Importing 'os' module

> os is the Python standard library's os module, which provides a way to interact with the operating system, including file and directory operations

```
import os
```

Import the glob module: When you import the glob module in Python, you gain access to functions for working with file paths and directories. The glob module is particularly useful for searching for files that match specific patterns and for listing files or directories within a given path.

```
import glob
```

In Google Colab, the from IPython.display import Image, display statement is used for displaying images or other media content directly in your Colab notebook.

```
from IPython.display import Image, display
```

Running !nvidia-smi in a Google Colab notebook is a command used to check and display information about the available NVIDIA GPU (Graphics Processing Unit) resources on the Colab virtual machine

The ! at the beginning of the command is used in Colab notebooks to run shell commands in a code cell.

```
!nvidia-smi
```

```
Sun Oct 15 12:11:33 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   34C    P8     9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

The line HOME = os.getcwd() is used to store the current working directory (CWD) of your Python script or program in the variable HOME. This is often done to create a reference to the current working directory so that you can easily use it throughout your script.

os.getcwd() is a function from the os module that stands for "get current working directory.

HOME is a variable where you store the result of os.getcwd(). You can name this variable whatever you want; HOME is often used to represent the home or root directory for a user.

```
HOME = os.getcwd()
```

Double-click (or enter) to edit

print(HOME) will display the path to the current working directory within your Colab environment.

```
print(HOME)
```

```
/content
```

Running !pip install ultralytics in a Google Colab notebook is used to install the "Ultralytics" library. Ultralytics is a popular open-source computer vision and deep learning library that provides a high-level, easy-to-use interface for training and evaluating computer vision models, particularly those related to object detection, image classification, and more.

```
!pip install ultralytics
```

```
Collecting ultralytics
  Downloading ultralytics-8.0.198-py3-none-any.whl (641 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 641.7/641.7 kB 5.2 MB/s eta 0:00:00
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (3.7.1)
Requirement already satisfied: numpy>=1.22.2 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.23.5)
```

```
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.8.0.76)
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.4.0)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (6.0.1)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.31.0)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.11.3)
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.0.1+cu118)
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.15.2+cu118)
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.66.1)
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.5.3)
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.12.2)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.0.0)
Collecting thop>=0.1.1 (from ultralytics)
  Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->ultralytics) (2023.3.post1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (2023.7.22)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.12.4)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (4.5.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch>=1.8.0->ultralytics) (3.27.6)
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch>=1.8.0->ultralytics) (17.0.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=3.3.0->ultralytics) (1
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.8.0->ultralytics) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.8.0->ultralytics) (1.3.0)
Installing collected packages: thop, ultralytics
Successfully installed thop-0.1.1.post2209072238 ultralytics-8.0.198
```

Once you've installed Ultralytics in your Google Colab notebook using !pip install ultralytics, you can import the Ultralytics library and start using its functionalities for various computer vision tasks. To import Ultralytics, you can simply use the import ultralytics

```
import ultralytics
```

This function checks the installed versions of PyTorch and Torchvision, and warns if they're incompatible according to the provided compatibility table based on

```
ultralytics.checks()
```

```
Ultralytics YOLOv8.0.198 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.8/78.2 GB disk)
```

The command !mkdir {HOME}/datasets is used to create a new directory named "datasets" within the current working directory (HOME) in your Google Colab notebook. This is a useful command when you want to organize your workspace by creating a directory to store datasets or other files.

Here's how it works:

! is used to run shell commands in a Colab notebook.

mkdir is a shell command for creating a new directory.

{HOME} is a placeholder for the current working directory obtained using os.getcwd()

```
!mkdir {HOME}/datasets
```

The %cd magic command in Google Colab is used to change the current working directory to the specified directory. In your code snippet, %cd {HOME}/datasets, you are using it to change the working directory to the "datasets" directory that you previously created within your HOME directory.

Here's how it works:

%cd is a Colab magic command that allows you to change the working directory.

{HOME} is a placeholder for the path to your home directory.

/datasets is the directory you want to change into.

