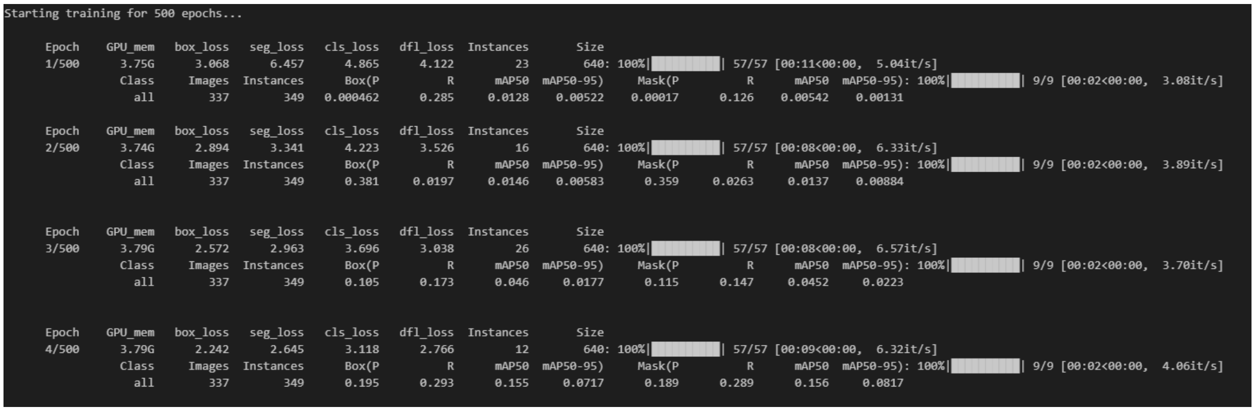
User Guide for Training Classification Models



```
Starting training for 500 epochs...

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
      1/500      3.75G      3.068      6.457      4.865      4.122         23        640: 100%|██████████| 57/57 [00:11<00:00,  5.04it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|██████████| 9/9 [00:02<00:00,  3.08it/s]
                   all        337        349   0.000462      0.285     0.0128    0.00522    0.00017      0.126    0.00542    0.00131

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
      2/500      3.74G      2.894      3.341      4.223      3.526         16        640: 100%|██████████| 57/57 [00:08<00:00,  6.33it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|██████████| 9/9 [00:02<00:00,  3.89it/s]
                   all        337        349      0.381     0.0197     0.0146    0.00583      0.359     0.0263     0.0137    0.00884


      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
      3/500      3.79G      2.572      2.963      3.696      3.038         26        640: 100%|██████████| 57/57 [00:08<00:00,  6.57it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|██████████| 9/9 [00:02<00:00,  3.70it/s]
                   all        337        349      0.105      0.173      0.046     0.0177      0.115      0.147     0.0452     0.0223


      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
      4/500      3.79G      2.242      2.645      3.118      2.766         12        640: 100%|██████████| 57/57 [00:09<00:00,  6.32it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|██████████| 9/9 [00:02<00:00,  4.06it/s]
                   all        337        349      0.195      0.293      0.155     0.0717      0.189      0.289      0.156     0.0817
```

Note:
The following example shows how the model training was done and how to recreate it. To save time and storage space, the batch size and epochs were reduced, and only a small fraction of the training data was included.

Step 1:
Clone the project repo:
git clone https://github.com/tchiang0/data_515_brain_tumor_computer_vision.git

Step 2:
Install pip if not already installed:

python -m pip install

Step 3:
Run requirements.txt to ensure all dependencies exist:

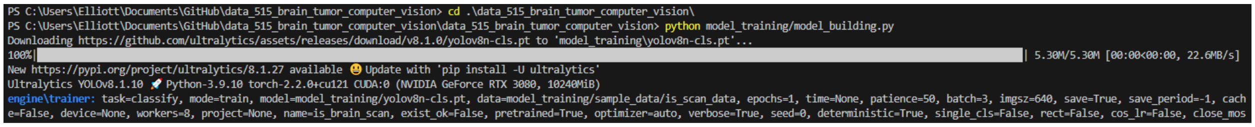
pip install -r requirements.txt

Step 4:
Navigate into the data_515_brain_tumor_computer_vision project directory and then into data_515_brain_tumor_computer_vision folder:

cd data_515_brain_tumor_computer_vision/data_515_brain_tumor_computer_vision

Step 5:
Run the following command from inside the data_515_brain_tumor_computer_vision folder to train the two classification models (one to determine if the uploaded image is a brain scan, one to determine if the uploaded brain scan has a tumor).

