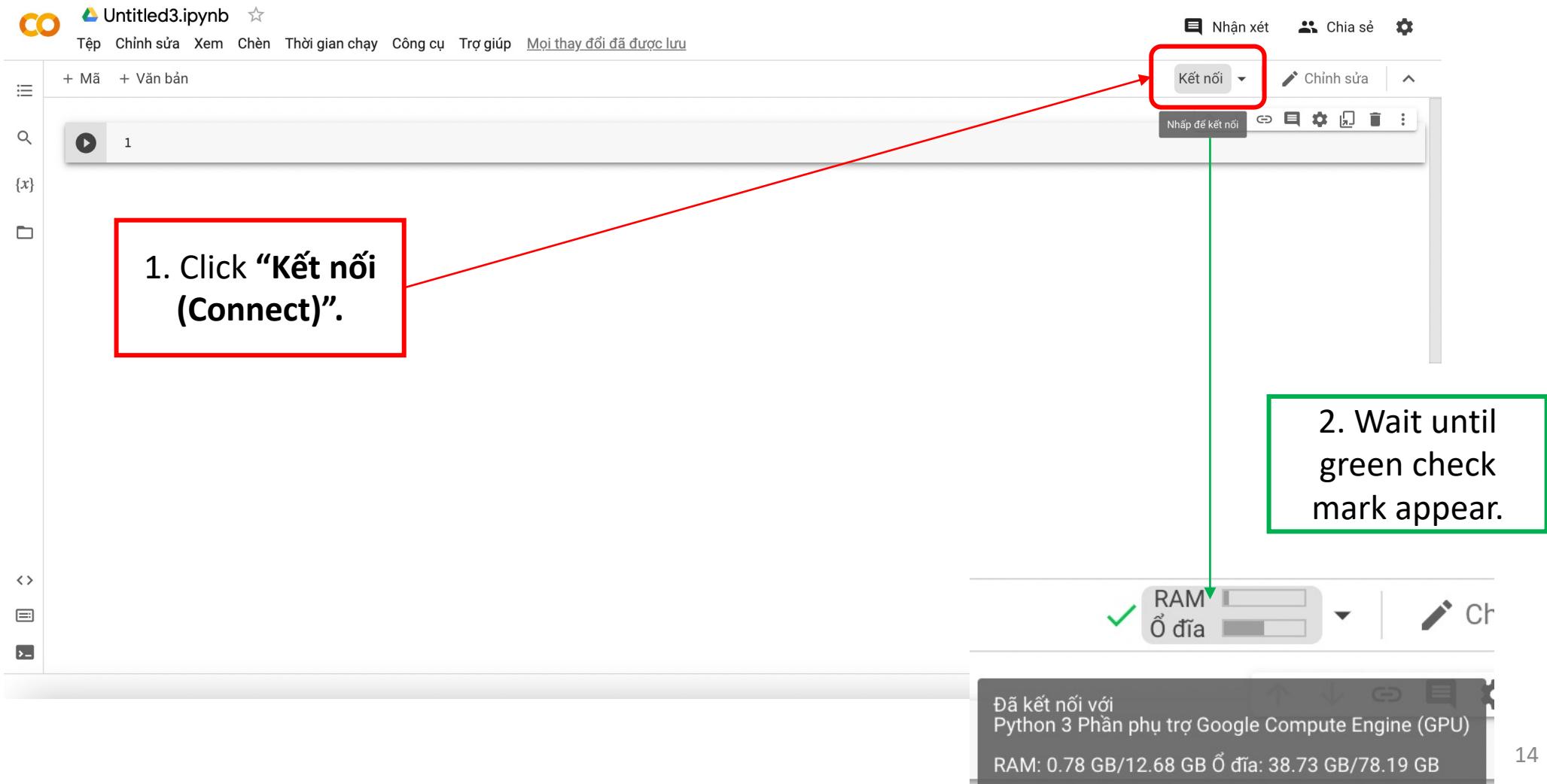


3. Click to change from **None** to **GPU**.



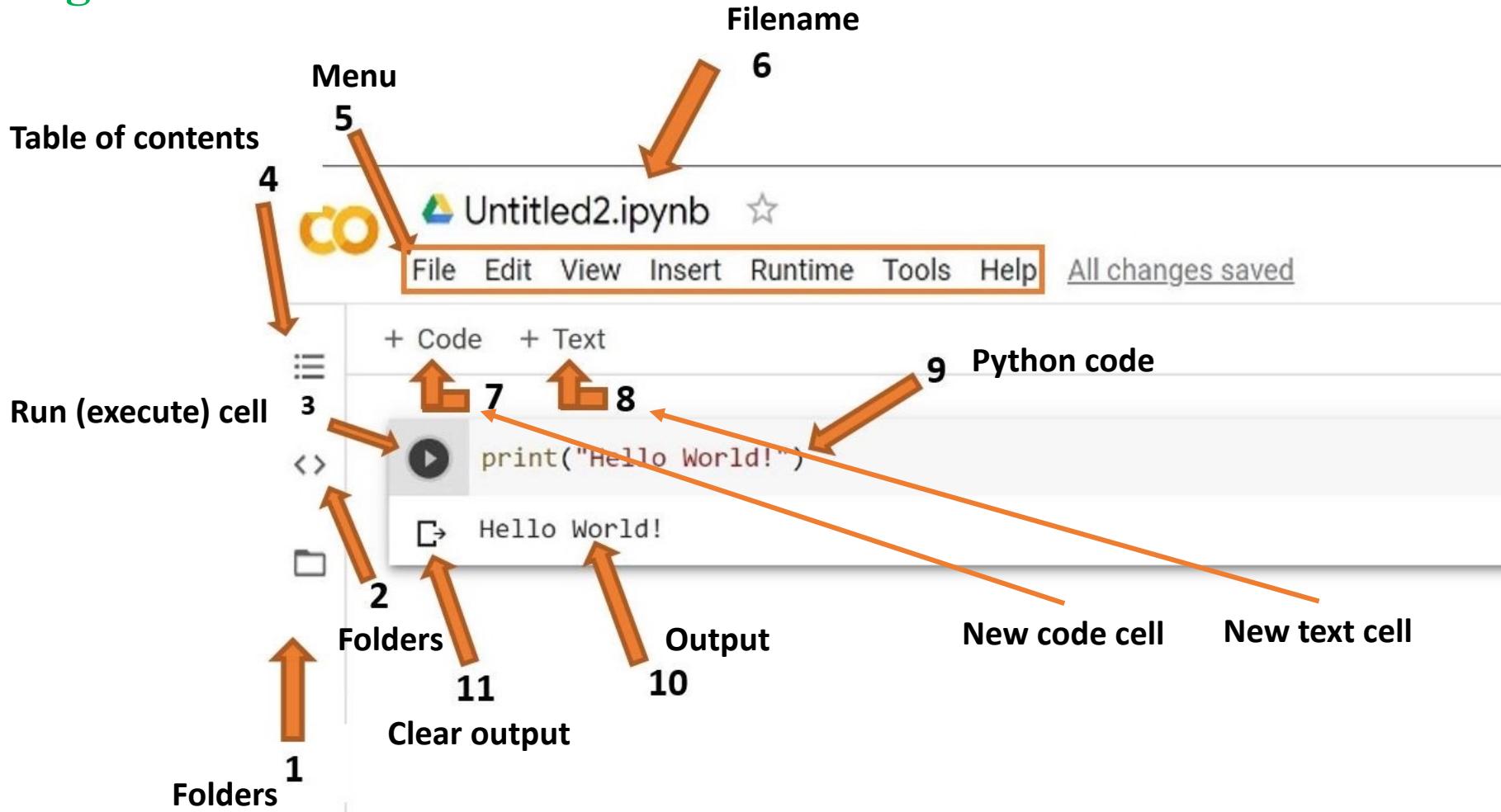
Code Environment

❖ Google Colab Instruction



Code Environment

❖ Google Colab Interface



Outline

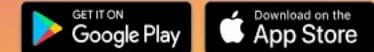
- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

YOLOv8 – How to use

❖ Introduction



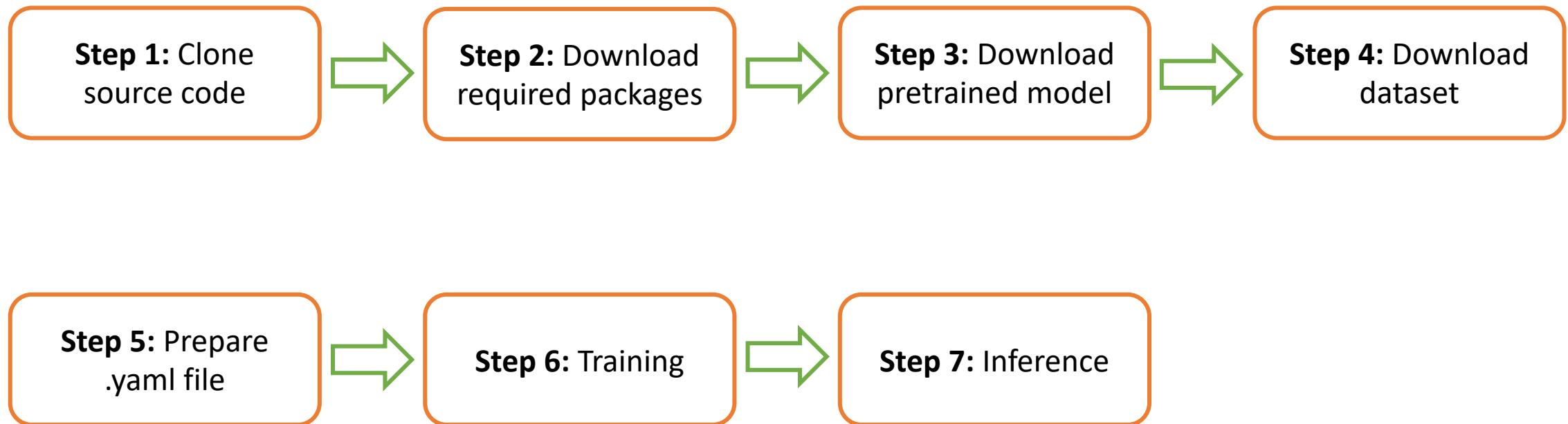
DOWNLOAD THE APP



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

YOLOv8 – How to use

❖ YOLOv8 Installation and Usage



YOLOv8 – How to use

❖ Step 1: Clone YOLOv8 source code

The screenshot shows the GitHub repository page for ultralytics / ultralytics. The 'Code' tab is selected. At the top, there are buttons for 'Sponsor', 'Watch 60', 'Fork 1.5k', and 'Star 7.9k'. Below the header, there are links for 'Issues 610', 'Pull requests 46', 'Discussions', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. A dropdown menu shows 'main' branch, 21 branches, and 0 tags. Buttons for 'Go to file', 'Add file', 'Code', and 'Use this template' are present. The main area displays a list of recent commits:

Commit	Message	Time Ago
.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

On the right side, there is an 'About' section with the following text:
NEW - YOLOv8 🚀 in PyTorch > ONNX >
CoreML > TFLite

Links include: docs.ultralytics.com, machine-learning, deep-learning, hub, pytorch, yolo, image-classification, object-detection, yolov3, yolov5, ultralytics, yolov8.

