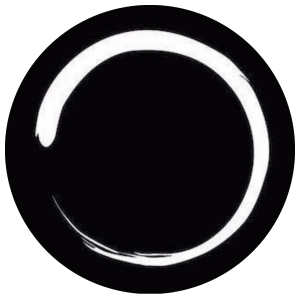
inzva DLSG

Homework 2



Coding Assignment

The main objective of this assignment is to train and evaluate a U-Net model for binary image segmentation, particularly focused on medical images related to stomach segmentation. The U-Net architecture is well-suited for medical image analysis, offering precise localization through an encoder-decoder structure with skip connections. The assignment consists of separate scripts for training and testing, along with utility modules for data handling, visualization, and metric computation.

The data provided is from a public dataset used in a Kaggle competition called UW-Madison GI Tract Image Segmentation. However, there is no need to interact with Kaggle or the competition itself. The dataset has already been processed in a way that is suitable for this specific task.

Detailed Task Description

1. Model Architecture

- As you know from our sessions, U-Net is a popular deep learning architecture for segmentation, featuring encoder and decoder paths with skip connections.
- In this assignment you will implement a U-Net model for image segmentation, defined in `model/unet.py`.

2. Training Script (`train.py`)

Functionality:

- Loads and prepares the dataset using the custom `MadisonStomach` class, which manages medical image data and segmentation masks.
- Initializes the model, optimizer, and loss function.
- Uses the `train_model` function to train the model over several epochs, compute training and validation losses, and save model checkpoints periodically.
- Visualizes the training and validation loss curves, as well as the Dice coefficient curve, to monitor performance.

3. Testing Script (`test.py`)

Functionality:

- Loads the trained U-Net model and prepares the test dataset. (Choose the model with last epoch you saved.)
- Computes segmentation predictions for the test set and calculates performance metrics such as the Dice coefficient.
- Visualizes predictions and saves the results for analysis.

4. Utilities (`utils` Directory)

- `viz_utils.py`: Contains functions for visualizing predictions and plotting loss/-metric curves.
- `metric_utils.py`: Defines functions for computing performance metrics, such as the Dice coefficient.
- `model_utils.py`: Provides utility functions like argument parsers for training/testing configuration and functions for reproducibility.
- `data_utils.py`: Defines the `MadisonStomach` dataset class, handling loading, preprocessing, and augmenting medical image data and masks.

Goals and Expected Outputs

- **Train the Model:** Successfully train the U-Net model to accurately segment medical images.
- **Evaluate the Model:** Assess performance using the test dataset, focusing on the Dice coefficient.
- **Visualize Results:** Generate and save segmentation results and loss curves to analyze model performance.
- **Save Checkpoints:** Periodically save model checkpoints for easy reloading and further experimentation.
- **Dice Coefficient:** Achieve a Dice coefficient of 0.7 or higher on both the validation and test sets.

Additional Lookups

- U-Net paper.
- Custom datasets in Pytorch
- You can read about how to use an argument parser and understand why it is useful. One suggestion is here

Delivery Instructions

- Rename the HW2 folder to your name_surname and zip your code and upload it.
- Delete all trained models except the one you tested and found to work best.
- Do not include the dataset folder in the zip file.

Visualization Examples