Assorted Techniques for MRI Brain Tumor Classification

Pranith Koppula, Geonhyeok Jeong, Jason Hu, Oscar Liu, Elena Tsai

Summary: Brain tumor prognosis is highly dependent on early and accurate diagnosis. In an effort to automate tumor classification, various machine learning techniques were applied to sort MRI scans into one of four major categories: no tumor, glioma, meningioma and pituitary. In recognition of resource constraints, a focus was placed on traditional methods (KNN, SVM, etc.) and simple CNNs. KNN, Random Forest, and CNN all performed exceedingly well, though slightly below average manual classification. Differing methods were found to favor identification of different tumor types, reflecting a need to recognize the individual costs of misclassification.

Motivation

According to the National Institute of Health, the accuracy of diagnosing glioma, a type of brain tumor, was around 87% via MRI [3]. Although this percentage may be considered high, we aimed to improve the accuracy even further using machine learning techniques. What we expected to gain from this project was three things. First, it could reduce the workload of doctors. With this technique, doctors would not need to spend time on diagnosis. Alternatively, they could focus on more pressing matters, such as surgical procedures or research. Second, implementing this technology could significantly reduce the cost of medical diagnosis by simply providing images and using a computer. Third, the goal of this technology is to improve patient outcomes. Our expectation is that this technology could potentially achieve higher levels of accuracy in diagnosis and lead to higher rates of survival and recovery for patients.

Although our dataset included images without tumors and images containing other types of tumors, our ultimate goal was to improve the accuracy of diagnosing glioma, which is the most aggressive and dangerous type of brain tumor. To achieve this, we analyzed the data in this order: conducted exploratory data analysis, constructed basic and CNN models, and interpreted the results.

Exploratory Data Analysis

Our dataset consists of 3264 MRI brain scans sourced from Sartaj Bhuvaji on Kaggle, with a seventh of the data consisting of no tumor images, while the rest are equally distributed between glioma, meningioma, and pituitary tumors [1]. While the dataset was pre-divided into a training and test set, due to discrepancies in pixel intensity, the dataset was shuffled and rearranged into a 80:20 training-testing split. In the data cleaning process, all images were rescaled to 100 by 100 pixels for ease of processing. To conduct meaningful summarization of dataset features, dimension reduction was necessary. While various methods such as T-SNE were attempted, the best visual clustering was produced by PCA, as represented by Figure 1. Considering the distinct boundaries between clusters, it appears traditional classification methods such as KNN and SVM will be effective. As a buffer against the curse of high dimensionality, only the first 391 PC scores, which explains 90% of the variation in the images, was used.

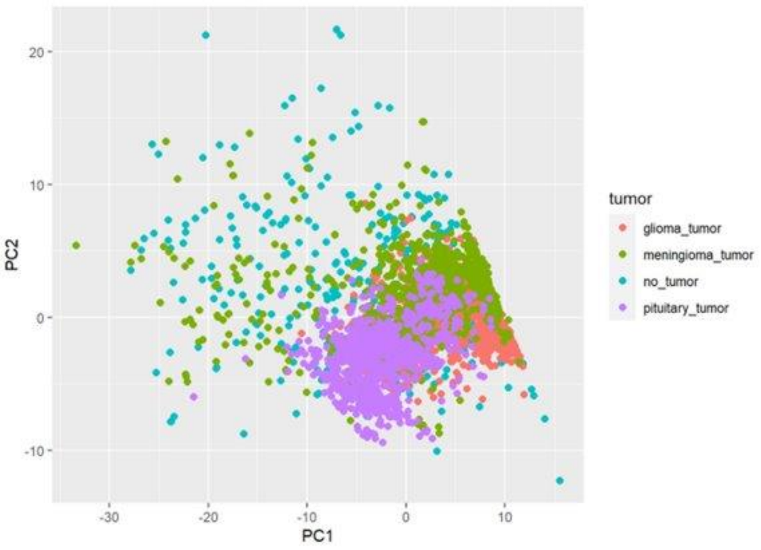


Figure 1. Scatterplot of first two PC scores

Methodology

In order to maximize accuracy, a variety of traditional models, namely KNN, SVM, Naive Bayes, and Random Forest, along with various Convolutional Neural Networks were applied to the dataset.

