Multi-Modal 2D+3D Semantic Segmentation for UXB Dataset

Python Script: `train_multimodal_fusion_ce.py`

July 18, 2025

Overview

This script performs joint training of a **2D+3D semantic segmentation model** on the **UXB dataset**, which includes aerial 2D TPI images and 3D LAS point-clouds. The training pipeline incorporates class-weighted losses, early fusion in the 3D space, and a custom sampler to balance minority class representation.

1 Dataset Structure

- Each sample is stored under `pair_XXXX/` directories.
- Each pair contains:
  - 2D image: `tpi_100m_*.png`
  - 2D mask: `uxb_msk_pl_st_*.png`
  - 3D point-cloud: `*.las`

2 Model Architecture

- **2D branch:**
  - Backbone: SegFormer-B0 (`nvidia/segformer-b0-finetuned-ade-512-512`)
  - Head: 1x1 convolution for segmentation
- **3D branch:**
  - Backbone: MONAI Swin UNETR-large (`feature_size=96`)
- **Fusion:**
  - Feature fusion: mean spatial encoding of 2D features expanded across 3D volume
  - Fusion layers: Conv3D → ReLU → Conv3D (output NUM_CLASSES)

3 Loss Functions

Both branches use class-weighted Cross Entropy Loss:

$$\text{Class Weights (2D/3D)} = [0.05, 0.45, 0.50]$$

Heavier weight is placed on minority classes (Plazuela and Structure).

4 Training Schedule

- Warm-up: 2-stage schedule for head and encoder learning rates
- Epochs: up to 100
- Early stopping: after 10 epochs with no validation improvement
- Optimizer: AdamW with weight decay
- Batch size: 1 (due to high memory from 3D volumes)

Learning Rate Schedule

- `head_lr`: linearly increases from `LR_WARMUP` to `LR_MAIN`
- `enc_lr`: frozen at 0 during warm-up; then gradually increases

5 Sampling Strategy

BalancedSampler ensures ∼70% of training samples contain minority class voxels (label 1 or 2 in 3D). It computes voxel distributions during initialization.

6 Metrics

Evaluated using:

- Global accuracy
- Per-class accuracy
- Precision, Recall, F1-score
- IoU per class

7 Evaluation & Output

- **Best Weights:** saved to `FusedSwinCrossEntropy_tpi_clr_swin_best.pth`
- **Inference Output:**
  - `pred_XXXX.npy`: predicted voxel grid
  - `pred_XXXX.las`: LAS file with updated `classification` field

8 Key Implementation Modules

Dataset

- Processes both 2D and 3D data
- Converts LAS point-clouds into voxel grids using spatial normalization

- Remaps class labels using:

```
LABEL_MAP_2D = {0: 0, 76: 1, 150: 2}
LABEL_MAP_3D = {2: 0, 27: 1, 6: 2}
```

Model

The model implements:

- SegFormer 2D segmentation head
- Swin UNETR 3D segmentation
- Feature fusion between 2D latent space and 3D volume features

Training Loop

Each epoch:

- Adjusts learning rates per schedule
- Unfreezes encoder weights after warm-up
- Computes combined 2D + 3D cross-entropy loss
- Saves best model based on validation loss
- Triggers early stopping if no improvement

Inference

- Loads predict split
- Generates class predictions on voxel grid
- Maps voxel predictions back to LAS file and saves

9 Usage Instructions

1. Place your dataset under: `Old_data_2d+3d/uxb_tpi_clr_with_masks_paired/`
2. Structure:

```
train/
test/
predict/
    pair_XXXX/
        *.png
        *.las
```

3. Run the script:

```
python train_multimodal_fusion_ce.py
```

4. Monitor console logs for training stats, early stopping, and inference output paths

10 Requirements

- Python 3.8+
- torch, transformers, monai, laspy, PIL, numpy

11 Conclusion

This implementation fuses 2D and 3D data for robust semantic segmentation on archaeological UXB data. With a modular architecture, balanced sampling, and fine-grained metric tracking, it offers a reproducible baseline for multimodal fusion in geospatial analysis.