Nuclei Detection in Colon Cancer Histology Images Using U-Nets with TensorFlow

Piper Williams, Charlie Carpenter

Abstract—Today, accurate diagnosis of colon cancer remains heavily reliant on manual examination of H&E- stained microscopic images of collected tissue by an expert pathologist. This process remains somewhat subjective depending on the pathologist's experience and is an extremely time-consuming process. In an attempt to circumvent these issues, various machine learning and deep learning methods have been proposed and developed in the recent years. In particular, the U-Net, a fully convolutional network, was created specifically for biomedical image segmentation. The goal of this project was to train a U-Net for semantic segmentation of colon cancer histology images via TensorFlow and evaluate classification performance via precision, recall, and F1-score in the presence of heavily-imbalanced data. We wanted to highlight the effects of varying dropout rates and validation batch sizes on classification performance as well. There were numerous limitations to this project that largely stemmed from the imbalanced data, and significant improvements such as more rigorous data augmentation and resampling techniques should be further explored in future research. Regardless, this project showcases the potential for fully convolutional networks, such as the U-Net, to aid in the accurate and timely detection of colon cancer.

Index Terms—Convolutional Neural Network; U-Net; TensorFlow; Computer Vision; Deep Learning; Colon Cancer; Histology Images

1 Introduction

Colorectal cancer is the third-most prevalent cancer diagnosis and the fourth-most prevalent cancer-related death worldwide [1]. It is routine to process colon biopsy tissue samples by staining them with hematoxylin and eosin (H&E) when colon cancer is a likely diagnosis. These H&E stains help highlight key structures within the tissue samples, such as cell nuclei. The current gold standard is a trained pathologist's diagnosis of these H&E-stained tissue samples based on their visual examination of the nuclei color, shape, and other features within the tissue in order to gain a qualitative and quantitative assessment of the sample. This requires a deep understanding of several biological characteristics such as cell morphology and cell protein markers. Accuracy of any diagnosis is subject to the pathologist's experience. In addition, this process is extremely time consuming, and these challenges are compounded by the high prevalence of colon cancer. The combination of all of these factors can lead to diagnostic errors, which can ultimately result in incorrect treatment decisions.

Developing imaging software that can reliably detect and count nuclei within these H&E-stained tissue samples has the potential to significantly cut down on time. In addition, this software could also reduce diagnostic errors by preventing pathologists from missing nuclei due to cell clusters and overlap, poor image quality, or poor staining quality.

Many machine learning techniques are shallow in that their learning algorithm doesn't contain many layers. Machine learning methods often require some features to be designed by hand to work along with linear classifiers in order to be effective. Creating these hand-crafted features can be a demanding task and call for an expert-level understanding of the underlying biology as well as advanced skills in computer engineering. This project aims to create a U-Net trained to detect nuclei in three-channel images of H&E-stained colorectal tissue samples. Deep learning models' classification performance can be impacted by many different hyperparameter settings within the training stage. In this project, we also explore the impact of the training dropout rate and validation batch sizes on accuracy, precision, recall, and F1-score.

2 Related Works

This project uses a subset from Sirinukunwattana et al.'s [2] research using locally sensitive deep learning for semantic segmentation. Nuclei detection is a relatively common task in the field of computer vision. Papers by Rogojanu et al. [3], Nawandhar et al. [4], Joon Ho et al. [5], and Cui et al. [6] all focus on nuclei segmentation. Their methods include convolutional neural networks, thresholding, and other deep learning techniques. U-Nets are also rather

- *Piper Williams is with the Department of Biostatistics and Informatics, University of Colorado, Denver, School of Public Health*
- *Charlie Carpenter is with the Department of Biostatistics and Informatics, University of Colorado, Denver, School of Public Health*

common in biomedical image segmentation papers. However most authors, such as Sevastopolsky et al. [7], make adjustments to the method to best suit the problem at hand.

