Report: Deep Learning for Brain Cancer Detection

Samuele Aglieco

21/08/2024

1 Introduction

Detecting brain cancer early is crucial for giving patients the best chance at recovery. In this project, I worked on developing a deep learning model that can analyze MRI images and identify whether they contain signs of a tumor. This report will walk you through my journey, highlighting the key steps and decisions that led me to the final, most effective model.

2 Sample MRI Images

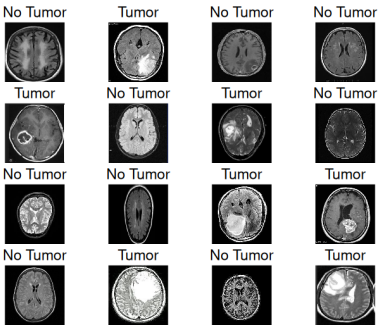

Figure 1: Sample MRI images from the dataset showing various cases. The grid includes images labeled as "Tumor" and "No Tumor".

These images give a visual understanding of the types of data my models analyzed. It's essential to see the kind of patterns the models needed to learn to differentiate between healthy and tumor-affected brain tissues.

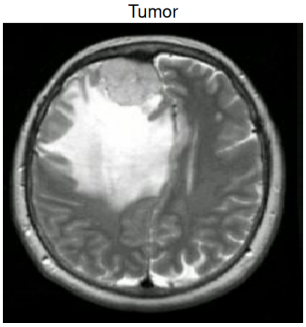

Figure 2: Example of an MRI image with a visible tumor.

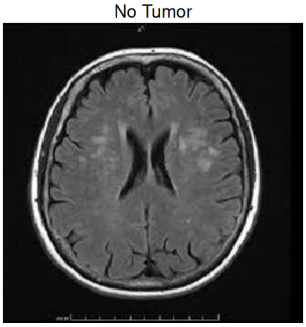

Figure 3: Example of an MRI image without a tumor.

3 Dataset and Preparation

I started with a dataset of MRI images categorized into "Yes" (with a tumor) and "No" (without a tumor). The dataset was divided into three parts: 70% for training, 15% for validation, and 15% for testing. To make the model more robust, I applied various data augmentation techniques like flipping, rotating, and zooming in on the images. However, as noted in Model 3, excessive rotation or zoom of the images resulted in worse model performance in terms of accuracy and loss.

All images were resized to 32x32 pixels, and pixel values were normalized to a range of 0 to 1. This preprocessing helped speed up the training process and made the model more effective in learning the relevant features from the images.

The dataset used in this project is sourced from Kaggle and can be found at the following link: `https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection/data`.

4 Modeling Journey

I explored several models to find the one that works best. My journey started with a simple CNN and gradually moved towards more complex architectures. I'll briefly touch on each model before diving into the details of my final, most successful model—Model 7.

- **Model 1**: A simple CNN with three convolutional layers. It gave me a decent start but needed improvement.
- **Model 2**: I tried a pre-trained VGG16 model. It improved accuracy, but I noticed some overfitting.
- **Model 3 and 4**: I experimented with different levels of data augmentation. While some augmentation helped, too much actually hurt the model's performance.
- **Model 5**: A more complex CNN with four convolutional layers. This model showed good performance but was further improved by reducing the batch size from 32 to 16, leading to the development of the final model, Model 7.
- **Model 6**: I retrained Model 5 with a separate validation set, which improved its generalization but wasn't perfect yet.

4.1 Focus on Model 7

After refining my approach with the previous models, I arrived at Model 7, which delivered the best results.

4.1.1 Architecture

- Four convolutional layers with increasing filter sizes: 32, 64, 128, and 256.
- Max-pooling and dropout layers after each convolutional layer to prevent overfitting.
- A dense layer with 256 neurons, followed by a dropout layer.
- Finally, a single neuron in the output layer with a sigmoid activation function, perfect for our binary classification task.

4.1.2 Optimization Strategy

- I used the Adam optimizer, which is well-suited for this kind of deep learning task.
- The loss function was binary crossentropy, the standard choice for binary classification problems.

- I trained the model for 35 epochs and adjusted the batch size to 16. This combination allowed the model to learn effectively without much overfitting to the training data.

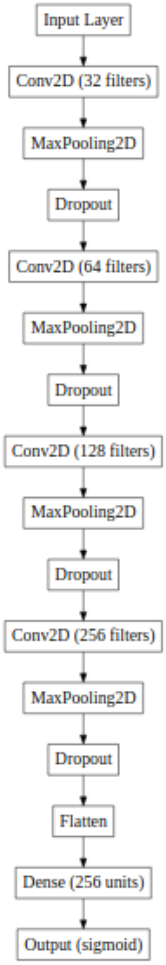

Figure 4: Model Diagram

4.2 Results

Model 7 showed the best performance across all models. The final evaluation metrics for Model 7 are as follows:

- **Loss**: 0.122
- **Accuracy**: 96.66%

This model demonstrated high accuracy and low loss, indicating that it performed exceptionally well in distinguishing between tumor and non-tumor images.

4.3 Visualizing Performance

To better understand the training process, I plotted the loss and accuracy for both the training and validation sets over the 35 epochs. The following plot, generated from Keras, shows both metrics on a single graph.

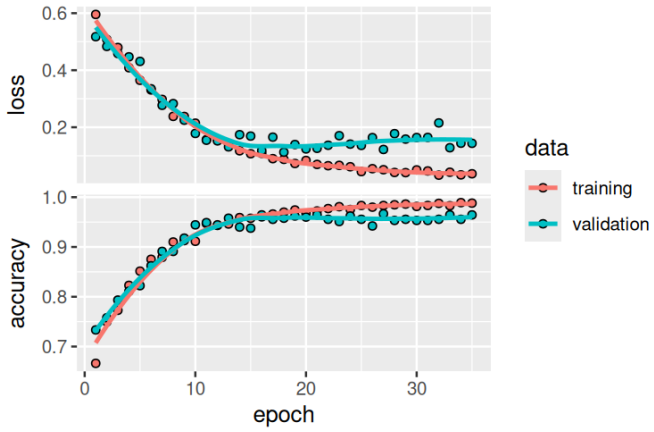

Figure 5: Training and Validation Loss and Accuracy over 35 Epochs. The plot provides insights into how the model's loss and accuracy evolved during training. Ideally, both training and validation loss should decrease, and accuracy should increase and stabilize. Any divergence, such as increasing validation loss while training loss decreases, may indicate overfitting.

These plots will help you see how Model 7 performed during training and if there were any signs of overfitting. The model showed slight signs of overfitting in the end. The best accuracy was achieved at epoch 27, with a value of 96.66

5 Evaluation

After training, I evaluated Model 7 on the test set. The results were good, with a final loss of 0.122 and an accuracy of 96.66%. These metrics show that Model 7 performs effectively in detecting brain tumors from MRI images.

The low loss value indicates that the model's predictions are close to the actual labels, while the high accuracy demonstrates that the model correctly classified most of the test images. These results confirm that Model 7 is a reliable tool for this task.

6 Conclusion

Through my work, I developed a CNN model that effectively detects brain tumors from MRI images. Model 7, with its refined design and tuned parameters, performed the best among the tested models. It's ready for further testing and could be useful in real-world applications.

In the future, I might look into using larger datasets or trying out different model architectures to improve even more.

7 Dataset

The dataset used for this project is available on Kaggle and can be accessed via the following link: `https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection/data`. This dataset includes labeled MRI images of brain tumors.