BRAIN TUMOUR DETECTION AND CLASSIFICATION USING DEEP LEARNING MODELS

TEAM_NO: 26
Kunche Aishwarya : ai22btech11015
Talasani Sri Varsha: ai22btech11028
Yashwanth Nayak Rathlavath: em24mtech11007
Rega Sai Prasanth: em22mtech11003

Abstract

The classification of brain tumors is a crucial task in medical imaging, as accurate and timely diagnosis is essential for effective treatment planning and patient outcomes. Manual analysis of MRI images can be labor-intensive and subject to human error, necessitating automated solutions that can enhance diagnostic precision. In this paper, we propose a framework for automatic brain tumor classification leveraging state-of-the-art Convolutional Neural Networks (CNNs) and transfer learning techniques. We plan to utilize pre-trained models such as VGG16, ResNet50, and EfficientNet to extract robust features from MRI scans, allowing for effective fine-tuning tailored to our specific classification tasks. Our approach will involve comparing these CNN architectures against traditional machine learning classifiers, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests, implemented using scikit-learn. We anticipate that our CNN-based model will significantly outperform conventional methods, aiming for an accuracy benchmark exceeding 98%.

1. Introduction

Brain tumors are abnormal growths of tissue in which cells grow uncontrollably, often disrupting normal brain function. They may vary from benign (non-cancerous) to malignant (cancerous). Brain tumors are classified based on the cell type of the parent origin. Among the most common forms of brain tumors include gliomas, meningiomas and pituitary tumors. Gliomas are tumors originating from glial cells, which provide support and insulation to neurons. Meningiomas are tumors that originate from the membranes covering the brain and the spinal cord, and pituitary tumors arise within the pituitary gland, the organ responsible for the overall regulation of hormones in the human body. Early diagnosis of brain tumors can reduce damage dramatically because tumor progression can cause severe impairment of neurological function and general health.

Magnetic Resonance Imaging is one of the main diagnostic imaging devices in the characterization and evaluation of brain tumors. Still, interpretation of MRI images remains a time-consuming, relatively inaccurate process that relies on human decision-making, leading to inconsistency in diagnosis and categorization of the tumors. That is where artificial intelligence can help, particularly deep learning models, bringing radical innovation in interpretation procedures. CNNs are the most potent variant in the classification of images, such as in identifying and classifying various types of brain tumors based on MRI scans. The network feeds on images, which enables it to learn important features automatically so it is able to differentiate between various types of tumors and even the state that the brain scans are healthy with high accuracy.

In this paper, we will implement a deep learning model to automatically detect and classify brain tumors from MRI images into four categories: glioma tumor, meningioma tumor, pituitary tumor, and no tumor. This study would be intended to develop a reliable tool that may help radiologists in attaining quicker and more accurate decision-making related to the identification and treatment planning of the tumors. Under training, the model uses a CNN architecture with several layers of convolution, pooling, and dropout to make it optimize the performance and application of data augmentation techniques in order to increase the size of the dataset and prevent overfitting.

We will be able to overcome the two major problems in brain tumor diagnosis: the variability of manuals in image interpretation and the time needed in the process of diagnosis. Therefore, AI-powered tools can act as supplementary aids to human expertise in healthcare settings where timely and accurate decisions can play the difference between life and death. Our model assesses classification performance of our method in terms of accuracy, precision, recall, and the F1-score that evaluate robustness of it in practical life scenarios.

2. Literature Review

2.1. Approaches and methodologies for classifying medical images to predict brain tumors.

2.1.1 Convolutional Neural Networks (CNNs)

Krizhevsky, A.; Sutskever, I.; Hinton, G.E. (2017) ImageNet classification with deep convolutional neural networks. This seminal paper introduced AlexNet, a deep

CNN that significantly improved image classification performance on the ImageNet dataset. It demonstrated the power of deep learning for large-scale image recognition tasks.

Pashaei, A.; Sajedi, H.; Jazayeri, N. (2018) Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines. This paper explores the combination of CNNs and Extreme Learning Machine(ELMs) for brain tumor classification. It discusses the integration of these models and their performance. The combined approach of CNNs and ELMs showed improved classification accuracy, highlighting the benefits of integrating different deep learning techniques.

