

CALIFORNIA STATE UNIVERSITY SAN MARCOS

THESIS SIGNATURE PAGE

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

THESIS TITLE: Improved Glioma Grading using Deep Learning Techniques

AUTHOR: SAJJAD SABAHUDDIN

DATE OF SUCCESSFUL DEFENSE: 03 / 27 / 2024

THE THESIS HAS BEEN ACCEPTED BY THE THESIS COMMITTEE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN
COMPUTER SCIENCE.

DR. SREEDEVI GUTTA
THESIS COMMITTEE CHAIR

Sreedevi Gutta
SIGNATURE

05/03/2024
DATE

DR. MUHAMMED LUTFOR RAHMAN
THESIS COMMITTEE MEMBER

Md Lutfor Rahman
SIGNATURE

05/08/2024
DATE

THESIS COMMITTEE MEMBER

SIGNATURE

DATE

THESIS COMMITTEE MEMBER

SIGNATURE

DATE

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor, Dr. Sreedevi Gutta, for her invaluable support, guidance, and expert mentorship throughout every stage of my research project. From helping me write intricate lines of code to streamlining the paper writing process, her commitment and vision have been instrumental in shaping the trajectory of my research. Moreover, her patient explanations and comprehensive teachings of core concepts have equipped me with a solid foundation upon which to build my academic journey.

I am also greatly thankful to Dr. Muhammed Lutfor Rahman, whose essential feedback and thoughtful evaluation have enhanced the development of my research paper and project. Furthermore, I extend my sincere appreciation to the California State University San Marcos for granting me with the opportunity to embark on this fulfilling research project. Their assistance and resources have been vital in facilitating my academic growth and exploration.

On a final note, I am profoundly grateful to all those who have contributed to this endeavor, whether through silent encouragement or direct collaboration. Your contributions have played a fundamental role in shaping the outcome of this work, and for that, I am truly thankful.

TABLE OF CONTENT

| | |
|--|-----------|
| <i>ABSTRACT</i> | 5 |
| <i>Section 1: INTRODUCTION</i> | 6 |
| <i>Section 2: RELATED WORK</i> | 7 |
| <i>Section 3: DATASET DESCRIPTION</i> | 9 |
| <i>Section 4: METHODS</i> | 12 |
| <i>4-1: Machine Learning</i> | 12 |
| <i>4-2: Deep Learning Model</i> | 17 |
| <i>4-2-1: Data pre-processing</i> | 17 |
| <i>4-2-2: ResNet</i> | 18 |
| <i>4-2-3: MobileNet</i> | 19 |
| <i>4-2-4: DenseNet</i> | 19 |
| <i>4-2-5: EfficientNet</i> | 19 |
| <i>4-2-6: Proposed Model - VGG16 with Attention</i> | 20 |
| <i>4-2-7: Fine Tuning for Deep Learning</i> | 22 |
| <i>Section 5: RESULTS</i> | 23 |
| <i>5-1: Performance Measures</i> | 23 |
| <i>5-2: Machine Learning</i> | 24 |
| <i>5-3: Deep Learning model</i> | 29 |
| <i>Section 6: DISCUSSION</i> | 34 |
| <i>Section 7: CONCLUSION</i> | 35 |
| <i>Section 8: FUTURE WORK</i> | 36 |
| <i>Section 9: REFERENCES</i> | 37 |

LIST OF THE FIGURES

| | |
|--|----|
| Figure 1: Scans of a patient with High Grade Glioma..... | 10 |
| Figure 2: Scans of a patient with Low Grade Glioma.. | 10 |
| Figure 3: Flowchart of Machine Learning..... | 11 |
| Figure 4: Demonstration of how radiomic features are calculated | 14 |
| Figure 5: Effectiveness of the DL models. | 20 |
| Figure 6.1: Detailed illustration of the proposed network architecture. | 21 |
| Figure 6.2: Illustration of the gray blocks from the two attention modules..... | 24 |
| Figure 7: Confusion Matrix | 23 |
| Figure 8: Bar Plots | 24 |
| Figure 9: Confusion Matrix of Training. | 26 |
| Figure 10: Confusion Matrix of Testing. | 27 |
| Figure 11: Feature Importance of all the features arranged based on importance..... | 28 |
| Figure 12: Pie chart of the Feature Importance | 28 |
| Figure 13: Bar Plot for Accuracy of the Models..... | 32 |
| Figure 14: Bar Plot for Precision of the Models | 30 |
| Figure 15: Bar Plot for Recall of the Models..... | 33 |
| Figure 16: Bar Plot for F1-score of the Models..... | 31 |
| Figure 17: Visualizations from 6 different test patients correctly classified and focus on the tumor. | 32 |
| Figure 18: Visualization of LGG patient misclassified as HGG. | 33 |

LIST OF THE TABLES

| | |
|--|----|
| Table 1: Summary of Machine Learning Data. | 12 |
| Table 2: Summary of Deep Learning Data | 18 |
| Table 3: ML results of all the model..... | 24 |
| Table 4: Top 28 important features from the Radiomic features..... | 29 |
| Table 5: Results from Deep Learning Model | 30 |

Thesis Title: Improved Glioma Grading using Deep Learning Techniques

Supervisor: Dr. Sreedevi Gutta

ABSTRACT

BACKGROUND AND PURPOSE

Gliomas are a sort of primary brain tumor that originates from glial cells. These cells are accountable for supporting the central nervous system of the human body. These tumors vary widely based on their nature, they could be either malignant or aggressive. Glioma grading is an essential part of diagnosis and treatment planning, as it can give critical information regarding the tumor's characteristics and its behavior. The main objective of this study is to use MR images and develop an attention-based deep learning model to detect glioma and visually locate the tumor. Additionally, machine learning models are deployed on the radiomic features, which are extracted from the MRI scans.

MATERIALS AND METHODS:

The dataset includes a total of 285 patients with both high- and low-grade gliomas. The preprocessing steps applied on this dataset include interpolation to a standardized resolution of 1 cubic millimeter, alignment to a common anatomical template, and finally skull stripped. A pretrained VGG16 model with 2 attention modules is used for grade prediction. These attention modules extract intermediate features from the main architecture and predict the area of tumor by highlighting it. The proposed model's performance is evaluated with other pretrained models like ResNet, DenseNet, MobileNet, and EfficientNet. Along with the deep learning models, popular machine learning models are also used to evaluate the performance.

RESULTS

The proposed model was able to achieve an f1-score of 91.18%, demonstrating its robustness and capability to grade the tumor. Furthermore, the attention maps enabled detailed visualization of the tumor regions, enhancing the interpretability of the model's predictions. The proposed model produced almost the same results as the pretrained model ResNet50 but with additional visualization of the tumor. In comparison to the machine learning model, we can notice an improvement of about 6% in F1-score, from the top-performing machine learning model, i.e. random forest.

CONCLUSION:

To conclude, the proposed model, leveraging its fundamental capability to automatically learn features, has proven remarkable effectiveness in glioma tumor detection and its classification. The integration of two attention maps into the VGG16 pretrained model has enhanced its capability of precisely focusing and detecting the tumor region, a feature that was

not available in earlier models. This holds a promising advancement in disease diagnosis and medical imaging.

Section 1: INTRODUCTION

A brain tumor is a condition where human body cells grow aberrantly and start harming the spinal cord or brain [1]. According to the survey conducted by World Health Organization (WHO), the tenth most common cause of death, for both men and women, is brain or central nervous system cancer [2]. Since 2000, 8 June has been marked as world brain tumor day, where people are educated, and awareness is spread about brain tumors [1]. Glioma is a type of brain tumor which can be found in at least 5-10 adults per 100,000 people every year [3]. These are malignant in nature [3]. Based on the molecular properties and histology, the WHO categorized gliomas into four grades, ranging from I to IV [1]. Here the I and II grades fall into the low-grade category [1]. Likewise, III and IV grade comes under high-grade category [1]. The survival rate for a patient with a high-grade tumor is low [4, 5]. Structured treatment course can be formed by precise and early diagnosing of the tumor. This will help in saving not just thousands but millions of lives [5].

The initial signs of brain tumor can be drawn through image modalities, like computerized tomography (CT) and magnetic resonance imaging (MRI), and through neurological examination [5, 6]. Moreover, tumor grade can be determined by performing tests such as biopsy and biomarker [7]. The CT scans capture the image of a body by emitting ionized radiation, which is harmful to the body [8]. On the other hand, MRI scans capture high-quality images of the body without emitting radiation, making it safe. [8, 9]. MRI produces several sequences like T1-weighted contrast enhancement (T1ce), T1-weighted (T1), T2-contrast (T2c), T2-weighted (T2), and fluid-attenuated inversion recovery (FLAIR) [9]. Each sequence has a unique tissue structure because the physical method of capturing those sequences is distinct [8, 10, 11]. Hence, it is expected that each one of these will yield a different output in brain tumor classification [7].

Accurate determination of tumor grade is important for treatment planning. The current gold standard way of determining tumor grade is through biopsy [12, 13, 14]. Biopsy is an invasive process, where a small part of the tissue is extracted and visualized under the microscope. Hence, biopsy is invasive and expensive [14]. Thus, a computer aided diagnosis (CAD) tool, which is fast, automated, and discreet, is needed for determination of tumor grade [12].

With the latest innovations and improvements of Artificial Intelligence (AI) methods, numerous CAD tools were developed for various medical applications [15, 16, 17]. The two significant branches of effective AI algorithms are Deep Learning [18, 19] and Machine Learning [20, 21]. A major drawback with ML techniques is the need for explicit extraction of features [22, 23]. Convolutional neural networks (CNN) on the other hand automatically extract the most relevant features from scanned images [24]. These algorithms extract those minute details which are not detectable through the naked eye [24]. Consequently, deep learning (DL) is

most used in medical image analysis for problems like segmentation [25], image registration [26], and classification [27]. The main motive of this thesis is to develop a non-invasive and an accurate DL model to classify whether a patient has a High-Grade Glioma (HGG) or Low-Grade Glioma (LGG).

Section 2: RELATED WORK

In the prior research paper “Improved Glioma Grading Using Deep Convolutional Neural Networks” [3] the convolutional neural network (CNN) model achieved an accuracy of 87%, while the ML algorithms support vector machine (SVM), random forest (RF), and gradient boosting (GB) got 56%, 58%, and 64% accuracies respectively [3]. More details about preprocessing and results of other metrics can be found in Gutta et al. 2021 [3]. This paper has performed multiclass classification problem (grade I vs II vs III vs IV) with few additional steps in data preprocessing [3]. As per the result in this paper [3], confusion matrices displayed that CNN model was better than SVM, RF, and GB models which were trained with radiomic features. It also depicts that it is 23% more accurate than the best performing model i.e., GB [3]. Other experiments where these models were compared using the performance metrics; recall, precision, and F1 score illustrates that CNN performed better, implying that learned features are indeed valuable in increasing the accuracy of the prediction [3].

Another research article “A Novel System for Precise Grading of Glioma” [28] developed a CAD system which predicts the grade ranging from I to IV. A total of 99 patient with gliomas were considered, including 49 male and 50 female [28]. Their proposed system utilizes three different MRI scans sequences, namely, diffusion-weighted (DW-MR), T2-MR known as FLAIR, and contrast-enhanced T1-MR [28]. According to Abdel [28], “These sequences are used to extract the following features: (i) functional features by estimating voxel-wise apparent diffusion coefficients (ADC) and contrast-enhancement slope, (ii) first and second orders textural features by constructing histogram, gray-level run length matrix (GLRLM), and gray-level co-occurrence matrix (GLCM), and (iii) morphological features based on constructing the histogram of oriented gradients (HOG) and estimating the glioma volume”. To get the final set of important features, the above features were integrated all together and treated using a Gini impurity-based selection [28]. The final set of features are then given as input to a multi-layer perceptron artificial neural networks (MLP-ANN) classification model to produce the final grade [28]. The evaluation was performed using the leave-one-subject-out (LOSO) and cross-validation approach, k-fold stratified [28]. The model’s overall accuracy is 95.8% with k values as 5 [28]. The proposed MLP-ANN system outperformed the random forest and support vector machine [28].

In another research article “Brain Tumor and Glioma Grade Classification Using Gaussian Convolutional Neural Network” [29] a CAD system is proposed which predicts the Brain Tumor type and a Gaussian Convolutional Neural Network (GCNN) model to predict glioma grade. They have considered two datasets, where one dataset is utilized to predict the type of tumor, including glioma-tumor, pituitary-tumor, and meningioma-tumor [29]. The second

dataset is used to predict the grade of glioma, i.e., grade II, III, and IV [29]. The number of patients are 233 and 73 with 3064 and 516 T1-weighted complexity images in each dataset respectively [29]. The proposed model marked 99.8% and 97.14% accuracies for the two datasets [29]. The preprocessing involves two steps, first the data is passed through a gaussian imaging filter, then it is passed through a 16-layer based network [29]. According to its authors, data augmentation played a vital role in delivering better outcomes, despite the dataset being small [29].

