Instance Segmentation

Project Report

Syed Kumayl Raza Moosavi

Reg. No.: 364013

MS AI 2K21

June 12, 2022

CS 893 - Advanced Computer Vision

Instructor - Dr. Muhammad Moazam Faraz

NETWORK DETAILS

The standard model of the maskRCNN, as shown in Figure 1, uses a ResNet50 as the backbone. The second model backbone used is basically a rendition of the DeepLab V3 network in which xception is used as the backbone, atrous convolution is used in the last few blocks of the backbone. Model checkpoints are created at every iteration and the weights are saved at every epoch (10 epochs in total) and the weights of the trained model are saved at each epoch and used later for inference purposes. mrcnn library is imported from HoverNet GitHub repo [1]. The model configurations edited for nuclei dataset are shown in Table I. Tranfer learning is also performed in which mask_rccnn weights are loaded from the MS COCO image dataset trained on 80 classes. Support has also been taken from [2] and [3].

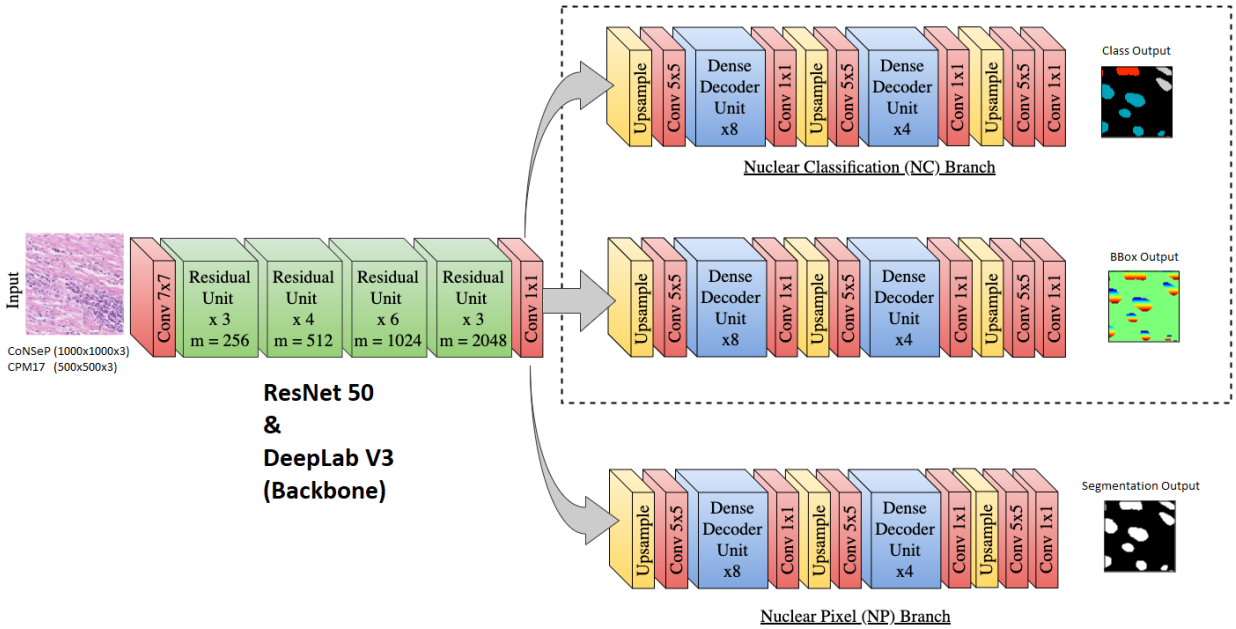

Figure 1: Mask RCNN Architecture

DATASET

The dataset downloaded contains 27 training and 14 test images in CoNSeP and 32 training and 32 test images in CPM17 datset. In conjunction with these images, relevant MAT files are present in which inst_map array are used as labels. For training purposes Data Augmentation is also added such as rotation, width/height shift, horizontal flip, etc.

Table I: Model Configurations

S.No.	Parameter	Value
1	NUM CLASSES	1+1
2	IMAGE MIN DIM	512
3	IMAGE MAX DIM	1024
4	DETECTION MIN CONFIDENCE	0
5	STEPS PER EPOCH	100
6	BACKBONE	resnet50 & xception
7	IMAGE MIN SCALE	2.0
8	RPN ANCHOR SCALES	(8, 16, 32, 64)
9	POST NMS ROIS TRAINING	50
10	POST NMS ROIS INFERENCE	100
11	RPN NMS THRESHOLD	0.9
12	RPN TRAIN ANCHORS PER IMAGE	32
13	TRAIN ROIS PER IMAGE	128
14	MAX GT INSTANCES	100
15	DETECTION MAX INSTANCES	200

QUANTITATIVE ANALYSIS

This section shows the quantitative results for the networks trained at 10 epochs. Train_validation split of the input dataset is 80% by 20%. For testing purposes the test dataset provided in the dataset is preprocessed and evaluated. Table II shows the model metrics of training and testing results. The performance metrics of AJI, PQ, SQ, Dice, and DQ can be imported from mrcnn.stat_util library but I was unable to include those metrics without getting some unexpected errors.

