Deep Learning for Digital Pathology: Tumor Detection using the PCam Dataset

Simon Zou
Department of Electrical and Computer Engineering
University of Washington
Seattle, WA 98195
xyzou1@uw.edu

Yehchan Yoo
Department of Statistics
University of Washington
Seattle, WA 98195
yehchany@uw.edu

Abstract

Accurate identification of metastatic tissue is critical for cancer diagnosis, yet healthcare systems face a severe shortage of pathologists. To address this bottleneck, our paper focused on automating tumor detection in histopathological images using deep learning architectures. The study evaluated a custom convolutional neural network against a baseline equivariant model and a pre-trained residual network. While all models achieved high diagnostic accuracy and the fine-tuned residual network with data augmentation attained the highest overall accuracy, the custom model demonstrated vastly superior computational efficiency. Additionally, interpretability analyses showed that the custom model relied on highly localized, granular cellular structures to make its predictions, whereas the pre-trained networks focused on broader spatial features. These findings indicate that lightweight, interpretable, and purpose-built diagnostic tools can perform competitively with deeper networks, offering an efficient triage solution to alleviate clinical workloads without compromising patient safety.

1 Introduction

Accurate identification of metastatic tissue in lymph node histopathological scans is essential for cancer diagnosis. Unfortunately, various developed countries such as the United States, Canada, and Italy are suffering from a shortage of pathologists for tumor detection – leading to high workload for the remaining pathologists and the aging of the current pathologist workforce. In the United States alone, the number of active pathologists decreased by a little less than 20% from 2007 to 2017 while their overall workload increased by more than 40%.[1] This trend remains far worse in low- and middle-income countries with far less resources.[2]

Our project aims to address this issue by harnessing deep learning to automate tumor identification in histological image patches using the Patch Camelyon (PCam) dataset.[3] The use of artificial intelligence (AI) for tasks like tumor detection can help a lot in reducing the workload of the existing oncologists and allowing them to focus on more complex tasks.[1]

2 Literature Review

Srikantamurthy et al.[4] developed a hybrid deep learning model to classify breast cancer histopathology images into four benign and four malignant subtypes. Their approach utilized a Convolutional Neural Network (CNN) combined with a Long Short-Term Memory (LSTM) recurrent neural network. The CNN module leveraged transfer learning from pre-trained architectures, utilizing models like

ResNet50 to extract spatial features, which were then merged with the sequential processing power of the LSTM module for the final classification. This study is highly relevant to our PCam project because it highlights the effectiveness of using transfer learning and fine-tuning established models on histopathological data, which directly supports our methodology of deploying a fine-tuned ResNet-18 as a robust predictive baseline.

Fu et al.[5] proposed StoHisNet, an advanced hybrid model designed to categorize gastric pathology images into normal tissue and three distinct types of adenocarcinoma. To overcome the inherent limitations of standard convolution operations in capturing global context, they engineered a dual-channel network. This network uses an Xception-based module to extract local cellular features alongside a Swin Transformer block to learn global, long-range dependencies across the tissue. While our PCam project currently focuses on binary classification, StoHisNet's success in fusing different architectural strengths provides a strong theoretical foundation for our upcoming plan to ensemble our custom CNN with the ResNet-18 model, proving that combining unique feature-extraction techniques maximizes overall accuracy.

Dabeer et al.[6] demonstrated a straightforward, highly effective approach for the binary classification of breast cancer histopathology images into benign and malignant categories. They designed a custom Convolutional Neural Network built from scratch, utilizing sequential convolutional layers, max-pooling for subsampling to prevent overfitting, and fully connected layers for the final prediction. To reduce computational overhead, they heavily emphasized image preprocessing, specifically resizing the original high-resolution slides to smaller, manageable dimensions before training. This paper directly parallels our project's initial phase; it validates our strategy of building and evaluating a simple, custom VGG-style CNN to learn the spatial hierarchies of our 96×96 histological patches before we transitioned to heavier, pre-trained networks.

3 Dataset: Patch Camelyon (PCam) Dataset

For training, validation, and testing, our paper primarily uses the Patch Camelyon (PCam) dataset, which consists of 327,680 colored images of lymph node histopathological scans. Each image is accompanied by a binary label that indicates whether there is metastatic tissue in that image.[3] The dataset was made for Veeling et al.'s 2018 paper "Rotation Equivariant CNNs for Digital Pathology" from the Camelyon16 dataset to help validate the authors' own rotation equivariant CNN models.[7] In our paper, we will consider images with metastatic tissue as positive data points and as part of the "Tumor" class, while we will consider images without as negative data points and as part of the "Normal" class. Some sample images from the dataset can be seen in Figure 1.

