Explainable AI for Brain Tumor Detection

Deep Learning Final Project

Group 1: Andreas Kaag Thomsen, Asger Song Høøck Poulsen,
Emil Hilligsøe Lauritsen, Niels Viggo Stark Madsen

December 11, 2024

1 Introduction

Brain tumor detection is a critical task in medical diagnostics, where early and accurate identification can significantly improve patient outcomes. With advances in deep learning, automated systems for detecting and classifying brain tumors using MRI scans have shown great promise. However, the complexity of these models often makes it difficult to understand their decision-making processes, which is crucial in high-stakes domains like healthcare. This project aims to develop a deep learning model for brain tumor detection while incorporating Explainable AI (XAI) techniques to provide insights into the model's predictions. By leveraging state-of-the-art architectures and interpretability methods, we strive to build a system that is not only accurate but also transparent, enabling clinicians to trust and verify the model's outputs.

2 Dataset

The Brain Tumor MRI Dataset[1] is a collection of 7023 grayscale MRI images organized into four distinct categories: glioma, meningioma, pituitary tumors and no tumors, each in its own folder. The dataset is divided into training, testing, and validation sets, following an approximate 80-10-10 split. Deliberate efforts were made to ensure a near-uniform class distribution to avoid any bias.

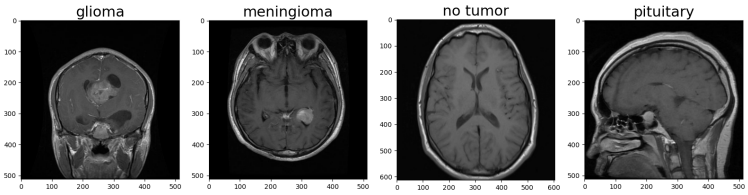


Figure 1: Images from the dataset with their respective labels.

2.1 Data Augmentation

Given the limited size of our dataset, we employed data augmentation techniques to address this constraint. Specifically, we focused on expanding the size of the dataset using the existing data rather than simply increasing the variability of the data. To ensure definitive results during prediction, augmentation was applied only on the training dataset [1].

For this purpose, we utilized the `albumentations`[2] framework, setting up five different transformations, each applied with a specific probability (0.2-0.6). These transformations were chosen to preserve medical relevance in the images while introducing controlled variability. The individual transformations are available in the code, and includes `HorizontalFlip` to enhance dataset diversity and `ImageCompression` to simulate lossy storage artifacts.

By applying these augmentations, we effectively increased the data size by a factor of N and chose $N = 2$ as we determined further replication would not generate enough variance.

2.2 Data Pre-processing

We have preprocessed the data through the following pipeline: First, each image is converted to grayscale and then expanded to 3 channels using `transforms.Grayscale(num_output_channels=3)`. This ensures compatibility with the used model that expects 3-channel inputs, as it is pre-trained on RGB images. The images are then converted into PyTorch tensors using `transforms.ToTensor()` to ensure proper compatibility. Finally, the images are resized to 224×224 pixels using `transforms.Resize` to ensure uniformity in sizes across the dataset. We wanted to have applied a custom `CropImgTransform` inspired by the guidelines provided in the dataset[3]. This transform identifies the extreme points in the image and crops it, ultimately discarding irrelevant areas. However, we encountered issues with this when introducing the transformations used for data augmentation, and thus discarded it. As we achieved good results without it, we deemed that it did not pose an issue. However, it is possible that having used this transform might have increased the performance of the model a bit.

3 Network Architecture

The architecture is based on the Vision Transformer (ViT) proposed by Dosovitskiy et al. in *'An Image is Worth 16x16 Words'* [2], which includes ViT-B(ase) (86M parameters), ViT-L(arge) (307M) and ViT-H(uge) (623M). All models share the structure shown in fig. 2.

