Classifying Brain Tumors with CNNs

Niyati Prabhu
Department of Computer Science
University of Texas at Austin
Austin, TX 78712
niyatiprabhu@utexas.edu

Monique Tran
Department of Computer Science
University of Texas at Austin
Austin, TX 78712
mktran@utexas.edu

Abstract

We constructed several CNNs and chose the best amongst them for the image classification of four different groups of brain tumors using MRI scans: gliomas, meningiomas, pituitary tumors, and a no tumor group. We then used the best performing model to perform transfer learning on a second dataset, in which the classes were somewhat similar but with greater number and granularity of brain tumor classes. These classes include: gliomas (T1, T1C+, T2), meningiomas (T1, T1C+, T2), neurocitomas (T1, T1C+, T2), schwannomas (T1, T1C+, T2), other (T1, T1C+, T2), and a no tumor control group (T1, T2).

1 Background

Brain tumors are formed by tissue that grows abnormally due to the uncontrolled multiplication of cells [1]. They form abnormal masses in the brain that can either be benign or malignant. Almost 20,000 adults die annually from this disease [2]. Additionally, the number of people who have died from brain tumors has increased as much as 300% in the last 30 years [3].

Brain tumors can either form in the brain (primary tumors), or elsewhere in the body and metastasize to the brain (metastatic tumors) [1].

1.1 Motivation

Diagnosing a brain tumor can be difficult, due to their similarity to other non-neoplastic neurological diseases on neuroimaging. This includes, but is not limited to, strokes, tuberculosis, and toxoplasmosis [4]. The diagnosis of such tumors is heavily reliant upon magnetic resonance imaging, or MRI, which is able to provide more data on their structure and allow for segmentation. However, because the brain is a very intricate organ, even professionals require extensive time to manually diagnose these tumors [2].

Automation of diagnosis could prove incredibly effective in this area [2]. Instead of seeking to replace professionals in the field, they instead offer a second opinion. This will serve to improve the accuracy of professional diagnoses, offload the burden of neurologists so that they can distribute their time to other responsibilities, and eliminate errors made out of fatigue or by mistake [3]. These diseases are difficult to detect, hard to treat at later stages, and highly fatal. Therefore, we seek to develop a model with the potential to improve patient quality of life and longevity.

1.2 Goals

This paper will seek to use deep learning to perform automated brain tumor classification using MRI scans of brain tumors. We aim to 1) gain a greater understanding of how to classify images using CNNs, especially regarding which architectures and approaches work best, 2) practically implement

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

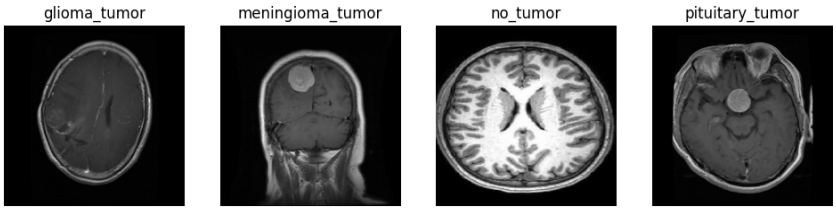

Figure 1: Tumor Groups in Dataset One

techniques that were only discussed in class theoretically, and 3) contribute our own work to a field that is high-impact would greatly benefit from any insight that improve existing deep learning infrastructure being used within the field currently.

1.3 Related Works

- Work [5] utilized CNNs with small kernel sizes to implement automatic detection of brain tumors with 97.5% accuracy. Because this allows for efficient parsing of mass amounts of MRI data, this relieves the strain on medical providers to manually process each image.

- Because medical classification requires very high precision in order to be utilized, [6] tested various neural network architectures, such as ResNet, U-Net, SegNet, and YOLO v3, alongside different data pre-processing methods in order to yield a 94% accuracy.

- Work [7] experimented with multiple neural network architectures, combining the resulting classifier with discrete wavelet transform and principal component analysis to attain high performance in classifying four classes of brain tumors.

2 Datasets

Our first dataset consisted of four different classes of MRI scans – gliomas, meningiomas, pituitary tumors, and a control group with no tumors (Figure 1) [8]. It was pre-separated into training and testing sets, which we preserved (see Figure 1). The training set was mostly evenly distributed with sample sizes of 800, except for the no tumor group which was about only 400 samples large. The testset was also mostly evenly distributed with sample sizes of 100 across each class, with only the pituitary tumor group having about 75 samples.

