– Independent Work Report Spring 2023 –

Estimating Tumor Necrosis in Glioblastoma Histologies

Konstantin Howard
Advisor: Dr. Olga Troyanskaya

Abstract

Glioblastoma is an aggressive cancer in the brain capable of killing a victim within six months and with a 17% survival rate after two years [12]. Immediate diagnosis and rapid treatment is essential to survival. However, pathologists still diagnose and analyze tumors manually by studying microscopic slide images (histologies) from tumor biopsies, an inefficient and error-prone process. This project contributes to ongoing research efforts to automate histopathology with machine learning tools by attempting to build a Convolutional Neural Network to classify the percent tumor necrosis in glioblastoma tumor histologies. This metric of tumor analysis proved to be a difficult problem: our model achieved at best a 37% accuracy in ten-way classification by patch, but did well in estimating a range of necrosis percent values by considering its top predicted classes. Despite a lightweight design and minimal training, our model shows promise for future research in tumor necrosis detection and estimation with highly-labeled data.

1. Introduction

1.1. Problem Overview

Cancer constitutes the largest health threat of the modern world, accounting for 10 million deaths or every sixth death globally every year. The effort to vanquish cancer is many-fold, from prevention to treatments to diagnoses and more, animating enormous research efforts worldwide.

Around the same time that the scientific crusade against cancer took off, so too did the modern world of computing. From the earliest, biomedical research and computer science relied on each other: research was made easier with computational tools, powered by tremendous amounts of data, and in turn biomedical research presented novel applications and data for the implementation of

abstract methods. This burgeoning field is known as Precision Health and, as much as it inspires research today, it is creating the healthcare of tomorrow. Healthcare is becoming more accurate and effective by becoming less biased and more personalized to natural biological differences among people and populations.

Our project emerges in this context: the application of well-known machine learning tools to a novel, biomedical problem: the diagnosis and analysis of cancer from a tumor biopsy. When a tumor develops in an organ and cancer is possible, doctors will extract a sample of the tumorous tissue for examination. Pathologists study the extracted tissue as slide images under a microscope, stained with Hematoxlyn & Eosin to reveal internal cell structures, in a field known as histopathology (see Figures 1 & 2). Manual analysis of slide images yields a pathology report, containing not only a malignancy classification but also a variety of other important measures such as tumor grade, tumorous cell ratio, percent necrosis, and others. These measurements tell the doctor vital information about the strength and nature of the tumor. Crucially, this data is used to inform the treatment plan for that patient.

Unfortunately, histopathology as it is performed today is inefficient, prone to human error, and lacks standardization. Pathologists work in labs physically distant from doctors and patients, delaying diagnosis. Histopathology is a long and arduous task. It requires selecting an appropriate area of the tumor for examination and a discerning eye: cancer cells can appear in a variety of ways and often look little different from healthy cells. Inaccurate results can be actively harmful to the patient. In one study of Papanicolaou (pap) tests for cervical cancer, 45% of errors in pathology diagnoses carried some harm, ranging from unnecessary delays in treatment to increased morbidity [6]. Even while malignancy classification is largely correct, other measures such as percent tumor cell have been shown to be regularly mistaken. An experiment testing pathologists on their ability to calculate this measure, which involves counting tumorous and nontumorous cells, found that they on average misestimated by 2.0 categories or up to 20% [9]. Even secondary reviews by other pathologists, which are very common in practice, are not standardized nationally[6]. In the end, a pathology report is only as good as those who wrote it and can only be confirmed *post facto* by the

patient's future health.

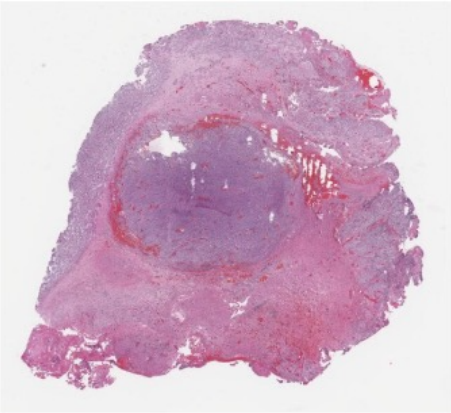

Figure 1: H&E stained histology of a glioblastoma tumor.

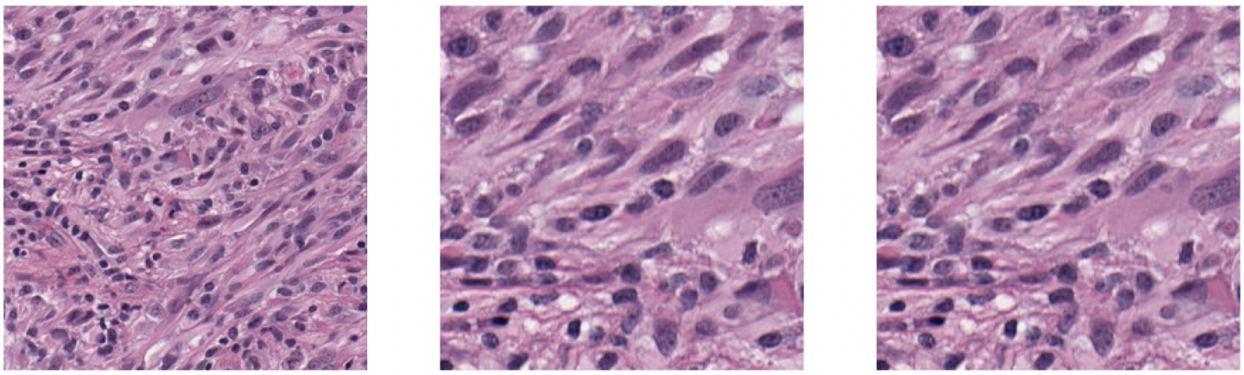

Figure 2: Section of histology at under varying magnification. Dark spots are tumor cell nuclei.

