Assignment Computer Vision Research Engineer

Assignment Title: Modular Vision-Based Multi-task Learning for Eye Disease Diagnosis

---

Objective:

Design and implement a modular deep learning system for multi-task learning in ophthalmology, aimed at efficiently handling multiple tasks such as classification and segmentation from retinal images. This assignment focuses on leveraging dynamic task-specific routing within a unified vision model to simulate intelligent task-aware learning in a medical imaging context.

Tasks:

1. **Model Architecture Implementation**:
    - Build a modular vision model with shared base features and multiple specialized processing units ("experts").
    - Implement a dynamic routing mechanism that selects appropriate expert modules based on input data characteristics.
    - Ensure that the model supports at least two tasks concurrently (e.g., disease grading classification and lesion segmentation).
2. **Dataset and Preprocessing**:
    - Use the **IDRiD (Indian Diabetic Retinopathy Image Dataset)** for training and evaluation.
        - Task 1: Disease Grading (multi-class classification).
        - Task 2: Lesion Segmentation (binary or multi-label segmentation).
    - Perform necessary preprocessing, including resizing, normalization, and augmentation (if applicable).

   - Prepare the dataset by mapping each image to its corresponding label and segmentation mask.

3. **Training Pipeline Setup**:
   - Implement the training loop supporting both tasks with appropriate loss functions (e.g., CrossEntropy for classification, BCEWithLogits for segmentation).
   - Use a combined loss function that weights each task’s loss to ensure balanced optimization.
   - Integrate logging of training loss and task-specific performance metrics.
4. **Model Training and Evaluation**:
   - Train the model on the IDRiD dataset for a fixed number of epochs.
   - Evaluate classification accuracy and segmentation performance (e.g., Dice score or IoU).
   - Include both quantitative results and visual outputs (e.g., predicted masks overlaid on original images).
5. **Performance Comparison**:
   - Compare the multitask modular model's performance with a single-task baseline model (either classification-only or segmentation-only).
   - Analyze trade-offs in accuracy, efficiency, and generalizability.

Deliverables:

- Complete Python code implementing the modular multi-task learning pipeline.
- A short report (~2 pages) covering:
  - Model design and justification
  - Dataset handling and preprocessing steps
  - Training and evaluation strategy
  - Performance metrics and analysis
  - Visualizations of results

Additional Notes:

- Use PyTorch and standard libraries for implementation.
- Maintain modularity in code for easy extensibility (e.g., adding more tasks).

- Carefully handle overfitting using techniques such as dropout, data augmentation, or early stopping.
- Ensure reproducibility by fixing random seeds and organizing code cleanly.