Our second dataset consisted of seventeen different classes of MRI scans - gliomas, meningiomas, neurocitomas, schwannomas, other tumors, and a no tumor group, each of which was further categorized into three subtypes - T1, T1C+, and T2 (Figure 2) [9]. The dataset - with 4449 total images - was not separated into train and test sets, so we manually enforced an 80-20 split, with 3559 images in the training set and 889 in the test set. The number of images in each class differed greatly, but we left the sample sizes as is to preserve the rarity of each tumor type across the population,

3 Experiment 1 — Model Architecture Exploration

3.1 Justification

Computational Experiment #1 – Model exploration is important, as it allows you to test across a variety of architectures to see what suits a given problem best. The architecture has a significant impact of the model results, therefore weighing different trade-offs and advantages with different approaches is an integral step towards achieving optimal results. Model exploration also fosters innovation by allowing for unconventional methods to be tested or for the discovery of optimal structures for specific problems, which promotes a greater understanding of how neural networks function and drives progress in the field.

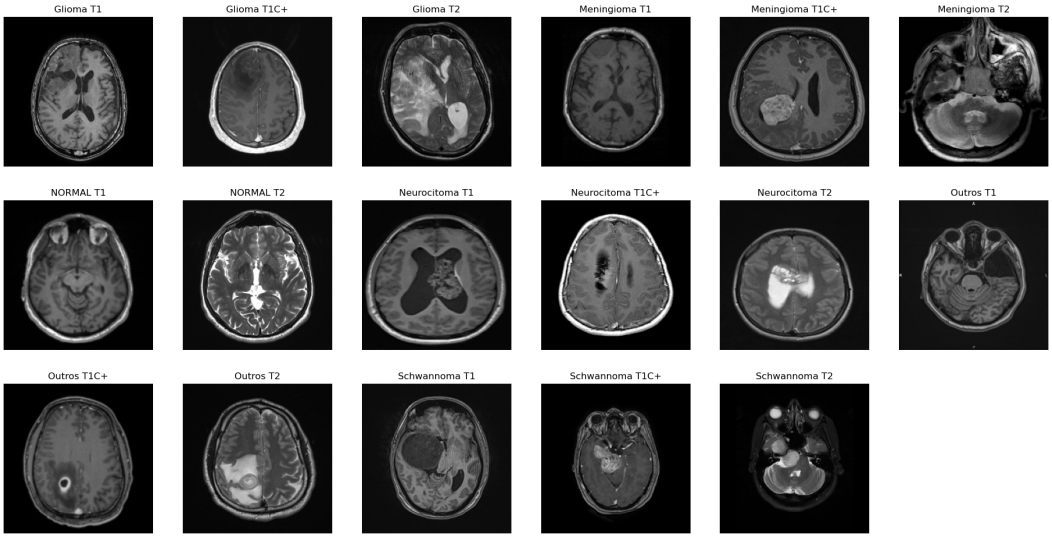

Figure 2: Tumor Groups in Dataset Two

3.2 Hypothesis

We posit that different neural network architectures have varying abilities to learn complex patterns from the MRI scans. We expect that more layers with more kernels and some form of regularization will perform better than models with the contrary.

3.3 Model Construction

First, a vanilla model was constructed that consisted of a single convolutional layer, a max pooling layer, and two fully connected (FC) layers, the last of which collapsed the dimensionality down to the number of classes in the first dataset.

Fifteen additional models were then constructed as derivations of the first model. Changes in the architecture included stacking convolutional and max pooling layers; adding activations between the FC layers; changing the size of the FC layers; changing the number of kernels, padding, stride, and kernel size in the convolutional and max pool layers; changing the activations used on the convolutional layers to leaky ReLU and sigmoid; using normalization such as dropout and batch norm; changing the type of pooling to average pooling; and increasing and decreasing the batch size with the vanilla model. A mixture of these changes were done across all fifteen models. All models were trained with an epoch size of 30 due to computational cost trade-offs in this process. The variations in the model were selected for to test 1) how a change in architecture reflected a change in performance, i.e. small variations between models were tested to observe how certain architectures performed, and 2) to use models with significant differences to exhaustively explore a wider range of possibilities, given time constraints.

