Dara Rouholiman

Stanford, CA
dara.rouholiman@gmail.com

March 30, 2018

Udacity Machine Learning Nanodegree
Capstone Project Report
Find Nuclei in pathological images using CNNs

i. Definition

Project Overview

Accurate segmentation of nuclei has diagnostic significance in histopathological examinations.[1] At the moment, manual pathological examinations by humans serve as the gold standard for reliable morphological statistics in many clinical protocols. However, the manual segmentation and detection of nuclei structures is expensive, error-prone, and time-consuming. Therefore, automating nucleus detection and segmentation will improve the efficiency, reliability, and scalability of clinical analysis in different domains.

Existing work in the field can be classified into two categories: image processing methods and machine learning techniques.[2]–[5] Recent approaches in machine learning have focused on deep neural models. [6], [7] The peculiar fact that finding nuclei in pathological images has a high impact on healthcare and is also an opportunity to learn more about image segmentations forms the basis of my attraction to this study.

The dataset used for this project contains 670 training images and 65 testing images of microscopic images of varying sizes showing ensembles of cells and their nuclei. For the training images the nuclei are segmented by humans such that we know their number and location within each image. The dataset is provided by Booz Allen Hamilton's *2018 Data Science Bowl: Find the nuclei in divergent images to advance medical discovery* challenge under Creative Commons license.[8] The images were acquired under a variety of conditions and vary in the cell type, magnification, and imaging modality (brightfield vs. fluorescence) to challenge an algorithm's ability to generalize across these variations.

Problem Statement

Development of an algorithm for semantic segmentation of nuclei in pathological images rather than instance segmentation which will be discussed as a future study later on this paper. In this study, An U-Net neural network consisting of several convolutional and max-pooling layers is developed and used for semantic segmentation of nuclei in the images. The algorithm uses TensorFlow as its backend and Keras as the framework. The input of the network are images of quantifiable shape (height, width, channels) while the output are corresponding binary masks of shape (height, width, 1). The output result will be a list of indices for the segmentation, pairs of values that contain a start position and a run length. E.g. '1 3' implies starting at pixel 1 and running a total of 3 pixels (1,2,3). Thus, an outline of tasks are as follows:

1. Data exploration
2. Data pre-processing
3. CNN benchmark model design and training
4. U-net model design and training

5. Accuracy and loss graphs
6. Final encoded results
7. Future work discussion

Metrics

The final solution will be evaluated on the mean average precision (mAP) at different intersection over union (IoU) thresholds. mAP is the de facto standard for the major object detection and image segmentation competitions in computer vision. The IoU of a proposed set of object pixels and a set of true object pixels is calculated as:

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

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. In other words, 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:

$$\frac{\text{TP}(t)}{\text{TP}(t) + \text{FP}(t) + \text{FN}(t)}$$

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:

$$\frac{1}{|\text{thresholds}|} \sum \frac{\text{TP}(t)}{\text{TP}(t) + \text{FP}(t) + \text{FN}(t)}$$

ii. Analysis
iii.

Data Exploration

The dataset for this project is provided by Booz Allen Hamilton's *2018 Data Science Bowl: Find the nuclei in divergent images to advance medical discovery* challenge and contains 670 training images and 65 testing images of microscopic images of varying size showing ensembles of cells and their nuclei. For the training images the nuclei are segmented by humans such that we know their number and location within each image. The images were acquired under a variety of conditions and vary in the cell type, magnification, and imaging modality (brightfield vs. fluorescence). Each image is represented by an associated Image Id. Files belonging to an image are contained in a folder with this Image Id. Within this folder are two subfolders of images and masks, contains the segmented masks of each nucleus. Each mask contains one nucleus. Masks are not allowed to overlap (no pixel belongs to two masks). Data exploration has revealed that there were a few outliner masks present that only represent a single pixel line and not a nucleus. A function will be written during the data pre-processing step to find all false masks and exclude such masks from the training.

Exploratory Visualization

The visual representation of two samples are presented in Figure 1. The figure represents two training images on the top and their corresponding combined masks annotated by humans right below them. This two samples were picked to help the reader understand the challenge of distinguishing the overlapping nuclei, how the images are from different conditions, and moreover how a nucleus can be in different "oval" shape.

