

Cell Segmentation and Nuclei Detection

Debdas Ghosh

Email: deg107@pitt.edu

Piu Mallick

Email: pim16@pitt.edu

Siyu Cai

Email: sic52@pitt.edu

April 21, 2020

Abstract

We address a central problem of neuroanatomy, namely, the automatic segmentation of neuronal structures depicted in stacks of electron microscopy images. Low contrast of such images hinder image segmentation and visualization of these data-sets. With recent advances in computer vision and deep learning, such data-sets can be segmented and reconstructed with greater ease and speed than with previous approaches. In this study, we implement Mask R-CNN, a deep convolutional neural network for instance segmentation. We have evaluated the model using two data-sets, one is for cell segmentation, and another is for nuclei detection. We show that Mask R-CNN can achieve reliable performance on tasks of neuronal structures segmentation.

1 Introduction

Among a variety of imaging modalities, segmenting histopathology images is challenging. Such data-sets are notoriously difficult to work with, as the texture and intensity variations across all parts of the images are similar. Automatic cell nuclei segmentation and detection of histopathology images is an important task in medical image processing and analysis. Identifying nuclei allows pathologists to identify each individual cell in a sample, and by detecting the shape of cells, especially cancerous cells, the pathologists can determine the severity of the cancer. In this project, we will compare and contrast between two methods/techniques, namely **U-Net** and **Mask R-CNN**. However, upon detailed discussion and analysis, we will choose Mask R-CNN technique to segment cell nuclei in histopathology images.

2 Objective

Our project aims at implementing instance segmentation, which is the task of identifying object outlines at the pixel level. We will consider this task as a binary classification, which is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule. In our project, the object is the positive target and the background as the negative target.

3 Hypothesis

The problem follows a NULL hypothesis as the pixels do not belong to the cell segments. If the pixel belongs to the cell segment, we reject the NULL hypothesis and accept alternate

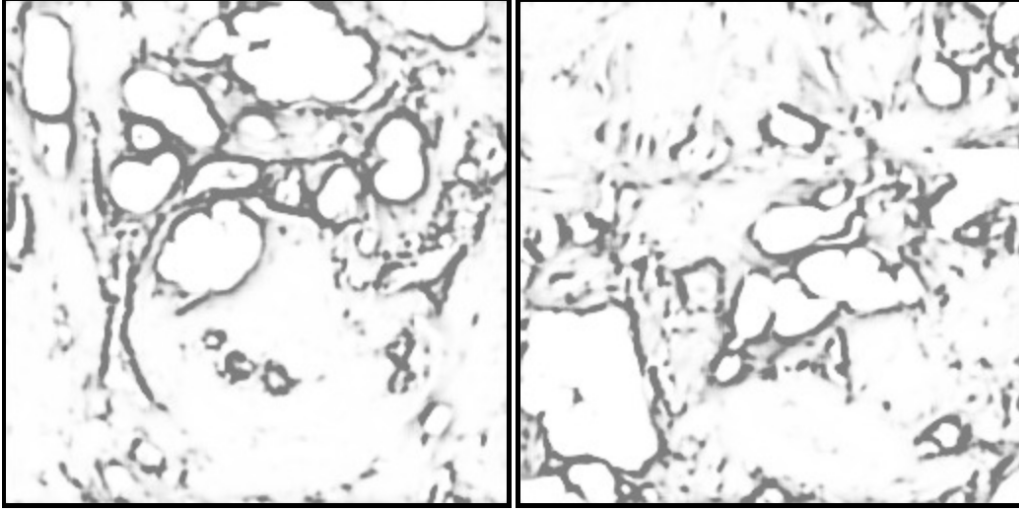
hypothesis.

4 Methods/Models Involved

4.1 Initial Method: U-Net

Initially, we used **U-Net** [1] (a convolutional neural network that was developed for biological image segmentation at the Computer Science Department of the University of Freiburg, Germany: it is based on fully convolutional neural network and its architecture was modified and extended to work with fewer training images and to yield more precise segmentation) and obtained results like shown in the **Figure 1**:

Figure 1: **Example of U-Net results.** Segmentation did not work well; quite blurry and distorted. It cannot be segmented only by the instances of the cancerous cell regions, instead it segmented at different regions in a semantic segmentation method.

