Final Papers for STA561D Probability Machine Learning, 2019

Wenwen Ye,[*] Zicheng Yuan,[†] Xue Zou[‡]

Advances in the quality of clinical images have enabled the systematic digital recognition of metastatic cancer from pathologic scans [1], and deep learning algorithm has proved its effectiveness in identifying metastatic lymph nodes among MRI images [2]. To evaluate the accuracy and efficiency of digital assistance on identifying clinical-patches at different severity, we conducted a study utilizing deep learning algorithm for the detection of cancer metastasis in small image patches taken from larger digital pathology scans. Two hundred twenty thousand twenty-five images were used in the training and evaluation set, and two models were used to predict the binary labels and compared: four-layer Convolutional Neural network (CNN) and ImageNet pre-trained model. This study demonstrates the potential of a deep learning algorithm to improve the computer-assisted diagnostic in pathology and clinical care.

1 Introduction

Image-based features of tissue cancers have an essential role in clinical prognostics. Specifically, tumor grade is strongly associated with survivorship, and metastasis is a very late cancer developing stage that needs attention. Methods for identifying patients who are likely to benefit from further molecular testing are required. The development of the digitalization of clinical slide samples produces an increasing need for helpful tools based on Artificial Intelligence (AI).

Among AI technologies, deep learning has demonstrated strong performance in many electronic image-recognition applications for its proven efficiency, consistency and accuracy [4]. The essence of deep learning is to build higher-level concepts on top of the lower ones via hierarchy of features. Automatically learning these general features enables the system to learn complex functions mapping an input to output without the need for hand-identified features. Significant advances in this area have begun to show promise for tumor detection, metastatic cancer detection [2]. However, the general limitation of deep learning algorithms is that models are inherently limited to the training dataset which sets restrictions on the application area. Therefore, we are comparing the pre-trained model adapted from ImageNet dataset to our four-layer CNN model.

Our findings demonstrate that the use of four-layer CNN to identify metastatic cancer patches focusing on regions of interest is very time-efficient. While the accuracy of trained model based on the pre-trained model from ImageNet is not competitive as expected, which might require further deeper training and parameter tuning. But that is not the main focus of this project. With appropriate design and implementation, we believe the benefits of an automate computational tool such as this could extend the application to other images of cancer tissues for which there is a similar need to accurate and efficient identify cancer risk.

2 Data Description

Image data is consisting of a large number of small pathology images 96*96p pixels. They are a modified version of the PatchCamelyon (PCam) benchmark dataset (Original PCam contains duplicate pictures due to its probabilistic sampling). A positive label indicates that the center 32*32px region of a patch includes at least one pixel of tumor tissue, tumor tissue in the outer area of the patch does not influence the label.

[*]Department of Statistics, Duke University (wenwen.ye@duke.edu)
[†]Department of Electrical and Computational Engineering, Duke University (zicheng.yuan@duke.edu)
[‡]Computational Biology and Bioinformatics Program, Duke University (xue.zou@duke.edu)

Table 1: Summary of input training images

Image	DESCRIPTION
Metastasis	classified by clinician (50%)
Non-Metastasis	classified by clinician (50%)

