

Downloading and Setting Up A Dataset From Roboflow for Training YOLOv8

39 Steps

[View most recent version on Tango.ai](#) 

Created by

Hamza Hamami

Creation Date

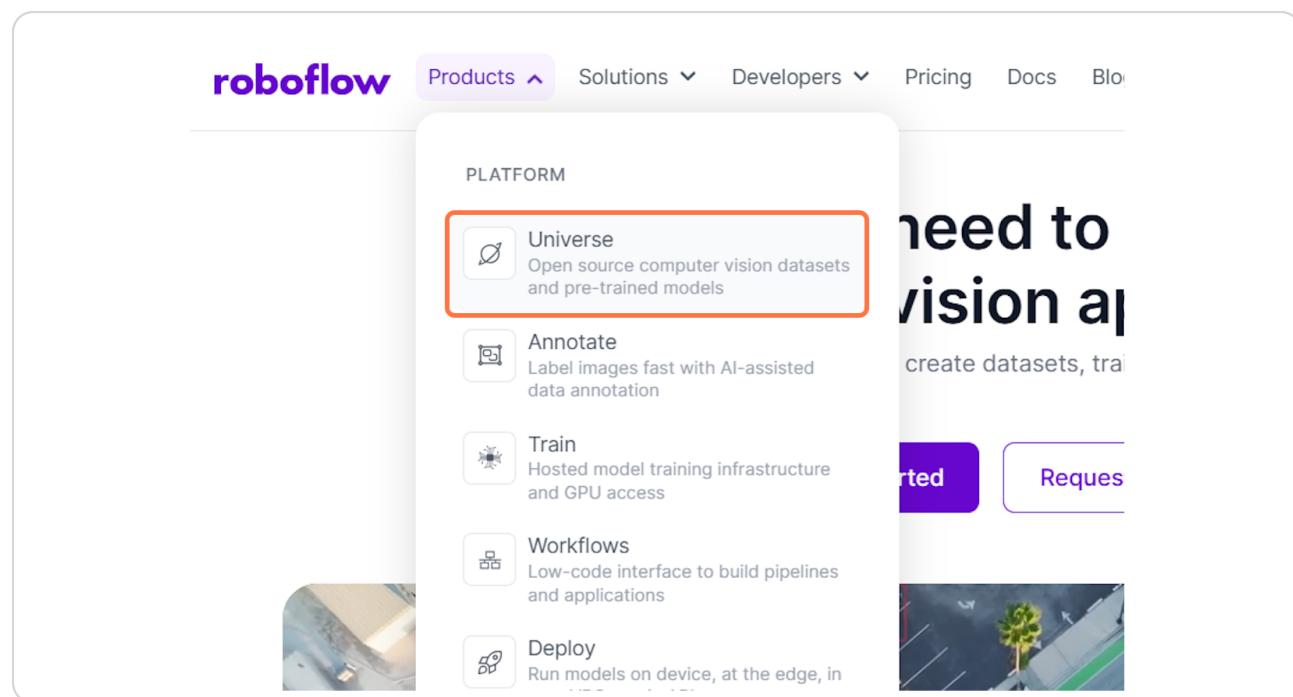
Feb 05, 2025

Last Updated

Feb 05, 2025

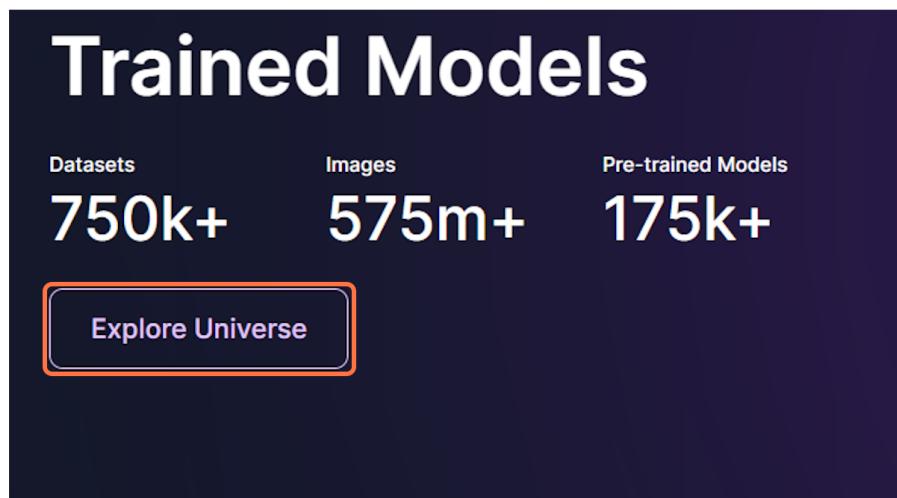
STEP 1

Find 'Universe'



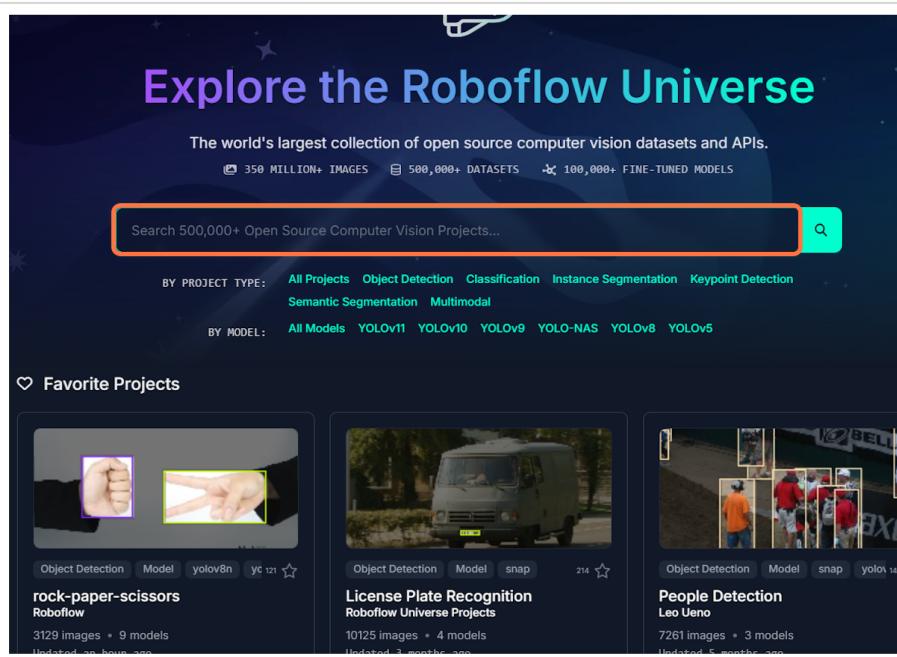
STEP 2

Click on Explore Universe



STEP 3

head to search



STEP 4

Search for desired dataset

The screenshot shows the Roboflow Universe homepage with a search bar at the top containing the text "rocket". Below the search bar, there are filters for "BY PROJECT TYPE" and "BY MODEL". Under "BY PROJECT TYPE", options include All Projects, Object Detection, Classification, Instance Segmentation, Keypoint Detection, Semantic Segmentation, and Multimodal. Under "BY MODEL", options include All Models, YOLOv11, YOLOv10, YOLOv9, YOLO-NAS, YOLOv8, and YOLOv5. The main content area displays "Favorite Projects" with three cards:

- rock-paper-scissors** by Roboflow: 3129 images, 9 models, updated 2 hours ago.
- License Plate Recognition** by Roboflow Universe Projects: 10125 images, 4 models, updated 3 months ago.
- People Detection** by Leo Ueno: 7261 images, 3 models, updated 5 months ago.

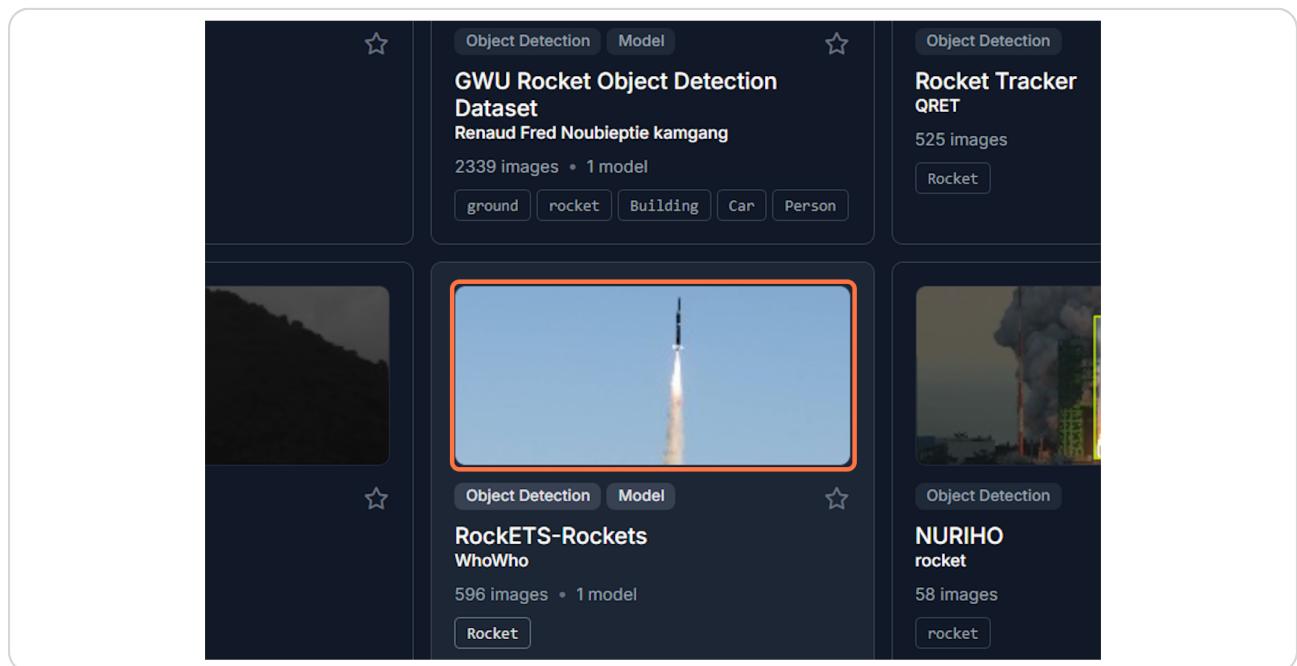
STEP 5

Click on Search

This screenshot shows the same search results as the previous one, but the search button in the search bar has been highlighted with a red rectangle. The search bar contains the text "rocket". The rest of the interface, including the project cards and filters, remains the same.