Other repository options: Readme, AGPL-3.0 license, Code of conduct, Security policy, Cite this repository, 7.9k stars.

YOLOv8 GitHub Repository

YOLOv8 – How to use

❖ Step 1.1: About GitHub

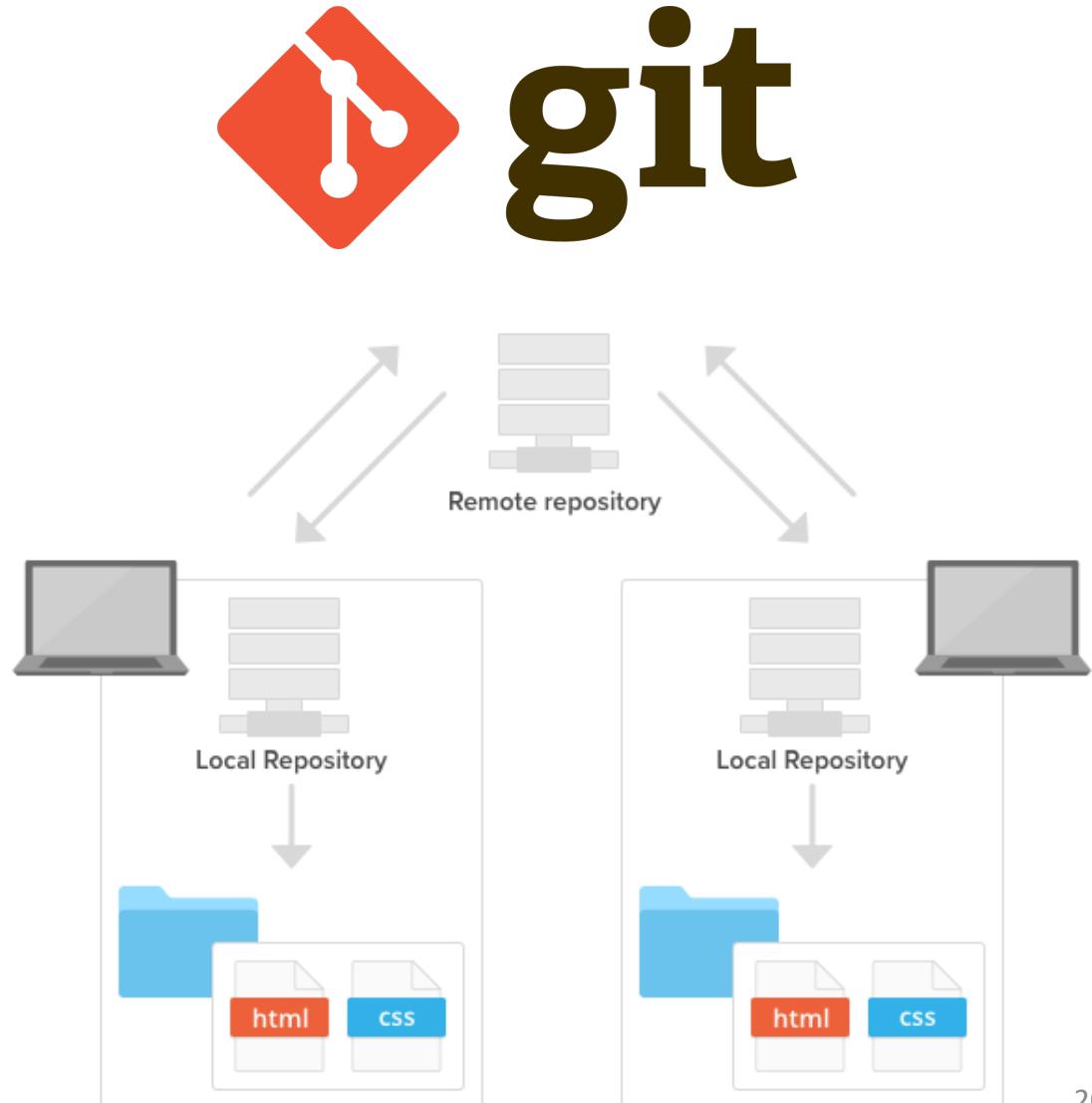


GitHub

A code hosting platform for collaboration and version control

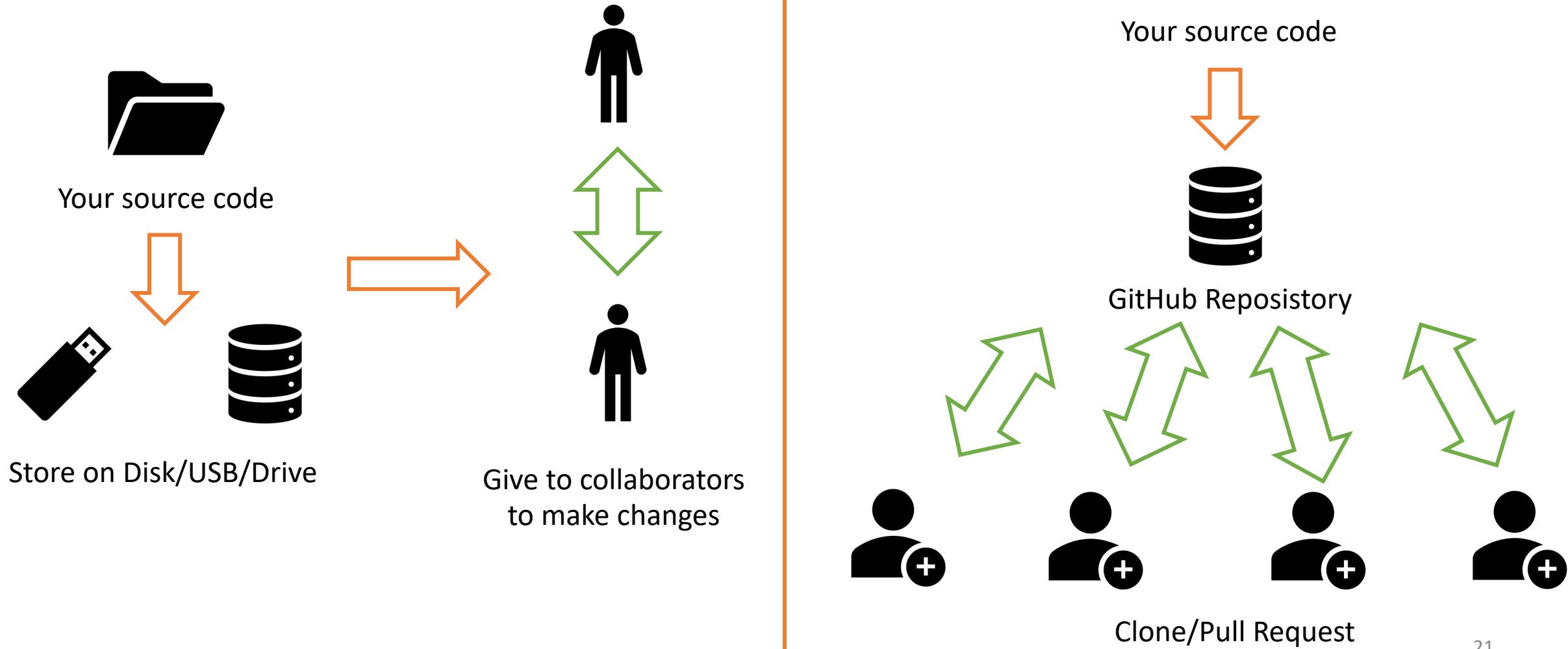
GitHub usages:

