sciendo
ISSN 1335-8871
MEASUREMENT SCIENCE REVIEW
Journal homepage: https://content.sciendo.com

Multimodal Brain Tumor Classification using Capsule Convolution Neural Network with Differential Evolution Optimization Process

Senthilkumar C[1*], Eatedal Alabdulkreem[2], Nuha Alruwais[3], Suresh K[4]

[1]*Department of ECE, Sri Krishna College of Technology, Coimbatore, Tamilnadu, India, psk.uce@outlook.com*
[2]*Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh 11671, Saudi Arabia*
[3]*Department of Computer Science and Engineering, College of Applied Studies and Community Services, King Saud University, Saudi Arabia, P.O. Box 22459, Riyadh 11495, Saudi Arabia*
[4]*Department of IT, Sri Venkateswara College of Engineering Kancheepuram, Tamilnadu, India*

Abstract: Manual identification of brain tumors is error-prone and time-consuming for radiologists. Therefore, automation of the process is crucial. Although binary classification, such as distinguishing between malignant and benign tumors, is often straightforward, radiologists face significant challenges when classifying multimodal brain tumors. In this study, we present an automated approach that uses deep learning to classify brain tumor types using many types of data. The proposed method consists of three sequential phases. First, the median filter is used to eliminate any noise. For feature extraction in the second stage, linear contrast enhancement is used on VGG-16. The meningioma, glioma, and pituitary images are identified in the third stage of the brain tumor classification (BTC) process, which uses a modified capsule convolution neural network (CNN) design. The experimental results show that the brain tumor detection technique presented in this study successfully identifies the locations of tumor lesions. The results obtained were notably superior, with an accuracy of 98.34 %, a precision of 97.84 %, a recall of 05.34 %, and an F1-score of 94.56 %.

Keywords: Brain tumor, multimodal, MRI images, CT images, optimization, convolution neural network, segmentation.

1. INTRODUCTION

A brain tumor is one of the most alarming diseases worldwide [1]. In the United States, there are 24530 malignant tumors and 59040 non-malignant tumors in the central nervous system (CNS). In addition, 18600 people will die from this condition [2]. Brain tumor detection and diagnostic procedures often rely on the expertise of neuroexperts and radiologists who evaluate the images. This process is complex, time-consuming, and prone to human error [3]. Therefore, there is now a great need for computer-aided diagnostics. Moreover, in a low-income country like Bangladesh, the cost of advanced medical treatments is likely to be prohibitive for the majority of the population. Several companies are using similar strategies to improve medical diagnostics and predict diseases at an early stage [4]. Human detection and tracking of brain tumors is tedious and error-prone. Therefore, automated technologies are needed to replace traditional human procedures. Over the past decade, deep neural networks (DNNs) have consistently shown excellent performance as evidenced by recent multimodal challenges [5]. Convolution neural networks (CNNs) are a recognized deep learning method that provides excellent results on both two-dimensional (2D) and three-dimensional (3D) medical images. Transfer learning is often used when there is a lack of data and computational resources to save time [6]. In this approach, the information obtained from one task is used to solve similar tasks. There is a lack of advanced frameworks that seamlessly fuse multimodal data using neural networks to enable more accurate and robust brain tumor classification. Techniques for deep feature fusion, such as combining spatial and temporal features across modalities, are still under-researched.

2. RELATED WORKS

In this section, existing studies were categorized as deep learning and neural networks using pre-trained methods. In [7], multi feature scaling neural network (MultiFeNet) used a CNN to classify brain tumors. Multi-level feature scaling was used for feature extraction. An optimized CNN was developed in [8]. Experimental results on the BRATS 2018 dataset show improved performance (precision of 97.41 %, recall of 95.78 %, and dice score of 97.04 %) compared to

DOI: 10.2478/msr-2024-0031
*Corresponding author: psk.uce@outlook.com (Senthilkumar C)

previous frameworks. In [9], the Harris Hawks optimized convolution network (HHOCNN) was presented. Various features are extracted from the segmented domain and identified using a CNN. In [10], an effective method for brain tumor classification (BTC) using a hierarchical deep-learning neural network (HieDNN) classifier is presented. In [11], a new parallel DCNN architecture is proposed to extract both global and local features from the two parallel stages and overcome the problem of overfitting by using dropout regularization in conjunction with batch normalization. The CNN is used to detect brain tumors in X-ray images. The accuracy of MobileNetV2, InceptionV3, and VGG19 was 92 %, 91 %, and 88 %, respectively.

