IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 January 2023, accepted 11 March 2023, date of publication 16 March 2023, date of current version 22 March 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3257767*

RESEARCH ARTICLE

# CrossTransUnet: A New Computationally Inexpensive Tumor Segmentation Model for Brain MRI

**ANDRÉS ANAYA-ISAZA[1], LEONEL MERA-JIMÉNEZ[1,2], AND ALVARO FERNANDEZ-QUILEZ[2,3]**
[1]Indigo Technologies, Bogota 410010, Colombia
[2]SMIL, Department of Radiology, Stavanger University Hospital, 4068 Stavanger, Norway
[3]Department of Electrical Engineering and Computer Science, University of Stavanger, 4021 Stavanger, Norway

Corresponding author: Leonel Mera-Jiménez (leonel.mera@udea.edu.co)

This work was supported by Indigo Technologies Division Research (https://indigo.tech/).

**ABSTRACT** Brain tumors are usually fatal diseases with low life expectancies due to the organs they affect, even if the tumors are benign. Diagnosis and treatment of these tumors are challenging tasks, even for experienced physicians and experts, due to the heterogeneity of tumor cells. In recent years, advances in deep learning (DL) methods have been integrated to aid in the diagnosis, detection, and segmentation of brain neoplasms. However, segmentation is a computationally expensive process, typically based on convolutional neural networks (CNNs) in the UNet framework. While UNet has shown promising results, new models and developments can be incorporated into the conventional architecture to improve performance. In this research, we propose three new, computationally inexpensive, segmentation networks inspired by Transformers. These networks are designed in a 4-stage deep encoder-decoder structure and implement our new cross-attention model, along with separable convolution layers, to avoid the loss of dimensionality of the activation maps and reduce the computational cost of the models while maintaining high segmentation performance. The new attention model is integrated in different configurations by modifying the transition layers, encoder, and decoder blocks. The proposed networks are evaluated against the classical UNet network, showing that our networks have differences of up to an order of magnitude in the number of training parameters. Additionally, one of the models outperforms UNet, achieving training in significantly less time and with a Dice Similarity Coefficient (DSC) of up to 94%, ensuring high effectiveness in brain tumor segmentation.

**INDEX TERMS** Artificial intelligence, cancer, deep learning, magnetic resonance imaging, image segmentation.

## I. INTRODUCTION

Brain tumors are abnormal cellular masses or growths that cause severe damage to the nervous system [1]. These are classified into heterogeneous neoplasms ranging from differentiable lesions (e.g., meningiomas) to highly invasive and poorly differentiable lesions such as multiform gliomas [2]. Glioma has the highest mortality rate among brain tumors since it is the most progressive and represents almost 80% of malignant tumors [3], generating a 5-year survival rate of less than 21% in people older than 40 years [4]. However, early and accurate tumor regions' detection significantly reduces these figures [5].

On the other hand, Magnetic Resonance Imaging (MRI) is one of the main noninvasive brain scanning techniques [6]. Besides, MRI is the standard technique for monitoring neoplasms due to the high contrast between soft tissues [7], [8], [9], which allows the affected regions to be seen as changes in intensity and irregular shapes.

In medical practice, segmentation can be performed manually, where the radiologist or professional in charge delimits or segments the affected tissue region [10]. However, the process is tedious, time-consuming, and subject to the professional's interpretation. Besides, this is also subject to

The associate editor coordinating the review of this manuscript and approving it for publication was Syed Islam.

other factors such as skills, experience, and external factors, generating results that are subject to great intra- and inter-evaluator variability [11]. Therefore, an automatic tool is a primary need for detecting tumor tissues. The implications would allow to eliminate human error, make the process more efficient and faster, and even improve treatments due to accurate segmentation in the follow-up of neoplasms undergoing treatments [12].

Automatic developments have had a significant position in the different bioinformatics areas, and interest in these has increased with Deep Learning (DL) techniques [13]. Consequently, it is easy to find DL networks focused on tasks such as classification, detection, prediction, and segmentation [14], [15]. For example, the Convolutional Neural Network (CNN) is widely used in medical image segmentation [16], [17], [18], [19], [20].

Automatic segmentation of brain tumors is not a recent problem. In fact, the first approaches were born from image processing techniques and the emergence of computer vision [21]. Similarly, segmentation is a problem that has been addressed in various disciplines, and new and more robust techniques emerge every day. Implementations range from basic image processing techniques to new artificial intelligence techniques [22]. The latter has even experienced exponential growth in recent years due to the versatility and high performance of the techniques (e.g., such as CNN). in this regard, we present recent works on brain tumor segmentation in structural MRI.

As mentioned above, segmentation is approached from conventional image processing techniques. For example, Mascarenhas et al. [23] define a histogram equalization method, intensity adjustment, binarization, and segmentation through the brain region's coordinates. On the other hand, in a more robust approach, Chen et al. [24] use a support vector machine and an extended Kalman filter, achieving close to 98% accuracy. These approaches are promising; however, most current research focuses on CNNs. Clear examples of this are Jacobo and Mejia [25], Shehab et al. [26], Naveena et al. [27], Jungo et al. [28], Banerjee and Mitra [29], Zhou et al. [30], Chen et al. [31], Baur et al. [32], and Pei et al. [33].

Jacobo and Mejia [25] use a fully convolutional multi-branch architecture with filters of different sizes. Similarly, Shehab et al. [26] use CNNs incorporating blocks with a residual connection in a UNet-like conformation (ResNets), achieving a DSC close to 86% in the segmentation of the whole tumor. Likewise, Naveena et al. [27] reach this same result (Dice Similarity Coefficient (DSC) of 86%) but implementing the network on multi-channel resonance images. For their part, Jungo et al. [28] observe a score of 88% in their quantitative analysis of uncertainty estimation methods, using UNet-like networks trained with the cross-entropy loss function. In contrast, Banerjee and Mitra [29] achieved 90.21%, performing volumetric segmentation from two-dimensional CNNs applied on axial, coronal, and sagittal slices. Besides, Banerjee integrates a consensus fusion strategy with a post-refinement based on conditional random fields in an encoder-decoder network. The model is trained through DSC and weighted log loss functions. In the same vein, Zhou et al. [30] approach the 3D problem from an efficient residual network. The development uses the 3D ShuffleNetV2 network as an encoder, reducing the computational cost. Additionally, Zhou integrates a fusion loss function constituted by the cross-entropy and DSC, achieving a score of 91.21%. In other more ingenious approaches, Chen et al. [31] start from the principle of the two brain hemispheres' structural symmetry to create a symmetric network, looking for asymmetries present with brain tumors. Chen incorporates the residual connection blocks and the focal loss function and adds a post-refinement of the image, obtaining a score of 85.2%.

In recent studies, it is possible to find approaches within unsupervised training or architectures that are not fully convolutional, as Baur et al. [32] and Pei et al. [33]. Baur implements an encoder trained with healthy patients through a reconstruction function, capturing healthy subjects' physiology through an unsupervised network. In contrast, Pie et al. [33] designed a 3D encoder-decoder architecture with a full connection at the architecture's deepest layers. In addition, Pie et al. compare their results with ResNet, UNet, and UNet-VAE architectures and train their design using a two-factor loss function (Dice and semantic loss), achieving the highest scores with their network (values close to 89%) [33].

In different approaches to brain tumors, Chen et al. [34], propose a segmentation network using transformers as the main structure in their model. In particular, the model counts a Transformer at the input of the network, which encodes tokenized image patches from a CNN feature map as the input sequence to extract general contexts. Subsequently, the network is integrated with a decoder that upsamples the encoded features which are then combined with the high-resolution CNN feature maps to enable accurate localization of the lesion/organs of interest. In this research, the results showed that the proposed network managed to outperform the state of the art, achieving an average DSC value of 77.48%. Similarly, Lin et al. [35] proposed a hierarchical swin transform in both the encoder and decoder of the UNet architecture. The network achieved a DSC segmentation score of 94%, but the network used approximately 287 million training parameters. In turn, Xie et al. [36] designed a network for organ segmentation integrated with a CNN and a Deformable Transformer. The architecture achieved a DSC of 85% using approximately 41.9 million training parameters. It is worth mentioning that Chen et al. and Xie et al. recognize the UNet as the de-facto standard, due to its great success [34], [36].

In the specific case of brain tumors, most of the challenges and developments have been based on the BraTs database that includes axial resonance images with their respective segmentation masks, performed by expert radiologists. For example, Kajal and Mittal [37] use this database for the development of a 3D network inspired by the UNet network. The authors compare the performance of their model with different state-of-the-art networks, concluding that their model outperforms other models in terms of accuracy and IoU (accuracy of 98.19% and IoU of 65.88%). Similarly, Pei and Liu [38] propose a 3D U-network (UNet), but including residual connections in their model. The network achieved a DSC of 81.96%, 91.95%, and 85.03% in the segmentation of enhanced tumor, whole tumor, and tumor core, respectively. In the same classes, Hsu et al. [39] achieve DSC of 81.59%, 87.34%, and 91.93%, through their proposed SegResNet-based model implemented with different loss functions. The model includes post-processing, which improves the model performance. Meanwhile, Di Ieva et al. [40] evaluate the accuracy of the best-performing model of the BraTs 2018 challenge, reaching DSC of 87.8%, 73.2%, and 69.9% for the whole tumor, core tumor, and active tumor, respectively. Similarly, Jena et al. [41] analyze the performance of UNet under the DSC and accuracy using cross-entropy as a loss function. The results showed that UNet achieved scores of 98.81% and 99.34% in accuracy and DSC, respectively.

Rahman et al. [42] develop a 3D UNet-Context encoding for improved segmentation. In addition, the model includes epistemic and random uncertainty quantification using Monte Carlo Dropout and Test Time Augmentation to provide confidence in segmentation performance. The results showed that the proposed development achieved DSC of 77.87%, 84.99%, and 91.59% for enhancing tumor, tumor core, and whole tumor, respectively.

Abdullah et al. [43] propose a lightweight network (LBTS-Net) for fast and accurate brain tumor segmentation. The LBTS-Net is based on the VGG architecture, but has half of the convolutional filters in the first layer and uses depth convolution to reduce the number of parameters. In addition, it incorporates transfer learning to tune the network and achieve robust tumor segmentation, achieving an overall accuracy of 98.11% and a DSC of 91%, being significantly more efficient than the standard VGG network. For their part, Micallef et al. [44] propose a variation of the U-Net++, using a different loss function, number of convolutional blocks, and deep supervision method than the standard model. It also incorporates data augmentation and post-processing techniques. The proposed approach achieved a DSC of 71.92%, 87.12%, and 78.17% for the enhancing tumor, whole tumor, and tumor core classes, respectively. Moreover, the proposed model is lightweight and performs similarly to peer-reviewed methods on the same dataset.

In different approaches, Gryska et al. [45] replicate two segmentation models (3D dual-path CNN and 2D single-path CNN), seeking to determine the reproducibility and replicability of the studies. The study found that one of the two methods was successfully reproduced, but the second method could not be reproduced due to insufficient description of the preprocessing pipeline. Nevertheless, the first method showed promising results in terms of the DSC and sensitivity. Meanwhile, Mehta et al. [46] explore and evaluate a metric for quantifying uncertainty in segmentation models. The metric was developed during the BraTS 2019-2020 challenge on uncertainty quantification and is designed to evaluate and rank uncertainty estimates in segmentation. The results confirm the importance and complementary value of uncertainty estimates in medical image analysis and highlight the need to quantify uncertainty in these tasks.

Table 1 shows the metrics reported by the authors in their respective investigations.

While the results are promising, most research focuses on conventional UNet, based on filters or convolutional operators, shaping thousands or millions of training parameters. The approach requires a large number of computational resources, state-of-the-art graphics cards, or equipment with RAM with sufficient capacity to support the demand of the models. The limitation of computational resources has always been an inherent problem in artificial intelligence developments. Consequently, efforts are focused on reducing the computational load without losing the high effectiveness of the models. For example, backward propagation was one of the fundamental algorithms that allowed reducing the computational cost since the gradient is preserved as one moves backward between the layers of the network [47], [48], [49]. On the other hand, techniques such as stochastic downward gradient have been implemented, which, although producing stochastic variation before reaching the optimal values, reduces RAM usage [50], [51]. Based on these motivations, in this research, we propose three computationally inexpensive models for brain tumor segmentation in magnetic resonance imaging. The models are based on separable convolutions, which split the conventional operation into a depthwise convolution and a pointwise convolution. Although these convolutions reduce the computational model cost, this could affect model performance. Therefore, we propose new connections between stages of the UNet network to extract the abstract features of the model to achieve state-of-the-art segmentation while preserving the low computational cost. Our proposed models were integrated with transition layers or layers based on the attention model, modified to the new concept that we call cross-point product, which avoids the loss of dimensionality by combining keys, queries, and values in the multi-head-attention product.