Table II: Training and Validation Metrics

Train Metric	Network Trained on CoNSeP	Network Trained on CPM17
Loss	2.3175	2.8385
RPN Class Loss	0.1486	0.3236
RPN Bbox Loss	1.8442	1.8289
MRCNN Class Loss	0.035	0.0472
MRCNN Bbox Loss	0.0765	0.2796
MRCNN Mask Loss	0.2131	0.3592

Validation Metric	Network Trained on CoNSeP	Network Trained on CPM17
Loss	4.115	2.7268
RPN Class Loss	0.1879	0.361
RPN Bbox Loss	1.8133	1.2191
MRCNN Class Loss	0.0663	0.07
MRCNN Bbox Loss	0.1908	0.4918
MRCNN Mask Loss	0.2176	0.5096

Time is also an evaluation metric for model comparison and evaluation. Table III shows the train and evaluation time of the model to achieve the requisite metrics.

Model training graphs is also shown in Figure 2 for CoNSeP dataset and Figure 3 for CPM17 dataset. The processor used for computation was an Intel Core i7-9750, GPU: Nvidia GTX-1660 Ti.

Table III: Time Analysis

Time (minutes)	CoNSeP Dataset	CPM17 Dataset
Training Time	118.32 minutes	25.35 minutes

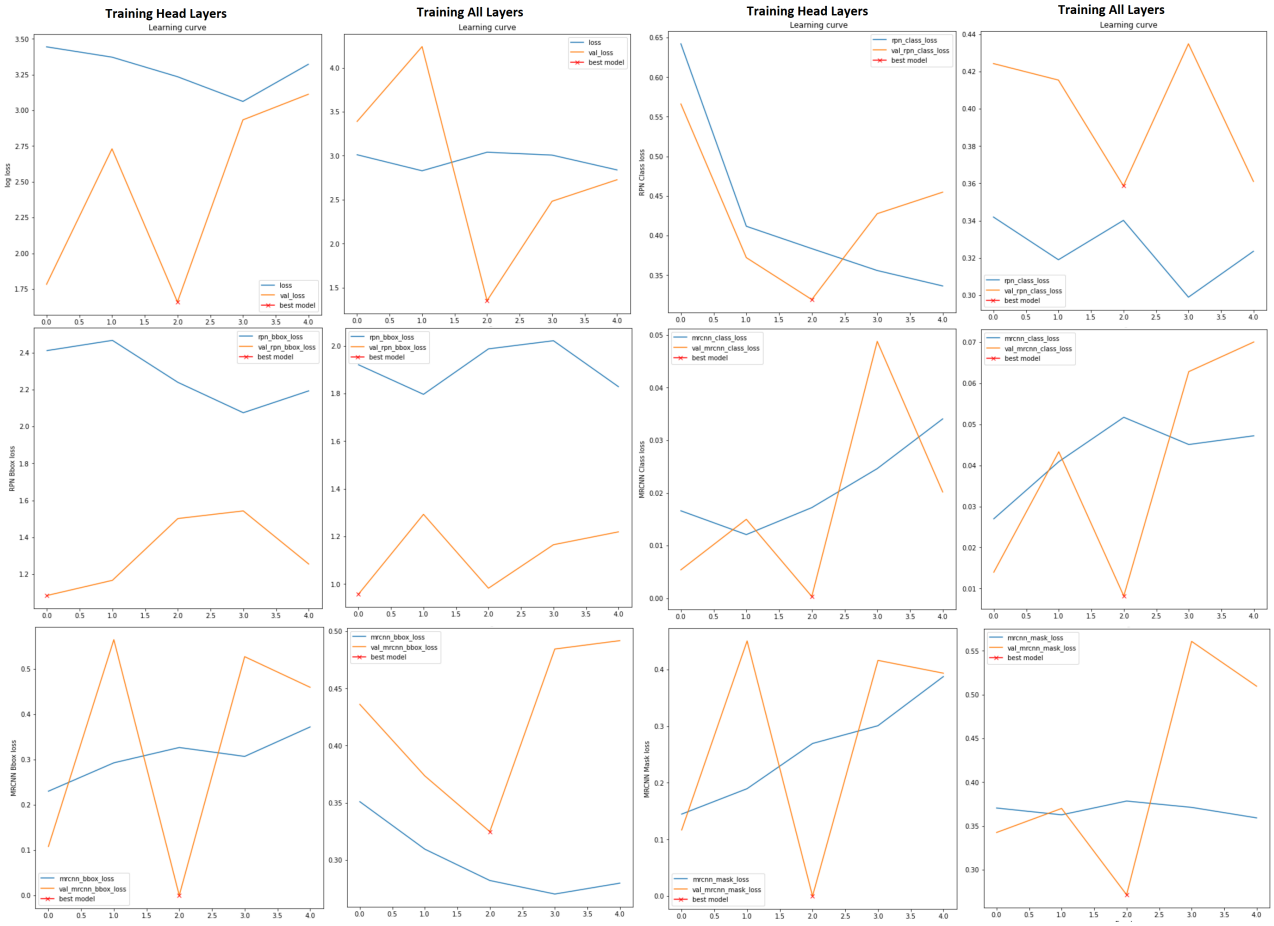

Figure 2: CoNSeP Network Learning Curves

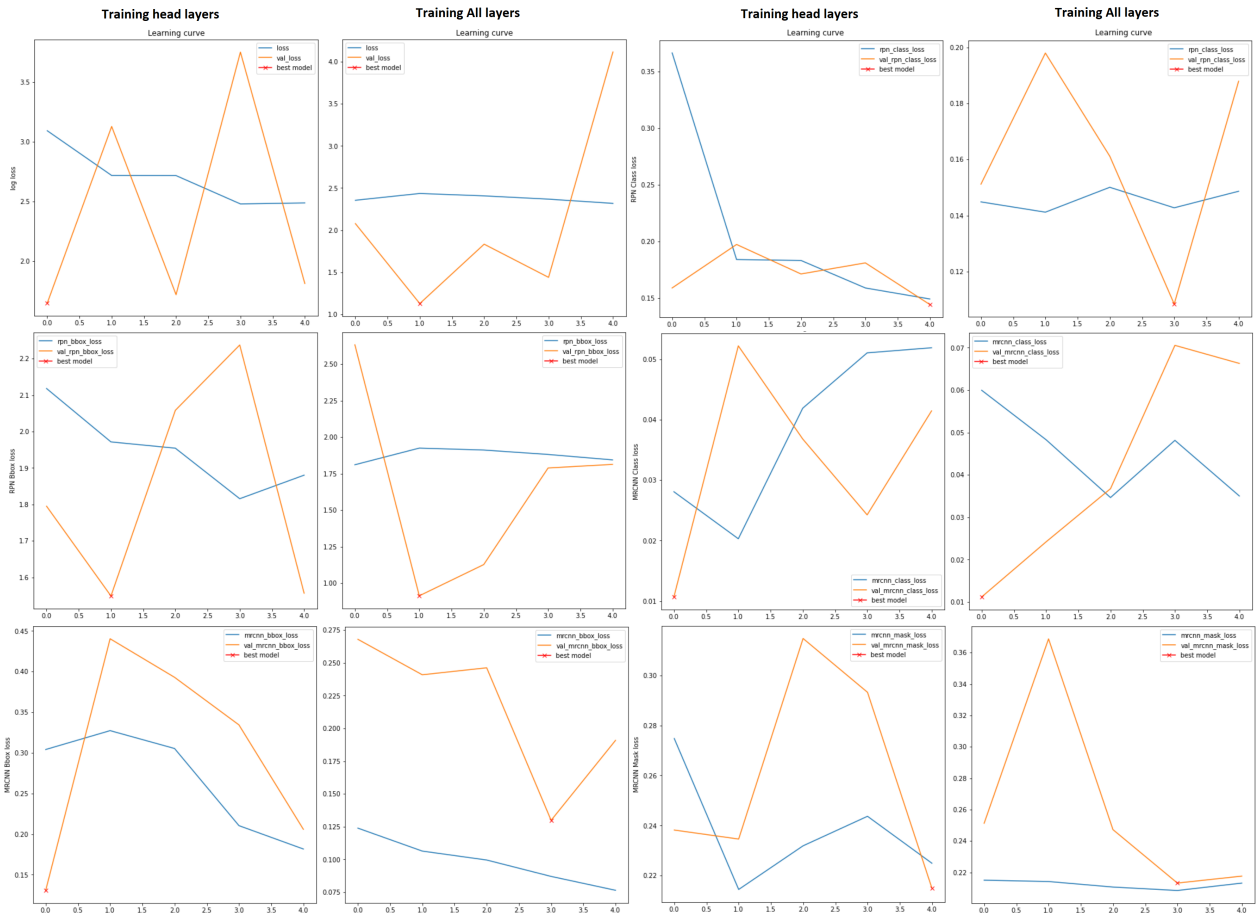

Figure 3: CPM17 Network Learning Curves

QUALITATIVE ANALYSIS

