Enable Kaggle APIlink text

```
# Install Kaggle API
!pip install kaggle

# Create directory
!mkdir -p ~/.kaggle

# Copy kaggle.json (must be uploaded first)
!cp /content/kaggle.json ~/.kaggle/

# Change permissions
!chmod 600 ~/.kaggle/kaggle.json

print("Kaggle API Successfully Configured!")
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.12
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.1
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/pytho
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/di
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.1
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.12/di
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/d
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist
Kaggle API Successfully Configured!
```

Clone YOLOv8

```
!git clone https://github.com/ultralytics/ultralytics
%cd ultralytics
```

```
Cloning into 'ultralytics'...
remote: Enumerating objects: 76112, done.
remote: Counting objects: 100% (117/117), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 76112 (delta 76), reused 64 (delta 51), pack-reused 75995 (from
Receiving objects: 100% (76112/76112), 40.53 MiB | 15.23 MiB/s, done.
Resolving deltas: 100% (57220/57220), done.
/content/ultralytics
```

Install YOLOv8 (editable mode)

```
!pip install -e .
import torch
print("GPU:", torch.cuda.get_device_name(0))
```

```
Obtaining file:///content/ultralytics
  Installing build dependencies ... done
  Checking if build backend supports build_editable ... done
  Getting requirements to build editable ... done
  Preparing editable metadata (pyproject.toml) ... done
Requirement already satisfied: numpy>=1.23.0 in /usr/local/lib/python3.12/d
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/pytho
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.12/d
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.12/d
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.1
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.12/di
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.12/di
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3
Requirement already satisfied: psutil>=5.8.0 in /usr/local/lib/python3.12/d
Requirement already satisfied: polars>=0.20.0 in /usr/local/lib/python3.12/
Collecting ultralytics-thop>=2.0.18 (from ultralytics==8.3.233)
  Downloading ultralytics_thop-2.0.18-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/di
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pytho
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/py
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/di
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/p
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/d
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-pack
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/d
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/loc
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/li
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/li
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/l
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/l
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/py
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/p
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/li
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/di
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3
```

```
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/
Downloading ultralytics_thop-2.0.18-py3-none-any.whl (28 kB)
Building wheels for collected packages: ultralytics
```

Double-click (or enter) to edit

Prepare Dataset Folder

```
import os, shutil

print("🧹 Cleaning datasets directory...")
if os.path.exists('/content/datasets'):
    shutil.rmtree('/content/datasets')

os.makedirs('/content/datasets')

%cd /content/datasets
```

```
🧹 Cleaning datasets directory...
/content/datasets
```

Download Brain Tumor Dataset

```
print("⬇️ Downloading Brain Tumor Dataset...")
!kaggle datasets download -d pkdarabi/medical-image-dataset-brain-tumor-dete
```

```
⬇️ Downloading Brain Tumor Dataset...
Dataset URL: https://www.kaggle.com/datasets/pkdarabi/medical-image-dataset-b
License(s): Attribution 4.0 International (CC BY 4.0)
Downloading medical-image-dataset-brain-tumor-detection.zip to /content/datas
 98% 291M/297M [00:06<00:00, 37.7MB/s]
100% 297M/297M [00:06<00:00, 45.0MB/s]
```

unzip data set

```
print("📦 Unzipping...")
!unzip -q medical-image-dataset-brain-tumor-detection.zip
!rm medical-image-dataset-brain-tumor-detection.zip
```

```
📦 Unzipping...
```

Auto-Fix data.yaml

```
import os

print("🔍 Locating data.yaml...")
```

```
yaml_path = None

for root, dirs, files in os.walk('/content/datasets'):
    if 'data.yaml' in files:
        yaml_path = os.path.join(root, 'data.yaml')
        break

if yaml_path is None:
    raise FileNotFoundError("❌ data.yaml not found!")

dataset_root = os.path.abspath(os.path.dirname(yaml_path))

yaml_content = f"""
path: {dataset_root}
train: train/images
val: valid/images
test: test/images

nc: 3
names: ['glioma', 'meningioma', 'pituitary']
"""

with open(yaml_path, 'w') as f:
    f.write(yaml_content)

print("✅ FIXED data.yaml at:", yaml_path)
print("📍 Dataset root:", dataset_root)
```

```
🔍 Locating data.yaml...
✅ FIXED data.yaml at: /content/datasets/BrainTumor/BrainTumorYolov11/data.ya
📍 Dataset root: /content/datasets/BrainTumor/BrainTumorYolov11
```

