**Att-DCRNet with Info-WGANGP**

This repository provides Pytorch implementation for our paper “Att-DCRNet with Info-WGANGP: A Deep Learning Self-Attention Cross Residual Network with Info-WGANGP for Mitotic Cell Patterns Identification in HEp-2 Medical Images”.

This paper proposes the Att-DCRNet with Info-WGANGP for HEp-2 mitotic vs. interphase cell patterns classification. This framework is composed of two cascaded steps. The first is to generate new minority mitotic samples using Info-WGANGP to synthetically balance the skewed training dataset in order to train the downstream Att-DCRNet model.

**Dataset**

The used UQ-SNP\_HEp-2 Task-3 dataset could be downloaded from [here](https://outbox.eait.uq.edu.au/uqawilie/UQSNP_HEp2_datasets/Task%203/).

This implementation follows the Pytorch torchvision dataset library. Therefore, each class of the dataset should be separated in a class-named subfolder inside the dataset parent folder. This configuration should be followed for the training, validation, and testing sets. For example:

|  |  |  |
| --- | --- | --- |
| dataset  | | | |
| |------------- | --------------|------------ | ----------| |
| training | validation | testing |
| |- mitotic | |- mitotic | |- mitotic |
| |-interphase | |-interphase | |-interphase |

**Info-WGANGP**

WGANGP with information maximization (Info-WGANGP) is used for generating new mitotic samples for oversampling purposes. This model is constructed based on the original paper [1]. Our implementation was built based on the original open source Pytorch implementation [[here](https://github.com/bohu615/nu_gan)] Here we provide the Info-WGANGP generating function that used a pre-trained generator to synthesize new mitotic images.

**Usage**

* Synthesized new miotic images using Info-WGANGP:
  + python infoWGANGP\_generator.py --generate\_num 100

The model pretrained weights [download] should be located where the python code is (with subfolder name:'.\model'). The number of synthesized images could be specified using the argument --generate\_num.

**Att-DCRNet**

Here we provide Pytorch implementation of the proposed Self-attention Att-DCRNet, which is an improved version of the baseline DCRNet [2]. The attention mechanism is implemented based on the adopted convolutional-based attention module (CBAM) [3] [[here](https://github.com/Jongchan/attention-module)]

**Usage**

1. Training Att-DCRNet:
   * python AttDCRNet\_Main.py --train\_path YOUR DIRECTORY --val\_path YOUR DIRECTORY

The directories of the training and validation set should be provided (validation set is optional). For listing all arguments:

* + Python AttDCRNet\_Main.py -h

1. Testing Att-DCRNet:
   * python testing.py --test\_path YOUR DIRECTORY --model\_path YOUR DIRECTORY

The directories of the test set folder and the Att-DCRNet model pretrained weights should be provided.

**Requirements**

* Pytorch 1.X
* Python 3

The code is validated under below environment:

Windows 10, RTX 2080 SUPER GPU device, anaconda environment (PyTorch 1.4, CUDA 10.1, Python 3.6))

**References**

[1] B. Hu, Y. Tang, E. I. C. Chang, Y. Fan, M. Lai, and Y. Xu, “Unsupervised learning for cell-level visual representation in histopathology images with generative adversarial networks,” IEEE J. Biomed. Heal. Informatics, vol. 23, no. 3, pp. 1316–1328, 2019, doi: 10.1109/JBHI.2018.2852639.

[2] L. Shen, X. Jia, and Y. Li, “Deep cross residual network for HEp-2 cell staining pattern classification,” Pattern Recognit., vol. 82, pp. 68–78, Oct. 2018, doi: 10.1016/j.patcog.2018.05.005.

[3] S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: Convolutional block attention module, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision - ECCV 2018, Vol. 11211 LNCS, Springer International Publishing, Cham, 2018, pp. 3-19. doi:10.1007/978-3-030-01234-2\_1.