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



YOLOv8 – How to use

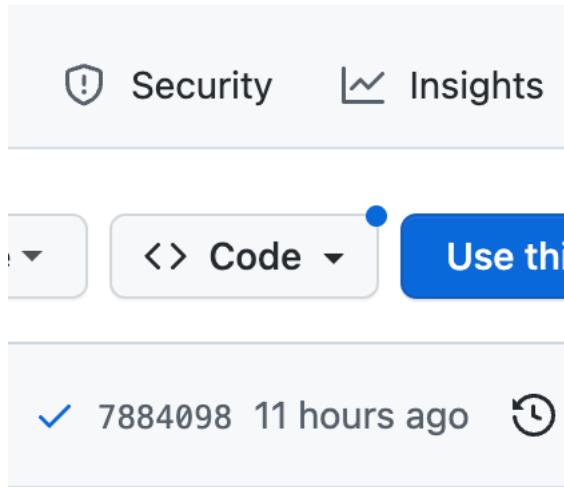
❖ Step 1.1: About GitHub



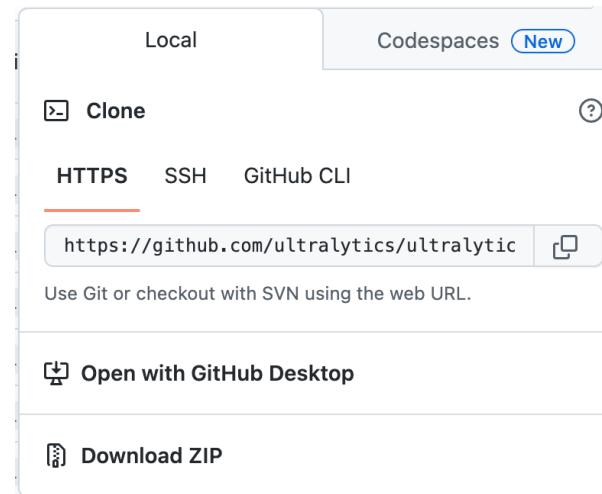
YOLOv8 – How to use

❖ Step 1: Clone YOLOv8 source code

Step 1: Click Code



Step 2: Copy link

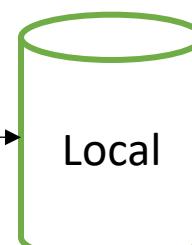


Step 3: Clone source code

```
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:

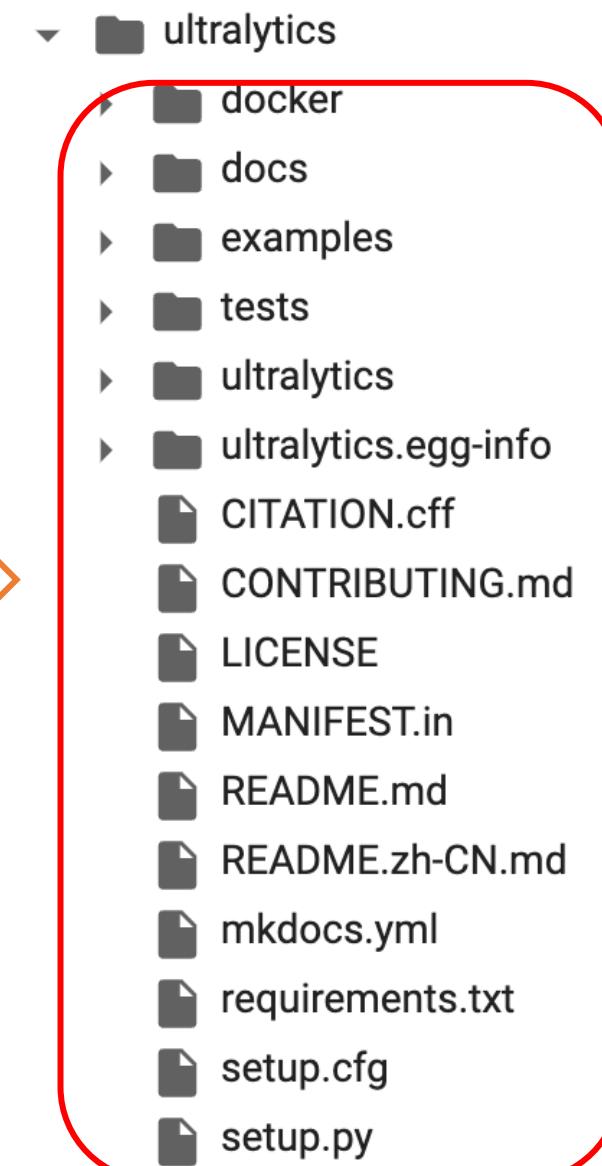
git clone <link>: Clone (copy) a repository into a new directory.



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

- ▼ ultralytics
 - ▶ docker
 - ▶ docs
 - ▶ examples
 - ▶ tests
 - ▶ ultralytics
 - ▶ ultralytics.egg-info
 - CITATION.cff
 - CONTRIBUTING.md
 - LICENSE
 - MANIFEST.in
 - README.md
 - README.zh-CN.md
 - mkdocs.yml
 - requirements.txt
 - setup.cfg
 - setup.py

```
requirements.txt ×  
1 # Ultralytics requirements  
2 # Usage: pip install -r requirements.txt  
3  
4 # Base -----  
5 matplotlib>=3.2.2  
6 opencv-python>=4.6.0  
7 Pillow>=7.1.2  
8 PyYAML>=5.3.1  
9 requests>=2.23.0  
10 scipy>=1.4.1  
11 torch>=1.7.0  
12 torchvision>=0.8.1  
13 tqdm>=4.64.0  
14  
15 # Logging -----  
16 # tensorboard>=2.13.0  
17 # clearml  
18 # comet  
19  
20 # Plotting -----  
21 pandas>=1.1.4  
22 seaborn>=0.11.0
```



Download all
theses 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

Files



You are here

..
sample_data
ultralytics

1 !pwd
/content



sample_data
ultralytics

1 %cd ultralytics
/content/ultralytics

Move to here

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 ultralytics>=8.0.0)
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: torchvision>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.0)

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.

YOLOv8 – How to use

❖ Step 4: Download dataset

Image:



Label:

frame007.25.00-07.30.00.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

```
└── human_detection_dataset
    ├── train
    │   ├── images
    │   ├── labels
    │   └── labels.cache
    └── val
        ├── images
        ├── labels
        └── labels.cache
```

- frame007.25.00-07.30.00.jpg
- frame007.30.00-07.35.00.jpg
- frame007.35.00-07.40.00.jpg
- frame007.40.00-07.45.00.jpg
- frame007.25.00-07.30.00.txt
- frame007.30.00-07.35.00.txt
- frame007.35.00-07.40.00.txt
- frame007.40.00-07.45.00.txt

- 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:

```
1 !gdown --id 1--0QuKMwj31K-CSvD8oq5fceFweiFPuN&
2 !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:

```
1 from google.colab import drive
2
3 drive.mount('/content/gdrive')
```

/content
Mounted at /content/gdrive

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

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

YOLOv8 – How to use

❖ Step 4: Download dataset

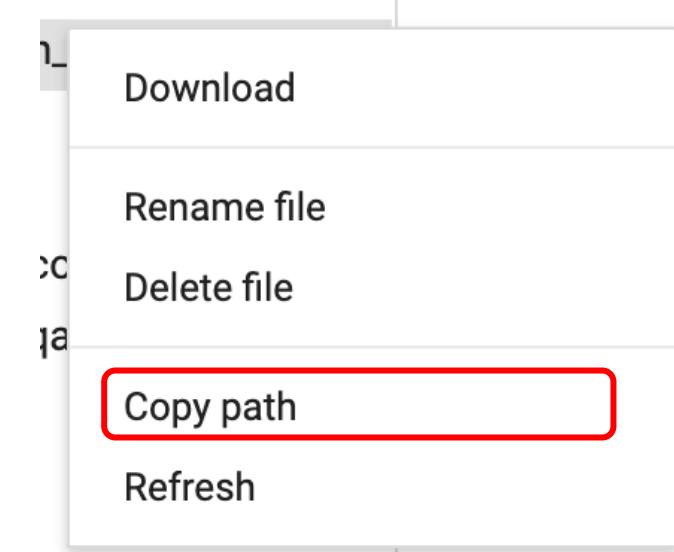
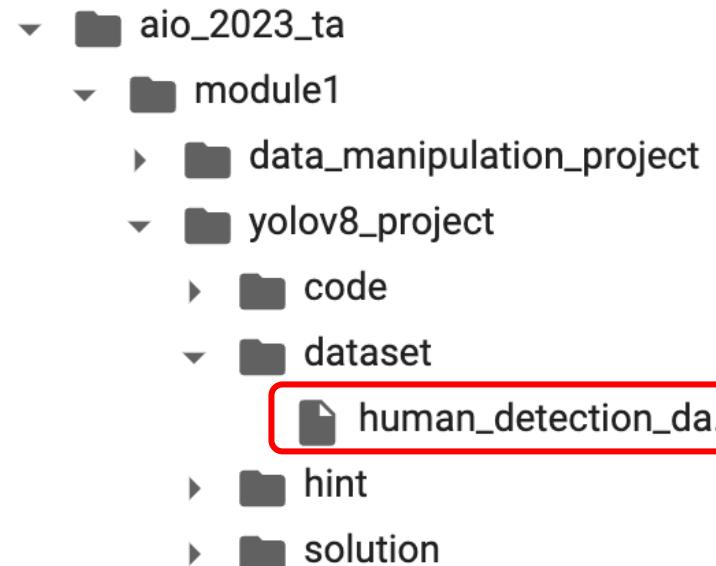
2. Copy from google drive example

```
1 from google.colab import drive  
2  
3 drive.mount('/content/gdrive')
```

/content

Mounted at /content/gdrive

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



```
path_to_human_dataset.zip =  
/content/gdrive/MyDrive/Coordinate/aio_2023_ta/module1/yolov8_pro  
ject/dataset/human_detection_dataset.zip
```

YOLOv8 – How to use

❖ Step 5: Prepare .yaml file

```
data
  ▶ hyps
  ▶ images
  ▶ scripts
  ▶ Argoverse.yaml
  ▶ GlobalWheat2020.yaml
  ▶ Objects365.yaml
  ▶ SKU-110K.yaml
  ▶ VOC.yaml
  ▶ VisDrone.yaml
  ▶ coco.yaml highlighted
  ▶ coco128.yaml
  ▶ helmet.yaml
  ▶ market.yaml
  ▶ xView.yaml
```

```
coco.yaml ×
1 # YOLOv5 🚀 by Ultralytics, GPL-3.0 license
2 # COCO 2017 dataset http://cocodataset.org by Microsoft
3 # Example usage: python train.py --data coco.yaml
4 # parent
5 #   └── yolov5
6 #     └── datasets
7 #       └── coco ← downloads here (20.1 GB)
8
9
10 # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt
11 path: ..//datasets/coco # dataset root dir
12 train: train2017.txt # train images (relative to 'path') 118287 images
13 val: val2017.txt # val images (relative to 'path') 5000 images
14 test: test-dev2017.txt # 20288 of 40670 images, submit to https://compe
15
16 # Classes
17 nc: 80 # number of classes
18 names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'tr
19   'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', '
20   'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella',
21   'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'bas
22   'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife',
23   'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
24   'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop',
25   'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book',
26   'hair drier', 'toothbrush'] # class names
27
```

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

```
1 import yaml
2 import os
3
4 dataset_info = {
5     'train': './train/images',
6     'val': './val/images',
7     'path': os.path.abspath('./human_detection_dataset'),
8     'nc': 1,
9     'names': ['Human']
10 }
11
12 yaml_filepath = './human_detection_dataset/data.yaml'
13 with open(yaml_filepath, 'w') as f:
14     doc = yaml.dump(
15         dataset_info,
16         f,
17         default_flow_style=None,
18         sort_keys=False
19     )
```

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.

