Multi-Task Convolutional Neural Network for Brain Tumor Detection

Andre Thomas Gil Cifuentes
thomas.gil@udea.edu.co

December 2, 2025

1 Introduction

Accurate and efficient analysis of **Magnetic Resonance Imaging (MRI)** scans is critical for the diagnosis and treatment planning of brain tumors. Manual **segmentation** (delineating the tumor boundaries) and **classification** (determining the tumor grade or subtype) of these deceases are time-consuming and prone to variability.

In recent years, **Deep Learning (DL)** models, especially **Convolutional Neural Networks (CNNs)**, have demonstrated state-of-the-art performance in various computer vision and medical image analysis tasks.

This project proposes a Multi-Task Deep Learning architecture for the simultaneous segmentation and classification of brain tumors from MRI data.

2 Dataset Description

The dataset used in this project is BRISC [1], which has expert-annotated images for **brain tumor segmentation and classification tasks**. Addresses common limitations of similar datasets (e.g BraTS), including class imbalance and annotation inconsistencies.

Task	Train	Test	Total
Classification	5000	1000	6000
Segmentation	3933	860	4793

Table 1: Samples per Task

As shown in 1, the dataset consists of more than 3000 images for training. Moreover, for classification, we have balanced classes for *Glioma, Meningioma, Pituitary*, and *No Tumor*. Finally, Figure 1 shows a binary mask for segmentation purposes.

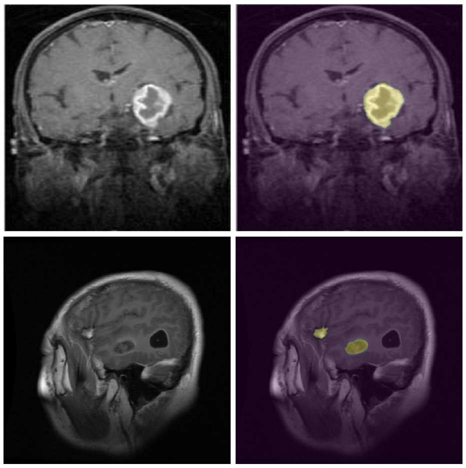

Figure 1: Right column shows MRI scans and the left column a segmentation mask is shown on top of the scan highlighting the area where the tumor is located.

2.1 Data Processing

The images discussed in Section 2 were first pre-processed to ensure they were in a consistent format suitable for model input. This involved several key steps:

- **Resizing:** All images were initially scaled to a standard size of 512×512 pixels. This uniform sizing is crucial as it ensures all inputs to the neural network have the same dimensions.
- **Grayscale Conversion:** The images were converted from color (RGB) to grayscale, effectively reducing the number of input channels from three to one. This step helps simplify the feature learning process, especially if color information is not critical to the classification task.
- **Training Crop:** For the images used during the training phase, a 224×224 pixel region was randomly selected and cropped from the larger 512×512 image. This introduces variability and focuses the model on different parts of the image content. The 224×224 size represents the final required input dimension for the model.

Finally, for both the training and testing datasets, the pixel data was converted into the specific tensor format and data type (32-bit floating-point) required by the PyTorch deep learning framework.

2.2 Data Augmentation

To enhance the robustness and generalization capabilities of the model, **Data Augmentation** techniques were applied exclusively to the training dataset.

This process artificially expands the effective size of the training data by applying random geometric transformations to the existing images.

The following augmentations were included in the training pipeline:

- **Random Cropping:** As noted above, a random crop was performed, which serves the dual purpose of preparing the input size and acting as an augmentation by varying the spatial context of the features.
- **Random Vertical Flipping:** With a 30% probability, an image was flipped vertically.
- **Random Horizontal Flipping:** With a 30% probability, an image was flipped horizontally.

These random transformations help prevent the model from overfitting to specific orientations or compositions present in the initial training images. Crucially, no such random augmentation was applied to the testing dataset, ensuring that model performance was evaluated fairly on the original image content.

2.3 Baseline CNN

We began our experiments by implementing a straightforward CNN architecture based on concepts introduced during the course. This initial model provided a simple baseline and helped us establish the overall workflow for data preprocessing, training, and evaluation. However, despite achieving low training loss, the model exhibited strong overfitting and failed to generalize well to the validation data. Although its performance was limited, this experiment served as a valuable first step to understand the complexity of the classification task.

