Brain Tumor Detection: Machine Learning Classification

University of Wisconsin - Madison

Computer Science 539 - Introduction to Neural Networks

Group 6

Nicholas Beninato - Department of Computer Sciences - Undergraduate
Sebastian Murrell - Department of Computer Sciences - Undergraduate
Garrison Waugh - Department of Computer Sciences - Undergraduate

Thursday, December 15th, 2022

1 - Abstract

The goal of this project is to create a convolutional neural network that will be trained for multi-class classification on several images of human brain MRIs. This image set contains MRIs of three types of brain tumors–glioma, meningioma, and pituitary–and MRIs without tumors. The ultimate goal is the classification of glioma, meningioma, pituitary, or no tumor. To start, we followed the example model from Kumar (2021) on Kaggle to implement our own CNN to perform the binary classification if an MRI has a tumor. After we had this model working, we developed a multi-class model to classify the type of tumor.

2 - Introduction

Medical imaging analysis, while not a new field, is certainly a growing discipline with enormous potential for saving lives. With that being said, this abundant growth has not come without shadows of doubt around the technology — and for good reason. It is no surprise that patients would want another human to review MRI images, cell biopsies, or other scans to assure their cancer diagnosis, rather than relying on a pre-trained algorithm to confirm their diagnosis. In the United States today, cancer of all forms remains one of the leading causes of death across all cohorts (Centers for Disease Control and Prevention 2022). And, while a doctor is still a necessary middle-man when diagnosing cancer with an algorithm, new research suggests that machine learning algorithms can diagnose diseases consistently with medical professionals (Varoquaux & Cheplygina 2022). Promising results with other diseases begs the question: why not apply deep learning methods to cancer image classification?

The current workflow when it comes to a brain tumor diagnosis typically begins with a series of neurological tests to assess brain function regarding balance, vision, and reflexes. Then, based on how these tests go, a doctor will order a scan of some sort (typically an MRI) and review the imaging results to determine if there is a tumor present in the brain. Then, finally, the doctor may order a biopsy of the suspected tumor to assess its type and whether or not it is cancerous (Johns Hopkins Medicine 2022).

It is easy to see where one may apply deep learning principles here. Instead of having doctors review these MRI images, healthcare professionals could instead look to deep learning algorithms to help assess these images. In fact, a recently published study found that deep learning models trained for brain tumor classification performed better in classifying and identifying brain tumors than neuroradiologists. Additionally, neuroradiologists that were provided with this same model outperformed their counterparts without the model in accuracy of diagnosing brain tumors (Gao, et al 2022).

3 - Data

Original Datasets

Our dataset is a compilation of three existing datasets from Kaggle and FigShare. Altogether, it contains 7022 MRI brain scans in a JPEG format. This directory contains 4 sub-directories, each of which corresponds to one of three types of brain cancer, with the fourth and final sub-directory containing images of brain scans with no brain tumor present. This collection of images is a composite of 3 separate data repositories found online, each one providing varying brain scans:

Dataset 1: Brain Tumor Dataset (Cheng 2017)

Cheng's provides a large portion of the entire sample. The imaging comes from brain scans of 233 different patients with different forms of brain tumors: meningioma (n=708), glioma (n=1426), and pituitary tumor (n=930).

Dataset 2: Brain Tumor Classification (Bhuvaji, et al. 2020)

This dataset contains several more samples from each of the tumor types, as well a control set. The control samples (n=500) come from Navoneel Chakrabarty's Kaggle dataset (Chakrebarty 2018). The remaining samples (meningioma (n=937), glioma (n=926), pituitary (n=901)) come from a private dataset published by Swati Kanchan, though these images are publicly available in the main dataset above.

Dataset 3: Brain Tumor Detection 2020 (Hamada 2020)

This dataset contains data for binary classification of brain tumors. As such, we are only going to be using the "no tumor" subset (n=1500) in order to expand the size of our control group. We are opting not to use the images labeled as having a brain tumor as it would require a professional to manually label 1500 more images of brain tumors.