The model starts with **image tokenization**, converting an input image into $N_P \times N_P$ patches (e.g. $N_P \in \{16, 32\}$). A learnable **linear projection** adjusts each token to length D. A `class` token is concatenated with the image tokens for prediction, and a **positional embedding** is added to encode token order. These tokens pass through L **transformer encoder blocks**, each with H attention heads and an MLP of size M, where the model learns from the tokens. Finally, the `class` token passes through an **MLP Head** for prediction. We use the ViT-B model ($L = 12$, $H = 12$, $D = 768$, $M = 3072$, $N_p = 16$) to accelerate training. We use the `IMAGENET1K_V1` weights as the pre-trained

[1] https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset/data

[2] https://github.com/albumentations-team/albumentations

[3] https://github.com/masoudnick/Brain-Tumor-MRI-Classification/blob/main/Preprocessing.py

initialization and add our output layer with 4 classes. Using the smaller patch size, we divide the image into a higher resolution of $14 \cdot 14 = 196$ patch tokens, which in turn allows higher resolution in our XAI attention rollout section 5.1.

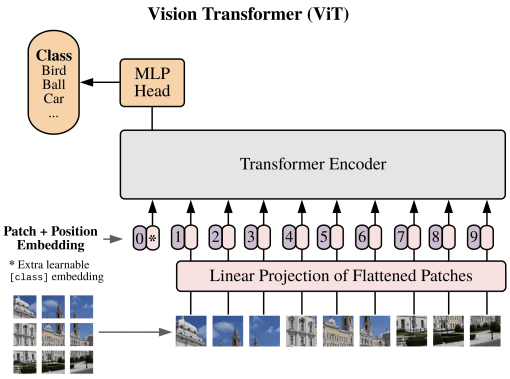


Figure 2: Model Overview - Taken from *An Image is Worth 16x16 Words* [2]

4 Training & Hyperparameter Tuning

To optimize the coding process, we have utilized the `pytorch-lightning` framework[4], which abstracts away a lot of boilerplate code. This framework simplifies model training and evaluation by providing a structured workflow, making it easier to integrate complex components like callbacks, learning rate schedulerss and checkpoints.

For the training process, we adopted the commonly favored Adam optimizer for its efficiency and stability. We used Optuna[5] for hyperparameter tuning via random grid search, focusing on batch size $bs \in \{32, 64, 128\}$ and learning rate $lr \in [10^{-4}, 10^{-3}]$. Batch sizes were categorical while learning rates were continuous. Training was limited to `max_epochs=20` with `early_stopping` (patience = 5). In 12 trials, Optuna optimized the validation loss to identify the best combination of hyperparameters with $bs = 128$ and $lr = 0.000165$, also summarized in the parallel coordinate plot in fig. 3[6]. The model training used cross-entropy loss as the objective function between targets and outputs.

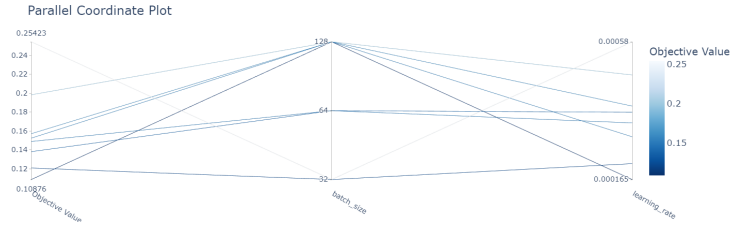


Figure 3: A parallel coordinate plot of all trials from the Optuna study with the objectice value (validation loss) on the far left.

[4] https://github.com/Lightning-AI/pytorch-lightning
[5] https://github.com/optuna/optuna
[6] Optuna did not account for the `early_stopping` callback, and as a result, the best loss value shown in fig. 3 does not correspond to the final model used.

5 Explainable AI Techniques

This project uses XAI techniques, specifically Attention Rollout and LIME, to explain our Vision Transformer model. The next section briefly overviews these methods.

