Evaluating and Optimizing Training and Interference Performances Between Variations of UNET Models for Nuclei Detection and Segmentation

Richard Dong[1,2,*]
r.dong@mail.utoronto.ca

Chris Xiao[1,3,*]
yl.xiao@mail.utoronto.ca

Sylvia Xu[1,2,*]
sylviaxiaoxiao.xu@mail.utoronto.ca

1. Department of Medical Biophysics, University of Toronto, Toronto, ON M5G 1L7, Canada

2. Princess Margaret Cancer Centre, University Health Network, Toronto, ON M5G 1L7, Canada

3. Sunnybrook Research Institute, Sunnybrook Health Sciences Centre, Toronto, ON M4N 3M5, Canada

* These authors contributed equally to the work

April 1st, 2024

Abstract

Cell morphology images taken by microscope are necessary for understanding cell behaviors and predicting cellular fates. However, it is often hard and time-consuming to manually identify cells from microscopy images before any other analysis can happen. Therefore, various approaches have been dedicated to developing a cell segmentation model that can streamline and automate this process. In this report, we aim to evaluate the performance of two popular machine learning architectures, UNET and UNETR, with various modifications such as the addition of a learning rate scheduler and the usage of different optimizers (Adam and SGD). Based on the testing result, we found that when the models are trained on a small dataset that is class imbalanced, the UNET model with an SGD optimizer was found to have the highest accuracy while also requiring low hardware resources for training. It is hoped that this finding can serve as a guideline for future researchers when they are designing a general-purpose cell segmentation model for their own dataset. The code for this project is available at https://github.com/YinniKun/mbp1413-final.

Keywords: machine learning, cell segmentation, computer vision, UNET, UNETR

Introduction

Cell morphology is used to visualize cell phenotype and cell activity [1]. It refers to the structural characteristics and features of cells, including size, shape, and internal organization [2]. In general, cell morphology shows the physical appearance and properties of cells as observed under the microscope or other imaging techniques [3]. In cell biology, it is important to understand cellular processes, functions, and behaviors through morphology [4]. For instance, stem cells change their morphology in a more visible and significant way when they get infected or have function changes [2]. Since cancer stem cells have a higher tendency for morphology changes due to the fact of cell-cell adhesion loss, cell motility induction, cytoskeleton alteration, pleomorphism, epithelial-mesenchymal transition, and cell polarity alterations [5], the morphology change of cancer cells is a good indication that they start to migrate as they separate from others, change their pathway, and try to develop genetic heterogeneity and undergo clonal evolution [2]. As a result, morphologies have been largely used in cell classification for medical diagnosis and personalized treatment [6].

Morphology determination is usually done with the help of micrographs. Various types of micrographs are classified by microscope features, including the light source, staining materials, resolution of the objective lenses, etc. [7]. The most commonly used microscopes for cell morphology visualization are light microscopes and fluorescence microscopes, as they can readily identify the edges of cells with staining [3]. The light and fluorescence microscopy uses a halogen lamp or LED as a light source, and the light passes through a condenser lens which directs the light to the specimen [8]. Under the microscope, the edge of the cell shows in grey color without stain, and therefore sometimes stains are used to better visualize the cell morphology by making them appear to be more clear and decent [3]. For instance, in H & E staining, giemsa staining, and methylene staining, the stains are negatively or positively charged so that they can bind on the cell membrane and help visualization [9]. However, the resolution of the cell is limited by visible light, which makes scientists change to the use of fluorescence microscopy, as a more detailed image can be produced with the light generated by the

attached fluorophores. There are several typical fluorescence dyes, including DAPI and phalloidins, which bind to cell nuclei and actin filaments respectively [10]. All of the microscopies mentioned above are state-of-the-art microscopies for cell morphology identification and can also be combined to better visualize the cellular morphology.

Even though microscopy images are often easily acquired so that a lot of data can be available, it is often hard to visualize and compare all cells manually to find their common characteristics when analyzing a big batch of images [11]. This is because the vague contrast between cell boundary and background due to illumination can lead to difficulties in cell segmentation and identification before morphological classification is even possible [11]. Therefore, developing a method for automated image segmentation has been an interest of biologists for a long time. Recently, machine learning (ML) has gained much attention thanks to the improvement in computing hardware [12]. ML is a subfield of artificial intelligence that enables computers to learn and make predictions based on the input data [13]. To obtain an ML model, firstly one has to train the model with a set of training data and then validate the reliability of the model with another set of testing data [12]. In general, there are two big categories of learning frameworks: supervised learning and unsupervised learning. Supervised learning requires the user to label the testing data beforehand [14], while unsupervised learning mainly clusters the dataset into different categories [15]. For cell image analysis, supervised ML is often used as the first step for cell segmentation [16]. Cell segmentation models require two noncontinuous variables, ”cell” and ”not cell”, attached to each pixel of the original image. During training, the labeled data will be used for the machine to learn about the two different variables, and eventually be able to segment the cell from micrographs [16].

To perform this task better, a convolutional neural network (CNN) is one of the ways to aid cell segmentation as it can learn more local information while also reducing the number of required parameters so that the learning process is more efficient [17]. In general, the input for CNN is an image or grid-like structure, and each pixel is represented as a numeric value as greyscale or RGB/RGBA [18]. In the middle of the

pipeline, there are several convolutional layers as building blocks which are made up of convolutional kernels that cut the image into pieces and filter slides that multiply the local image into a feature map [19]. After the convolution operation, an activation function is applied element-wise to involve non-linearity and help to capture features in the image [20]. The feature maps will be further reduced with the pooling layer but the important information remains, and the feature maps will be flattened into one vector in the fully connected layer [17]. This layer performs the learning with high-level reasoning and decision-making [21]. After learning, the output layer can represent the predicted values, and the network is tested on new input data to evaluate its performance.