1.2. Neural Networks

The problem of histopathological analysis is ripe for the application of machine learning because it is at its core an image classification problem. The modern machine learning revolution has shown that computers can learn the features of and interpret images remarkably well, especially for images which can be difficult for the human eye to discern. The Convolutional Neural Network (CNN), a variety of the generalized Neural Network (NN), is the golden standard for image classification problems today. (Interestingly, the design of NNs was itself inspired by biomedical research into how neurons fire in the brain). A NN is composed of layers of neurons with multiplicative weights and activation functions propagating input data across layers to the output layer. They work well

despite zero prior knowledge about the problem or data. For CNNs, that input data represents an image. CNNs make use of convolution to increase the third dimension of a image and extract features that are important for classification while downsampling the size of the image. The number of classes for classification determines the length of the output layer and corresponds to assigning probabilities to output classes with the help of functions like sigmoid and softmax. In general, more classes makes for a more difficult classification problem.

1.3. Necrosis

One of the measures included in pathological reports is necrosis, whether detecting its presence in a tumor or quantifying the necrotic regions as a percentage of the total tumor area. There are two processes by which a cell can die: natural and healthy apoptosis and unnatural necrosis. Necrosis is brought on by health problems external to the cells affected, such as when they are starved of oxygen (hypoxia) and other nutrients like glucose that power their internal processes. Necrotic cells are marked by "compromised plasma membrane integrity, swelling of cellular organelles, random DNA degradation, and uncontrolled release of pro-inflammatory molecules" [13]. Not all necrosis is related to cancer; gangrene is also an instance of necrosis. Research into the causes of cell necrosis are ongoing, as many other factors can influence its prevalence in a tissue.

While it is little surprise that necrosis appears in tumorous tissue, its underlying relationship with tumors remains poorly understood. Rapid tumor growth surpasses the nutrient supply available and the tumor cells become necrotic as a result. Tumor necrosis is widely agreed to be indicative of poor health prognosis, tumor aggressiveness, size, stage, etc. for a variety of cancers including breast, lung, endometrial, and glioblastoma [13]. Still undecided, though, is the circular question of "whether necrosis is an epiphenomenon accompanying tumor progression or a direct cause of tumor aggressiveness" [13]. Regardless, necrosis is an important measure for pathologists to get right and for doctors to understand for treatment. Beyond prognosis and a gauge of severity, tumor necrosis can help determine treatment strategies, like surgical removal of necrotic regions, or serve as a binding target for drug therapies [5].

1.4. Glioblastoma

Glioblastoma (GBM) is rapid, aggressive brain tumor. The most common among brain tumors, it affects 3.21 out 100,000 people [12]. It often develops *de novo*, that is, having not spread from elsewhere in the body [12]. Originating in the frontal and temporal lobes of the brain, it can also spread into neighboring tissues [12]. Diagnosis of glioblastoma involves first locating the tumor via MRI or CT scans and then taking a biopsy for neuropathologists to examine. Surgical removal, or craniotomy, is the most common method of treatment, along with radiation and chemotherapy [12]. Treatment for glioblastoma is made difficult for a variety of reasons. The brain is a delicate and complicated place for a tumor, poorly accessible to drug therapies, fragile to more aggressive treatments, and disconnected from tumors elsewhere in the body that might be more easily treatable [12]. Even successful treatment may only prolong survival. Without treatment, a patient can die within sixth months. Survival rates after one and two years are a shocking 40% and 17%, respectively [12].

Notably, necrosis is a common feature of glioblastoma as it is a high grade tumor [10]. Necrotic regions appears pinkish in histologies and without the dark spots that tend to populate the rest of the slide. Whereas the dark spots are tumor cell nuclei, the nuclei of necrotic cells fade due to DNA degradation. Another visible marker of a necrotic region is its border with viable tumor cells. Such regions exhibit a palisading effect in which elongated tumor cells are arranged perpendicular to the necrotic region and parallel to each other. See 3 and 4 for an example analysis of necrosis in a glioblastoma histology.

1.5. Project Overview

With a little background, even an untrained eye can begin to recognize the difference between tumorous and necrotic regions in histologies. This project sought to build a CNN to perform that very task and estimate the percent tumor necrosis for glioblastoma histologies. Slide images were divided into smaller, magnified patches and labeled with in classes of 10%. Due to this simplification and sparsely-labeled histologies, the model was not very successful in accurately predicting the

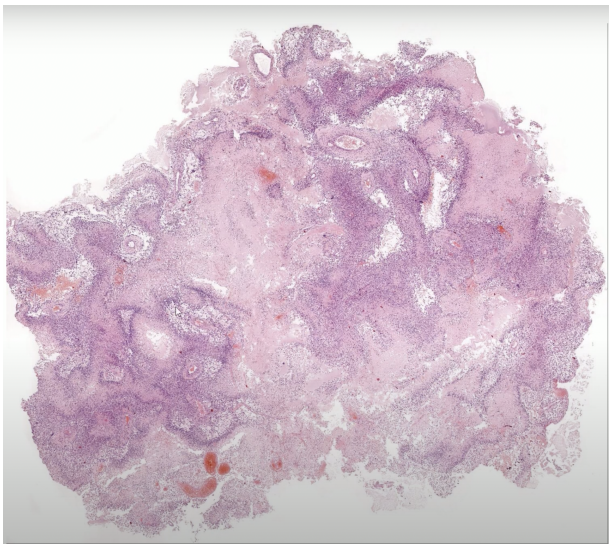

Figure 3: Tumorous and necrotic regions are visible even in this unmagnified glioblastoma histology.

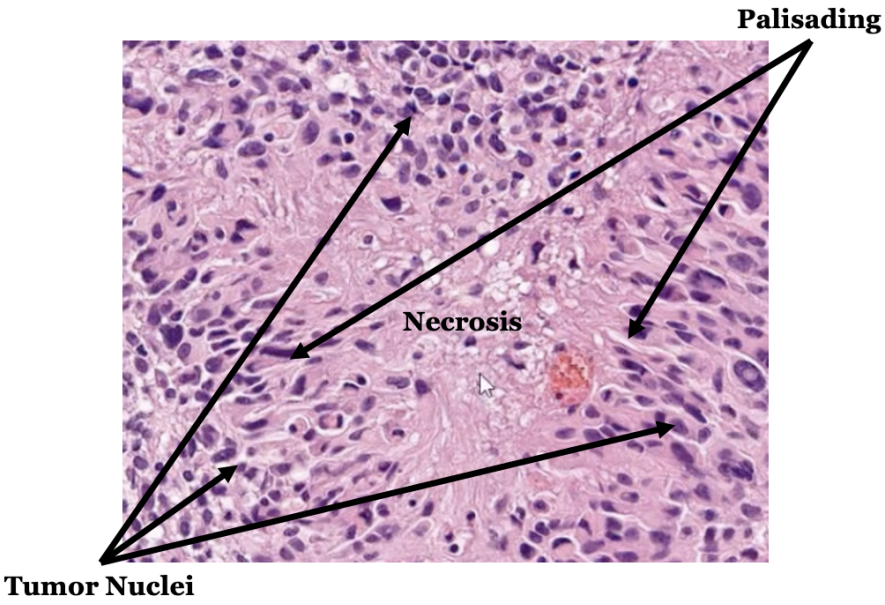

Figure 4: Labeled necrotic region from the above glioblastoma histology.

correct class of necrosis percent, whether validated patch-to-patch or patch-to-whole slide. The best performance was achieved by lowering the learning rate and extending training duration. However, the performance was effective enough (better than random guessing among 10 classes) to indicate that future research in the same vein is viable and ought to be pursued with the lessons learned from this project.