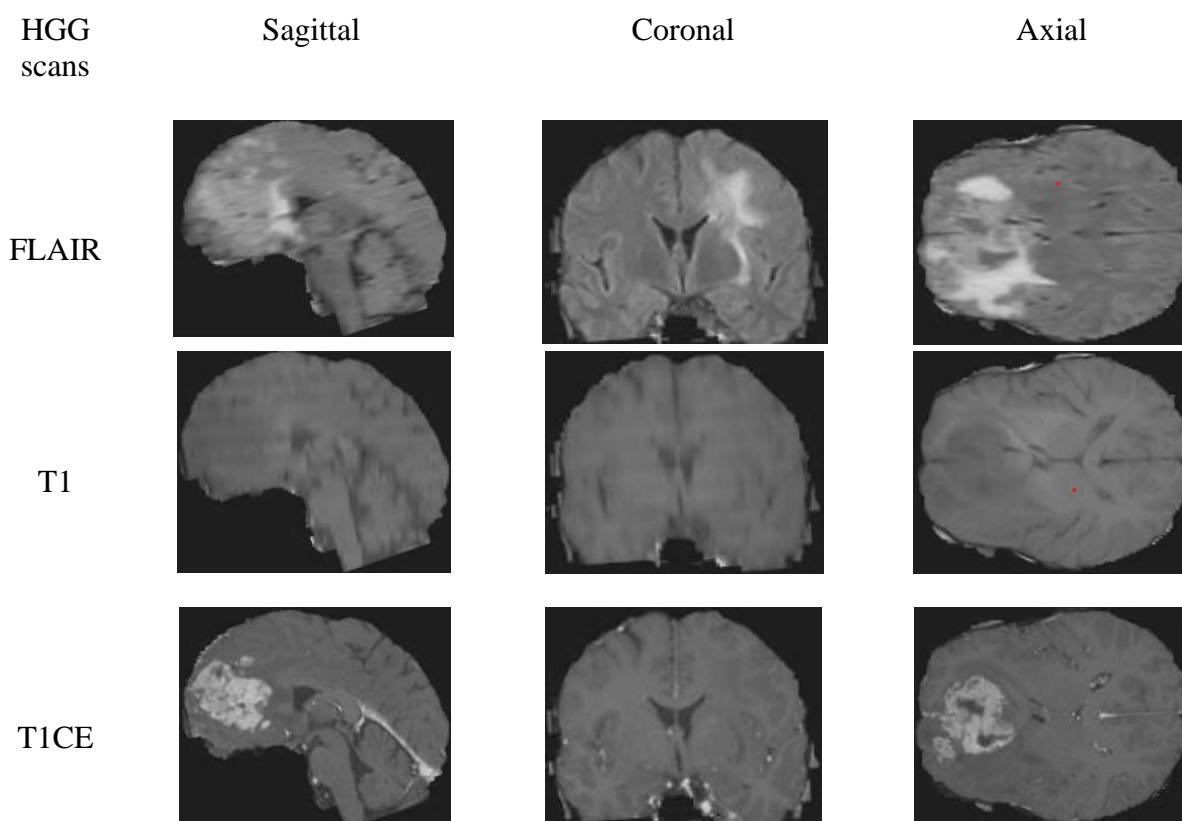
In another research article “Role of Ensemble Deep Learning for Brain Tumor Classification in Multiple Magnetic Resonance Imaging Sequence Data” [30], a CAD tool is proposed to classify the glioma grade between low-grade and high-grade. The datasets considered in this article are derived from three different MRI sequences; FLAIR, T2-weighted (T2W), and T1-weighted (T1W) [30]. The five convolutional neural networks implemented for tumor classification are: VGG16 [31], GoogleNet [32], AlexNet [33], ResNet18 [34], and ResNet50 [35]. An ensemble model was developed using the majority vote from the above-mentioned models, this will certainly be more consistent and will yield better results than any individual model [30]. K-fold cross validation approach, with $k=5$, was implemented in both training and testing [30]. The proposed ensemble classifier achieved an accuracy of 98.88% with FLAIR dataset, the other datasets T2W and T1W achieved an accuracy of 97.98% and 94.75% respectively [30].

In this thesis report, we have developed a CAD tool which will classify between the low grade and the high grade. The key difference from other research articles are as follows:

1. Determine the significance of radiomic features with machine learning models.
2. All the prior works require an explicit segmentation step to locate the tumor region. Note that this pre-processing step of locating the tumor is challenging. In this thesis, the goal is to propose a deep learning model that helps in eliminating the segmentation step by using an attention layer that allows you to focus on the tumor regions automatically.

Section 3: DATASET DESCRIPTION

The dataset was collected from “Multimodal Brain Tumor Segmentation Challenge 2018” hosted on a website of Perelman School of Medicine, University of Pennsylvania [36]. The following three preprocessing steps were considered before publishing the data online. Firstly, it is listed using the same anatomical template. Secondly, it is interpolated to the same resolution, which is 1 mm^3 . Lastly, it is skull-stripped, to remove the skull. This dataset has multimodal 3D brain MRI scans of 285 patients. Out of 285 patients, 210 belong to the glioblastoma (GBM/HGG) category and the rest 75 belong to LGG category. The scans are in Neuroimaging Informatics Technology Initiative (NIfTI) format, which has an extension of “.nii”. Each patient has 4 sequences, FLAIR, T1, T1CE, and T2 and each sequence has three types of orientations: namely axial plane, coronal plane, and sagittal plane [37]. When an MRI scan is viewed from top to bottom, that plane is referred to as axial plane [37]. Whereas the plane from front to back is called coronal plane and from side-to-side plane is called sagittal plane [37]. In our case, each sequence has a shape of (240, 240, 155). Where the first 240 slices belong to sagittal plane and second 240 slices belong to coronal plane. The axial plane has 155 slices.



T2

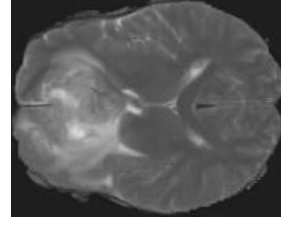
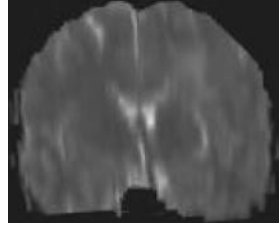
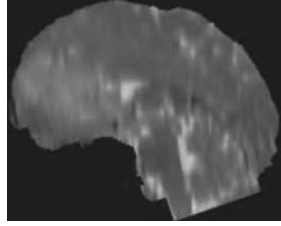


Figure 1: Scans of a patient with High Grade Glioma. Each row corresponds to a sequence. The four sequences are FLAIR, T1, T1CE, and T2. Each column corresponds to a plane. The first, second, and third columns correspond to sagittal, coronal, and axial planes respectively.

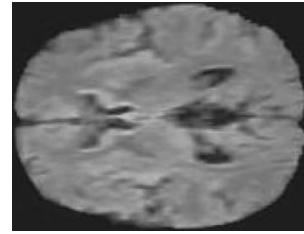
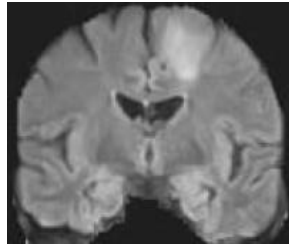
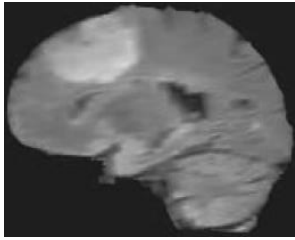
LGG
scans

Sagittal

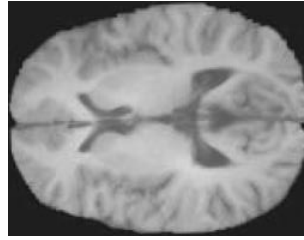
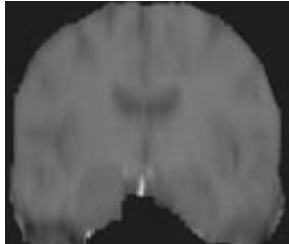
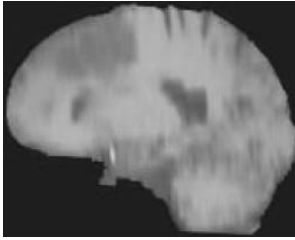
Coronal

Axial

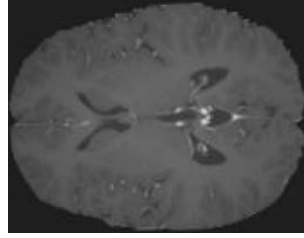
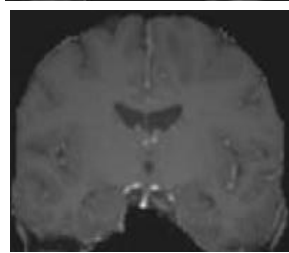
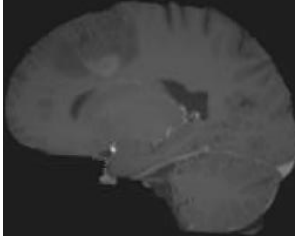
FLAIR



T1



T1CE



T2

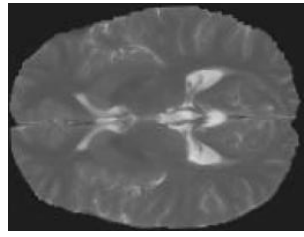
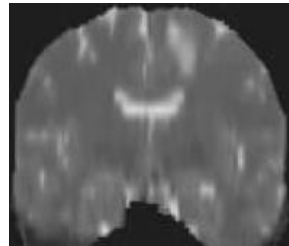
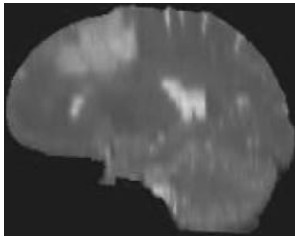


Figure 2: Scans of a patient with Low Grade Glioma. Each row corresponds to a sequence. The four sequences are FLAIR, T1, T1CE, and T2. Each column corresponds to a plane. The first, second, and third columns correspond to sagittal, coronal, and axial planes respectively.

Four rows of figures 1 and 2 demonstrate four sequences, FLAIR, T1, T1CE, and T2. As mentioned earlier each sequence will have a different tissue structure [8, 10, 11]. This will help

in locating the tumor from different perspectives. According to Tiwari [30], “FLAIR data sequence produced the best performance using the majority voting classifier”. The columns represent the three different planes of each sequence.

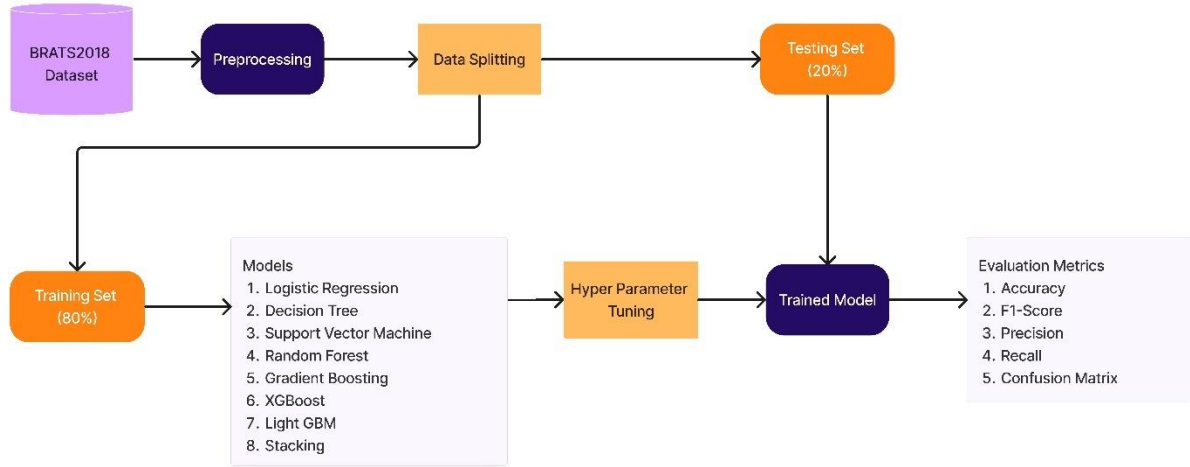


Figure 3: Flowchart of Machine Learning. BraTS 2018 dataset was used for evaluating machine learning and deep learning models. Preprocessing is performed soon after the dataset is loaded. Followed by splitting the data into training and testing sets. Training set is used to train all the 7 models. Then hyperparameter tuning is performed on those trained models. Finally, predictions are made on the testing set and based on the evaluation metrics the quality of the models are determined.

Section 4: METHODS

4-1: Machine Learning

Machine learning models require explicit extraction of radiomic features [38]. The extracted features are then used by ML models to determine the tumor grade.

Radiomic Feature Extraction using PyRadiomics Package

For the Machine Learning models, we have used PyRadiomics [38], an open-source python package to extract quantitative metrics, also called radiomics features, from the image dataset. Radiomic features capture lesion and tissue characteristics like shape and heterogeneity [39]. They are either considered solely or in combination with genomic, histologic, demographic, or proteomic data, for solving clinical problems [39]. Studies have depicted that radiomic features have high correlations with heterogeneity indices present at the cellular level [39, 40]. Biopsies only concentrate at one anatomic site and capture the heterogeneity of a tiny portion of tumor only [41]. On the other hand, radiomic features cover the entire tumor volume while capturing heterogeneity. Expectedly, radiomic features are linked with tumor aggressiveness [41]. It is being reported that radiomic features are of great help in predicting the clinical endpoint like treatment response and survival and to be directly linked to proteomic, transcriptomic, or genomic characteristics [41, 42, 43].

We have extracted radiomic features from each sequence and it resulted in 107 Radiomic features, which includes 14 shape-based features, 18 first-order statistics-based features, and other commonly used texture features like gray-level co-occurrence matrix (GLCM) (24 features), gray-level dependence matrix (GLDM) (14 features), gray-level run length matrix (GLRLM) (16 features), gray-level size zone matrix (GLSZM) (16 features), and neighboring gray tone difference matrix (NGTDM) (5 features).

The extracted radiomic features were saved in a csv file that can be used later for training ML models. A total of 107 features were extracted for each sequence and with a total of 4 sequences present for each patient, there are a total of 428 features extracted. The data is split into training and testing in the ratio of 80:20. Thus the training set consists of 168 patients with HGG and 60 patients with LGG. The testing set consists of 42 patients belonging to HGG and 15 patients belonging to LGG.

| Number of samples | HGG | LGG | Rows |
|-------------------|------------------------|----------------------|------|
| Training (80%) | $0.8 \times 210 = 168$ | $0.8 \times 75 = 60$ | 228 |
| Testing (20%) | $0.2 \times 210 = 42$ | $0.2 \times 75 = 15$ | 57 |
| Total | 210 | 75 | 285 |

Table 1: Summary of Machine Learning Data. The data is split into training and testing in the ratio of 80:20. There are 228 patients in the training set (168 belonging to HGG and 60 belonging to LGG) and 57 patients in the testing set (42 belonging to HGG and 15 belonging to LGG).

Shape-Based Features

The shape-based features are used to define geometric features of regions of interest (ROIs) [44]. Shape based features are easy to interpret when compared to other features because they have 2D or 3D diameters, axes, and their ratios [44]. The 14 shape features extracted from the sequences are: Elongation, Flatness, Least Axis Length, Maximum2DDiameterColumn, Maximum2DDiameterRow, Maximum2DDiameterSlice, Maximum3DDiameter, Mesh Volume, Minor Axis Length, Sphericity, Surface Area, Surface Volume Ratio, and Voxel Volume [38].