The images are all 96 by 96 pixels in resolution, and are labeled to have tumor by the presence of a minimum of one pixel of tumor tissue in the center 32 by 32 pixel region for each image. Without such presence, an image is labeled to be normal. The dataset is split into three groups by the dataset creators: a training set of 262,144 images, a validation set of 32,768 images, and a test set of 32,768 images – all with a 50/50 balance between positive and negative examples.[8]

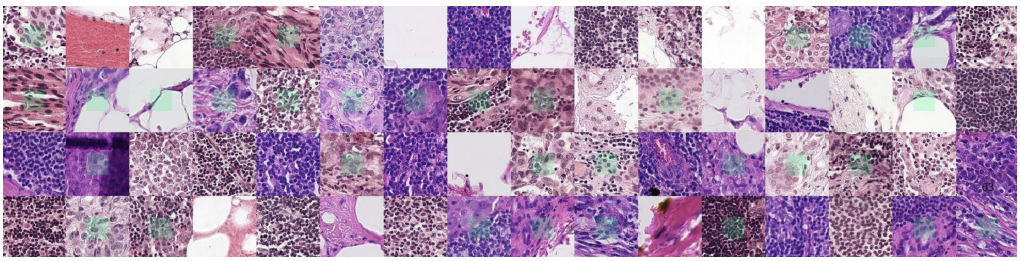

Figure 1: Sample images from Patch Camelyon, provided by the Grand Challenge platform.[3]

4 Methods

4.1 Models

4.1.1 Baseline Model: G-CNN

As noted previously, Veeling et al.[7] introduced the Patch Camelyon (PCam) dataset in their 2018 paper "Rotation Equivariant CNNs for Digital Pathology" to address the unique challenge that histological tissue samples lack a canonical orientation—meaning a tumor remains a tumor regardless of how the diagnostic slide is rotated. To exploit this geometric property, the authors proposed the use of Group Equivariant Convolutional Networks (G-CNNs), replacing standard convolutional layers with equivariant layers that mathematically guarantee consistent feature extraction across 90-degree rotations and reflections (the p4m symmetry group) without relying on massive data augmentation. This seminal work is the direct foundation of our project, as it provides the exact dataset we are analyzing.

Since the PCam dataset was created with the authors' G-CNN models in mind, we trained and tested a baseline G-CNN model so that we could compare our own models to this baseline model for comparison.

4.1.2 Custom CNN

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed to process data with a grid-like topology, such as images. Unlike traditional neural networks that treat input pixels as independent features, CNNs preserve the spatial relationship between pixels. This is achieved through convolutional layers, which "slides" small filters across the input image to detect local features like edges, textures, and curves. As the data passes deeper into the network, these simple features are combined to recognize increasingly complex structures, such as cell nuclei or tissue boundaries. Key components of a CNN include pooling layers and activation functions.[9] In medical imaging, CNNs are particularly powerful because they can automatically learn to identify disease specific biomarkers without the need for manual feature engineering.

We have our custom application of CNN, which utilizes a VGG-style architecture composed of three sequential convolutional blocks with 32, 64, and 128 filters, respectively, to progressively learn complex cellular patterns. Each block integrates batch normalization to stabilize training, ReLU activation for non-linearity, and max pooling to downsample the spatial footprint from 96×96 to 12×12. For binary classification, the network concludes with a fully connected head featuring a 512-neuron layer. This final stage incorporates a 0.5 dropout rate to prevent overfitting, ultimately outputting a single probability score to determine the presence of a tumor. If the probability score is 0.5 or more for an image, then that image is predicted to contain tumor.

From here, we adjusted the number of epochs, the batch size, and the learning rate of our custom CNN model to potentially change the model as follows:

- Raising the number of epochs should give the model more opportunities to learn from the entire dataset, while lowering it should prevent the model from overfitting.
- Raising the batch size lets the model process more data at once for faster training steps but demands more memory, whereas lowering it updates the model's weights more frequently and can help it generalize better.
- The learning rate dictates how drastically the model updates those weights after each step; a high rate learns quickly but might overshoot the best solution, while a low rate learns slowly but precisely.
  - The later iterations of our custom CNN model did not use fixed learning rates. Rather, we used the `ReduceLROnPlateau` scheduler, which acts as a smart monitor that automatically lowers our model's learning rate whenever the model performance stops

improving — allowing it to take smaller, more careful steps to fine-tune its way to the optimal solution.[10]

Furthermore, we implemented data augmentation for the final iteration of our custom CNN model. Data augmentation artificially expands the training dataset by applying random geometric transformations and color jittering. This forces the model to learn the actual structural features of a tumor rather than memorizing the orientation or specific chemical dye lighting of the training slides.[11]

Our data augmentation randomly applied the following transformations to the training set (though not the test or the validation set):

- Horizontal flip
- Vertical flip
- Rotation by up to 90 degrees either direction (clockwise or counterclockwise)
- Brightness changed up or down by up to 20%
- Contrast changed up or down by up to 20%
- Saturation changed up or down by up to 20%
- Hue changed up or down by up to 20%

Eventually, we ended up with the following working trials of our custom CNN model:

1. A batch size of 64, 10 epochs, and a fixed learning rate of 0.001
2. A batch size of 32, 20 epochs, and a fixed learning rate of 0.001
3. A batch size of 64, 10 epochs, and a learning rate scheduler (specifically, `ReduceLROnPlateau`)
4. A batch size of 64, 10 epochs, a learning rate scheduler (specifically, `ReduceLROnPlateau`), and robust data augmentation

It should be noted that all of our custom CNN models were trained on our M3 MacBook Pro and did not require the use of any computing cluster.

