Classification of Brain Tumors in Multi-Modal MR Images Using Deep Spatio-Spatial Model

Aysu Kaymak
Gazi University
Computer Enginnering
aysu.kaymak@gazi.edu.tr

Muhammed İzzet Demir
Gazi University
Computer Engineering
muhammedizzet.demir@gazi.edu.tr

-

Abstract

This paper introduces a novel approach, the "Deep Spatio-Spatial Model," for brain tumor classification using multi-modal magnetic resonance imaging (MRI) data. Recognizing the significance of early diagnosis and precise treatment planning, the proposed model aims to enhance the accuracy of brain tumor categorization by capturing both spatial and temporal information in MRI scans. The paper reviews existing literature on deep learning methods in brain tumor classification, highlighting approaches such as volumetric Convolutional Neural Networks (CNNs) and fusion frameworks. Identifying a gap in the exploration of unified spatio-spatial models, the paper presents a detailed methodology, experimental design, and results obtained from applying the proposed model to a dataset of multi-modal MRI scans. The model, based on a modified ResNet18 architecture with pretrained weights from ImageNet, undergoes data preprocessing, including normalization and augmentation. The experimental results showcase the model's robust learning dynamics, achieving notable accuracy improvements during the 30-epoch training period. The validation accuracy peaks at 90.87%, demonstrating the model's strong generalization capabilities. The paper concludes with considerations for further analysis and refinement, emphasizing the potential of the Deep Spatio-Spatial Model in advancing brain tumor classification for improved patient care.

1. Introduction

Brain tumor classification using medical imaging plays a crucial role in early diagnosis and treatment planning. In recent years, deep learning techniques have shown significant promise in enhancing the accuracy of brain tumor classification, particularly when dealing with multi-modal magnetic resonance imaging (MRI) data. This paper presents an in-depth exploration of a novel approach termed "Deep Spatio-Spatial Model" for the classification of brain tumors using multi-modal MR images. The motivation behind this research stems from the need to improve the precision of brain tumor categorization, ultimately aiding medical professionals in making informed decisions for patient care.

Several studies in the last few decades have delved into the application of deep learning for brain tumor classification. Notable approaches, such as volumetric Convolutional Neural Networks (CNNs) [1], multi-step segmentation for grading-based categorization [2], and a fusion framework utilizing multiple imaging modalities [3], have contributed to the evolving landscape of brain tumor research. However, there remains a gap in the literature concerning the exploration of a unified "Deep Spatio-Spatial Model" that leverages both spatial and temporal information present in multi-modal MR images.

This paper proposes to fill this gap by introducing a deep learning model designed to capture spatio-spatial relationships within multi-modal MR images. The utilization of spatial and temporal correlations is expected to enhance the classification accuracy of brain tumors, providing a more comprehensive understanding of the underlying pathological features. The subsequent sections of this paper detail the methodology, experimental design, and results obtained from applying the Deep Spatio-Spatial Model to a dataset comprising MRI scans from different patients

2. Related Work

Many studies conducted in the last few decades have investigated the use of deep learning methods for brain tumor categorization. Using T1 contrast-enhanced images as a basis, Mzoughi et al. [1] suggested a volumetric Convolutional Neural Networks (CNNs) technique for identifying high-grade gliomas and low-grade gliomas. Their method used T1 contrast-enhanced imaging's unique characteristics to distinguish between various glioma grades.

A similar experiment on the grading-based categorization of gliomas was carried out by Pei et al. [2]. Their investigation began with a segmentation procedure to define the tumor region and then classified the samples as either Low-Grade Gliomas (LGG) or High-Grade Gliomas (HGG). This multi-step procedure is designed to improve the precision of glioma classification by first isolating the tumor location.

Ge et al. (2010) [3] presented a novel fusion framework for tumor classification that jointly used T1 contrast-enhanced, T2, and FLAIR (Fluid-Attenuated Inversion Recovery) images. This is in contrast to the current trend in the categorization and grading of glioma tumors, where most studies concentrated on examining one MR contrast image at a time. Ge et al. were able to obtain a more complete set of characteristics by the integration of different imaging modalities, which may have enhanced the accuracy of glioma classification.

These varied studies demonstrate the applicability of deep learning techniques in the field of brain tumor classification, exhibiting various approaches like multi-step segmentation, volumetric CNNs, and fusion of multiple imaging modalities to improve the classification models' robustness and accuracy. Investigating alternative approaches advances the continuous endeavors to create efficient and dependable instruments for the identification and categorization of brain cancers.

3. Method

The utilization of deep learning models has gained prominence in various computer vision tasks, including image classification. Models pretrained on extensive datasets like ImageNet have shown effectiveness in feature extraction and transfer learning.

Our main aim is to build a binary image classification model. We'll use a modified ResNet18 architecture, leveraging pretrained weights from ImageNet to boost the model's performance.

3.1. Data Preprocessing

- **Normalization:** We'll define mean and standard deviation values for image normalization, aligning with widely accepted defaults in ImageNet preprocessing. These values will be applied during training, validation, and testing phases.