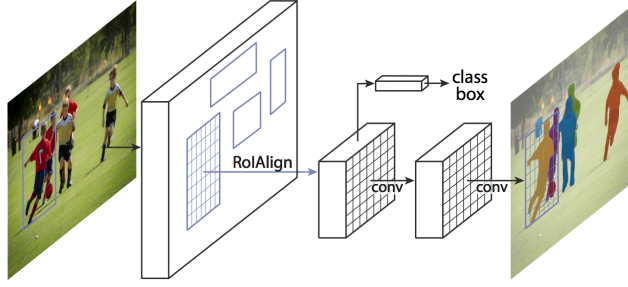


The problem with **U-Net** is that it does not split into adjacent masks, which is why we needed to modify it in one way or another to split the output. We will be discussing this in the next section of our paper.

4.2 Final Method: Mask R-CNN

We then implemented **Mask R-CNN** [2] on **Python 3**, using **TensorFlow**. **Mask R-CNN** is a deep learning model that generates a bounding box and mask for each instance of an object. This is a great foundation for instance segmentation projects and has produced better results than other models. Mask R-CNN consists of two stages: the first stage scans the image and generates proposals, which are the areas likely to contain an object, and the second stage classifies the proposals and generates bounding boxes and masks.

Figure 2: Mask R-CNN framework (He, Gkioxari, Dollár, Girshick, 2017).
Model architecture of Mask R-CNN.



5 Experiments Involved

5.1 Exploring the Data

We used two publicly available data-sets in our project. The first data-set was obtained from Case Western Reserved University’s publicly available data-set for Nuclei Detection (Xu, Xiang, Hang, Wu, 2014)[3]. It contains over 40,000 primary histological images. We selected 68 images for training and 15 images for validation, and then manually annotated cancerous cell boundaries on them using an open sourced tool **VGG Image Annotator (VIA)** by University of Oxford (Dutta Zisserman, 2019)[4]. **VIA** generates a JSON file consisting of the coordinates of the masks. The second data-set includes a large number of segmented nuclei images across different types of scans. It is sourced from the 2018 Data Science Bowl Competition on Kaggle, which targets nuclei detection (2018 Data Science Bowl, n.d.). We extracted 670 patches for training, and 65 patches for validation.

5.2 Interesting Summary Statistics

Some interesting statistics were noticed while data exploration. The cell image sizes are 2200 px X 2200 px and the nucleus image height is approximately 256 px, but they are of different widths. It was quite challenging to detect the cells and segments as they do not have any specific size or shape or color.

5.3 Training the data

We trained the cell segmentation model on the first data-set, and trained nuclei segmentation model on the second data-set. Since we annotated the images manually, which is quite time-consuming, the data-set is tiny and hardly met the training requirement for a deep learning model. In order to compensate, we involved the transfer learning method, which means that instead of training a model from scratch, we started with a weights file that has been pre-trained on the **COCO** [5] data-set. It is a large-scale object detection and segmentation data-set containing 330K images across a range of objects. By incorporating those data, the trained weights have already learned a lot of features common in object detection, which helps in the further cell segmentation task. In the process of training (ResNet50 backbone model), the learning rate was set to 0.001, and weight decay was set to 0.0001 in Stochastic Gradient Descent optimizer. The training and validation class was set to 2 (i.e. cell and background) for both experiments. Because we were using a very small data-set, and starting from the trained weights, we only trained a part of layers.

5.4 Special Considerations for the execution of the model

Initially, the model execution was started in a standalone machine. However, the system crashed and we had to switch to **Google Colab**, with an initial RAM requirement of 12.5 GB. We faced the same issue with RAM requirement there as well until Google Colab itself recommended and upgraded its RAM requirement to 25 GB in order to accommodate the training and it took hours in training. Hence, all the experiments were performed on a GPU cluster on each node at Google Colab, because of the high GPU and RAM requirement.

