AL-UNet: a Self-adaptive Multi-receptive-field Segmentation Neural Network for Cytology Image

Junwei Deng
School of Information
junweid@umich.edu

Yirui Gao
School of Information
yiruigao@umich.edu

Dongjian Chen
School of Information
djichen@umich.edu

Yunzhe Jiang
School of Information
yunzhej@umich.edu

Shuwan Feng
School of Information
averyfe@umich.edu

Abstract

We present an advanced network - **A**ggregated **L**ayer **U-Net** (**AL-UNet**) to do cytology image semantic segmentation on multiple applications such as microscope scanned cervical cancer image. Our work is mainly based on classical medical image segmentation convolutional networks, U-Net, but with more advanced settings and adaptations. A powerful novel model updated from U-Net which considers different receptive field sizes is proposed. Moreover, we design an auxiliary model to decide the cell size and provide the segmentation model pre-knowledge. By carrying out comparative experiments, we can prove that our proposed model outperforms another state-of-art segmentation network, U-Net++.

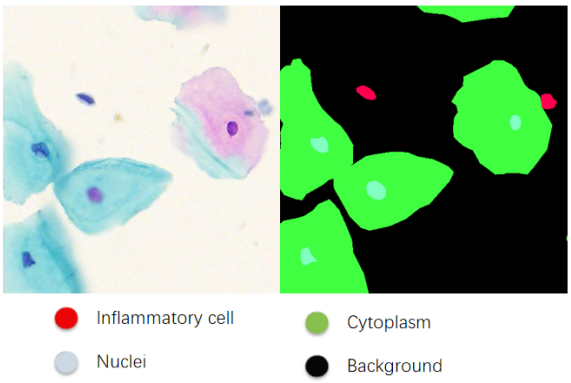

Figure 1: **An example of cell segmentation. The left figure is the original image, and the right one is the resulted figure after cell segmentation called *mask*. For each image, each units are categorized and mapped into the mask, which contains only four different components. The black is the background, and pink the cytoplasm is colored. The nucleus is blue, and the inflammatory cells are red.**

1 Introduction

Biomedical cytology image segmentation is a crucial technique in order to provide automatic and accurate characterization of cells in medical practices, like cancer research and drug discovery. Based on the classification of each pixel on a cellular image, cell image segmentation turns out to generate resulted images with clear cell boundaries and annotated cell types. For example, in Figure 1, we can see after the cell segmentation, the sub-cellular compartments are essentially structured and organized under the 2D microscopy, which greatly helps in medical diagnostics and researches.

The traditional models for biomedical cell image segmentation are variants of the encoder-decoder architecture such as fully convolutional network (FCN) (Long et al., 2014) and U-Net (Ronneberger et al., 2015). U-Net++ (Zhou et al., 2018) is another state-of-art powerful architecture based on the structure of U-Net as a deeply-supervised encoder-decoder network where the encoder and decoder sub-networks are connected through a series of nested, dense skip pathways.

Potential issues of the traditional methods turn out to be, first, models like U-Net doesn't have multiple adaptive receptive fields for different types of cells. For small cells like cancer cells (inflammatory cells), the segmentation effect is much worse than larger cells (normal cells with nuclei and cytoplasm). In our experiment, a network with 4 downsample(large receptive field) has better performance on large cells and worse performance on small cells. For large networks that provide multiple receptive fields, it heavily relies on artificial pruning to the model based on the whole validation dataset evaluation metric result(Zhou et al., 2018), while it doesn't take consideration of different types of cells in one image thus is not feasible in some real-life scenarios. Domain-specific training typically makes "data hungry" methods such as segmentation DCNN work better. In our project, we consider an improvement of this seg-

