SMU Data Science Review

Volume 2 | Number 1 Article 8

2019

Automate Nuclei Detection Using Neural Networks

Jonathan Flores
Southern Methodist University, floresj@smu.edu

Thejas Prasad
Southern Methodist University, tprasad@smu.edu

Jordan Kassof
Southern Methodist University, jkassof@smu.edu

Robert Slater
Southern Methodist University, rslater@smu.edu

Recommended Citation

Flores, Jonathan; Prasad, Thejas; Kassof, Jordan; and Slater, Robert (2019) "Automate Nuclei Detection Using Neural Networks," *SMU Data Science Review*: Vol. 2: No. 1, Article 8.
Available at: https://scholar.smu.edu/datasciencereview/vol2/iss1/8

Automate Nuclei Detection Using Neural Networks and Image Detection Techniques

Jonathan Flores, Thejas Prasad, Jordan Kassof, Dr.Robert Slater
Master of Science in Data Science
Southern Methodist University
Dallas, Texas USA
{jflores, jkassof, tprasad, rslater}@smu.edu

Abstract. Nuclei identification is a pivotal first step in many areas of biomedical research. Pathologists often observe images containing microscopic nuclei as part of their day to day jobs. During research, pathologists must identify nuclei characteristics from microscopic images such as: volume of nuclei, size, density and individual position within image. The pathology field can benefit from image detection enhancements done through the use of computer image segmentation techniques. This research presents methods that can be used to identify all the cell nuclei contained in images. Multiple techniques were experimented with such as edge detection and Convolutional Neural Networks with U-Net architecture. The data for training these models was sourced from the 2018 Data Science Bowl sponsored by Kaggle and Booz, Allen, Hamilton. As a result, there were various methods identified to assist the pathology industry for automating nuclei detection by using computer image detection methods. These computer methods rapidly process images for research purposes, with a reasonably high accuracy which has the potential to greatly accelerate the pace of research.

1 Introduction

The goal of this paper is to add value to the biological research process through the use of image classification techniques - Edge Detection, CNN and U-Net. Edge Detection worked very well on images that had isolated nuclei. This technique detected the edges of the nuclei and highlighted the density within the nuclei. Images of Edge Detections highlighting nuclei are shown in the Methods section. The Morphological Chan Vese Segmentation, specifically, worked very well for identifying nuclei; however, this method fails when nuclei is extremely dense within the image. The neural network models performed more consistently across the entire dataset; however, the precision requires more fine tuning to improve the accuracy. There are opportunities for future work in order to improve nuclei classification/segmentation. However, this paper demonstrates there are multiple potential solutions to automate this detection.

The fundamental unit of all known biological life is a cell. Cells consist of a liquid cytoplasm, surrounded by a membrane, contained within the cytoplasm are a myriad of organic molecules such as proteins and nucleic acids. There are two primary types of cells - prokaryotic and eukaryotic. Prokaryotes are typically single-cell organisms such as bacteria and archaea whereas eukaryotes are the more familiar forms of life such as plants, animals, and fungi. Since the focus of this paper is on biomedical research for human disease the focus will be exclusively on eukaryotic cells going forward. The flagship feature of eukaryotic cells is their compartmentalization, eukaryotic cells contain many smaller membrane-bound organelles such as mitochondria, Golgi apparatus, and most relevantly, a nucleus.

The nucleus is an organelle found in most eukaryotic cells, its primary function is to contain and protect the organism's genetic material, DNA. DNA is a special biological molecule, and is the key to the complex and diverse forms of life that can be seen across the world. DNA is the instruction manual that tells a cell everything it needs to know about itself, from how it should be shaped as it grows to how the cell’s internal systems should be regulated. Given the crucial role the nucleus and the DNA contained within in cell play in cell function, it is commonly referred to as the control center for the cell.

Traditionally, identifying nuclei is a manual process in which a scientist looks at images and tags the location of nuclei. Nuclei detection is a good fit for compute image classification since it is a repetitive, manual and rules-based process. The goal of this research is to optimize the step of cell nuclei identification in the biological research pipeline - with the intention to increase the throughput of the medical research process. The data used in in this research is from the 2018 Data Science Bowl, sponsored by Booz, Allen, Hamilton and Kaggle. The data consists of images of cells, and image masks which indicates where on the image a cell nuclei is present.

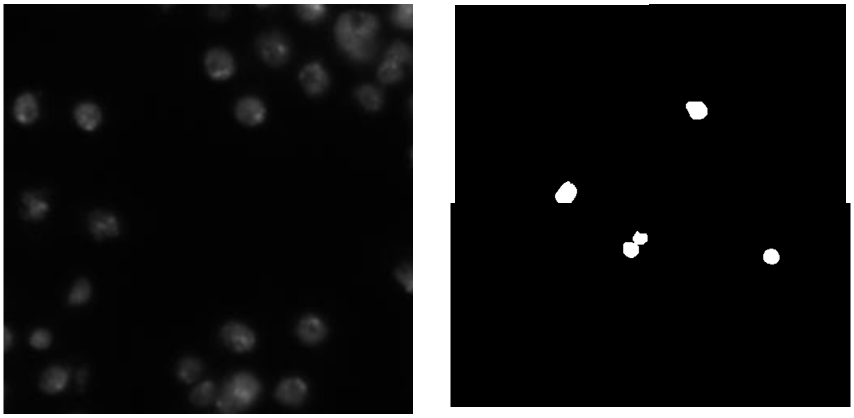


Fig. 1. Sample images from 2018 Data Science Bowl data, first image is input of unlabeled cells, second (four) images are sample masks that show location of cell nuclei [1]

Machine or computer vision is a field within computer science in which the goal is to design systems that recover useful information about a scene from their two-dimensional projection [2]. Machine vision has grown rapidly in recent years, correlated with the rise of smartphones and social networks [8]. As an example, when a picture is posted on Facebook, and Facebook identifies individuals in the photo and suggests to tag them, this is computer vision at work. The output of a computer vision system is not necessarily a single label (cat, dog, happy, sad), it can be a binary label for each pixel meant to indicate the presence of an object. By stitching all those labels together, a "mask" that indicates the position of individual objects within an image can be created. This process of taking one image and automatically identifying all the constituent objects within that image is known as image segmentation.