Combined Dataset

To generate our data, we combined these 3 datasets. We went through each, which were composed of various formats, such as jpeg, matlab, and png, with many different shapes and channel numbers. As such, our first step was to make our data consistent in format and shape. In terms of format, we needed to first convert our .mat files (from the figshare dataset) into JPEG images. Following this, our newly converted JPEG images, along with several images from the other datasets, contained only one channel as they were mapped to be grayscale. To solve this, we duplicated the channels of grayscale images so that all images contain three channels. Our final challenge was regarding the shape of the images. Images originally ranged in size from around 100 x 100 to over 1300 x 1300. Additionally, not all of the images were square in nature. We used OpenCV to resize all images to 150 x 150 x 3. Once we had all images in a consistent format, we flattened them and saved them as a csv with the label as the first column and each pixel in their own column. Finally, we split this at a 75% / 25% ratio for training and testing. Examples of brains with tumors are found in Figures 3.1, 3.2, and 3.3, while Figure 3.4 presents a healthy brain.

Data Example (MRI Images)

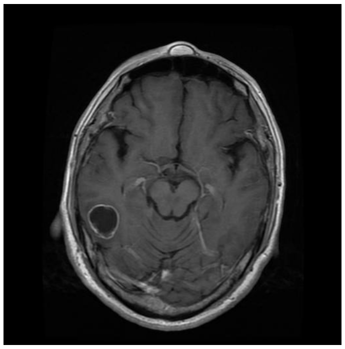


Figure 3.1:
Glioma Tumor

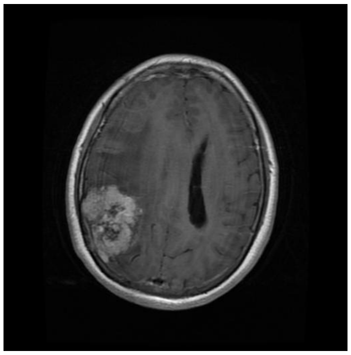


Figure 3.2:
Meningioma Tumor

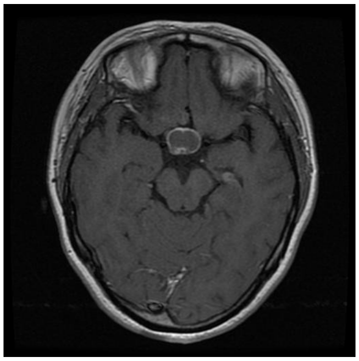


Figure 3.3:
Pituitary Tumor

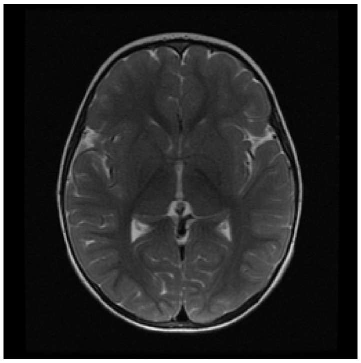


Figure 3.4:
Healthy Brain (No Tumor)

Tumors appear as large white masses superimposed on the surfaces of the brains. Due to this phenomena, our model is able to achieve relatively high accuracy through the use of average pooling to detect significant changes in grayscale coloring for each convolution.

4 - Tasks Performed

Model

For our model, we constructed and trained a convolutional neural network (CNN) that is able to successfully classify the existence of tumors as well as their type from an MRI image. Currently, there is a large amount of work available online in the field of tumor classification, most of which takes a similar approach to that of our group. As mentioned in the introduction, this is a widely used method in the medical field due to its accuracy and cost-effectiveness. For example, the National Institute of Health released a study outlining the use of CNN's in tumor identification, specifically highlighting the contrasts between discriminate image analysis such as K-Nearest Neighbors (kNN), Artificial Neural Networks (ANN) and CNN's (Díaz-Pernas, et al. 2021).

