Spot Nuclei: Segmentation with U-Net

CS230 Final Report

Ta Tang SUNet ID: tatang

1 Introduction

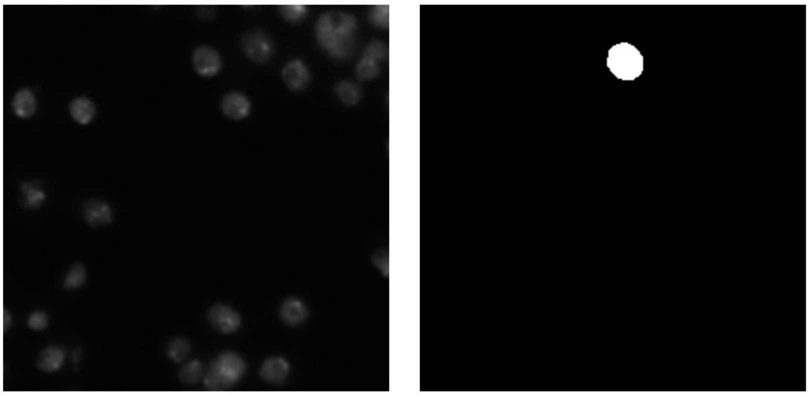
This project is a competition on Kaggle (https://www.kaggle.com/c/data-science-bowl-2018). The goal is to find nuclei from images of cells. Because finding nuclei is usually the first step to analyze images of cells under various experimental conditions, and this could be time-consuming without the help of an accurate algorithm. The result of this project could potentially accelerate results analysis of biological experiments and thus speed up research progress and new drugs development.
The challenge of this project would be the capability of the model to generalize. Since there are various cells of human body, the same type of cell would manifest various forms under different conditions, images are taken under various conditions, etc., learning from training set and expecting your model to handle unseen types of cells images would be a very challenging task. This is actually an instance segmentation task. I treat it as a segmentation task first, then I tried a method to do the instance segmentation.

2 Dataset

The dataset is provided by the Kaggle competition organizer. There are two stages of datasets. First stage dataset is used to train your model.The first stage dataset contains a training set and a dev set. The training set contains 670 images and each image is associated with several tens of masks (each mask corresponds to a nuclei in the image). The test set contains 65 images without masks. The second stage of dataset contains a test set used to evaluate the performance of the model. The second stage test dataset will have unseen types of cells images, therefore it will have different data distribution compared to first stage data.
Figure (a), (b) is one example of the data (I only show one of its masks here).

3 Metrics

This competition is evaluated on the mean precision at different intersection over union (IoU) thresholds. Detailed description is here(https://www.kaggle.com/c/data-science-bowl-2018#evaluation). For just segmentation, I use accuracy as the metric while training the model.



(a) Image (b) A mask of (a)

4 Approach

4.1 Segmentation with U-net

The original task is an instance segmentation task. We not only have to tell wether a pixel belongs to a nuclei, but also need to know which nuclei it belongs to. To do instance segmentation, a good way to go is Mask R-CNN[1, p. 1]. However, to begin with, I choose to use a much simpler U-net[2, p. 2] architecture to just do segmentation. U-net is known to be good at segmentation of biological cell images. I use an architecture very similar to Figure 2.

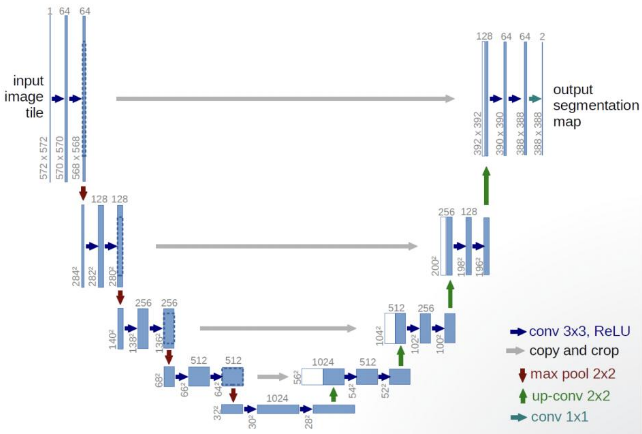


Figure 2: U-net Architecture

The input is reshaped(128*128*3) and normalized image. The output is a 128*128*1 segmentation map. The ground truth label is just adding all the corresponding masks. The loss is binary crossentropy. I train the model with Adam Optimizer and early stopping. This model achieves 97.36% accuracy on training set, and 97.22% accuracy on dev set. Figure 3 is an example image in the training set. Figure 4 is the ground truth segmentation and the prediction made by the model.

