DeepGlio:
A Novel Computational Pipeline for the Segmentation, Classification, and Characterization of Low and High-Grade Brain Tumors

Julia Christina Camacho
Texas Academy of Mathematics and Science
June 2019–February 2020

Presented at the following competitions in 2020:
Texas State Science and Engineering Fair (competition canceled due to COVID-19)
Fort Worth Regional Science and Engineering Fair (2nd Place in Computer Science and multiple special awards)

Abstract:

Every year, over 23,800 adults in the United States are diagnosed with brain tumors, and only 33% survive for 5 years or longer following diagnosis. Early diagnosis and treatment of gliomas, the most common type of brain tumors, are crucial for improving patient prognosis. However, due to the innate complexity and heterogeneity of these tumors, current analysis procedures are challenging and time-consuming for radiologists and clinicians. To address this problem, this project developed DeepGlio, a computational pipeline to segment, classify, and characterize low and high-grade gliomas using deep learning methods and patient MRI, genetic, demographic, and survival data.

DeepGlio consists of three main components: first, visual tumor detection and segmentation into 3 distinct subregions using patient MRI scans; second, histopathological grade classification using these segmentation outputs as well as patient genetic data; and third, patient overall survival (OS) prediction using segmentation, grade, genetic, and demographic data. To achieve these goals, state-of-the-art convolutional and deep neural networks were used to construct predictive engines.

Evaluation of the pipeline using validation data showed promising results. Tumor detection and segmentation produced excellent visual mask and bounding-box predictions, with an average whole-tumor Dice accuracy score of 0.85. The grade classification and OS prediction engines achieved near-perfect ROC AUC and accuracy scores. After pipeline evaluation, feature selection and Grad-CAM visual analysis were performed, identifying 7 key data features and the model mechanisms behind accurate grade classification and OS prediction. Through its novel sequential and multi-input construction and implementation, DeepGlio enables accurate and efficient computational glioma diagnosis and analysis.

Research Assistance and Timeline:

I conducted my research by myself from June 2019 to February 2020.

Table of Contents

Introduction: Problem Statement and Project Objectives

In this project, I developed DeepGlio, a novel computational pipeline for the segmentation, classification, and characterization of gliomas, the most common type of primary brain tumors.

Every year, over 23,800 adults in the United States are diagnosed with brain or central nervous system tumors. These tumors are one of the most deadly types of cancers—of these patients, only 33% will survive for 5 years or longer following diagnosis (National Cancer Institute, 2019). For patients with glioblastoma multiforme (GBM), the most aggressive variant of brain cancer, the survival rates drop even further: only 5% of patients survive for 5 years or longer. If a patient's tumor is left untreated, they will only survive for a median of 3 months (Tamimi & Juweid, 2017).

Early diagnosis and treatment of gliomas is crucial for the improvement of patient prognosis. Surgical resection, radiotherapy, and chemotherapy are the most common treatment options available today, but to properly diagnose patients and choose the most effective course of treatment for each individual, several key steps must first be taken. These steps, which can be broken down into three main components, are as follows: first, tumor detection and segmentation, in which the tumor is visually identified and segmented into different histological sub-regions; second, grade classification, or the determination of the tumor's histopathological grade; and finally, characterization, which encompasses genomic analysis and assessment of patient survival.

However, due to the intrinsic complexity and heterogeneity of gliomas, these procedures are currently very challenging and time-consuming for radiologists and clinicians to undertake. With the increasing availability of patient data in our modern age of big data, computational methods and machine learning present an opportunity to develop high-speed and low-cost solutions to this problem.

DeepGlio is a pipeline that satisfies the following objectives: first, visual tumor detection and segmentation into 3 distinct subregions using patient magnetic resonance imaging (MRI) scans;

second, histopathological grade classification using these segmentation outputs as well as patient genetic data; and third, patient overall survival (OS) prediction using segmentation, grade, genetic, and demographic data. To achieve these three goals, I used state-of-the-art convolutional and deep neural networks to construct machine learning predictive engines.

Background

Gliomas, or tumors that originate from glial cells (astrocytes, oligodendrocytes, microglia, and ependymal cells) in the brain, are the most common primary brain tumors. Gliomas can vary greatly in aggressiveness and location, and symptoms may include headaches, seizures, numbness, dizziness, loss of vision, and speech problems (Johns Hopkins Medicine, 2019).

The main types of gliomas include astrocytomas, oligodendrogliomas, oligo-astrocytomas, ependymomas, brain stem gliomas, and optic pathway gliomas. According to the most recent World Health Organization (WHO) Classification of Tumors report, there are four main grades (aggressiveness/growth-potential classes) of gliomas (I-IV), determined by both histological and molecular criteria. The inclusion of molecular criteria, which include the mutational characteristics of genes such as IDH and ATRX, in tumor grading is a relatively recent development, as it was first included in the 2016 report (Louis *et al.*, 2016). Thus, this component of tumor analysis still has much more to be explored, as only a few genes have been identified to have correlations with glioma development and accurate grading and diagnosis currently has not been achieved based on genetic data alone.

Diagnosis of gliomas can consist of several steps, including brain scans, biopsies, and neurological exams. Early diagnosis is crucial for improving patient prognosis and survival outcomes; left untreated, gliomas can become increasingly aggressive, leading to tumors that are harder to treat and decreased chances of patient survival.

The four most common treatment strategies available today are observation, surgery, radiotherapy, and chemotherapy. Treatment choice depends on the class of the tumor, and is usually based upon tumor location, patient symptoms, and case-by-case analysis of patient risks. For low-grade tumors (generally defined as classes I and II), treatment usually consists of observation or surgery. For high-grade tumors (classes III and IV), treatment may consist of surgery, radiation, or

chemotherapy (Mayfield Clinic, 2018). Surgery may also be used to relieve pressure in the brain due to rapid tumor growth.

As treatment strategies differ based on tumor grade and characteristics, accurate glioma grading and analysis are crucial for treatment planning and prediction of patient prognosis. In addition, tumor segmentation into various histological subcomponents is greatly helpful when determining optimal surgical strategies (prioritization of removal, avoidance of critical brain areas, etc.). Due to the innate heterogeneity of glioma appearance and shape, visually segmenting gliomas is a challenging and time-consuming task.

To segment and grade tumors, visual depiction methods are needed. Current noninvasive methods for visual glioma analysis include neuroimaging such as magnetic resonance imaging (MRI), which uses a magnetic field and radiofrequency pulses to distinguish between different types of tissues based on their proton relaxation times. There are three subregions in particular that glioma MRI scans are often segmented into: peritumoral edema, enhancing tumor core, and necrotic/non-enhancing tumor core. These regions are differentiated by their visual appearance using different MRI sequences (T1, T1-Gd, T2, T2-FLAIR, etc.) These sequences differ in their Repetition Time (time between radiofrequency pulses) and Time to Echo (time between pulses and emitted radiofrequency energy signals, which are then converted to pixel intensities), which cause different proton relaxation times (Preston, 2006).

Computational methods are becoming increasingly useful in grading and characterizing gliomas as they can recognize underlying patterns (visual features) in imaging data and more accurately and efficiently perform crucial diagnosis and analysis procedures. In particular, machine learning methods such as convolutional neural networks (CNNs) can be utilized to process MRI images and predict tumor characteristics.

Deep neural networks function by adjusting feature weights in each layer in order to optimize prediction accuracy through backpropagation. Convolutional neural networks (CNNs) can take in

image data, as opposed to simple numerical data, and identify and learn from visual features in the image in order to make predictions. CNNs achieve this through convolutional layers, which use numerical filters to extract features from the images and recognize key patterns in similar images. As the layers progress, the features become higher level (progressing from partial edges of shapes to entire objects, for instance). An example of a filter being applied to an image in a convolutional layer is to the right.

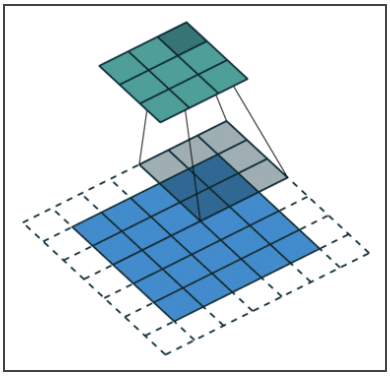


Figure 1. Convolutional Filter.

The CNN structure that I used in this project is Mask R-CNN, a state-of-the-art CNN structure used for object detection and image segmentation (He *et al.,* 2017). Below is an overview of the structure of the model, which produces bounding box, class label, and output mask predictions.

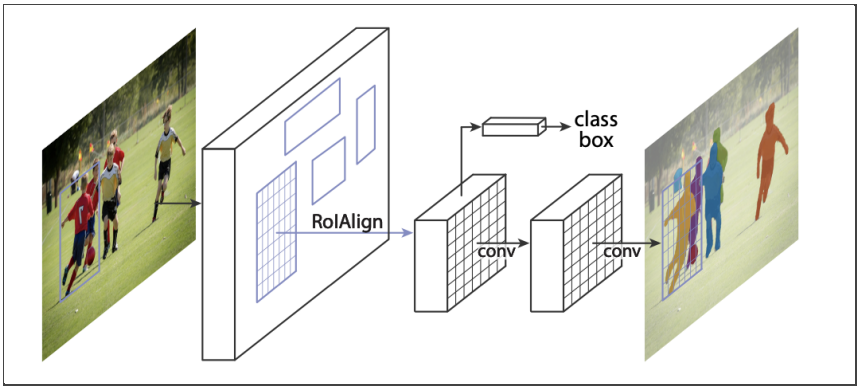


Figure 2. Mask R-CNN model structure.

To facilitate more effective and efficient CNN training, I used transfer learning, which involves re-training a model previously trained on different data (either for the same or a different objective). This aims to "transfer" fundamental model knowledge over from one problem to another application, giving the model an advantage before it begins learning. To further improve prediction accuracy, I also employed methods such as ensemble learning, feature selection and hyperparameter optimization.

Materials, Methods, and Procedures

I had three key aims in my project methodology: first, to detect and segment tumors in MRI images; second, to use the images and patient genetic data to classify the tumors' grades; and finally, to use the images and genetic data along with the tumor grade and patient demographic data to predict overall survival (OS) outcomes for each patient.

I took several key steps in developing my DeepGlio pipeline: first, data preparation; second, construction and training of the tumor detection and segmentation engine; third, construction and training of the grade classification engine; fourth, construction and training of the OS prediction engine; fifth, full evaluation of the pipeline; sixth, feature analysis; and finally, 2D to 3D visualization. I coded all of my engines using Python 3.6 and various image processing and machine learning libraries such as OpenCV and Keras.

