Brain Tumor MRI Classification

Megi Xhafka
megi.xhafka@georgebrown.ca
101445718

Michelle Ohikhena
michelle.ohikhena@georgebrown.ca
101428168

Tasneem Pipalyawala
tasneem.pipalyawala@georgebrown.ca
101439685

Taha Sathalia
taha.sathalia@georgebrown.ca
101444968

August 06, 2023

Abstract

Brain tumor classification using magnetic resonance imaging (MRI) is a critical task in the field of medical image analysis. Accurate and efficient classification of brain tumors is essential for early diagnosis and timely treatment planning.
In this project, we leverage the power of deep learning and state-of-the-art convolutional neural network (CNN) architectures, including EfficientNetB0, MobileNet, Xception, and ResNet50, to address the crucial task of brain tumor classification. The primary goal of this project is to develop a robust classification model capable of accurately categorizing brain MRI images into four distinct classes: glioma, meningioma, no tumor, and pituitary. After multiple attempts of fine-tuning, it can be identified that ***EfficientNetB0 with the highest accuracy of 97.70%*** *on the test set is the recommended model for brain tumor classification from MRI images due to its superior performance in terms of accuracy and other evaluation metrics.*

Figure 1. AI generated brain tumor illustration

1. Introduction

Brain tumor classification using magnetic resonance imaging (MRI) is a critical task in the field of medical image analysis. Accurate and efficient classification of brain tumors is essential for early diagnosis and timely treatment planning. In recent years, deep learning techniques have shown remarkable success in various image classification tasks, motivating their application to brain tumor MRI analysis. This project aims to develop a deep learning-based approach for brain tumor classification using MRI images. The proposed model leverages convolutional neural networks (CNNs) to automatically learn discriminative features from MRI scans.

2. Dataset

This dataset is a comprehensive collection of 7023 human brain MRI images, carefully curated from three primary sources: figshare, SARTAJ dataset, and Br35H. The main objective of this dataset is to classify brain MRI images into four distinct categories: glioma, meningioma, no tumor, and pituitary.
The distribution of the images among the classes is as follows:
Glioma: Images depicting brain MRI scans showing glioma tumors.
Meningioma: Images depicting brain MRI scans showing meningioma tumors.
No Tumor: Images of healthy brain MRI scans with no detectable tumors. These images were extracted from the Br35H dataset.
Pituitary: Images depicting brain MRI scans showing pituitary tumors.

2.1 Data Preprocessing

The dataset contains images of varying sizes, which can potentially introduce challenges during model training. To address this, the dataset should undergo preprocessing steps, including resizing the images to a standardized size. This resizing process will involve eliminating any extra margins that may be present in the images. By doing so, the accuracy and effectiveness of the model's preprocessing code will be significantly enhanced. In figure 2 you can see the data distribution of 4 classes and it is not a balanced dataset. Data augmentation is applied using the ImageDataGenerator from Keras to increase the dataset size and enhance model generalization. Augmentation techniques include rotation, horizontal flip, vertical flip, zoom, and shift. The dataset is split into a training and testing set. The images are resized to 150x150 pixels.

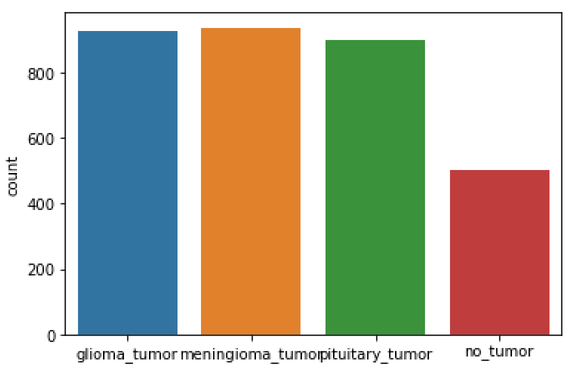

Figure 2. Dataset distribution

3. Methods

EfficientNetB0, MobileNet, Xception, and ResNet are popular convolutional neural network (CNN) architectures used for various computer vision tasks, including image classification. When applied to brain tumor MRI classification, these models can help accurately identify and classify brain tumors from MRI images. Here's a brief overview of each architecture.