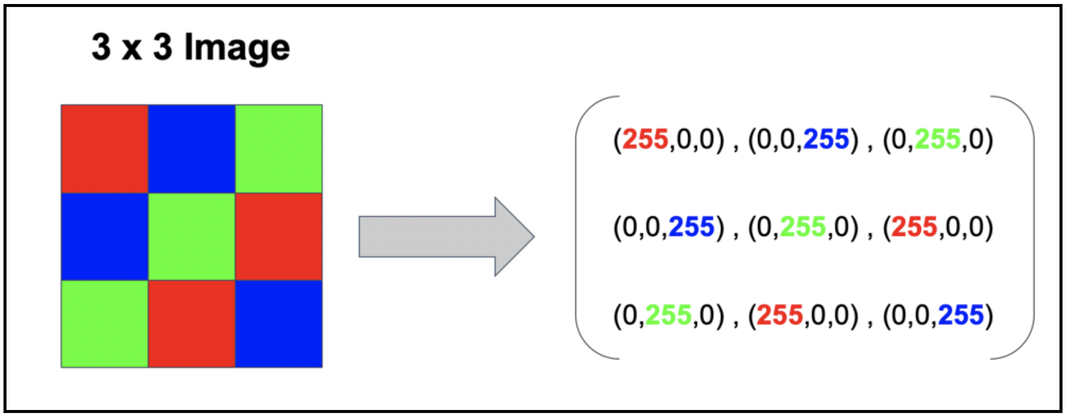
In recent years, deep convolutional networks have shown large efficiency gains in many visual recognition tasks [9, 19]. The historical limiting factor of using convolutional neural networks in biological research was the requirement for very large training data sets. It may be easy to get millions of images of cats, but it is not so easy to get millions of high quality images of rare cancerous cells. U-Net architecture, as proposed by Ronnenberger, Fischer, and Brox [7] has shown great success in implementing effective image segmentation algorithms that can train on relatively small data sets.

A key insight in U-Net architecture is the use of data augmentation to simulate a larger training dataset. Training the algorithm on many altered versions of the same image effectively generates a more generalized and robust network, capable of performing quite well on test data. To further push the concept of image augmentation as a performance booster, a series of visual filters can be applied to images and train with those augmented images. TensorFlow - an open-source deep learning framework originally developed by Google, was used to implement a U-Net CNN, specifically leveraging the Keras Tensorflow API. Keras is a high-level API for TensorFlow which allows rapid experimentation of different deep learning architectures.

2 Image Data Processing Background

The methods used for this analysis require the use of convolution arithmetic, as this is the basis for decomposing and analyzing images. Images are composed of pixels. Pixels are small or tiny squares within an image that contain a specific color. The combination of these tiny squares (pixels) and colors would then compose a larger image which would be visible to the human eye. A good example to explain image processing and convolution are TV images. TVs that have a clear and vivid picture quality have a very high resolution. In this generation, a high resolution TV would be a 4K and 2K pixel TVs[1]. The more pixels a TV has, the higher the definition it will have, and therefore the clearer the image will be.

The explanation above illustrates the basis of the approach to solve the problem at hand, which is identify the Nuclei from an image dataset. These images contain pixels, and each individual pixel contains a color. In order to identify Nuclei, the images in the dataset need to be decomposed at the pixel level in order to teach the computer the target element (Nuclei). In addition, the pixels need to be categorized accordingly as to which pixels contain the target image using 0s and 1s. This way the data frame to be used after the analysis, is one that contains data of nuclei pixels vs non-nuclei pixels. Once this pixel segmentation is performed, later in the process, an aggregation method with all the pixels would be required to identify the larger image and segmented features. This aggregation method of pixels is important as no single pixel can tell what the image is about unless there is a combination of pixels.



(255,0,0) , (0,0,255) , (0,255,0)

(0,0,255) , (0,255,0) , (255,0,0)

(0,255,0) , (255,0,0) , (0,0,255)

Fig. 2. A visual representation of a 3x3 matrix of pixels

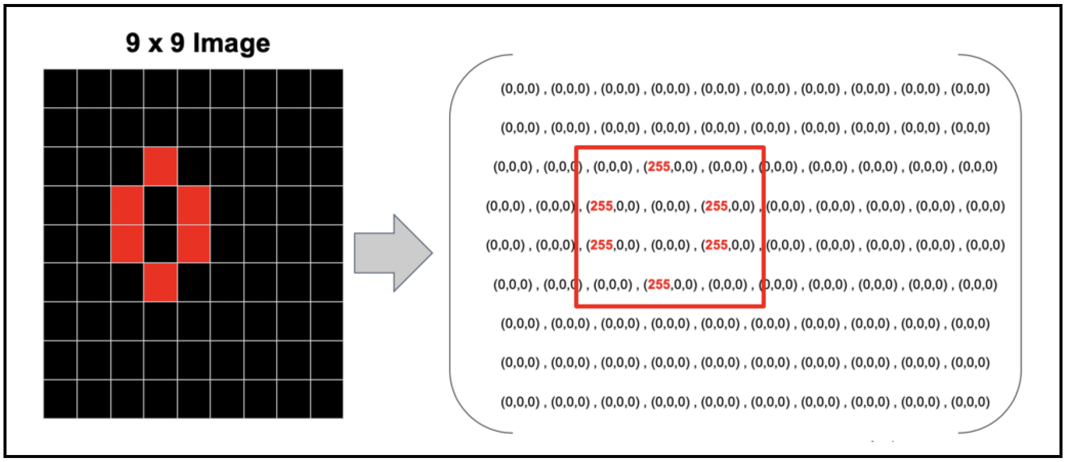
The image processing is done through raw data using a preferred computing tool. The image/pixel decomposition exercise previously mentioned is translated into sequences of numbers or arrays in the form of mathematical matrices[2]. Each individual pixel would represent a numeric indicator in the mathematical matrix. This is done in order to compute any mathematical calculation and be flexible into breaking the pixel composition of the image further, in case this is needed. Moreover, the process to decompose an image is done through a grid method identifying if each pixel contains a specific color. A color in computing is a combination of the three primary colors: red, green and blue. This data is also stored in raw data in the form of mathematical matrices.

A technique commonly used in convolution, is to apply a 'filter' of a different color scheme to the image to be analyzed or trained. This is to isolate the target features from the rest of the picture. These filters would have a numeric composition very similar in format to the one decomposed in pixels on the original image. In order to apply a filter to a specific image, these filter matrices are integrated with the original image by applying a matrices calculations, which is where the convolution process begins. The figure below indicates the progression of pixel calculations for this

[1] Note this technology continues to evolve and more and more pixels are added to technologies to produce a more vivid image.