The main problem with using neural networks to categorize MRI and CT scans is the amount of imagesin the database. Also, MRI and CT scans are acquired in multiple planes, sousing all accessible planes could expand the collection and solve the problem of overfitting.

3. PROPOSED CLASSIFIER MODEL

First, the multimodal images such as MRI and CT are loaded and preprocessed with a median filter. Then feature extraction is performed with VGG-16, followed by dimension reduction. Finally, the tumor regions and their types are classified using the transformation matrix capsule convolution neural networks (TMCapsNet) as shown in Fig. 1:

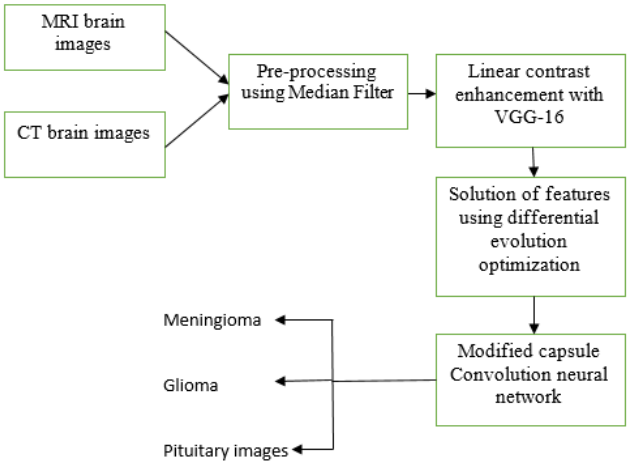

Fig. 1. BTC model.

A. *Preprocessing of multimodal data*

Preprocessing brain MRI and CT images with a median filter can help reduce noise while preserving important features in both modalities. During preprocessing useful information is extracted, and grouped as radiometric or geometric corrections. Median filtering replaces the center pixel of a sliding window that moves pixel by pixel across the entire image with the median value of the gray levels. Consider an $(A * B)$, where A indicates represents an MRI image and B indicates represents a CT image, with the image $I(p,q)\epsilon\{1,2,\dots A\} * \{1,2 \dots . B,\}$. If p,q commonly generally indicate the intensity or value of a pixel at coordinates (p,q) in an image, a 2-dimensional median filter is given as:

$$I(p,q) = median\,\{I(p+u,q+v)\} \tag{1}$$

where $u,v \in (-\left(\frac{x-1}{2}\right),\dots.\left(\frac{x-n}{2}\right))$ and $x(i,j)$ represents the pixel at (i,j). This preprocessing approach with median filters produces smoothed images while preserving the edges, which is an important feature.

B. *VGG-16 based feature extraction*

Consider the preprocessed images $\Delta = \{\tau_1, \tau_2, \dots \tau_N\}$, $\tau_N \in R^d$. Suppose $\mu(A,B)$ is a fused CT and MRI image of size $N \times M$, where $N = 256$ and $M = 256$ are rows and columns, respectively. τ_i is the average of clusters K_i, and the criterion function is:

$$S' = \sum_{i=1}^{K} \sum_{\tau \epsilon k_i} |\tau - \tau_i|^2 \tag{2}$$

where S' is the total sum of squared errors of all pixels. The symbol τ represents the input images, and τ_i denotes the number of clusters that are initialized. We used the generated images described by $\tau_1(x,y)$ to implement the edge-based texture histogram (HC) equalization, where $\tau_1(x,y) \in S'$. The gradient of the generated image $\tau_1(x,y)$ was calculated in the following way:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{3}$$

where G_x and G_y represent the x and y derivatives of $\tau_1(x,y)$, respectively. The edge map was then created by applying a threshold function in the following way:

$$EMp(x,y) = \begin{cases} 1 & G(x,y) < T \\ 0 & G(x,y) \geq T' \end{cases} \tag{4}$$