3 Data

The data provided included 30 three-channel images (500x500 in dimension) from the original batch of 100 H&E-stained tissue sample images from Sirinukunwattana et al., along with the images' corresponding annotation files. Four subtypes of nuclei were annotated by hand for each of the images.

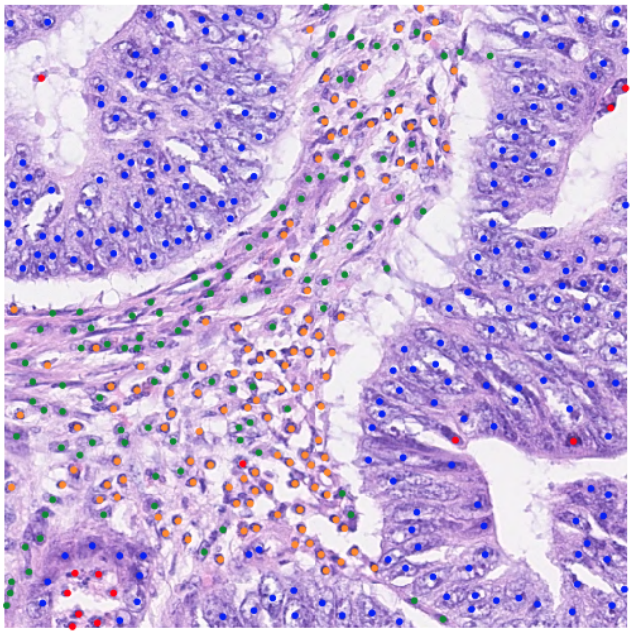

Fig. 1. Example of one of the original training images with nuclei labels designated from annotation files. Blue: epithelial cells. Green: fibroblast cells. Yellow: inflammatory cells. Red: "other" cells.

Exploratory data analysis revealed that each of the four cell subtypes made up less than 2% of the entire training data set. Due to the sparsity of the four subtypes of nuclei, the objective of this paper was changed to pixelwise binary classification of the histology images.

4 Methods

All data preparation and analysis was completed using select libraries from Python3 [8] and TensorFlow [9]. Select graphics were created using the *ggplot2* package in R-Studio [10].

4.1 Image Preprocessing and Data Augmentation

The 30 images were split 50/50 into a training/validation set and a testing set. The corresponding annotation files included both the pixel-level X-Y coordinates of the center of the nuclei as well as the label of the nuclei which indicated the cell subtype. Annotation images were then generated using the coordinates and labels from the annotation files. The single-pixel, labeled images were then dilated using a 7x7 kernel. This enlarged the single-pixel centers to 7x7 squares, which more accurately reflected the size of the nuclei. The dilated annotation images served as the final annotation masks and ground truth for the training phase.

The 15 training images in the training/validation set and their corresponding annotation files were each randomly cropped four times to 350x350 images. This relatively large cropping window was chosen due to the sparsity of nuclei in some of the training images. Each cropped image was then rotated 0°, 90°, 180°, or 270°. This gave us a total of 60 images to train and validate our model. These steps were taken in order to increase our training/validation set size as well as introduce more variation in the training/validation set. In theory, this will help create a more generalizable model and guard against overfitting the U-Net to the training data.

4.2 U-Nets

It is well understood that deep learning networks require large, annotated data sets. This is often a major challenge within the biomedical image domain, since imaging and creating the annotations are both costly and time-consuming. The biomedical imaging field often relies heavily on data augmentation techniques, such and random cropping and rotations, to boost the amount of data within the training set.

U-Nets were originally developed by Olaf Ronneberger et al. [11] with biomedical image segmentation specifically in mind. Their end-to-end, fully convolutional structure allows for any image size to be used for training and testing. It was designed with data augmentation in mind which also allows for a relatively small training and validation set. These properties have made U-Nets rather popular in biomedical image segmentation problems and other related computer vision problems.

