Nucleus Detection using Deep Learning

James Pao (jpao), Benjamin Knapp (bknapp8), Burak Bartan (bbartan)
CS230: Deep Learning, Winter 2018, Stanford University

Introduction

According to the World Health Organization, cancer is the second-leading cause of mortality worldwide, being responsible for 1 in 6 deaths [1]. Considering this, the ability to automatically and accurately detect cell nuclei in images is highly desired, as doing so would allow researchers to study the effects of treatments and drugs at a cellular level without expending too much manpower or time manually searching for cells. This would allow scientists the ability to iterate more rapidly and focus more of their effort on developing treatments and cures. As much promise has been shown by a variety of CNN architectures [2][3] for semantic segmentation tasks, we investigate using deep learning for accurate nuclei detection.

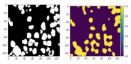
Methods

Loss: binary cross-entropy loss
Metric: Sørensen-Dice coefficient → $\frac{2(Y_{true} \bigcap Y_{pred})}{Y_{true} + Y_{pred}}$

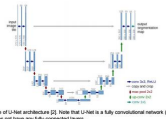
Summary: Implement simple CNN as baseline. Since we need to localize, try semantic seg CNN's such as U-Net and tune hyperparameters. Investigate transfer learning using available ResNet50 weights. Look into techniques such as data augmentation to allow our model to generalize better.

Baseline: Simple CNN

Architecture:
(Conv2 -> MaxPool) x 3 -> Conv2 -> (Upsamp -> Conv2) x 3 -> sigmoid



UNet

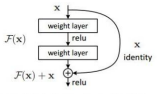


Example of U-Net architecture [2]. Note that U-Net is a fully convolutional network (FCN) as it does not have any fully-connected layers.

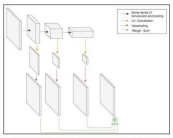
- Basic CNN's use max-pooling to reduce the spatial size of feature maps, limiting the number of parameters and reducing overfitting
- However, downsampling inhibits localization ability: okay for context-primary tasks (i.e. does this picture contain some object?) but poorer on semantic segmentation (i.e. what class does this pixel belong to?)
- For nucleus detection, important to have clear and accurate boundaries, necessitating retention of pixel-level information through the network
- U-Net: for encoding step, double-convolve and keep feature map to use when decoding, then downsample.
- U-Net: for decoding step, upsample, concatenate saved feature maps, then double convolve. At the end, convolve using sigmoid $g(z)$ for each pixel. **Feature maps from encoding step helps localization during decoding.**

Transfer Learning w/ ResNet50

- ResNet50 consists of residual blocks that have skip connections, which help with vanishing/exploding gradients
- Makes it possible to train deeper networks more easily
- Our model is a ResNet50-based FCN with connections from the last 32×32, 16×16, and 8×8 layers of the ResNet50 [4]
- We initialized the weights using ResNet50 weights pre-trained on ImageNet (transfer learning)
- Even though ResNet50 weights were optimized for a different task on ImageNet, transfer learning still helps. It leads to a faster convergence.



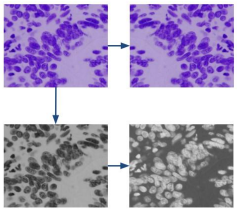
Residual Block (building block of ResNet)



Fully convolutional neural network architecture

Data Augmentation

- Original training set yields poor performance on RGB images with dark nuclei
- Convert entire training set to grayscale
- Invert images and add to original training set
- Translate/rotate images
- Mirror image about horizontal/vertical axes



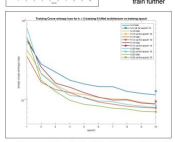
Dataset

The dataset, provided via a Kaggle competition sponsored by Booz Allen Hamilton [5]], is comprised of divergent images of human cells, which likely have been stained for nuclei detection under fluorescence microscopy. Imaging conditions vary by modality, camera, and magnification (to motivate generalizability of the algorithm). There are **670 images**, along with their human-labeled nuclei masks. Given the relatively small training dataset, we chose to split our training dataset into **90%-train/10%-validation** (i.e. 603/67).

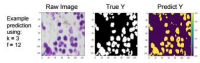
Results

UNet

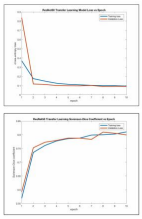
- Investigated various k-values (2, 3, 4) for f = 12 over 10 epochs
- Chose best k-value to investigate number of filters (f)
- Chose best k-value (3) and f-value (12) architecture to train further



Best model: k = 3, f = 12, trained for 25 epochs
Train dice: **0.8707**
Val dice: **0.8732**

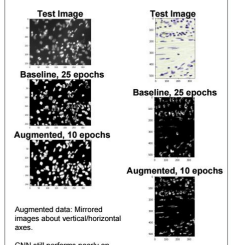


ResNet



**all trained 10 epochs, meant for comparison*	*Basic CNN*	*U-Net*	*ResNet (transfer)*
Train Dice	0.7328	**0.8499**	0.8104
Val Dice	0.7080	**0.8488**	0.7996

Data Augmentation



Augmented data: Mirrored images about vertical/horizontal axes.

CNN still performs poorly on stained tissue images, but increases performance on simpler data

Discussion and Future Work

Our best model, a U-Net architecture, performs reasonably well on our training set, but generally fails to identify crowded nuclei in tissue section images with general RGB components, even with our initial data augmentation attempts. Future work could include generating GANs for producing new images for this purpose, as well as refining our data augmentation approach.

We also find that transfer learning applied with ResNet 50 gives us a performance in between a basic CNN and U-Net. We believe this is possibly due to the pre-trained ResNet weights not translating well to our segmentation task, as well as the ResNet architecture being potentially situationally suboptimal. We will continue to tune the U-Net model and train it for longer to see if performance improvements can be made. We will also implement SegNet, which uses max pooling indices instead of feature maps as in U-Net to compare performance and network training speed, as it is also an architecture that seeks to retain localization ability, which we have seen is clearly necessary with our images.

References

[1] http://www.who.int/mediacentre/factsheets/fs297/en/
[2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015.
[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," Oct. 2016.
[4] https://www.azavea.com/blog/2017/05/30/deep-learning-on-aerial-imagery/
[5] https://www.kaggle.com/c/data-science-bowl-2018#description