3.1. EfficientNetB0

EfficientNetB0 is one of the variants from the EfficientNet family of models, which are designed to achieve state-of-the-art performance with high accuracy and efficiency. It is the smallest model in the EfficientNet series and is particularly useful when computational resources are limited, making it suitable for brain tumor MRI classification tasks. EfficientNetB0 is built upon the compound scaling method, which balances model depth, width, and resolution. It uses a combination of depthwise and pointwise convolutions to extract features from the input MRI images. The architecture is designed to achieve high accuracy while keeping the model relatively small in terms of parameters and computations.

3.2. MobileNet

MobileNet is another lightweight CNN architecture designed for mobile and embedded vision applications. It uses depthwise separable convolutions to significantly reduce the number of parameters and computations while maintaining decent accuracy. MobileNet is suitable for resource-limited scenarios and can be employed for brain tumor MRI classification tasks when computational resources are constrained. The training process involves passing the brain MRI images through the network, calculating the loss, and updating the weights using backpropagation. The process continues until the model converges and achieves satisfactory results.

3.3. Xception

Xception (Extreme Inception) is a CNN architecture that is based on the Inception model. It goes a step further by using depth wise separable convolutions similar to MobileNet. The depthwise separable convolutions aim to capture spatial information more effectively while reducing the computational complexity.

3.4. ResNet50 (Residual Network)

ResNet is a groundbreaking CNN architecture that introduced the concept of residual learning. It addresses the vanishing gradient problem by utilizing skip connections or shortcuts, which allow the model to learn the residual (difference) between the input and the output. This enables the training of very deep neural networks without degradation in performance. The modified ResNet50 model is trained on the brain tumor MRI dataset using the training set. The training process involves passing the MRI images through the network, computing the loss, and adjusting the weights of the last layer through backpropagation. Training continues until the model converges, capturing tumor-specific features from the MRI scans.

4. Experiments

We have conducted 2 experiments throughout our project where we have changed and played around with the fine-tuning of parameters.

4.1. Experiment 1

First experiment includes the Brain Tumor Classification from MRI using Four Models (EfficientNet, Xception, MobileNet, ResNet).

4.1.1. EfficientNetB0 Model

The EfficientNetB0 model is imported with pre-trained weights from 'imagenet' and initialized without the top layers. The top layers are added, including a Global Average Pooling 2D layer, a Dropout layer, and a Dense layer with softmax activation for classification. The model is compiled with the categorical cross-entropy loss function, Adam optimizer, and accuracy metric. Callbacks used during training include Tensor Board, Model Checkpoint, and Reduce LROnPlateau.

4. 1. 1. 1. EfficientNetB0 Evaluation

The model is trained for 10 epochs. The accuracy and loss metrics are plotted for training and validation data as seen on figure 3. The model achieves a testing accuracy of approximately 98.16%. The classification report shows high precision, recall, and f1-score for each class, indicating good performance. The confusion matrix visualizes the model's performance on each class.

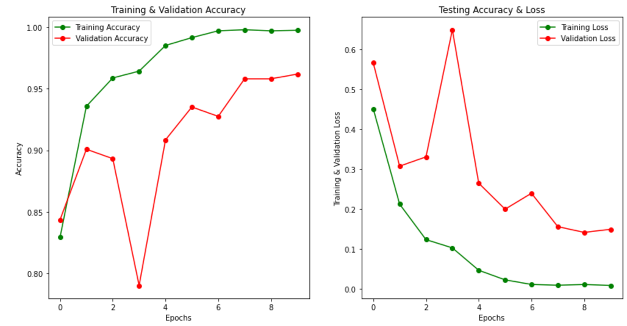

Figure 3. EfficentNetB0 Training vs Testing Plot

4. 1. 2. Xception Model

The Xception model is imported with pre-trained weights from 'imagenet' and initialized without the top layers. The top layers are added, including a Global Average Pooling 2D layer, a Dropout layer, and a Dense layer with softmax activation for classification. The model is compiled with the categorical cross-entropy loss function, Adam optimizer, and accuracy metric. Callbacks used during training include Tensor Board, Model Checkpoint, and ReduceLROnPlateau.

