PROSTATE CANCER DIAGNOSIS BY DEEP LEARNING

By

Shuijing Liu

Senior Thesis in Computer Engineering

University of Illinois at Urbana-Champaign

Advisor: Aditya Parameswaran, Jian Peng

May 2018

Abstract

Prostate cancer diagnosis by biopsy images of human tissue requires experienced trained pathologists and the cost is high. To facilitate prostate cancer diagnosis, we built and trained binary classifiers using deep convolutional neural networks (CNNs) on two datasets: one contains cancerous and healthy biopsy images of prostate tissues (referred as Dataset1), and the other contains biopsy images of tissues with recurred cancer and fully recovered tissues (referred as Dataset2). We extracted patches from biopsy images of human tissues, then built and trained CNN models to classify the patches. We achieved 82% test accuracy on Dataset1 and 63% accuracy on Dataset2.

In addition, we used ensemble methods to further boost the performance. With predictions of all patches in our datasets, we performed majority voting on the image level, and the accuracy increases by 5% to 10% on the first dataset. Then we used Bootstrap Aggregation (Bagging) to further increase accuracy to 100% on Dataset1. However, the two-step ensemble methods above have little influence on the accuracy of Dataset2. When visualizing the predictions on the second dataset returned by our models, no clear patterns are found that can distinguish the two classes.

Subject Keywords: Image Processing, Machine Learning, Bioinformatics, Computer Vision

Acknowledgments

I would like to give special gratitude to my advisors Dr. Aditya Parameswaran and Dr. Jian Peng for their continuous support and guidance.

I would like to thank my collaborators Andrew Kuznetsov and Ke Xu for their insights and contributions to my research.

Also, I would like to thank my parents, Qing Luo and Dongxu Liu, for their love and support.

Contents

1. Introduction

1.1 Motivation and Goals

Cancer is the second most common cause of death in the United States from 2012 2017. It causes 22.5% of total deaths in this period. According to American Cancer Society [1], prostate cancer, the third most common cause of death of males in the United States, is reported with 164,690 new cases and 29,430 deaths in 2018 [1].

While early detection and diagnosis of prostate cancer often allows for more treatment options and increases a person's chances of survival, detection by professional pathologists, which involves examining prostate tissue biopsies, is labor-intensive and error-prone. Moreover, different pathologists may have different methods for detection and may have conflicting opinions on the same case. Therefore, developing a systematic rule for cancer diagnosis is necessary. Moreover, automating the diagnosis process can largely reduce the time and costs, as well as minimize errors.

For this reason, we used biopsy images of prostate tissue from the US Biomax as our datasets. We used a deep learning approach to classify two datasets: the first dataset contains biopsy images of healthy and cancerous tissues (referred as Dataset1), and the second dataset contains biopsy images of tissues with recurred prostate cancer and fully recovered tissues (referred as Dataset2).

1.2 Related Work

Ciresan et al. [2] detected mitosis in breast cancer histology images using deep neural network, which classifies patches from images that center on the cell nucleus and yields a probability of the nucleus being in the mitosis process. [2] computed the local maximum probability in each image and label these areas as mitotic. Liu et al. [3] successfully classified tumor patches and normal patches using multi-scale convolutional neural network (CNN) and generated heatmaps for tumorous regions on Camelyon 16 dataset. However, few previous researchers studied the automatic detection of prostate cancer, which leads us to this project.

Wang [4] trained a 7-layer LeNet to classify our Dataset1 and achieved around 70% accuracy [4]. He also proposed that in theory, ensemble methods can boost the performance of CNN models to nearly perfect accuracy. Following his discovery, we trained CNN models with better performance based on deeper and more modern architectures than LeNet. And we explored the classification of Dataset2 using our CNN models and pushed this project forward to a new level.