Our current iteration of this model uses multi-class classification to determine presence of tumors in our MRI scans. We have taken inspiration from the works of Jay Kumar and implemented transfer learning in the development of this classifier which is built on top of EfficientNetB0. We chose this model as a base for our classifier as it has already been trained on Computer Vision benchmarks, so the ImageNet weights will be useful in any image classification. By fine tuning to the specifics of brain scans using this pretrained model, we hope to experience much more efficient training times than if we were to start from scratch.

After we completed the easier goal of binary classification, we modified our model to then perform multi-class classification on MRI images. Through using transfer learning, we were able to easily freeze our trainable layers, and modify the loss functions, dense nodes, and layer types to convert it from binary to multi-class classification. The total number of parameters is visible in Figure 4.1.

Our model used a base of EfficientNetB0, which we then added more additional layers to fine tune the model. These include GlobalAveragePooling2D, Dropout (for training only), and a Dense layer with softmax activation. The model was compiled with categorical cross entropy loss and optimized with Adam.

Total Params:	4,054,695
Trainable Params:	4,012,672
Non-trainable Params:	42,023

Figure 4.1: Model Summary

Platform

For this project, we have written the majority of our code in IPython-Notebooks (ipynb) hosted in Google Colab. Hosting our ipynb in Colab allows them to be collaborative, whereas an ipynb hosted on a virtual machine or on a local machine will not allow for collaborative development. Additionally, the use of notebooks as opposed to modules allows us to make small changes to our model efficiently. This code has also been uploaded to our GitHub repository (here). In addition to the code for our model, the code used for transforming our original files into the consistent images used for training, evaluation, and testing has also been uploaded to the GitHub repository.

4 - Results

Our baseline comes from Kumar's (2021) Kaggle Project, which achieved accuracy of 98%. Using their code, we were able to replicate the same results on our own dataset. The baseline uses the EfficientNetB0 model with weights from ImageNet.

Figure 4.4 highlights the successes we had with our binary classification model, while Figure 4.5 explores the statistics and deeper metrics associated with our binary classification model. This initial model scored an accuracy of greater than 99% on testing data drawn from our initial 75%/25% train-test-split.

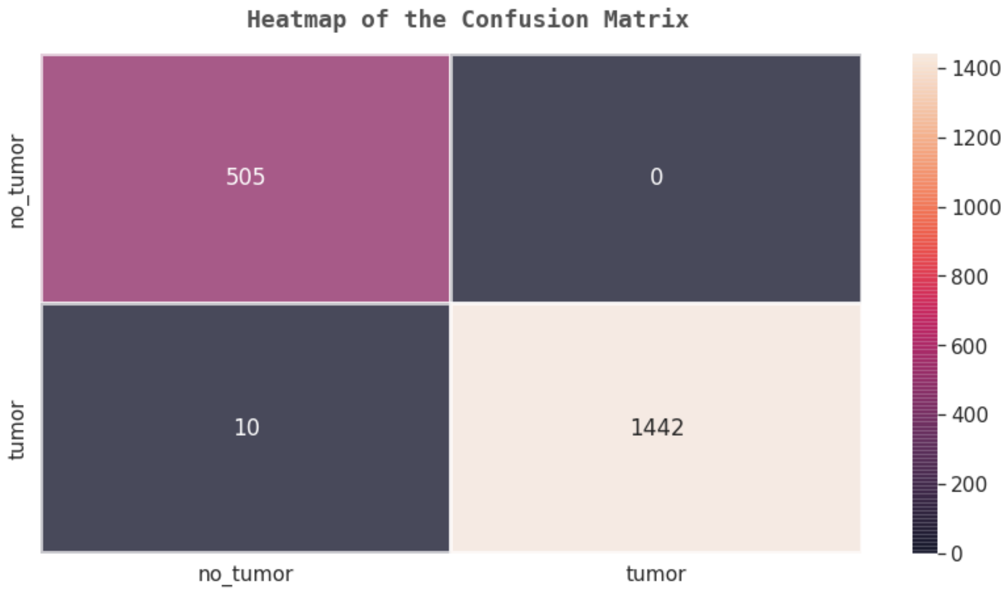
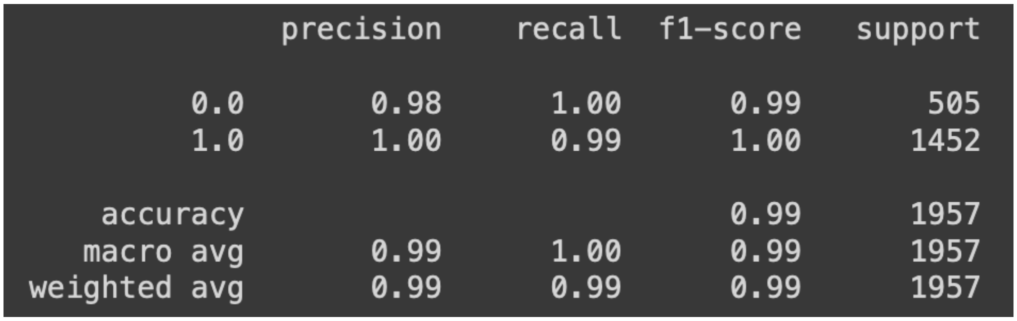


