Differentiating Tumor Cells from Healthy Cells in a Tumor Biopsy using CNNs

Aditya Mundada
Stanford University
amundada@stanford.edu

Shiv Raj Kaul
Stanford University
shivkaul@stanford.edu

Varsha Sankar
Stanford University
svarsha@stanford.edu

Abstract

In this paper, we attempt to classify cells seen on a cancer biopsy as malignant or healthy using different semantic segmentation techniques. Specifically, we investigate super-pixel based methods and a pixel by pixel classification using fully-convolutional networks. While fully-convolutional networks show promise, we get best results with 16x16 size super-pixel method. We work with image samples obtained from breast cancer tissue from two different institutions, NKI and VGH as in [3].

In this paper, we first review the literature in section[2] for similar work or methods developed to solve similar problem. We then present the dataset in section[3] and discuss its nuances. Section[4] covers the methods that we explored to solve the defined problem and section[5] discusses the results that we have obtained from our methods.

1. Introduction

Identifying cancer cells and healthy cells from pathology images of cancer biopsies is an important part of the disease diagnosis. In our project, we apply Computer Vision techniques to a set of cancer biopsy images to differentiate between tumor cells and healthy cells. The motivation behind the project is that these methods would assist doctors and pathologists immediately identify cancerous cells thereby enabling them to focus on investigating finer details about the disease. A logical extension of this technique would be to calculate the ratio of malignant cells to normal cells to gauge the progress of the disease, which could then be used to predict the number of years the patient would survive. These techniques, if successful, could be extended to understand the differences in visual manifestation of different types of cancer cells. This will not only help in detecting if the cancer has spread in the body, but will also enable us to detect early signs of cancer.

We will test several different deep-learning models (Convolutional Neural Networks) for this task. For each one of our models, the input to our network will be a cropped image obtained from the cancer pathology images. The size of this crop will depend on the model being used. For our super-pixel models, the labels are scalar values indicating tumor, non-tumor and background obtained corresponding to Red, Green and Black regions in the label images. The network thus outputs a single class label for each input image crop. Fully-Convolutional Networks (FCNs) are trained with labels being images of the same size as the input, where each each pixel corresponds to a class label. Thus this network produces an output containing a class label for each pixel in the input crop. We will discuss the details of our input data, class labels, and deep-learning models in the subsequent sections.

2. Related Work

Our project is based on the problem discussed in [3]. In this paper, the authors present a quantitative approach to detecting features in breast cancer epithelium and stroma to eventually determine the cancer's histological grade. They derive models that analyze morphological features that identify characteristics of prognostic relevance and provide means to assess prognosis from microscopic data. This work inspires us to take a CNN based approach where we believe a good CNN model should automatically learn the features.

Proposed deep-learning methods of addressing the tumor detection problem have recently come into existence. For example, in [14] they discovered that the classification of sub-types of cells helps in survival prediction. Another [15] study used feature extraction from raw pixel values to distinguish between epithelium and stromal cells. Current state of the art deep learning models for cancer detection are able to reduce human error by 85 percent when combined with a pathologist's labeling [13].

Semantic segmentation method lends beautifully to this task as discussed in [12], [5] and [6]. Authors in [12] as well as in [10] propose a fully convolutional network for semantic segmentation with a special skip architecture that combines semantic information from deep coarse layer and appearance information from shallow fine layer to produce detailed segmentations. We believe this architecture could work well for our problem as well as it shows best results on PASCAL VOC and SIFT Flow.

Authors in [4] attempt to segment neuronal structures depicted in stacks of electron microscopy images. They predict the label of each pixel based on the raw values of pixels in a window centered around the pixel under consideration. Authors in [9] [8] take a similar approach where in they create a 'supervoxel' from small clusters of voxels of similar intensity. Working with supervoxels speeds up their algorithm by several orders of magnitude while maintaining reasonable accuracy. Yet another superpixel based approach is explored in [11] for Gleason grading system in prostate cancer diagnosis. The idea to work with pixel crops in our project is vaguely motivated from these papers.

3. Dataset

The dataset consists of H&E-stained histological images from breast cancer tissue TMAs from two independent institutions: NKI and VGH. All images are available at tma.stanford.edu/tma_portal/C-path/. The dataset was preprocessed as described in the Beck paper.