2. Image Preprocessing

Deep learning requires a large amount of data to learn feature representations with different abstractness and achieve high performance. An example of our image data is shown in Figure 1 and 2. The number of images in our datasets is limited to several hundreds, and the size of our images is too large to be efficiently processed by our CNN models. Moreover, we had severe data imbalance problem in Dataset1. For these reasons, we need to sample smaller patches from our image datasets as ideal inputs to CNN models, and up-sample the minority classes or down-sample the majority class to solve the data imbalance problem.

2.1 Sampling the Patches

The size of tissue biopsy images in our datasets is around 1100 pixel x 1100 pixel. If we directly feed the raw images into our CNN models, the training time will be too long due to the large size of the images. Also, we need around 10,000 data to train a good CNN binary classifier. Thus, sampling patches from original images becomes a feasible and necessary step. As shown in Figures 1 and 2, since we are only interested in the red tissue regions, we developed an algorithm to detect the elliptical boundary between tissues and white blank areas (Figure 3). Then we randomly sampled patches of size 128 pixel x 128 pixel within the red ellipse, which served as input to our CNN models. Finally we divided the patches into training, validation, and testing subsets with ratios equal to 75 : 15 : 15. Validation data was used to tune parameters in the training phase, and testing data was used to test the performance of trained CNN models.

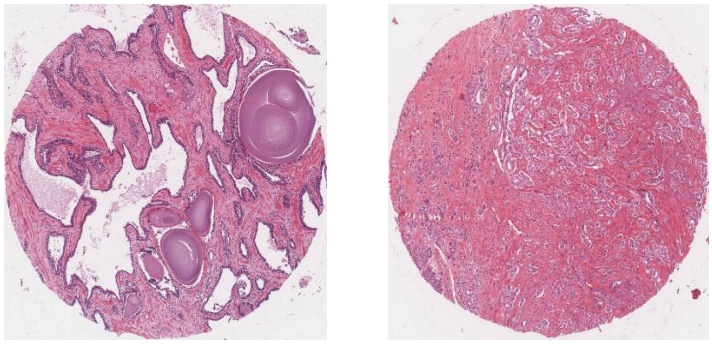

Figure 1. Sample images from Dataset1 (Left: biopsy of healthy tissues, Right: biopsy of cancerous tissues)

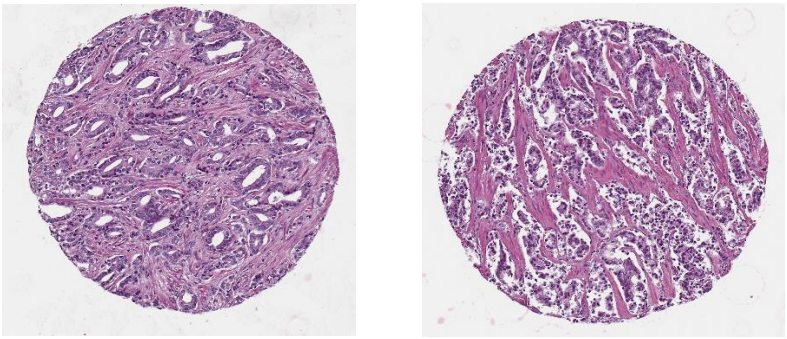

Figure 2. Sample images from Dataset2 (Left: biopsy of recovered tissues, Right: biopsy of tissues with recurred cancer)