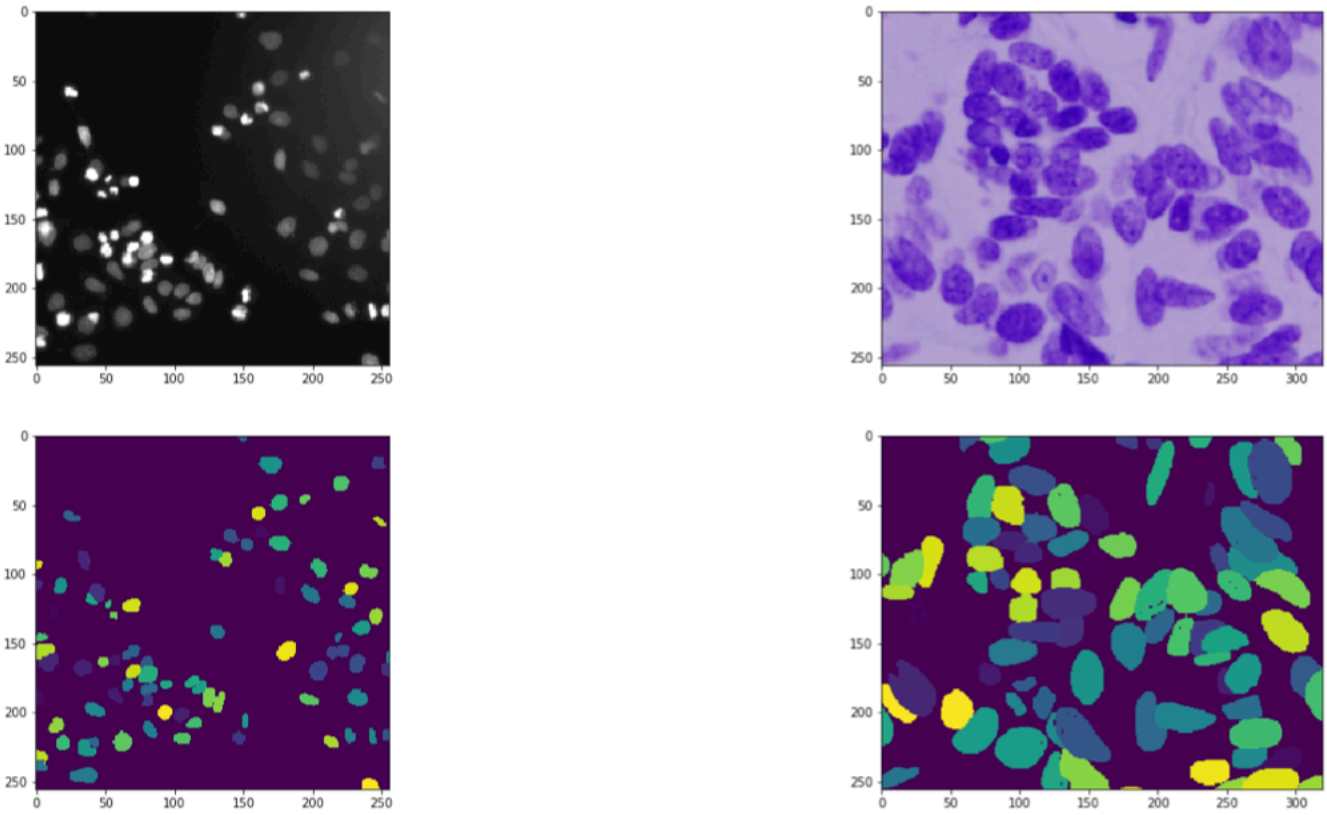

Figure1: Sample of training images and masks

Algorithms and Techniques

The U-net network architecture is illustrated in Figure 2. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by an exponential linear unit (ELU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by an ELU. At the final layer a 1x1 convolution is used to map each 16-component feature vector to the one class. In total the network has 23 convolutional layers, uses TensorFlow as its backend, and Keras as the framework. The input of the network is images of quantifiable shape (height=256, width=256, channels=3) while the output is corresponding binary masks of shape (height=256, width=256, classes=1).

One important modification in U-net architectures is that in the upsampling part features a large number of feature channels and allows the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting path and yields a u-shaped architecture. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image.