Qureshi et al. (2022) proposed an Ultra-Light Deep Learning Architecture (UL-DLA) for multi-class brain tumor detection. The model integrates deep features with textural features extracted using the Gray Level Co-occurrence Matrix (GLCM). These features are combined to form a Hybrid Feature Space (HFS), which is then fed into the deep learning model. The objective is to achieve high prediction accuracy with limited computational resources. The study demonstrated the model's effectiveness in detecting various types of brain tumors while maintaining a low computational load.

Zahoor et al. (2022) introduced a Deep Hybrid Boosted and Ensemble Learning-Based model for brain tumor analysis using MRI. The methodology involves combining multiple deep learning techniques, including convolutional neural networks (CNNs), to enhance the accuracy and robustness of tumor detection and classification. The model leverages the strengths of different deep learning approaches to improve diagnostic performance. The study achieved high accuracy in brain tumor analysis, demonstrating the effectiveness of the hybrid and ensemble learning approach.

2.1.2 Hybrid and Ensemble Learning Models

Ranjbarzadeh et al. (2021) presented a deep learning-based segmentation method using an attention mechanism and multi-modal MRI images. The methodology involves incorporating an attention mechanism into the deep learning model to focus on relevant features in the MRI images. The model is trained on multi-modal MRI datasets to improve segmentation accuracy. The study demonstrated significant improvements in segmentation accuracy and efficiency, highlighting the potential of attention mechanisms in deep learning models for brain tumor detection.

2.1.3 Transfer Learning and Ensemble Methods

Mazurowski, M.A.; Buda, M.; Saha, A.; Bashir, M.R. (2019)The paper discusses various deep learning architectures, with a particular focus on Convolutional Neural Networks (CNNs). It highlights the use of transfer learning, where pre-trained models such as VGGNet, ResNet, and Inception are fine-tuned on medical imaging datasets to improve performance.

They emphasize the importance of transfer learning in medical image analysis due to the limited availability of labeled medical data. The methodology involves:

Transfer Learning: Utilizing pre-trained models on large datasets (e.g., ImageNet) and fine-tuning them on specific medical imaging tasks.

Data Augmentation: Techniques such as rotation, scaling, and flipping to artificially increase the size of the training dataset.

Ensemble Methods: Combining predictions from multiple models to improve overall

The use of pre-trained models reduces the need for large labeled datasets and accelerates the training process. Ensemble methods further improve accuracy and robustness, making deep learning a valuable tool in medical image analysis[1].

Chartrand, G.; Cheng, P.M.; Vorontsov, E.; Drozdzal, M.; Turcotte, S.; Christopher, J.P.; Kadoury, S.; Tang, A. (2017) This primer introduces radiologists to the principles of deep learning, focusing on CNNs. It discusses popular architectures such as AlexNet, VGGNet, ResNet, and U-Net, which are commonly used in medical imaging.

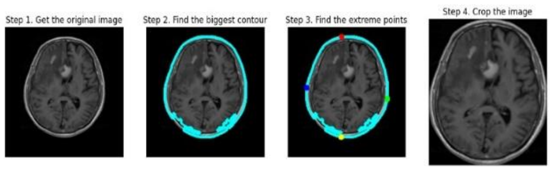
3. methodology:

3.1. Selecting the data set

We have sourced our dataset from Kaggle, and the link to the dataset is provided. The images are classified into four categories: Glioma, Tumour, Pituitary, and Meningioma. All the data is assumed to be well-labeled and processed. We believe that binary classification (predicting yes or no) is not sufficient for better treatment. Therefore, we have opted for a four-class classification, as the treatment varies according to the type of tumor.

3.2. Data preprocessing and argumentation

The process begins with data collection, gathering images that represent the desired classification categories. Next, data cleaning is performed to remove corrupted images and correct labels. Images are then resized to a uniform size, such as 224x224 pixels, to match the input shape required by the CNN. Pixel values are normalized to enhance convergence during training. Data augmentation techniques, including rotation, shifting, flipping, shearing, and brightness/contrast adjustments, are applied to artificially increase the size and variability of the training dataset. The data is then split into training, validation, and test sets, with 2870 images for training and 394 for testing. Images are batched to efficiently feed them into the CNN during training. Finally, label encoding is applied to encode the class labels.



