Received: 23 January 2024 | Revised: 18 March 2024 | Accepted: 29 March 2024 | Published online: 1 April 2024

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2024, Vol. 00(00) 1–15
DOI: 10.47852/bonviewJDSIS42022514

BON VIEW PUBLISHING

CT-γ-Net: A Hybrid Model Based on Convolutional Encoder–Decoder and Transformer Encoder for Brain Tumor Localization

Punam Bedi[1], Ningyao Ningshen[1], Surbhi Rani[1,*], Pushkar Gole[1] and Veenu Bhasin[2]

[1]*Department of Computer Science, University of Delhi, India*

[2]*PGDAV College, University of Delhi, India*

Abstract: Brain tumor is a life-threatening disease, and its early diagnosis can save human life. Computer-aided brain tumor segmentation and localization in magnetic resonance imaging (MRI) images have emerged as pivotal approaches for expediting the disease diagnosis process. In the past few decades, various researchers combined the strengths of convolutional networks and transformer to perform brain tumor segmentation. However, these models require a large number of trainable weights parameters, and there is still scope for performance improvement in them. To bridge these research gaps, this paper proposes a novel hybrid model named "CT-γ-Net" for effective and efficient brain tumor localization. The proposed CT-γ-Net model follows an encoder-decoder structure in which the convolutional encoder (CE) and transformer encoder (TE) are used for encoding, whereas the convolutional decoder (CD) is utilized for decoding the combined output of CE and TE to generate the segmentation masks. In CE and CD components of the CT-γ-Net model, conventional convolutional layers are replaced by depth-wise separable convolutional layers, as these layers significantly reduce trainable weights parameters. The proposed model achieves 95.5% MeanIoU, 94.82% Dice score, and 99.24% pixel accuracy on a publicly available dataset named the Cancer Imaging Archive. These experimental results demonstrate that the CT-γ-Net model outperformed other state-of-the-art research works, despite using roughly 28% fewer trainable weights parameters. Hence, the proposed model's lightweight nature and its high performance make it a suitable candidate for deployment on mobile devices, facilitating the precise localization of brain tumor regions in MRI images.

Keywords: brain tumor segmentation, transformer, convolutional encoder–Decoder, deep learning, disease diagnosis using artificial intelligence

1. Introduction

A brain tumor is a mass of lumps in the brain caused by abnormal brain tissue that develops rapidly. According to the American Association of Neurological Surgeons, more than 150 different types of brain tumor have been documented, which are either benign or malignant. In the survey conducted by central nervous system (CNS) in 2020, tumor occurrence in India lies between 5 and 10 per 100,000 population; moreover, there has been a rapid increase in the number of tumor patient [1]. Therefore, concerning the growing number of brain tumor patients and its threat to life, it is important to get diagnosed and undergo treatment in the early stage to improve the chances of successful treatment and prolong life expectancy.

While diagnosing brain tumors, medical experts use imaging tests such as magnetic resonance imaging (MRI) to identify the suspected tumor region and its probable chance of spreading to other CNS [2]. Diagnosing a brain tumor is laborious and requires careful examination, which can lead to failure of timely diagnosis before complications arise. Therefore, numerous researchers have developed various artificial intelligence models to support professional medical practitioners and aid less experienced doctors in diagnosing brain tumors accurately and efficiently [3]. Initially, researchers tried to utilize machine learning (ML) algorithms to diagnose brain tumors from MRI scans [4, 5]. However, ML techniques have two major shortcomings: first, they require manual features extraction, which is a laborious task, and second, they are unable to take advantage of modern graphics processing units (GPUs), as they are not implemented in that manner. Therefore, researchers have used deep learning (DL) techniques for segmenting brain tumors, as these techniques conquer the aforementioned limitations of ML models [6–10]. DL methods comprise multiple layers which can automatically extract various important features from raw data. Convolutional neural network (CNN) is a DL model that is extensively utilized to address many computer vision tasks like image classification, object detection, etc. [11]. They can automatically learn hierarchical features from images, which makes them well-suited for capturing spatial patterns from image data. Though CNN can perform image classification tasks, they are not inherently designed for image segmentation. This is due to the absence of deconvolutional layers or transpose convolutional layers to upsample the dimensions of feature maps equivalent to the

*Corresponding author: Surbhi Rani, Department of Computer Science, University of Delhi, India. Email: srani@cs.du.ac.in

dimensions of the input images, which is necessary for any image segmentation model to generate segmentation masks. Therefore, CNN is modified into a fully convolutional network (FCN) for image segmentation tasks. The architectures of CNN and FCN are different, as CNN follows a sequence of convolutional, pooling, and dense layers to extract features; on the other hand, FCN can map back the learned features to their original spatial dimensions, and this allows it to generate segmentation masks where each pixel corresponds to a class label.

In recent years, transformer-based models have gained much popularity as they achieved high performance over different CNN models in image classification tasks such as plant disease detection [12] and brain tumor diagnosis. This outperforming nature of the transformer-based models can be argued on the fact that they can obtain global and local representation from shallow layers, and these extracted features are similar to the features extracted by CNN in deeper layers [13]. Also, skip connections in the transformer encoder (TE) are even more influential than a popular CNN architecture named ResNet, which increases the performance of transformer-based models. Hence, various researchers have exploited the aforementioned advantages of transformer-based models and merged them with existing FCN models to build a state-of-the-art (SOTA) system for segmenting brain tumors from MRI scans [14, 15]. Though the existing research works achieved better results with faster computation than SOTA models, these models require a large number of trainable weights parameters to achieve high performance, which is computationally expensive. Moreover, there is still a scope for performance improvement to perform brain tumor segmentation more accurately. To bridge these research gaps, a novel hybrid model named "CT-γ-Net" based on convolutional encoder–decoder and TE has been proposed in this research work for brain tumor segmentation or localization. In the proposed model, depth-wise separable convolutional layers are used in CE and CD instead of conventional convolutional layers to reduce the trainable weights parameters by a significant factor. The depth-wise separable convolutional layers decrease weight parameters by performing convolution operations in two stages, namely, the filtering stage and the combination stage. In the filtering stage, a depth-wise convolution operation is applied in which each kernel independently processes individual channels of the input feature map. In the combination stage, pointwise convolution operation is employed to combine the output from depth-wise convolution operation that helps to create new feature maps by linearly concatenating the information across different channels. Hence, in this way, the number of trainable weights parameters is decreased by utilizing the depth-wise separable convolutional layer in place of conventional convolutional layers, which reduces the computational cost during the training and testing phase of the model.

In the CT-γ-Net model, CD is used for decoding; however, an alternative decoding approach could involve utilizing the modified transformer decoder (MTD) which is a modification of transformer decoder (TD). Therefore, two alternate architectures named "CE+TE→CD+MTD" and "CE+TE→MTD" have also been built to analyze the ability of MTD in segmenting the tumor region from MRI images. In CE+TE→CD+MTD, both the CD and MTD are utilized for segmenting the brain tumor from MRI images, and the output generated from CD and MTD is combined to get the final segmentation mask. Meanwhile, in CE+TE→MTD, the segmentation operation is exclusively performed by MTD. The performances of CT-γ-Net, CE+TE→CD+MTD, and CE+TE→MTD architectures have been compared in this research work, and it is found that the CT-γ-Net model outperformed the other two models. Thus, combined with low trainable weights parameters and high performance, the CT-γ-Net model can be utilized in the real world for performing effective and efficient brain tumor segmentation from MRI images, which can lead to faster brain tumor diagnostics.

The remainder of this paper is organized as follows. Section 2 describes several existing research works on brain tumor segmentation and localization. Section 3 describes the necessary background concepts that are required to build the proposed CT-γ-Net model. Section 4 describes the CT-γ-Net model. Moreover, in Section 5, other variants of the proposed models, namely, CE+TE→CD+MTD and CE+TE→MTD models which are the combination of CNN and transformer techniques, are discussed. The experimentations conducted in this research are outlined in Section 6. In Section 7, results obtained from the experimentation are given, and Section 8 discusses these results. Finally, the paper is concluded in Section 9.

2. Related Work

Recently, numerous researchers across the world have developed various SOTA ML and DL models for automatic segmentation or localization of brain tumor. This section discusses relevant research works present in the literature, and it is divided into three subsections. Section 2.1 discusses the research works based on ML techniques, Section 2.2 focuses on the DL-based research works, and in Section 2.3, transformer-based research works are discussed.

2.1. Brain tumor segmentation using ML

Initially, researchers leveraged the potential of ML techniques for segmenting brain tumors for MRI images. They have explored the potential of the Wiener filter [16] to enhance the quality and clarity of the input data through noise reduction [17, 18]. In Dehariya and Shukla [17], the output of the Wiener filter served as an input to the intelligent water drop (IWD) genetic algorithm, which defines an objective function to evaluate the quality of the feature subsets based on relevance for segmentation through an iterative process. Their experiment was conducted on 100 brain MRI images, and they obtained 97.97% pixel accuracy. Pixel accuracy is a metric that measures the percentage of pixels that are correctly classified by a model out of the total number of pixels in the image. From their experiment, it is evident that the Wiener filter can filter noise from the input brain MRI data to improve segmentation accuracy. Meanwhile in Zhang et al. [18], K-means++ [19] and Gaussian kernel-based fuzzy C-means algorithm [20] were combined to segment images. Initially, cluster centroids were initialized through K-means++ followed by clustering operation using Gaussian kernel-based fuzzy C-means algorithm to reduce the sensitivity of the clustering parameters, which in turn improved the robustness of their proposed model. Their model was trained with 100 pairs of FLuid-Attenuated Inversion Recovery (FLAIR) images from BRATS2012 collected from 20 patients and evaluated using data from patients 1, 2, and 3 individually and obtained a Dice score of 92.61%, 94%, and 89.78% for the patients, respectively. Jayanthi et al. [21] presented a novel model based on a fuzzy integrated active contour segmentation technique, which overcomes the limitation of "active contour without edges" [22] in segmenting images with weak boundaries. They evaluated their model on BRATS2012 and BRATS2015 datasets and achieved an average Dice score of 81%.

In the abovementioned research works, despite their high performance in brain tumor segmentation, they require manual feature extraction which is time-consuming and laborious. Therefore,

to overcome this limitation, DL techniques are used by various researchers for automating brain tumor diagnosis, and in the next subsection, several DL-based research works are discussed in which the tumor regions are segmented or localized from brain MRI images.