Based on this calculation, we selected the pixels with values above the threshold $T = 0.55$. These pixels were used to create the HC. Then, α and β were calculated, with α representing the lowest pixel value and β representing the highest pixel value. The gray levels were designated as HC if their values were between α and β. Finally, to get a better image, the transfer functions and the cumulative distribution function (CDF) were used. The VGG-16 model consists of 12 convolution layers, 15 ReLU activation layers, five max-pooling layers, three fully connected (FC) layers, and one classification Softmax layer. The input layer has dimensions of 224 x 224 x 3. The first convolution layer consists of 64 filters with dimensions of 3 x 3 x 3. A step size of 1 unit was used. In the subsequent convolution layer, the number of filters remained the same, but the dimensions of each filter were augmented to 3 x 3 x 64. Finally, the dense layer is used to extract features. The features can include intensity-based characteristics (such as voxel intensities and histogram statistics), texture characteristics (derived from gray-level co-occurrence matrices and local binary patterns), shape characteristics (such as volume and surface area), or other sophisticated characteristics (such as fractal dimensions and wavelet transform coefficients).

C. Dimensionality reduction

Following feature extraction, use differential evolution optimization to reduce the dimensionality of the features found. Differential evolution is a population-based optimization approach used to identify the best subset of characteristics that retains the highest degree of discriminative information. Vector discrepancies between two randomly selected individuals from the same population are calculated to perform the mutation. By combining the weighted difference of two vectors $F(X_{r2,g} + X_{r3,g})$ with the basic vector $X_{r1,g}$, its organization is disrupted to produce the mutant individual $V_{i,g}$. The following formula generates a mutant vector:

$$V_{i,g} = X_{r1,g} + F(X_{r2,g} + X_{r3,g}) \tag{5}$$

where, $r1, r2$ and $r3$ are randomly selected from the range $[1, N]$, where N is the total number of people in the population. The variables g and F respectively are the current generation and a constant mutation factor, respectively, which can assume values between 0 and 2. The target vector $X_{i,g}$ and the mutant vector $V_{i,g}$ are recombined to generate the trial vector $U_{i,g}$. The crossover factor indicates how high the probability is that components of the mutant vector will be included in the trial vector. Trial vector construction formula is:

$$U_{i,j,g} = \begin{cases} V_{i,j,g} & if\ rand_{i,j} \le CR \ \ or\ J = J_{rand} \\ X_{i,j,g} & if\ rand_{i,j} > CR \ \ or\ J \ne J_{rand} \end{cases} \tag{6}$$

where $i = 1,2, \ldots N$, where N is the size of the population. Similarly, let $j = 1,2, \ldots D$, where D is the dimension of a single vector. $rand_{i,j}$ is a randomly generated number in the range of $[0,1]$. J_{rand} is a randomly generated integer from the set $[1,2, \ldots D]$. Based on the fitness function formula, the trial vector $U_{i,g}$ is compared with the target vector $X_{i,g}$ and the one with the lowest function value is selected to generate $g + 1$,

$$X_{i,g+1} = \begin{cases} U_{i,j,g} & if\ f(U_{i,g}) < f(X_{i,g}) \\ X_{i,g} & otherwise \end{cases} \tag{7}$$

The performance of the selected feature subset can be evaluated by cross-validation or other suitable validation techniques based on the fitness function.

D. Capsule convolution neural network classifier

In this study, we implemented a neural network architecture called TMCapsNet, in which a matrix is created in each capsule.