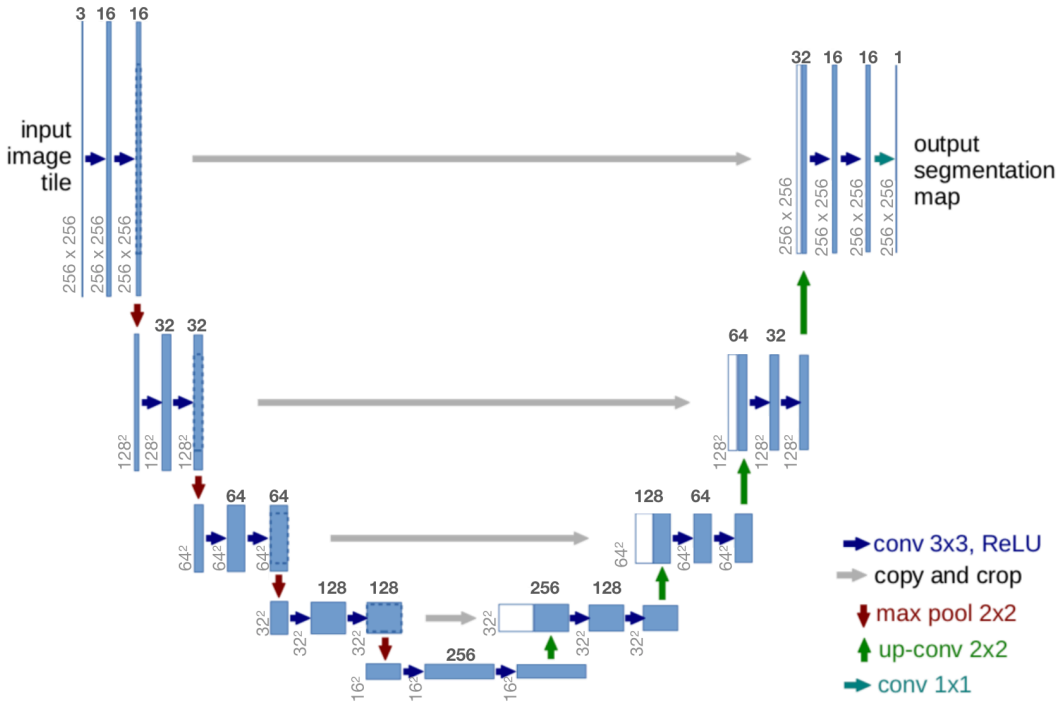

Figure 2: U-net architecture with blue boxes correspond to a multi-channel feature map. The number of channels is denoted on top of the box. The image size (width, height) is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Benchmark

To measure the performance of our U-net model and to provide an appropriate benchmark model, a simple CNN model is trained and tested with the same dataset and pre-processing steps to get a comparable baseline score. The CNN has 8 CNV layers followed by 3 PL layers which is followed by 4 upsampling layers as illustrated in Figure 3. The input of the CNN is images of quantifiable shape (height=256, width=256, channels=3) while the output is corresponding binary masks of shape (height=256, width=256, classes=1). By using the same dataset and having the same input and output shape, the CNN will be used as the benchmark to measure the U-net's performance on its loss, accuracy, and precision (IuO).

