Using CNN's to classify GBM and LGG pathology images

Matthew Millett
Stanford University
450 Serra Mall, Stanford, CA 94305
millett@stanford.edu

Abstract

Pathology slides have long been useful in various diagnoses, including cancer. Convolutional Neural Networks can be a useful tool in diagnosis, but large and heterogeneous pathology slide images make it difficult to apply them. Here I adapt common CNN architectures, some pretrained on ImageNet, to classify Pathology whole-slide images from The Cancer Genome Atlas as two different types of cancer, Glioblastoma and Low Grade Glioma. The best model in this paper achieved a classification accuracy of approximately 95%.

1. Introduction

One in four deaths in the United states is due to cancer [9]. Individual cancers have their own quirks that affect treatment. Additionally, different treatments- types of radiation therapy, pharmaceutical interventions- for a given cancer will be more and less effective at different stages of a particular cancers evolution. One tool commonly used by pathologists is the hematoxylin and eosin (H&E) stain, which gives valuable information about cell morphology and spatial arrangement. A slide consists of a slice of tissue that has been fixed and stained with certain dyes to make morphological features, e.g. nuclei, more visible. Currently H&E slides are usually interpreted by unaided pathologists who stand to gain significantly from computer assistance. However, inter-reader variability has been shown to be quite high between pathologists [5]. Digital image processing methods have already been shown to improve consistency, efficiency, and accuracy in evaluating histopathology stains, and can be used for decision support [11].

The input to my algorithm is a histopathology slide from The Cancer Genome Atlas (TCGA) [15]. I then use various CNN architectures with some preprocessing to output a predicted cancer type: Low Grade Glioma (LGG) or Glioblastoma (GBM).

Overall, while transfer learning seems to help slightly, the selection, cell-density and magnification level of the input data have a large impact on accuracy. That said, the best model in this paper was able to achieve a classification accuracy of approximately 95%.

2. Related Work

Currently, most pathologists interpret images on their own without much help from computer-aided sources [2]. Even many machine learning approaches have had a pathologist manually designate areas of interest on the pathology slide [1] [3] [7]. Mousavi et al specifically had a pathologist highlight several areas of interest on potential LGG and high-grade glioma (GBM and clinically more dire LGG) slides to separate them by pathology [13], achieving 95% detection accuracy of high-grade glioma, but of course the model used the pathologist's time, attention and expertise. Manual selection is also so common because these slides can contain around 10^{10} pixels [2], making most classical feature extraction from a whole image computationally intractable.

Barker et al calculated manually-selected features of pathology images to attain 90% accuracy on the same dataset, selecting which images from a slide to use with a voting scheme [2]. Deep learning for tumor grade classification has also been done using an ensemble method with gliomas [8]. Xu et al [16] classified tumors into different subtypes using transfer learning with an AlexNet model. They first trained with ImageNet [6] and fine-tuned on the MICCAI 2014 Brain Tumor Digital Pathology Challenge dataset to account for small sample sizes and large images, and achieved 97.5% accuracy on the dataset. Their CNN goes across the entire pathology image, which can go up to 100,000 by 100,000 pixels, but is the highest reported accuracy for this particular problem.

The presented framework still provides automated classification of histopathology slides, but is less complex and still attains comparable accuracy on The Cancer Genome Atlas (TCGA), a less carefully-curated dataset [15]. Unlike Xu et al, I choose a simple method that only takes the 10 most dense images as defined by Yu et al [17] in the entire image and use it as training data. This reduces billions of

pixels (for a 55000x55000 pixel image) to examine down to millions (at most, 2000x2000x10), which greatly saves computation and still achieves high accuracy. And unlike Barker et al, there is no extra math or domain knowledge required, which means this problem can potentially be extended to other disease types or other problems with unusually large images.