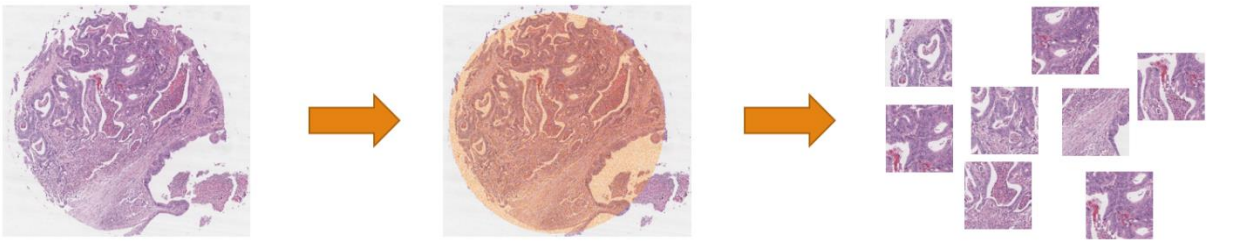

Figure 3. An illustration of our image processing steps

2.2 Upsampling and Downsampling

Due to the difficulty of obtaining biomedical data and thus limited data resource, we faced severe data imbalance problem in Dataset1 that contains 112 images in benign class and 409 images in cancerous class. Dataset2 is only slightly imbalanced and the ratio of recovered class to recurred class is 7:6. The data imbalance has a negative effect on CNN models since the majority class would dominate the training process, thus the model loses the chance to sufficiently learn the features of minority class and it tends to predict testing data as majority class. To solve this problem, we adjust the number of patches we sample from the majority class and the minority class to ensure that the total number of patches is roughly the same in the two classes.

3. Convolutional Neural Network

Convolutional neural network (CNN) is made up of multiple layers to learning image features with multiple levels of abstraction [5]; also, sequences of convolutional – batch normalization – activation layers in CNN allow it to learn features invariant to various image transformations. Because CNN is the most powerful tool in deep learning, and it has led to numerous breakthroughs in fields related to machine learning, we chose this architecture to build our classifier.

3.1 Architectures

We primarily used ResNet to train our datasets. ResNet has residual blocks with shortcut connections that are easier to optimize and can gain accuracy from considerably increasing depth [6]. As shown in Figure 4, each residual unit can be expressed as:

$$y_l = h(x_l) + \mathcal{F}(x_l, \mathcal{W}_l),$$

$$x_{l+1} = f(y_l)$$

where x_l and x_{l+1} are input and output of the l-th unit, $\mathcal{F}$ is a residual function, and $h(x_l) = x_l$ is an identity mapping and f is a ReLU activation function. Soon afterwards, [7] came up with an updated version of residual blocks with pre-activation structures that lead to better performance (see Figure 5), which can be expressed as:

$$x_{l+1} = x_l + \mathcal{F}(x_l, \mathcal{W}_l).$$

since f is also an identity mapping in this case. Let L denote the loss function of ResNet, then the backpropagation of the equation above can be expressed as

$$\frac{\partial L}{\partial x_l} = \frac{\partial L}{\partial \mathcal{F}(x_l, \mathcal{W}_l)} \frac{\partial \mathcal{F}(x_l, \mathcal{W}_l)}{\partial x} = \frac{\partial L}{\partial \mathcal{F}(x_l, \mathcal{W}_l)} (1 + \frac{\partial \mathcal{F}(x_l, \mathcal{W}_l)}{\partial x_l})$$

The term $\frac{\partial L}{\partial \mathcal{F}(x_l, \mathcal{W}_l)}$ ensures that the gradient can be propagated to any of the previous units. But our experiments showed that the newer ResNet had almost the same accuracy as the original ResNet in both of our datasets. Since our dataset size and number of classes are much smaller than the dataset in [6], we found that ResNet18 with half of its layers is sufficient without compromising the performance. The testing result on Dataset1 is 82% accuracy.

