DA312: Advanced Topics in Machine Learning - Lab Assignment 7

April 1, 2024

Submission Guidelines

1. Implement a UNet architecture from scratch using **PyTorch**.
2. Ensure your code includes comprehensive comments and explanations.
3. Upload your complete Google Colab notebook to the designated Teams assignment portal.

Dataset Description

The dataset comprises Fundus photographs, capturing the interior surface of the eye including the retina, optic disc, optic cup, and vasculature, which are essential for diagnosing and managing ocular diseases.

- Each image in the dataset is accompanied by Ground Truth annotations for two distinct features: the optic disc and the optic cup.
- The dataset also includes an Excel file that categorizes each image as either Normal or Glaucoma (disease).
- Access to the dataset is provided through the following hyperlink: Fundus Image Dataset.

Task 1: Segmentation with UNet

- Develop a PyTorch model for the simultaneous segmentation of the optic cup and optic disc. The UNet architecture is shown in Figure 1.
- Employ the Dice Coefficient as the segmentation loss function, defined by:

$$Seg_{\text{loss}} = 1 - \frac{1 + 2 \times P_s \times Y_s}{1 + 2 + P_s + Y_s} \tag{1}$$

  where P_s denotes the predicted segmentation result, while Y_s denotes the actual segmented result.

Task 2: Glaucoma Classification

- Extend your Task 1 model to classify the images as Normal or indicative of Glaucoma. Refer to Figure 2 for the extended architecture.
- Use Multitask loss as discussed in the class.