UNET is one of the innovative CNN architectures created to tackle semantic segmentation [22], which provides highly effective solutions for precise pixel-level segmentation, particularly in the imaging field. It is piped by Ronneberger et al. and utilizes an encoder-decoder architecture with skip connections [23]. This addition makes the segmentation precise even with relatively little training data. The encoder part identifies important and relevant features from the original image to condense the information. After condensation, the decoder part builds a segmentation mask based on the features and recovers the original image with required characterizations [24]. One problem of this network is that the condensation of information might lead to a loss in features and details [25]. Therefore, UNET also includes skip connections, which concatenate the decoder to its corresponding encoder [22]. This feature would allow UNET to ”remember” more details than the encoder-decoder structure, therefore increasing segmentation accuracy. Based on the UNET architecture, UNET transformer (UNETR) was proposed later by Ben-Cohen et al., which uses a transformer-based approach that enhances the scalability of the original UNET [26]. The transformer layer is a convolutional layer in the skip connection where more details of the original image are processed with an attention mechanism [25]. Other UNET architectures can be used for cell segmentation, but UNET and UNETR are two of the most commonly used ones for cell segmentation.

Despite that UNET and UNETR models have had many successes in cell segmen-

tation, ultimately each task is different and people often need to modify and retrain a model based on their data. To further increase the efficiency and scalability during training, training parameters should also be considered. Learning rate schedulers and optimizers are two major additional things to further optimize model training. The learning rate scheduler can adjust and optimize the learning rate by calculating the loss value step-wise [27]. During the training process, the learning rate scheduler changes the learning rate based on the schedule, which can be set to after a certain number of epochs or iterations. This helps to change the learning rate dynamically to adjust to the appropriate loss value and convergence speed [27]. On the other hand, optimizers adjust the model parameter to reduce losses, and is key to how the model learns [28]. The training of the model starts from initial preset parameters, which will be further adjusted to minimize the loss function. Some of the commonly used optimizers are Adam [29] and Stochastic Gradient Descent (SGD) [30], which are an adaptive optimization algorithm and a basic form of gradient descent for loss function respectively.

To facilitate the design of the most optimal model for cell segmentation, in this report, we evaluated the cell segmentation performance of UNET and UNETR training with the same set of microscope data. To further optimize both models, we compared the performance of models with and without the use of a learning rate scheduler and models with Adam or SGD. It is hoped that our findings can provide a practical guide for choosing the most suited model architecture for future researchers who wish to accomplish such tasks.

Results

The choice of data allows the model to be compatible with a diverse array of modalities

The dataset we used to train, test, and build our model comes from the *Kaggle 2018 Data Science Bowl: Find the nuclei in divergent images to advance medical discovery* (see **Materials and Methods** for a more detailed description). This set of data is

very diverse and contains a variety of cellular imaging modalities such as bright field microscopy, dark field microscopy, and fluorescence microscopy (**Figure 1A**), where some images are stained and some are non-stained. Those modalities capture various cell densities, cell types, cell morphologies, imaging coloring schemes, and resolutions. This diverse dataset was designed to allow the model to be trained to be generalizable across modalities and cell states to be deployed in various settings.

UNETR requires significantly more resources for training

Hardware availability is one of the key factors for researchers to consider when designing and training a deep learning model as computational resources are not infinite [31]. Generally speaking, the more parameters a model has, the heavier its requirements are on hardwares such as GPU and GPU memory for efficient training. Here we compared the average training time for running 200 epochs of training with various conditions for both UNET and UNETR (see **Materials and Methods** for details on model building and training). It was found that UNETR models take significantly more time ($p = 9 \times 10^{-25}$, Student-T test, **Figure 1B**) when compared to UNET models. This is primarily because of the heavier computation required by the multi-head attention mechanisms that are found in UNETR models when compared to the regular UNET models that only employ simple convolutional layers and concatenations.

SGD is a better optimizer when compared to Adam for this task

We started by experimenting with different optimizers, in particular - SGD and Adam, when training the model. For UNET models, it was found that models that were trained with Adam converge faster, evident from the loss curves that stabilize faster (**Figure 2 a and e**), while SGD provides minimal improvement for the performance on testing inference (**Table 1**). For UNETR models, the use of SGD significantly smooths out the loss curves (**Figure 3 a and e**) and provides a much greater improvement for the performance on testing inference (**Table 1**).