We have 107 labeled images from NKI institution and about 51 labeled images from the VGH institution. Each image is 1128 x 720 pixels in resolution. While the number of images seems to be small for deep learning models, we work around this by taking crops of the original image. The sizes of these crops vary based on the deep-learning model we used. In the super-pixel method, we considered crops of sizes 32 x 32 and 16 x 16. By using 32 x 32 non-overlapping crops, we obtained a total of $121,660$ images and by using 16 x 16 non-overlapping crops, we obtained a total of $497,700$ images. For our Fully Convolutional model, we took overlapping 256 x 256 pixel crops of the original input image, instead of feeding in the whole images in full resolution to keep the number of parameters of the model within reasonable limits. In this case, each image would produce 15 crops of this size, giving us a dataset of 2370 crops.

A sample pathology image is shown in Figure 1. This image is 1128 x 720 pixels in dimension. For every such image in the dataset, there is a corresponding label image to indicate which regions contain cancer cells and which regions do not. Figure 2 shows the label image.

In figure 2, the red regions indicate the presence of cancer cells while the green regions indicate healthy cells. The black region represents either background or a region with cells which the pathologist could not classify as cancer with certainty. If figure2 is overlapped with 1, the dark violet regions in figure 1 with thick blobby patch of cells can be seen as cancerous cells. The dark and small, well defined black spots can be inferred as healthy cells. The rest of the image, shown as black in figure 2, contains other cells and tissue matter.

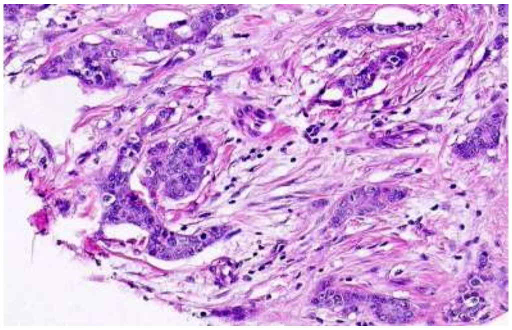


Figure 1. An image depicting one of the tissue biopsy under the microscope.

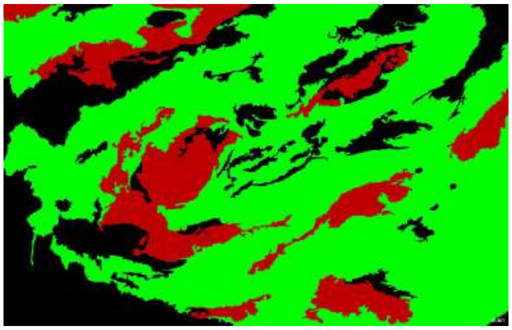


Figure 2. Image of a tissue biopsy under the microscope.

The train, validation and test splits were different in different methods and thus they are discussed in the Methods section. As mentioned earlier, pre-processing by taking smaller crops from these fairly high resolution images, we were able to increase the size of our dataset. When we tried to further augment our training set by including horizontal and vertical flips, significant improvement was not observed.

4. Methods

We can broadly divide our methods into two categories, as discussed below.

4.1. Super-pixel Based Approach

In this method, we divide the input pathology images and the corresponding label images in to small crops of sizes 32 x 32 and 16 x 16. The reason for choosing these numbers is that, a 16 x 16 patch captures atleast one cell. So if we take smaller crops, the CNN models may not be able to learn meaningful features, even though we would have a larger dataset.

We treat each image crop as a super pixel and assign a scalar value ϵ {0,1,2} as the label, based on the color of the majority of the pixels in the corresponding label image crop. Here, 0 indicates tumor cells, 1 indicates non-tumor cells and 2 indicates background. Figure 3 shows a few sample 32 x 32 input crops (along the first row) and the respective label crops (in the second row). Based on the label crops, we assign labels 1, 2 and 2 to the input crops here.

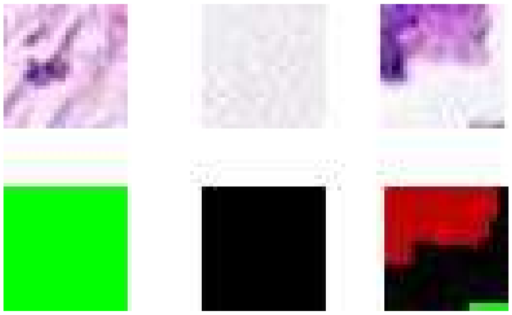


Figure 3. Sample 32 x 32 crops from original and label images.

The dataset was divided into training, validation and test splits consisting of the crops corresponding to 107, 5 and 46 input images. Several CNN models were trained on the data, the architectures of some of which are discussed below. These models were implemented using tensorflow[1].