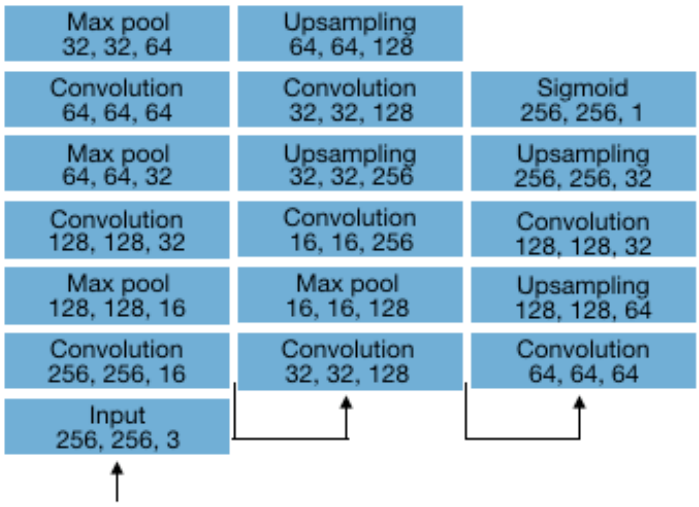

Figure 3: The benchmark CNN architecture

iii. Methodology

Data Preprocessing

The data preprocessing step has three sections: data normalization, data cleanup, and data augmentation. In data normalization, two functions were written to handle reading and resizing the training and testing images to 256 x 256 pixels. The normalized image size enables consistent input image feeding to the model and help the network learn the features at each level with fixed padding and sizes, otherwise multiple models need to be trained for different image sizes.

In data cleanup, we address the outliners and abnormalities we found in data exploration. During data exploration, some edge masks were presented to be just line as common marking errors during image annotations, thus 1px or empty edge masks were found and excluded from the training set.

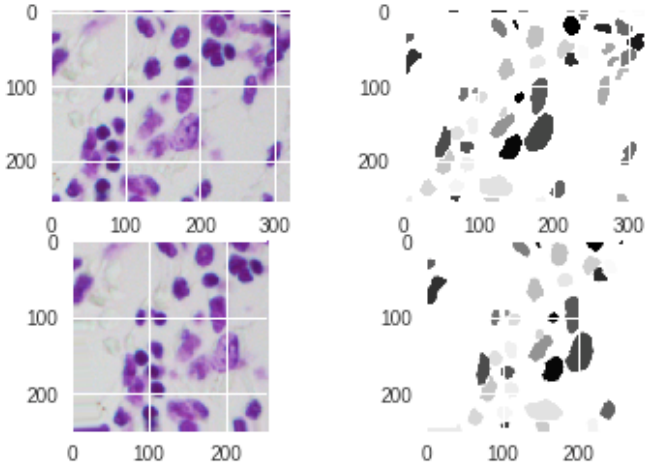

Figure 4: Sample data augmentation

Moreover, there is very little training data available, thus data augmentation was used by applying affine transformations to the available training images and masks. This is particularly important in biomedical

segmentation since deformation is the most common variation in tissue, and realistic deformations can be simulated efficiently. Figure 4 displays the original image and its mask on the top and the augmentated image and its mask at the bottom. The following steps were taken to implement this affine transformation:

1. random 0.9 scaled down version of the annotation and image
2. random 30° rotations
3. random crop at the 'edge'

The dataset contains colored images from different labs with various tools and techniques and we can't classify them by the original lab or technique. One might choose to normalize all the images and masks to one grayscale channel, however since the dataset is not classified by the lab or the technique that was used, I chose to increase the generalizability of the algorithm and keep the three RGB channels as the input parameter of the model.

Implementation

All experiments were performed using Keras with TensorFlow backend in Python 3.5 on Google Colaboratory equipped with a single Tesla K80. The backend was used for automatic differentiation and optimization during training. We used zero-padding in convolutional layers in all architectures. Therefore, output channels have the same dimensions as the input. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image.

Refinement

In the refinement process different number of epoch, validation/test splits, and additional CNV layers were trained and tested and it was found that the best model is the original 23 CNV layer U-net with 0.1 validation/test split and 250 epoch with 0.328 IoU score. The original U-net was trained at epoch 50 with three different validation/test splits: 0.05, 0.1, and 0.2; it was found that the 0.1 split has a better result. Table 1 summarizes these three experiments and their IoU scores.

Table 1: different validation/test splits and their IoU scores of U-net model at epoch 50

Validation/test split	*IoU Score*
0.05	0.304
0.1	0.315
0.2	0.295

The 23 CNV was compared to the 8 CNV layered CNN and 27 CNV layered U-net models at different epoch numbers with 0.1 validation/test split. Figure 5 shows these different experimental approaches and their IoU precision score on the unseen test data.

