# DeepCell: Automating cell nuclei detection with neural networks

Cristian Bartolome (cbartolm@stanford.edu), Yiguang Zhang (yiguang@stanford.edu), Ashwin Ramaswami (ashwin99@stanford.edu)
Stanford University

## Introduction

The purpose of this project is detecting nuclei in various types of microscopic images. Finding nuclei is a very important task in the microscopic study of cell morphology. Solving this problem will help introduce more automation into the time-consuming process of identifying nuclei, which is a large bottleneck in the process of analyzing cells during drug tests.

An effective machine-learning solution will shorten drug tests, making it easier to research treatments for a wide variety of diseases.

## Data

The data given consisted of a set of 670 images provided by the Broad Institute. These images have been obtained from microscopes in many different conditions. They came with human-labeled mask files, one for each nucleus detected. We merged them to get the mask file for all the nuclei. Additionally, the images were of varying sizes, so we resized them all to a consistent size before using them for training.

Furthermore, we used techniques of data augmentation such as horizontal / vertical flipping, random cropping and the elastic transformation, which involves local distortion and affine transformation.

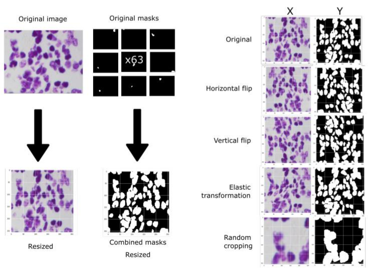


Figure 1: Data pre-processing and data augmentation

## Evaluation Metric

The model is evaluated on the mean average precision at different intersection over union (IoU) thresholds. At each threshold value $t$, a precision value is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP) resulting from comparing the predicted object to all ground truth objects:

$$\frac{TP(t)}{TP(t)+FP(t)+FN(t)}. \tag{1}$$

The average precision of a single image is calculated as the mean of the precision values at each IoU threshold:

$$\frac{1}{|thresholds|}\sum_t \frac{TP(t)}{TP(t)+FP(t)+FN(t)}. \tag{2}$$

We achieved a Mean-IoU of 0.415 in the training set and a Mean-IoU of 0.339 in the public test set of Kaggle.

## Models

### Otsu's Method

We first explored a non-neural network based method, in order to see the the applicability and limitations of a more traditional approach. Otsu's method assumes that the image contains two classes of pixels (cell pixels and non-cell pixels). It computes a threshold that minimizes the intra-class variance for both classes. This method works fine only for a very limited set of pictures (gray pictures mainly).

### Modified U-net

The U-net architecture is built upon the so-called "fully convolutional network", which supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. U-net changes the upsampling part to contain a large number of feature channels, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting path, and yields a u-shaped architecture.

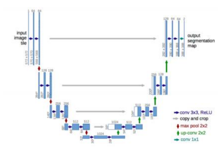

Figure 2: Original U-net architecture (example for 32x32 pixels in the lowest resolution)

We implemented various modified U-net architectures by adding batch-normalization and regularization.

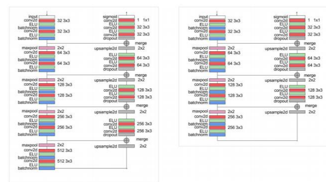

Figure 3: Modified U-net architectures.

## Results

### Otsu's Method

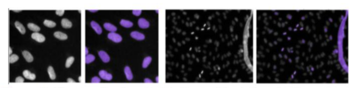

Figure 4: Results for Otsu's Method. The left pair shows a successful classification. However, the right pair evinces the limitations of this algorithm

### Modified U-net

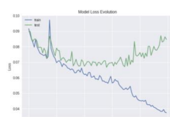


Figure 5: Original U-Net

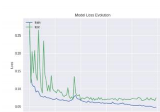


Figure 6: Modified U-net

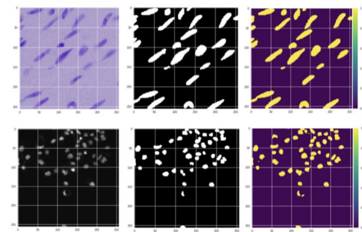

Figure 7: Modified U-net results: Original image, ground truth mask and predictions (left, center and right, respectively)

## Discussion and Future Work

The above results show that:

- The modified U-net is less prone to overfit as it includes dropout layers in the post-merge paths and an extra L2/weight-decay regularization term in all the convolutional filters of the encoder
- The addition of Batch-Normalization after the activation layers in the encoder blocks has proven to be beneficial
- Hyperparameter tuning reveals that exponential linear units (ELU) are more suitable for this problem than ReLU units after the Conv2D layers
- The modified U-net is slightly more computationally expensive than the original architecture (the number of parameters increases by less than 1%) . As future work, we plan to implement TernausNet: a new modification of U-net

## References

[1] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. *"U-net: Convolutional networks for biomedical image segmentation."* Springer, Cham, 2015.

[2] Simard, Steinkraus and Platt, *"Best Practices for Convolutional Neural Networks applied to Visual Document Analysis"*, 2003.