I. KNN

KNN was computed using PC scores, since proximal pixel intensities are highly correlated. Similarity between data points was measured with Euclidean distance, as PC scores are comparable metrics. The optimal value of k was selected through 5-fold cross-validation, with k ranging from 1 to 101, with a step size of 2 to reduce the uncertainty associated with ties.

Validation accuracy was found to consistently decrease with an increase in k, peaking at 87.8% at k=1. This is a cause for some concern, as it creates a decision boundary that is highly susceptible to noise variance within the training set.

II. SVM

SVM was conducted on the scaled image pixel data as well as the PC scores to identify images based on which tumor class was depicted in them. Two types of kernels were used in modeling: linear and radial. When using PC scores, the number of variables was reduced from 10000 to around 400, both the linear and radial kernel models were run through 10-fold-cross validation with cost ranging from 0.001 to 100 and gamma ranging from 0.5 to 4 (gamma only for radial). Accuracy of the models using PC scores were not very high due to the nature of PCA and losing information in the loss of variables. 10-fold-cross validation was run on both kernel models using the full scaled pixel data. In general, the radial kernel performed better than the linear. Accuracy also increased as cost increased and gamma was low. The best model had a total accuracy of 82.7%, with high sensitivity for glioma, meningioma, and no tumors, and extremely low sensitivity for pituitary tumors. The best model being high cost implies that the model could be prone to overfitting training data.

III. Naive Bayes

Before fitting a Naive Bayes model, We have scaled pixel sizes to 100 * 100 and standardized the data for better modeling. Initially, we attempted to predict the 3 tumors and 1 no tumor class. However, the accuracy from the train data was only around 50%, which was poor. We tried to fit models with different prior probabilities using 10-fold cross-validation but the result was almost either the same or worse. Ultimately, the accuracy from the test data was around 30% indicating that Naive Bayes is not suitable for image analysis. The basic assumption, the features in the data are conditionally, independent given the class variables, was a major obstacle to achieving higher accuracy. Therefore, it is recommended to explore other models that can handle dependencies for better performance in image analysis.

IV. Random Forest

We fit a random forest model to directly predict the 3 tumor classes and the 1 non tumor classes. We did not standardize the data, because it actually made the classifier worse. This is due to the fact that random forest does not require the data to be standardized because it based on an ensemble of weak decision trees, and decision trees don’t require standardization. We found that the model performed well overall achieving about 86% accuracy. We performed GridSearchCV, a cross-validation technique that searches through a grid of combinations of the hyperparameters. With this, we found that the best number of estimators were 16 and the max depth of the trees were 16. Therefore, random forest chose a full set of decision trees. Random forest did very well for a simple algorithm, especially considering that many of the more sophisticated models required a lot more pre-processing and tuning. Some improvements that we did not get to do was to split the decision process into tumor vs no-tumor, and then try to classify the specific tumor from the tumor predictions. This two step classification like transfer learning has proved to work well.

V. CNN

CNN requires many iterations to maximize accuracy, due to the exhaustive number of hyper-parameters involved. As all computations are limited to a CPU, the number of trainable parameters was bound by a maximum of 10,000,000. Thus, our neural networks focus on transfer learning and efficient designs. Transfer learning involves using a model pre-trained on a large, general database to solve a new task such as brain MRI image classification on a smaller target dataset. For our project, we utilized the VGG-16 and ResNet-50 CNNs, both of which were trained on the ImageNet database. The VGG-16 architecture was modified for the new task by freezing the final convolution layers, adding a new model on top of it, and then training it on the target MRI brain dataset. The new model had a global average pooling layer, three dense layers, a dropout layer with a rate of 0.2 and an output layer that used the softmax activation function. The input image data was resized to 150 by 150 pixels and the dataset was split into 64% training, 16% validation and 20% testing.

