MRI Brain Tumor Patch-Based Classification

Ronai, Or
orronai@gmail.com

Alt, Nitzan
nitzan.alt12@gmail.com

June 28, 2023

GitHub link: https://github.com/orronai/MRIBTCPB

Abstract

This paper explores the classification of brain tumors in MRI images using neural networks. Limited access to medical data and the high cost of obtaining labeled images pose challenges for training accurate models. To overcome these challenges, we employ transfer learning with pre-trained networks and self-supervised learning techniques. The proposed algorithm aims to achieve high accuracy while minimizing resource consumption. The study utilizes the Brain Tumor MRI Dataset and applies data augmentation and hyperparameter optimization. The results demonstrate the effectiveness of the approach in classifying different types of brain tumors within MRI images.

1 Introduction

MRI image classification and the broader application of neural networks in the medical field have emerged as prominent topics in recent years. The accessibility of medical data is restricted due to patient privacy concerns and the expense associated with obtaining classified images, necessitating the involvement of medical professionals in the classification process. Furthermore, the availability of freely accessible medical data online is severely limited, posing a significant challenge for training neural networks, which inherently require substantial amounts of data.

To overcome this challenge, one approach involves leveraging pre-trained large-scale networks such as ImageNet and employing transfer learning techniques to extract crucial features from MRI images. This strategy offers the advantages of reduced training time and enhanced generalization capabilities. However, when working with limited data, another viable method to enhance model accuracy is to utilize self-supervised training methods. By utilizing self-supervised techniques, the model can extract additional information from the same dataset, effectively leveraging the inherent structure or patterns within the data itself. This approach enables the model to make the most of the available data resources and further improve its performance.

Brain tumor classification in MRI images is a critical capability that holds significant potential for aiding humanity in several ways. Firstly, it can effectively reduce the time and manpower required for doctors to classify tumors accurately. Additionally, the application of automated classification algorithms can lead to faster tumor classifications, expediting the diagnostic process. It is important to emphasize that the objective is not to replace the work of doctors, but rather to provide valuable assistance in their medical assessments and decision-making processes, which is called: "CAD" (Computer-Aided Diagnosis Systems).

Our main goal in this paper is to present an algorithm that achieves both low resource consumption and high accuracy in classifying different types of brain tumors within MRI images. The research project focuses specifically on the classification of brain tumor images obtained from the Brain Tumor MRI Dataset [3].

2 Related Work

In this section, we discuss the prior works that have served as the basis for our project. We have drawn inspiration from the following studies:

- "Brain MRI Classification using PyTorch EfficientNetB0" [2]: This work focused on brain tumor classification using the EfficientNetB0 pretrained architecture in the PyTorch framework.
- "PatchResNet: Multiple Patch Division–Based Deep Feature Fusion Framework for Brain Tumor Classification Using MRI Images" [6]: This study explored brain tumor classification using the ResNet50 pretrained architecture. Additionally, the authors incorporated feature selectors and employed a multiple patch division-based deep feature fusion framework.

It is important to note that the first related work, which utilized the PyTorch EfficientNetB0 architecture, employed a different data splitting strategy compared to our project. Furthermore, their evaluation did not include a separate and independent test set. Instead, they relied solely on a validation set. Consequently, it remains uncertain whether they calibrated their hyperparameters in accordance with this validation set, potentially leading to overfitting of the model to the validation set's accuracy score.

In our project, we explored alternative methods to address brain tumor classification. These methods included dividing the MRI images into not overlapping patches and utilizing a self-supervised learning technique known as BYOL (Bootstrap Your Own Latent). By incorporating these approaches, we aimed to enhance the performance and robustness of our classification model.

3 Methods

3.1 Data-set

We utilized the Brain Tumor MRI Dataset [3], which consists of 7,022 labeled images, for our project. To ensure appropriate evaluation and avoid overfitting, we employed the train-validation-test splitting methodology.