4.1.1 Simple CNN

We used a simple CNN consisting of 1 Convolutional layer with 32 filters of size 7 x 7 and stride 2, followed by a ReLU and a Fully connected layer which outputs a vector of size 3, containing the scores for the 3 classes. We apply Hinge Loss and Adam Optimizer. This model was trained for 20 epochs on both 32 x 32 and 16 x 16 crops. From the results, we observed that the performance on 32 x 32 was not good enough, as we are approximating a relatively large region to have a single label, even though it contains two or three classes. However, 16 x 16 was doing better in terms of the quality of the output images. To improve the performance on 16 x 16, we tried several deeper CNNs and one of the models which performed well is discussed below.

4.1.2 Complex CNN

The architecture of the complex CNN trained on 16 x 16 crops consisted of 17 layers, i.e., conv, ReLU activation and Batch Norm layers, 4 of each, 2 max pool layers and 3 FC layers[1]. The arrangement of these layers can be seen more clearly in Figure 4

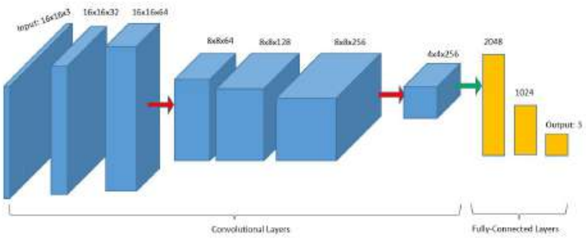


Figure 4. Complex CNN architecture (for 16 x 16 crops)

As seen in the figure, we feed input RGB images of size 16 x 16 into the CNN. It is passed through a convolutional layer with a zero padding to retain the same spatial dimensions at the output and then through ReLU activation and Batch Norm layers. The same structure repeats, now with different number of filters, before a max pool layer of kernel size 2 and stride 2, to down sample. This is followed by another set of the same architecture and then 3 Fully Connected layers consisting of 2048, 1024 and 3 units respectively. The kernel size for all convolutional layers is 5 x 5. The number of filters in the conv layers increases from 32 to 256 by a factor of 2 at each of them.

4.2. Fully Convolutional Network

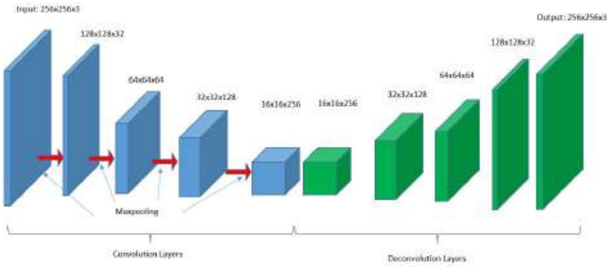


Figure 5. Our fully convolutional model architecture

In order to achieve better resolution in our output images, we needed an architecture that could classify each individual pixel as tumor, non-tumor, or background. We could accomplish this by using a fully convolutional network. We split the previously mentioned 2370 256x256 pixel crops into 1659 training samples, 237 validation samples, and 474 test samples. The specific architecture we

[1] Used CS231N HW Assignment-2 code as starter code for implementation.

used can be seen in Figure 5.

The figure shown displays the dimensions of the output of each convolution/deconvolutional layer. The first two dimensions shown above the rectangles correspond to the height and the width of the output, while the last dimension corresponds to the number of filters used at the corresponding convolutional/deconvolutional layer. The conv/deconv layers were followed by ReLU activation and batch normalization. In between convolution layers we used max-pooling with a stride of 2 pixels in order to reduce the pixel dimensions. Similarly, we used upsampling deconvolution as a means of increasing our pixel dimesions in order to produce an network output that is the same size as the network input.

We choose relatively small kernel dimensions (11 x 11, 5 x 5 and 3 x 3) for our convolutional layers. This was done because from our inspection of the ground-truth/label-image pairs, we observed that classification seemed to depend on local information. In addition, models with larger kernel sizes would likely require more data for training or run the risk of overfitting or even memory storage issues due to the large number of parameters. We pass a 256 x 256 x 3 crop into the image, where the last dimension corresponds to the RGB values. The output of the network is also 256 x 256 x 3, however now the last dimension represents a probability distribution of sorts over our three possible class labels. This allows us to compute a softmax cross entropy loss for each pixel in a 256 x 256 crop.

5. Experiments and Results

We have both quantitative and qualitative results from the two methods. We quantify our results by the training, validation and test accuracies as well as the confusion matrices which gives information about fraction of true and incorrect classifications. Quality of our results is determined by the actual output images from the CNNs (Reconstructed Images) and Saliency Maps. They are described in the subsequent sections.