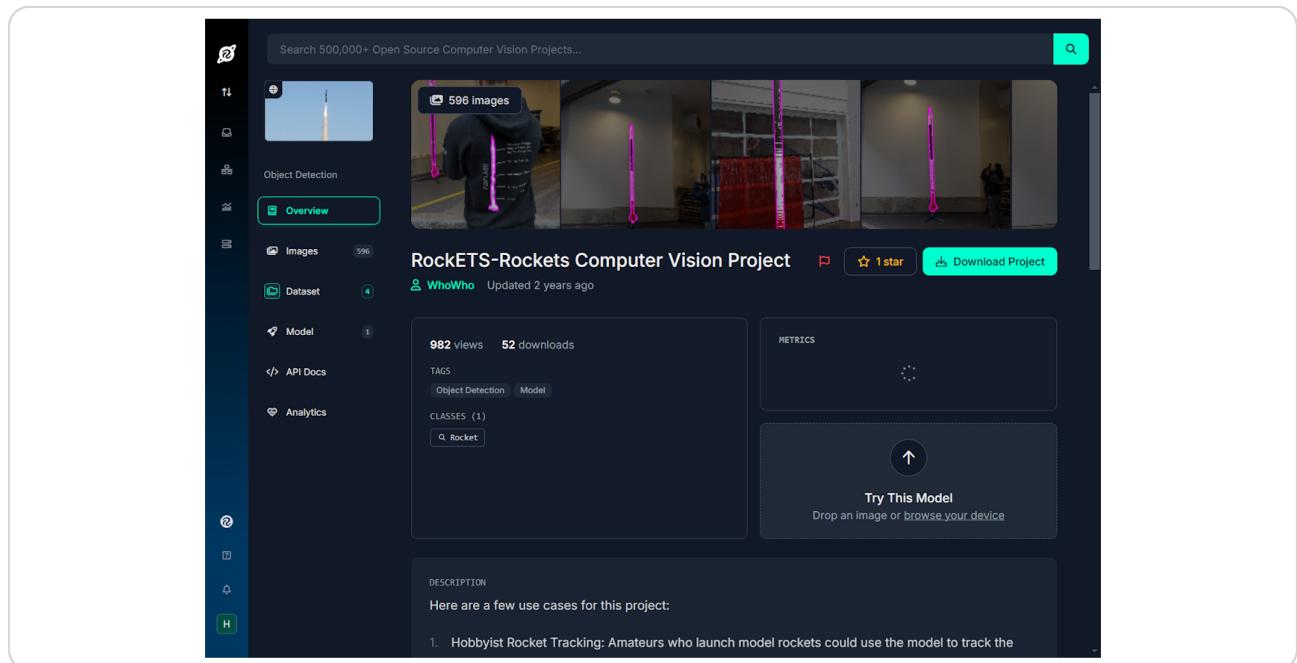
STEP 6

find from many open source projects, the more data the better



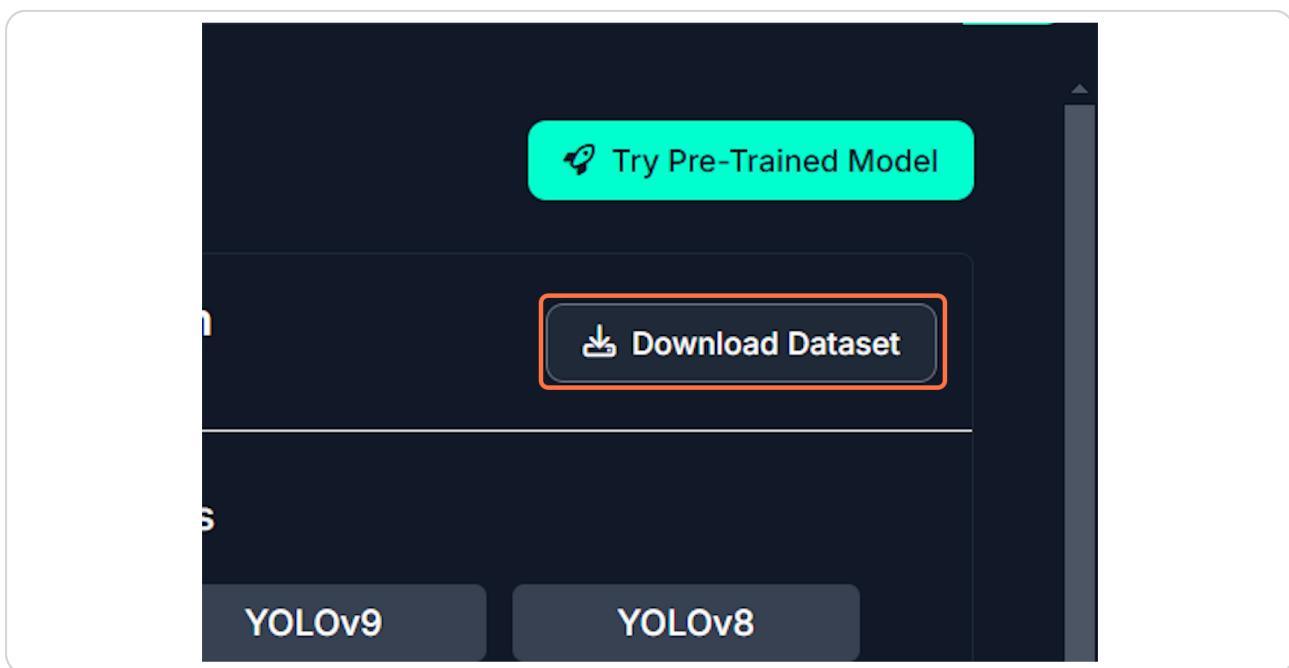
STEP 7

once in the page, click on dataset



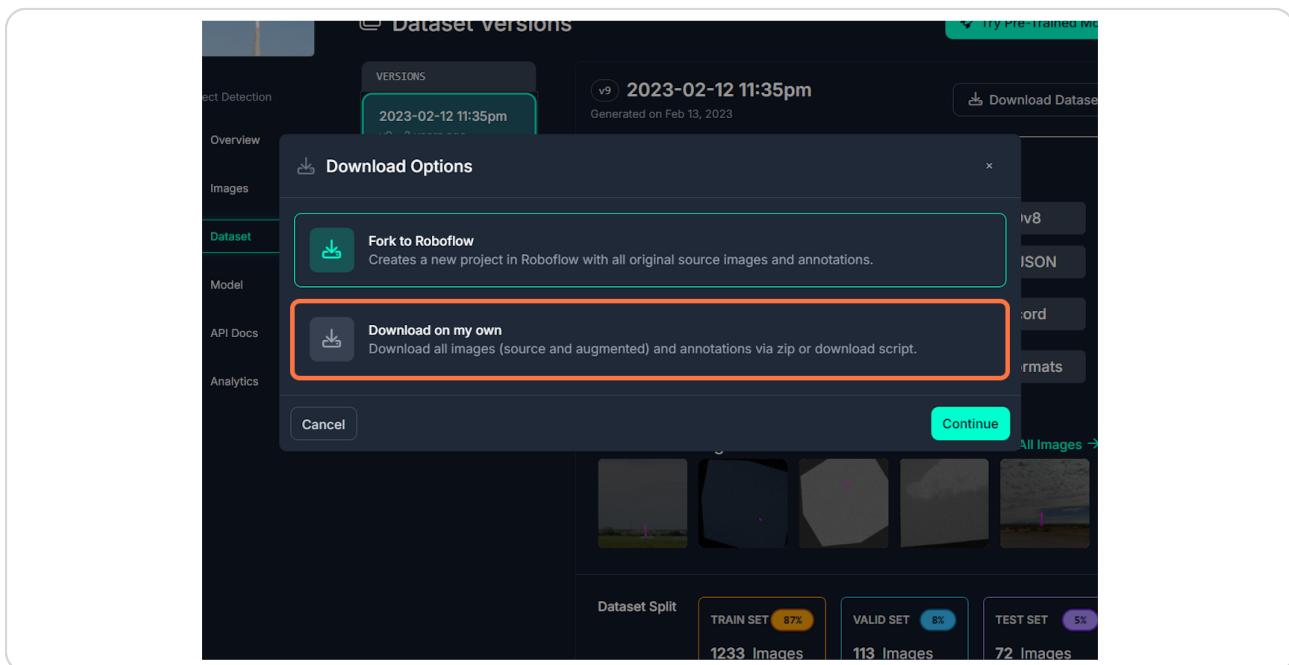
STEP 8

Click on Download Dataset



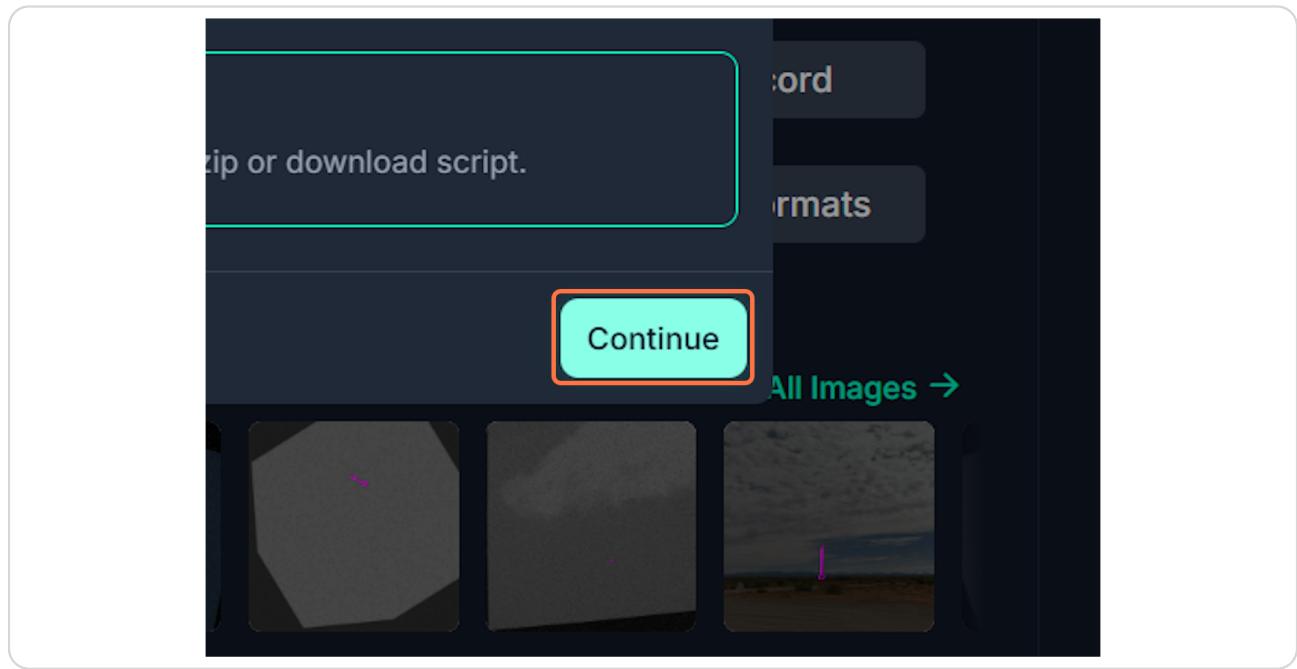
STEP 9

Click on Download om my own



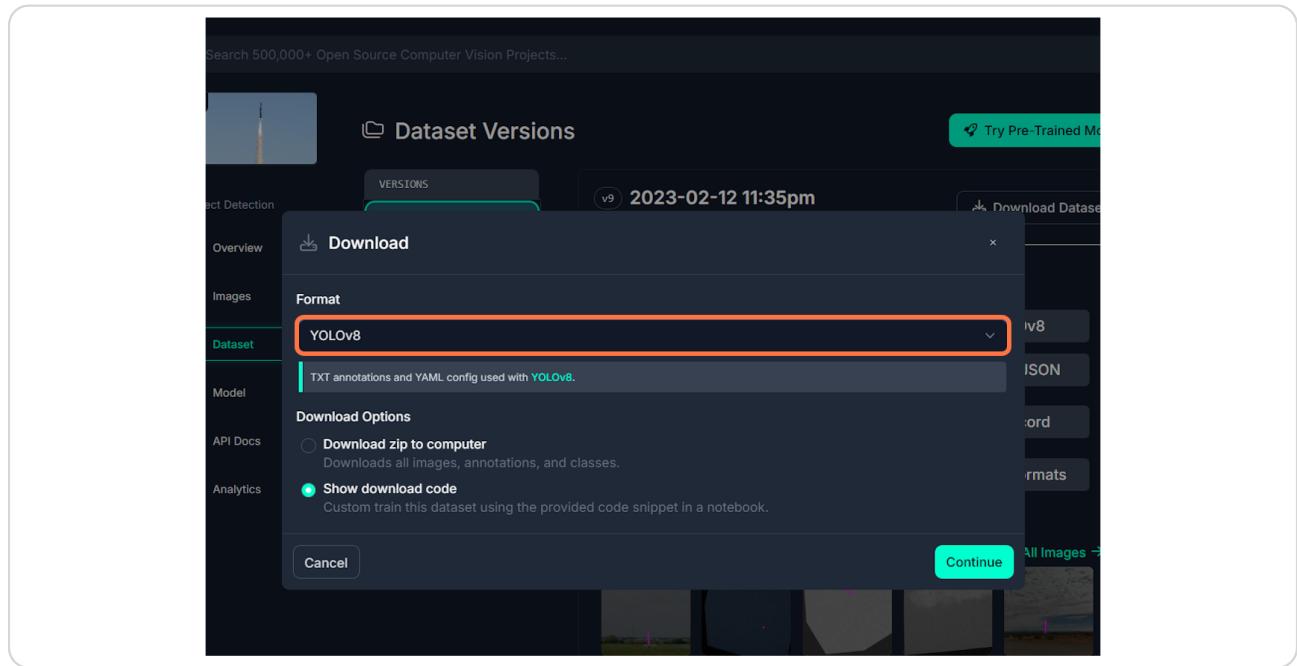
STEP 10

Click on Continue



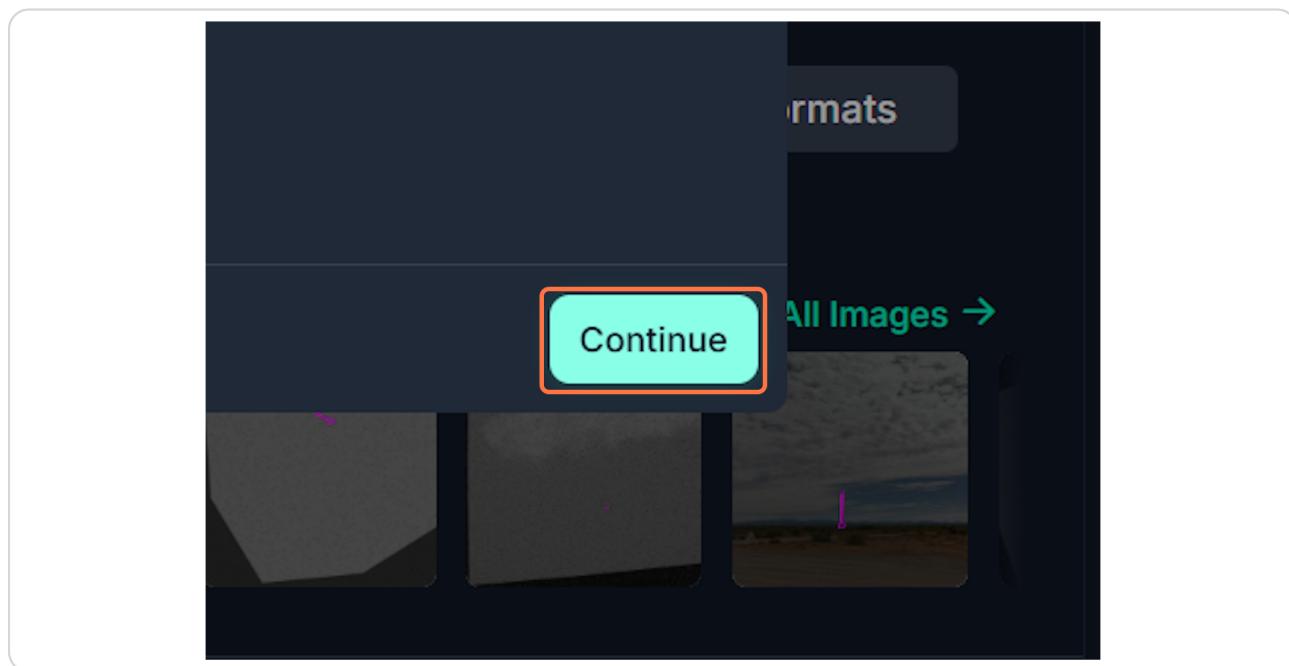
STEP 11

Select YOLOv8 format and choose 'show download code'



