Advanced Brain Tumor Detection Using Enhanced Visual Information Processing

Xining Feng[1]

[1] Computer science and engineering, The Ohio State University, Columbus, OH 43210, USA
feng.1246@osu.edu

Abstract. In the fields of computer science, artificial intelligence, and medical science, brain tumor detection is quite important. This paper proposes a strategy utilizing Convolutional Neural Networks (CNNs) to build a detection model. CNNs, by simulating the human brain's processing of visual information, can automatically learn advanced features from images and accomplish tasks such as image recognition and classification. The goal of this paper is to utilize basic and straightforward methods to achieve accurate brain tumor detection. Additionally, the paper compares the influence of the early stopping method and thresholding approach on the proposed model. The outcomes reflect the influence of 2 methods is similar. The test accuracy for the model with the early stopping method is approximately 97.5%, while the model employing the thresholding approach achieves a test accuracy of about 98%. Brain tumor detection using CNNs signifies a significant advancement in computer science and artificial intelligence, with immense practical significance in the medical field and for patient well-being. By leveraging the power of automation, CNNs aid radiologists in detecting tumors with greater speed and precision, paving the way for earlier diagnosis and therapeutic interventions. This, in turn, can significantly enhance patient outcomes, reduce the risk of complications, and potentially save lives.

Keywords: Brain Tumor Detection, Convolutional Neural Networks (CNNs), Early Stopping Method.

1 Introduction

The human body is a complex system comprised of diverse cells, with each cell fulfilling a specific function. Among these, brain cells stand out as highly specialized components that constitute the central nervous system. However, when these cells undergo unregulated growth, a pathological condition known as a brain tumor emerges, posing a significant threat to human health. Since the intricate and detailed anatomical structure of the brain, brain tumor detection poses a significant challenge [1]. An analysis was furnished from the Central Brain Tumor Registry of the United States, approximately 39,550 individuals received a diagnosis of both malignant and

Y. Wang (ed.), *Proceedings of the 2024 International Conference on Artificial Intelligence and Communication (ICAIC 2024)*, Advances in Intelligent Systems Research 185,
https://doi.org/10.2991/978-94-6463-512-6_76

benign brain tumors, highlighting an incidence rate of 14 cases per 100,000 individuals for both malignant and benign primary brain tumors [2].

In medical diagnosis, Magnetic Resonance Imaging (MRI) is vital, especially for brain tumor detection. It offers superior diagnostic capability compared to Computed Tomography (CT) scans, as it does not use radiation. MRI utilizes a magnetic field to generate detailed images of internal structures, including the brain, where radiologists can assess the presence of tumors [3]. Automated tumor detection in MRI images is essential for precision in medical transactions involving human lives. A technique for identifying brain cancers in MRI data was presented by Natarajan et al. [4]. They preprocess images with a median filter, segment them using thresholds, apply morphological operations, and use image subtraction to obtain the tumor region. This approach accurately identifies the tumor's shape. Megeed and Aminproposed an intelligent system consisting of hybrid neural nets based on the Principal Component Analysis (PCA) for global segmentation and extraction of MRI features depending on the Wavelet Multiresolution Expectation Maximization (WMEM) for local feature extraction [5]. Multi-Layer Perceptron (MLP) classifies these features. The study compares MLP performance using both approaches, aiming to improve diagnosis speed, accuracy, and assist non-experts. Zhang and Xu presented a novel approach combining threshold segmentation and morphological operations to automatically detect suspicious regions or tumors [6]. This hybrid method leverages the strengths of both techniques, resulting in improved detection and extraction of tumor zones. Iftekharuddin et al. fused texture features and intensity data from multimodal MRI for pediatric classification and categorization of brain tumors using an unsupervised neural network called the Self-Organizing Map [7]. Segmentation of brain tumors utilizing Berkeley wavelet transformation (BWT) was investigated by Bahadure et al. who also retrieved relevant features to improve support vector machine (SVM) classification [8]. Validation is done on MRI data using accuracy, sensitivity, specificity, and dice similarity index. Abdalla and Esmail proposed to identify brain tumor combining MRI and artificial neural networks (ANN) [9]. It uses feedforward ANNs for segmentation, preprocessing, classification, and statistical feature extraction. Saeedi et al. analyzed brain MRI utilizing Convolutional Neural Network (CNN), 6 machine-learning techniques, and auto-encoder for improved categorization of gliomas, meningiomas, pituitary tumors, and healthy brains [10].