Figure 4.2: Confusion Matrix for Binary Classification



```
              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99       505
         1.0       1.00      0.99      1.00      1452

    accuracy                           0.99      1957
   macro avg       0.99      1.00      0.99      1957
weighted avg       0.99      0.99      0.99      1957
```

Figure 4.3: Binary Model Classification Metrics

Figure 4.4 illustrates our transition from binary classification to multi-class classification. It evidences our success in accurately labeling tumors, and further reveals that the primary difficulty that our model had was in falsely labeling meningioma tumors as being glioma. Again, the second figure, Figure 4.5, shows the underlying class-specific metrics for our model, as well as it overall performance of 97% accuracy on testing data.

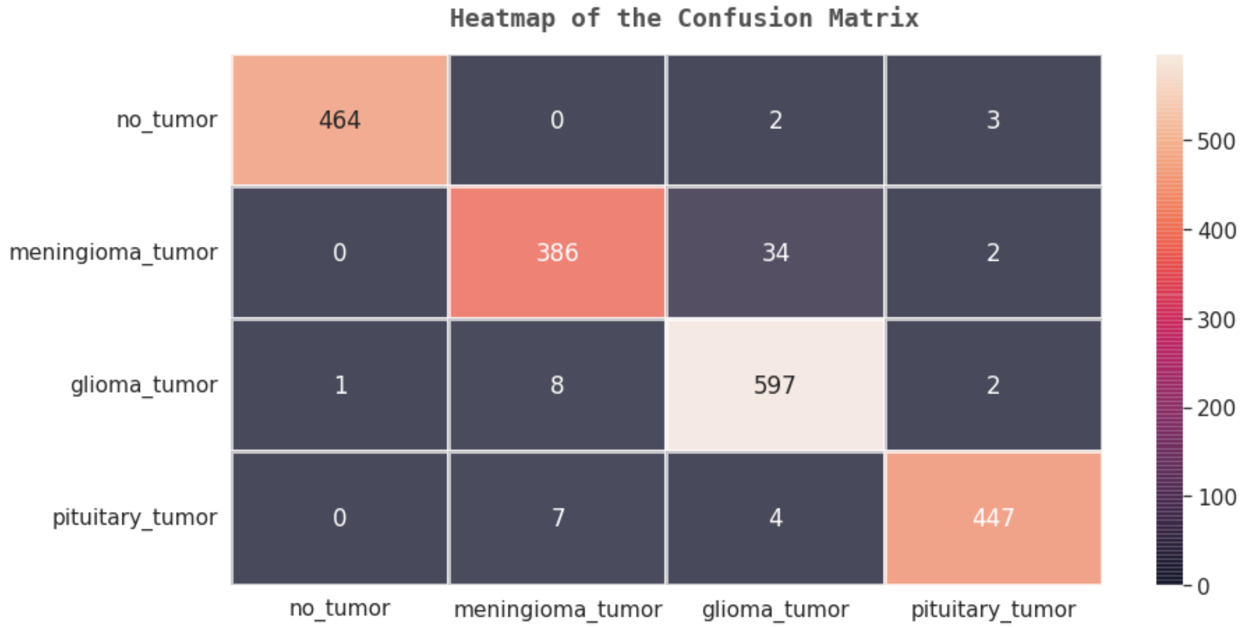
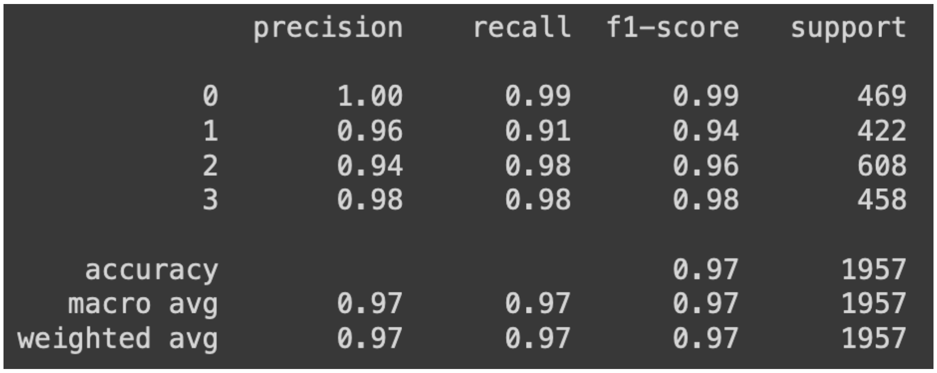