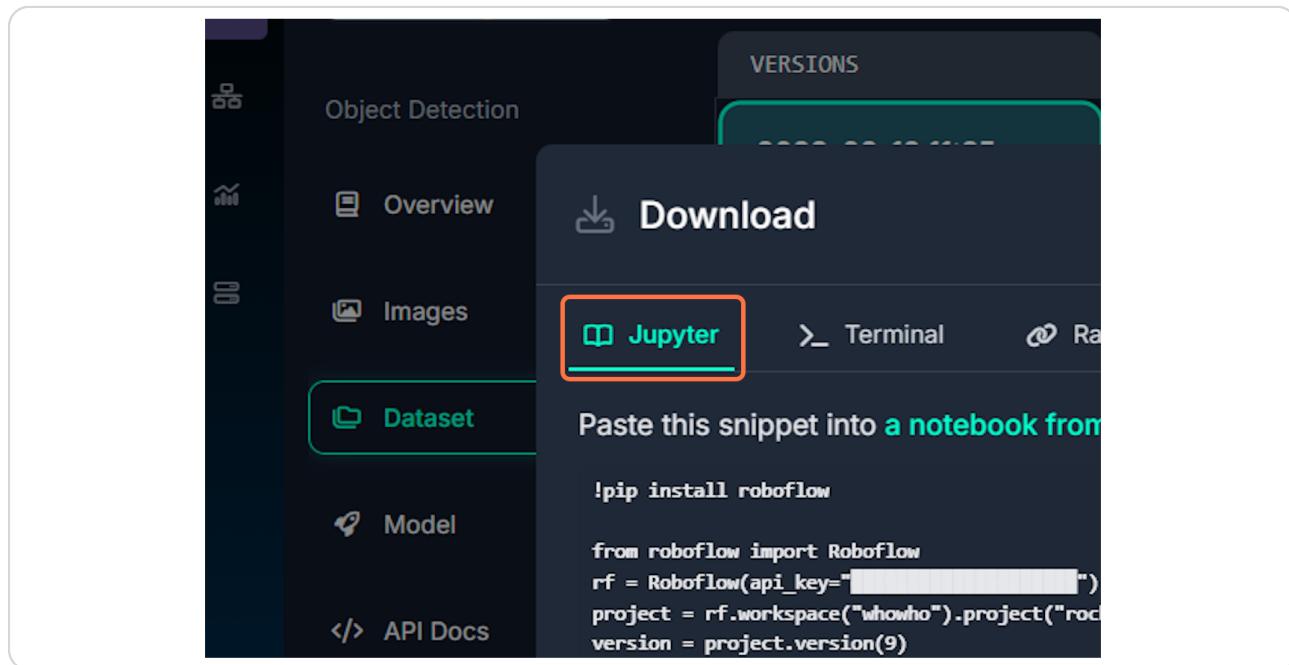
STEP 12

Click on Continue



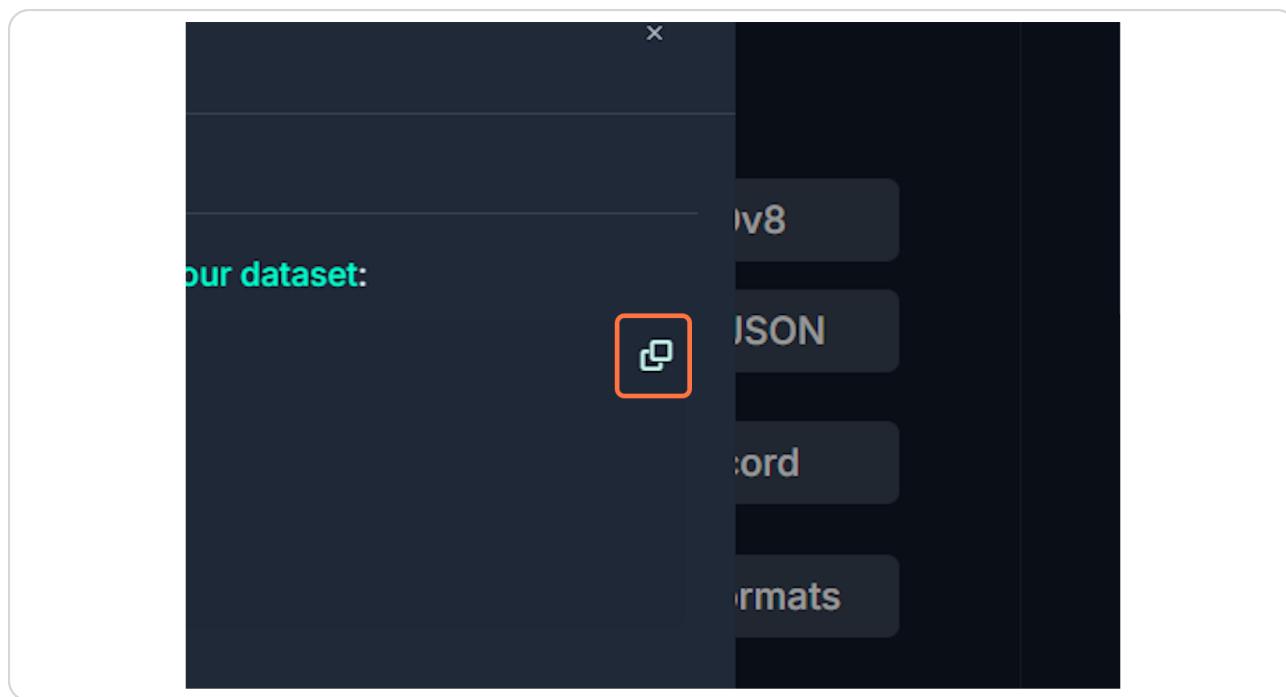
STEP 13

Click on Jupyter



STEP 14

copy

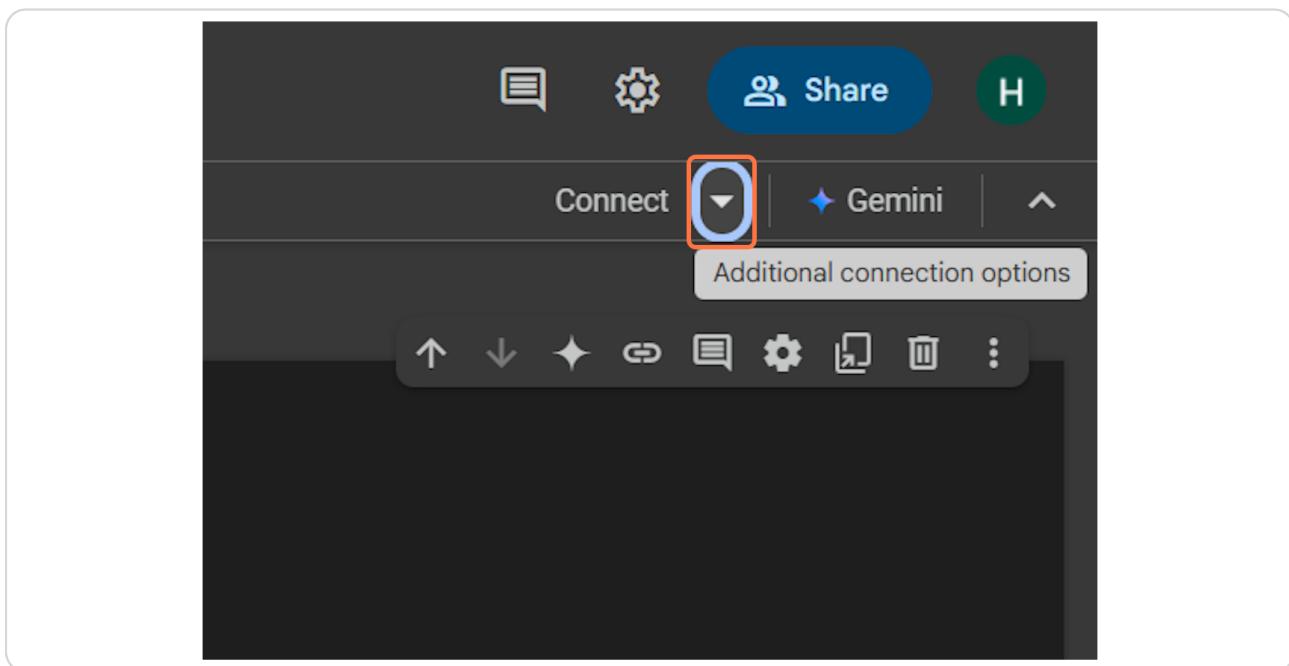


STEP 15

[Next, head to Jupyter or Google Colab; the upcoming step will show how to run on Colab.](#)

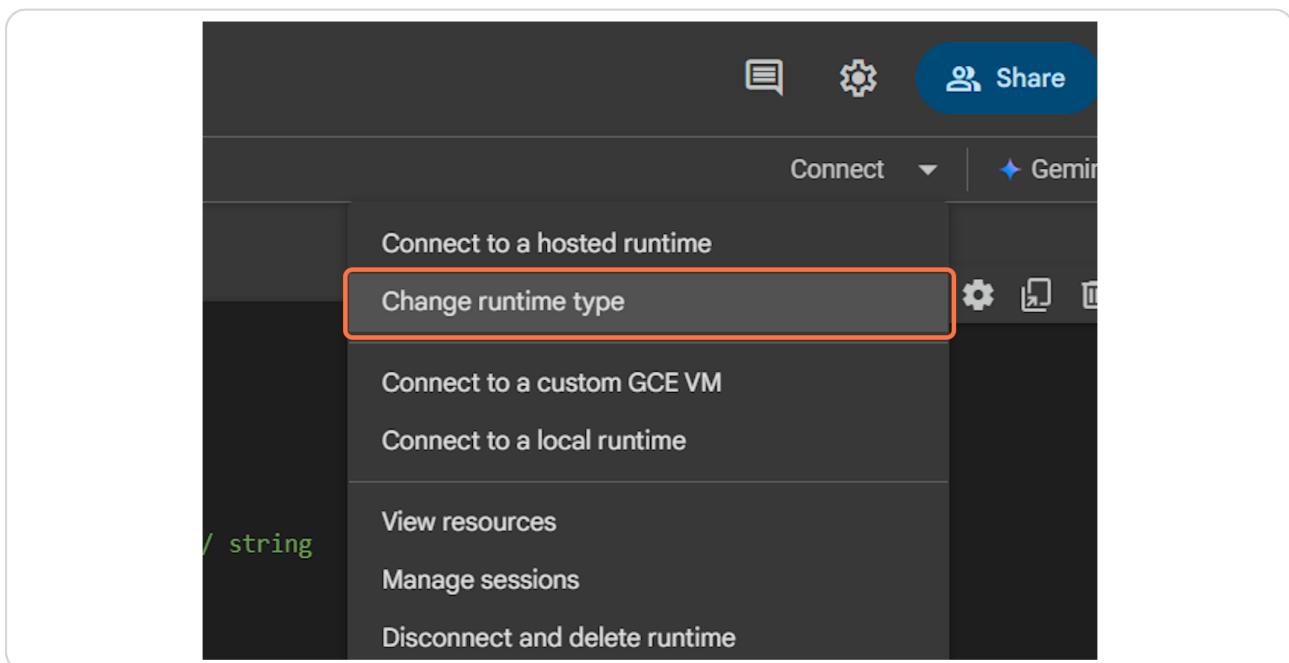
STEP 16

we need switch to GPU so click on arrow next to 'connect'



