I Can See U Now:
Nuclei Segmentation and Classification Using U-Net

Artificial Intelligence Project Report

Zhao Lutong 1002872
Wang Zijia 1002885
Tang Xiaoyue 1002968
Lin Xiaohao 1003144

Contents Page

Abstract

Automated nuclei segmentation and classification have huge potential in advancing biomedical research. However, technical challenges such as difficulty in identifying the nuclei in varied conditions and the lack of image data are hindering progress. To overcome these challenges, we present the U-Net , a 23-layered Convolutional Neural Network with a symmetric contracting and expansive path (Ronneberger et al, 2015). Coupled with data augmentation methods like cropping to increase the amount of training daa, our U-Net is able to efficiently segment and classify nuclei within 80 epochs. Our segmentation results have produced a satisfactory Jaccard Index (IOU) score of 0.45, and our classification accuracy is 92%, which outperformed other state-of-the-art-networks. The dataset used is the Warwick ConSEP nuclei image dataset (S. Graham et al, 2019) with 41 images, each of size having the size of 1,000 x 1,000 pixels, at 40 objective magnification.

Project Background

Automated nuclear segmentation and classification is promising in speeding up biomedical research. With automated processes, researchers are able to identify and categorise nuclei in a cell effortlessly and measure how cells react to various treatments. Subsequently, researchers can understand the underlying biological processes at work, bringing the industry closer to cures in diseases, such as cancer, heart disease, and diabetes.

However, nuclear segmentation and classification are highly challenging because nuclei are available in a varied number of positions, and due to limited training data (S. Graham et al, 2019) . Without access to thousands of annotated training samples, which are typical of Deep Neural Network training, this project explores the efficient solution of using U-Net with data augmentation as an alternative. U-Net is made up of 23 convolutional layers, which are extremely strong in capturing translational invariance. Moreover, U-Net has a unique architecture of a symmetric contracting and expanding path (Ronneberger et al, 2015). The contracting path captures context while the expanding path captures specific localisation. We have achieved stellar results with U-Net, demonstrating that it is suitable for the task of nuclear segmentation.

Dataset Description

The dataset used in this report is the colorectal nuclear segmentation and phenotypes (CoNSeP) dataset. It consists of 41 H&E (Haematoxylin & Eosin) stained image tiles, each of size having the size of 1,000 x 1,000 pixels, at 40 objective magnification.

There are 24,319 nuclei in total, which are classified into seven categories, with the respective labels as detailed below:

0 = background
1 = other
2 = inflammatory
3 = healthy epithelial
4 = dysplastic/malignant epithelial
5 = fibroblast
6 = muscle
7 = endothelial

The labels are colour coded as shown in Fig 1 below:

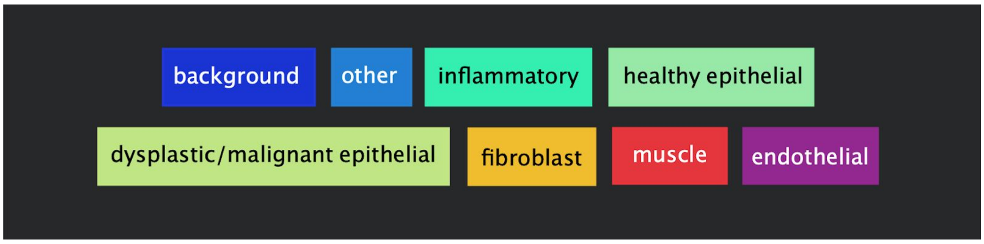


Figure 1: Colour Code For Labels

Data Augmentation and Preprocessing

Image cropping

We cropped input images and result labeling as well as masked into 256x256 size to avoid large numbers of nuclei gathering in one image. Since there are 24,319 nuclei on 27 images, for clearer results with better accuracy, and also for data augmentation, we divided each image into sixteen 256*256 sub-images.

