AIRS MEDICAL

[SwiftSight] AI Research Scientist Assignment

I. Assignment Topic

Mini-SynthSeg for Lumbar Spine Segmentation

II. Assignment Description

As an AI Research Scientist at SwiftSight, you need to design a minimal CPU-viable problem and implement a simplified version of the SynthSeg approach for robust lumbar spine segmentation. The goal is to train a segmentation model using synthetic MRI data that generalizes better than traditional supervised learning. Please submit a ZIP file containing working code and documentation.

Your submission must include:

1. **Working Implementation**
    - Python script(s) implementing synthetic MRI generation from SPIDER spine label maps
    - Two trained models: one using synthetic data, one using real data
    - Evaluation pipeline comparing robustness
    - One example script demonstrating the full pipeline
2. **README.md** containing:
    - **Setup instructions**: How to install and run
    - **Usage examples**: How to generate synthetic data and train models
    - **AI tool usage notes**: Which tools you used and how (see guidelines below)
3. **ANALYSIS_REPORT.md** containing:
    - **Background Research**: What is SynthSeg? Why does synthetic training improve generalization?
    - **Minimal Problem Design**: How you defined the task (e.g., simplified class labels, model size, data size) and why
    - **Implementation Details**: Your synthetic data generation strategy
    - **Experimental Results**: Comparison between synthetic vs real training
    - **Analysis**: Why does synthetic training work (or not) for spine segmentation?

III. AI Tool Usage Guidelines

We strongly encourage you to use AI tools (ChatGPT, Claude, Copilot, etc.) for any part of this assignment.

In your README, briefly document: - Which AI tools you used - 2-3 example prompts that were particularly helpful - One instance where you disagreed with or had to correct an AI suggestion

IV. Technical Requirements

Suggested Structure (flexible):

```
synthseg_spine_assignment/
├── README.md                  # Setup, usage, AI usage
├── ANALYSIS_REPORT.md        # Background, results, analysis
├── requirements.txt           # Dependencies
├── synthetic_generator.py     # Label-to-image generation
├── train.py                   # Training script for both models
├── evaluate.py                # Robustness evaluation
├── example.py                 # Demo script
├── models/
│   ├── unet.py                # Simple U-Net architecture
│   └── saved/                 # Trained model checkpoints
├── data/
│   ├── SYNTH_T1_SEG/          # Generated synthetic MRI from T1 labels
│   ├── SPIDER_T1_train/       # Real T1 MRI for training (subset)
│   └── SPIDER_T2_val/         # Real T2 MRI for evaluation (subset)
└── results/
    └── evaluation_results/    # Performance metrics
```

Core Features Required:

1. **Synthetic Data Generation**
    - Generate synthetic spine MRI from label maps
    - Implement intensity sampling for different tissues (vertebrae, discs, spinal canal)
    - Add at least ONE realistic artifact (bias field, noise, or motion)
    - Support contrast variation (T1-like to T2-like spectrum)
    - Resolution variation (1-4mm)
2. **Model Training**
    - Train Model A: Using ONLY synthetic data (no real images)
    - Train Model B: Using real SPIDER images (baseline)
    - Same architecture for fair comparison
    - Log training metrics
3. **Evaluation**
    - Standard test on T2 validation images
    - Contrast shift test (e.g., test on T2 when trained on T1-like)
    - Resolution degradation test
    - Report Dice scores per structure (vertebrae, disc, canal)
4. **Open Research Questions**: Propose and explore your own investigation into synthetic training. (For inspiration, not selection: reality gap tuning, pathology handling, feature visualization, failure modes...)

Dataset: SPIDER lumbar spine MRI (open-source). Please select a subset of the data for your experiment for proof of concept (e.g., ~40 samples).

V. Implementation Details

Part 1: Synthetic Generator

Implement `SpineSynthGenerator` class with:

```
def generate_synthetic_mri(self, label_map):
    """
    1. Sample tissue intensities based on contrast type
    2. Add one artifact (bias field, motion, or noise)
    3. Apply resolution variation
    """
    return synthetic_image
```

Part 2: Model Training

- Use provided 2D U-Net (`models/unet.py`) or modify as needed
- Training loop with Dice loss
- Model A: Synthetic-only training (from T1 labels)
- Model B: Real T1-only training

Part 3: Evaluation

Test on T2 validation set with metrics: - Per-structure Dice coefficient - Robustness to contrast shifts - Robustness to resolution changes

Part 4: Open Research Question

Propose and investigate your own research questions (see Core Features #4).

VI. FAQ

Q. Do I need to implement my own model? A. No, we provide a 2D U-Net in `models/unet.py`. Feel free to modify as you needed.

Q. Do I need a GPU? A. No, the provided U-Net is lightweight and all tasks are CPU-runnable.

Q. What if my synthetic model performs worse? A. That's totally fine! Analyze why and document your findings for potential improvement strategies.

Q. Can I simplify the task? A. Yes! Feel free to simplify the problem to make it more tractable (e.g., fewer structures labels). Document your simplifications and justify why they help demonstrate the core SynthSeg concept.

Q. How much data do I need? A. For the proof of concept, download a small desired subset from the open-source SPIDER challenge dataset.

Q. How much time should this take? A. Aim for ~8 hours of work. Again, we strongly recommend you to work with AI tools. Focus on core concepts over perfect performance.

VII. Evaluation Points

- Does the problem clearly defined and compact?
- Is the implementation correct and well-documented?
- Are the results properly evaluated and interpreted?
- Is there clear understanding of domain randomization principles?
- Is the research question well-defined and investigated?

VIII. Submission

- **Timeline:** One week (extensible - just let us know)
- **Format:** ZIP file named `[YourName]_SynthSegSpine.zip`
- **Contact:** For questions, email [jung.woojin@airsmed.com]

9. References & Resources

References:

- SynthSeg: Billot et al., "Segmentation of brain MRI scans of any contrast and resolution without retraining" (2023) - https://arxiv.org/abs/2107.09559

- SPIDER dataset: van der Graaf et al., "Lumbar spine segmentation in MR images: a dataset and a public benchmark" (2024) - https://arxiv.org/abs/2306.12217
- SPIDER Challenge & Dataset: https://spider.grand-challenge.org/

Provided: - 2D U-Net implementation (`models/unet.py`)

Note: This is a simplified version of SynthSeg. Focus on understanding the core concept of synthetic training for domain generalization rather than implementing all features of the original paper.