The basic architecture follows a symmetric path. First, the U-Net consists of down-sampling through convolution layers, rectified linear unit (ReLU) activation functions, and max pooling layers. It is then followed with up-sampling through transposed convolution (or "deconvolution") layers. After each up-sampling step, cropped convolutional layers from the opposite side of the "U" are copied onto the current layer before further convolution is performed. Finally, a pixelwise image segmentation map is output. A visual of this architecture from the original publication is shown below in Figure 2.

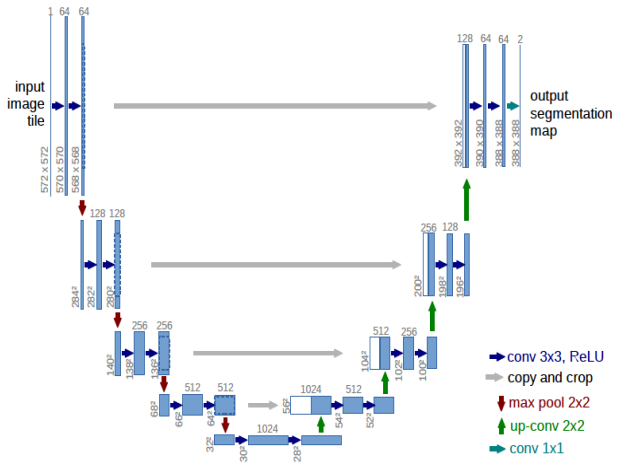

Fig. 2. The basic architecture of a U-Net. Blue boxes represent a multi-channel feature map. The number of channels (for this example) is indicated by the number above the box. White boxed represent cropped and copied feature maps from previous stages. Different operations are indicated by different colored arrows.

Each U-Net for this project was created using the *tf_unet* [12] package available through TensorFlow. The predicted probabilities for each pixel ranged from 0 to 1, 0 being background and 1 being foreground. Predicted probabilities greater than 0.5 were classified as foreground, and predicted probabilities below 0.5 were classified as background.

It is worth noting that the U-Net consists of unpadded convolution layers. This means that the output image segmentation map will always be smaller in dimension compared to the input image. To combat this, we considered zero padding the original images. However, we were advised not to zero pad the images, as this changes the pixel-level distributions.

4.3 Model Structures

At first, we wanted to explore the two different optimizer functions built into *tf_unet*: the momentum optimizer and the Adam optimizer. It quickly became clear that the momentum optimizer was inappropriate for this problem. Several different hyperparameter combinations and model architectures were tested including increasing the number of layers in the U-Net from three to five and using the dice coefficient as the loss function in place of weighted cross entropy. However, these models continued to perform poorly. When applied to the testing set, the background class was predicted nearly 100% of the time. This is likely due to the heavy imbalance in our imaging data set. The ratio of foreground to background in our 60 augmented training images was approximately 1:16. To account for this heavy imbalance, Adam optimizers with a weighted cross entropy loss function were used for all four of our final models.

For this project, we wanted to test the effect of varying dropout rates and validation batch sizes on the U-Nets' performance. We explored dropout rates of 0.75 (package default) and 0.9, along with validation batch sizes of 4 (package default) and 8. U-Nets with a dropout rate of 0.5 were fit. However, the U-Nets with a lower dropout rate did not perform well. For this reason, those results were not reported for this project. Our final U-Nets were all trained using 3 layers, 32 feature roots, a constant learning rate of 0.0001, and 100 epochs with 20 training iterations per epoch. Both the training and validation learning curves for each of the four models were plotted to ensure good model fit.

5 Results

Learning curves are common diagnostic tools in machine learning and deep learning. They visualize whether or not a model is underfitting, overfitting, or a good fit. Initially, we planned to use TensorBoard to plot the training and the validation loss. However, only the training loss is output to TensorBoard directly. Due to this limitation, the training and validation loss was collected manually and visualized using the *ggplot2* package in R-Studio.