4. Model training of data For classification task’

4.1. Using cnn

The model is designed with three convolution layers (1, 2, 3) of sizes 32, 64, and 128, respectively. It includes two dense layers (0, 1, 2) and performs a four-class classification of glioma, meningioma, no tumor, and pituitary. This approach focuses on hyperparameter tuning to find the optimal model configuration. The model follows a standard CNN architecture with ReLU activation and max pooling after each convolution layer. It uses sparse categorical cross-entropy loss and the Adam optimizer, and is trained for 20 epochs. Early stopping is introduced to prevent overfitting, and a 33% dropout is applied for regularization. The classification targets are glioma, meningioma, no tumor, and pituitary.

```
dense_layers = [0, 1, 2]
layer_sizes = [32, 64, 128]
conv_layers = [1, 2, 3]

for dense_layer in dense_layers:
    for layer_size in layer_sizes:
        for conv_layer in conv_layers:
            NAME = f"{conv_layer}-conv-{layer_size}-nodes-{dense_layer}-dense-{int(time.time())}"
```

4.2. Using VGG

The input for the model is an image of size 224x224x3. All convolutional layers use a kernel size of 3x3, a stride of 1, and padding of 1. After each convolutional layer, the ReLU activation function is applied. The output from the convolutional layers is flattened into a 1D vector, which is then fed into a fully connected (Dense) layer with 4096 units. The model is trained on 2870 images and validated on 394 images, over a span of 20 epochs.

4.3. Using VGG fine tune model

Fine-tuning VGG19 rather than training it from scratch yields superior performance for several reasons. Leveraging pre-trained features allows the model to use already-captured, general-purpose image patterns. This leads to faster convergence since the model starts with a solid foundation of learned features. Additionally, it can generalize more effectively on smaller datasets, reducing the risk of overfitting.

4.4. ResNet model

ResNet50, a pre-trained convolutional neural network (CNN) architecture from Keras applications, processes images with an input shape of (224, 224, 3). It randomly drops 40% of neurons for regularization. The network classifies images into four classes and uses cross-entropy as the loss function. It processes 32 images at a time (batch size) and trains for seven epochs. Early stopping is implemented, halting training if the validation loss doesn't improve for five consecutive epochs.

4.5. ResNet model with improved parameters

The data augmentation process involves using the `ImageDataGenerator` function with adjustments such as a rescale factor of 0.2, rotation range increased from 30 to 40, and added width and height shifts and fill mode. These changes help the model generalize better by exposing it to a wider variety of transformations. The number of epochs for training was changed from 7 to 10. Batch normalization was introduced to potentially improve training speed and accuracy by normalizing the input to each layer. Learning rate scheduling was also added, with `ReduceLROnPlateau` monitoring validation loss, reducing the learning rate by a factor of 0.2 after 3 patience epochs, down to a minimum learning rate of 0.00001. Finally, L2 regularization was added to the Dense layer to reduce overfitting by penalizing large weights, thus helping achieve better generalization on unseen data.

4.6. Vision transformer model and improved Vision transformer model.

The model architecture begins with an input layer of shape (224, 224, 3), corresponding to the image size and three color channels (RGB). It utilizes the Vision Transformer (ViT) base model from TensorFlow Hub, which is trainable. The output of the ViT is reshaped into a 4D tensor, adjusted as needed to fit the ViT's output shape (1, 1, 768). Following this, a Global Average Pooling 2D layer reduces each feature map to a single value by averaging, resulting in a flattened output shape of 768. A dropout layer with a rate of 0.4 is used to prevent overfitting by randomly dropping units during training. The final dense (output) layer contains 4 units, corresponding to the number of classes, and employs a softmax activation function to generate a probability distribution over these classes. The model parameters include sparse categorical cross-entropy as the loss function and the Adam optimizer with default settings. This setup ensures effective multi-class classification.