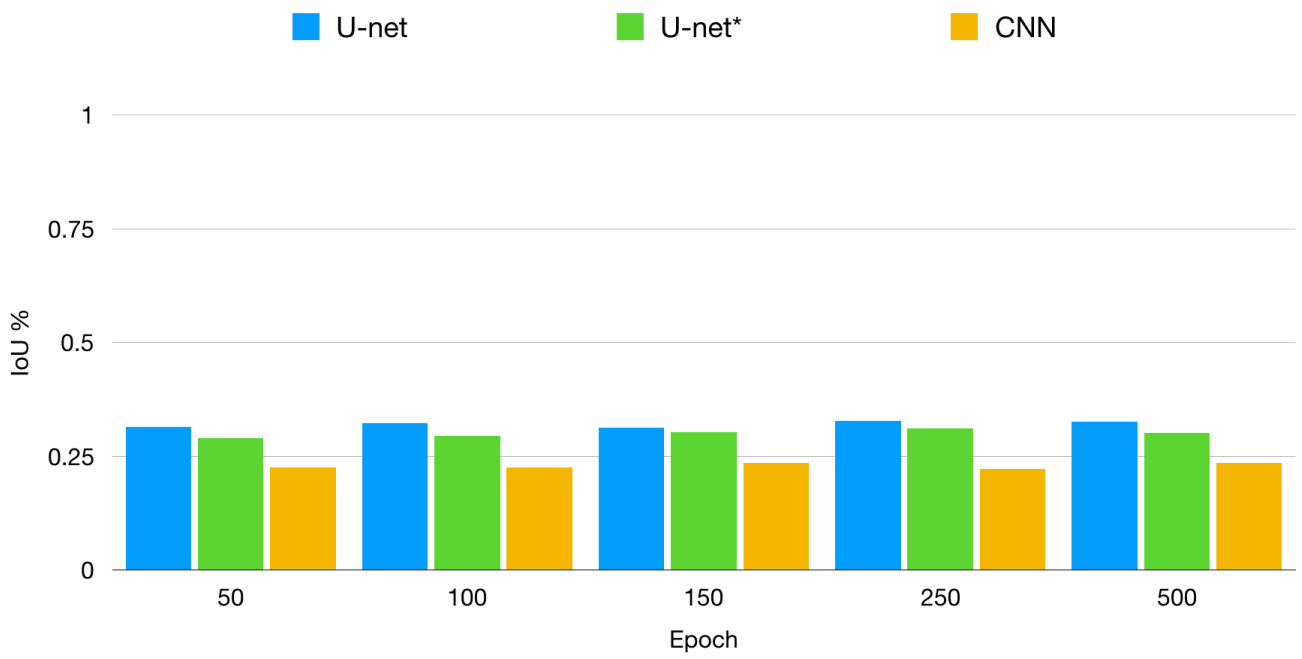

Figure 5: Results of models at different epoch numbers. The 23 CNV layered U-net outperforms the 27 CNV layered U-net and our benchmark 8 CNV layered CNN models.

iv. Results

Model Evaluation and Validation

The final model, 23 CNV layered U-net at 250 epoch with 0.1 validation/test split meets the solution expectations with appropriate final parameters. As shown in Figure 6 the model's accuracy and loss graphs at epoch 250 the training set meet the accuracy of 0.95 and the test set meet the accuracy of 0.9. Moreover the model has the loss of 0.05 on the training set and 0.15 on the training set.

The various sizes, colors and shapes of the 669 training images and the fact that the final 65 testing images have not been used during training at all and their annotated masks are not available yet, make our evaluation process a wise choice that evaluates generalizability of our solution.

Perhaps the ultimate weakness of the proposed U-net model is its robustness on finding the nuclei on very different images. The size, shape, condition/colors of each image differs greatly and our simple preprocessing step and the data augmentation, both made an improvement of 2% to the final IoU score. However, although the accuracy and loss of the model is adequate, the IoU results are not, thus the model should be improved further and/or complemented with another machine learning model(s).

In summary, the results found from the model can be trusted and used to ensemble a final model with a better IoU score. To complement the proposed U-net model, I am going to consider alternatives to

convolutional networks and use models better suited for instance segmentation rather than semantic segmentation.