Initially, the dataset was divided into a train set and a test set. Subsequently, to establish a robust model and fine-tune hyperparameters, we further split the train set. The train-validation split resulted in approximately 60% of the data allocated for training, while 20% was designated for validation and an additional 20% for the final evaluation on the test set.

By employing this train-validation-test splitting approach, we aimed to calibrate the hyperparameters of our model based on the performance observed on the validation set. This process helps mitigate overfitting by ensuring that the model's performance is evaluated on unseen data during the validation and test phases, leading to a more reliable and generalized assessment of its accuracy.

3.2 Augmentations

In our project, we incorporated augmentations as a technique to enhance the performance of our model, mitigate overfitting, and improve the overall accuracy. Augmentations involve applying various transformations to the input data, creating additional training examples with slight variations while preserving the underlying information.

The specific augmentations utilized in our project include techniques such as random rotations, horizontal or vertical flips, zooming, blurring, scaling, or changes in brightness, sharpness and contrast.

The addition of augmentations acts as a form of regularization, introducing subtle variations that prevent the model from relying too heavily on specific features or patterns in the training data. By exposing the model to a wider range of examples, augmentations encourage it to learn more robust and generalized representations, leading to improved accuracy on unseen data.

The utilization of augmentations is particularly crucial when working with small datasets, such as in our case, as it helps to artificially expand the training data and introduce additional variation, thus enabling the model to learn from a more diverse set of examples and improve its generalization capabilities.

3.3 Architecture

3.3.1 Supervised Learning

We employed transfer learning techniques by utilizing pre-trained neural networks that had been trained on the ImageNet dataset. The objective was to compare and evaluate the performance of these networks after fine-tuning them on our specific dataset.

To process the scan images effectively, we adopted a patch-based approach. Each scan image was divided into smaller not overlapping patches, and these patches were preserved without reshaping them to fit the batch dimension. Instead, we reshaped the batch dimension itself to accommodate all of the image's patches. By adjusting the batch dimension to include all the patches from the same image, we ensured that they were processed together, allowing for comprehensive feature extraction.

The patches were then forwarded through the pre-trained neural network until reaching the last layer. After this step, the patches were reshaped back to their original dimensions, retaining the individual characteristics of each patch. This process enabled us to extract meaningful features from the patches, capturing the essential information from the original scan image.

Following the feature extraction stage, a linear layer or classifier was applied to the extracted features. This layer utilized the features to construct a classifier specific to the brain tumor classification task.

In our project, we utilized several pre-trained neural networks, namely ResNet50 [7], EfficientNet-B4 [5], and DenseNet201 [4]. By leveraging these established architectures, we aimed to benefit from their learned representations from the ImageNet dataset. Fine-tuning these networks on our dataset allowed us to adapt their knowledge and features to the specific task of brain tumor classification, potentially enhancing the performance and accuracy in our domain.

These networks were trained using Cross Entropy Loss.

3.3.2 Self-Supervised Learning

In our pursuit of improving the accuracy of the model, we explored the application of self-supervised learning techniques. Given the limited size of our dataset, we recognized the potential benefits of learning rich data representations that could enhance performance. To this end, we implemented the BYOL (Bootstrap Your Own Latent) [1] method, utilizing a pretrained ResNet50 as our encoder to extract meaningful representations from the data.

By training with the BYOL method, we aimed to capture important latent features and structures within the dataset. The encoder learned to create useful representations by predicting the augmented version of the input samples from their augmented counterparts. This self-supervised learning approach allowed us to derive informative representations without the need for extensive labeled data.

Following the self-supervised learning phase, we extracted all the layers of the online encoder (excluding the last layer) and added a new linear layer. This new linear layer was introduced to create our classifier. Subsequently, we trained this modified network using supervised learning techniques, leveraging the labeled data available in our dataset.

By combining the benefits of self-supervised learning for representation learning and supervised learning for classification, we aimed to enhance the accuracy of our model. The self-supervised learning phase provided a foundation for learning informative representations, while the subsequent supervised learning phase fine-tuned the network for the specific task of brain tumor classification.