2.4 Addressing Overfitting

To mitigate the overfitting observed in the baseline model, we focused on reducing model complexity and increasing data variability. First, we replaced the large fully connected layers with a global average pooling (GAP) layer inspired by [3], which significantly reduced the parameter count—from several million to only a few hundred thousand. This architectural adjustment encouraged the network to extract more meaningful representations of features while lowering the risk of memorization. We also incorporated data augmentation techniques such as random rotations, shifts, and flips to improve generalization. These modifications led to more stable learning dynamics and noticeably better performance in the validation set.

2.5 Transfer Learning with Pretrained Backbones

After improving the generalization of our baseline models, we extended our exploration by applying transfer learning. We experimented with several pretrained backbone architectures from [6, 5, 2, 3], and used partial freezing to preserve their low-level feature extractors while fine-tuning higher-level layers to the specifics of MRI data. This allowed us to benefit from the rich feature representations learned on large-scale datasets, leading to more accurate and reliable predictions than our earlier architectures. Across the models tested, transfer learning consistently provided stronger performance, confirming its suitability for this medical imaging task.

2.6 Segmentation with U-Net Decoder

Building on the best-performing pretrained backbones, we then trained full segmentation models by pairing each encoder with a U-Net decoder with the same structure as in [4]. This approach allowed us to combine strong feature extraction with a powerful, skip-connected decoder architecture specifically designed for dense prediction tasks. Using these hybrid models, we achieved our best tumor segmentation results, with clear improvements in boundary reconstruction and overall Dice performance. These experiments highlight the effectiveness of coupling modern pretrained encoders with U-Net–style decoders for brain tumor segmentation.

3 Results

3.1 Classification Task

Table 3.1 compares the performance of the evaluated classification models. While several architectures achieve strong results, MobileNet-V2 stands out with the highest F1 score (96.31%), recall (96.58%), and F1 precision (96.11%), despite having by far the smallest parameter count (59K). EfficientNet-B0 and ResNet-34 follow closely in terms of accuracy but require substantially more parameters. These results indicate that MobileNet-V2 provides the best balance between performance and efficiency, making it the most suitable choice for this classification task. For this reason, we would select MobileNet-V2 as our final classification model.

Model	F1 Score	Recall	F1 Precision	# Params
ImprovedBaseline	91.20%	90.77%	91.87%	393K
EfficientNet-B0	96.20%	96.30%	96.10%	417K
RestNet-34	94.57%	94.83%	94.37%	4.1M
MobileNet-V2	**96.31%**	**96.58%**	**96.11%**	**59K**
DenseNet	87.93%	88.21%	88.32%	6.1M

Table 2: Classification Results

3.2 Segmentation Task

Table 3.2 shows that EfficientNet + U-Net achieves a higher Dice score (84.1%) compared to MobileNet + U-Net (78.6%), although it also requires more parameters (4.6M vs. 2.9M). This indicates that the additional model capacity of EfficientNet leads to noticeably better tumor segmentation performance. For this reason, we would choose the EfficientNet + U-Net architecture for the MRI brain tumor segmentation task, as the gain in accuracy outweighs the increase in model size.

Model	Dice Score	# Params
EfficientNet + Unet	**84.1%**	4.6M
MobileNet + Unet	78.6%	**2.9M**

Table 3: Segmentation Results

4 Conclusions

In summary, although we did not fully achieve our initial goal of building a unified multi-task CNN capable of performing both classification and segmentation simultaneously, we made substantial progress toward it, laying a clear foundation for future work. Throughout the project and the course, we gained valuable insights into model design, regularization, and the practical challenges of working with medical imaging data. There remains significant room for improvement, particularly in experimenting with larger models and more extensive training schedules; however, our progress was constrained by the limited hardware available. These findings nonetheless provide a solid starting point for continued development in future iterations of the project.

References

[1] A. Fateh, Y. Rezvani, S. Moayedi, S. Rezvani, F. Fateh, M. Fateh, and V. Abolghasemi. Brisc: Annotated dataset for brain tumor segmentation and classification with swin-hafnet. *arXiv preprint arXiv:2506.14318*, 2025.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[3] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks, 2018.

[4] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.

[6] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.