Figure 4.4: Confusion Matrix for Multi-Class Model



```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       469
           1       0.96      0.91      0.94       422
           2       0.94      0.98      0.96       608
           3       0.98      0.98      0.98       458

    accuracy                           0.97      1957
   macro avg       0.97      0.97      0.97      1957
weighted avg       0.97      0.97      0.97      1957
```

Figure 4.5: Multi-Class Model Classification Metrics

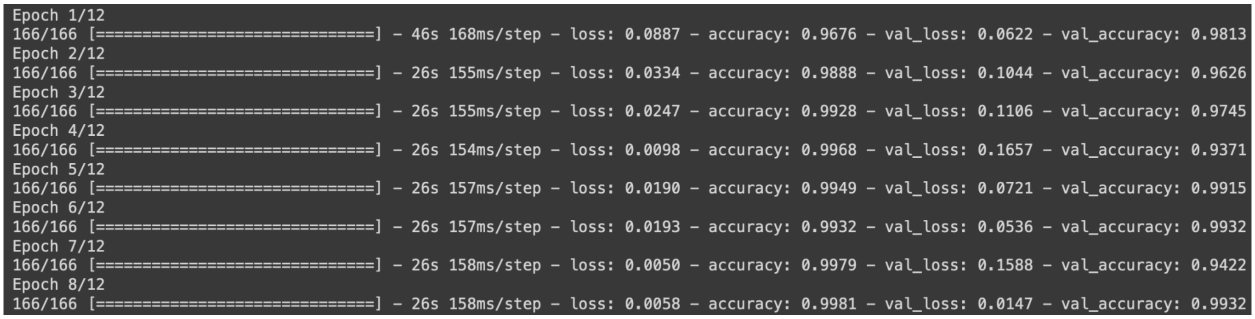
Our model had extremely high accuracy for the binary classification of tumors, and only slightly lower for the classification of which type of tumor. We believe that this reason for such high accuracy comes from the transfer learning of EfficientNetB0. Since the model comes pre-trained on ImageNet, the accuracy after the first epoch was 97%. With such a high starting classification rate, the model was still able to make gains from its 12 epochs of training, ending at a misclassification rate of less than 0.01.