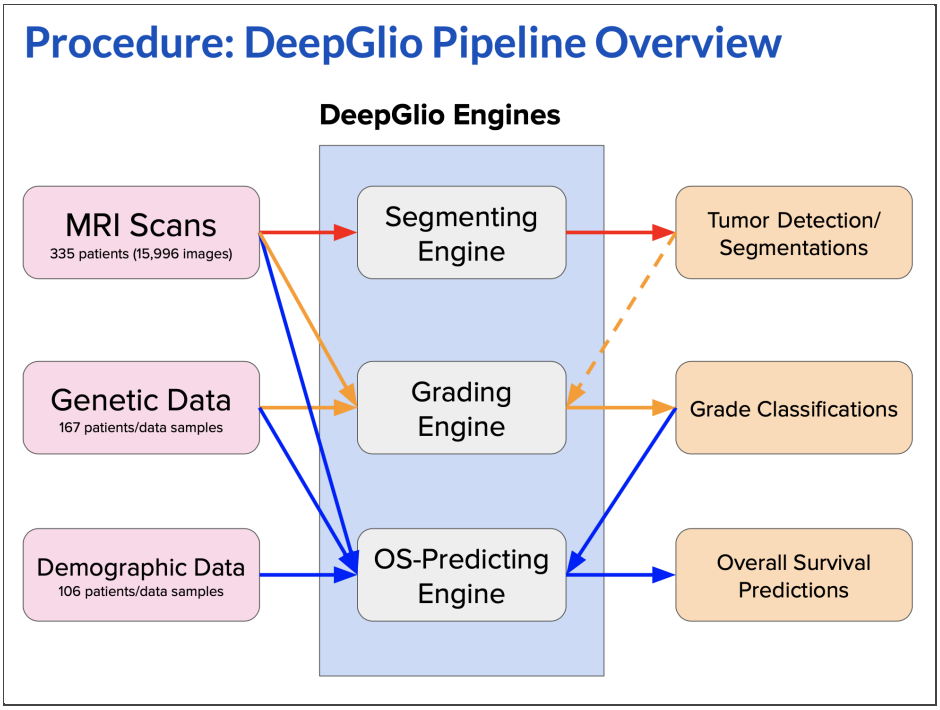


Figure 3. Complete Pipeline Overview

Step 1: Data Preparation

To begin, I obtained publicly available patient MRI scan images and genetic, demographic, and survival data from a 2019 study conducted by the Perelman School of Medicine at the University of Pennsylvania.The data consisted of 355 individuals in total, 167 for whom genetic data were available (through the NIH NCI Genomic Data Commons Data Portal), and 106 for whom demographic and survival data were available. Of these patients, 259 had low-grade glioma (LGG), and 76 had high-grade glioma/glioblastoma (HGG/GBM).

Number of Patients with **LGG** (Low-Grade)	Number of Patients with **HGG** (High-Grade)
259	76

Table 1. The dataset patient statistics for the grade classification dataset.

Number of **Segmentation** Patients/Data Samples	Number of **Grade Classification** Patients/Data Samples	Number of **OS Prediction** Patients/Data Samples
335	167	106

Table 2. The dataset statistics for the segmentation, classification, and characterization datasets.

Each individual had NIfTI-format MRI scan “frame” files, which contained incremental cross-sectional image “slices” of the patients’ brains, and NfFTI-format tumor “mask” files, which showed the ground-truth tumor segmentations for each slice that were created by neuro-radiologists. There were 4 types of MRI scan files, each corresponding to a different MRI contrast: T1 (native), T1-gd (post-contrast T1-weighted), T2 (T2-weighted), and T2-FLAIR (T2-Fluid Attenuated Inversion Recovery).

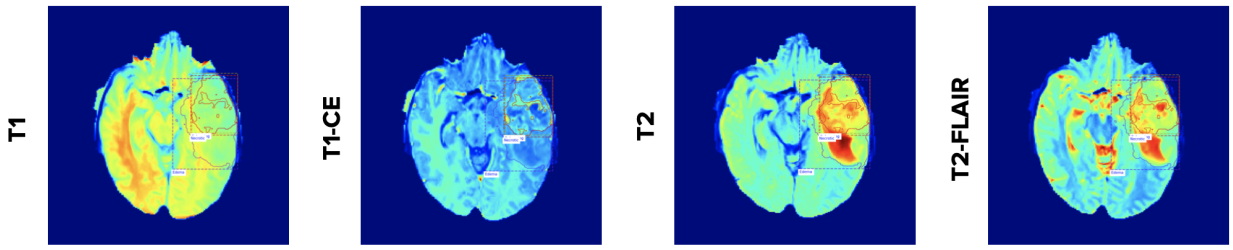


Figure 4. The 4 MRI sequence types with ground-truth segmentations.

In each of the mask files, the MRI images were segmented into 4 distinct areas: background (no tumor), peritumoral edema, tumor GD-enhancing core, and tumor necrotic/non-enhancing core. Thus, each pixel in the mask files had one of the following values: either 0 for background, 2 for edema, 4 for enhancing core, or 1 for necrotic core.

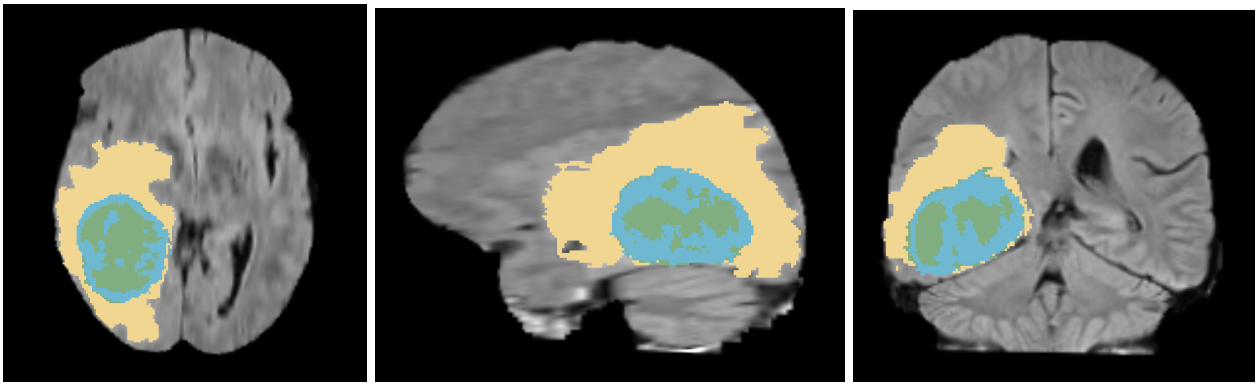


Figure 5. Ground truth segmentations (shown in axial, sagittal, and coronal views) for each of the 3 tumor subregions (segmentations): peritumoral edema (in yellow), enhancing tumor core (in blue), and necrotic/non-enhancing tumor core (in green).

In order to convert these MRI files into numerical arrays that could be used to train a convolutional neural network, I first converted the scans from NIfTI files to PNG images of each slice using the SimpleITK and PNG Python packages. For each of the MRI scans and mask files, slice PNGs were only generated if the slice contained at least one pixel of tumor area (which meant that the mask

slice, or the MRI slice's corresponding mask slice, had at least 1 non-zero, or non-background, pixel value).

Then, these PNG files for both the MRI-scan frames and corresponding tumor-segmentation masks were read and converted to numerical arrays. To achieve this, I created data loading functions using the OpenCV package. For the MRI frame slices, I resized each PNG image to 256x256 pixels and assigned each of the 4 contrast type slices to a different image channel, thus resulting in 256x256x4 arrays. For each of the segmentation mask slices, I once again resized each PNG image to 256x256 pixels, then saved binary segmentation masks for each of the 3 segment categories (pixels assigned to a value of 1 if they were the particular category and 0 if non-category) to different image channels, thus resulting in 256x256x3 arrays.

I obtained the genetic and demographic data for each patient through the NIH NCI Genomic Data Commons (GDC) Data Portal. These data consisted of point mutation data for 5 genes that may have a correlation with glioma development: ATRX (alpha thalassemia X-linked), IDH1 (isocitrate dehydrogenase 1), PTEN (phosphatase and tensin homolog), TP53 (tumor protein 53), and EGFR (epidermal growth factor receptor). After one-hot encoding (converting the categorical data to numerical data) and additively coding (0, 1, 2, etc. based on the number of times that the mutation appeared in the patient's genome) the mutation data, the patient genetic data consisted of 15 features in total. The patient demographic data consisted of 3 features: gender, race, and age, which I then also one-hot encoded. The final number of genetic and demographic data features was 18.

Number of **MRI Images**	Number of **Genetic Features**	Number of **Demographic Features**
15,996	15	3

Table 3. The dataset statistics for all types of data features.

Step 2: Segmentation (Segmenting Engine)

To construct the segmentation engine, I used Mask R-CNN, a state-of-the art convolutional neural network framework for object detection and image segmentation.

I began by dividing the MRI image data (15,996 images in total) into training and testing datasets with a 85/15 split. This resulted in 13,776 images in the training set and 2,220 images in the testing set. Then, I initialized the MR-CNN model configuration and loaded pre-trained model weights (trained on the COCO (Common Objects in Context) image segmentation dataset) in order to facilitate more efficient and effective model training through transfer learning.

Number of Patients in **Segmentation Training Dataset**	Number of Patients in **Segmentation Testing Dataset**
288 (13,776 MRI images)	67 (2,220 MRI images)

Table 4. Number of patients and MRI images in the Segmentation train/test datasets.

To further increase the number of images used for training, I also employed image augmentation. Using the imgaug library, I constructed a custom augmentation configuration to flip and rotate images. I then trained the model "heads" layer for 20 epochs using this augmentation setup and a batch size of 4 images.

After completing this training, which took approximately 13 hours using an NVIDIA Quadro P6000 GPU, I then trained the entire model (all layers) for 40 additional epochs using the same augmentation setup and batch size of 4 images. This training took approximately 40 hours using the same GPU. (Multiple iterations of this training were conducted to determine optimal epoch amounts). After completing this, I downloaded the model weights (h5 file) corresponding to the epoch with the lowest validation loss as well as the TensorBoard output file for loss plotting and model structure visualization.

I then coded my Dice scoring algorithm. The Dice similarity coefficient, a popular image segmentation evaluation metric, measures the spatial overlap between ground truth and prediction segmentations (Taha & Hanbury, 2015).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} = \frac{2TP}{2TP + FP + FN}.$$

Equation 1. Dice similarity coefficient (DSC) formula.

For each slice, I computed the Dice coefficient of element-wise similarity (comparing the value of each pixel) between the ground truth mask and the prediction mask, which I both encoded as four binary masks (one for each subregion type, and one for the whole tumor). I then averaged the sum of these scores for only slices that had at least 100 pixels of each tumor subregion, which prevented skewing due to inclusion of scores of 0 obtained when evaluating slices with ground truth masks lacking in one or more tumor subregions.

Step 3: Classification (Grading Engine)

To construct the grade classification engine, I began by constructing the train and test datasets. To obtain the best representative slices for each tumor and minimize the computations needed to train the predictive model, I coded an algorithm to find the slice with the maximum whole-tumor cross-section area and extract that slice and the two above and below it (in terms of slice order) to use in the datasets. I also addressed the class imbalance issue (102 patients with HGG and 65 with LGG) of the dataset by restricting the number of patients with HGG to 65 to create an evenly split dataset. Through these procedures, I obtained 649 images; using a 75/25 train/test split, I ended up with 486 images in the training set and 163 images in the testing set.

Number of Patients in **Grade Classification Training Dataset**	Number of Patients in **Grade Classification Testing Dataset**
98 (486 MRI images)	32 (163 MRI images)

Table 5. Number of patients and MRI images in the Grade Classification train/test datasets.

I then constructed my predictive CNN model using the Keras Sequential library to create 12 layers in total. I then trained and tested this model using MRI frame images, then ground truth masks. After this, I constructed a multi-layer perceptron (MLP) neural network to train and test using only the genetic data (with no feature selection). Finally, I constructed a multi-input model combining the CNN with the MLP classifier (feeding their individual outputs to a final layer in a neural network structure) to use both the imaging data and genetic data together. I trained and tested this multi-input model using the frame images and genetic data, then the ground truth masks and genetic data. I evaluated the performance of these models using ROC AUC scores, standard accuracy scores, and confusion matrices.

$$accuracy = \frac{TP+TN}{total\ predictions} \qquad true\ pos.rate = \frac{TP}{TP+FN} \qquad false\ pos.rate = \frac{FP}{FP+TN}$$

TP	FP
TN	FN

Equation 2. Accuracy and true positive/false positive formulas and confusion matrix layout.