In this paper, I use several previously published convolutional models: VGG [14], ResNet [10], and SqueezeNet [12]. VGG is notable in that it takes only convolutional and only uses convolution, pooling, and fully connected layers [14]. ResNet allows backpropagation gradients to skip over layers so there is less of a vanishing gradient and the network can continue to train [10]. Squeezenet is special because it achieves similar results to AlexNet, but using 50x fewer parameters, whichi allows it to fit onto smaller devices and train more quickly [12].

3. Data

The Cancer Genome Atlas (TCGA) contains large amounts of H&E pathology stain images from cancer patients and has many opportunities for learning [15]. TCGA was chosen because the data is publicly accessible and well-labeled, and is also heterogeneous- these images come from many different whole-slide image scanners. The other benefit of using TCGA is that it hosts data for many different types of tumors, meaning that adapting the methods below to other TCGA datasets should be relatively simple. Also, since none of the features are hand-selected apart from cell density, these methods should apply to many pathology slides.

The input images are scanned Hemotoxylin and Eosin (H&E) pathology slides of GBM or LGG cells, which pathologists usually examine manually for signs of disease. The slides are fixed slices of tissue that are stained for various morphological features, for example nuclei and cell membranes. Despite their widespread usage, histology stains pose unique challenges as compared to other medical imaging modalities such as CT or MRI scans. The images themselves are massive, are not uniformly sized, and contain large amounts of detail. For example, one randomly selected slide was 21973x17532 pixels, and some pathology slides can contain more than 10^10 pixels. The slides can also contain one or multiple globs of cells. I am running a parallel analysis using traditional machine learning methods (my BMI260 final project- due next week so results are still coming in) to see how the respective models perform, and may attempt to combine and compare them at the end of the second project. Currently, the neural network uses padded versions of the entire image but will later incorporate the processing described below.

For evaluation purposes, the images were split into a 90%-10% training-validation split.

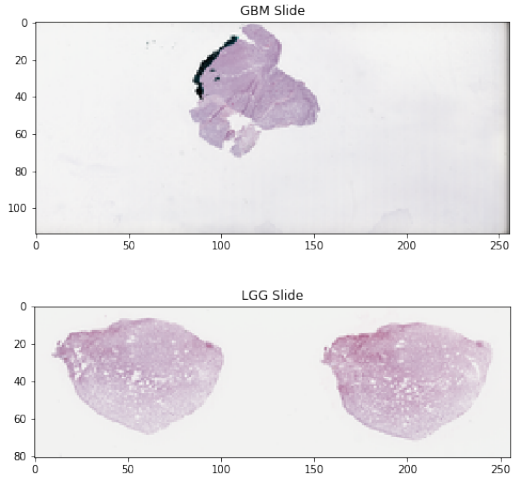

Figure 1. Examples of the input data, significantly scaled down. Images courtesy TCGA.

The images were hosted in .svs format on a Google Cloud instance, so it was fairly simple to download them. However, accessing the image itself is rather complex as each image is stored as a series of sub-images at different magnifications. As an image can contain over 10^10 pixels, the SVS format does not allow loading of the entire full-resolution image into memory; only chunks are accessible at one time.

Proportionally, there are 1193 GBM and 711 LGG full pathology slide images, so images are about 63% GBM and 37% LGG. However, as mentioned below in Methods, the number of images used varied based on choice of pathology slide subparts; since images are heterogeneous in dimension as well as morphology, I was more likely to extract a larger number of cell-dense chunks out of larger images.

4. Methods

There are two main subproblems to solve in this classification task: a coarse search of deciding what subparts of the image to examine, and a finer examination of those selected subparts.

4.1. Subpart Selection of Pathology Slides

One of the most important problems in pathology slide classification is deciding what to look at. The papers mentioned above employ several methods, from voting based on coarse features [2] to simply using a convolution over the entire image, excluding whitespace [16]. I decided to explore different possibilities, and began with a simple way to look at the entire image.