2.2. Brain tumor segmentation using deep learning

In DL, FCN is considered de facto for image segmentation as it can automatically learn various spatial and temporal features of brain MRI images. Thus, various researchers utilized FCN for building the SOTA model to segment brain tumors from MRI images. Pereira et al. [23] conducted their experiment for brain tumor segmentation using the CNN technique. In their experiment, two models, namely, HGG-CNN and LGG-CNN were developed, which were trained and evaluated for HGG and LGG tumor grade separately. The HGG-CNN involves 11 layers, and LGG-CNN has 9 layers with a filter size of (3×3) in the convolutional layers. Moreover, ReLU activation functions were applied consistently throughout the layers in both models. Their evaluation with the BRATS2013 dataset yielded a Dice score of 88%, 83%, and 77% in segmenting complete, core, and enhance tumor regions, respectively. Sun et al. [24] proposed a novel DL-based framework to segment brain tumors from MRI images. Their proposed framework involved three different segmentation techniques, namely, Cascaded Anisotropic Convolutional Neural Network [25], DFKZ Net [26], and 3D-UNet [27]. Subsequently, a majority voting strategy was employed in an ensemble technique without weighted scheme. Their model achieved a mean Dice score of 71.71%, 87.62%, and 79.97% on enhancing, whole, and tumor core during testing with the BRATS2018 dataset. Daimary et al. [28] presented a hybrid architecture based on U-Net, SegNet, and ResNet models. Different combinations of these models were configured to find the best-performing one, and it was found that Seg-UNet, that is, a fusion of SegNet and U-Net, outperforms other combinations by scoring 73.4% MeanIoU on the BRATS dataset. In the architecture proposed by Balamurugan and Gnanamanoharan [29], they exploited a VGG-16 model for brain tumor segmentation and classification between glioma and meningioma tumors through a LuNet classifier. Their experimental dataset consisted of 173 total samples of which they achieved 99.7% accuracy.

Although FCN-based techniques are effective in learning local features, they cannot extract global features of the images effectively and efficiently due to small receptive field. Additionally, these techniques are unable to capture the long-range dependencies within the images. These drawbacks have been conquered by the transformer model, and the next subsection discusses the research works based on this model.

2.3. Brain tumor segmentation using transformer-based model

Several researchers have explored the use of transformer in brain tumor segmentation or localization to address the deficiency observed in FCN and CNN models, which generally arises due to its incapability to provide solutions for long-range dependencies. However, transformer-based architecture, which was originally designed for textual data, can capture long-range dependencies through the self-attention mechanism. As such, Wang et al. [30] proposed the TransBTS model, which utilized 3D convolutional layers and transformer. Their model follows a structure similar to U-Net; however, TE layers were added between the encoder and decoder of U-Net instead of the default bottleneck layer to allow capturing semantic correlation. Their model achieved Dice scores of 78.73%, 90.09%, and 81.73% on segmenting, enhancing, whole, and core tumors. Following the U-Net architecture, Hatamizadeh et al. [14] developed the Swin-UNET-Transformer (Swin-UNETR) model for brain tumor segmentation, which is based on Swin-UNet [31]. In this architecture, the decoder of Swin-UNet is modified by replacing it with a residual block consisting of two $3 \times 3 \times 3$ convolutional layers. Their model has a total of 61.98 M parameters. While testing on the BRATS2021 dataset, their model scored an average Dice score of 89.10%, 93.30%, 91.70%, and 91.30 on enhancing, whole, and core tumors. Following the suit of Swin Transformer, Jiang et al. [32] proposed SwinBTS with a similar architecture to Swin Transformer. However, the self-attention mechanism in the multi-head attention (MHA) is replaced with 3D convolutional layers. Their model achieved 83.21%, 84.75%, and 91.83% on enhancing, core, and whole tumor. Furthermore, Liang et al. [33] proposed an architecture named TransConver which utilized the convolution and transformer blocks to find local and global information, respectively. The skip connection with cross-attention fusion mechanism was used with an enhanced skip connection to improve their model's performance which reduce semantic discrepancies between encoder and decoder features. Their proposed model achieved 83.72% and 86.32% Dice scores on BRATS2018 and BRATS2019, respectively.

In the preceding discussion, it is evident that existing models demand a substantial number of trainable weights parameters for achieving favorable results. Additionally, models based on FCN face challenges in capturing long-range dependencies due to their reliance on local receptive fields. On the other hand, Transformer-based architectures showcase limitations in capturing short-term dependencies. Therefore, this paper proposes a novel hybrid model named "CT-γ-Net" for brain tumor segmentation, which is based on a convolutional encoder–decoder and TE, to bridge these research gaps. Moreover, the CT-γ-Net model employed depth-wise separable convolutional layers to decrease the number of trainable weights parameters. The next section of this paper explains the necessary concepts required to understand the proposed architecture.

3. Background

This section provides a comprehensive description of fundamental components derived from CNN and the transformer model to develop the proposed CT-γ-Net model. This section is further subdivided into two subsections. Subsection 3.1 explains CNN and its layers, that is, the convolutional layer and pooling layer. Subsection 3.2 describes the transformer model and its components, that is, TE and TD.

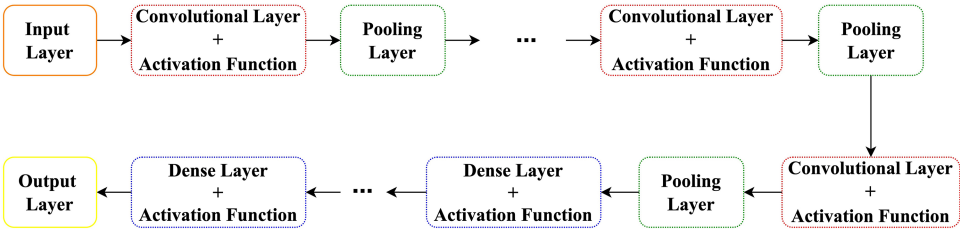
3.1. Convolutional neural network

CNN is a DL technique well-suited for handling data with a grid-like topology, such as time series and image data. The idea behind CNN is to exploit spatial and temporal information by using a specialized operation called convolution which operates on two real-valued functions, in contrast to simple matrix multiplication in conventional neural networks [11]. A typical representation of CNN is given in Figure 1 which comprises an input and an output layer and a set of convolutional, pooling, and dense layers.

3.1.1. Convolutional layer

A convolution operation (denoted by $*$) is a mathematical operation that is used in signal processing, image analysis, etc. It involves a binary operation between two real-valued functions (say $l(x)$ and $k(x)$).

Figure 1
Convolutional neural network (CNN)



The mathematical representation of convolution operation for discrete data is given in Equation (1).

$$z = (p * q)(x) = \sum_{v=-\infty}^{\infty} p(v).q(x-v) \tag{1}$$

where $z = (p * q)(x)$ represents the value of convolution of functions p and q at position x, v is an offset that determines the alignment of the kernel with the input, and $q(x-v)$ represents the function g shifted by v units. In the conventional convolution layer, convolution operations are applied to all input channels using multiple filters, which slide across the input feature maps to perform element-wise multiplication where the results are summed together to produce a single output. Thus, a significant number of multiplications are required in standard convolutional layers. Therefore, the depth-wise separable convolution layer has been used to address the drawbacks of the conventional convolution layer as it significantly reduces the number of multiplications by performing convolution operations in two stages, namely, the filtering stage and the combination stage. In the filtering stage, a depth-wise convolution operation is applied in which each kernel independently processes individual channels of the input feature map. These kernels are responsible for detecting local patterns and features within the data specific to that channel. Moreover, by processing each channel independently, the depth-wise convolution layer captures distinctive information within each feature map. Following the filtering stage, the combination stage involves merging the information from different channels using a point-wise convolution, typically with a 1×1 kernel. It combines filtered information across channels by performing a linear combination. This step helps create new feature representations that capture cross-channel correlations. Hence, combining the information from different channels enriches the overall representation, allowing the model to learn complex hierarchical features while maintaining computational efficiency. The visual representation of the standard convolution layer and depth-wise separable convolution layer is given in Figure 2.

To illustrate the number of multiplications involved in both standard and depth-wise separable convolution layer, consider an input feature map with dimensions of (W, H, C), where W denotes the width, H the height, and C the number of channels. A convolution operation with N filters of dimension $(K,\ K)$ where K is the filter size is applied to produce a feature map of the height and width of G. The total number of multiplications required in standard convolution layer (N_s^{mul}) and depth-wise separable convolution layer (N_{dsc}^{mul}) are given in Equations (2) and (3), respectively.

$$N_s^{mul} = C.G^2.K^2.N \tag{2}$$

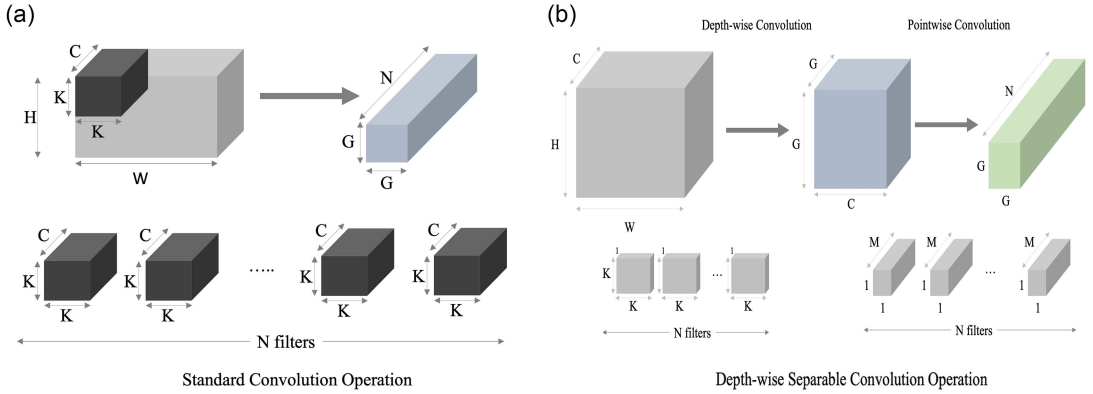
$$N_{dsc}^{mul} = C.G^2.(K^2 + N) \tag{3}$$

The ratio between the number of multiplications required in the conventional convolution layer and the depth-wise convolution layer can be computed by Equation (4).

$$\frac{N_{dsc}^{mul}}{N_s^{mul}} = \frac{C.G^2.(K^2+N)}{C.G^2.K^2.N} = \frac{K^2+N}{K^2 \times N} \tag{4}$$

By putting $K = 3$ and $N = 2014$ in Equation (4), we get $\frac{3^2 + 2014}{3^2 \times 2014} = \frac{2023}{18126}$; thus, we can conclude that the number of multiplications in depth-wise separable convolution layer is approximately one-ninth of the number of multiplications in standard convolution layer. Drawing from the given example, the depth-wise separable convolution layer demonstrates practical benefits that notably influence

Figure 2
A comparative example of standard convolution layer and depth-wise separable convolution layer



neural networks by lowering computational complexity and enhancing efficiency, particularly on devices with constrained resources.