## II. MATERIALS AND METHODS

### A. DATASET

The proposed approach was based on the BraTS2020 challenge database [52], [53], [54]. The set has 369 MR images, and each volume consists of 155 axials 240 × 240 slices in uint8 (8-bit unsigned integer) format. Each image (slice) has four channels corresponding to native (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) acquisition sequences. In addition, all images were manually

**TABLE 1. Metrics reported by related work in medical image segmentation.**

| Main author | Year | Network | #TP | DSC(%) | IoU | ACC | SP | SE | PR | HD | PPV | TPR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mascarenhas *et al.*, [23] | 2020 | Images processing | NA | - | - | - | - | - | - | - | - | - |
| B, Chen *et al.*, [24] | 2020 | SVM | NA | - | - | - | - | - | - | - | - | - |
| Jacobo & Mejia [25] | 2020 | Fully convolutional multi-branch | NR | ***96.4** | 93 | - | - | - | - | - | 96 | 97.5 |
| Shehab *et al.*, [26] | 2020 | Residual UNet | NR | 86 | - | 86 | 91 | - | 92 | - | - | - |
| Naveena *et al.*, [27] | 2020 | CNN on multi-channel | NR | 86 | - | - | 74 | 73 | - | - | - | - |
| Jungo *et al.*, [28] | 2020 | UNet | NR | 88 | - | - | - | - | - | - | - | - |
| Banerjee & Mitra [29] | 2020 | CNN two-dimensional | NR | 90.2 | - | - | 99.3 | 91.4 | - | 4.75 | - | - |
| Zhou *et al.*, [30] | 2021 | ERV-Net | 17.3 | 91.2 | - | - | - | - | - | 3.88 | - | - |
| H, Chen *et al.*, [31] | 2020 | Symmetric and residual CNN | NR | 85.2 | - | - | - | - | - | - | - | - |
| Baur *et al.*, [32] | 2020 | Unsupervised network | NR | 53.7 | - | - | - | - | - | - | - | - |
| Pei *et al.*, [33] | 2020 | 3D encoder-decoder CANet | NR | 89.5 | - | - | - | - | - | 4.9 | - | - |
| J, Chen *et al.*, [34] | 2021 | CNN with Transformers | NR | 77.5 | - | - | - | - | - | 31.7 | - | - |
| Lin *et al.*, [35] | 2022 | Dual swin Transform UNet | 287.75 | 94.2 | 89.4 | - | - | 95 | 93.7 | - | - | - |
| Xie *et al.*, [36] | 2021 | CNN and deformable Transformer (CoTr) | 41.9 | 85 | - | - | - | - | - | - | - | - |
| Kajal & Mittal [37] | 2022 | Modified U-Net | NR | ***78.2** | 64.1 | 97.6 | - | - | - | - | - | - |
| Gryska *et al.*, [45] | 2022 | Dual-path and single-path CNN | NR | 77 | - | - | - | 88 | - | - | 72 | - |
| Pei & Liu [38] | 2022 | 3D ResUNet | 7.8 | 92 | - | - | - | - | - | 6.17 | - | - |
| Hsu *et al.*, [39] | 2022 | SegResnet | 27.5 | 87.3 | - | - | - | - | - | 7.99 | - | - |
| Di Ieva *et al.*, [40] | 2021 | DL CNN | NR | 87.8 | - | - | - | - | - | - | - | - |
| Rahman *et al.*, [42] | 2022 | UNet-ContextEncoding (UNCE) | NR | 75.5 | - | - | - | - | - | 62.8 | - | - |
| Jena *et al.*, [41] | 2022 | UNet | NR | 92.3 | - | 90.4 | - | - | - | - | - | - |
| Abdullah *et al.*, [43] | 2021 | LBTS-Net | 65 | 91 | - | 98.1 | - | - | - | - | - | - |
| Micallef *et al.*, [44] | 2021 | UNet++ | 4.5 | 87.1 | - | - | - | - | - | - | - | - |

segmented by one to four radiologists, and experienced neuroradiologists approved their annotations. The neoplasms were segmented into Necrotic and Non-Enhancing Tumor Core (NCR/NET), GD-Enhancing Tumor (ET), and Peritumoral Edema (ED) conforming all tumor-involved tissue. It is worth mentioning that, in order to reduce the computational cost and training time of the proposed models, axial slices without tumor tissue were excluded; therefore, the base was reduced to 22410 MR images with their respective segmentation masks. In addition, only the images with the T2-FLAIR sequence and the mask of the whole neoplasm were taken for the training of the proposed models. Figure 1 shows two examples from the database.

### B. DATA PREPROCESSING

DL differs from machine learning in that DL can extract features from data automatically. In other words, DL can be implemented on raw data [55], [56], [57]. Therefore, the preprocessing only focused on three basic processes. First, the data type was changed to float32 format. Second, it was normalized, leaving all pixel values in the range of 0 to 1. Finally, all images were adjusted to the size of 128 × 128 to reduce the computational cost during network training.

### C. PROPOSED MODEL

Although convolutional models largely conform to segmentation networks, these can be integrated with new elements or developments. Therefore, we propose integrating the Transformer [58] into the classical UNet structure. In addition, we propose a new approach to the care model, which we call the cross-care model. The approach is born under the idea of limiting dimensionality loss in the attention model's scalar product.

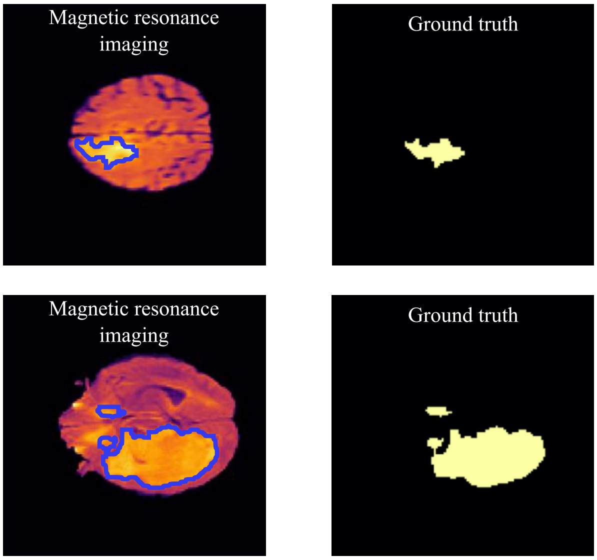


**FIGURE 1. Examples of brain tumors from the BraST2020 database [52], [53], [54].**

Transformers still maintain the same intuitions of DL neural networks but dispense with convolutional and recurrent networks. Figure 2 shows the structure of the Transformer, consisting of an encoder and a decoder described by Vaswani et al. [58].

The architecture is based on the self-attention models, generating the keys, queries, and values from the input data (see Figure 2b). Moreover, the main difference is that it does not depend on previous states, as they are implicit in the positional encoding (see Figure 2d). Therefore, it is possible to use a single attention model or, alternatively, multiple attention models in parallel. The multiple models are known as multiheaded attention. It generates the three matrices of keys, queries, and values for each input embedding vector. For example, let be an embedding vector $x \in R^d$ ($d$ dimensions), then the queries, keys, and values are described by Equations (1), (2) and (3).

$$Q_i = xW_i^q \quad W_i^q \in \mathbb{R}^{d\times d_q} \tag{1}$$

$$K_i = xW_i^k \quad W_i^k \in \mathbb{R}^{d\times d_k} \tag{2}$$

$$V_i = xW_i^v \quad W_i^v \in \mathbb{R}^{d\times d_v} \tag{3}$$

$d_q$, $d_k$ and $d_v$ are the columns of the $i$-th header matrices, whose values must be equal.

The matrices are the outputs of the linear blocks (see Figure 2b) and are used in the scalar product of the attention model (see Figure 2c). The product is the central part of the model, consisting of the matrix product between the keys and queries. The result is scaled by dividing it by $\sqrt{d_k}$, subjected to the softmax activation function, and multiplied with the values, as shown in Equation (4) and Figure 2c.

$$\text{Head}_i = \text{softmax}\left(\frac{Q_i \cdot K_i^T}{\sqrt{d_k}}\right)V_i \tag{4}$$

Although the model is quite efficient, Equation (4) shows that the scalar product $Q_i \cdot K_i^T$ loses the inherent dimensionality in the product. Starting from this premise, the proposed model is based on a cross-care model, i.e., the model is generated with the scalar product in the three possible combinations of matrices (1), (2) and (3). In this sense, the proposed cross products are expressed by Equations (5), (6) and (7).

$$V_{\text{Head}_i} = \text{softmax}\left(\frac{Q_i \cdot K_i^T}{\sqrt{d_k}}\right)V_i \tag{5}$$

$$K_{\text{Head}_i} = \text{softmax}\left(\frac{V_i \cdot Q_i^T}{\sqrt{d_q}}\right)K_i \tag{6}$$

$$Q_{\text{Head}_i} = \text{softmax}\left(\frac{K_i \cdot V_i^T}{\sqrt{d_v}}\right)Q_i \tag{7}$$

In other words, the scalar product in Figure 2c is replaced by the model depicted in Figure 3. The process is weighted and repeated for each header, generating the output of the first sublayer (see Figure 2b), described by Equation (9).

$$\text{Head}_i = \text{mean}\left(V_{\text{Head}_i}, K_{\text{Head}_i}, Q_{\text{Head}_i}\right) \tag{8}$$

$$\text{MultiHead} = (\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_k)W^o \tag{9}$$

Here, $W^o \in R^{d_k h \times d}$ is the matrix of the linear operation shown in Figure 2b, and $h$ is the number of headers of each of the $N$ stacks.

### D. SEGMENTATION NETWORKS BASED ON THE CROSS-ATTENTION MODEL

Our new multi-head-cross-attention layer can be integrated into different configurations or architectures to perform tasks such as classification or segmentation. In this sense, the new block was integrated into low-cost convolutional structures (separable convolutions) to perform brain tumor segmentation. The new layer was added in three models named: model 1, 2, and 3, respectively. Each model is schematically represented by Figure 4, Figure 8 and Figure 11. In the following, each of the three models is described in more detail.

#### 1) MODEL 1

Initially, model 1 (see Figure 4) is a 4-stage deep UNet architecture. Each encoding stage increases the number of features maps and reduces the dimensions of the feature maps to half their initial dimensions. The first stage is a convolution with 32 3 × 3 filters and strides of 2 (see Appendix B). The three subsequent stages are the encoder blocks, depicted in Figure 5.

The block is divided into two trajectories. The main trajectory consists of two separable convolutional layers (see Appendix C), where the number of filters in each layer is twice the number of input feature maps. Subsequently, the dimensions of the feature maps are reduced to half the input size through maximal pooling (see Appendix E). The secondary trajectory only has a 1 × 1 convolution, with strides of 2 and twice as many filters as the input number, adjusting the dimensions of this trajectory to the output of the second one, guaranteeing the residual connection (see Appendix G).

The process described above is repeated throughout the three stages, as illustrated in Figure 4. It should be noted that the output of each stage is the input of the next stage, and similarly, each output is used in the transition layers to concatenate with the output of the encoder stages.

A transition layer was implemented for each encoder-decoder stage (see Figure 6). The layer receives the output of the corresponding stage and reorganizes the maps by taking patches from them, i.e., for a set of feature maps with dimensions (batch, $r$, $c$, $n$) it is restructured to dimensions (batch, $r/m$, $c/m$, $m^2n$), where $m^2$ is the number of patches per map. Then, the patches are embedded in a lower dimensionality tensor using a linear operation and including positional coding, as described in the original Transformer.

The embedding is normalized (see Appendix D) and used on the multi-head cross-attention block. The block is shaped by the proposed mathematical model described in the previous section (see C. proposed model). Subsequently, the output of the block is added with the normalization output in a residual connection. The output is again normalized to give way to a feed-forward layer with the residual connection.