Improving the precision of MRI brain tumor detection is the main goal of the research. Specifically, the study begins with Exploratory Data Analysis (EDA) to uncover potential patterns, relationships, outliers, and other significant observations in the data, providing a foundation for subsequent analysis and modeling. Multiple convolutional layers are then employed to enhance precision. The images are also transformed from the blue-green-red (BGR) color format to the red-green-blue (RGB) format because Open-Source Computer Vision (OpenCV) library reads images in BGR by default, and the RGB format is more suitable for the algorithm. By introducing CNN to enhance the representation of features, a multi-level network architecture ensures effective feature extraction. This approach improves the local and global feature summarization of MRI images through combined convolutional and fully connected operations, enhancing the visual information. The outcomes of the

experiment reveal that the precision of the two-dimensional CNN can reach 99%. This study aims to assist healthcare professionals in improving the effectiveness of disease monitoring and prevention.

2 Methodology

2.1 Dataset Description and Preprocessing

In the research, the dataset has been used is called Br35H: Brain Tumor Detection 2020, which is from Kaggle [11]. This data set involves 1500 Brain MRIs, which are all depicting tumors, and 1500 Brain MRIs that do not display any tumorous characteristics. Each image is grayscale and different in size. The features that dataset contains are tumorous label and non-tumorous label, which indicate if the brain that is displayed is tumorous or not. Fig. 1 exemplifies several examples drawn from the dataset.

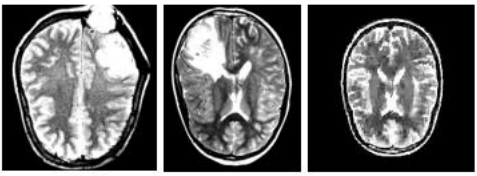

Fig. 1. Pictures in the Br35H: Brain Tumor Detection 2020 dataset [11].

2.2 Proposed Approach

The method for tumor detection is comprised of Exploring Data Analysis (EDA), data scaling and transformation, and Sequential model. This method is based on the sequential model, combined with a two-dimensional CNN. When integrated, these technologies will proficiently extract and harness spatial as well as contextual data from brain imagery, ultimately enhancing the efficiency of brain tumor detection tasks. The diagram depicted in Fig. 2 below outlines the architecture of the system. For the preprocessing part, to align with the processing requirements of subsequent steps, the BGR format is converted to RGB for the images, as the OpenCV library natively recognizes BGR-formatted images. To ensure standardization and compatibility across images, all images are resized to a predetermined dimension (128*128). After resizing, the images are compiled into lists, along with their associated labels, to streamline accessibility and handling during subsequent evaluations. Also, As the data comprises images, and images are comprised of pixels, a normalization step is applied where each pixel's value is divided by 255 (since pixels can have values ranging from 0 to 255). This guarantees that all values for pixels are scaled to the same [0,1] range.

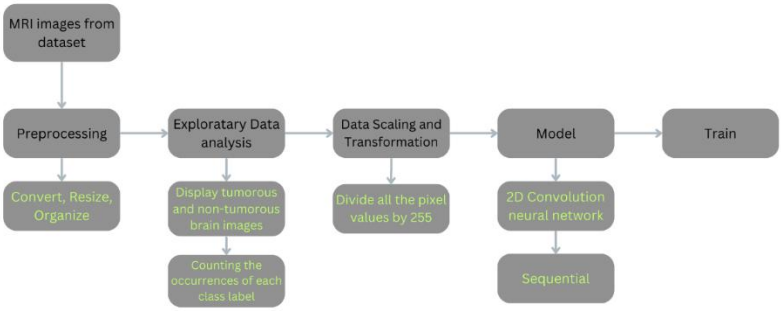

Fig. 2. The architecture of the system (Photo/Picture credit: Original).

Exploring Data Analysis (EDA)

EDA serves as a technique to analyze a dataset, aiming to capture its key characteristics, typically achieved through the utilization of statistical graphics and various data visualization tools. Fig. 3 illustrates the images of brain that is tumorous and brain that is non-tumorous. Fig. 4 presents the distribution of class labels.

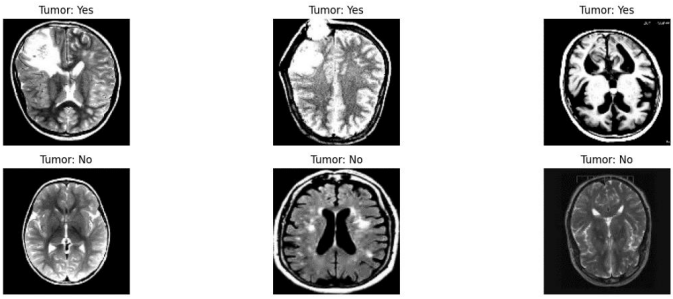

Fig. 3. Pictures in dataset with labels (Photo/Picture credit: Original).