Despite that both SGD and Adam are gradient-based algorithms for parameter op-

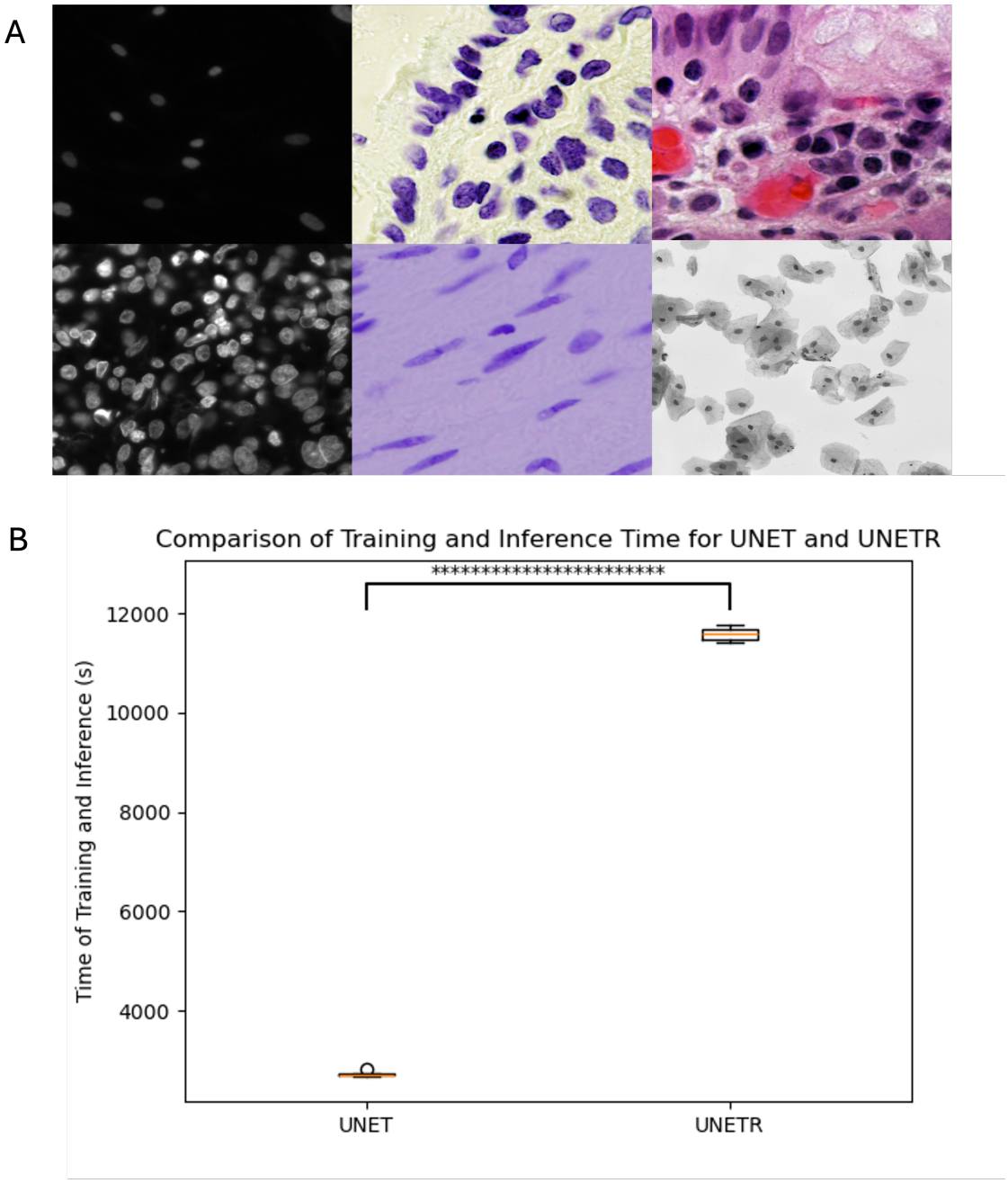

Figure 1: An overview of the dataset used in the study and the training performance. A) A selection of the images from the dataset, where a variety of imaging techniques with many different image characteristics are present. B) A comparison between the training and inference time required for UNET and UNETR. The mean training and inference times for UNET and UNETR models are 2,721 seconds and 11,582 seconds respectively, with $p = 9 \times 10^{-25}$ by Student-T test.

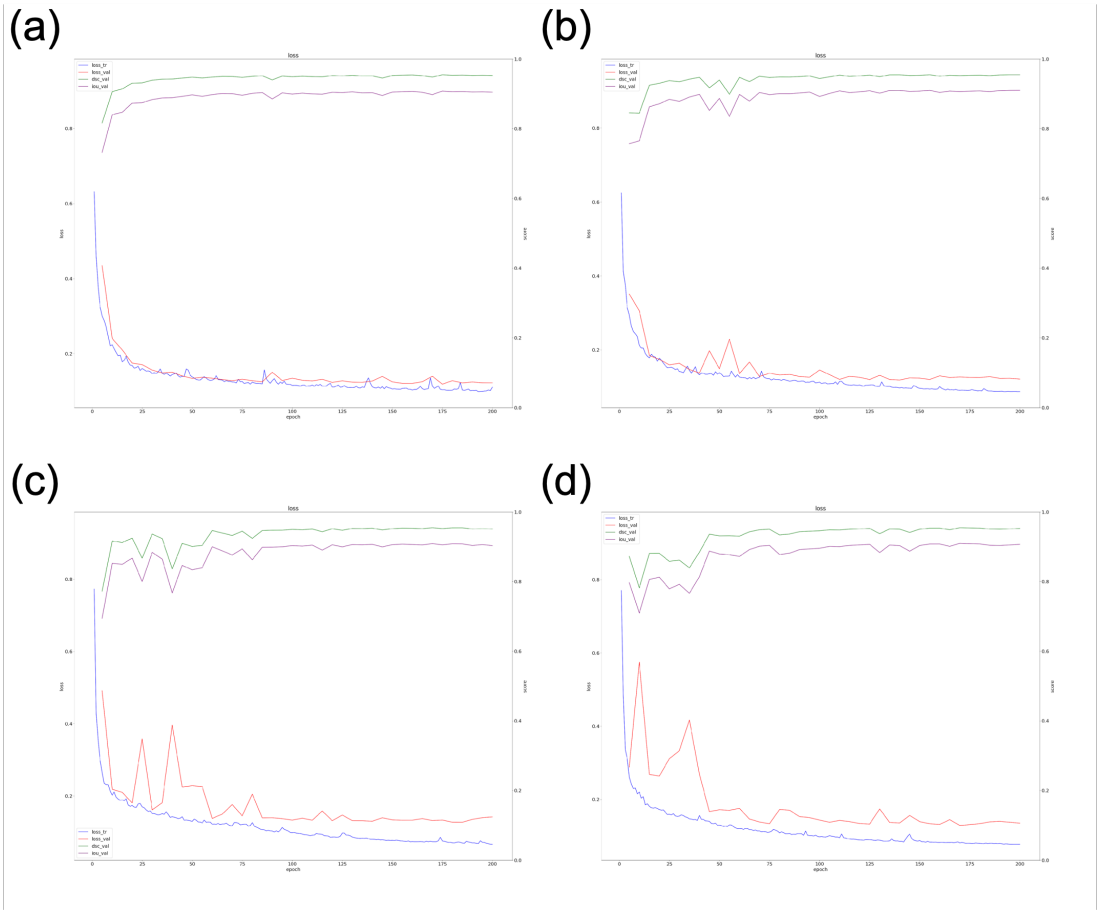

Figure 2: Loss and validation curves of various UNET models during training and validation for each epoch. Each plot represents the loss curve of a model with one particular combination of training parameters, including (a) the model with an Adam optimizer only, (b) the model with an Adam optimizer and a learning rate scheduler, (c) the model with an SGD optimizer only, and (d) the model with an SGD optimizer and a learning rate scheduler. In each figure, the blue line is the training loss, the red line is the validation loss, the green line is the validation Dice score, and the purple line is the validation IoU score.