2. Background

Lots of research in recent years has sought to use CNNs for the classification problems pathologists encounter when analyzing histologies. Widespread approval and adoption of such tools is still far off, though, and histopathological analysis remains largely manual to this day. The extent to which digitization exists in the world of pathology today is limited to the interface through which pathologists view slide images, which only in 2017 acquired FDA approval to replace physical glass slides.

The most success in this domain has been achieved with the most important classification: whether a tumor is malignant or benign. In a 2019 paper (by no means the first of its kind), researchers developed a CNN for breast cancer malignancy classification and achieved an impressive 99.86% accuracy for the true class [4]. Their model increased the third dimension of the input images via two convolutional layers, the last of which contained 128 channels; the relatively lightweight design became effective with an intensive 500 epochs of training [4]. Other research with breast cancer extended beyond malignancy classification to the problem of classifying the eight subordinate types of breast cancer. The multiclass CNN achieved 93.2% accuracy [7]. These and other papers like it demonstrate the viability of using CNNs in histopathology and that they will someday become common tools for pathologists to use. Indeed, my student collaborators (see Acknowledgements) were able to replicate such research with even simpler CNN architectures and less training for lung and uterine cancers.

Using CNNs to classify other measurements in histopathology have proven to be more complicated but still successful. Oftentimes, they are estimation, not classification, problems and the models require more highly processed and labeled data to learn on. One example is the tumor cell ratio, that is the ratio of tumorous to healthy cells in a histology, which has obvious implications as to the severity of the tumor. Calculating the tumor cell ratio requires counting every cell in the slide and then classifying each as tumorous or not. In 2021, researchers at NEC Labs developed a system of models to do just that, combining a U-net CNN architecture for image segmentation with a binary

classifier to estimate tumor cell ratio [3]. The combined model achieved a mean absolute error of just 6%, compared to an average of 20% for human pathologists [3]. This paper demonstrated that even complicated measures like tumor cell ratio can be estimated with machine learning tools.

Research attempts to classify tumor necrosis have also found some success, especially in solving the detection problem (that is, a binary classification problem) of identifying the presence of necrosis in a slide image or patch. One paper studying osteosarcoma (bone cancer) trained several different machine learning models on labeled patches of viable tumor, necrotic, and non-tumor tissue to produce predicted tumor maps for test histologies [2]. They were able to achieve a patch-to-patch accuracy of 93.3% [2]. A similar paper sought to identify necrosis in canine soft tissue sarcoma. Their model trained on highly-labeled data, where each patch of a whole slide histology containing at least 30% necrosis according to on-hand veterinary pathologists was labeled as necrotic [8]. They used a pre-trained DenseNet161 CNN model to predict tumor maps of necrotic regions in the whole slide image [8] with a patch-to-patch accuracy of 92.7% [8]. Note that the data was labeled down to the patch level, while available data sets tend to provide labels just for whole slide histology images.

As for research with glioblastoma, one paper successfully classified the grade of glioblastoma histologies, a ternary classification problem, with 96.5% accuracy [14]. This they achieved with a data labeling scheme that patched whole slide histologies, choosing patches with suitable content (nuclei) for input [14]. This meant that individual patch labels were extrapolated from the whole slide histology label. Their CNN model featured an intensive 7 convolutional layers and dropout layers for regularization [14].

These papers and the wealth of research in this domain demonstrate the successes and difficulties of using CNNs to classify necrosis and inspired our project. Key lessons include using statistical simplifications to compensate for sparsely labeled data, the success of simple CNN architectures, the availability of histology data and CNN tools, the difficulty of multiclass classification problems, and that many different cancers, including glioblastoma, have potential for research. Critically, it signaled the need for more research into tumor necrosis and the percentage measurement.

3. Approach

3.1. Model

This project attempted to design a CNN model to classify the percent tumor necrosis of glioblastoma histologies. In contrast to some complex, extensively trained models in previous research, our model was extremely lightweight. It was built with simple PyTorch functions and featured two convolutional layers and three fully-connected layers. Training did not exceed 20 epochs and the number of channels in the convolutional layers reached 64 at most. The output (classification) layer was ten-way, a simplification of percentage measurements made to reduce the number of classes in the model.

3.2. Data

The data used was sparsely labeled. Each whole slide histology, segmented into its constituent patches, was mapped to a single percent tumor necrosis and then reduced to labels of 10% increments. Necrotic regions were not identified in the data set and so the CNN had to learn the appropriate label without knowing what necrosis actually looked like as input. This simplification was necessary given the labeling available. Notably, it mimicked the simplification used in [14] as mentioned previously.

3.3. Glioblastoma

Research into glioblastoma and brain tumors has not yet focused on tumor necrosis. Given that it is such a common and prevalent feature of glioblastoma and the need of rapid diagnosis for such an aggressive cancer, it was the cancer chosen for our data.

The over 90% accuracy rates in the cited background work were never to be expected; rather, we sought to explore how well a lightweight model could classify histologies and the feasibility of estimating percent tumor necrosis directly from whole slide histologies without the labeling of necrotic regions in the dataset.