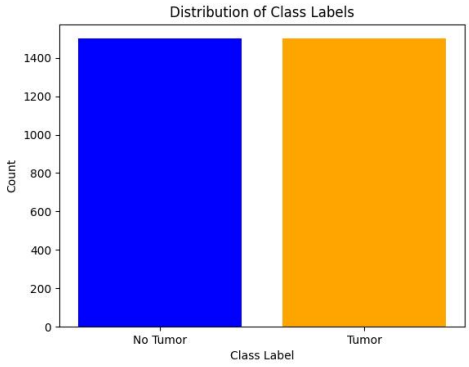

Fig. 4. The distribution of class labels (Photo/Picture credit: Original).

Two-dimensional CNN

The type of model for the system is sequential. The 2-dimensional CNN system is comprised of eleven layers, which are three convolutional layers, one flatten layer, three pooling layers, two dropout layers, two dense layers. The input size is 128*128 with RGB. Fig. 5 showcases the system's organizational framework.

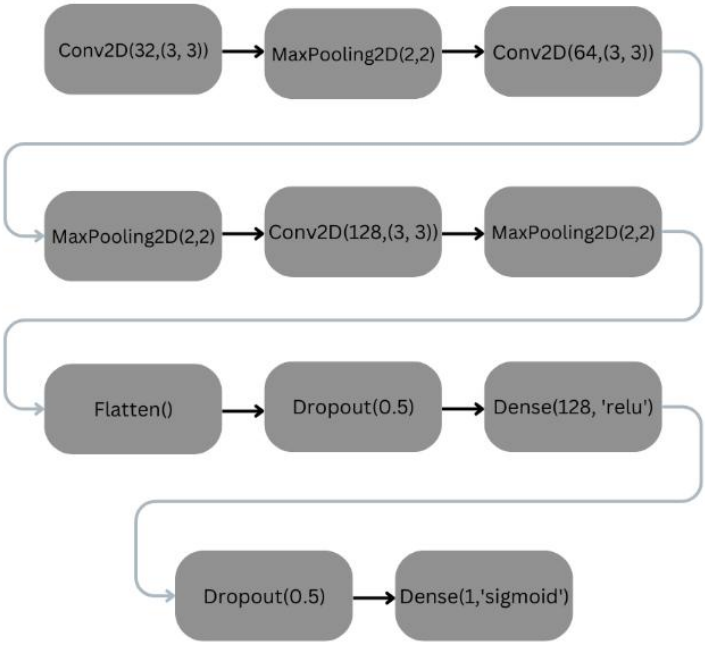

Fig. 5. The framework of 2D convolution network (Photo/Picture credit: Original).

The convolutional layer encompassed inputs from the initial layer obtained the features that were fully connected to the Universal Fully Connected (Ufc) layer, which has 128 hidden layers. The weights' count (Wconv) relied on the count of hidden layers, filters' quantity (n), and the dimension of the convolution output layer before it (z1*z2). Consequently, the calculation for the convolutional layer was the 2D convolutions' construction network determined as follows [10]: Wconv = z1*z2*n*Ufc = 14*14*128*128 = 3,211,264. This encompassed 3,211,264 parameters for the first fully connected layer, along with 128 biases, totaling 3,211,392 parameters.

A condensed overview of the network's learning parameters is presented within Table 1. According to Table 1, the total of each parameter's value, which categorize the network into 4 distinct groups, is arrived at by aggregating the values mentioned in Table 1's parameter column. The result amounts to 3,304,769, indicating that all parameters are eligible for training.

Table 1. The framework of 2D convolution network.