The BYOL method we used was trained using Negative Cosine Similarity Loss, and the final network was trained using Cross Entropy Loss.

3.4 Hyper-Parameters Search

To enhance the performance of our model, we used the Optuna Python package to conduct a hyperparameter search. By leveraging Optuna, we aimed to optimize key hyperparameters that significantly impact the model's accuracy and convergence.

The following hyperparameters were subjected to the hyperparameter search using Optuna:

- Learning rate
- Batch size
- Number of patches
- Optimization algorithm
- Scheduler

By exploring a range of values for these hyperparameters, we sought to identify the optimal configurations that would maximize the model's performance for brain tumor classification.

In addition, we utilized the pruning method provided by Optuna. This technique allows for early termination of unpromising trials, thereby conserving computational resources and expediting the search process. By incorporating pruning, we ensured a more efficient exploration of the hyperparameter space.

For the hyperparameter search, we conducted a total of 50 trials, allowing for a thorough exploration of the various hyperparameter combinations and their impact on the model's performance.

4 Results

In this section we'll present the performances of all the models as discussed. all models was trained for 30 epochs with different hyper-parameters that will be introduce in each related subsection.

Each model we trained on both augmented and not augmented train set, except of BYOL method which was trained only with the augmented version.

4.1 ResNet50

4.1.1 Selected Hyper-Parameters

- Learning rate: 0.0148
- Batch size: 32
- Number of patches: 49
- Optimization algorithm: SGD
- Scheduler: CosineAnnealingLR

4.1.2 Performance

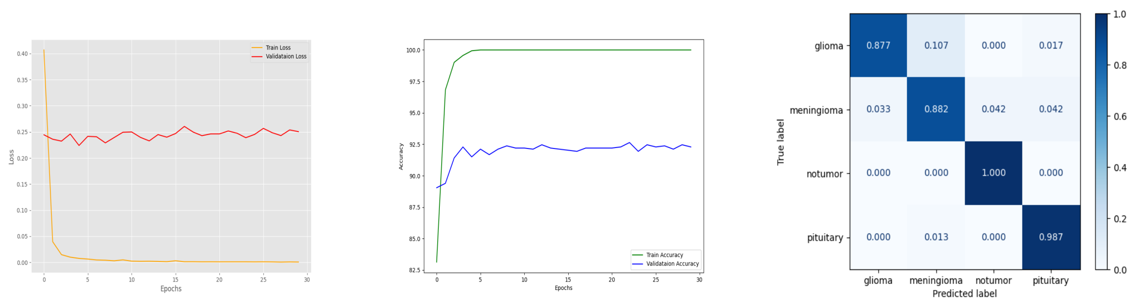

Figure 1: Loss, Accuracy and Confusion Matrix of Not-Augmented ResNet50

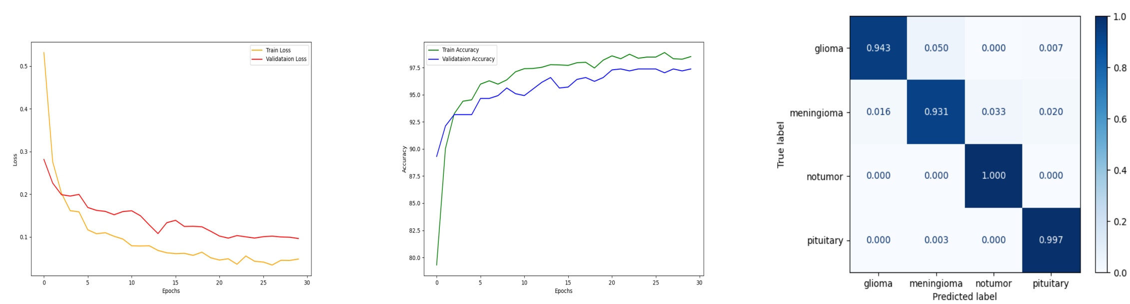

Figure 2: Loss, Accuracy and Confusion Matrix of Augmented ResNet50

4.2 DenseNet201

4.2.1 Selected Hyper-Parameters

- Learning rate: 5.577e-5
- Batch size: 32
- Number of patches: 4
- Optimization algorithm: RMSprop
- Scheduler: StepLR