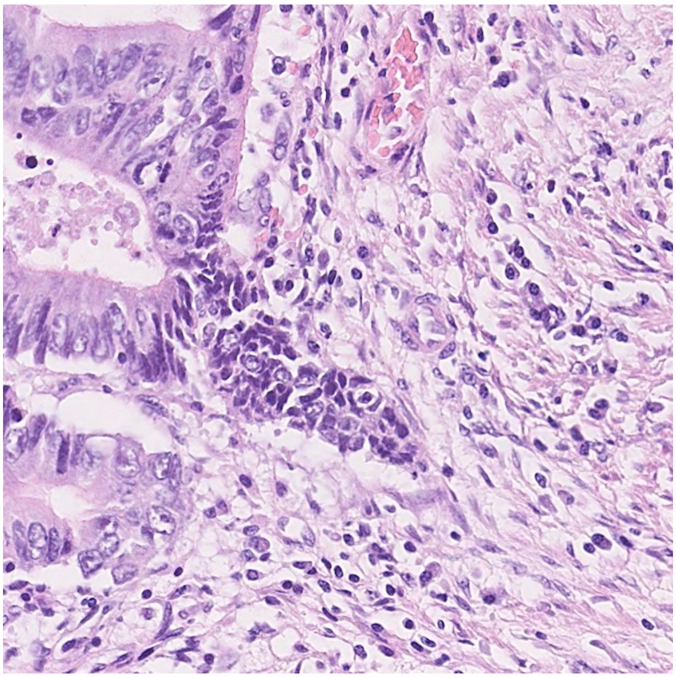


Figure 2: Sample Image Before Cropping

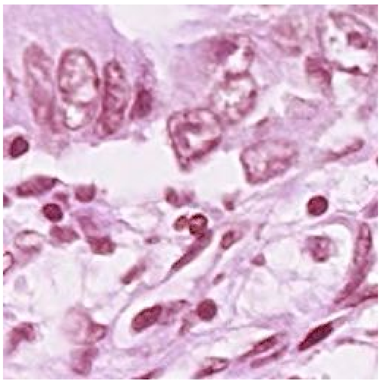
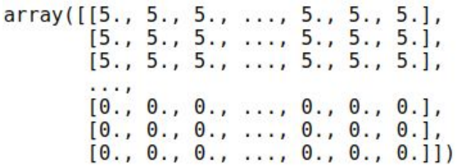


Figure 3: Sample Image After Cropping

Preprocessing

We used 10 subimages for testing, and the other 646 sub images for training and validation set with a ratio of 9:1.

As part of pre-processing, we also cropped the same size pixel level label for later part of classification. An example of the generated label is as follows:

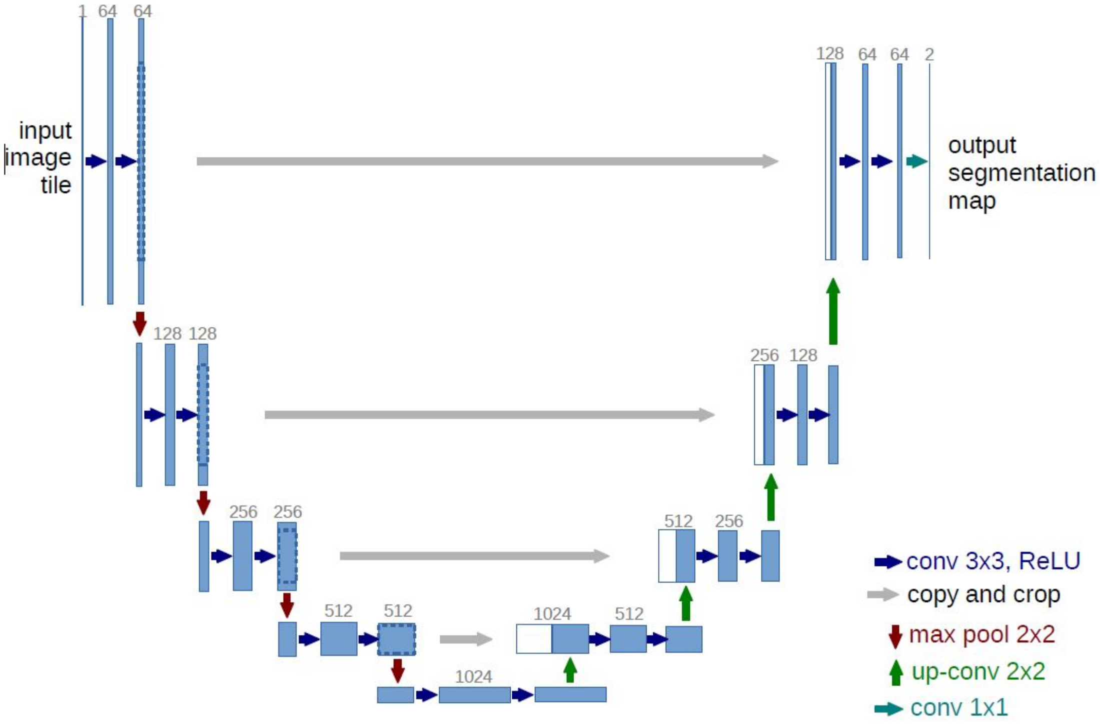


```
array([[5., 5., 5., ..., 5., 5., 5.],
       [5., 5., 5., ..., 5., 5., 5.],
       [5., 5., 5., ..., 5., 5., 5.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Figure 4: Data Transformation

Model Architecture

Convolutional Neural Network (CNN), with its robust feature learning capability and local invariance property, is a promising algorithm for image segmentation. In addition, given that there was a small training set of only 646 images, we believed that U-Net, together with data augmentation, was the best approach to train efficiently. We adopted two U-Net architectures, UNet1 and UNet2. Both are identical in structure with only one difference: the number of out_channels in U-Net 2 is only a **quarter** of the U-Net 1. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The arrows denote the different operations. In total, there are 23 convolutional layers.



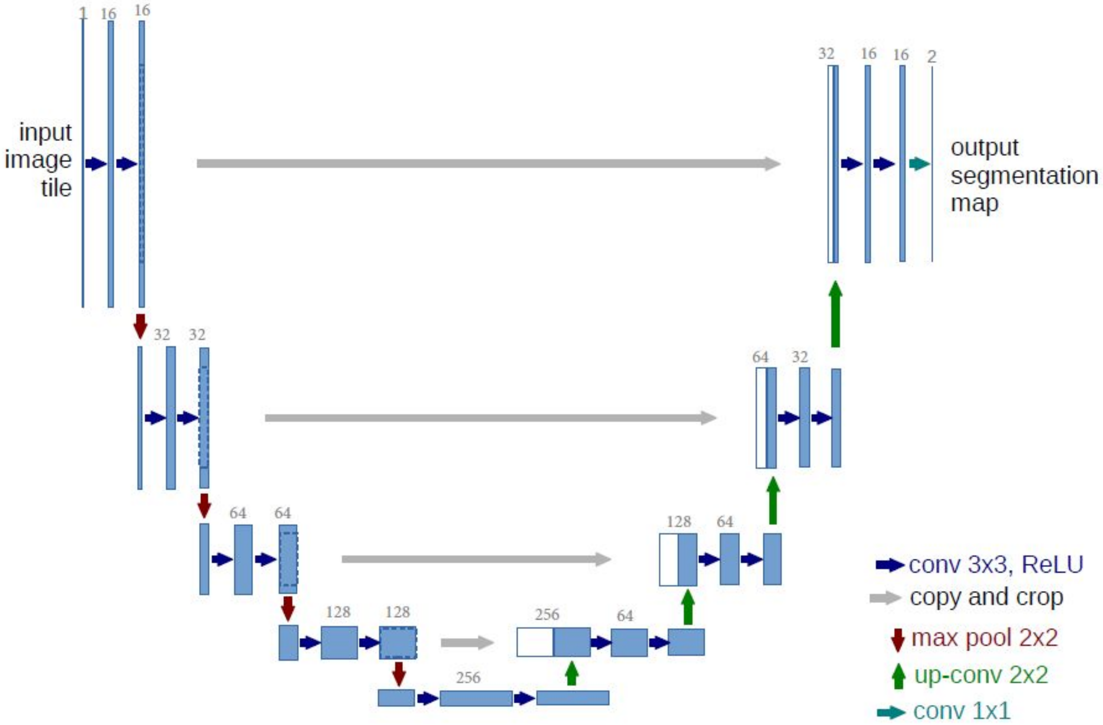


Figure 5: Comparison of U-net 1 (1st img) and U-Net 2 (2nd-img) Architectures
Source: (Ronneberger et al, 2015)

The network consists of a contracting path (left side) and an expansive path (right side). More details are listed in the table below:

Contracting path[1]	Expansive path [2]
Repeated application of two 3x3 convolutions, each followed by ReLU.	Repeated two 3x3 convolutions, each fol-lowered by a ReLU.
2x2 max pooling operation with stride 2 for downsampling.	2x2 convolution ("up-convolution") for un-sampling
At each downsampling, number of features doubled	At each un-sampling, number of feature channels halved
-	Repeated concatenation with the correspondingly cropped feature map from the contracting path
-	Final layer: a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes.