[2] Carlo Tomasi, "Image Correlation, Convolution and Filtering", 1-10

process and how pixels can be combined, reduced, aggregated and calculated in the convolution process.



```
(0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (0,0,0) , (255,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (255,0,0) , (0,0,0) , (255,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (255,0,0) , (0,0,0) , (255,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (0,0,0) , (255,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
(0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0) , (0,0,0)
```

Fig. 3. A visual representation of a 9x9 Image with its pixels

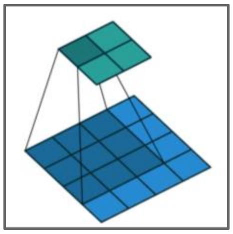
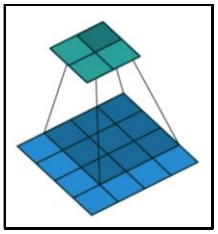
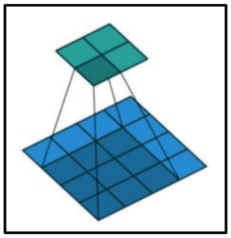
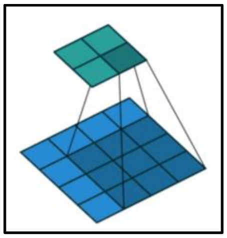
Pixel Calculation 1	Pixel Calculation 2
	
Pixel Calculation 3	**Pixel Calculation 4**
	

Fig. 4. A visual representation of the process of applying an image filter across a set of pixels

A final process after defragmenting an image in order to read its individual components, is to recreate the image with the features highlighted or identified. In the image above, the green picture represents a matrix of the possible output of a new image. This in essence would be an image that was modified from the original image highlighting the target features.

3 Related Work

This paper builds on the ideas of a wide range of research and topics. Humayun Irshad, Antoine Veillard, Dr. Ludovic Roux and Daniel Racoceanu have done a detailed research and shared many important details related the nuclei detection and its importance in the field of digital pathology using traditional techniques [2]. They have also highlighted the challenges currently faced in the medical diagnosis process especially in image processing for nuclei detection, segmentation and classification. The paper shares the details of conventional image process methods such as thresholding, morphology, region growing, watershed, active contour models and level sets, K-Means Clustering, probabilistic models and graph cuts. Through this research paper an attempt is made to address some of these challenges which exist today in the medical field and try to enhance the overall process in addition to the traditional methodologies.

Christopher D Malon and Eric Cosatto in their research article have discussed how the automatic analysis could lead to reduction in the pathologist labor and thus improving the quality and efficiency of overall medical diagnosis procedures [3]. They used Support Vector Machine and deep learning technique of Convolutional Neural Networks in training the model to solve the problem of image segmentation for nuclei detection. Their trained model achieved F1 scores up to 0.659 on color scanners and 0.589 on multispectral scanner. This article was created as part of mitotic figure recognition contest at the 2012 International Conference on Pattern Recognition (ICPR) challenges a system to identify all mitotic figures in a region of interest of hematoxylin and eosin stained tissue, using each of three scanners (Aperio, Hamamatsu, and multispectral). This paper builds on these methods and processes defined by various researches already done on this problem. Based on the observation on reviewing many researches already done related to medical diagnosis, nuclei identification and classification appears to be a tedious tasks for the pathologists and consumes significant amount of time in the overall process.

Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi highlights the challenges in the conventional image segmentation processes like Otsu and Watershed in their research IEEE paper [4]. They have used Convolutional Neural Networks to generate the model and have obtained an overall F1 score of 0.8267. They have used Hematoxylin and eosin (H&E) stained images for their research in developing the model for nuclei segmentation. Their reason for using H&E stained images is that Hematoxylin renders nuclei dark blueish purple and epithelium light purple, while eosin renders stroma pink. Together, H&E enhance the contrast between nuclei, epithelium and stroma for examination under a microscope. The basic challenges in nuclear segmentation using traditional technologies are, as the original H&E stained tissue images show as crowded and chromatin-sparse nuclei, the Otsu thresholding method leads to merged nuclei (under-segmentation). Marker controlled watershed segmentation method leads to fragmented nuclei (over-segmentation). The technique used in their research paper is CNN which detects and segments almost all nuclei well. Each segmented nucleus is shown in a separate color in this model. There seems to be a vast amount of untapped information in H&E stained images that can be used for specific diagnoses such as cancer molecular subtypes determination.

Vahadane, A., & Sethi, A. in their IEEE paper Towards generalized nuclear segmentation in histological images [5], shares the enhancement details applied on traditional image segmentation processes. The main reason for proposing the enhancements is due to the fact that watershed segmentation is prone to errors like over segmentation when applied to histological images. They propose specific enhancements to improve segmentation of cell nuclei in histological images. At a high level these are the proposed enhancements - Foreground seeds were generated by fast radial symmetry transform (FRST). Otsu thresholding was used on enhanced image to estimate tentative foreground map. Background markers were computed from the tentative foreground map. False detections in the segmented output were removed by logical AND with the tentative foreground map. They have confirmed that by using these enhancements the nuclear segmentation improved significantly for the historical images like H&E stained breast and intestinal tissue images.

The research presented in this paper builds off of everything above, but will try to solve the problem of nuclei segmentation by combining all the best features and ideas presented above. In addition, the team focuses on Machine Learning & Deep Learning techniques that has a higher potential in significantly improving the efficiency in the image segmentation process over the current traditional methods used. The methods proposed in this research paper can be applied in other areas with minor customizations.

4 Data

Datasets used in this research paper are publicly accessible datasets obtained primarily from Kaggle and other external sources, it contains a large number of segmented nuclei images. In an effort to diversify training data, the team used these four sources of data: (1) Kaggle Data Science Bowl 2018, (2) IEEE paper on Transactions on Medical Imaging "A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology", (3) Murphy Lab & (4) Zenodo.

Datasets obtained from Kaggle[3] are images that were acquired under a variety of conditions and vary in the cell type, magnification, and imaging modality (brightfield vs. fluorescence). The dataset used is designed to challenge an algorithm's ability to generalize across these variations. The dataset includes a diversity of nuclear appearances from several patients, disease states, and organs, techniques trained on it are likely to generalize well. This dataset has 670 images and associated masks for training the model and 65 images to test the model by predicting the masks as they will be not exposed while training the models. As these are human annotated datasets, various forms of error exists in this dataset which needs to be accounted for before being used for training the the models.

The dataset obtained from Multi-Organ Nuclei Segmentation Challenge 2018 (MoNuSeg) is publicly accessible and are annotated datasets of H&E stained tissue images with painstakingly annotated nuclear boundaries[4]. The quality of the annotations was validated by a medical doctor. As the dataset includes a diversity of nuclear appearances from several patients, disease states, and organs, techniques trained on it are likely to generalize well and is expected to work fine on other H&E stained images. This dataset was generated by carefully annotating tissue images of several patients with tumors of different organs and who were diagnosed at multiple hospitals. This dataset was created by downloading H&E stained tissue images[5] captured at 40x magnification from archive. H&E staining is a routine protocol to enhance the contrast of a tissue section and is commonly used for tumor assessment (grading, staging, etc.). Given the diversity of nuclei appearances across multiple organs and patients, and the richness of staining protocols adopted at multiple hospitals, the training dataset will enable the development of robust and generalizable nuclei segmentation technique. This dataset was used for the research work in IEEE paper “Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology” [4].

The dataset[6] obtained from Murphy Lab Carnegie Mellon University are Hand-Segmented Datasets. The lab is a multidisciplinary environment with people working on projects in computational cell biology. The dataset contains original images from two microscopes for two cell types stained with Hoechst 33342 as PNG files. It also contains images showing hand-segmentation of the Hoechst images into regions containing single nuclei.