5 - Discussion

Our initial binary classification model worked exceptionally well at distinguishing between brain scans with tumors and without tumors. The final iteration of our model demonstrated a transition from a binary classifier to multi-class classifier. This meant that we retrained our model on the four classes in the additional layers, and also opted for a categorical cross-entropy loss function to better suit the training of such a model. Since the model was at a place where it was able to determine the presence of tumors, the ability to distinguish the type of tumors only involved the addition of layers while maintaining the current weights for binary classification.

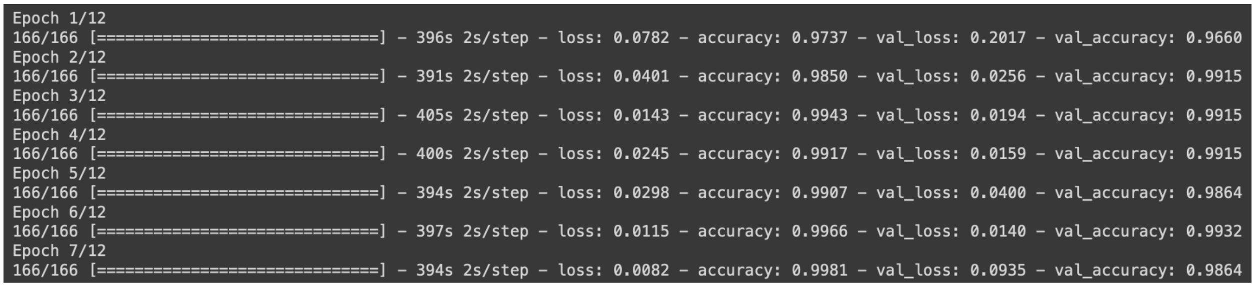
We did encounter some difficulties training our model. The free version of Google Colab did not have enough ram to support learning on our combined csv of all three datasets. We suspect this is because there were over 7000 total images. We overcame this issue by recruiting a GCP virtual machine with a 13GB graphics processing unit. This took our training time per epoch down from 7 minutes to 20 seconds and enabled us to make improvements to our code at a much faster rate. This can be seen in the images at the end of this section.

Overall, we were only 1% lower than our baseline in terms of accuracy. This is likely due to us using a much larger dataset. It could also be due to random chance, as the difference between 97% and 98% is not much. If we had more time, we could explore more options for trying to get even higher accuracy, such as rotating images for augmentation, or using grid search for hyper parameter tuning.



```
Epoch 1/12
166/166 [==============================] - 46s 168ms/step - loss: 0.0887 - accuracy: 0.9676 - val_loss: 0.0622 - val_accuracy: 0.9813
Epoch 2/12
166/166 [==============================] - 26s 155ms/step - loss: 0.0334 - accuracy: 0.9888 - val_loss: 0.1044 - val_accuracy: 0.9626
Epoch 3/12
166/166 [==============================] - 26s 155ms/step - loss: 0.0247 - accuracy: 0.9928 - val_loss: 0.1106 - val_accuracy: 0.9745
Epoch 4/12
166/166 [==============================] - 26s 154ms/step - loss: 0.0098 - accuracy: 0.9968 - val_loss: 0.1657 - val_accuracy: 0.9371
Epoch 5/12
166/166 [==============================] - 26s 157ms/step - loss: 0.0190 - accuracy: 0.9949 - val_loss: 0.0721 - val_accuracy: 0.9915
Epoch 6/12
166/166 [==============================] - 26s 157ms/step - loss: 0.0193 - accuracy: 0.9932 - val_loss: 0.0536 - val_accuracy: 0.9932
Epoch 7/12
166/166 [==============================] - 26s 158ms/step - loss: 0.0050 - accuracy: 0.9979 - val_loss: 0.1588 - val_accuracy: 0.9422
Epoch 8/12
166/166 [==============================] - 26s 158ms/step - loss: 0.0058 - accuracy: 0.9981 - val_loss: 0.0147 - val_accuracy: 0.9932
```