First-Order Statistics-Based Features

The simplest statistical descriptors are completely relied on the global-level histogram [45, 46]. These include gray-level minimum, maximum, mean, variance, and percentiles [45, 46]. Since these features are based on single voxel or single pixel analyses, they are labeled as first-order statistics-based features [45, 46]. The 18 first-order features are 10Percentile, 90Percentile, Energy, Entropy, Interquartile Range, Kurtosis, Maximum, Mean Absolute Deviation, Mean, Median, Range, Skewness, Robust Mean Absolute Deviation, Root Mean Squared, Total Energy, Uniformity, and Variance [38].

Gray-Level Cooccurrence Matrix (GLCM)

First defined by Shanmugam et al. [47], the gray-level co-occurrence matrix is a gray-level histogram of second order [47]. It captures spatial relations among the voxel or pixel pairs, which are already defined with gray-level intensities [47]. These are in different directions (13 different directions in a 3D analysis or diagonal, vertical, or horizontal for 2D analysis). Along with the predefined intensities, the distance among the voxels or pixels are also predefined (refer figure 2 for radiomic features calculations) [44]. The 24 GLCM features extracted are Autocorrelation, Cluster Prominence, Cluster Shade, Cluster Tendency, Contrast, Correlation, Difference Average, Difference Entropy, Difference Variance, ID, Idm, Idmn, Idn, Imc1, Imc2, Inverse Variance, Joint Average, Joint Energy, Joint Entropy, MCC, Maximum Probability, Sum Average, Sum Entropy, and Sum Squares [38].

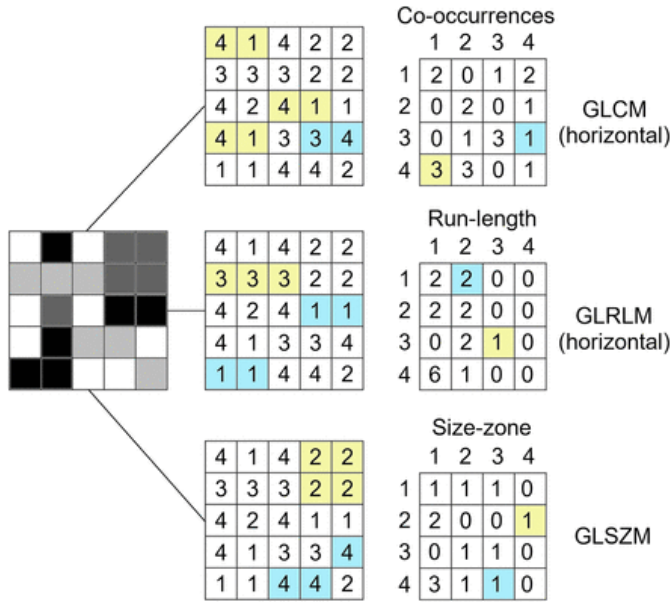


Figure 4: Demonstration of how radiomic features are calculated [44]. While GLSZM is dependent on the neighboring pixel areas with same gray-level, the GLRLM is dependent on the runs, and the GLCM is dependent on the pixel pairs (in this case, distance of the interpixel = 0) [44]

Gray-Level Dependence Matrix (GLDM) (14 features)

This feature is responsible for calculating the gray-level dependencies of the image scan. According to Pyradiomics, “Gray-level dependency can be defined as number of connected voxels with a certain distance are dependent on the center voxel” [48]. The 14 GLDM features extracted are Dependence Entropy, Dependence Non-Uniformity, Dependence Non-Uniformity Normalized, Dependence Variance, Gray Level Non-Uniformity, Gray Level Variance, High Gray Level Emphasis, Large Dependence Emphasis, Large Dependence High Gray Level Emphasis, Large Dependence Low Gray Level Emphasis, Low Gray Level Emphasis, Small Dependence Emphasis, Small Dependence High Gray Level Emphasis, and Small Dependence Low Gray Level Emphasis [38].

Gray-Level Run Length Matrix (GLRLM) (16 features),

The GLRLM, as defined by Galloway [49], provides data regarding the dimensional distribution of runs of successive pixels having the same gray level. They are either in 2 or 3 dimensions with one or more directions [49]. The 16 GLRLM features extracted are Gray Level Non-Uniformity, Gray Level Non-Uniformity Normalized, Gray Level Variance, High Gray Level Run Emphasis, Long Run Emphasis, Long Run High Gray Level Emphasis, Long Run Low Gray Level Emphasis, Low Gray Level Run Emphasis, Run Entropy, Run Length Non Uniformity, Run Length Non Uniformity Normalized, Run Percentage, Run Variance, Short Run Emphasis, Short Run High Gray Level Emphasis, and Short Run Low Gray Level Emphasis [38].

Gray-Level Size Zone Matrix (GLSZM) (16 features)

As defined by the Meyer [50], the GLSZM follows the same principle as the GLRLM, the difference comes while forming the matrix. The GLSZM considers the number of interconnected neighboring voxel or pixel groups having the same gray level [50]. The 16 GLSZM features extracted are Gray Level Non-Uniformity, Gray Level Non-Uniformity Normalized, Gray Level Variance, High Gray Level Zone Emphasis, Large Area Emphasis, Large Area High Gray Level Emphasis, Large Area Low Gray Level Emphasis, Low Gray Level Zone Emphasis, Size Zone Non-Uniformity, Size Zone Non-Uniformity Normalized, Small Area Emphasis, Small Area High Gray Level Emphasis, Small Area Low Gray Level Emphasis, Zone Entropy, Zone Percentage, and Zone Variance [38].

Neighboring Gray Tone Difference Matrix (NGTDM) (5 features).

The NGTDM, first proposed by Amadasun [51], calculates the sum of differences between the mean gray level of adjacent voxels or pixels, whose distance is already defined, and the gray level of a voxel or pixel. The key features of NGTDM are complexity, busyness, and coarseness. The 5 features extracted in our case are Busyness, Coarseness, Complexity, Contrast, and Strength [38].

For feature scaling we have used the Standard scaler. This will standardize the features by subtracting the mean with the training samples and scaling to unit variance. The formula is $z = (x - u)/s$ where 'x' are the training samples, 'u' is the mean, and 's' is the standard deviation. And with regards to null values, there are none.

ML models on these radiomics

The ML models we have used are Logistic Regression, Decision Tree, Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GB), XGBoost, Light Gradient-Boosting Machine (LGBM), and Stacking. All the ML models are accessible through scikit learn toolbox except LGBM and XGBoost, which are available as python packages.

i. Logistic Regression

Logistic Regression falls under the classification category in machine learning [52]. It is also called Logit Regression [52]. This algorithm calculates probabilities of each instance belonging to a specific class [52]. In our case, it is the probability that these features belong to HGG or LGG. If the calculated probability is higher than 50 percent then the model's prediction is, that instance belongs to label 1 class, which is the positive class [52]. If less than 50 percent, then it belongs to label 0 class, which is negative class [52]. Hence, this is a binary classifier [52]. The scikit learn offers the Logistic Regression class, and we can import it using the following python code "from sklearn.linear_model import LogisticRegression" [53].

ii. Decision Tree

Decision tree can be used to perform both regression and classification tasks [52]. It has a tree structure which starts from the root node and connects through various branches and ends at

leaf nodes [52]. At each node a decision is made based on the conditions [52]. Non-linear datasets can be easily interpreted and visualized using this algorithm [52]. The scikit learn offers the Decision Tree Classifier class, and we can import it using the following python code “from sklearn.tree import DecisionTreeClassifier” [54].

iii. Support Vector Machine (SVM)

This algorithm works by increasing the dimension of the data to better categorize the data [52]. A decision boundary is drawn which separates the two classes [52]. Based on this boundary, the closest training instances are separated as far as possible [52]. That separator with widest possible street is called as large margin classification [52]. Feature scaling is necessary because SVMs are sensitive to them [52]. SVM is a good fit for small and medium-sized datasets [52]. The scikit learn offers the Support Vector Classifier class, and we can import it using the following python code “from sklearn.svm import SVC” [55].

iv. Random Forest (RF)

This ML model works on the concept of bagging [52]. In bagging, a group of models, in random forest case it is decision tree, is trained on unique subsets [52]. The final output is produced by averaging all the decision trees together [52]. The main advantages of using random forest are it produces high accuracy, versatile, robust, and scalable [52]. The random forest adds additional randomness to the growing trees [52]. While splitting a node, it looks for the best parameters among a random subset of parameters, rather than searching for the best parameters [52]. The scikit learn offers the Random Forest Classifier class, and we can import it using the following python code “from sklearn.ensemble import RandomForestClassifier” [56].

v. Gradient Boosting (GB)

Gradient Boosting is a type of boosting algorithm, where many models work sequentially, and each model will rectify the errors made by the previous model [52]. Even if the model selected is weak it will yield better results, because the algorithm will be fitting the new model to the residual errors made by the previous model [52]. The base model for the gradient boosting is also decision trees [52]. Gradient boosting does not aggregate the results after all the models give their output, in its place the results of each model are aggregated [52]. The scikit learn offers the Gradient Boosting Classifier class, and we can import it using the following python code “from sklearn.ensemble import GradientBoostingClassifier” [57].

vi. LightGBM

LightGBM is also a gradient boosting algorithm, where the splitting of the tree is done leaf-wise [52]. Other boosting algorithms split the data level-wise or depth-wise [52]. The training speed and efficiency of this algorithm is higher than others because it consumes less memory [52]. This algorithm is capable of handling large datasets [52]. The base estimator for LightGBM is also a decision tree [52]. This package can be installed using the python command “pip install lightgbm” and can be imported using the code “import lightgbm as lgb” [58]

vii. XGBoost

XGBoost, short for Extreme gradient boosting, is a more regularized form of GB [52]. The model regularization capabilities are increased with the help of L1 & L2 normalization [52]. XGBoost performs better than other ensemble algorithms like random forest when the dataset has a class imbalance [52]. This package can be imported using the code “from xgboost import XGBClassifier” [59].

viii. Stacking

While boosting is mainly focused on decreasing the bias, and bagging is mainly focused on decreasing the variance, stacking on the other hand aims to increase accuracy of the predictions [60]. While boosting and bagging combine weak machine learning models to produce better results, stacking combines the best machine learning algorithms [60]. The final estimator takes the input of three predictions from three different models and makes it ultimate prediction [60]. We have tried three different final estimators namely: Logistic Regression, Random Forest, and Support Vector Regressor. Based on the results produced we chose logistic regression because it produced the highest accuracy. The scikit learn offers the stackingclassifier class, and we can import it using the following python code “from sklearn.ensemble import StackingClassifier” [31].

ix. Fine Tuning for Machine Learning

For the Fine Tuning of the models, we have chosen RandomizedSearch CV [61]. In this search only selected random combination of hyper parameters are considered and tests are conducted on them to find the best model. Cross validation is splitting the data into folds, either 5 or k-fold, and training the model using one set and validating it with another. As for the stacking algorithm, we have tried three different final estimators: Random Forest Regressor, Logistic Regression, and SVM.

4-2: Deep Learning Model

4-2-1: Data pre-processing

As discussed earlier in the dataset description section, the whole nifty dataset consists of 210 HGG patients and 75 LGG patients. Here each sequence has a shape of 240 x 240 x 155. To maintain the same proportion of classes in both the training and testing datasets as in the original dataset. We have considered 80% of the samples from both HGG and LGG to form the training dataset. This training dataset contains 168 HGG patients and 60 LGG patients. This gives a total of 228 training patients. For this research, we have decided to not consider the t1 sequence because of the limited information present in this sequence. The shape for one patient looks like this: (155, 240, 240, 3), where the 3 represents three different sequences. When all 155 images are visualized, it is noted that starting 50 and the last 50 slices had only partial image of the brain, and hence was removed. Now the shape of each patient is (56, 240, 240, 3). When all 228 patients are considered, the shape looks like this (12768, 240, 240, 3).

The proposed model is a pre-trained model, originally trained on images with a size of 224 x 224 pixels. Hence, we have resized our images to best fit the model. We performed permutation operation on the dimensions of the dataset as PyTorch prefers the "channel-first" ordering. The training data now has a shape (12768, 3, 224, 224), where image dimensions are 224 by 224. Similarly, we have 57 testing patients, and the testing dataset shape is (3192, 3, 224, 224). We are interested in knowing the grade of each patient, from the 56 slices of a patient, if the majority belongs to HGG then its HGG else LGG.

| Class Samples | Training (80%) | Testing (20%) | Total samples |
|---------------|----------------|---------------|---------------|
| HGG | 9,408 | 2,352 | 11,760 |
| LGG | 3,360 | 840 | 4,200 |
| Total | 12,768 | 3,192 | 15,960 |

Table 2: Summary of Deep Learning Data. There are a total of 11,760 slices which belong to HGG. And a total of 4,200 slices for LGG.

As part of the data preprocessing, we calculated the mean and the standard deviation across all the images and pixels in both training and testing dataset. This is followed by normalizing both the training and testing datasets separately using their respective computed values. This normalizing step is crucial because it establishes a consistent scale for the input features, assisting the training process and improving model convergence.

In the data Augmentation techniques, we have introduced random rotation and horizontal flipping to increase the diversity in the training dataset. While horizontal flipping creates mirror images, random rotation is responsible for introducing variability in the alignment of the images. These are applied only to 50% of the images randomly.

Deep Learning models

4-2-2: ResNet

ResNet stands for Residual Network, a Convolutional Neural Network (CNN) architecture which can support up to hundreds or thousands of convolutional layers. The drawback with the earlier CNN architectures was that it could not be scaled to a large number of layers. Due to this weakness, they were underperforming. ResNet was able to solve the problem of vanishing gradient, also called "skip connections". ResNet piles up the identity mapping, then skips those layers, and the activations of the earlier layers are reused again. By skipping these layers, the initial training is boosted, because the network is not compressed. [62]

When the network is passed for retraining, the skipped layers, known as the residual parts, are now considered to study the input image in a much better way. The different versions of ResNet available are ResNet-50, ResNet-101, and ResNet-152. Each of them varies in complexity and number of layers [62]. In this research we have used ResNet-50 which has 50 layers including conventional layers, fully connected layers, and activation functions.