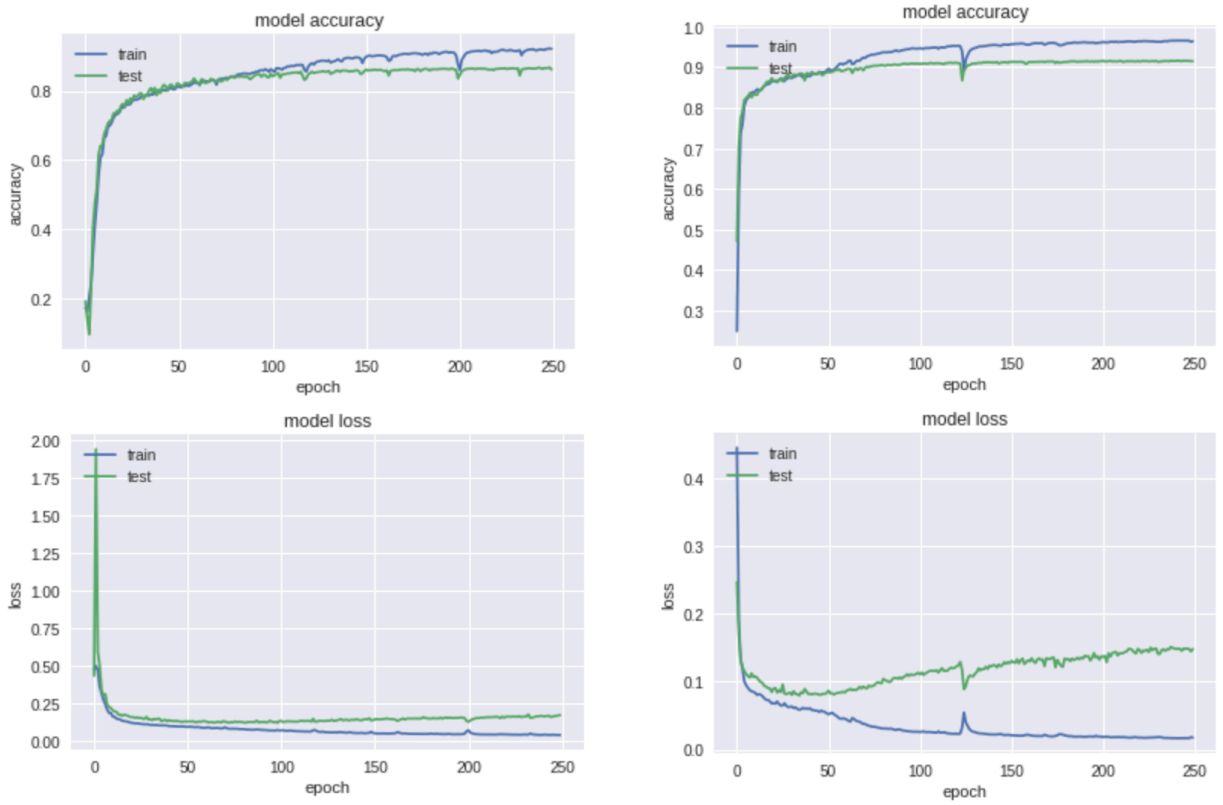

Figure 6: The U-net model's accuracy and loss graph on the right vs. the benchmark CNN model's accuracy and loss graph on the left.

Justification

The proposed U-net model has a higher accuracy of 9% than the benchmark CNN model with accuracy values of 0.95 and 0.9 (on training and testing set respectively) vs the benchmark CNN model with accuracy scores of 0.86 and 0.81. Moreover the U-net model has a lower loss of 5% than the benchmark CNN model with loss of 0.05-0.15 on training and testing set respectively, compared to the benchmark model with loss of 0.1-0.2. Finally the U-net model has a IoU score of 0.328 which is 10% higher than the IoU score of the benchmark CNN model with IoU score of 0.220.

The Proposed U-net model is tested with 65 unseen and unmasked test images, it has predicted the nuclei in each image and has achieved the IoU score of 0.328. The results of the model suggest that the U-net is superior to the benchmark CNN model however it does have the overfitting issue which further suggests the model needs to be complemented.

The proposed U-net model alone is not accurate and robust enough for finding the nuclei in real-world images due to the fact that real-world images get prepared under various conditions. The best model that can solve this general problem should be an ensemble of a few different models.