4. Implementation

The project made use of the following tools and software:

- Jupyter Notebooks with Python 3
- PyTorch library for building the CNN model
- NumPy library for data handling
- SciKit Learn library for evaluating model performance
- The Clinical Proteomic Tumor Analysis Consortium Glioblastoma Multiforme Collection (CPTAC-GBM) via the National Institutes of Health (NIH) The Cancer Imaging Archive (TCIA) [1]
- Aspera Connect for high-speed downloads from TCIA
- py-wsi open-source library for SVS file image handling [11]
- kaggle for hosting large datasets and GPU

4.1. Data Processing

Data was downloaded manually from TCIA due to the limitations of computer storage and in order to curate the distribution of labels in the dataset. For the final training and test runs, a total of 255 whole slide histologies were downloaded. The train and test sets consisted of 205 and 50 slide images, respectively. All histologies came from biopsies of the frontal and/or temporal lobes of the brain, where glioblastoma always originates before spreading elsewhere. Histologies and many kinds of medical images are often stored as SVS files, meaning they were created by an Aperio ScanScope slide scanner. This file type stores images at varying magnifications all at once, allowing for manipulation without loss of clarity. The magnification levels can be visualized as a pyramid of constituent tiles at each level, as in Figure 5.

The py-wsi library facilitated viewing and working with SVS files. Just as a human looking at an entire histology at once cannot determine much about it, so to do CNNs require sections of whole slide histologies to be interpreted. py-wsi offers functions for sampling and storing these smaller sections, known as patches, as PNG files. It was decided that 256 x 256 pixel patches would be used and that the magnification level be 13 out of 17. The level was chosen qualitatively as the

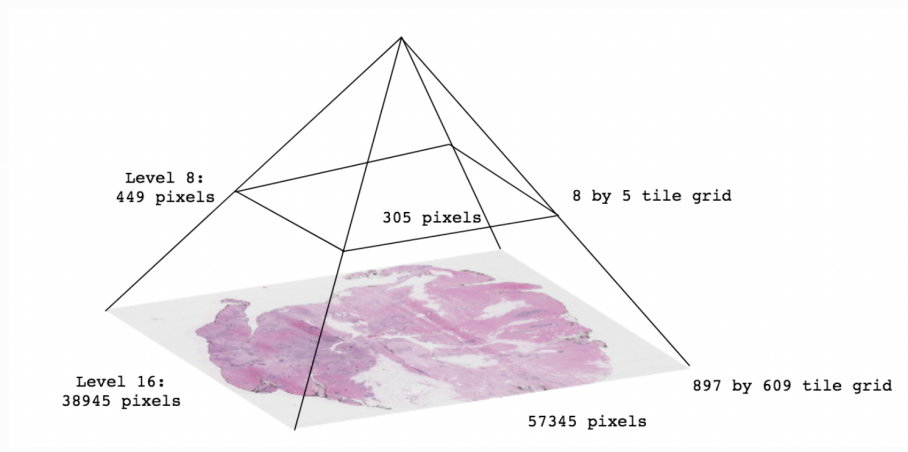

Figure 5: Example whole slide histology stored as SVS file.

level best interpretable for human vision: nuclei and internal structures do not appear fuzzy, etc. No overlap was used in patch sampling to prevent overcounting of regions. Thus, the first step in the data pipeline converted the 255 slide images into thousands of constituent patches like those in Figure 6.

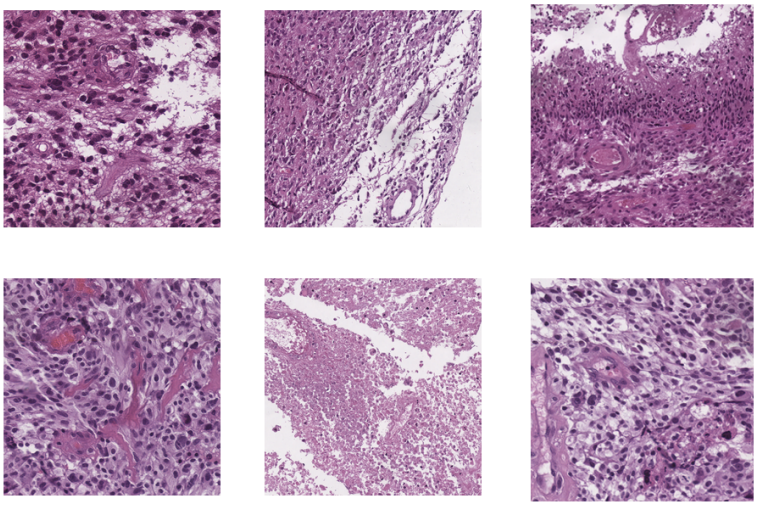

Figure 6: Example patches at level 13 magnification.

As visible in the above example patches, many patches turn out to contain some whitespace from edges and background the original histology slide. These are not helpful for the CNN to train or test on and had to be filtered. There are several methods for filtering whitespace, but this step turned out to be a bottleneck when processing all 255 slides. The algorithm used and described in 1 is executed efficiently with NumPy functions and the resulting patches were qualitatively suitable.

Algorithm 1: Whitespace Filter

```
for patch do
    w ← 0 ;
    for px in patch do
        m ← min_RGB px;
        if m > 200 then
            w ← w + 1;
        end
    end
    if w / 256^2 > 0.35 then
        Remove(patch);
    end
end
```

The maximum allowed whiteness of the minimum RGB value is 200 and the ratio of "white" pixels to all pixels in the image must be at most 0.35. After filtering, there remained 13,088 patches for training and 2,540 patches for testing.

4.2. Data Labeling and Distribution

The entire TCIA database contained far more slide images than needed for training and testing. More importantly, the distribution of percent tumor necrosis values was very uneven. As a result, the training and test sets were carefully curated to balance the need for a large quantity of data and an even distribution of labels. Curation was done at the slide image level, as the number of patches produced per histology after whitespace filtering was not consistent. Table 1 explains the labeling scheme and distribution used in the end. The bottom row totals the slides in each column and the constituent patches after whitespace filtering.

The paucity of highly necrotic slides might have lead to bias such that the model was better at predicting lower necrosis labels than higher labels. However, the latent distribution in the database also indicates that such high percent tumor necrosis values are far less common in reality than the middle and lower end values. To supply this labeled data to the CNN, a custom PyTorch Dataset class was overwritten to map each patch to its original whole slide histology label. TCIA provided a cohort CSV file with slide IDs and percent tumor necrosis values, then rounded to necrosis labels, that was used in the Dataset class to provide labels to patches as they were retrieved in the CNN.