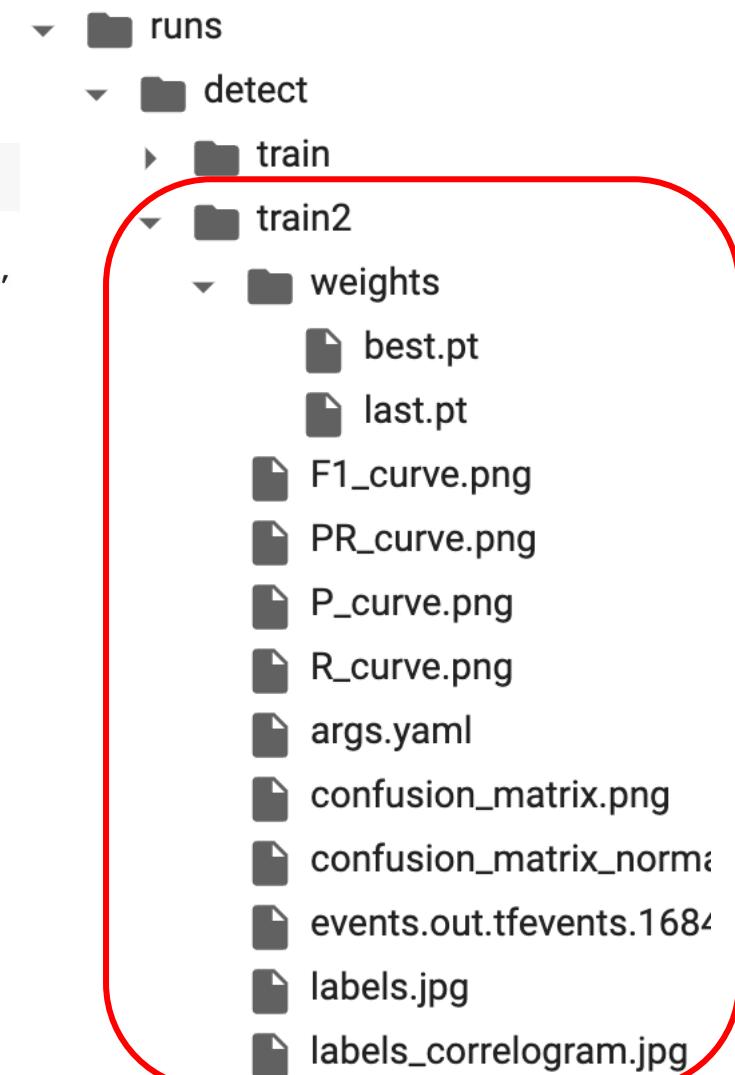
```
Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
19/20 4.05G 0.7462 0.4929 0.9263 158 640: 100% 139/139 [04:49<00:00, 2.08s/it]
          Class Images Instances Box(P) R mAP50 mAP50-95: 100% 52/52 [01:06<00:00, 1.28s/it]
          all 1642 13171 0.86 0.745 0.843 0.588

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 0.956 0.742 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

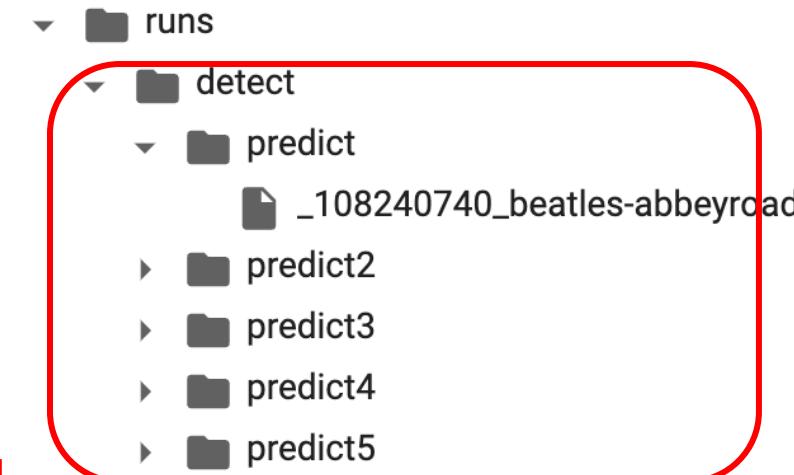
Validating /content.ultralytics/runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.104 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients
          Class      Images    Instances    Box(P       R           mAP50  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
```

Predict command

```
1 # With uploaded image
2 !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
```

Path to trained weight file (.pt)



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 <input checked="" type="checkbox"/>	'vid.mp4'	str, Path	
directory <input checked="" type="checkbox"/>	'path/'	str, Path	
glob <input checked="" type="checkbox"/>	'path/*.jpg'	str	Use * operator
YouTube <input checked="" type="checkbox"/>	'https://youtu.be/Zgi9g1ksQHc'	str	
stream <input checked="" type="checkbox"/>	'rtsp://example.com/media.mp4'	str	RTSP, RTMP, HTTP

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

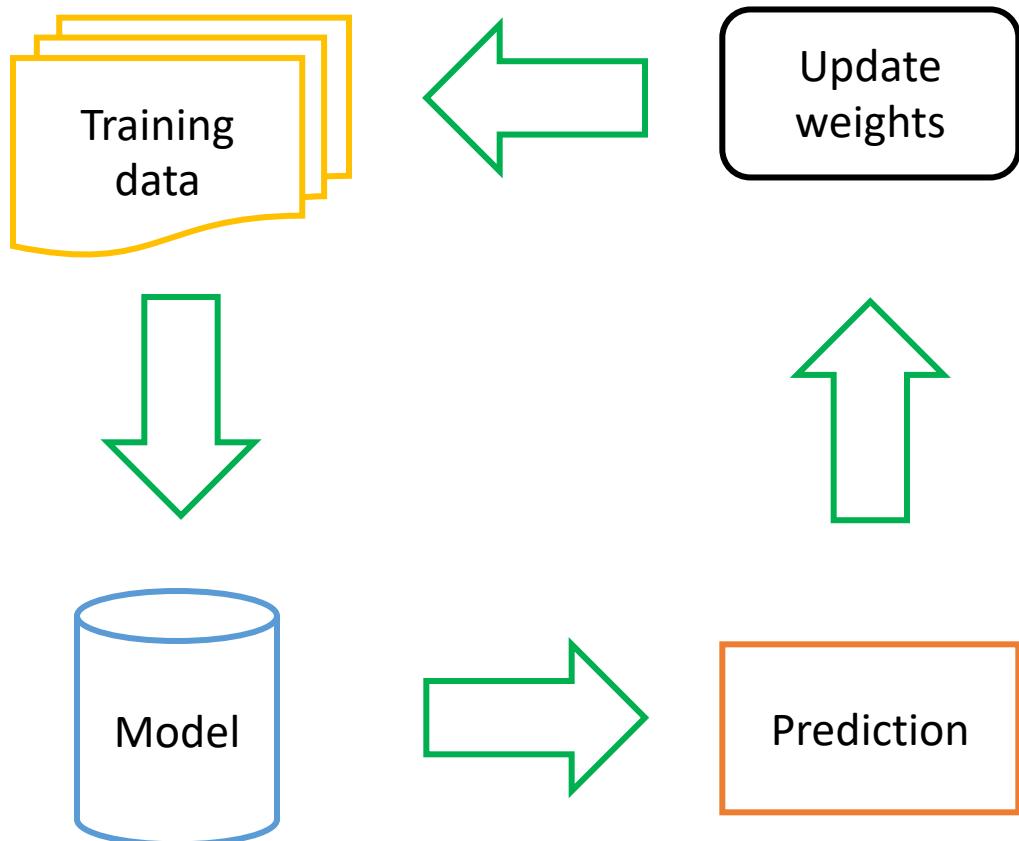
```
1 # With uploaded image
2 !yolo predict model=yolov8n.pt source=filepath
```

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

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

YOLOv8 – How to use

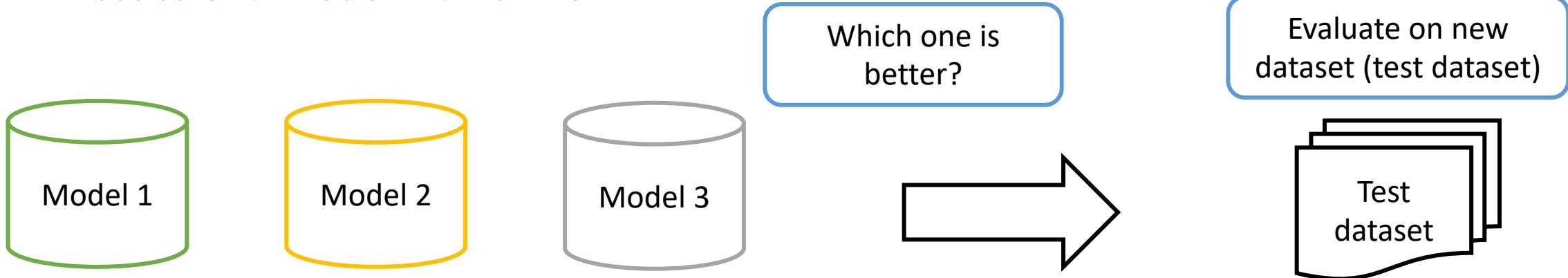
❖ Discussion: Training command flags meaning



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

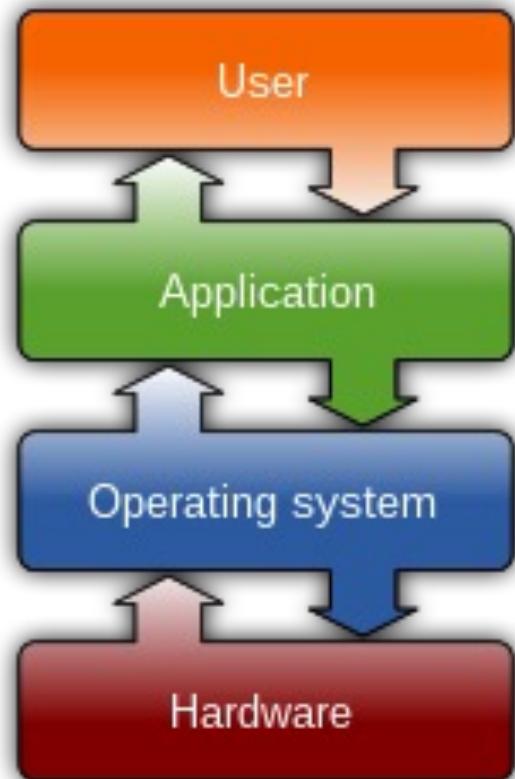


macOS

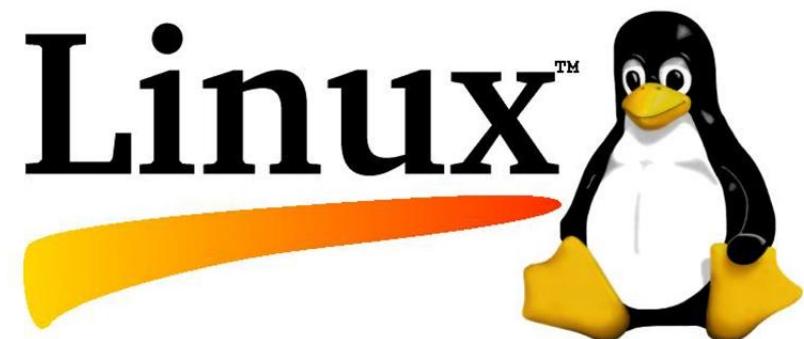
These are called Operating System (OS)

