Deep learning for image and video processing

Brain Tumor Detection Project Report

University of Maastricht

Daridis Theologos Christos id: 6330897

December 2022

Abstract

A Brain Cancer is very critical disease which causes deaths of many individuals. The brain tumor detection and classification system is available so that it can be diagnosed at early stages. Cancer classification is the most challenging tasks in clinical diagnosis.The main objective of this project is to predict if the subject has tumor or not, using Deep learning architectures. generally known as NN (Neural Networks) and two transfer learning methods as [4]VGG 19(visual geometric groups) and [2] Inception-V3.Moreover we analyze the structure of each model and compare the experiment based on their accuracy.

1 Motivation

Nowadays Brain Tumor [1] is one of the most dangerous diseases, requiring early and precise detection methods. Currently, most detection and diagnosis methods rely on the judgment of neurospecialists and radiologists for image evaluation, which can lead to human error and be time-consuming.As it mentioned above detection plays a crucial role in the treatment of brain tumors, as early and accurate detection allows for the implementation of appropriate treatment strategies, which can significantly improve the prognosis and quality of life for individuals with this condition. Various image processing techniques are employed in this application, including techniques to feature extraction, and classification, which analyze medical images such as MRI(Magnetic Resonance Imaging).

2 Data set

There are several datasets available for brain tumor detection, which consist of brain MRI images. The first dataset [1]selected contained 98 MRI images without a tumor and 155 images with a tumor. It was believed that the training

[1]https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

process would not be time-consuming due to the size of the dataset, allowing for a large number of experiments to be conducted. In fact, a significant number of experiments were carried out, which greatly enhanced my understanding of deep learning architectures and the modifications necessary to improve the accuracy of my results.However, i encountered several issues during the evaluation process, as will be discussed in the following sections.Therefore, the next step was to conduct the same experiments using a larger [2]dataset consisting of 1500 MRI images without a tumor and 1500 images with a tumor.

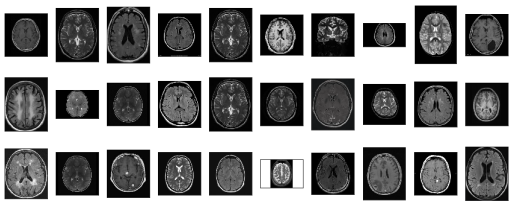

Figure 1: No Tumor

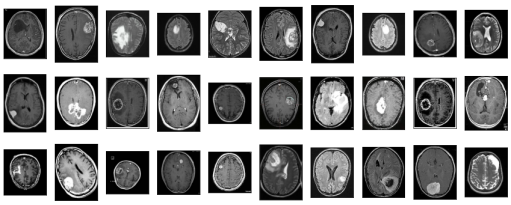

Figure 2: Caption

3 Methodology

In this section,we illustrate the following steps of our general workflow(input data, data preprocessing,data augmentation,model training and image classification) and we propose three methodologies for the detection of brain tumors using convolutional neural networks[3] (CNNs). The first method involves the use of a 10-layer CNN. The second and third methods utilize transfer learning, employing the VGG19 and INCEPTION-v3 models, respectively.It is important to note that the effectiveness of these methods will depend on the quality and diversity of the data used, as well as the chosen hyperparameters and optimization algorithm.