According to this matrix criterion, the outputs are distributed to all parent capsules in the subsequent layer, but their coupling coefficients are not uniform. Each capsule tries to predict the output of its parent capsules. If this prediction matches with the actual output of the parent capsule, the coupling coefficient between these two capsules is increased. The prediction for the parent capsule j is obtained by considering u_i as the output of capsule i,

$$u_{j,i} = W_{ij} u_i \tag{8}$$

where the prediction vector of the output of the j^{th} capsule on a higher layer, computed from the capsule W_{ij} in the layer below, is denoted as $u_{j,i}$. The weighting matrix to be learned in the backward pass is represented as W_{ij}. The coupling coefficients C_{ij} are determined using a softmax function based on the degree of conformation between the capsules in the underlying layer and the parent capsules:

$$C_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \tag{9}$$

where b_{ij} is the logarithmic probability of whether capsule i should be connected to capsule j. At the beginning of the routing by agreement procedure, it is originally assigned the value of 0. The input vector for the parent capsule j is therefore calculated as follows:

$$s_j = \sum_i c_{ij}\, u_{j,i} \tag{10}$$

Finally, the following non-linear squashing function is used to prevent the output vectors of the capsules from exceeding the value one and to generate the final output of each capsule depending on its initial vector value, which is given in the following equation:

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{11}$$

where the input vector of capsule j is denoted as s_j, while the output is represented by v_j. In the routing process, the log probabilities need to be updated by considering the agreement between v_j and $u_{j,i}$. This can be done by using the fact that if the two vectors agree, their inner product will be larger. Therefore, the agreement $a_{i,j}$, which is used to update the log probability and the coupling coefficients, is calculated as follows:

$$a_{i,j} = v_j . u_{j,i} \tag{12}$$

The last layer's loss function l_k provides a high loss value for capsules with extended output instantiation parameters if the represented object does not exist. The loss function l_k is calculated as follows:

$$l_k = T_k \max(0, m^+ - \|v_k\|^2 + \tau(1 - T_k) \max(0, \|v_k\| - m^-)^{-2} \tag{13}$$

where T_k is equal to 1 if class k is present and 0 otherwise. The hyperparameters m^+, m^-, and τ must be defined prior to the learning process. Meningioma, glioma, and pituitary tumors are categorized as a consequence. With a 16-mini-batch size and data shuffling per iteration, an Adam optimizer trained the network. The early-stopping criterion, which determines when the network training process is terminated, is based on the completion of one epoch. Some of the limitations and possible directions are listed here: Capsule networks often require more computational resources due to their complex routing mechanisms and dynamic routing between capsules. This can lead to longer training times and higher energy consumption. CapsNets can be sensitive to the

initialization of the weights. Poor initialization can lead to slow convergence or complete failure to learn, making them less robust.

4. EXPERIMENTAL ANALYSIS

Experimental setup: For training the model, we used an NvidiaGeforce 2080 GPU with a storage capacity of 8 GB. The training set comprised 100 epochs and a batch size of 8. The processing system used in this study consists of a PC equipped with an Intel Core i3 CPU and 8 GB of RAM. The software used for the classification and improvement objectives is Python 3.9.7. Table 1 indicates modified capsule CNN configuration details [12].

Table 1. Modified capsule CNN.

Layer	Size
Kernel size	3×3
Activation function	ReLU
Input layer	28×28×1
Number of filters	32 to 256
Filter size	3×3

BRATS is an abbreviation for "The Multimodal Brain Tumor Image Segmentation Benchmark". Evaluating the effectiveness of different brain cancer image segmentation methods against the most advanced techniques is a challenging endeavor. The performance metric for disease classification is significantly influenced by the detection rate, which represents the proportion of contaminated pixels relative to the total number of pixels. Several researchers in the literature review have used several significant parameters, including accuracy, precision, recall and F1-score. Table 2 lists the evaluation parameters.

Table 2. Performance metrics.

Performance matrix	Formula
Accuracy	$ac = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$pr = \frac{nt}{nt + pf}$
Recall	$re = \frac{pt}{pt + nf}$
F1-score	$ppv = \frac{pt}{pt + pf}$

Note: $pt = true\ positive\ rate$ $nt = true\ negative\ rate$
$pf = false\ positive\ rate$ $nf = false\ negative\ rate$

Fig. 2 shows the evaluation of accuracy. The proposed TMCapsNet achieves the highest accuracy with 98.45 %, significantly outperforming MultiFeNet with 94.89 % and CNN with 90.46 %. This proves that TMCapsNet is the most reliable model in terms of overall correctness.