Linux Commands (Colab)

❖ Introduction



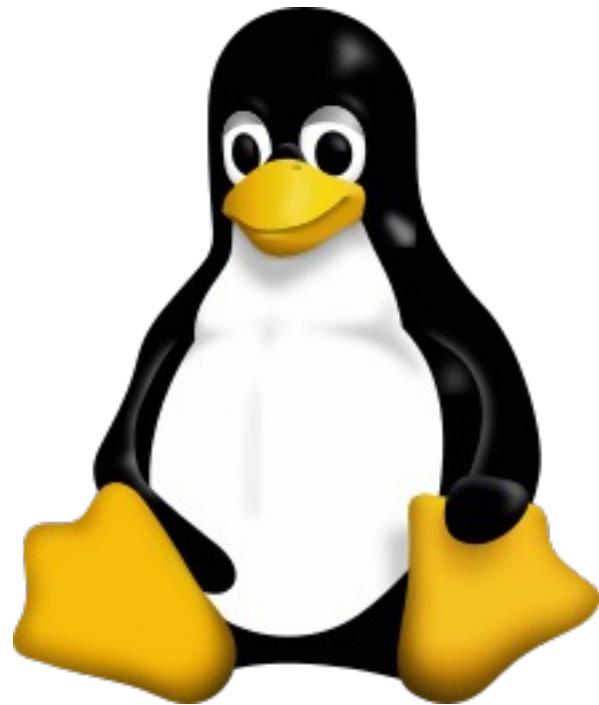
Operating System: A system software that manages computer hardware, software resources...



Some famous operating systems

Linux Commands (Colab)

❖ Linux



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



debian



Ubuntu



CentOS



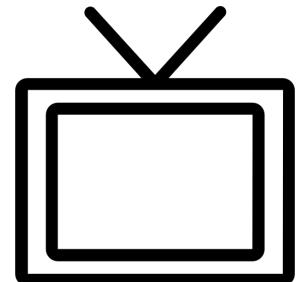
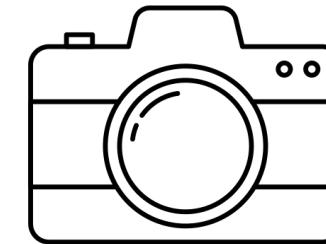
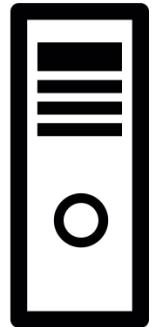
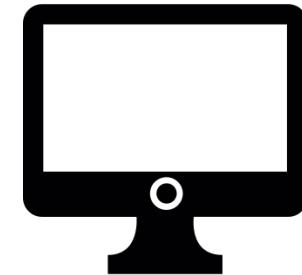
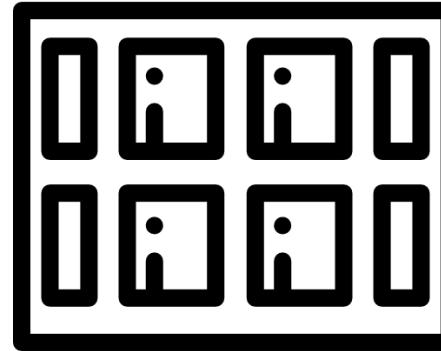
Linux Distributions: Operating System that based on Linux

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



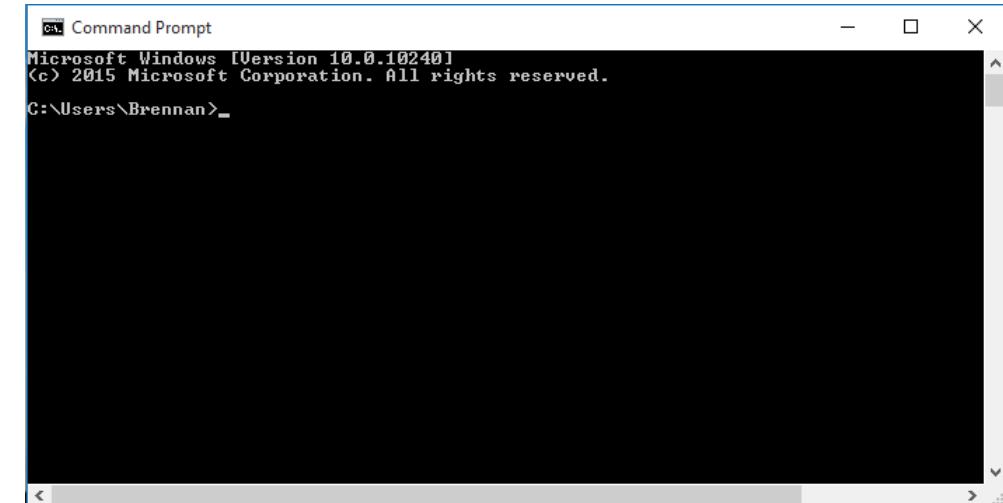
Lots of machines use Linux

Linux Commands (Colab)

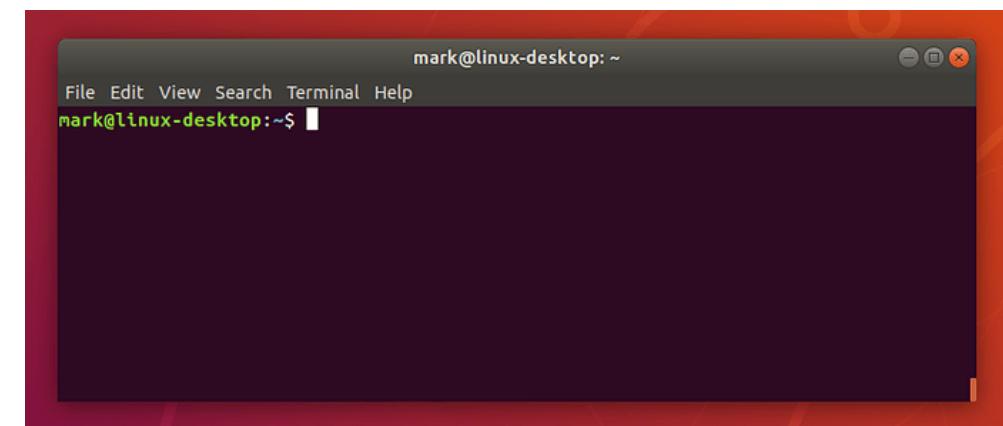
❖ Command Line Interface (CLI)

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmta.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmta.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
rpmfusion-free-updates/primary_db
rpmfusion-nonfree-updates
updates/metalink
updates
updates/primary_db
    73% [=====] 62 kB/s | 2.6 MB 00:15 ETA
| 2.7 kB 00:00
| 206 kB 00:04
| 2.7 kB 00:00
| 5.9 kB 00:00
| 4.7 kB 00:00
```

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



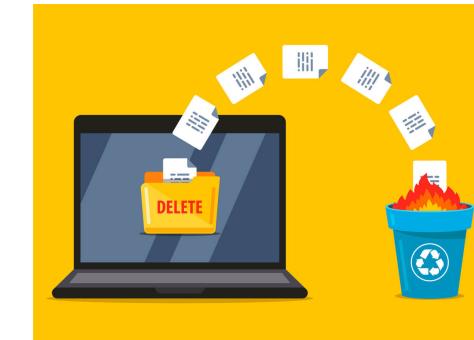
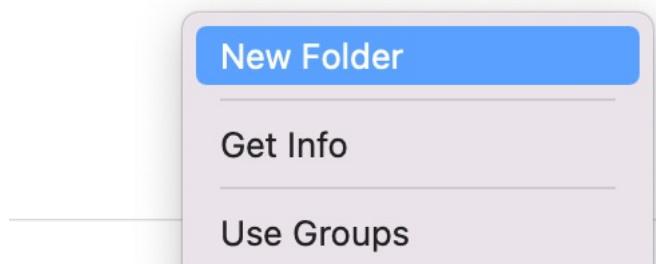
Windows Command Line (cmd)



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 % ]
```



new_folder

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 % ]
```

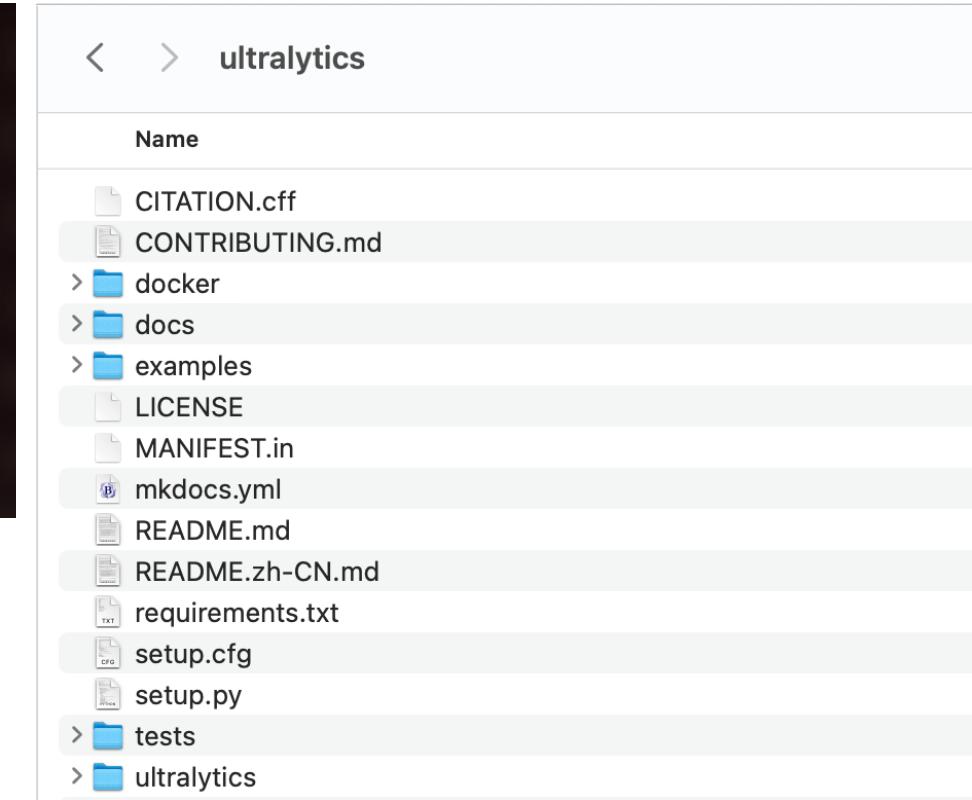
argument: Input given to a command line

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

Linux Commands (Colab)

❖ Common Linux Commands: ls

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

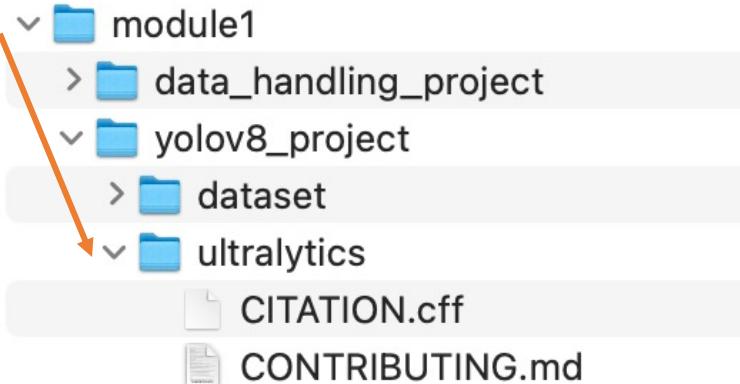


- ❖ **Command:** ls
- ❖ **Usage:** To list all file/folder names in current directory.

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
GlobalWheat2020.yaml VisDrone.yaml
Objects365.yaml     coco.yaml
SKU-110K.yaml       coco128.yaml
[thangduong@Duongs-MacBook-Pro data % cd ..
[thangduong@Duongs-MacBook-Pro yolov5 % ls
CONTRIBUTING.md    export.py
LICENSE              hubconf.py
README.md            models
data                requirements.txt
detect.py           setup.cfg
thangduong@Duongs-MacBook-Pro yolov5 %
```