5.1 Attention Rollout

Attention Rollout is a technique from *'Quantifying Attention Flow in Transformers by Abnar and Zuidema'* [3]. Unlike single-layer attention visualization, which interprets attention maps in isolation, this method combines attention maps across all layers to quantify the cumulative *flow* of attention from input to output. It assumes earlier layers influence later ones, necessitating a layer-wise aggregation of attention. Let A_i represent the attention matrix at the i-th layer averaged across all H heads, where $1 \leq i \leq L$. The attention rollout matrix R is recursively defined as:

$$R_{i+1} = R_i(A_i + I), \qquad R_0 = I \tag{1}$$

where I is the identity matrix, added to account for residual connections, ensuring information from previous layers is preserved. Normalizing $A_i + I$ ensures consistent attention scaling across layers. The final rollout matrix R_L represents the cumulative attention flow. To explain the classification of the model, we extract the column R_L corresponding to the class token, visualizing how the input patches attend to the classification decision.

5.2 LIME

Local Interpretable Model-agnostic Explanations (LIME), proposed by Ribeiro et al. in *'Why Should I Trust You?'* [4], is an occlusion-based framework designed to explain predictions from machine learning models.

For image classification, LIME begins by segmenting the input image into regions called superpixels using the algorithm SLIC. SLIC clusters pixels according to their intensity and spatial proximity, creating coherent regions that serve as interpretable units for explanation. Next, LIME perturbs the image by selectively turning superpixels on and off, when off, it replaces them with gray. This generates a set of perturbed images, which are then passed through the black-box model, in our case the ViT, to obtain predictions. These predictions provide insight into how the presence or absence of specific superpixels influences the model's output. Finally, LIME fits a linear regression model to approximate the relationship between superpixels and model predictions. The weights of this regression model indicate the importance of each superpixel, providing an interpretable explanation of the model's decision.

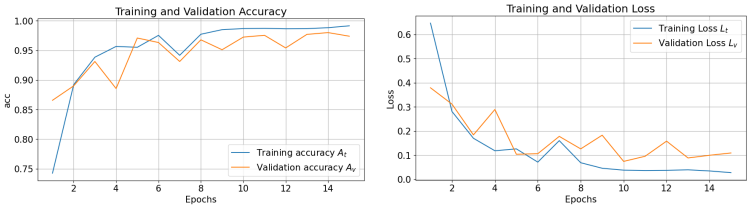
6 Results

This section provides an overview of the model's general performance using standard metrics and evaluates the explanations generated by our two tools. Then these explanations are compared with those provided by a medical student[7] for further validation.

[7] stud.med Josefine Halkjær

6.1 Model Performance

Figure 4 shows the accuracy and loss on the training and validation sets during training (L_t, L_v, A_t, A_v). L_t and A_t appear to be smooth and stabilize around epoch 10, while L_v and A_v fluctuate more. L_v reaches its lowest value (0.07) in epoch 10 where A_v is 0.97. Since L_v does not improve for 5 epochs, the early stopping is invoked. In conclusion, we arrive at the best model in epoch 10 with: $L_t = 0.03, L_v = 0.07, A_t = 0.99$, and $A_v = 0.97$. As the curves are relatively close to each other, the model does not seem to overfit. They both have high accuracy. This means we have a low bias and a low variance. With this model, we run the test dataset, which reaches an **accuracy of 0.97 and a loss of 0.090**.



(a) Accuracies. (b) Losses.

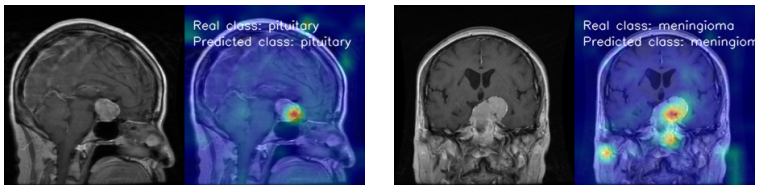
Figure 4: Loss and accuracy during training.