4. 1. 2. 1. Xception Evaluation

The model is trained for 12 epochs. The accuracy and loss metrics are plotted for training and validation data as seen on figure 4. The model achieves a testing accuracy of approximately 95.25%. The classification report shows high precision, recall, and f1-score for each class, indicating good performance. The confusion matrix visualizes the model's performance on each class.

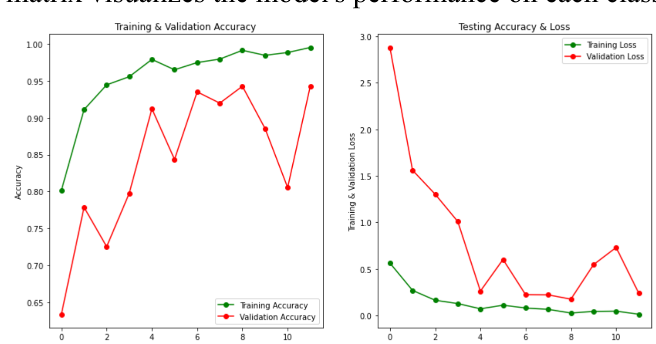

Figure 4. Xception Training vs Testing Plot

4. 1. 3. MobileNet Model

The MobileNet model is imported with pre-trained weights from 'imagenet' and initialized without the top layers. The top layers are added, including a Global Average Pooling 2D layer, a Dropout layer, and a Dense layer with softmax activation for classification. The model is compiled with the categorical cross-entropy loss function, Adam optimizer, and accuracy metric. Callbacks used during training include Tensor Board, Model Checkpoint, and ReduceLROnPlateau.

4. 1. 3. 1. MobileNet Evaluation

The model is trained for 12 epochs. The accuracy and loss metrics are plotted for training and validation data as seen on figure 5. The model achieves a testing accuracy of approximately 96.02%. The classification report shows high precision, recall, and f1-score for each class, indicating good performance.The confusion matrix visualizes the model's performance on each class.

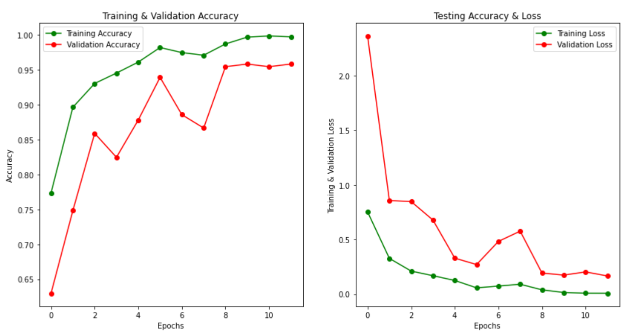

Figure 5. MobileNet Training vs Testing Plot

4. 1. 4. ResNet50 Model

The ResNet50 model is imported with pre-trained weights from 'imagenet' and initialized without the top layers. The top layers are added, including a Global Average Pooling 2D layer, a Dropout layer, and a Dense layer with softmax activation for classification. The model is compiled with the categorical cross-entropy loss function, Adam optimizer, and accuracy metric. Callbacks used during training include Tensor Board, Model Checkpoint, and ReduceLROnPlateau.

4. 1. 4. 1. ResNet50 Model Evaluation

The model is trained for an unspecified number of epochs. The accuracy and loss metrics are plotted for training and validation data as seen on figure 6. The model achieves a testing accuracy that is not mentioned in the provided information. The classification report shows high precision, recall, and f1-score for each class, indicating good performance. The confusion matrix visualizes the model's performance on each class.