Step 4: Characterization (OS-Predicting Engine)

To construct the OS prediction engine, I began by adding the tumor grade and patient demographic data to the imaging and genetic data previously used to construct the grading engine. Selecting the patients who had data available in all of these categories resulted in a dataset of 106 patients. Using the same maximum cross-section algorithm as above, I obtained 530 images in total; using a 75/25 train/test split, I ended up with 397 images in the training set and 133 images in the testing set.

Number of Patients in **OS Prediction Training Dataset**	Number of Patients in **OS Prediction Testing Dataset**
79 (397 MRI images)	27 (133 MRI images)

Table 6. Number of patients and MRI images in the OS Prediction train/test datasets.

I then constructed my predictive CNN model using the Keras Sequential library to create 12 layers in total. I then trained and tested this model using MRI frame images, then ground truth masks. After this, I constructed a multi-layer perceptron (MLP) neural network to train and test using only the patient genetic and demographic data. Finally, I constructed a multi-input model combining the CNN with the MLP classifier to use both the imaging data and patient genetic, grade, and demographic (GGD) data together. I trained and tested this multi-input model using the frame images and GGD data, then the ground truth masks and GGD data together.

To explore ways to further improve accuracy, I also implemented feature selection for the GGD data, which has 17 features in total, using permutation shuffling. I computed the permutation importance for each of the 17 features. This was done by permutation-shuffling the sample values for each data feature and calculating the decrease in prediction accuracy, using a Random Forest classifier, that results for each feature being shuffled. Through this, I identified the most important features that could be used to possibly increase OS classification accuracy; I then re-trained the multi-input models using the selected-feature GGD subset. I evaluated the performance of each of the models using standard accuracy scores and confusion matrices.

Step 5: Pipeline Evaluation

I constructed the final DeepGlio predictive engines by choosing the best-performing models for grade classification and OS prediction. To determine statistically significant improvements in prediction accuracy, I conducted t-tests using the data from each model evaluation trial. The final DeepGlio pipeline was then constructed using the maximized-accuracy predictive engines.

Step 6: Feature Analysis

To analyze the biological significance of the genetic features that I had discovered through feature selection to be important for OS Prediction, I utilized the NCBI database to identify the genomic locations of the mutations and the oncological significance of each.

To determine the predictive mechanisms of the CNNs trained for grade classification and OS prediction, I used Grad-CAM (Gradient-weighted Class Activation Mapping) through Keras Visualization to determine the visual features that the models used to determine their predictions. Using Grad-CAM, which produces heatmaps demonstrating the most important regions of the image for prediction, I confirmed that the models were looking at plausible regions of the image and identified key characteristics of both the MRI frame images and segmentation masks crucial for model prediction (Selvaraju *et al.,* 2017).

Step 7: 2D to 3D Visualization

In order to visualize the tumor segmentations in a 3D space, I converted the segmentation masks to 155-slice NIfTI files using the Nibabel library. Then, I used the 3D Slicer MRI scan visualization software to load the NIfTI files and plot them in 2D and 3D spaces. The segmentation files were also converted to files for 3D printing.

Results

Section 1: Tumor Detection and Segmentation

The segmenting engine produced excellent results for both visual tumor detection and segmentation (bounding boxes and prediction confidences) and Dice accuracy scores.

Below are the training and per-epoch testing (validation) losses for the Mask R-CNN model over all 60 epochs of training. At epoch 20, the model training shifted from training of the heads layer only to training of all layers for the final 40 epochs. The clear downward trend in both of the losses indicate that the model was successfully learning from the training data and making increasingly accurate predictions on the testing data.

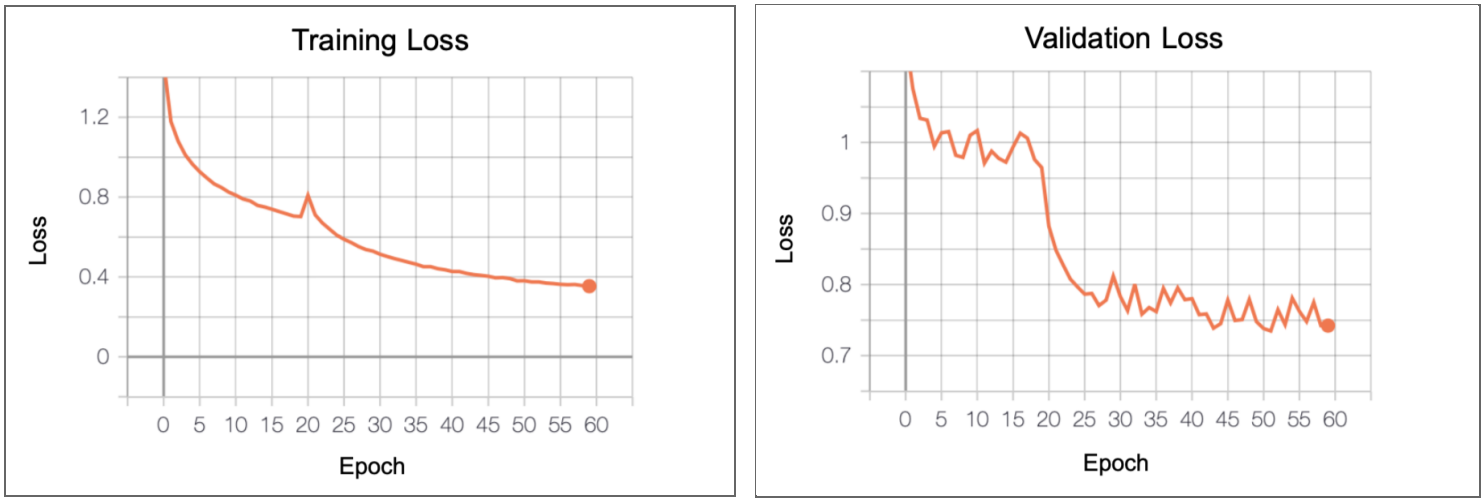


Figure 6. Segmentation engine training and validation model loss graphs

Below are two examples of the model predictions on two slices from the testing dataset. These predictions consist of bounding boxes for the regions of interest (ROI), which are shown by the rectangular boxes surrounding the masks, the segmentation masks themselves, which are shown by contrasting shading colors in the tumor area, and class prediction confidences (shown by the numbers next to the label names) for each of the box/masks' label classifications as either peritumoral edema, enhancing tumor core, or necrotic tumor core.

For each of the tumor segment classes, as well as the whole-tumor area, I calculated Dice accuracy scores to evaluate the quality of the mask detection and area predictions. This was done using the methodology explained in the previous section, and resulted in four sets of scores: one for each subregion class and one for the whole tumor.

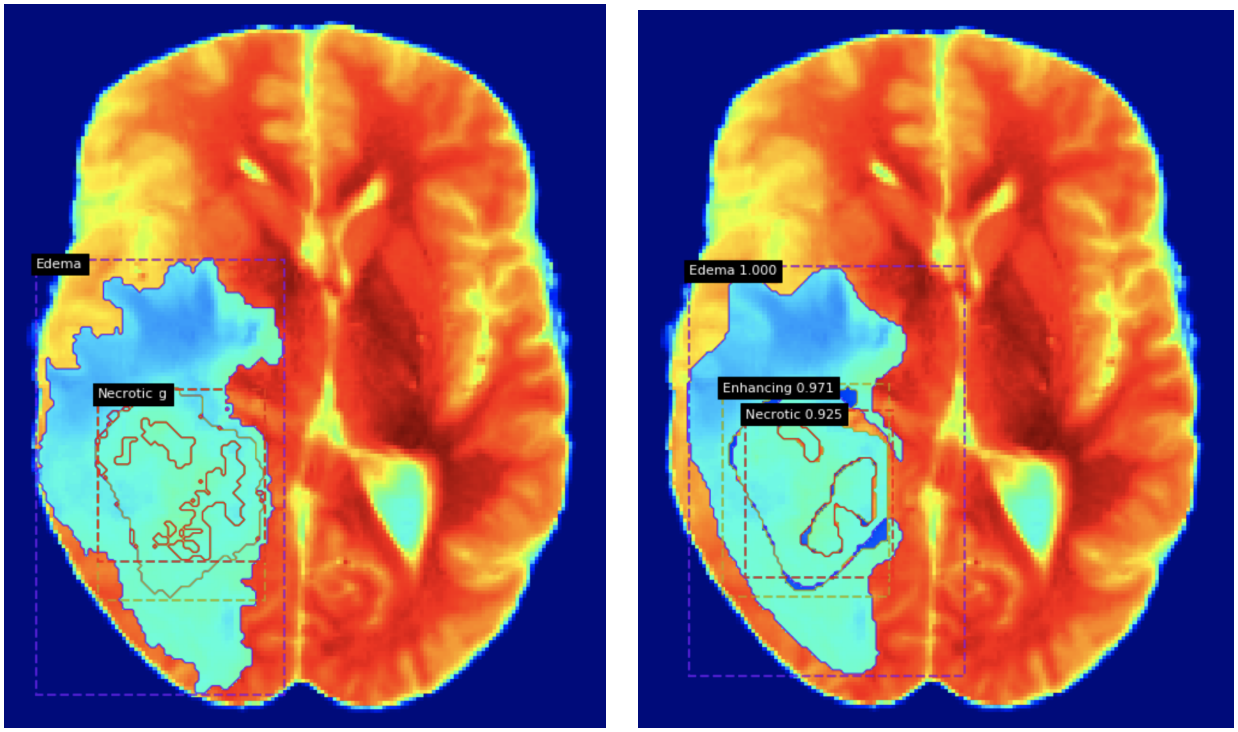


Figure 7. Example of tumor detection and segmentation prediction (ground truth on the left, model prediction on the right) from the Segmenting Engine.

Tumor Section	Dice Score
Edema	0.838
Enhancing Core	0.868
Necrotic Core	0.644
Whole-Tumor	0.909

Table 7. Dice scores for prediction example #1.

To the left, these are the Dice scores for each tumor subregion as well as the entire tumor area.

As shown by the labels on the image on the right, the model segment class prediction confidences showed that the engine was confident in its

predictions, with values of 1.00 for edema, 0.971 for enhancing core, and 0.925 for necrotic core.

The accuracy of the visual detection and segmentation are reflected by the excellent Dice scores. These predictions were made using the Mask R-CNN model weights from the training epoch with the lowest validation loss (which was epoch 51, as shown through the loss graphs previously shown above).

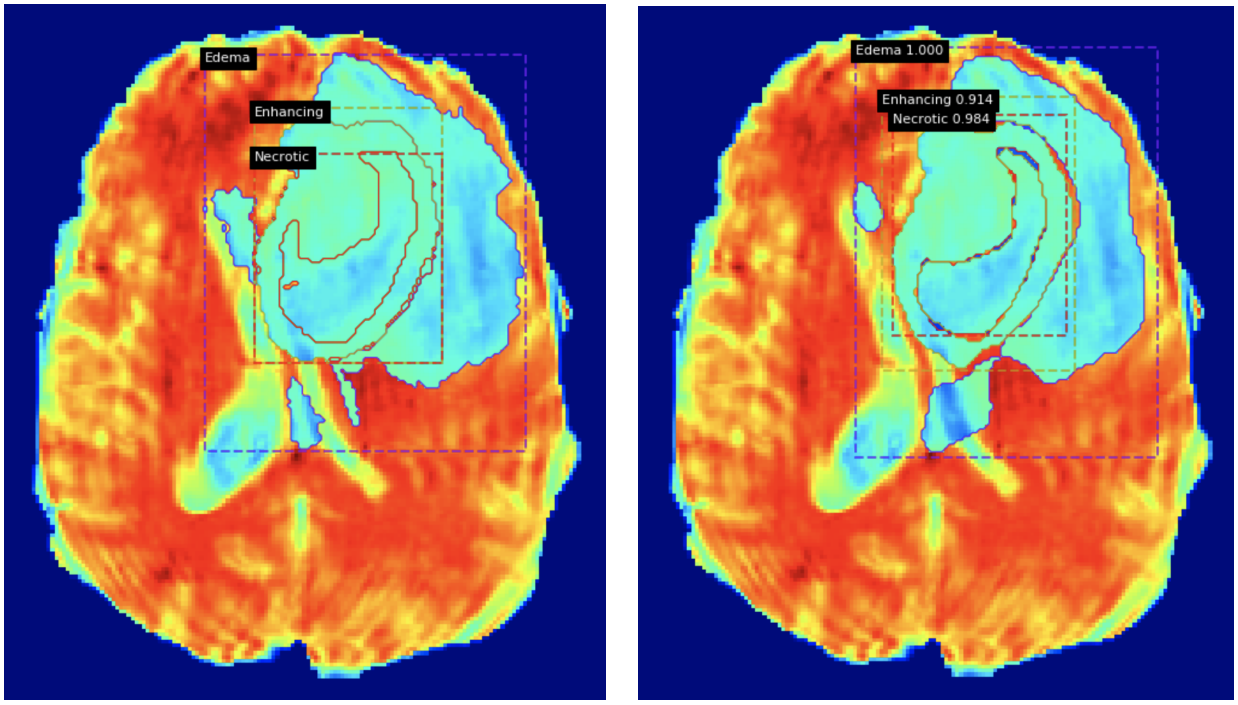


Figure 8. Example of tumor detection and segmentation prediction (ground truth on the left, model prediction on the right) from the Segmenting Engine.