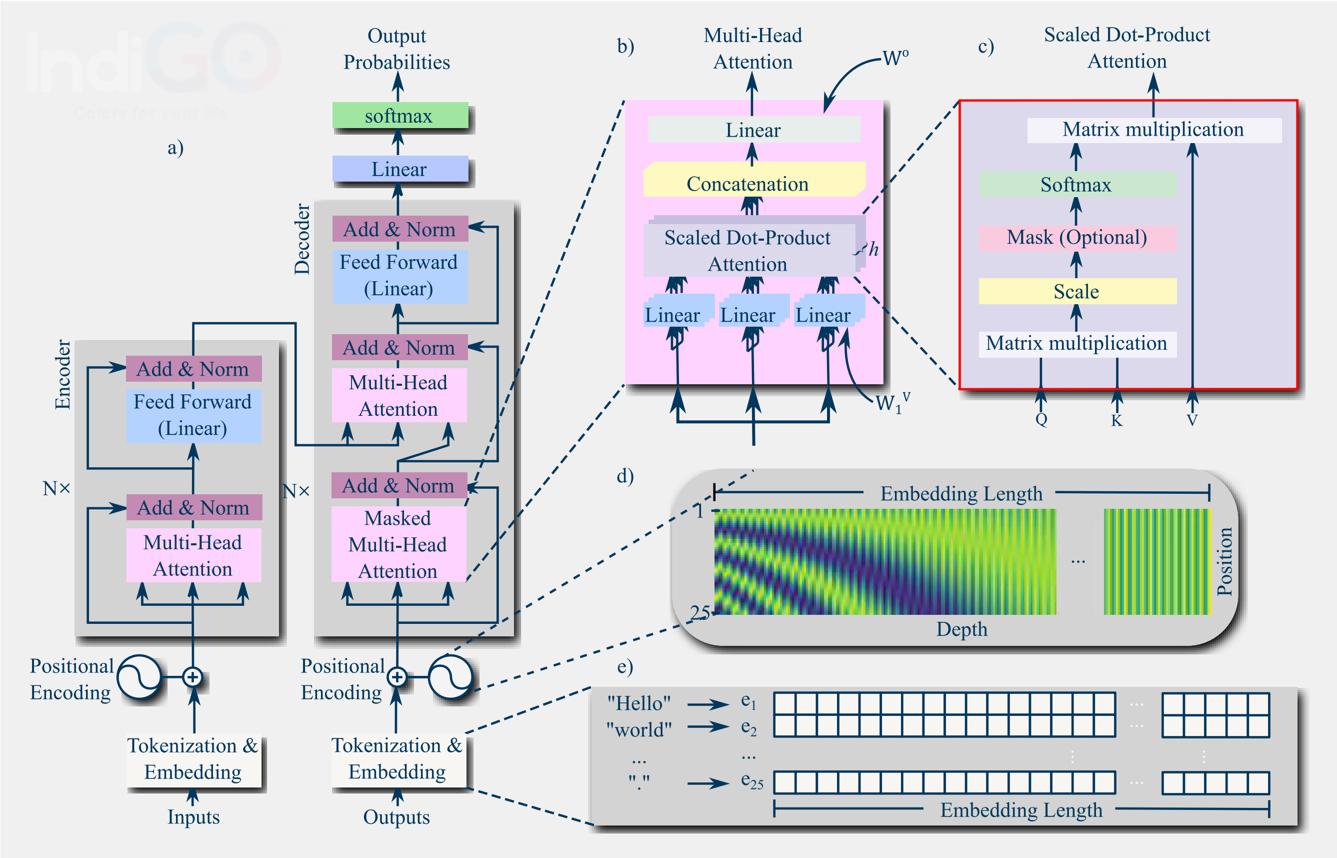


**FIGURE 2.** General structure of the transformer network described by Vaswani et al. a) Conformation of the model as encoder-decoder. b) Multi-head attention block. c) Scaled Dot-Product Attention. d) Positional encoding. e) Tokenization and embedding.

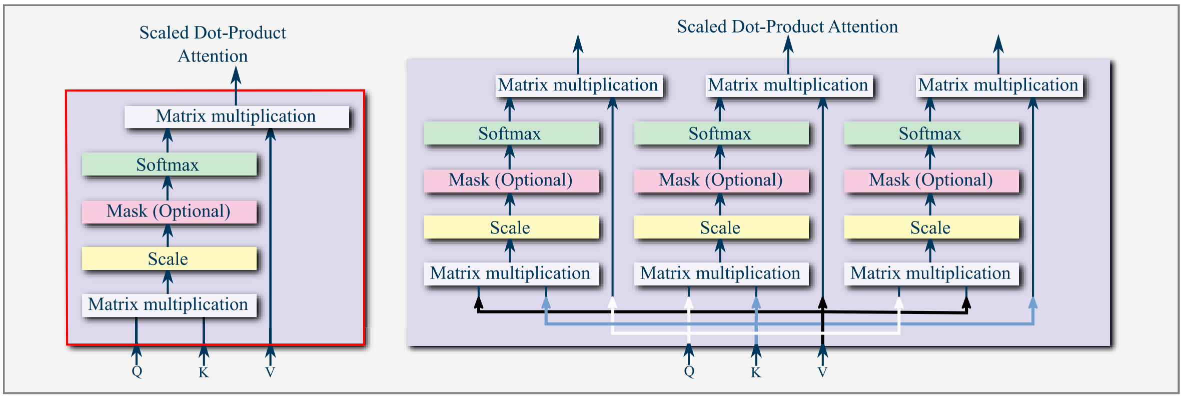


**FIGURE 3.** Schematization of the proposed model; cross-care model between queries, keys, and values.

Finally, the embedding process is reversed, generating new patches used to reconstruct the shapes of the original feature maps.

As mentioned above, the transition layers outputs are concatenated with the decoder blocks outputs, except for the most profound stage, which is concatenated with a copy of the last encoder block.

The encoder blocks perform the opposite process to the encoder blocks (see Figure 7). Each block increases the size of the feature maps, adjusting those maps for concatenation with the respective stage. Again, the encoder block consists of two paths, the first of which has two serially transposed convolutions, with the number of filters corresponding to each stage and a 3 × 3 filter size. Subsequently, the output is upsampled (see Appendix F), doubling the size of the maps. The second trajectory is directly upsampled and convolved 1 × 1, adjusting the dimensions of this trajectory to make the residual connection with the first one.

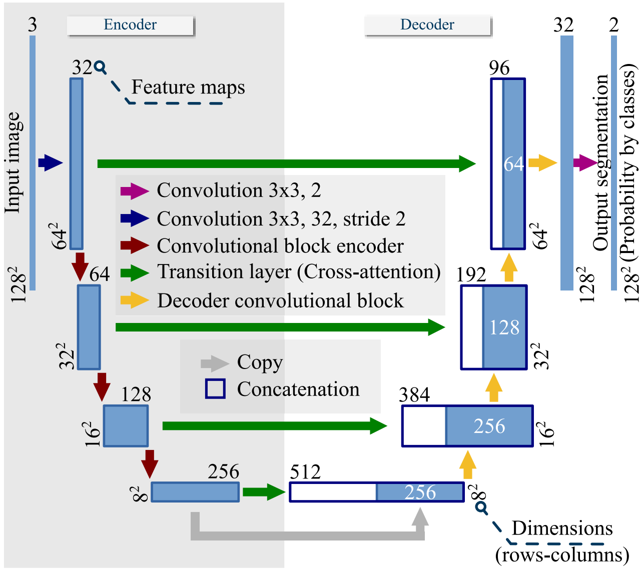

Encoder
Decoder
3
Input image
$128^2$
32
Feature maps
$64^2$
64
$32^2$
128
$16^2$
256
$8^2$
Convolution 3x3, 2
Convolution 3x3, 32, stride 2
Convolutional block encoder
Transition layer (Cross-attention)
Decoder convolutional block
Copy
Concatenation
96
64
$64^2$
192
128
$32^2$
384
256
$16^2$
512
256
$8^2$
Dimensions (rows-columns)
32
2
$128^2$
$128^2$
Output segmentation (Probability by classes)

**FIGURE 4.** General structure of model 1, implemented with the transition layer based on the cross-attention model.

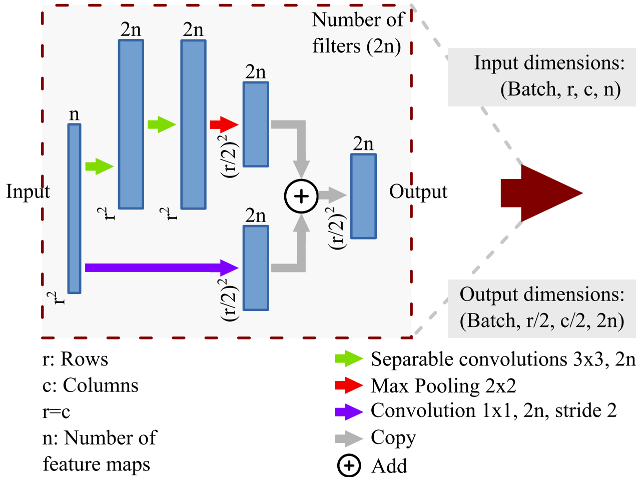

Number of filters (2n)
Input dimensions: (Batch, r, c, n)
Output dimensions: (Batch, r/2, c/2, 2n)
Input
n
$r^2$
2n
2n
$r^2$
$r^2$
2n
$(r/2)^2$
2n
$(r/2)^2$
2n
$(r/2)^2$
Output
r: Rows
c: Columns
r=c
n: Number of feature maps
Separable convolutions 3x3, 2n
Max Pooling 2x2
Convolution 1x1, 2n, stride 2
Copy
⊕ Add

**FIGURE 5.** Convolutional block encoder of model 1.

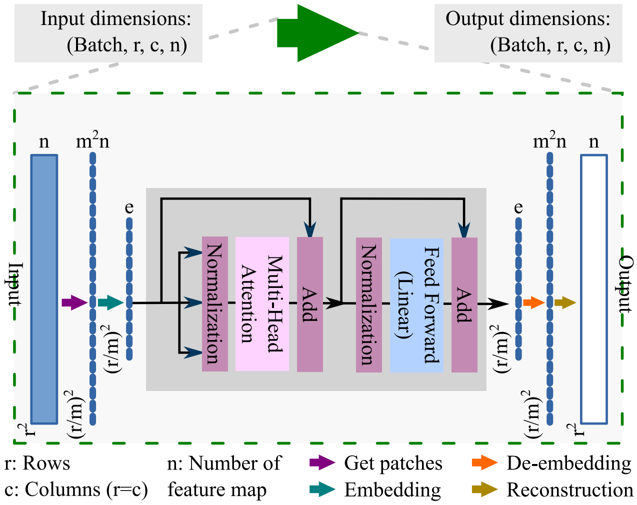


**FIGURE 6.** Transition layer based on the cross-attention model. The transition layer is used in the different parts of the three models.

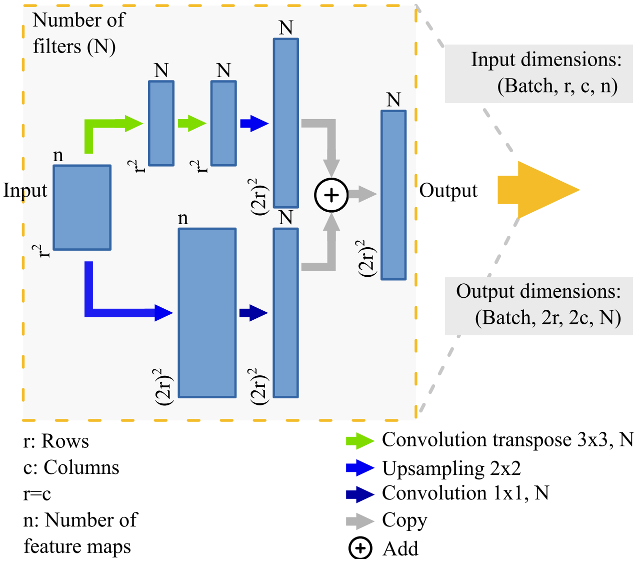


**FIGURE 7.** Decoding convolutional block of model 1.

The process is repeated until the last 3 × 3 convolutional layer with two convolutional filters is reached. The two filters correspond to the probability outputs of the two elements of interest, i.e., a map corresponding to the probability that each pixel is a tumor and a map for the non-tumor pixels.

### 2) MODEL 2

The second model is like the first one; it has the same elements but partially different structural conformations. First, this model has only the transition layer in the last encoder-decoder stage, as shown in Figure 8.

However, the transition layer is included in both the encoder and decoder blocks over the residual connection path, as illustrated in Figure 9 and Figure 10. Additionally, the convolutional layers were changed to separable convolutions in the two blocks but retaining a structure like the first blocks described above.

It should be noted that the transition layers included in Figure 9 and Figure 10, retain the same layout as described in Figure 6.

### 3) MODEL 3

Model 2 follows the same general structure as model 2; however, all convolutions were replaced by separable convolutions. The change was made both in the general structure (see Figure 11 and in the encoder and decoder blocks (see Figure 12 and Figure 13).

Additionally, the transition layers based on the cross-attention model were removed from these blocks. In other

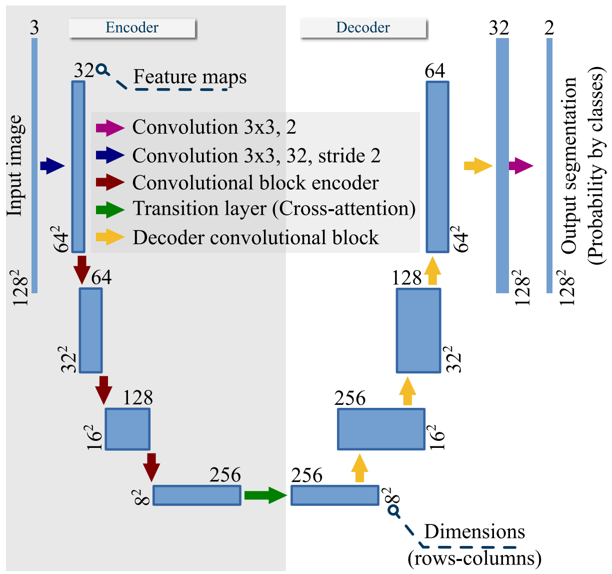


**FIGURE 8.** General structure of model 2, implemented with the transition layer based on the cross-care model.

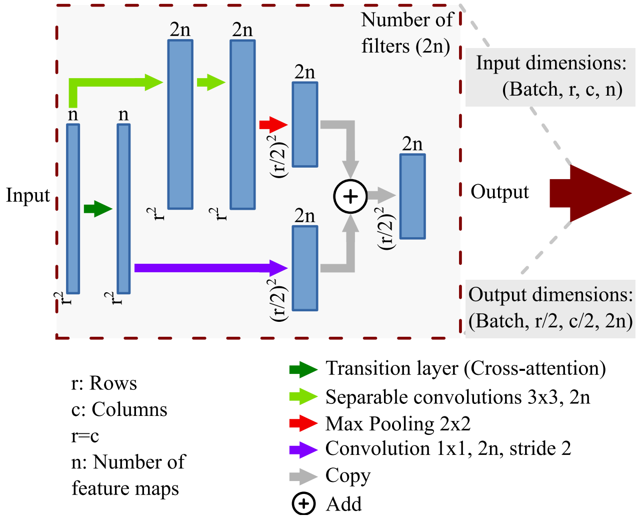

Input
n
n
r²
r²
2n
2n
r²
r²
Number of filters (2n)
2n
(r/2)²
2n
(r/2)²
2n
(r/2)²
Input dimensions: (Batch, r, c, n)
Output
Output dimensions: (Batch, r/2, c/2, 2n)
r: Rows
c: Columns
r=c
n: Number of feature maps
Transition layer (Cross-attention)
Separable convolutions 3x3, 2n
Max Pooling 2x2
Convolution 1x1, 2n, stride 2
Copy
Add

**FIGURE 9.** Convolutional block encoder of model 2. The block implements the transition layer based on the cross-attention model.

words, model 3 only has a single transition layer between the encoder-decoder of the last stage (see Figure 11).