```
%cd {HOME}/datasets
```

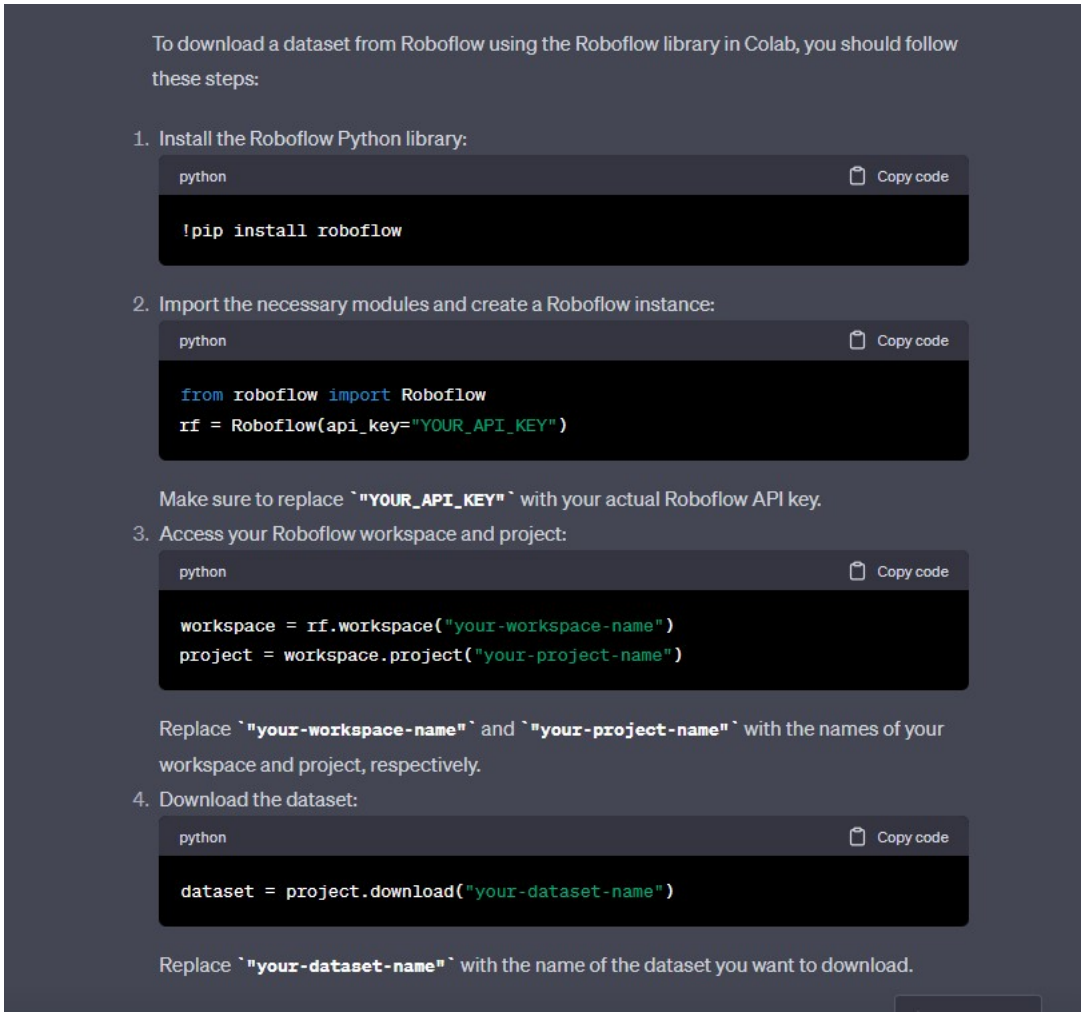
```
/content/datasets
```

The !pwd command is used to print the current working directory (CWD) in a Google Colab notebook. When you run this command, it will display the full path of the directory where your notebook is currently located.