Tumor Section	Dice Score
Edema	0.860
Enhancing Core	0.887
Necrotic Core	0.869
Whole-Tumor	0.921

Once again, the Dice scores for each segment are shown on the left, and once again, the model prediction confidences were solid, with values of 1.00 for edema, 0.971 for enhancing core, and 0.925 for necrotic core.

Table 8. Dice scores for prediction example #2.

These predictions were then generated for all 2,220 slice images in the testing dataset. Below are the average Dice scores (using slices that contained at least 100 pixels of each tumor subregion class, which amounted to 1,377 slices in total) for each of the three subregion classes and the whole-tumor area.

Tumor Section	**Dice Score** (averaged over all validation samples)
Edema	0.777
Enhancing Core	0.768
Necrotic Core	0.517
Whole-Tumor	0.845

Table 9. Averaged segmentation Dice accuracy scores for the validation dataset.

These Dice scores show a drastic improvement in whole-tumor and enhancing-core predictions over existing tumor segmentation methods such as Benson *et al*., 2019.

Section 2: Tumor Histopathological Grade Classification

Next, I constructed the classification engine for tumor grade (low and high-grade) prediction. This process consisted of training and testing the CNN classifier (methodology described in the previous section) on the frame and mask imaging data, training and testing the MLP neural network classifier on the patient genetic data, then developing a multi-input ensemble model to incorporate both imaging and genetic data into a multi-input classifier.

First, I began by training the CNN on frame image data only. (For each of these trials and all following, I repeated the training 3 times on 3 distinct train-test data splits.) This resulted in the

following prediction accuracy scores: 0.97, 0.95, and 0.93 (average: 0.95), respectively in correspondence with the confusion matrices shown below.

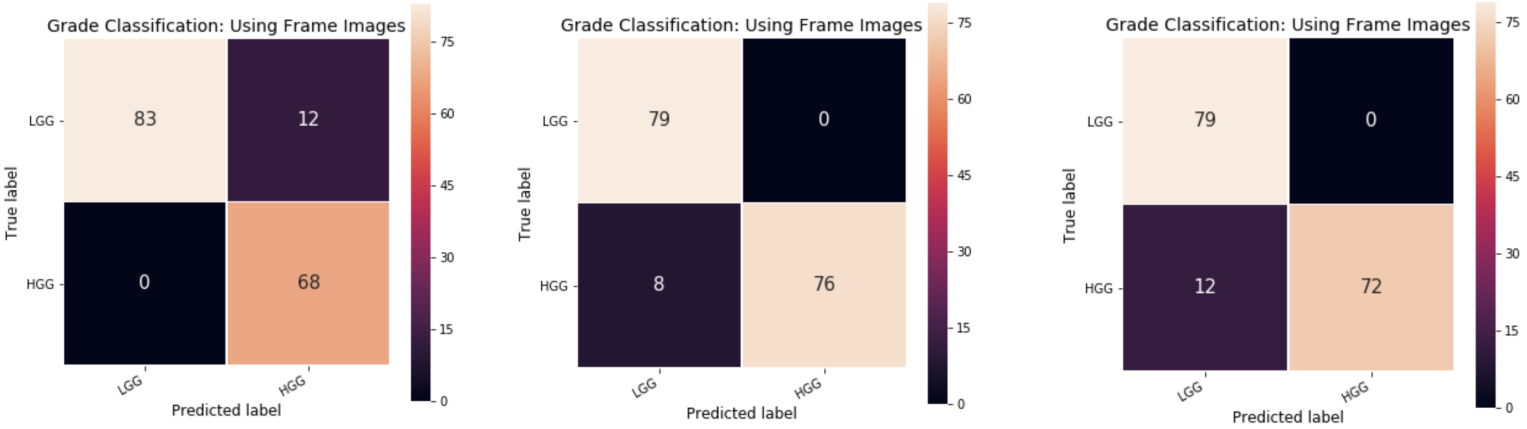


Figure 9. Grade classification predictions using frame imaging data only.

I then trained and tested the CNN on the mask image data only. This resulted in the following prediction accuracy scores: 0.99, 0.88, and 0.74 (average: 0.87), in correspondence with the confusion matrices shown below. In the second and third trials in particular, the model appeared to struggle with LGG and HGG classification, respectively.

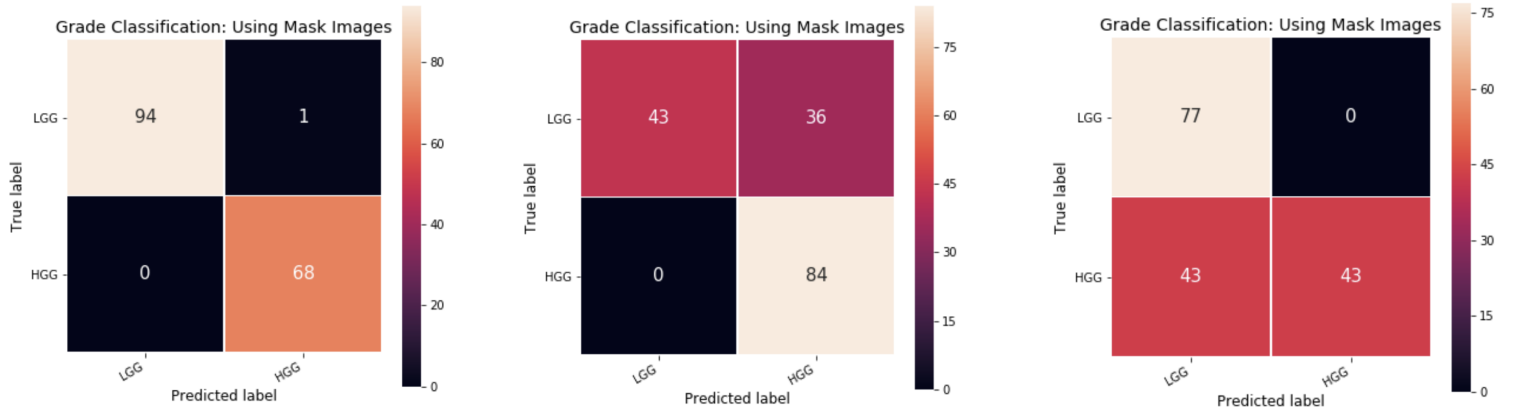


Figure 10. Grade classification predictions using mask imaging data only.

Next, I trained and tested the MLP neural network classifier on the patient genetic data (all 15 features) only. This resulted in the following prediction accuracy scores: 0.81, 0.88, and 0.90 (average: 0.86), in correspondence with the confusion matrices shown below. (The dataset used here was 1/5 of the size of the previous datasets used above, as there was only one genetic data sample per patient, as opposed to five images per patient in the frame and mask datasets.)

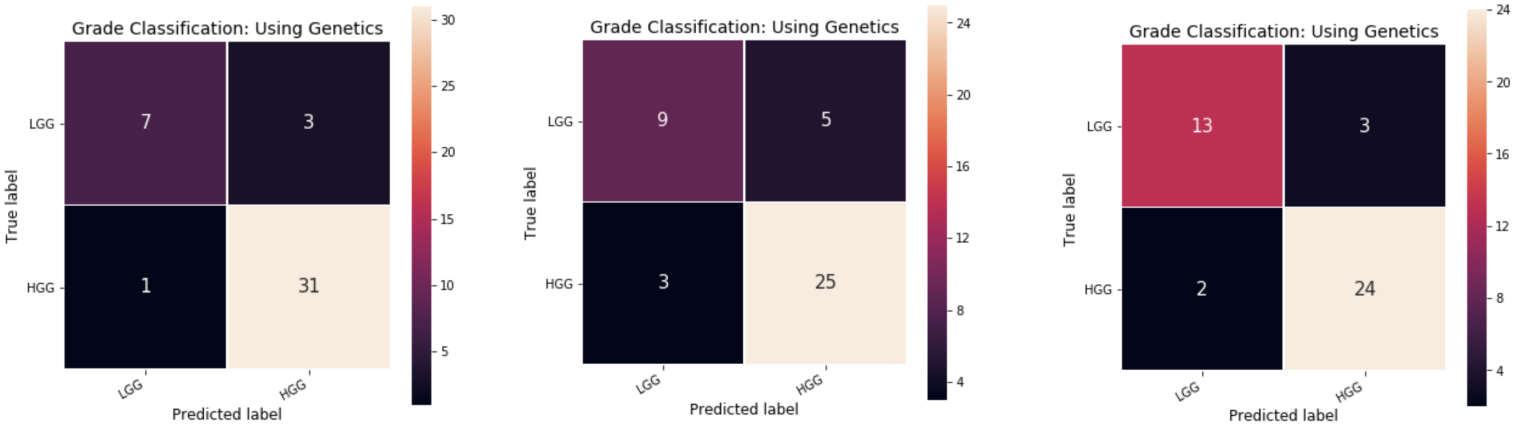


Figure 11. Grade classification predictions using patient genetic data only.

I then constructed a multi-input classification model by combining the CNN and MLP networks (specific methodology described in the previous section). After training and testing using the combined frame images and genetic dataset, I obtained the following accuracy scores: 0.98, 1.00, and 1.00 (average: 0.99). This showed a significant improvement over the results of the models trained on imaging and genetic data alone (results of numerical comparison described in the "Comparison of Predictive Models and Final Pipeline Construction" section).