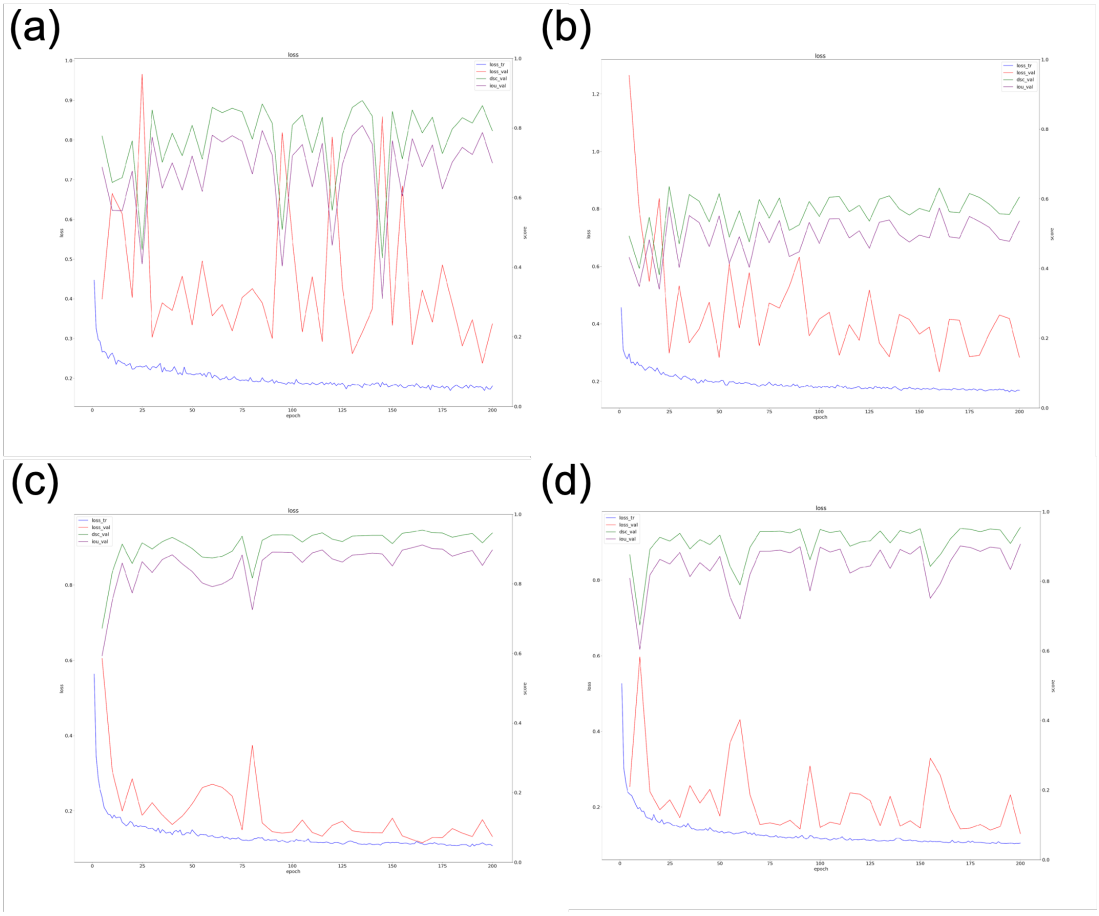

Figure 3: Loss and validation curves of various UNETR models during training and validation for each epoch. Each plot represents the loss curve of a model with one particular combination of training parameters, including (a) the model with an Adam optimizer only, (b) the model with an Adam optimizer and a learning rate scheduler, (c) the model with an SGD optimizer only, and (d) the model with an SGD optimizer and a learning rate scheduler. In each figure, the blue line is the training loss, the red line is the validation loss, the green line is the validation Dice score, and the purple line is the validation IoU score.

Table 1: Summary of the performance metrics (average Dice and IoU scores) of variations of UNET and UNETR models on the testing dataset. Sche indicates that the model used a learning rate scheduler during training.

		Adam	Adam + Sche	SGD	SGD + Sche
UNET	Dice	0.82	0.81	0.82	0.82
	IoU	0.73	0.72	0.73	0.71
UNETR	Dice	0.72	0.73	0.79	0.78
	IoU	0.59	0.60	0.68	0.68

timization, the main difference between Adam and SGD is that Adam is an adaptive algorithm that optimizes the learning rate for each parameter in the model individually based on the first and second moments of the gradients [32]. This kind of algorithm architecture makes Adam robust to variances in initial parameters to allow faster convergence during training when compared to models using SGD as the optimizer [33], yet some studies found that this faster optimization can cause the model to not be very generalizable [34, 35]. This phenomenon is observed in our UNET models, where a faster convergence in training for Adam models does not lead to an increase in performance in testing, potentially due to overfitting (**Table 1**). On the other hand, we hypothesize that the small sample size for training makes the first and second moments of the gradients to be noisy, which might explain why SGD performs significantly better for UNETR models.

Using a learning rate scheduler has limited and even negative effects on performance improvement

We then investigated the effect of using a learning rate scheduler. In general, the use of learning rate scheduler seems to have minimal if not negative effect on the performance of both UNET and UNETR models, as evident in both the training loss curves (**Figure 2 b and d** and **Figure 3 b and d**) and the testing results (**Table 1**).