```
!pwd
```

```
/content/datasets
```

To download a dataset from Roboflow using the Roboflow library in Colab, you should follow code:




```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="U7dTYxWd6GhveGuZJTEv")
project = rf.workspace("college-ze5w1").project("sign_language-m6orh")
dataset = project.version(12).download("yolov8")
```

```
Collecting roboflow
  Downloading roboflow-1.1.7-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.8/58.8 kB 1.9 MB/s eta 0:00:00
Collecting certifi==2022.12.7 (from roboflow)
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.3/155.3 kB 7.8 MB/s eta 0:00:00
Collecting chardet==4.0.0 (from roboflow)
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 178.7/178.7 kB 21.6 MB/s eta 0:00:00
Collecting cycler==0.10.0 (from roboflow)
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting idna==2.10 (from roboflow)
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.8/58.8 kB 8.5 MB/s eta 0:00:00
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.23.5)
Collecting opencv-python-headless==4.8.0.74 (from roboflow)
  Downloading opencv_python_headless-4.8.0.74-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (49.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.1/49.1 MB 18.7 MB/s eta 0:00:00
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (9.4.0)
Collecting pyparsing==2.4.7 (from roboflow)
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.8/67.8 kB 9.7 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.31.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Collecting supervision (from roboflow)
  Downloading supervision-0.15.0-py3-none-any.whl (69 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 69.0/69.0 kB 10.9 MB/s eta 0:00:00
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.0.6)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.66.1)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0.1)
Collecting requests-toolbelt (from roboflow)
  Downloading requests_toolbelt-1.0.0-py2.py3-none-any.whl (54 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.5/54.5 kB 8.2 MB/s eta 0:00:00
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.1.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.43.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (23.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.3.0)
Requirement already satisfied: scipy<2.0.0,>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from supervision->roboflow) (1.11.3)
Installing collected packages: python-dotenv, pyparsing, opencv-python-headless, idna, cycler, chardet, certifi, supervision, requests-toolbelt, r
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.1.1
    Uninstalling pyparsing-3.1.1:
      Successfully uninstalled pyparsing-3.1.1
  Attempting uninstall: opencv-python-headless
    Found existing installation: opencv-python-headless 4.8.1.78
    Uninstalling opencv-python-headless-4.8.1.78:
```

The %cd magic command in Google Colab is used to change the current working directory. In your code snippet, %cd {HOME}, you are using it to change the working directory back to your HOME directory, which typically represents your home directory.

Here's how it works:

%cd is a Colab magic command to change the working directory.

{HOME} is a placeholder for the path to your home directory.

```
%cd {HOME}
```

```
/content
```

Double-click (or enter) to edit

```
WARNING: The following packages were previously imported in this runtime:
```

The %cd magic command in Google Colab is used to change the current working directory. In your code snippet, %cd {dataset.location}, you are using it to change the working directory to the location of a specific dataset.

Here's how it works:

%cd is a Colab magic command to change the working directory.

{dataset.location} is a placeholder for the location or path of a specific dataset.

By running %cd {dataset.location}, you are changing the current working directory to the directory where the dataset is located. This can be useful when you want to work with files or data within that specific dataset directory.