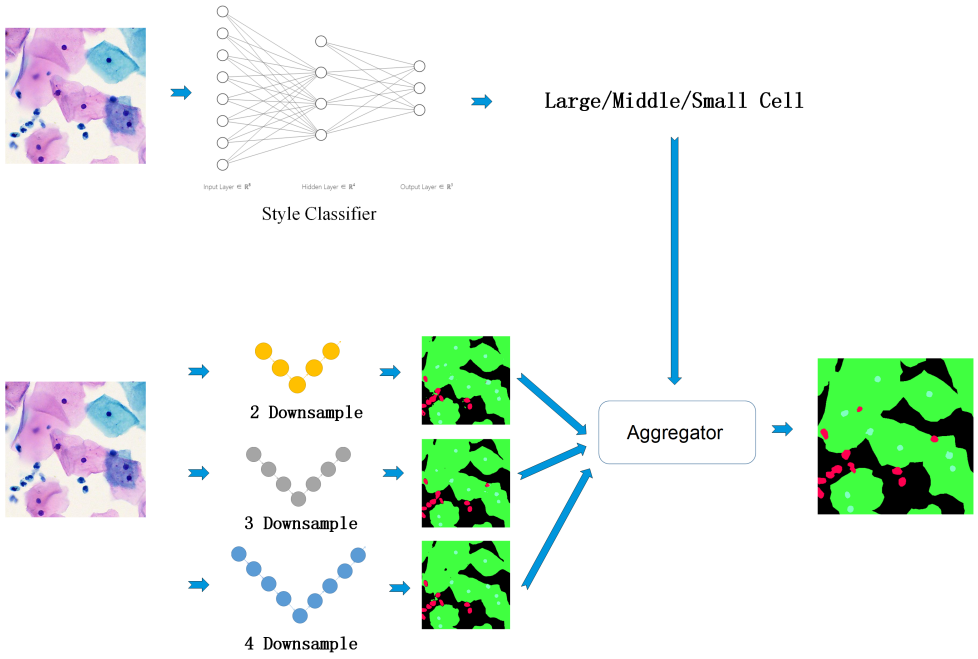

Figure 2: **The overall diagram for our proposed model. Our model is composed of a style classifier served for predicting different sizes of cells and a complex network utilizing structures like U-Net to perform down sampling. The final output is the aggregation of the products for the two components.**

mentation technology, particularly based on U-Net and U-Net++ structures.

Our main contributions in this project are:

- Based on U-Net, we propose a model based on domain specific learning.
- We design a style classifier to perform domain divisions based on the size of cells.

2 Related Works

Segmentation task has been a classical branch in deep learning. Fully Convolutional Network (FCN) (Long et al., 2014) is the first practical segmentation network for general image. U-Net(Ronneberger et al., 2015) shows an encoder-decoder baseline for medical images. For our dataset we have two state-of-art methods, which is (Deng et al., 2020) for exactly the same dataset we will use and (Zhou et al., 2018) for general medical image.

2.1 FCN

FCN is a traditional convolutional-based neural network, which connects convolutional components with fully connected layers. Adopting up-sampling and being trained end-to-end and pixels-to-pixels, FCN serves as a well tool to capture key information for deep features from high-dimensional input images, thus it is commonly used for semantic segmentation tasks.

2.2 U-Net

U-Net released in 2015 shows how to use encoder-decoder network to work on biomedical image segmentation tasks. It outperforms FCN by providing multiple layers for up-sampling and using skip connections and concatenates instead of direct adding. Like most existing medical image segmentation researches, our work will use U-Net as baseline to do further modification and improvement. Our work will consider multiple receptive fields for different types of cells in a network to improve the performance.

2.3 ResNet-based U-Net

Deng et al. (Deng et al., 2020) showed how to complete segmentation tasks of cells and subtype by designing a two-stage network in 2020. The work is done by one of our author and it is specifically designed to segment the dataset we use in this project. It is the state of art model for this dataset but the work did not compare their result with U-Net++.

2.4 U-Net++

U-Net++ (Zhou et al., 2018) released in 2018 is also a inspiration for our work since it showed that it will be helpful to propose a net-work with multiple receptive fields. And to improve the speed, it also provide a pruning method. It is the state of art method generally for medical image.

Other related works including:

- Ke et al. (Ke et al., 2019b) proposed VGG and dilated convolutions network based on U-Net and analyzed the pros and cons of two networks in 2019.

3 Proposed Method

We basically have three main components in our proposed model. A three-class style classifier which can decide the largeness of the cells in the input image. Three U-Net based models with different down-sample times and a mask aggregator which take the cell largeness and four mask produced by four U-Net based models. The structure is shown in figure 2.

3.1 Style Classifier

For the three-class style classifier, we will decide if the cells in our input image are large, middle or small. We will give a image-level label for one

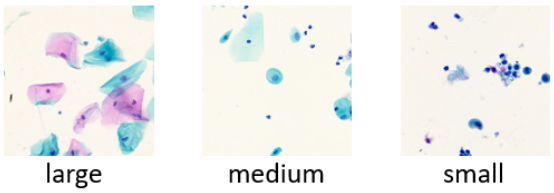

Figure 3: **Examples of three Categories of Image.**

input image rather than a label for each cell. The task is relatively easy, we will use a Resnet to complete the task.