\NOW YOUR BASIC DATASET + YOLO SETUP IS READY

PolarizedSelfAttention to tasks.py

```
import re

tasks_path = "/content/ultralytics/ultralytics/nn/tasks.py"

psa_code = """
# --- PolarizedSelfAttention START ---
import torch
import torch.nn as nn

class PolarizedSelfAttention(nn.Module):
    \"\"\"Lightweight Polarized Self-Attention:
    - Channel branch: global pooling + MLP (channel focus)
    - Spatial branch: 1x1 conv + sigmoid (spatial focus)
    Output = x * channel_att * spatial_att
```

```
        \"\"\"
        def __init__(self, c1, c2=None, reduction=16):
            super().__init__()
            if c2 is None:
                c2 = c1
            if c1 != c2:
                c1 = c2
            self.c = c1

            mid = max(8, c1 // reduction)

            # Channel attention (like SE)
            self.ca_mlp = nn.Sequential(
                nn.Linear(c1, mid, bias=False),
                nn.ReLU(inplace=True),
                nn.Linear(mid, c1, bias=False),
                nn.Sigmoid()
            )

            # Spatial attention (1x1 conv → 1 channel)
            self.sa_conv = nn.Conv2d(c1, 1, kernel_size=1, bias=False)

        def forward(self, x):
            b, c, h, w = x.shape

            # ----- Channel branch -----
            gap = x.view(b, c, -1).mean(-1)          # (b, c)
            ca = self.ca_mlp(gap).view(b, c, 1, 1)   # (b, c, 1, 1)

            # ----- Spatial branch -----
            sa = torch.sigmoid(self.sa_conv(x))      # (b, 1, h, w)

            # Broadcast and combine: polarized (channel * spatial)
            x = x * ca * sa

            return x
    # --- PolarizedSelfAttention END ---
    """

    print("🔧 Injecting PolarizedSelfAttention into tasks.py...")

    # Read file as list of lines
    with open(tasks_path, "r") as f:
        lines = f.readlines()

    # Find where BaseModel starts
    insert_index = None
    for i, line in enumerate(lines):
        if line.strip().startswith("class BaseModel"):
            insert_index = i
            break

    if insert_index is None:
        print("❌ ERROR: Could not find class BaseModel.")
    else:
```

```
    # Insert our code block above BaseModel
    new_lines = lines[:insert_index] + [psa_code + "\n"] + lines[insert_inde

    with open(tasks_path, "w") as f:
        f.writelines(new_lines)

    print("✅ PolarizedSelfAttention class inserted above BaseModel.")

    # Now register in the safe module list
    with open(tasks_path, "r") as f:
        content = f.read()

    content = content.replace(
        "if m in (",
        "if m in (PolarizedSelfAttention, ",
        1
    )

    with open(tasks_path, "w") as f:
        f.write(content)

    print("✅ PolarizedSelfAttention registered in module list.")
```

```
🔧 Injecting PolarizedSelfAttention into tasks.py...
✅ PolarizedSelfAttention class inserted above BaseModel.
✅ PolarizedSelfAttention registered in module list.
```

Create yolov8n-psa.yaml

```
psa_yaml = """
# YOLOv8n + Polarized Self-Attention

nc: 3  # glioma, meningioma, pituitary

backbone:
  - [-1, 1, Conv, [64, 3, 2]]
  - [-1, 1, Conv, [128, 3, 2]]
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]]
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, Conv, [512, 3, 2]]
  - [-1, 6, C2f, [512, True]]
  - [-1, 1, Conv, [1024, 3, 2]]
  - [-1, 3, C2f, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]]

  # Polarized Self-Attention at deepest feature map (1024 channels)
  - [-1, 1, PolarizedSelfAttention, [1024]]

head:
  - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
  - [[-1, 6], 1, Concat, [1]]
  - [-1, 3, C2f, [512]]
```

```
  - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
  - [[-1, 4], 1, Concat, [1]]
  - [-1, 3, C2f, [256]]

  - [-1, 1, Conv, [256, 3, 2]]
  - [[-1, 13], 1, Concat, [1]]
  - [-1, 3, C2f, [512]]

  - [-1, 1, Conv, [512, 3, 2]]
  - [[-1, 10], 1, Concat, [1]]
  - [-1, 3, C2f, [1024]]

  - [[16, 19, 22], 1, Detect, [nc]]
"""

with open("/content/ultralytics/yolov8n-psa.yaml", "w") as f:
    f.write(psa_yaml)

print("✅ Created yolov8n-psa.yaml")
```

```
✅ Created yolov8n-psa.yaml
```