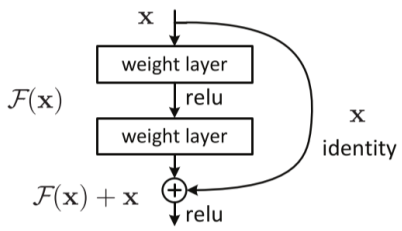

Figure 4. A residual block in ResNet [6]

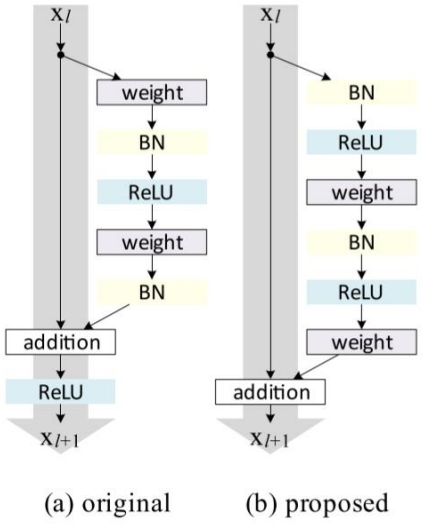

Figure 5. (a) Original residual block; (b) New residual block proposed by He et al. [7]

We explored ResNet, VGG Net, and Inception-v3 in Dataset2, all of which are previous winners of ImageNet Large Scale Visual Recognition Challenge (ILSVRC). VGG Net is much deeper than LeNet and therefore allows for more levels of feature representation. Inception-v3 has auxiliary units that ensures fast convergence and reduces gradient vanishing problem [8]. For the same reason as Dataset1 above, we used smaller versions of these CNNs. We find that ResNet not only achieved best accuracy, but also needs shortest training time due to its smallest number of parameters among the three. Therefore, the rest of our experiments are done on ResNet models.

Table 1. Comparison of 3 different CNN architectures trained on Dataset2

CNN architecture	ResNet9	VGG13	Inception-v3 (5 blocks)
Number of parameters	4,000,000	138,000,000	50,000,000
Training time (based on Amazon EC2 g3.4xlarge instance)	3.5 hours	18 hours	24 hours
Accuracy	63%	51%	59%

3.2 Overfitting Issue

As our learning curves shown in Figure 6 and Figure 7, we got 100% training accuracy on both datasets, and 82% testing accuracy on Dataset1 and 62% accuracy on Dataset2. Large gap between training and testing performance indicated an overfitting problem in our models, especially for Dataset2.

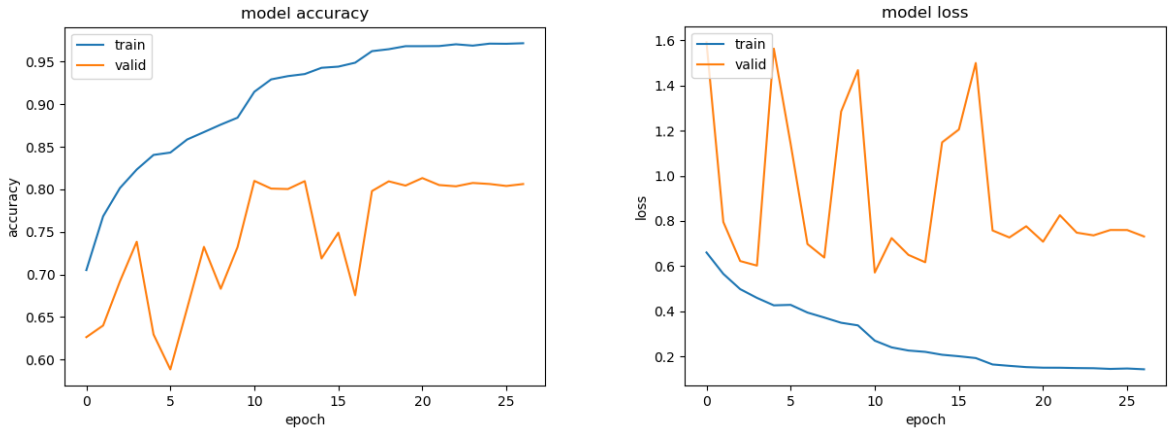

Figure 6. Learning curves of ResNet on Dataset1 (Left: Accuracy v.s. Epoch; Right: Loss v.s. Epoch)