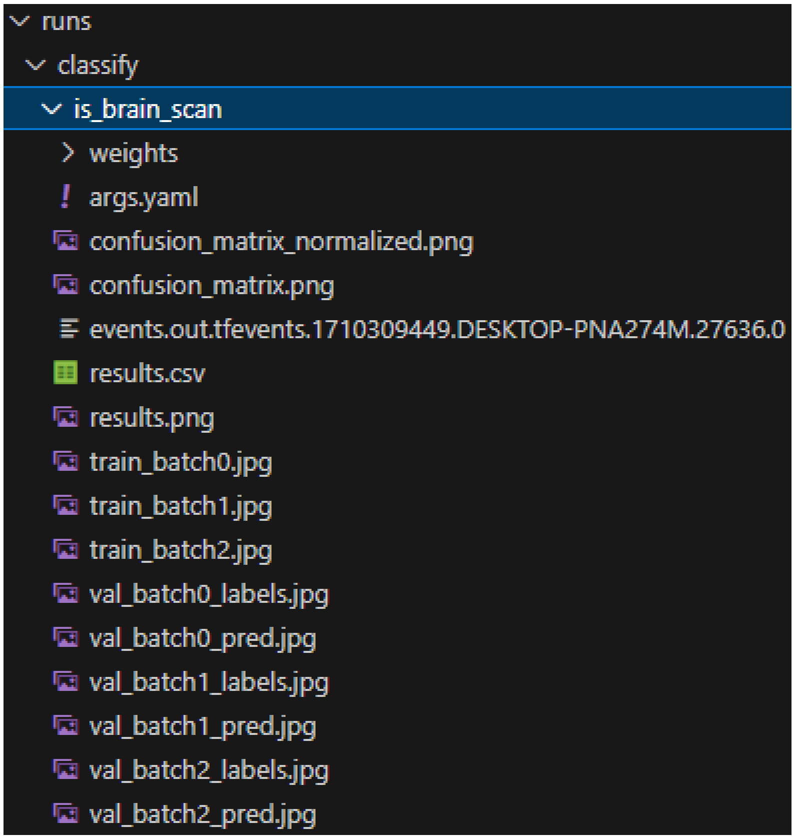
python model_training/model_building.py



```
PS C:\Users\Elliott\Documents\GitHub\data_515_brain_tumor_computer_vision> cd .\data_515_brain_tumor_computer_vision\
PS C:\Users\Elliott\Documents\GitHub\data_515_brain_tumor_computer_vision\data_515_brain_tumor_computer_vision> python model_training/model_building.py
Downloading https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n-cls.pt to 'model_training\yolov8n-cls.pt'...
100%|████████████████████████████████████████████████████████████████████████████████| 5.30M/5.30M [00:00<00:00, 22.6MB/s]
New https://pypi.org/project/ultralytics/8.1.27 available 😃 Update with 'pip install -U ultralytics'
Ultralytics YOLOv8.1.10 🚀 Python-3.9.10 torch-2.2.0+cu121 CUDA:0 (NVIDIA GeForce RTX 3080, 10240MiB)
engine\trainer: task=classify, mode=train, model=model_training/yolov8n-cls.pt, data=model_training/sample_data/is_scan_data, epochs=1, time=None, patience=50, batch=3, imgsz=640, save=True, save_period=-1, cach
e=False, device=None, workers=8, project=None, name=is_brain_scan, exist_ok=False, pretrained=True, optimizer=auto, verbose=True, seed=0, deterministic=True, single_cls=False, rect=False, cos_lr=False, close_mos
```

Step 6:
Navigate to the folder where the weights and results are stored:

runs/classify/is_tumor
runs/classify/is_brain_scan



```
runs
  classify
    is_brain_scan
      weights
      args.yaml
      confusion_matrix_normalized.png
      confusion_matrix.png
      events.out.tfevents.1710309449.DESKTOP-PNA274M.27636.0
      results.csv
      results.png
      train_batch0.jpg
      train_batch1.jpg
      train_batch2.jpg
      val_batch0_labels.jpg
      val_batch0_pred.jpg
      val_batch1_labels.jpg
      val_batch1_pred.jpg
      val_batch2_labels.jpg
      val_batch2_pred.jpg
```

Final notes:

These steps are also executed automatically when running program tests. To see which training data is used, navigate to the model_training/sample_data folder.