The dataset obtained from in Zenodo Org[7] was used in the IEEE research paper "Segmentation of Nuclei in Histopathology Images by deep regression of the distance map" in Transaction on Medical Imaging by Naylor, P., Lae, M., Reyal, F., & Walter, T. [6]. This dataset consists of 50 annotated images, divided into 11 patients (DOI: 10.5281/zenodo.1174342.) by extracting three to eight 512 × 512 patches from different areas of the tissue. Its mentioned in this paper [6] that this dataset was generated at the Curie Institute consisting of annotated H&E stained histology images at 40× magnification. All slides were taken from a cohort of Triple Negative Breast Cancer (TNBC) patients and were scanned with Philips Ultra Fast Scanner 1.6RA. The annotation was performed by expert pathologists in the institute. This data set represents both intra- and inter-patient variability for the same cancer type.

High-quality labeled training datasets are integral part of supervised and semi-supervised machine learning algorithms. The quality of dataset chosen will significantly impact the outcome of the research, as it can lead to inaccurate findings in the research if the data quality is not good. Hence identifying the right set of data to train the model for this research is the most important task. As

[3] https://www.kaggle.com/c/data-science-bowl-2018/data

[4] https://nucleisegmentationbenchmark.weebly.com

[5] https://cancergenome.nih.gov

[6] http://murphylab.web.cmu.edu/data/2009_ISBI_Nuclei.html

[7] https://zenodo.org/record/1175282#.Ws2n_vkdhfA

mentioned above the main focus in this paper would be on the Kaggle dataset for training and testing the model initially, later other datasets will be used as a supplement. This helps us to improve the model accuracy and efficiency by using a combination of variety of datasets and enable the development of a powerful model and also universalize the nuclei segmentation technique which can then be applied to other areas with minimal modifications.

5 Methods

There are specific techniques and methods to process images in the data science space. Some of these methods are: **1)** Edge Detection and Image Filters **2)** Convolution Neural Networks (CNN) **3)** U-Net and **4)** Mask R-CNN

5.1 Edge Detection and Image Filters

Edge detection is a method to identify borders in an image by identifying changes in contrast, brightness and sharpness. Several 'Edge Detectors' were tested as one of the methods to identify nuclei within the image dataset. The edge detectors used to run these tests are as follow: NDImage (image filter) [17], Canny Detector [16], Shape Index [18], Image Filter, and Morphological Chan Vese Segmentation [14]. All edge detectors are from the scikit-image library [15].

Some of the edge detectors were unsuccessful. These unsuccessful filters and edge detectors created more noise and nuclei was not identified properly. However, others edge detectors performed well. The images below give a very clear representation on how each one of the image filters, contour and edge detectors performed. Two high level requirements are needed for nuclei detection **1)** Be able to clearly identify the nuclei in the image and **2)** Convey this image into a numeric form in order to report how many nuclei was identified and where. Since all of the images below can be transformed into its numeric composition this second requirement is met when using Edge Detection and Image Filters.

As part of the first requirement it can be observed that the NDImage and Shape Index produce noisy data for this specific nuclei use case. Even though the image displays the nuclei, it is harder to see the nuclei compared to the rest of the edge detectors and filters. When looking at the Canny Feature example and Magma Image Filter, there is a nicer visualization and identification of nuclei. The Magma Image Filter does a very good job at highlighting the contour and edges. At the same time, this filter highlights where there is more concentration of color or edges. Even though the Canny Feature identifies the edges, it fails to connect the individual nucleus as there are broken nucleus.

The best edge detector seems to be the Morphological Chan Vese Segmentation. This edge detector provides not only the edges and its concentration, just like the Magma Image Filter, but also highlights the area of the nuclei. Highlighting the area of the nuclei is important as it can be used to calculate the Intersection over Union, metric later covered in detail.

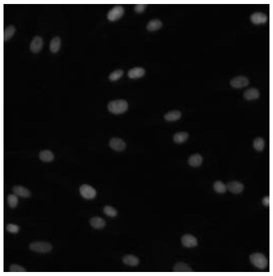
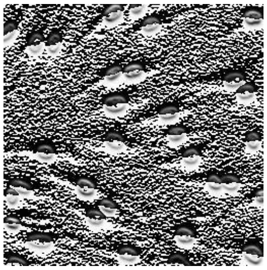
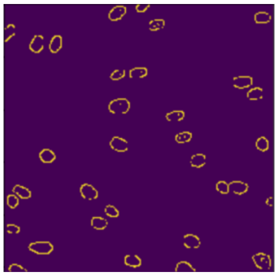
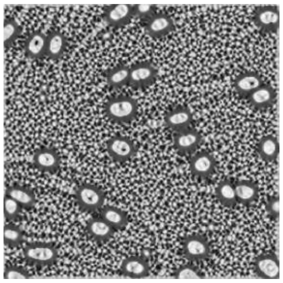
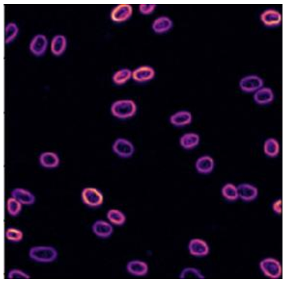
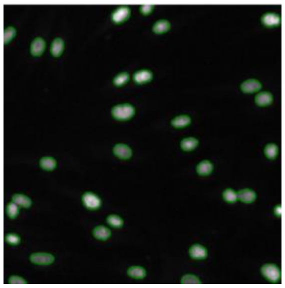
Original Image	**NDImage**	**Canny Feature**
		
Shape Index (σ=1)	**Magma Image Filter**	**Morphological Chan Vese Segmentation**
		

Fig. 5. Sample original images and its edge detected images.

The Morphological Chan Vese Segmentation' - Active contours without edges implemented with morphological operators. It can be used to segment objects in images and volumes without well defined borders. It is required that the inside of the object looks different on average than the outside (i.e., the inner area of the object should be darker or lighter than the outer area on average).

5.2 Convolution Neural Networks (CNN)

For this Nuclei Analysis Identification, 'Convolution Neural Networks‘ was used to process, read, analyze and generate the nuclei identification algorithm. This technique breaks down the images into pixel arrays. The image is segmented in a data grid with all of the pixels mapped with 0s and 1s. A 0 is tagged in those pixels where there is irrelevant data to be marked and a 1 is tagged in those pixels where relevant data needs to be marked. After this process is completed, the data is now formatted in a flat data format as a pixel array matrix with only 0s and 1s. This process makes it easier to make respective calculations against various other filters for convolution processing. This convolution between filters and flat data (previously formatted as an image) are passed into feature identification models. CNN can ultimately segment any trained feature and identify features within images as previously explained, in this case Nuclei.