3.1.2. Pooling layer

Pooling layer is a fundamental component within FCN. It is used to reduce the spatial dimension of feature maps by selecting the most dominant feature within a pooling region to create a more compact feature map. This makes the FCN model more robust against small shifts of position in the input feature maps. There are two types of pooling layers, namely, max pooling and average pooling. The maximum value is selected within each pooling region in max pooling, allowing it to select the most relevant information. Moreover, it provides some degree of translation invariance. Nevertheless, max pooling discards non-maximal values, leading to a loss of information. In tasks like brain tumor segmentation where subtle features or average intensity levels are essential, this loss may hinder the model's performance, whereas in average pooling, the average value is taken from each pooling window, thereby preserving more information. This can be advantageous when the overall intensity or distribution of values is important for the segmentation task. The mathematical equations for max and average pooling given a kernel size k × k are given in Equations (5) and (6), respectively.

$$\text{Max Pooling}(X)_{i,j} = \max_{m=1}^{k} \max_{n=1}^{k} X_{(i+m-1,\, j+n-1)} \tag{5}$$

$$\text{Average Pooling}(X)_{i,j} = \frac{1}{\text{k} \times \text{k}} \sum_{m=1}^{k} \sum_{n=1}^{k} X_{(i+m-1,\, j+n-1)} \tag{6}$$

where i, j represents the indices of the output feature map, and $X_{(i+m-1,\, j+n-1)}$ represents the value of the input feature map at position $(i+m-1,\ j+n-1)$. The pictorial representation of max and average pooling is given in Figure 3.

Apart from max pooling and average pooling layers, there exist additional pooling layers such as global pooling that considers the entire input feature map as a single pool and fractional pooling, which is an adaptive pooling operation introduced to address limitations in traditional max or average pooling, when the input dimensions do not neatly divide by the pooling size. Therefore, these pooling techniques could be considered in the architecture.

CNN is primarily used for classification tasks, and they are not inherently designed for image segmentation. Therefore, researchers have modified CNN into a convolutional encoder–decoder architecture for performing the image segmentation task efficiently and effectively. Hence, in this research work, convolutional encoder-decoder architecture has been utilized for brain tumor segmentation.

3.2. Transformer

Transformer is a DL architecture used for natural language processing. It utilized the concept of self-attention that computes the importance of different tokens in a sequence [34]. The encoder–decoder structure follows transformer architecture where the encoder extracts the features from the input sequence; meanwhile, the decoder takes the output of the encoder as well as the target sequence to predict the output sequence. The complete description of TE and TD is discussed in Subsections 3.2.1 and 3.2.2, respectively.

3.2.1. Transformer encoder

TE is one of the components of a transformer. However, to perform computer vision tasks, the TE requires embedded patches as input. Therefore, each input image with dimensions of $H \times W$ (where W denotes the width and H the height) is divided into $N_p = \frac{H.W}{P^2}$ nonoverlapping patches of size $P \times P$ to control the size of the resulting sequence of patches. Moreover, it ensures that the length of the sequence (the number of patches) is determined by the ratio of the image dimensions to the square of the patch size. Afterward, these patches are flattened and then linearly embedded using an embedding function $E : \mathbb{R}^{P \times P \times C} \rightarrow \mathbb{R}^{D}$ where D, which is the embedding dimension of the model, is chosen to strike a balance between maximizing the model's capacity and ensuring computational efficiency. Through patch embedding, it allows TE to capture local information within the patch. After patch embedding, the positional information of these patches is then added to the embedded patches. This ensures that the model can differentiate between patches based on their relative positions. Moreover, positional encoding is essential for preserving spatial information as well as conveying the spatial relationships and order of the patches.

The TE encompasses multiple stacked encoder blocks where each encoder block is comprised of MHA, layer normalization, and multi-layer perceptron (MLP) modules. Moreover, the TE encompasses two skip connections around MHA and MLP to mitigate the vanishing gradient problem as well as regain positional information. The visual representation of the TE has been depicted in Figure 4 and its components are described below.

Figure 3
Comparison between max and average pooling layers

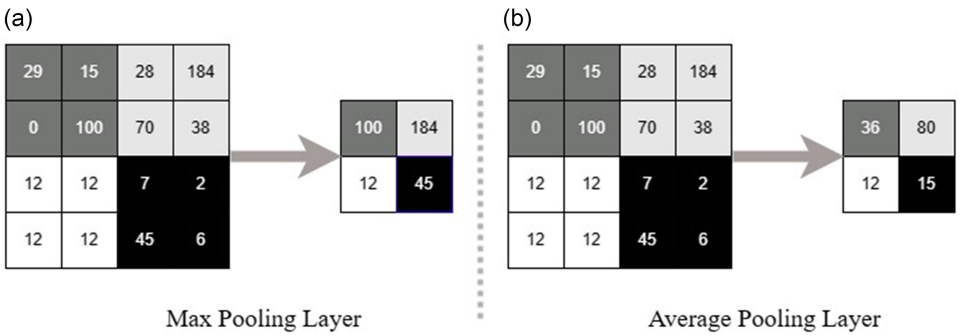
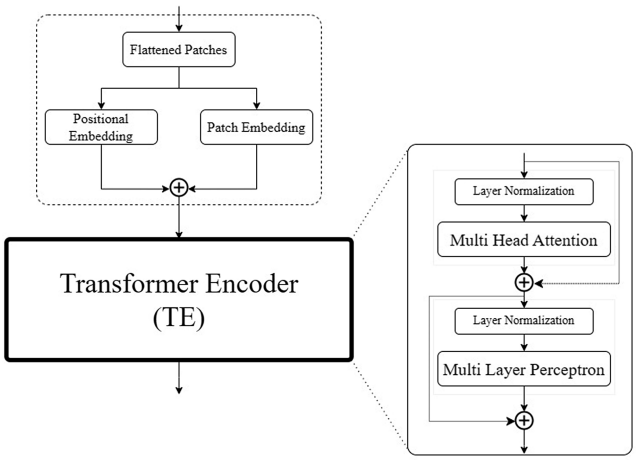


Figure 4
Transformer encoder (TE)



a. Multi-head attention

In MHA, n self-attention operations are employed simultaneously to learn different representations and global relationships among patches where n is the number of heads used in the MHA. Given $E \in \mathbb{R}^{(N_p \times D)}$, where E is the input embedding matrix, N_p denotes the number of patches, and D is the embedding dimension. The weight matrices $Q_W \in \mathbb{R}^{D \times q_D}$, $K_W \in \mathbb{R}^{D \times k_D}$, *and* $V_W \in \mathbb{R}^{D \times v_D}$ are trained where q_D, k_D, and v_D are the number of columns in Q_W, K_W, *and* V_W weight matrices. After weight initialization, Query (Q), Key (K), and Value (V) matrices have been calculated by multiplying input embedding matrix E with the weights matrices, that is, $Q = EQ_W$, $K = EK_W$, and $V = EV_W$. Thereafter, the self-attention score (Z) is computed by using Equation (7).

$$Z = softmax\left(\frac{QK^T}{\sqrt{k_D}}\right).V \tag{7}$$

Further, the outputs of n heads $Z_1, Z_2, \ldots . Z_n$ are concatenated and multiplied by a transformation matrix $O_W \in \mathbb{R}^{N_p \times nv_D}$ that allows the model to adapt and optimally fuses the distinct pieces of information from various attention heads to produce a complete representation. The formula for the concatenation operation is given in Equation (8).

$$Z_{concat} = concat(Z_1, Z_2, \ldots, Z_n) \times O_w \tag{8}$$

b. Layer normalization

Layer normalization was given by Ba et al. [35] to overcome the limitation of batch normalization, that is, batch size dependence. Layer normalization normalizes activations along the features rather than the Batch. This reduces the model's reliance on batch statistics which makes the activations more robust to the distributional differences between the training and testing sets. Initially, mean and variance are calculated using Equations (9) and (10), respectively. It is followed by the normalization of the feature map. A smoothing factor ε has been utilized to avoid zero division. The formula for normalizing the feature is given in Equation (11). Finally, two learnable parameters, that is, a scaling factor γ and a shifting factor ζ, are used to shift and scale as given in Equation (12).

$$\mu_{t,f} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} x_{tijf} \tag{9}$$

$$\sigma^2_{t,f} = \sum_{i=1}^{H}\sum_{j=1}^{W}\left(x_{tijf} - \mu_{t,f}\right) \tag{10}$$

$$\overline{x}_{(t,\ ijf)} = \frac{\left(x_{tijf} - \mu_{t,f}\right)}{\sqrt{\sigma^2_{t,f} + \varepsilon}} \tag{11}$$

$$y_t = \gamma\overline{x}_t + \beta \equiv LN_{\gamma,\zeta}(x_t) \tag{12}$$

where C, H, and W are the channel, height, and weight of the feature maps, respectively.

c. Multi-layer perceptron

In addition to the attention and Layer Normalization layers, the encoder of the transformer also comprises an MLP module. Two linear transformations and a nonlinear activation function are combined in this module to create nonlinearity and enable the model to understand intricate correlations between the input features. The mathematical representation of the MLP module is given in Equation (13).

$$\mathrm{MLP}(x) = \sigma(W_2\ (\sigma(xW_1 + b_1))\ + b_2) \tag{13}$$

3.2.2. Transformer decoder

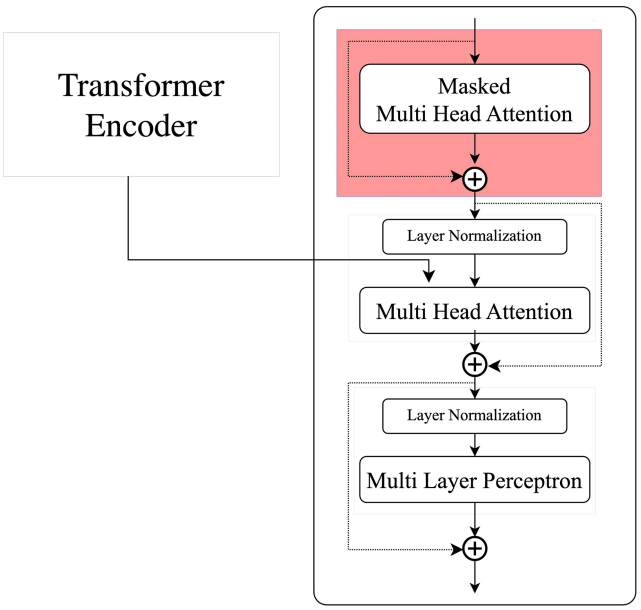
The TD is a key component of the transformer architecture, composed of multiple decoder blocks. Each decoder block comprises 2 × (MHA, layer normalization) followed by MLP modules. Taking the target sequence as input, the TD component processes it through the first MHA module. The second MHA module, along with the output of the first MHA module, also takes the output of the TE as input, capturing representations from both the input and target sequences. Additionally, three skip connections are incorporated around the first and second MHA modules, as well as the MLP module, to address the vanishing gradient problem and restore positional information. The architectural diagram of TD has been given in Figure 5.