Free-form Visualization

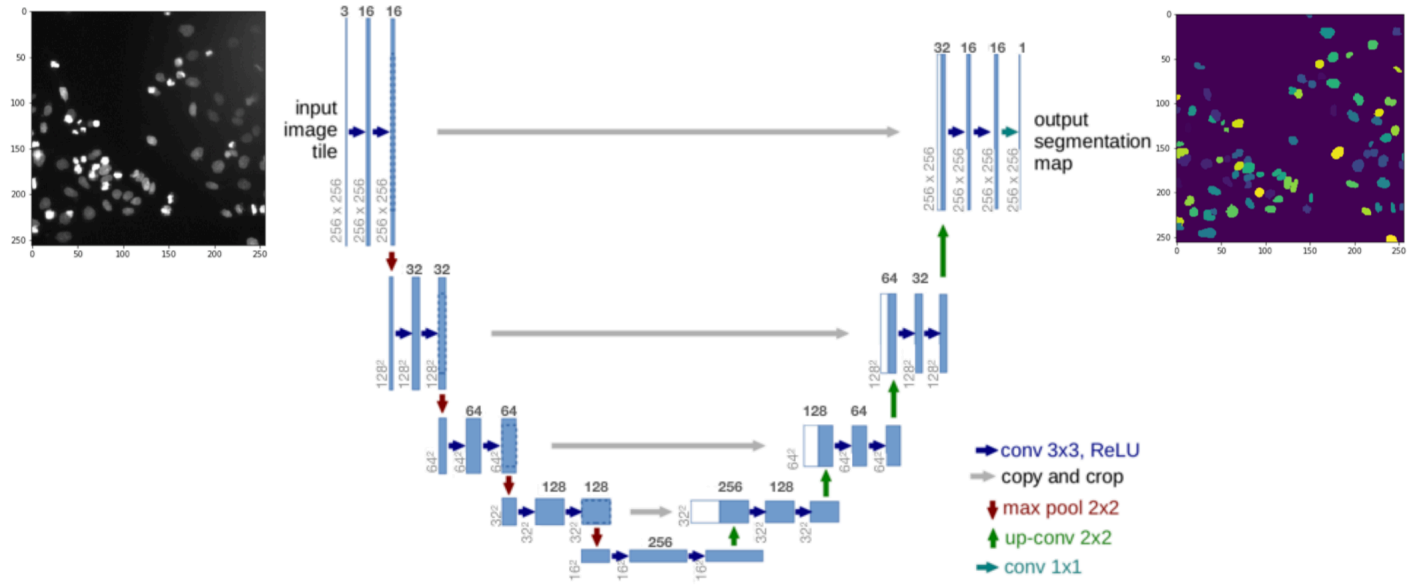

Figure 7: The 23 CNV layered U-net model and its function in finding nuclei in pathological images

The most important quality of this project is understanding the challenges of image segmentation in pathology images with respect to the strengths and weaknesses of U-net learning model. Thus, Figure 7 illustrates both the strengths and weaknesses of the proposed U-net model while highlighting the challenge of working with pathological images. The unique contraction and expansion architecture of the model is definitely its strength, however like all convolution nets, the weakness is in generalizing to new viewpoints. The U-net has the ability to deal with translations but for other dimensions of an affine transformation, replicating feature detectors based on number of dimensions or increasing the size of the training set with labeled masks are both exponential inefficiencies of the model.

Reflection

The project started with a simple data exploration, during which it became evident that a data pre-processing step is necessary to normalize the sizes of images to 256 by 256 pixels, eliminate the empty masks, and augment images to complement the training set. The research on different models to address the problem suggested that U-net is an adequate first approach due to its speed with a single GPU (22 seconds per epoch), ease of implementation, and the built-in contraction and expansion.

The interesting aspect of the project is the effect of pre-processing step to have significant better weights that result in better segmentation. The significant better results after the pre-processing step suggest the importance of the pre-processing and post-processing steps in image segmentation, one should spend a good quality amount of time designing these steps as they spend would designing and training the appropriate model.

The main difficulty of the project was with encoding the data image to arrays; different models have different input/output which holds true with different functions and libraries, thus understanding the documentation and understanding the data and the code very well is the key.

Overall the expectation of finding a preliminary solution to finding nuclei in pathological images was met and much more than expected was learned on the subject as the result: differentiating between instance and semantic segmentations, their respective potential solutions, handling images with different shapes and conditions, encoding the masks, calculating the correct IoU, and designing and training an U-net network model with a limited single GPU.