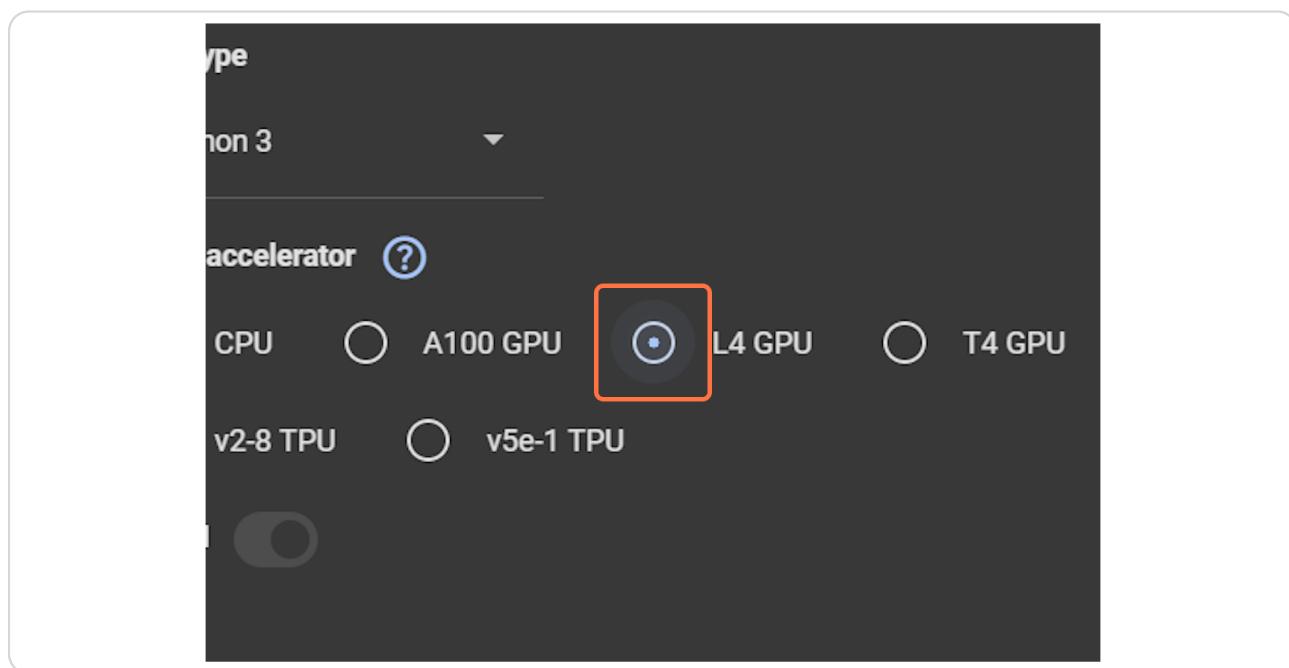
STEP 17

Click on Change runtime type



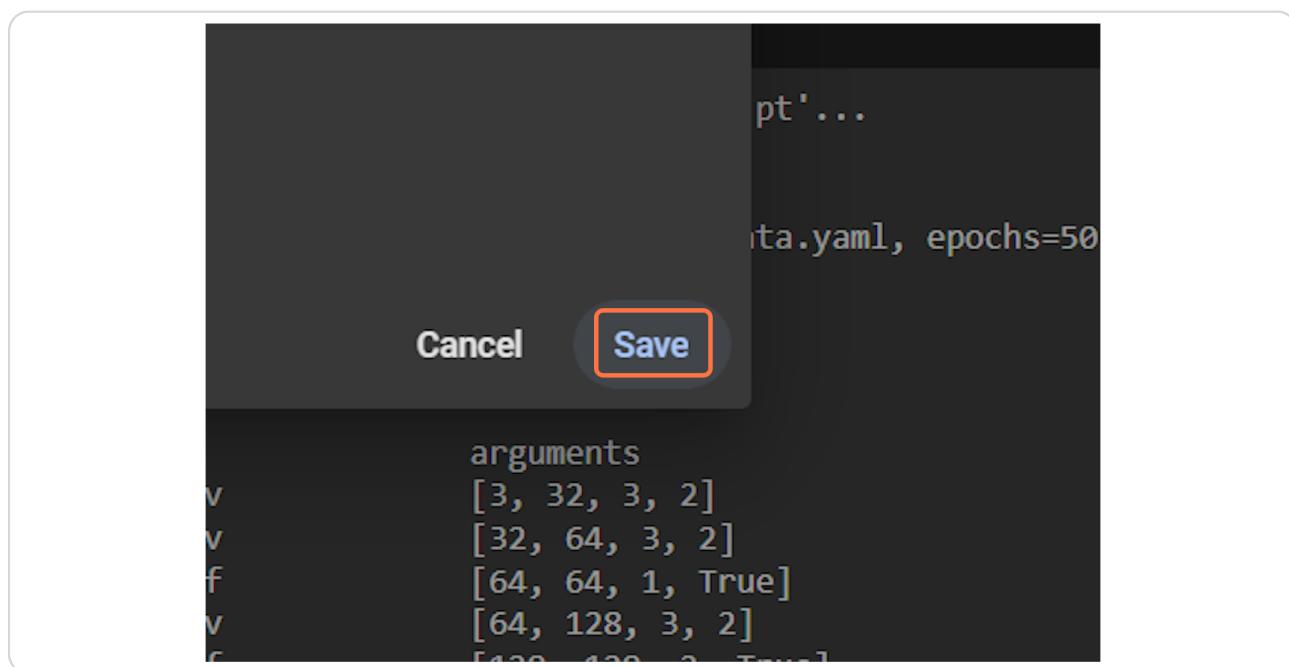
STEP 18

Select GPU, the free one is T4



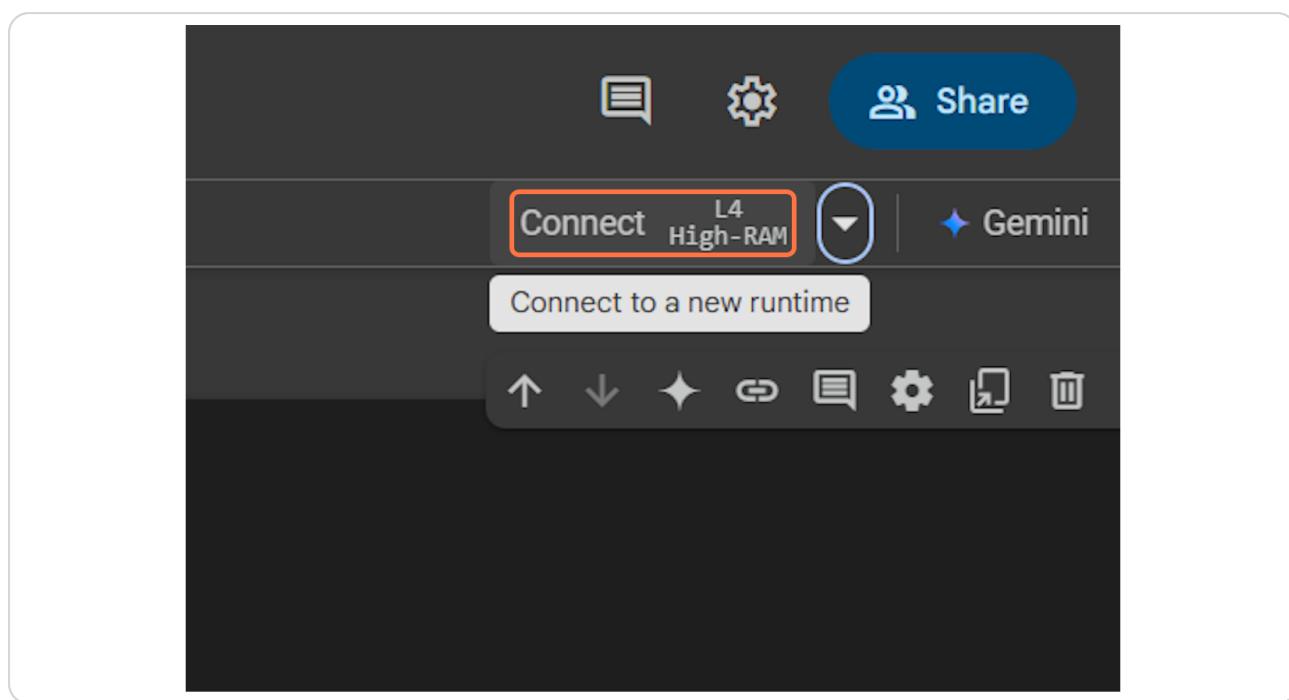
STEP 19

save



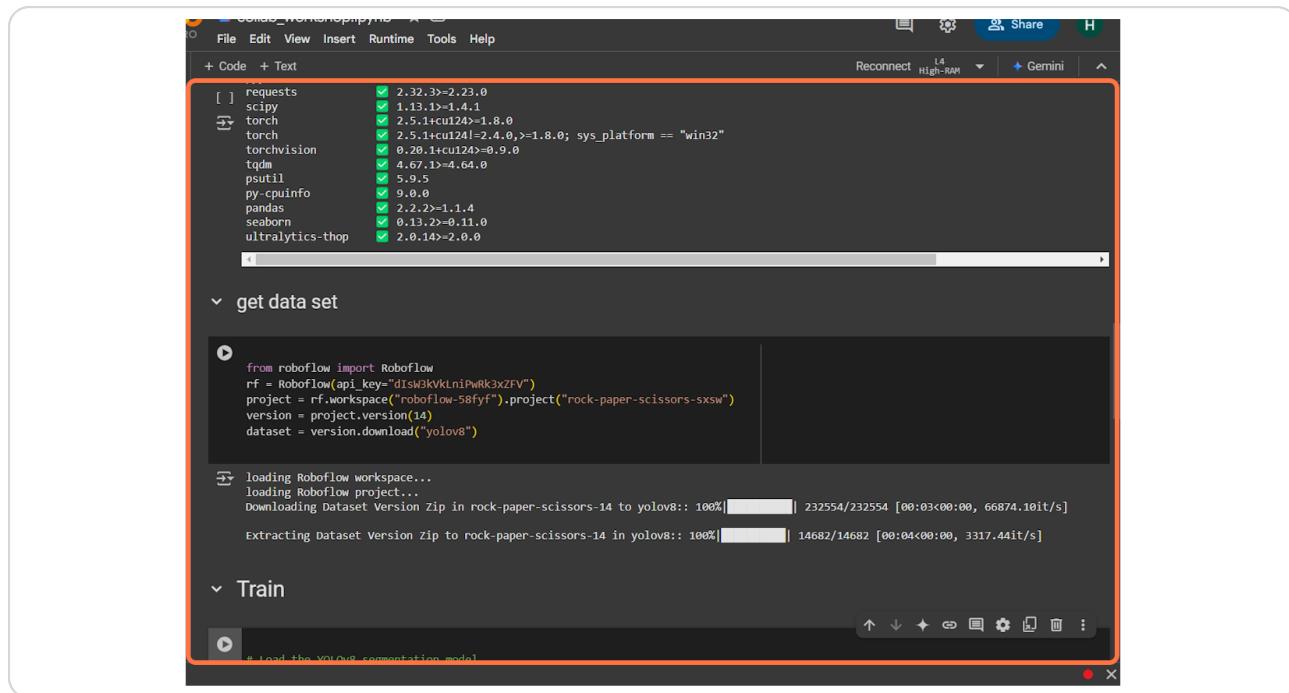
STEP 20

GPU name should change, click to connect to PC or run a code CELL



STEP 21

Before running the code, copy the set of code you received from Roboflow. The first two lines are redundant and can be removed.



The screenshot shows a Jupyter Notebook interface with a code cell containing Python code. The code imports Roboflow and initializes a project. It then attempts to download and extract a dataset version named 'yolov8'. The output shows the progress of the download and extraction process. Below the code cell, there is a section titled 'Train' with a play button icon.

```
[ ] requests      ✓ 2.32.3>=2.23.0
[ ] scipy          ✓ 1.13.1>=1.4.1
[ ] torch          ✓ 2.5.1+cu124>=1.8.0
[ ] torchvision   ✓ 2.5.1+cu124>=2.4.0,>=1.8.0; sys_platform == "win32"
[ ] tqdm           ✓ 0.20.1+cu124>=0.9.0
[ ] psutil         ✓ 5.9.5
[ ] py-cpuinfo    ✓ 9.0.0
[ ] pandas         ✓ 2.2.2>=1.1.4
[ ] seaborn        ✓ 0.13.2>=0.11.0
[ ] ultralytics-thop ✓ 2.0.14>=2.0.0

▼ get data set

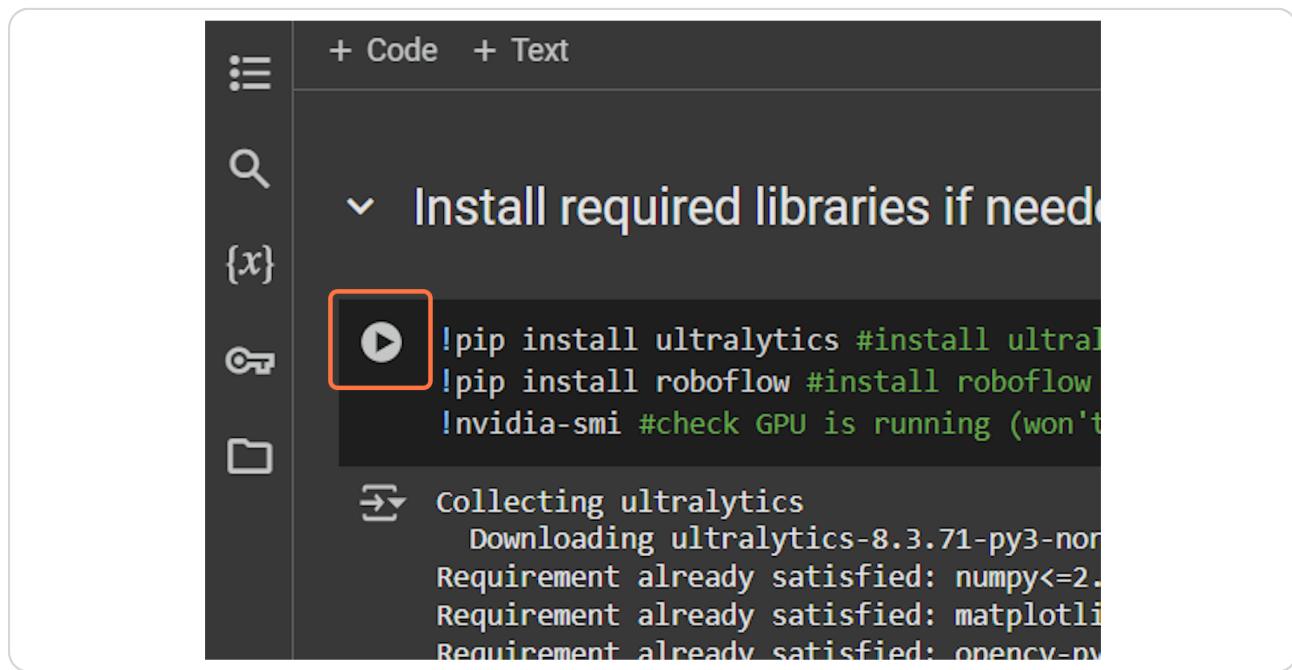
▶ from roboflow import Roboflow
rf = Roboflow(api_key="d1sw0kVklniPwRk3xZEV")
project = rf.workspace("roboflow-5afyf").project("rock-paper-scissors-sxsw")
version = project.version(14)
dataset = version.download("yolov8")

▶ loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Zip in rock-paper-scissors-14 to yolov8:: 100%|██████████| 232554/232554 [00:03<00:00, 66874.10it/s]
Extracting Dataset Zip to rock-paper-scissors-14 in yolov8:: 100%|██████████| 14682/14682 [00:04<00:00, 3317.44it/s]

▼ Train
```