5.3 U-Net

Another technique known for image processing is U-Net. Like CNN, this technique decomposes the images using a sigmoid function (values between 0 and 1) to read respective features. U-Net's architecture passes input images through a series of convolutions and matrices reduction processes, then as it reaches the final convolution composition and the smallest matrix reduction, the matrices start building back up with now a combination and a comparison with the real image. The final output is a new recreated image with segmented features.

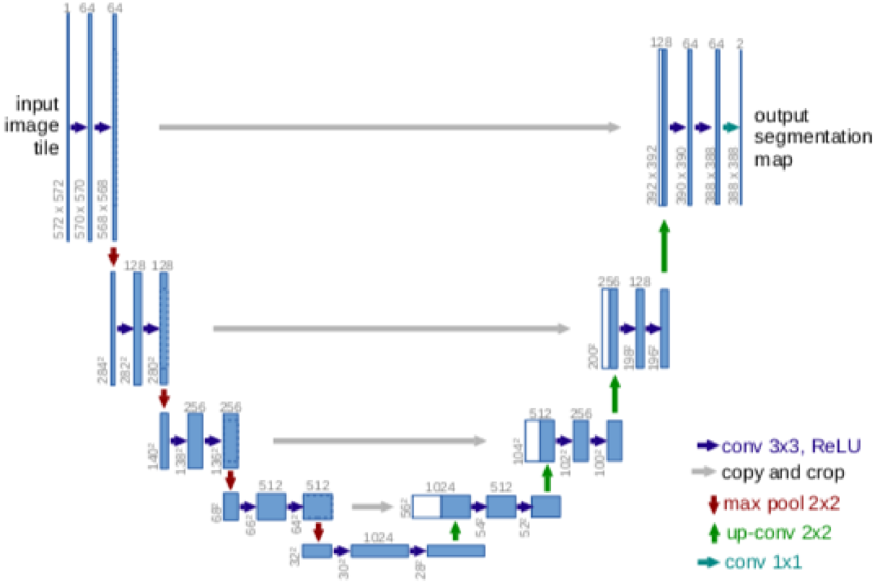


Fig. 6. A U-Net architecture. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations [7].

5.4 Mask R-CNN

This method combines two methods for a more powerful method. Mask R-CNN generates bounding boxes and segmentation masks for each instance of an object in the image. This method is useful since the Nuclei dataset not only contains a single nuclei per image, it contains multiple nuclei per image. In object detection, many objects can be classified in an image by creating a shape around the target object. In Instance Segmentation, the objects can be colored around the object contour with a primary color for object detections. These two primary image detection algorithms are the base of Mask R-CNN as it both identifies multiple objects of the same kind while highlighting in primary colors the contour of that specific object

6 Metrics To Analyze Accuracy and Precision for Nuclei Predictions

The nuclei prediction requires to be measured for accuracy. Once the images are converted into series of pixels and matrices notation, then further steps can be performed to measure how accurate those predictions were against the original masks. In addition, due to the many number of masks within a specific image, it is necessary to choose a metric that not only takes into account the precision of just one individual mask but all the multiple masks within an image. Once the level of accuracy is calculated at the mask and image level, then a third computation can aggregate the accuracy of all masks and images at the dataset level. In the explanation below, 3 high level steps are explained conceptually for the overall computation of accuracy for the nuclei detection: **1)** Step 1 - Computing Mask Accuracy **2)** Computing Image Accuracy and **3)** Computing Nuclei dataset detection accuracy

Step 1 - Computing Mask vs Predicted Mask Accuracy Using 'Intersection over Union' (IoU)

'Intersection over Union" is a metric that can assist with measuring the accuracy of nuclei 'Prediction' versus the 'Ground Truth' [11]. Below is the 'Intersection over Union' formula, in which the numerator explains the area overlap between the 'Ground Truth' and the 'Prediction', while the numerator explains the 'Area of Union" between the 'Ground Truth' and the 'Prediction.

$$IoU\ (A,B)\ =\ \frac{A \cap B}{A \cup B}$$

Formula 1. Intersection over Union (IoU)

In the display below, 'Intersection over Union' is represented conceptually for the Nuclei Prediction use case in the form of a progression to compute the predictions' accuracy. There are four nuclei areas that need to be computed to derive IoU: **1)** Ground Truth's Area **2)** Prediction's Area **3)** Overlap Area and **4)** Area of Union.

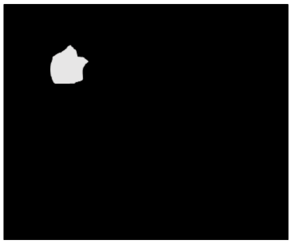
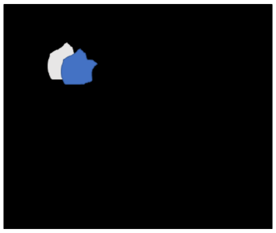
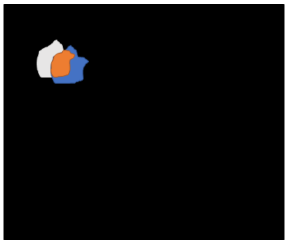
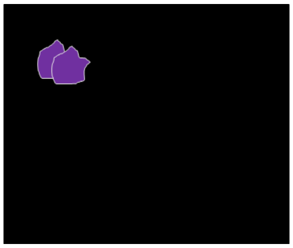
Display 1 - Identify "Ground Truth" (Grey Area)	**Display 2 - Ground Truth (grey) and Prediction (Blue Area)**
	
Display 3 - Overlap Area (Orange Area): $\boldsymbol{A \cap B}$	**Display 4 - Area of Union (Purple Area):** $\boldsymbol{A \cup B}$
	

Fig. 7. A visual representation of the IOU metrics

In the 'Display 1' from the representation above, the "Ground Truth" becomes the masks derived from the actual image. The characteristics of the "Ground Truth" becomes the basis of comparison in further steps for computing 'Intersection over Union'. When the prediction is generated by the algorithm or model of choice, then the predicted mask (Display 2- Blue Area) is compared to the "Ground Truth" (Grey Area). In this comparison, there will be two overlaps that are required to compute IoU: **1)** The area of overlap (Display 3 - IoU's numerator) and **2)** the Area of Union (Display 4- IoU's denominator). Once these areas are computed for that individual mask, then the ratio of these two would indicate how accurate was the 'Ground Truth' from the 'Prediction'. With this principle in mind, if the prediction is very accurate, the overlap and union areas of the predicted mask would be very similar to the Ground Truth's area. If the prediction is inaccurate and far off from the Ground Truth, the areas would be dissonant and not have the overlapping required to be accurate.