Table 2 highlights the main structural differences of the four models.

### E. DATA AUGMENTATION

Data augmentation is a series of strategies to artificially increase the amount of data or images used to train DL models. The strategies range from the most straightforward geometric transformations to synthesis with artificial neural networks. Although there are many methods, the conventional and simple ones have shown their high effectiveness on DL models. In this sense, only the methods of random horizontal flipping and rotation of the images were integrated

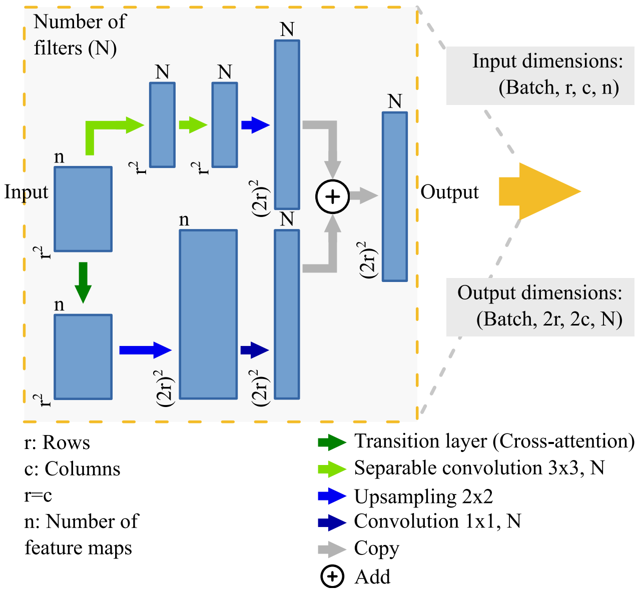

Number of filters (N)
Input
n
r²
N
r²
N
r²
N
(2r)²
n
r²
(2r)²
n
(2r)²
N
(2r)²
N
(2r)²
Output
Input dimensions: (Batch, r, c, n)
Output dimensions: (Batch, 2r, 2c, N)
r: Rows
c: Columns
r=c
n: Number of feature maps
Transition layer (Cross-attention)
Separable convolution 3x3, N
Upsampling 2x2
Convolution 1x1, N
Copy
Add

**FIGURE 10.** Decoder convolutional block of model 2. The block implements the transition layer based on the cross-attention model.

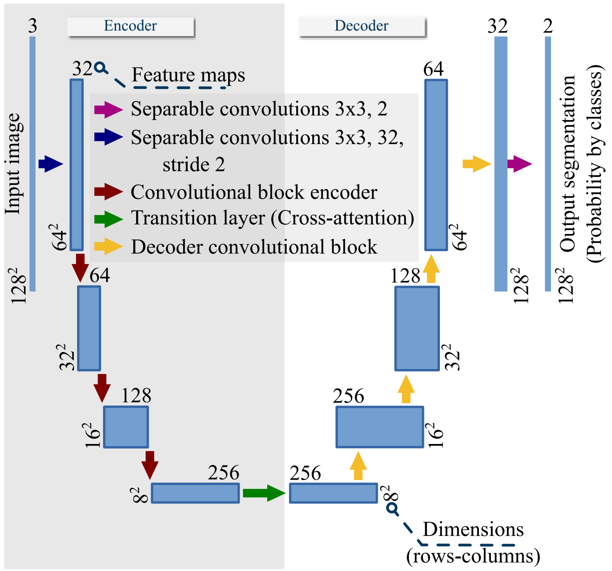


**FIGURE 11.** General structure of model 3, implemented with the transition layer based on the cross-attention model. The model uses only separable convolutions.

into this research. Rotation was performed at 90° angles using the same probability for any of the 4 possible positions (including the original position). Similarly, flipping was used using the same probability for the two positions. The methods are illustrated in Figure 14.

### F. LOSS FUNCTION

The loss function is an objective function used to quantify the difference between the ground truth (labeled) values and the values predicted by the network, i.e., the function allows determining the degree of accuracy or performance of the model. In the case of segmentation, the most used loss is the Dice coefficient, which is described below.

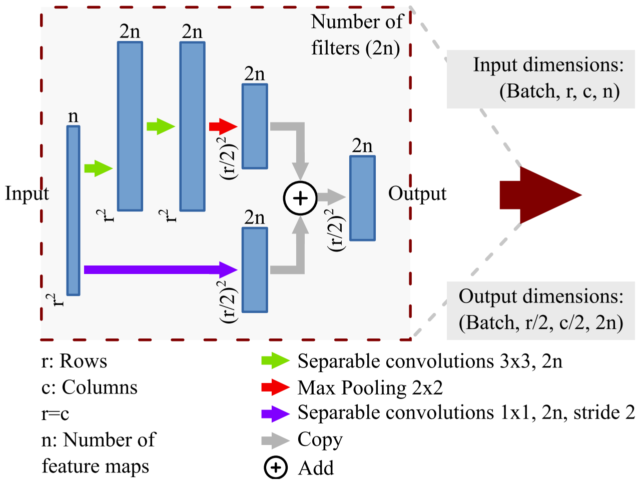

Number of filters (2n)
Input dimensions: (Batch, r, c, n)
Output dimensions: (Batch, r/2, c/2, 2n)
Input
Output
n
2n
2n
2n
2n
2n
$r^2$
$(r/2)^2$
r: Rows
c: Columns
r=c
n: Number of feature maps
Separable convolutions 3x3, 2n
Max Pooling 2x2
Separable convolutions 1x1, 2n, stride 2
Copy
Add

**FIGURE 12.** Convolutional block encoder of model 3. The model uses only separable convolutions.

**TABLE 2.** Main structural characteristics of the three proposed models.

| Model | External convolutions* | Encoders | Transition layers | Decoders |
|---|---|---|---|---|
| 1 | Conventional convolutions | Separable convolutions | Applied to all 4 stages | Conventional and upsampling convolutions |
| 2 | Conventional convolutions | Separable convolutions and internal transition layer | Applied to the last stage | Separable convolutions, upsampling and internal transition layer |
| 3 | Separable convolutions | Separable convolutions | Applied to the last stage | Separable convolutions and upsampling |

*External convolutions refer to convolutions that are not inside the encoder or decoder blocks.

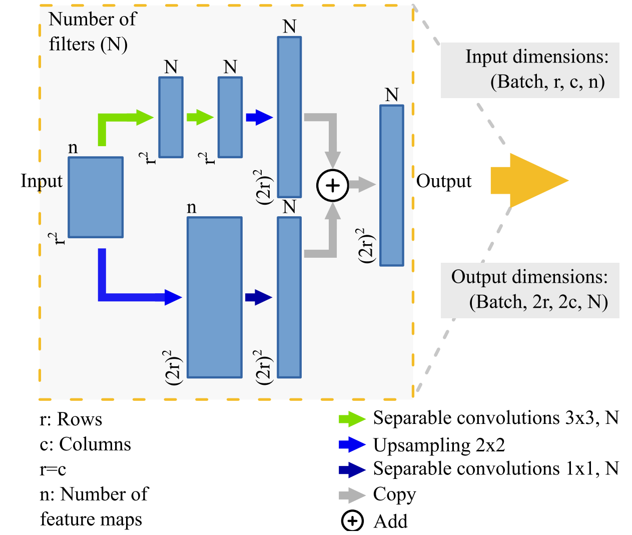

Number of filters (N)
Input dimensions: (Batch, r, c, n)
Output dimensions: (Batch, 2r, 2c, N)
Input
Output
n
n
N
N
N
N
N
$r^2$
$(2r)^2$
r: Rows
c: Columns
r=c
n: Number of feature maps
Separable convolutions 3x3, N
Upsampling 2x2
Separable convolutions 1x1, N
Copy
Add

**FIGURE 13.** Decoding convolutional block of model 3. The model uses only separable convolutions.

### 1) DICE COEFICIENT LOSS

The Dice coefficient is a statistic used to calculate the similarity between two samples, or in the case of computer vision, the similarity between two images [59]. The coefficient is defined mathematically as expressed in Equation (10) and can be used as a loss function through the modification expressed in Equation (11).

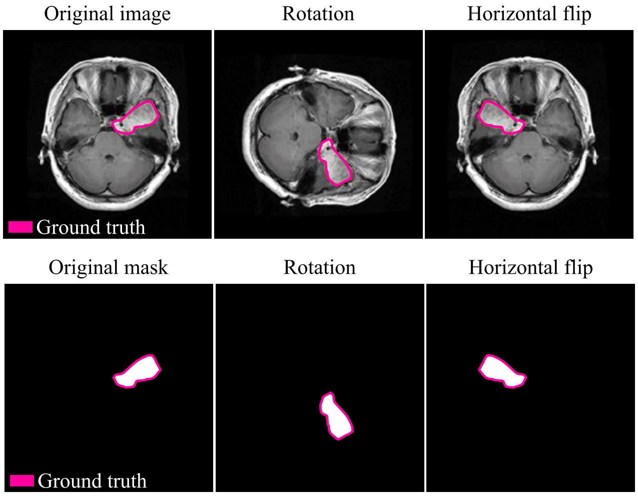


**FIGURE 14.** Examples of data augmentation with image flipping and rotation operations.

$$D(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{10}$$

$$D_{\text{loss}}(A, B) = 1 - \frac{2|A \cap B|}{|A| + |B|} \tag{11}$$

Here, $A$ and $B$ are the actual and predicted regions by the network, respectively. Furthermore, for the case of binary classification, the loss can be rewritten as expressed in Equation (12).

$$D_{\text{loss}} = \frac{FN + FP}{2 \cdot TP + FN + FP} \tag{12}$$

where, TP, FN, and FP are true positives, false negatives, and false positives, respectively.

## G. EVALUATION METRICS

Currently, there are different metrics for the performance evaluation of IA networks; therefore, for the objective evaluation of model performance, it was proposed to use 3 of the most reported metrics in the literature: Jaccard distance or Intersection-Over-Union (IoU), Dice Similarity Coefficient (DSC) and Hausdorff distance (HD). The metrics are expressed mathematically as shown in Equations (13), (14) and (15) [59], [60], [61].

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{13}$$

$$\text{DSC} = \frac{2TP}{2TP + FP + FN} \tag{14}$$

$$\text{HD}(A, B) = \max\{h(A, B), h(B, A)\} \tag{15}$$

IoU and DSC are expressed in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Moreover, $h(A, B)$ is the directed Hausdorff distance, which refers to the minimum distance from the farthest point of $A$ to $B$ and vice versa for $h(B, A)$.

**TABLE 3.** Hyperparameters used in segmentation models.

| Hyperparameters | |
|---|---|
| Loss function | Dice coefficient |
| Optimizer | Adam |
| Epochs | 100 |
| Number or repeated runs per fold | 3 |
| Batch size | 16 |
| Initialization of weights | Uniform Glorot |
| Bias initialization | Zeros |
| Hidden layers activation function | ReLU |
| Output layer activation function | Softmax |

### H. STATISTICAL ANALYSIS

The Kruskal Wallis test was used for statistical estimation between groups, which evaluates whether two or more samples belong to the same distribution based on the median of these samples. The test uses the null hypothesis with the assumption that all samples come from the same distribution. Then, for a $p$ value less than 0.05, it would imply that the null hypothesis is false and, therefore, a statistically significant difference would be established between the two groups tested. Note that the value of 0.05 or significance level can have a lower or higher value. However, this value is the most accepted since it represents only 5% of concluding that there is a difference when there is none [62]. The method, assuming $k$ groups with $n$ observations, defines the $H$ statistic given by the mathematical expression of Equation (16).

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} n_i(\bar{r_{i.}} - \bar{r})^2 \tag{16}$$

$$\bar{r_{ij}} = \frac{\sum_{i=1}^{n_i} r_{ij}}{n_i} \tag{17}$$

where, $n_i$ is the number of observations in the $i$-th group, $N$ is the total number of observations in the two groups, $r_{ij}$ is the rank of the $i$-th observation over the $j$-th observation among all observations, and $k$ is the number of groups [63], [64].