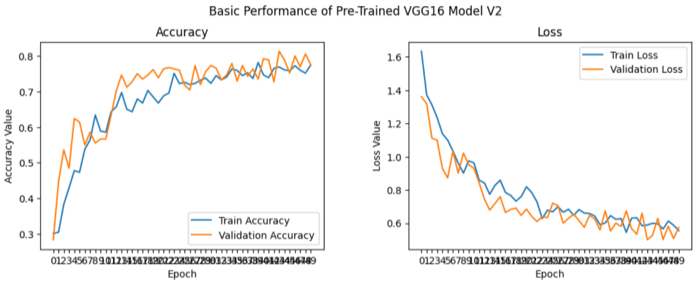


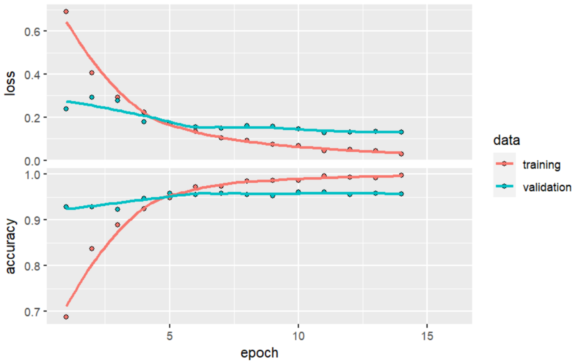
Figure 2. Accuracy and Loss vs Epoch for Training and Validation Sets

VGG-16 has 16 layers (13 convolutional and 3 dense) and roughly 138 million trainable parameters, significantly larger and more complex than ResNet-50 with around 25 million trainable parameters. This contributed to its relatively long training time and possibly impacted its accuracy rate. As seen in Figure 2, the validation accuracy was overall higher than the train accuracy while the validation loss was overall lower than the training loss, potentially due to the dropout layer not being activated during validation. Multiple runs of the VGG16 model had the test accuracy remaining at 74%-78%.

In order to increase the overall depth and accuracy of the CNN, hierarchical deep-learning CNN, a model also originally developed on the ImageNet dataset, was employed to reduce the workload [2]. HD-CNNs first classify images into coarse categories, such as “fruits and vegetables” or “vehicles”, before final classification into a reduced number of fine categories, such as “apples” or “oranges.” Yan et. al contends “that HD-CNN can achieve lower error than the corresponding [regular] CNN, at the cost of a manageable increase in memory footprint and classification time [2].” However, it should be noted that the training time of each individual CNN is reduced, allowing for better fine-tuning. On the basis of semantic and visual differences, as evidenced by the PCA scatterplot, the first CNN will determine the presence of a tumor, while the sequential CNN will classify the tumor as glioma, meningioma, or pituitary.

Since the cost of a false negative is much greater than a false positive for tumor detection, a model that prioritizes sensitivity is preferred. Furthermore, as the accuracy of the sequential CNN is dependent on the performance of the first CNN, a stable validation accuracy is necessary. Thus, transfer learning was used to greatly reduce the number of free parameters. ResNet-50 was chosen due to its relative simplicity and speed, especially compared to other models, such as VGG16. After the adapted convolution layers from ResNet, a single dense layer was implemented to increase the model’s transition capacity. Both batch normalization and dropout (rate = 0.2) were applied to the dense layer to prevent overfitting. The CNN was trained with Adam optimizer, with a reduced learning rate of 0.0001 to deter suboptimal solutions. As shown in Figure X, using batch size = 32, the CNN reaches 96% validation accuracy within a few epochs. However, this model greatly favors positive classification, with a 98% validation accuracy, as opposed to the negative validation classification accuracy of 89%.

Because the fine categories are similar in cost, the sequential CNN will seek to minimize the loss function, cross-entropy. Thus, a custom CNN was built, since it was assumed the greater adaptability would yield a higher accuracy. Due to the significantly larger number of free parameters, architecture design focused on maintaining a stable validation accuracy to prevent statistical artifacts. In order to decrease the number of convolutional layers necessary, images were initially reduced from the 224 by 224 pixels required by ResNet50 to 112 by 112. Each of the four convolutional layers consists of a standard 3 by 3 kernel, batch normalization, and a max pooling layer with padding. The convolution layers are connected to the single dense layer by global average pooling, to reduce the number of required nodes. Once again, the model was trained with Adam optimizer and a learning rate of 0.0001, to further stabilize the validation accuracy. As seen in Figure Y, the model was chosen where validation accuracy appears to stabilize and peak around 93%, between 26 and 29 epochs.