3.2 U-Nets with different downsample times

In our proposed model, we will take advantage of the adaptive receptive field by have three U-Net based models with different down-sample times. The more the times of down-sample, the larger the receptive field and the better the performance to larger cells. Each U-Net model will provide a segmentation mask, we can express them as m_2, m_3, m_4 for masks produced by model with 2,3,4 downsample.

3.3 Aggregator

Aggregator is the most important and complicated part in our proposed model. First we will simply aggregate the mask by overlap all four masks by priority, the priority of four class is background, cytoplasm, nuclei and inflammatory cell from low to high. The we can have a pre-aggregated mask represented as m_p. Then we will find all the connected area in the image. In this step we will see nuclei and inflammatory to be one class. Here is a example of connected area search. Then we get

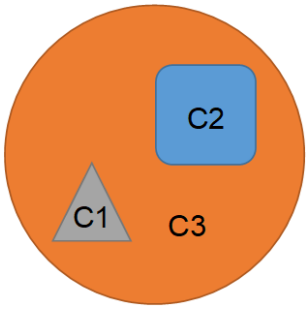

Figure 4: **An example of connected area, C_1, C_2 and C_3 are three connected area in this mask.**

a list of connected area $C_1, C_2, ...C_n$. For each connected area, we can have a weighted vote. The weight is related to the cell largeness classified by the style classifier. We can write a formula as following.

$$Class[C_k] = argmax_{\ l \in l_{list}} \sum_{p \in C_K} \sum_{i \in range(3)} w_i * I[m_i[p] == l]$$

In this formula, we have l_{list} as the four class we segment, which is background, cytoplasm, nuclei and inflammatory cell. w_i is the weighted decided by the background, cytoplasm, nuclei and inflammatory cell for mask m_i. $p \in C_K$ means that we sum all the pixel in one connected area. So, we will have one segmentation result in one connected area. For w_i, we train it through the loss of the training set. A more detailed training method is stated in experiment section.

We choose a stacking ensemble like model to do this because that we want to choose representative field adaptively for different cell size.

4 Evaluation

We use typical segmentation metrics to evaluate our result, which includes pixel accuracy, mean pixel accuracy and mean IoU. **Pixel accuracy** is a simple comparison for each pair of pixels in the predicted mask and the ground truth, which cannot tell the overall difference in performance since for many images there are large area of backgrounds, which are the easiest to be distinguished.

$$pixel\ accuracy = \frac{\sum_{i}^{M} n_{ii}}{\sum_{i}^{M}\sum_{i}^{M} n_{ij}},$$

Mean pixel accuracy uses similar methods as pixel accuracy, but an additional consideration of separating different classes by taking the mean of pixel accuracy for each class.

$$mean\ pixel\ accuracy = \frac{1}{M+1}\sum_{i}^{M}\frac{n_{ii}}{\sum_{j}^{M} n_{ij}},$$

IOU is applicable is segmentation task (Cordts et al., 2016). It is calculated by finding the number of intersection and union. In our project, we take the mean of IOU for each class and evaluates the results on the mean IOU.

$$mean\ IoU = \frac{1}{M+1}\sum_{i}^{M}\frac{n_{ii}}{\sum_{j}^{M} n_{ij} + \sum_{j}^{M} n_{ji} - n_{ii}}.$$

For the above three evaluation metrics, M is the number of class and n_{xy} represents the number of pixel classified to class y and its ground truth is x. Specially, in our final evaluation, as mentioned above, we eliminate the influence of background pixels that are easy to classify, we only consider the inflammatory cell, cytoplasm and the nuclei regions, therefore here $M = 3$.

5 Experiments

We use the state-of-art Pytorch framework and train different depths of U-Net structures including containing only 2, 3 and 4 downsample layers, which are called **2L U-Net**, **3L U-Net** and **4L U-Net** correspondingly in our project for simplicity. Our aggregated structure is mainly based on the network of these three models using different weights from the style classification. The modeling strategies can be found in the following subsections. What's more, we also trained a complete U-Net++ network for comparison of our model. The codes for the U-Net and U-Net++ models are majorly based on the Github links [1] and [2].

