HIGH SCHOOL EDITION
Journal of Student Research

Volume 12 Issue 2 (2023)

Brain Tumor Detection Using Convolutional Neutral Network

Falak Chhatre[1], Sudhanva Deshpande[2], Sidhant Malik[2], Grace Yan[1] and Suresh Subramaniam[#]

[1]Mission San Jose High School
[2]Monta Vista High School
[#]Advisor

ABSTRACT

Early and accurate diagnosis of brain tumors, a lethal disease caused by the abnormal growth of cells in the brain, is imperative to increase survival rates. A popular method for detection, diagnosis, and treatment is magnetic reasoning imaging (MRI) because it is non-invasive and provides high-quality visuals. Unfortunately, analyzing them manually can often be time-consuming and requires medical expertise. Image classification, a subset of computer vision, is a computer's ability to classify and interpret objects within images. It can support a doctor's diagnosis and serve as an entry-level screening system for brain tumors. This study aims to build an accurate machine learning model to predict the existence of brain tumors from magnetic resonance images. We used the Br35H dataset to build two different convolutional neural network (CNN) models: Keras Sequential Model (KSM) and Image Augmentation Model (IAM). First, images from our dataset were preprocessed, augmented, and standardized to improve efficiency and reduce inaccuracies. Then, the data was normalized, and our models were trained. Lastly, aside from the validation accuracy and loss observed while training, we cross-referenced the accuracy of our model using the accuracy validation dataset. Of our two models, the IAM outperformed the KSM. The IAM had a validation accuracy of 97.99% and a validation loss of 4.94% on the Br35H dataset, and a 100% accuracy when classifying MRIs from the accuracy validation dataset.

Introduction

Due to their low survival rate, brain tumors are a deadly disease. They can either be benign (non-cancerous) or malignant (cancerous). Benign brain tumors gradually grow and are usually contained in one area of the brain, while malignant brain tumors rapidly grow and invade healthy brain tissue. The 5-year survival rate for these malignant tumors is only 36%, although it can vary based on age and the location of the tumor.

It is more difficult to detect tumors in the brain compared to other body parts because ordinary radioactive indicators have difficulty detecting tumor cells due to the blood-brain barrier. For that reason, two widely used methods to detect brain tumors are magnetic resonance imaging (MRI) and computer tomography (CT). Moreover, Computer-aided diagnosis (CAD) and machine learning help radiologists detect tumors.

A faster, more efficient method to classify MRIs of the brain is imperative to the early detection and successful treatment of brain tumors. Automating or introducing computer-assisted screening rounds in medical imaging will contribute to combating the increasing healthcare worker shortage and will help medical professionals in developing countries. The effects of the recent pandemic have showcased the shortcomings of the healthcare system at times of crisis and emphasized the importance of assistive technologies at the starting stages of diagnosis.

In this study, we built a model to examine brain MRIs and accurately predict the existence of a tumor. Our model is trained on the Br35H:: Brain Tumor Detection 2020 (training) dataset and its performance is evaluated using Brain MRI Images for Brain Tumor Detection (accuracy validation) dataset. The model had a validation accuracy of 97.99% and a validation loss of 4.94% on the training dataset and had a 100% accuracy when classifying MRIs from the accuracy validation dataset.

Methods

Data Standardization

We built our models using images from a publicly available dataset: Br35H :: Brain Tumor Detection 2020. It consists of 2998 images, split evenly into two categories: 'yes' tumor and 'no' tumor. Below are three samples from each category of the original dataset, with no manipulation.