- **Data Augmentation:** To improve model generalization, random horizontal flipping will be applied during training. Additionally, we'll resize images to a consistent size for model input across different phases.

3.2. Dataset Handling

The dataset will be organized into three phases: Training (TRAIN), Validation (VAL), and Testing (TEST). We'll leverage PyTorch's ImageFolder to structure and load the dataset. Also, data loaders will be created for each dataset phase, specifying batch size, shuffle, and num_workers. We'll calculate and store the size of each dataset phase (TRAIN, VAL, TEST) and retrieve class names from the training dataset.

3.3. ResNet Model

The Resnet models were introduced in the paper titled "Deep Residual Learning for Image Recognition." In this context, there are five variations of Resnet models with 18, 34, 50, 101, and 152 layers, respectively. Refer to Figure 1. for detailed information on the model architectures. The 1-crop error rates on the ImageNet dataset, using pretrained models, are provided below. [4]

Model structure	Top-1 error	Top-5 error
resnet18	30.24	10.92
resnet34	26.70	8.58
resnet50	23.85	7.13
resnet101	22.63	6.44
resnet152	21.69	5.94

Figure 1. The 1-crop error rates on the ImageNet dataset, using pretrained models

The ResNet (Residual Network) is a deep neural network architecture that was introduced to address the challenges of training very deep networks. It was proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their paper titled "Deep Residual Learning for Image Recognition," which was presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

The key innovation in ResNet is the use of residual learning blocks, which allow the network to learn residual functions instead of directly learning the desired mapping. Traditional deep networks stack layers on top of each other, and during training, as the network learns, it can become difficult for the network to make further improvements. This problem is known as the vanishing gradient problem, where gradients become very small as they are backpropagated through the layers during training. [4]

In ResNet, residual blocks are introduced to mitigate this issue. A residual block consists of a shortcut connection (or skip connection) that bypasses one or more layers. This shortcut connection allows the network to learn the residual between the input and output of the block. Mathematically, the output of a residual block can be expressed as [4]:

Output = Input+ F(Input)

where F(Input) is the residual function that the block aims to learn. The presence of the skip connection ensures that, during training, even if the network is having difficulty learning the desired transformation, it can still learn the identity mapping (the input is directly passed to the output through the shortcut connection). [4] This makes it easier for the network to learn the residual function.

ResNet architectures come in different depths, denoted by the number of layers. Common versions include ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, which correspond to the number of layers in each network, shown in Figure 2. [5] The deeper versions generally have more parameters and can capture more complex features but may also require more computational resources.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2. Resnet models

In summary, ResNet's success lies in its ability to train very deep neural networks by using residual blocks and skip connections, which help overcome the vanishing gradient problem and facilitate the training of deep models. This architecture has been widely adopted and achieved state-of-the-art performance on various computer vision tasks, including image classification and object detection.

3.4. Model Architecture and Training

A pretrained ResNet18 model from torchvision models will be loaded. The number of input features for the original model's fully connected layer will be extracted. We'll replace the fully connected layer with a new layer suitable for binary classification (2 classes) and move the model to the specified device (CPU or GPU).

The CrossEntropyLoss will be defined as the loss function for classification tasks. Stochastic Gradient Descent (SGD) will be utilized as the optimization algorithm with a specified learning rate. We'll implement a StepLR scheduler with a step_size of 7 and a gamma value of 0.1 to adjust the learning rate during training. The training process will iterate over the specified number of epochs, each comprising training and validation phases. The best model weights based on validation accuracy will be monitored and recorded, and the best model will be saved to a file.

4. Experimental Design

4.1. Dataset

The dataset consists of MRI scans from different patients, each scanned using four different techniques. The goal is to predict the presence or absence of the MGMT Biomarker based on radiogenomic features extracted from these scans. The associated CSV file provides labels for training the model, making it a binary classification problem.

4.1.1. Data Structure

Two main folders: "train" and "test." Each patient is identified by a code (e.g., "00000," "00001") and has a corresponding folder inside the "train" and "test" directories. [6]

4.1.2. Patient Folders

Inside each patient's folder, there are four subfolders shown below in Figure 3.: "FLAIR," "T1w," "T1wCE," and "T2w." Each subfolder contains MRI scans acquired using specific techniques: FLAIR, T1-weighted pre-contrast (T1w), T1-weighted post-contrast (T1Gd), and T2-weighted (T2). [6]

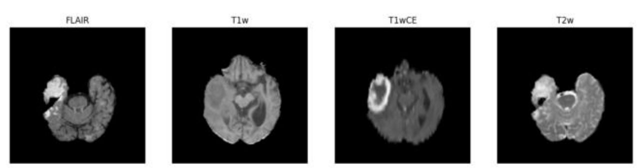

Figure 3. Dataset’s subfolders

4.1.3. Label Information

The dataset includes a CSV file named "train_labels.csv.". Columns in the CSV file:

- **"BraTS21ID":** Subject code representing each patient.