Label	TCIA Total	Train Set	Train Set
0: 0-9%	241	44	11
1: 10-19%	118	42	9
2: 20-29%	54	45	9
3: 30-39%	34	27	7
4: 40-49%	15	12	3
5: 50-59%	16	13	3
6: 60-69%	10	8	2
7: 70-79%	11	9	2
8: 80-89%	3	2	1
9: 90-100%	4	3	1
Total	505	205 (13088)	50 (2540)

Table 1: Data Label Distribution

Patch sampling with py-wsi extended the slide IDs to patch ID naming scheme. With a dataset curated for label distribution and filtered for whitespace, the following CNN model could be trained and tested.

4.3. CNN Architecture

4.3.1. Design The CNN model developed was a customized PyTorch ConvNet. All layers and functions were implemented with basic PyTorch functions. The design is summarized in Table 2. Designing CNNs always involves extensive hyperparameter tuning and ambiguity as to how many and what layers to include. Since our goal was a lightweight model, the CNN used only two convolutional layers and three fully-connected layers. Hyperparameters were tuned with a smaller training set of 30 slides and validation set of 16 slides while in development. Batch size was kept constant at 16. Smaller batch sizes are associated with lower losses and better performance at the cost of longer runtimes. Kernel size for both convolutional layers was found to be best at 5 x 5 with 2 pixels of padding. The convolutional layers extend third dimension of the image array from 3 channels on input (the three RGB values 0-255) to 32 channels and then to 64 channels. This stage of a CNN is known as feature extraction. Just as pixels in the original patch image are represented by three channels – red, green, and blue – CNNs learn other features associated with an image across each convolutional layer with the growth of channel count and assign weights to

those features. While convolutional layers extend the third dimension of the image, downsampling with 2D max pooling layers reduce the height and width of the image. Hyperparameter tuning determined a kernel size of 2 x 2 and a stride of 2 for the pooling layers. Activation functions are used in all kinds of NNs to introduce nonlinearity. Without them, a NN is one long series of reducible matrix multiplication operations. As is common for CNNs, our model used the Rectified Linear Unit (ReLU) function after each convolutional layer, followed by max pooling. The ReLu function returns the input if it is positive, or else it returns zero 1. By remaining linear for half the input domain, ReLU preserves the ease of optimizing with gradient methods.

$$ReLU(x) = max(x, 0) \tag{1}$$

After the convolutional layers, the model flattens the image to a 1D array of length 64x64x64 (the final height and width after downsampling and the final number of output channels are all 64). The fully connected-layers function like a general NN and reduce that array to 512, then to 100 (recall that it is a percentage that the CNN is approximating), and then to 10 for the final output layer. The output layer corresponds to the 10 labels of percent tumor necrosis. Selecting a prediction from these 10 output values is akin to a probability distribution for the most likely classification. The model used the softmax function to actually convert the output layer to a probability distribution 2.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

In the above softmax formula, $\vec{z}$ represents the output layer as a vector and K as the length of that vector. Thus, for any output vector, it reduces it to a probability distribution from which the model selects the highest probability class as its prediction.

While experimenting with layers and parameters, it became clear that the model was overfitting to the training data: training accuracy was very high while validation accuracy was very low. To combat this perennial problem in machine learning, dropout layers were added. Dropout layers are a method of introducing regularization to a NN during training. They randomly drop values as they

propagate from one layer to the next with probability p. Our model used two dropout layers with probability $p = 0.2$ located after each convolutional plus max pooling layer.

Layer Type	Channels	Output Size	Details	Activation
Input	3	256 x 256	-	-
Conv 1	32	256 x 256	Kernel: 5, Pad: 2, Stride: 1	RELU
Max Pool 1	32	128 x 128	Kernel: 2, Stride: 2	-
Dropout 1	32	128 x 128	$p = 0.2$	-
Conv 2	64	128 x 128	Kernel: 5, Pad: 2, Stride: 1	RELU
Max Pool 2	32	64 x 64	Kernel: 2, Stride: 2	-
Dropout 2	32	64 x 64	$p = 0.2$	-
Fully Connected 1	1	512	-	-
Fully Connected 2	1	100	-	-
Fully Connected 3	1	10	(Output Layer)	softmax

Table 2: CNN Architecture Specifications

4.3.2. Training Training the model used all 205 slide images as 13,088 individual patches. Initializing the model with random weights, a PyTorch Trainer was used to implement the backpropagation algorithm. The loss function was Cross Entropy, the most popular function for training machine learning models. The optimizer used was Adam. Adam (not an acronym) is a relatively recent innovation, building on typical stochastic gradient descent. It is favored in deep learning models such as this one. Our study collected models trained with different learning rates, ranging from 0.001 to 0.000001, and for different durations, either 10 or 20 epochs. This produced eight parameterizations of the same model for evaluation. Figure 7 visualizes the training loss over epochs for each learning rate used. As evident in the graphs, the loss for learning rates of 0.001 and 0.0001 quickly plateaued. However, increasing the duration of training meaningfully decreased the final loss for a learning rate of 0.00001 from 0.754 to 0.115.

5. Results

5.1. Evaluation

There were two methods of evaluating performance: patch-to-patch and patch-to-whole slide. The model learned by evaluating its performance on the true labels for patches. For the purposes of