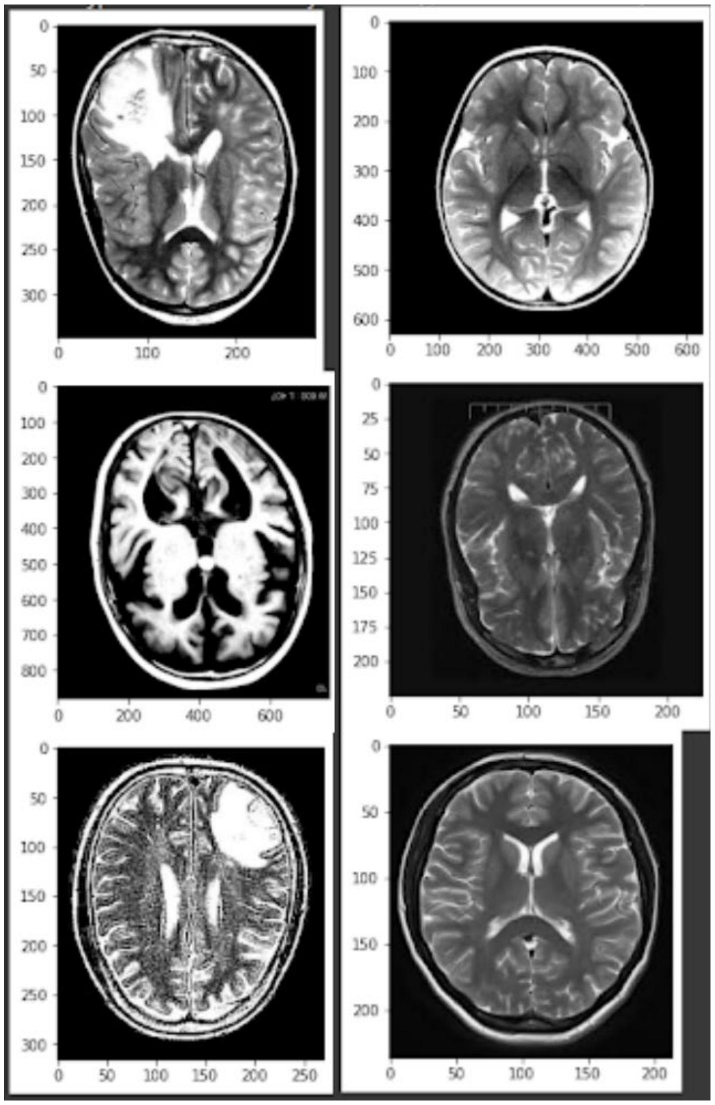

Figure 1. Raw images from the original dataset

As can be observed from Figure 1, our input images were not standardized and contained unnecessary pixels that could reduce the performance of our model. Therefore, we first pre-processed the data using the OpenCV library. There were two main manipulations of the original images. First, we converted the images to grayscale to reduce the input layers from three to one. Second, we scaled all of the images to 250x250 pixels. The pre-processed data was saved and organized appropriately. The images from Figure 1 were processed and are displayed in Figure 2.

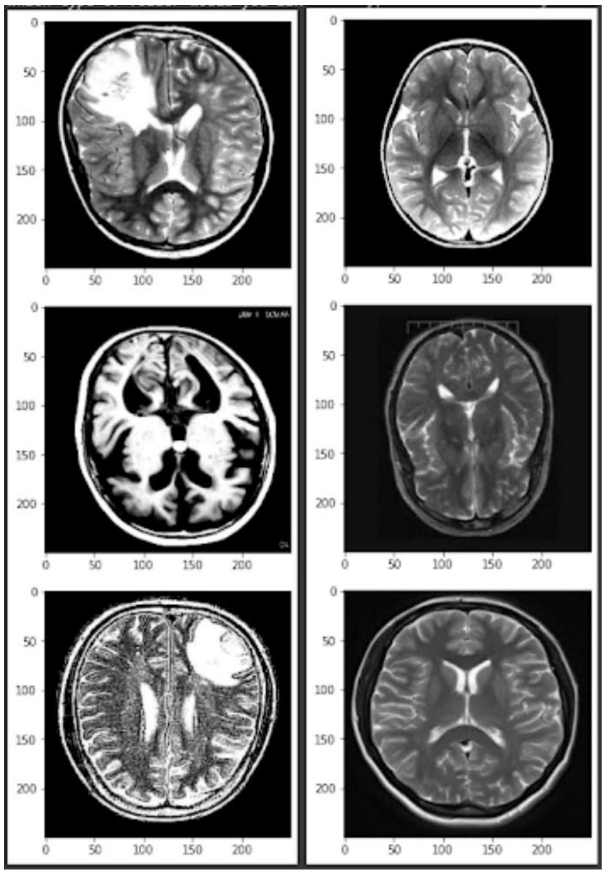

Figure 2. Images from the original dataset after standardization

The dataset was then divided into training, validation, and testing categories. The breakdown of these categories can be observed in Table 1.