5.1. Hyperparameter Selection and Preprocessing

We trained both our super-pixel based models and our FCN using an Adam Optimizer using a learning rate of $1e{-4}$. Using lower learning rates required far too many epochs to train. Higher learning rates lead to our losses plateauing at a suboptimal value or losses increasing after an initial decrease. We chose a batch size of 64 for our super-pixel based models. Batch sizes smaller than these resulted in higher variance in batch losses and accuracies, while larger batch sizes increased the training time considerably. However for the FCN, we used a Batch size of 32, because it had relatively smaller number of training samples, which at the same time were much bigger than the ones in the super-pixel method. When we tried augmenting our training set by taking horizontal and vertical flips, it didn't result in any significant improvement and was also taking considerably longer time to train. We included Batch Normalization layers after convolutional and Fully connected layers in most of the models to speed up the training and improving the accuracy.

5.2. Losses and Accuracies

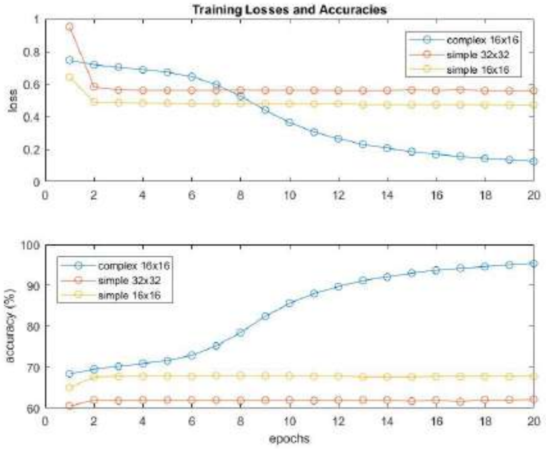


Figure 6. Training losses and accuracies for super-pixel based method

Figure 6 shows the loss and accuracy curves across 20 epochs for the super pixel based method. It can be seen that the simple 16 x 16 model performed the best on the training set, giving the lowest training loss (less than 0.2) and highest training accuracy of 95.4%. But it also indicates that, it might have overfitted the training data. The plot also suggests that training the 16 x 16 model for more epochs would cause the loss to decrease further. The simple 32 x 32 model and the complex 16 x 16 performed similarly, where the loss plateaus off early in time and at sub-optimal values. Also the accuracy gets saturated around 60% and 70% respectively.

Our FCN's training time losses and accuracies can be seen in Figure 7. We only trained for 15 epochs due to time constraints, but like the simple 16 x 16 super-pixel model, the plot seems to suggest that if we had trained for longer we would have continued to see the loss decrease. However, in practice, we found that training for a longer duration would cause the FCN to overfit more severely.

On comparing the super-pixel based method and the

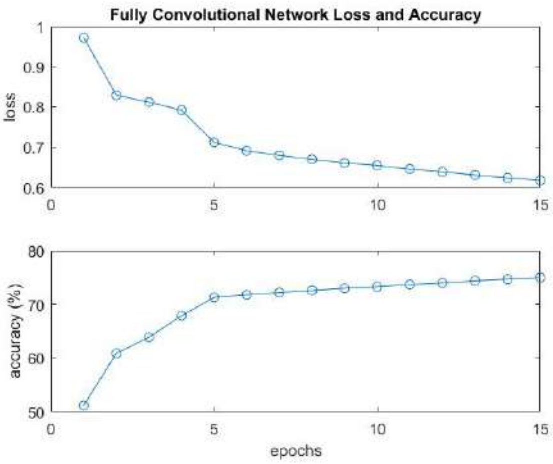
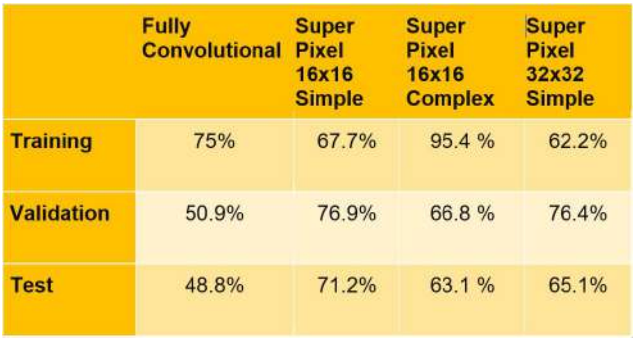