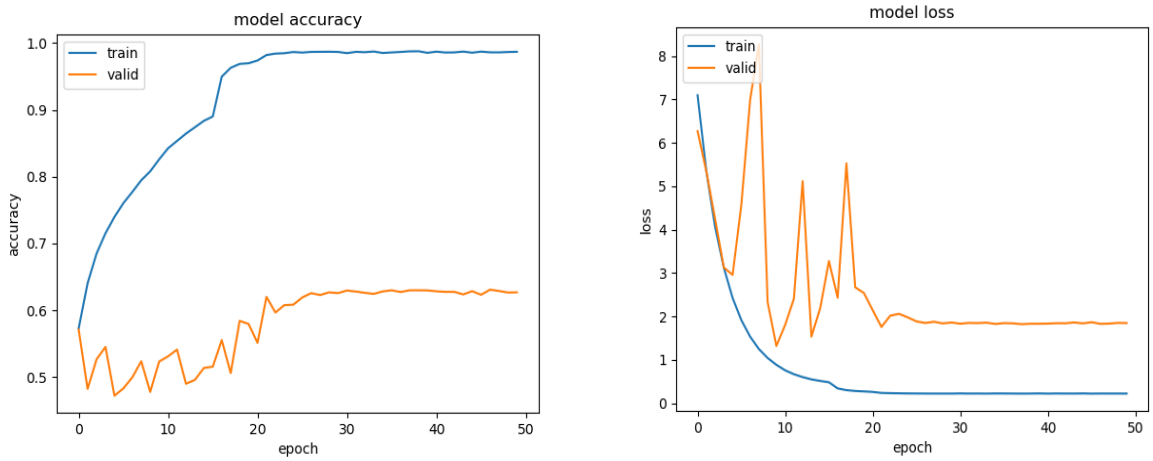

Figure 7. Learning curves of ResNet on Dataset2 (Left: Accuracy versus Epoch; Right: Loss versus Epoch)

To reduce overfitting, we tried the following three methods: (1) Increasing the size of the dataset; (2) adding dropout layers; and (3) reducing the number of layers in CNN models to decrease variance. We will discuss each of these methods one by one.

After we obtained more images, extracted patches from them, and merged the new patches with Dataset2, we trained the CNN model on the enlarged dataset but obtain the same result as the original smaller Dataset2.

As, shown in Figure 7, dropout addresses the overfitting problem by randomly removing a unit temporarily from the network with probability p [9]. In this way, the trained model acts as a combination of many different models, which usually improves the performance.

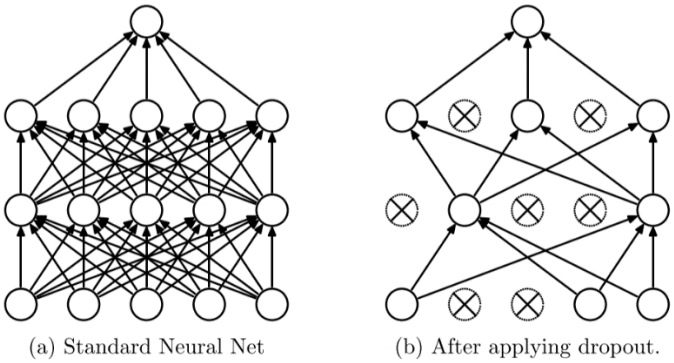

Figure 8. Dropout neural network model. Left: A standard neural network with 2 hidden layers; Right: An example of a thinned net produced by applying dropout to the NN on the left. [9]

We trained models with different dropout probabilities from 0 (no dropout) to 0.8, and as Figure 9 shows, accuracy constantly drops as dropout probability increases. Therefore, adding dropout layers does not improve the performance of our models.