Fig. 3 shows the evaluation of precision. In this case, MultiFeNet has the highest precision with 96.56 %, followed by TMCapsNet with 95.78 %, and CNN with 94.98 %. While MultiFeNet has a slight edge in precision, indicating it is slightly better at reducing false positives, the proposed TMCapsNet still performs competitively.

Fig. 4 shows the evaluation of recall. The proposed TMCapsNet has the highest recall with 96.89 %, followed by MultiFeNet with 95.78 %, and CNN with 93.68 %. This shows that TMCapsNet is more effective in capturing all true positives, making it more sensitive in its predictions compared to the other models.

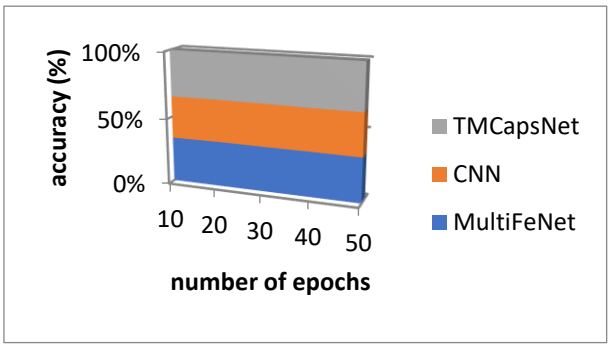

Fig. 2. Evaluation of accuracy.

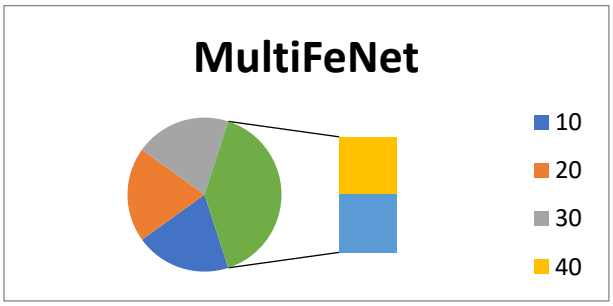

Fig. 3. Evaluation of precision.

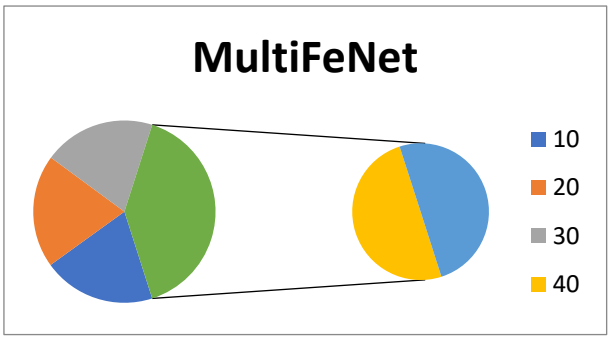

Fig. 4. Evaluation of recall.

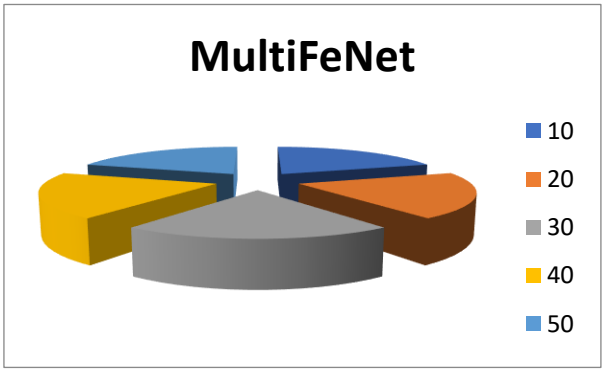

Fig. 5. Evaluation of F1-score.

Table 3. Overall comparative analysis.

Parameters [%]	MultiFeNet [16]	CNN [17]	TMCapsNet [proposed]
Accuracy	96.4	90.34	98.34
Precision	93.4	97.41	97.84
Recall	95.7	93.67	95.34
F1-score	96.2	95.78	94.56

Fig. 5 shows the analysis of the F1_score. The existing MultiFeNet and the proposed TMCapsNet both achieve an F1-score of 96.56 %, which is higher than CNN's 94.98 %. Table 3 indicates overall comparative analysis between existing and proposed methods.