Figure 7. Training losses and accuracies for our fully convolutional network



	Fully Convolutional	Super Pixel 16x16 Simple	Super Pixel 16x16 Complex	Super Pixel 32x32 Simple
Training	75%	67.7%	95.4 %	62.2%
Validation	50.9%	76.9%	66.8 %	76.4%
Test	48.8%	71.2%	63.1 %	65.1%

Figure 8. Training, validation, and test accuracies for all of our models

FCN in terms of the learning curves, super-pixel based method appears better due to large training accuracy. But to see if it is overfitting the underlying training set, we compute the accuracy on Validation and test set using all the methods. From the chart in Figure 8 we can see clear overfitting in both the FCN and the simple Super Pixel 16 x 16 models. Additional forms of regularization like dropout, however, are needed in order to improve the aforementioned models. Other super-pixel models did not suffer from overfitting.

5.3. Confusion Matrices

The quantitative results for all our super pixel-based methods in terms of fraction of the true classification and misclassification are summarized in the confusion matrices ([7]) in Figures 9, 10 and 11. It can be seen that the simple CNN using 32 x 32 crops classifies the tumor (red) and non-tumor (green) regions reasonably well, while fails in classifying the background (black) correctly. It in turn seems to classify most of the black pixels as non-tumor. On the other hand, the simple CNN using 16 x 16 crops is much better in that regard. It classifies about 50% of the black pixels correctly. Also, there is huge increase in true classification of the tumor cells to 90%, which is the most important requirement. The Complex CNN on 16 x 16 is comparable to that of the simple CNN. Though it performs better with respect to the non-tumor and background cells, it is worse at correctly classifying the tumor cells. Thus the simple CNN 16 x 16 seems to be the best among the 3 with respect to the confusion matrices.

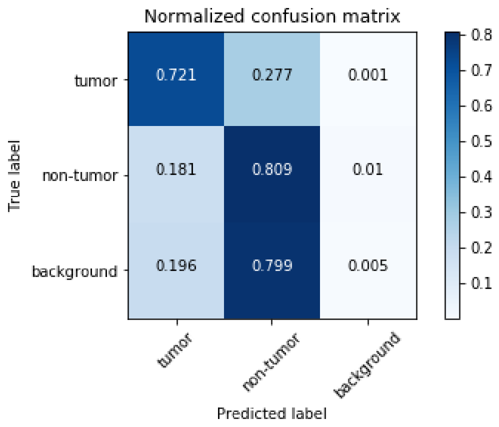


Figure 9. Confusion Matrix of Simple CNN 32 x 32

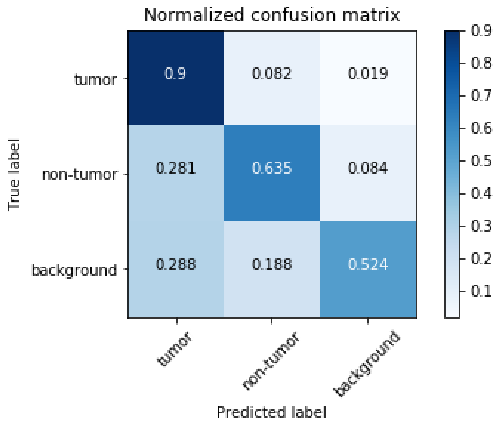


Figure 10. Confusion Matrix of Simple CNN 16 x 16

Figure 12 shows us that a major shortcoming of our FCN is that it fails to correctly classify any pixels as non-tumor. It classifies nearly all non-tumor pixels as background. This may be due to the fact that the so-called ”background” class corresponds to two different labels - the biopsy slide background, as well as the cellular regions the pathologist could not classify with certainty. Tumor cell pixels were classified with 69.7% accuracy. When they were misclassified they were exclusively predicted to be background cells.