3.4 Results

Most models performed equally well, achieving an accuracy around 0.7 and an F1-score around 0.65. The performance of our best model achieved an accuracy of 0.7, an F1-score of 0.66, recall of 0.7, and precision of 0.77 (Figure 3). We selected for the best model based on F1-score because it provides a balance between precision and recall. We value precision because we want to make sure that when our model predicts a patient's MRI positively, it is correct; we value recall because we want to make sure that we can identify as many patients with tumors as possible.

We believe our models achieved similar performance due to 1) a lack of augmentation to the input — that is, we did not alter or modify the images in any way that might've allowed the models to pick up on other patterns, and 2) the limitation to the epoch size, which may have not have allowed the models enough time to diverge in performance.

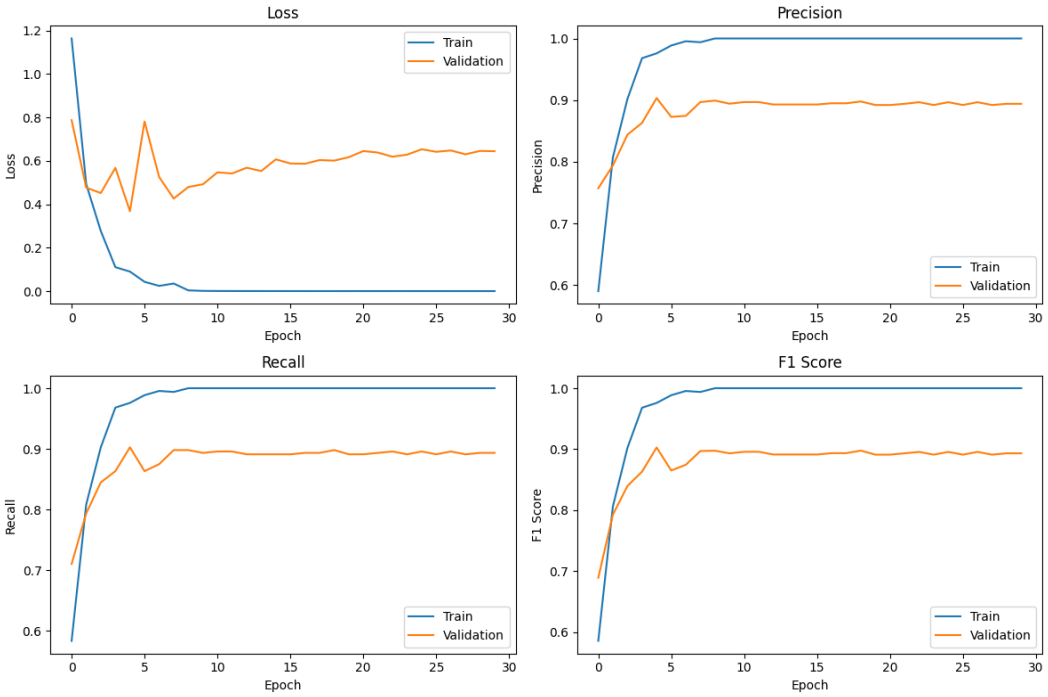

Figure 3: Results of Best-Performing Model

The poorest results we received were through the use of the sigmoid function, which for replication purposes we advise against testing, as it is only suitable for binary classification problems, of which this dataset is not.

In terms of result interpretation, our best-performing model correctly predicts 70% of the instances across all classes; 77% of the instances it predicts as positive for a certain class are actually true positives; and it identifies 70% of the positive instances correctly.

3.5 ChatGPT

Outputs from ChatGPT were used for architectural variations 13 and 14 in our experiment.

4 Experiment 2 — Transfer Learning

4.1 Justification

Computational Experiment #2 – Transfer learning as a methodology is a great tool to evaluate the generalizability of an existing model in solving a different problem than the one it was originally geared to solve. It is also invaluable in identifying the kinds of problems a given model is capable of adapting to. It answers the question - what other problems are within the model's reach? Furthermore, variation in fine-tuning techniques can offer a multitude of models with differing levels of specificity and flexibility.