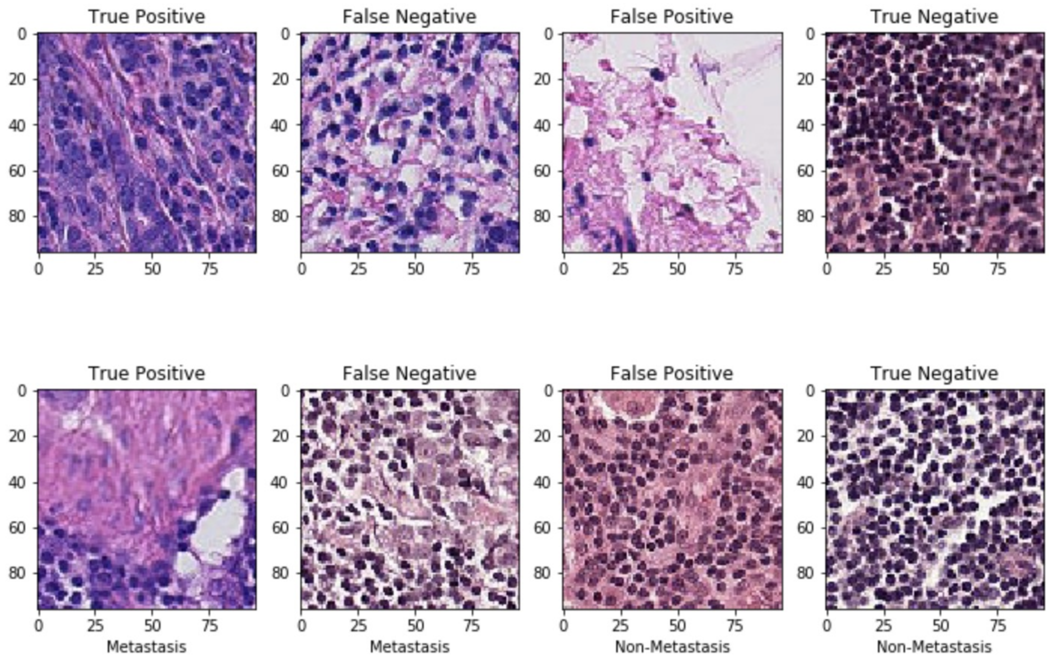

Figure 1: Two random selected examples for FP, FN, TP, TN, respectively

3 Method

3.1 CNN

Convolutional neural network (CNN) is the most popular neural network model used for feature identification and classification of the images. A typical CNN consists of multiple convolutional layers, ReLU layers, pooling layers as feature learning and fully connected layer and softmax as classification.

Convolutional Layer extracts features from the input image by carrying out convolution operation with a Kernel/Filter, which has the same depth of input image. The first convolutional layer is typically used to capture low-level features such as edges. The additional layers then adapt to high-level features, giving a network of the overall understanding of the images in the data set.

Rectified Linear Unit (ReLU) Layer enhances the training efficiency by only maintaining positive values of a convolved output and carrying those activated features into the next layer. The activation function is represented as:

$$f(x) = max(0, x)$$

Pooling Layer performs nonlinear down-sampling that reduces the spatial size of a convolved output. Specifically, Max Pooling partitions the convolved output into a set of non-overlapping regions and outputs the maximum of each region.

Fully Connected Layer performs high-level reasoning by learning a possibly non-linear function of the high-level features.

Softmax produces a vector containing the probabilities of an image being classified into each image class. The softmax functin is written as:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} for i = 1, ..., K and \mathbf{z} = (z_1, \ldots, z_K) \in R^K$$

where z is the input probability vector.

Adam Optimization is an extension to traditional stochastic gradient descent methods.It make use the moving average of first and second moments of the gradient to update weight as $w_t = w_{t-1} - \alpha \frac{\hat{m_t}}{\sqrt{\hat{v_t}} + \epsilon}$, $\hat{m_t} = \frac{m_t}{1-\beta_1^t}$,$\hat{v_t} = \frac{v_t}{1-\beta_2^t}$,where m_t and v_t are the updated biased first and second moment.

3.1.1 Four-layer CNN

Figure 2 below illustrates the basic structure of our four-layer CNN. Each layer consists of a 2-dimensional convolutional layer with ReLu as the activation function, followed by **batch normalization**, possibly one **maxpooling layer** and **dropout**.

The batch normalization layer normalizes the output of each layer before it enters next layer. It improves the robustness and effectiveness of learning process. The dropout layer randomly drops a proportion of connected nodes in previous layer. Dropout layer and batch normalization are our major methods to reduce overfitting.

Those structures together are designed to extract features in training images. At the end of the four layers, a flatten layer and several fully-connected layers are used to make final classifications.

The losses that we used to evaluate models is **binary cross-entropy loss**, which is defined as below:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i))$$

where N is the total number of images used in training, y_i is the true label of image i(positive=1, negative=0), and $p(y_i)$ is the softmax probability output by the model that represents how likely image i being classified as positive. Comparing to the simple metric of Classification Error, binary cross-entropy loss penalizes based on the confidence of that classification. In other words, binary cross-entropy penalizes the most for classifications that are wrong and have high softmax probability. A full-detailed list of all layers is in table4 attached at the end.

3.1.2 Retraining pre-trained ImageNet Model

ImageNet project aims to provide a large visual database for research purposes and there are powerful image classifiers trained on ImageNet. Retraining a pre-trained model having the advantages of taking pieces of a model already trained on the related task and effectively reusing it in a new model. Since many lower level features are largely same throughout the image data, the trained weights could be directly utilized which speeds up the training process[3].

Among various pre-trained models, the Inception V3 is selected for a retraining, since the Inception V3 model contains fewer parameters while able to achieve higher accuracy within fewer operations. The inception V3 model contains multiple layers which are divided into image feature extraction modules. The method here is to retrain the officially-defined seventh Mixed Layer. Also, the final fully connected layer, is also retrained.