This is surprising as learning rate schedulers are known to reduce overfitting and therefore improve model performance [36]. We hypothesize the reason for this minimal and potential negative improvement is that the learning rate scheduler is not tailored

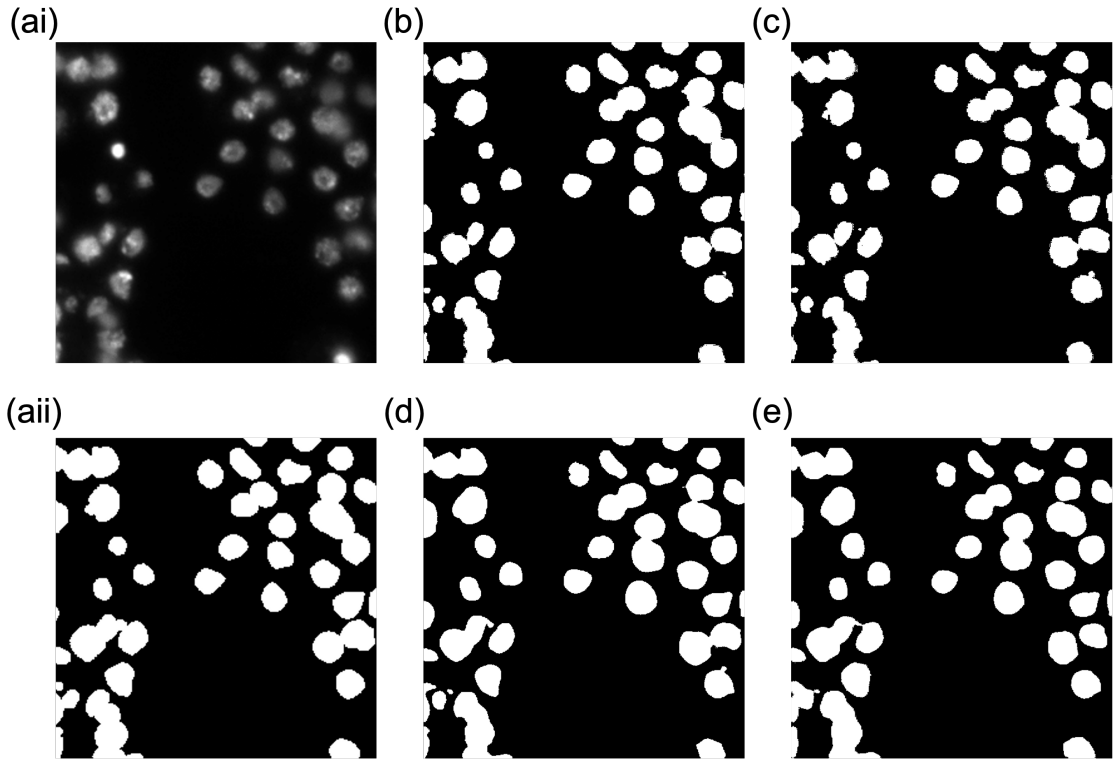

Figure 4: Representative testing results of non-stained images by various UNET models. (ai) The original micrograph that was used as input. (aii) The manually annotated mask that was used as the ground truth. (b) Masks that are generated with models using an Adam optimizer only. (c) Masks that are generated with an Adam optimizer and a learning rate scheduler. (d) Masks generated by models with an SGD optimizer only. (e) Masks generated by models with an SGD optimizer and a learning rate scheduler.

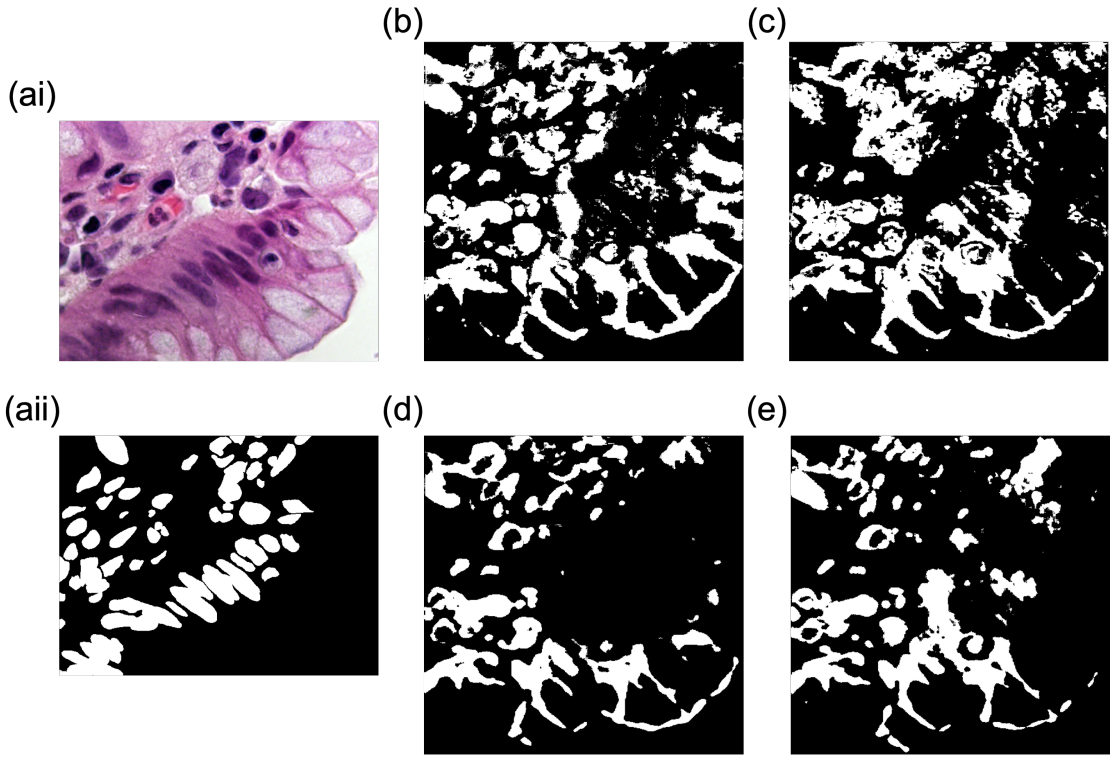

Figure 5: Representative testing results of stained images by various UNET models. (ai) The original micrograph that was used as input. (aii) The manually annotated mask that was used as the ground truth. (b) Masks that are generated with models using an Adam optimizer only. (c) Masks that are generated with an Adam optimizer and a learning rate scheduler. (d) Masks generated by models with an SGD optimizer only. (e) Masks generated by models with an SGD optimizer and a learning rate scheduler.