### I. EXPERIMENTAL DESIGN AND HYPERPARAMETERS

The three proposed models based on the cross-attention model were trained together with the classical UNet model to obtain a benchmarking. All models were trained on the BraTS dataset, using an 80% and 20% split for training and testing. In addition, the models were run through 5-fold cross-validation and under the model settings shown in Table 3.

The architectures were modeled with the Python programming language by utilizing the main Keras and TensorFlow libraries. The models were executed on a Colab platform configured with 25 GB of RAM and a Tesla P100 GPU. Each training was evaluated with the Dice Similarity Coefficient (DSC), Jaccard distance or Intersection-Over-Union (IoU), and Hausdorff distance (HD) metrics [59], [60], [61]. Finally, the results were compared with the Kruskal-Wallis non-parametric test statistic, based on the hypothesis that the sample means to come from the same population or distribution [63], [64].

**TABLE 4.** Total number of training parameters in each of the explored networks.

| | Number of training parameters. | Number in millions |
|---|---|---|
| Model 1 | 5256130 | 5.26 |
| Model 2 | 5505378 | 5.51 |
| Model 3 | 1050429 | 1.05 |
| UNet | 43594920 | 43.59 |

## III. RESULTS

This section shows the overall results of the methodology described in the previous section. The four models were trained under the same conditions but repeating the runs to obtain the metrics distributions and a detailed description of the models.

Initially, Table 4 shows the total number of training parameters. The results show a marked difference between the conventional UNet and the proposed models, reaching a reduction of up to an order of magnitude. This represents a significant computational cost reduction due to the reduced number of training parameters in the proposed models.

Table 5 shows the representative values for the three-evaluation metrics and the training time of the models. The results show that model 1 presented the best average performance for the DSC metric, reaching up to an average value of 93.06% with a standard deviation of 0.83%. In other words, the model has a high probability of obtaining a DSC close to 93%. Similarly, model 3 and the reference network (UNet) presented good performance values, around 90% DSC. On the other hand, model 2 was the network with the worst performance, even falling below the conventional UNet network. Additionally, shows the highest metrics achieved by the four models, revealing again that model 1 was the best-performing network, reaching a DSC of 94%.

Regarding the HD metric, the results reflect the same behavior as the DSC metric. Model 1 presented the lowest average HD with the smallest deviation, ensuring the high effectiveness of the model. Similarly, model 1 achieved the lowest HD value, ensuring that the predicted segmentation contour is quite close to the ground truth. Conversely, model 2 presented the highest HD with the largest standard deviation, i.e., the HD metric demonstrates the low segmentation quality of this model.

The scores of the different trainings are shown in Figure 15, except for the outliers. The distributions of the scores give a more detailed description of the behavior of the models. For example, the IoU metric shows that models 1, 3, and UNet had scores between 80 and 90%. However, the UNet network had a higher heterogeneity of scores, while models 1 and 3 presented more homogeneous distributions, ensuring a

**TABLE 5. Representative results of the three metrics for model evaluation and model training time.**

| Models | DSC (%) | IoU (%) | ACC (%) | HD (pixels) | Time (Hours) |
|---|---|---|---|---|---|
| 1 | 93.06 ±0.83 | 87.03 ±1.43 | 99.43±0.08 | 15.10 ±10.25 | 2.63 ±0.12 |
| 2 | 46.96 ±8.72 | 31.13 ±8.00 | 63.91±20.54 | 69.18 ±55.95 | 4.81 ±0.34 |
| 3 | 90.47 ±0.87 | 82.61 ±1.44 | 99.20±0.08 | 36.49 ±13.56 | 3.23 ±0.04 |
| UNet | 88.69 ±7.11 | 80.30 ±10.17 | 99.23±0.29 | 27.35 ±28.24 | 3.65 ±0.86 |
| | Best values (Maximum values of DSC, IoU and Accuracy. Minimum HD values and minimum and maximum range for time) | | | | |
| 1 | 94.00 | 88.68 | 99.51 | 1.20 | 2.49 - 2.79 |
| 2 | 62.78 | 45.75 | 97.88 | 22.56 | 4.43 - 5.87 |
| 3 | 91.66 | 84.60 | 99.30 | 13.24 | 3.18 - 3.30 |
| UNet | 93.13 | 87.14 | 99.46 | 1.33 | 2.95 - 4.95 |

higher probability of obtaining models with values close to the average values reported in Table 5.

Additionally, the box-and-whisker plot shows again that model 2 was the worst performer, stagnating with IoU scores below 40%. Similarly, the DSC presented a similar behavior in the four models; however, model 1, 3, and UNet had distributions close to 90% or above this value. Furthermore, it is again confirmed that model 1 presents the most homogeneous distribution, guaranteeing a high probability of obtaining models with scores close to the mean value (see Table 5).

Figure 15 also presents the distribution of the training times of the models. The results show that model 3 had a more homogeneous behavior than the other models; however, the interquartile range of this model is above the range of model 1. In other words, model 1 had an average training time above 2.5 hours, while model 3 was above 3 hours. In the case of model 2, it had training times between 4.5 and 5 hours. On the other hand, the conventional UNet network presented more significant heterogeneity in training time, ranging from 3 to 5 hours.

Figure 16 shows the training of the four models through the 100 epochs with the respective 95% error bands for loss and validation through the DSC. The curves show that models 1, 3, and UNet performed similarly. All three models converged above 0.9 (90% in percent equivalent), partially above the values generated by the test data (see Table 5 y Figure 15). It is worth noting that the error bands of these results were significantly reduced from epoch 80 onwards. Therefore, this would imply that there was no overtraining of the models or that the overtraining was almost null. In the opposite case, the training curves of model 2 again reflect the model's difficulty in reaching the optimal values of the training parameters, generating an error band that increases with increasing training epochs.

Finally, the scores generated by the 30 runs were compared via the Kruskal Wallis non-parametric test statistic. The p-value of the test statistic is shown in Table 6, comparing all possible combinations. The results indicate that there were no values above the significance level for the IoU and DSC metrics, i.e., there were no values above 0.05. In other words, all models have statistically significant differences. Therefore, our proposed models are different from each other and generate different scores from those of the conventional UNet network.

The $p$-values also show that there is no statistically significant difference between models 2 and 3 for the Hausdorff distance. Similarly, the UNet network and model 3 do not have statistically significant training times, as can be seen in the time plot in Figure 15.

As shown in Table 5, model 1 presented the best performance in brain tumor segmentation. Figure 17 presents seven segmentation samples predicted by model 1. The figure contains the MR image in the axial slice, the ground truth, and the segmentation predicted by that model. From a qualitative point of view, the model segmentations are highly effective for different tumor types, i.e., the predicted segmentation is quite close to the ground truth for both small and large tumors. Moreover, the segmentations differ faintly from the actual contours, ensuring the high effectiveness of the segmentation model despite its low computational cost.

Finally, although a precise comparison with related works cannot be made due to the complexity of the models and the different hyperparameters that affect their performance, in this section we chose to make a comparison based solely on the metrics and hyperparameters reported by the authors. First, we found that our proposed models achieved higher DSC scores than the research performed on the same database (see Table 7). The maximum score reached by our most efficient model (model 1) was 1.73% higher than the highest value reported up to the present year 2022, and the average of the different runs was 0.79% higher than the value reported in the state of the art. Even, our second most efficient model was only below the research of Jena et al. and Pie & Liu, which guarantees the high effectiveness and performance of our proposed models.

Regarding the number of training parameters, Table 8 shows only the related papers that reported the number of training parameters along with the evaluation metrics DSC,

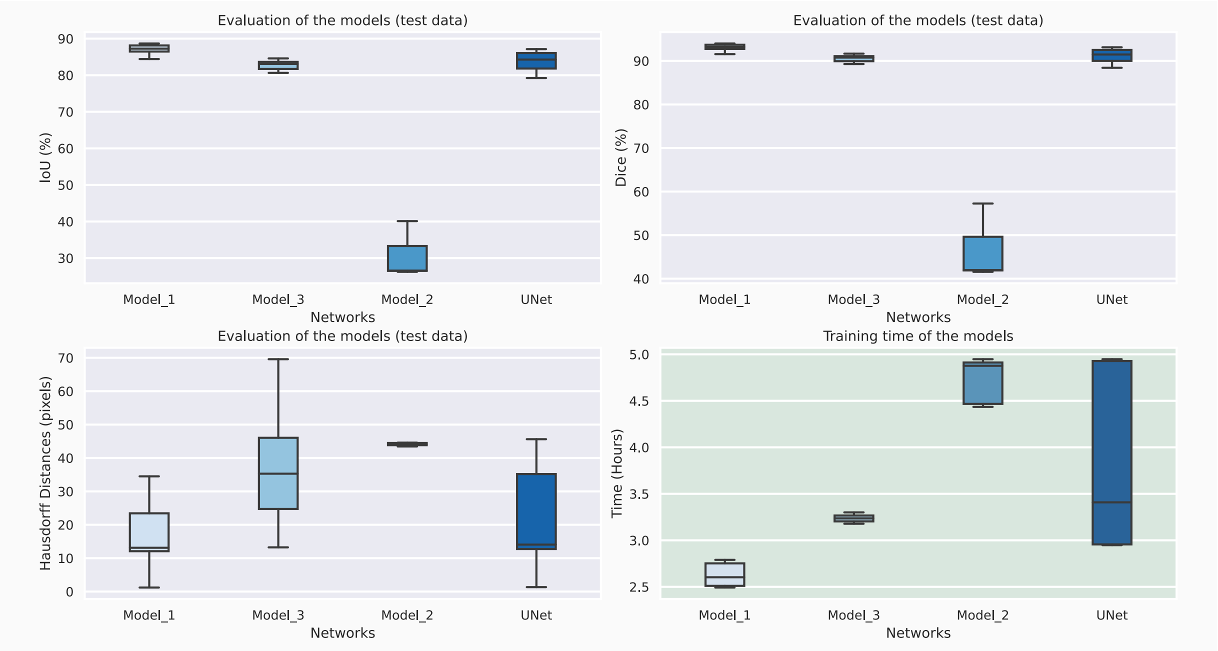


**FIGURE 15.** Distribution of the evaluation metrics of the four models. The graphs show the IoU, DSC (Dice), and Hausdorff metrics and the training time of the models.

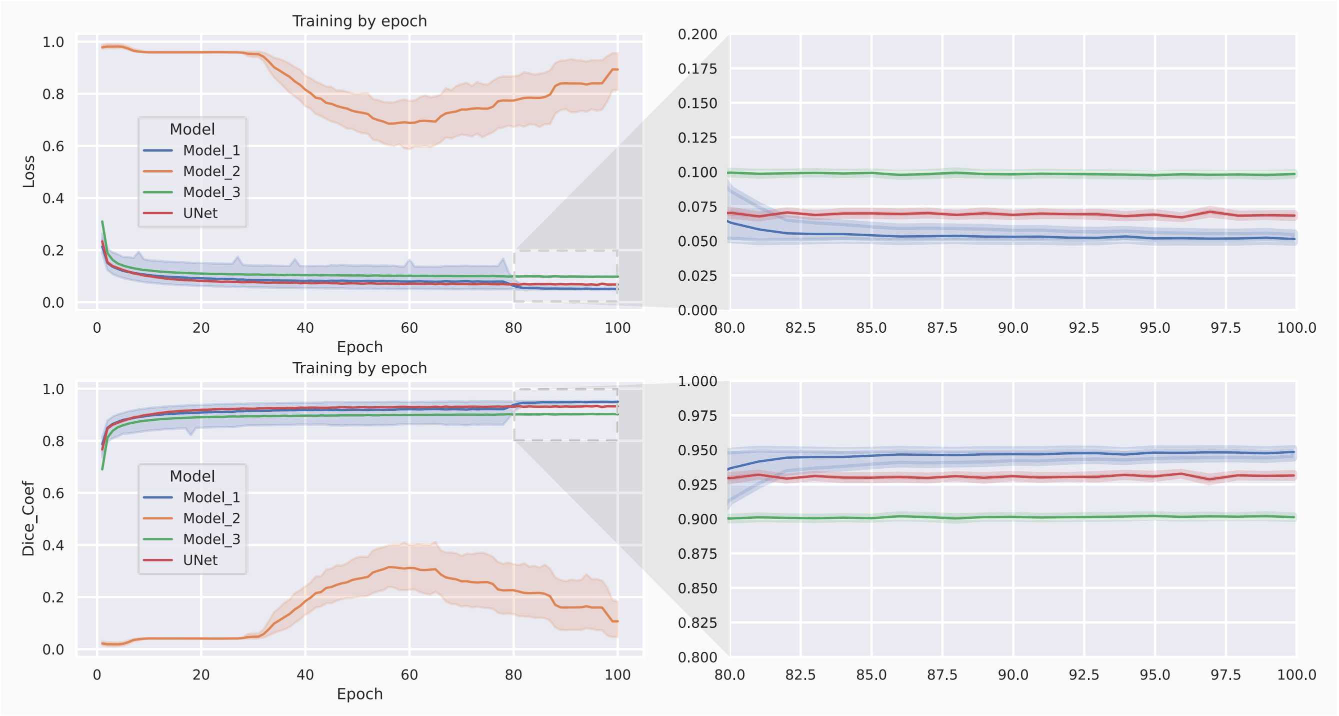


**FIGURE 16.** Training curves of the four models. The curves are given for the model loss and the Dice coefficient as evaluation metrics during the training data execution.