Table 4: Description of the U-Net Paths

Therefore, when we implemented this two UNet structure, there are two more stacks of convolutional layers for Uet2, with each stack consisting of Conv2d + BatchNorm + ReLU + Conv2d + BatchNorm + ReLU structure.

Loss Function

For pure segmentation, we used BCE (Binary Cross Entropy) loss after the final sigmoid layer of the U-net. We read the ground true label as boolean and compare the prediction that went through logits with the 0/1 labels. Both shapes are (batchsize, 256,256).

To perform class-wise segmentation, we used cross-entropy loss for multi-class recognition. If the output channel of each prediction is (batchsize,8, 256, 256), we will take the argmax of each position of the image to be its predicted class, then use color mapping from matplotlib to display the class-wise segmentation results. The label

[1] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[2] IBID

passed in to the loss function is in the shape of (batchsize, 256, 256), and it is generated from the .mat label's 'type_map' category with pixel_level class value.

Hyperparameters Settings

1. **Learning rate**

With a learning rate of **0.01**, the validation loss oscillates and is very unstable, as shown in the figure below:

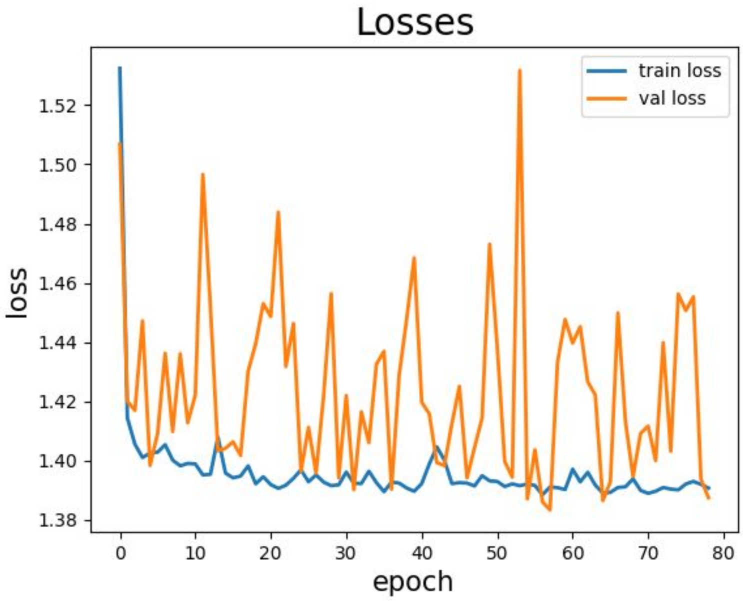


Figure 6: Training and Validation Loss for Learning Rate of 0.01

However, with lower learning rates o**f 0.001 and 0.0001,** we can see that the validation loss smoothens, and converges. Since the learning rate of 0.001 converges faster, it was chosen.

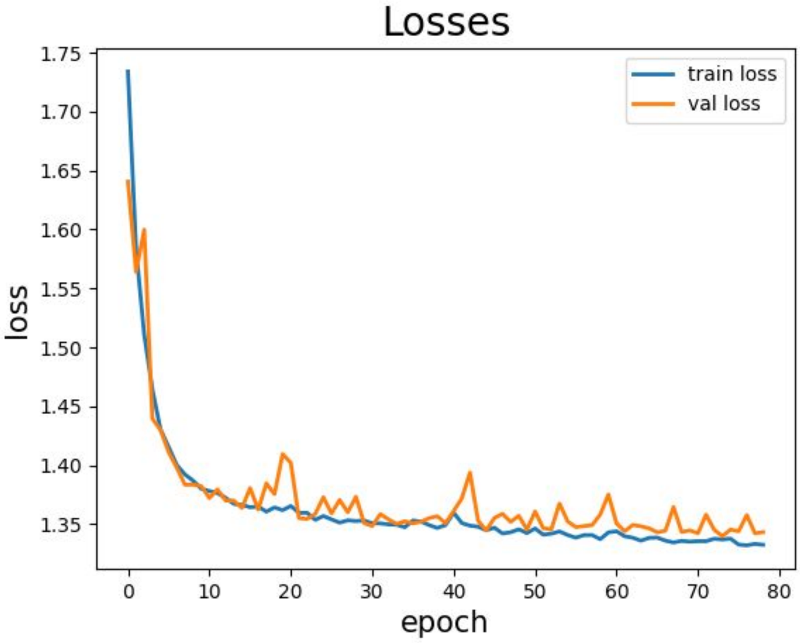


Figure 7: Training and Validation Loss for Learning Rate of 0.001

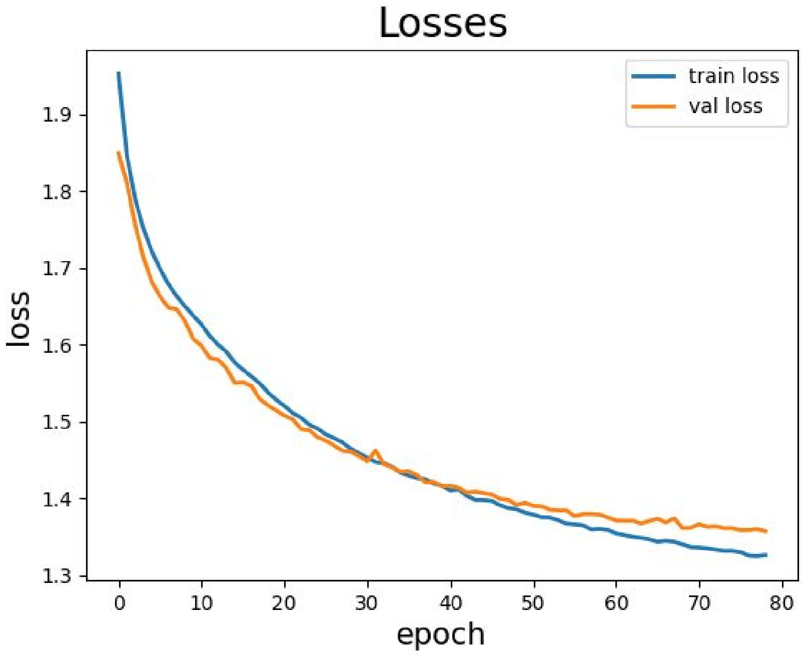


Figure 8: Training and Validation Loss For Learning Rate of 0.0001

2. Number of Training Epochs