[1] https://github.com/milesial/Pytorch-UNet

[2] https://github.com/MrGiovanni/UNetPlusPlus

5.1 Data Set

Our data set contains both training set and testing set of cervical cancer cell images. The training images come from 2,000 samples captured from 4 large images of size 3,000×4,000 from different resources. Some samples are overlapped to make the same cell appears in different position, which is a way of data augmentation. However, in order to prevent there are too many overlapped regions, images with different cell sizes are divided and sampled. In other words, we have large-cell images, medium-cell images, and small-cell images in our training image set. Each training sample is a 776×776 size of image. For each training image, it has a corresponding annotated mask, which contains a black background, several red inflammatory cells, several normal cells with both green cytoplasm and blue nuclei, each mask is of size 500×500. As mentioned, the annotated masks reflect the central cropped parts of the corresponding original images.

For the test data, the components are the same as training set and the sample size of it is 50, and 20 for validation. The test set is cropped outside of the 4 large images of size 3,000×4,000 of training set so that there's not any overlapping between the training set and the test set.

5.2 Deep Framework & Machines

We use Pytorch as the major tool for training and testing since it provides well-wrapped deep learning network components. And in order to run our models smoothly and rapidly, we utilize Google Cloud Platform[3] technology with powerful GPUs(1 Nvidia tesla K80) to train and test our proposed network.

5.3 Style Classifier

This section explains the working of the Convolutional Neural Network (CNN) image classifier. The network consists of four convolutional layers, which are followed by max-pooling layers, and one fully-connected layers with a final over 10000-way softmax. In order to improve the results, We attempt various model structure such as the ReLU activation function, local normalization and dropout layers. The purpose of training algorithm is to train a network such that the error is minimized between the network output and the desired categories label.

[3] https://cloud.google.com/

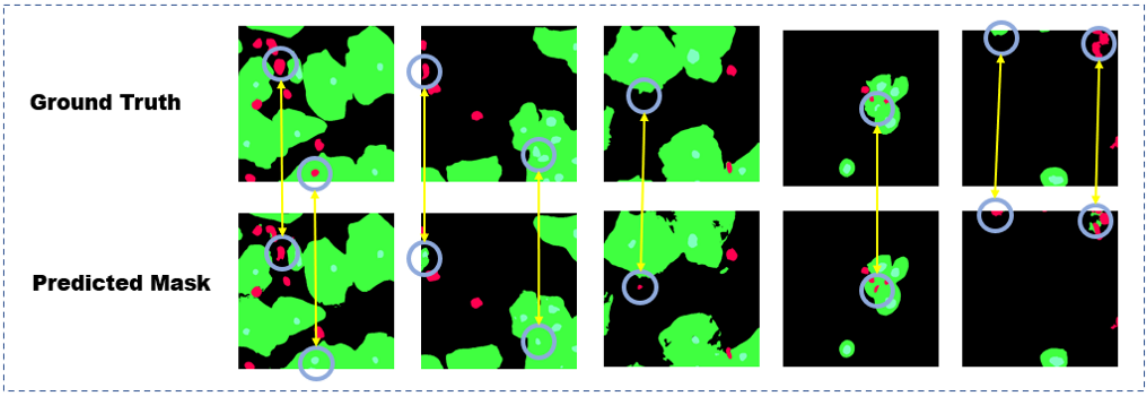

Figure 5: **Several resulted images for U-Net++ on our test set. Some obvious classified mistakes on cells can be observed, which indicate the limitation of U-Net++ on our data set. For each pair of comparison areas which are significantly different are marked by blue circles and corresponding circles are connected by a double-directional orange arrow.**

We use the Adam optimizer with a learning rate of 1e-3 and we trained the model for 3 epoch, with batch size of 64, all max pooling layers with pooling size of 2*2. Our model is build upon the popular tensorflow libraries and all models is train on google colab GPU.