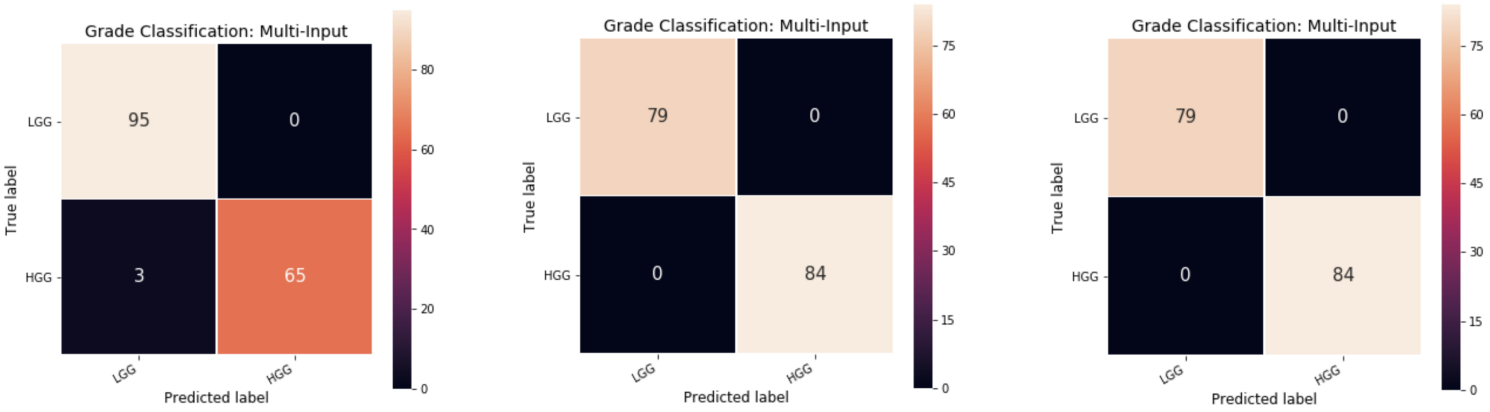


Figure 12. Grade classification predictions using both frame imaging and patient genetic data.

I then also trained and tested the multi-input model using the mask imaging and genetic data. Through this, I obtained the following accuracy scores: 0.98, 1.00, and 0.95 (average: 0.98). This also showed a significant improvement over the results of the models trained on imaging and genetic data alone.

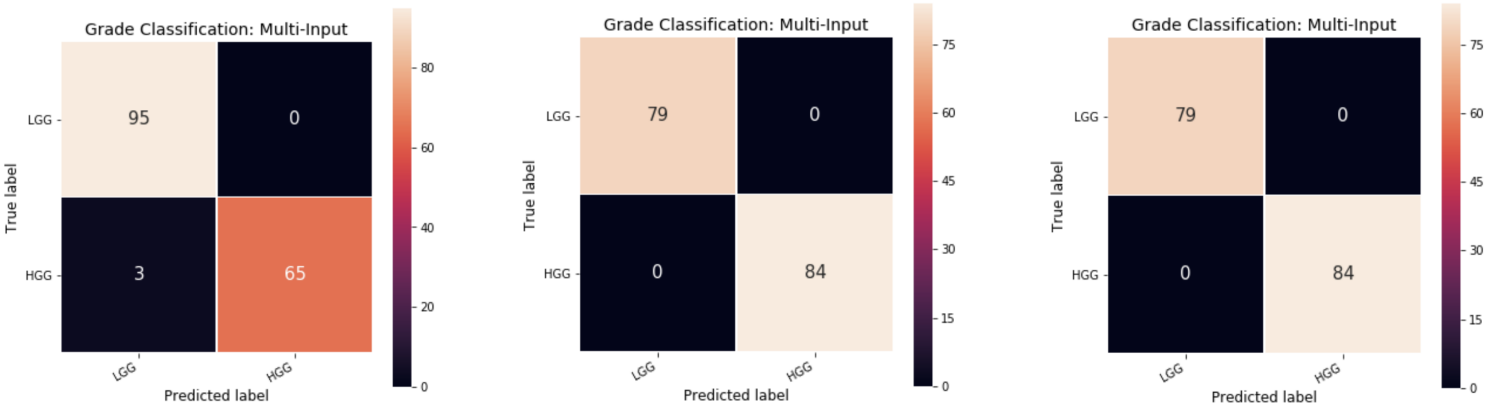


Figure 13. Grade classification predictions using both mask imaging and patient genetic data.

The best multi-input models, with accuracy scores of 1.00, also had ROC AUC scores of 1.00. The ROC curve graph of these models is shown below. This excellent score, which indicates complete prediction accuracy, is an improvement over current grade classification methods employing either images or genetic data alone.

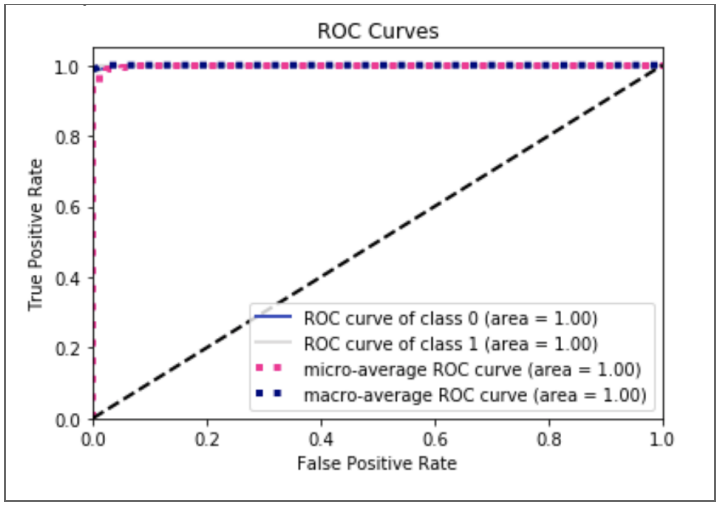


Figure 14. The ROC curve of the highest-performing multi-input models (AUC score: 1.00).

Below are examples of the image-trained model's prediction for both low and high-grade tumors . The prediction confidences were 0.97 and 0.83, respectively.

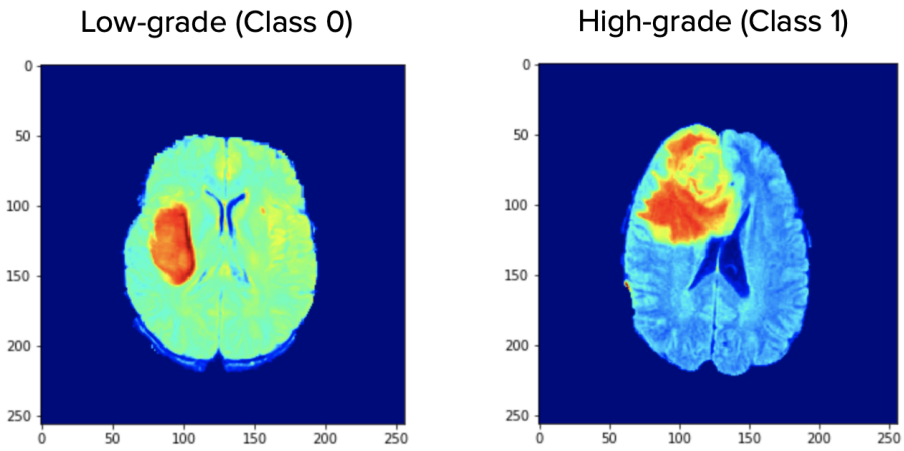


Figure 15. Examples of tumor grading engine predictions (left: low-grade; right: high-grade).

Section 3: Tumor Characterization (Patient Overall Survival Prediction)

Finally, I constructed the OS-predicting engine for tumor characterization (short, medium, or long patient survival time after glioma diagnosis). This process consisted of consisted of training and testing the CNN classifier (methodology described in the previous section) on the frame and mask imaging data, training and testing the MLP neural network classifier on the patient demographic and genetic data, conducting feature selection on the patient genetic, grade, and demographic (GGD) data, and then constructing and evaluating multi-input prediction models incorporating both imaging and the GGD data.

I began by training the CNN on the frame imaging data alone. This resulted in the following accuracy scores: 0.60, 0.94, and 0.97 (average: 0.84), in correspondence with the confusion matrices shown below. In the first trial, the model struggled to correctly classify short-survival images.

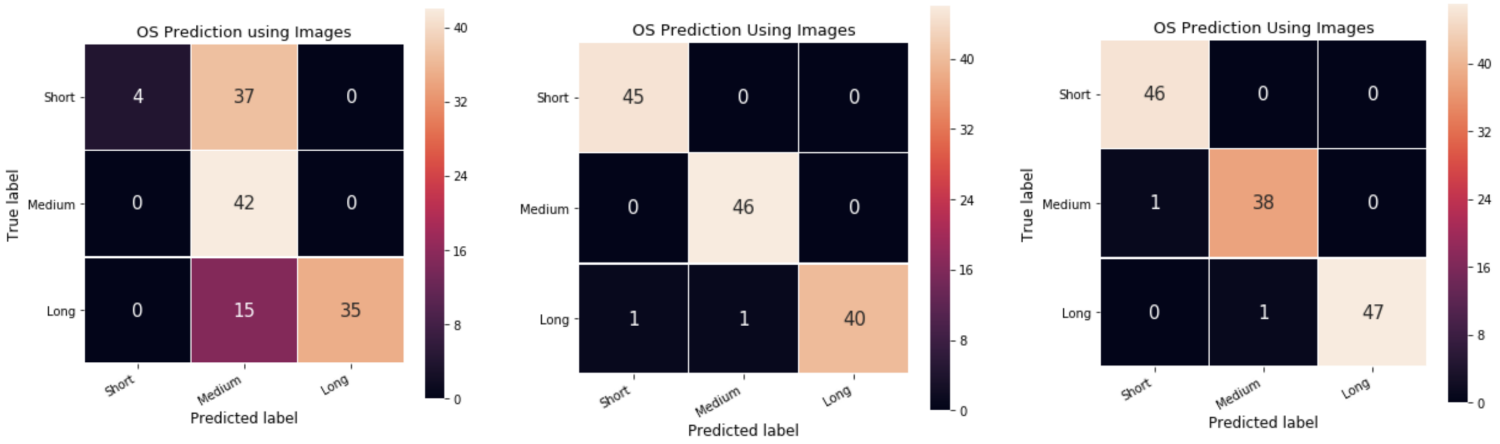


Figure 16. OS classification predictions using frame imaging data alone.

I then trained the CNN on the mask imaging data alone. This resulted in the following accuracy scores: 0.93, 0.90, and 0.84 (average: 0.89), in correspondence with the confusion matrices shown below. The models mainly had difficulty only with long survival prediction.

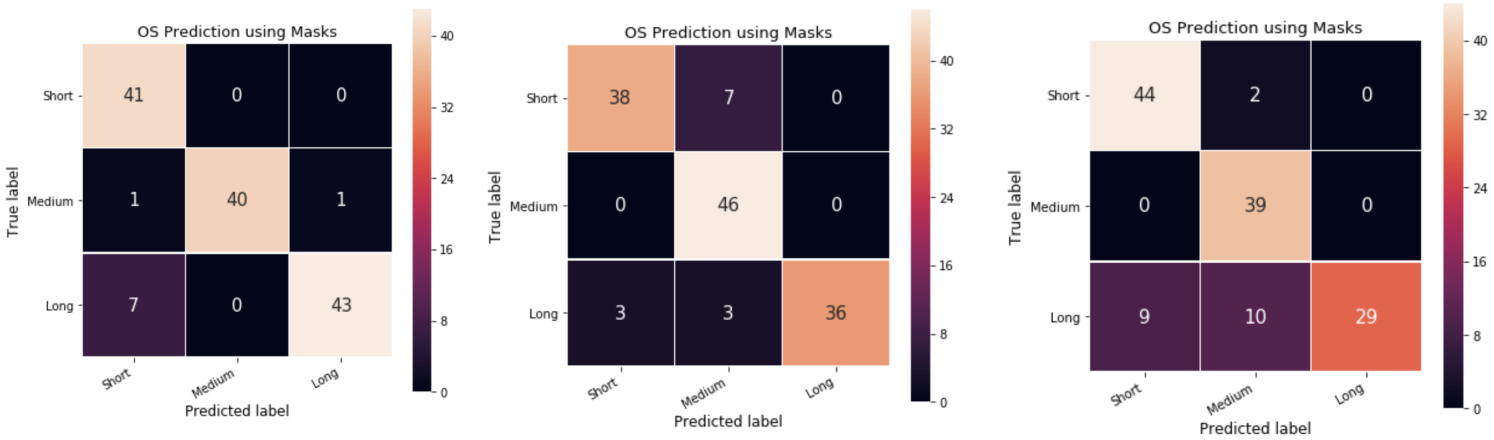


Figure 17. OS classification predictions using mask imaging data alone.