4-2-3: MobileNet

MobileNet is simple yet efficient CNN which is designed for mobile vision applications. Despite it being not very computationally intensive it has high efficiency. The key features of MobileNet include Depthwise Separable Convolutions, and Width Multiplier and Resolution Multiplier. The Depthwise Separable Convolutions technique is utilized to break the usual convolution operation into a pointwise convolution and depthwise convolution. This is helpful in reducing the number of computations and parameters making it efficient. Whereas the Resolution Multiplier and Width Multiplier are the hyperparameters, introduced by the MobileNet, which allows the users to limit the computational cost and the size of the model. When there are certain constraints with the target device, these parameters help in customization [63]. For our thesis, we have decided to use MobilenetV2 [64]. This is an updated version of the original MobileNet design. This new design's key feature is Inverted Residuals with Linear Bottlenecks. This helped in maintaining capturing and maintaining more records within the network [64].

4-2-4: DenseNet

A typical ConvNet, passes the input image through many convolutions and gathers high level features. While in DenseNet, concatenation is used, and each layer receives additional inputs from all the previous layers. And then it forwards its own feature maps to all the succeeding layers. This way every layer will be obtaining a “collective knowledge” from all the previous layers. This results in a lower number of channels with a thinner and more compact network. Furthermore, it will tend to have richer patterns with additional diverse features. A very notable advantage of DenseNet is that it has a strong gradient flow [65]. For this research, we have decided to go with DenseNet-201 [66]. This model has 201 layers and can capture more complex hierarchical features.

4-2-5: EfficientNet

EfficientNet is based on a concept named “compound scaling”. This concept helped overcome common issues like accuracy, model size, and computational efficiency. The basic idea behind this is to scale the three dimensions of a neural network, such as depth, width, and resolution. Here the depth scaling refers to the total number of layers of a network. The deeper the model, the better representation of data, but this comes at the expense of computational resources. However, if the models are designed in a shallow manner, then it costs some accuracy. With respect to width scaling, it is pertained to the number of channels in every layer of the network. Just like depth scaling, the increase in width captures complex features and patterns, resulting in better accuracy. On the contrary, reducing the width leads to a lightweight model. Lastly, the resolution scaling involves altering the image sizes. An input image with better resolution yields better performance. Nonetheless, fine-grained details are missed in lower-resolution input images [67].

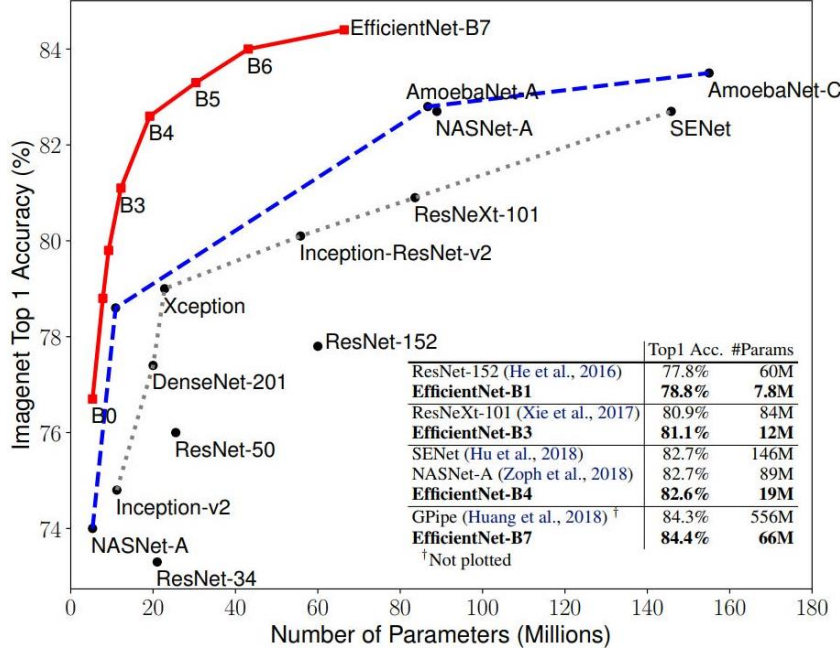


Figure 5: Effectiveness of the DL models. EfficientNet curve is highlighted using a red line [67].

The above figure shows the effectiveness of the model in comparison to other models like ResNet, EfficientNet, etc. A significant difference can be seen in the parameters of GPipe [68], a giant neural network using pipeline parallelism, with over 556 million and EfficientNet with just 66 million. EfficientNet model achieved slightly better accuracy with over 8 times less parameters than GPipe. EfficientNet is able to achieve this by intelligently adjusting the depth, width, and resolution. This also makes this model more adaptable to several hardware constraints [67]. EfficientNet architectures are ordered based on their compound scaling coefficient, like EfficientNet_B0, EfficientNet_B1, EfficientNet_B2, and so on. In our experiments we have utilized EfficientNet_B2 architecture, which is a variant of EfficientNet. This model is larger and more powerful than EfficientNet_B1 architecture [67].

4-2-6: Proposed Model - VGG16 with Attention

For the proposed model, we have adopted VGG-16 model [31], which has 16 layers in total. The convolution layers have a filter size 3x3 and stride = 1. The activation function is ReLU [31]. The maxpooling layer is of 2x2 filters and stride = 2 [31]. It also has 2 dense layers [31]. And lastly a softmax layer [31]. Right now, we have VGG16 as the backbone without any dense layers [31].

While analyzing an image or some MRI scan, we humans tend to focus only on the objects that are related to the task at hand [31]. For instance, dermatologists, who treat skin cancer, focus only on the lesion and not on irrelevant parts of the image like hair or background [31]. To copy this image exploration pattern, we are going to implement an attention module, which will calculate a spatial (pixel-wise) attention map [69]. Detailed illustration of the

proposed network architecture is in figure 4.1 [69]. The gray blocks are the two attention modules (Refer to Fig.4.2 for details) [69]. One of the attention modules is pool-3 and the other is pool-4 [69]. The output which is the final feature vector is derived from concatenating the three feature vectors. These feature vectors were derived via global research pooling [69]. The final feature vector is passed to the classification layer, which is not in this figure [69].

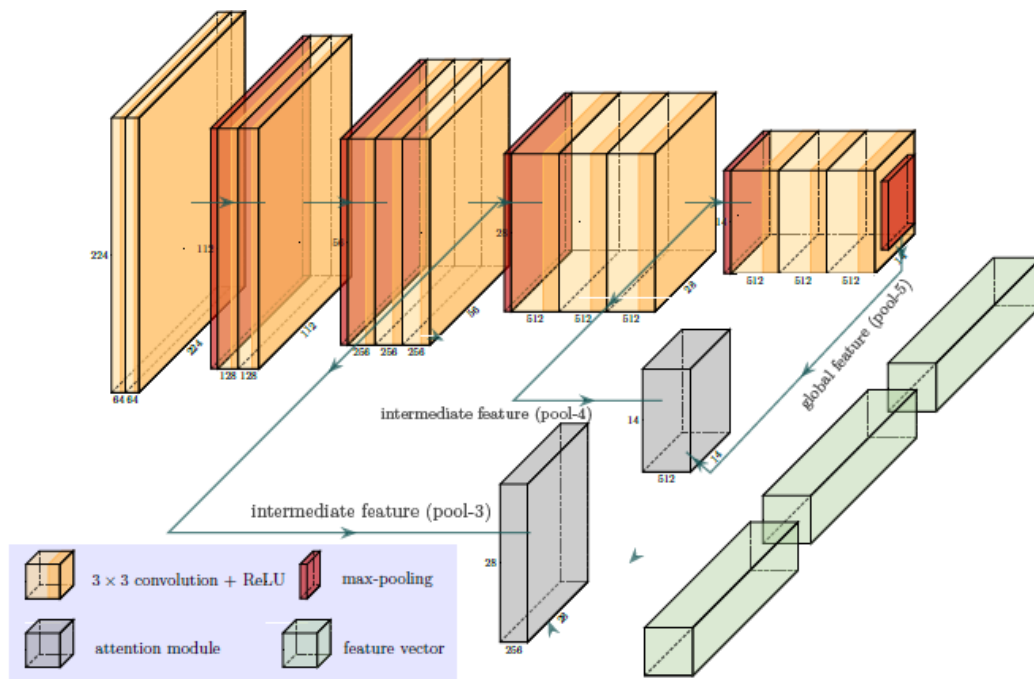


Figure 6.1: Detailed illustration of the proposed network architecture. The gray blocks are the two attention modules (Refer to Fig 4.2 for details) [69]. One of the attention modules is pool-3 and the other is pool-4 [69]. The output which is the final feature vector is derived from concatenating the three feature vectors. These feature vectors were derived via global research pooling [69]. The final feature vector is passed to the classification layer, which is not in this figure [69].

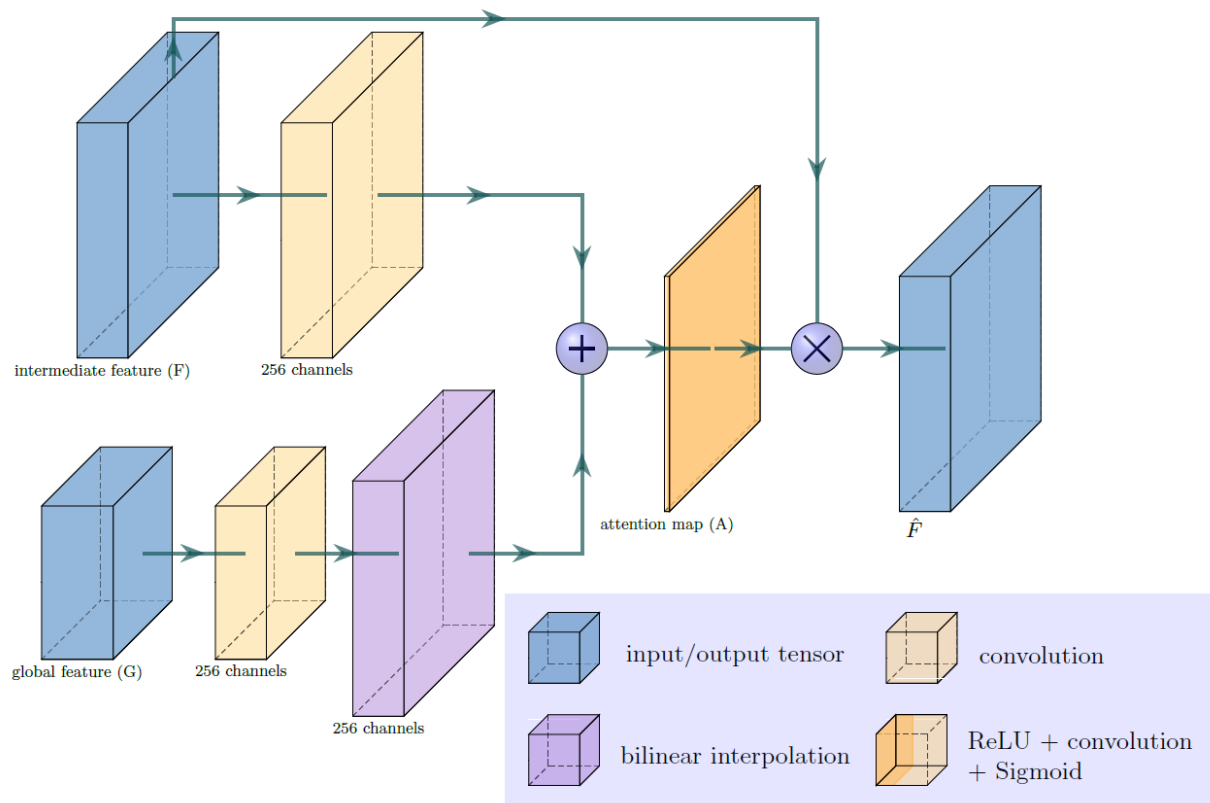


Figure 6.2: Illustration of the gray blocks from the two attention modules. Feature unsampling is performed with the help of bilinear interpolation, when the spatial size of both intermediate and global features is unique [69]. The multiplication operation is “pixel-wise”, whereas the sum operation is element wise [69].

4-2-7: Fine Tuning for Deep Learning

In our final stage of this process, we focused on fine-tuning hyperparameters like dropout rate, learning rate, and weight decay. The learning rate, which is a key parameter influencing the convergence of our model, was altered across four values: 0.01, 0.001, 0.0005, and 0.0001. As for the dropout rate, we explored a range of values from 0.2 to 0.5. Lastly, we adjusted the weight decay across the following range: 0.01, 0.001, and 1e-05.

Section 5: RESULTS

5-1: Performance Measures

The five metrics we have used for measuring the models are accuracy, precision, recall, f1 score, and confusion matrix. Accuracy (Table 5.1) is the ratio of model's number of correct predictions to the sum of total predictions [70]. Precision (Table 5.2) is the ratio of true positive to all the positive predicted cases [70]. Recall (Table 5.3) is the ratio of true positive to all the cases that shall be predicted as positive [70]. F1 score (Table 5.4) is the harmonic mean of precision and recall [70]. Although when we look at metrics like accuracy, precision, recall, and f-1 score it looks like the model is performing well. But the confusion matrix has shown that the models are not able to learn LGG features and are predicting them incorrectly as HGG.

A confusion matrix helps in visualizing the outcome in a tabular format, where the outputs i.e., model's prediction and actual values are compared together [71]. In short, it plots a table with all the actual and predicted values by a model [71]. Refer figure 5 below for a basic structure of a confusion matrix.

The confusion matrix produces four combinations from the actual and predicted values of a classifier model [71]. In the below figure we can spot those four combinations.

