Deep Learning Applications: Detecting Metastatic Tissue in Lymph Node H&E Stains

Jude Lascano
lascanoj@bc.edu

November 30, 2023

Abstract

Hematoxylin and eosin (H&E) staining has historically provided great morphology of tissue samples; from a single H&E slide, histologists can extract information about the presence of cancerous cells, among other health abnormalities. Previous work has demonstrated that deep neural networks can prove helpful in medical classification contexts. This paper examines how deep learning models can assist in the classification of H&E stains. This paper examines the PatchCamelyon (PCam) dataset, a set of ∼300,000 H&E stained lymph node scans by testing various architectures. Regular convolutions, residual networks, and transformers were used to analyze how different deep learning models can be applied to medical image classification problems. These models were tasked with learning whether or not cancer was present in a lymph node H&E stain. This paper reasserts that deep neural nets have the capacity to learn such information in focused medical contexts. This is followed by a discussion of the results and improvements to current algorithms to improve accuracy, such as test-time augmentation. This paper concludes with the findings' potential impact on data analysis in medical imaging fields.

1 Introduction

The hematoxylin and eosin (H&E) stain is the principal stain used in histology. It cited as the "golden standard" due to the detailed morphology of the stained tissue samples [7]. Cell nuclei are stained purple and the extracellular matrices are stained pink. Other cell structures are typically stained with a variety of lighter pink tones. This coloration allows histologists to have a clear visualization of cells within a tissue sample.

Due to this level of detail, the H&E stain is also a critical component in assessing whether cancer is present in an organ. Since the growth of cancer cells is associated with the movement of cells into abnormal regions and rapid mitotic division, H&E stains allow pathologists to clearly see the state of cell nuclei and the potential deterioration of cell structures.

However, diagnosis via H&E stain is not without flaw; the human labor required is cited as one of the major struggles with H&E staining [2]. Due to a focus on maximizing inter-observer agreement, many pathologists follow simple decision trees that classify stains. In spite of this, discordance among pathologists still persists. To amend this, specialists have employed the use of deep learning algorithms in order to gain extra information about a stain. [2].

1.1 Previous Work

Deep learning has become increasingly important in various medical imaging contexts. Several studies [1] [4] [6] have demonstrated that deep neural network (DNN) architectures have the capability to classify images with accuracy and thereby make diagnoses with similar accuracy to human specialists.

With respect to the histology of H&E stains, the BRACE project [11] has demonstrated that DNNs can outperform existing techniques in breast cancer diagnosis. BRACE shows that with DNNs, an AI can identify tumor morphology, mitotic activity, and the aggressiveness of a tumor [11].

1.2 My Contribution

This paper proposes a simpler investigation into an effective architecture for H&E classification tasks. This paper will approach apply various models to a classification task; I will be evaluating (1) a self-defined DNN architecture, (2) residual networks, and (3) vision transformers.

2 Methodology

2.1 Data

The dataset used in this project is PatchCamelyon (PCam), a benchmark dataset of 327,680 images of H&E stained lymph node tissues. This dataset was originally created by Veeling et al. [10] and is a derivation of Camelyon16 [4]. PCam is divided into the following: 262,144 of the stains are for training, 32,768 will be used for validation, and 32,768 will be used for testing.

Each image is a 96 * 96px RGB image annotated with a binary label that indicates if cancer is present or not. A positive label (i.e. = 1) indicates that within the 32 * 32 center region, at least one pixel is considered within the cancerous region. Tumors outside of this area do not impact the label. Likewise, a negative label (i.e. = 0) indicates that there is no cancer in the center region. There is a 50/50 split between positive and negative labels across the entire dataset.

The images are all subregions of the greater whole-slide image of the H&E stain. Additionally, there is no overlap in H&E stains; for all images, there are no two images that share a common region of pixels.

2.2 Training Settings and Parameters

This project primarily focuses on a classification problem, therefore cross entropy loss was used as the optimizer. Due to the sheer size of the dataset and the limitations of Google CoLab, each model was trained on one epoch (one full run of the training set).

The trainloader class written for this paper also had a variable batch size. Since I wanted to maximize CoLab's capabilites, the trainloader class was instantiated with a

varying number of workers and batch size. Generally, the project aimed to maximize GPU memory without crashing the program. Shuffle was also enabled. For the models for which transfer learning is relevant, the trainloader class was also modified to load images with the necessary preprocessing steps that were specified by those models' documentation.