Next, I trained and tested the MLP neural network classifier using the patient genetic and demographic data alone. This resulted in poor accuracy scores: 0.30, 0.33, and 0.22 (average: 0.28). The models appeared to struggle to correctly classify short and medium-length survival patients in particular, as shown in the confusion matrices below.

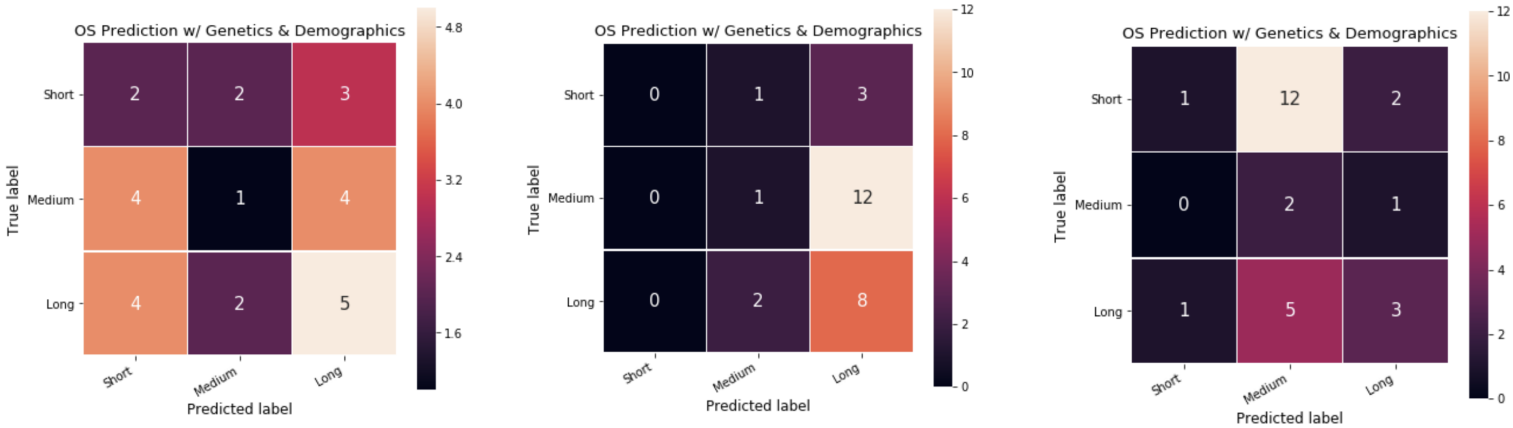


Figure 18. OS classification predictions using patient genetic and demographic data alone.

To remedy these decreases in accuracy, I conducted feature selection to find a feature subset with reduced noise. To achieve this, I used permutation shuffling (described in the "Materials and Methods" section). The results of this procedure are shown below.

Weight	Feature
0.1037 ± 0.0864	Grade
0.0963 ± 0.1005	TP53 Missense
0.0889 ± 0.0755	Age (Years)
0.0889 ± 0.0593	PTEN Missense
0.0370 ± 0.0000	EGFR Missense
0.0074 ± 0.0296	IDH1 Missense
0.0074 ± 0.0864	Race (White)
0 ± 0.0000	PTEN 3' UTR
0 ± 0.0000	PTEN Stop Gained
0 ± 0.0000	PTEN Frameshift
0 ± 0.0000	TP53 Frameshift
0 ± 0.0000	TP53 Stop Gained
0 ± 0.0000	ATRX Frameshift
0 ± 0.0000	ATRX Synonymous
0 ± 0.0000	ATRX Missense
0 ± 0.0000	TP53 Splice Site
0 ± 0.0000	Race (Asian)
0 ± 0.0000	EGFR Intron
-0.0222 ± 0.0363	Race (Black)
-0.0296 ± 0.0864	Gender

Figure 19. Data feature importance results using Random Forest permutation-shuffling feature selection.

I obtained 7 features with high importance (shown in the green rows). The numbers in the "Weight" column show the decreases in accuracy for each feature when the values are shuffled and the Random Forest classifier is re-trained and tested. (The number after the plus or minus is a measure of the randomness of the performance reduction for each reshuffle tried.)

I used the top 4 features (tumor grade, TP53 missense mutation status, patient age, and PTEN missense mutation status) for future training and testing.

I then re-trained and tested the MLP classifier using the selected genetic and demographic data. This resulted, as hypothesized, in increases in prediction accuracy, with scores of 0.44, 0.44, and 0.37 (average: 0.42).

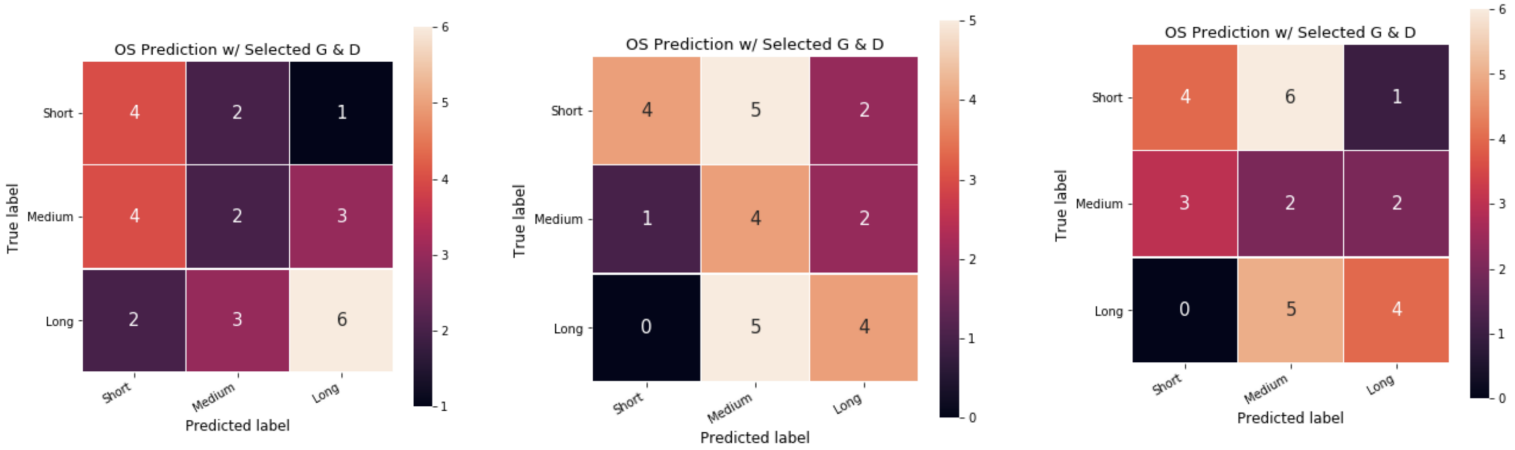


Figure 20. OS classification predictions using selected genetic and demographic data alone.

Next, I constructed my multi-input models by combining the CNN and MLP classifiers using the same methodology as before. I then trained and tested this model using the frame imaging and selected GGD data and obtained accuracy scores of 0.92, 0.70, and 0.92 (average: 0.85), which was a slight increase in accuracy in comparison to the CNN model trained on frame imaging data alone. These multi-input models seemed to have difficulty accurately classifying medium and long-length survival, as shown in the confusion matrices below.

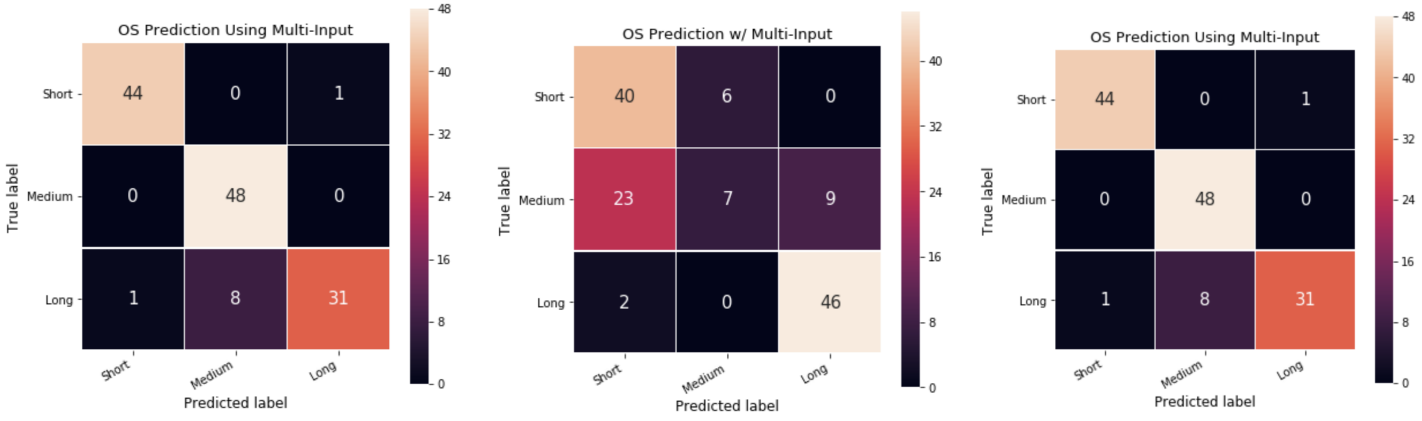


Figure 21. OS classification predictions using both frame imaging and selected GGD data.

Finally, I trained and tested my multi-input model using mask imaging data and the selected GGD data. This resulted in accuracy scores of 0.98, 0.68, and 0.83 (average: 0.83), which was a slight decrease in accuracy in comparison to the CNN model trained on mask imaging data alone. These multi-input models appeared to struggle with short and long-length survival classification, as shown by the second and third matrices below.

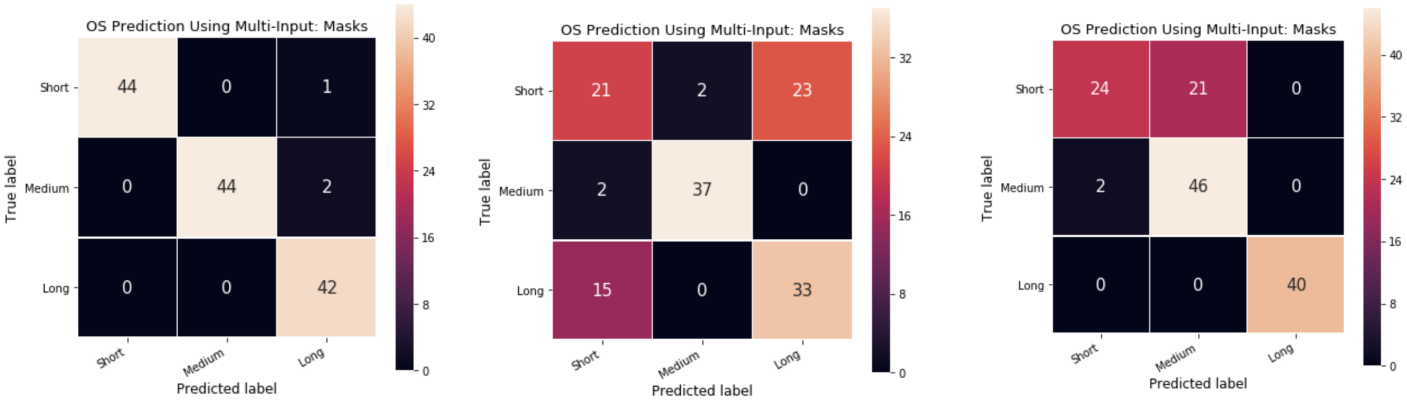


Figure 22. OS classification predictions using both mask imaging and selected GGD data.