- True Positives: The number of times the model's predicted positives are equal to the actual positive values, then they are said to be true positive [71].
- False Positive: The number of times the model's predicted positive value is incorrect. Then it is said to be false positive [71].
- True Negative: The number of times the model's predicted negative values are indeed negative values. Then it is said to be true negative [71].
- False Negative: The number of times the model's predicted negative values are incorrect and misclassified as negative. Then is said to be false negative [71].

| | | Predicted | |
|--------|---------------|--------------------------------------|-------------------------------------|
| | | Negative (N) - | Positive (P) + |
| Actual | Negative - | True Negative (TN) | False Positive (FP) Type I Error |
| | Positive + | False Negative (FN) Type II Error | True Positive (TP) |

Figure 7: Confusion Matrix [72]

5-2: Machine Learning

| | Accuracy | | Precision | | Recall | | F-1 Score | |
|------------------------|--------------|-------------|--------------|-------------|--------------|-------------|--------------|-------------|
| | Training (%) | Testing (%) | Training (%) | Testing (%) | Training (%) | Testing (%) | Training (%) | Testing (%) |
| Logistic Regression | 94.64 | 73.68 | 98.70 | 73.68 | 90.47 | 100 | 94.41 | 84.85 |
| Decision Tree | 93.15 | 68.42 | 100 | 73.08 | 100 | 90.48 | 100 | 80.85 |
| Support Vector Machine | 100 | 73.68 | 100 | 73.68 | 86.31 | 100 | 92.65 | 84.85 |
| Random Forest | 100 | 73.68 | 100 | 73.68 | 100 | 100 | 100 | 84.85 |
| Gradient Boosting | 100 | 73.68 | 100 | 73.68 | 100 | 100 | 100 | 84.85 |
| LightGBM | 100 | 73.68 | 100 | 73.68 | 100 | 100 | 100 | 84.85 |
| XGBoost | 100 | 73.68 | 100 | 73.68 | 100 | 100 | 100 | 84.85 |
| Stacking | 93.45 | 73.68 | 100 | 73.68 | 86.90 | 100 | 92.99 | 84.85 |

Table 3: ML results of all the model

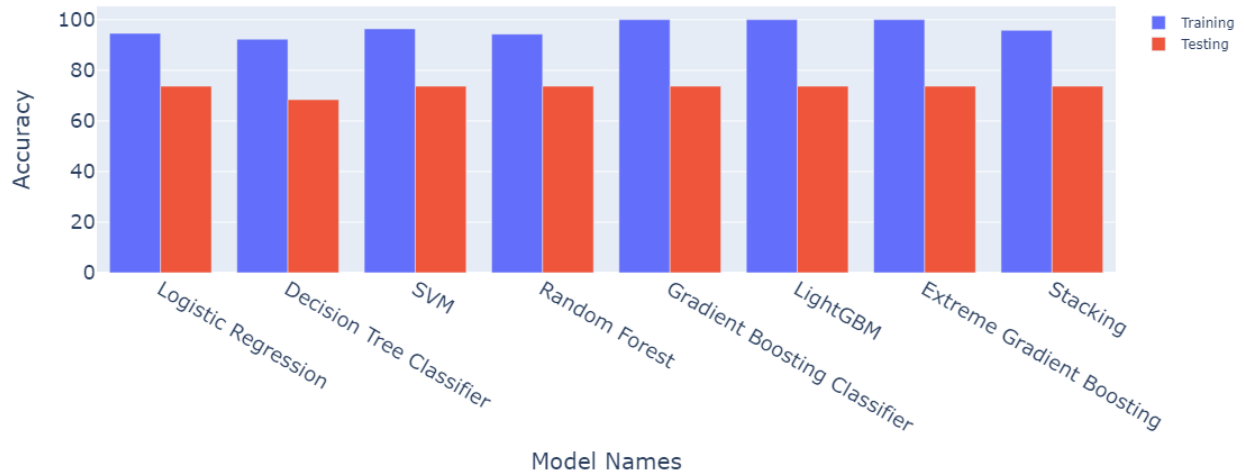


Figure 8.1: Bar Plot for Accuracy

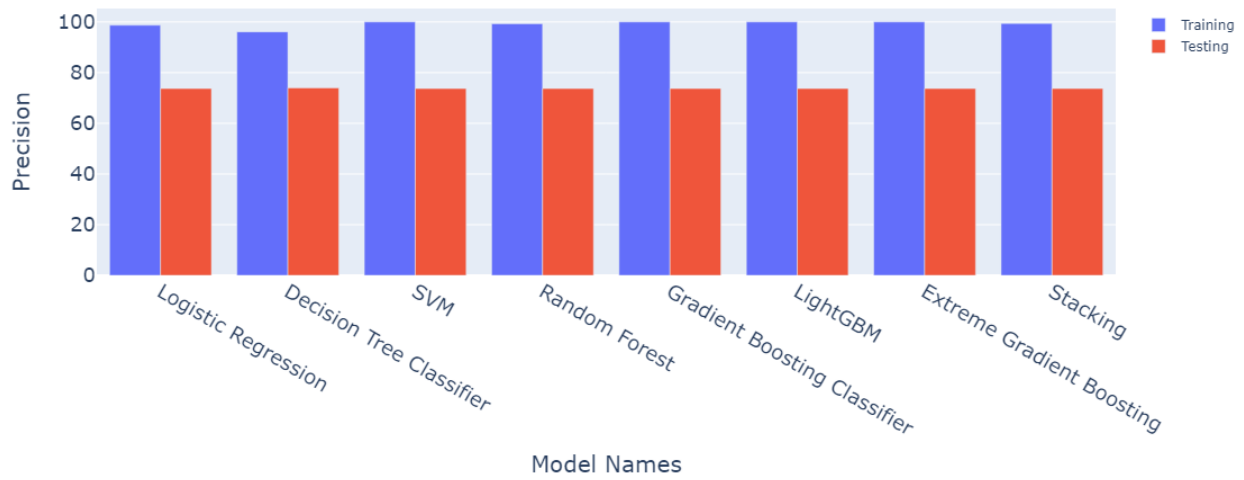


Figure 8.2: Bar Plot for Precision

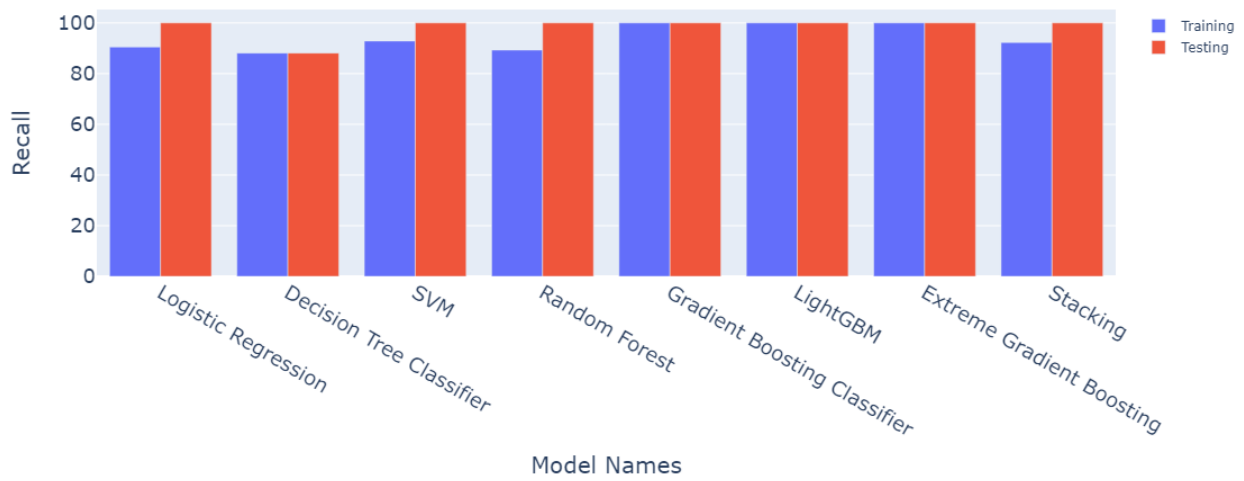


Figure 8.3: Bar Plot for Recall

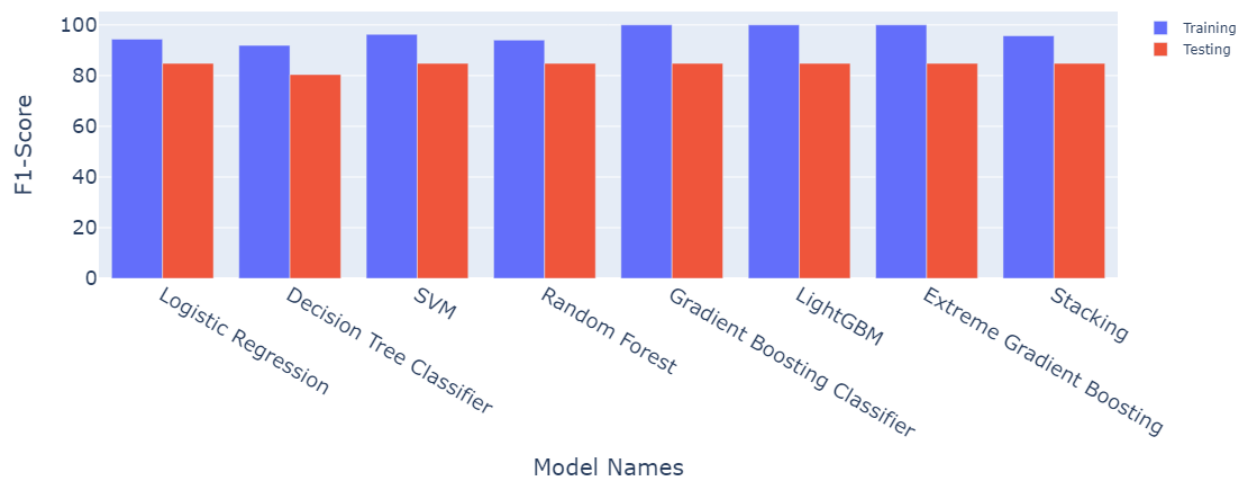


Figure 8.4: Bar Plot for F1-score

Confusion Matrix of Training

| | | | |
|---------------|-----|---|-----|
| | | Logistic Regression Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 166 | 2 |
| | HGG | 16 | 152 |

| | | | |
|---------------|-----|-----------------------------------|-----|
| | | Decision Tree Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 162 | 6 |
| | HGG | 20 | 148 |

| | | | |
|---------------|-----|---------------------------------------|-----|
| | | Gradient Boosting Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 168 | 0 |
| | HGG | 0 | 168 |

| | | | |
|---------------|-----|-------------------------------|-----|
| | | Light GBM Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 168 | 0 |
| | HGG | 0 | 168 |

| | | | |
|---------------|-----|--|-----|
| | | Support Vector Machine Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 168 | 0 |
| | HGG | 12 | 156 |

| | | | |
|---------------|-----|-----------------------------------|-----|
| | | Random Forest Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 167 | 1 |
| | HGG | 18 | 150 |

| | | | |
|---------------|-----|-----------------------------|-----|
| | | XGBoost Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 168 | 0 |
| | HGG | 0 | 168 |

| | | | |
|---------------|-----|---------------------------|-----|
| | | Stack Predicted Values | |
| | | LGG | HGG |
| Actual Values | LGG | 167 | 1 |
| | HGG | 13 | 155 |

Figure 9: Confusion Matrix of Training.

Confusion Matrix of Testing

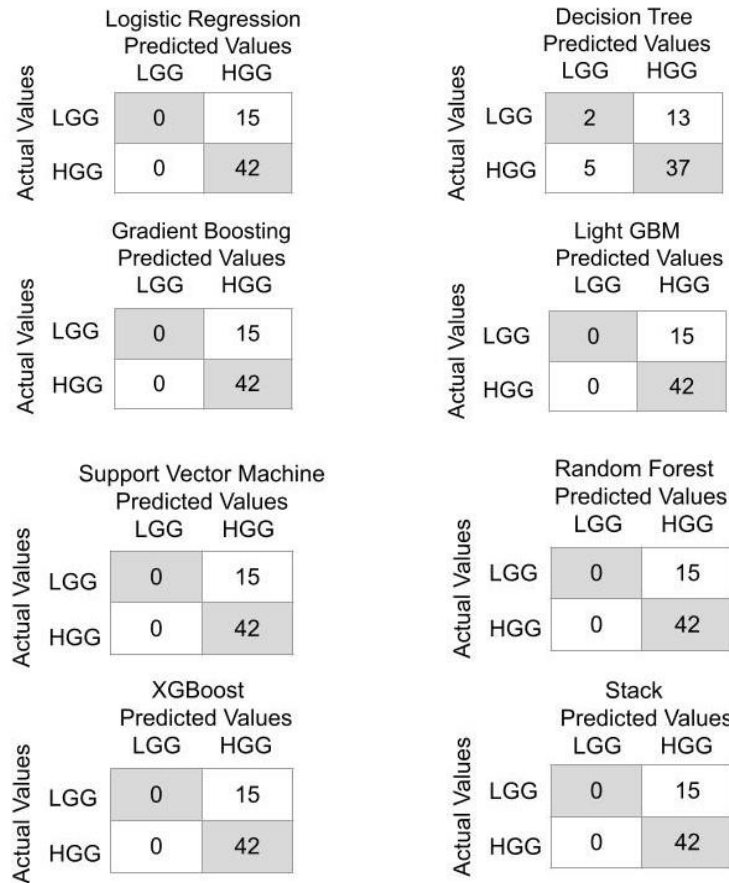


Figure 10: Confusion Matrix of Testing.

Hyperparameters from the Random Forest model

- Number of Estimators (n_estimators): 100
- Minimum Samples Split (min_samples_split): 2
- Minimum Samples Leaf (min_samples_leaf): 3
- Maximum Features (max_features): 'sqrt'
- Maximum Depth (max_depth): 10
- Criterion (criterion): 'gini'
- Bootstrap: False

Feature Importance

The below figure 8 illustrates a graphical representation of features' importance when they are arranged based on their importance in a decreasing order. These features are extracted with the help of Random Forest model, since it has a built-in function for calculating feature importance in scikit-learn [56]. We can notice that there is a significant drop of importance in the initial 30 features. Later the feature importance started decreasing gradually and around 200 features the slope became a plateau.