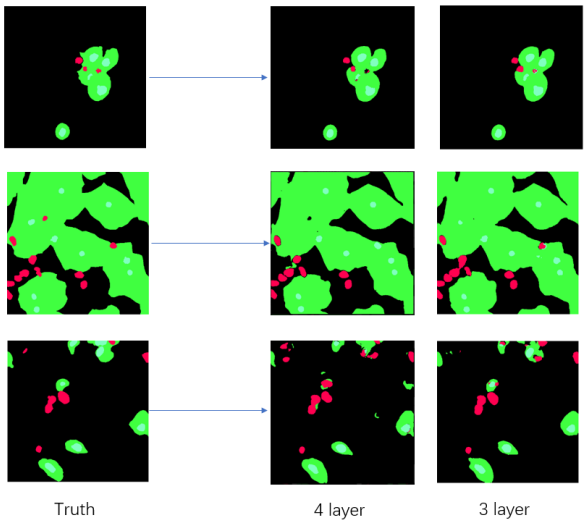

Figure 6: **Examples of images that have performance of a 3L U-Net structures that are better than 4L U-Net.**

5.3.1 U-Net Structures

For the U-Net implementation, as mentioned above, we perform the original modeling as well as two layer-reduction versions, 2L U-Net and 3L U-Net. First, for the original implementation, we just replicate the proposed U-Net model in the original paper (Ronneberger et al., 2015) on Pytorch from existing resources. We train our U-Net model for 30 epoch for all training samples with batch size 1.

Then, we also implemented the 2L U-Net and 3L U-Net by eliminating the last downside convolution and the first upside convolution layers, also the last two downside convolution and first two upside convolution layers, respectively. We assume that for some image, the new versions of U-Net may have better performances than the original model. So we modified the original U-net to change the layer of encoder and decoder from 4 layers to 3 and 2 layers. We use the updated 2L U-Net and 3L U-Net to train models with epoch 30 and batch size 1.

As we expected, the results of the 2L U-Net is much worse than the 4L U-Net, which is the complete U-Net structure. However, for the 3L U-Net, it achieves surprisingly good results, even outperform the 4L U-Net on our data set. A sample of three test images and the results from the two U-Net networks is shown in Figure 6. Particularly, by eliminating the the fourth horizontal level (including two layers) from the original U-Net model, for some images the 3L U-Net has better performance than the original 4L U-Net. From these figures, we can see that the 3L U-Net has better performance on classification small nuclei and inflammatory cells. Our assumption is that for some small objects we don't need so deep network as the original U-Net model. The experiment proves our assumption and can be an evident that our idea of using U-Net models with different layers on different image to improve accuracy can be feasible.

5.4 U-Net++ Implementation

For the more complex U-Net++ implementation, we perform the original model on full training set with with epoch 22 and batch size 1.

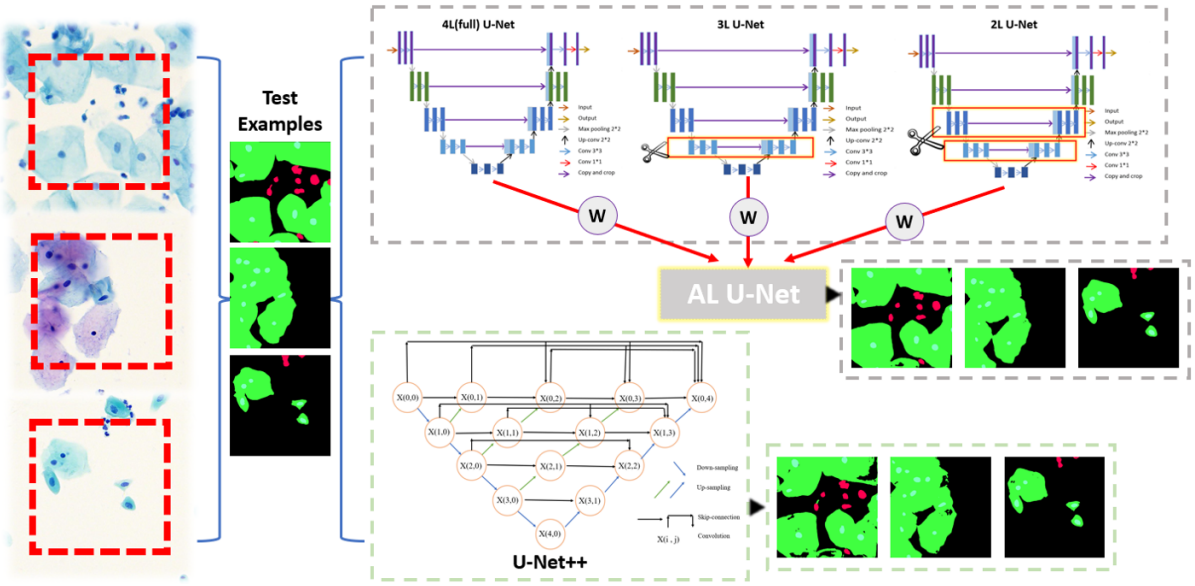

Figure 7: **An example of comparison between the performances of AL U-Net and U-Net++. The given three examples are sampled from large, medium and small classes of test images. The results show that the outputs of AL U-Net are very close to the ground truths while the outputs of U-Net++ have obvious specks and incorrectness.**

