YOLOv11 Training in Google Colab - Explained Code

Name : Shehryar Khan

Mount Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Explanation: *Mounts your Google Drive to access files from within Google Colab.*

Define Paths

```
zip_file_path_in_drive = "/content/drive/My Drive/Colab_Datasets/archive (3).zip"
destination_path_in_colab = "/content/archive.zip"
```

Explanation: *Defines the original zip file path in Google Drive and sets a simplified path for use in Colab.*

Copy ZIP File

```
!cp "{zip_file_path_in_drive}" "{destination_path_in_colab}"
```

Explanation: *Copies the zip file from Google Drive to Colab's working directory.*

Remove Old Folder (Optional)

```
!rm -rf /content/BrainTumor
```

Explanation: *Deletes any previous extracted dataset to prevent conflicts.*

Unzip Dataset

```
!unzip -o -q "{destination_path_in_colab}" -d "/content/"
```

Explanation: *Extracts the dataset zip file into Colab's working directory.*

List Directory Contents

```
!ls /content
```

Explanation: *Lists all files and directories in /content to verify dataset extraction.*

Import YOLO and OS

```
from ultralytics import YOLO
import os
```

Explanation: *Imports the YOLO model class and Python's os module.*

Set Path to YAML File

```
data_yaml_path =
"/content/BrainTumor/BrainTumorYolov11/data.yaml"
```

Explanation: *Defines the path to your dataset’s configuration file (data.yaml).*

Verify YAML File Exists

```
if os.path.exists(data_yaml_path):
    print(f"The file '{data_yaml_path}' exists in Colab.")
else:
    raise FileNotFoundError(f"Error: The file '{data_yaml_path}' does
not exist in Colab.")
```

Explanation: *Checks if the data.yaml file exists to prevent training errors.*

Load YOLOv11 Model

```
model = YOLO("yolo11n.pt")
```

Explanation: *Loads the pre-trained YOLOv11n model. Make sure the model file is available.*

Start Model Training

```
print("Starting training...")
trained_model = model.train(
   data=data_yaml_path,
   epochs=18,
   imgsz=640,
   batch=16,
   device=0
)
```

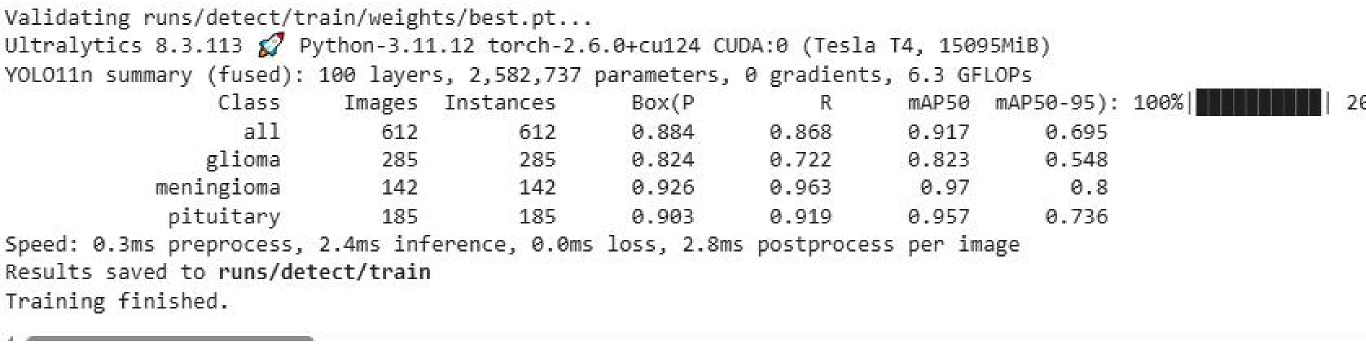
Explanation: *Begins training the model with specified settings (epochs, image size, batch size, and device).*

Training Completion Message

```
print("Training finished.")
```

Explanation: *Indicates that the training process has finished.*

Output



```
Validating runs/detect/train/weights/best.pt...
Ultralytics 8.3.113 🚀 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,582,737 parameters, 0 gradients, 6.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 20
                   all        612        612      0.884      0.868      0.917      0.695
                glioma        285        285      0.824      0.722      0.823      0.548
            meningioma        142        142      0.926      0.963       0.97        0.8
             pituitary        185        185      0.903      0.919      0.957      0.736
Speed: 0.3ms preprocess, 2.4ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to runs/detect/train
Training finished.
```