4.1.1 Rescaled 256x256 Images at Weakest Magnification

It seemed surprising that no one had simply fed the whole-slide images to a convolutional network. Because of the

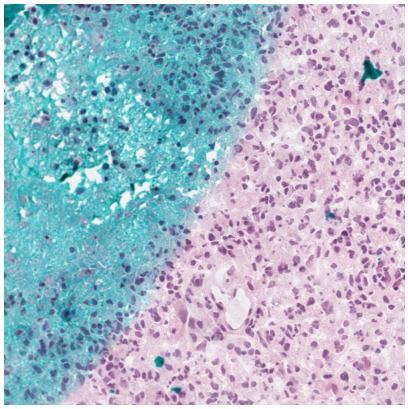

Figure 2. An example image selected by the cell-density pipeline.

heterogeneity of the slides and the somewhat small amount of data (on a deep learning scale), I chose to begin by just resizing and padding the images to be 256x256 pixel squares, to be the same shape as ImageNet data. Distorting the images with resizing, including in any data augmentation, doesn't necessarily make sense with pathology. Tumor x and y dimension ratios could be an important feature. Rescaling the data to 256x256 images also made it amenable to feed directly into known ImageNet architectures. Although clearly this loses a lot of potentially useful information at a detailed level, it is a quick and dirty way to get started.

4.1.2 Images Chosen by Cell Density at Closer Magnification

Because slides are heterogeneous and include significant portions of blank space, this project also adapts a relatively simple approach by Yu et al [17] to select the most cellularly dense portions of each slide for training. For each image, the 10 most dense 1000x1000 tiles were chosen, where image density was calculated as the percentage of nonwhite (all RGB values under 200) pixels in the tile. While choosing the most dense images only might miss out on important data needed to make a diagnosis, it is one of the simple ways to make studying such large images possible and performed appreciably with traditional machine learning methods with manual feature selection. It is important to note that these images were selected at the 2nd-closest possible magnification, because although some of the most detailed features might still be missed, the processing to find the right 1000x1000 slides takes much less time and also does still capture more information across the cell, such as the color change shown in Figure 2.

4.1.3 Different Scaling Methods

Primarily, these 1000x1000 tiles were rescaled with simple bilinear interpolation down to 256x256 to remain easily compatible with existing ImageNet architectures without significant modification. However, I also used the same 1000x1000 images with a modified ResNet architecture. The ResNet architecture fed the pixels from the 1000x1000 images into a 196x196 kernel with stride 8 to form a 224x224 "image" of the maximal convolved values, because perhaps less information is lost in a maxpool than bilinear scaling. The kernel size and stride were chosen in an attempt to minimize computational time, hence the stride was chosen to be as large as possible while maintaining the 224x224x3 output data to plug into ResNet.

4.2. Neural Network selection of finer features

Once subimages had been selected, there were a few different architectures and approaches that I tried with the data.

4.2.1 Data Augmentation

Data augmentation allowed me to increase the amount of data available for training. By nature of how they're made, pathology slides can be randomly rotated flipped before being put on the scanner. Due to randomness in electronics and biology, some color jitter is likely not uncommon. I augmented the data by randomly adding artificial changes to images in the data pipeline.

4.2.2 Normalization

As different scanners have their own settings, each image was self-normalized to keep intensities reasonable. As suggested in the PyTorch transfer learning tutorial, red, green, and blue channels were each normalized to slightly different means and standard deviations [4].

4.2.3 Transfer Learning

Many deep learning image repositories such as ImageNet [6] have tens of thousands of images, but medical imaging generally suffers from a lack of abundant, standardized data and thus data augmentation and transfer learning become useful. I adapted my images to fit the 256x256 architecture created by ImageNet competitors. In all architectures tested, transfer learning conferred a notable benefit upon the model's classification accuracy (See Experiments/Results).

4.2.4 Different Architectures, Different Inputs

I tried a few different architectures that were all available in the Pytorch libraries and had the option to be pre-trained on ImageNet. I used the 18-layer Resnet architecture [10],

the VGG 11-layer architecture without BatchNorm [14], and the SqueezeNet v1.1 architecture [12], both with and without pretrained weights, to assess the benefit of transfer learning on classification. I only ran each model for 5 epochs because models seemed to train quite well over only a few epochs and it didn't take much for the models to differentiate themselves. Of the 3 models, note that Resnet is the only one that uses BatchNorm layers.