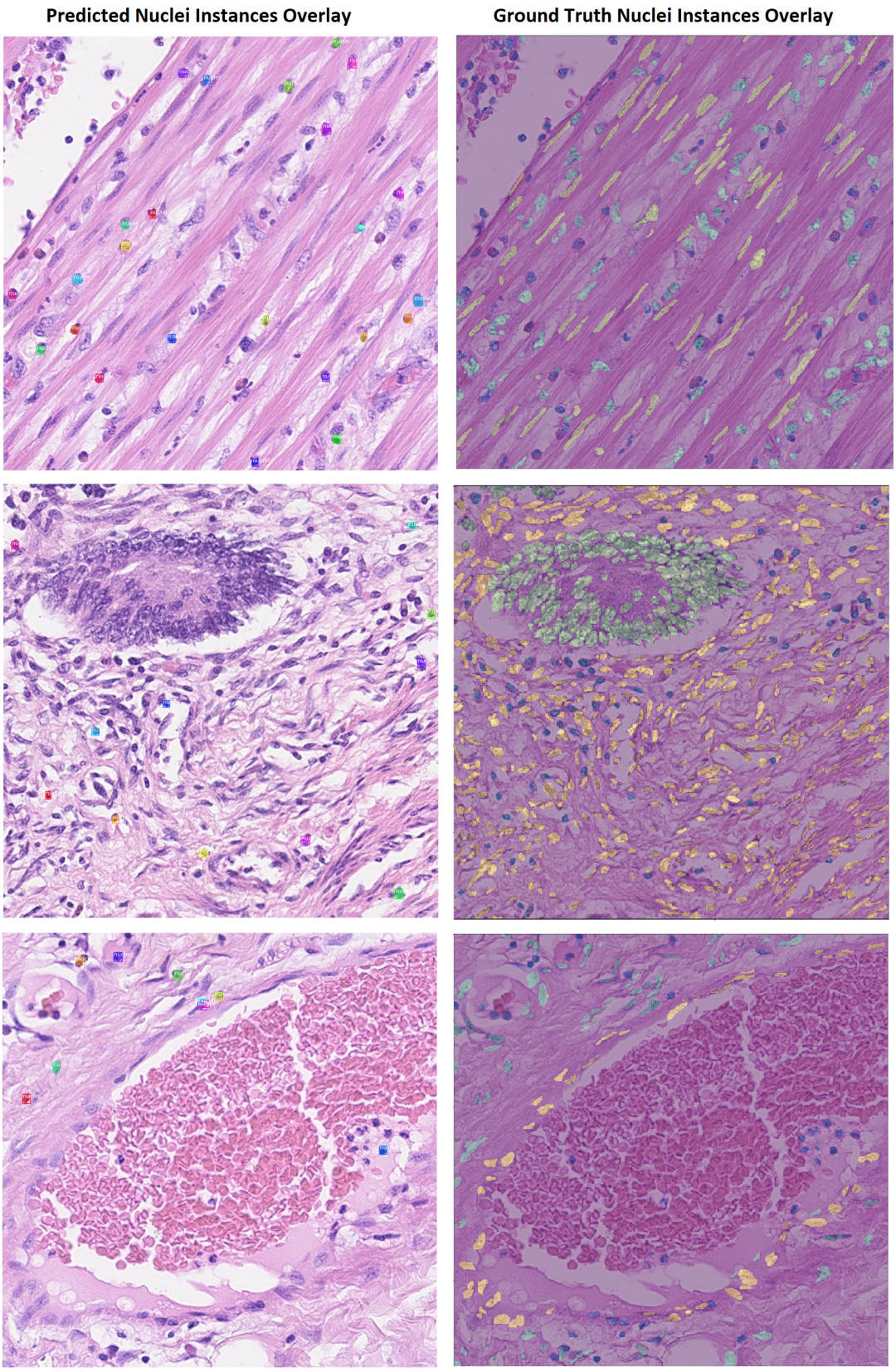

Figure 4: CoNSeP Images overlay

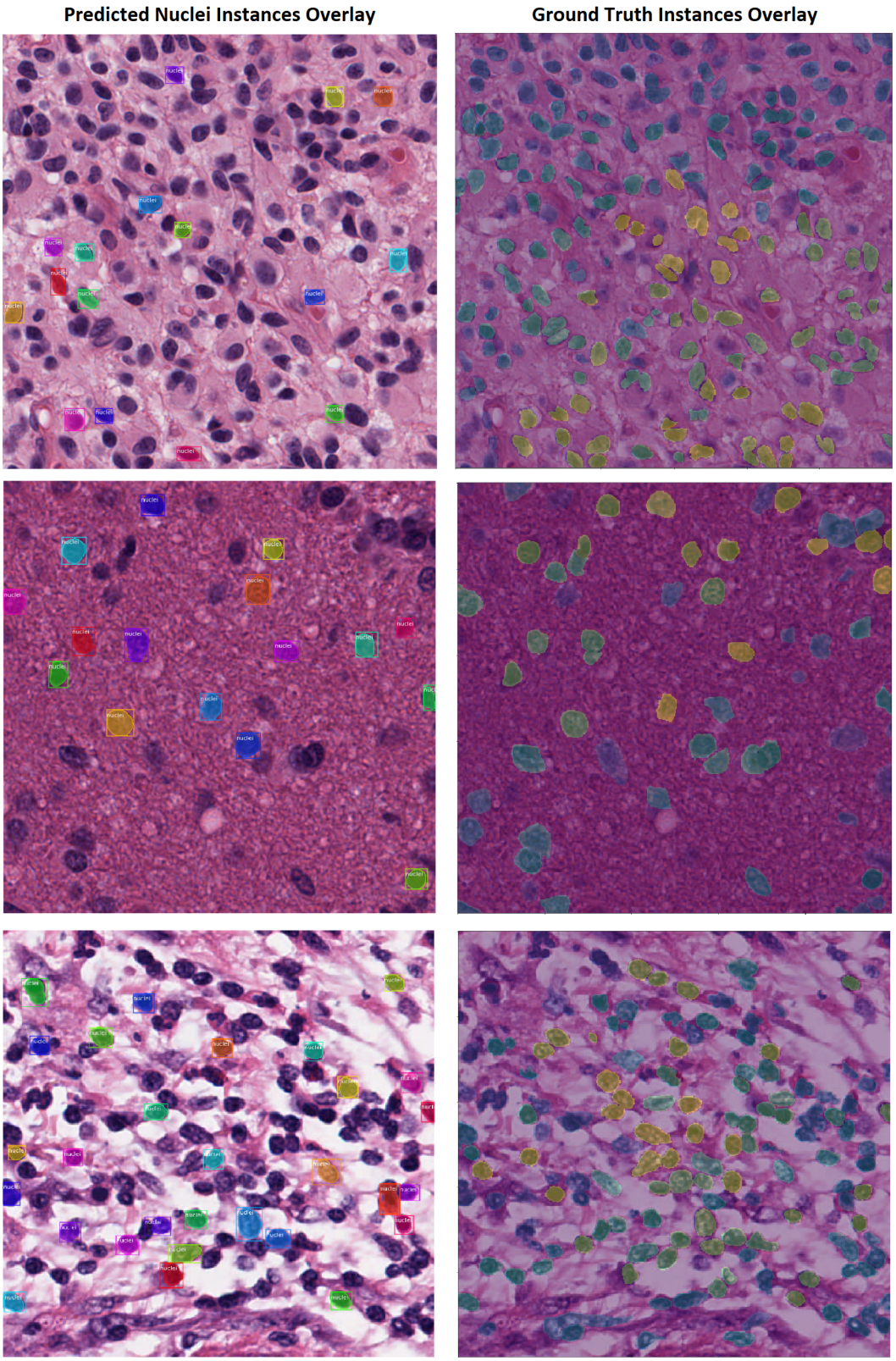

Figure 5: CPM17 Images overlay

HOW TO RUN THE CODE

Download the rar file containing the Project folder from the Github repository from the link provided as under:

`Github`: Github Repo

The project zip file and short presentation is also available attached with this report

Download dataset zip file from the following link:

`Google Drive` : CoNSeP_Dataset CPM17_Dataset

The Github link contains the python notebook. Unzip dataset folder. Folder 'A3_Dataset' and python notebook file should be in the same folder.

Run the jupyter notebook file

REFERENCES

[1] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot, "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," https://github.com/vqdang/hover_net, p. 101563, 2019.

[2] B. Bancher, A. Mahbod, I. Ellinger, R. Ecker, and G. Dorffner, "Improving mask r-cnn for nuclei instance segmentation in hematoxylin & eosin-stained histological images," https://github.com/bbanc/Improved-Mask-R-CNN-for-nuclei-segmentation, PMLR, pp. 20–35, 2021.

[3] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.