The following epoch numbers were experimented: 45, 80, and 150, which are shown in Fig 9, 10, 11 respectively. With 45 epochs, the training loss only reaches 1.35 and the trend indicates it could be lower with more epochs. With 150 epochs, the final loss is almost the same with 80 epochs. It seems to reach a bottleneck for validation loss even when the training loss is still decreasing. There, the number of epochs is set to 80 for faster convergence.

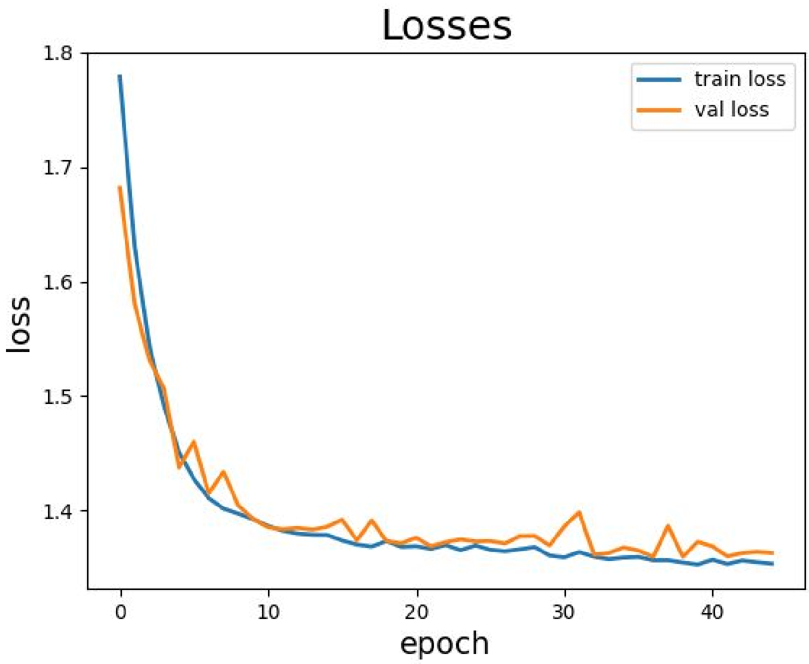


Figure 9: Training and Validation Loss for 45 Epochs.

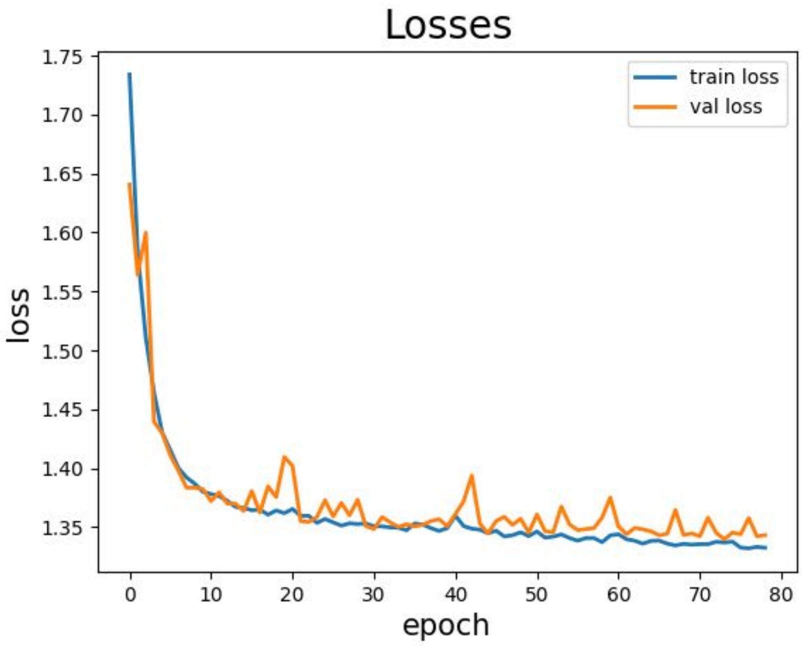


Figure 10: Training and Validation Loss for 80 Epochs

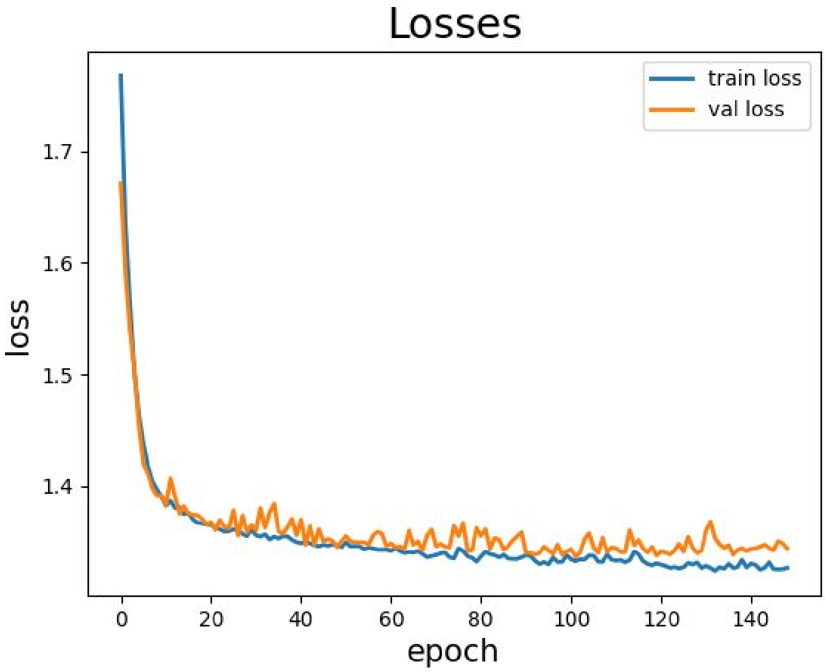


Figure 11: Training and Validation Loss for 150 Epochs

Performance Evaluation

Evaluation Criteria

1. Nuclei segmentation

We used the Jaccard index as our evaluation matrix. The Jaccard index, also known as Intersection over Union (IOU) and the Jaccard similarity coefficient, is a statics used for gauging the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

2. Nuclei classification

The evaluation criterion used is vanilla accuracy for every pixel, the precision, recall and F1 scores.

Results Discussion

1. Nuclei Segmentation

The model has achieved stellar results, as exemplified in the figure below, where most of the visible nuclei are identified.

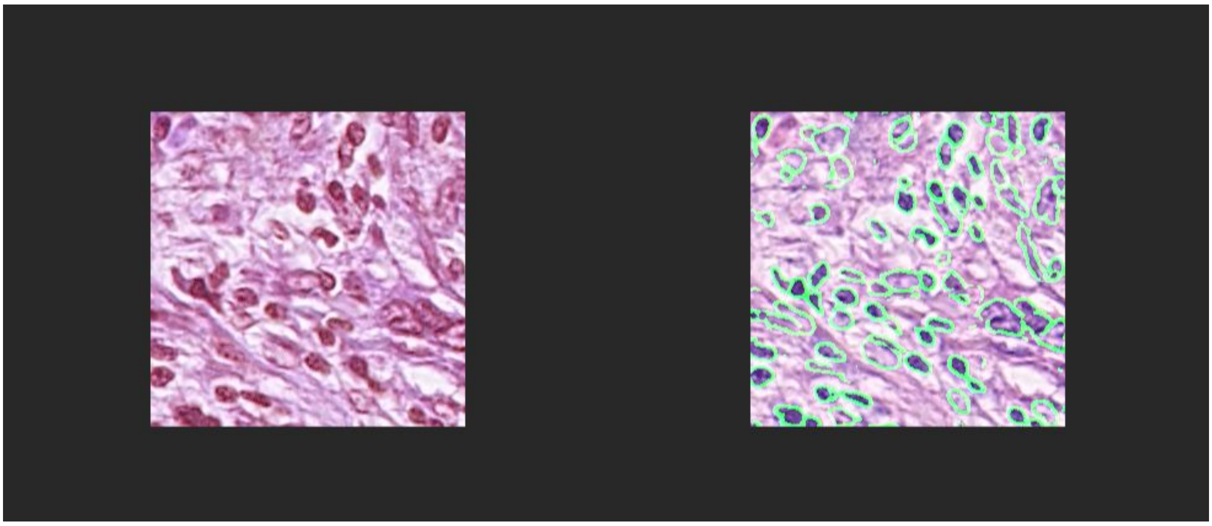


Figure 12: Segmentation Result

This tables below shows our Jaccard Index compared with other state-of the-art network architectures[3], and enhanced versions of the U-Net[4], which shows our network has done reasonably well, but there is still room for improvement.

	U-Net1	U-Net2	PangNet	DeconvNet	FCN	Ensemble
Jaccard index	0.53	0.804	0.722	0.814	0.782	0.804