Furthermore, CNN and transformer both are utilized for the proposed model of this paper. A detailed description of the proposed model is given in the next section.

4. Proposed Work

In this section, we present the CT-γ-Net model, a novel approach that combines convolution and transformer methodologies for effective brain tumor segmentation. This integration addresses limitations observed in depth-wise separable convolution layers, particularly in handling long-range dependencies, by leveraging transformer techniques and vice versa. Additionally, the distinctive architecture of the CT-γ-Net model enhances performance capabilities while maintaining a low number of trainable parameters, achieved through the utilization of depth-wise separable convolution layers.

Figure 5
Transformer decoder (TD)



The CT-γ-Net model combines convolutional encoder–decoder and TE modules to perform brain tumor segmentation or localization. The convolutional encoder–decoder modules can learn spatial as well as temporal features effectively and efficiently, while TE can capture global features and learn long-range dependencies from the brain MRI images. Therefore, by amalgamating the convolutional encoder–decoder and TE, the CT-γ-Net model has the potential to seamlessly learn spatial and temporal features along with the long-range dependencies inherent in brain MRI data. In the proposed model, depth-wise separable convolutional layers have been utilized in place of conventional convolutional layers to decrease the count of trainable weights parameters by a significant factor. The architectural diagram of the proposed CT-γ-Net model is given in Figure 6.

4.1. Components of the proposed model

CT-γ-Net model follows an encoder–decoder structure, and it comprises three components, namely, convolutional encoder (CE), TE, and convolutional decoder (CD). In the encoder of the proposed model, CE and TE are utilized; on the other hand, the decoder encompasses CD which decodes the summed output of CE and TE. These aforementioned components of the CT-γ-Net model are described below.

4.1.1. Convolution encoder

It encompasses five stacked ConvBlocks, and each of these blocks comprises two ConvBatchNormBlocks followed by a max pooling layer. Every ConvBatchNormBlock encompasses a depth-wise separable convolutional layer and batch normalization layer. The depth-wise separable convolutional layer in ConvBatchNormBlock significantly minimizes trainable weights parameters while preserving feature extraction capability, as discussed in Section 3.1. Additionally, a batch normalization layer is employed following each depth-wise separable convolutional layer to standardize the feature maps within a batch. Furthermore, CE employs a max pooling layer in every ConvBlock to reduce the dimensions of the feature by a factor of two. Given that the

Figure 6
Architectural diagram of the "CT-γ-Net" model

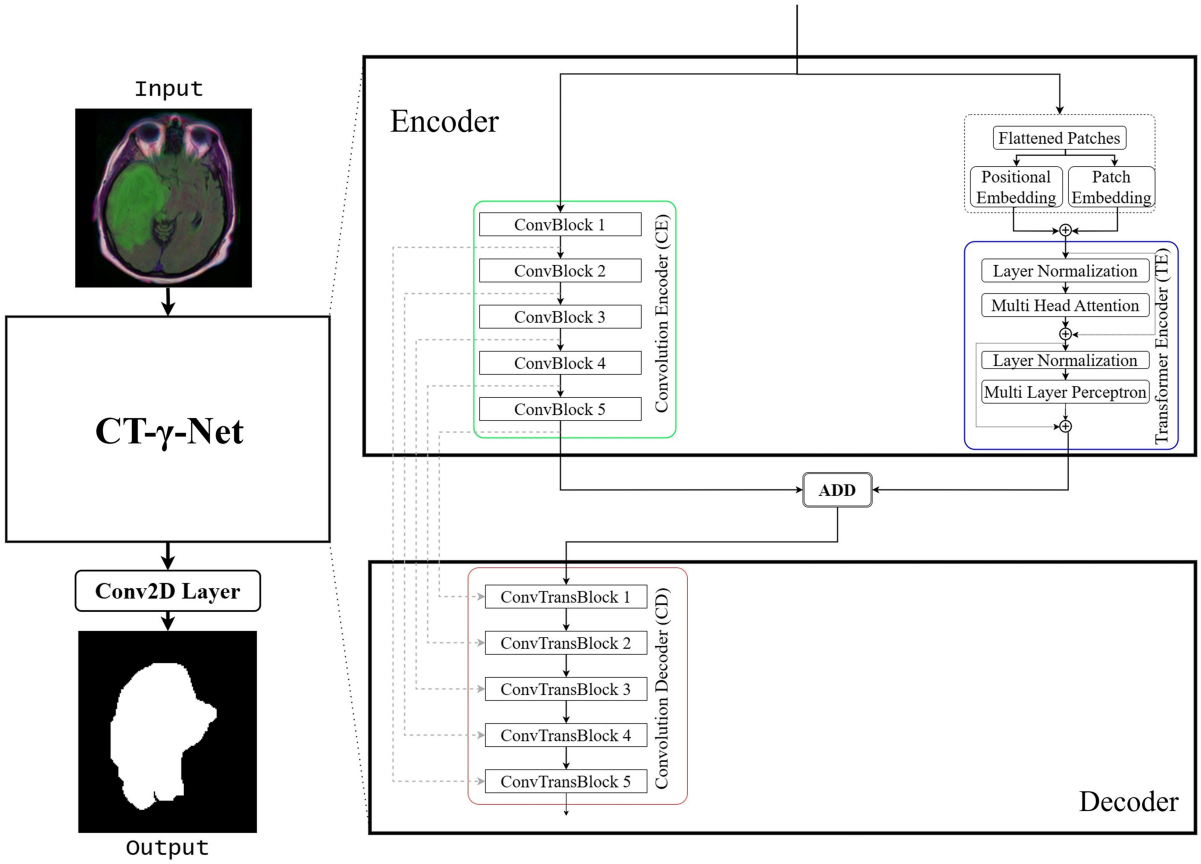
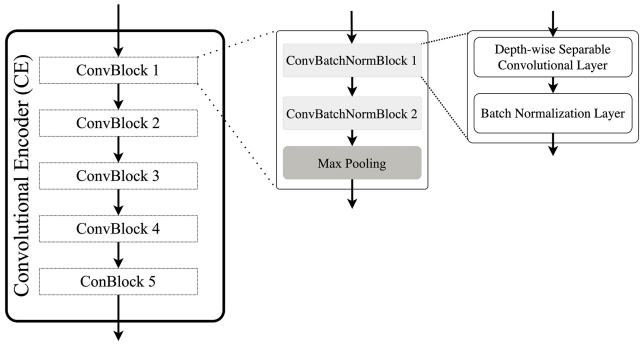


Figure 7
Convolutional encoder (CE)



dimension of the feature maps decreases by a factor of 2 after each ConvBlock, it is crucial to thoughtfully determine the number of ConvBlocks in the CE. It aims to avoid an excessively small feature map dimension, as such reduction could introduce challenges in the reconstruction process and impede the recovery of lost information. The diagram of CE is given in Figure 7.

4.1.2. Transformer encoder

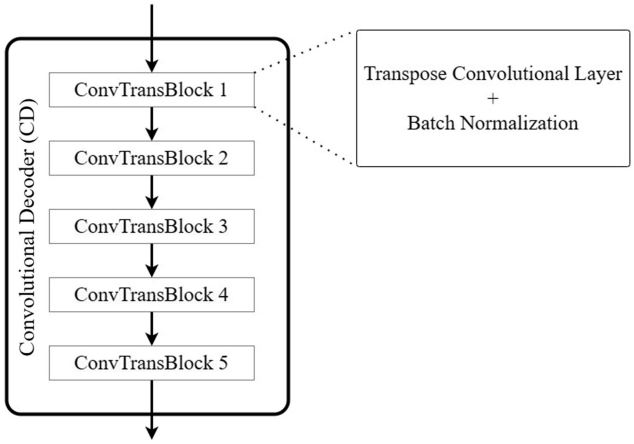
It comprises three modules, namely, layer normalization, MHA, and MLP. In TE, the values of the input embedded patches are normalized using layer normalization. This is followed by feature extraction in the MHA module through a self-attention mechanism. In the MHA, multiple self-attention is employed parallelly to learn different types of global features as well as long-range dependencies within the input patches. Subsequently, the normalized output of MHA is concatenated with the skip connection that is employed around MHA and is further passed to the MLP module that consists of two dense layers, which enables the model to enhance its capacity in capturing important information. Finally, the output of MLP is concatenated with the skip connection that surrounds MLP, and the feature maps are then normalized via layer normalization. The block diagram of TE is given in Figure 4.

To potentially enhance the feature extraction capability through the TE, one could consider incorporating multiple TE blocks. This approach enables the model to conduct a more comprehensive and hierarchical analysis of the input data, with each module processing information from the preceding one. This sequential processing contributes to the extraction of progressively abstract and global features. Additionally, a higher count of modules in the TE facilitates an extended and refined modeling of long-range dependencies. However, it is important to note that an increased number of TE modules raises the overall model complexity, and the associated trade-off may not necessarily justify the added complexity.

4.1.3. Convolution decoder

It consists of five ConvTransBlocks and each ConvTransBlock contains a transpose convolutional layer followed by a batch normalization layer. The transpose convolutional layer upsamples the feature maps, that is, increasing the spatial dimensions by a factor of two. Ultimately, the final ConvTransBlock ensures that the spatial dimensionality of the feature map matches that of the original input image. Moreover, the skip connections are employed between each corresponding ConvBlock and ConvTransBlock in the proposed model to prevent it from vanishing gradient problems. Furthermore, the batch normalization layer has been utilized for normalizing output feature maps of the transpose convolutional layer. It is important to observe that each ConvTransBlock is paired with a corresponding ConvBlock to facilitate the recovery of lost information through a skip connection. Consequently, introducing additional ConvTransBlocks could compromise the quality of the feature representation, as there will not be corresponding ConvBlocks for information recovery. While it is conceivable to direct the output of a single ConvBlock to ConvTransBlocks lacking corresponding ConvBlocks, the extent of performance improvement will remain a subject for further experimentation. The diagram of the CD is given in Figure 8.