5.5 Reasons behind choosing one model over the other (Mask R-CNN over U-Net)

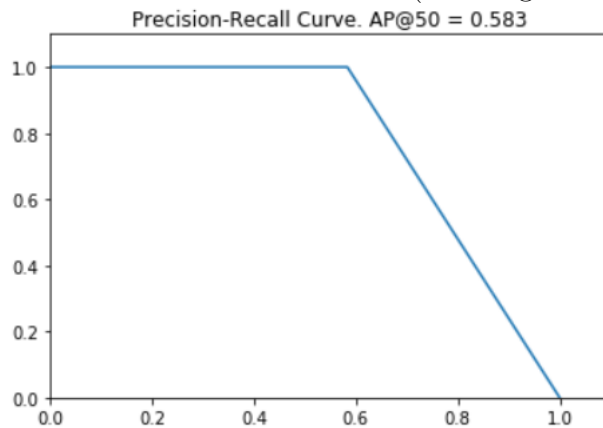
U-net extended the architecture of Fully Convolutional Networks (FCNs) and utilized up-sampling in opposite expansive path with skip connections to obtain the output mask within an end-to-end learning framework (Helmstaedter, et al., 2013). It has been widely used by both medical and biological image analysis communities and is still regarded as one of the most successful models in this context. Such method depends on large ground-truth samples, however, according to one study (Xu, Xiang, Hang, Wu, 2014), manual data segmentation or annotation costs \$10 per m3 on average for EM volumes, which might incur thousands of dollars for a relatively large data volume. Additionally, providing ground-truth samples for microscopy images is challenging due to their inherently low contrast. Hence, although U-Net is less computationally expensive, Mask R-CNN is a better option since it provides better results under the circumstances of limited training data. Moreover, U-Net does not split into adjacent masks, whereas Mask R-CNN enables to simultaneously generating a high-quality segmentation mask for each instance while efficiently detecting objects in an image.

6 Results

6.1 Precision-Recall Curve (Cell Segmentation)

After training with only 68 images we got good object detection results, with mean average precision 0.583.

