Neuclei Detection

Ankita Joshi
Department of Computer Science
University of Georgia
Ankita.joshi25@uga.edu

Vibodh Fenani
Department of Computer Science
University of Georgia
vibodh.fenani25@uga.edu

Vyom Shrivastava
Department of Computer Science
University of Georgia
vyom@cs.uga.edu

Abstract

The human body is composed of trillions of cells and each cell contains nucleus. Each nucleus contains DNAs which are the genetic code that program each cell. Researchers are developing various treatments for number of diseases by understanding the biological process of how a cell reacts to a treatment. One of a challenge in this process is to identify each individual cell in a sample which is usually done by identifying nuclei. We propose a way to detect these nuclei from a given sample images by creating a convolutional neural network model that would take an image as an input and will spit out the regions where nuclei are spotted. We created a U-net model, with minor modifications in the model and proper pre-processing techniques, we achieved a score of 0.335 on Kaggle "2018 Data Science bowl" [1] challenge.

1 Overview

Today, researchers are developing various methods to cure diseases better and faster. Identifying nucleus in a cell is a promising challenge which can help researchers to perform better treatments by understanding inherent biological processes. We present a deep learning model using Unet which helps to detect nuclei from distinct images.

The problem to solve diseases using identification of nucleus from a cell sample has appeared in Kaggle competition. The solution to accurately identifying nucleus and its fundamental chemical processes is vital to diagnosis of numerous diseases and more importantly, it can speed up the different diagnostic practices as nucleus analysis can directly help researchers in finding the irregularities on genome and DNA level.

Deep learning and convolutional networks are considered very effective for image analysis especially in recent decades. It was restricted due to limited datasets and hardware capabilities. Convolutional neural network which is one of the deep learning architectures which are conventionally used for image and visual analysis. It also represents biological processes in its architecture and known to be required little preprocessing. The Unet model [2], derivative of convolutional neural network is famous for efficient and precise segmentation of biological images [3]. Our model basically takes input of large number of segmented nuclei which may be from any cell types and provides the coordinates of nucleus within that cell by performing precise image segmentation.

1.2 Problem Statement

To create a model that detects nuclei from a given sample images by creating a model that would take an image as an input and will spit out the regions where nuclei are spotted.

2 Preliminaries

The system was operational with the following for feature selection and building a segmentation model.

2.1 Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. [4]

2.2 U-Net

U-Net is a deep learning network that can be trained end-to-end from very few images and outperformed its prior state of the art models in the field of biomedical image segmentation. It uses strong data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. [2]

3 Dataset

The dataset that we will be using for this project is from the Kaggle challenge, “2018 Data Science Bowl” [5]. This dataset consists of a training and testing dataset, the training dataset contains a large number of segmented nuclei of different cell type, magnification and imaging modality. Each image in training data is associated with an image ID and contains a corresponding mask dataset which contains multiple masks for each nucleus in the image. The training dataset consists of 670 images in either Brightfield modality or fluorescence modality. The images are of different sizes but the most common image dimension in the dataset is 256x256. Each image has multiple mask associated with the same hash, each mask contains the information of a single neuron. Majority of the images are 3 channel images, but some images are grayscale single channel images.

The testing dataset consists on unseen images which were tested with our model and create the corresponding mask to identify the regions. The images are of different sizes and modality but are much more complicated then the training images, these images may or may not contain the neurons, most images contain a lot of noise and difference in magnification is very large compared to the training images. Some examples of images of the dataset are shown below.

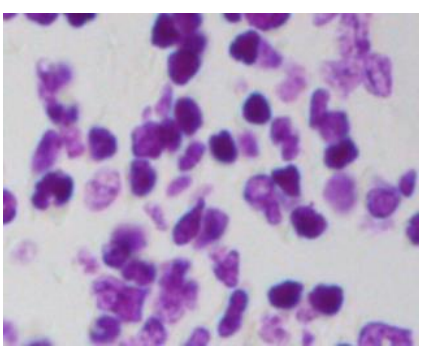

Figure 1. Brightfield Image

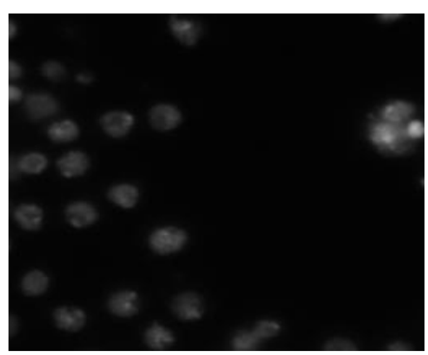

Figure 2. Fluorescence Image

Figure 3. Noisy Image with different dimension

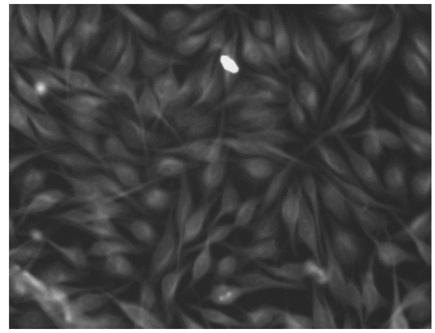

Figure 4. Image with different magnification

4 Approach

This section speaks of the preprocessing and model building strategies.