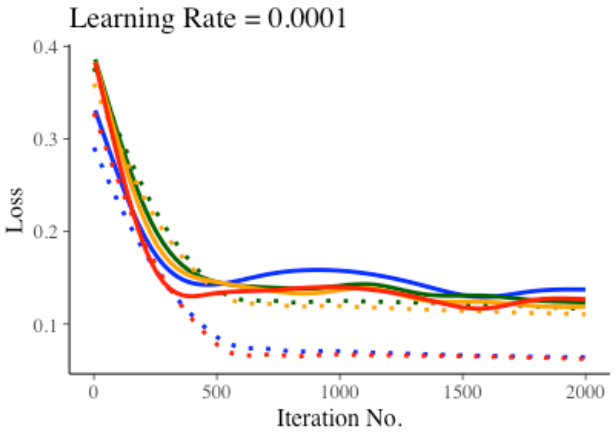

Fig. 3. Learning curves displaying the training and validation loss over training iterations. Solid lines: training loss. Dotted lines: validation loss. Blue: validation batch size of 4, dropout of 0.75. Green: validation batch size of 8, dropout of 0.75. Yellow: validation batch size of 8, dropout of 0.9. Red: validation batch size of 4, dropout of 0.9.

As mentioned previously, the default validation batch size and dropout for the U-Net in the *tf_unet* package were 4 and 0.75, respectively. Our first model was trained using these default values. In visualizing the training and validation loss over the training iterations, we noticed an interesting phenomenon for this particular model. As can be seen in Figure 3, as shown by the blue solid and dotted lines, the majority of the validation loss appeared to be lower than the training loss. While uncommon, this relationship is not impossible. In fact, lower validation loss compared to training loss is often an artifact of including dropout in the U-Net. A lower validation

loss also indicates a potentially unrepresentative validation set in comparison to the training set. To combat this issue, it is often suggested that more data is allocated to the validation set. This initial observation made while training the U-Net with the default validation batch size and dropout was the key motivation to further explore the effect of varying validation batch size and dropout rates.

Figure 3 shows that each of the four U-Nets had relatively similar training loss over the course of training iterations. It also appears that increasing the validation batch size from 4 to 8 resulted in a validation loss that is closer in value to the corresponding training loss, which is encouraging as well as expected. Increasing the dropout from 0.75 to 0.9 did not seem to have a pronounced effect on the training and validation loss. It also appears that the training and validation loss for each of the four trained U-Nets plateaued after approximately 500 training iterations. Ultimately, the learning curves indicate that there were no obvious issues with underfitting or overfitting.

To evaluate the performance of each of the trained U-Nets, accuracy, precision, recall, and F1 scores were each calculated and compared. It is worth noting that, in the presence of heavily-imbalanced data, it is generally recommended that accuracy should not be reported and should not be used to assess model performance when applied to a testing set. In these situations, accuracy may be relatively high in value. However, it is merely reflecting the underlying class distribution. Precision, recall, and F1 scores are typically the more preferred metrics to report. We chose to report accuracy to reiterate that it is a rather misleading metric when the data are imbalanced.

	VS=4, D=0.75	**VS=4, D=0.9**	**VS=8, D=0.75**	**VS=8, D=0.9**
Accuracy	0.9313	0.9337	0.9296	0.9280
Precision	0.6488	0.6771	0.6114	0.5932
Recall	0.4374	0.4171	0.4970	0.5184
F1-Score	0.5226	0.5162	0.5483	0.5533

Table 1. Accuracy, precision, recall, and F1-score of the four trained U-Nets. VS: validation batch size. D: dropout. Hyperparameter settings held constant: layers = 3, feature roots = 32, learning rate = 0.0001, epochs = 100, training iterations/epoch = 20, optimizer = Adam, loss function = weighted cross entropy.