4.2 Hypothesis

Because training the model on the new dataset allows for more specific feature detection, we expect that the model will work increasingly well as we raise the number of parameters that are retrained with the new dataset. Using the original model "off-the-shelf" will likely be unsuccessful in solving the problem.

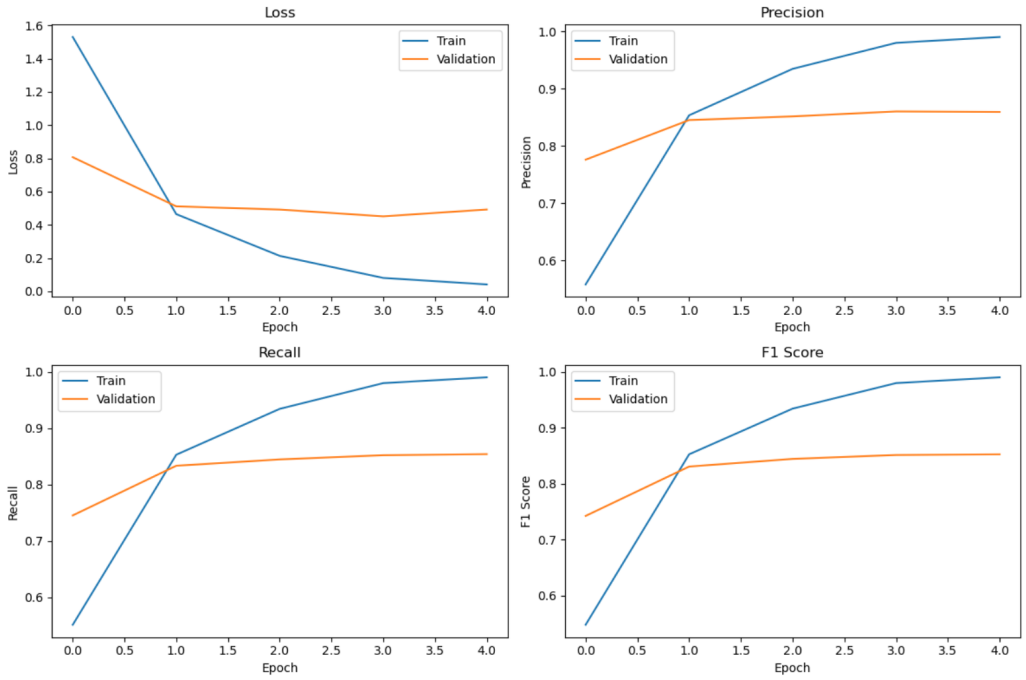

Figure 4: Results of Second Approach

4.3 The Three Approaches

- **Off-the-shelf**: In this approach, we used the learned parameters from the final model from Experiment 1, as is, in a new model for the transfer learning problem. We observed the effectiveness of retaining all learned parameters of the model, only replacing the "classification head" to represent a new number of classes.
- **Freeze all pre-trained**: In this approach, we used the same pre-trained model with all learned parameters preserved from Experiment 1. This time, before feeding the model data from the new dataset, we trained the parameters of the modified output layer on new data for 5 epochs. The remaining pre-trained parameters remained as they were, i.e. "frozen". Finally, we tested the trained model to observe any improvements from the previous iteration.
- **Gradual unfreezing**: In this approach, we tried a fine-tuning method involving "gradual unfreezing" of the frozen pre-trained layers from Experiment 1. The new output layer of the model began as the only unfrozen layer. Then, during the training process (5 epochs), the parameters of the next outer-most layers of the model were iteratively unfrozen, allowing the model more flexibility in learning features that are specific to this classification problem. The "unfreezing" process occurs on layers in reverse order because features learned by the parameters become less broad and less generalizable moving through the model. The parameters towards the end should be malleable to adapt to the specifics of the new problem. Parameters at the beginning of the model should remain frozen because they capture the broad similarities between the two problems.

4.4 Results

As anticipated, the initial approach of testing the model from Experiment 1 "off-the-shelf" to solve a different classification problem was unsuccessful. Across multiple testing iterations, this approach achieved somewhere between 4 and 12 % accuracy on average. Evidently, the two classification problems were not quite similar enough to warrant using the same model without any additional training with new data.