Move to “1 level up” directory

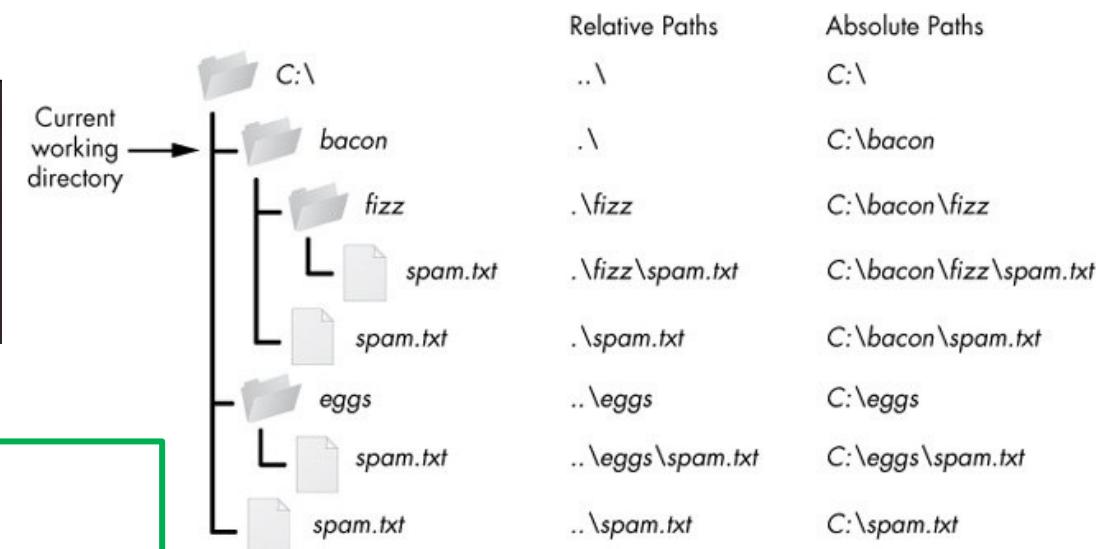
```
[thangduong@Duongs-MacBook-Pro yolov5 % cd ..
[thangduong@Duongs-MacBook-Pro ~ % pwd
/Users/thangduong
thangduong@Duongs-MacBook-Pro ~ %
```

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
[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
README.md         requirements.txt
README.zh-CN.md   setup.cfg
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

❖ 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
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
(base) thangduong@Duongs-MacBook-Pro ultralytics %
```

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

E.g: Delete tutorial.ipynb in current 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
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
MANIFEST.in                           mkdocs.yml
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
datasets                             overrides
(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
LICENSE                                docs
MANIFEST.in                           examples
README.md                            mkdocs.yml
(base) thangduong@Duongs-MacBook-Pro ultralytics % mkdir abc
(base) thangduong@Duongs-MacBook-Pro ultralytics % ls
CITATION.cff                         abc                         setup.cfg
CONTRIBUTING.md                      docker
LICENSE                                docs
MANIFEST.in                           examples
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.

```
[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 %
```

❖ **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 % 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

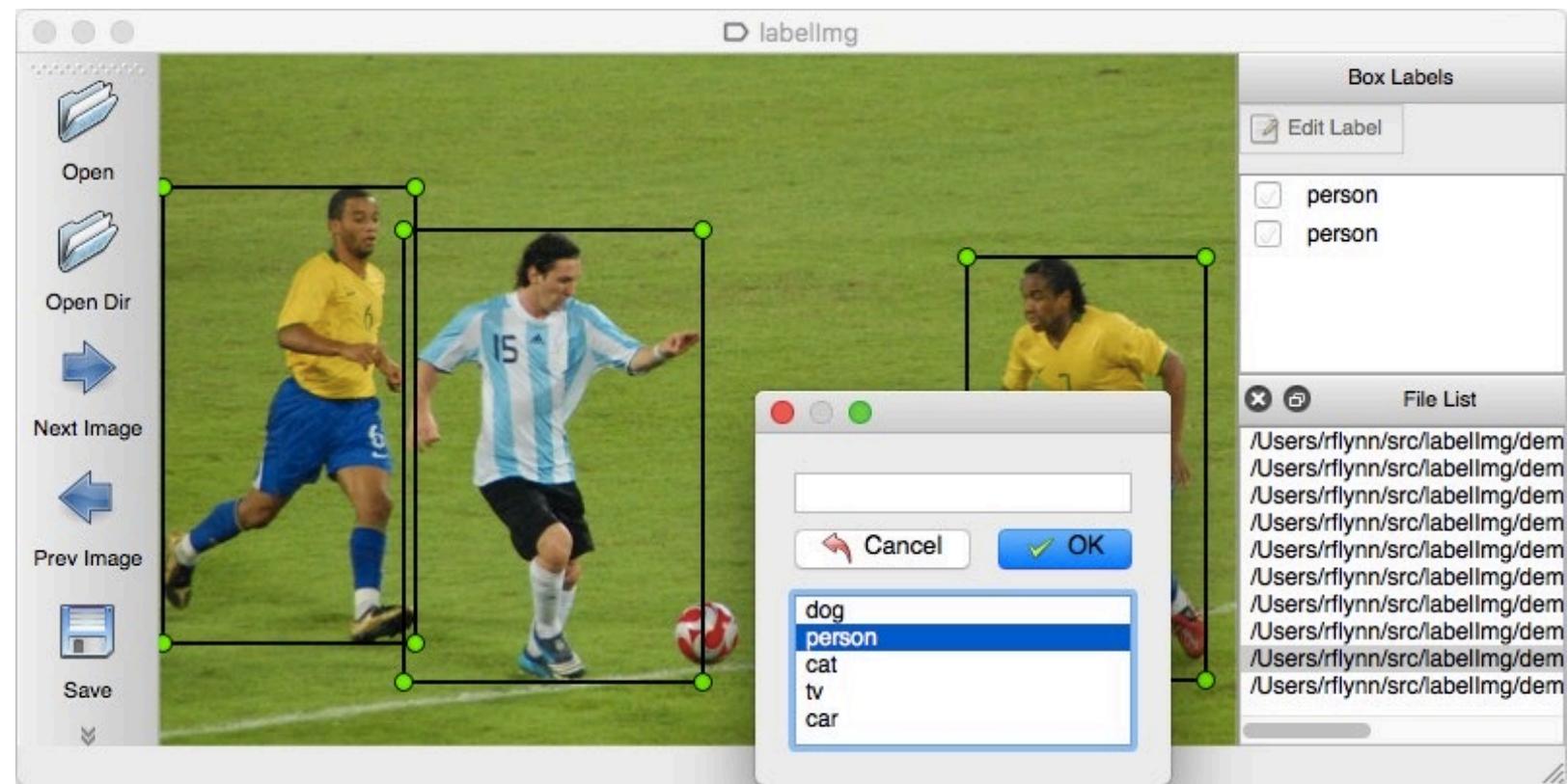
- Introduction
- Code Environment
- YOLOv8 – How to use
- Linux Commands (Colab)
- LabelImg
- Question

labelImg

❖ Introduction



A graphical image annotation tool



❖ 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

heartexlabs / labelImg Public

<> Code Issues 376 Pull requests 48 Actions Projects Wiki Security Insights

master 5 branches 25 tags

Go to file Add file <> Code

Isell Readme updates (#950) ... 1 b33f965 on Sep 23, 2022 449 commits

.github Revert "Create pylint.yml" last year

build-tools Fix typo 4 years ago

data Modified the default label text box into a drop down (#824) 2 years ago

demo Screenshot of macOS High Sierra usage added. 6 years ago

libs CreateML fixes (#906) 10 months ago

readme Readme updates (#950) 8 months ago

requirements Bump lxml from 4.6.5 to 4.9.1 in /requirements (#909) 10 months ago

resources Updated translations last year

tests CreateML fixes (#906) 10 months ago

tools move convert dir to tools dir 2 years ago

.gitignore Update .gitignore last year

CONTRIBUTING.rst Create new files for pypackage 6 years ago

About

LabelImg is now part of the Label Studio community. The popular image annotation tool created by Tzutalin is no longer actively being developed, but you can check out Label Studio, the open source data labeling tool for images, text, hypertext, audio, video and time-series data.

youtu.be/pOnR2YsCY_U

recognition tools deep-learning
detection annotations python3
imagenet image-classification python2

Readme MIT license 19.8k stars 382 watching 5.9k forks Report repository

Go to file Add file <> Code

Local Codespaces New

<- Clone ?

HTTPS SSH GitHub CLI

https://github.com/heartexlabs/labelImg.git

Use Git or checkout with SVN using the web URL.

< Open with GitHub Desktop

< Download ZIP

Step 2: Copy link and do git clone

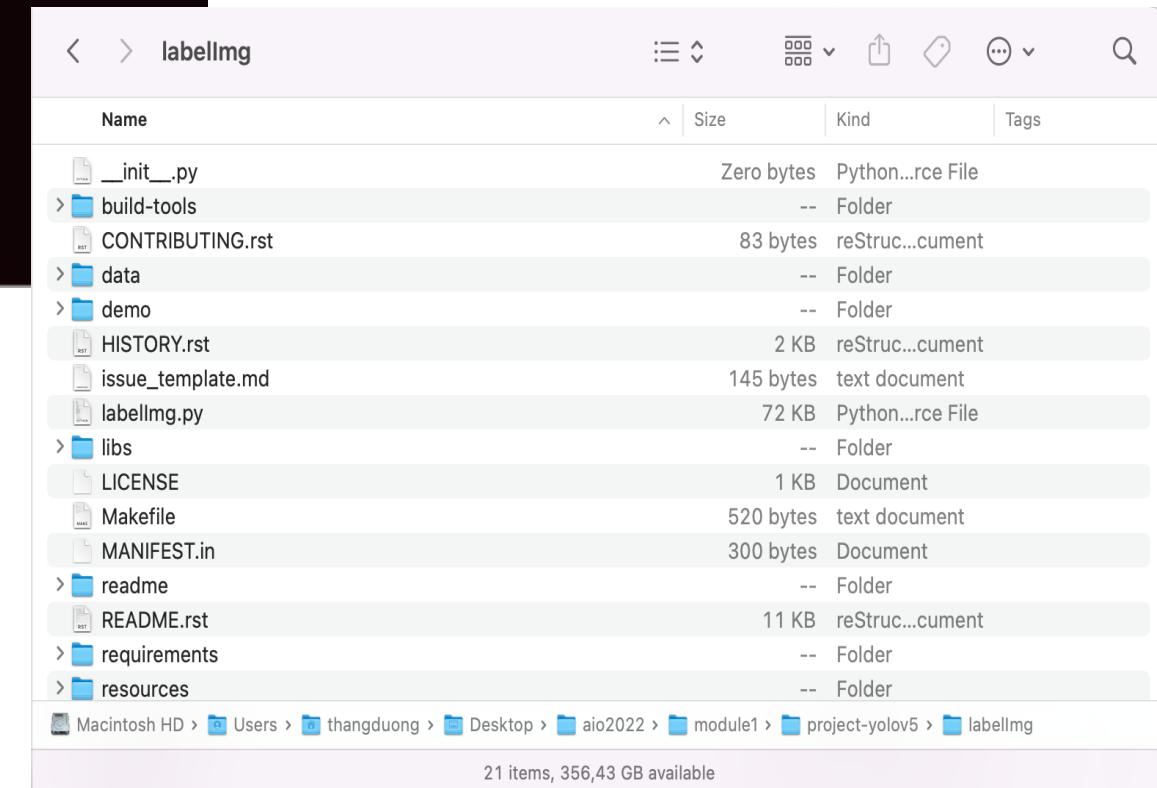
labelImg

❖ labelImg Installation