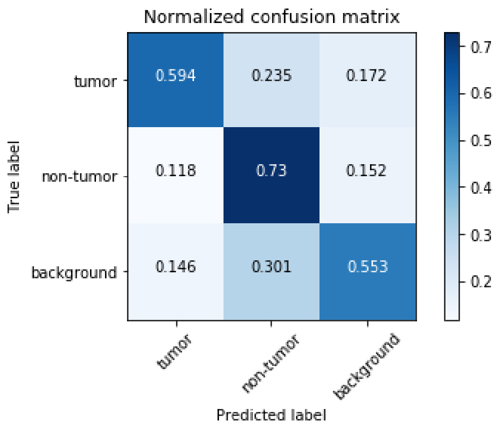


Figure 11. Confusion Matrix of Complex CNN 16 x 16

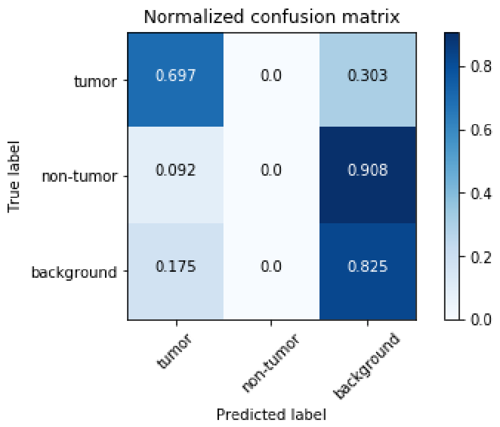


Figure 12. Confusion matrix for our fully-connected network

5.4. Visualizing Network Output

In the super-pixel based method, the output of our network is a single class label for any given super pixel. With this label, we generate a super-pixel sized RGB image representing the crop. More concretely, if our super-pixel size is 16 x 16, then we will generate a 16 x 16 RGB image which is completely red, green or black based on the label obtained from the model. We stitch together these images for all the super-pixels that constitute a given test image. Such reconstructed output images from various methods along with the actual test input and label images are shown in Figure [13].

Notice that the 32 x 32 output is very pixelated. This is expected considering that we are labeling 1024 pixels with the same color as opposed to a very smooth approach in the original labeled image. The pixelation reduces in 16 x 16 crops and the image appears much more smoother. Our best model - simple 16 x 16 matches very closely to the original labeled image. While the complex model in 16 x 16 is able to detect the features, we see a lot of noise in form of spurious black labels in between due to overfitting. It makes us conclude that due to minimal underlying features of the cells represented in the image, complex models over fit very easily while simple models are more adept at learning the right features.

In Figure [14], we see that the output of FCN is as smooth as our original labeled images. However, our FCN model is not able to distinguish between green and black correctly and ends up classifying all the green pixels as black. However, the correlation between the original labeled image and output image for red pixels is apparent. Note that our output correlates very highly with the original biopsy image. The dark violet colored region is correctly marked as tumor by the algorithm whereas the labeled image has a mixture of labels with sharp feature distinction. We conclude that the algorithm is broadly successful in learning the features of the image but in a dense region with multiple labels, fails to segment the features.

Figure[15] and Figure[16] shows examples that were misclassified by 16x16 and 32x32 super pixel method respectively. For the 32x32 method, we see that the black labels were not identified. This is attributed to the fact that black labels are marked in two regions with two completely distinct features - background region which appears completely white as in Figure[16] and unclassified region as in Figure[15]. This makes it difficult for the algorithm to tune parameters for black label which frequently gets misclassified as green or red. The semantic segmentation output in Figure[14] is a very good example of this observation. In this figure, we see that the green label has been misrepresented as black.