enough for the dataset. Studies have shown that the choice of the learning rate scheduler should depend on the characteristics of the dataset to maximize its benefit [37]. For example, if the dataset has a class imbalance, a more aggressive learning rate scheduler should be used to prevent the model from favoring the majority class. Since in the training dataset, there are many more non-stained images than stained ones, the polynomial learning rate scheduler we used in our models might not be enough to overcome this imbalance to increase generalizability. This is evident from the much worse performance in the inference of stained testing images (**Figure 5** and **Figure 7**) when compared to the performance of inference on non-stained testing images (**Figure 4** and **Figure 6**) for both the UNET and UNETR models. Therefore, when choosing the learning rate scheduler for training models, it is important to first examine the characteristics of the dataset in order to pick the most optimal scheduler for better performance.

Performance of UNET models are generally better across UNETR models

We finally compared the overall performance of UNET and UNETR models. Regardless of the optimizer and whether a learning rate scheduler is used, it is evident that UNETR models perform worse than UNET models on both the testing dataset (**Table 1**) and the training dataset, with the UNETR models trained on Adam not even having stable loss curves (**Figure 1 a and c**). The primary reason for that is that studies have found that vision transformer-based models (such as UNETR with its attention-based encoder) suffer from performance issues on small datasets. This is because when the dataset is small, vision transformers often fail to learn the local features, despite that those features are learnable when the dataset becomes larger [38]. Since the training data is relatively small with only 670 images (see **Materials and Methods** for details on the dataset), transformer-based models such as UNETR might not be the best candidate for this segmentation task given this dataset.

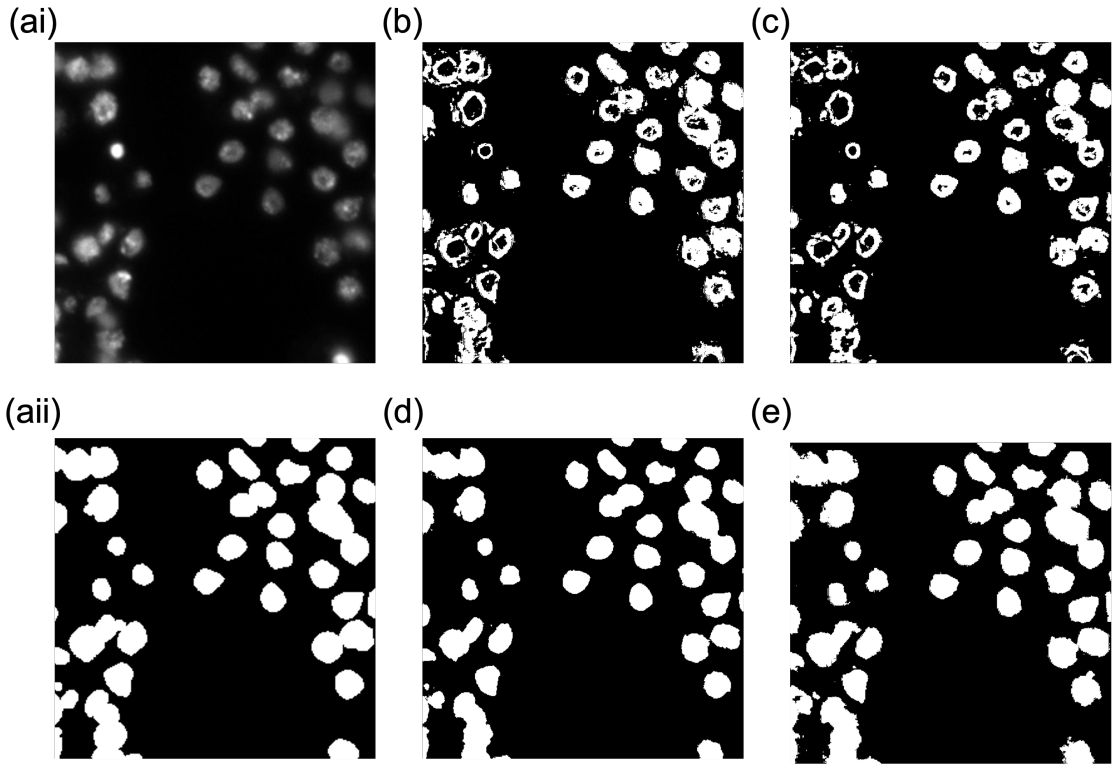

Figure 6: Representative testing results of non-stained images by various UNETR models. (ai) The original micrograph that was used as input. (aii) The manually annotated mask that was used as the ground truth. (b) Masks that are generated with models using an Adam optimizer only. (c) Masks that are generated with an Adam optimizer and a learning rate scheduler. (d) Masks generated by models with an SGD optimizer only. (e) Masks generated by models with an SGD optimizer and a learning rate scheduler.