Table 1. The distribution of images in training, testing, and validation stages

	training	testing	validation	total
Br35H	2399	299	299	2998

A data normalization layer is used to scale the pixel values from the greyscale MR images from 0 to 1. This improves the stability and consistency of the model and decreases the needed training time, as the model has a stronger learning ability due to the extra layer.

Keras Sequential Model

Our initial model was a basic Keras sequential model. It contains a linear stack of layers and is flexible because its variables (batch size, activation, padding) can be manipulated easily. The individual layers of this model can be observed in Figure 3. This model is basic and cannot provide high accuracy, even with the large amount of data provided.

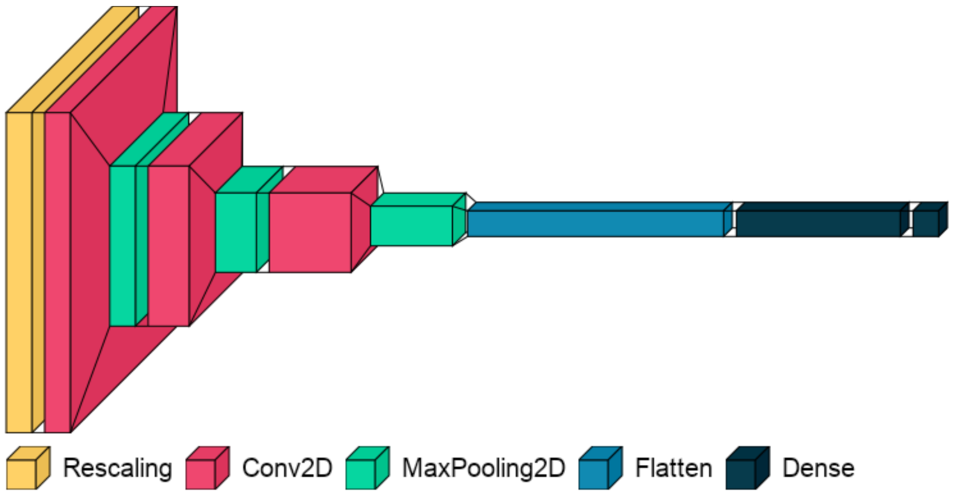

Figure 3. Architecture of the Keras Sequential Model

Image Augmentation Model

When we plotted the training and validation accuracy and loss, it was apparent that the model was overfitting. This issue, combined with our small dataset size, led us to try a different approach. We created a second sequential model using image augmentation to increase the input numbers and avoid overfitting. Specifically, we used three augmentation techniques: random horizontal flip, random rotation, and random zoom shown. Figure 4 is an example of the pre-processing layers of the model.

