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

The multi-modal semantic segmentation model integrates 2D aerial imagery and 3D point-cloud data through an **early fusion strategy in the 3D spatial domain**. This architecture comprises distinct 2D and 3D processing branches, followed by a dedicated fusion block that combines their respective latent representations.

The **2D branch** leverages a **SegFormer-B0 backbone**, pre-trained on the ADE20K dataset, to extract rich contextual features from the aerial TPI images. The SegFormer model, based on a transformer encoder, generates multi-scale feature maps from the input image $I_{2D} \in \mathbb{R}^{H_{2D} \times W_{2D} \times 3}$. These features are then passed through a **1x1 convolutional head** to produce a latent 2D feature representation, denoted as $F_{2D} \in \mathbb{R}^{C_{2D} \times H'_{2D} \times W'_{2D}}$. This representation captures semantic information from the visual cues present in the 2D imagery.

Concurrently, the **3D branch** employs a **MONAI Swin UNETR-large backbone**, a transformer-based U-Net architecture, to process the 3D LAS point-clouds after their conversion into a voxelized grid $V_{3D} \in \mathbb{R}^{D_{3D} \times H_{3D} \times W_{3D} \times C_{3D}}$. The Swin UNETR efficiently extracts hierarchical spatial features from the volumetric data, producing a high-dimensional 3D feature tensor $F_{3D} \in \mathbb{R}^{C_{3D} \times D'_{3D} \times H'_{3D} \times W'_{3D}}$. The choice of Swin UNETR is motivated by its ability to capture long-range dependencies and local information within the volumetric data, crucial for accurate 3D semantic understanding.

The core of the fusion model lies in the **integration of these disparate modalities**. The 2D features, F_{2D}, are first spatially encoded and then expanded across the depth dimension of the 3D volume. This operation can be conceptualized as projecting the 2D semantic understanding into the 3D voxel space, effectively augmenting the 3D features with contextual information derived from the aerial view. Let MeanSpatialPooling$(\cdot)$ denote an operation that reduces the spatial dimensions of F_{2D} to a global descriptor, which is then broadcast or tiled to match the spatial and depth dimensions of the 3D feature space. The fused feature F_{fused} is then obtained by concatenating or summing these expanded 2D features with the 3D features, potentially after alignment:

$$F_{\text{fusion\_input}} = \text{Concatenate}(\text{Expand}(F_{2D}), F_{3D})$$

This concatenated feature tensor is then fed into a series of **3D convolutional layers** for final fusion and classification. Specifically, the fusion layers consist of Conv3D $\rightarrow$ ReLU $\rightarrow$ Conv3D, which ultimately output the predicted class probabilities for each voxel in the 3D volume, mapping the fused features to the NUM_CLASSES output channels. The final segmentation map $S_{3D} \in \mathbb{R}^{D'_{3D} \times H'_{3D} \times W'_{3D} \times \text{NUM\_CLASSES}}$ is then derived from these probabilistic outputs. This early fusion approach allows the model to leverage the complementary strengths of both 2D and 3D data from the initial stages of feature learning, leading to a more robust and comprehensive semantic understanding of the environment.

Model Architecture Summary (from original input for quick reference)

- **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 $\rightarrow$ ReLU $\rightarrow$ 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 approximately 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

- 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.