Figure 5 shows 5 examples of the predicted masks on the test set. We can see that for even for the U-Net++ structure, there exist several obvious specks, which are incorrect predictions on important cells on some images.

According to the results and comparisons, a general observation is that the overall segmentation borders are mostly correct while there exists wrong class predictions, especially the major differences lie in the predictions related inflammatory cells, which are the red areas in the figures. The mis-classification in U-Net++'s predictions occurs mostly in the way that inflammatory cells are incorrectly classified as nuclei cells and cytoplasm cells(red areas are incorrectly classified as light blue areas and green areas). In relatively rare cases, nuclei cells and cytoplasm cells are incorrectly classified as inflammatory cells (light blue areas and green areas are incorrectly classified as red areas). This incorrect classification related to inflammatory cells is exactly the problem we want to solve with our advanced model. One potential reason for the difficulty in correctly classifying inflammatory cells is that inflammatory cells are normally smaller and harder to distinguish even by human beings.

5.5 Weights Learning

While aggregating models, we need three 3-d parameters for images classified as small, middle and large. And for an image classified as certain size, we have three-dimension parameters for U-Net with two, three and four layers. We can learning these weights by train them on images of difference size in our train data set. And we assume that the parameter trained by our process can have better performance on the aggregating model.

We first divide our images in training data set into large, small and middle size. Then we can train the parameters for image of certain size on certain data set. During the process of training, in each iteration, we calculate the loss of a single image from three U-Net models. For the model with the lowest loss, we give it reward and for the model with the highest loss we give it penalty. For the reward, we multiple the weight with a number larger than 1 and for the penalty we multiple the weight with a number smaller than 1. The penalty and reward number are the hyper-parameters for us to do experiments on. In our final result we choose **1.01** as reward number and **0.99** as penalty number.

The result weight of our training process is shown in Table 5.4. From the final weight we can see that 3L U-Net has the highest weight on small and medium image and 4L U-Net has the highest weight on Large image which is agree with our experiment on the performance of each U-Net models on different sizes of images.

Metrics	Class	2L U-Net	3L U-Net	4L (full) U-Net	U-Net++	**AL U-Net**
pixel accuracy	Large	85.07	94.07	91.73	92.77	**95.10**
	Medium	79.41	89.73	87.32	88.50	**91.33**
	Small	69.79	78.02	74.50	76.91	**82.01**
	Total	81.24	90.54	87.99	89.22	**92.39**
Mean pixel accuracy	Large	51.15	60.15	58.43	61.33	**62.80**
	Medium	45.05	61.73	57.57	60.87	**63.89**
	Small	48.24	53.94	51.03	53.87	**57.80**
	Total	49.04	59.47	57.12	60.00	**62.33**
Mean IOU	Large	45.82	54.05	52.88	53.38	**56.61**
	Medium	40.29	58.36	54.24	57.19	**59.84**
	Small	37.00	42.22	38.75	40.21	**43.73**
	Total	42.98	53.15	51.05	52.08	**55.49**

Table 1: **The overall results of our AL U-Net and the combinations of different set of U-Net and the compared baseline U-Net++ model. For different kinds of metrics, AL U-Net outperforms other models for all sizes of classes and the overall test set. The results indicate the success of the proposed AL U-Net.**

Size, Layer number	2L U-Net	3L U-Net	4L (full) U-Net
Small	1.0793e-03	0.8387	0.1602
Medium	5.9392e-03	0.9912	2.8870e-03
Large	2.8971e-05	0.3633	0.6367

Table 2: **The three sets of trained weights from our learning.**

5.6 AL-UNet Implementation

AL-UNet is a combination of style classifier, U-Net and the weight. An aggregator get the result of classifier, three U-Net masks and the weight. With the help of opencv built-in connected area search function, aggregator does weighted vote for each connected area and get the final result mask.