STEP 22

run cell 1 to install libraries



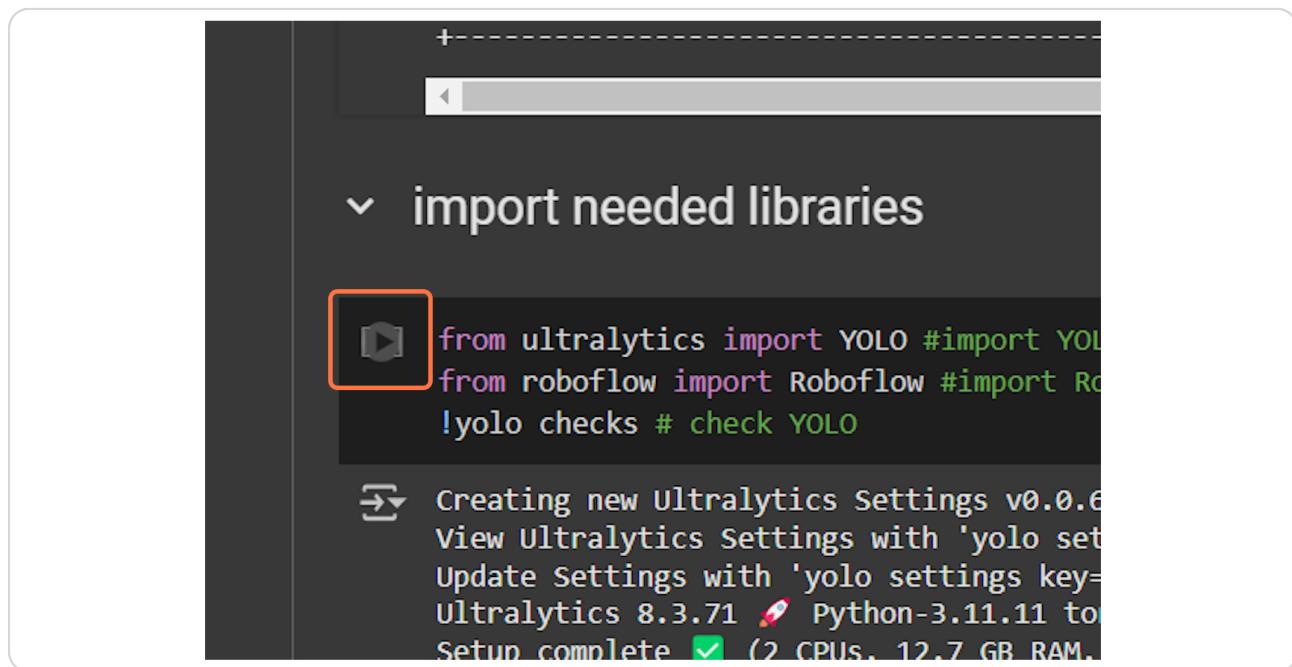
The screenshot shows a Jupyter Notebook interface. On the left is a sidebar with icons for file operations. The main area has two tabs at the top: '+ Code' and '+ Text'. Below the tabs, there's a section titled 'Install required libraries if needed' with a downward arrow. A play button icon is highlighted with an orange box. To its right is a code cell containing three pip commands:

```
!pip install ultralytics #install ultralytics  
!pip install roboflow #install roboflow  
!nvidia-smi #check GPU is running (won't work on colab)
```

Below the code cell, a progress bar indicates the status of the 'Collecting ultralytics' step. The progress bar is mostly filled, with the text 'Requirement already satisfied' repeated several times.

STEP 23

run cell 2 to check YOLO and import



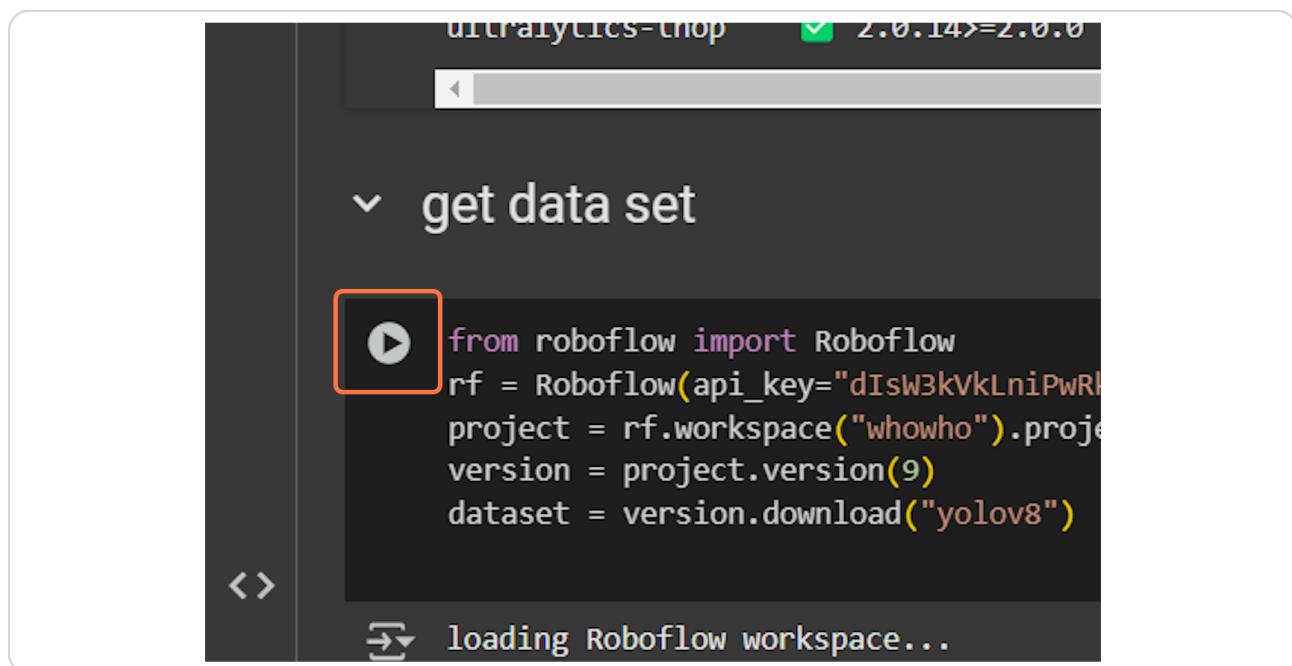
The screenshot shows a Jupyter Notebook interface. On the left is a sidebar with icons for file operations. The main area has two tabs at the top: '+ Code' and '+ Text'. Below the tabs, there's a section titled 'import needed libraries' with a downward arrow. A play button icon is highlighted with an orange box. To its right is a code cell containing three import statements and one command:

```
from ultralytics import YOLO #import YOLO  
from roboflow import Roboflow #import Roboflow  
!yolo checks # check YOLO
```

Below the code cell, a progress bar indicates the status of the 'Creating new Ultralytics Settings v0.0.6' step. The progress bar is mostly filled, with the text 'Setup complete' followed by a checkmark and '(2 CPUS, 12.7 GB RAM)'.

STEP 24

run cell 3 to download your dataset from roboflow



```
ultralytics=ultralytics
2.0.14>=2.0.0

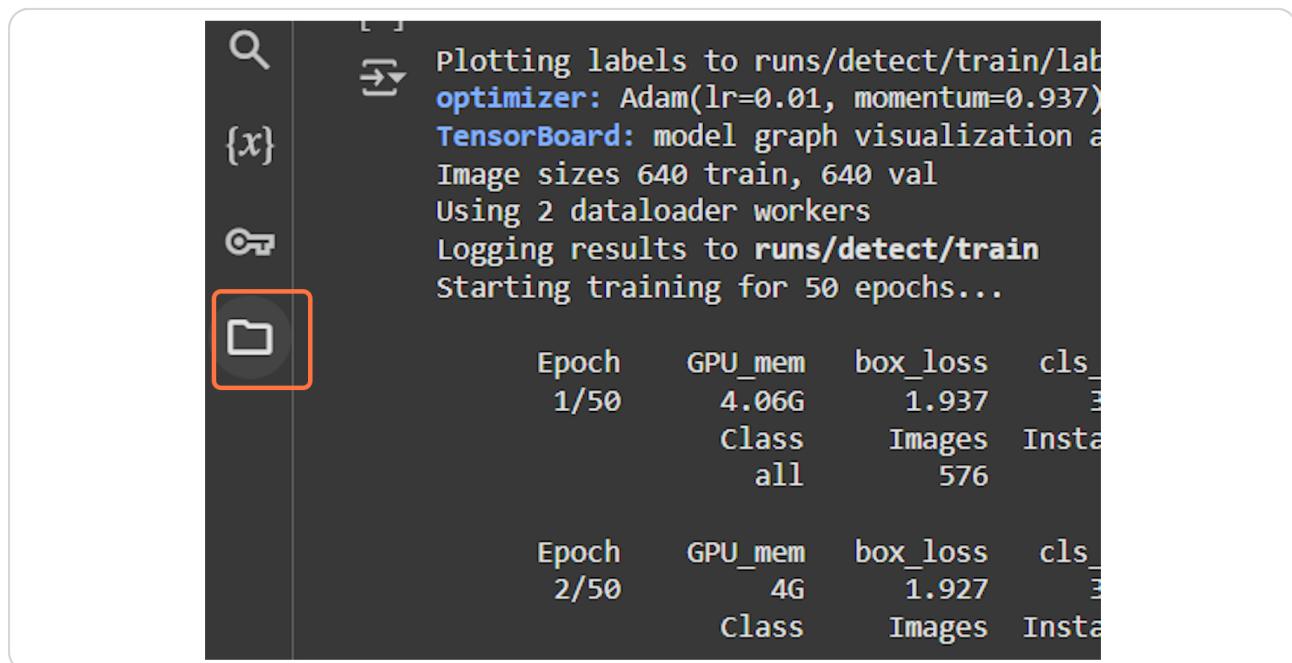
get data set

from roboflow import Roboflow
rf = Roboflow(api_key="dIsW3kVkJniPwRl")
project = rf.workspace("whowho").project()
version = project.version(9)
dataset = version.download("yolov8")

loading Roboflow workspace...
```

STEP 25

go to files

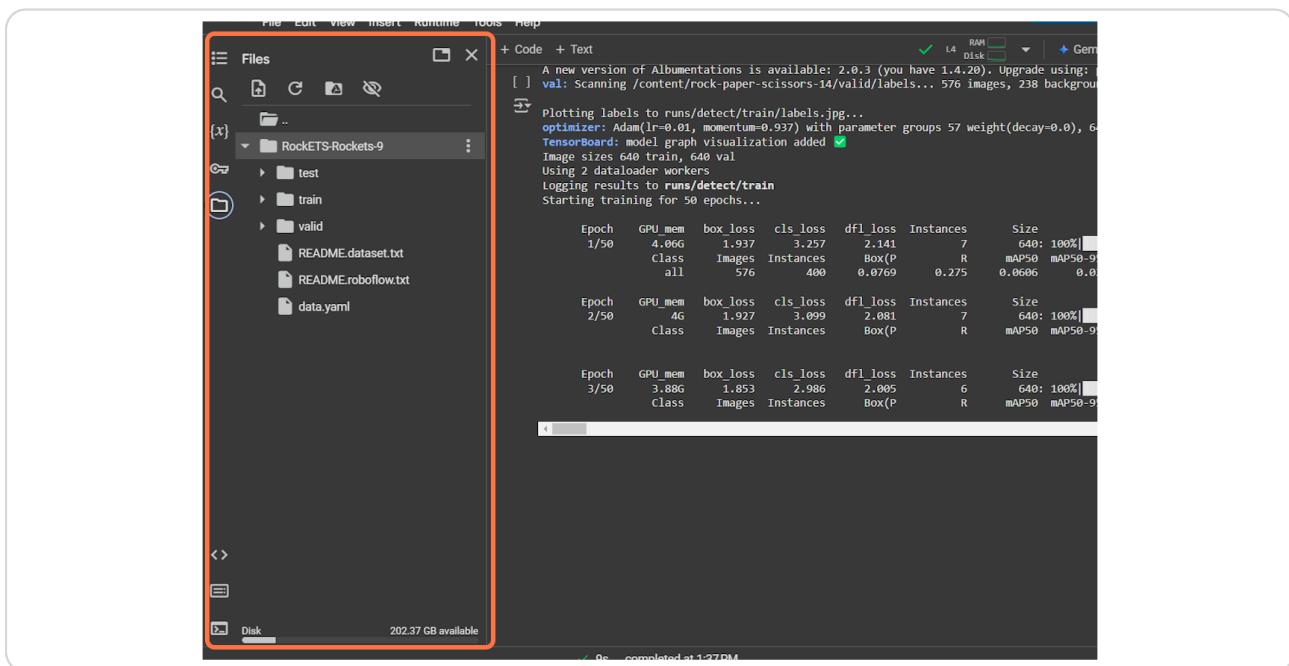


```
Plotting labels to runs/detect/train/labels
optimizer: Adam(lr=0.01, momentum=0.937)
TensorBoard: model graph visualization at http://localhost:6006
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/detect/train
Starting training for 50 epochs...

Epoch      GPU_mem    box_loss   cls_
1/50       4.06G      1.937    3.0
                           Class    Images  Instances
                           all        576
Epoch      GPU_mem    box_loss   cls_
2/50       4G          1.927    3.0
                           Class    Images  Instances
```