Train YOLOv8 + Polarized Self-Attention

```
%cd /content/ultralytics
import os
os.environ["WANDB_DISABLED"] = "true"

!yolo detect train \
  data=/content/datasets/BrainTumor/BrainTumorYolov11/data.yaml \
  model=/content/ultralytics/yolov8n-psa.yaml \
  epochs=50 \
  imgsz=640 \
  batch=16 \
  name=YOLOv8_PSA
```

```
/content/ultralytics
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralyt
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir
Ultralytics 8.3.233 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 1
engine/trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=ra
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultra

                   from  n    params  module
  0                  -1  1      1856  ultralytics.nn.modules.conv.Conv
  1                  -1  1     73984  ultralytics.nn.modules.conv.Conv
  2                  -1  3    279808  ultralytics.nn.modules.block.C2f
  3                  -1  1    295424  ultralytics.nn.modules.conv.Conv
  4                  -1  6   2101248  ultralytics.nn.modules.block.C2f
  5                  -1  1   1180672  ultralytics.nn.modules.conv.Conv
  6                  -1  6   8396800  ultralytics.nn.modules.block.C2f
```

```
  7                  -1  1   4720640  ultralytics.nn.modules.conv.Conv
  8                  -1  3  17836032  ultralytics.nn.modules.block.C2f
  9                  -1  1   2624512  ultralytics.nn.modules.block.SPPF
 10                  -1  1    132096  ultralytics.nn.tasks.PolarizedSelfAtte
 11                  -1  1         0  torch.nn.modules.upsampling.Upsample
 12             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat
 13                  -1  3   4985856  ultralytics.nn.modules.block.C2f
 14                  -1  1         0  torch.nn.modules.upsampling.Upsample
 15             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat
 16                  -1  3   1247744  ultralytics.nn.modules.block.C2f
 17                  -1  1    590336  ultralytics.nn.modules.conv.Conv
 18            [-1, 13]  1         0  ultralytics.nn.modules.conv.Concat
 19                  -1  3   4592640  ultralytics.nn.modules.block.C2f
 20                  -1  1   2360320  ultralytics.nn.modules.conv.Conv
 21            [-1, 10]  1         0  ultralytics.nn.modules.conv.Concat
 22                  -1  3  18360320  ultralytics.nn.modules.block.C2f
 23        [16, 19, 22]  1   7059673  ultralytics.nn.modules.head.Detect
YOLOv8n-psa summary: 214 layers, 76,839,961 parameters, 76,839,945 gradients

Freezing layer 'model.23.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/y
AMP: checks passed ✅
train: Fast image access ✅ (ping: 0.0±0.0 ms, read: 812.0±279.6 MB/s, size
train: Scanning /content/datasets/BrainTumor/BrainTumorYolov11/train/labels
train: New cache created: /content/datasets/BrainTumor/BrainTumorYolov11/tra
WARNING ⚠️ Box and segment counts should be equal, but got len(segments) =
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_lir
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 483.1±212.2 MB/s, size:
val: Scanning /content/datasets/BrainTumor/BrainTumorYolov11/valid/labels..
val: New cache created: /content/datasets/BrainTumor/BrainTumorYolov11/valid
Plotting labels to /content/ultralytics/runs/detect/YOLOv8_PSA/labels.jpg..
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937'
optimizer: AdamW(lr=0.001429, momentum=0.9) with parameter groups 97 weight
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to /content/ultralytics/runs/detect/YOLOv8_PSA
Starting training for 50 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Siz
       1/50      10.1G      3.032      4.629      3.734         31        64
```