Based on Table 1, increasing the dropout rate from 0.75 to 0.9 increased both the accuracy and the precision. Increasing both the validation batch size from 4 to 8 and the dropout rate from 0.75 to 0.9 increased resulted in the highest recall and F1-score of any of the four trained U-Nets. Accuracy is relatively high for each of the four trained U-Nets. This was included in the final results table was to provide a real-world example of how accuracy can be a misleading measure of classification performance when the data are imbalanced. As mentioned previously, the foreground to background ratio in the training data was approximately 1:16. Thus, the high accuracy likely means that the background pixels in the testing set were accurately classified. However, accuracy does not give us an accurate picture of foreground classification.

6 Limitations

There are numerous limitations of this project that should be addressed and discussed. First, our experience in building and fitting deep learning algorithms is rather limited. With more experience in training various deep learning algorithms, we would have a better understanding of how to adjust the hyperparameters accordingly to improve the performance of the U-Nets.

Second, the variety of hyperparameters explored in this project was rather limited. This was largely due to the fact that training the models took a substantial amount of time and computer power. With more time, we would have expanded our hyperparameter exploration. More time would have also allowed for deeper architectures and increased number of epochs and training iterations within each epoch to be explored.

Third, the fact that the data were imbalanced likely resulted in the reduced classification performance. Weighted cross entropy was implemented in an attempt to alleviate this issue. However, it may have been beneficial to implement other methods. For example, more rigorous data augmentation methods and resampling techniques that over-sample the minority class and/or under-sample the majority class could have been used.

Fourth, the original U-Net developed by Ronneberger et al. was designed to analyze grayscale (single-channel) images. For this project, RGB (three-channel) images were used to train the U-Nets. Using single-channel, grayscale images may have improved the classification performance.

Fifth, the annotation masks used during the training process may not have reflected the actual ground truth. As described in the methods, the original annotation files for the images included the X-Y coordinates of the center of the nuclei and the cell subtype label. To enlarge each annotated point into a small region, the original annotation masks were dilated using a 7x7 kernel, and these dilated images were used as the final annotation masks used in the U-Net training process. Unfortunately, the dilation operation created annotation masks that assumed the cells were perfect 7x7 squares. Thus, the underwhelming classification performance of each of the four U-Nets may have been exacerbated by using annotation masks that are not representative of the actual ground truth.

Finally, further exploration of the test set predictions indicated that epithelial cells were predicted poorly overall. The epithelial subtype generally seemed to have a cellular structure that is less

distinguishable from the background, in comparison to the other three subtypes.

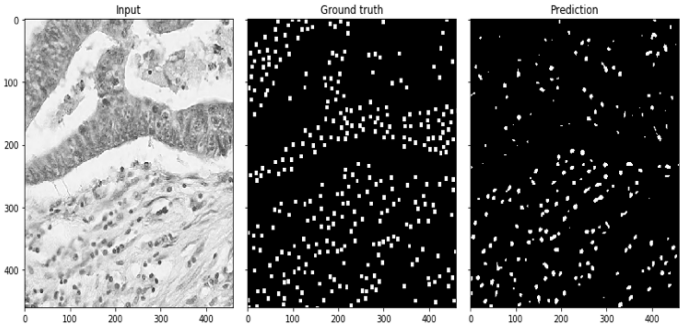

Fig. 4. Comparison of the input, ground truth, and prediction of the ground truth for an image in the testing set.

In Figure 4, it is obvious that the trained U-Net did not predict the epithelial cells well. The epithelial cells are located in the upper half of the image. The prediction image of the ground truth shows that these cells were difficult to distinguish from the background. This phenomenon was observed for other testing images as well. With an already heavily-imbalanced imaging data set, the U-Nets inability to detect the epithelial subtype well likely contributed the poor classification performance.