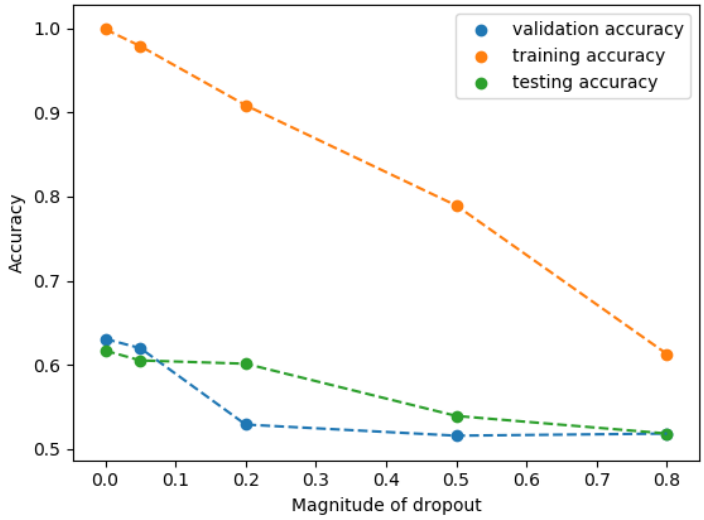

Figure 9. Accuracy v.s. Dropout magnitude on ResNet with 1 residual unit

In machine learning, variance is defined as an error from sensitivity to small fluctuations in the training data, and high variance can cause a model to fit noise data and lead to overfitting [10]. To reduce variance in our models, we experimented with ResNet from 4 residual units (i.e. ResNet18) to 0 residual units with only input layer and fully-connected layer at output. The accuracy versus number of residual units is plotted in Figure 10:

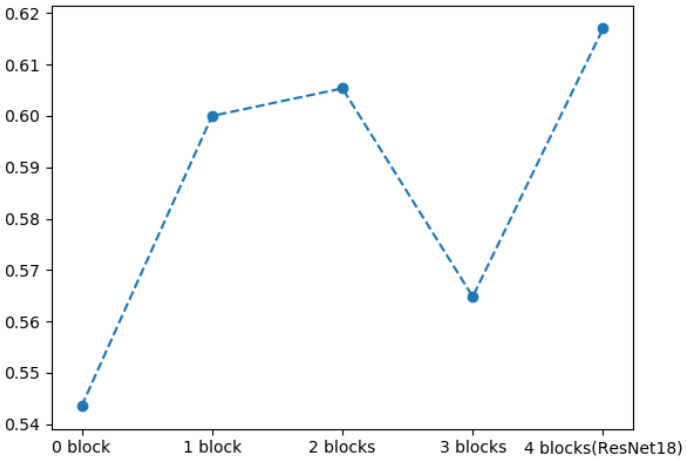

Figure 10. Accuracy v.s. number of residual units in ResNet

From Figure 10, it is obvious that smaller networks only reduce accuracy. It is due to the limit of feature representation of different abstractness in very reduced number of convolutional layers. Therefore, none of the three proposed methods was effective in solving the overfitting problem, which is still haunting us now.

4. Ensemble Methods

Ensemble methods use multiple learning algorithms to obtain better predictive performance than single learning algorithms [11]. In the first step of our ensemble method, we used the labels of patches from the same image to vote for a label for the image. In the second step, we performed Bootstrap Aggregating to give predictions by combining those of multiple CNN models. Both steps gave effective improvements on Dataset1, but did not have much effect on Dataset2.

4.1 Majority Voting: From Patches to Images

After we obtained labels of all patches from our CNN classifiers, we aggregated the patches from the same image and give the image a single label using majority voting. Remember that we randomly extracted 100 patches from each image. For any image in our datasets, suppose m patches are labeled as "positive" and $(100 - m)$ patches are labeled as "negative", then the image is labeled as "positive" if $m > 50$; otherwise it is labeled as "negative". An illustration of this process is shown below in Figure 11.