	U-Net1	U-Net2	U-Net (Long F. et al, 2020)	U-Net++	U-Net+ (T.C.)	U-Net+ (U.S.)
Jaccard index	0.53	0.804	0.533 ± 0.173	0.552 ± 0.217	0.552 ± 0.217	0.551 ± 0.187

[3] P. Naylor, M. Laé, F. Reyal and T. Walter, "Nuclei segmentation in histopathology images using deep neural networks," 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, VIC, 2017, pp. 933-936, doi: 10.1109/ISBI.2017.7950669.

[4] Long, F. Microscopy cell nuclei segmentation with enhanced U-Net. *BMC Bioinformatics* 21, 8 (2020). https://doi.org/10.1186/s12859-019-3332-1

Table 2: Benchmarking Segmentation Results with Other Networks

2. **Nuclei Classification**

Comparison with state-of-the-art

The accuracy score of our U-Net is compared with the state-of-the-art performance[5], as shown in the table below. Our U-Net has outperformed all of them, showing stellar performance.

Score type	U-NET1	U-NET2	Farsight (%)	MI (%)	DCN (%)	MSER-based (%)	FCM-based (%)
Average Accuracy	85.55	89.15	82.27	89.55	91.65	84.63	86.60
Precision	71.54	76.45	61.03	-	84.89	61.37	64.04
Recall	76.2	81.01	98.23	-	80.84	88.57	94.49
F1	57.48	68.71	74.15	77.33	82.34	72.50	76.34

Table 3: Benchmarking Classification Results with Other Networks

Therefore, because UNET-2 has better results in both segmentation and classification, it is chosen to be the final architecture. Moreover, we acknowledge that the main limitation for our model is that we used two model weights for two tasks: segmentation and classification, and this could potentially slow down the training process.

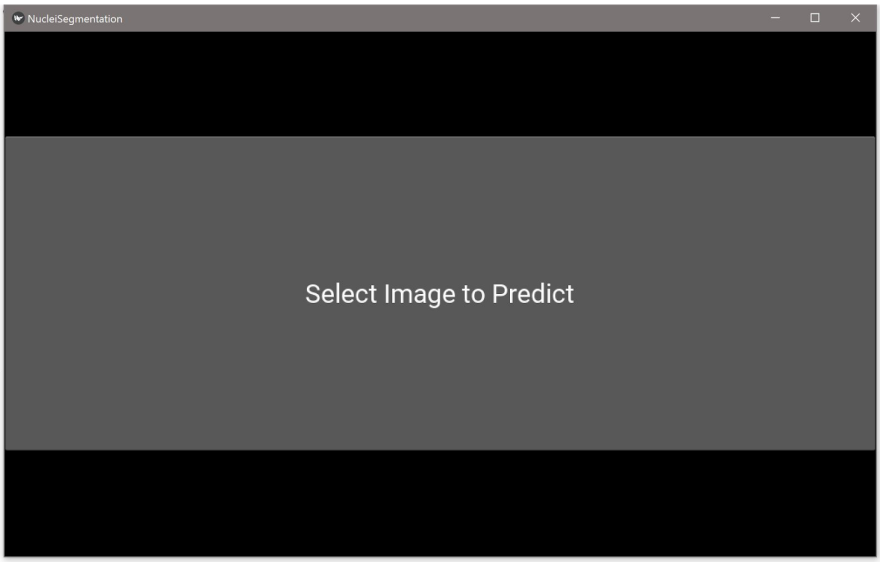
Graphical User Interface

1. How to set up our GUI
   Step 1): Install the Kivy packages in requirements.txt, using $ pip install -r requirements.txt
   Step 2): Run gui.py in UNet-pytorch/GUI