7 Conclusions

In conclusion, we determined that 1.) the momentum optimizer did not perform well in the presence of heavily-imbalanced data, 2.) increasing the validation batch size resulted in validation loss that more closely reflected the training loss, and 3.) increasing the validation batch size and dropout rate resulted in an increase in both recall and F1-score. In further exploration of the default hyperparameters of the *tf_unet* U-Net, it was discovered that the default coefficient of momentum was set to 0.2, which seems to be relatively low. In general, the coefficient of momentum is set to 0.9. Thus, it is possible that the momentum optimizer would have performed better if the coefficient of momentum was set to a higher value. However, even if the coefficient of momentum was increased, previous applications of the *tf_unet* U-Net suggested that the Adam optimizer would likely perform better in the presence of heavily-imbalanced data.

Today, the current gold standard for diagnosis of colon cancer is a trained pathologist's visual examination of the nuclei color, shape, and other features of H&E-stained tissue samples. However, the accuracy of diagnosis is heavily dependent on the pathologist's experience. The process of visual examination is also an extremely time-consuming process. The combination of these two factors can potentially lead to diagnostic errors, incorrect treatment decisions, and missed opportunities to begin treatment promptly. In an attempt to improve detection of colon cancer, numerous fully convolutional networks have been developed with the goal of eliminating the need for manual visual inspection by a trained pathologist.

For this project, we decided to use the U-Net, a deep learning algorithm developed specifically for the purpose of biomedical image segmentation, for pixelwise, binary classification of colon cancer histology images. In particular, the results illustrated how difficult classification tasks are when the training data are heavily imbalanced. Weighted cross entropy was used as the loss function to try and combat the imbalanced data. However, the results from this project indicate that substantial improvements are necessary to improve classification performance to a clinically-appropriate level. Incorporating more rigorous data augmentation techniques and/or resampling techniques are all potential options that would likely improve U-Net performance. Regardless of the limitations of the project, these results illustrate that fully convolutional networks, such as U-Nets, are viable alternatives for nuclei detection in H&E-stained colorectal tissue sample images.

Acknowledgments

The authors wish to thank Dr. Fuyong Xing for providing the original data, annotation files, and guidance throughout this project.

References

[1] R. Seigel, C. Desantis, and A. Jemal, "Colorectal Cancer Statistics," *CA Cancer Jounal,* 2014.

[2] K. Sirinukunwattana, S. E. Ahmed Raza, Yee-Wah Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images," (in eng), *IEEE Trans Med Imaging,* vol. 35, no. 5, pp. 1196-1206, 05 2016, doi: 10.1109/TMI.2016.2525803.

[3] R. Rogojanu, G. Bises, C. Smochina, and V. Manta, "Segmentation of cell nuclei within complex configurations in images with colon sections," ed. IEEE Xplore: IEEE, 2010.

[4] A. A. Nawandhar, L. Yamujala, and N. Kumar, "Image segmentation using thresholding for cell nuclei detection of colon tissue," ed. IEEE Xplore: IEEE, 2015.

[5] D. Joon Ho, C. Fu, P. Salama, K. W. Dunn, and E. J. Delp, "Nuclei Segmentation of Fluorescence Microscopy Images Using Three Dimensional Convolutional Neural Networks," ed. IEEE Xplore: IEEE, 2017.

[6] Y. Cui, G. Zhang, Z. Liu, X. Zheng, and H. Jianjun, "A Deep Learning Algorithm for One-step Contour Aware Nuclei Segmentation of Histopathological Images," *CoRR,* 2018.

[7] A. Sevastopolsky, "Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network," *Springer,* 2017.

[8] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. CreateSpace, 2009.

[9] M. Abadi, A. Agarwal, P. Barham, and E. Brevdo, "TensorFlow: Large-scale machine learning on heterogeneous systems," ed. tensorflow.org, 2015.

[10] *R: A language and environment for statistical computing*. (2018).

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *MICCAI,* 2015.

[12] *TensorFlow UNet* (2017). Astronomy and Computing.