4.1.3 ResNet-18

In addition to the baseline and our custom CNN models, we were curious on how fine-tuning an already pre-trained general-purpose computer vision model would work out for cancer detection. So, we also decided to fine-tune a pre-trained ResNet-18 model on the PCam dataset to see how the fine-tuned model would perform. ResNet-18 model comes from the ResNet family of models, which were developed and released by researchers at Microsoft Research in their 2015 paper "Deep Residual Learning for Image Recognition". ResNet-18 is the smallest model in the ResNet family and is an 18-layer convolutional neural network model that utilizes residual learning, which involves approximating a residual function for the underlying mapping function with identity mappings and is visualized in Figure 2. ResNet models were introduced as a framework to improve the performance and ease the training of deep neural networks.[12]

From the ResNet family of models, we picked the ResNet-18 model, as we found this model to be the most appropriate for the limited computing resources we had for this paper. The ResNet-18 model was fine-tuned and tested on Kaggle with its Nvidia P100 GPU using the training set from the PCam dataset; the base ResNet-18 model was imported from the PyTorch library.[13]

For the ResNet-18 model, we trained two versions of this model: one without data augmentation and one with data augmentation, mentioned in Section 4.1.2. Each version of the model also went through hyperparameter tuning using the Bayesian optimization process from the `optuna` library.[14] The hyperparameter tuning process was set to select a learning rate between 10^{-5} and 10^{-2}; a batch size of 32, 64, or 128; and either the Adam or the stochastic gradient descent (SGD) optimizer.

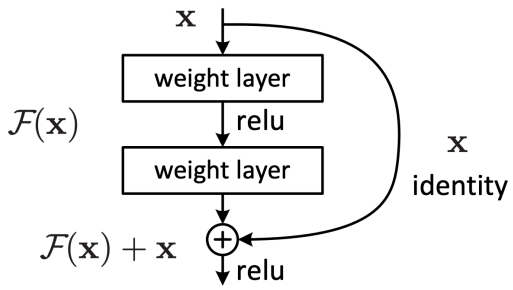

Figure 2: Residual learning: a building block; image from the 2018 paper "Deep Residual Learning for Image Recognition". [12]

4.2 Interpretability

For our paper, we did not just focus on the general accuracy and performance of our models. We also focused on the interpretability of our models. While complicated models like ours can make accurate predictions on complex image data, such models also can be lacking in interpretability, meaning that the users of these models may struggle to understand how these models come to make the predictions they make. [15] So, we employed two methods for checking the interpretability of our models: Shapley Additive Explanations (SHAP) and Gradient-weighted Class Activation Mapping (GradCAM).

4.2.1 SHAP

To systematically evaluate the interpretability of our models, we first utilized SHAP (SHapley Additive exPlanations), a unified and game theory-based framework for interpreting model predictions. SHAP works by assigning an importance value to each feature for a particular prediction. Based on cooperative game theory, SHAP unifies six existing additive feature attribution methods to specifically attribute the change in the expected model prediction to each feature when conditioning on that feature. [15] It should be noted that, for our tasks, we treated each pixel as a feature — especially since every image in the PCam dataset is 96 by 96 pixels in dimension.

4.2.2 GradCAM

Additionally, we also utilized Gradient-weighted Class Activation Mapping (GradCAM) to evaluate the interpretability of our models. GradCAM is an interpretability technique that makes CNN models more transparent by generating visual heatmaps to highlight the image regions most influential to a model's specific prediction. By using the gradients of a target concept flowing into the final convolutional layer, it produces a localization map that identifies the discriminative features the model used to reach its conclusion. [16]

We selected SHAP and GradCAM, as these two methods of interpretability complement each other by balancing computational efficiency with interpretive depth, particularly in high-stakes fields like healthcare. While Grad-CAM offers fast, spatially oriented explanations that provide an intuitive "at-a-glance" look at broader regions of interest, SHAP provides a more granular and mathematically rigorous attribution of individual input features. [17] This synergy could allow pathologists to use GradCAM to quickly localize suspicious regions within medical imaging and then employ SHAP for a fine-grained analysis of the specific textural or structural features that triggered the classification, ensuring that AI-driven clinical decisions are both efficient and transparent.

5 Results

5.1 Performance

After training and tuning the models, we assessed the performance of the models on the test set from the PCam data by looking at the confusion matrix from each model and by using the following metrics:

- Accuracy is the proportion of total correct predictions out of all predictions made.[18]
- AUC-ROC (Area under the Receiver Operating Characteristic Curve) is the probability that – when given a positive and a negative data point – the model will correctly predict the positive data point to be of a higher probability value (of being positive) than the negative data point. The perfect model would have an AUC-ROC of 1.0. A random guess model would have an AUC-ROC of 0.5.[19]
- Recall, precision, and F1 score were also calculated separately for data points in the tumor class and data points in the normal class.
  - Recall is the the ratio of correctly predicted instances of a class to the total actual instances of that class.
  - Precision is the ratio of correctly predicted instances of a class to the total instances predicted to be in that class.
  - F1 score is the harmonic mean of precision and recall, calculated as $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.[18]

As noted previously in Section 3, we will consider images with metastatic tissue as positive data points and as part of the “Tumor" class, while we will consider images without as negative data points and as part of the “Normal" class.