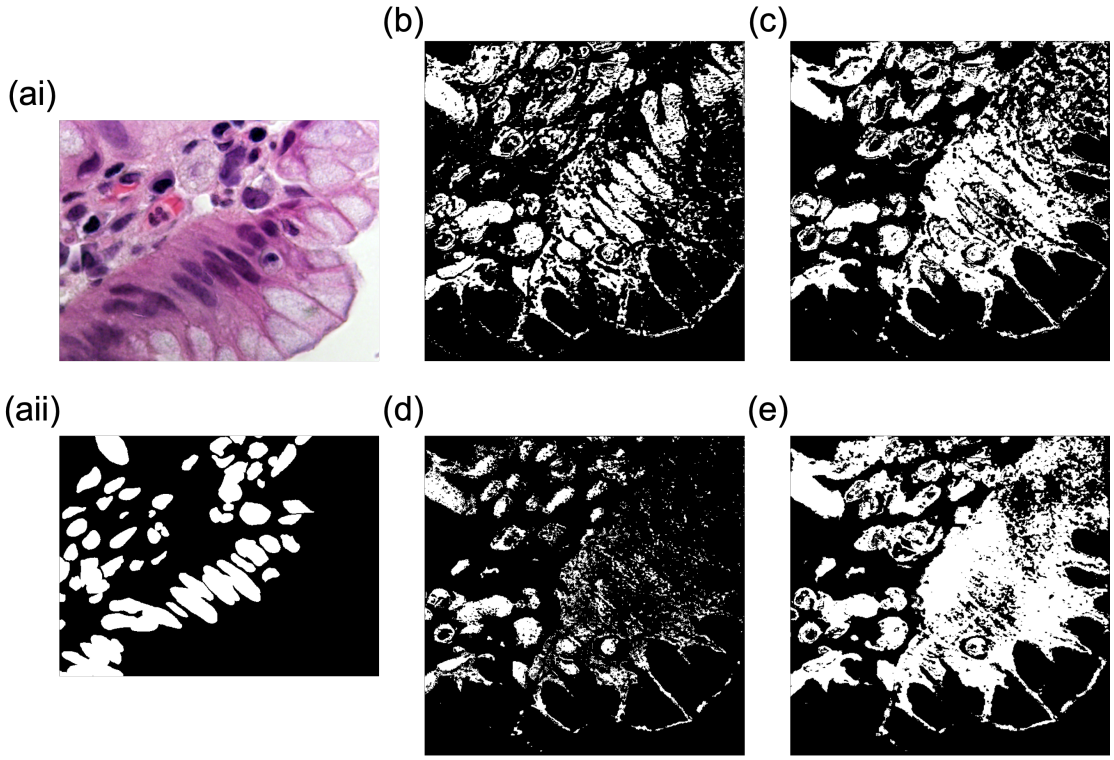

Figure 7: Representative testing results of stained images by various UNETR models. (ai) The original micrograph that was used as input. (aii) The manually annotated mask that was used as the ground truth. (b) Masks that are generated with models using an Adam optimizer only. (c) Masks that are generated with an Adam optimizer and a learning rate scheduler. (d) Masks generated by models with an SGD optimizer only. (e) Masks generated by models with an SGD optimizer and a learning rate scheduler.

Discussion and Conclusion

In this study, we compared the performance of UNET and UNETR with different optimizers and learning rate schedulers for cell segmentation. It was found that given this small and imbalanced dataset that largely favors non-stained images, using a UNET model with SGD as an optimizer can outperform other combinations of the model training architecture while still requiring relatively low training hardware resources. To build a model that best suits cell segmentation tasks, future researchers should ensure that they have a large and balanced dataset for training to reach the best outcome. It is hoped that with the advice provided by this study, better cell segmentation models can be developed in the near future so that cell morphology identification and disease diagnosis can be facilitated and optimized.

Materials and Methods

Data and Unprocessed Results Availability

The training and validation datasets and unprocessed results generated from the study can be found at this Google Drive. In short, 670 image and mask pairs were used during training and 64 image and mask pairs were used during testing.

UNET and UNETR Model Description

The input data and labels were preprocessed by first resizing to a dimension of (4, 512, 512) as RGBA images to correct for the inconsistency in the dimension of the raw images. They were then processed by pixel intensity scaling for normalization. Both the UNET and UNETR models were built with 4 input channels and 2 output channels with channel sizes of (16, 32, 64, 128, 256, 512). For convolutional layers, the stride size is 2, the kernel size is 3, and the number of residual units is set to 5. Leaky ReLU is used as the activation function [39] and batch normalization is used [40] with a batch size of 5.

Both the UNET and UNETR models were built with the MONAI Python library (version 1.3.0) and the PyTorch Python library (version 2.2.0).

Training and Validation Details

All models were trained and validated on one NVIDIA V100 Volta GPU on the Graham Cluster on Digital Research Alliance of Canada (previously known as Compute Canada), where the training set was used with an 80/20 train/validation split and a separate test set was used. The training was done with a learning rate of 0.005 for 200 epochs, with an objective function that sums both the Dice loss and Focal loss. Various combinations with or without a polynomial learning rate scheduler and an optimizer of either SGD (with momentum of 0.99) or Adam with a dropout ratio of 0.2 were tested to optimize the performance on inference. Dice and IoU scores were computed for the validation and testing sets of each model for every 5 epochs.

Acknowledgement

The authors would like to thank all instructors of *MBP1413H Winter 2024* for their well-prepared lectures on the background of machine learning and the opportunity to complete this project. The authors would also like to thank the TA of *MBP1413H Winter 2024*, Mr. Ahmadreza Attarpour, for the useful tutorial sessions, as well as Dr. Gregory Schwartz from Princess Margaret Cancer Centre for providing access to the Cluster for running all the computations required for this project.

