ELEC/COMP 447/546
Assignment 5
Due: Apr. 9, 2023, Midnight

Problem 1: Semantic Segmentation (7 points)

In this problem, you will train a simple semantic segmentation network. Recall that in semantic segmentation, the algorithm must assign each pixel of an input image to one of *K* object classes. We have provided you with a Colab notebook with skeleton code to get you started.

We will use a portion of the CityScapes dataset for this problem, consisting of 2975 training images and 500 validation images. The second cell in the notebook will automatically download the dataset into your local Colab environment.

Each image also comes with annotations for 34 object classes in the form of a segmentation image (with suffix 'labelIds.png'). The segmentation image contains integer ids in [0, 33] indicating the class of each pixel. This page provides the mappings from id to label name.

a. Fill in the `init` and `forward` functions for the `Segmenter` class, which will implement your segmentation network. The network will be a convolutional encoder-decoder. The encoder will consist of the first several 'blocks' of layers extracted from the VGG16 network pretrained on ImageNet (the provided Colab notebook extracts these layers for you). You must implement the decoder with this form:

Layer	**Output channels for Conv**
3 x 3 Conv + ReLU	64
Upsample (2 x 2)	X
3 x 3 Conv + ReLU	64
Upsample (2 x 2)	X
3 x 3 Conv + ReLU	64

Upsample (2 x 2)	X
3 x 3 Conv	n_classes (input to `init`)

Use PyTorch's Upsample function. Remember that the size of the image should not change after each Conv operation (add appropriate padding).

b. Train your model for 7 epochs using the nn.CrossEntropy loss function. Using the GPU, this should take about 30 minutes.

c. Using the final model, report the average intersection-over-union (IoU) per class on the validation set in a table. For more on IoU, see this page. Which class has the best IoU, and which has the worst? Comment on why you think certain classes have better accuracies than others, and what factors may cause those differences.

d. For each of the following validation images, show three images side-by-side: the image, the ground truth segmentation, and your predicted segmentation. The segmentation images should be in color, with each class represented by a different color.
   i. frankfurt_000000_015389_leftImg8bit.jpg
   ii. frankfurt_000001_057954_leftImg8bit.jpg
   iii. lindau_000037_000019_leftImg8bit.jpg
   iv. munster_000173_000019_leftImg8bit.jpg

e. Look at the lines of code for resizing the images and masks to 256 x 256. We use bilinear interpolation when resizing the image, but nearest neighbor interpolation when resizing the mask. Why do we not use bilinear interpolation for the mask?

f. Look at the `__getitem__` function for the CityScapesDataset class and notice that we apply a horizontal flip augmentation to the image and mask using a random number generator. Why do we apply the flip in this way instead of simply adding `T.RandomHorizontalFlip` to the sequence of transforms in `im_transform` and `mask_transform` (similar to what you did in Homework 4)?

Submission Instructions

All code must be written using Google Colab (see course website). Every student must submit a zip file for this assignment in Canvas with 2 items:

1. An organized report submitted as a PDF document. The report should contain all image results (intermediate and final), and answer any questions asked in this document. It should also contain any issues (problems encountered, surprises) you may have found as you solved the problems. **Please add a caption for every image specifying what problem number it is addressing and what it is showing**. The heading of the PDF file should contain:
    1. Your name and Net ID.
    2. Names of anyone you collaborated with on this assignment.
    3. A link to your Colab notebook (remember to change permissions on your notebook to allow viewers).
2. A pdf copy of your Colab notebook.

Collaboration Policy

I encourage collaboration both inside and outside class. You may talk to other students for general ideas and concepts, but you should write your own code, answer questions independently, and submit your own work.

Plagiarism

Plagiarism of any form will not be tolerated. You are expected to credit all sources explicitly. If you have any doubts regarding what is and is not plagiarism, talk to me.