Layer	Output Shape	Parameter
convolution	Batch: None, Filters: 32, Size: 126*126	896
Max pooling	Batch: None, Depth: 32, Size: 63*63	zero
convolution	Batch: None, Filters: 64, Size: 61*61	18,496
Max pooling	Batch: None, Depth: 64, Size: 30*30	zero
convolution	Batch: None, Filters: 128, Size: 28*28	73,856
Max pooling	Batch: None, Depth: 128, Size: 14*14	zero
flatten	Batch: None, Vector: 25088	zero
dropout	Batch: None, Features: 25088	zero
dense	Batch: None, Neurons: 128	3,211,392
dropout	Batch: None, Features: 128	zero
dense	Batch: None, Neurons: 1	129

The initial layer of convolution contains 32 convolution kernels with a size of 3x3. The Rectified Linear Unit (ReLU) activation function is utilized to introduce non-linearity into the model. The function is as follows:

$$f(\gamma) = max\,(0, \gamma) = \frac{\gamma + |\gamma|}{2} = \begin{cases} \gamma, \ if\ \gamma > 0 \\ 0, \ otherwise \end{cases} \tag{1}$$

Following the layer of convolution, the max pooling layer with a pooling window size of 2x2 is implemented. By down sampling, this max pooling layer works to decrease the feature maps' spatial dimensions, thereby lessening the computational demands and aiding in the prevention of overfitting. There are 64 3x3 convolution kernels in the second layer of convolution. Following the convolutional layer, a layer of max pooling having a pooling window dimension of 2x2 is utilized to execute a downsampled. The third convolutional layer comprises 64 convolution kernels, each having a dimension of 3x3. These convolutional layers progressively extract features from low-level to high-level in the images. Subsequent to the layer of convolution, a layer for maximum pooling is introduced, utilizing a pooling window dimension of 2x2. After convolutional and pooling layer combination, multi-dimensional feature maps become converted into a one-dimensional feature vector by the flatten layer. Then, the model introduces a Dropout layer, which randomly "turn off" a portion of the neurons to enhance the robustness and generalization ability of model. Next, a fully connected layers (Dense) with the initial layer contains 128 neurons and utilizes ReLU function. A layer of dropouts is next. The second dense layer is the layer of output, containing only one neuron and applying sigmoid activation function to output a probability score indicating the confidence of the image belonging to the target category. Sigmoid function is as follows:

$$Sigmoid: \sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{1+e^{-x}} = 1 - \sigma(-x) \tag{2}$$

Loss Function

The loss function is an essential metric that measures the distinction between the predictions and the true labels. For the loss function, the binary cross-entropy is selected for handling binary categorizing problems. The binary cross-entropy loss function is suitable for handling classification problems with two categories, effectively measuring the distinction between the actual probability distribution and the model's anticipated probability distribution. By minimizing binary cross-entropy loss, this paper can gradually improve the model's prediction accuracy during training, as:

$$T = \frac{1}{N} \quad {}_i T_i = \frac{1}{N} \quad {}_i - [\theta_i \cdot log\,(s_i) + (1 - \theta_i) \cdot log\,(1 - s_i)] \tag{3}$$

where, θ_i shows the sample (i) label, where one denotes the positive class and zero the negative class and s_i indicates the expected likelihood that sample (i) will be in the positive class.

2.3 Implementation Details

This approach was realized utilizing Python 3.12.1, relying on the Keras library. The Adam optimizer is the optimizer for the system, which is efficient in high-dimensional spaces. The optimizer is one of the core components in the training process of models for deep learning, responsible for updating the weights in the model based on the differences between the true labels and the predictions for the model. The optimizer, Adam, is chosen to compile this model. The beneficial effects of adaptive gradient algorithms and root mean square propagation optimization methods are combined in the Adam optimizer, enabling adaptive learning rate adjustments and effective handling of sparse gradients and non-stationary objectives. This optimizer has demonstrated excellent performance in various deep learning tasks, thus becoming preferred choice. The size of batch is 32. The count of epochs is 100. The validation split is set to 0.2. For validation, 20% of the initial training set is set aside. Throughout training, the performance is assessed utilizing the validation set, allowing for early stopping or other interventions if the model starts too overfit.

3 Results and Discussion

Within this study, a method with 2-dimensional CNNs was employed to detect brain tumors from a dataset containing 3060 images, each assigned a distinct label that shows if the brain is tumorous or not. The author tested two different stop approaches. The model A used an early stopping approach which monitored validation loss. If after several epochs (2 epochs in this study) the patient's validation loss does not improve, the training will be stopped early. The model B used Thresholding method. The training stops when the accuracy hits 99.5%. A comparative evaluation of the accuracy achieved by model A to model B is presented in Fig. 6 and Fig. 7, providing an analysis of the two models' performance in detecting brain tumors.