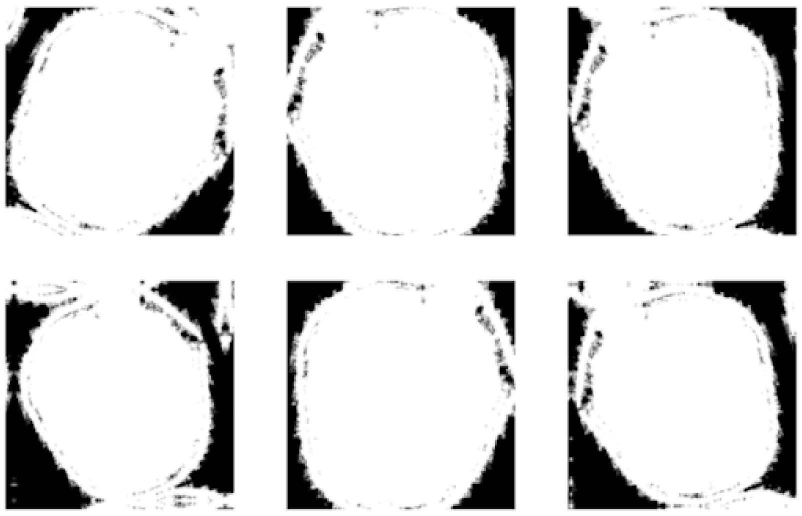

Figure 4. Images generated from the Image Augmentation Model

The elements of this CNN model are more complex than that of the KSM. The IAM consists of rescaling, convolutional, pooling, flattening, and dropout layers. Figure 5 displays the architecture of the CNN that we built in this study. Convolutional layers use kernels, matrices of weights learned through training, that convolute over the input to identify the important and relevant features of the images. Their outputs are called feature maps because they display the location of different features (ex. edges, straight lines, objects). The MaxPooling2D layers decrease the size of feature maps by extracting the maximum value of a (2,2) matrix. The Flatten layer is usually towards the end of a neural network because it takes a feature map and transforms it into a one-dimensional vector. This one-dimensional vector is then imputed into a Dense layer that classifies the image. A dense layer is fully-connected, meaning that each value in this layer is connected to every value in the previous layer. When this model was trained, there was a significant decrease in overfitting and the validation accuracy was higher.

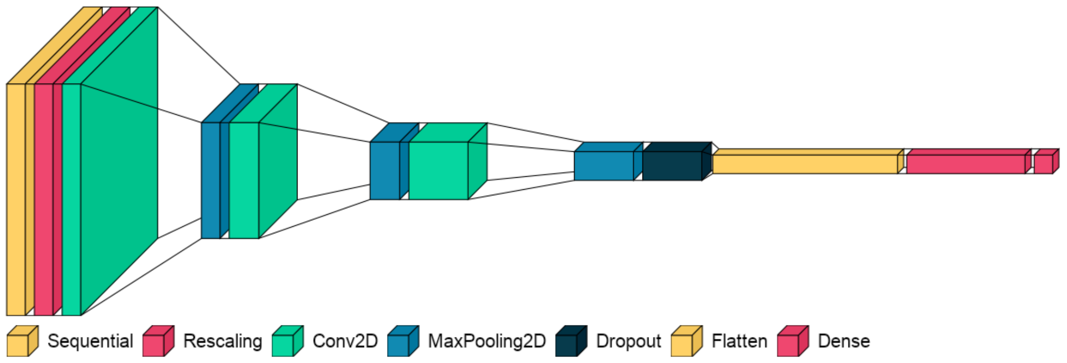

Figure 5. Architecture of the Image Augmentation Model

Finding the Optimal Number of Epochs and Batch Size

After implementing the Image Augmentation Model, the validation accuracy was adequately high, but to further enhance the model, we experimented with the number of epochs and batch sizes. Since we noticed that after a certain number of iterations, our model's validation accuracy peaked, our goal was to identify an optimal epoch and batch size to avoid unnecessarily iterating through our dataset.

In our code, we added an EarlyStopping callback that monitored when the validation loss stopped improving and concluded the training at that point. We ran this callback multiple times with different batch sizes to find the highest validation accuracy that showed the least signs of overfitting. A summary of our findings can be found in Table 2.

Table 2. Validation Accuracy and Loss for Different Batch Sizes and the Optimal Number of Epochs

Epochs When Training Concluded	Batch Size	Validation Accuracy (%)	Validation Loss (%)
28	256	94.65	18.19
36	128	97.99	4.94
13	64	89.32	24.91
20	32	92.98	16.73

The optimal batch size was 128, and the optimal number of epochs was 36.