IoU and accuracy. The results show that, our models comprised the top 4 models with the lowest number of parameters. Although model 3, with only 1.05 million parameters, did not have the best DSC, it was only below our model 1 and that of Pei and Liu [38] with 7.8 million parameters. It is worth mentioning that, although the proposed model of Micallef et al. [44] presented the second model with the lowest number of parameters, it achieved a DSC of 87.12%,

**TABLE 6. p-value for the comparison of the models through the non-parametric Kruskal Wallis test.**

P-value from Kruskal Wallis

| | **IoU** | | | | | **DSC** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Model_1 | Model_2 | Model_3 | UNet | | Model_1 | Model_2 | Model_3 | UNet |
| Model_1 | - | <0.001 | <0.001 | <0.001 | Model_1 | - | <0.001 | <0.001 | <0.001 |
| Model_2 | <0.001 | - | <0.001 | <0.001 | Model_2 | <0.001 | - | <0.001 | <0.001 |
| Model_3 | <0.001 | <0.001 | - | 0.013 | Model_3 | <0.001 | <0.001 | - | 0.013 |
| UNet | <0.001 | <0.001 | 0.013 | - | UNet | <0.001 | <0.001 | 0.013 | - |
| | **Hausdorff Distances** | | | | | **Time** | | | |
| | Model_1 | Model_2 | Model_3 | UNet | | Model_1 | Model_2 | Model_3 | UNet |
| Model_1 | - | <0.001 | <0.001 | 0.019 | Model_1 | - | <0.001 | <0.001 | <0.001 |
| Model_2 | <0.001 | - | **0.082** | <0.001 | Model_2 | <0.001 | - | <0.001 | 0.021 |
| Model_3 | <0.001 | **0.082** | - | 0.001 | Model_3 | <0.001 | <0.001 | - | **0.404** |
| UNet | 0.019 | <0.001 | 0.001 | - | UNet | <0.001 | 0.021 | **0.404** | - |

*Values in bold are those that exceed the significance level ($\alpha = 0.05$).

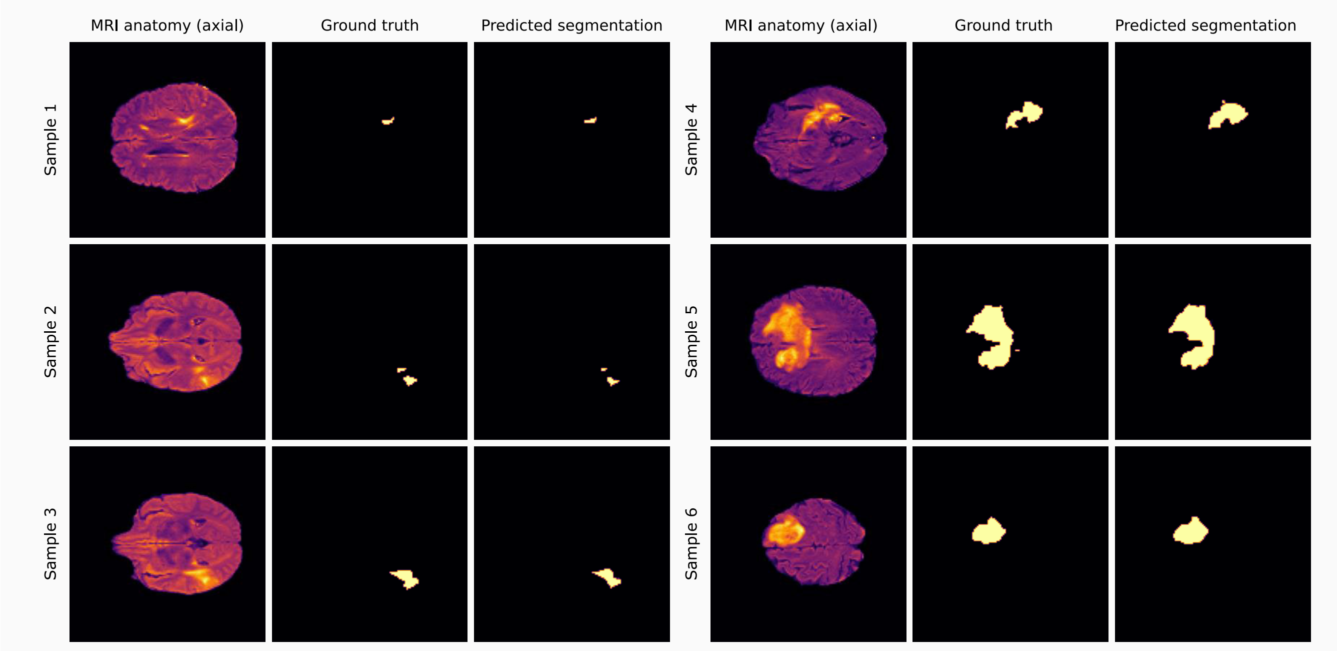


**FIGURE 17. Segmentation results of the best model (Model 1) for different brain lesions.**

lower performance than our models 1 and 3. Additionally, our models had the lowest number of parameters compared to the other two developments based on Transformers, where Lin et al. outperformed our model, but on a different database and with a marked difference in training parameters of up to two orders of magnitude. It is worth mentioning that, Abdullah et al. [43] also integrated depth convolutions with the same purpose of reducing the number of training parameters, however, our models 1 and 3 had lower number of parameters and higher performances in terms of DSC, IoU and accuracy. In this sense, our models guarantee the best performance/number of parameters ratio, generating the best segmentation metrics with the lowest number of training parameters.

In the case of training times, most authors do not report such time or limit themselves to stating that the models had an inference time that exceeds manual execution without giving a quantitative description. In this sense, the works related to training or inference time were reduced to those shown in Table 9. The results show that, our models were outperformed by the development done by Shehab et al. [26]. However, the authors do not report what type of GPU they used in training the model, and, therefore, the comparison between models is not clearly evidenced. On the other hand, Micallef et al [44]

**TABLE 7. Comparison of the metrics of the proposed model and related work based on the BraTs database.**

| Main author | Year | Network | DSC (%) | IoU (%) | ACC (%) |
|---|---|---|---|---|---|
| **Our proposed model 1** | **2022** | **Crosstransunet (max)** | **94.00** | **88.68** | **99.51** |
| **Our proposed model 1** | **2022** | **Crosstransunet (mean)** | **93.06** | **87.03** | **99.43** |
| Jena et al, [41] | 2022 | UNet | 92.27 | - | 90.41 |
| Pei & Liu [38] | 2022 | 3D ResUNet | 91.95 | - | - |
| **Our proposed model 3** | **2022** | **Crosstransunet (max)** | **91.66** | **84.60** | **99.30** |
| Zhou et al, [30] | 2021 | ERVNet | 91.21 | - | - |
| Abdullah et al, [43] | 2021 | LBTSNet | 91.00 | - | 98.11 |
| Banerjee & Mitra [29] | 2020 | CNN twodimensional | 90.21 | - | - |
| Pei et al, [33] | 2020 | 3D encoderdecoder CANet | 89.50 | - | - |
| Jungo et al, [28] | 2020 | UNet | 88.00 | - | - |
| Di Ieva et al, [40] | 2021 | DL CNN | 87.80 | - | - |
| Hsu et al, [39] | 2022 | SegResnet | 87.34 | - | - |
| Micallef et al, [44] | 2021 | UNet++ | 87.12 | - | - |
| Shehab et al, [26] | 2020 | Residual UNet | 86.00 | - | 86.00 |
| Naveena et al, [27] | 2020 | CNN on multichannel | 86.00 | - | - |
| H, Chen et al, [31] | 2020 | Symmetric and residual CNN | 85.20 | - | - |
| Kajal & Mittal [37] | 2022 | Modified UNet | *78.15* | 64.13 | 97.59 |
| Gryska et al, [45] | 2022 | Dualpath and singlepath CNN | 77.00 | - | - |
| Rahman et al, [42] | 2022 | UNetContextEncoding (UNCE) | 75.51 | - | - |

*Metric calculated from metrics reported in the article

**TABLE 8. Related work that reported the number of training parameters in their developments, along with DSC, IoU and accuracy performance metrics.**

| Author | Year | Network | #TP | DSC | IoU | ACC |
|---|---|---|---|---|---|---|
| **Our proposed model 3** | 2022 | Crosstransunet | 1.05 | 91.66 | 84.60 | 99.30 |
| Micallef *et al.* [44] | 2021 | UNet++ | 4.50 | 87.12 | - | - |
| **Our proposed model 1** | 2022 | Crosstransunet | 5.25 | **94.00** | 88.68 | 99.51 |
| **Our proposed model 2** | 2022 | Crosstransunet | 5.50 | 62.78 | 45.75 | 97.88 |
| Pei & Liu [38] | 2022 | 3D ResUNet | 7.80 | 91.95 | - | - |
| Zhou *et al.* [30] | 2021 | ERV-Net | 17.30 | 91.21 | - | - |
| Hsu *et al.* [39] | 2022 | SegResnet | 27.50 | 87.34 | - | - |
| Xie *et al.* [36]* | 2021 | CNN and deformable Transformer (CoTr) | 41.90 | 85.00 | - | - |
| Abdullah *et al.* [43] | 2021 | LBTS-Net | 65.00 | 91.00 | - | 98.11 |
| Lin *et al.* [35]* | 2022 | Dual swin Transform UNet | 287.75 | **94.22** | 89.39 | |

*Segmentation not applied to the BraTs dataset
#TP: Number of training parameters in millions.

implemented UNet++ under the same data set and partially accelerated with the Tesla P100 GPU, showing a high processing time of approximately 14.33 hours per 100 epochs, i.e., almost 3 times our slowest model implemented with the same GPU (model 2).

## IV. DISCUSSION

Cancer is one of the diseases with the highest incidence worldwide, has a high impact on public health costs, and has a high impact on patients' quality of life. Brain tumors are among the cancers with the highest mortality rate since they involve part of the central nervous system. In this sense, classification, diagnosis, and delimitation of the affected areas are vital tasks to provide timely patient care. Healthcare professionals perform many of these tasks, but they are often tedious tasks that require excessive time or repetitive processes, as in the case of segmentation. Fortunately, automatic segmentation methods based on DL have proven to be highly effective, despite the high computational cost required. Based on these considerations, this research focused on developing a DL network with a reduced number of training parameters.

We proposed three new models consisting of separable convolutions and attention blocks with a new model that we call the cross-attention model. The three networks were designed in the UNet form integrating the new proposed

**TABLE 9. Related work that reported the model training or inference time.**

| Author | Year | Network | Time | GPU |
|---|---|---|---|---|
| Banerjee & Mitra [29] | 2020 | CNN two-dimensional | 10 min* | NR |
| H. Chen *et al.* [31] | 2020 | Symmetric and residual CNN | 7-10 s* | NVIDIA GTX 1080 |
| Shehab *et al.* [26] | 2020 | Residual UNet | 1.03 hrs/128 epoch | NR |
| **Our proposed model 1** | **2022** | **Crosstransunet** | **2.63 hrs/100 epochs** | **Tesla P100** |
| **Our proposed model 3** | **2022** | **Crosstransunet** | **3.23 hrs/100 epochs** | **Tesla P100** |
| **Our proposed model 2** | **2022** | **Crosstransunet** | **4.81 hrs/100 epochs** | **Tesla P100** |
| Micallef *et al.* [44] | 2021 | UNet++ | 43hrs/300 epochs | Tesla K80 GPU and Tesla P100 |

*Inference time and not training time

model in different structural conformations (models 1, 2, and 3. see materials and methods, section E).

The results evidenced the reduced number of training parameters, but showed the high effectiveness of the models, comparable to the UNet. For example, all three models were highly efficient (see Table 5); model 1 even managed to outperform the conventional UNet, despite having less than one-fifth of training parameters (see Table 4). Furthermore, model 1 achieved a maximum DSC of 94% outperforming the most robust CNN-based model for brain tumor segmentation (DSC of 91.21% achieved by Zhou et al. [30]). In this sense, it is clear that the cross-attentional models were efficient for feature extraction in MR images, allowing efficient segmentation to be achieved with fewer training parameters. Although the latter did not achieve the performance of model 1, it is an acceptable segmentation considering that only 1.05 million training parameters were used, i.e., only 2.4% of the conventional UNet network (see Table 4). Additionally, the structure of model 3, with a single transition layer, makes it an efficient autoencoder. In this sense, image synthesis could be explored in possible future work or address investigations with the latent variables of the encoder.

On the other hand, the training times show the high effectiveness of the designs, being model 1 the architecture with the shortest average training time, reaching the total training of the model in less than 3 hours. This represents a significant reduction in computational resources (machine execution time). Likewise, the reduced number of training parameters also guarantees the reduction of computational resources, since a greater number of training parameters requires more RAM memory for the adjustment of these parameters.