```
[thangduong@Duongs-MacBook-Pro project-yolov5 % git clone https://github.com/tzutalin/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
LICENSE                data              resources
MANIFEST.in            demo              resources.qrc
Makefile               issue_template.md setup.cfg
README.rst             labelImg.py     setup.py
__init__.py            libs              tests
readme                readme           tools
thangduong@Duongs-MacBook-Pro labelImg %
```



labelImg

❖ labelImg Installation

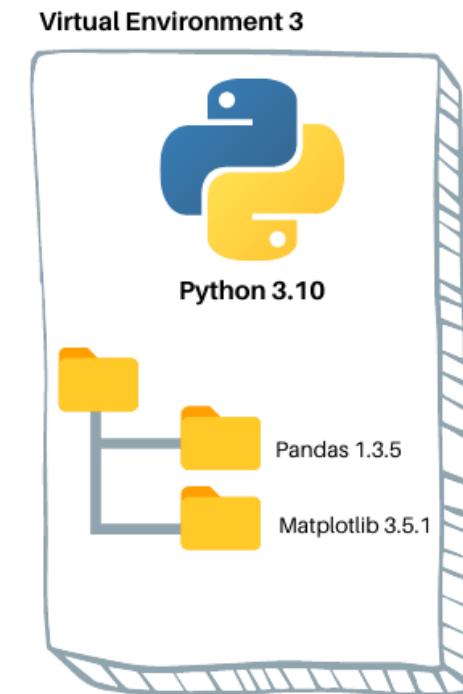
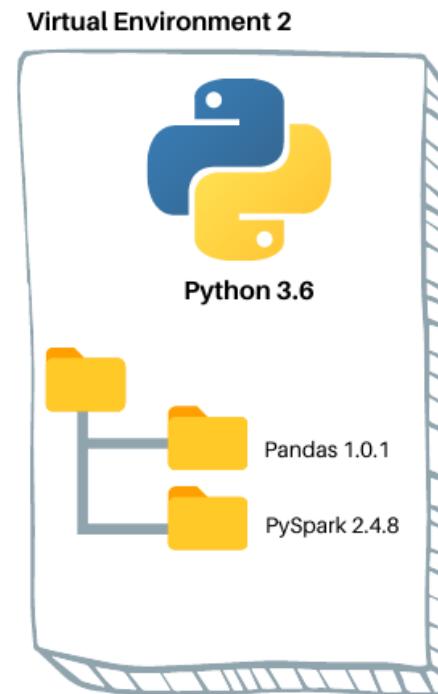
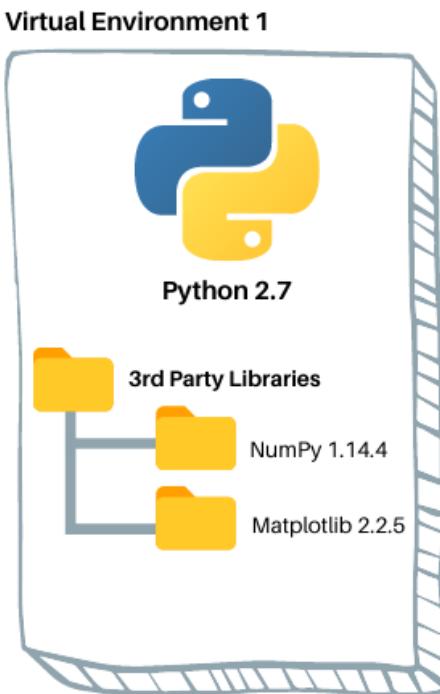
Step 2: Download Anaconda



```
thangduong -- zsh -- 80x24
Last login: Sun Jul 10 16:37:47 on ttys000
(base) thangduong@Duongs-MacBook-Pro ~ %
```

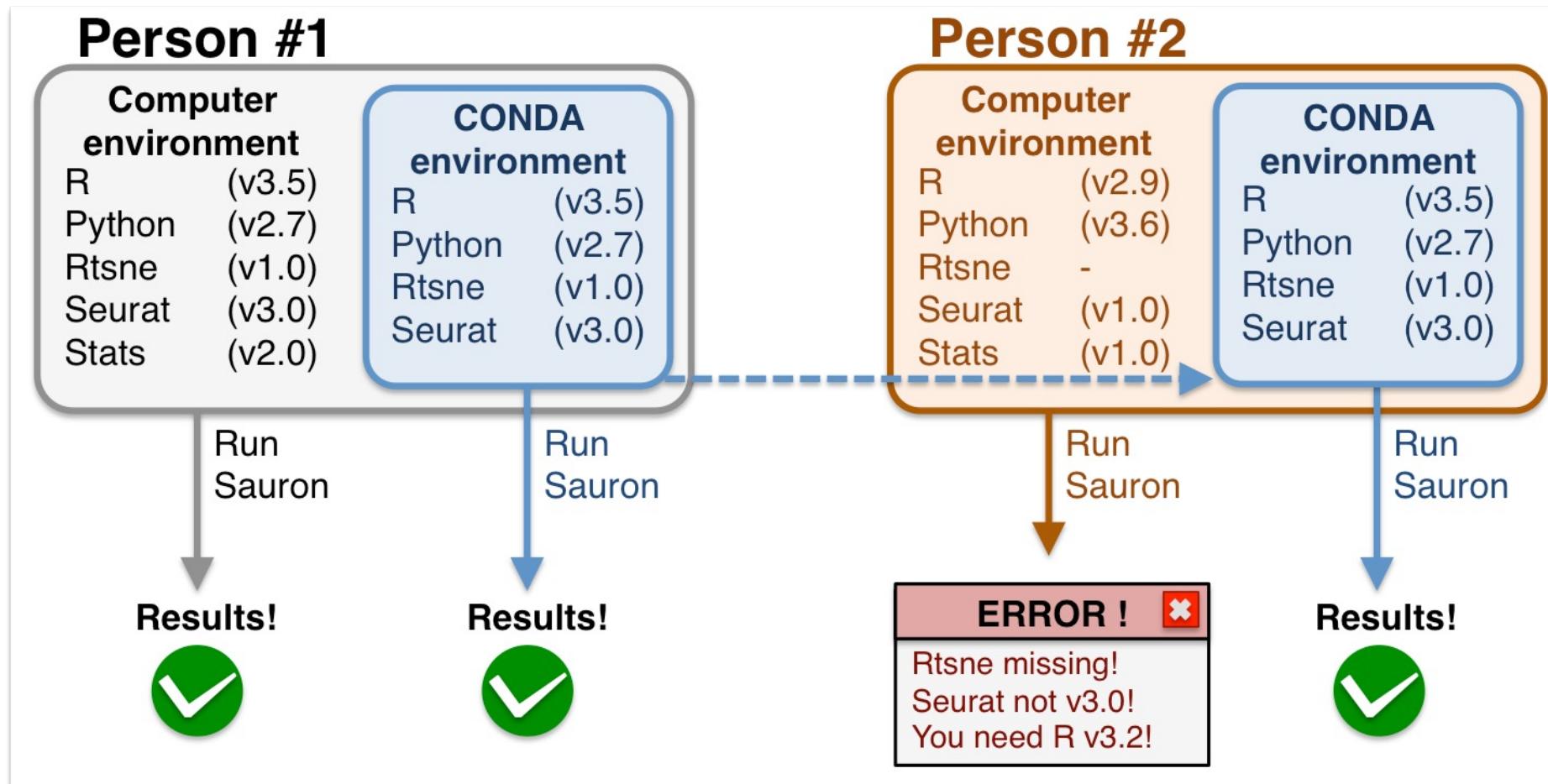
labelImg

❖ About virtual environment



labelImg

❖ About conda



labelImg

❖ About conda: Different requirements

master ➔ yolov5 / requirements.txt

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

42 lines (37 sloc) | 1.2 KB