Step 2 - Computing Precision for Individual Images

Once the mask accuracy is computed with the IoU for each mask, 'Precision' can be calculated for each individual image. An image can have multiple masks; therefore, a calculated 'Precision' for an image could be impacted if there are predicted objects with no associated Ground Truths (True Negatives) for that particular image. Moreover, if there are Ground Truths with no associated prediction (False Negatives) can also impact the Precision ratio for that specific image. A good nuclei image 'Precision' should have the number of predictions matching the number of ground truths with a certain IoU threshold. IoU thresholds higher than 0.5 are considered a IoU hit. The precision formula is as follow:

$$Precision(t) = \frac{TP(t)}{TP(t) + FP(t)}$$

Formula 2. Precision

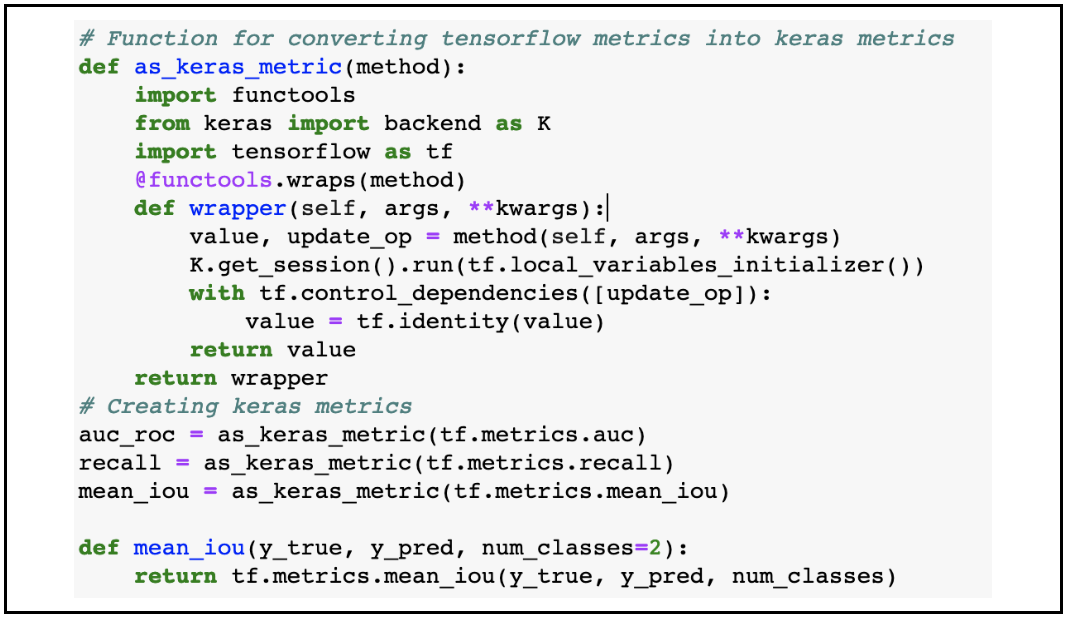
Step 3 - Average Precision for the Entire Nuclei Dataset

Once the precision is generated for each individual image, then an overall precision average can be computed for the entire dataset. This computation can be used to compare multiple models' accuracy. As the models are being refined over and over, it is essential to have this level of aggregation to determine which model is performing better than another. The average precision takes individual image precisions and averages. At this point this aggregation is much simpler than earlier steps.

$$Avg.Precision = \frac{1}{n_{threshold}} + \sum_{t=1}^{n} \blacksquare \text{precision (t)}$$

Formula 3. Average Precision for Entire Nuclei Dataset

Below is a code excerpt from where the Keras metrics used for experimentation and training were defined.



```
# Function for converting tensorflow metrics into keras metrics
def as_keras_metric(method):
    import functools
    from keras import backend as K
    import tensorflow as tf
    @functools.wraps(method)
    def wrapper(self, args, **kwargs):
        value, update_op = method(self, args, **kwargs)
        K.get_session().run(tf.local_variables_initializer())
        with tf.control_dependencies([update_op]):
            value = tf.identity(value)
        return value
    return wrapper
# Creating keras metrics
auc_roc = as_keras_metric(tf.metrics.auc)
recall = as_keras_metric(tf.metrics.recall)
mean_iou = as_keras_metric(tf.metrics.mean_iou)

def mean_iou(y_true, y_pred, num_classes=2):
    return tf.metrics.mean_iou(y_true, y_pred, num_classes)
```

Fig. 8. IOU metrics python code

7 Current Code and Basic Architecture

This section gives an overview of data preprocessing and the Neural Network architecture used in this image segmentation problem. The project folder structure contains two folders at the top level, training and testing data are in these two separate folders. The train directory is further subdivided into separate folders for each image, this image folder name will be the actual image id. This image folder contains two sub folders one for the actual image and the other for the pre-labeled masks. The mask folder has one image per nuclei, *i.e.* if the original image has five nuclei then there will be five corresponding images in the mask folder. As the folders are in a well-defined structure, that structure will be integrated with the code to load the data for preprocessing before it is passed to the model.

The entire data processing and model training pipeline relies on a range of open source Python libraries. They include: numpy, pandas, skimage, tensorflow, and keras.

Once the data is loaded, it is resized to 512 (image width) by 512 (image height) for further analysis. One of the major data processing tasks is to combine all the mask images to one image, in that way there will be one labeled output per input image in the training set. The input images are loaded and stored in python as numpy arrays. All the images are then converted to gray scale for enhanced processing. Test images are also loaded in the same format as explained above. Figure 9 shows a sample image after initial preprocessing.