The training curves confirmed that the models were not overtrained. Moreover, these could be trained with a smaller number of epochs, since the curves converged from epoch 80 onwards, and the error bands were reduced from this epoch. In other words, the models reached the optimal values of the training parameters from epoch 80 onwards, except for model 2. It is worth mentioning that model 2 showed the worst performance. The above would imply that it is not efficient to add the cross-attention model in residual block connections in both encoders and decoders. This would imply that the transition layers within the encoder and decoder blocks are limited in mapping the residual connection identity function and, therefore, prevent the blocks from reaching the optimal values in their training parameters.

Although attention models have been shown to be highly efficient in feature extraction on image tasks, many authors point out that a larger amount of data is needed to reach optimal training parameters [65]. This could be the cause that limits model 2 because it has the cross-cutting care model in each block. However, this opens the possibility for future work. For example, model 2 could be explored in the segmentation task with a large corpus of data to observe the model's behavior in detail.

When it comes to related work, it is challenging to make objective comparisons between deep learning (DL) networks due to the unique design and training of each network. The performance of a DL network can vary based on factors such as the quality and quantity of training data, the number of layers and neurons in each layer, the learning algorithm, hyperparameters, and even hardware acceleration time, which can have a significant impact on both training and inference time. This makes it difficult to evaluate different networks accurately and fairly, particularly when the necessary information to replicate the networks and results is not reported. Gryska et al. [45] even caution about the difficulties in replicating segmentation models and argue that established reproducibility criteria do not adequately emphasize the importance of describing the preprocessing chain. They conclude that a detailed description of the entire preprocessing chain is necessary to establish solid evidence of the generalizability of segmentation methods. Given these limitations, we compared related works based on evaluation metrics, the number of training parameters, and the training or inference time of the model. Our results showed that our models have better segmentation in terms of DSC, IoU, and accuracy. Additionally, the number of training parameters in our models was lower than in most other studies, and Model 3 had the lowest number of parameters. Our results were also significantly superior to studies that aimed to reduce the number of training parameters, such as Abdullah et al. [43], who used separable convolutions for this purpose, but still had several million more parameters than our models.

Clearly, the use of separable convolutions allows to significantly reduce the number of training parameters. In our case, this means that by limiting the number of parameters in the convolutional layers, it was possible to increase the number of parameters in the attention layers, improving feature extraction for segmentation without losing the spatial distribution preserved by the convolutional layers. In other words, the processing load falls on the attention models, but the integration with separable layers preserves the spatial distribution of pixels. Consequently, this shows the high efficiency of the models, as the Transformers allow processing all the input information simultaneously, thanks to the multi-heads, which makes the network very efficient in capturing abstract relationships in all the information. In addition, multiple attention blocks allow information to be extracted from different parts of the input dynamically and adaptively, making it possible to learn more complex and subtle relationships.

Regarding the models using Transformers, it is worth noting that model 1, performed approximately 17% better than the architecture of Chen et al. [34]. Moreover, it achieved a maximum score of 94% equal to the Lin et al. network, but with 5.26 million training parameters in contrast to the 287 million of Lin et al. [35], showing a marked difference in the computational cost of our network. Similarly, model 1 used just over one-fifth of the training parameters of Xie's network (5.26 vs. 41.9 million) [36], but outperformed Xie's score by 9%. Although these results are not fully comparable, due to the set of images used, the values give a good intuition of the excellent performance of cross-attention models. This implies that, the cross-point product between keys, queries, and values, improves feature extraction by avoiding dimensionality loss between such mathematical operations, however, a deeper analysis of the architecture is required, which we will be addressing in future research.

The test statistics revealed that there are statistically significant differences among the four models, i.e., the performance of each model is substantially different among the others, for the DSC and IoU metrics. For the case of Hausdorff distance, the metric is widely used to evaluate the quality of segmentation in images. High quality segmentation is achieved when the Hausdorff distance is small and poor-quality segmentation is achieved when the distance is large. While the metric is widely used to validate the quality of segmentation, the mathematical model shows that the metric is sensitive to small imperfections in image segmentation, since it takes the largest distance between the maxima of the minimum distances between sets. Consequently, a small imperfection can significantly increase the Hausdorff distance, which can give a false impression of the quality in the segmentation. In this sense, the sensitivity of the metric affects the distribution of distances, causing such distributions to have overlapping inter-quartile ranges, which in turn increases the test statistic, i.e., this makes the metric not statistically significant between models, as indicated for models 2 and 3. It is worth mentioning that, although models 2 and 3 are not statistically different for a significance level of 0.05, they have a p-value of 0.08, i.e., the probability of committing type I error is below 10%, which is an acceptable percentage for the case of neural networks. In the case of the test statistic between training times, the p-values show that network 3 and UNet are not statistically different. In other words, it is highly probable that model 3 can take the same training time as the UNet model. This statement can be seen in Figure 15 where model 3 has a homogeneous distribution but close to the median (Q2 quantile) of UNet.

## V. CONCLUSION

In this study, three new DL networks with a reduced number of training parameters were developed for automatic segmentation of brain tumors in structural magnetic resonance images. The networks were based on our proposed new model, called CrossTransUNet, a novel attention model using separable convolutions and inspired by Transformer that integrates a dot product between the queries, keys, and values in the three possible combinations that allows to reduce the computational cost of the models while achieving highly accurate segmentations, measured by DSC.

The results showed that the three proposed models were highly efficient and achieved superior results to UNet, even with significantly fewer training parameters. In addition, training times were significantly shorter for model 1, and it was confirmed that the models were not overtrained. Similarly, comparison with related work demonstrates the superiority of the models, exceeding the values reported in the state of the art in terms of DSC, IoU, Accuracy Metrics, inference times, and even with a lower number of training parameters by up to two orders of magnitude.

Overall, it is concluded that CrossTransUnet models are efficient for feature extraction in MRI images and allow efficient segmentation with fewer training parameters.

## APPENDIX

### A. CONVOLUTIONAL NEURAL NETWORK

Artificial intelligence is one of the areas of computational science with many ramifications. AI ranges from the most straightforward systems composed of linear models to the most recent DL methods. One of the fundamental elements in AI is Convolutional Neural Networks (CNN) or simply convolutional networks. CNNs are one of many bio-inspired systems based on the functioning of the central nervous system, specifically the brain. These emulate the primary visual cortex to perform tasks such as segmentation [66]. In particular, as the name implies, CNNs are shaped by a convolutional operator (or filter) that highlights features of the input images. Currently, there are many networks that differ substantially from the first convolutional model developed by Lecun et al. [67] or from the first convolutional model of DL, known as AlexNet [66]. However, although developments report novel architectures, UNet remains the base structure par excellence for segmentation networks [68].

### B. CONVOLUTION AND CONVOLUTIONAL LAYERS

In general, convolution is a mathematical operation described as the sum of the product of two functions with the argument of one of them inverted and offset by a value $t$. The operation can be applied to discrete space with a few minor modifications and is the fundamental basis of convolutional neural networks. The two-dimensional convolution $C$ between two matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{3 \times 3}$ is mathematically denoted as expressed in Equation (A.1).

$$C = A * B \tag{A.1}$$

That is, if the matrices $A$ and $B$ are made up of the elements described in equation (A.2), these would produce the matrix $C \in \mathbb{R}^{(n-2)\times(n-2)}$ (see Equation (A.3)).

$$C = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} * \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,1} & b_{3,3} \end{bmatrix} \tag{A.2}$$

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n-2} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-2,1} & c_{n-2,2} & \dots & c_{n-2,n-2} \end{bmatrix} \tag{A.3}$$

Here, each element of matrix $C$ is given by Equation (A.4).

$$c_{i,j} = \sum_{k=1}^{3}\sum_{m=1}^{3} a_{i+k-1,j+m-1} \cdot b_{k,m} \tag{A.4}$$

Note that the $(*)$ represents convolution (it is not a conventional matrix product).

In the case of artificial neural networks, the smaller matrix is called the filter or kernel. In addition, convolution is generally a multi-channel operation corresponding to the multiple feature maps that are generated throughout the network. That is, for a set of $N$-channel feature maps, one must have a filter $K$ with the same number of channels, such that the output of the convolutional layer can satisfy the model described by Equation (A.5).

$$C = \varphi\left(b + \sum_{i=1}^{N} A_i * K_i\right) \tag{A.5}$$

Here, $b$ is the bias associated with the filter $K$, and $\boldsymbol{\varphi}$ is the activation function of the model. Then, for a convolutional layer of $k$ filters, a feature map (or channel) is generated for each filter as described in Equation (A.6).

$$C_j = \varphi\left(b_j + \sum_{i=1}^{N} A_i * K_{ij}\right) \quad j = 1, 2, 3, \dots, k \tag{A.6}$$

It should be emphasized that Equation (A.6) is the convolutional model of the neural network; therefore, the weights that constitute each filter $K_{ij}$ and the biases $b_j$ are the training parameters of the network.

### C. SEPARABLE CONVOLUTION

Separable convolutions are like the conventional convolutions described in the previous section. However, they differ from the former in that the operation is divided into two convolutions: a depthwise convolution and a pointwise convolution. That is, for a convolution equivalent to the previous one (set of $N$-channel feature maps and $3 \times 3$ filter), the depthwise convolution described by Equation (A.7), must be performed, where each filter $D_i$ has the exact dimensions ($D_i \in \mathbb{R}^{3\times 3}$).

$$A_{D_i} = A_i * D_i \quad \text{i} = 1, 2, 3, \dots, \text{N} \tag{A.7}$$

Subsequently, the operation is used in the pointwise convolution described by Equation (A.8).

$$C = \varphi\left(b + \sum_{i=1}^{N} A_{Di} * P_i\right) \tag{A.8}$$

Here, the filter $P$ has dimensions $1 \times 1$ and has the same number of channels as the input. Therefore, for a separable convolutional layer of $k$ filters, the maps described by Equation (A.9) are generated.

$$C_j = \varphi\left(b_j + \sum_{i=1}^{N} A_{Di} * P_{ij}\right) \quad j = 1, 2, 3, \dots, k \tag{A.9}$$

Note that regardless of the number of filters selected, the depthwise convolution is the same for all $k$ filters; therefore, the number of training parameters is smaller in this operation.

### D. BATCH NORMALIZATION

Batch normalization was proposed by Ioffe and Szegedy [69]. The method allows for faster and more stable training through neural networks by normalizing, centering, and scaling the inputs to each layer. In principle, the method was devised to mitigate the problem of the internal covariates change produced by the internal distribution change of each feature map and the random initialization of the weights, which limit the learning rate. This unfavorable effect can be reduced by adjusting the distribution toward a standard normal, i.e., a distribution with a mean of 0 and a standard deviation of 1. The process is mathematically expressed by Equation (A.10).

$$\hat{x}_j = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{A.10}$$

Here, $\hat{x}_j$ presents the input activation maps of the j-th mini-batch. $\mu_\mathrm{B}$ and $\sigma_\mathrm{B}^2$ are the mean and variance of the activation maps, and $\varepsilon$ is a constant added for numerical stability of the variance. Additionally, the normalization is fit to an optimal distribution ($y_j$) by learning the coefficients of a linear transformation, as expressed in Equation (A.11).

$$y_j = \gamma \cdot \hat{x}_j + \beta \tag{A.11}$$

The model learns the parameters $\gamma$ and $\beta$ generating the new distribution and improving the model performance [70]. The transformation also smooths the gradient flow and acts as a regularization layer [69].

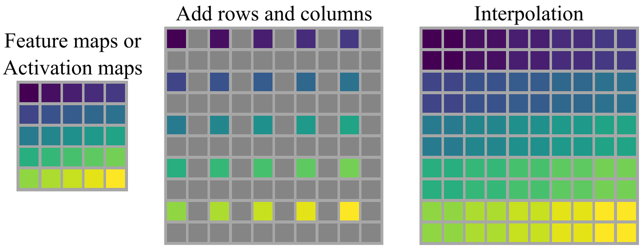


**FIGURE 18.** Upsampling interpolated to the nearest values.

### E. MAX POOLING

The method is a pooling operation that calculates the maximum value across patches or windows along the feature maps. Generally, for a set of feature maps $A$ of dimensions $R \times C$, the pooling expressed in Equation (A.12) would be generated for an operation with 2 × 2 patches or windows.

$$MaxPooling\left(A\right) = \begin{bmatrix} M_{1,1} & M_{1,2} & \ldots & M_{1,c} \\ M_{2,1} & M_{2,2} & \ldots & M_{2,c} \\ \vdots & \vdots & \ddots & \vdots \\ M_{r,1} & M_{r,2} & \ldots & M_{r,c} \end{bmatrix} \tag{A.12}$$

where, the $M_{i,j}$ are the maxima of the windows, as expressed in Equation (A.13).

$$\begin{gathered} M_{m,n} = max \begin{bmatrix} A_{2m,2n} & A_{2m,2n+1} \\ A_{2m+1,2n} & A_{2m+1,2n+1} \end{bmatrix} \\ m = 1, 2, 3, \ldots, \left(\frac{R}{2}\right) \quad n = 1, 2, 3, \ldots, \left(\frac{C}{2}\right) \end{gathered} \tag{A.13}$$

It should be emphasized that the windows are applied along the rows and columns; therefore, the number of channels is not affected.