5. Results

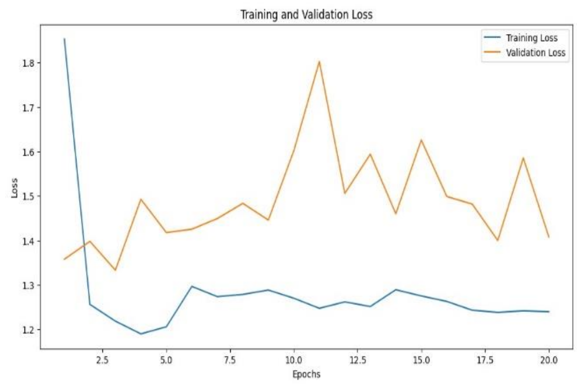
5.1. CNN

Model with layers	Test accuracy	Test loss
1-conv-32-nodes-0-dense	0.7487309575080872	2.17889690399169
2-conv-32-nodes-0-dense	0.7055837512016296	1.83383941650390
3-conv-32-nodes-0-dense	0.6573604345321655	2.20265197753906
1-conv-64-nodes-0-dense	0.7715736031532288	1.66127645969390
2-conv-64-nodes-0-dense	0.7588832378387451	1.42008006572723
3-conv-64-nodes-0-dense	0.6852791905403137	1.56843495368957
1-conv-128-nodes-0-dense	0.7436548471450806	1.89341390132904
2-conv-128-nodes-0-dense	0.7055837512016296	1.82416605949401
3-conv-128-nodes-0-dense	0.6675127148628235	1.97315526008605
1-conv-32-nodes-1-dense-	0.7055837512016296	1.14198994636535
2-conv-32-nodes-1-dense	0.510152280330658	1.19252824783325
3-conv-32-nodes-1-dense	0.5685279369354248	1.43919408321380
1-conv-64-nodes-1-dense	0.4720812141895294	1.18280303478240
2-conv-64-nodes-1-dense	0.703045666217804	1.96063685417175
3-conv-64-nodes-1-dense	0.6522842645645142	1.48325562477111
1-conv-128-nodes-1-dense	0.7081218361854553	2.55973386764526
2-conv-128-nodes-1-dense	0.7487309575080872	1.33904373645782
3-conv-128-nodes-1-dense	0.6725888252258301	1.66419303417205

1-conv-32-nodes-2-dense	0.25380709767341614	1.43077993392944
2-conv-32-nodes-2-dense	0.586294412612915	1.37160062789917
3-conv-32-nodes-2-dense	0.5203045606613159	1.36755216121673
1-conv-64-nodes-2-dense	0.5532994866371155	1.24637031555175
2-conv-64-nodes-2-dense	0.6751269102096558	1.07399523258209
3-conv-64-nodes-2-dense	0.5583756566047668	1.36648464202880
1-conv-128-nodes-2-dense	0.703045666217804	2.37447738647460
2-conv-128-nodes-2-dense	0.6269035339355469	1.63504171371459
3-conv-128-nodes-2-dense	0.4974619150161743	1.44549107551574

5.2. VGG original

The training of the model took 230 minutes, achieving a validation accuracy of 43.91%. The training accuracy started at 30.84% in Epoch 1 and increased to 37.53% by Epoch 20, with the training F1 score peaking at 0.3665 in Epoch 20. Initially, the validation accuracy improved from 35.79% in Epoch 1 to a peak of 43.91% in Epoch 3. However, it began to fluctuate and eventually declined, reaching its lowest at 31.73% in Epoch 18. The highest validation F1 score of 0.3588 was achieved in Epoch 7, after which the model's performance failed to improve, indicating potential overfitting. The VGG19 model, when trained from scratch, struggled to learn effectively from the dataset. Despite a slight improvement in training accuracy, the validation performance lagged. This may be attributed to the insufficient data, as VGG19 typically requires large datasets for effective feature learning.



Training and Validation Performance:

Epoch	Train Loss	Train Accuracy	Train F1 Score	Val Loss	Val Accuracy	Val F1 Score
1	1.8531	30.84%	0.3004	1.3577	35.79%	0.2962
2	1.2560	38.22%	0.3808	1.3978	40.10%	0.3449
3	1.2183	40.49%	0.4055	1.3330	43.91%	0.4060
4	1.1896	40.56%	0.4000	1.4925	32.23%	0.3055
5	1.2056	43.17%	0.3857	1.4175	40.61%	0.3935
6	1.2964	33.48%	0.2858	1.4252	35.03%	0.3129
7	1.2734	34.49%	0.3048	1.4492	38.83%	0.3588
8	1.2784	33.48%	0.2791	1.4833	40.36%	0.3699
9	1.2882	33.38%	0.2867	1.4457	32.99%	0.2965
10	1.2698	33.87%	0.3255	1.6019	36.29%	0.3130
11	1.2472	36.48%	0.3454	1.8025	30.20%	0.2469
12	1.2616	35.89%	0.3218	1.5057	36.04%	0.2324
13	1.2512	35.96%	0.3398	1.5940	35.53%	0.3017
14	1.2889	33.87%	0.3014	1.4598	34.77%	0.2471
15	1.2750	33.48%	0.3095	1.6259	36.29%	0.2992
16	1.2626	35.05%	0.3373	1.4988	40.36%	0.3194
17	1.2428	36.38%	0.3575	1.4815	36.80%	0.2739
18	1.2380	36.86%	0.3622	1.3997	31.73%	0.2292
19	1.2415	36.79%	0.3223	1.5854	41.12%	0.3458
20	1.2395	37.53%	0.3665	1.4080	35.28%	0.2322

5.3. VGG fine tuned

The final model reached its best validation performance in Epoch 7, achieving an accuracy of 84.52% and an F1 score of 0.8332. The overall final validation accuracy was 85%, and the final F1 score on validation was 0.7962. This indicates a strong performance and effectiveness in classification tasks for the given model.

