

Capstone Proposal
Machine Learning Nanodegree
Dillon Donohue

Applied Image Segmentation - Cell Nuclei

Domain Background

This project aims to study and apply image segmentation to biological cells for the purpose of automating nucleus detection. Medical research, including drug development, pathological research, and tissue engineering, all begin with studying cell cultures and how they respond to chemical and physical stimuli. This project is motivated by the 2018 Data Science Bowl competition held by Kaggle (www.kaggle.com). The resulting computer vision model will be used in the competition. Image segmentation of human cells, and mammalian cells in general, is a core component of biological and medical research. The importance stems from the vast amount of information learned from studying cell phenotype and morphology. For this reason, computers have been employed to automate the task of identifying and interpreting cells for the past several decades (Meijering, 2012). The cell nucleus is the focal point for image segmentation in this project because it is often the most easy to see feature of an individual cell.

I am personally very motivated to contribute to the effort of improving automation of cell segmentation because I personally have spent many hours manually labeling and masking images of cells. As an undergraduate studying Chemical and Biological Engineering at the University of Colorado Boulder, I worked in a tissue engineering lab that extensively studies human cells and their interactions with synthetic polymers designed to emulate biological conditions. I fully recognize how freeing scientists from the task of manually studying images of cells can both increase throughput of images and reduce variation in segmentation that arises from human interpretation. Enhanced throughput of cell images has the power to expedite medical and health advancements by allowing scientists to focus on the more challenging aspects of research and increase the throughput of research itself.

Problem Statement

The focus of this project is to employ computer vision and convolutional neural networks (CNN) for the task of identifying and masking cell nuclei in divergent images. This computer vision model will be built with the intention to generalize across multiple input variables including cell type, magnification, and image modality (brightfield and fluorescence) as seen in Figure 1. The ultimate goal is to push the state of the art technology in cell nuclei segmentation by submitting the most performant model from this project to the 2018 Data Science Bowl competition hosted by Kaggle.

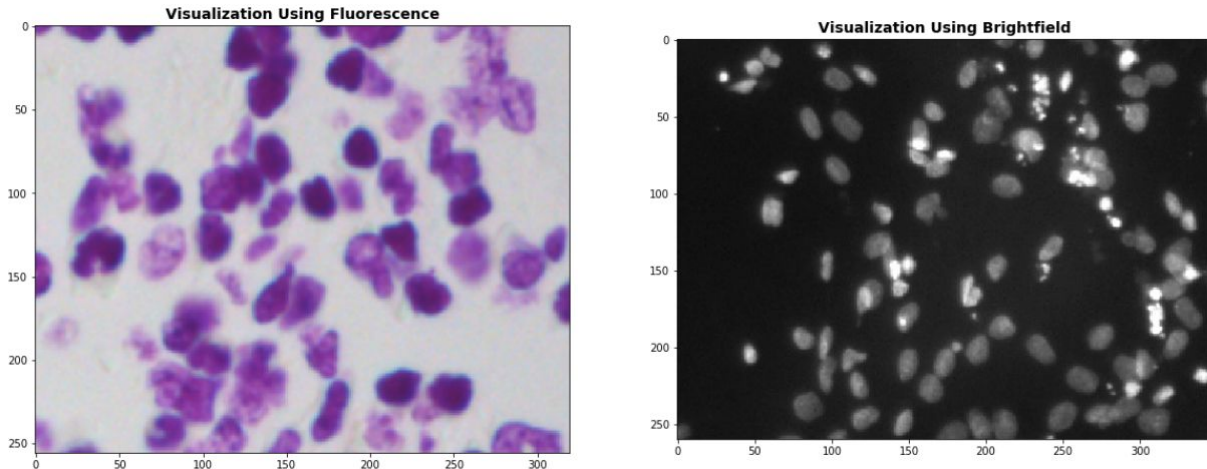


Figure 1 - Raw image samples from training dataset.

Possible solutions to this project and Kaggle competition include classical computer vision and modern CNN architectures that have been designed for similar image segmentation tasks. For the purposes of this project, classical computer vision techniques are those that rely primarily on background vs. foreground classification and edge detection.

Dataset

The dataset used in this project will be the dataset provided by Kaggle for the 2018 Data Science Bowl competition ([kaggle](https://kaggle.com/c/data-science-bowl-2018)). This dataset provided by Kaggle has already been broken into training and testing sets. The training set contains 670 image samples. Each raw image has an associated set of segmentation mask images for each nucleus found in the image. For example, if a raw image contained 10 nuclei there would be an accompanying 10 mask images. One for each segmented nucleus. The segmented mask images serve as labels for each nucleus in the original image. Therefore the masks are used to calculate error between the models predicted mask and the human annotated mask (supplied mask). This is the error that will be used for backpropagation through the model to update the model parameters. Additionally, the dataset contains a training labels csv file that stores the mask information for each nucleus in the training image. The csv file is a redundant form of labels for the training images.