5.1.1 Baseline Model

The baseline Group Equivariant CNN (G-CNN) achieved an overall accuracy of 85.86% and an AUC-ROC of 0.9301. As seen in the confusion matrix, the model demonstrated strong predictive balance with a Tumor precision of 0.90 and a Normal recall of 0.91. This robust performance is driven by the architecture’s multi-directional filtering and group pooling, which mathematically guarantee consistent predictions regardless of how the tissue patch is rotated or mirrored. Comparatively, this baseline model outperformed our custom CNN by inherently leveraging these spatial symmetries from the start, though its overall performance ultimately fell slightly short of the deeper, pre-trained ResNet-18 architecture. Figure 3 shows the confusion matrix for G-CNN.

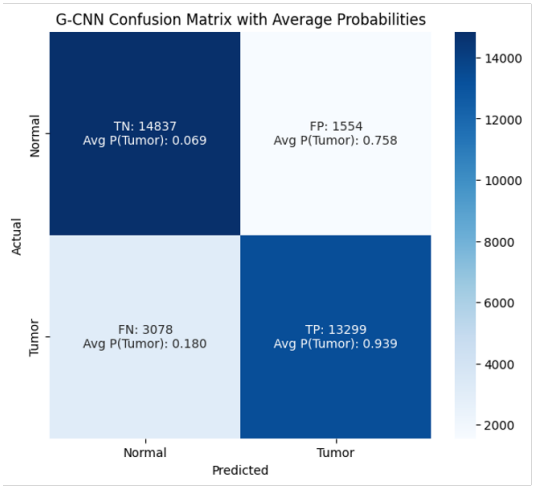

Figure 3: Confusion matrix for the baseline G-CNN model

5.1.2 Custom CNN

The initial trials of the custom CNN model showed steady performance, though the first trial exhibited a notable bias toward identifying normal tissue over tumors (Figure 4). While the second trial offered marginal improvements in AUC-ROC and F1 scores, the classification distribution remained largely similar to the first iteration.

Performance improved noticeably in the third trial, which achieved a higher accuracy of 83.91% and a significantly improved F1 score of 84.00%. More importantly, this trial resolved previous biases, demonstrating a well-balanced ability to distinguish between normal and tumor classes as illustrated in Figure 6.

Table 1: Performance Comparison of Custom CNN Model Trials on the PCam Dataset

Trial	Accuracy	AUC-ROC	F1 Score
First Trial	0.8043	0.9055	0.8000
Second Trial	0.8074	0.9132	0.8100
Third Trial	0.8391	0.9148	0.8400

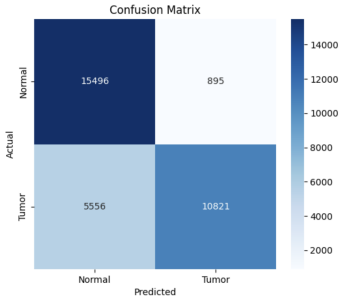

Figure 4: Confusion matrix for trial 1 of custom CNN model

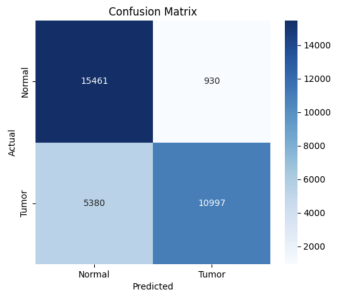

Figure 5: Confusion matrix for trial 2 of custom CNN model

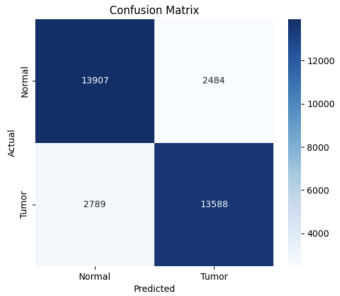

Figure 6: Confusion matrix for trial 3 of custom CNN Model

The fourth and final trial demonstrated the highest overall performance and robustness, reaching an accuracy of 85.49%, an AUC-ROC of 94.64%, and an F1 score of 85.50%. This peak result proves that the integration of geometric and color augmentations significantly enhanced the model's ability to generalize to unseen tissue variations. Figure 7 shows the confusion matrix for the final iteration of our custom CNN model.

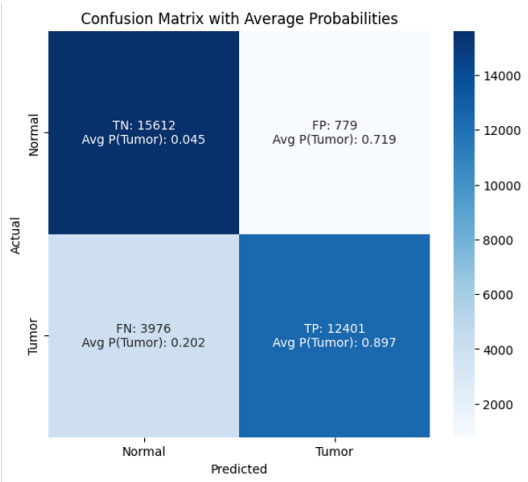

Figure 7: Confusion matrix for trial 4 of custom CNN model

5.1.3 ResNet-18

After tuning, the first of our ResNet-18 model without data augmentation was trained with a learning rate of approximately 0.00017, a batch size of 128, and the Adam optimizer. On the test set, our first ResNet-18 model achieved an accuracy of 84.10% and AUC-ROC of 94.18% — outperforming the first three trials of our custom CNN model but not its final iteration. The first of our ResNet-18 models also had a higher AUC-ROC but a lower accuracy than the baseline model. The confusion matrix for the first version of our ResNet-18 model can be found in Figure 8.