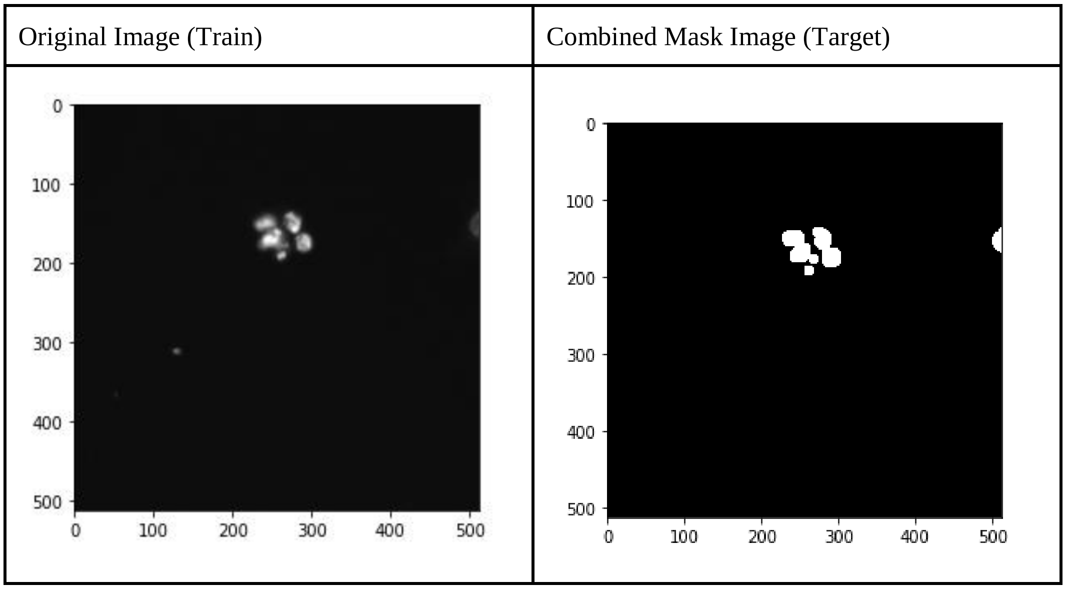
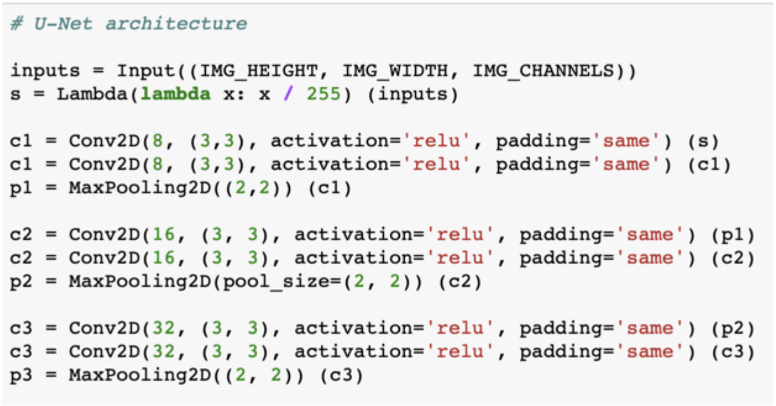


Fig. 9. This is a sample image printed after loading the image data into python and storing it in numpy arrays.

Using keras and tensorflow as backend the Neural Network layers are built using Conv2D and Conv2dTranspose for convolution and deconvolution process in the architecture. In the current architecture there is an input layer and an output layer. The hidden layers details are as follows, there are eight convolution layers and four max pooling layers in the first half of the architecture. The second half of the architecture consists of four deconvolution layers or convolution transpose layers followed by two convolution layers after each convolution transpose layers. The filters used are 3 by 3 in the convolution layers and 2 by 2 for the max pooling layers. The activation function used is ReLu for the hidden layers and Sigmod for the output layer. The loss function used is 'binary_crossentropy' and optimizer used is 'adam'. For all the convolution layers the padding used is 'same', this ensures that there will be no information loss during the feature extraction process. The data is scaled by dividing by 255 before passing into the neural network model for training. Below is a small sample of the Keras code used to build up the network. Keras provides a relatively ergonomic API for stacking a wide range of Tensorflow layers [10].



```
# U-Net architecture

inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
s = Lambda(lambda x: x / 255) (inputs)

c1 = Conv2D(8, (3,3), activation='relu', padding='same') (s)
c1 = Conv2D(8, (3,3), activation='relu', padding='same') (c1)
p1 = MaxPooling2D((2,2)) (c1)

c2 = Conv2D(16, (3, 3), activation='relu', padding='same') (p1)
c2 = Conv2D(16, (3, 3), activation='relu', padding='same') (c2)
p2 = MaxPooling2D(pool_size=(2, 2)) (c2)

c3 = Conv2D(32, (3, 3), activation='relu', padding='same') (p2)
c3 = Conv2D(32, (3, 3), activation='relu', padding='same') (c3)
p3 = MaxPooling2D((2, 2)) (c3)
```

Fig. 10. U-Net CNN Code snippet.

Fifty epoches with a batch size of 10 is used for model training. The validation split is 0.1 - meaning 90% training data and 10% validation data. To be more efficient an early stopper has been implemented by monitoring the validation loss with a patience value of 5. The ModelCheck function will save the best model built so far to the local system during the training process. With 50 epochs the time taken to complete the training of the model is 12 minutes 59 seconds on a retail home computer GPU. Once the training process is complete the best model saved is loaded and then the predictions are made on the unseen data. The metrics used in the training process is mean_iou (Intersection over Union). The best mean_iou value obtained on the validation set is 0.4236. Based on the results obtained so far it can be assumed that there could be still options to improve the model and the inference accuracy scores which needs to explored further.

7 Ethics

To understand why a rigorous and exhaustive analysis of the ethics of machine learning is important, consider that machine learning algorithms detects underlying correlation between some inputs variables and some outcome. This process on its own is completely agnostic to whether or not those correlations fit within any sort of human derived ethical framework. An algorithm may notice that a particular type of candidate is hired less likely for a given role, and thus recommend to not hire candidates in that category. But what if the reason that type of person is hired less often is to due to existing biases and prejudices held by the hiring manager. This creates a feedback loop that causes the existing injustices in a system to be amplified by an algorithm.

The application of data science to biological sciences in particular is one of the most ethically difficult spaces there are. The stated goal of the research in this paper is to accelerate the speed of biological research. At first thought this sounds like a universal good - but upon further consideration one must recognize it is inextricably linked to the nature of the research that is being accelerated. Cancer research being accelerated is definitely good, whereas facilitating research into biological weapons that can target people based on their genetic profile clearly is bad. It's impossible to know all of the possible outcomes of one's research, but that doesn't make one immune from considering and attempting to ameliorate the negative consequences.

There are many levels to thinking about the ethics of this research. At the highest or most immediate level there is concern around the direct application of this (theoretical) technology - this concept is mentioned in the previous paragraph. The next level down would be to think about the secondary effects of a breakthrough in the throughput capabilities of research teams. How could it alter the economic systems that drive the research world? Would it change job prospects for lab technicians or physicians themselves? What could a possible sudden uptick in research volumes do to the business models of the labs, journals, equipment manufacturers, etc. that support and facilitate research in a general way. Given the nature of software and the internet, an algorithm, once published, can spread around the world and be adopted very quickly. Extreme Gradient Boosting, colloquially called XGBoost, is a relatively new algorithm - it was created in 2015 or so. Almost immediately XGBoost was one of the most popular algorithms out there, it was used for 17 out of the 29 Kaggle competition winners in 2015 [20].

Another angle when considering the ethics of this research is focusing on the data the models are trained on. Biology is a complex field, and there can be subtle but incredibly important differences between individuals and groups. During data collection and model training processes it is of the utmost importance that enough data is collected from all types of people - these models must be broadly generalizable or else the subsequent research will be limited to the those whose biology most closely resembles that of the training data. The underlying issue here is bias in data, and a simple Google search will reveal many cases where bias in data led to poor outcomes such as technology malfunctioning for certain types of people, or for groups to be disproportionately targeted by an algorithm.