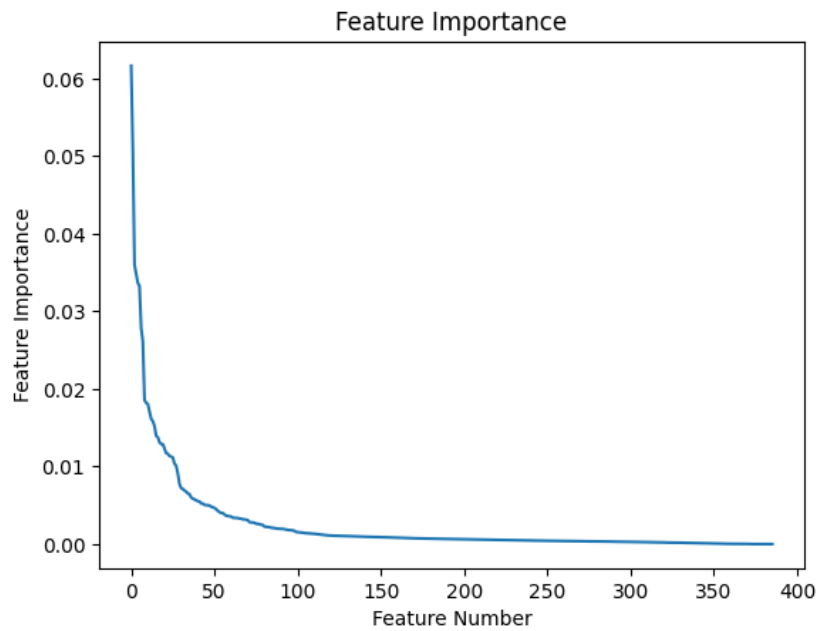


Figure 11: Feature Importance of all the features arranged based on importance.

Among the total of top 135 features, most of the features belong to GLCM category. GLDM, first order, GLRLM and GLSZM have similar number of features. The second last category is NGTDM, and the least number of features belongs to Shape.

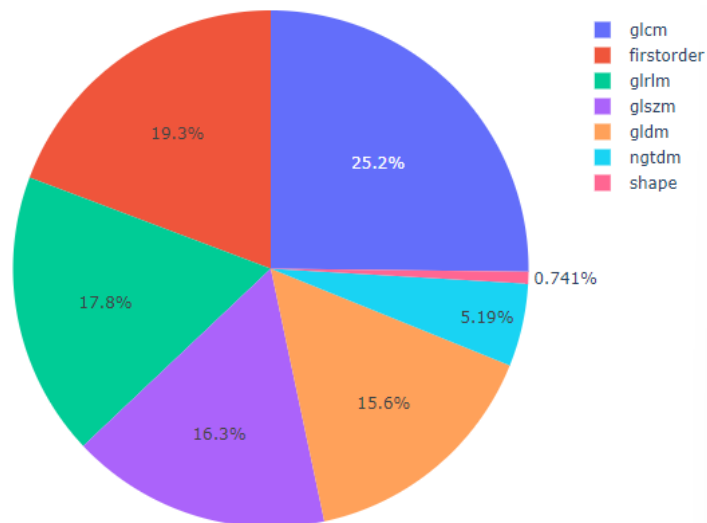


Figure 12: Pie chart of the Feature Importance

Refer table 7 below for the top 20 important features according to the Gradient boosting algorithm, which is one of the best models as per the test results. With 1 being the most important feature and 27 being the 27th most important feature.

| No. | Features | Importance |
|-----|--|------------|
| 1 | T1 Original GLCM Inverse Variance | 0.061645 |
| 2 | T1 Original GLRLM Run Entropy | 0.050814 |
| 3 | T1 Original GLDM Dependence Entropy | 0.035988 |
| 4 | T1 Original GLSZM Size Zone Non-Uniformity Normalized | 0.034862 |
| 5 | T1 Original GLSZM Small Area Emphasis | 0.033656 |
| 6 | T1 Original GLRLM Long Run High Gray Level Emphasis | 0.033251 |
| 7 | T1 Original GLRLM Long Run Emphasis | 0.027826 |
| 8 | T1 Original GLCM ID | 0.026129 |
| 9 | T1 Original GLCM IDM | 0.018572 |
| 10 | T1 Original GLRLM Run Percentage | 0.018205 |
| 11 | T1 Original GLCM Correlation | 0.018050 |
| 12 | T1 Original GLCM Cluster Tendency | 0.017074 |
| 13 | T1 Original GLSZM Gray Level Non-Uniformity Normalized | 0.016142 |
| 14 | T1 Original GLDM Small Dependence Emphasis | 0.015835 |
| 15 | T1 Original Firstorder Mean Absolute Deviation | 0.015224 |
| 16 | T1 Original GLCM MCC | 0.013951 |
| 17 | T1 Original GLCM IDN | 0.013778 |
| 18 | T1 Original GLSZM Zone Percentage | 0.013129 |
| 19 | T1 Original GLCM Sum Entropy | 0.012893 |
| 20 | T1 Original GLRLM Run Variance | 0.012878 |
| 21 | T1 Original Firstorder Entropy | 0.012326 |
| 22 | T1 Original GLRLM Gray Level Non-Uniformity Normalized | 0.011710 |
| 23 | T1 Original GLDM Large Dependence Emphasis | 0.011656 |
| 24 | T1 Original Firstorder Uniformity | 0.011354 |
| 25 | T1 Original GLCM IMC2 | 0.011280 |
| 26 | T1 Original Firstorder Variance | 0.011159 |
| 27 | T1 Original GLDM Gray Level Variance | 0.010377 |
| 28 | T1 Original GLCM IMC1 | 0.010110 |

Table 4: Top 28 important features from the Radiomic features

5-3: Deep Learning model

The below table contains all the results from the models we have explored. Starting with ResNet 5 experiments were conducted on the complete dataset. We have used five metrics

Confusion Matrix [71], Accuracy [73], Precision [74], Recall [75], and F1-score [76]. For our research we are more focused on the f1-score because it provides a balanced measure of precision and recall. F1-score studies both the ability to predict all positive cases, i.e. recall, and the ability to accurately detect positive cases, i.e. precision. The proposed model was able to record an f1-score of 91.18% with only 8 misclassifications from the total of 57 test patients. The hyperparameters which produced these results are as follows:

- Learning_Rate = 0.01
- Dropout_Rate = 0.4
- Weight_Decay = 1e-05

| No. | Model Names | Confusion Matrix | Accuracy (%) | Precision (%) | Recall (%) (Sensitivity) | F1-score (%) | | | | |
|-----|-------------------------------------|---|--------------|---------------|--------------------------|--------------|-------|-------|-------|-------|
| 1 | ResNet50 | <table><tr><td>8</td><td>7</td></tr><tr><td>1</td><td>41</td></tr></table> | 8 | 7 | 1 | 41 | 85.96 | 85.42 | 97.62 | 91.18 |
| 8 | 7 | | | | | | | | | |
| 1 | 41 | | | | | | | | | |
| 2 | MobileNetV2 | <table><tr><td>4</td><td>11</td></tr><tr><td>6</td><td>36</td></tr></table> | 4 | 11 | 6 | 36 | 70.18 | 76.59 | 85.71 | 80.85 |
| 4 | 11 | | | | | | | | | |
| 6 | 36 | | | | | | | | | |
| 3 | DenseNet201 | <table><tr><td>8</td><td>7</td></tr><tr><td>6</td><td>36</td></tr></table> | 8 | 7 | 6 | 36 | 77.19 | 83.72 | 85.71 | 84.70 |
| 8 | 7 | | | | | | | | | |
| 6 | 36 | | | | | | | | | |
| 4 | EfficientNet_B2 | <table><tr><td>6</td><td>9</td></tr><tr><td>7</td><td>35</td></tr></table> | 6 | 9 | 7 | 35 | 71.93 | 79.55 | 83.33 | 81.48 |
| 6 | 9 | | | | | | | | | |
| 7 | 35 | | | | | | | | | |
| 5 | Proposed Model VGG16 with Attention | <table><tr><td>8</td><td>7</td></tr><tr><td>1</td><td>41</td></tr></table> | 8 | 7 | 1 | 41 | 85.96 | 85.42 | 97.62 | 91.18 |
| 8 | 7 | | | | | | | | | |
| 1 | 41 | | | | | | | | | |

Table 5: Results from Deep Learning Model

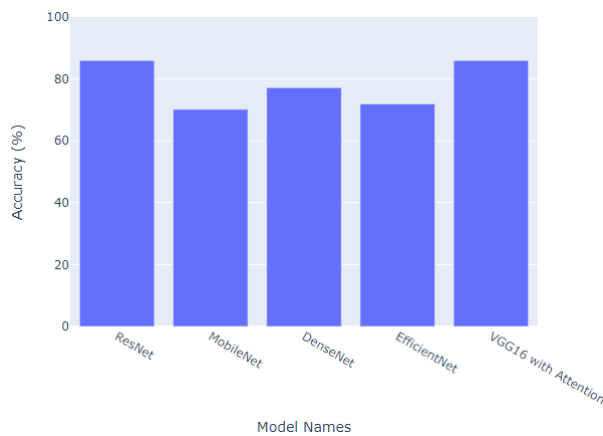


Figure 13: Bar Plot for Accuracy of the Models

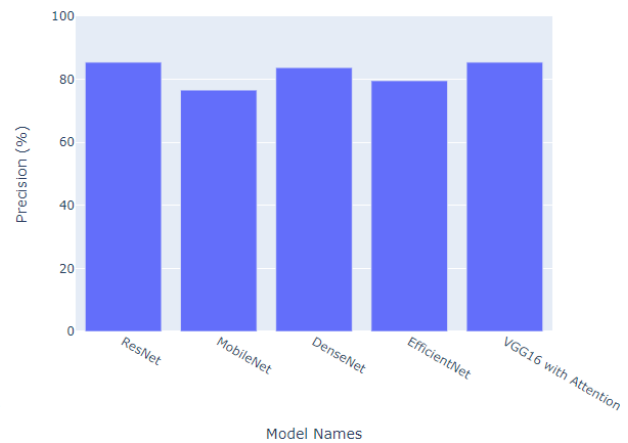


Figure 14: Bar Plot for Precision of the Models

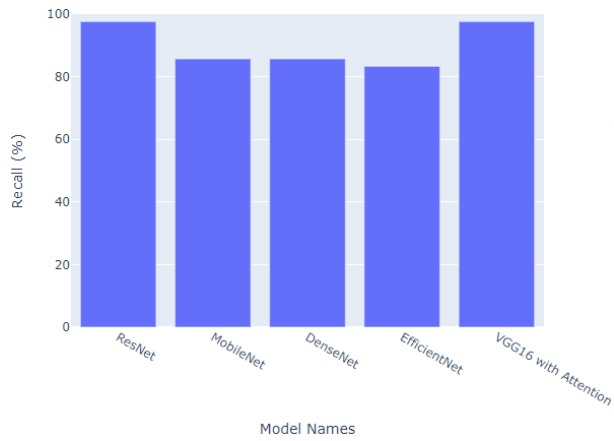


Figure 15: Bar Plot for Recall of the Models

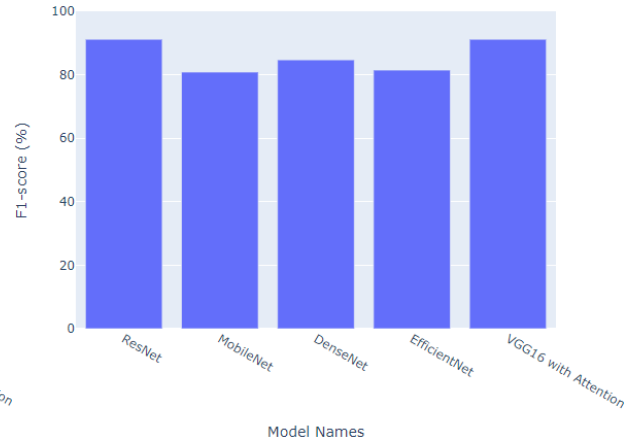
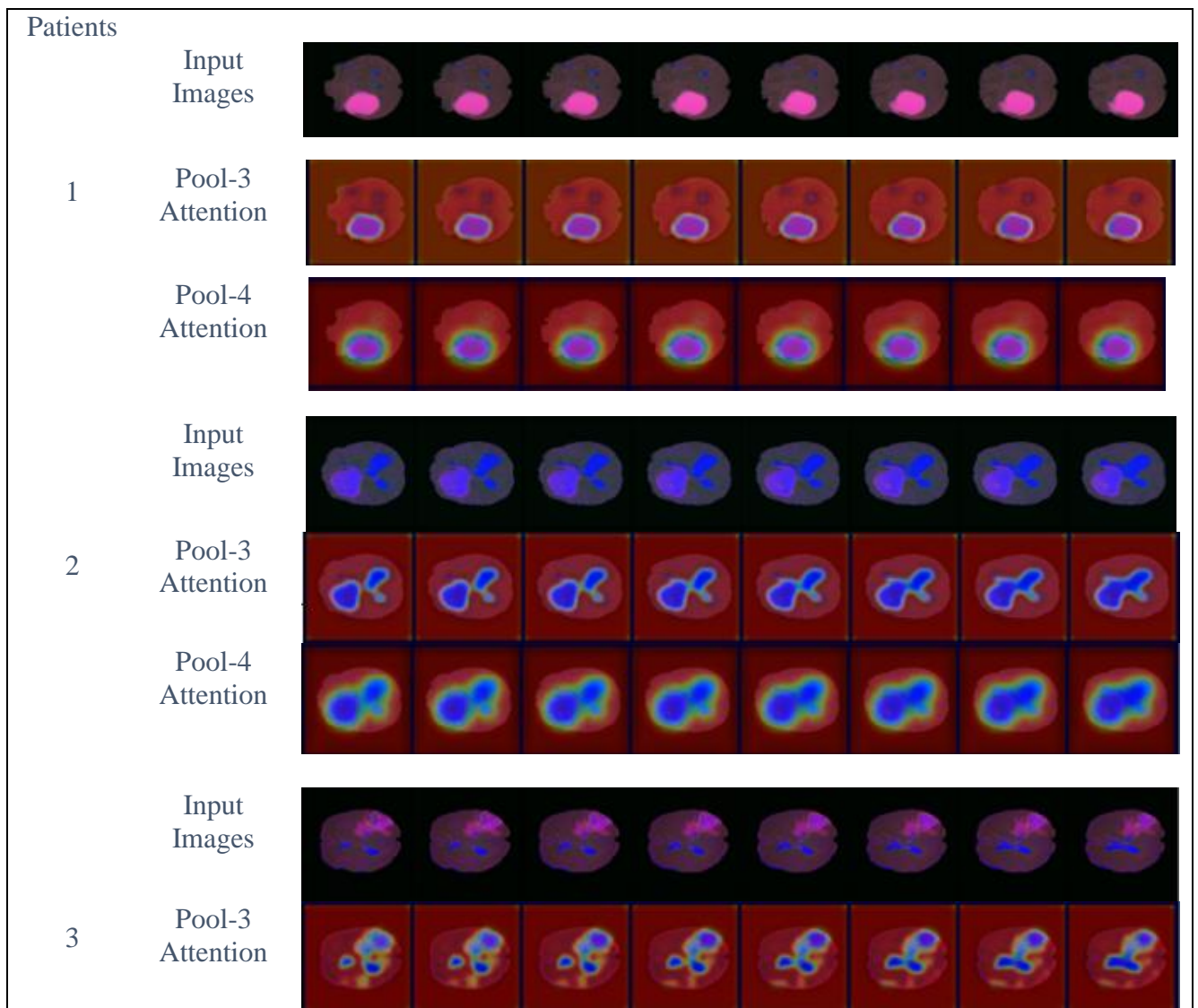


Figure 16: Bar Plot for F1-score of the Models.

