# Spot Nuclei, Speed Cure, Save Lives

Khyteang Lim
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01851

***Abstract*— The purpose of this project is to present a method for nuclei segmentation using the convolutional networks known as U-Net. U-Net is known to outperform prior best method (a sliding-window convolutional network) in term of speed and accuracy when it comes to biomedical image segmentation. This project was inspired by the 2018 Data Science Bowl challenge on Kaggle. The challenge for this year 2018 Data Science Bowl is to design an algorithm to efficiently spot nuclei in both hematoxylin and eosin (H&E) stained and Hematoxylin-channel grayscale images.**

***Keywords—convolutional networks, U-Net, Kaggle, nuclei segmentation***

## I. INTRODUCTION

In an individual human body, there are over 30 trillion cells that contain nuclei full of DNA code. These cells are the smallest unit of life that can replicate independently in order to form a functional living thing. However, what makes these cells very important is the DNA that lives inside the nucleus. These DNA are the genetic codes that program each and every cell in our body. Researchers have been studying these cells in order to try to understand the patterns in how these infected cells are reacting when treated with various treatments. Having this ability to quickly evaluate these cells with the new treatments, researchers can hope to find cure to these deadly diseases once and for all.

The topic of automated image analysis of cells and tissues has been an active research field in medical due to the rise of computing power and greater microscopy hardware. Researchers have been studying cells in order to find cures for many diseases such as cancers, heart disease, chronic obstructive pulmonary disease, and so many more [1]. At the same time, researchers are trying to understand the patterns of the DNA in these infected cells in order to better diagnose patients. The ultimate goal of this project is to build a model-based algorithm, that given a microscopy image (Figure 1), can detect and identify the location of the nucleus in the image. This would utilize the power of technology to better diagnose a patient and can hopefully save many lives before it is too late.

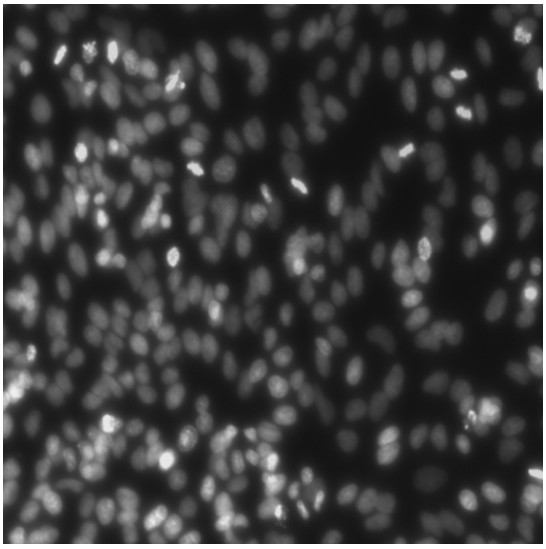

**Figure 1: Microscopy image from dataset**

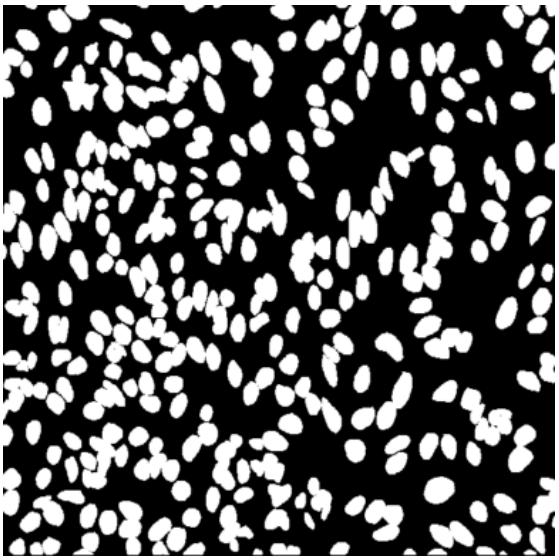

**Figure 2: Processed Mask for Figure 1 image**

## II. Background

The search for cells/nucleus segmentation module of microscopy images is an ongoing research topic in the field of medical. There are various studies and approaches that have been proposed and implemented over the years that included the use of markers detection [2], instance segmentation using Mask RCNN, and stepwise merging of rules based and gradient vector flow snake for segmenting white blood cells [3]. Another approach that many have tried is the use of priori information about cellular features in order to build a generic model of the cell for robust detection [5]. This approach performs great on homogeneous dataset of cells that have similar shape and size. However, this approach is problematic in most relevant cases in pathology such as cancer cell segmentation where the cells are highly heterogeneous in shape and size [4].