4.2.2 Performance

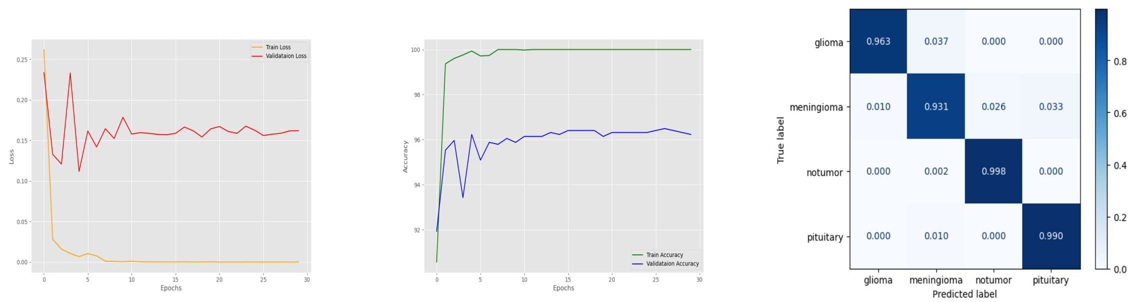

Figure 3: Loss, Accuracy and Confusion Matrix of Not-Augmented DenseNet201

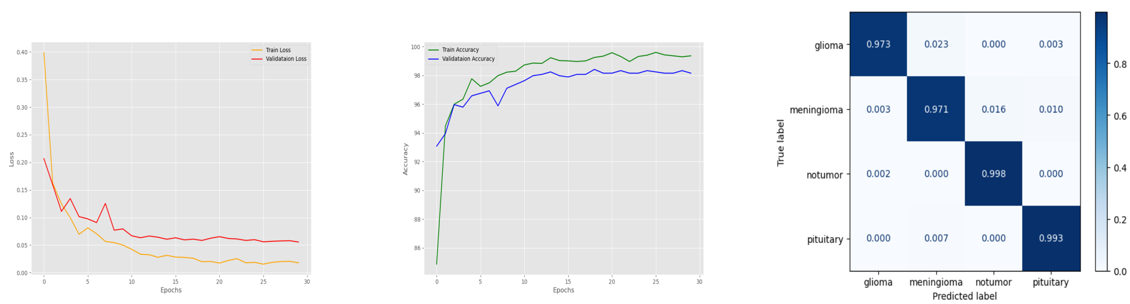

Figure 4: Loss, Accuracy and Confusion Matrix of Augmented ResNet50

4.3 EfficientNet-B4

4.3.1 Selected Hyper-Parameters

- Learning rate: 2.169e-4
- Batch size: 32
- Number of patches: 4
- Optimization algorithm: RMSprop
- Scheduler: CosineAnnealingLR