- **"MGMT_value":** Binary label (0 or 1) indicating the presence (1) or absence (0) of the MGMT Biomarker.

4.1.4. Label Distribution

The CSV file contains 585 entries. The "MGMT_value" column indicates whether the MGMT Biomarker is present (1) or absent (0) for each patient. [6]

4.1.5. Problem Type

Classification of Brain Tumors in Multi-Modal MR Images Using Deep Spatio-Spatial Model: Predicting the presence or absence of the MGMT Biomarker based on MRI scans.

5. Experimental Results

The training of the model occurred over 30 epochs, with periodic evaluation on the validation dataset. The key findings from these results are:

5.1. Training Process

The training loss steadily decreased from an initial value of 3.0781 to 0.5954 over the course of 30 epochs. The training accuracy improved from 60.49% to 93.00%.

5.2. Validation Performance

The validation loss decreased from 2.2196 to 1.7253 during the training process. The validation accuracy exhibited fluctuations but reached a peak of 90.87% around epoch 10.

5.3. Observations

The model demonstrates a strong learning capability, achieving high accuracy on both training and validation sets. The validation accuracy, while fluctuating, remains consistently high, indicating robust generalization.

5.4. Considerations

Further analysis, such as confusion matrices or precision-recall curves, may provide a more detailed understanding of the model's performance. Fine-tuning hyperparameters or exploring additional data augmentation techniques could potentially enhance results.

In summary, the model exhibits promising learning dynamics and generalization capabilities, achieving notable accuracy on both training and validation datasets. Further analysis and refinement could contribute to even better performance. Also, train and validation values are shown in Figure 4.

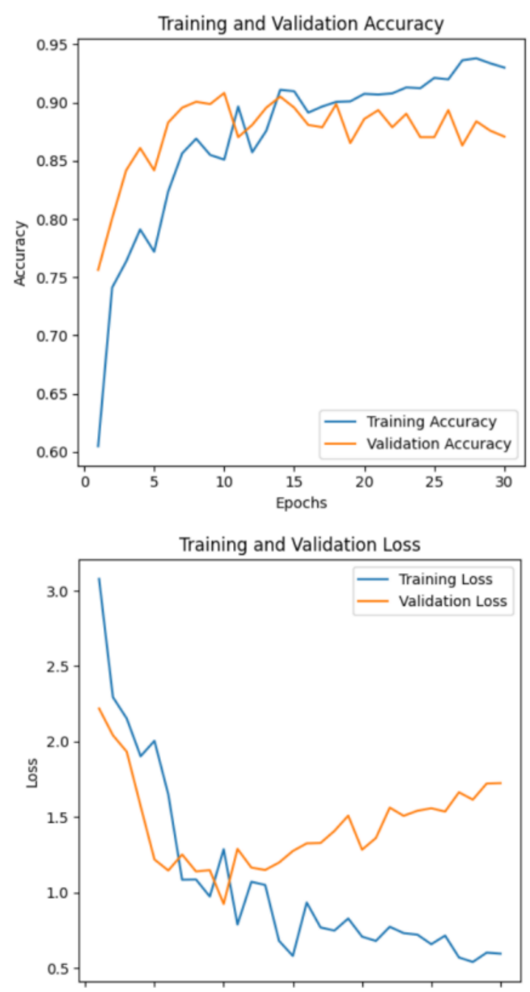

Figure 4. Train and validation values

6. Conclusions

A deep learning model utilizing a modified ResNet18 architecture with pretrained weights from ImageNet was developed for binary brain tumor classification. The RSNA-MICCAI Brain Tumor Radiogenomic Classification dataset, containing multi-modal MRI images, was used. Data preprocessing involved normalization and augmentation. The dataset was organized into training, validation, and testing phases. The model, trained for 30 epochs, demonstrated strong learning with an accuracy increase from 60.49% to 93.00%, and achieved a peak validation accuracy of 90.87% around epoch 10. Further analysis, like confusion matrices, could provide additional insights.

7. Bibliography

[1] Mzoughi, H. et al. Deep multi-scale 3d convolutional neural network (cnn) for mri gliomas brain tumor classification. J. Digit. Imaging 33, 903–915 (2020).

[2] Pei, L. et al. Brain tumor classification using 3d convolutional neural network. In International MICCAI Brain lesion Workshop, 335–342 (2019).

[3] Ge, C. et al. Deep learning and multi-sensor fusion for glioma classification using multistream 2d convolutional networks. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 5894–5897 (2018).

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[5] Shafiq, M., & Gu, Z. (2022). Deep residual learning for image recognition: A survey. Applied Sciences, 12(18), 8972.

[6] Flanders, A., Carr, C., Calabrese, E., Kitamura, F., inversion, Rudie, J., Mongan, J., Elliott, J., Prevedello, L., Riopel, M., sprint, Bakas, S., Ujjwal. (2021). RSNA-MICCAI Brain Tumor Radiogenomic Classification. Kaggle. https://kaggle.com/competitions/rsna-miccai-brain-tumor-radiogenomic-classification