4.1 Preprocessing

Many image filtering techniques were applied on the images and the ones resulting in the best preprocessed images were used to train the model. Python library OpenCV was used to apply filters on images [6]. The following pre-processing techniques were performed on images.

4.1.1 Median Filter

Median Filter is a non-linear digital filtering technique that is used to remove noise and preserve edges. One of the biggest challenge in using U-net for image segmentation is to preserve the boundaries of the segmented entity. We used Median Filter for better edge detection.

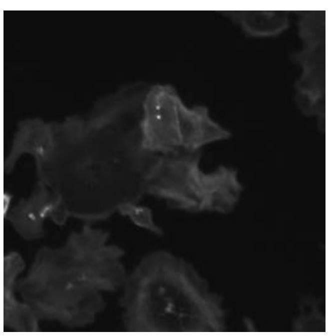

Figure 5. Original Image

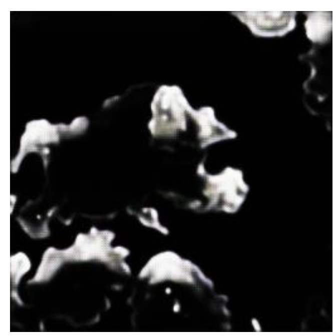

Figure 6. Prediction after Median Filtering

4.1.2 Bilateral Filter

Bilateral filter is a non-linear filter which is helpful in smoothing, removing noise and preserving edges. Again, the main reason to use this filter was to remove noise and preserve edges.

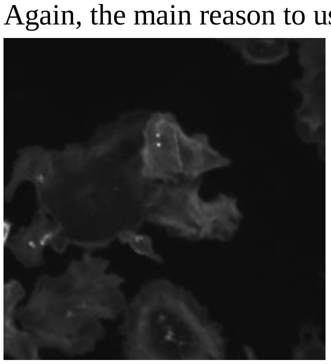

Figure 7. Original Image

Figure 8. Prediction after Bilateral Filtering

4.1.3 Band-Pass Filter

Band-Pass filter allows only pixel within an upper and lower pixel intensity bound to pass through the filter, thus removing noise from the image. As the testing images were very noisy, band-pass filtering was tried to remove extra noise.

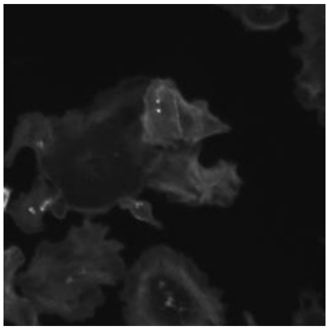

Figure 9. Original Image

Figure 10. Prediction after Band-Pass Filtering

4.1.4 Thresholding Filter

It was found that most of the noise in the images had pixel value intensity above 200, so a threshold filter was applied on the images to remove the noise from the images.

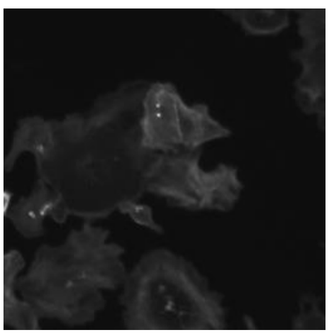

Figure 11. Original Image

Figure 12. Prediction after Thresholding Filtering

4.1.5 Thresholding-Median Filter

It was found that thresholding performed well with removing the noise but did not preserve edges well and Median filter performed well in preserving edges but did perform well in predicting body. A combination of both thresholding and median filter was applied to image to preserve edges and remove noise more efficiently.

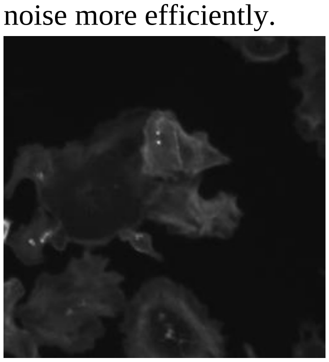

Figure 13. Original Image

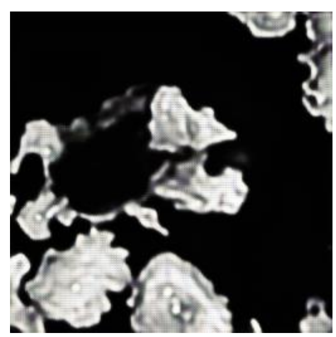

Figure 14. Prediction after combined Filtering

4.1.4 Other Pre-processing Techniques on Training Images

Other filters and pre-processing techniques like Laplacian filter, Histogram equalization were also tried but they performed well on one modality and performed poorly on other modality images and due to their poor performance during pre-processing phase they were not used further.