The second approach of maintaining all pre-trained parameters from the Experiment 1 model as "frozen" had remarkable performance - considerably better than the level we expected. An average

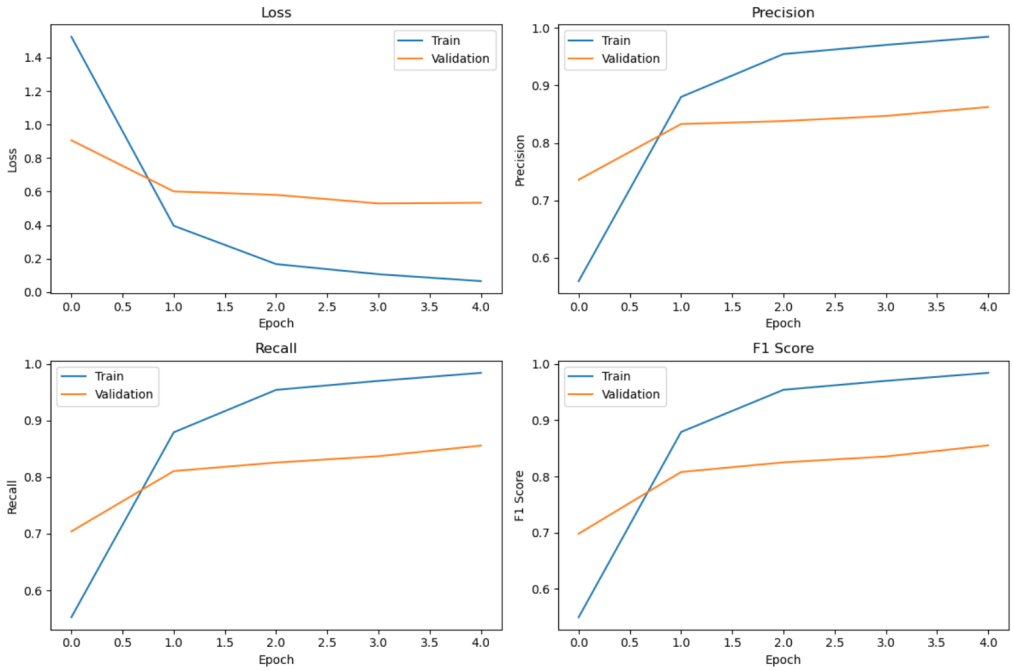

Figure 5: Results of Third Approach

F1 score, precision, and recall between 87 and 88% was achieved by only training the output layer of the model on data from the second dataset.

The third approach of gradually unfreezing parameters starting at the back of the model performed almost identically to the model from the second approach. The attained F1 score, precision, and recall also averaged between 87 and 88% for this model.

Tuning the hyperparameters of the model could change the performance of each approach. Increasing the number of epochs could reveal a greater difference between the latter two models' performances. Additionally, modifying the frequency of "unfreezing" - changing the number of remaining frozen parameters - offers less/additional flexibility for the new classification problem.

4.5 ChatGPT

Outputs from ChatGPT were used for ideation of the three approaches listed above and in aiding the implementation of the final fine-tuning approach.

References

[1] Abd-Ellah, M. K., Awad, A. I., Khalaf, A. a. M., & Hamed, H. F. A. (2019). A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. *Magnetic Resonance Imaging, 61*, 300–318. https://doi.org/10.1016/j.mri.2019.05.028

[2] Hu, A., & Razmjooy, N. (2020). Brain tumor diagnosis based on metaheuristics and deep learning. *International Journal of Imaging Systems and Technology, 31*(2), 657–669. https://doi.org/10.1002/ima.22495

[3] El-Dahshan, E. A., Mohsen, H., Revett, K., & Salem, A. M. (2014). Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm. *Expert Systems With Applications, 41*(11), 5526–5545. https://doi.org/10.1016/j.eswa.2014.01.021

[4] Herholz, K., Langen, K., Schiepers, C., & Mountz, J. M. (2012). Brain tumors. *Seminars in Nuclear Medicine, 42*(6), 356–370. https://doi.org/10.1053/j.semnuclmed.2012.06.001

[5] Seetha, J., & Raja, S. S. (2018). Brain Tumor Classification Using Convolutional Neural Networks. In Biomedical and Pharmacology Journal (Vol. 11, Issue 3, pp. 1457–1461). Oriental Scientific Publishing Company. https://doi.org/10.13005/bpj/1511