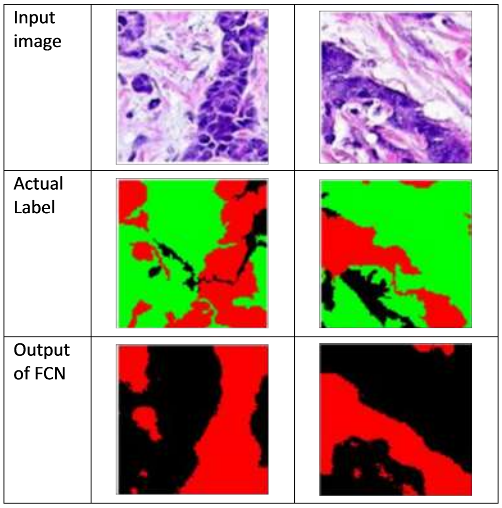


Figure 14. Input, Label and the output crops from FCN

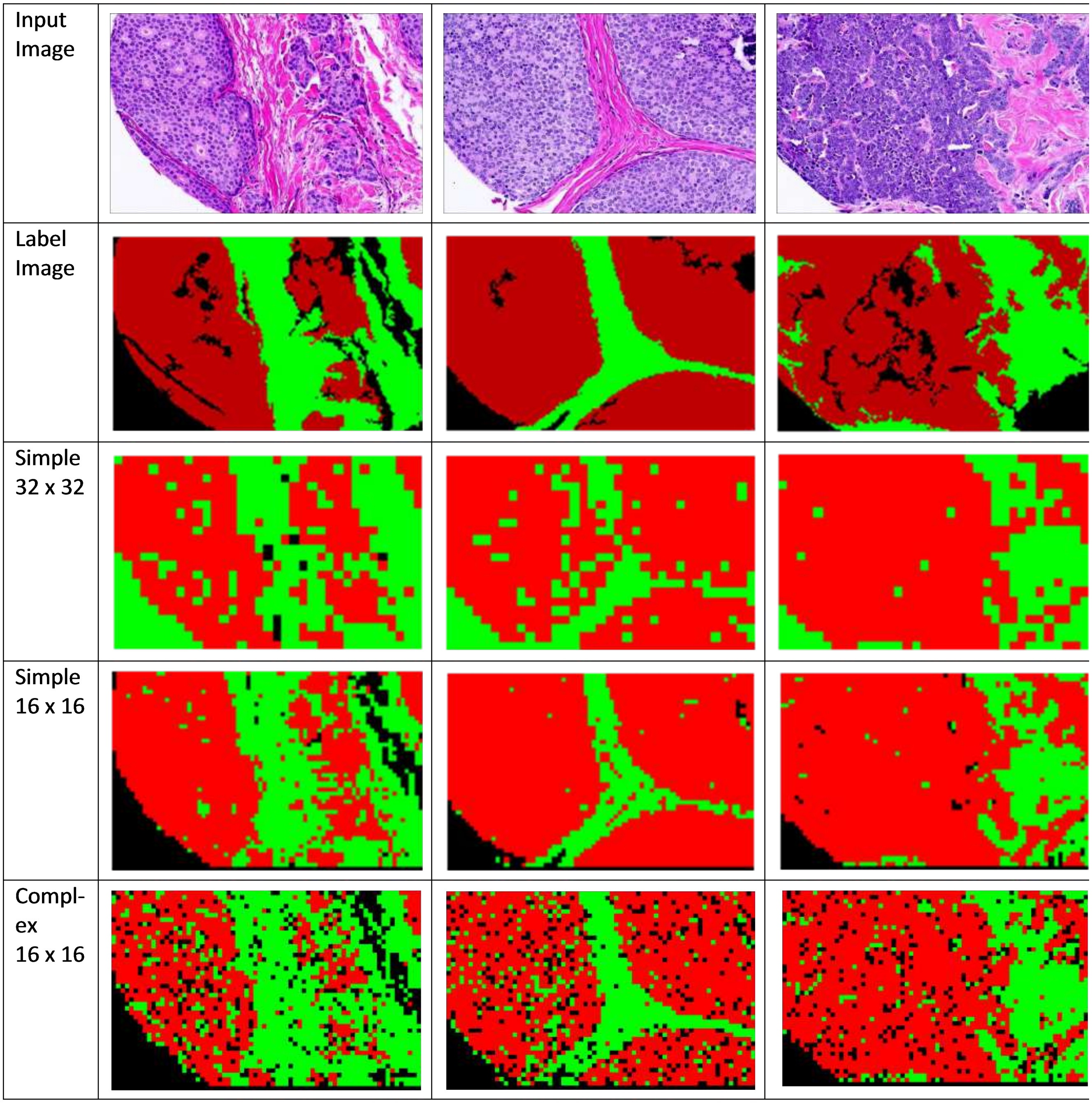


Figure 13. Reconstructed output images from Super-pixel method with corresponding inputs and labels