[2]https://www.kaggle.com/datasets/abhranta/brain-tumor-detection-mri

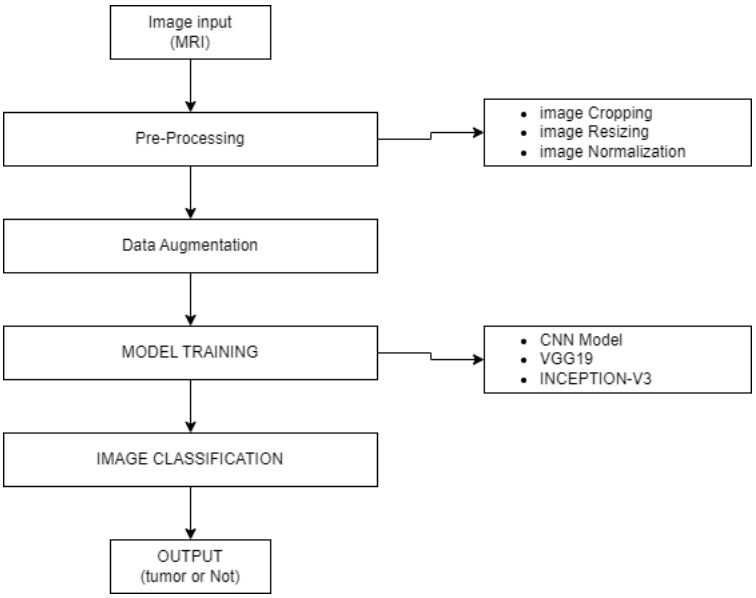

Figure 3: Workflow

3.1 Image Preprocessing

As you can see in figure 4, according to the distribution of image ratios, there are many images with different widths and heights, different sizes of black corners, and some of them have numbers or letters that add noise to our dataset. Therefore, the first step of "normalization" was to crop the brain out of the images. We used a technique with OPEN-CV that is perfectly described on the pyimagesearch blog for finding the extreme points on the image and cropping the rectangular region around them.

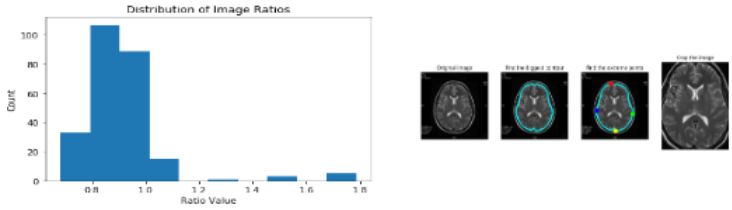

Figure 4:

3.2 Data Augmentation

Since we had small data set and in order to prevent overfiting we use real time data augmentation to increase our training examples according to kerasblog.We carry out some random rotations and brightness changes.

4 Experiments

In this section, we will focus on the architecture of our models and carefully analyze the processes that occur within the model's layers. We will also present the results of training each method using two datasets: one containing 253 MRI images and the other containing 3000 MRI images.Additionally We illustrate the results of these training sessions for 30 epochs to better understand how each method performs and how the size of the dataset affects the result.

4.1 Working of CNN 10-layers

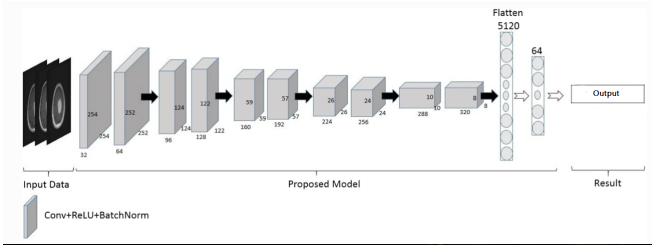

Figure 5: CNN architecture

- layers of CNN model
  - Convolution 2D
  - Max pooling 2D
  - Dropout
  - Flatten
  - Dense
  - Activation

- Convolution 2D
  Convolution 2D layer is used to extract features from images and other types of data. For example, a CNN might use a 2D convolution to detect edges in an image, or to identify patterns of pixels that are indicative of a particular object or feature

- ❖ Max pooling 2D
  Max pooling 2D layer reduce the spatial dimensions of the input data while also retaining the most important features. It is typically applied after a convolutional layer to reduce the number of parameters and computational cost, as well as to control overfitting.

- ❖ Dropout
  Dropout is randomly sets input units to zero with a frequency of rate at each step during training time.

- ❖ Flatten
  Flatten layer takes the input from the previous layer which is typically a multi-dimensional array and flattens it into a one dimensional array.

- ❖ Dense
  A dense layer takes as input a one-dimensional array and produces an output array of the same size, where each output unit is a weighted sum of the input units plus an optional bias term

- ❖ Activation
  A Sigmoid activation function used in our case since we binary classification so the output must be 0 or 1.

Results

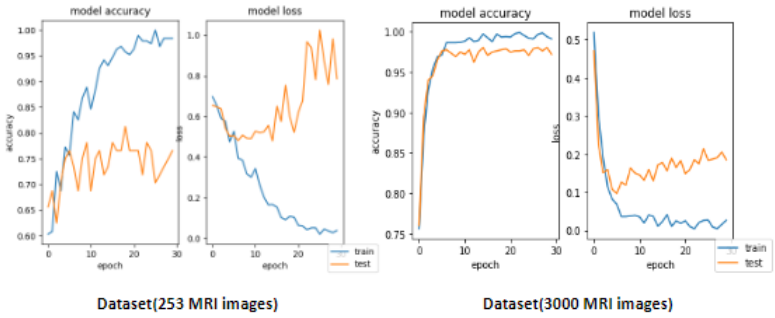

Figure 6: CNN method

4.2 Transfer learning Methods

Transfer learning is a way to speed up the process of building a deep learning model by using the knowledge from a pre-trained model. It can save time, reduce the amount of training data needed, and lower computational costs. Transfer learning has been used in a variety of fields, such as identifying tumors, predicting software defects, recognizing activities, and classifying sentiments. In this case, the performance of a Deep CNN model we compare to a popular transfer learning approach [5] VGG19 and INCEPTION-V3