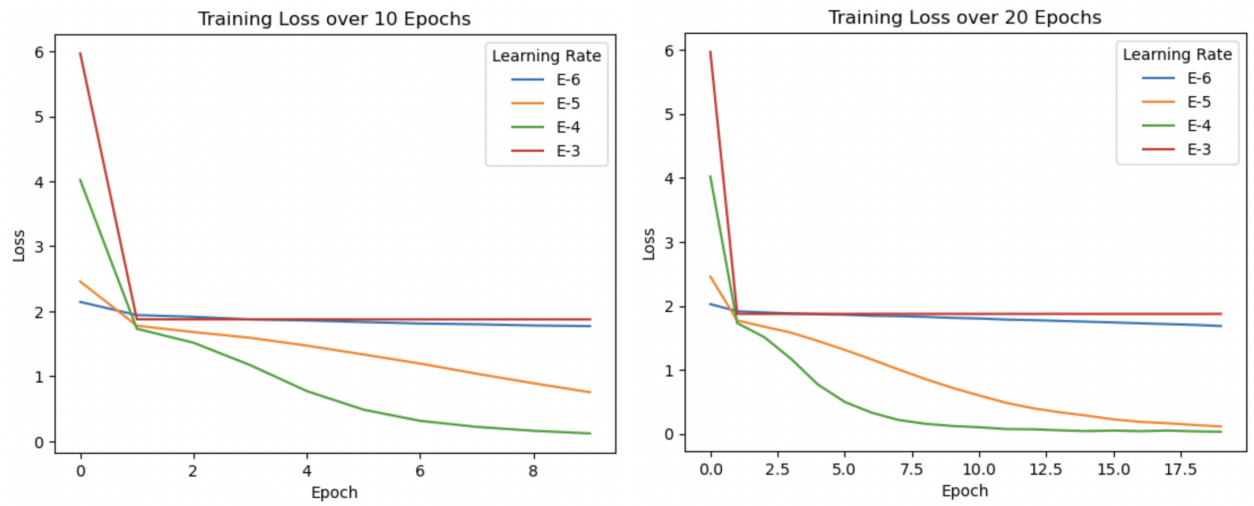

Figure 7: Training loss by learning rate on 10 and 20 epochs.

the project, though, we sought to estimate the percent tumor necrosis for an entire slide image. A number of metrics were used to evaluate performance by either method, namely accuracy, balanced accuracy, F1 score, Area-Under-Curve score, Receiver Operating Characteristic Area-Under-Curve score, and Top-k Accuracy. Given the uneven distribution of labels in the datasets, balanced accuracy was essential because it weighs the accuracy relative to the quantity of each class that appears. The Area-Under-Curve score for multiclass classifiers has two methods for determining true and false classes: one-versus-one and one-versus-all. These two methods always yielded the same values and are not differentiated in our Results. Top-k accuracy was the most insightful metric. It computes how often the true label was among the top k predicted labels in the softmax output layer. Because our model provided estimates very roughly – indeed it was trained on approximations – this metric was more applicable. Note that it cannot be used for patch-to-whole slide evaluation because the model produces no probability distribution for whole slide label prediction. Rather, it averages the patch label of every constituent patch to determine the whole slide's predicted label. A steeper plot of top-k accuracies for k classes means better performance. For the AUC score, the baseline is 0.5, as random guessing will yield just as many true positives as false positives. This is visualized as line $y = x$ when plotting the area under the curve. For all scores, higher values are better. Note that the baseline for evaluation is random guessing among the ten classes, or an accuracy of 10%.

5.2. Findings

The results for all eight parameterizations by patch and slide are summarized in Tables 3 and 4. Figure 8 graphs the top-k accuracy by patch for each model within its training class. Figures 9, 10, and 11 plot the area under the curve for each class and model. The area under the curve is not plotted for the models with learning rate 0.001 as they do not differ from random guessing.

Metric	**0.001**	**0.0001**	**0.00001**	**0.000001**
—	—	*By Patch*	—	—
Final Loss	1.874	0.120	0.754	1.771
Accuracy	28.4%	29.39%	32.59%	34.69%
Balanced Accuracy	10%	15.09%	15.03%	12.96%
F1 Score	0.044	0.143	0.129	0.108
AUC Score	0.5	0.713	0.663	0.588
ROC AUC Score	0.5	0.531	0.532	0.52
—	—	*By Slide*	—	—
Accuracy	23.91%	15.97%	17.39%	21.74%
Balanced Accuracy	10%	13.33%	8.89%	10.91%
F1 Score	0.039	0.1	0.051	0.063
AUC Score	0.5	0.519	0.495	0.506

Table 3: Results Summary for 10-Epoch Models

Metric	**0.001**	**0.0001**	**0.00001**	**0.000001**
—	—	*By Patch*	—	—
Final Loss	1.875	0.032	0.115	1.685
Accuracy	28.43%	31.21%	32.99%	36.87%
Balanced Accuracy	10%	15.57%	16.2%	16%
F1 Score	0.044	0.162	0.166	0.128
AUC Score	0.5	0.687	0.684	0.681
ROC AUC Score	0.5	0.534	0.537	0.539
—	—	*By Slide*	—	—
Accuracy	23.91%	28.26%	21.74%	23.92%
Balanced Accuracy	10%	14.8%	11.11%	12.02%
F1 Score	0.039	0.12	0.078	0.092
AUC Score	0.5	0.531	0.509	0.514

Table 4: Results Summary for 20-Epoch Models

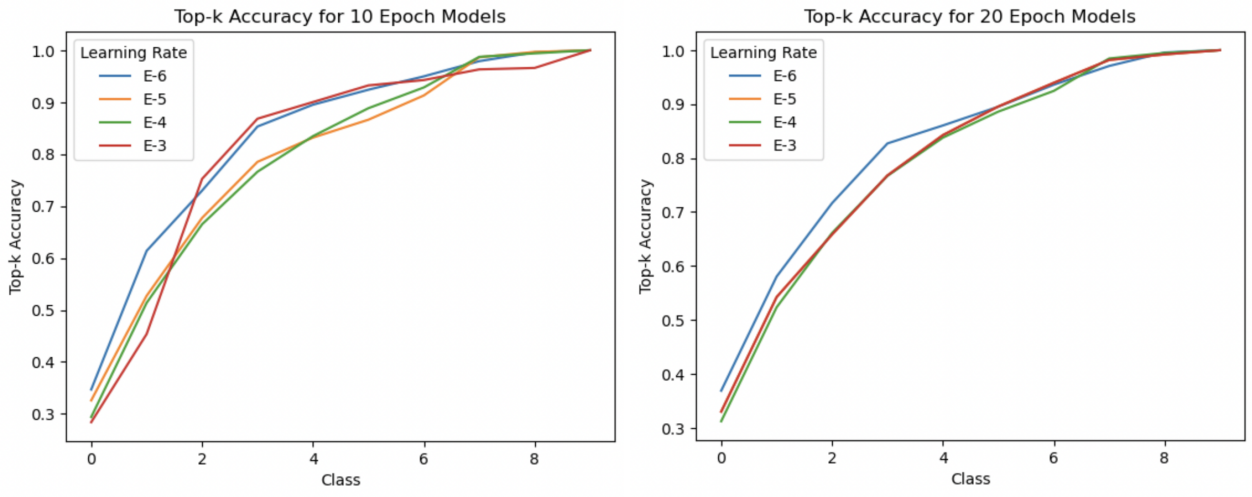

Figure 8: Top-k Accuracy Scores by patch for all k Classes.