The answers to these questions and paths toward resolving these conflicts aren't clear, but all consequences, both positive and negative, of technological research must be considered for the good of the species. This is a problem that data scientists alone cannot solve - it will require the interdisciplinary efforts of data scientists, social scientists, philosophers, regulators, and subject matter experts alike. There is no amount of professional accreditation or "best-practices" that can truly address the scale of the ethical concerns of leveraging machine learning in ways that can alter the life of so many people. Regulatory frameworks will need to be put in place to assess and monitor the risks associated with technological developments. Governments have long been the system that human societies use as a safeguard against societal level problems, and this shouldn't stop when it comes to technology.

Today's concerns are commonly focused narrowly on the effects technology can have on particular individuals. As machine learning continues to proliferate into daily life, ethical conversations need to also consider effects on a broader, societal level. Fleets of self-driving cars will be making life or death decisions on our behalf constantly, cutting edge breakthroughs in biological technology could have dramatic effects on our health as a species. Institutions will need to be in place to help shape the development of these technologies to ensure that they remain an asset and boon to human prosperity.

8 Conclusions and Future Work

In this paper microscopic images with annotated nuclear boundaries from a diverse set of body parts, captured by the process of modern techniques in digital pathology were used. The metric proposed to be used is of Intersection over Union - which seems to be an ideal choice among various metrics available in this setting for training and testing the models to obtain more generalized nuclear segmentation model as well as its further optimization. The U-Net architecture technique with Convolutional Neural Networks performed the best out of all the techniques used for nuclei segmentation during this research. Keras was used for building the Convolutional Neural Networks for model and implementing the U-Net architecture. Tensorflow was used as a backend. The Morphological Chan Vese Segmentation (edge detection technique), specifically, worked very well for identifying nuclei; however, this method fails when nuclei is extremely dense within the image.

The U-Net model built in this research had reasonable performance and generalized fairly well to test data - but ultimately did not come within the upper echelon of Kaggle scores. The team acknowledges that the model has issues in segmenting correctly the overlapped nucleus and highly complex and stained images. There is more work to be done in a variety of areas to further the effectiveness of the model. There are many options to explore - one of which is to modify the network architecture itself: use a different number of hidden layers and its corresponding output shape or change the filter size and strides for instance. Additionally, there are many image processing techniques that could be applied upstream in the model pipeline and have profound effects on the model. Image filters have been found to have extensive use in computer vision applications, that would be another good area to experiment. Smoothing and noise reduction are even more image processing techniques that could provide value to the end goal of accelerating biological research by rapidly identifying cell nuclei. The team intend to collect more data and revisit to the model development process over the next few years for further enhancements to the model in terms of accuracy and generalization.

A fundamentally different approach to future work is to focus on hardware. The training of the models in this research were largely done on personal computers and laptops. Deep learning, neural networks in particular, are broadly known for requiring large volumes of data to be accurate. As this is an era of big data, and powerful computers are needed to effectively handle that data. Due to the nature of neural networks, building these models on the available hardware was time-intensive. Leveraging cutting-edge machine learning targeted hardware would allow for more rapid iteration and experimentation of the model training process. Practically infinitely scalable high-end GPU clusters are available through major cloud service providers. Google has even developed a novel type of processor architecture, called a "Tensor Processing Unit" (TPU) which were designed and engineered specifically with running Tensorflow models in mind. The images used for training the model have thousands of pixels which get translated into enormous Numpy arrays within Python during the model training process. To efficiently perform scientific computational operations on these large numpy arrays requires high computational power. By integrating the model development process with cloud hardware specifically targeted for machine learning, the throughput of the model refinement process could enormously increase.

Keeping pathologists in mind who may be not very familiar with machine learning techniques and may not have a computer science background, the team propose to build a software that have a less complex functionalities and a easy to use graphical user interface where pathologists can easily mask the nuclei in the microscopic images by just loading the images to the software and generate the segmented images with reasonably high accuracy. This will greatly improve the diagnosis process in the medical field by improving the time currently used in segmenting the nucleus in the initial stages of the diagnosis process. The team welcomes and encourage ideas, suggestions or contributions from public towards this project.

The team strongly believe that this is a good-will project aimed at the betterment of human life and will be highly satisfied if this work contributes in speeding up the diagnosis process even by a small percentage during critical stages of life.

Acknowledgment

The authors thank Dr. Robert Slater for supporting this study with his vast knowledge and experience in machine learning and deep learning and thank Dr. Klyne Smith for reviewing and suggesting new experiments and ideas to strengthen the paper.

References

1. Booz Allen and Kaggle. Data Science Bowl - Kaggle Problem 2018
2. Humayun Irshad, Antoine Veillard, Ludovic Roux, and Daniel Racoceanu. Methods for Nuclei Detection, Segmentation, and Classification in Digital Histopathology: A Review—Current Status and Future Potential (2014).
3. Malon CD, Cosatto E. Classification of mitotic figures with convolutional neural networks and seeded blob features. J Pathol Inform 2013;4:9.
4. Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi.:A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology (Jan, 2017).
5. Abhishek Vahadane and Amit Sethi.:Towards Generalized Nuclear Segmentation in Histological Images. 13th IEEE International Conference on BioInformatics and BioEngineering. (2013).
6. Naylor, P., Lae, M., Reyal, F., & Walter, T.Segmentation of Nuclei in Histopathology Images by deep regression of the distance map. IEEE Transactions on Medical Imaging, 1–1. (2018).
7. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

8. Jain, Ramesh, Rangachar Kasturi, and Brian G. Schunck. Machine vision. Vol. 5. New York: McGraw-Hill, 1995.
9. Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2018): 834-848.
10. Chollet, François. "Keras: The python deep learning library." Astrophysics Source Code Library (2018)
11. Stephen Bailey "Step-By-Step Explanation of Scoring Metric". Data Science Bowl - Kaggle Problem 2018.
12. Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner: Machine Bias. ProPublica (May,2016)
13. Laura Douglas, AI is not just learning our biases, it is amplifying them. Medium (Dec, 2017)
14. Walt, Stéfan "Scikit Image - Morphological Chan Vese Segmentation Edge Detection". Open-Source Image Processing Library (Aug. 2009)
15. Walt, Stéfan "Scikit Image - Watershed Segmentation Edge Detection". Open-Source Image Processing Library (Aug. 2009)
16. Walt, Stéfan "Scikit Image - Canny Feature Edge Detection". Open-Source Image Processing Library (Aug. 2009)
17. Walt, Stéfan "Scikit Image - NDimage Edge Detection". Open-Source Image Processing Library (Aug. 2009)
18. Walt, Stéfan "Scikit Image - Shape Index Edge Detection". Open-Source Image Processing Library (Aug. 2009)
19. Bianco, Remi Cadene, Luigi Celona, and Paolo Mapolentano - Benchmark Analysis of Representative Deep Neural Network Architectures. University of Milano-Bicocca, Department of Informatics, Systems and Communication, viale Sarca, 336, 20126 Milano, Italy.
20. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.