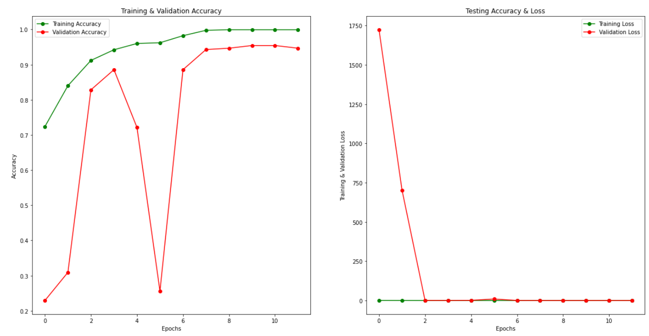

Figure 6. ResNet50 Training vs Testing Plot

4.2. Experiment 2

In this experiment, we trained four different deep learning models (EfficientNet, Xception, MobileNet, and ResNet) for brain tumor classification from MRI images. The goal was to determine which model performs best in terms of accuracy and other evaluation metrics. Each model was initialized with pre-trained weights (Imagenet) and added top layers for classification. The models were compiled with the categorical cross-entropy loss function and the Adam optimizer. EarlyStopping, Model Checkpoint, ReduceLROnPlateau, and TensorBoard were used as callbacks to monitor and control training. Each model was trained on the training set and evaluated on the testing set. The evaluation metrics included accuracy, precision, recall, F1-score, and the confusion matrix.

4. 2. 1 EfficientNetB0 Model

EfficientNetB0 is a deep convolutional neural network with a high efficiency and accuracy trade-off. It achieved the following performance:

- Training Accuracy: 99.99%
- Validation Accuracy: 96.95%

- Testing Accuracy: 97.70%
- Precision, Recall, and F1-score were all above 95% for all classes.

Diagonal values of the Confusion Matrix as seen on figure 7 indicate correct predictions, and off-diagonal values indicate misclassifications.

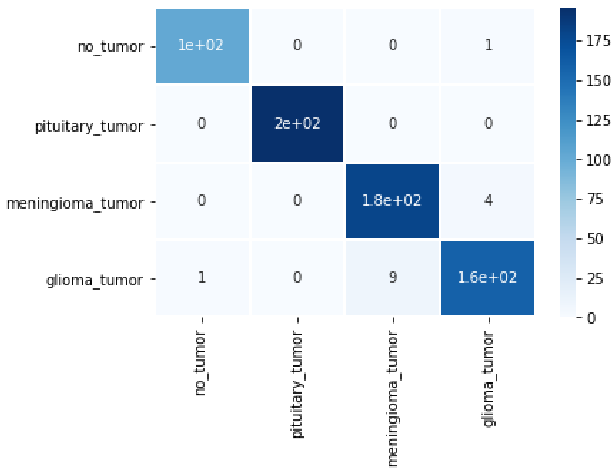

Figure 7. Confusion Matrix of EfficientNetB0

4. 2. 2. Xception Model

Xception is an extended version of the Inception architecture and performed as follows:

- Training Accuracy: 99.76%
- Validation Accuracy: 97.71%
- Testing Accuracy: 96.48%
- Precision, Recall, and F1-score were all above 92% for all classes.

Some misclassifications in the Confusion Matrix in the figure 8 between the meningioma_tumor and glioma_tumor classes were observed.

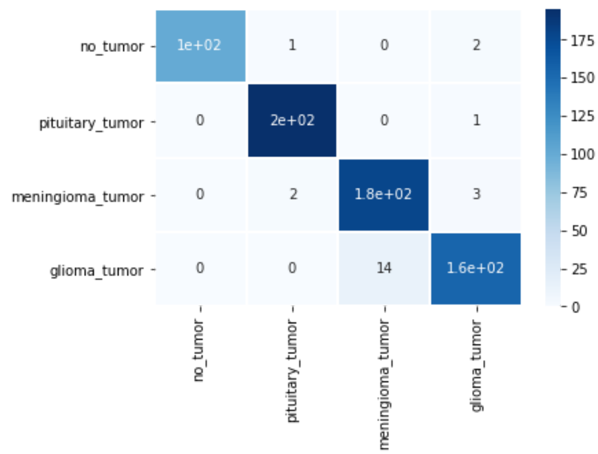

Figure 8. Confusion Matrix of Xception

4. 2. 3 MobileNet Model

MobileNet is a lightweight deep learning model suitable for mobile and embedded devices. It achieved the following results:

- Training Accuracy: 98.28%
- Validation Accuracy: 94.66%
- Testing Accuracy: 95.87%
- Precision, Recall, and F1-score were all above 93% for all classes.

Some misclassifications in the confusion matrix as seen on the figure 9 between the glioma_tumor and no_tumor classes were observed.