```
1 # YOLOv5 requirements
2 # Usage: pip install -r requirements.txt
3
4 # Base -----
5 matplotlib>=3.2.2
6 numpy>=1.18.5
7 opencv-python>=4.1.1
8 Pillow>=7.1.2
9 PyYAML>=5.3.1
10 requests>=2.23.0
11 scipy>=1.4.1 # Google Colab version
12 torch>=1.7.0,!>=1.12.0 # https://github.com/ultralytics/yolov5/issues/8395
13 torchvision>=0.8.1,!>=0.13.0 # https://github.com/ultralytics/yolov5/issues/8395
14 tqdm>=4.41.0
15 protobuf<4.21.3 # https://github.com/ultralytics/yolov5/issues/8012
16
```

master ➔ PyTorch_YOLOv4 / requirements.txt

WongKinYiu Update requirements.txt
1 contributor

9 lines (9 sloc) | 115 Bytes

```
1 numpy == 1.17
2 opencv-python >= 4.1
3 torch == 1.6
4 torchvision
5 matplotlib
6 pycocotools
7 tqdm
8 pillow
9 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 ~ %
```

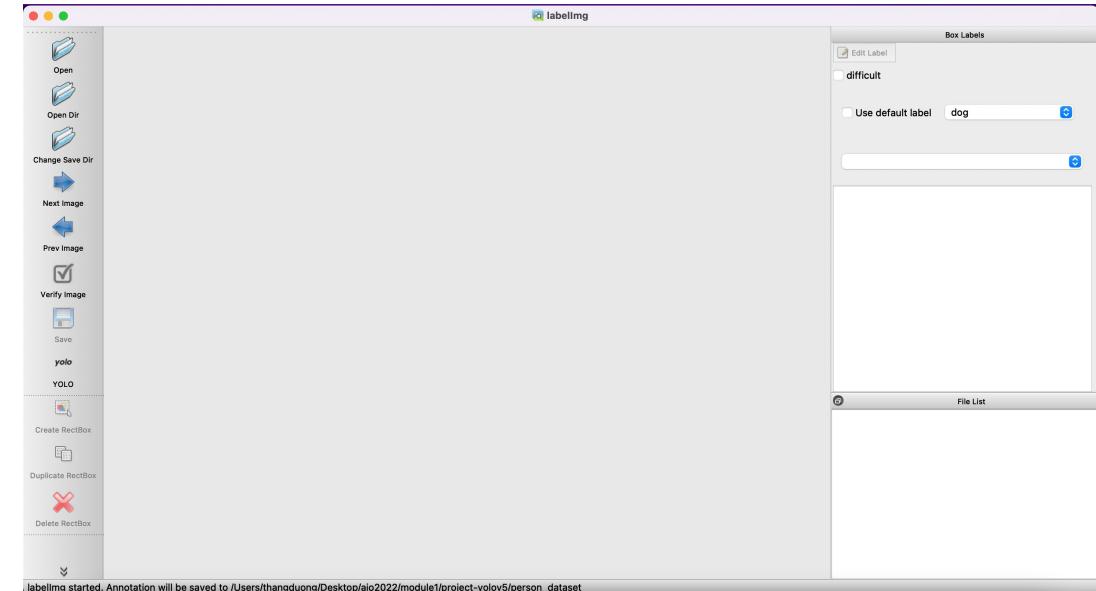
labelImg

❖ 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



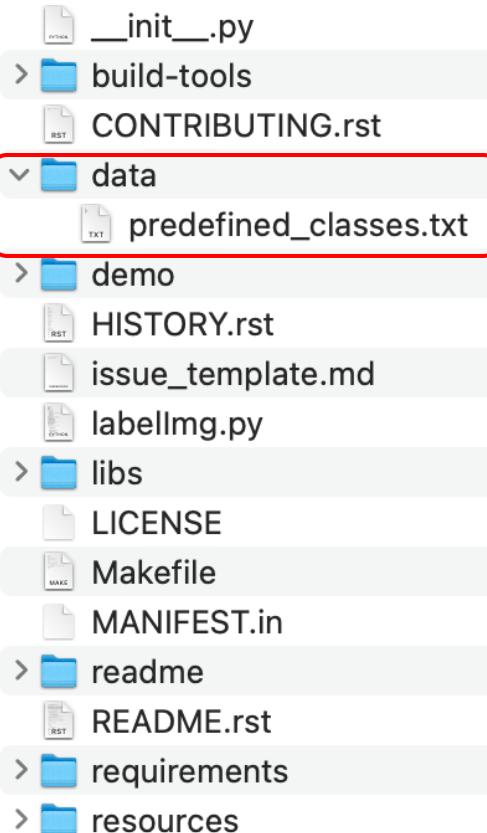
```
(labelimg_env) thangduong@Duongs-MacBook-Pro labelImg % python labelImg.py
/Users/thangduong/Desktop/aio2022/module1/project-yolov5/labelImg/labelImg.py:21
  3: DeprecationWarning: an integer is required (got type DockWidgetFeatures). Implicit conversion to integers using __int__ is deprecated, and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

labelImg

❖ How to use?

< > labelImg

Name



predefined_classes.txt

dog
person
cat
tv
car
meatballs
marinara sauce
tomato soup
chicken noodle soup
french onion soup
chicken breast
ribs
pulled pork
hamburger
cavity

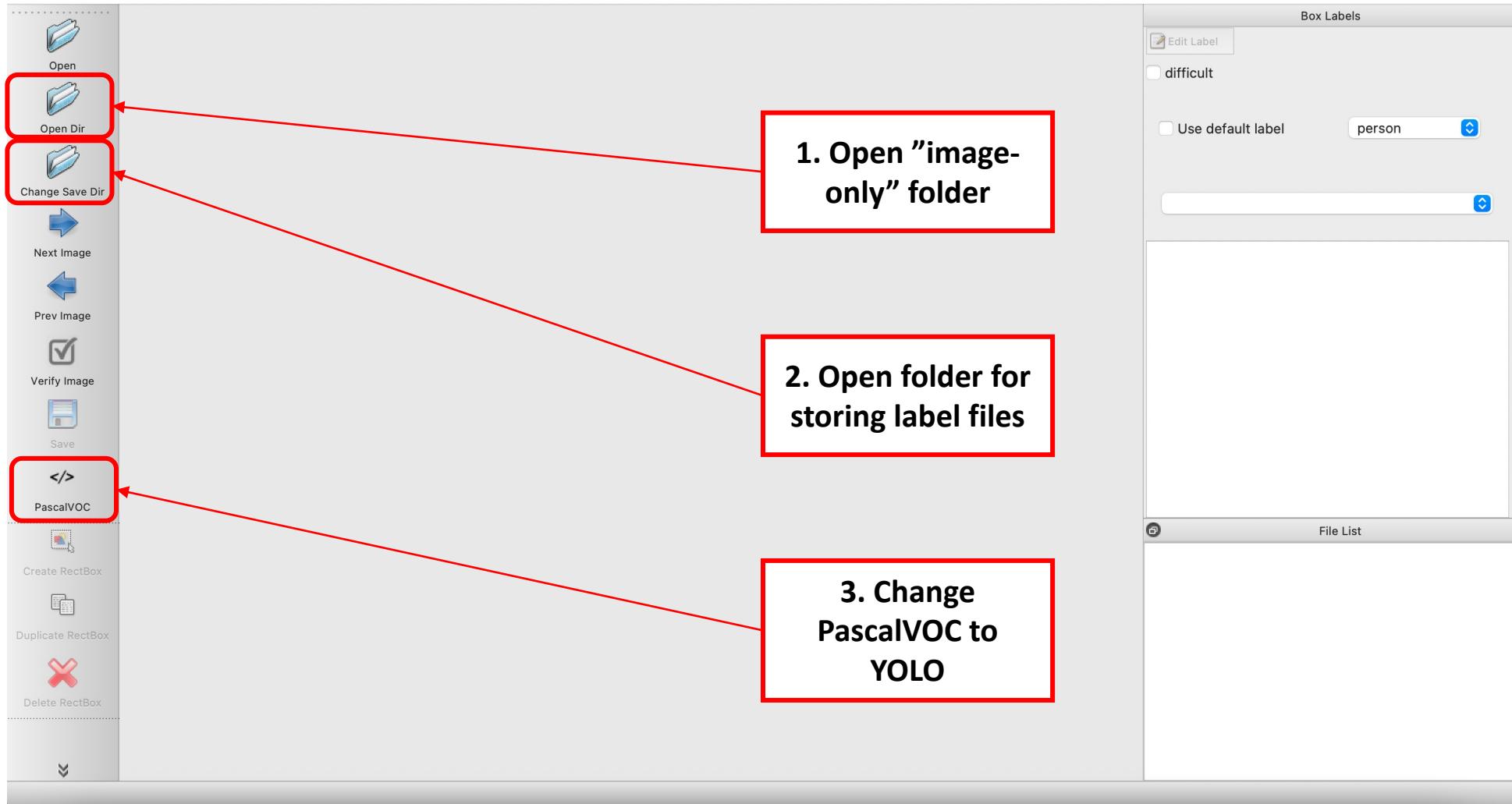
predefined_classes.txt

Human|

Change to classnames we work with

labelImg

❖ How to use?



labelImg

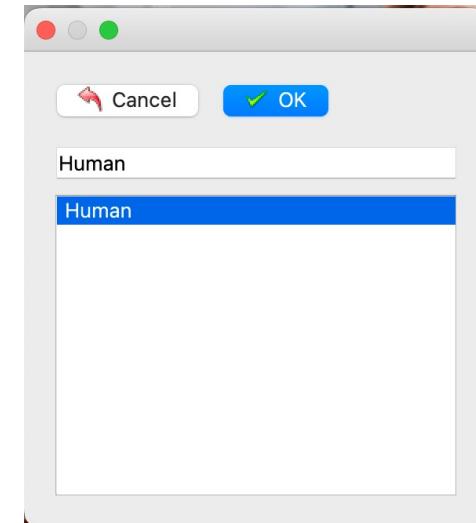
❖ How to use?

1. Click
“RectBox”



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

2. Draw the rectangle so that it fits the
object



Iterate the
process with
other objects



Question