Improvement

It is clear from the results that I need to focus on generalizability and precision rather than accuracy. A better IoU score will be achieved with an average score across multiple experimental conditions than good scores on one or two conditions. Unfortunately, a little information is provided regarding the different experimental conditions (grouping the images based on the origin of experiment or the lab or the biologist) which could have helped us put the images in different classes. There are three initial areas where I should consider improving the solution as the future work: 1. Combat the overfitting issue 2. Complement the U-net model with other(s) models to create an ensemble model 3. Address the nuclei with wide holes and gaps.

1. Combat the overfitting issue

The results suggest a big overfitting problem where the train/test split accuracy and loss rates are acceptable yet the final test set IoU doesn't reflect that, one approach could be experimenting with different input image sizes during training. The input images have different shapes and conditions, we resized them all to 256 x 256 pixels but a nucleus can be shown in an image with different sizes and shapes, for example an image of size 1098 x 800 pixels that has a single nucleus with 500 by 200 pixels when the image gets resized to 256 x 256, it has lost the valuable information regarding the actual size and shape of the nucleus, hence the model can not extract that learning feature about an nucleus. Training the U-net model with different input image sizes or yet better maybe finding a way to train the model without a set image size but maybe a range of sizes and use k-fold cross validation where k is the number of experimental conditions.

2. Create an ensemble model

To make the designed networks complementary to our proposed U-net, an approach of detecting objects in an image while simultaneously generating a segmentation mask for each nucleus should be added to the final solution. My literature review suggests Mask R-CNN as a promising approach that has outperformed the models in COCO 2016 challenge winners. Mask R-CNN adopts the same two stage procedure of Faster R-CNN: 1. Region Proposal Network (RPN) and 2. Feature extraction by RoIPool. However, in the second stage Mask R-CNN outputs a binary mask for each RoI in parallel to predicting the class and box offset.

3. Address the nuclei with wide holes and gaps

Perhaps a solution here would be the use of a weighted loss, where the separating background labels between touching cells obtain a large weight in the loss function, a custom weighted loss function in which the cross entropy penalizes at each position the deviation of pixel(l) from 1 using:

$$E = \sum w(x) \log\left(pixel(l)\right)$$

where $l : \Omega \rightarrow \{1,...,K\}$ is the true label of each pixel and $w : \Omega \rightarrow R$ is a weight map that gives some pixels more importance in the training.

vi. References

[1] K. Nguyen, A. Sarkar, and A. K. Jain, "Structure and context in prostatic gland segmentation and classification," *Med. Image Comput. Comput. Interv. - MICCAI 2012*, vol. 7510, pp. 115–123, 2012.

[2] A. Ahmad, A. Asif, N. Rajpoot, M. Arif, and F. ul A. A. Minhas, "Correlation Filters for Detection of Cellular Nuclei in Histopathology Images," *J. Med. Syst.*, vol. 42, no. 1, 2018.

[3] M. Veta *et al.*, "Assessment of algorithms for mitosis detection in breast cancer histopathology images," vol. 20, pp. 237–248, 2015.

[4] Y. Al-kofahi, W. Lassoued, W. Lee, B. Roysam, and S. Member, "Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images," vol. 57, no. 4, pp. 841–852, 2010.

[5] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam, "Original Article A Hybrid 3D Watershed Algorithm Incorporating Gradient Cues and Object Models for Automatic Segmentation of Nuclei in Confocal Image Stacks," vol. 36, pp. 23–36, 2003.

[6] P. Kainz, M. Pfeiffer, and M. Urschler, "Semantic Segmentation of Colon Glands with Deep Convolutional Neural Networks and Total," pp. 1–15.

[7] Y. Xu *et al.*, "Gland Instance Segmentation Using Deep Multichannel Neural Networks," 2017.

[8] Booz Allen Hamilton and Kaggle, "2018 Data Science Bowl: Find the nuclei in divergent images to advance medical discovery," *Kaggle*, 2018. [Online]. Available: https://www.kaggle.com/c/data-science-bowl-2018.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," pp. 1–8, 2015.

[10] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P. Heng, "DCAN : Deep contour-aware networks for object instance segmentation from histology images," *Med. Image Anal.*, vol. 36, pp. 135–146, 2017.