However. This competition is actually an instance segmentation task. This requires that you have to distinguish between individual cells. In these images, there are usually some cells attached together. Although U-net achieves a high accuracy in segmentation, when measured by mIoU, this model achieves a 0.291 score. This is not bad but also not that good.

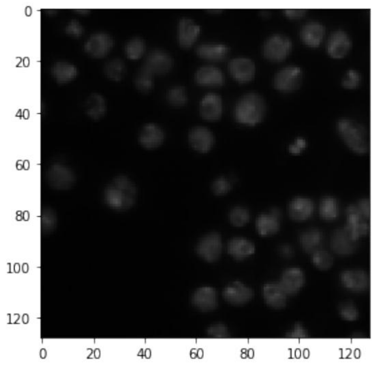


Figure 3: Image in the training set

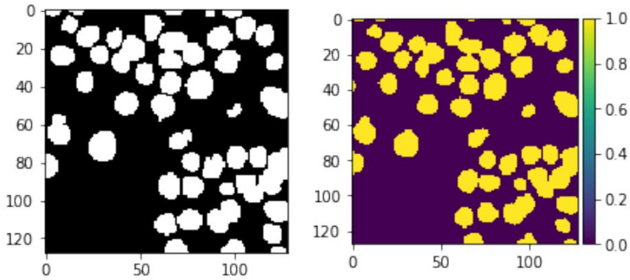


Figure 4: Left: ground truth nuclei pixels. Right: prediction of the model

4.2 Boundary Detection

We can consider this instance segmentation task as a simpler segmentation task, if we classify pixels into 3 categories. First category is background, the second is nuclei, and the third is pixel at the intersections of different nuclei.

Basically, I propose to add an output branch at the end of a U-Net. So there are two outputs. One of them will output pixels which belong to nuclei. The other one will output pixels which belong to the boundaries of different nuclei. Combine the two outputs to distinguish different nuclei.

However, pixels belonging to boundaries will be much less than pixels belonging to nuclei and background. Thus it will be much difficult for the network to learn pixels belonging to boundaries. I add 2 conv layers and 1 output layers to form the boundary pixels output branch. I processed the dataset to get the ground truth boundary pixels. Then I feed the musk of nuclei and musk of boundaries respectively to the two output layer as the ground truth label. I use weighted cross-entropy loss. (I assign more weights to the loss of pixels at the boundary of nuclei). Then I separate the 670 training images into training/dev set (0.9/0.1). I use mini-batch size of 16, learning rate 0.001, learning rate decay 0.001 and early stopping by monitoring the loss of the dev set. Only save the model which gives the minimum of dev set loss. I use accuracy to evaluate my model performance. After 39 epochs, on training set, the model achieves 96.13% accuracy on output2(boundary pixels). On dev set, the accuracy is 95.99%. Here is an example of the results:

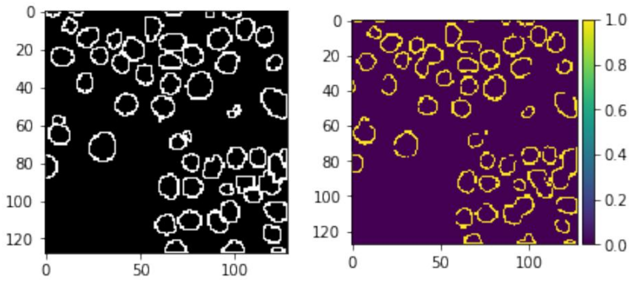


Figure 5: Left: ground truth boundary pixels. Right: prediction of the model

5 Summary

I tried U-net to do segmentation of nuclei pixels, which results in a very good performance(97.22% accuracy on dev set). I also tried to detect the boundaries of nuclei and hope by combining the two outputs we can distinguish between different nuclei. However, boundary is harder to detect. Although eventually I can achieve high accuracy on detecting boundaries, the performance of the model to detect boundaries at the intersection of closely attached nuclei is very poor. (Because the pixels at the intersection of different nuclei only take up a very small fractions of the total pixels, it is too difficult for the model to learn to detect them.) In the future, I would like to try Mask R-CNN to do this instance segmentation task.

6 Code Link

https://github.com/taaatang/CS230_Spot_Nuclei

References

[1] Kaiming He et al. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.