The test set in the dataset holds 65 images for evaluating model performance. The competition is broken into two stages and two testing sets. During the first stage of the competition, the test set only contains images and therefore model performance on this set can only be evaluated by submitting the model predictions to Kaggle. When the second stage of the competition begins, the labels for the first test set will be released. At this time a second test set will also be released. The second stage test set will serve as the final evaluation of model performance.

This dataset is ideally suited for this project as it provides high quality training data specifically focused on cell nuclei. The images and masks were hand selected and annotated by scientist from the Broad Institute of MIT and Harvard for the purpose of nucleus segmentation

and masking. Additionally, it should be mentioned that the images have been released under the Creative Commons license 0 and are therefore within the public domain ([2018 Data Science Bowl](#)).

Solution Statement

Image segmentation, and specifically image segmentation of cell nuclei, performed by computer vision techniques is far from a novel concept. There are several deep CNN models that perform well on these types of tasks. Two models that will be considered and explored as solutions to nuclei segmentation are the U-Net model and the Mask R-CNN model. The U-Net model was developed by at the University of Freiburg for the purpose of biomedical image segmentation (Ronneberger, Fischer, & Brox, 2015). Alternatively, the Mask R-CNN model was developed by Facebook AI Research as a flexible model that generalizes well for object segmentation (He, Gkioxari, Dollar, & Girshick, 2017; Ronneberger et al., 2015). These models will be explored as possible solutions for this project by first obtaining publicly released model architectures followed by training and testing the models.

The U-Net model architecture can be obtained from Github found here: <https://github.com/jocicmarko/ultrasound-nerve-segmentation>. The model will have to be trained using the Kaggle provided dataset. The Mask R-CNN model and pre-trained weights from the MS COCO dataset are available on Github found here: https://github.com/matterport/Mask_RCNN. The pre-trained weights can be omitted in favor of training the model on the Kaggle dataset.

The performance of these models on the Kaggle dataset will be determined using the testing set to generate segment masks of the nuclei in the images. The accuracy of the generated masks will be measured against a ground truth set of masks held by Kaggle to assign a model score for the test set.

Benchmark Model

The benchmark model will be based on classical computer vision techniques. Specifically this model will rely on the computer vision library OpenCV for python. The underlying approach of this pure computer vision, benchmark model is separating foreground from background in the image through statistical analysis of pixel intensity. Ideally, each image has a bimodal distribution of pixel intensity where a threshold intensity can be specified to classify all pixels into two groups. A foreground group and a background group. Additional image processing techniques including opening and closing can be used to improve separation of adjacent nuclei. Finally, all foreground objects can be counted and their mask areas calculated. The benchmark model can generate moderate results for nuclei segmentation masking and requires no training as it employs no machine learning techniques. The test set of images will be run through the benchmark model to generate nuclei masks. These masks will be compared to the ground truth set of masks held by Kaggle to assign a score for the model. With this benchmark model score, we can quantify the benefits of the solution model and therefore the strengths of convolutional neural networks in image segmentation.

Evaluation Metrics

Conceptually speaking, evaluation of the benchmark model and solution model are performed by comparing the segment masks produced by the model with a set of ground truth masks held by Kaggle. The ground truth masks for the first test set will be released by Kaggle when stage two of the competition begins. For convenience, I anticipate using a portion of the training data as a validation set. Due to the relatively small size of the training set, k-fold cross-validation will be used to make all training data available for training. For this project, Kaggle's evaluation metrics will be used to both determine performance of the solution model and compare this model's performance with the Kaggle leaderboard. The evaluation metrics for the 2018 Data Science Bowl are a comparison of predicted masks and ground truth masks using intersection over union (IoU) ([2018 Data Science Bowl Evaluation](#)). Intersection over union of a set of predicted pixels in an object versus the true pixels of the object is found using the equation:

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

A series of IoU threshold values have been defined over which an average precision will be calculated for all masks in an image for a given threshold. A predicted mask is considered correct if the IoU of the predicted mask versus the ground-truth mask is greater than the threshold value. The threshold values have been specified as 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95. Precision for a given threshold and image is calculated as:

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

In the above equation, TP is the count of true positives, FP is false positives, and FN is false negatives. A mean precision for an image is then calculated by summing the precision for each IoU threshold and dividing by the number of thresholds:

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

Finally, the mean precision for each image is averaged over all images in the test set to produce a mean average precision. The mean average precision is the metric used in this project and by Kaggle to assign a score for each model.

Project Design

The objective of this project is detecting and identifying cell nuclei through image segmentation and masking. Well established computer vision techniques have been applied to this challenge and have demonstrated moderate results. This project aims to apply recent advances in convolutional neural networks to this image segmentation challenge. The ultimate goal is to enhance the automation of this task and help drive medical research and innovation via the 2018 Data Science Bowl.