All architectures trained with the same binary cross-entropy loss function, where the loss for each class is shown below:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right)$$

4.2.5 Tweaking the loss function

All models seemed to have higher precision than recall, with precision in the 90% range and recall in the 35% range (see Discussion), so I decided to tweak the weights of the loss function in my highest-accuracy model overall to see if I could increase recall. I changed the weight of LGG to be 0.5 and left the weight of GBM as 1 in the loss function. Thus, the weight of an GBM example remains the same but the weight of a LGG example is below:

$$\text{loss}(x, LGG) =$$

$$0.5 * \left(-x[LGG] + \log\left(\sum_j \exp(x[j])\right)\right)$$

,

5. Experiments/Results/Discussion

My primary metric for assessing models was accuracy per **selected** images on a validation set, although I do list precision and recall values for my most impressive model and did use those values to tweak my loss function, which I mention towards the end of this paper. Note that in the case of cell-dense images, I classify each chunk of the entire image instead of the image itself. This is also how I have chosen to evaluate accuracy A voting algorithm for the aggregation of each chunk could be created, but since more than 90% of the time the algorithm is correct about a slide by looking at a single cell-dense chunk, this seemed unnecessary. Note that the trivial baseline of classifying every single image as GBM is approximately 62% for all models.

I chose hyperparameters by fiddling slightly with the transfer learning defaults set in the PyTorch transfer learning tutorial [4], but ultimately modified those parameters very little because they yielded high accuracy after fine-tuning on pretrained models. I did not do cross-fold validation because it would take too much time to run on my machine, which I was also using for other projects. To make sure the learning rate wasn't too slow, I increased it significantly and did not see my loss continue decreasing over a few epochs, so I knew my initially chosen learning rate was approximately correct.

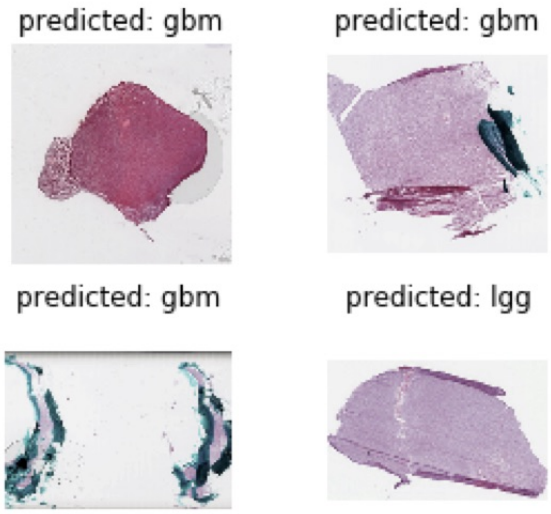

Figure 3. Predictions made by the ResNet model on whole pathology slides. Images courtesy TCGA, but rescaled and padded to be 256x256 pixels to fit into the neural network model. Note the heterogeneity in the data and different blobs of cells in multiple locations.

Overall, VGGNet using transfer learning performed most accurately, but training took 1 hour to run 5 epochs for a 2% improvement without significant improvement in recall. ResNet without modification on dense images trained in half the time with very similar accuracy, so I used it to test tweaking the loss function. My model that did "scaling" using a MaxPool layer was unsatisfactory and took 2 hours to train, but yielded a 60% accuracy rate, suggesting that the MaxPool is not as effective as bilinear interpolation for extracting information from an image.

5.1. Image selection

It is somewhat surprising that the simple technique of looking at the entire slide as a 256x256 image yields up to 83% accuracy in results, simply because many of the known characteristics of GBM and LGG are mentioned at a cellular level [2].