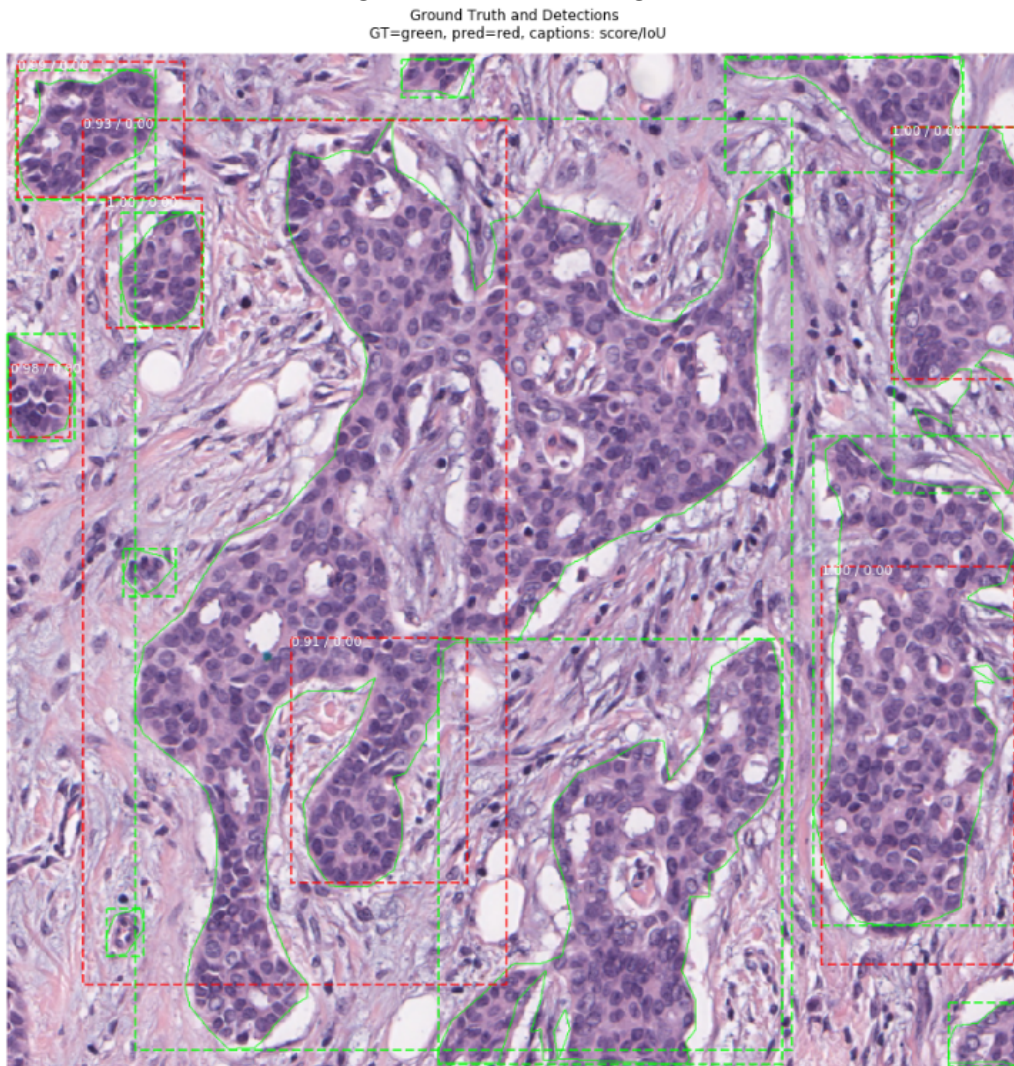
Figure 3: Precision-Recall Curve (Cell Segmentation)



However, we did not get masking. Our understanding is that the model needs more training images, as it cannot learn features from such a small data-set. Although, we used pre-trained weights for COCO data-set, the COCO image categories (like cat, dog, person, aeroplane etc.)

are very different from microscopic images. Apart from that, cell regions have no particular shape unlike person or aeroplane, and thus making it a difficult task to identify microscopic images.

Figure 4: Detected Regions



6.2 Precision-Recall Curve (Nuclei Detection)

We got a good mAP of 0.795 as well as accurate masking.

Figure 5: Precision-Recall Curve (Nuclei Detection)