Visualizations



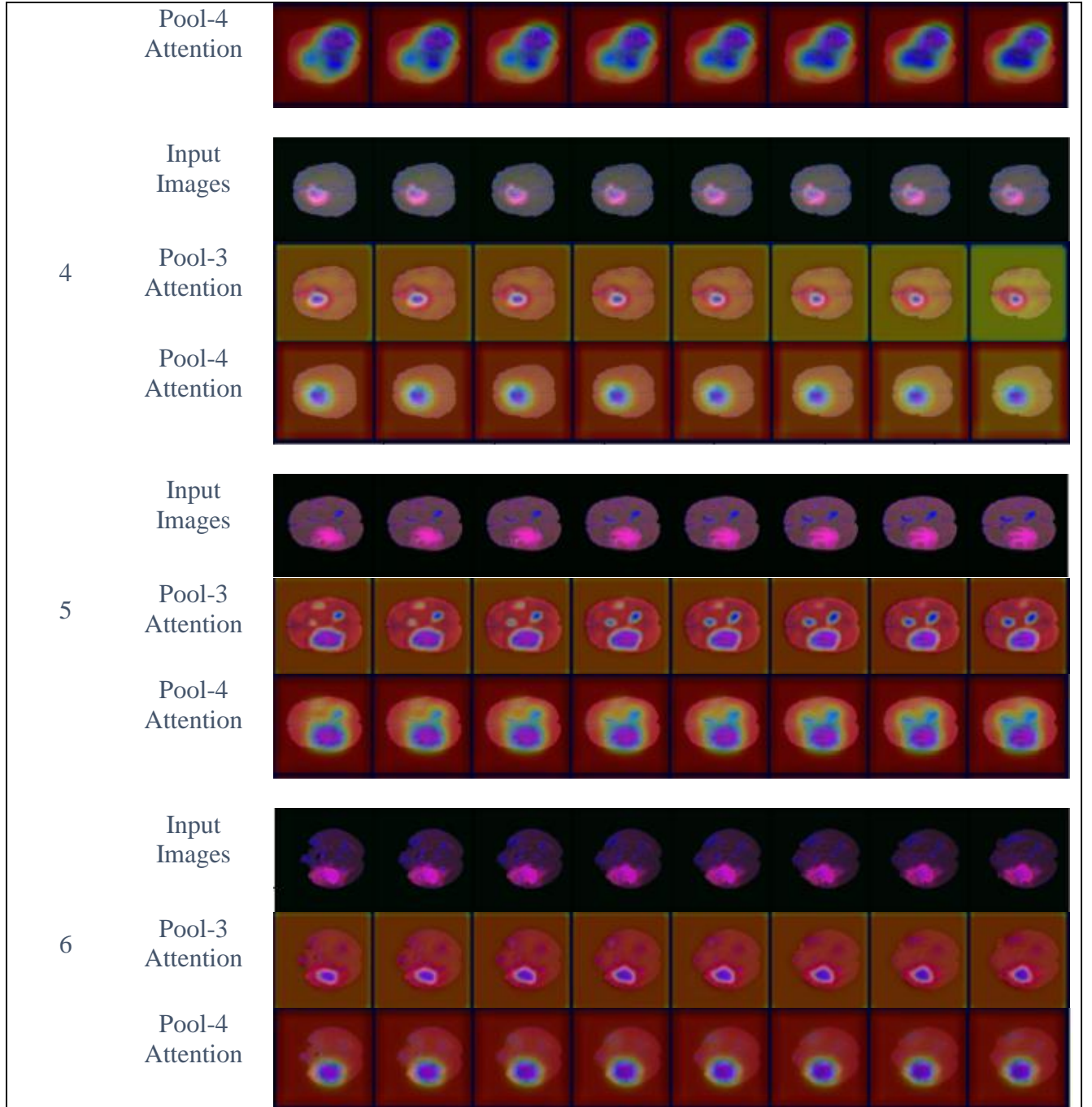


Figure 17: Visualizations from 6 different test patients correctly classified and focus on the tumor.

As we examine the above visualizations of 6 different patients, we can notice that the attention layers were able to detect the tumor from the input images. The model is coloring the tumor blue. The rest of the image, that is irrelevant, is either colored red or any another color. The gray blocks in figure 4.1 are the pool-3 and pool-4 attentions which produce the above visualization outputs.

We can notice that the confusion matrix of ResNet50 and the proposed model, from the results table above, had most of the LGG patients wrongly classified as HGG. This is because the tumor in some of the LGG patients seemed to be bigger. Below is one such example.

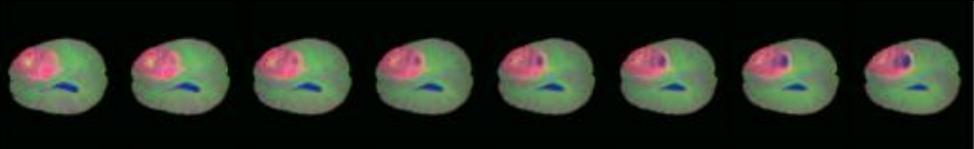
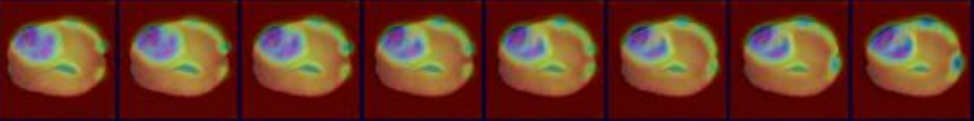
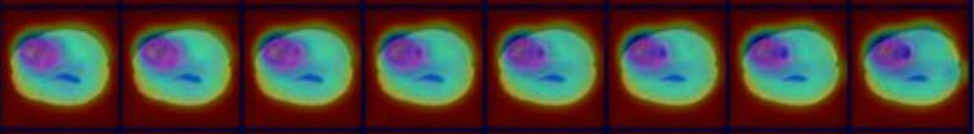
| | |
|------------------|--|
| Input Images |  |
| Pool-3 Attention |  |
| Pool-4 Attention |  |

Figure 18: Visualization of LGG patient misclassified as HGG.

Section 6: DISCUSSION

In a nutshell, we have developed a CAD tool which classifies the tumor grade of a patient, whether it is HGG or LGG. The dataset provided 210 patient scans with HGG and 75 patient scans with LGG. For the machine learning algorithm, we have generated a csv file by extracting all the radiomic features from the MRI sequences. The algorithms we have considered are logistic regression, decision tree, support vector machine, random forest, gradient boosting, extreme gradient boosting, light GBM, and lastly stacking. The metrics considered to measure these algorithms are confusion matrix, accuracy, precision, recall, and f1 score. We used the Random Forest model to derive all the important feature categories. As for the deep learning algorithm, we have used 5 neural network models, Resnet-50, MobileNetV2, DenseNet201, EfficientNet_B2, and lastly the proposed model - VGG16 with attention to evaluate all the MRI sequences. The metrics used for evaluating are confusion matrix, accuracy, precision, recall, and f1-score. In the proposed model, the attention layer produces visualization which locates the tumor.

In the ML results, almost all models gave similar results. Despite trying out a variety of models, they were not able to learn a lot from the input data. This indicates that they require feature engineering.

With regards to the Deep Learning model, the proposed model gave an f1-score of 91% on the test set. Traditionally, models were missing the attention feature, where the model provides visuals of it focusing on the tumor and classifying them accurately. In our proposed model we have added this feature with the help of attention maps. Here 2 intermediate features are extracted and passed on to the final classification layer after concatenating them. In visualizations table, we can observe that the model is detecting the tumor. While performing testing we noticed that a few patients were repeatedly getting misclassified, we have planned to examine them further in the future. Another notable aspect of the proposed model is that it is very sensitive to change in hyperparameters. Changing the weight decay from $1e-5$ to $1e-4$ has F1-score dropped from 91.18% to 79.64%.

Section 7: CONCLUSION

To conclude, glioma grading is required for treatment planning. The treatment process, biopsy, is invasive and expensive. So, there is a need for better ways to determine the grade of the tumor. Our goal is to determine the grade through non-invasive approach. ML and DL algorithms can be trained based on the MRI scans or features extracted from it. 107 radiomic features are extracted from the MRI scans to serve as an input for the ML models. The ML models we have seen in this study are Logistic Regression, Decision Tree, Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GB), XGBoost, Light Gradient-Boosting Machine (LGBM) and Stacking. And for the Deep Learning, we have trained 5 different models, Resnet-50, MobileNetV2, DenseNet201, EfficientNet_B2, and lastly the proposed model - VGG16 Convolutional Neural Network (CNN) model with attention. Proposed DL model does not require explicit feature extraction and by using attention removes the need of tumor segmentation. The algorithms are evaluated using accuracy, f1-score, precision, recall, and confusion matrix. An f1-score of 91.18% was achieved by the proposed model. The model also displayed where the tumor was located.

All in all, in the ML algorithms, we have noticed that radiomic features have not produced satisfactory results and indeed require further feature engineering. And for deep learning, the proposed model produced an f1-score of 91%. And with the help of attention maps in the model, we can visually see how the tumors have been detected properly.

Section 8: FUTURE WORK

Nevertheless, the proposed model – VGG16 with Attention for predicting glioma grades has demonstrated promising results with a notable F1 score of 91.18%, following are the possible areas which will be helpful in future research and improvements. The first one would be to incorporate multi-modal data. In the data preprocessing steps of deep learning, we have dropped the t1 sequence and only considered t1ce, t2, and flair sequences. For future experiments, we can keep the t1 sequence instead of dropping it. This can incorporate diverse information and can help the model in learning other subtle features and improve overall diagnostic accuracy.

Another aspect to consider is investigating other architectures. Powerful architectures like ResNet, Inception, etc. can be integrated with the attention mechanism we have used in the proposed model. The process of integration involves adapting the attention block's input dimensions and matching it with the target architecture's structure. Along with this experiment, we can also investigate state-of-the-art models, for instance ensemble models or transformer-based models.

Last but not least, handling imbalanced datasets. A part of the reason for misclassification of few LGG patients to be classified as HGG could be lack of MRI scans of LGG patients. As viewed earlier, there are only 75 LGG patients which is almost three times less than HGG patients. This could impact on the decision-making process of the model. Data augmentation techniques like over-sampling, cluster-based over-sampling, stratified sampling, etc. can help mitigate this issue.

Section 9: REFERENCES

- [1] Tandel G.S., Biswas M., Kakde O.G., Tiwari A., Suri H.S., Turk M., Laird J.R., Asare C.K., Ankrah A.A., Khanna N.N., et al. A Review on a Deep Learning Perspective in Brain Cancer Classification. *Cancers*. 2019; 11:111. <https://doi.org/10.3390/cancers11010111>
- [2] Cancer Statistics. 2022. [(accessed on 17 November 2022)]. Available online: <https://www.cancer.net/cancer-types/brain-tumor/statistics>
- [3] Gutta, S., Acharya, J., Shiroishi, M. S., Hwang, D., & Nayak, K. S. (2021). Improved glioma grading using deep convolutional neural networks. *American Journal of Neuroradiology*, 42(2), 233-239. doi: 10.3174/ajnr.A6882
- [4] Pereira S., Pinto A., Alves V., Silva C.A. Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images. *IEEE Trans. Med. Imaging*. 2016; 35:1240–1251. doi: 10.1109/TMI.2016.2538465.
- [5] Xie Y., Zaccagna F., Rundo L., Testa C., Agati R., Lodi R., Manners D.N., Tonon C. Convolutional Neural Network Techniques for Brain Tumor Classification (from 2015 to 2022): Review, Challenges, and Future Perspectives. *Diagnostics*. 2022; 12:1850. doi: 10.3390/diagnostics12081850
- [6] American Society of Clinical Oncology Brain Tumor Diagnosis. 2021. [(accessed on 5 November 2021)]. Available online: <https://www.cancer.net/cancer-types/brain-tumor/diagnosis>
- [7] Role of Ensemble Deep Learning for Brain Tumor Classification in Multiple Magnetic Resonance Imaging Sequence Data. doi: 10.3390/diagnostics13030481
- [8] Bauer S., Wiest R., Nolte L.-P., Reyes M. A survey of MRI-based medical image analysis for brain tumor studies. *Phys. Med. Biol.* 2013;58:R97. doi: 10.1088/0031-9155/58/13/R97
- [9] Işın A., Direkoğlu C., Şah M. Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods. *Procedia Comput. Sci.* 2016; 102:317–324. doi: 10.1016/j.procs.2016.09.407
- [10] Balafar M.A., Ramli A.R., Saripan M.I., Mashohor S. Review of brain MRI image segmentation methods. *Artif. Intell. Rev.* 2010; 33:261–274. doi: 10.1007/s10462-010-9155-0
- [11] Leung D., Han X., Mikkelsen T., Nabors L.B. Role of MRI in Primary Brain Tumor Evaluation. *J. Natl. Compr. Cancer Netw.* 2014; 12:1561–1568. doi: 10.6004/jnccn.2014.0156
- [12] Kumar R., Srivastava R., Srivastava S.K. Detection and Classification of Cancer from Microscopic Biopsy Images Using Clinically Significant and Biologically Interpretable Features. *J. Med. Eng.* 2015; 2015:457906. doi: 10.1155/2015/457906
- [13] Wang M., He X., Chang Y., Sun G., Thabane L. A sensitivity and specificity comparison of fine needle aspiration cytology and core needle biopsy in evaluation of suspicious breast lesions: A systematic review and meta-analysis. *Breast*. 2017; 31:157–166. doi: 10.1016/j.breast.2016.11.009
- [14] Moiin A., Neill B.C. A novel punch biopsy technique without scissors or forceps. *J. Am. Acad. Dermatol.* 2021;85: e71–e72. doi: 10.1016/j.jaad.2018.05.1253