As for the second and final version of our ResNet-18 model with data augmentation, the model was trained with a learning rate of about 0.00286, a batch size of 32, and the SGD optimizer. On the test set, the second of our ResNet-18 models achieved a higher accuracy of 86.65% and a higher AUC-ROC of 94.41% — outperforming the baseline model and the first three versions of our custom CNN model. Our second ResNet-18 had a higher accuracy but a lower AUC-ROC than the final iteration of our custom CNN model. The higher learning rate indicates that our second ResNet-18 model required larger updates to its weights compared to our first ResNet-18 model [10], while the lower batch size indicates that our second ResNet-18 model utilized noisier updates for better generalization compared to the first ResNet-18 model [20]. The confusion matrix for the second version of our ResNet-18 model can be found in Figure 9.

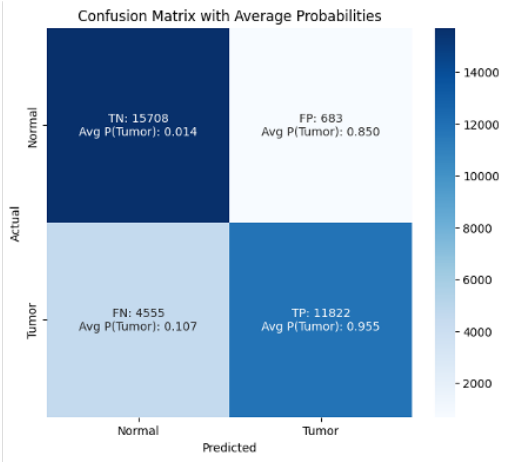

Figure 8: Confusion matrix for first ResNet-18 model without data augmentation

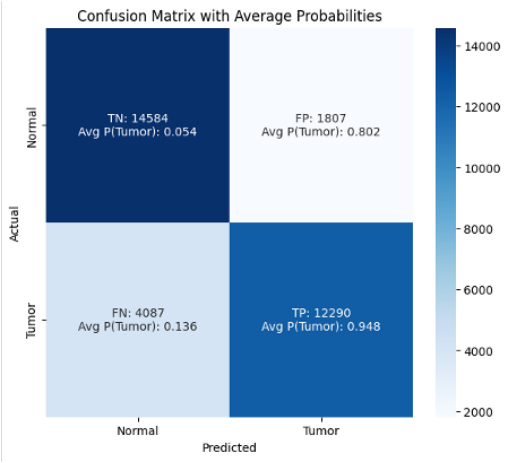

Figure 9: Confusion matrix for second ResNet-18 model with data augmentation

5.1.4 Model Performance Comparison

After training all our models, we complied our metrics in a table to compare the performance of the following models: the baseline G-CNN model, the custom CNN model (fourth trial), the first version of the ResNet-18 model, and the second version of the ResNet-18 model. The complied metrics can be found in Table 2.

Table 2: Model comparison

Model	Accuracy	ROC-AUC	Precision		Recall		F1 Score	
			Tumor	Normal	Tumor	Normal	Tumor	Normal
Baseline	0.8586	0.9301	0.90	0.83	0.81	0.91	0.85	0.86
Custom (Final)	0.8549	0.9464	0.94	0.80	0.76	0.95	0.84	0.87
ResNet-18 (Ver 1)	0.8401	0.9418	0.95	0.78	0.72	0.96	0.82	0.86
ResNet-18 (Ver 2)	0.8665	0.9441	0.93	0.82	0.79	0.94	0.86	0.88

Overall, looking at the table and comparing the confusion matrices, we found that the second ResNet-18 model achieves the highest accuracy and the highest F1 scores (for both classes). However, the second ResNet-18 model only achieves the second highest ROC-AUC score, the second highest recall values, the second highest precision for the Normal class, and the third highest precision for the Tumor class — all surpassed by the custom CNN model. As a whole, the second ResNet-18 model and the final version of the custom CNN model remain the top performing models from our paper.

The confusion matrices and the table also reveal distinct shifts in the False Positive (FP) and False Negative (FN) distributions across the models. The baseline model demonstrated the highest Tumor recall (0.81) among the listed models. This resulted in the lowest number of False Negatives (3,078) but a relatively high volume of False Positives (1,554). As for the custom CNN model, this model yielded higher Tumor precision (0.94) and Normal recall (0.95) compared to the baseline. This came at the cost of an increase in False Negatives (3,976), which brought the Tumor recall down to 0.76.