5.4. ResNet

The model consists of a total of 23,587,712 parameters (89.98 MB), with 23,534,592 being trainable (89.78 MB) and 53,120 being non-trainable (207.50 KB). It was trained on 1,443 images across 4 classes and validated on 78 images belonging to the same 4 classes. Despite achieving good training accuracy in all epochs, the overall test accuracy remained low, prompting us to retrain the model with updated parameters. The accuracy shown is 25.6%.

5.5. ResNet with improved training parameters.

With the improved parameters training model the accuracy of the model has increased from 25.6 percentage in the original model to 47.43%. We believe that the accuracy of the model using ViT is low here because the trend images are very less and the data is not pre trained that is the model took raw data.

5.6. Vision transformer original model and with improved parameters.

With the improved parameters training model the accuracy of the model has increased from 24% in the original model to 41.02%. We believe that the accuracy of the model using ViT is low here because the trend images are very less and the data is not pre trained that is the model took raw data.

References

[1] Vinod Kumar Dhakshnamurthy Murali Govindan,Kannan Sreerangan Manikanda Devarajan Nagarajan andAbhijith Thomas Brain Tumor Detection and Classification Using Transfer Learning Models

[2] Mohan, G.; Subashin, M. MRI based medical image analysis: Survey on brain tumor grade classification. *Biomed. Signal Process. Control* **2018**, *39*, 139–161. [**Google Scholar**] [**CrossRef**]

[3] Al-Antari, M.A.; Al-Masni, M.A.; Choi, M.-T.; Han, S.M.; Kim, T.-S. A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification. *Int. J. Med. Inform.* **2018**, *117*, 44–54. [**Google Scholar**] [**CrossRef**]

[4] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Comms. ACM* **2017**, *60*, 84–90. [**Google Scholar**] [**CrossRef**]

[5] Qureshi, S.A.; Raza, S.E.A.; Hussain, L.; Malibari, A.A.; Nour, M.K.; Rehman, A.U.; Al-Wesabi, F.N.; Hilal, A.M. Intelligent Ultra-Light Deep Learning Model for Multi-Class Brain Tumor Detection. *Appl. Sci.* **2022**, *12*, 3715. [**Google Scholar**] [**CrossRef**]

[6] Zahoor, M.M.; Qureshi, S.A.; Bibi, S.; Khan, S.H.; Khan, A.; Ghafoor, U.; Bhutta, M.R. A New Deep Hybrid Boosted and Ensemble Learning-Based Brain Tumor Analysis Using MRI. *Sensors* **2022**, *22*, 2726. [**Google Scholar**] [**CrossRef**]

[7] Woźniak, M.; Siłka, J.; Wieczorek, M. Deep neural network correlation learning mechanism for CT brain tumor detection. *Neural Comput. Appl.* **2023**, *35*, 14611–14626. [**Google Scholar**] [**CrossRef**]

[8] Al-Galal, S.A.Y.; Alshaikhli, I.F.T.; Abdulrazzaq, M.M. MRI brain tumor medical images analysis using deep learning techniques: A systematic review. *Health Technol.* **2021**, *11*, 267–282. [**Google Scholar**] [**CrossRef**]

[9] Mahmud, M.I.; Mamun, M.; Abdelgawad, A. A Deep Analysis of Brain Tumor Detection from MRImages Using Deep Learning Networks. *Algorithms* **2023**, *16*, 176. [**Google Scholar**] [**CrossRef**]

[10] ZainEldin, H.; Gamel, S.A.; El-Kenawy, E.-S.M.; Alharbi, A.H.; Khafaga, D.S.; Ibrahim, A.; Talaat, F.M. Brain Tumor Detection and Classification Using Deep Learning and Sine-Cosine Fitness GreyWolf Optimization. *Bioengineering* **2023**, *10*, 18. [**Google Scholar**]

[11] Pareek, M.; Jha, C.K.; Mukherjee, S. Brain Tumor Classification from MRI Images and Calculation o