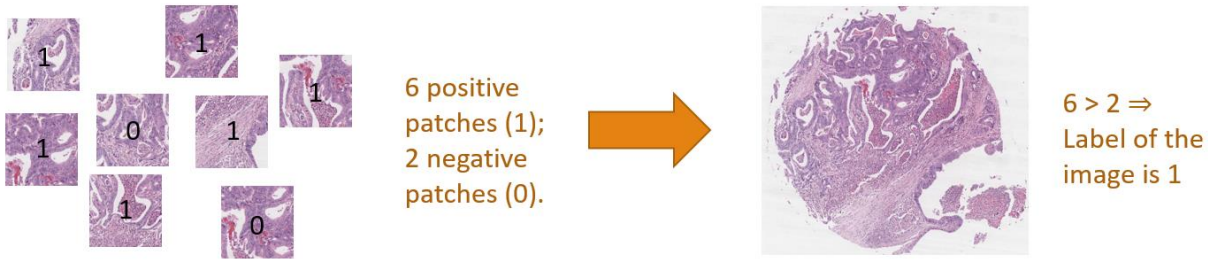

Figure 11. A simplified example of majority voting process (1 stands for "positive", 0 stands for "negative")

From Figure 13, the green bars indicate that accuracy in Dataset1 increases for 4% to 16% after Majority Voting.

4.2 Bootstrap Aggregating

Bootstrap aggregating, or bagging, is an ensemble algorithm that combines multiple algorithms and gives collective outputs by averaging the outputs of all algorithms (commonly used in regression) or voting (commonly used in classification) [12]. It also reduces variance and helps to avoid overfitting. With image-wise predictions given by majority voting, we applied voting again among our models trained by different hyperparameters such as learning rate, number of convolution layers, and regularization magnitude, and give collective predictions to images. As shown in Figure 13 and 14, bagging increased accuracy for another 2% to 3% in Dataset1 but had [13] no obvious effect on Dataset2.

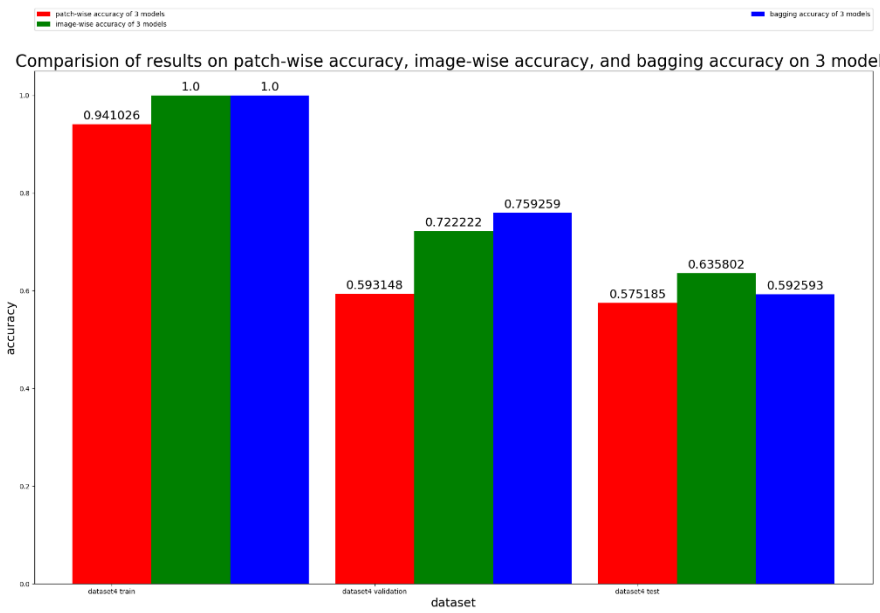

Figure 12. Accuracies of single CNN model (red), Majority Voting (green), and bagging (blue) on training, validation and testing data in Dataset1