Selecting and rescaling the top 10 cell-dense images yielded a jump of approximately 10% in accuracy from rescaled whole-slide images, and indicates that the cell carries important characteristics of cancer on an individual level that cannot be assessed as well from an aggregate standpoint. This is consistent with scientific knowledge [2]. (There is an exception in the case of non-pretrained

Model / Dataset	Sq_n	Sq_p	Res_n	Res_p	VGG_n	VGG_p
Rescaled WSI	0.7419	0.7681	0.7903	0.8319	-	-
Rescaled Cell-Dense	0.6215	0.9237	0.9107	0.9469	0.9118	0.9559
Max-Pool input	-	-	-	-	0.6005	-

Figure 4. Accuracy values of each model on the validation set after 5 training epochs.
Key: Sq = SqueezeNet v1.1, Res = ResNet18, VGG=VGG.
Subscripts: n = not pretrained on ImageNet, p = pretrained on ImageNet.
WSI = Whole-Slide Image rescaled to 256x256.

SqueezeNet, which seemed not to learn at all on the dense 256x256 images). Notably, all models (with the exception of SqueezeNet) seemed to perform within about 10% of each other, which speaks to the importance of the data coming into the model. Another marker of the importance of image selection and processing is shown in my attempt to "rescale" the image using MaxPooling and feeding the result into a (non-pretrained) ResNet, which took 2 hours to train and yielded the dismal validation accuracy of 46%. It seems that a MaxPooling layer might not be able to capture the nuances of an image as well as bilinear interpolation, although having a 196x196 kernel size and stride 8 did properly feed into the ResNet architecture, which seems to imply that MaxPooling is not a good way to summarize visual information for a convolutional network designed for normal image input.

5.2. Neural Network Architectures

VGG pretrained on ImageNet and fine-tuned on the scaled cell-dense images performed the best in terms of accuracy, but it took over an hour to train. ResNet trained similarly was able to train in under an hour and achieved results within 2% of VGG. It seems that zooming closer into cells helps with classification and transfer learning helps slightly as well. Although precision is close to 0.93, recall hovers around 0.38 for the best-performing models.

5.2.1 Transfer Learning

Most models gained 2-4% accuracy from transfer learning (See Figure 4). The pretrained VGG network initially started with a higher accuracy and declined slightly upon more training with the model. The small boost gained in the other models seems more consistent with the differences between ImageNet images and pathology slides. It makes sense that dumbbells, monitors, and necklaces (3 example classes from ImageNet [6]) do not have similar morphology to pathology slides, partially because they exist on a microscopic as opposed to a macroscopic scale, so the benefits are limited. Transfer learning did prove to be immensely useful for SqueezeNet model, bringing up accuracy from 62% to 92% , although that seemed like an initialization fluke possibly caused by the model being caught in a local minimum.

5.3. Precision and Recall

As mentioned in the Methods section, most of my models had high precision and low recall. The best-performing model , VGG pretrained on ImageNet and fine-tuned on rescaled cell-dense images (Accuracy of 0.9559), had a precision of 0.9328, but a recall of only 0.3694. The ResNet model, trained similarly, had a similar precision and recall of 0.9224 and 0.3687, respectively. Interestingly, the corresponding models not pretrained on Imagenet had slightly higher recall values (0.3841 and 0.3749, respectively), although their precision values were slightly worse (0.9254 and 0.9015, respectively). This suggests that pretraining on ImageNet increases model sensitivity to GBM tumors, perhaps because particular features are seen as more salient.

The low recall means that fewer GBM's will be seen by the pathologist using this system, which is unfortunate because those are images a pathologist would probably need to see. GBM is a more dire condition [2]. A valiant effort was made to tweak the loss function and make the classifiers more sensitive to GBM, but the transfer learning did not complete in time for submission in this report.

5.4. Comparison with Traditional Machine Learning Methods

In my other project for BMI260, I use various calculated morphological features of the selected 1000x1000 to predict cell type and achieve an accuracy of approximately 90% on the test set. The best model discussed here, the pretrained VGG fine-tuned on cell-dense images, achieves approximately 95% accuracy on the same images. This is useful because, as mentioned with Barker's paper [2], there is no need to calculate biology-specific features. This means that this kind of algorithm and data selection shows more generalizable promise that does not limit itself to LGG and GBM classification.