On top of using different end-to-end frameworks of convolutional neural networks algorithm and traditional segmentation methods, data augmentation also plays a major role with improving the performance of these algorithms that are being used in some of the existing experiments. Data augmentation can be flipping, rotating, scaling, or translating the images. Another preprocessing step that most experiments perform is normalizing the color of the dataset images. The reason for this is depending on your dataset, some may have a combination of H&E RGB (Figure 3) and Hematoxylin-channel grayscale image of the cell (Figure 1). Using the H&E stain normalization methods can help eliminate the negative interference caused by color variation [10].

## III. Approach

The approach that is being proposed in this paper is to perform semantic segmentation on these microscopy images of cells using the network and training strategy called U-Net, a convolutional networks approach for biomedical images. This approach relies heavily on the use of data augmentation in order to efficiently use the available training data that are given in this challenge. The data that are being used in this project come from the 2018 Data Science Bowl challenge in Kaggle.

When using convolutional neural networks to train a model, the performance of the model depends heavily on having a large amount of dataset for training. With convolutional neural network, a learning model can have millions of parameters that required tuning for optimization and that is why having a large set of data can help optimize the tuning of these parameters. However due to the limitation of the dataset in this challenge, data augmentation played a major role with improving the accuracy and performance of the training model. Without a large sample of training data, training a model with convolution neural networks can lead to the problem of overfitting which we actually experienced at the beginning of the experiment. On top of data augmentation, we had to perform some preprocessing of the training images in order to normalize the dataset. As mentioned earlier, the dataset that we used for this experiment has a combination of both Hematoxylin-channel grayscale and H&E images, so we had to convert all the images into black and white images before we started training the model on these training set.

With the limitation on annotated images and overlapping cells in images, U-Net is the perfect approach for this application. In the U-Net architecture shown below (Figure 2), there are two components to the network flow, contraction and expansion. The purpose of the contraction flow of the network is to increase the data of what we are looking for by increasing the number of filters or feature channel by 2. In the contraction phase, there are a few operations that the input training images have to go through such as convolutions with a 3 x 3 matrix followed by an activation function and max-pooling which purpose is to down-sample the input representation. The other stage of the U-Net architecture is expansion which purpose is to create a high-resolution segmentation map. In the expansion stage, the input images go through convolutions with a 3 x 3 matrix followed by an activation function and up-convolutions with concatenations with the high-resolution features from the contraction path. The output of propagating these input images through the U-Net Architecture is a segmentation map that can be used to compared with the mask of the input data.

## IV. Dataset/Method

*1)* Split dataset: In this challenge, we are given training dataset and testing dataset that we have to evaluate our trained model on and submit a csv file of the result for grading. For the training dataset, we have to do a training-testing split of the dataset for actual training dataset and validation dataset. Validation dataset is used to give estimation of the model skill while tuning the model's hyperparameters.

*2)* Data setup: The training dataset for this challenge contains both the raw microscopy images of the cells and with each image, it comes with *n*-number of masks (shown in Figure 3) that correspond to all the cells in the raw image. However, since we are training our model with U-Net framework we are required to have a mask per training image with an initial size of 572 x 572. In order to do this, we combine all the masks for one raw image into a single image file. Also, during this stage we perform our preprocessing of the training images such as cropping the images in order to increase our dataset. We do the same for both the mask and the raw training data. Other preprocessing operations that we perform on the images are flipping, rotating and scaling.