Furthermore, the for all training instances, learning rate was set to 0.01 and momentum was set to 0.9. These values were set based on previous experiments in class, which demonstrated that they provide a stable baseline for healthy loss.

2.3 Models Used

2.3.1 MyNet

The first model that was utilized was a neural network designed by myself. For simplicity, this paper will refer to this model as "MyNet." This model's architecture uses five convolutional layers, followed by four linear layers. Between each later, the ReLU activation function was used, and between each convolution, 2D batch normalization was used. The `nn.Flatten` function was also used to transition between the last convolutional layer and the first linear layer.

2.3.2 Residual Networks

The next models this paper evaluates are the ResNet18 and ResNet50 models, both of which are pretrained models made by He et al. [5]. Residual networks implement skip connections, which directly address the limitations of backpropagation and the Vanishing/Exploding Gradient Problem. Notably, He et al.'s residual network architecture was the most accurate algorithm in evaluating the ImageNet 2015 dataset. Their ResNet model won 1st place in a competition to produce a model that could accurately categorize 1000 objects from an input image [5].

Therefore, we can reasonably apply transfer learning for the pretrained models and adapt them to PCam's task, since both ImageNet and PCam focus on image classification. For both ResNet models, they were loaded in using the pretrained weights. This project fine-tunes[1] this model to the PCam dataset and adjusts the output vector to be our two categories. Images going into ResNet18 and ResNet50 were preprocessesd accordingly; this paper opted to have the images match the input tensor by adding padding.

2.3.3 Transformers

The last model we are evaluating is ViT_L_16, a pretrained model created by Dosovitskiy et al. [3]. This paper will refer to it as "ViT" for simplicity. This model is built off on the attention mechanism and transformer architecture proposed by Vaswani et al. [9] and adapts it to computer vision tasks. ViT has been proven to excel at various image recognition

[1] In this paper, "train" will be used synonymously with "fine-tune," but do note that this project did not train a residual network or transformer model from the ground up.

benchmarks (ImageNet, CIFAR-100, VTAB, etc.). Similar to the ResNets, this project used the pretrained default weights and fine-tuned ViT to work with PCam's data and output. This paper preprocessed images as specified by Vaswani et al., meaning that images were resized and then cropped [9].

2.4 Evaluation

The models trained on PCam are tasked with classifying the lymph node stain into two categories: metastatic or non-metastatic. Accuracy in classifying novel H&E stains is the metric by which all models are evaluated. All models produce an output of size 1 * 1 * 1. If the number in this output is closer to 0, then the model has deduced that the H&E stain has no metastatic tissue present. If this number is closer to 1, then the model has determined that there is metastatic tissue. Rounding to 0 or 1 (whichever is closer) will be used for evaluation. Model accuracy is then defined as follows:

$$\text{Accuracy} = \frac{\#\text{ of test set images correctly identified}}{\#\text{ of test set images}}$$

As a baseline, if a model was simply guessing, its score would be 0.5. Scores < 0.5 mean that the model has overfitted to the data and has failed to interpret novel data. Scores > 0.5 indicate that the model is able to successfully differentiate between metastatic and non-metastatic stains.

3 Results

Each model was successfully trained on the PCam dataset. This paper reports the loss graph, time taken, and final accuracy of the models. Each of the figures below shows the change in loss over time. The x axis is the current batch of the trainloader, and the y axis is the loss at a given batch. Notably, each model had a variable amount of batches due to the aforementioned limitations of CoLab.

