Deep Learning (IME/UERJ) – 2023.1
Final Assignment - Semantic Segmentation

1. Objectives

- Build and evaluate two CNN models for semantic segmentation using Google Colab.
- Most of the code for pre- and post-processing de input data is already in the Google Colab notebook provided with this task.
- You should implement at least two fully convolutional CNN models. I suggest picking two of the following: U-Net, Deeplabv3+, ParseNet, SegNet, and PSPNet. But you can use any other models, including a custom one (designed by you).
- The final objective is to perform semantic segmentation on images from the ISPRS 2D Semantic Labeling Contest (Vaihingen city data): https://www2.isprs.org/commissions/comm2/wg4/benchmark/semantic-labeling/
- The images are available at:
  https://github.com/gilson-costa/files/raw/main/Images_Semantic_Segmentatio_Assignment.part1.rar
  https://github.com/gilson-costa/files/raw/main/Images_Semantic_Segmentatio_Assignment.part2.rar
  https://github.com/gilson-costa/files/raw/main/Images_Semantic_Segmentatio_Assignment.part3.rar
- They consist in three orthophotos, with 9 cm spatial resolution, which channels correspond to the near infrared, red and green bands delivered by the camera. Each orthophoto comes together with a Digital Surface Model (DSM), in which each pixel has height information (in pixels); and a reference (ground truth) image.
- The images are named:
  - *Test_Image.tif; Test_DSM.tif; Test_Reference.tif;*
  - *Train1_Image.tif; Train1_DSM.tif; Train1_Reference.tif;*
  - *Train2_Image.tif; Train2_DSM.tif; Train2_Reference.tif.*
- The first triplet should be used for evaluating the CNN, and the other two triplets for training the network.
- Be sure to normalize both the images (orthophotos) and DSMs before using them (the normalization procedure in Section 3 (Loading the data) must be changed to process the DSM data (which has only one channel).

2. Experimental Protocol

Creating the CNN model

- Start with the code (Google Colab notebook) provided with this task.
- You must include in Section 8 (Building the CNN model) the structure/layers of the selected CNNs.

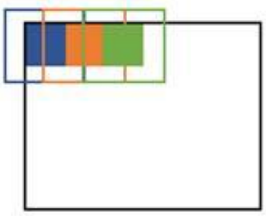
- If you decide to use the U-Net, you can start with the structure given in the Colab notebook (in the output of the last cell of Section 8), but you must change it to use Transpose Convolutions in the decoder section of the network.

Training the CNN model

- In order to tune the training process to obtain better performances, you can alter the training parameter values set in Section 10 (Defining hyperparameters), specially: the number of epochs, class weights, data augmentation, optimizer and respective optimizer parameter values.
- As in the last assignment, you should also change the data augmentation parameters in Section 7 (Data augmentation).
- Change the data augmentation procedure to generate new patches by flipping and rotating (by multiples of 90 degrees) the original training patches, using the functions of tf.data.Dataset.
- Observe that you should change the code of Section 11 (Start training) to train with the whole training set.

Evaluation of the CNN model

- You should train and evaluate the both CNN models using the following combinations (which will lead to four experiments):
  a) Use only one training image (Train1); and use two training images (Train1 and Train2) for training the network;
  b) For each of the alternatives in (a): use only the orthophotos as inputs to the network; and use the orthophotos + DSMs as inputs. When using the orthophoto + DSM, concatenate the three orthophoto channels with the DSM channel, to obtain a 4-channel raster, which should be used as input to the models. Observe that this should be done both for training and testing the network.
- Both in training and testing, you must extract patches with overlap.
- You must evaluate/test of the CNNs over the test patches (Section 15. Prediction over test patches) and over the reconstruction of the whole prediction image.
- The reconstruction of the prediction image is done by mosaicking the prediction of the individual test patches (16. Mosaic of the test patches predictions). The mosaic considers only the center of the test patches predictions.



- Accordingly, the report must present the classification results as label images, and report accuracy metrics (overall and average class accuracies, F1-score) for each patch size.

- Accuracies in the order of 80% or more (overall accuracies) over the test set are expected.

3. Report

The final report must contain:

1) A link to the notebook containing your code.
2) A pdf file reporting your experiments, the results and a discussion of the results.