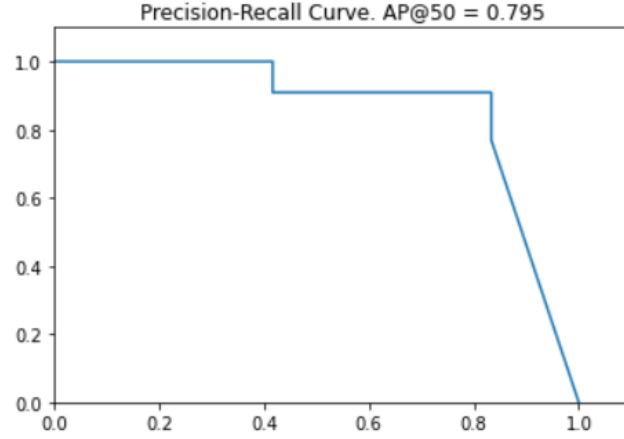
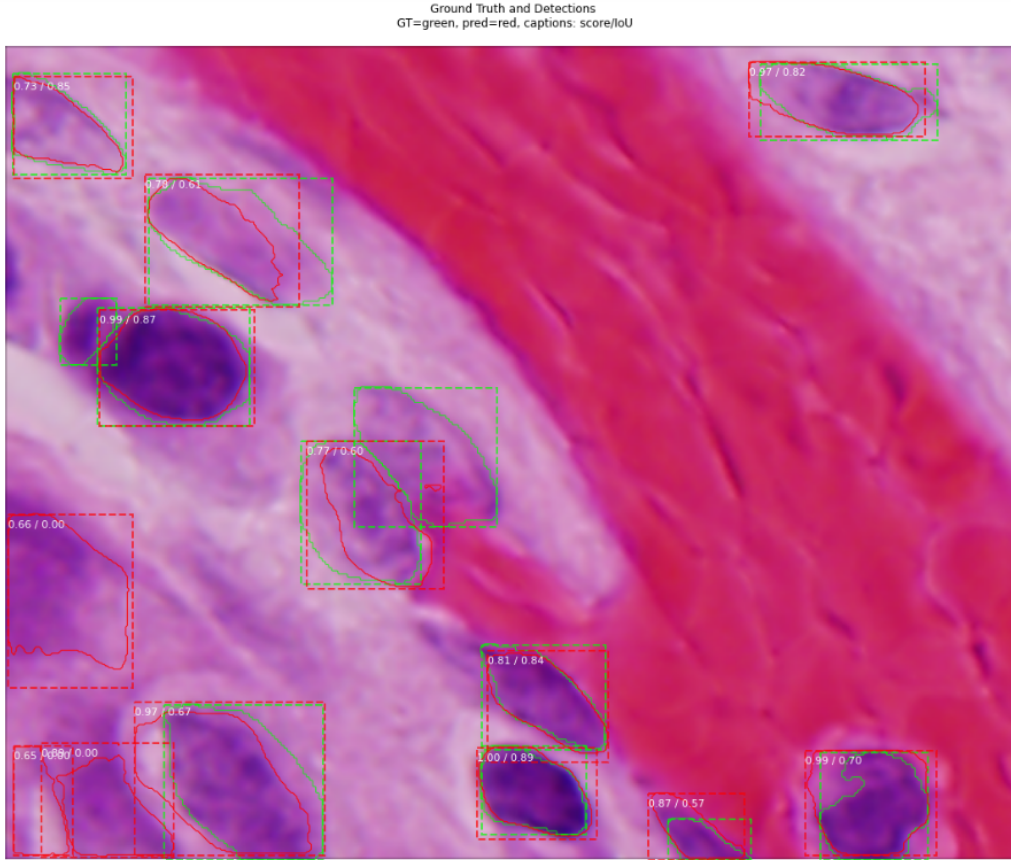


Figure 6: Detected Nucleus



7 Challenges Faced

Few challenges were faced while implementing our models, which will be discussed briefly below:

- Annotating each image by VGG Image Annotator (VIA) is a very time-consuming process and it took almost 1-2 minutes for annotation per image. Hence, masking for 68 images manually was a time-cumbersome process.
- Executing the code-base on a standalone machine was a huge challenge. Switching on to Google Colab gave us a breath of relief in terms of high GPU and RAM usage. However, we needed to upload the data-sets, weight files and configuration files and libraries each and every time while starting a new session.

8 Future Work

Looking ahead in the future, the following improvements could be made:

- More training images could be incorporated in order to get image mask prediction for cell segments.
- Experiments can be done by ensembling both the models used in our experiments separately, namely U-Net and Mask R-CNN, in order to get better results. We haven't explored this part yet due to time constraints. However, we look forward to experimenting more on this in future.

9 Confidence on the Results

Post-execution of the model, we have 79.5% confidence rate on nuclei detection and we are pretty confident that with the help of more training data, we can improve on the accuracy. Nonetheless, even with an 79.5

10 Keywords

R-CNN: Convolution Neural Network (by Region)

COCO Images: Common Objects in Context

mAP: Mean Average Precision

References

- [1] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *arXiv*, 2015.
- [2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” *arXiv*, 2018.
- [3] X. Jun, X. Lei, L. Qingshan, G. Hannah, W. Jianzhong, T. Jinghai, and M. Anant, “Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images,” *IEEE*, 2015.
- [4] A. Dutta and A. Zisserman, “The via annotation software for images, audio and video,” *ACM*, 2019.
- [5] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft coco: Common objects in context,” *arXiv*, 2015.