6 Results

The results for different structured networks are acquired after feeding the test images. With the final results of our proposed AL U-Net. For the qualitative result, three examples are shown to compare the performances between AL U-Net and U-Net++. For the quantitative result, the numerical calculations are shown in Table 5.4.

Our proposed AL U-Net achieves very good performance on the test set, largely outperforming U-Net, and also goes beyond U-Net++. Since we have found the 3L U-Net can generally has better performance even than the 4L U-Net, we focus on the comparison of 3L U-Net, U-Net++ and our AL U-Net. For the total test set, AL U-Net can have an about **3**% increasing of mean pixel accuracy and mean IOU than 3L U-Net, and around **4**% increasing than U-Net++. For different classes of images, the advantages of AL U-Net are even clearer. The results are surprisingly good to us, but it reflects the truth that combining style classifier and assigning different weights based on several layered U-Nets can actually achieve a better result.

7 Future Work

In future, We will complete the weight update(sec 5.5) by gradient decent by training a 1*1 convolution layer. By doing this we can eliminate the manual parameter and make the method can be generalized to more dataset.

Moreover, we will carry out our method on another public dataset.

8 Conclusion

In this project, we focus on the biomedical task with cell segmentation from detecting clinical cervical cancer cells. We propose a model(AL-UNet) that is adapted from the current U-Net structure. We have performed several experiments and found some encouraging patterns on models. The U-Net models have good performance on our data set in our experiment. The comparison we did on U-Net with different layers can be an evidence that our idea of combining the results of U-Net with different layers to improve accuracy is feasible.The U-

Net++ model performs relatively well but didn't reach our expectation given its model complexity. We will adjust this model to improve its performance in the later experiment.

9 Contribution

Junwei Deng completes data collcetion, dataloader, basic pytorch implementation of U-Net and U-Net++, aggregator implementation, proposed method section in report and revise of other sections. **Yirui Gao** was mainly responsible for original 4L U-Net and 2L U-Net model training and testing part, with addition to this part in the report and several other sections of report. Also Yirui contibutes on implementing the metrics and all the evaluation jobs. **Shuwan Feng** contributes on debugging and training U-Net++ model on small dataset, focuses on implementing the the CNN image classifier and also accomplish part of U-Net++ implementation part, style classifier and some other parts in the report. **Dongjian Chen** focuses on modifying and testing the 3-layer U-Net and comparing its performance with original 4-layer U-Net, training the weight for aggregating models and some other parts in the report. **Yunzhe Jiang** is mainly responsible for part of U-Net++ implementation, training and testing on full data set, with addition to this part in the report and some other sections of the report.

References

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 3213–3223.

Junwei Deng, Yizhou Lu, and Jing Ke. 2020. An accurate neural network for cytologic whole-slide image analysis. In *Proceedings of the Australasian Computer Science Week Multiconference*. Association for Computing Machinery, New York, NY, USA, ACSW '20. https://doi.org/10.1145/3373017.3373039.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 770–778.

J. Ke, J. Deng, Y. Lu, D. Wang, Y. Song, and H. Zhang. 2019. Assessment and elimination of inflammatory cell: A machine learning approach in digital cytology. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*. pages 1–8. https://doi.org/10.1109/DICTA47822.2019.8946065.

Jing Ke, Junwei Deng, and Yizhou Lu. 2019a. Noise reduction with image inpainting: An application in clinical data diagnosis. In *ACM SIGGRAPH 2019 Posters*. Association for Computing Machinery, New York, NY, USA, SIGGRAPH '19. https://doi.org/10.1145/3306214.3338593.

Jing Ke, Zhaoming Jiang, Changchang Liu, Tomasz Bednarz, Arcot Sowmya, and Xiaoyao Liang. 2019b. Selective detection and segmentation of cervical cells. In *Proceedings of the 2019 11th International Conference on Bioinformatics and Biomedical Technology*. Association for Computing Machinery, New York, NY, USA, ICBBT'19, page 55–61. https://doi.org/10.1145/3340074.3340081.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2014. Fully convolutional networks for semantic segmentation. *CoRR* abs/1411.4038. http://arxiv.org/abs/1411.4038.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *CoRR* abs/1505.04597. http://arxiv.org/abs/1505.04597.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2018. Unet++: A nested u-net architecture for medical image segmentation. *CoRR* abs/1807.10165. http://arxiv.org/abs/1807.10165.