Figure 8
Convolution decoder (CD)



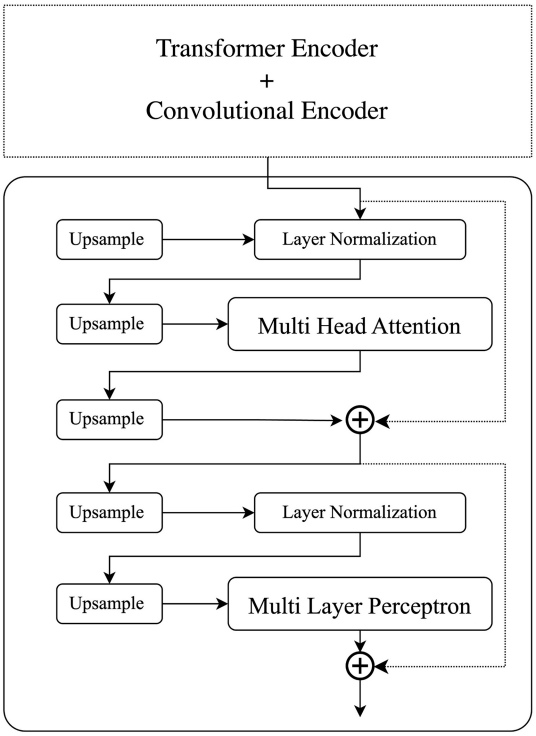
In this section, various components utilized in the development of the proposed CT-γ-Net model have been described. The discussion of other model variants, namely, CE+TE→MTD and CE+TE→CD+MTD, is presented in the next section.

5. Other Variants Based on the Combination of CNN and Transformer

As discussed in the previous section, the proposed CT-γ-Net model exploits CD for decoding the combined output of CE and TE. However, an alternative decoding approach could involve utilizing the TD to decode the combined output of CE and TE. Nevertheless, the TD of the transformer model utilized in Vaswani et al. [34] is not suitable for image segmentation; therefore, the TD is modified for image segmentation in this research work, and it is referred to as MTD throughout the paper. The block diagram of MTD is shown in Figure 9. The modification involves the removal of the masked MHA module from TD, which is highlighted in Figure 5. Furthermore, in the modified design, the input from the encoder is directed through layer normalization located above the MHA module. In addition to these changes, upsample layers are added before each module of TD to facilitate the dimensionality expansion of the feature maps by a factor of two.

To assess the decoding capability of MTD compared with CD and to substantiate the selection of CT-γ-Net as the proposed model, two alternative architectures, namely, "CE+TE→CD+MTD" and "CE +TE→MTD," were designed and developed in this research work. The block diagrams of these architectures are given in Figures 10 and 11, respectively. In CE+TE→CD+MTD represented in Figure 10, the encoding operation on input brain MRI images is performed

Figure 9
Modified transformer decoder (MTD)



simultaneously by CE and TE, which is subsequently combined to produce a unified encoded feature map. In the subsequent decoding phase, CD and MTD decode the encoded feature maps simultaneously. The results undergo normalization through a Conv2D layer employing a sigmoid activation function to produce segmentation masks. Similar to CT-γ-Net, skip connections are integrated between the respective ConvBlock of CE and ConvTransBlock of CD, to address the vanishing gradient problem and restore positional information. In conjunction, an alternate model CE+TE→MTD represented in Figure 11 is also developed to experiment and further explore the efficacy of MTD as compared with CD. In this architecture, the CD component along with skip connections from CE is omitted from the CE+TE→CD+MTD, leaving only MTD for decoding, as depicted in Figure 11. The sole output of MTD is utilized to generate segmentation masks.

6. Experimental Study

This section seeks to offer a comprehensive understanding of the experimental study conducted for this research work that aims to achieve the best-performing combination of CE, TE, CD, and MTD, that is, CT-γ-Net, CE+TE→MTD, or CE+TE→CD+MTD. This section is organized into three subsections. Subsection 6.1 describes the Cancer Imaging Archive (TCIA) dataset, and Subsection 6.2 provides details of various experimentations done in this research work. In Subsection 6.3, some evaluation metrics are discussed which are used to evaluate CT-γ-Net, CE +TE→MTD, and CE+TE→CD+MTD architectures.

Figure 10
Architectural diagram of CE+TE→CD+MTD

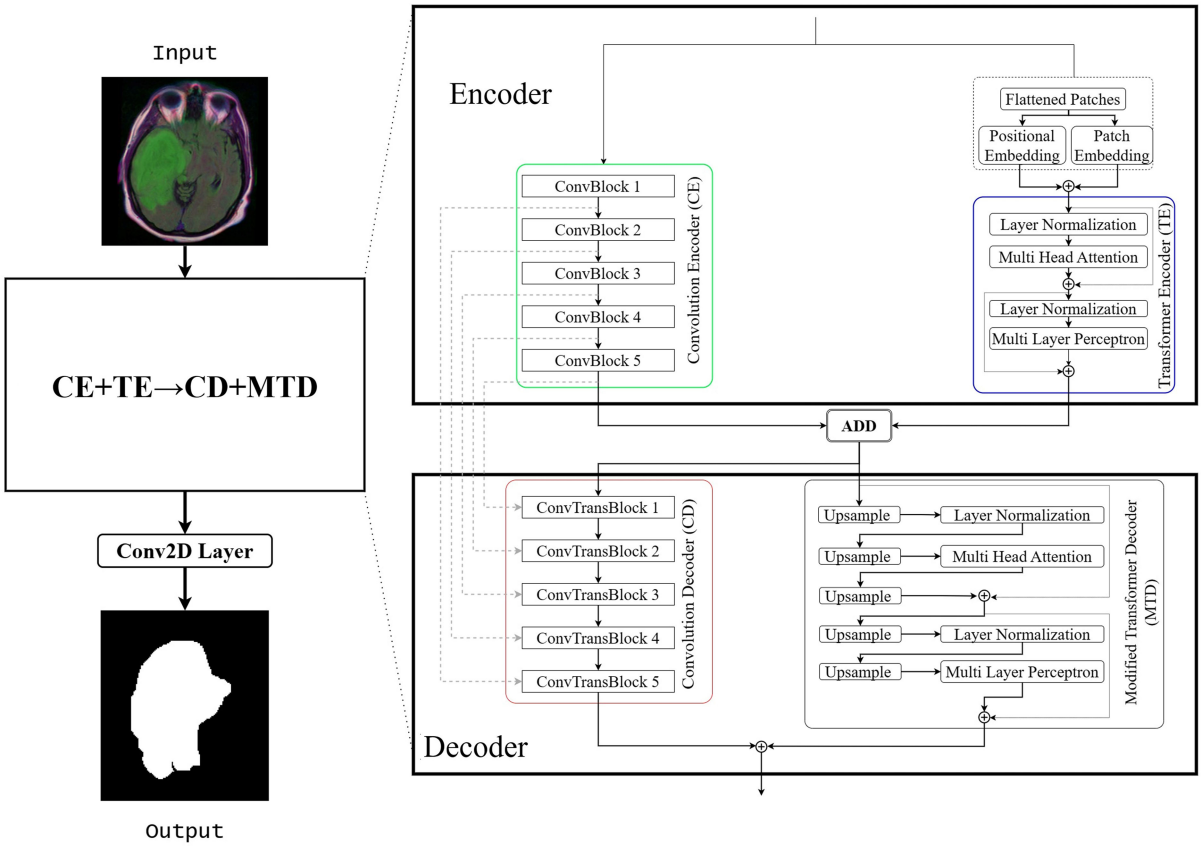
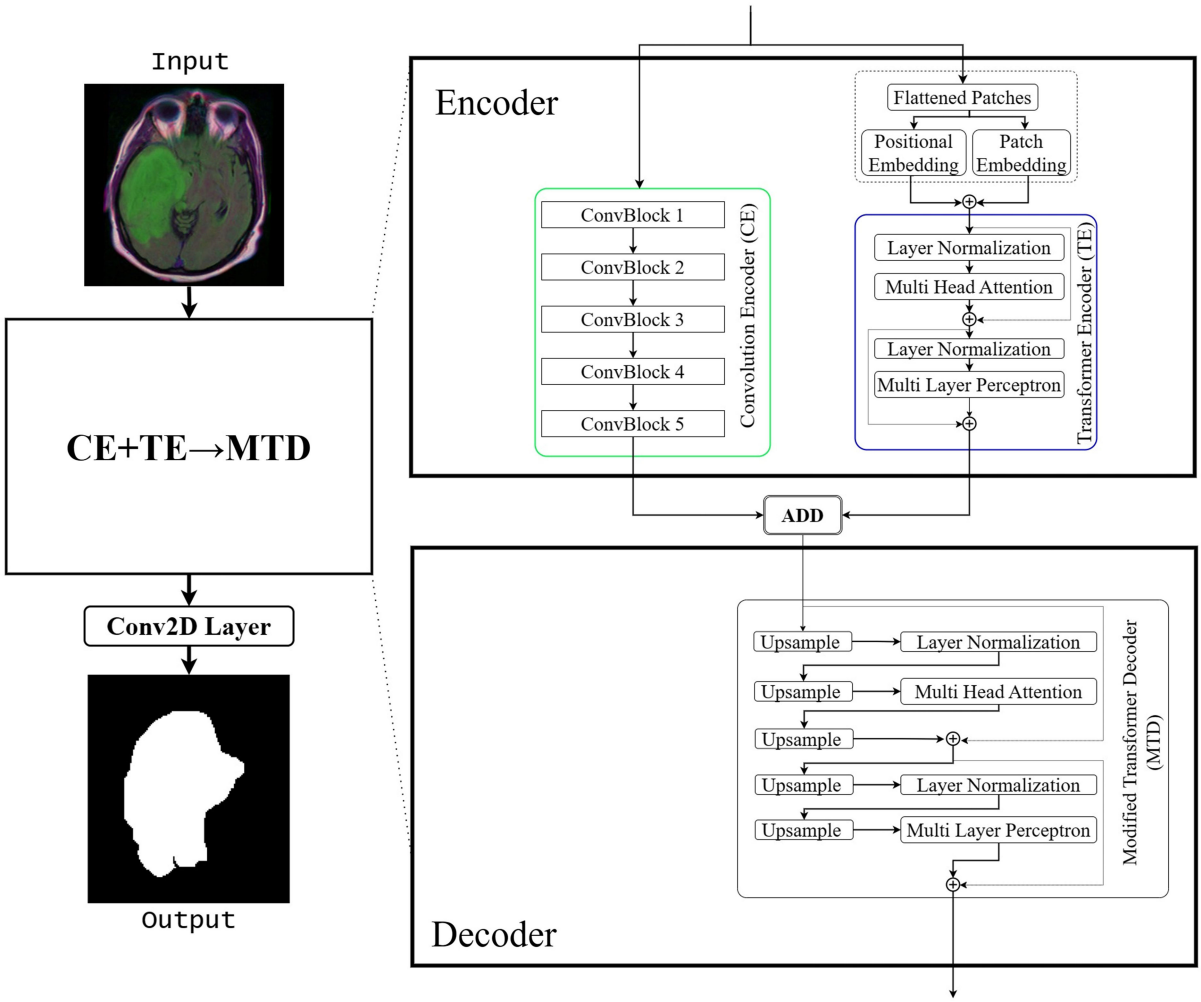