6.2 XAI Demonstration & Validation

To demonstrate the two XAI techniques, we apply them to test dataset samples, as shown in figs. 5 to 7. Due to space constraints, only a few samples are presented.

In fig. 5, the model successfully classifies the images, with the Attention Rollout highlighting specific regions. In fig. 5a, a distinct area is emphasized, clearly standing out. In contrast, in fig. 5b, multiple regions are highlighted, making it harder to pinpoint the location of the tumor. However, one of the highlighted areas may still indicate the tumor position, still offering practical value.

In fig. 6 the model highlights the superpixel that carries the highest weight in the local model. In fig. 6a we can see that the superpixel highlighted corresponds almost 1-1 with that of the medical student in fig. 7a. In fig. 6b the superpixel covers a larger area than the one in fig. 7b, however, still containing the most important information.

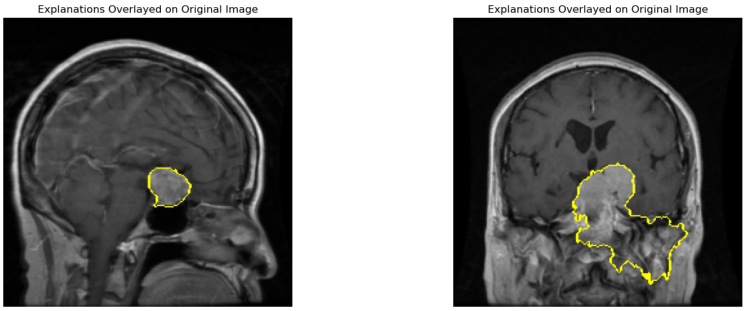


(a) Explanation of tumor with `pituitary` label. (b) Explanation of tumor with `meningioma` label.

Figure 5: Demonstration of XAI using Attention Rollout

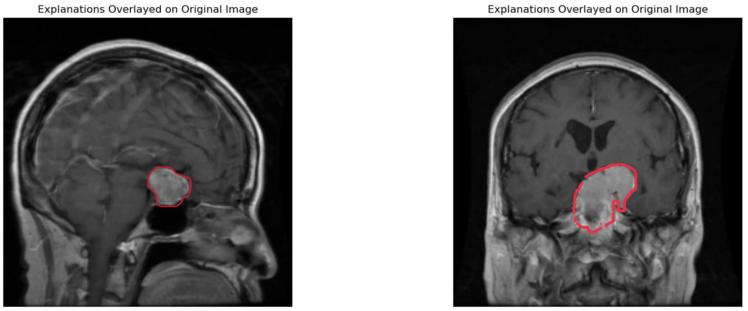
To validate our explanations, we have provided MRI scans to a medical student who is tasked with identifying the location of a given tumor. We then compare the student's findings to the explanations we have generated.

In general, both LIME and attention rollout generate explanations that largely correspond to those of the medical student. However, certain challenging test cases, not included in the report, reveal areas where both methods still face difficulties. These cases are not always the same; thus if one XAI method fails to give meaningful insights, the other will possibly be better.



(a) Explanation of tumor with `pituitary` label. (b) Explanation of tumor with `meningioma` label.

Figure 6: Demonstration of XAI using LIME; Visualizing The Most Important Superpixel



(a) Explanation of Tumor With `glioma` label. (b) Explanation of Tumor With `Meningioma` Label.

Figure 7: Demonstration of a Medical Student Explaining Tumors

References

[1] Aston Zhang, Zachary C. Lipton, Mu Li, et al. *Dive into Deep Learning*. https://D2L.ai. Cambridge University Press, 2023.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[3] Samira Abnar and Willem Zuidema. *Quantifying Attention Flow in Transformers*. 2020. arXiv: 2005.00928 [cs.LG]. URL: https://arxiv.org/abs/2005.00928.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. URL: https://doi.org/10.1145/2939672.2939778.