Specifically, 40, 000 samples of each result type are randomly selected for retraining among the overall 220, 026 objects, while 4, 000 other samples of each result type are used to validate the model. The start learning rate is set to $1\mathrm{e}{-4}$ with a decay rate of 0.98 each 1587 steps. The batch size is 50 for each result type.

4 Results

4.1 Four-layer CNN

The accuracy and losses of 20 epochs in the training process are plotted in Figure 3. There is a clear slowdown in losses' decreasing during the last few epochs. Furthermore, the validation accuracy bounces around the training accuracy, but starts to show a slight tendency to decline. All of the above implies that further training with more epochs may have limited improvement in performance.

The best model we selected has the highest validation accuracy, which is 92.24%, and its losses is 0.2015.

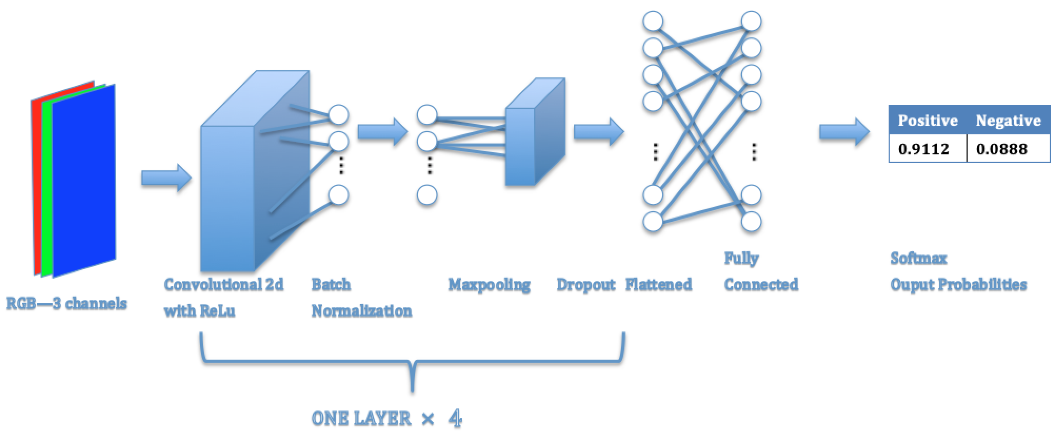

Figure 2: (4-layer CNN) An example CNN with multiple convolutional layers and pooling layers. A fully connected layer and the softmax decide the final classification of the image into one of several categories

4.2 Pre-trained ImageNet Model

The accuracy and losses of approximate 65,000 epochs in the retraining process are plotted in Figure 4. The accuracy implies that the growth rate is gradually decreasing and might convergence around 0.8. Moreover, the loss also ceases to decrease and will be stable around 0.18.

The accuracy after 65,000 epochs is 78.57%, and its losses is 0.1841.

5 Discussion

One potential method of improving accuracy is combine these two methods together. Build Custom layers upon pre-trained deep neural networks and continue training on our dataset may let us take the advantages of complexity

Another direction of improvement might be cropping the images. Since the label of images in initial dataset depends only on the central 32×32 pixels, it might be helpful if we crop the image with only the central part left, and train the model with the central part.

The result shows that the four-layer CNN model performs with a higher accuracy than the retrained model, which has two reasonable explanations. One reason is that the training data for retraining is not enough. For the deep network, in order to obtain high accuracy, large volume of data is required, which results in a time-consuming training procedure that typically required computation on GPU. The other assumption is that the result are only classified into two types, causing the advantages of pre-trained model could not be fully unitized. If the result type could be more various, the accuracy of retraining pre-trained model might be better than the implemented four-layer CNN model.