5. CONCLUSION

The proposed classification method for brain tumor detection achieves a higher level of accuracy. The new technology outperforms the current techniques in the detection and categorization of brain tumors using MRI and CT. It is also more is visually appealing and provides better results. Future work will focus on the integration of other images such as functional magnetic resonance imaging (fMRI), positron emission tomography (PET) or diffusion tensor imaging (DTI), which can provide complementary information on tumor morphology, metabolism, and functional connectivity. In addition, the development of methods to visualize and interpret model predictions, highlight relevant image features, and provide context-aware explanations could improve the transparency and reliability of automated tumor detection systems.

ACKNOWLEDGMENTS

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R161), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSPD2024R608), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] Miller, K. D., Ostrom, Q. T., Kruchko, C., Patil, N., Tihan, T., Cioffi, G., Fuchs, H. E., Waite, K. A., Jemal, A., Siegel, R. L., Barnholtz-Sloan, J. S. (2021). Brain and other central nervous system tumor statistics, 2021. *CA: A Cancer Journal For Clinicians*, 71 (5), 381-406. https://doi.org/10.3322/caac.21693

[2] Amin, J., Sharif, M., Raza, M., Saba, T., Anjum, M. A. (2019). Brain tumor detection using statistical and machine learning method. *Computer Methods and Programs in Biomedicine*, 177 (3), 69-79. https://doi.org/10.1016/j.cmpb.2019.05.015

[3] Ibrahim, I. M., Abdulazeez, A. M. (2021). The role of machine learning algorithms for diagnosing diseases. *Journal of Application Science and Technology Trends*, 2 (1), 10-19. https://doi.org/10.38094/jastt20179

[4] Jacily Jemila, S., Jayasankar, T. (2011). An automated cancer recognition system for MRI images using neuro fuzzy logic. *International Journal of Computer Information Systems*, 2 (5), 18-22. ISSN 2229-5208.

[5] Rashid, M., Khan, M. A., Alhaisoni, M., Wang, S.-H., Naqvi, S. R., Rehman, A., Saba, T. (2020). A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection. *Sustainability*, 12 (12), 5037. https://doi.org/10.3390/su12125037

[6] Agrawal, T., Choudhary, P., Shankar, A., Singh, P., Diwakar, M. (2024). MultiFeNet: Multi-scale feature scaling in deep neural network for the brain tumour classification in MRI images. *International Journal of Imaging Systems and Technology*, 34 (1), e22956. https://doi.org/10.1002/ima.22956

[7] Ranjbarzadeh, R., Zarbakhsh, P., Caputo, A., Tirkolaee, E. B., Bendechache, M. (2024). Brain tumor segmentation based on optimized convolutional neural network and improved chimp optimization algorithm. *Computers in Biology and Medicine*, 168, 107723. https://doi.org/10.1016/j.compbiomed.2023.107723

[8] Kurdi, S. Z., Ali, M. H., Jaber, M. M., Saba, T., Rehman, A., Damaševičius, R. (2023). Brain tumor classification using meta-heuristic optimized convolutional neural networks. *Journal of Personalized Medicine*, 13 (2), 181. https://doi.org/10.3390/jpm13020181

[9] Shajin, F. H., Salini, P., Rajesh, P., Nagoji Rao, V. K. (2023). Efficient framework for brain tumour classification using hierarchical deep learning neural network classifier. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 11 (3), 750-757. https://doi.org/10.1080/21681163.2022.2111719

[10] Rahman, T., Islam, M. S. (2023). MRI brain tumor detection and classification using parallel deep convolutional neural networks. *Measurement: Sensors*, 26, 100694. https://doi.org/10.1016/j.measen.2023.100694

[11] Tazin, T., Sarker, S., Gupta, P., Ayaz, F. I., Islam, S., Khan, M. M., Bourouis, S., Idris, S. A., Alshazly, H. (2023). Retracted: A robust and novel approach for brain tumor classification using convolutional neural network. *Computational Intelligence and Neuroscience*, 2023, 9760861. https://doi.org/10.1155/2023/9760861

Received May 30, 2024
Accepted October 10, 2024