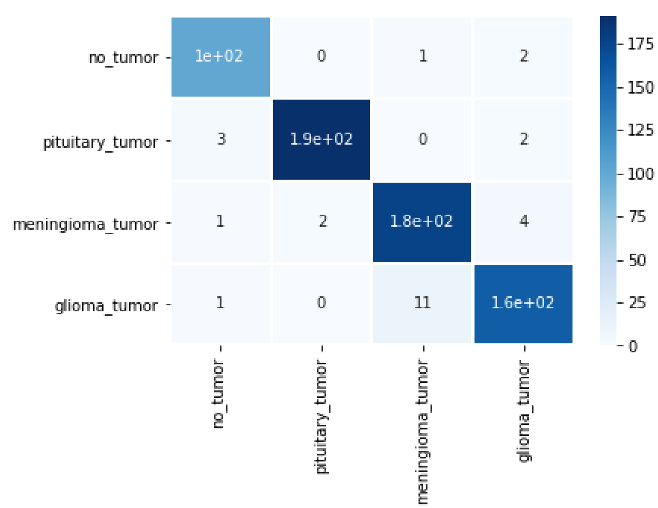

Figure 9. Confusion Matrix of MobileNet

4. 2. 4. ResNet Model

ResNet is a popular deep learning architecture with residual connections. It performed as follows:

- Training Accuracy: 95.42%
- Validation Accuracy: 91.22%
- Testing Accuracy: 95.42%
- Precision, Recall, and F1-score were all above 90% for all classes.

Some misclassifications between the glioma_tumor and pituitary_tumor classes were observed as seen on figure 10.

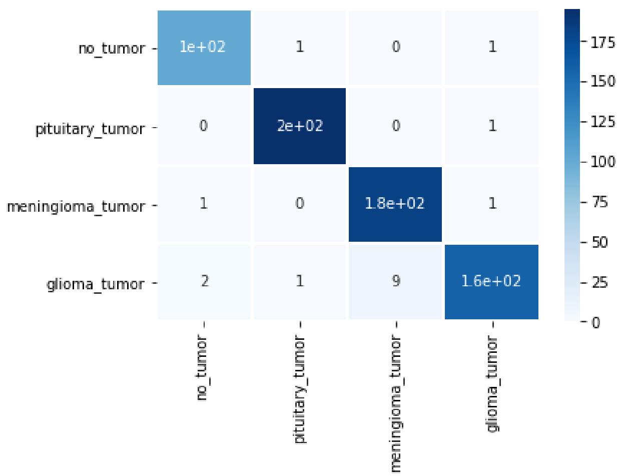

Figure 10. Confusion Matrix of ResNet50

6. Website building

We decided to create "a real website" for our project and we used to build it using Jekyll + GitHub pages. It was challenging and fun at the same time because we haven't used the Jekyll theme before but we learned it. This project website is created using the theme Start Bootstrap - Clean Blog Jekyll - Official Jekyll Version. Clean Blog Jekyll is a stylish, responsive blog theme for Bootstrap created by Start Bootstrap. Copyright 2013-2021 Start Bootstrap LLC. Code released under the MIT license. We managed to contribute all people in the team to the website building by cloning the repository into our local computers and doing the necessary changes. The website contains the homepage, an about page that explains the project and posts page where you can find our notebook files Experiment 1 and Experiment 2 converted into html. Also, the Streamlit Application is embedded into our website. The link to the website is as follows here: https://megixhafka.github.io/ .
More information about the website building procedure is found on the github repository README file. Link to the github repository:
https://github.com/megixhafka/megixhafka.github.io .
Enjoy!

7. Streamlit App Deployment

The classification model used in this application is based on EfficientNetB0, a state-of-the-art deep learning architecture for image recognition tasks. It has been pretrained on a large dataset and fine-tuned for brain tumor classification. The model is capable of processing MRI images of size 150x150 pixels. The model will predict the class of the brain tumor based on the uploaded image and display the result below the image.The accuracy of the model's predictions depends on the quality and nature of the uploaded MRI image. This app is for demonstration purposes only and should not be used for medical diagnosis or decision-making. Always consult a qualified healthcare professional for accurate medical advice. We have embedded the streamlit application into the website https://megixhafka.github.io/streamlit .
Also, here is the link to the streamlit github repository: https://github.com/taspips/Brain_Tumor_Classification_Experiment.git . You can see an example in figure 11, of how the model performs.