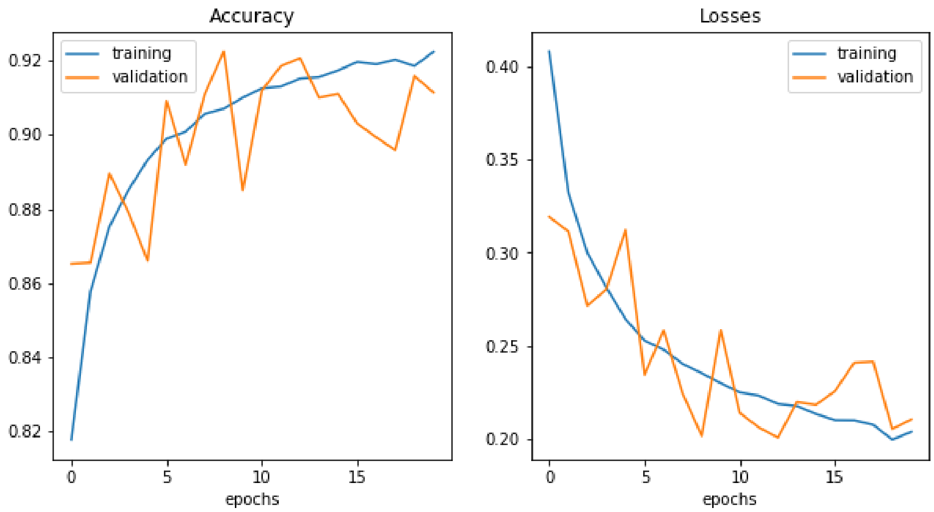

Figure 3: (4-layer CNN) Accuracy and Losses of 20 epochs

Accuracy	Losses
0.9224	0.2015

Table 2: (4-layer CNN) Accuracy and Loss of the Best model

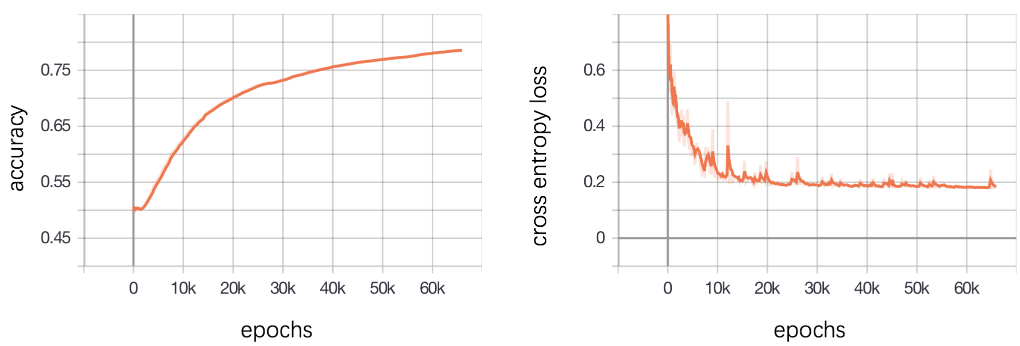

Figure 4: (ImageNet pre-trained model) Accuracy and loss of 65,000 epochs

Accuracy	Losses
0.7857	0.1841

Table 3: (ImageNet pre-trained model) Accuracy and Loss of the retrained model

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None 94 94 32)	896
batch_normalization_13 (Batc	(None 94 94 32)	128
conv2d_10 (Conv2D)	(None 92 92 32)	9248
batch_normalization_14 (Batc	(None 92 92 32)	128
max_pooling2d_5 (MaxPooling2	(None 46 46 32)	0
dropout_11 (Dropout)	(None 46 46 32)	0
conv2d_11 (Conv2D)	(None 44 44 64)	18496
batch_normalization_15 (Batc	(None 44 44 64)	256
dropout_12 (Dropout)	(None 44 44 64)	0
conv2d_12 (Conv2D)	(None 42 42 128)	73856
batch_normalization_16 (Batc	(None 42 42 128)	512
max_pooling2d_6 (MaxPooling2	(None 21 21 128)	0
dropout_13 (Dropout)	(None 21 21 128)	0
flatten_3 (Flatten)	(None 56448)	0
dense_7 (Dense)	(None 512)	28901888
batch_normalization_17 (Batc	(None 512)	2048
dropout_14 (Dropout)	(None 512)	0
dense_8 (Dense)	(None 128)	65664
batch_normalization_18 (Batc	(None 128)	512
dropout_15 (Dropout)	(None 128)	0
dense_9 (Dense)	(None 2)	258

Table 4: (4-layer CNN) Full-detailed list of all layers

Acknowledgments

We appreciate the help from Professor Sayan Mukherjee and all TAs from STA561D probability machine learning course!

References

[1] Steiner, F.D., et al., Impact of Deep Learning Assistance on the Histopathologic Review of Lymph Nodes for Metastatic Breast Cancer. Am J Surg Pathol, 2018.

[2] Lu, Y., et al., Identification of Metastatic Lymph Nodes in MR Imaging with Faster Region-Based Convolutional Neural Networks. Cancer Res, 2018. 78(17): p. 5135-5143.

[3] Hoong, P. (2017, July 22). Explore and Have Fun with TensorFlow: Transfer Learning. Retrieved fromhttps://www.slideshare.net/kuanhoong/explore-and-have-fun-with-tensorflow-transfer-learning

[4] Bi, W.L., et al., Artificial Intelligence in Cancer Imaging: Clinical Challenges and Applications. CA CANCER J CLIN, 2019