References

1. Carpenter, A. E. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology* **7,** 1–11 (2006).
2. Baba, A. I. & Câtoi, C. in *Comparative oncology* (The Publishing House of the Romanian Academy, 2007).
3. Alberts, B. *et al.* in *Molecular Biology of the Cell. 4th edition* (Garland Science, 2002).
4. Li, Y., Tang, W. & Guo, M. The cell as matter: Connecting molecular biology to cellular functions. *Matter* **4,** 1863–1891 (2021).
5. Kim, D. H. *et al.* Epithelial mesenchymal transition in embryonic development, tissue repair and cancer: a comprehensive overview. *Journal of clinical medicine* **7,** 1 (2017).
6. Rivenbark, A. G., O'Connor, S. M. & Coleman, W. B. Molecular and cellular heterogeneity in breast cancer: challenges for personalized medicine. *The American journal of pathology* **183,** 1113–1124 (2013).
7. Murray, R. & Robinow, C. F. Light microscopy. *Methods for General and Molecular Microbiology,* 5–18 (2007).
8. Yamagata, K. *et al.* Fluorescence cell imaging and manipulation using conventional halogen lamp microscopy. *PLoS One* **7,** e31638 (2012).
9. Alkhamiss, A. S. Evaluation of better staining method among hematoxylin and eosin, Giemsa and periodic acid Schiff-Alcian blue for the detection of Helicobacter pylori in gastric biopsies. *The Malaysian journal of medical sciences: MJMS* **27,** 53 (2020).
10. Takata, K. & Hirano, H. Use of fluorescein-phalloidin and DAPI as a counterstain for immunofluorescence microscopic studies with semithin frozen sections. *Acta histochemica et cytochemica* **23,** 679–683 (1990).

11. Sailem, H. Z., Cooper, S. & Bakal, C. Visualizing quantitative microscopy data: History and challenges. *Critical reviews in biochemistry and molecular biology* **51,** 96–101 (2016).

12. Jordan, M. I. & Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* **349,** 255–260 (2015).

13. Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A. & De Felice, F. Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability* **12,** 492 (2020).

14. Cunningham, P., Cord, M. & Delany, S. J. in *Machine learning techniques for multimedia: case studies on organization and retrieval* 21–49 (Springer, 2008).

15. Celebi, M. E. & Aydin, K. *Unsupervised learning algorithms* (Springer, 2016).

16. Kan, A. Machine learning applications in cell image analysis. *Immunology and cell biology* **95,** 525–530 (2017).

17. Kattenborn, T., Leitloff, J., Schiefer, F. & Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing* **173,** 24–49 (2021).

18. Ihor, S. *Malware Detection Using Visualization Techniques* MA thesis (České vysoké učení technické v Praze. Vypočetní a informační centrum., 2024).

19. Albawi, S., Mohammed, T. A. & Al-Zawi, S. *Understanding of a convolutional neural network* in *2017 international conference on engineering and technology (ICET)* (2017), 1–6.

20. Hao, W., Yizhou, W., Yaqin, L. & Zhili, S. *The role of activation function in CNN* in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)* (2020), 429–432.

21. Basha, S. S., Dubey, S. R., Pulabaigari, V. & Mukherjee, S. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing* **378,** 112–119 (2020).

22. Huang, H. *et al. Unet 3+: A full-scale connected unet for medical image segmentation* in *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2020), 1055–1059.

23. Ronneberger, O., Fischer, P. & Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* 2015. arXiv: 1505.04597 [cs.CV].

24. Kazerouni, I. A., Dooly, G. & Toal, D. Ghost-UNet: an asymmetric encoder-decoder architecture for semantic segmentation from scratch. *IEEE Access* **9,** 97457–97465 (2021).

25. Sha, Y., Zhang, Y., Ji, X. & Hu, L. Transformer-unet: Raw image processing with unet. *arXiv preprint arXiv:2109.08417* (2021).

26. Hatamizadeh, A. *et al. UNETR: Transformers for 3D Medical Image Segmentation* 2021. arXiv: 2103.10504 [eess.IV].

27. Wen, L., Gao, L., Li, X. & Zeng, B. Convolutional neural network with automatic learning rate scheduler for fault classification. *IEEE Transactions on Instrumentation and Measurement* **70,** 1–12 (2021).

28. Yaqub, M. *et al.* State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. *Brain sciences* **10,** 427 (2020).

29. Ogundokun, R. O., Maskeliunas, R., Misra, S. & Damaševičius, R. *Improved CNN based on batch normalization and adam optimizer* in *International Conference on Computational Science and Its Applications* (2022), 593–604.

30. Vrbančič, G. & Podgorelec, V. Efficient ensemble for image-based identification of Pneumonia utilizing deep CNN and SGD with warm restarts. *Expert Systems with Applications* **187,** 115834 (2022).

31. Sarker, I. H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and Research Directions. *SN Computer Science* **2** (2021).

32. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* 2017. arXiv: 1412.6980 [cs.LG].

33. Pan, Y. & Li, Y. *Toward Understanding Why Adam Converges Faster Than SGD for Transformers* 2023. arXiv: 2306.00204 [cs.LG].

34. Wilson, A. C., Roelofs, R., Stern, M., Srebro, N. & Recht, B. *The Marginal Value of Adaptive Gradient Methods in Machine Learning* 2018. arXiv: 1705.08292 [stat.ML].

35. Keskar, N. S. & Socher, R. *Improving Generalization Performance by Switching from Adam to SGD* 2017. arXiv: 1712.07628 [cs.LG].

36. Kim, C., Kim, S., Kim, J., Lee, D. & Kim, S. *Automated Learning Rate Scheduler for Large-batch Training* 2021. arXiv: 2107.05855 [cs.LG].

37. Wu, Y. *et al. Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks* 2019. arXiv: 1908.06477 [cs.LG].

38. Zhu, H., Chen, B. & Yang, C. *Understanding Why ViT Trains Badly on Small Datasets: An Intuitive Perspective* 2023. arXiv: 2302.03751 [cs.CV].

39. Xu, B., Wang, N., Chen, T. & Li, M. *Empirical Evaluation of Rectified Activations in Convolutional Network* 2015. arXiv: 1505.00853 [cs.LG].

40. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* 2015. arXiv: 1502.03167 [cs.LG].