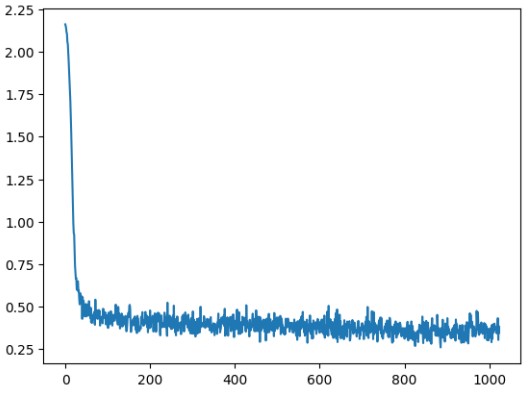
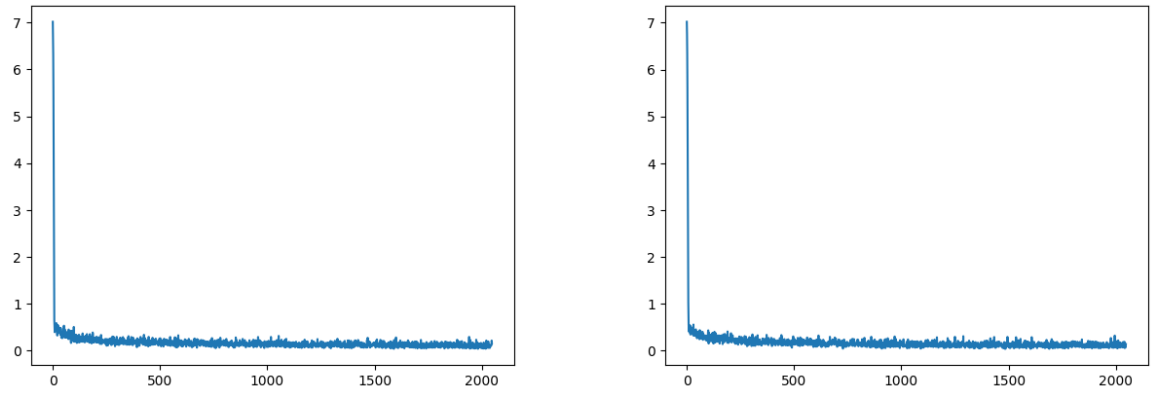


Figure 1: Loss graph for MyNet

Fig. 1 depicts the loss of MyNet. It took 26 minutes to train and its final accuracy ended up being 0.71775.



(a) ResNet18 loss graph

(b) ResNet50 loss graph

Figure 2: Loss graphs for ResNet models

Fig. 2 shows the loss graphs for each of the pretrained ResNet models that were used. ResNet18 took 29 minutes to train and had an accuracy of 0.85507. ResNet50 took 50 minutes to train and had an accuracy of 0.82369.

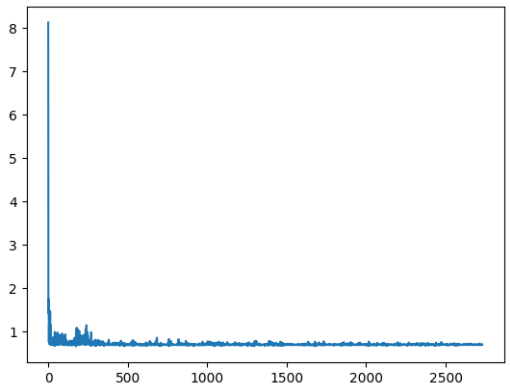


Figure 3: Loss graph for ViT

Fig. 3 is the loss graph for ViT. It ended up taking 129 minutes to train and had a final accuracy of 0.8468.

Table 1: Compiled Results

Architecture	Model name	Time to train (min)	Accuracy (out of 1)
CNNs	MyNet	29	0.717
Residual Networks	ResNet18	26	0.855
Residual Networks	ResNet50	50	0.823
Transformers	ViT_L_16	129	0.846

The table aggregates the numerical information of each model.

4 Discussion and Conclusions

4.1 Interpretation of Results

All models achieved accuracies greater than 0.5, demonstrating that each model was able to learn information from the PCam dataset and apply it to novel information. Moreover, this proves that the "detect cancer" task is indeed learnable. Expectedly, the more complex and intensive architectures, ResNet50 and ViT, took longer to train.

MyNet's performance demonstrates that complex architectures are not necessarily required for a DNN to be successful at this task, though the performance of the other models does reveal that alternate DNN architectures are conducive for better results. Expectedly, the ResNet models and ViT outperformed MyNet by a notable margin. For the ResNet models, the results show that they performed similarly to each other. Chiefly, this similarity shows that an increase in layers does not necessarily correlate to an increase learning. This is especially true, given that ResNet's architecture directly addresses the issues with many-layered DNNs that do not address backpropagation's flaws.

One notable observation is the steady decrease in loss. Even at the end of training, each model's loss graph was still decreasing towards 0. This means that all models still had the capability to learn more about PCam. In turn, this means that if given more epochs, these models could achieve a higher accuracy score.

However, one discrepancy that was not expected was the similar performance between ResNets and ViT. I expected ViT to outperform the ResNets, since Dosovitskiy et al. demonstrated how ViT beats ResNets in various classification benchmarks [3]. But, the results show that given one epoch, these models achieve a similar accuracy. I believe this is primarily due to the simplicity of the task. Since these pretrained models are tasked with making a binary classification, this paper does not fully take advantage of the full classification abilities of each architecture. If the task were to identify more information (such as the stage of the tumor, its type, etc.), then perhaps the ViT would be the most effective choice.

