Final Project Challenge - Microscopy Cell Segmentation

Assaf Arbelle, Tammy Riklin Raviv

Deep Learning and Applications Course 2018

1 Introduction

This challenge checks the results on the network you trained. We supply two new images without their segmentations. The ground truth segmentations are kept by the course staff. Each group should submit a zip file, explained in the document, containing the segmentation files. We will evaluate the segmentations of all groups and publish a score board. Since all the evaluation is done automatically, we ask that you strictly follow the submission instructions. Failure to do so will result in a 0 score for the challenge and a reduction of 20% from the project grade. In order to reduce errors, we supply a script that validates the submission file. Make sure to validate the file before submission.

1.1 Data

The raw images are locate in 'Data/Test/RawImages' Please copy the merge this folder with the original Data folder supplied at the beginning of the course.

2 Running The Code

2.1 Designing Your Network

The main file for running the test script is *test_SegNet.py* The parameters are located in file *test_Params.py*. Please copy both files to the project directory The test script assumes that your network architecture is implemented in *Network.py* (located in the same directory)

2.2 Setting Parameters

Open the "*test_Params.py* "file and edit the parameters under the parameter class *TestParams*() (line 56). DON NOT CHANGE *TestParamsBase*() !! The following is a list of all the parameters:

1. *net_build_params* Parameters for your network.
2. *use_gpu* IF NO GPU AVAILABE, SET TO FALSE
3. *gpu_id* IF MORE THAN 1 GPU IS AVAILABLE, SELECT WHICH ONE
4. $load\_checkpoint_patg$ Path to checkpoint of trained network
5. *experiment_name* Name of your experiment. Will be used for directory names

2.3 Running the Code

Run the function "$tes\_SegNet.py$"The function creates a .zip file ready for submission. If for any reason you change the $test\_SegNet.py$ file or would like to create a valid zip file.

2.4 Contents of the ZIP file

Each zip file should contain only two files:

1. $seg\_calibrate2 - P01.068.TIF$
2. $seg\_calibrate2 - P01.072.TIF$

Each image should be of size $(512, 640)$, identical to the input images. The images should be of type uint8 and contain only values $[0, 1, 2]$ for classes background, foreground and edge respectively.

2.5 Checking the submission

Before submission, please validate the zip file using $CeckSubmission.py$. Run the file from comman line:
$python3 PATH\_TO\_FILE$
Replace $PATH\_TO\_FILE$ with the full path to the zip file.

2.6 Evaluation Metric

The code measures two values, (1) the global Jaccard index, (2) the mean instance Jaccard index.

For a given image $NetSeg$ and label ground image GT the foreground is calculated as the pixels where the label is of cell nucleus (1):

$$F_{net} = (x, y) : NetSeg(x, y) = 1 \tag{1}$$

$$F_{gt} = (x, y) : GT(x, y) > 0 \tag{2}$$

The global Jaccard index is defined as:

$$J = |\frac{F_{net} \cap F_{gt}}{F_{net} \cup F_{gt}}| \tag{3}$$

The mean instance Jaccard instance checks the Jaccard index of seperate cells. The image F_{net} is transfomed into a labeled image of connected components L_{net}. For each pare $l_{net} \in L_{net}$ and $l_{gt} \in GT$

$$F_{net}^{l_{net}} = (x, y) : NetSeg(x, y) = l_{net} \tag{4}$$

$$F_{gt}^{l_{gt}} = (x, y) : GT(x, y) = l_{gt} \tag{5}$$

A match is considered if:

$$|\frac{F_{net}^{l_{net}} \cap F_{gt}^{l_{gt}}}{F_{gt}^{l_{gt}}}| > \frac{1}{2} \tag{6}$$

if the cells are matched then their Jaccared index is added to the mean:

$$J = |\frac{F_{net}^{l_{net}} \cap F_{gt}^{l_{gt}}}{F_{net}^{l_{net}} \cup F_{gt}^{l_{gt}}}| \tag{7}$$

3 Report Submition Instructions

When submitting the project please submit the training code, along with a trained checkpoint. Please also submit a document clearly explaining your project. The document should include at least:

1. Names and IDs.
2. Detailed explanation of the project (i.e Network Architecture, paramterts ets)
3. Detailed explanation of the training process (Losses, regularizations, batch size, crop size, etc)
4. figures of the loss curve and examples of results (from tensorboard).
5. A summary of your work. (What you did, what worked or didn't work, analysis etc)
6. Conclusions from your work
   Please submit all the files in a zip file. ONLY ONE SUBMISSION PER GROUP. This Should be a sepperate