Results

Table 3. The training accuracy, training loss, validation accuracy, and validation loss for each of the two models

	Training Accuracy (%)	Training Loss (%)	Validation Accuracy (%)	Validation Loss (%)
Keras Sequential Model	100.00	0.004055	99.33	15.77
Image Augmentation Model	96.87	8.75	97.99	4.94

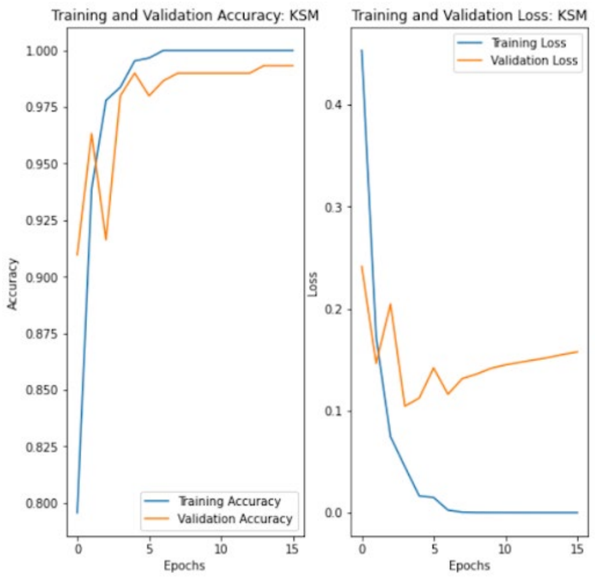

Figure 6. Plotted Training and Validation Accuracy: Keras Sequential Model

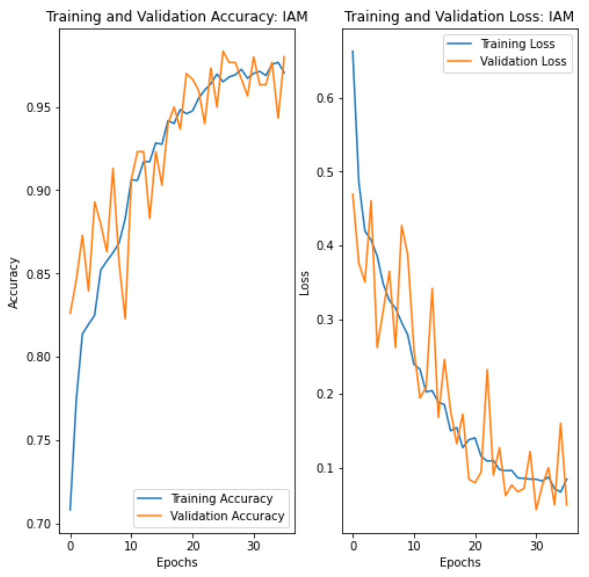

Figure 7. Plotted Training and Validation Accuracy: Image Augmentation Model

With such a high training accuracy and nearly negligible training loss (shown in Figure 6), we concluded that the Keras Sequential Model was overfitting. This meant that our model was extremely accurate against the input images, possibly including arbitrary traits and background noise, most likely due to the lack of data provided to our model.

The Image Augmentation Model had reasonable training and validation accuracies and closely resembled each other (shown in Figure 7). This meant that our model was not only accurate against the input data, but it was able to correctly classify images from the validation set.

To further validate these results, we decided to use our model to classify 100 random images in the Images Dataset for Brain Tumor Detection dataset. We referred to this as the accuracy validation dataset. The images were manually renamed to be consistent. This includes standardization such as renaming the files and fixing image numbering. Next, each image was inputted into our model which predicted if the brain MRI had a tumor and calculated a confidence percent. The results of this validation are summarized in Figure 8. Note that the minimum for the y-axis is 75.