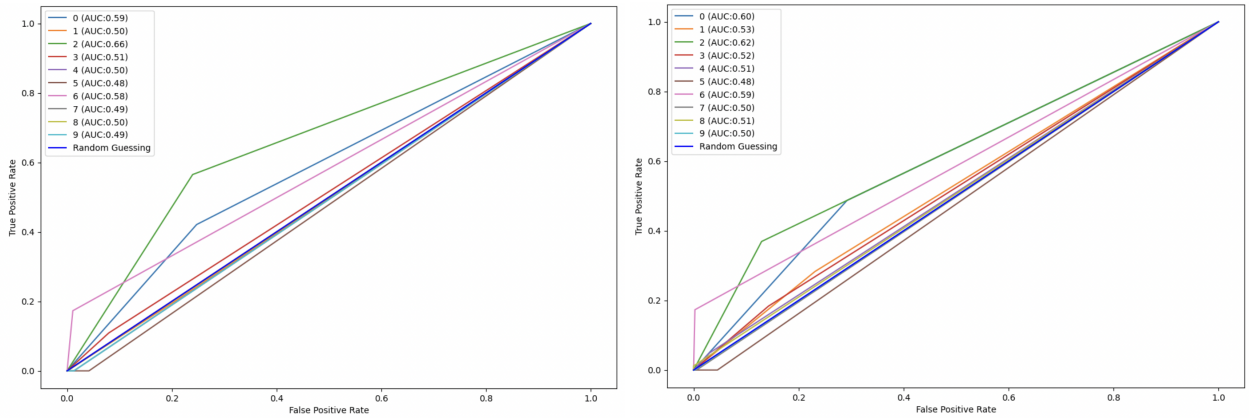

Figure 9: AUC for 0.0001 Learning Rate on 10 and 20 Epochs

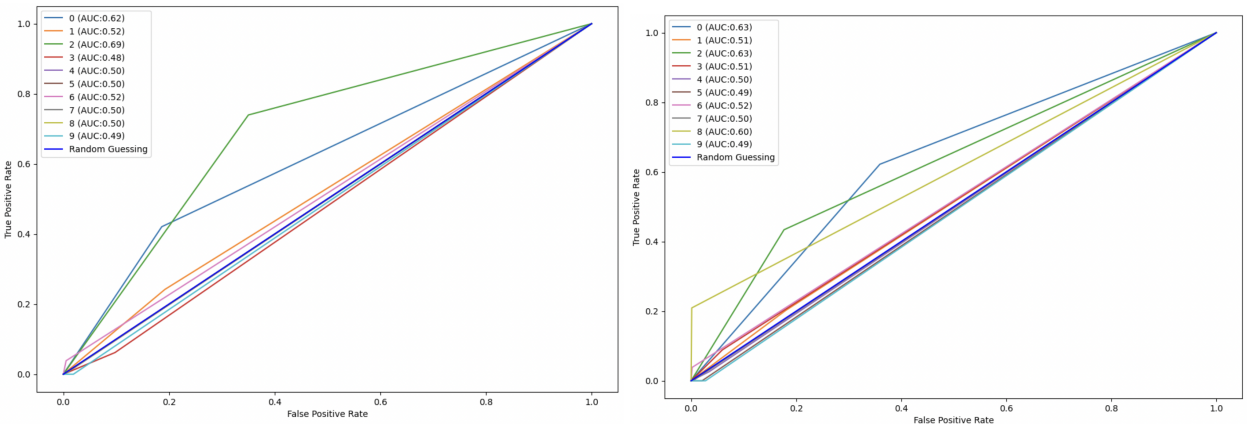

Figure 10: AUC for 0.00001 Learning Rate on 10 and 20 Epochs

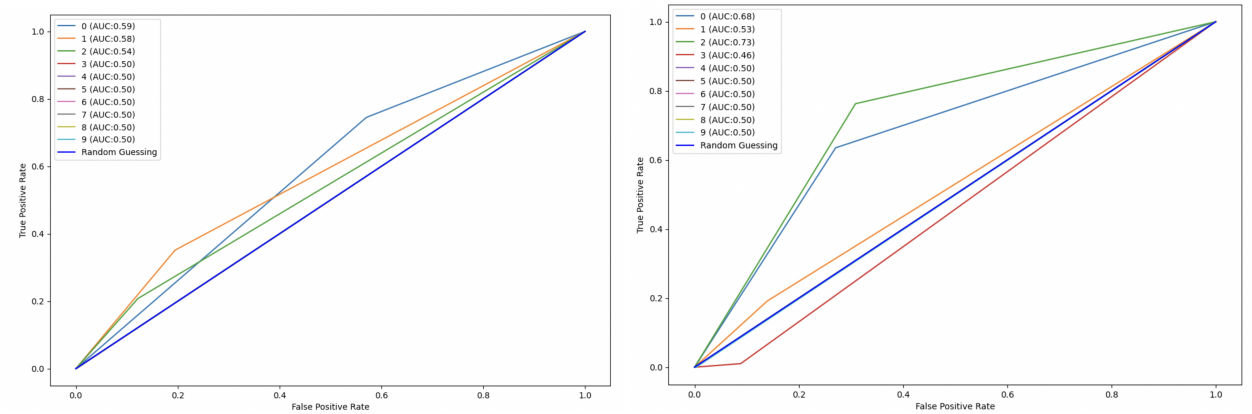

Figure 11: AUC for 0.000001 Learning Rate on 10 and 20 Epochs

5.3. Analysis

As expected, estimating percent tumor necrosis as a multiclass classification problem proved to be a challenging problem. At best, patch-to-patch accuracy reached 36.87%, more than three times better than random guessing in a 10-way classification problem. When considering balanced accuracy, however, the best achieved was 16.2%, barely better than random guessing. These two best scores were not achieved with the same model parameters. The uneven distribution of labels in the training set was likely to blame for this and the weak performance overall. The AUC plots confirm this hypothesis: higher labels of tumor necrosis, of which there were fewer histologies in the database and datasets, hover around scores of 0.5, meaning that the model is predicting them no better than random guessing, whereas the lower labels, of which there were many histologies, reach AUC scores in the 0.6s and 0.7s consistently across different model parameterizations. The greater frequency of these labels in the data likely pulled up the average AUC score as reported in the summary tables, when in reality most classes had poor AUC scores individually. F1 scores were very low for all models. The ROC AUC score differed little from random guessing. A smaller learning rate and more epochs of training tended to improve performance across metrics.