Below are examples of the image-trained model's prediction for short, medium, and long survival. The class prediction confidences were 0.52, 0.80 and 0.73, respectively.

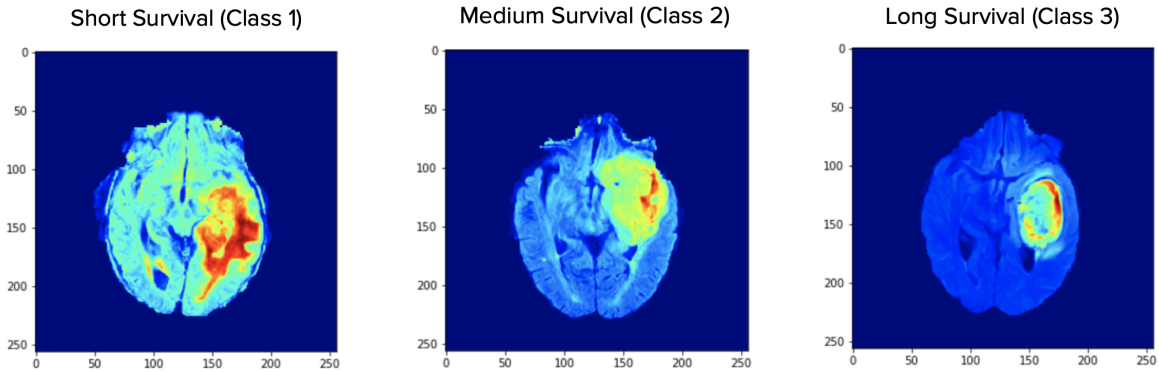
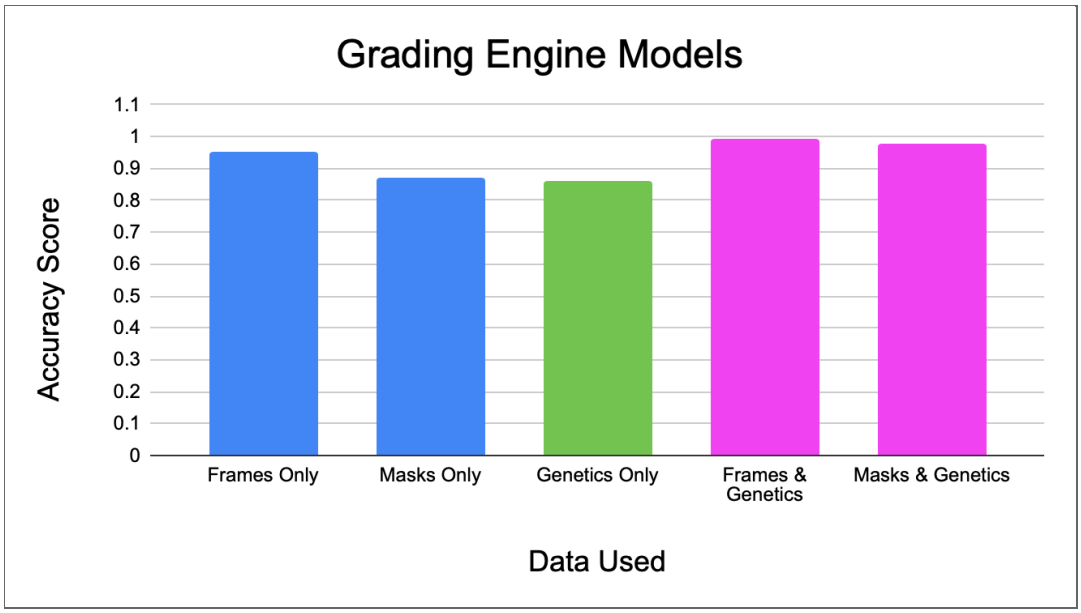


Figure 23. OS-Predicting Engine Predictions (left: short survival; middle: medium survival; right: long survival).

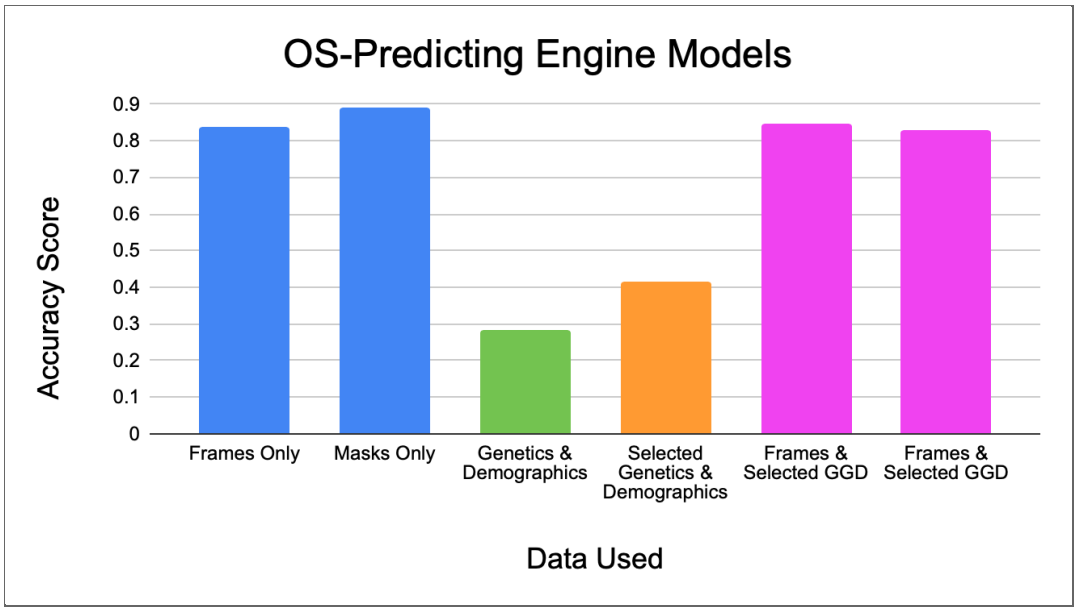
Section 4: Comparison of Predictive Models and Final Pipeline Construction

Below are the accuracy scores of the predictive models that I trained during Grading Engine construction. As visible, there is an increase in accuracy between the multi-input model trained on both frame images and genetic data and the CNNs trained on images/the MLP neural network trained on genetics alone.



Graph 1. Accuracy scores of the various predictive models for grade classification.

Below are the accuracy scores of the predictive models that I trained during OS-Predicting Engine construction. The CNN trained on mask images had the highest accuracy score, and feature selection on the patient genetic and demographic data resulted in an increase in accuracy.



Graph 2. Accuracy scores of the various predictive models for OS classification.

To determine if the construction of the multi-input models as well as feature selection resulted in statistically significant improvements in accuracy, I conducted t-tests for each of the accuracy scores. The p-values are below; the frames multi-input model for grade classification and feature selection for OS prediction both resulted in statistically significant improvements (p-value < 0.05).

	Grade: Frames vs. Frames & Genetics	**Grade:** Masks vs. Masks & Genetics	**OS:** Genetics & Demographics vs. Selected	**OS:** Frames vs. Frames & Selected GGD	**OS:** Masks vs. Masks & Selected GGD
P-Value	0.0314	0.2218	0.0296	0.9463	0.5438

Table 7. P-values of t-tests determining if the multi-input models and feature selection resulted in statistically significant improvements in accuracy.

To construct the final DeepGlio pipeline, I used the most accurate models for the predictive engines. This resulted in the following pipeline: 1) Segmentation Engine consisting of Mask R-CNN, 2) Grading Engine consisting of the Frame Images and Genetics-Trained Multi-Input

Neural Network, and finally, 3) OS-Predicting Engine consisting of the Mask Images-Trained CNN.

Section 5: Genetic and Visual Feature Analysis

To determine correlations between patient genetics and tumor grade as well as patient overall survival, I determined the biological significance of the most important genetic features that were found to be helpful for model predictions by the Random Forest Permutation Shuffling feature selection procedure. Below are the four mutations that were identified as important features; these have all been proven through genetic studies to be linked to glioma development. This computational demonstration that these mutations play a factor in tumor grade and patient survival paves the way for the efficient discovery of more mutations that influence patient prognosis, and thus will help enable more effective treatments and a more thorough understanding of glioma development.

Gene & Chromosome	TP53 (17)	PTEN (10)	EFGR (7)	IDH1 (2)
Mutation Type	Missense	Missense	Missense	Missense
Functional effects of mutation	Faulty tumor suppression (gene expression)	Faulty tumor suppression (AKT/ PKB pathway)	Abnormal/increased cell proliferation	Faulty NADPH regeneration/ production

Table 8. The 4 genetic mutations identified to be correlated with glioma grade and patient survival.

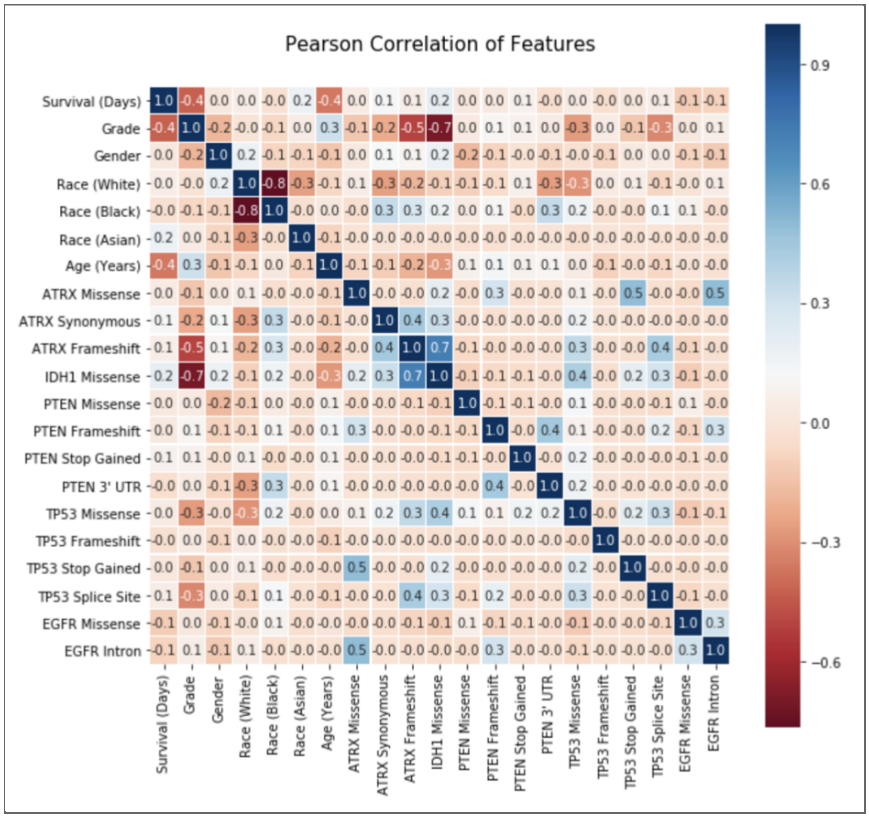


Figure 22. Pearson correlation values between the data features.

I also calculated the Pearson correlation coefficients (indicates the strength of the linear relationship between two variables) between each of the data features, including tumor grade and patient survival.

Through this, I found that the highest correlations were between grade and IDH1 missense mutations, grade and ATRX frameshift mutations, survival and age, and survival and grade. This aligns with the fact that genetic data improved grade classification, while demographic and grade data improved survival prediction.

In addition, I conducted Grad-CAM visual analysis to determine the most important visual features that the CNNs were using in the grading and OS-predicting engines. When making predictions about tumor grade and patient survival, the models seemed to pay the most attention to areas of peritumoral edema and tumor infiltration into surrounding brain tissue, which aligns with the biological idea of the extent of tumor spread directly correlating with glioma aggressiveness and overall patient prognosis.