4.2 Sources of Error

One flaw of the project was the failure to utilize the validation set of PCam's images. When training, the use of the validation set was omitted and not used. I believe that if the validation set was taken advantage of in the training phase, each of the models may have seen a notable increase in accuracy.

The graphs are also inconsistent with each other. While the figures indeed demonstrate that each model was able to learn over time, each graph represents a slightly different trainloader class, as shown by the varying x axis lengths. This is due to CoLab's GPU restrictions, which forced the larger models (ResNet50 and ViT) to have a significantly lower batch size than ResNet18 and MyNet. This also means that the times to train are variable; however, this does not invalidate the time scores because each model was fully

taking advantage of CoLab's 15GB GPU VRAM limit. Thus, the times can also be thought of as the time needed to train given that 15GB of a GPU was being fully utilized.

4.3 Limitations

As mentioned, we trained only on one epoch. However, each graph does show that even at the end of this epoch, loss was still steadily decreasing. This suggests that all models still have more to learn about PCam, thus implying the models could have achieved a higher accuracy given more epochs. Training on additional epochs was not done out of the interest of time.

4.4 Improvements

As mentioned, some simple additions to this project would be to use a consistent batch size for more comparable graphs, to take advantage of the validation set, and to add extra epochs for better accuracy. Additionally, one could run multiple trainings for the same model to get a mean accuracy, which would better inform us of an architecture's expected accuracy. For example, given ResNet18, one run multiple instances of training and testing. Afterwards, one could take the average of all those accuracies to get a more precise accuracy; this notably means an individual model would likely run slightly better or worse than that final mean accuracy.

One technique that I believe would significantly contribute towards a higher accuracy is test-time augmentation (TTA). I say this because TTA has been proven to be an excellent strategy in improving image classification models [8]. Though this would increase the train time, this strategy of polling the model on augmentations of the same image can help the model better generalize the data. In the context of our data, the most appropriate augmentations would likely be horizontal flips, slight tilts, and slight shifts in any direction.

4.5 Generalization & Future Work

Thus far, this paper has demonstrated that pretrained models have the capability to perform adequately in an H&E-centric context, but the success here demonstrates that even the basic pretrained model can be applied in other various medical imaging contexts. Pathologists can use these pretrained models as the building blocks for future architectures. Some projects [11] [1] have experimented with using pretrained models as a step in their analysis of medical images and have found success. To this end, DNNs have great potential as tools for data analysis in both research and diagnostic contexts.

References

[1] Bagheri Rajeoni, A., Pederson, B., Clair, D. G., Lessner, S. M., and Valafar, H. (2023). Automated Measurement of Vascular Calcification in Femoral Endarterectomy Patients Using Deep Learning. *Diagnostics*, 13(21).

[2] Djuric, U., Zadeh, G., Aldape, K., and Diamandis, P. (2017). Precision histology: how deep learning is poised to revitalize histomorphology for personalized cancer care. *npj Precision Oncology*, 1(1):22.

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

[4] Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J. A. W. M., , and the CAMELYON16 Consortium (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*, 318(22):2199–2210.

[5] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition.

[6] Mercan, C., Balkenhol, M., Salgado, R., Sherman, M., Vielh, P., Vreuls, W., Polónia, A., Horlings, H. M., Weichert, W., Carter, J. M., Bult, P., Christgen, M., Denkert, C., van de Vijver, K., Bokhorst, J.-M., van der Laak, J., and Ciompi, F. (2022). Deep learning for fully-automated nuclear pleomorphism scoring in breast cancer. *npj Breast Cancer*, 8(1):120.

[7] Rosai, J. (2007). Why microscopy will remain a cornerstone of surgical pathology. *Lab Invest*, 87(5):403–408.

[8] Shanmugam, D., Blalock, D., Balakrishnan, G., and Guttag, J. (2021). Better Aggregation in Test-Time Augmentation.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention Is All You Need.

[10] Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. (2018). Rotation Equivariant CNNs for Digital Pathology.

[11] Wahab, N., Toss, M., Miligy, I. M., Jahanifar, M., Atallah, N. M., Lu, W., Graham, S., Bilal, M., Bhalerao, A., Lashen, A. G., Makhlouf, S., Ibrahim, A. Y., Snead, D., Minhas, F., Raza, S. E. A., Rakha, E., and Rajpoot, N. (2023). AI-enabled routine H&E image based prognostic marker for early-stage luminal breast cancer. *npj Precision Oncology*, 7(1):122.