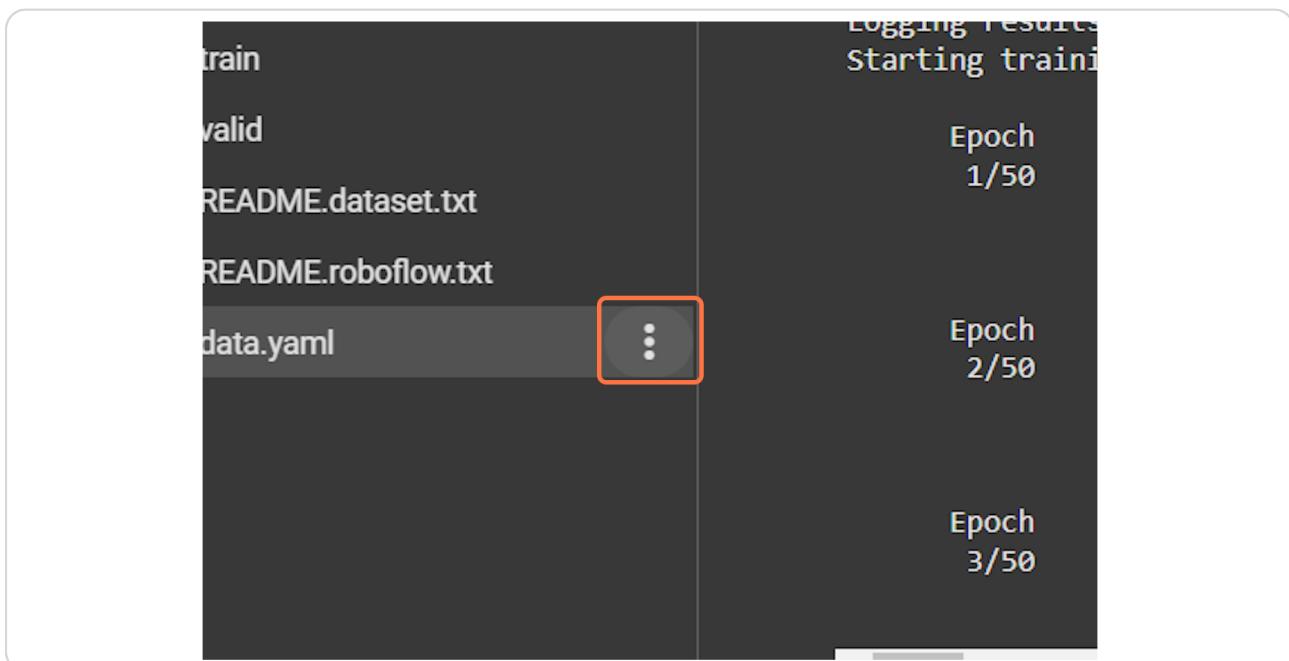
STEP 26

fin the dataset



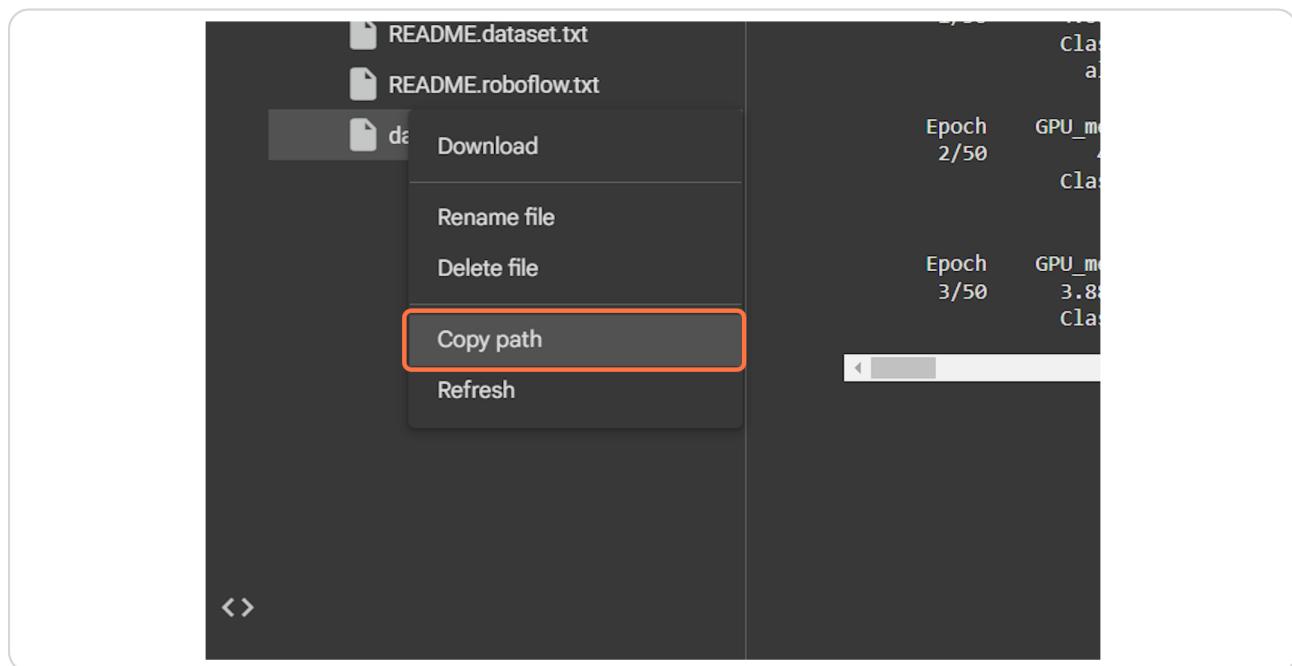
STEP 27

click on three dots next to data.yaml



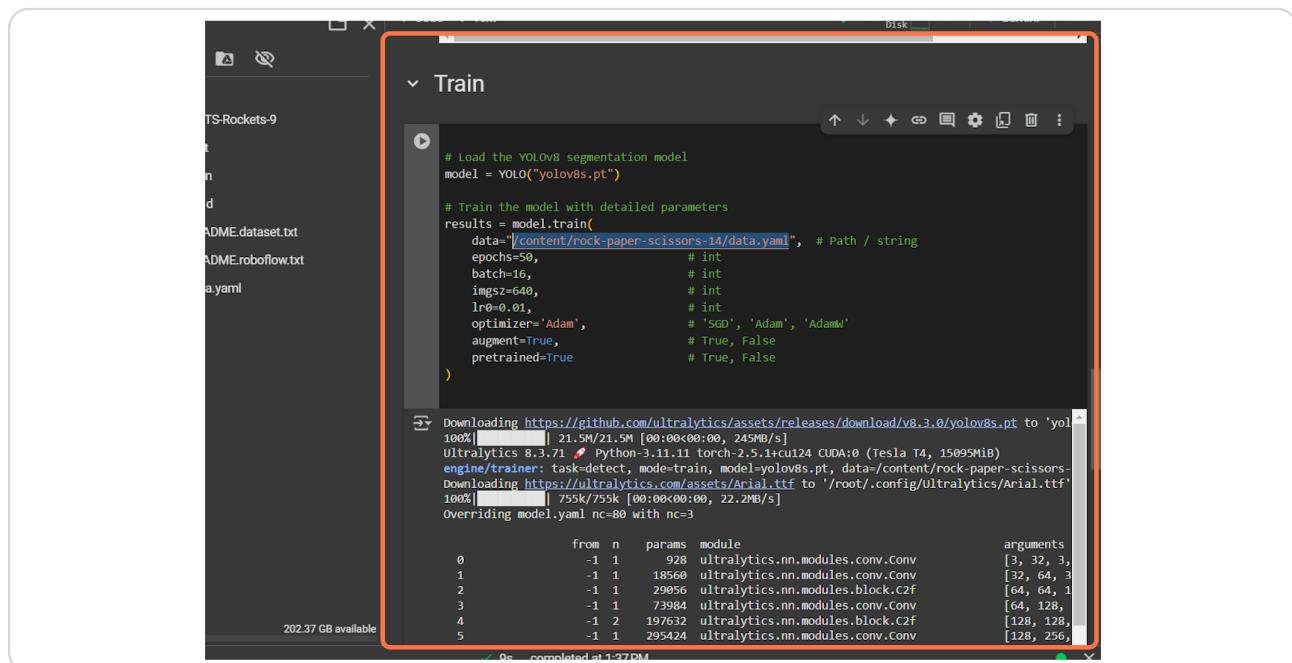
STEP 28

Click on Copy path



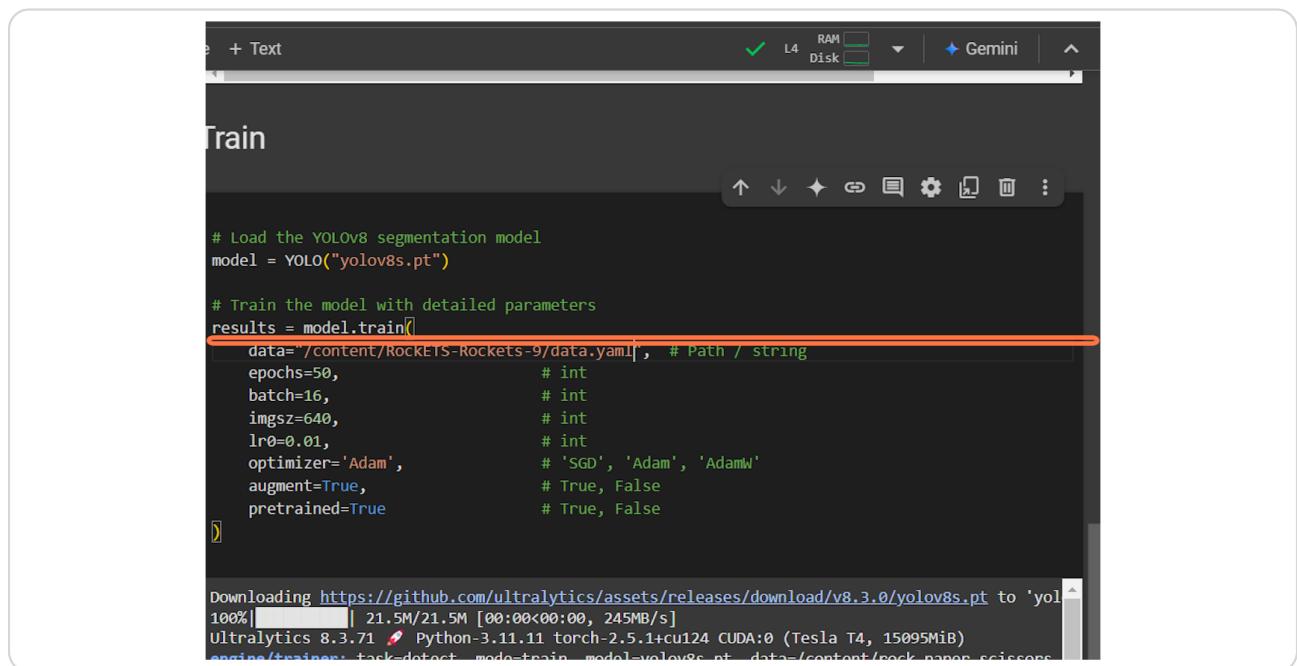
STEP 29

go to training cell



STEP 30

change 'data' path



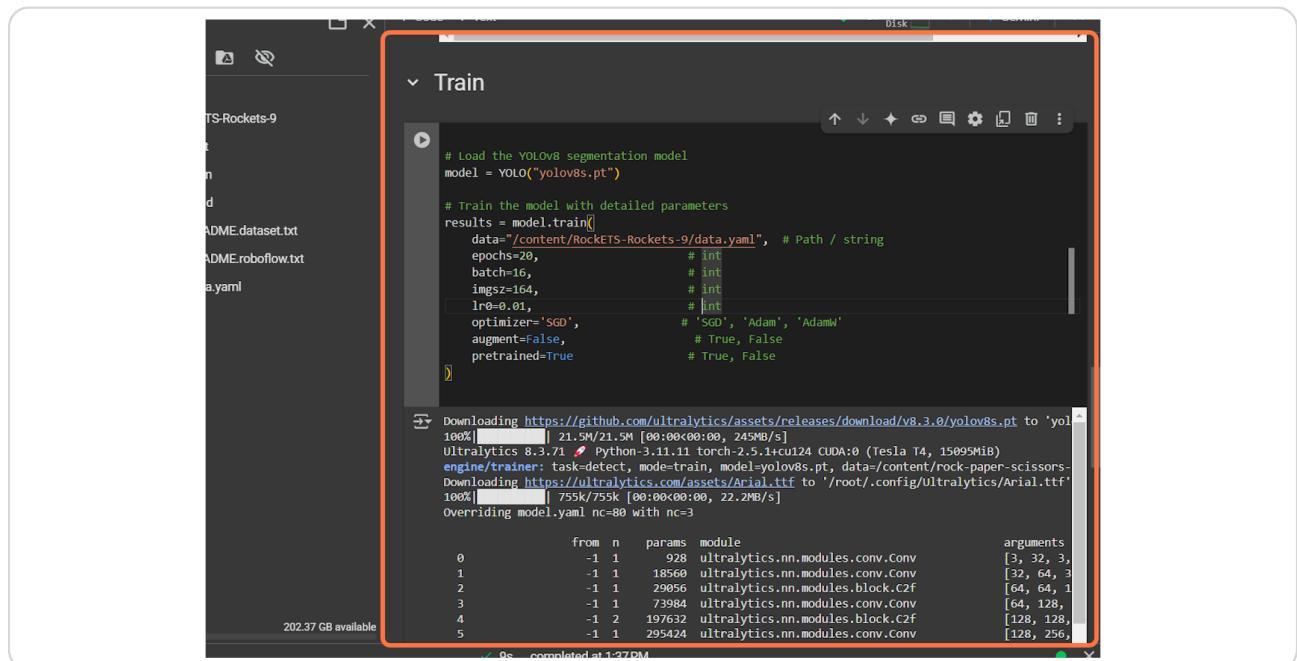
```
# Load the YOLOv8 segmentation model
model = YOLO("yolov8s.pt")

# Train the model with detailed parameters
results = model.train(
    data='/content/RockETS-Rockets-9/data.yaml', # Path / string
    epochs=50, # int
    batch=16, # int
    imgsz=640, # int
    lr0=0.01, # int
    optimizer='Adam', # 'SGD', 'Adam', 'AdamW'
    augment=True, # True, False
    pretrained=True # True, False
)

Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8s.pt to 'yolov8s.pt'
100%[██████████] 21.5M/21.5M [00:00<00:00, 245MB/s]
Ultralytics 8.3.71 🚀 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: task=detect, mode=train, model=yolov8s.pt, data=/content/rock-paper-scissors
```