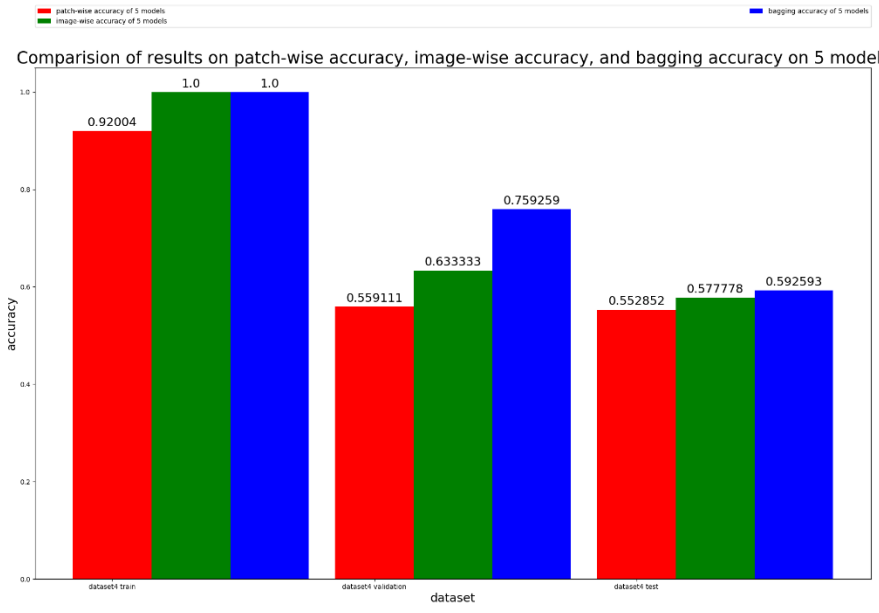

Figure 13. Accuracies of single CNN model (red), Majority Voting (green), and bagging (blue) on training, validation and testing data in Dataset2

5. Conclusions

Both our CNN models and ensemble methods achieved strong performance on classification of cancer/benign biopsy images in Dataset1. Our final accuracy is 100%, which indicates that CNN is capable of detecting key features of different abstractness that can distinguish between cancer tissues and healthy tissues.

However, the overfitting problem on Dataset2 is inconclusive despite attempts of three methods mentioned above. We hypothesize that the reason is the ambiguous nature of cancer recurrence diagnosis. As non-experts, we cannot find any strong visual clue that can distinguish the two classes in Dataset2. Since our CNN models have no expert knowledge either, it may converge to and get stuck at a local maximum while misses a far better global maximum. We believe that insights from pathologists may allow us to give prior knowledge to guide our CNN to converge to a better result.

6. Future Work

Our models are only focused on classification of prostate cancer datasets. It would be interesting to explore the performance of CNN models on other biological datasets, such as other human cancers or tissue of other animals.

Another interesting task is object detection and segmentation on prostate cancer datasets. It would be very useful in practice if we could locate the cancerous tissues and segment them from the rest of the neighborhood.

References

[1] "Prostate Cancer", American Cancer Society, 2018. [Online]. Available: https://www.cancer.org/cancer/prostate-cancer.html.

[2] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks," 2012.

[3] Y. Liu, K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado, J. D. Hipp, L. Peng, and M. C. Stumpe, "Detecting Cancer Metastases on GIgapixel Pathology Images," in *MICCAI*, 2017.

[4] B. Wang, "Prostate Cancer Diagnosis with Deep Learning," Undergraduate Senior thesis, University of Illinois at Urbana-Champsign, Urbana, IL, 2017.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* p. 1, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *European Conference on Computer Vision*, Amsterdam, 2016.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 2015.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research,* p. 1930, 2014.

[10] "BIas-variance tradeoff," 1 April 2018. [Online]. Available: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff.

[11] "Ensemble learning," 20 April 2018. [Online]. Available: https://en.wikipedia.org/wiki/Ensemble_learning.

[12] "Bootstrap aggregating," 20 February 2018. [Online]. Available: https://en.wikipedia.org/wiki/Bootstrap_aggregating.