4.2.1 Working of VGG19 model

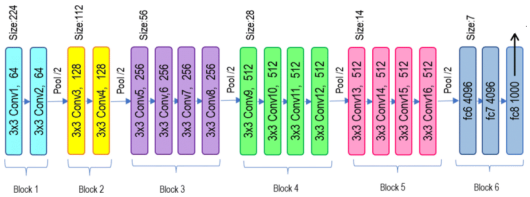

Figure 7: VGG 19 Architecture

VGG19 is a convolutional neural network (CNN) model with 19 layers, including 13 convolutional layers and 3 fully connected layers. Here is a brief description of the process of input data passing through the layers of VGG19:The input data is passed through the first convolutional layer, which applies a set of filters (also known as kernels or weights) to the input data to compute a set of feature maps.The output of the first convolutional layer is passed through additional convolutional layers.The output of the convolutional layers is passed through a series of pooling layers, which retaining the most important feature.After the convolutional and pooling layers, the output is passed through a flatten layer, which converts the multi-dimensional output of the previous layers into a one-dimensional array as we mentioned above.The output of the flatten layer is passed through the first fully connected layer, which applies a matrix multiplication followed by an optional bias offset and an activation function to produce a prediction.The output of the first fully connected layer is passed through additional fully connected layers and using the activation functions produce a final prediction.

Results

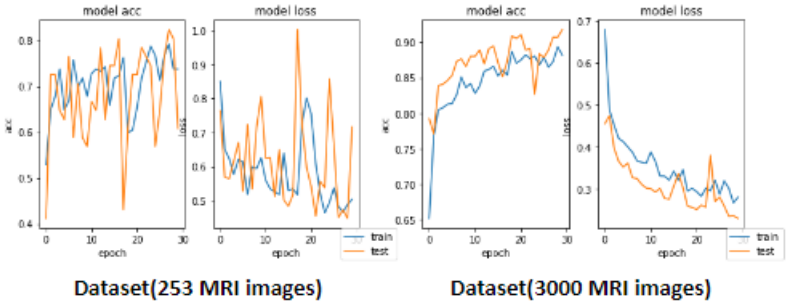

Figure 8: VGG 19 model

4.2.2 Working of Inception V3

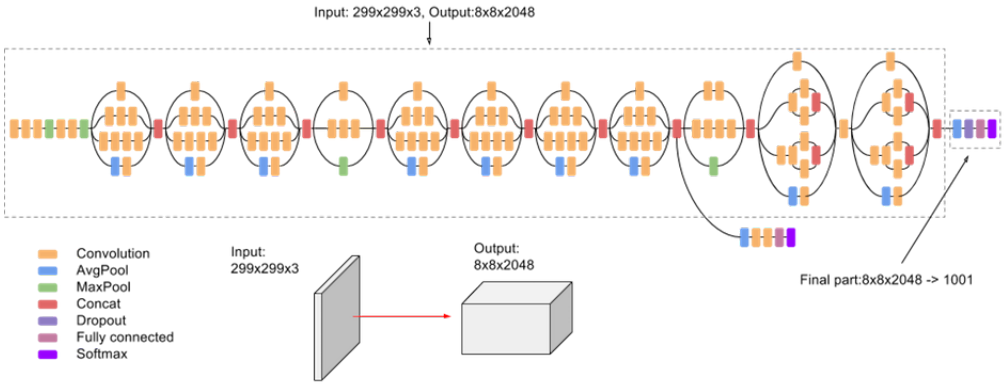

Figure 9: Inception V3 architecture

Like VGG19, Inception V3 is a pretrained model that has been trained on the ImageNet dataset. We have chosen it because it is known for its good performance.One of the main differences between Inception V3 and VGG19 is their architecture. VGG19 has a simpler and shallower architecture compared to Inception V3, which makes it faster to train and evaluate. However, this simplicity may also make it less effective at recognizing patterns in images compared to Inception V3.Another difference is the type of layers used in the two models. VGG19 primarily uses convolutional and fully connected layers, while Inception V3 uses a combination of these layers as well as inception modules, which are layers that include multiple parallel convolutional and pooling layers. This allows Inception V3 to learn a variety of features at different scales, which may improve its performance on a wider range of tasks.Overall,seems that Inception V3 are more powerful and widely used for image recognition tasks.