Figure 11
Architectural diagram of CE+TE→MTD

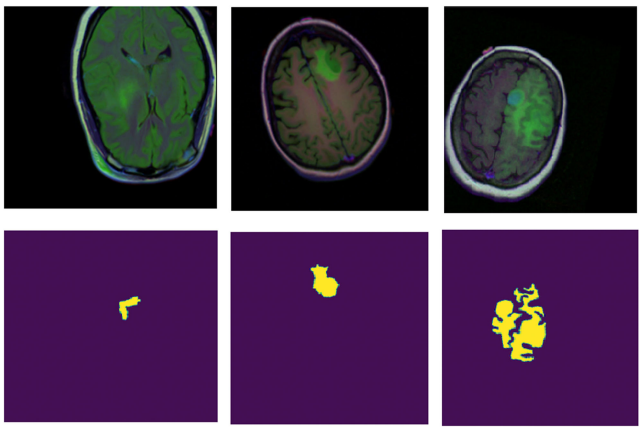


6.1. Dataset description

The TCIA is a benchmark dataset widely used to evaluate any ML or DL-based models for brain tumor segmentation. This dataset is publicly available in the Kaggle data science community. It comprises 2149 brain MRI images along with their corresponding annotated mask images. The images were taken from 110 patients, who were diagnosed with brain tumor disease. From each patient, 26 samples of brain MRI images were recorded that fall under different stages of the tumor. The MRI images are of dimensions (256 * 256 * 3), denoting a height and width of 256 pixels each, with 3 channels. In contrast, the mask images possess dimensions (256 * 256, 1), indicating grayscale images. It is noteworthy that the TCIA dataset solely comprises native volumes, posing a limitation in capturing comprehensive information and achieving robustness to imaging variability.

During preprocessing, the TCIA dataset underwent a random partition, with 70% for training and 30% for testing. Furthermore, during each epoch, 30% of the training data is randomly chosen for validation purposes. Additionally, to increase the number of samples, various data augmentation techniques such as 20^o degree right rotation, ± 2 % horizontal and vertical shift, and horizontal and vertical flip are performed. Moreover, the pixel value was normalized by dividing with $1/255$ to scale the pixel values to a range between 0 and 1 to prevent numerical instability and improve convergence. The visual illustration of some samples of the TCIA dataset is given in Figure 12.

Figure 12
Augmented sample images of the TCIA dataset with their corresponding annotated mask



6.2. Experimental configuration

The research experiments were conducted on Kaggle, a data science community platform, using Python 3.10.9 as the programming language. Nevertheless, these experiments can also be executed using alternative programming languages. Additionally, TensorFlow 2.12 library is used for model implementation. Furthermore, the Adam optimizer is used to minimize the loss computed by the Jaccard loss function. The mathematical formula for the Jaccard loss is depicted in Equation (14). Furthermore, to accelerate the training process, GPU NVIDIA Tesla T4 is used. The training of all the models has been done for 500 epochs with a batch size of 32. Additionally, an early stopping technique is applied, meaning that if the validation MeanIoU of the model shows no improvement for 60 consecutive epochs, the training of the model will be halted.

$$\text{Jaccard Loss} = 1 - \sum_{i=1}^{M} \frac{|\mathrm{A_i} \cap \mathrm{B_i}|}{|\mathrm{A_i} \cup \mathrm{B_i}|} \tag{14}$$

6.3. Evaluation metrics

To evaluate the performance of CT-γ-Net, CE+TE→CD+MTD, CE+TE→MTD architectures, the three most commonly used evaluation metrics, namely, mean intersection over union (MeanIoU), Dice score, and pixel accuracy, are used. Nevertheless, it is crucial to highlight that both MeanIoU and Dice score exhibit sensitivity to class imbalance, showing a bias toward the dominant class. Additionally, MeanIoU and Dice score may encounter difficulties in accurately assessing the performance of the model during class overlap especially when the model struggles to precisely describe the boundaries between adjacent classes. To address these concerns, pixel accuracy is also introduced to measure the percentage of correctly classified pixels over the total number of pixels in the image. The mathematical formulas for the aforementioned metrics are given in Equations (15)–(17). In these equations, M is the number of instances, and A_i and B_i represent the predicted and ground truth segmentation masks, respectively. Moreover, TP = true positive, TN = true negative, FN = false negative, and FP = false positive.

$$\text{MeanIoU}(\mathrm{A}, \mathrm{B}) = \frac{1}{M} \sum_{i=1}^{M} \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \tag{15}$$

$$\text{Dice score}(\mathrm{A}, \mathrm{B}) = \frac{1}{M} \sum_{i=1}^{M} \frac{2 \times |A_i \cap B_i|}{|A_i| + |B_i|} \tag{16}$$

$$\text{Pixel accuracy} = \frac{(\mathrm{TP} + \mathrm{TN})}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{17}$$

7. Results

This section provides the result of the experimentation done in this paper. The performance of CT-γ-Net, CE+TE→CD+MTD, and CE+TE→MTD architectures has been evaluated by using MeanIoU, Dice score, and pixel accuracy on the test subset of the TCIA dataset. The performances of these architectures are compared to find the best-performing architecture. Thereafter, the best-selected architecture is compared with the existing SOTA research works.

The MeanIoU, Dice score, and pixel accuracy of CT-γ-Net, CE+TE→MTD, and CE+TE→CD+MTD architectures on the test subset have been depicted by a line chart given in Figures 13–15, respectively. It can be observed from these figures that the architecture CE+TE→MTD achieved 69.07% MeanIoU and 55.48% Dice score. Meanwhile, CT-γ-Net outperforms both CE+TE→MTD and CE+TE→CD+MTD by achieving 95.5% MeanIoU and 94.82% Dice score. However, CE+TE→MTD achieved the highest pixel accuracy compared with CE+TE→CD+MTD and CT-γ-Net by scoring 99.38% indicating high overall correctness of the pixel-wise predictions.

Furthermore, the total number of trainable weights parameters employed in CE+TE→CD+MTD, CE+TE→MTD, and CT-γ-Net architectures is compared in Table 1. It can be observed from Table 1 that CE+TE→MTD and CT-γ-Net require 346011 and 342431 trainable weights parameters, which are approximately 26.7% and