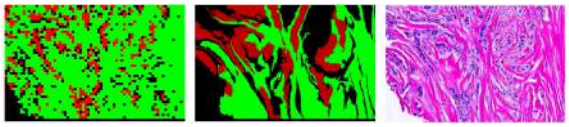


Figure 15. Misclassified output-1

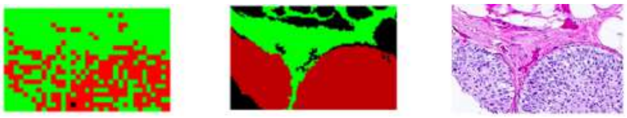


Figure 16. Misclassified output-2

5.5. Saliency Maps

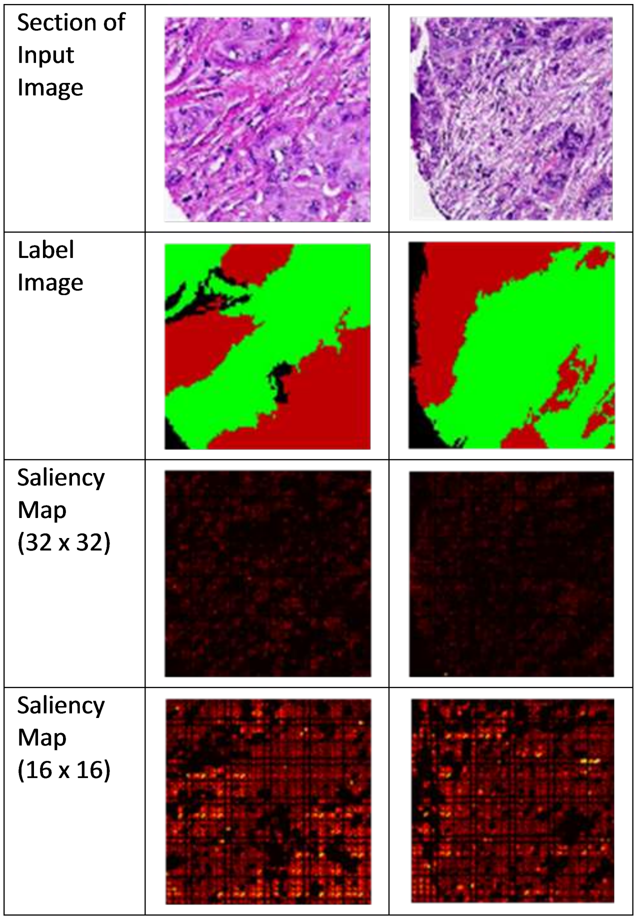


Figure 17. Saliency maps corresponding to image crops after reconstructing using 32 x 32 and 16 x 16 methods

To observe what features the network learns from the training set, we visualize the saliency maps corresponding to the network outputs[2]. Figure 17 shows the saliency maps from 32 x 32 and 16 x 16 methods corresponding to some sections of input images. Since the output from these networks are also of the same size as the superpixel, the saliency maps corresponding to the superpixels had to be stitched together to obtain the ones shown in the figure.

As we can see, the 16 x 16 method seems to be capturing more features than the 32 x 32 method. However, we are not able to interpret what kind of features are being captured as in the original input images, there is not much significant visual distinction between the cells belonging to different categories.

6. Conclusion/Future Work

Although we expected our FCN to perform the best, it turned out that our simple 16x16 super-pixel model gave the best looking labeled output images. This was also reflected in this model's quantitative performance. One reason for the simple model to outperform the complex could be that the input images have very minimal visually distinguishable features and at the same time black label is assigned to cells as well as background.

If we could have pathologists label the images with four classes - tumor, healthy, uncertain, and background - we would expect that our FCN would produce more desirable results. Additionally, if we had more data, all of our models would benefit. In order to improve our FCN, we could use the method proposed by [2], in which the up-sampling in the deconvolutional layers uses the indices from the max-pooling done in the convolutional layers.

In absence of more detailed labeling, we can augment the black labels into two different classes based on the underlying pixel values in the original image. This would help us identifying the background from regions that are unclassified. While 16x16 super-pixel model performed really well, semantic segmentation lends itself better to this problem and we would like to explore different models in FCN to better learn the features of the images. Augmenting black labels along with better FCN models should intuitively outperform other approaches.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[3] A. H. Beck, A. R. Sangoi, S. Leung, R. J. Marinelli, T. O. Nielsen, M. J. van de Vijver, R. B. West, M. van de Rijn, and D. Koller. Systematic analysis of breast cancer morphology uncovers stromal fea-

[2] Used CS231N HW Assignment-3 code as starter code for implementation.

tures associated with survival. *Science translational medicine*, 3(108):108ra113–108ra113, 2011.

[4] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[5] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[6] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.

[7] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[8] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua. Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *IEEE transactions on medical imaging*, 31(2):474–486, 2012.

[9] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua. A fully automated approach to segmentation of irregularly shaped cellular structures in em images. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pages 463–471, 2010.

[10] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[11] J. Ren, E. Sadimin, D. J. Foran, and X. Qi. Computer aided analysis of prostate histopathology images to support a refined gleason grading system. In *SPIE Medical Imaging*, pages 101331V–101331V. International Society for Optics and Photonics, 2017.

[12] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.

[13] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[14] S. Wang, J. Yao, Z. Xu, and J. Huang. Subtype cell detection with an accelerated deep convolution neural network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 640–648. Springer, 2016.

[15] J. Xu, X. Luo, G. Wang, H. Gilmore, and A. Madabhushi. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing*, 191:214–223, 2016.