Automated Nuclei Detection

Suvrat Bhooshan
Department of Computer Science
Stanford University
suvrat@stanford.edu

Aditya Garg
Department of Computer Science
Stanford University
agarg94@stanford.edu

William Marshall
Department of Computer Science
Stanford University
wfm@stanford.edu

1 Introduction

In our project we built a system to detect the location of cell nuclei in microscopy images. In medical research, it is often the first step to find the nuclei of cells in a particular sample in order to assess their health or the affect a drug is having on them. However, there is currently no robust autonomous system to reliably perform this task. As much of the rest of a typical research process is already automated, nuclei detection is currently a large bottleneck as it needs to manually done. Automating the process of identifying nuclei allows for more efficient drug testing, and can shorten the 10 years it takes for each new drug to come to market [1]. Here, we modify state of the art Deep Learning Segmentation Models to detect nuclei in microscopy images.

2 Literature Review

Nuclei detection falls under the broader category of image segmentation which is a topic of much current research in the vision community. Fully convolutional networks (FCNs) have been shown to have significant success on image segmentation tasks and have been one of the most trusted architectures for this task. Traditional FCNs however are not instance aware meaning that they can not distinguish multiple instances of the same type of object from each other. Methods have been developed to separate pixels of the same class into different instances and have met with varying results. Recent work by Le et al solved this problem by creating the Fully Convolutional Instance Segmentation (FCIS) architecture [4]. Although FCIS still struggles somewhat with discerning instances of the same class and still makes systematic errors on overlapping instances, it still performs very well on many image segmentation tasks [4]. The other major approach to image segmentation is the Mask R-CNN. The Mask R-CNN is motivated by the successes of region proposal based networks for other vision tasks such as Faster R-CNN [2]. Unlike the "segmentation first" strategy of the FCN, the Mask R-CNN takes an "instance first" approach, and it is a relatively simple to the effective Faster R-CNN architecture. Mask R-CNN is the current state of the art [3].

There have also been previous attempts specifically at the problem of nuclei detection. Notably Ronneberger et al produced a model that performs quite well on a similar task in the segmentation of microscopy images [5]. They use a relatively small FCN architecture which helps reduce complexity in the absence of large amounts of data as is common for many medical applications. The authors also discuss the significant impact that data augmentation had on their results. Their techniques included standard shifts and rotations as well as grey scale variations and smooth deformations using bicubic interpolation [5]. These techniques show promise for our application.

Additionally, Ciresan et al attempted a similar microscopy segmentation problem and attempted to use a different architecture to overcome the relative lack of data. They trained a network that used a

sliding window to predict the value class of the pixel at the center of the window [6]. This method, however, has several drawbacks as a a sliding window approach requires much more computation at test time and so is slow. Additionally this approach is again a "segmentation first" approach, and so there would be significant difficulties extending it to an application which required instance level identification. The results of this network to not come close to human performance [6].

Still other approaches seek to avoid the difficulties of instance level segmentation by predicting the location of the middle of a nucleus instead of the nucleus itself or by exploiting properties specific to a certain type of image and microscopy technique [7] [8]. However, even by attempting to solve a slightly different problem none of the approaches has had significant success, underscoring the need for better solutions. We sought to combine the domain insights from these works with the recent success shown by instance first segmentation models to create a more accurate and useful model.

3 Approach

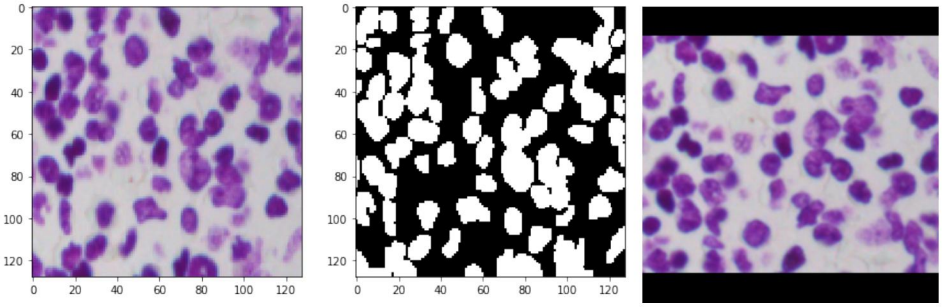
3.1 Dataset

We are using data from a pre-prepared and labeled dataset online from the 2018 Kaggle Data Science Bowl[1]. This data had the advantage of already being cleaned and labeled by medical experts. We did have to perform data preprocessing to restructure it in the format acceptable by both our deep learning models. The dataset comprises of 670 images which we split into Training (80%), Validation (10%), and Test (10%) sets. There are roughly 25,000 nuclei distributed in these images. There are 6 different types that the types of images can be classified into.

