Automated Nuclei Detection

Category: Computer Vision (Image Segmentation)

William Marshall, Suvrat Bhooshan, Aditya Garg

Introduction

In medical research, finding the nuclei of cells is often the first step in order to assess their health or the affect a drug is having on them. Identifying nuclei allows researchers to identify each individual cell in a sample, and by measuring how cells react to various treatments, the researcher can understand the underlying biological processes at work [1]. However, 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].

Goal: Modify state of the art Deep Learning Segmentation Models to detect nuclei in microscopy images.

Dataset

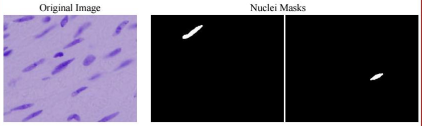
Source: Kaggle - 2018 Data Science Bowl [1]

Input: Each sample contains the raw image containing multiple nuclei. It also has masks of individual nuclei in order to train a multi-object segmentation model.

Dataset Statistics

- ~25,000 individual Nuclei
- 670 unique examples/images
- Train:Test:Validation = 80% : 10% : 10%
- 5 types of images

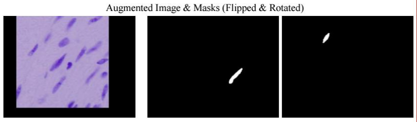
Original Image Nuclei Masks



Dataset Augmentation: Since we did not have enough data, we applied the following data augmentation techniques increasing the training dataset by over 30 times:

- Vertifical Flips
- Horizontal Flips
- Image Rotation (90°, 180°, 270°)

Augmented Image & Masks (Flipped & Rotated)

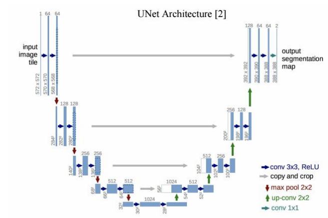


*Images not shown to scale

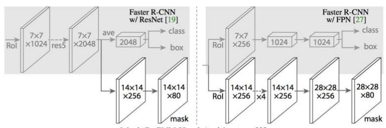
Algorithms & Approach

We tried 2 popular image segmentation models:

- **UNet Architecture** [2]



- **Facebook Mask R-CNN (Detectron)** [3]



Mask R-CNN Head Architecture [3]

- **Transfer Learning**
  - Even post augmentation, we had 17,000 valid images with only 536 raw images.
  - We used weights pre-trained on the MS COCO dataset to provide some pre-learnt knowledge of any generic image
  - Freezing the RPN portion of the network and only training the heads that performed the actual segmentation predictions, allowed us to reduce the effects of overfitting.
  - As an additional benefit, the pre-trained weights allowed us to train our networks faster, which was particularly useful since we were short on computational resources.

- **Evaluation Metrics**
  - Mean Average Precision (mAP)

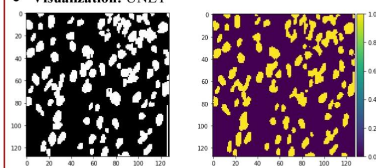
$$\frac{1}{|thresholds|}\sum_t \frac{TP(t)}{TP(t)+FP(t)+FN(t)}.$$

Results

- **Numerical Results**

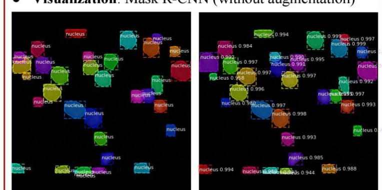
Model	Precision (with threshold 0.5)	Mean Average Precision (mAP)
UNet	0.361	0.246
Mask R-CNN (without data augmentation)	0.814	0.558
Mask R-CNN (with data augmentation)	0.841	0.584

- **Visualization:** UNET



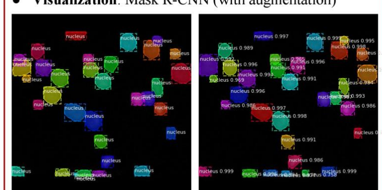
Ground Truth Prediction

- **Visualization**: Mask R-CNN (without augmentation)



Ground Truth Prediction

- **Visualization**: Mask R-CNN (with augmentation)



Ground Truth Prediction

Discussion

- We initially tried a semantic segmentation approach thorough training a UNET architecture.
- Semantic level segmentation classifies objects on a pixel level.
- This lead to worse results as overlapping objects were not distinguished. Average Precision was affected.
- Object level segmentation through Mask R-CNN was the next approach.
- It performed very well, as we could segment even overlapping objects, and assign same pixel to multiple object even if they were occluded.
- Highest mAP on the hidden data set on the kaggle leaderboard is 0.57. On our validation set the number is in the same ballpark. It will be interesting to see how the model generalizes on their hidden data.

Future Work

- Advanced Transfer Learning by freezing initial layers and training later ones
- Improve Network to handle images of different sizes
- Analyze the precision over different image types to see if one type has better prediction than others
- Try more sophisticated data augmentation techniques like jittering & distortion

References

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

[2] 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.

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

[4] https://github.com/matterport/Mask_RCNN