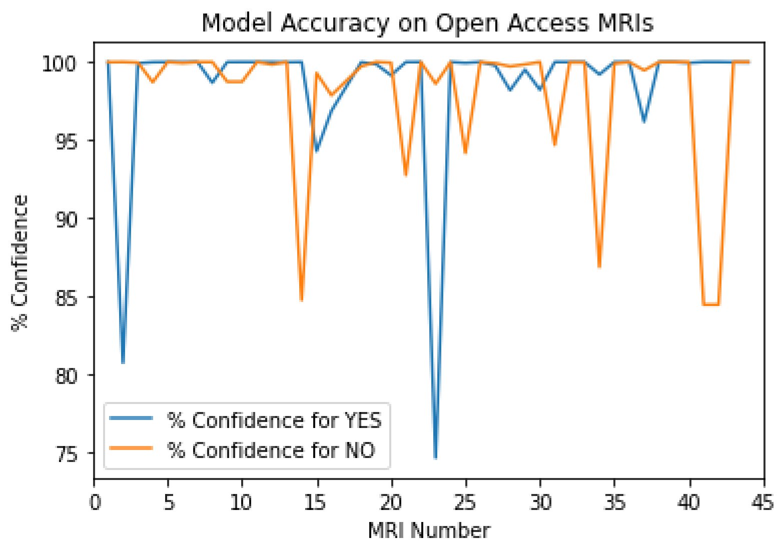

Figure 8. Model Confidence for accuracy validation dataset

Although 100 images are a small cross-validation number, our model was able to accurately predict 100% of the images with relatively high confidence.

Conclusion

Through these methodologies, we were able to build an accurate CNN to assist in the early detection and classification of brain tumors. During our research, we came across many problems, some solvable and others not. Solvable issues included finding a usable dataset in terms of content and size, troubleshooting the initial Keras Sequential model with high overfitting, and finding a method to further validate our results.

Our research supports the idea that Machine Learning Models can exponentially improve classification times for brain MRI. In our testing, our model needed an average of 16 seconds to classify an image. This is considerably faster than it would take a medical professional to perform the same task and more accessible as it does not require human involvement in the earlier stages.

In the future, there are many ways to improve the accuracy and versatility of this model. For example, we could employ transfer learning and tweak expert models to create a model that would apply to a wider set of MRI orientations and perspectives. Moreover, we could develop the complexity of our research by using a larger dataset consisting of different angles and varied brain segments. To further analyze our results, we could create confusion matrices and calculate numbers on our False Positive and False Negative rates.

Acknowledgments

We would like to thank Mr. Suresh Subramaniam, our advisor, for guiding and mentoring us throughout this project. We would also like to thank the Aspiring Scholars Directed Research Program (ASDRP) for giving us the opportunity and resources to conduct this research.

Frequently Used Terms

Epoch: The number of times the algorithm trains the neural network with all the data. One cycle of training represents one epoch.

Batch Size: The amount of data passed through the neural network at a time.

Overfitting: The model accurately classifies training images because it has adapted to details and noise from the dataset rather than drawing generalized conclusions. It is unable to perform well with new data.

Underfitting: The opposite of overfitting, where the model is unable to find the between input and output values and performs poorly on training data.

Training Loss/Accuracy: The accuracy of the model's predictions from the training dataset, which is the used to train the model.

Validation Loss/Accuracy: The accuracy of the model's predictions from the validation dataset, which is separated from the training data and used in fine tuning the model's hyperparameters.

References

[1] Asif, S. (2022, February 17). *Improving Effectiveness of Different Deep Transfer Learning-Based Models for Detecting Brain Tumors From MR Images*. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9718269

[2] *Brain Tumors and Brain Cancer*. (n.d.). Johns Hopkins Medicine. https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor

[3] *Brain Tumor: Statistics*. (2022, February). Cancer.Net. https://www.cancer.net/cancer-types/brain-tumor/statistics

[4] Chakrabarty, N. (2019, April 14). *Brain MRI Images for Brain Tumor Detection*. Kaggle. https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

[5] Géron, A. (2017). *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

[6] Hamada, A. (2021, November 14). *Br35H :: Brain Tumor Detection 2020*. Kaggle. https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection

[7] Naseer, A., Yasir, T., Azhar, A., Shakeel, T., & Zafar, K. (2021, June 13). Computer-Aided Brain Tumor Diagnosis: Performance Evaluation of Deep Learner CNN Using Augmented Brain MRI. *International journal of biomedical imaging*. 10.1155/2021/5513500

[8] Suresh, R. (2022, January 10). *In silico modeling of emodin's interactions with serine/threonine kinases and chitosan derivatives*. Journal of Emerging Investigators. https://emerginginvestigators.org/articles/i-in-silico-i-modeling-of-emodin-s-interactions-with-serine-threonine-kinases-and-chitosan-derivatives