- Black foreground and white background
- Purple background and (purple, white, yellow) foreground
- White foreground and black background

3.1.1 Data Augmentation

As 670 images is quite a small dataset, we performed data augmentation to increase its size. We used vertical mirroring, horizontal mirroring, and rotations at 90 degree increments in order to increase the size of our training set by over thirty times. None of these transformations change the essential characteristics of the nuclei, and so all are valid for use in augmentation.



(a) Dataset Image (b) Image Mask (c) Augmented Image (horizontal axis flip)

Figure 1: Dataset Examples

All input images were re-shaped to 256×256 and padded with zeros in order to preserve the image scaling.

3.2 Error Metrics

We focused 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:

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

The metric sweeps over a range of IoU thresholds, at each point calculating an average precision value. The threshold values range from 0.5 to 0.95 with a step size of 0.05. I.e., at a threshold of 0.5, a predicted object is considered a "hit" if its intersection over union with a ground truth object is greater than 0.5. 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. A true positive is counted when a single predicted object matches a ground truth object with an IoU above the threshold. A false positive indicates a predicted object had no associated ground truth object. A false negative indicates a ground truth object had no associated predicted object. The average precision of a single image is then calculated as the mean of the above precision values at each IoU threshold [1].

$$\text{Mean Average Precision} = \frac{1}{|\text{thresholds}|} \sum_t \frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

3.3 Algorithms

We built our models from two main architectures: U-NET and Mask R-CNN [5, 3].

3.3.1 U-Net

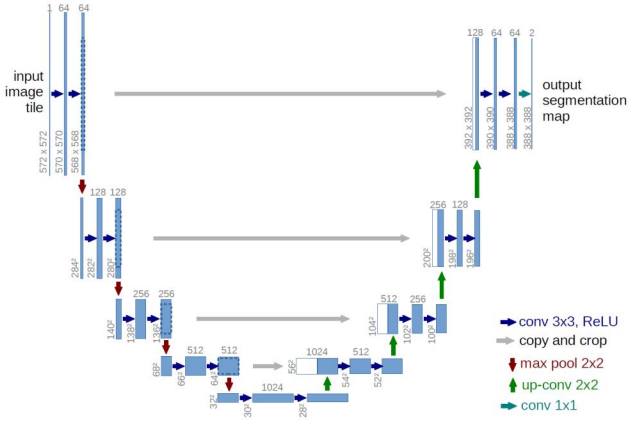


Figure 2: UNet Architecture [5]

U-Net (see Figure 2) is a fully convolutional architecture that performs semantic segmentation. It has been shown to perform quite well on many similar medical imaging tasks, which motivated us to apply it to our dataset as a baseline [5]. We used the Fully Connected UNET architecture to classify each pixel in the image as Background (0) or Nucleus (1). We reshaped the input image to 256 by 256 shape. We used binary cross-entropy loss over individual pixels. We also measured the results using the Intersection over Union metric.

3.3.2 Mask RCNN

Mask R-CNN is an extension of the Faster RCNN model [2]. Faster R-CNN is a Region Proposal network (RPN) based model which has two outputs for each candidate object: a class label and a bounding-box for the object. Mask R-CNN extends the model by adding in a third branch which outputs an object mask in addition to the other two outputs. This mask lets us do more accurate object segmentation apart from object detection and object localization done by a normal Faster RCNN network.