Figure 5.1: Training with GCP virtual machine (~26 s / epoch)



```
Epoch 1/12
166/166 [==============================] - 396s 2s/step - loss: 0.0782 - accuracy: 0.9737 - val_loss: 0.2017 - val_accuracy: 0.9660
Epoch 2/12
166/166 [==============================] - 391s 2s/step - loss: 0.0401 - accuracy: 0.9850 - val_loss: 0.0256 - val_accuracy: 0.9915
Epoch 3/12
166/166 [==============================] - 405s 2s/step - loss: 0.0143 - accuracy: 0.9943 - val_loss: 0.0194 - val_accuracy: 0.9915
Epoch 4/12
166/166 [==============================] - 400s 2s/step - loss: 0.0245 - accuracy: 0.9917 - val_loss: 0.0159 - val_accuracy: 0.9915
Epoch 5/12
166/166 [==============================] - 394s 2s/step - loss: 0.0298 - accuracy: 0.9907 - val_loss: 0.0400 - val_accuracy: 0.9864
Epoch 6/12
166/166 [==============================] - 397s 2s/step - loss: 0.0115 - accuracy: 0.9966 - val_loss: 0.0140 - val_accuracy: 0.9932
Epoch 7/12
166/166 [==============================] - 394s 2s/step - loss: 0.0082 - accuracy: 0.9981 - val_loss: 0.0935 - val_accuracy: 0.9864
```

Figure 5.2: Training with Local Machine (~400 s / epoch)

6 - References

Bhuvaji, S., Kadam, A., Bhumkar P., Dedge S., & Kanchan S. (2020). Brain Tumor Classification (MRI), Version 2. Retrieved December 15, 2022. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/1183165

Centers for Disease Control and Prevention. (2022, September 6). *FASTSTATS - leading causes of death*. Centers for Disease Control and Prevention. Retrieved December 15, 2022, from https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm

Chakrebarty, N., (2018). Brain MRI Images for Brain Tumor Detection, Version 1. Retrieved December 15, 2022 from https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection.

Cheng, J. (2017). brain tumor dataset (Version 5). figshare. https://doi.org/10.6084/m9.figshare.1512427.v5 ([])

Díaz-Pernas, F. J., Martínez-Zarzuela, M., Antón-Rodríguez, M., & González-Ortega, D. (2021). A deep learning approach for Brain Tumor Classification and segmentation using a multiscale convolutional neural network. *Healthcare*, *9*(2), 153. https://doi.org/10.3390/healthcare9020153

Gao, P., Shan, W., Guo, Y., Wang, Y., Sun, R., Cai, J., Li, H., Chan, W. S., Liu, P., Yi, L., Zhang, S., Li, W., Jiang, T., He, K., & Wu, Z. (2022). Development and validation of a deep learning model for brain tumor diagnosis and classification using Magnetic Resonance Imaging. *JAMA Network Open*, *5*(8). https://doi.org/10.1001/jamanetworkopen.2022.25608

Hamada A. (2020). Br35H :: Brain Tumor Detection 2020, Version 12. Retrieved December 15, 2022 from https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection?select=no.

Johns Hopkins Medicine. (n.d.). *Brain tumors and brain cancer*. Brain Tumors and Brain Cancer | Johns Hopkins Medicine. Retrieved December 15, 2022, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor

Kumar, J., (2021). Brain Tumor MRI Classification: TensorFlow CNN, Version 1. Retrieved December 15, 2022 from https://www.kaggle.com/code/jaykumar1607/brain-tumor-mri-classification-tensorflow-cnn.

Varoquaux, G., & Cheplygina, V. (2022, April 12). *Machine Learning for Medical Imaging: Methodological Failures and recommendations for the future*. Nature News. Retrieved December 15, 2022, from https://www.nature.com/articles/s41746-022-00592-y