The first step in this project, as with most machine learning projects, is reviewing and cleaning the input data. This project will be using image data that comes as both black and white images and colored images as well as varying cell types and image magnification. In addition to the raw images, the training dataset also includes segmented and masked nuclei

generated from the raw images. These masks must also be reviewed for correctness with respect to the nucleus they are intending to mask. Discrepancies between mask images and the nuclei in the original image can be manually annotated or allowed to remain. Ultimately, resolving discrepancies will lead to better CNN training results. See Figure 2 for an example of a nucleus that was not masked in the provided training set.

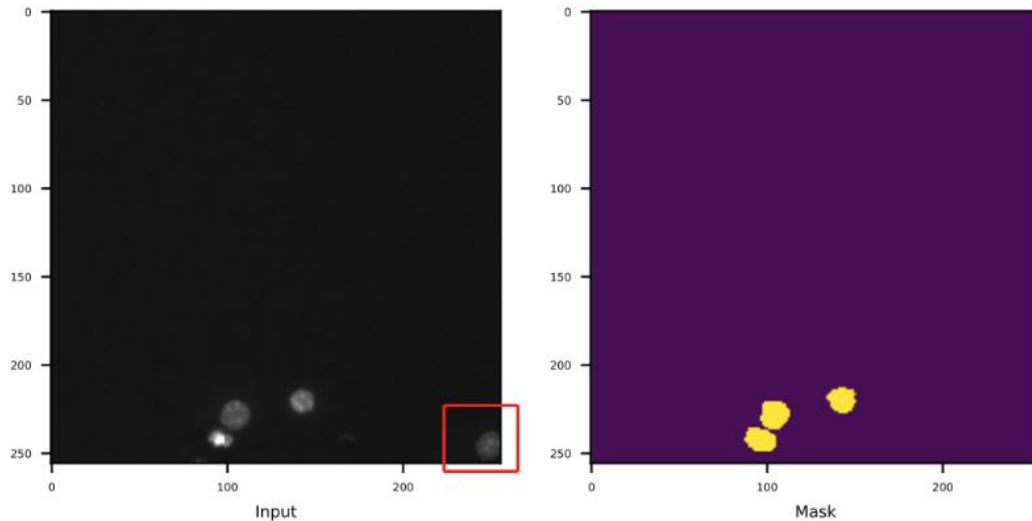


Figure 2 - Nucleus in bottom right corner was not segmented and masked in the provided training set. Manually annotating the masked training images like this one will improve solution model. ([2018 Data Science Bowl - Data Quality Issues Discussion](#))

The next step in the project will be building a simple computer vision model based on existing python libraries and techniques. A sample model using the OpenCV library can be seen below. It should be noted that this model requires no training as it does not employ machine learning techniques. See Figure 3 for source code for a simple computer vision model. The source code seen in Figure 3 is a fork of the code found here: <https://www.kaggle.com/ahassaine/pure-image-processing-lb-0-274>

```

import os
import cv2
import numpy as np

test_dirs = os.listdir("data/stage1_test")
test_filenames=["data/stage1_test/" + file_id + "/images/" + file_id + ".png" for file_id in test_dirs]
test_images=[cv2.imread(imagefile) for imagefile in test_filenames]

def segment_mask(img_rgb):
    # convert color image to gray scale
    img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)

    # apply morphology transformation to open gaps between adjacent nuclei by suggesting shapes are ellipses
    ellipse = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (7,7))
    img_open = cv2.morphologyEx(img_gray, cv2.MORPH_OPEN, ellipse)

    # use OTSU method of thresholding for bimodal pixel intensity
    img_th = cv2.threshold(img_open, 0, 255, cv2.THRESH_OTSU)[1]

    # invert the image if nuclei are dark and background is light i.e. fluorescence microscopy image
    if (np.sum(img_th == 255) > np.sum(img_th == 0)):
        img_th = cv2.bitwise_not(img_th)

    # perform morphological opening on binary image
    img_mask_open = cv2.morphologyEx(img_th, cv2.MORPH_OPEN, ellipse)

    # segment masked nuclei
    seg_masks = cv2.connectedComponents(img_mask_open)[1]

    return seg_masks

test_segmented_masks = [segment_mask(img) for img in test_images]

def rle_encoding(segment_masks):
    values = list(np.unique(segment_masks))
    values.remove(0)
    RLEs = []
    for v in values:
        dots = np.where(segment_masks.T.flatten() == v)[0]
        run_lengths = []
        prev = -2
        for b in dots:
            if (b > prev + 1):
                run_lengths.extend((b + 1, 0))
                run_lengths[-1] += 1
                prev = b
            RLEs.append(run_lengths)
    return RLEs

test_RLEs = [rle_encoding(segmented_img) for segmented_img in test_segmented_masks]

with open("submission_improved_cv.csv", "a") as myfile:
    myfile.write("ImageId,EncodedPixels\n")
    for i, RLEs in enumerate(test_RLEs):
        for RLE in RLEs:
            myfile.write(test_dirs[i] + "," + " ".join([str(i) for i in RLE]) + "\n")

```

Figure 3 - Sample computer vision model for cell nuclei segmentation and masking.