   (If you are running on GPU, please make sure is_cuda = False under utils.py)

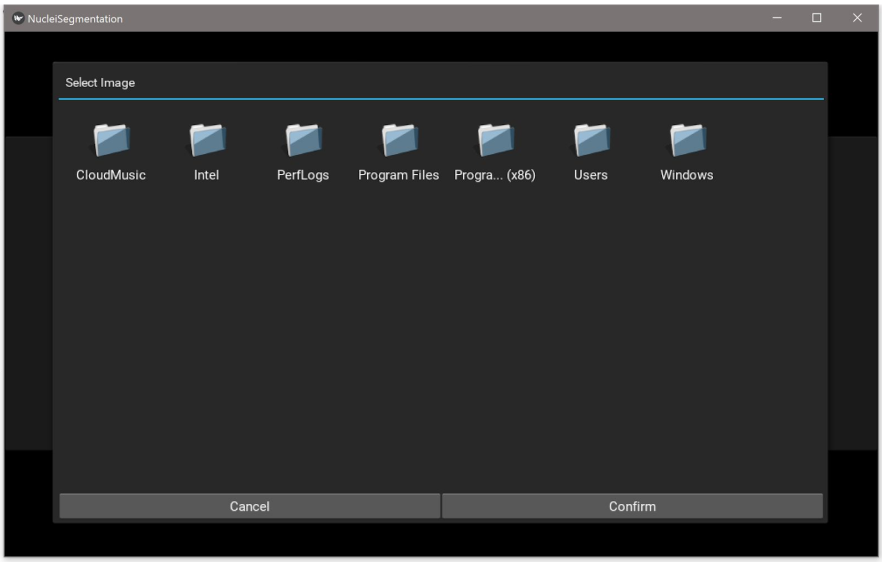
2. How to use our GUI

[5] Abdolhoseini, M., Kluge, M.G., Walker, F.R. *et al.* Segmentation of Heavily Clustered Nuclei from Histopathological Images. *Sci Rep* 9, 4551 (2019). https://doi.org/10.1038/s41598-019-38813-2

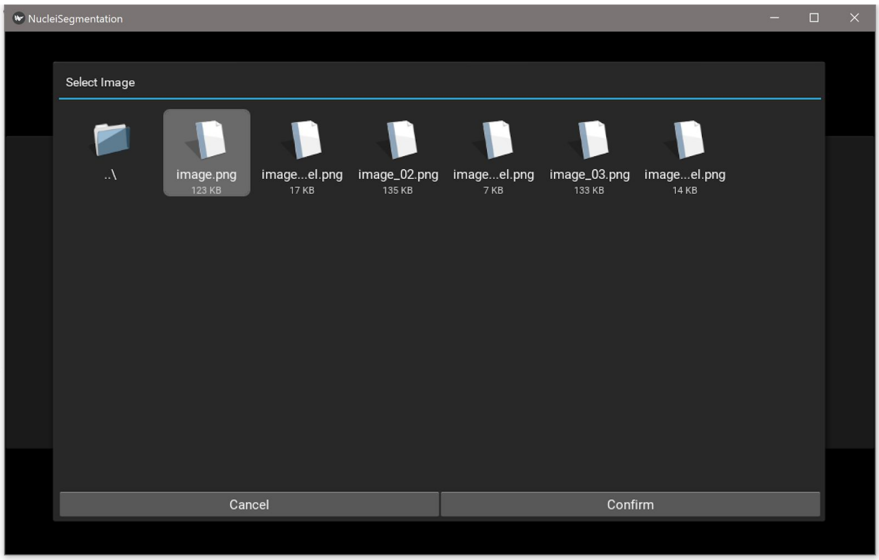
Step 1): Click the only button "Select Image to Predict" in the main page



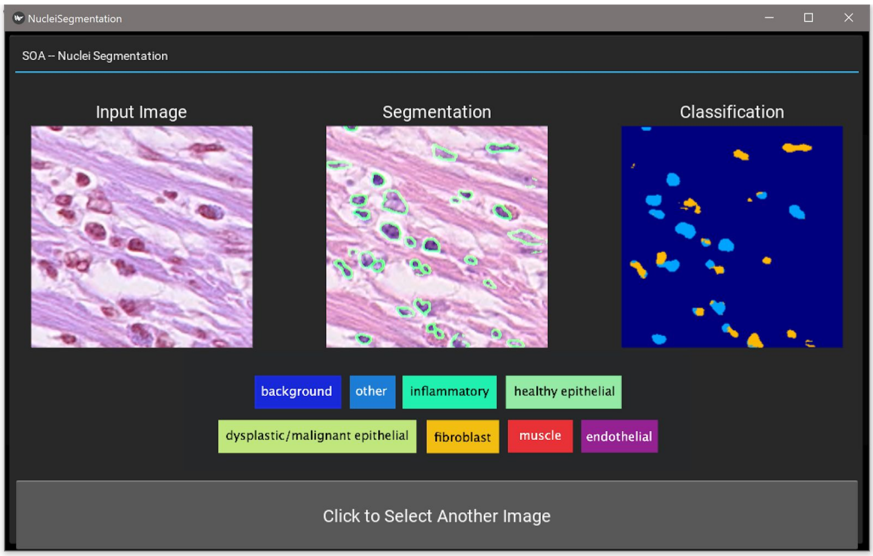
Step 2): Go to the location of the nuclei image in your local machine that you want to segment or classify



Step3): Select the image and click the button "Confirm"



Step 4): Wait for a little while, you would see the results of segmentation and classification. You may click the button at the bottom to select another image or just quit the APP.



3. Important Features
    a. The results of segmentation and classification are shown against each other in a horizontal fashion. This makes it significantly convenient for the user to compare the results of multiple tasks.
    b. There is an indicator which labels the class represented by each color. When the user is interested in the task of classification, he can easily determine the nuclei type by comparing the colors in the image to the ones in the indicator.

Conclusion

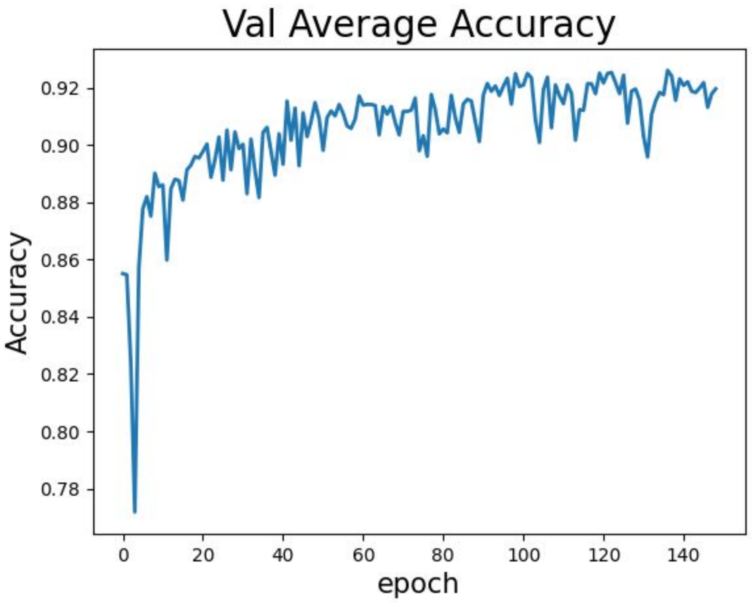
With only 80 epochs, our U-Net has performed reasonably well in segmentation and classification. Our classification accuracy is 89.15%, and outperformed other state-of-the-art networks. Our U-Net2 segmentation results are also as impressive as the state-of-the-art. Furthermore, we believe our UNET2 acts as a good starting point for us to gain insights of this special network, and how it can aid meaningful endeavours like biomedical research. In the future, we would improve on existing architecture, and also experiment with other state-of-the-art architectures to strive for better results.