Figure 13
MeanIoU for CE+TE→MTD, CE+TE→CD+MTD, and CT-γ-Net

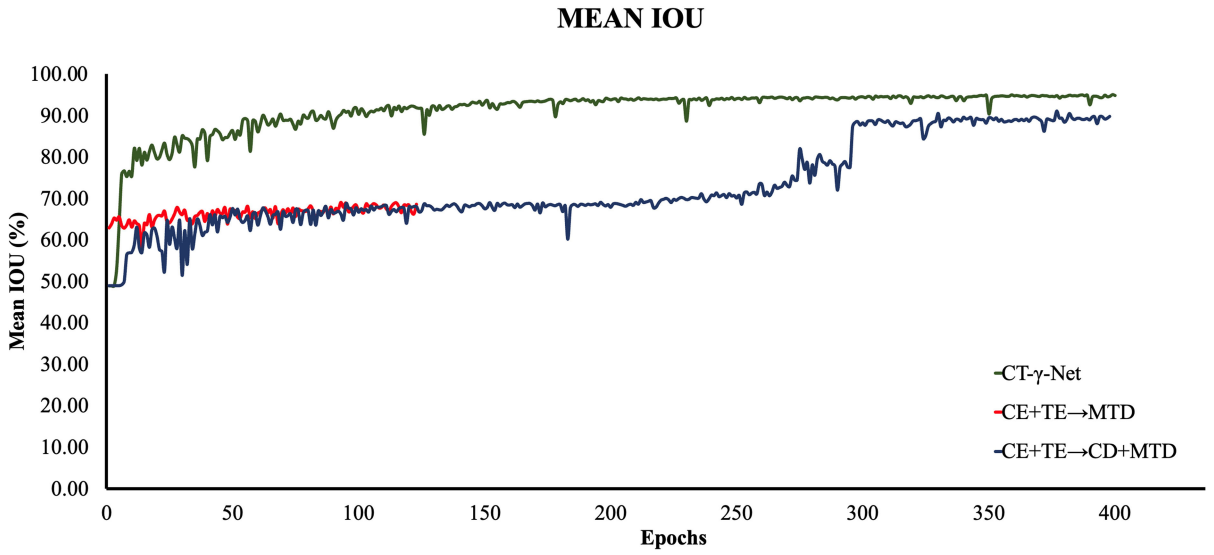


Figure 14
Dice score for CE+TE→MTD, CE+TE→CD+MTD, and CT-γ-Net

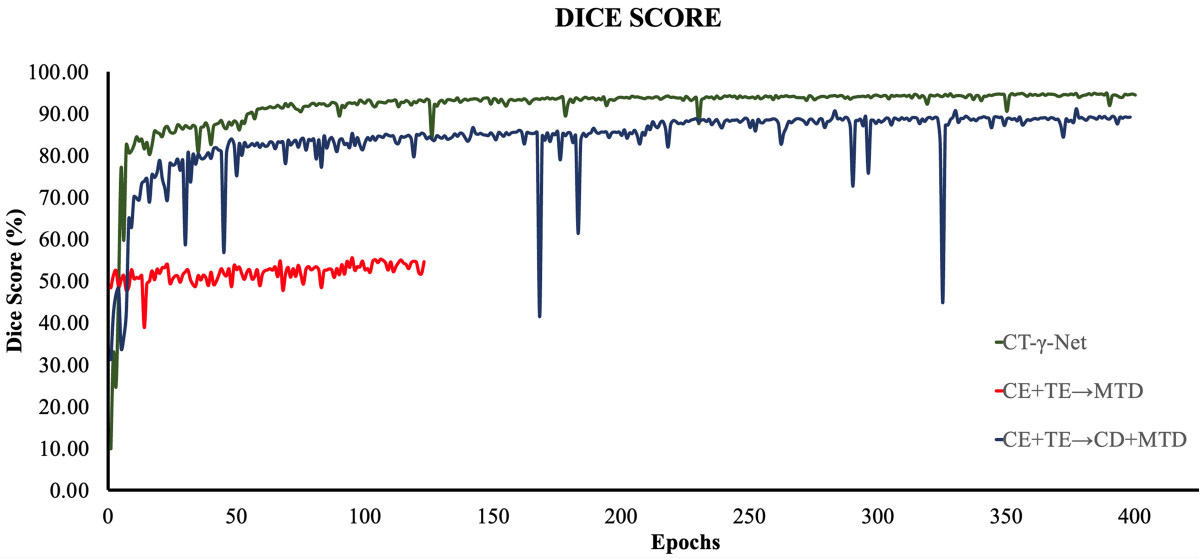
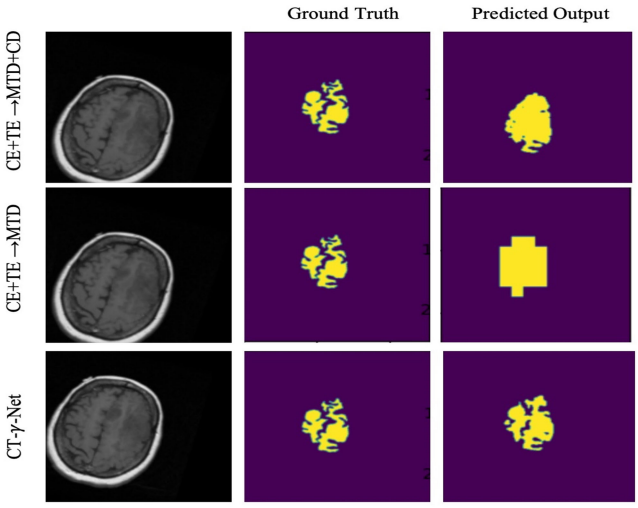


Figure 15
Pixel accuracy for CE+TE→MTD, CE+TE→CD+MTD, and CT-γ-Net



27.4% less than CE+TE→CD+MTD architecture. In addition, output segmentation masks of the aforementioned architectures have been given in Figure 16. It can be visualized from Figure 16 that

Table 1
Parameters of the architectures CE+TE→CD+MTD, CE+TE→MTD, and CT-γ-Net

Model	Parameters
CE+TE→CD+MTD	471,986
CE+TE→MTD	346,011
CT-γ-Net	342,431

the predicted segmentation mask of CE+TE→MTD architecture significantly differs from the ground truth segmentation masks, as the MeanIoU and Dice score of this architecture are minimum among other architectures, but pixel accuracy was the highest in CE+TE→MTD, which indicates the high overall correctness in pixel-wise prediction predictions. However, it can also be seen from Figure 16 that the predicted segmentation mask of CT-γ-Net architecture is similar to the ground truth segmentation mask.

Furthermore, a performance comparison was conducted between the CT-γ-Net model and existing SOTA models as given in Table 2. It reveals that, despite using fewer trainable weights parameters, our proposed model demonstrates superior performance, achieving the highest MeanIoU and Dice scores compared to other works found in the literature.

The findings above demonstrate that the CT-γ-Net model shows significant efficacy in segmenting brain tumors. This can be credited

Figure 16
Segmentation results CE+TE→CD+MTD, CE+TE→MTD, and CT-γ-Net

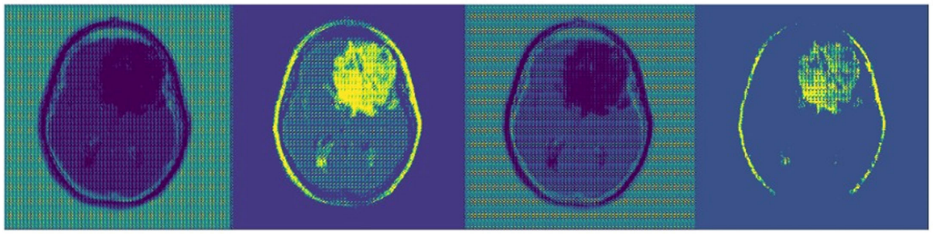


Table 2
Comparison of existing models with the CT-γ-Net model

Research work	MeanIoU	Dice score	Pixel acc.	Params
Kadry et al. [36]	–	90.36%	–	–
Gagan et al. [37]	80.06%	–	–	483,197
Wu et al. [38]	89.9%	82.3%	–	–
CT-γ-Net	**95.50%**	**94.82%**	**99.24%**	**342,431**

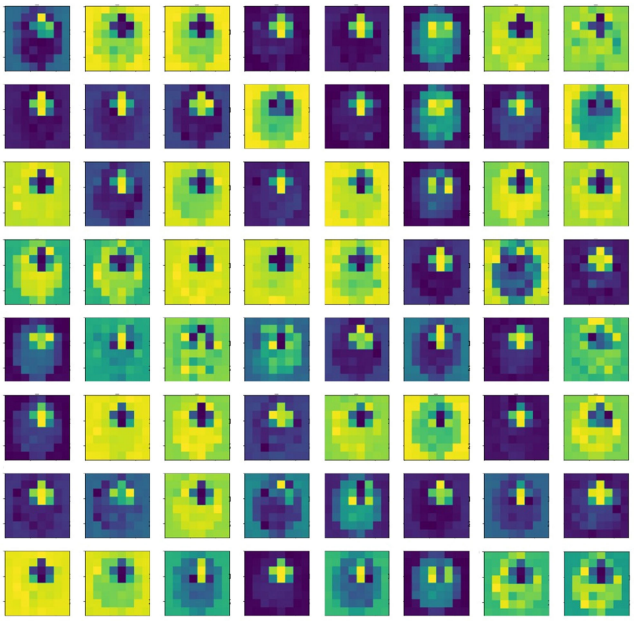
to the innovative architecture design, which integrates convolution layers and TE to effectively capture both local and global features from MRI images of the brain. Additionally, the utilization of depth-wise Separable convolution layers reduces the number of trainable weights parameters, enabling deployment on low-performing mobile devices without compromising performance in brain tumor segmentation.

8. Discussion

This research work is aimed at building a SOTA machine intelligence model to segment tumor regions effectively and efficiently from brain MRI images. Various researchers have utilized ML [36] and DL [39] techniques for brain tumor segmentation and localization. However, their models require manual feature extraction, or they are computationally expensive, that is, necessitate a large number of trainable weights parameters. Additionally, there is still scope for performance improvement in the existing research works [28, 38]. Therefore, to tackle these limitations of the existing research work, this paper proposes a novel hybrid model named "CT-γ-Net" based on FCN and transformer for segmenting brain tumor with a lesser number of trainable weights parameters. To reduce the trainable weights parameters, depth-wise separable convolutional layers have been utilized in place of conventional convolutional layers in the CT-γ-Net model. The proposed model follows the encoder–decoder structure, and it encompasses CE, TE, and CD components based on FCN and transformer. CE and TE are part of encoder, whereas CD has been utilized for decoding the combined output of CE and TE for generating the segmentation masks. Although TD can also be used for decoding the summed output obtained from CE and TE, it is not suitable for image segmentation as transformer architecture as it is designed for natural language processing. Therefore, TD is modified into MTD (described in Section 5), and to verify its potential as a decoder, two more architectures named CE+TE→CD+MTD and CE +TE→MTD have been designed and developed in this research work. Thereafter, the performance of these architectures has assessed TCIA dataset, which is publicly available with the help of three evaluation metrics, namely, MeanIoU, Dice score, and pixel accuracy.

After analyzing the results obtained from experimentation, it is found that the CT-γ-Net model outperformed the other two architectures by achieving maximum MeanIoU, Dice score, and pixel accuracy despite using a significantly lesser number of trainable weights parameters. However, it was also observed during experimentation that the performance of the CE+TE→MTD architecture which utilizes only MTD in the decoder is minimal. Therefore, to investigate the possible reason behind the subpar performance of this architecture, the output feature maps obtained from MTD and CD modules of CE+TE→MTD and CT-γ-Net models have been plotted in Figures 17, respectively. It has been observed that MTD's decoding process was unable to recover the potential information loss during encoding. Moreover, it is also found that the upsampling operations

Figure 17
Output feature maps of CD in CT-γ-Net model



in MTD might result in the loss of positional information and global feature representations. It can also be seen from these figures that the features obtained from CD are much richer as compared to the output of MTD. This can be argued by the fact that the skip connections have the potential to recover the loss of spatial information between CE and CD.

Conclusively, it can be stated that despite employing a considerably smaller number of trainable weights parameters, the CT-γ-Net model outperformed CE+TE→CD+MTD and CE+TE→MTD architectures and other SOTA research works. Hence, considering the impressive performance and lightweight design CT-γ-Net model, it can be implemented in real-world scenarios, making it suitable for deployment on a broader range of devices including devices with lower computational power. This deployment could offer valuable support to medical diagnosticians, facilitating faster diagnoses, and aiding less experienced medical practitioners in identifying brain tumors from MRI images. Furthermore, the application of this service could be extended to end users for self-service purposes.

In this research work, the proposed model experimented with the TCIA dataset; however, in future research work, the CT-γ-Net model can also be trained on different medical image datasets like BRAin Tumor Segmentation (BRATS), Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases (IQ-OTH/NCCD), etc. This will not only facilitate a more in-depth evaluation but also enhance the generalization capabilities of the proposed CT-γ-Net across various modalities in brain tumor datasets. Moreover, a wider range of evaluation metrics can be considered, which will contribute to a more thorough understanding of the CT-γ-Net model. Additionally, future endeavors can involve improving the segmentation performance in CE+TE→CD+MTD and CE+TE→MTD by devising a more efficient and effective design for MTD components. This should be coupled with addressing the substantial computational requirements due to a high number of trainable weights parameters in TE and MTD because of the use of the MHA mechanism for brain tumor segmentation.

9. Conclusion

A brain tumor is a cluster of masses in the brain resulting from the rapid growth of abnormal brain tissue. It is a life-threatening disease, and its early diagnosis is a crucial step in saving human life. However, manual segmentation or localization performed by medical experts requires valuable time. Therefore, many researchers have used various ML and DL techniques for segmenting brain tumor from MRI images. However, the models utilized in existing SOTA require manual features extraction or a large number of trainable weights parameters. Moreover, there is still scope for performance improvement in the existing research works. Therefore, to address these limitations, a novel hybrid model named "CT-γ-Net" was proposed in this research work for localizing or segmenting brain tumors from MRI images. The CT-γ-Net model utilized CE and TE to encode the input images, whereas CD was used to decode the combined output of CE and TE for generating segmentation masks. Moreover, TD could also be employed for decoding the summed output of CE and TE. However, as TD was originally designed for natural language processing; in this research work, it was modified to perform image segmentation and referred to as MTD in the paper. To validate the performance of MTD for decoding the summed output of CE and TE, two alternate architectures were built by utilizing MTD for decoding. These alternate architectures were referred to as "CE+TE→CD+MTD" and "CE+TE→MTD" in this paper. After analyzing the experimental results, it was concluded that CT-γ-Net outperforms the other two architectures by achieving 95.05% MeanIoU, 94.82% Dice score, and 99.24% pixel accuracy. Moreover, it was also found that the CT-γ-Net model outperforms the existing SOTA architectures presented in the literature despite using a significantly lesser number of trainable weights parameters. Hence, the proposed model can be deployed on various low-computational powered devices like mobile phones, Raspberry Pi, etc., to effectively and efficiently segment or localize the brain tumors from MRI images, due to its high performance and lightweight nature.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in [Kaggle] at https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation.