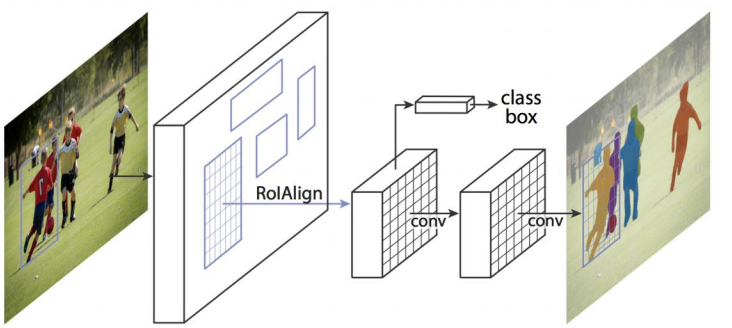


Figure 3: MaskRCNN Architecture [3]

We ran four different Mask R-CNN experiments: one for each combination of dataset (raw or augmented) and transfer learning technique (fine-tuning or no fine-tuning). See Section 3.4 for details. As fine-tuning on the raw dataset had an immediate and significant negative impact on the performance of the model, we do not report these results.

3.4 Transfer Learning

As is typical for many medical learning problems, our datasets were quite small. Even our augmented dataset with over 17,000 images is tiny by computer vision standards. This meant that transfer learning was a natural step in order to reduce over-fitting. For our Mask R-CNN models, we used weights pre-trained from the MS COCO dataset. In addition to reducing over-fitting, transfer learning helped us train our models faster, which was valuable helping us conserve our computational resources. As UNET was our baseline and was a much smaller and simpler network, we did not experiment with transfer learning for this model.

We applied two different types of transfer learning to our Mask R-CNN models. First we froze the RPN and trained only the heads of the network. Next for select experiments we fine-tuned all layers of the network on our train data. Although our datasets were both small, we still saw a positive effect from the fine-tuning (see Section 4).

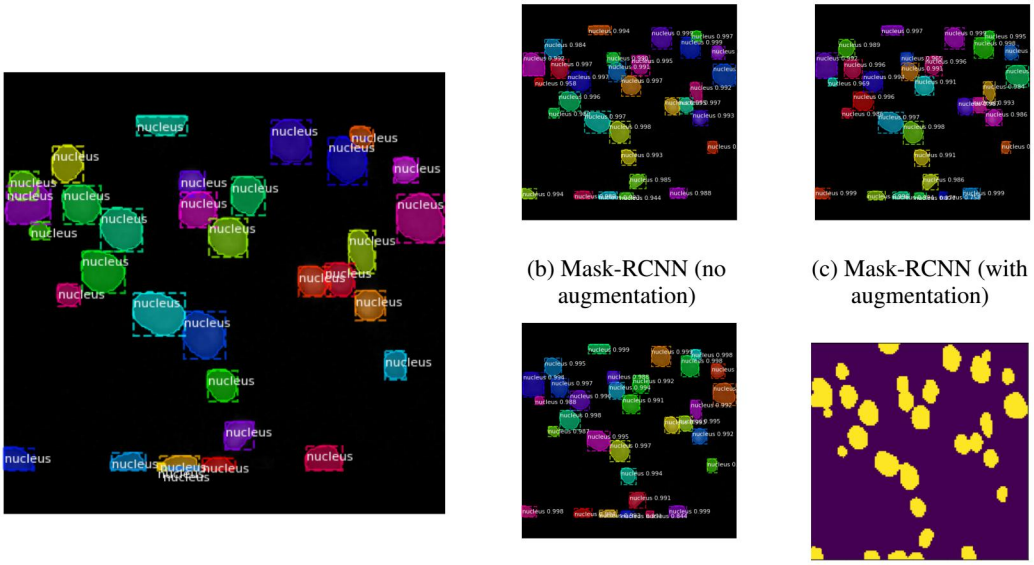
4 Results

Table 1: Results

Model	**AP at 0.5**	**mAP**
U-NET	0.361	0.246
Mask R-CNN (raw data)	0.814	0.558
Mask R-CNN (augmented data)	0.841	0.584
Mask R-CNN (augmented data and fine-tuning)	0.851	0.600
Current competition leaders	NA	0.569[1]

The results are shown in table 1. Our best performing model was the Mask R-CNN trained on the augmented dataset, first freezing the RPN and the fine-tuning the entire network. All of the Mask R-CNN models significantly outperformed the baseline UNET models. Additionally, both Mask R-CNN models trained on the augmented dataset outperformed the current competition leader. This statistic might be slightly misleading as the metrics for the competition leader were calculated on a hidden test set that might have a different distribution from our test set. However, even if this is the case, our models are still competitive with the current best.