[6] Kondratenko, Y., Sidenko, I., Kondratenko, G., Petrovych, V., Taranov, M., Sova, I. (2021). Artificial Neural Networks for Recognition of Brain Tumors on MRI Images. In: Bollin, A., *et al.* Information and Communication Technologies in Education, Research, and Industrial Applications. ICTERI 2020. Communications in Computer and Information Science, vol 1308. Springer, Cham. https://doi.org/10.1007/978-3-030-77592-6_6

[7] Mohsen, H., El-Dahshan, E.-S. A., El-Horbaty, E.-S. M., & Salem, A.-B. M. (2018). Classification using deep learning neural networks for brain tumors. In Future Computing and Informatics Journal (Vol. 3, Issue 1, pp. 68–71). Future University in Egypt. https://doi.org/10.1016/j.fcij.2017.12.001

[8] *Brain Tumor Classification (MRI)*. (2020, May 24). Kaggle. https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri/data

[9] *Brain Tumor MRI Images 17 Classes*. Kaggle. https://www.kaggle.com/datasets/fernando2rad/brain-tumor-mri-images-17-classes

5 Contributions

Niyati Prabhu – Dataset two preprocessing, visualization, and write up; dataset two and experiment two write up; creating modified training function; implementing and testing three different transfer learning methods for experiment two

Monique Tran – Dataset one preprocessing, visualization, and write up; background, dataset one, and experiment one write up; creating train/validation/test functions; designing and running different models for experiment one

6 Model Card for Experiment One

6.1 Model Details

This CNN is designed to classify four different brain tumors. It takes images with 3 channels as input and predicts one of four classes. The architecture consists of two convolutional layers followed by average pooling, then two fully connected layers.

- **Developed by:** Monique Tran
- **Model type:** Convolutional neural network
- **Task:** Image classification

6.2 Uses

The model is intended to be used for the classification of brain tumors from MRI scans. Its main audience would likely consist of medical professionals such as radiologists, oncologists, and neurosurgeons who routinely analyze MRI images for diagnosis and treatment planning. Researchers who reside in the intersection of medical imaging and machine learning might also utilize the model for their studies and experiments.

6.3 Bias, Risks, and Limitations

This model provides predictions, not definitive diagnoses. It should be used to supplement professional decision-making rather than a substitute.

The model may not perform well for rare or uncommon brain tumor types due to limited training data.

6.4 Recommendations

Patients diagnosed with brain tumors rely on accurate and timely diagnoses for appropriate treatment planning. Therefore, the model's performance directly affects patient outcomes and quality of life. It's imperative that this model is tested more thoroughly before being deployed in clinical settings.

6.5 Data

See section 7.

6.6 Training Hyperparameters and Procedure

The model was tested for 30 epochs, using cross entropy loss and the Adam optimizer, with a learning rate of 0.001. The validation score was computed in each epoch.

6.7 Evaluation

The model was tested after being trained for 30 epochs on an unseen test set.

6.8 Metrics

We selected this model as our best model based on its F1-score, because this metric provides a balance between precision and recall. We value precision because we want to make sure that when our model predicts a patient's MRI positively, it is correct; we value recall because we want to make sure that we can identify as many patients with tumors as possible. We also considered accuracy, but moreso for interpretability.

6.9 Results

This model achieved an accuracy of 0.7, an F1-score of 0.66, recall of 0.7, and precision of 0.77. This means that it correctly predicts 70% of the instances across all classes; 77% of the instances it predicts as positive for a certain class are actually true positives; and it identifies 70% of the positive instances correctly.

7 Datasheet for Experiment One

Who created the dataset? Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, with acknowledgements to Navoneel Chakrabarty and Swati Kanchan

How many instances are there in total? 3264 files

Is there a label or target associated with each instance? Yes, each image is classified into four groups: gliomas, meningiomas, pituitary tumors, and a no tumor control group

Are there recommended data splits? Yes, the data comes pre-partitioned into training and test sets with about a 15% split for the test set.

Does the dataset contain data that might be considered confidential? Yes, these scans are released anonymously, but were at some point bound between doctor-patient confidentiality.

Does the dataset identify any subpopulations? No, from the MRI scans gender or age cannot be determined.