3) Covert image to tfrecord: Since we are using Tensorflow, an open source machine learning framework, most of their libraries required us to convert the input data into tfrecord, a built-in type, of tuple. During the conversion, we also normalize the color of the training images into black and white tfrecord. The result tfrecord of this conversion is [width, height, intensity] where intensity is either 1 or 0 for black or white pixel.

4) Train the model: Once we have our training dataset as tfrecord, we can feed these tfrecords into our U-Net framework that we have implement using tensorflow libraries.

5) Validate model: Once we have a trained model, it is time to evaluate the trained model on validation dataset in. order to get an unbiased estimation of the model's skill. In this step, we also perform any tuning that needs to be done to the model's hyperparameters.

6) Test the model: Once we have our final model, we can evaluate the test dataset provided by the challenge. The output of this evaluation is a csv file that we submit for a grading.

## V. Evaluation

### A. Experiment #1

During the first experiment, we used an open-source project that has implemented U-Net with Keras and Tensorflow libraries. This implementation follows the whole U-Net architecture and there were no preprocessing or data augmentation that were done to the training dataset. Shown below in Figure 3 is one of the raw images in the test dataset while the image Figure 4 is the predicted nuclei segmentation using the trained model. Without any preprocessing of the training dataset, we saw that the segmentation of the prediction is very distinguish, especially for H&E images. While for Hematoxylin-channel grayscale images, the prediction contains very faded segmentation of the nuclei.

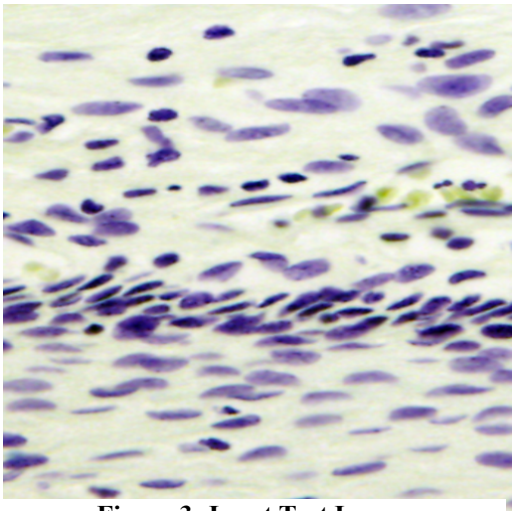

**Figure 3: Input Test Image**

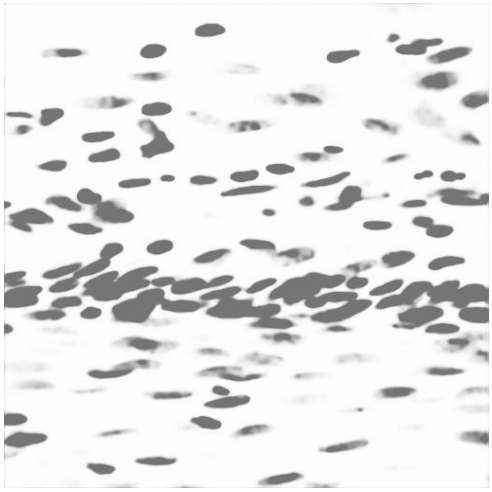

**Figure 4: Prediction segmentation from experiment 1 model**

### B. Experiment #2:

During the second experiment, we implement our own U-Net architecture using Tensorflow machine learning libraries. Trying to implement the U-Net framework from scratch took a lot of work but we managed to get this done with a lot of trial and error. Luckily, Tensorflow has a GPU version which can get install and run on enabled-GPU. Using a GPU acceleration framework helps us to get real prediction and feedback on our implementation of the U-Net framework. Once there is a working U-Net framework that we can start training our model on, we are able to get great (too good to be true) prediction where our accuracy is in the 90% (Figure 5). However, when we start validating our machine learning algorithm with our validation dataset, we see that our model is not making the accurate prediction that we are hoping for. What we notice is that our trained model does not generalize to the new data very well. Our trained model models the training data too well that it negatively impacts the performance. The model does not only learn from the training data, it picks up detail and noise of random fluctuations which the model learns as concepts. Once we have confirmed that this is the cause of overfitting in our model, we begin brainstorming ways to get more generalize dataset for training.