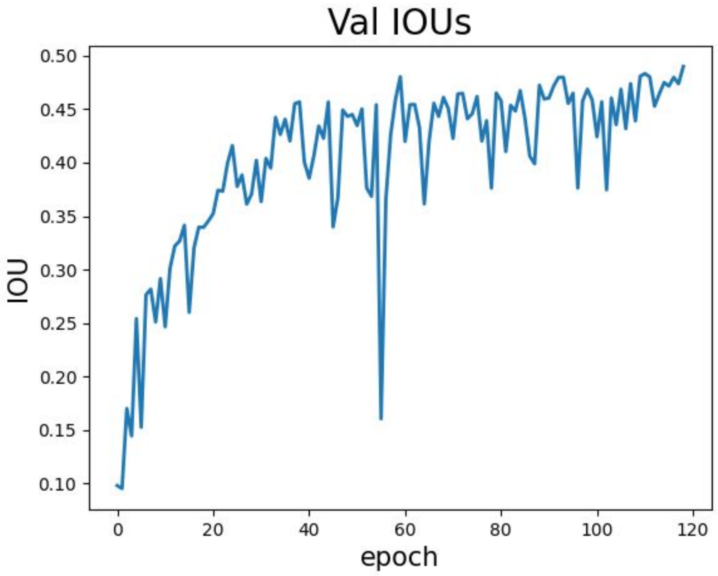
Reference

1. Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28
2. 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." Medical Image Analysis, Sept. 2019.
3. limingwu8. "limingwu8/UNet-Pytorch." *GitHub*, github.com/limingwu8/UNet-pytorch/tree/cf4ad9363509bd151c94a6aa8065ceb2965f6a43.
4. P. Naylor, M. Laé, F. Reyal and T. Walter, "Nuclei segmentation in histopathology images using deep neural networks," 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, VIC, 2017, pp. 933-936, doi: 10.1109/ISBI.2017.7950669.
5. Long, F. Microscopy cell nuclei segmentation with enhanced U-Net. *BMC Bioinformatics* 21, 8 (2020). https://doi.org/10.1186/s12859-019-3332-1
6. Abdolhoseini, M., Kluge, M.G., Walker, F.R. *et al.* Segmentation of Heavily Clustered Nuclei from Histopathological Images. *Sci Rep* 9, 4551 (2019). https://doi.org/10.1038/s41598-019-38813-2

Appendices

Appendix A: Graphs for Average Accuracy and Jaccard Index (IOU)





Appendix B: Instructions to run our code

Testing/Inference Mode:

1. Make sure the trained model (e.g. “model-150-color.pt”) is under root/checkpoints
2. Make sure is_train = False in utils.py
3. Under root, run python3 train.py

Training Mode:

1. Before Training, we need to firstly prepare our data:

i).Please put the CoNSeP dataset under structure of Root/data/concept/CoNSeP/Train(Test)

ii) Under root, run python3 procee_data.py

2. Then we will need to pass training parameter under utils.py:

Under class Option, please set is_train = True and is _cuda = True if you are using GPU for training.

3. Under root, run python3 train.py