The performance of this model will be determined by running the test set through the model and submitting the produced segment masks to Kaggle for scoring. This model can be evaluated without uploading outputs to Kaggle by applying k-fold cross-validation on the training set to generate a test set. This test set will be scored manually using the same methodology as used in the 2018 Data Science Bowl.

The primary focus of the project will fall on developing and testing the solution model. This challenge will begin with exploring two established models for image segmentation, the U-Net model from the University of Freiburg and the Mask R-CNN model from the Facebook AI

Research group. The model architecture for Mask R-CNN is available on Github along with pretrained weights. Experimentation with this model will begin with importing and using the pretrained weights to establish baseline performance of this model. Next, retraining the model on the training dataset will be the first step in improving the model's performance. The Mask R-CNN model allows for selectively turning "on" and "off" layers during training. This feature will be used to initially focus on retraining the top layers of the model. Additionally, model parameters and hyperparameters will be explored and tuned using grid search to determine appropriate settings and values.

The U-Net model architecture is publicly available and can be found on Github. This model will have to be fully trained on the training dataset as there are no pretrained weights for any particular dataset currently available. The U-Net model architecture is significantly less deep than Mask R-CNN allowing for much quicker training times. Along with training the model, parameters and hyperparameters will be tuned using grid search to narrow in on an optimal configuration of the model. See Figure 4 for example of touching or overlapping nuclei being segmented and masked as a single nucleus as a result of simple segmentation techniques. The solution model will strive to better segment adjacent nuclei.

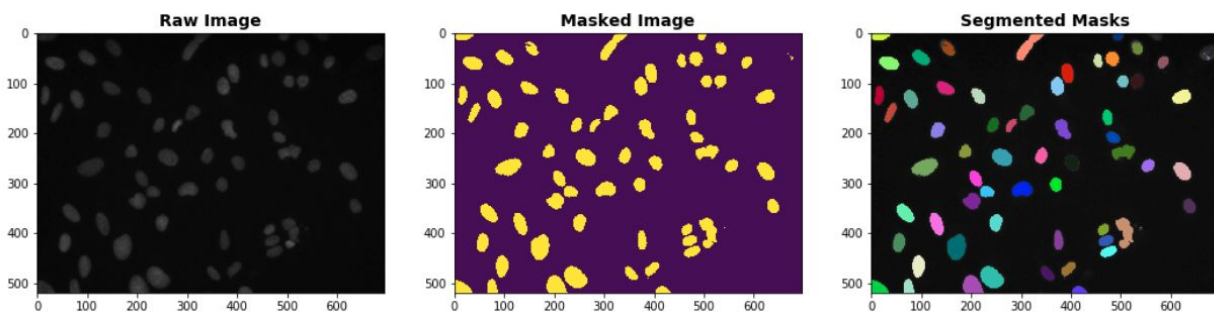


Figure 4 - Raw image, masked image, and segmented masked image. Image analysis performed using "classical" computer vision techniques. Note, touching or overlapping nuclei are considered one nucleus and is an incorrect interpretation by the model.

Evaluation of the model will be done following the methodology specified by Kaggle in the 2018 Data Science Bowl competition. Segment masks produced by the benchmark and solution models will be converted into a run-length encoding (RLE) file of the masks. The RLE file is then fed into the scoring algorithm to calculate mean average intersection over union of the segment masks generated on the test set versus the ground truth masks for the test set. The resulting score will be used to quantify performance of the model for generating segment masks. The RLE file generated by the solution model will also be submitted into the 2018 Data Science Bowl to compare with other competition entries.

References

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. <https://doi.org/10.1109/iccv.2017.322>

Meijering, E. (2012). Cell Segmentation: 50 Years Down the Road [Life Sciences]. *IEEE Signal Processing Magazine*, 29(5), 140–145.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Lecture Notes in Computer Science* (pp. 234–241).

<https://www.kaggle.com/c/data-science-bowl-2018/data>

<https://www.kaggle.com/c/data-science-bowl-2018/discussion/47864>

<https://www.kaggle.com/c/data-science-bowl-2018#evaluation>

<https://www.kaggle.com/c/data-science-bowl-2018/discussion/47770>

<https://github.com/jocicmarko/ultrasound-nerve-segmentation>

https://github.com/matterport/Mask_RCNN

<https://www.kaggle.com/ahassaine/pure-image-processing-lb-0-274>