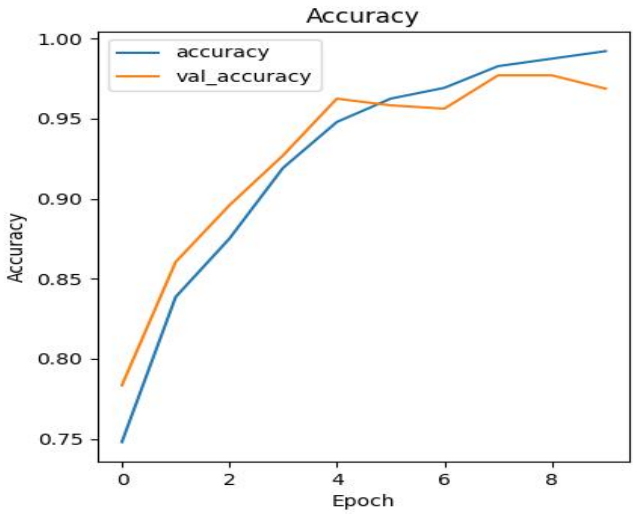

Fig. 6. The accuracy of the model with Early stop method (Photo/Picture credit: Original).

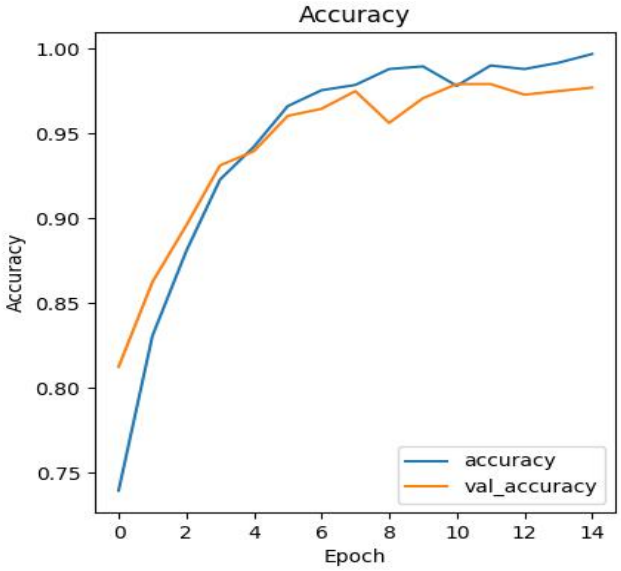

Fig. 7. The accuracy of the model with thresholding approach (Photo/Picture credit: Original).

From Fig. 6 and Fig. 7, both models' accuracy (blue line) and validation accuracy (orange line) are close. Also, both accuracies are on an increasing trend. This signifies that the model exhibits excellent performance on the training set, capable of precisely identifying the vast majority of samples. Model A's validation accuracy reaches 0.9771 after several epochs. Then, the validation loss starts to increase, which is the

signal of overfitting. Upon the consecutive increase of the validation loss for two epochs in succession, the early stop method stops training. In model B, the training was stopped early at the 15th epoch when the training accuracy reached 99.5%. The thresholding approach stops training for preventing overfitting and preserving computational resources.

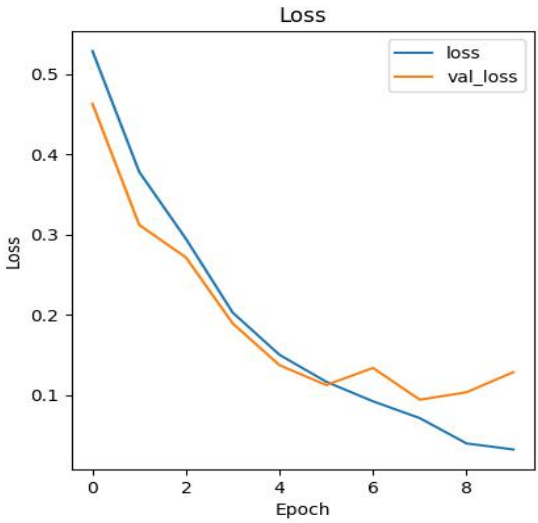

Fig. 8. The loss of the model with early stop method (Photo/Picture credit: Original).

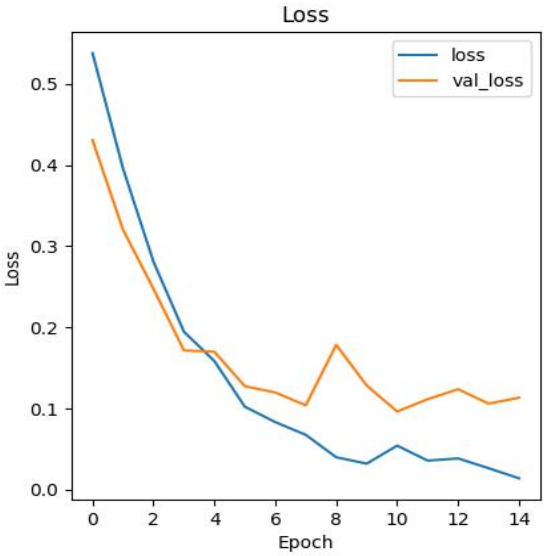

Fig. 9. The accuracy of the model with thresholding approach (Photo/Picture credit: Original).