[1]This mAP value was computed for a hidden test set that was different from the test set that we evaluated on. This might affect the relative performance of the two algorithms.



(a) Ground Truth

(b) Mask-RCNN (no augmentation)

(c) Mask-RCNN (with augmentation)

(d) Mask-RCNN (with augmentation and fine-tuning)

(e) U-NET (no augmentation)

Figure 4: Visualizations

Figure 4 shows an example of the output given by each models. All of our models qualitatively did a good job at detecting the nuclei. Errors seem to appear primarily for overlapping and especially small nuclei. The big jump in mAP from our U-NET baseline model to the Mask R-CNN approach was because U-NET does semantic level segmentation on a pixel by pixel level. It is very good at detecting whether a pixel belongs to a nucleus or not. However, it cannot distinguish between two overlapping nuclei. Therefore, the whole blob gets classified as one object, which leads to one True Positive and one False negative. Mask R-CNN overcomes this limitation by doing object instance segmentation where it learns that there are two separate overlapping objects and then creates individual segmentation masks for each of them. This allows the network to assign the same pixel to multiple objects as is needed in this problem. Hence, the mAP values for the Mask R-CNN model were much higher.

Data Augmentation helped increase the mean Average Precision by 0.026 points. Flipping the images vertically and horizontally meant that the model has more data to play around with, which resulted in in generalizing better to the validation and test set. Another significant boost to the model performance came from fine-tuning the entire network at the end of training. This indicates that the regions of interest trained for detection of every-day objects are not the same as regions of interest for nuclei detection. This makes intuitive sense as the microscopy images look very qualitatively different from images in the MS COCO dataset. This mismatch highlights the help that more data would bring, as the full network including the RPN is quite large and complex, meaning that you would need more data to train it well without over-fitting.

5 Conclusion and Future Work

Overall, we were able to demonstrate the effectiveness and potential of Mask R-CNN on medical image segmentation tasks. There are several opportunities for future work based off of this project. Although we did not have the computational resources to carry this out, it seems promising to modify the model to take only greyscale images of a set small size before initially training on MS COCO. This would reduce the size of the network without loosing much information from the images, thus reducing overfitting. Different augmentation techniques such as jittering and distortions also show potential and could increase the effective dataset size to help alleviate the problem of insufficient data. We believe that the techniques discussed in this paper have significant promise for the future course of the nuclei segmentation problem.

6 Contributions

Overall, we all had equal contributions. Aditya did most of the Amazon, Library and other infrastructure setups. He wrote about the data augmentation as well as the data restructuring. Suvrat ran the preliminary UNet case. Once we all together got the first Mask R-CNN running, Will re-structured it and Suvrat and Will ran and monitored the final results. We all contributed equally in all the writeups.

7 Link to Code

https://github.com/suvrat96/CS230Repo.git

References

[1] 2018 Data Science Bowl | Kaggle, www.kaggle.com/c/data-science-bowl-2018/data.

[2] Ren, Shaoqing, et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” [1506.01497] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 6 Jan. 2016, arxiv.org/abs/1506.01497.

[3] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988. doi: 10.1109/ICCV.2017.322

[4] Li, Yi, et al. “Fully Convolutional Instance-Aware Semantic Segmentation.” [1611.07709] Fully Convolutional Instance-Aware Semantic Segmentation, 10 Apr. 2017, arxiv.org/abs/1611.07709.

[5] Ronneberger, Olaf, et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” [1505.04597] U-Net: Convolutional Networks for Biomedical Image Segmentation, 18 May 2015, arxiv.org/abs/1505.04597.

[6] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In NIPS, pages 2852–2860, 2012.

[7] K. Sirinukunwattana, S. E. A. Raza, Y. W. Tsang, D. R. J. Snead, I. A. Cree and N. M. Rajpoot, "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images," in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1196-1206, May 2016.

[8] Gross, Polina, et al. “Nuquantus: Machine learning software for the characterization and quantification of cell nuclei in complex immunofluorescent tissue images.” Nature News, Nature Publishing Group, 23 Mar. 2016, www.nature.com/articles/srep23431.