- [15] El-Baz A., Gimel'farb G., Suri J.S. Stochastic Modeling for Medical Image Analysis. CRC Press; Boca Raton, FL, USA: 2015
- [16] Saba L., Biswas M., Kuppli V., Godia E.C., Suri H.S., Edla D.R., Omerzu T., Laird J.R., Khanna N.N., Mavrogeni S., et al. The present and future of deep learning in radiology. *Eur. J. Radiol.* 2019; 114:14–24. doi: 10.1016/j.ejrad.2019.02.038
- [17] Suri J.S., Biswas M., Kuppli V., Saba L., Edla D.R., Suri H.S., Cuadrado-Godia E., Laird J.R., Marinho R.T., Sanches J.M., et al. State-of-the-art review on deep learning in medical imaging. *Front. Biosci.* 2019; 24:380–406. doi: 10.2741/4725
- [18] Chang P., Grinband J., Weinberg B.D., Bardis M., Khy M., Cadena G., Su M.-Y., Cha S., Filippi C.G., Bota D., et al. Deep-Learning Convolutional Neural Networks Accurately Classify Genetic Mutations in Gliomas. *Am. J. Neuroradiol.* 2018; 39:1201–1207. doi: 10.3174/ajnr.A5667
- [19] Nalawade S., Murugesan G.K., Vejdani-Jahromi M., Fiscaro R.A., Yogananda C.G.B., Wagner B., Mickey B., Maher E., Pinho M.C., Fei B., et al. Classification of brain tumor isocitrate dehydrogenase status using MRI and deep learning. *J. Med. Imaging.* 2019; 6:046003. doi: 10.1117/1.JMI.6.4.046003
- [20] Zhou M., Scott J., Chaudhury B., Hall L., Goldgof D., Yeom K.W., Iv M., Ou Y., Kalpathy-Cramer J., Napel S., et al. Radiomics in Brain Tumor: Image Assessment, Quantitative Feature Descriptors, and Machine-Learning Approaches. *Am. J. Neuroradiol.* 2018; 39:208–216. doi: 10.3174/ajnr.A5391
- [21] Wang Q., Shi Y., Shen D. Machine Learning in Medical Imaging. *IEEE J. Biomed. Health Inform.* 2019; 23:1361–1362. doi: 10.1109/jbhi.2019.2920801
- [22] Srivastava S.K., Singh S.K., Suri J.S. Effect of incremental feature enrichment on healthcare text classification system: A machine learning paradigm. *Comput. Methods Programs Biomed.* 2019; 172:35–51. doi: 10.1016/j.cmpb.2019.01.011
- [23] Shrivastava V.K., Londhe N.D., Sonawane R.S., Suri J.S. A novel and robust Bayesian approach for segmentation of psoriasis lesions and its risk stratification. *Comput. Methods Programs Biomed.* 2017; 150:9–22. doi: 10.1016/j.cmpb.2017.07.011
- [24] Rehman A., Naz S., Razzak M.I., Akram F., Imran M. A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning. *Circuits Syst. Signal Process.* 2020; 39:757–775. doi: 10.1007/s00034-019-01246-3
- [25] Havaei M., Davy A., Warde-Farley D., Biard A., Courville A., Bengio Y., Pal C., Jodoin P.-M., Larochelle H. Brain tumor segmentation with Deep Neural Networks. *Med. Image Anal.* 2017; 35:18–31. doi: 10.1016/j.media.2016.05.004
- [26] Alam F., Rahman S.U., Ullah S., Gulati K. ScienceDirect Medical image registration in image guided surgery: Issues, challenges and research opportunities. *Biocybern. Biomed. Eng.* 2018; 38:71–89. doi: 10.1016/j.bbe.2017.10.001
- [27] Pereira S., Meier R., Alves V., Reyes M., Silva C.A. Understanding and Interpreting Machine Learning in Medical Image Computing Applications: MLCN 2018, DLF 2018 and IMIMIC 2018. Volume 11038.

Springer; Cham, Switzerland: 2018. Automatic Brain Tumor Grading from MRI Data Using Convolutional Neural Networks and Quality Assessment; pp. 106–114. (Lecture Notes in Computer Science)

[28] Alksas, A., Shehata, M., Atef, H., Sherif, F., Alghamdi, N. S., Ghazal, M., Abdel Fattah, S., et al. (2022). A Novel System for Precise Grading of Glioma. *Bioengineering*, 9(10), 532. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/bioengineering9100532>

[29] M. Rizwan, A. Shabbir, A. R. Javed, M. Shabbir, T. Baker and D. Al-Jumeily Obe, "Brain Tumor and Glioma Grade Classification Using Gaussian Convolutional Neural Network," in *IEEE Access*, vol. 10, pp. 29731-29740, 2022, doi: 10.1109/ACCESS.2022.3153108.

[30] Tandel, G. S., Tiwari, A., Kakde, O. G., Gupta, N., Saba, L., & Suri, J. S. (2023). Role of Ensemble Deep Learning for Brain Tumor Classification in Multiple Magnetic Resonance Imaging Sequence Data. *Diagnostics*, 13(3), 481. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/diagnostics13030481>

[31] Ibtesam, A. (n.d.). Image Classification with Attention. Paperspace.
<https://blog.paperspace.com/image-classification-with-attention/>

[32] A. Sekhar, S. Biswas, R. Hazra, A. K. Sunaniya, A. Mukherjee and L. Yang, "Brain Tumor Classification Using Fine-Tuned GoogLeNet Features and Machine Learning Algorithms: IoMT Enabled CAD System," in *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 3, pp. 983-991, March 2022, doi: 10.1109/JBHI.2021.3100758.

[33] Abhilasha, Kumari & Swati, Shipra & Kumar, Mukesh. (2022). Brain Tumor Classification Using Modified AlexNet Network. 10.1007/978-981-19-1018-0_36.

[34] H. Kibriya, M. Masood, M. Nawaz, R. Rafique and S. Rehman, "Multiclass Brain Tumor Classification Using Convolutional Neural Network and Support Vector Machine," 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), Karachi, Pakistan, 2021, pp. 1-4, doi: 10.1109/MAJICC53071.2021.9526262.

[35] Kumar, R & Kakarla, Jagadeesh & Isunuri, B & Singh, Munesh. (2021). Multi-class brain tumor classification using residual network and global average pooling. *Multimedia Tools and Applications*. 80. 10.1007/s11042-020-10335-4.

[36] <https://www.med.upenn.edu/sbia/brats2018/data.html>

[37] Padmanaban, Sriramakrishnan & Thiruvenkadam, Kalaiselvi & T., Padmapriya & Thirumalaiselvi, M. & Sivasakthivel, Ramkumar. (2020). A Role of Medical Imaging Techniques in Human Brain Tumor Treatment. 8. 565-568. 10.35940/ijrte.D1105.1284S219.

[38] <https://pyradiomics.readthedocs.io/en/latest/>

[39] Moon, S.H., Kim, J., Joung, J.G. et al. Correlations between metabolic texture features, genetic heterogeneity, and mutation burden in patients with lung cancer. *Eur J Nucl Med Mol Imaging* 46, 446–454 (2019). <https://doi.org/10.1007/s00259-018-4138-5>

- [40] Choi ER, Lee HY, Jeong JY, Choi YL, Kim J, Bae J, Lee KS, Shim YM. Quantitative image variables reflect the intratumoral pathologic heterogeneity of lung adenocarcinoma. *Oncotarget*. 2016 Oct 11;7(41):67302-67313. doi: 10.18632/oncotarget.11693. PMID: 27589833; PMCID: PMC5341876.
- [41] Sala E, Mema E, Himoto Y, Veeraraghavan H, Brenton JD, Snyder A, Weigelt B, Vargas HA. Unravelling tumour heterogeneity using next-generation imaging: radiomics, radiogenomics, and habitat imaging. *Clin Radiol*. 2017 Jan;72(1):3-10. doi: 10.1016/j.crad.2016.09.013. Epub 2016 Oct 11. PMID: 27742105; PMCID: PMC5503113.
- [42] Gillies RJ, Kinahan PE, Hricak H. Radiomics: Images Are More than Pictures, They Are Data. *Radiology*. 2016 Feb;278(2):563-77. doi: 10.1148/radiol.2015151169. Epub 2015 Nov 18. PMID: 26579733; PMCID: PMC4734157.
- [43] Yip SS, Aerts HJ. Applications and limitations of radiomics. *Phys Med Biol*. 2016 Jul 7;61(13): R150-66. doi: 10.1088/0031-9155/61/13/R150. Epub 2016 Jun 8. PMID: 27269645; PMCID: PMC4927328.
- [44] Mayerhoefer ME, Materka A, Langs G, Häggström I, Szczypiński P, Gibbs P, Cook G. Introduction to Radiomics. *J Nucl Med*. 2020 Apr;61(4):488-495. doi: 10.2967/jnumed.118.222893. Epub 2020 Feb 14. PMID: 32060219; PMCID: PMC9374044
- [45] Zwanenburg A, Vallières M, Abdalah MA, ... Löck S. The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping. *Radiology*. 2020 May;295(2):328-338. doi: 10.1148/radiol.2020191145. Epub 2020 Mar 10. PMID: 32154773; PMCID: PMC7193906.
- [46] Materka A. Texture analysis methodologies for magnetic resonance imaging. *Dialogues Clin Neurosci*. 2004 Jun;6(2):243-50. doi: 10.31887/DCNS.2004.6.2/amaterka. PMID: 22033841; PMCID: PMC3181797.
- [47] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.
- [48] <https://pyradiomics.readthedocs.io/en/latest/features.html#module-radiomics.firstorder>
- [49] Galloway, M. M. (1975). Texture classification using gray level run length. *Comput. Graph. Image Process*, 4(2), 172-179. Galloway, M. M. (1975). Texture classification using gray level run length. *Comput. Graph. Image Process*, 4(2), 172-179.
- [50] Thibault, G., Angulo, J., & Meyer, F. (2013). Advanced statistical matrices for texture characterization: application to cell classification. *IEEE Transactions on Biomedical Engineering*, 61(3), 630-637
- [51] Amadasun, M., & King, R. (1989). Textural features corresponding to textural properties. *IEEE Transactions on systems, man, and Cybernetics*, 19(5), 1264-1274.
- [52] Ray, Sunil. (2017). Top 10 Machine Learning Algorithms (with Python and R Codes). <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>

- [53] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- [54] <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
- [55] <https://scikit-learn.org/stable/modules/svm.html>
- [56] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [57] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>
- [58] <https://lightgbm.readthedocs.io/en/latest/index.html>
- [59] <https://towardsdatascience.com/getting-started-with-xgboost-in-scikit-learn-f69f5f470a97>
- [60] Gupta, Binay K. (2023). What is Stacking in Machine Learning?
<https://www.scaler.com/topics/machine-learning/stacking-in-machine-learning/>
- [61] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
- [62] <https://datagen.tech/guides/computer-vision/ResNet/#>
- [63] <https://medium.com/@godeep48/an-overview-on-mobilenet-an-efficient-mobile-vision-cnn-f301141db94d>
- [64] "MobileNetV2: Inverted Residuals and Linear Bottlenecks" by Mark Sandler, et al., presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2018.
<https://ieeexplore.ieee.org/document/8578572>
- [65] <https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>
- [66] "Densely Connected Convolutional Networks," presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2017. <https://ieeexplore.ieee.org/document/8099726>
- [67] "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," presented at the Conference on Neural Information Processing Systems (NeurIPS) in 2019.
<https://arxiv.org/pdf/1905.11946>
- [68] Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., ... & Wu, Y. (2019). Gpipe: Efficient training of giant neural networks using pipeline parallelism. Advances in neural information processing systems, 32. <https://doi.org/10.48550/arXiv.1811.06965>
- [69] Yan, Y., Kawahara, J., Hamarneh, G. (2019). Melanoma Recognition via Visual Attention. In: Chung, A., Gee, J., Yushkevich, P., Bao, S. (eds) Information Processing in Medical Imaging. IPMI 2019. Lecture Notes in Computer Science(), vol 11492. Springer, Cham. https://doi.org/10.1007/978-3-030-20351-1_62
- [70] Bressler, Noam. (2022). How to Check the Accuracy of Your Machine Learning Model.
<https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning->

[model/#:~:text=Accuracy%20score%20in%20machine%20learning%20is%20an%20evaluation%20metric%20that,the%20total%20number%20of%20predictions](#)

[71] Simplilearn. (2023). What is a Confusion Matrix in Machine Learning?.

<https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning#:~:text=A%20confusion%20matrix%20presents%20a,actual%20values%20of%20a%20classifier>

[72] <https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5>

[73] Srivastava, Niharika. (2023). Training, Validation & Accuracy in PyTorch.

<https://www.e2enetworks.com/blog/training-validation-accuracy-in-pytorch#:~:text=Training%20is%20the%20process%20of,correct%20output%20given%20the%20input>

[74] <https://torchmetrics.readthedocs.io/en/stable/classification/precision.html>

[75] <https://torchmetrics.readthedocs.io/en/stable/classification/recall.html>

[76] https://torchmetrics.readthedocs.io/en/stable/classification/f1_score.html