When considering patch-to-whole slide evaluation, our model performed even worse than with patch-to-patch. The best accuracy achieved was just 28.26% and the best balanced accuracy, 14.8%. Curiously, model performance did not track as closely with parameters as it did in the patch-to-patch

evaluation. The best performing model by slide was with learning rate 0.0001 on 20 epochs of training. Other models actually performed worse than random guessing. If the 0.001 learning rate models were no better than random guessing according to their balanced accuracies and AUC scores, the baseline unweighted accuracy by slide by slide is the reported 23.91%, higher due to the imbalance of labels. All ten-epoch parameterizations, except for the one with the same learning rate, actually reported worse unbalanced accuracies than 23.91%. As with patch-to-patch evaluation, F1 scores were very low, as were AUC scores.

The top-k accuracy scores give the most successful performance metric and are the most relevant, given the circumstances of the training and the goal of the project. The top-3 accuracy score for every model is well over 50%, meaning that the correct label lies within the top three labels predicted by the model over half the time. In fact, all the top-k curves climb steeply, typically crossing 90% by $k = 5$. This indicates that the model is very rough in its estimation, but not on the entirely wrong track. This is significant because the model is so simply designed and the assumptions of the labeling scheme so large that it is impressive that the model tends to be not far off in its top few predictions. Mid-range accuracy for any multiclass problem is not trivial and our model was off by only a few classes.

6. Summary

This project tackled the problem of estimating percent tumor necrosis in glioblastoma histologies with the novel approach of treating it as a multiclass classification problem split among ten 10% increments. Using a lightweight CNN and minimal training, the model achieved impressive success in roughly estimating the top 3-5 labels when evaluating performance by patch. The largest problems with the model proved to be related to the uneven distribution of data in training and test sets, an unavoidable imbalance inherited from the TCIA database. Fortunately, such an imbalance in the dataset is indicative of a similar imbalance in the real world of pathology: very rarely is a tumor biopsy sample more than 70% necrotic.

While future research may focus on other methods of estimating percent tumor necrosis, this

project has yielded some important insights. In particular, the power of such a lightweight model and so little training time to classify such complex images is not to be underestimated. Especially as machine learning for histopathology approaches the stage of efficient commercial implementation and widespread use, researchers would do well to recall what is still possible with fast, simple models. Top-k accuracy as an useful measure of multiclass classification performance should also be applied to better evaluate future studies with similar labeling assumptions. Were we to continue research on this topic with the same dataset, we would attempt a labeling scheme that probabilistically distributes labels to patches according to a Gaussian distribution about the whole slide's percent tumor necrosis. Label percentage ranges may also be changed such that they are not all 10% increments, given that reality does not bare out such a distribution anyways. Estimating tumor necrosis by classifying patches as necrotic or not, an easier binary classification problem, and computing the total necrosis percent by area is another strategy worthy of investigation.

7. Acknowledgements

I would like to thank my advisor, Dr. Olga Troyanskaya, for introducing me to the intersection of health sciences and computer science this semester and encouraging my pursuit of this project despite my lack of background in it. I would also like to thank Teaching Assistant and Ph.D. candidate Tavis Reed for his help in all the details of getting this project started and finished this semester. Lastly, I would like to thank my fellow undergraduate collaborators, Manya Zhu and Sydney Pittignano, who made our work on similar projects fun and less stressful. Data used in this publication were generated by the National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC).

8. Honor Code

This paper represents my best work in accordance with University regulations.

/s/ Konstantin Howard

References

[1] "The clinical proteomic tumor analysis consortium glioblastoma multiforme collection (cptac-gbm)," 2018. [Online]. Available: https://doi.org/10.7937/K9/TCIA.2018.3RJE41Q1

[2] H. Arunachalam *et al.*, "Viable and necrotic tumor assessment from whole slide images of osteosarcoma using machine-learning and deep-learning models," *PLoS One*, 2019.

[3] E. Cosatto *et al.*, "A multi-scale conditional deep model for tumor cell ratio counting," in *Medical Imaging 2021: Digital Pathology*, ser. SPIE Medical Imaging. International Society for Optics and Photonics, 2021.

[4] S. Dabeer, M. M. Khan, and S. Islam, "Cancer diagnosis in histopathological image: Cnn based approach," *Informatics in Medicine Unlocked*, vol. 16, 2019.

[5] A. Epstein, L. Kwahli, and P. Hu, "Multiple uses of tumor necrosis therapy (tnt) for the treatment and imaging of solid tumors: Preclinical considerations and progress," *Update on Cancer Therapeutics*, vol. 1, pp. 33–47, 2006.

[6] M. Fromer, "Pathology errors can have serious effect on cancer diagnosis & treatment," *Oncology Times*, vol. 27, pp. 25–26, 2005.

[7] Z. Han *et al.*, "Breast cancer multi-classification from histopathological images with structured deep learning model," *Scientific Reports*, vol. 4172, 2017.

[8] A. Morisi *et al.*, "Detection of necrosis in digitised whole-slide images for better grading of canine soft-tissue sarcomas using machine-learning," *Veterinary Sciences*, vol. 10, 2023.

[9] A. Smits *et al.*, "The estimation of tumor cell percentage for molecular testing by pathologists is not accurate," *Modern Pathology*, vol. 27, pp. 168–174, 2014.

[10] F. Sokol. Glioblastoma - histopathology. Filip Sokol on YouTube. Available: https://www.youtube.com/watch?v=tqGdlaYtrsE

[11] R. Stone. py-wsi. Open Source. Available: https://github.com/ysbecca/py-wsi

[12] J. Thakkar, P. P. Peruzzi, and V. Prabhu. Glioblastoma multiforme. American Association of Neurological Surgeons. Available: https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Glioblastoma-Multiforme

[13] P. Yee and W. Li, "Tumor necrosis: a synergistic consequence of metabolic stress and inflammation," *Bioessays*, vol. 7, 2021.

[14] A. Yonekura *et al.*, "Automatic disease stage classification of glioblastoma multiforme histopathological images using deep convolutional neural network," *Biomedical Engineering Letters*, vol. 8, pp. 321–327, 2018.