Author Contribution Statement

Punam Bedi: Conceptualization, Validation, Writing – review & editing, Supervision, Project administration. **Ningyao Ningshen:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Visualization. **Surbhi Rani:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Visualization. **Pushkar Gole:** Conceptualization, Validation, Writing – review & editing. **Veenu Bhasin:** Conceptualization, Validation, Writing – review & editing, Supervision.

References

[1] Insider Business. (2023). *Brain tumor cases rising "steadily" in India, 20% are children*. Retrieved from: https://www.businessinsider.in/science/health/news/brain-tumour-cases-rising-steadily-in-india-20-are-children-doctors/articleshow/100848073.cms

[2] Cancer.Net. (2023). *Brain tumor: Diagnosis*. Retrieved from: https://www.cancer.net/cancer-types/brain-tumor/diagnosis

[3] Coupet, M., Urruty, T., Leelanupab, T., Naudin, M., Bourdon, P., Maloigne, C. F., & Guillevin, R. (2022). A multi-sequences MRI deep framework study applied to glioma classification. *Multimedia Tools and Applications*, *81*(10), 13563–13591. https://doi.org/10.1007/S11042-022-12316-1

[4] Hassan, S., Hassan, A. A., Marshad, I., Al Hosain, M. A., Amin, M., Faisal, F., & Nishat, M. M. (2022). Comparative analysis of machine learning algorithms in detection of brain tumor. In *3rd International Conference on Big Data Analytics and Practices*, 31–36. https://doi.org/10.1109/IBDAP55587.2022.9907433

[5] Rinesh, S., Maheswari, K., Arthi, B., Sherubha, P., Vijay, A., Sridhar, S., . . . , & Waji, Y. A. (2022). Investigations on brain tumor classification using hybrid machine learning algorithms. *Journal of Healthcare Engineering*, *2022*. https://doi.org/10.1155/2022/2761847

[6] Bedi, P., Ningshen, N., Rani, S., & Gole, P. (2023). Explainable predictions for brain tumor diagnosis using inceptionV3 CNN architecture. In *6th International Conference on Innovative Computing and Communication*, 125–134. https://doi.org/10.1007/978-981-99-4071-4_11

[7] Corso, J. J., Sharon, E., Dube, S., El-Saden, S., Sinha, U., & Yuille, A. (2008). Efficient multilevel brain tumor segmentation with integrated Bayesian model classification. *IEEE Transactions on Medical Imaging*, *27*(5), 629–640. https://doi.org/10.1109/TMI.2007.912817

[8] Meier, R., Bauer, S., Slotboom, J., Wiest, R., & Reyes, M. (2014). Appearance-and context-sensitive features for brain tumor segmentation. In *Proceedings of MICCAI BRATS Challenge*. https://doi.org/10.13140/2.1.3766.7846

[9] Pei, L., Reza, S. M. S., Li, W., Davatzikos, C., & Iftekharuddin, K. M. (2017). Improved brain tumor segmentation by utilizing tumor growth model in longitudinal brain MRI. In *Medical Imaging 2017: Computer-Aided Diagnosis*, *10134*, 666–674. https://doi.org/10.1117/12.2254034

[10] Pinto, A., Pereira, S., Correia, H., Oliveira, J., Rasteiro, D. M. L. D., & Silva, C. A. (2015). Brain tumour segmentation based on extremely randomized forest with high-level features. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 3037–3040. https://doi.org/10.1109/EMBC.2015.7319032

[11] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. USA: MIT Press.

[12] Gole, P., Bedi, P., & Marwaha, S. (2023a). Automatic diagnosis of plant diseases via triple attention embedded vision transformer model. In *6th International Conference on Innovative Computing and Communication*, 879–889. https://doi.org/10.1007/978-981-99-4071-4_67

[13] Gole, P., Bedi, P., Marwaha, S., Haque, Md. A., & Deb, C. K. (2023b). TrIncNet: A lightweight vision transformer network for identification of plant diseases. *Frontiers in Plant Science*, *14*. https://doi.org/10.3389/fpls.2023.1221557

[14] Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H. R., & Xu, D. (2022). Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images. In *International*

MICCAI Brainlesion Workshop, 272–284. https://doi.org/10.1007/978-3-031-08999-2_22

[15] Wang, L., Li, R., Zhang, C., Fang, S., Duan, C., Meng, X., & Atkinson, P. M. (2022). UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *190*, 196–214. https://doi.org/10.1016/J.ISPRSJPRS.2022.06.008

[16] Mihailova, A., & Georgieva, V. (2016). Comparative analysis various filters for noise reduction in MRI abdominal images. *International Journal Information Technologies & Knowledge*, *10*(1), 47–66.

[17] Dehariya, A. K., & Shukla, P. (2021). Brain image segmentation to diagnose tumor by applying Wiener filter and intelligent water drop algorithm. *International Journal of Computer Theory and Engineering*, *13*(3), 84–90. https://doi.org/10.7763/IJCTE.2021.V13.1294

[18] Zhang, C., Shen, X., Cheng, H., & Qian, Q. (2019). Brain tumor segmentation based on hybrid clustering and morphological operations. *International Journal of Biomedical Imaging*, *2019*. https://doi.org/10.1155/2019/7305832

[19] Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 7*, 1027–1035.

[20] Ding, Y., & Fu, X. (2016). Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing*, *188*, 233–238. https://doi.org/10.1016/J.NEUCOM.2015.01.106

[21] Jayanthi, S., Ranganathan, H., & Palanivelan, M. (2019). Segmenting brain tumour regions with fuzzy integrated active contours. *IETE Journal of Research*, *68*(1), 514–525. https://doi.org/10.1080/03772063.2019.1615007

[22] Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, *10*(2), 266–277. https://doi.org/10.1109/83.902291

[23] Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Transactions on Medical Imaging*, *35*(5), 1240–1251. https://doi.org/10.1109/TMI.2016.2538465

[24] Sun, L., Zhang, S., Chen, H., & Luo, L. (2019). Brain tumor segmentation and survival prediction using multimodal MRI scans with deep learning. *Frontiers in Neuroscience*, *13*, 810. https://doi.org/10.3389/FNINS.2019.00810

[25] Wang, G., Li, W., Ourselin, S., & Vercauteren, T. (2017). Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop*, 178–190. https://doi.org/10.1007/978-3-319-75238-9_16

[26] Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., & Maier-Hein, K. H. (2018). Brain tumor segmentation and radiomics survival prediction: Contribution to the BRATS 2017 challenge. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017*, 287–297. https://doi.org/10.1007/978-3-319-75238-9_25

[27] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In Ourselin, S., Joskowicz, L., Sabuncu, M., Unal, G., Wells, W. (Eds.), *Lecture Notes in Computer Science* (pp. 424–432). Springer. https://doi.org/10.1007/978-3-319-46723-8_49

[28] Daimary, D., Bora, M. B., Amitab, K., & Kandar, D. (2020). Brain tumor segmentation from MRI images using hybrid convolutional neural networks. *Procedia Computer Science*, *167*, 2419–2428. https://doi.org/10.1016/J.PROCS.2020.03.295

[29] Balamurugan, T., & Gnanamanoharan, E. (2023). Brain tumor segmentation and classification using hybrid deep CNN with LuNet Classifier. *Neural Computing and Applications*, *35*(6), 4739–4753. https://doi.org/10.1007/S00521-022-07934-7

[30] Wang, W., Chen, C., Ding, M., Yu, H., Zha, S., & Li, J. (2021). TransBTS: Multimodal brain tumor segmentation using transformer. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 109–119.

[31] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). Swin-UNet: UNet-like pure transformer for medical image segmentation. In *European Conference on Computer Vision*, 205–218. https://doi.org/10.1007/978-3-031-25066-8_9

[32] Jiang, Y., Zhang, Y., Lin, X., Dong, J., Cheng, T., & Liang, J. (2022). SwinBTS: A method for 3d multimodal brain tumor segmentation using Swin transformer. *Brain Sciences*, *12*(6), 797. https://doi.org/10.3390/BRAINSCI12060797

[33] Liang, J., Yang, C., Zeng, M., & Wang, X. (2022). TransConver: Transformer and convolution parallel network for developing automatic brain tumor segmentation in MRI images. *Quantitative Imaging in Medicine and Surgery*, *12*(4), 2397–2415. https://doi.org/10.21037%2Fqims-21-919

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . , & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information processing Systems*, *30*, 6000–6010. https://doi.org/10.48550/arxiv.1706.03762

[35] Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv Preprint: 1607.06450*.

[36] Kadry, S., Rajinikanth, V., Raja, N. S. M., Jude Hemanth, D., Hannon, N. M. S., & Raj, A. N. J. (2021). Evaluation of brain tumor using brain MRI with modified-moth-flame algorithm and Kapur's thresholding: A study. *Evolutionary Intelligence*, *14*(2), 1053–1063. https://doi.org/10.1007/S12065-020-00539-W

[37] Gagan, K. R., Shlok, B., & Chary, Mr. V. R. (2022). MRI brain tumor segmentation using U-Net. *International Journal for Research in Applied Science and Engineering Technology*, *10*(6), 207–211. https://doi.org/10.22214/ijraset.2022.43774

[38] Wu, J., Fu, R., Fang, H., Zhang, Y., Yang, Y., Xiong, H., . . . , & Xu, Y. (2024). MedSegDiff: Medical image segmentation with diffusion probabilistic model. *Proceedings of Machine Learning Research*, *227*, 1623–1639.

[39] Sajid, S., Hussain, S., & Sarwar, A. (2019). Brain tumor detection and segmentation in MR images using deep learning. *Arabian Journal for Science and Engineering*, *44*(11), 9249–9261. https://doi.org/10.1007/S13369-019-03967-8

How to Cite: Bedi, P., Ningshen, N., Rani, S., Gole, P., & Bhasin, V. (2024). CT-γ-Net: A Hybrid Model Based on Convolutional Encoder–Decoder and Transformer Encoder for Brain Tumor Localization. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS42022514