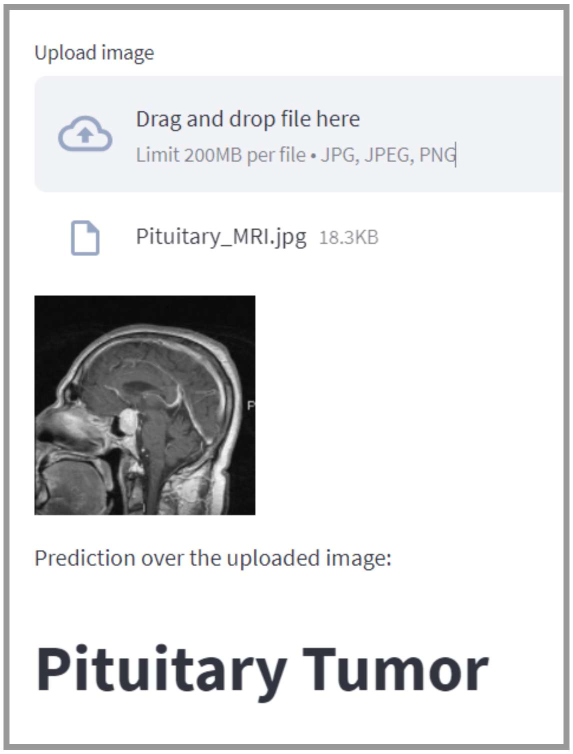

Figure 11. Example of the performance of the model. Uploaded image is an MRI image with a pituitary tumor, and the model gives the prediction to be a pituitary tumor.

8. Conclusion

Based on the evaluation metrics, EfficientNetB0 outperformed the other models with the highest accuracy of 97.70% on the test set. It also showed excellent precision, recall, and F1-score for all classes, indicating robust performance. Xception and MobileNet also performed well with accuracies of 96.48% and 95.87%, respectively. ResNet achieved a decent accuracy of 95.42%, but it had slightly lower precision and recall compared to other models.

In conclusion, **EfficientNetB0** is the recommended model for brain tumor classification from MRI images due to its superior performance in terms of accuracy and other evaluation metrics. However, the choice of model can also depend on other factors such as model complexity, deployment requirements, and hardware constraints.

9. References

[1] Msoud Nickparvar. (2021). *Brain Tumor MRI Dataset [Dataset]*. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/2645886

[2] Team, K. (n.d.). *Keras documentation: Image classification via fine-tuning with EfficientNet*. https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/

[3] Architjee. (2021). Brain Tumor Classification from MRI. *Kaggle*. https://www.kaggle.com/code/architjee/brain-tumor-classification-from-mri/notebook

[4] Mushfirat. (2022). Brain Tumor classification (Accuracy~96%). *Kaggle*. https://www.kaggle.com/code/mushfirat/brain-tumor-classification-accuracy-96

[5] Transfer learning and fine-tuning. (n.d.). *TensorFlow*. https://www.tensorflow.org/tutorials/images/transfer_learning

[6] *Fine-Tuning MobileNet on Custom Data Set with TensorFlow's Keras API*. (n.d.). Deeplizard.https://deeplizard.com/learn/video/Zrt76AIbeh4

[7] Benbrahim, Hayet & Behloul, Ali. (2021). Fine-tuned Xception for Image Classification on Tiny ImageNet. https://www.researchgate.net/publication/357755047_Fine-tuned_Xception_for_Image_Classification_on_Tiny_ImageNet

[8] Bakker, I. D. (n.d.). *Python Deep Learning Cookbook*. O'Reilly Online Learning. https://learning.oreilly.com/library/view/python-deep-learning/9781787125193/10750043-8f40-4339-be16-c3c036fec979.xhtml

[9] Manning Publications. (2021, October 6). *Fine-Tuning a Pre-Trained ResNet-50 - Manning*. Manning. https://freecontent.manning.com/fine-tuning-a-pre-trained-resnet-50/