The plot of the loss on the models is displayed in Fig. 8 and Fig. 9. For both models, In the early stages of training, both the validation loss and the training loss show a downward trend, which is a typical training occurrence showing how the model is gradually picking up new information and adjusting to the data. In Fig. 8, after approximately the 4th training epoch, the training loss of model A continued to decline but at a slower rate, while the validation loss suddenly increased at the 6th epoch and then stabilized afterward. This could suggest that the model began to show signs of mild overfitting after the 4th epoch, meaning that the model was performing increasingly better based on the training set of data, there was a little performance improvement on the validation data. In Fig. 9, as the quantity of training epochs rises (from Epoch 10 to higher Epochs), validation loss and loss both are decreasing, but the rate of decrease in validation loss seems to be slower than the loss. This might be an initial sign that the model may start to overfit to some extent. In conclusion, the model A and the model B exhibited good learning ability during the training process, achieving high accuracy on the sets for training and validation, and also achieved decent performance on the test set. Although there was some risk of overfitting, the model gradually stabilized its performance through subsequent training steps and ultimately achieved good accuracy on the test set.

4 Conclusion

This research introduces an approach to detect brain tumors using MRI. A two-dimensional CNN is proposed to analyze whether the brain is tumorous or non-tumorous. The input to the model is an MRI image, which is preprocessed by resizing and normalizing the pixel values. Three convolutional layers form the CNN architecture, each followed by a pooling layer for down sampling. After these layers, the features are flattened, subjected to dropout, and inserted in the dense layer. The second dropout layer and the second fully connected layer follow. Sigmoid activations are utilized in the output layer. The Adam optimizer and the binary cross-entropy loss function are utilized to train the model. The performance is primarily assessed using accuracy as the key metric, achieving a final accuracy score of 98% on the test dataset. Future research will focus on detecting the types of brain tumors, analyzing different tumor types to enhance diagnostic precision further.

References

1. Dhanwani, D.C., Bartere, M.M.: Survey on various techniques of brain tumor detection from MRI images. International Journal of Computational Engineering Research, 4(1), 24-26 (2014).
2. Amin, S.E., Megeed, M.A.: Brain tumor diagnosis systems based on artificial neural networks and segmentation using MRI. In 8th International Conference on Informatics and Systems, pp. MM-119. IEEE (2012).
3. Sharma, P., Diwakar, M., Choudhary, S.: Application of edge detection for brain tumor detection. International Journal of Computer Applications, 58(16), (2012).

4. Natarajan, P., Krishnan, N., Kenkre, N.S., Nancy, S., Singh, B.P.: Tumor detection using threshold operation in MRI brain images. In 2012 IEEE international conference on computational intelligence and computing research, pp. 1-4. IEEE (2012).
5. Amin, S.E., Megeed, M.A.: Brain tumor diagnosis systems based on artificial neural networks and segmentation using MRI. In 8th International Conference on Informatics and Systems, pp. MM-119. IEEE (2012).
6. Zhang, S., Xu, G.: A novel approach for brain tumor detection using MRI images. Journal of Biomedical Science and Engineering, 9(10), 44-52 (2016).
7. Iftekharuddin, K.M., Zheng, J., Islam, M.A., Ogg, R.J.: Fractal-based brain tumor detection in multimodal MRI. Applied Mathematics and Computation, 207(1), 23-41 (2009).
8. Bahadure, N.B., Ray, A.K., Thethi, H.P.: Image analysis for MRI based brain tumor detection and feature extraction using biologically inspired BWT and SVM. International journal of biomedical imaging, (2017).
9. Abdalla, H.E.M., Esmail, M.Y.: Brain tumor detection by using artificial neural network. In 2018 International conference on computer, control, electrical, and electronics engineering, pp. 1-6. IEEE (2018).
10. Saeedi, S., Rezayi, S., Keshavarz, H., Niakan, K.S.: MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. BMC Medical Informatics and Decision Making, 23(1), 16 (2023).
11. BR35H: Brain tumor detection 2020, https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection/data, last accessed 2020/8/20.