As for the first ResNet-18 model, this model maximized Tumor precision (0.95) and Normal recall (0.96). Figure 8 shows this aligns with the lowest FP count (683) but the highest FN count (4,555), resulting in the lowest overall Tumor recall (0.72). This model also displays the most polarized confidence, as it yields the highest average probability of tumor presence (that is, average P(Tumor)) for True Positives (0.955) and the lowest for True Negatives (0.014).

Introducing data augmentation improved the model's overall balance for the second version. Accuracy increased to 0.8665, and table metrics indicate Tumor recall improved to 0.79. Figure 9 reflects this shift with a decrease in FN (4,087) compared to the unaugmented model, though FPs increased to 1,807. The second ResNet-18 model also displays a less polarized confidence than the first ResNet-18 model, though more so than the final version of the custom CNN model.

5.2 Interpretability

We also were able to generate some SHAP and GradCAM interpretability plots for the final custom CNN model and the two versions of our ResNet-18 model.

First, we generated SHAP plots, which visualize the distribution of SHAP values for each pixel/feature. We analyzed feature importance using SHAP plots generated from the first five images of the test set. As shown in Figures 10, 11, and 12, the custom CNN model demonstrated a higher frequency of SHAP values near zero compared to both ResNet-18 variants. That is, we observed that the custom CNN highlighted fewer distinct spatial regions compared to the ResNet-18 models. We also found that, while the SHAP plots were fairly similar between the two ResNet-18 models, the regions prioritized by the custom CNN were largely unique from those prioritized by the ResNet-18 models. This divergence, combined with the custom model's higher density of SHAP values near zero, suggests that the custom CNN relies on a more sparse and distinct set of features for its classification decisions than the ResNet-18 architectures.

Next, we generated GradCAM heatmaps for these three models based on some randomly sampled images from the test set. Figures 13, 14, and 15 illustrate striking differences in how the three models analyze the same histopathology image to arrive at a correct "Tumor" prediction. Figure 13 shows that the custom CNN model produces a highly granular, fragmented heatmap — though less granular than the SHAP plots, as expected. It focuses on multiple small, distinct regions scattered across the tissue sample, suggesting the model relies on localized textural details or specific cellular structures.

Conversely, the ResNet-18 models in Figures 14 and 15 generate much smoother, broader activation maps, but with very different focal points. The first ResNet-18 model's heatmap in Figure 14 concentrates its attention entirely on a single large central structure. In stark contrast, the second ResNet-18 model's heatmap in 15 essentially ignores the center, heavily weighting the bottom two corners and the top left corner of the image instead. Based on observed patterns, this divergence indicates that — despite both ResNet versions arriving at the same correct prediction and having similar SHAP plots — they have still learned to focus on completely different macroscopic spatial features within the tissue to make their decisions in their final convolutional layers.