STEP 31

you can play around with other params



```
# Load the YOLOv8 segmentation model
model = YOLO("yolov8s.pt")

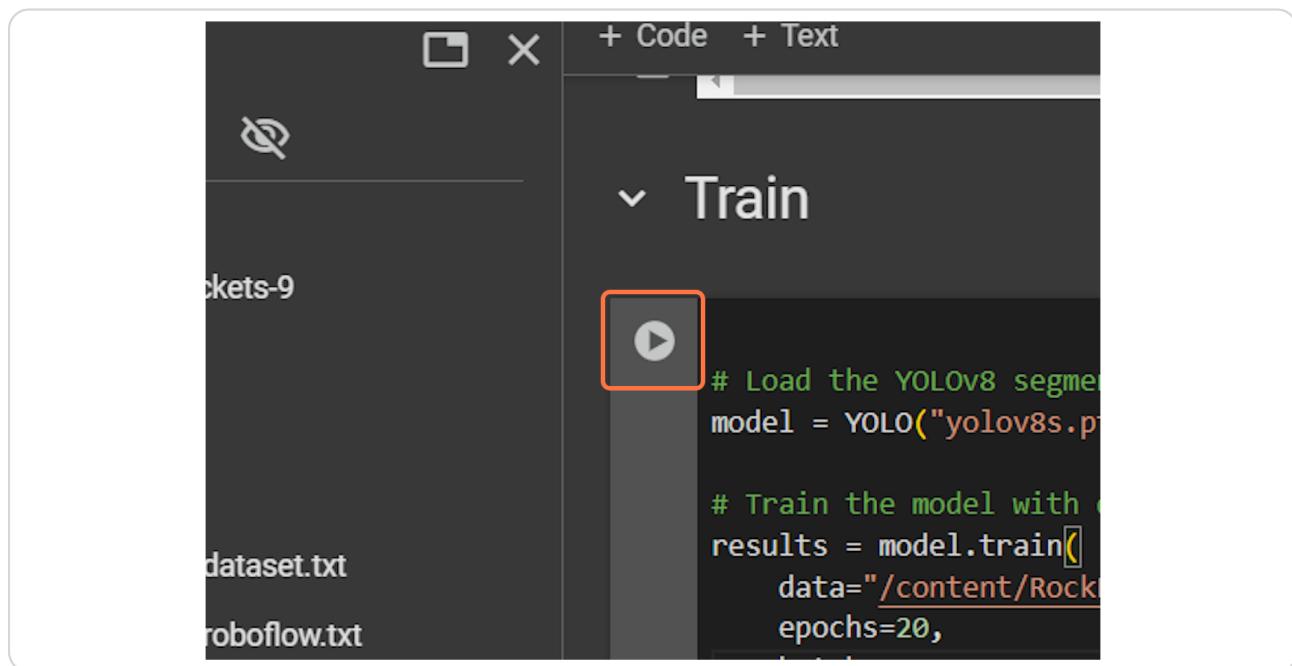
# Train the model with detailed parameters
results = model.train(
    data='/content/RockETS-Rockets-9/data.yaml', # Path / string
    epochs=20, # int
    batch=16, # int
    imgsz=164, # int
    lr0=0.01, # int
    optimizer='SGD', # 'SGD', 'Adam', 'AdamW'
    augment=False, # True, False
    pretrained=True # True, False
)

Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8s.pt to 'yolov8s.pt'
100%[██████████] 21.5M/21.5M [00:00<00:00, 245MB/s]
Ultralytics 8.3.71 🚀 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: task=detect, mode=train, model=yolov8s.pt, data=/content/rock-paper-scissors
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'
100%[██████████] 755k/755k [00:00<00:00, 22.2MB/s]
Overriding model.yaml nc=80 with nc=3

      from n  params module           arguments
0       -1  1      928 ultralytics.nn.modules.conv.Conv   [3, 32, 3]
1       -1  1     18560 ultralytics.nn.modules.conv.Conv  [32, 64, 3]
2       -1  1     29056 ultralytics.nn.modules.block.C2f  [64, 64, 1]
3       -1  1     73984 ultralytics.nn.modules.conv.Conv  [64, 128, 1]
4       -1  2    197632 ultralytics.nn.modules.block.C2f [128, 128, 1]
5       -1  1     295424 ultralytics.nn.modules.conv.Conv [128, 256, 1]
```

STEP 32

run train and now you are training

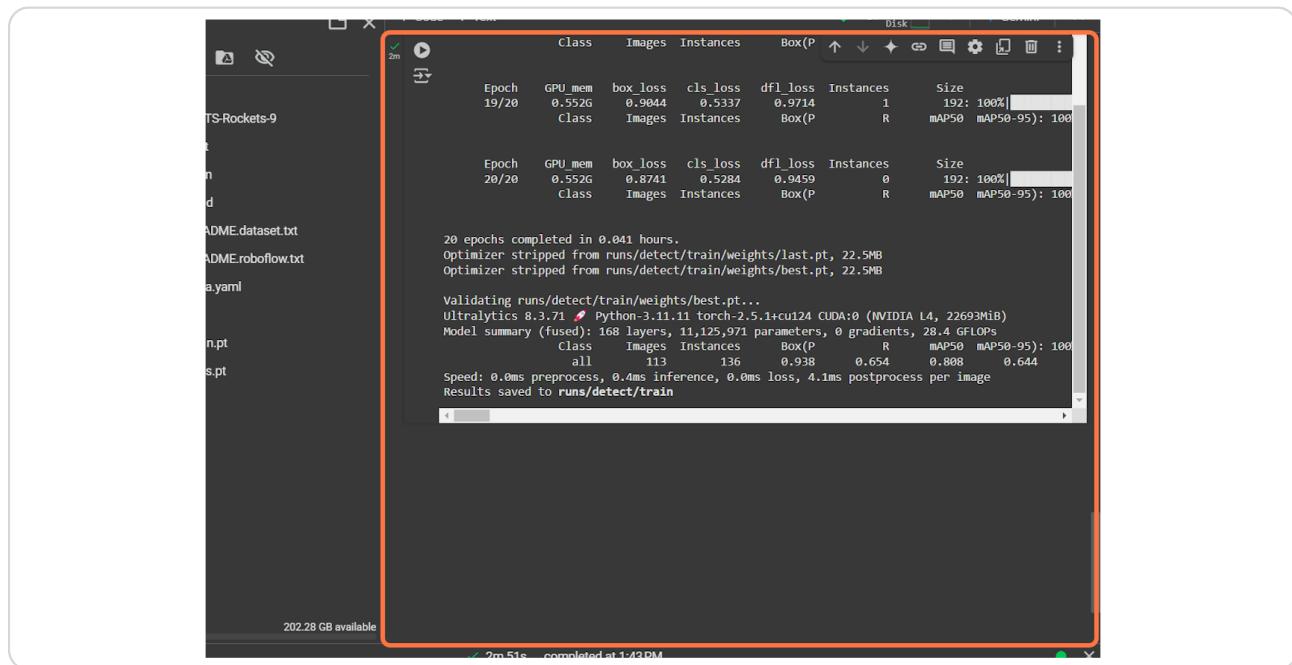


```
+ Code + Text
▼ Train
[play button]
# Load the YOLOv8 segmentation model
model = YOLO("yolov8s.pt")

# Train the model with our dataset
results = model.train(
    data="/content/Rocket-9/dataset.txt",
    epochs=20,
```

STEP 33

this will show up when you train

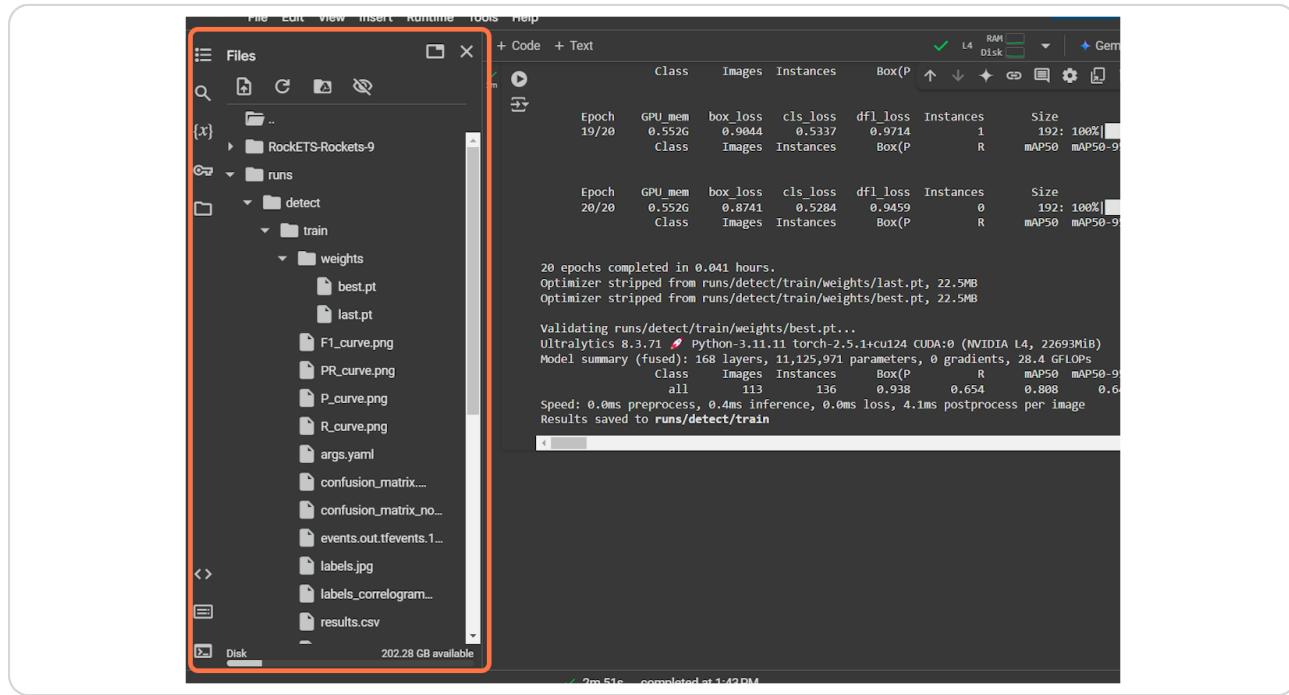


```
2m
Disk
2m
TS-Rockets-9
t
n
d
ADME.dataset.txt
ADME.roboflow.txt
a.yaml
n.pt
s.pt
202.28 GB available
2m 51s completed at 1:42 PM
Epoch 19/20 GPU mem box loss cls loss df1_loss Instances Size
0.552G 0.9044 0.5337 0.9714 1 192: 100% mAP50 mAP50-95: 100%
Class Images Instances Box(P) R
Epoch 20/20 GPU mem box loss cls loss df1_loss Instances Size
0.552G 0.8741 0.5284 0.9459 0 192: 100% mAP50 mAP50-95: 100%
Class Images Instances Box(P) R
20 epochs completed in 0.041 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB
Validating runs/detect/weights/best.pt...
Ultralytics 8.3.7! Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (NVIDIA L4, 22693MiB)
Model summary (Fused): 168 layers, 11,125,971 parameters, 0 gradients, 28.4 GFLOPs
          Class   Images  Instances   Box(P)      R      mAP50  mAP50-95: 100%
          all     113       136      0.938     0.654     0.808     0.644
Speed: 0.0ms preprocess, 0.4ms inference, 0.0ms loss, 4.1ms postprocess per image
Results saved to runs/detect/train
```