Results

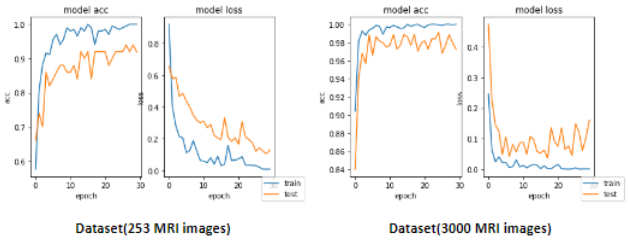

Figure 10: Inception V3

5 Comparison

Dataset (253 MRI images)

Model	Train loss	Val loss	Accuracy
CNN-(10 layers)	0.8052	0.0973	0.9654
VGG-19	0.5143	0.7209	0.7357
Inception-V3	0.0261	0.1804	0.9923

Dataset (3000 MRI images)

Model	Train loss	Val loss	Accuracy
CNN-(10 layers)	0.0368	0.1784	0.9754
VGG-19	0.2417	0.0489	0.8757
Inception-V3	0.0078	0.1309	0.9908

Figure 11: Model Comparison

6 Conclusion

In this project, we attempted to build a robust CNN model and find the most accurate transfer learning model using VGG19 and Inception-V3. Initially, we used a small dataset, but all of our models suffered from overfitting. Therefore, we made several scaling changes to the batch size and ultimately determined that a batch size of 32 would not converge in the first epochs, but we achieved better results for 30 epochs. As an optimizer, we selected the Adam optimizer because it can reduce noise in the training process. However, it appears that all of our models continued to overfit. This may have been due to the small size of the validation set in comparison to the training set, causing the models to fit better to the validation set than the training set.

Next, we used a larger dataset and implemented the same experiments with

the same epochs and hyperparameters. However, we encountered several issues, particularly with the Inception-V3 model, which we had expected to be the most accurate. After conducting extensive research, we discovered that the Imagenet dataset may not include a sufficient number of brain tumor MRI images and may not be able to quickly recognize the subject. Therefore, we did fine-tuning on Inception-V3 model,use RMS optimizer since he update the learning rate automatically and choose different learning rate for each parameter.Then we retrained the model, which has almost 20 million parameters. Finally, we obtained the results we had hoped for, with more than 99 per cent.We planned to do the same with the VGG19 model, but encountered difficulties with Google Colab and were unable to train the model. However, not every task is said to be perfect in this development field even more improvement may be possible in this application. I have learned so many things and gained a lot of knowledge about development field.

7 Tools

As a beginner, I have never done anything similar to this before in this field. To implement my brain tumor detection project in Python, I have been following many tutorials and blogs which i will mention below.

- Python was the selection language of this project
- Python packages like NumPy, Pandas and SciPy area unit freely available and well documented made the project simple
- Open cv used for image preprocessing
- Neural network libraries such as Tensorflow and Keras assist with the implementation of my model. There are many resources available for each model, including detailed information about the layers. Tensorflow is a highly flexible and efficient platform for building and training machine learning models, including neural networks. Keras, on the other hand, is a high-level API built on top of Tensorflow that simplifies the process of building and training deep learning models
- Youtube medical image classification videos

References

[1] Hussna Elnoor Mohammed Abdalla and M. Y. Esmail. Brain tumor detection by using artificial neural network. In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pages 1–6, 2018.

[2] Hassan Ali Khan, Wu Jue, Muhammad Mushtaq, and Muhammad Umer Mushtaq. Brain tumor classification in mri image using convolutional neural network. *Math. Biosci. Eng*, 17(5):6203–6216, 2020.

[3] Amjad Rehman, Muhammad Attique Khan, Tanzila Saba, Zahid Mehmood, Usman Tariq, and Noor Ayesha. Microscopic brain tumor detection and classification using 3d cnn and feature selection architecture. *Microscopy Research and Technique*, 84(1):133–149, 2021.

[4] N. Siddaiah, C. Vignesh, T. Narendra, K. Pujitha, and D. Madhu. Brain tumor detection from mri scans using vgg- 19 networks. In *2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22)*, pages 1–6, 2022.

[5] Zar Nawab Khan Swati, Qinghua Zhao, Muhammad Kabir, Farman Ali, Zakir Ali, Saeed Ahmed, and Jianfeng Lu. Content-based brain tumor retrieval for mr images using transfer learning. *IEEE Access*, 7:17809–17822, 2019.