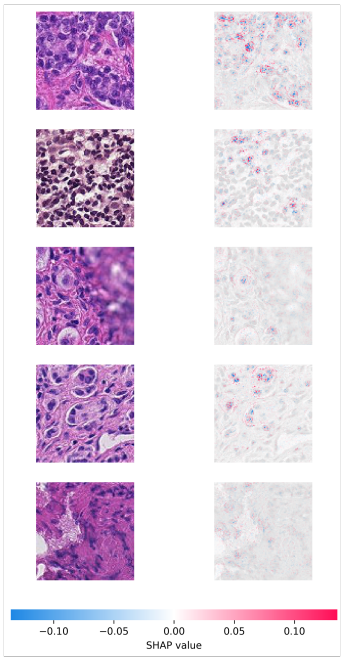

Figure 10: SHAP plots for trial 4 of custom CNN model

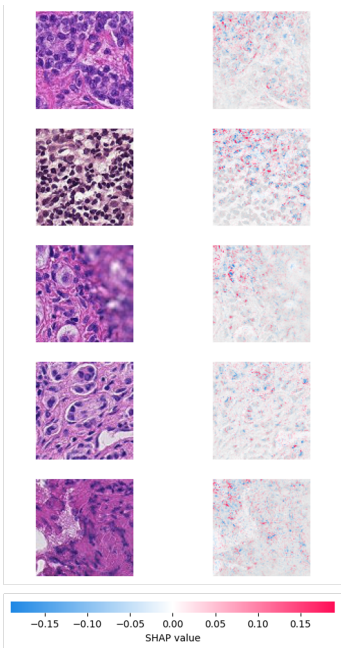

Figure 11: SHAP plots for version 1 of ResNet-18 model

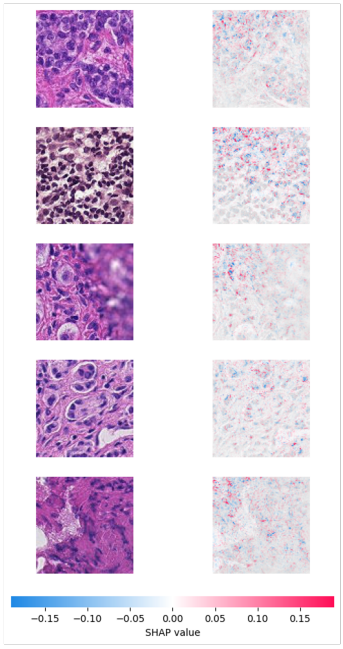

Figure 12: SHAP plots for version 2 of ResNet-18 model

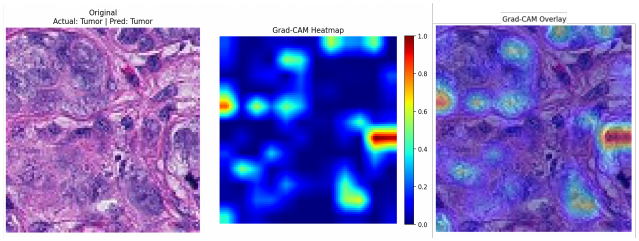

Figure 13: GradCAM heatmap for trial 4 of custom CNN model

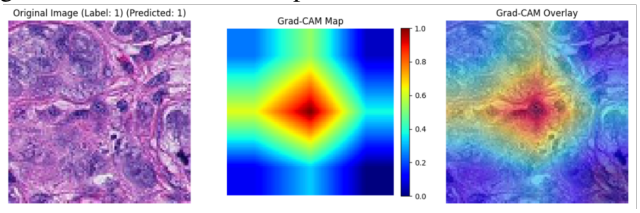

Figure 14: GradCAM heatmap for version 1 of ResNet-18 model

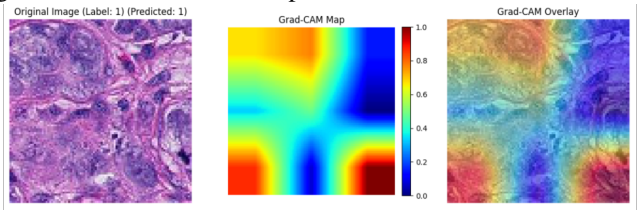

Figure 15: GradCAM heatmap for version 2 of ResNet-18 model

The difference in GradCAM heatmaps between the two ResNet-18 models was also found with another image from the test set, where the first ResNet-18 model correctly guesses the image to be positive/have tumor while the second model does not. Here, based on the heatmaps in Figures 16 and 17, it can still be seen that the first and second models focus on very different parts of the image and end up with a different prediction after the final convolutional layer.

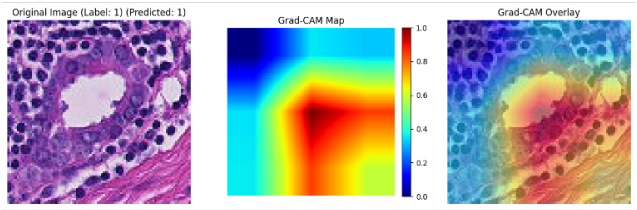

Figure 16: GradCAM heatmap for version 1 of ResNet-18 model on another image

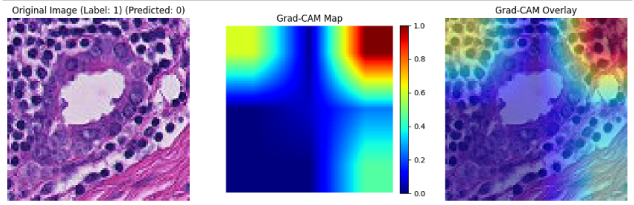

Figure 17: GradCAM heatmap for version 2 of ResNet-18 model on another image

6 Discussion

A critical advantage of our custom CNN architecture lies in its computational efficiency and rapid inference capabilities. While the fine-tuned ResNet-18 model achieved slightly higher overall accuracy, it required significant computational overhead, taking a little less than two hours to train on a Kaggle-hosted Nvidia P100 GPU. In stark contrast, our custom CNN converged in just over an hour using a local Apple M3 chip. This vastly superior training efficiency, coupled with much faster inference times, is a remarkable outcome; it demonstrates that a lightweight, purpose-built architecture can deliver diagnostic performance that closely rivals deeper, highly resource-intensive models. In a clinical setting, this reduced computational burden would translate directly to lower deployment costs and faster turnaround times for high-volume slide screening. Our custom CNN architecture also had the most interpretable SHAP plots as well, focusing on a small number of highlighted regions instead of many regions across the entirety of the images. Interestingly, we found that the first ResNet-18 model produced the most interpretable GradCAM heatmaps for diagnosis. Unlike the custom CNN's granular noise or the second ResNet-18's overly broad activations, this model highlighted a single, concise region of interest.

Building upon these promising results, future work will focus on expanding the network's diagnostic breadth and overall robustness. A primary objective is transitioning from simple binary detection to multi-class classification, enabling the model to not only flag malignancies but also differentiate between specific histopathological subtypes of tumors. Additionally, we plan to explore ensemble methodologies—strategically combining the rapid, localized feature-extraction of our custom CNN with the deep contextual learning of ResNet-18 and the mathematical rotational equivariance of the baseline G-CNN to maximize overall predictive accuracy and minimize false negatives. Finally, adapting and fine-tuning these architectures to identify metastatic tissue in other forms of cancer will be a crucial next step in validating the generalizability of our approach across the broader field of digital oncology.