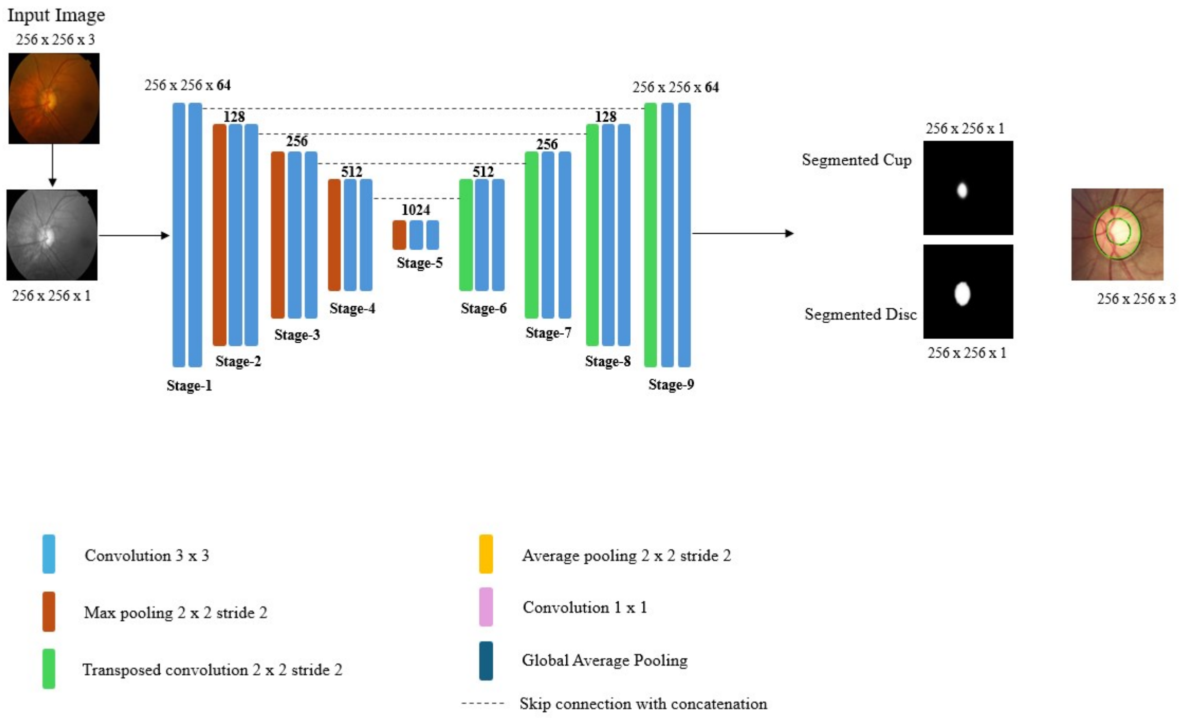

Figure 1: Joint Segmentation using UNet

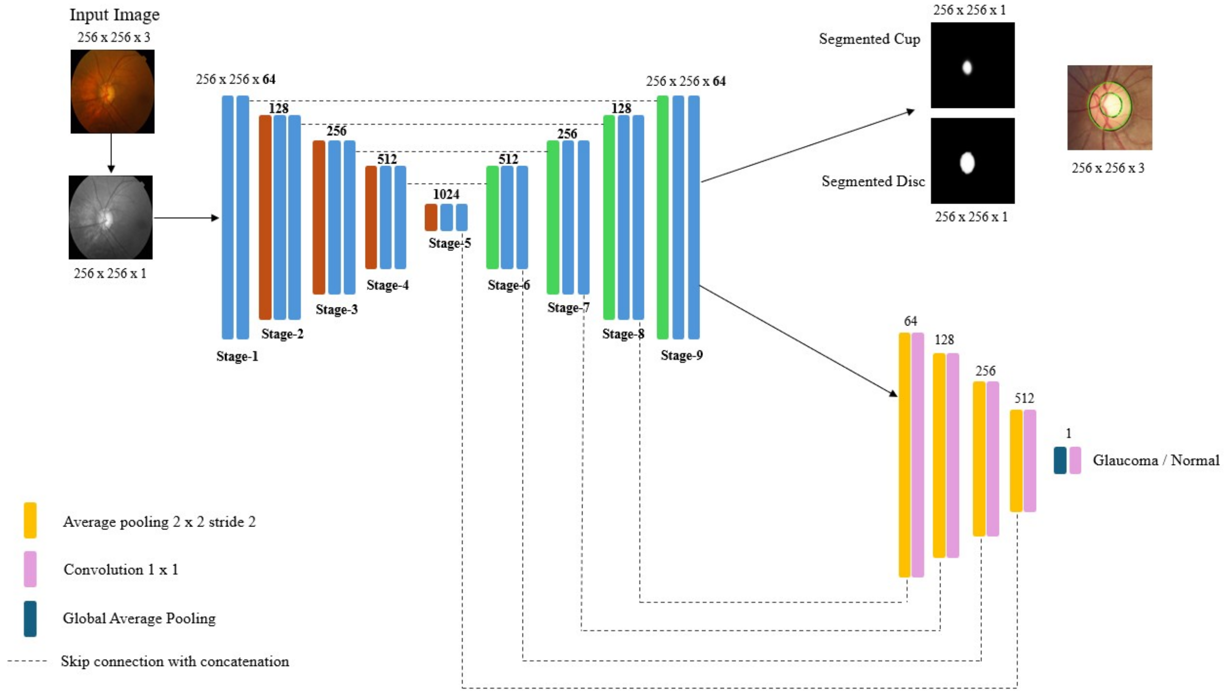

Figure 2: Classification in conjunction with Segmentation

Evaluation Metrics for Task 1 and Task 2

Evaluate your model on the following criteria:

1. Visualize feature maps for all nine stages (stage-1 to stage-9) of a test image.
2. Store segmentation outcomes for the entire set of test images.
3. Graph the trajectory of classification loss and compile a confusion matrix.
4. Compute the following metrics using PyTorch's built-in functions: Jaccard Similarity Index, Dice Coefficient, Accuracy, Precision, Recall, Specificity, F1-Score, Intersection over Union (IoU), Structural Similarity Index (SSIM), and Area Under the ROC Curve (AUC).