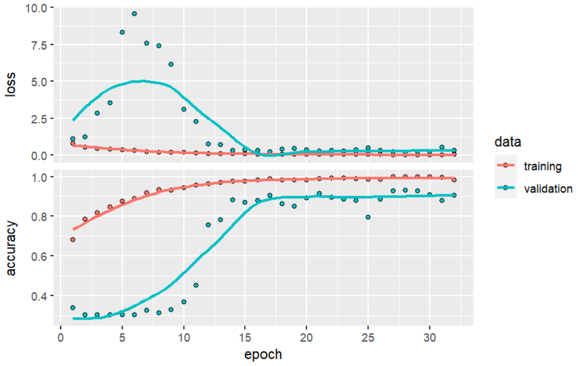


Figure X. Training Loss of First HD-CNN

Figure Y. Training Loss of Sequential HD-CNN

Finally, we also performed an asymmetric CNN model. This model was mainly based on the idea that one of the tumor classes, glioma, was known to be more malignant than the others. Therefore, we wanted to prioritize the recall and precision of this specific class. Thus, we employed an asymmetric loss to penalize the classification of this tumor more when it got it wrong. We did this through developing a simple CNN model with an architecture that was common in practice, but added our own custom cross entropy loss function with a weighting vector to penalize the glioma tumor class much more heavily than the others. We called this hyperparameter k , and this was multiplied to its loss and the others were kept as 1. To find this parameter, the optimal approach would have been to do k-fold cross validation and find which minimized the error for glioma. However due to computational constraints, specifically memory constraints that could not save the weights in RAM for the CNN models, we could do this. So, we ran the model with multiple tries of k ranging from 1-15. We found that 10 was the best in minimizing the error for glioma the best. However, this was a brute force method and more testing needs to be done to completely verify this result. The model performed the best regards to classifying the glioma tumor, achieving an accuracy of approximately 89.1%. However, there were tradeoffs with this model in that it lowered its no tumor classification accuracy. Its overall accuracy was also surprising and did well, achieving 91.1% which was the best across our models. This suggested that defining a loss function that helped to classify the glioma tumors was important and helped to push up the overall accuracy of the model. We also felt this was a good tradeoff as predicting a tumor accurately was much more important than not predicting someone had a tumor when they actually did. This tradeoff between false positives was really important especially in any medical imaging analysis.

Results and Conclusion

Table 1. Testing Set Accuracy

	No Tumor	Glioma	Meningioma	Pituitary	Total
KNN	80.0	88.0	88.6	94.0	89.2
Naive Bayes	30.4	28.0	13.0	74.3	32.7
SVM	98.6	88.4	94.8	64.2	82.7
Random Forest	89.7	79.5	88.2	95.6	87.3
VGG-16	85.0	71.0	73.0	86.0	78.0
HD-CNN	91.5	86.6	87.0	98.9	90.7
Asymmetric Loss	87.4	89.1	89.5	97.1	91.1

Overall, this project tried various methodologies and techniques to find ways to classify brain tumors. Some of the biggest challenges in this project were computational resources, and timing. There was ultimately only a limited amount of items we could do with the amount of time and resources that were available. However, there are still more extensions that we believe could really enhance the project. Firstly, our models have advantages over many complicated and deep architectures in that we save computational resources and time. However, it would be good to help to simplify these architectures while minimizing the amount of error made by the models. Moreover, better preprocessing techniques, such as autoencoders for dimensionality reduction could have helped to further improve computation time and make the models simpler. Furthermore, this application requires the explanation of why classifications are made. The health care application of the models make it imperative to explain in some form why the models make their decisions, maybe through explainable decision trees, and through feature mappings learned through the convolutional layers. These need to be explained in a clear and understandable way.

References

1. https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri
2. https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Yan_HD-CNN_Hierarchical_Deep_ICCV_2015_paper.pdf
3. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7794124/