4.3.2 Performance

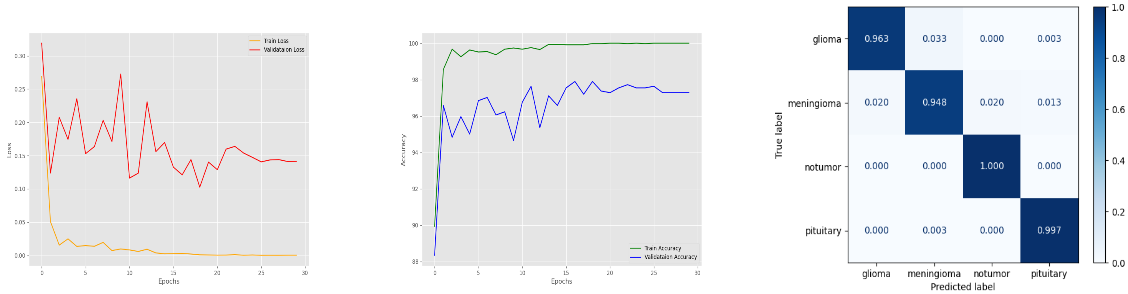

Figure 5: Loss, Accuracy and Confusion Matrix of Not-Augmented EfficientNet-B4

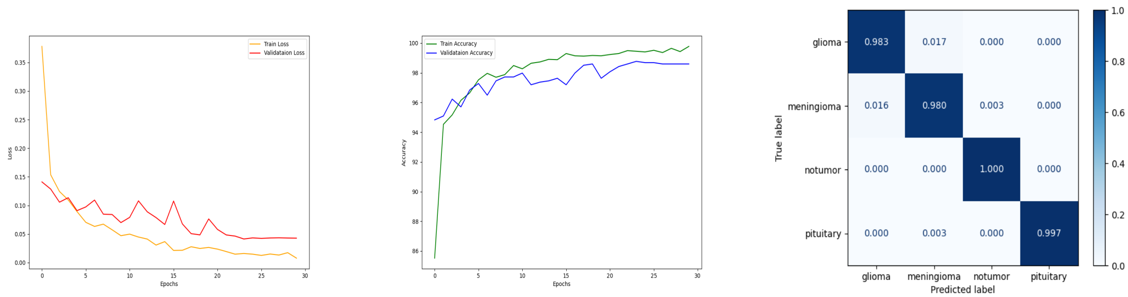

Figure 6: Loss, Accuracy and Confusion Matrix of Augmented EfficientNet-B4

4.4 BYOL+ResNet50

Due to limited resources, we employed the AMP (Automatic Mixed Precision) method to increase the batch size during training of BYOL. With the utilization of AMP, we were able to effectively handle larger batch sizes, thus enhancing the efficiency of the training process. Specifically, we used a batch size of 128 for this training.

4.4.1 Selected Hyper-Parameters For The Classifier Training

- Learning rate: 2.02e-4
- Batch size: 32
- Number of patches: 1
- Optimization algorithm: Adam
- Scheduler: CosineAnnealingLR

4.4.2 Performance

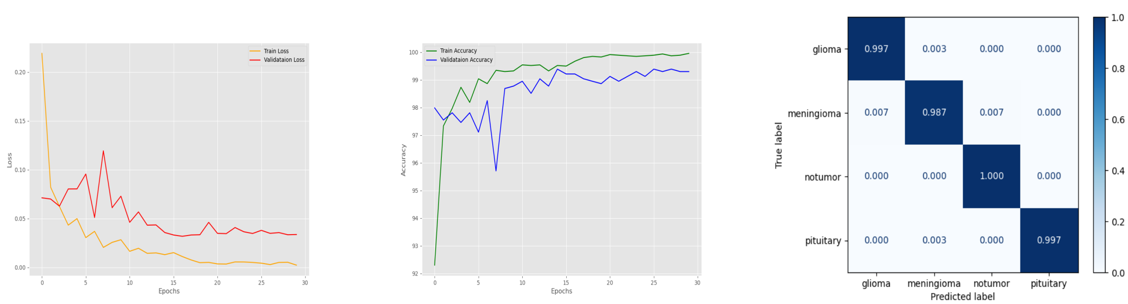

Figure 7: Loss, Accuracy and Confusion Matrix of BYOL+ResNet50

Additionally, we assessed the learned representation of the BYOL+ResNet50 method using T-SNE. T-SNE is a dimensionality reduction technique commonly used for visualizing high-dimensional data. In this evaluation, we applied T-SNE to the test set, randomly selecting 500 samples. We performed the visualization in both 2 and 3 dimensions, allowing us to gain insights into the distribution and separability of the learned representations:

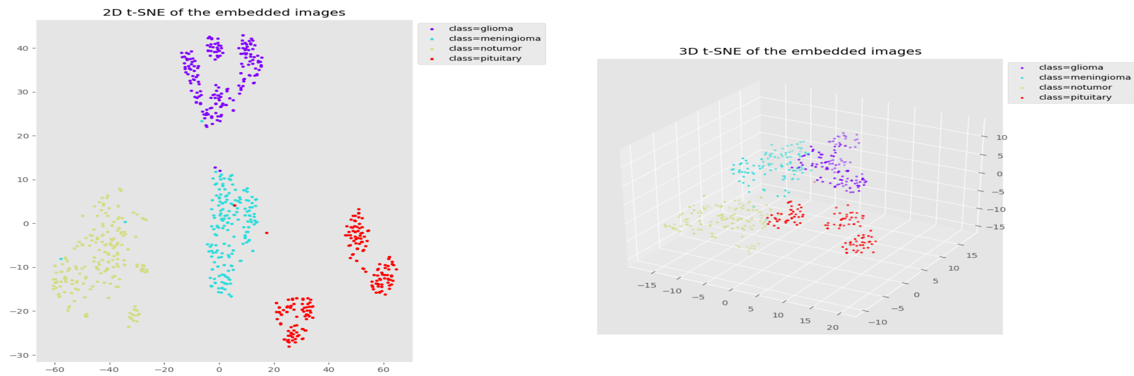

Figure 8: T-SNE Representation In 2D and 3D

4.5 Comparison

Model	Not-Augmented	Augmented
ResNet50	94.13%	97.03%
DenseNet201	97.25%	98.47%
EfficientNet-B4	97.86%	99.09%
BYOL+ResNet50	-	99.54%

Table 1: Accuracy of The Final Model on Test Set

The table clearly demonstrates that the patches method yielded favorable results, while the BYOL method outperformed it with even better results.

5 Conclusion

Based on the comprehensive analysis of the obtained results, it can be concluded that the selected augmentations played a crucial role in enhancing the performance and generalization of the models.

Among all the models evaluated, the one pretrained with the self-supervised learning method, specifically the BYOL method, demonstrated the highest accuracy. This outcome aligns with our expectations, considering the limited size of the dataset. Moreover, Figure 8 illustrates that this method effectively learned the representations, resulting in the formation of four distinct clusters, indicating its ability to capture meaningful patterns.

Moreover, it is noteworthy that employing the Patches method resulted in relatively poorer performance compared to other approaches. This observation indicates that the division of images into patches did not yield significant improvements in the overall model accuracy.

Overall, the findings emphasize the significance of augmentations in improving model performance, highlight the effectiveness of self-supervised learning methods such as BYOL, and underscore the limitations of the patch-based approach in the context of brain tumor classification.

6 Future Work

To further investigate the potential of the Patches method, which yielded promising results, future work could focus on implementing a network that divides each image into overlapping patches. This approach has the potential to capture more detailed and fine-grained information from the images, potentially leading to improved results in brain tumor classification.

By incorporating overlapping patches, the network can capture spatial relationships and contextual information more effectively, as neighboring patches would contain overlapping regions of the image. This approach can enhance the model's ability to detect subtle patterns and features that may be crucial for accurate tumor classification.

Furthermore, considering the potential computational overhead of processing overlapping patches, it would be beneficial to explore efficient implementation techniques and optimization strategies to ensure computational feasibility without sacrificing performance.

References

[1] *Bootstrap your own latent: A new approach to self-supervised Learning*. URL: https://arxiv.org/abs/2006.07733.

[2] *Brain MRI Classification using PyTorch EfficientNetB0*. URL: https://debuggercafe.com/brain-mri-classification-using-pytorch-efficientnetb0.

[3] *Brain Tumor MRI Dataset*. URL: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset?select=Training.

[4] *Densely Connected Convolutional Networks*. URL: https://arxiv.org/abs/1608.06993v5.

[5] *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. URL: https://arxiv.org/abs/1905.11946.

[6] *PatchResNet: Multiple Patch Division–Based Deep Feature Fusion Framework for Brain Tumor Classification Using MRI Images*. URL: https://link.springer.com/article/10.1007/s10278-023-00789-x.

[7] *Understanding ResNet50 Architecture*. URL: https://iq.opengenus.org/resnet50-architecture/.