6. Conclusion/Future Work

In this work I discuss different ways to use convolutional neural networks to classify tumors as Low Grade Glioma or Glioblastoma, based on their histopathology stains. The best-performing algorithm was a VGG net pretrained with ImageNet and fine-tuned on training with the most cellularly-dense subparts (as defined by presence of color) of histopathology slides. Other architectures were attempted, but attain lower accuracy. This method has classification accuracy approximately on par with others [16] [2], but saves computation for deep neural networks and does not require domain-specific knowledge.

One drawback of this model is the low recall rate, despite the high accuracy. If I had more time, I would tweak the loss function as described in the "Methods" section to favor recall over precision.

7. Contributions / Acknowledgments

All the work done in this paper was done by me, although I thank Professor Olivier Gevaert and Alex Momeni for helping me brainstorm on these ideas, and Aaron Dharna for helping me debug my neural network issues.

The only parts of this work used for BMI260 were the image processor I wrote to find the most dense images in a pathology slide and some inevitable overlap in the Introduction/Related Work section.

8. Appendix

All the code is currently hosted on GitHub at https://github.com/millett/pathology_learning.

References

[1] O. S. Al-Kadi. Texture measures combination for improved meningioma classification of histopathological images. *Pattern recognition*, 43(6):2043–2053, 2010.

[2] J. Barker, A. Hoogi, A. Depeursinge, and D. L. Rubin. Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles. *Medical image analysis*, 30:60–71, 2016.

[3] A. N. Basavanhally, S. Ganesan, S. Agner, J. P. Monaco, M. D. Feldman, J. E. Tomaszewski, G. Bhanot, and A. Madabhushi. Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology. *IEEE Transactions on biomedical engineering*, 57(3):642–653, 2010.

[4] S. Chilamkurthy. Pytorch transfer learning tutorial, 2017.

[5] S. W. Coons, P. C. Johnson, B. W. Scheithauer, A. J. Yates, and D. K. Pearl. Improving diagnostic accuracy and interobserver concordance in the classification and grading of primary gliomas. *Cancer*, 79(7):1381–1393, 1997.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] S. Doyle, M. D. Feldman, N. Shih, J. Tomaszewski, and A. Madabhushi. Cascaded discrimination of normal, abnormal, and confounder classes in histopathology: Gleason grading of prostate cancer. *BMC bioinformatics*, 13(1):282, 2012.

[8] M. G. Ertosun and D. L. Rubin. Automated grading of gliomas using deep learning in digital pathology images: A modular approach with ensemble of convolutional neural networks. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1899. American Medical Informatics Association, 2015.

[9] U. C. S. W. Group et al. United states cancer statistics: 1999-2014 incidence and mortality web-based report. *Atlanta: US Department of Health and Human Services, Centers for Disease Control and Prevention and National Cancer Institute*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] J. Hipp, T. Flotte, J. Monaco, J. Cheng, A. Madabhushi, Y. Yagi, J. Rodriguez-Canales, M. Emmert-Buck, M. C. Dugan, S. Hewitt, et al. Computer aided diagnostic tools aim to empower rather than replace pathologists: Lessons learned from computational chess. *Journal of pathology informatics*, 2, 2011.

[12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[13] H. S. Mousavi, V. Monga, G. Rao, and A. U. Rao. Automated discrimination of lower and higher grade gliomas based on histopathological image analysis. *Journal of pathology informatics*, 6, 2015.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113, 2013.

[16] Y. Xu, Z. Jia, L.-B. Wang, Y. Ai, F. Zhang, M. Lai, I. Eric, and C. Chang. Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC bioinformatics*, 18(1):281, 2017.

[17] K.-H. Yu, C. Zhang, G. J. Berry, R. B. Altman, C. Ré, D. L. Rubin, and M. Snyder. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nature communications*, 7:12474, 2016.