How was the data associated with each instance acquired? This is unknown.

Over what timeframe was the data collected? This is unknown.

Were any ethical review processes conducted? This is unknown.

Has the dataset been used for any tasks already? Yes, it is a popular dataset that has been used by several other researchers to build various neural networks.

Is there a repository that links to any or all papers or systems that use the dataset? https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri/code

How can the owner/curator/manager of the dataset be contacted? Through Github or Kaggle

8 Model Card for Experiment Two

8.1 Model Details

This CNN is designed to classify six different brain tumor groups, with each having three subgroups. It takes input in the form of MRI images with 3 channels and predicts one of seventeen classes. The architecture consists of two convolutional layers followed by average pooling, then two fully connected layers.

The model is built atop a pre-trained model for solving a broader brain tumor classification problem with fewer classes. With existing parameters frozen as is, the final output layer is trained with new data. The model's parameters may also be gradually unfrozen to accommodate for greater flexibility.

- **Developed by:**Niyati Prabhu
- **Model type:** Convolutional neural network
- **Task:** Image classification

8.2 Uses

The model is intended to be used for the classification of brain tumors from MRI scans. Its main audience would likely consist of medical professionals such as radiologists, oncologists, and neurosurgeons who routinely analyze MRI images for diagnosis and treatment planning. Researchers who reside in the intersection of medical imaging and machine learning might also utilize the model for their studies and experiments.

8.3 Bias, Risks, and Limitations

This model provides predictions, not definitive diagnoses. It should be used to supplement professional decision-making rather than a substitute.

The model may not perform well for rare or uncommon brain tumor types due to limited training data - specifically, for the most underrepresented tumor classes in the dataset (see section 9).

8.4 Recommendations

Patients diagnosed with brain tumors rely on accurate and timely diagnoses for appropriate treatment planning. Therefore, the model's performance directly affects patient outcomes and quality of life. It's imperative that this model is tested more thoroughly before its reliance in clinical settings.

8.5 Data

See section 9.

8.6 Training Hyperparameters and Procedure

The model was trained for 5 epochs, using cross entropy loss and the Adam optimizer, with a learning rate of 0.001. The validation score was computed in each epoch, using a validation set of 15% and training set of 85%.

8.7 Evaluation

The model was tested on 20% of the dataset after being trained for 5 epochs on the remaining 80% of the dataset. The tested data is unseen during the training procedure.

8.8 Metrics

Fine-tuning the model by training the output layer on new data allowed the model to obtain F1 score, precision, and recall. We value precision so that diagnoses are not unnecessarily issued; we value recall in order to identify as many patients with tumors as possible. We also considered accuracy, but this was mainly for interpretability.

8.9 Results

This model achieved an accuracy of 0.87, an F1-score of 0.87, recall of 0.87, and precision of 0.88. This means that it correctly predicts 87% of the instances across all classes; 87% of the instances it predicts as positive for a certain class are actually true positives; and it identifies 88% of the positive instances correctly.

9 Datasheet for Experiment Two

Who created the dataset? Fernando Feltrin

How many instances are there in total? 4449 files

Is there a label or target associated with each instance? Yes, each image is classified into seventeen groups: gliomas (T1, T1C+, T2), meningiomas (T1, T1C+, T2), neurocitomas (T1, T1C+, T2), schwannomas (T1, T1C+, T2), other (T1, T1C+, T2), and a no tumor control group (T1, T2).

Are there recommended data splits? No, but we chose to randomly split the data into 80% for training and 20% for testing.

Does the dataset contain data that might be considered confidential? Yes, these scans are released anonymously, but were at some point bound between doctor-patient confidentiality.

Does the dataset identify any subpopulations? No, from the MRI scans gender or age cannot be determined.

How was the data associated with each instance acquired? The data for each instance was exported from MRI workstations.

Over what timeframe was the data collected? This is unknown.

Were any ethical review processes conducted? This is unknown.

Has the dataset been used for any tasks already? Yes, it has been used by a few others to conduct classification experiments.

Is there a repository that links to any or all papers or systems that use the dataset? https://kaggle.com/datasets/fernando2rad/brain-tumor-mri-images-17-classes

How can the owner/curator/manager of the dataset be contacted? Through Github or Kaggle