7 Broader Impact

To safely integrate this model into clinical workflows as a high-speed triage system, we propose a risk-calibrated thresholding strategy. Because we believe that the cost of a false negative (missing a tumor) is significantly higher than a false positive (misdiagnosing a tumor), our proposed strategy would avoid the standard 0.5 classification cutoff in favor of a highly sensitive threshold. Any patch showing even a slight probability of malignancy (e.g., > 0.1) would trigger mandatory pathologist intervention, immediately flagging the slide and generating a GradCAM heatmap and a SHAP plot for human review. Conversely, high-confidence benign patches can be safely de-prioritized, allowing this asymmetric risk approach to act as an over-cautious assistant that alleviates the severe pathologist shortage without ever compromising patient safety.

Beyond the immediate clinical setting, the global deployment of such automated diagnostic systems carries profound societal implications, particularly for underdeveloped areas and low-to-middle-income countries. In these regions, the ratio of specialized pathologists to patients is critically low, frequently resulting in months-long delays in cancer diagnosis and treatment.[2] Deploying a lightweight, explainable AI triage system could profoundly democratize access to life-saving healthcare by empowering rural or underfunded clinics to process massive volumes of histological scans efficiently. By automatically filtering out explicitly benign cases and instantly elevating high-risk patients to the limited number of available human specialists, this technology can drastically reduce diagnostic bottlenecks. Ultimately, its global application has the potential to accelerate early-stage cancer interventions and actively bridge the severe healthcare disparities that currently exist between resource-rich and resource-constrained communities.

References

[1] Carbone FG. The Shrinking Workforce of Pathologists: Implications for Healthcare and Possible Solutions. *Pathologica* 2025; 117(4): 449–451. doi: 10.32074/1591-951x-n1156

[2] Bamodu OA, Chung CC. Cancer Care Disparities: Overcoming Barriers to Cancer Control in Low- and Middle-Income Countries. *JCO Global Oncology* 2024; 10(10). doi: 10.1200/go.23.00439

[3] Grand Challenge. PatchCamelyon - Grand Challenge. Accessed January 23, 2026. `https://patchcamelyon.grand-challenge.org/`.

[4] Srikantamurthy MM, Rallabandi VS, Dudekula DB, Natarajan S, Park J. Classification of benign and malignant subtypes of breast cancer histopathology imaging using hybrid CNN-LSTM based transfer learning. *BMC Medical Imaging* 2023; 23(1): 19. doi: 10.1186/s12880-023-00964-0

[5] Fu B, Zhang M, He J, Cao Y, Guo Y, Wang R. StoHisNet: A hybrid multi-classification model with CNN and Transformer for gastric pathology images. *Computer Methods and Programs in Biomedicine* 2022; 221: 106924. doi: 10.1016/j.cmpb.2022.106924

[6] Dabeer S, Khan MM, Islam S. Cancer diagnosis in histopathological image: CNN based approach. *Informatics in Medicine Unlocked* 2019; 16: 100231. doi: 10.1016/j.imu.2019.100231

[7] Veeling BS, Linmans J, Winkens J, Cohen T, Welling M. Rotation Equivariant CNNs for Digital Pathology. In: MICCAI '18. Springer-Verlag; 2018; Berlin, Heidelberg: 210–218. doi: 10.1007/978-3-030-00934-2_24

[8] Veeling B. PatchCamelyon (PCam). Published June 8, 2018. Accessed February 20, 2026. `https://github.com/basveeling/pcam`.

[9] GeeksforGeeks. Introduction to convolution neural network. Last updated February 17, 2026. Accessed February 19, 2026. `https://www.geeksforgeeks.org/machine-learning/introduction-convolution-neural-network/`.

[10] Amazon Web Services. Training parameters - Amazon Machine Learning. Accessed February 20, 2026. `https://docs.aws.amazon.com/machine-learning/latest/dg/training-parameters1.html`.

[11] Byrd J. The Ultimate Guide to Data Augmentation in Computer Vision. 2024. Published November 12, 2024. Accessed March 19, 2026. `https://encord.com/blog/data-augmentation-guide/`.

[12] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: ICCV '16. ; 2016: 770-778. doi:10.1109/CVPR.2016.90

[13] PyTorch. resnet18. Accessed February 19, 2026. `https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html`.

[14] Najib T. Hyperparameter Tuning Using Optuna. Published August 11, 2023. Accessed February 19, 2026. `https://medium.com/@taeefnajib/hyperparameter-tuning-using-optuna-c46d7b29a3e`.

[15] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: NIPS '17. Curran Associates Inc.; 2017; Red Hook, NY, USA: 4768–4777. doi: 10.5555/3295222.3295230.

[16] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: ICCV '17. ; 2017: 618-626. doi: 10.1109/ICCV.2017.74.

[17] Tempel F, Groos D, Ihlen EAF, Adde L, Strümke I. Choose your explanation: a comparison of SHAP and Grad-CAM in human activity recognition. *Applied Intelligence* 2025; 55(17). doi: 10.1007/s10489-025-06968-3

[18] Google Developers. Classification: Accuracy, recall, precision, and related metrics. Published January 12, 2026. Accessed March 20, 2026. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall.

[19] Google Developers. Classification: ROC Curve and AUC | Machine Learning Crash Course. Published January 12, 2026. Accessed March 20, 2026. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[20] Ultralytics. Batch Size. Accessed March 20, 2026. https://www.ultralytics.com/glossary/batch-size.