This assignment does not count toward the final grade.

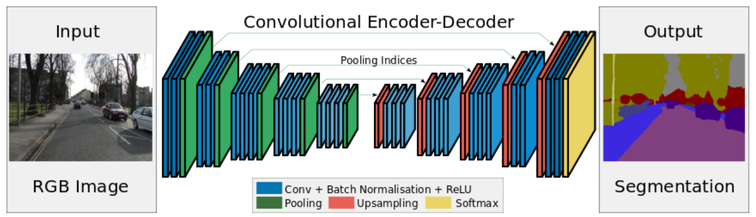
Bonus HW

New Attempt

Due Jun 7 by 10am **Points** 100 **Submitting** a file upload

Submit your Bonus HW on Canvas before 10am on June 7. Late submissions will not be graded.

This bonus homework is about semantic image segmentation by labeling all pixels of a given image with object classes. Your task is to train and evaluate a recent deep learning approach to semantic segmentation, called **SegNet** (https://mi.eng.cam.ac.uk/projects/segnet/).



SegNet architecture by Vijay Badrinarayanan, Alex Kendall, Ankur Handa and Roberto Cipolla.

SegNet has a convolutional encoder- decoder architecture. The encoder part has the 13 convolutional layers of the **VGG16 network** (https://neurohive.io/en/popular-networks/vgg16/) with a key difference that the encoder keeps the record of pooling indices computed in the max-pooling step. The corresponding decoder network upsamples its lower resolution input feature map(s), where the upsampling is based on the recorded max-pooling indices of the encoder. This removes the need for learning how to up-sample. The initial sparse up-sampled feature maps are convolved with trainable filters to produce dense feature maps. One of the main advantages of SegNet relative to other competing architectures is that it has significantly fewer parameters.

Task:

Use the provided skeleton pytorch code **pySegnet.zip** (https://canvas.oregonstate.edu/courses/1811612/files/85435566/download?download_frd=1) to train and validate SegNet on **VOC2012 training and validation images** (http://host.robots.ox.ac.uk/pascal/VOC/voc2012/). The VOC2012 dataset has been downloaded for you, and placed in the "\scratch" folder on the server. The skeleton code already specifies the right path to VOC2012 images in the "\scratch" folder.

- Specify your ONID user name in the script file "train.sh" for training SegNet on the training images.
- Specify your ONID user name in the script file "inference.sh" when evaluating your SegNet on the validation images.
- For training, specify the following hyper-parameters in file "train.py":
  - NUM_EPOCHS -- You should run training on a minimum of 2000 epochs, whereas some practitioners reported training on 6000 epochs. Note that it may take longer than 8 hours for training on 1000 epochs on our GPU server. So you need to estimate the right trade off between complexity and accuracy that works for you.
  - LEARNING_RATE -- Consider a few relatively small learning rates (e.g., 1e-1), and estimate on the validation data which learning rate gives the best result.
  - BATCH_SIZE -- Consider several different batch sizes (e.g., 8, 16, 32), and estimate on the validation data which batch size gives the best result.

What to turn in:

1. (50 points) model_best.pth -- file with parameters of your **best** SegNet model that can be found in folder "/scratch/ONID_username/pySegNet/saved_models/". Note that this file is very large and cannot be submitted via TEACH. Please upload your file to the Google drive (each OSU student has one with their OSU account), and provide the corresponding link to your file in "report.pdf".
2. (50 points) report.pdf -- a brief report of your training and validation:
   - (15 points) Two plots of your training error over 2000+ epochs for two different learning rates and the best batch size.
   - (15 points) Two plots of your training error over 2000+ epochs for two different batch sizes and the best learning rate.
   - (10 points) Your estimates of the best NUM_EPOCHS, LEARNING_RATE, and BATCH_SIZE.
   - (10 points) Figure with 10 example validation images and their corresponding semantic segmentations produced by your best SegNet model.