4.1.5 Combining Mask

The training images had multiple masks associated with each image, but the U-net architecture accepts only one mask as the label for each image. Thus, all mask associated with a single training image were combined to form a single mask which contained mask for each neuron in the image.

4.2. Building a Predictive Model

Training was performed using Deep learning segmentation architecture U-Net and a predictive model generated which was trained on 670 images from training dataset and their combined mask as labels. The U-Net model used was a modified version as specified in the U-Net paper [2], After each convolution layer in the architecture a Batch Normalization layer was added to handle the class imbalance problem. Also, a Dropout layer was added after each block of U-Net architecture to avoid overfitting.

Six models were trained, one for each pre-processing mentioned in section 4.1 and one model without pre-processing. Predictions from each model were compared manually and the best two predictions models were chosen for submission in the Kaggle Data Science Bowl 2018 challenge.

4.3. Post-Processing

The output mask for each image was converted into run length encoding format and run length encoding for each neuron was represented as a row in the table with its key as the image Hash-code. The table than generated was saved in csv format which was used for submission in Kaggle.

5 Performance

5.1. Training-Accuracy

[1] Unet with Thresholding	:	0.971
[2] Unet with Median Filtering	:	0.963
[3] Unet with Bilateral Filter	:	0.948
[4] Unet with Thresholding + Median Filter	:	0.978
[5] Unet with Band Pass Filter	:	0.890
[6] Unet without any processing	:	0.967

5.2 Testing Accuracy

Two submissions were made on Kaggle which got an accuracy of 0.278 and 0.335 on leaderboard.

Submission and Description	Private Score	Public Score	Use for Final Score
sub-dsbowl2018-median.csv a day ago by Vyom Shrivastava add submission details	0.335	0.000	☐

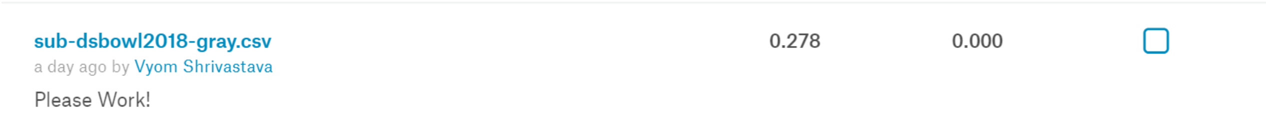

Figure 15. Submission on Kaggle

5.1 Evaluation Metric

Intersection over Union evaluation metric is used to evaluate the results. Intersection over Union can then be computed by dividing the intersection area by the union area of the two bounding boxes, taking care to subtract out the intersection area from the denominator.
This Kaggle competition is evaluated on the mean average precision at different intersection over union (IoU) thresholds. The IoU of a proposed set of object pixels and a set of true object pixels is calculated as:

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

6 Concluding Remarks

We performed segmentation using U-Net architecture and predicted the mask for all test dataset. We focused more on pre-processing images and less on using different models as this challenge had a diverse dataset that required images to be made as similar to each other as possible which can be done using better pre-processing as most of the models usually work on similar type of images.
The model resulted in an accuracy of 0.335 on Kaggle and can be further improved by applying combination of better pre-processing method and by using stronger models like Tiramisu.

7 References

[1] Kaggle "2018 Data Science Bowl" Challenge,
https://www.kaggle.com/c/data-science-bowl-2018

[2] U-Net: Convolutional Networks for Biomedical Image Segmentation
https://pdfs.semanticscholar.org/0704/5f87709d0b7b998794e9fa912c0aba912281.pdf

[3] Get acquainted with U-NET architecture + some keras shortcuts,
https://spark-in.me/post/unet-adventures-part-one-getting-acquainted-with-unet

[4] Image segmentation,
https://en.wikipedia.org/wiki/Image_segmentation

[5] Kaggle "2018 Data Science Bowl" Dataset,
https://www.kaggle.com/c/data-science-bowl-2018/data

[6] OpenCV smoothing filter documentation.
https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

[7] Kaggle Data Science Bowl 2018 Evaluation, https://www.kaggle.com/c/data-science-bowl-2018#evaluation