```
%cd {dataset.location}
```

```
/content/datasets/Sign_language-12
```

It seems to be running a YOLO training task with specific parameters.

Here's what each part of the command means:

!yolo: This is the command for running YOLO, which is a popular object detection framework.

task=detect: It specifies that you are performing the "detect" task, which typically involves training or running object detection models.

mode=train: It indicates that you are in training mode, where you are training an object detection model.

model=yolov8m.pt: This parameter specifies the model architecture you are using. In this case, it's "yolov8m.pt," which is likely a pre-trained YOLOv8 model.

data=/content/datasets/Sign_language-12/data.yaml: This parameter points to the configuration file for the dataset. The /content/datasets/Sign_language-12/data.yaml file is likely a YAML file containing information about the dataset, such as the data paths and model settings.

epochs=200: This specifies the number of training epochs. The model will be trained for 200 epochs.

imgsz=640: It specifies the image size for training. In this case, images are likely being resized to 640x640 pixels.

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8m.pt data=/content/datasets/Sign_language-12/data.yaml epochs=200 imgsz=640
```

```
[Errno 2] No such file or directory: '{HOME}'
/content
Ultralytics YOLOv8.0.198 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8m.pt, data=/content/datasets/Sign_language-12/data.yaml, epochs=200, patience=50, batch=16
Overriding model.yaml nc=80 with nc=5

                   from  n    params  module                                       arguments
  0                  -1  1      1392  ultralytics.nn.modules.conv.Conv             [3, 48, 3, 2]
  1                  -1  1     41664  ultralytics.nn.modules.conv.Conv             [48, 96, 3, 2]
  2                  -1  2    111360  ultralytics.nn.modules.block.C2f             [96, 96, 2, True]
  3                  -1  1    166272  ultralytics.nn.modules.conv.Conv             [96, 192, 3, 2]
  4                  -1  4    813312  ultralytics.nn.modules.block.C2f             [192, 192, 4, True]
  5                  -1  1    664320  ultralytics.nn.modules.conv.Conv             [192, 384, 3, 2]
  6                  -1  4   3248640  ultralytics.nn.modules.block.C2f             [384, 384, 4, True]
  7                  -1  1   1991808  ultralytics.nn.modules.conv.Conv             [384, 576, 3, 2]
  8                  -1  2   3985920  ultralytics.nn.modules.block.C2f             [576, 576, 2, True]
  9                  -1  1    831168  ultralytics.nn.modules.block.SPPF            [576, 576, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  2   1993728  ultralytics.nn.modules.block.C2f             [960, 384, 2]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  2    517632  ultralytics.nn.modules.block.C2f             [576, 192, 2]
 16                  -1  1    332160  ultralytics.nn.modules.conv.Conv             [192, 192, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  2   1846272  ultralytics.nn.modules.block.C2f             [576, 384, 2]
 19                  -1  1   1327872  ultralytics.nn.modules.conv.Conv             [384, 384, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  2   4207104  ultralytics.nn.modules.block.C2f             [960, 576, 2]
 22        [15, 18, 21]  1   3778591  ultralytics.nn.modules.head.Detect           [5, [192, 384, 576]]
Model summary: 295 layers, 25859215 parameters, 25859199 gradients, 79.1 GFLOPs

Transferred 469/475 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs/detect/train4', view at http://localhost:6006/
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...
AMP: checks passed ✅
train: Scanning /content/datasets/Sign_language-12/train/labels.cache... 870 images, 0 backgrounds, 0 corrupt: 100% 870/870 [00:00<?, ?it/s]
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile
val: Scanning /content/datasets/Sign_language-12/valid/labels.cache... 40 images, 0 backgrounds, 0 corrupt: 100% 40/40 [00:00<?, ?it/s]
Plotting labels to runs/detect/train4/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automaticall
optimizer: AdamW(lr=0.001111, momentum=0.9) with parameter groups 77 weight(decay=0.0), 84 weight(decay=0.0005), 83 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/detect/train4
Starting training for 200 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      1/200      6.84G      2.087      7.247      2.215         26        640:  15% 8/55 [00:07<00:42,  1.11it/s]
Traceback (most recent call last):
  File "/usr/local/bin/yolo", line 8, in <module>
    sys.exit(entrypoint())
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/cfg/__init__.py", line 444, in entrypoint
    getattr(model, mode)(**overrides)  # default args from model
  File "/usr/local/lib/python3.10/dist-packages/ultralytics/engine/model.py", line 341, in train
    self.trainer.train()
```