STEP 34

head to files and go 'runs' > 'detect' > 'train' > weights

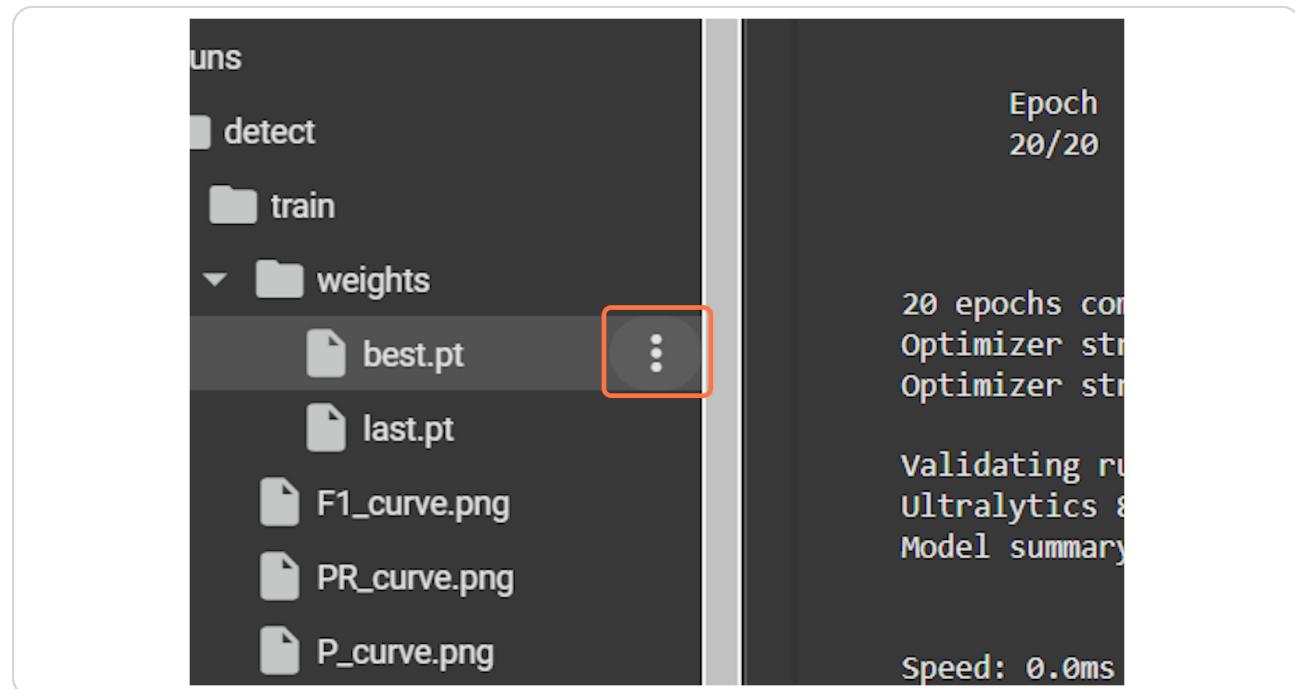
F1 curve and other generated plots can indicate how well ur model learned



STEP 35

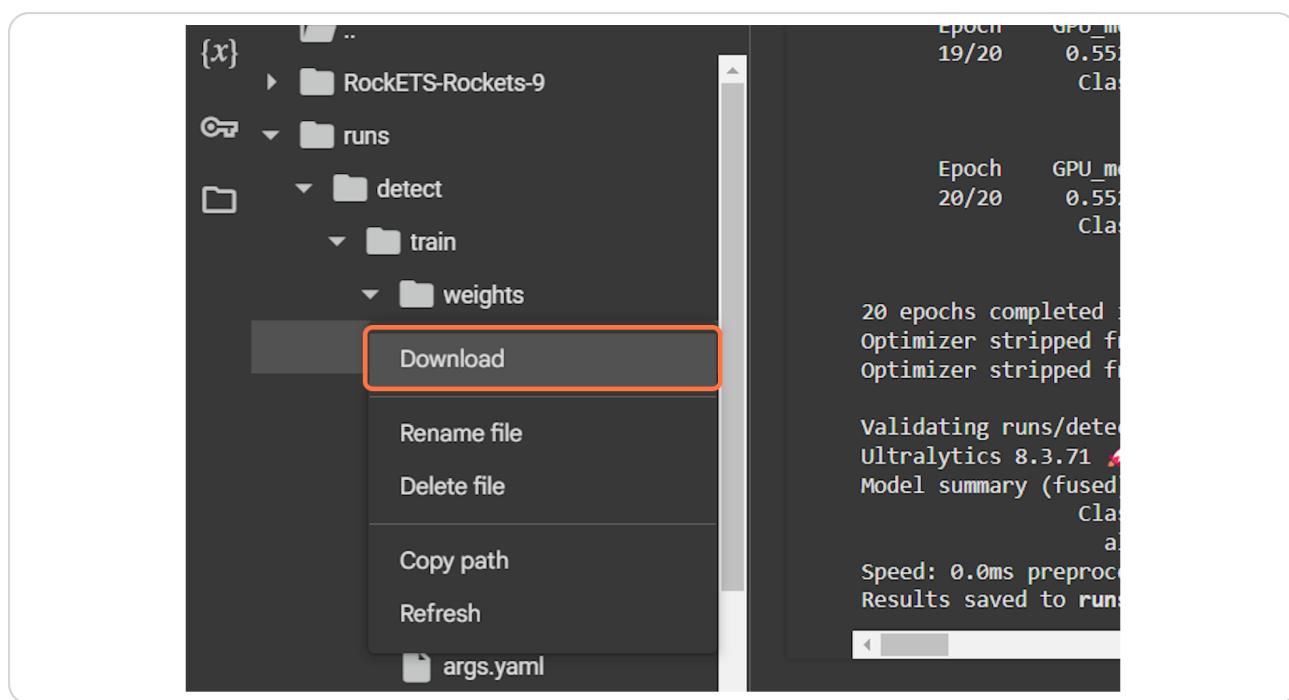
Click on 3 dots next to best.pt (or last.pt)

best is best epoch model, last is the last epoch model



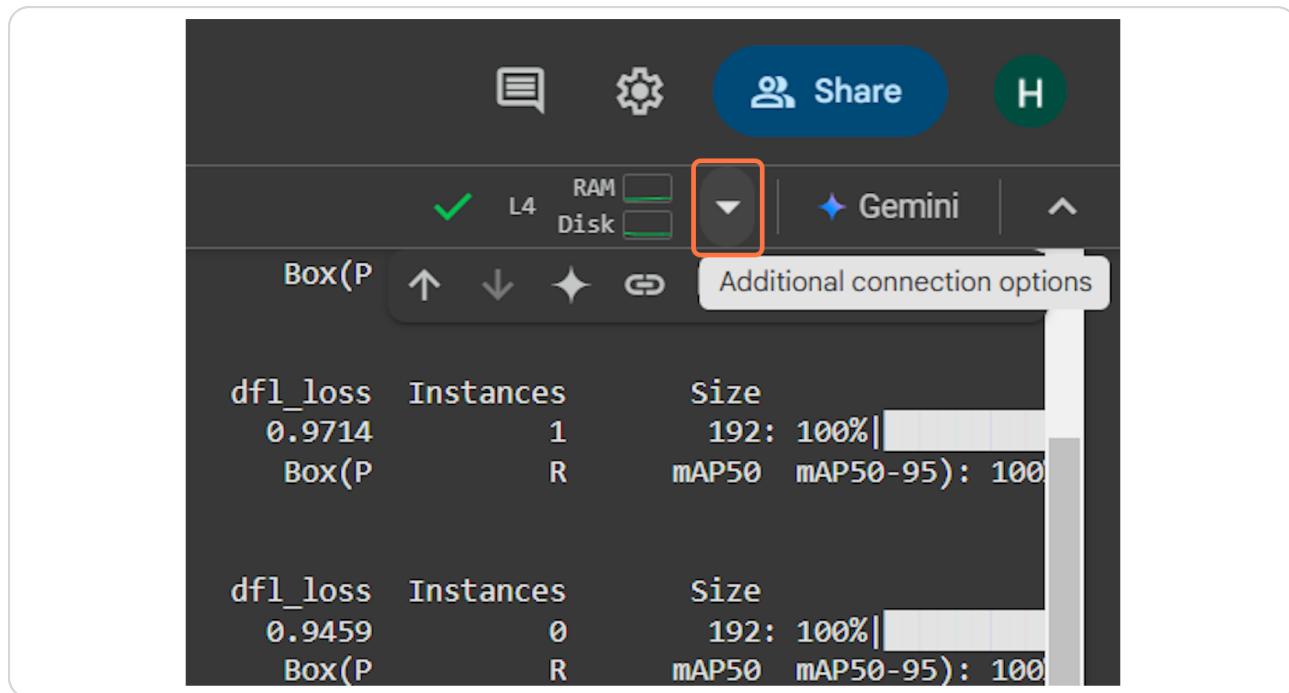
STEP 36

Click on Download



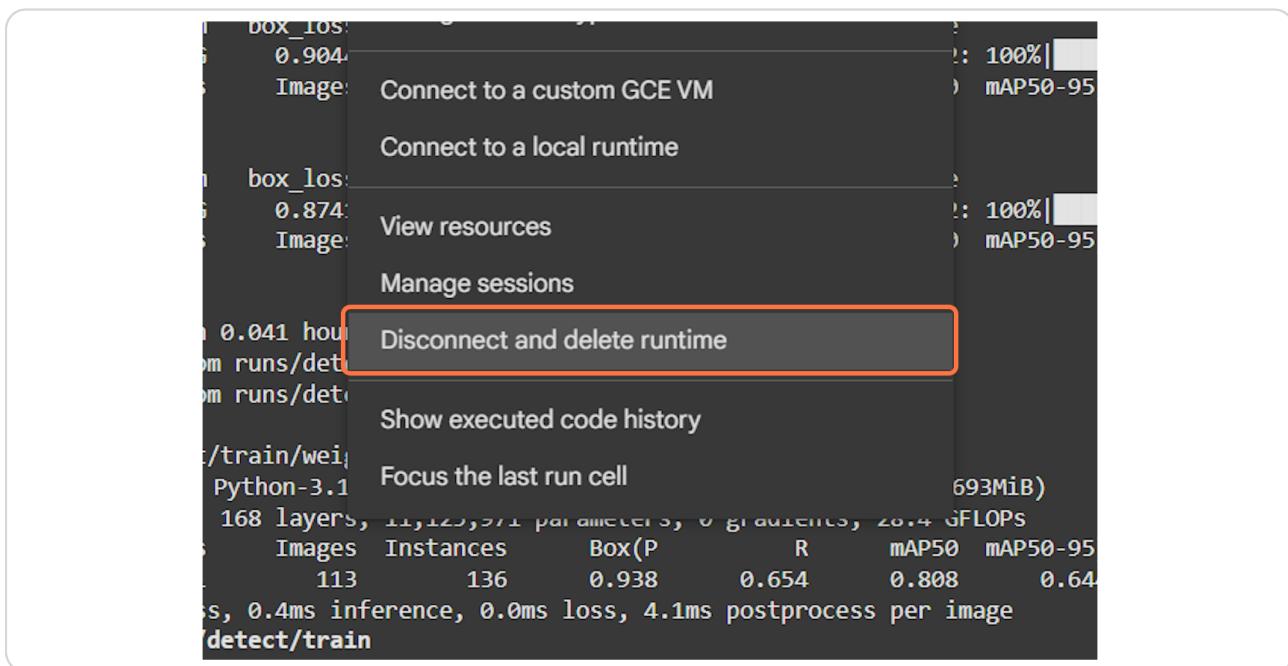
STEP 37

now you can shutdown your collab (not everything will be deleted download anything else needed)



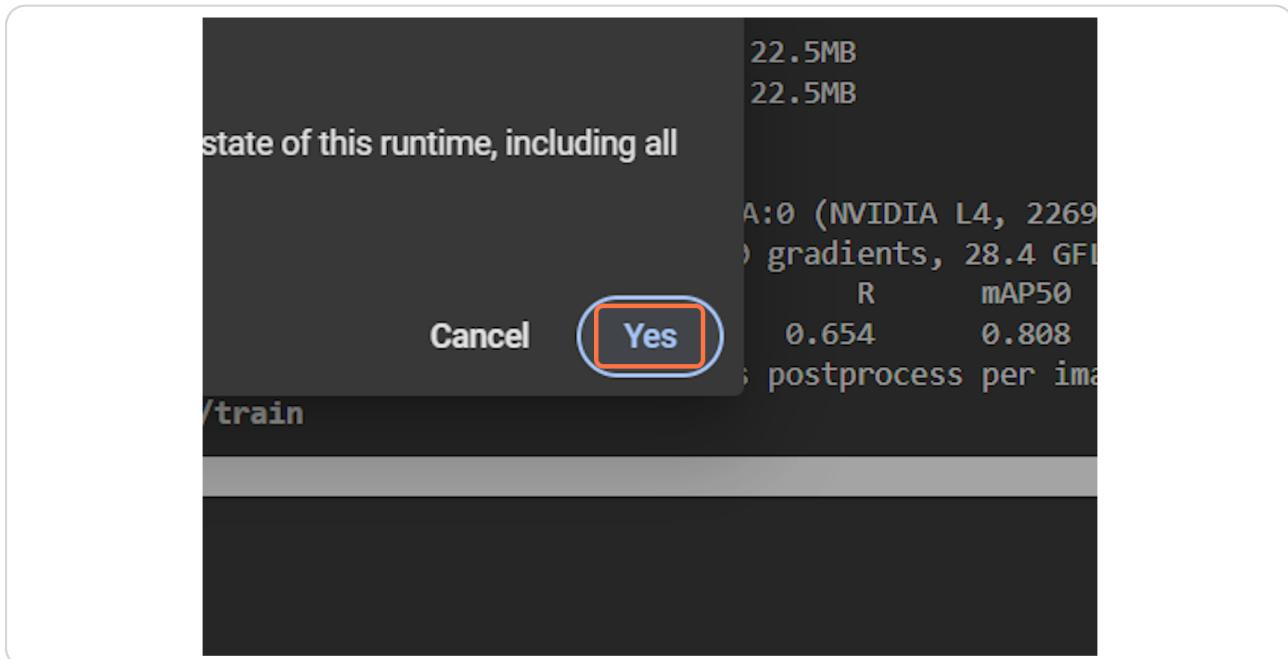
STEP 38

Click on Disconnect and delete runtime



STEP 39

Click on highlight



Tango

Never miss a step again. Visit Tango.ai