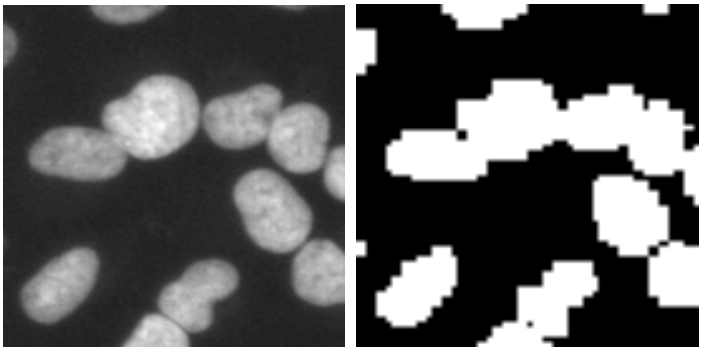

**Figure 5: Input [Left] and predicted output [Right] from experiment 2**

### C. Experiment #3

Once we realize that our model is modeling the training data too well that it negatively impacts the performance of the model, we begin brainstorming ways to both improve our model and increase our training dataset. One of the main causes of overfitting in supervised learning is the limited training dataset that the model can train from. The approach that we take to improve our dataset is to perform data augmentation of the given training dataset. We randomly crop roughly 45 new images from each training image and its corresponding mask. Due to the fact that U-Net framework requires a specific resolution for input images and masks, we have to scale up the cropped images and masks. Another transformation that we do to the images is rotation. Once we have the new set of training data, we feed them into the U-Net framework to start training the model. The prediction that we are seeing with this approach is confusing. We see blocks of prediction in each of the prediction images (Figure 6). We do not understand what the cause of this is, but the conclusion is that our model is underfitting in this experiment.