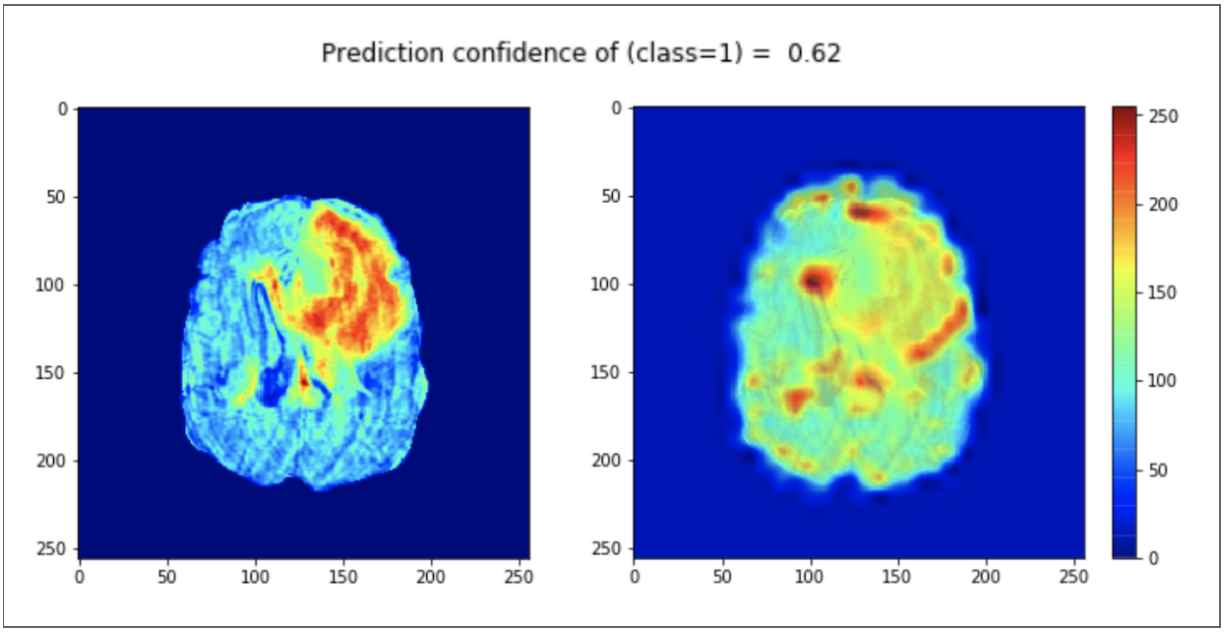


Figure 23. Grad-CAM Analysis Outputs from the Grading Engine

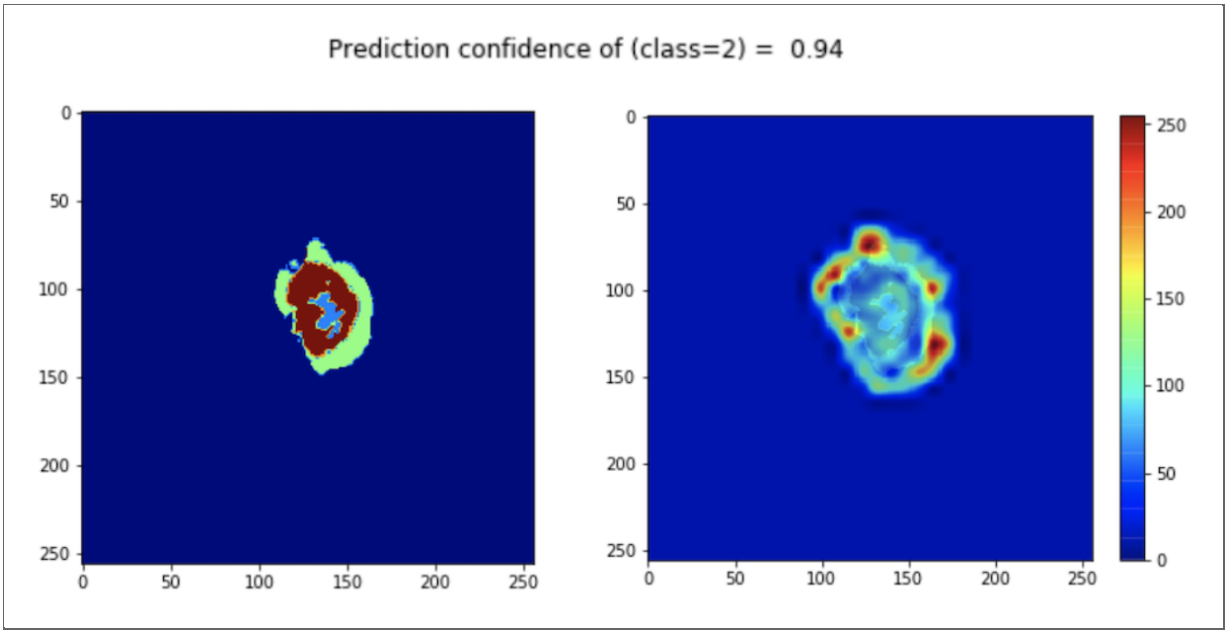


Figure 24. Grad-CAM Analysis Outputs from the OS-Predicting Engine

Discussion and Conclusions

DeepGlio fulfilled my original project objectives: first, to detect and segment gliomas using MRI scan images; second, to classify tumor grade using imaging/segmentation and patient genetic data; and third, to predict patient overall survival length using imaging/segmentation, tumor grade, and patient genetic and demographic data.

The success of the Segmenting Engine demonstrated that the Mask R-CNN model framework, paired with transfer learning and data augmentation techniques, can be used to accurately detect and segment glioma MRI images. The Dice scores achieved by the Segmenting Engine show an overall improvement over existing segmentation methods, as discussed previously, and confirm the viability of deep learning methods for advanced medical image processing.

The Grading Engine produced even more promising results, with near-perfect classification accuracy scores achieved by the imaging and genetic-based models, and perfect scores achieved by the multi-input models. This indicates that patient genetics is a crucial factor for accurate computational classification, which reinforces the most recent WHO glioma grading standards incorporating molecular characteristics in addition to pathologic features.

Likewise, the OS-Predicting Engine also achieved excellent results, with the best-performing model producing an average accuracy score of 0.89. During the engine construction process, feature selection identified 7 key features that were helpful for model prediction. These features were, in order of importance: tumor grade, TP53 missense status, patient age, PTEN missense status, EGFR missense status, IDH1 missense status, and patient race. Through this determination of the mutations that are most correlated with glioma grade and aggressiveness, drugs that target these specific genes and their protein products can be engineered in order to more effectively treat and halt the development of such cancer. Additionally, testing for the presence of these specific genetic variants instead of employing whole genome sequencing procedures can decrease the amount of resources needed to be spent during glioma treatment recommendation procedures.

When the selected feature subset (using the top 4 features) was used for training and testing of the genetic and demographic-based OS prediction model, there was a statistically significant increase in prediction accuracy. This showed that the model benefited from a reduction in data noise and was able to better learn using these particular features. However, while the multi-input frame image model improved slightly upon the original frame image-based model, combining the GGD data with the mask (segmentation) image data to create the multi-input mask prediction model, the OS prediction accuracy decreased. This suggests that, in comparison to grade classification, the particular genetic data used in this project were not as helpful for patient survival prediction. This is further evidenced by the Pearson correlation values between the data features and patient survival values, as only tumor grade and patient age achieved correlation coefficients larger than 0.30 in magnitude. On the other hand, in relation to tumor grade, four mutation data features—ATRX frameshift status, IDH1 missense status, TP53 missense status, and TP53 splice site status—all achieved correlation coefficient values of 0.30 or larger.

These results indicate that while the genetic features were extremely helpful for grade classification, they were less so for OS prediction. Future research needs to be conducted in order to more closely determine how glioma patient prognosis computationally and biologically relates to the genetic factors taken into account in this project and why.

During the predictive engine development, there were several factors that may have impacted my results. First, the dataset size was a key limitation on the accuracy of my models, as I had to utilize strategies such as data augmentation to artificially increase the number of images that I used for model training and testing. In addition, during grade classification, the class balance of patients with low and high-grade gliomas may not have been preserved during random train/test splitting, thus resulting in a class imbalance during model training (perhaps preventing full model recognition of low versus high-grade patterns) and possibly lower accuracy scores.

In future research, I aim to utilize more types of genetic data and further explore how these data can be used to improve predictions of patient prognosis. To maximize model accuracy, I will also implement ensemble learning and hyperparameter optimization methods. In addition, as more imaging and genomic data become readily available in the coming years, computational prediction accuracy will also increase as dataset sizes expand.

Through its novel construction and implementation, DeepGlio enables accurate and efficient computational glioma diagnosis and analysis. Its ability to deliver highly accurate predictions of three critical features—tumor detection and segmentation, tumor grade classification, and patient overall survival—can be utilized to aid clinicians in glioma analysis and treatment recommendation. By taking into account not only imaging results but also patient genetics and demographics, DeepGlio better facilitates precision medicine approaches to glioma diagnosis and treatment.

References

Benson, E., Pound, M. P., French, A. P., Jackson, A. S., & Pridmore, T. P. (2019). Deep Hourglass for Brain Tumor Segmentation. In BrainLes 2018: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, 419-428. Springer. doi:10.1007/978-3-030-11726-9_37

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*. doi: 10.1109/iccv.2017.322

Johns Hopkins University. (2019). Gliomas. Retrieved from https://www.hopkinsmedicine.org/health/conditions-and-diseases/gliomas

Louis, D.N., Perry, A., Reifenberger, G. *et al.* The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary. *Acta Neuropathol* 131, 803–820 (2016). https://doi.org/10.1007/s00401-016-1545-1

Mayfield Clinic. (2018, December). Gliomas, Astrocytoma, Oligodendroglioma, Glioblastoma. Retrieved from https://www.mayfieldclinic.com/pe-glioma.htm

National Cancer Institute. (n.d.). Cancer of the Brain and Other Nervous System - Cancer Stat Facts. Retrieved from https://seer.cancer.gov/statfacts/html/brain.html

Preston, D. C. (2006, November 30). MRI Basics. Retrieved from https://casemed.case.edu/clerkships/neurology/Web Neurorad/MRI Basics.htm

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *2017 IEEE International Conference on Computer Vision (ICCV)*. doi: 10.1109/iccv.2017.74

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, *15*(1). doi: 10.1186/s12880-015-0068-x

Tamimi AF, Juweid M. Epidemiology and Outcome of Glioblastoma. In: De Vleeschouwer S, editor. Glioblastoma [Internet]. Brisbane (AU): Codon Publications; 2017 Sep 27. Chapter 8. Available from: https://www.ncbi.nlm.nih.gov/books/NBK470003/doi: 10.15586/codon.glioblastoma.2017.ch8

Yu, N., Yu, Z., Gu, F., Li, T., Tian, X., & Pan, Y. (2017). Deep Learning in Genomic and Medical Image Data Analysis: Challenges and Approaches. *JIPS, 13*, 204-214.

Image References

Figure 1: Tsang, S.-H. (2019, March 20). Review: DRN - Dilated Residual Networks (Image Classification & Semantic Segmentation). Retrieved from https://towardsdatascience.com/review-drn-dilated-residual-networks-image-classification-semantic-segmentation-d527e1a8fb5

Figure 2: Garza, G. (2019, January 8). Mask R-CNN for Ship Detection & Segmentation. Retrieved from https://towardsdatascience.com/mask-r-cnn-for-ship-detection-segmentation-a1108b5a083