Homework 6
601.482/682 Deep Learning
Fall 2025

October 16, 2025

Due 11:59pm on Nov 5, 2024

Please submit 1) a single zip file containing your Jupyter Notebook and PDF of your Jupyter Notebook to "Homework 6 - Notebook" and 2) your written report (LaTeX generated PDF) to "Homework 6 - Report" on Gradescope (Entry Code: VWJRB3)

Important: You must program this homework using the PyTorch framework. We highly recommend using Google Colaboratory.

Important: If you don't have local GPU access, you should port the provided Python scripts to Colaboratory and enable GPU in that environment (under Edit->Notebook Settings). Training should converge in less than 30 minutes for T4 GPU. If your model does not make significant updates in that time, you should re-examine your code. Either way, this is a reminder to start the assignment early.

1. *Data Augmentation.* In this problem, you will attempt the 2024 SegSTRONG-C challenge.[1] You are given a pre-processed dataset of endoscopic frame images. The goal is to train a network that takes each RGB frame as an input and predicts a binary pixel-wise segmentation mask that labels the target robot tool. Additionally, we provided test data with non-adversarial corruption (blood). Data augmentation is the most common approach to increase your model's robustness against this type of corruption.

   Data Folder We have provided a well-structured dataset shown below. It consists of `train`, `val`, and `test` directories where `train` has 5 sequences under 2 configurations, `val` contains 1 sequence under 1 configuration, and `test` contains 1 sequence under 1 configuration.

```
SegSTRONGC_MLDL/
├── train/
│   ├── 3/0/
│   ├── 3/2/
│   ├── 4/0/
│   ├── 4/2/
│   ├── 5/0/
│   ├── 5/2/
│   ├── 7/0/
│   ├── 7/1/
│   ├── 8/1/
│   └── 8/2/
├── val/
│   └── 1/0/
└── test/
    └── 9/0/
```

[1] https://segstrongc.cs.jhu.edu/

Under each sequence in `train` and `val`, there are 'regular' and 'ground_truth' folders that contain only the 'left' folder, each containing 300 frames collected from the left camera on a stereo endoscope at 10 fps. Besides the 'regular' and 'ground_truth' folders, each sequence in the `test` folders contains the 'blood' folder which contains the non-adversarially corrupted version of the sequence.

The **main goals** of the homework are as follows. Concrete TODOs are enumerated on the next page.

- *Complete the Network structure for segmentation task.* The network structure we provide is a simplified U-Net, which is a very popular framework in medical image segmentation tasks. Read and understand the code in the notebook, the implementation is missing the last layer, the last activation function, and the forward function. Next, fill in the missing components for 1(a). For the segmentation task, train ONLY with the frames in '/train' and validate and test with the regular frames in '/val' and '/test' respectively. The original input image is a $270 \times 480 \times 3$ RGB image. The ground truth label is a grey-scale image that has the same dimension, where '255' indicates the robot tool type and '0' indicates background tissue.
- *Test the network under non-adversarially corrupted images.* Test our network trained on regular images on non-adversarially corrupted blood images.
- *Explore data augmentation to increase the model robustness.* Test our network trained on regular images on non-adversarially corrupted blood images.

Now that you know the broad goals and data sets, please complete the following TODOs.

(a) Train a segmentation network using the frames in the `train` folder. Please write from scratch a DICE loss function as your network loss[2]. Please train the network until convergence (should take under 30 min for 10 epochs) using the default provided hyperparameters and provide a figure of training loss and validation loss w.r.t. epochs (in a single figure). Please report your performance (DICE score) on both the regular and blood test datasets. (You will expect a DICE score over 0.8 for the regular test dataset if your implementation is reasonable.)

(b) Is your model's performance on blood images as good as the performance on regular images? Please explain why.

(c) Introduce meaningful data augmentation and train the network until convergence using the same hyperparameters as (a). Please plot the training loss and validation loss on a single figure again and report test dataset performance on both the regular and blood test datasets.

(d) Does the data augmentation improve the model's performance on the blood test datasets? Please explain why.

2. *Transfer Learning.* Please download the fashion MNIST dataset [3] as used in HW4 and download the VGG16 model (`https://pytorch.org/docs/stable/torchvision/models.html`).

(a) Randomly initialize all parameters in VGG16 and try to train your model to learn the Fashion MNIST classification task. What's the accuracy you achieve? Please report your test accuracy on the test dataset. You should expect an accuracy $> 85\%$.

(b) Load the pre-trained VGG16 model from torch vision models. Freeze all but the last layer: randomly initialize the last layer of your network and fine-tune this. What accuracy do you get now? Please again report your test accuracy on the test dataset. You should expect an accuracy $> 60\%$.

(c) Now, imagine a scenario in which you want to train the VGG16 model on an entirely new dataset and will fine-tune either the model from (2a) or (2b). Which pre-trained model is the preferred starting point for your new use case?

[2] Read more about the DICE score: `https://medium.com/datadriveninvestor/deep-learning-in-medical-imaging-3c1008431aaf`

[3] The full FashionMNIST dataset can be downloaded from the official website here: `https://github.com/zalandoresearch/fashion-mnist`