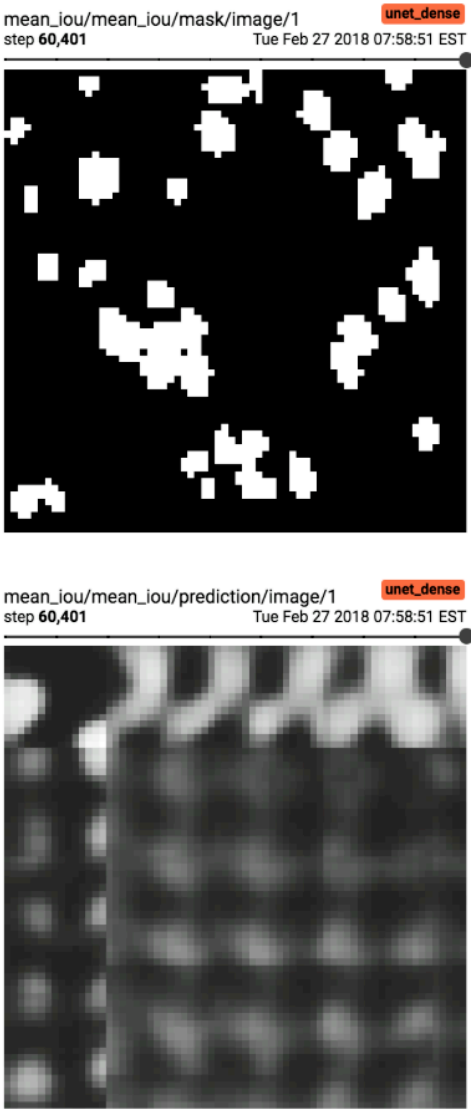


**Figure 6: Input [Top] and predicted output [Bottom] from experiment 3**

## VI. Conclusion

During the last few decades, there have been many state of the arts proposal for nucleus/cell detection and segmentation in digital pathology and microscopy images. Due to the fact that most of these approaches are applied on different dataset, it is difficult to determine that one proposed model is better than another. In this experiment, the new and latest convolutional network known as U-net is used to train a model for nuclei segmentation. The models that we have trained from the few experiments that we run do not have great performance since we have a model that is overfitting and another that is underfitting. This is my first introduction to machine learning and the world of using technologies to help researchers improve their process of studying cells. In order to find cures for diseases such as cancers, heart diseases, chronic obstructive pulmonary diseases, and many more, researchers have to understand the patterns of the DNA in each cell and how the cell is reacting to the various treatment. But the hardest part of their evaluation is the detecting and segmenting of the nuclei in order to find the DNA. This is definitely an exciting field that I am going to continue to invest my times and knowledges in in order to help find cures for many of these diseases. Despite not having a successful model for detecting and segmenting nuclei in microscopy images, I have learned a lot from this experiment and in conclusion, I believe the combination of researchers and technologies will one day find a cure to many of the world diseases.

## References


[1] *2018 Data Science Bowl | Kaggle,* www.kaggle.com/c/data-science-bowl-2018.

[2] Maurer CRJ, Qi R, Raghavan V. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. IEEE Trans. Pattern Anal. Mach. Intell. 2003 Feb;25(2):265–270

[3] Ko B. C., Gim J.-W. & Nam J.-Y. Automatic white blood cell segmentation using stepwise merging rules and gradient vector flow snake. Micron 42, 695–705 (2011).

[4] Bamford P. Unsupervised cell nucleus segmentation with active contours. Signal Processing 71, 203–213 (1998).

[5] Wienert, Stephan et al. "Detection and Segmentation of Cell Nuclei in Virtual Microscopy Images: A Minimum-Model Approach." *Scientific Reports* 2 (2012): 503. *PMC*. Web. 22 Feb. 2018.

[6] Xing, Fuyong, and Lin Yang. "Robust Nucleus/Cell Detection and Segmentation in Digital Pathology and Microscopy Images: A Comprehensive Review." *IEEE reviews in biomedical engineering* 9 (2016): 234–263. *PMC*. Web. 28 Apr. 2018.

[7] P. Wang, X. Hu, Y. Li, Q. Liu, and X. Zhu. "Automatic cell nuclei segmentation and classification of breast cancer histopathology images." Signal Processing: Elseviver, May 2016.

[8] Su, Jie et al. "Automatic Detection of Cervical Cancer Cells by a Two-Level Cascade Classification System." *Analytical Cellular Pathology (Amsterdam)*2016 (2016): 9535027. *PMC*. Web. 28 Apr. 2018.

[9] Plissiti Me, Nikou C, and Charchanti A. "Automated detection of cell nuclei in pap smear images using morpological reconstruction and clustering." *IEEE Trans Inf Technol Biomed.* Web. Oct 14. 2010.

[10] Y. Cui, G. Zhang, Z. Liu, Z. Xiong, J. Hu. "A deep learning algorithm for one-step contour aware nuclei segmentation of histopathological images." Columbia, SC: University of South Carolina, Mar 7, 2018.

[11] Żejmo M., Kowal M., Korbicz J., Monczak R. (2018) Nuclei Recognition Using Convolutional Neural Network and Hough Transform. In: Kościelny J., Syfert M., Sztyber A. (eds) Advanced

Solutions in Diagnostics and Fault Tolerant Control. DPS 2017. Advances in Intelligent Systems and Computing, vol 635. Springer, Cham.

[12] B. Pang, Y. Zhang, Q. Chen, Z. Gao, Q. Peng, and X. You. "Cell nucleus segmentation in color histopathological imagery using convolutional networks". in *Chinese Conference on Pattern Recognition (CCPR),* Wuhan 430074, China, Oct 23, 2015.

[13] S. Sadanandan, P. Ranefall, S. Guyader, and C. Wählby. "Automated training of deep convolutional neural networks for cell segmentation". in *Scientific Reports 7*, Article number: 7860, June 29, 2017.