### F. UPSAMPLING

Upsampling could be considered the opposite process of pooling. In general, the process consists of increasing the size of the activation maps by adding new rows and columns filled with values determined by some interpolation method. In the simplest case, the activation maps have rows and columns added to them interleaved with the original rows and columns (interpolation to the nearest). The new rows and columns are filled with the closest value, as illustrated in Figure 18.

Upsampling can be implemented with different interpolation methods such as area, bicubic, bilinear, Gaussian, etc. However, nearest-neighbor interpolation is usually the most commonly used method due to its low computational cost.

### G. RESIDUAL CONNECTION

Residual connection is a simple but highly effective concept to facilitate the training of neural networks, mitigating the effect produced by gradient fading. The residual connection creates a trajectory parallel to a set of convolutional layers by applying the identity mapping to the input layers and summing this result to the output, as illustrated in Figure 19.

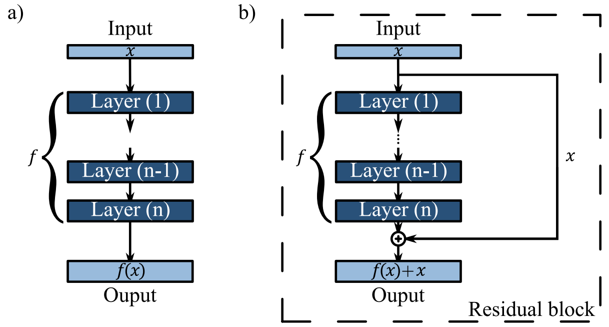


**FIGURE 19.** a) traditional network without residual connections, b) network with the residual connection. The identity mapping path can have a direct connection or a convolutional layer.

Generally, the identity mapping is accompanied by a 1 × 1 convolution with the number of filters corresponding to the central trajectory's output to allow summation of the output depicted in Figure 19.

## REFERENCES

[1] J. C. Buckner, P. D. Brown, B. P. O'Neill, F. B. Meyer, C. J. Wetmore, and J. H. Uhm, "Central nervous system tumors," *Mayo Clinic Proc.*, vol. 82, no. 10, pp. 1271–1286, Oct. 2007, doi: 10.4065/82.10.1271.

[2] K. M. Vernau and P. J. Dickinson, "Brain tumors," in *Consultations in Feline Internal Medicine*, J. R. August, Ed., 5th ed. Saint Louis, MO, USA: W.B. Saunders, 2006, ch. 54, pp. 505–516. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0721604234500573, doi: 10.1016/B0-72-160423-4/50057-3.

[3] J. Chen, W. Huan, H. Zuo, L. Zhao, C. Huang, X. Liu, S. Hou, J. Qi, and W. Shi, "Alu methylation serves as a biomarker for non-invasive diagnosis of glioma," *Oncotarget*, vol. 7, no. 18, pp. 26099–26106, May 2016, doi: 10.18632/oncotarget.8318.

[4] American Society of Clinical Oncology. (2021). *Brain Tumor: Statistics Cancer.Net Doctor-Approved Patient Information From ASCO*. Accessed: Aug. 31, 2021. [Online]. Available: https://www.cancer.net/cancer-types/brain-tumor/statistics

[5] M. K. Abd-Ellah, A. I. Awad, A. A. M. Khalaf, and H. F. A. Hamed, "A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned," *Magn. Reson. Imag.*, vol. 61, pp. 300–318, Sep. 2019, doi: 10.1016/j.mri.2019.05.028.

[6] F. Rabai and R. Ramani, "Magnetic resonance imaging: Anesthetic implications," in *Essentials of Neuroanesthesia*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 519–532, doi: 10.1016/B978-0-12-805299-0.00031-2.

[7] L. M. DeAngelis, "Brain tumors," *New England J. Med.*, vol. 344, no. 2, pp. 114–123, Jan. 2001, doi: 10.1056/NEJM200101113440207.

[8] R. Lang, L. Zhao, and K. Jia, "Brain tumor image segmentation based on convolution neural network," in *Proc. 9th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2016, pp. 1402–1406, doi: 10.1109/CISP-BMEI.2016.7852936.

[9] S. Bauer, R. Wiest, L.-P. Nolte, and M. Reyes, "A survey of MRI-based medical image analysis for brain tumor studies," *Phys. Med. Biol.*, vol. 58, no. 13, pp. R97–R129, Jul. 2013, doi: 10.1088/0031-9155/58/13/R97.

[10] M. P. A. Starmans, S. R. van der Voort, J. M. C. Tovar, J. F. Veenland, S. Klein, and W. J. Niessen, "Radiomics," in *Handbook of Medical Image Computing and Computer Assisted Intervention*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 429–456, doi: 10.1016/B978-0-12-816176-0.00023-5.

[11] A. Işın, C. Direkoğlu, and M. Şah, "Review of MRI-based brain tumor image segmentation using deep learning methods," *Proc. Comput. Sci.*, vol. 102, pp. 317–324, Jan. 2016, doi: 10.1016/j.procs.2016.09.407.

[12] M. Reuter, E. R. Gerstner, O. Rapalino, T. T. Batchelor, B. Rosen, and B. Fischl, "Impact of MRI head placement on glioma response assessment," *J. Neuro-Oncol.*, vol. 118, no. 1, pp. 123–129, May 2014, doi: 10.1007/s11060-014-1403-8.

[13] D. Ravi, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G. Z. Yang, "Deep learning for health informatics," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 4–21, Jan. 2017, doi: 10.1109/JBHI.2016.2636665.

[14] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.

[15] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017, doi: 10.1016/j.neucom.2016.12.038.

[16] S. Li, G. K. F. Tso, and K. He, "Bottleneck feature supervised U-Net for pixel-wise liver and tumor segmentation," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113131, doi: 10.1016/j.eswa.2019.113131.

[17] G. Pan, Y. Zheng, S. Guo, and Y. Lv, "Automatic sewer pipe defect semantic segmentation based on improved U-Net," *Autom. Construct.*, vol. 119, Nov. 2020, Art. no. 103383, doi: 10.1016/j.autcon.2020.103383.

[18] Y. Yang, C. Feng, and R. Wang, "Automatic segmentation model combining U-Net and level set method for medical images," *Expert Syst. Appl.*, vol. 153, Sep. 2020, Art. no. 113419, doi: 10.1016/j.eswa.2020.113419.

[19] P. Hambarde, S. Talbar, A. Mahajan, S. Chavan, M. Thakur, and N. Sable, "Prostate lesion segmentation in MR images using radiomics based deeply supervised U-Net," *Biocybern. Biomed. Eng.*, vol. 40, no. 4, pp. 1421–1435, Oct. 2020, doi: 10.1016/j.bbe.2020.07.011.

[20] B. Wu, Y. Fang, and X. Lai, "Left ventricle automatic segmentation in cardiac MRI using a combined CNN and U-Net approach," *Comput. Med. Imag. Graph.*, vol. 82, Jun. 2020, Art. no. 101719, doi: 10.1016/j.compmedimag.2020.101719.

[21] E. C. Freuder, "Affinity: A relative approach to region finding," *Comput. Graph. Image Process.*, vol. 5, no. 2, pp. 254–264, 1976, doi: 10.1016/0146-664X(76)90033-2.

[22] C. L. Chowdhary and D. P. Acharjya, "Segmentation and feature extraction in medical imaging: A systematic review," *Proc. Comput. Sci.*, vol. 167, pp. 26–36, Jan. 2020, doi: 10.1016/j.procs.2020.03.179.

[23] L. R. Mascarenhas, A. D. S. Ribeiro, and R. P. Ramos, "Automatic segmentation of brain tumors in magnetic resonance imaging," *Einstein (São Paulo)*, vol. 18, Feb. 2020, Art. no. eAO4948, doi: 10.31744/einstein_journal/2020AO4948.

[24] B. Chen, L. Zhang, H. Chen, K. Liang, and X. Chen, "A novel extended Kalman filter with support vector machine based method for the automatic diagnosis and segmentation of brain tumors," *Comput. Methods Programs Biomed.*, vol. 200, Mar. 2021, Art. no. 105797, doi: 10.1016/j.cmpb.2020.105797.

[25] M. Z. Jacobo and J. Mejia, "Segmentation of brain tumor on magnetic resonance imaging using a convolutional architecture," Mar. 2020, *arXiv:2003.07934*.

[26] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and M. S. El-Mahallawy, "An efficient brain tumor image segmentation based on deep residual networks (ResNets)," *J. King Saud Univ., Eng. Sci.*, vol. 33, no. 6, pp. 404–412, 2021, doi: 10.1016/j.jksues.2020.06.001.

[27] C. Naveena, S. Poornachandra, and V. N. M. Aradhya, "Segmentation of brain tumor tissues in multi-channel MRI using convolutional neural networks," in *Proc. Int. Conf. Brain Informat.*, vol. 12241, 2020, pp. 128–137, doi: 10.1007/978-3-030-59277-6_12.

[28] A. Jungo, F. Balsiger, and M. Reyes, "Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation," *Frontiers Neurosci.*, vol. 14, pp. 1–13, Apr. 2020, doi: 10.3389/fnins.2020.00282.

[29] S. Banerjee and S. Mitra, "Novel volumetric sub-region segmentation in brain tumors," *Frontiers Comput. Neurosci.*, vol. 14, pp. 1–13, Jan. 2020, doi: 10.3389/fncom.2020.00003.

[30] X. Zhou, X. Li, K. Hu, Y. Zhang, Z. Chen, and X. Gao, "ERV-Net: An efficient 3D residual neural network for brain tumor segmentation," *Expert Syst. Appl.*, vol. 170, May 2021, Art. no. 114566, doi: 10.1016/j.eswa.2021.114566.

[31] H. Chen, Z. Qin, Y. Ding, L. Tian, and Z. Qin, "Brain tumor segmentation with deep convolutional symmetric neural network," *Neurocomputing*, vol. 392, pp. 305–313, Jun. 2020, doi: 10.1016/j.neucom.2019.01.111.

[32] C. Baur, S. Denner, B. Wiestler, N. Navab, and S. Albarqouni, "Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study," *Med. Image Anal.*, vol. 69, Apr. 2021, Art. no. 101952, doi: 10.1016/j.media.2020.101952.

[33] L. Pei, L. Vidyaratne, M. M. Rahman, and K. M. Iftekharuddin, "Context aware deep learning for brain tumor segmentation, subtype classification, and survival prediction using radiology images," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Nov. 2020, doi: 10.1038/s41598-020-74419-9.

[34] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers make strong encoders for medical image segmentation," Feb. 2021, *arXiv:2102.04306*.

[35] A. Lin, B. Chen, J. Xu, Z. Zhang, G. Lu, and D. Zhang, "DS-TransUNet: Dual swin transformer U-Net for medical image segmentation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022, doi: 10.1109/TIM.2022.3178991.

[36] Y. Xie, J. Zhang, C. Shen, and Y. Xia, "CoTr: Efficiently bridging CNN and transformer for 3D medical image segmentation," 2021, *arXiv:2103.03024*.

[37] M. Kajal and A. Mittal, "A modified U-Net based architecture for brain tumour segmentation on BRATS 2020," *Res. Square*, 2022, doi: 10.21203/rs.3.rs-2109641/v1.

[38] L. Pei and Y. Liu, "Multimodal brain tumor segmentation using a 3D ResUNet in BraTS 2021," in *Proc. Int. MICCAI Brainlesion Workshop*, 2022, pp. 315–323, doi: 10.1007/978-3-031-08999-2_26.

[39] C. Hsu, C. Chang, T. W. Chen, H. Tsai, S. Ma, and W. Wang, "Brain tumor segmentation (BraTS) challenge short paper: Improving three-dimensional brain tumor segmentation using SegResNet and hybrid boundary-dice loss," in *Proc. Int. MICCAI Brainlesion Workshop*, 2022, pp. 334–344, doi: 10.1007/978-3-031-09002-8_30.

[40] A. Di Ieva, C. Russo, S. Liu, A. Jian, M. Y. Bai, Y. Qian, and J. S. Magnussen, "Application of deep learning for automatic segmentation of brain tumors on magnetic resonance imaging: A heuristic approach in the clinical scenario," *Neuroradiology*, vol. 63, no. 8, pp. 1253–1262, Aug. 2021, doi: 10.1007/s00234-021-02649-3.

[41] B. Jena, G. K. Nayak, S. Paul, and S. Saxena, "An exhaustive analytical study of U-Net architecture on two diverse biomedical imaging datasets of electron microscopy drosophila ssTEM and brain MRI BraTS-2021 for segmentation," *Social Netw. Comput. Sci.*, vol. 3, no. 5, p. 418, Aug. 2022, doi: 10.1007/s42979-022-01347-y.

[42] M. M. Rahman, M. S. Sadique, A. G. Temtam, W. Farzana, L. Vidyaratne, and K. M. Iftekharuddin, "Brain tumor segmentation using UNet-context encoding network," in *Proc. Int. MICCAI Brainlesion Workshop*, 2022, pp. 463–472, doi: 10.1007/978-3-031-08999-2_40.

[43] M. A. M. Abdullah, S. Alkassar, B. Jebur, and J. Chambers, "LBTS-Net: A fast and accurate CNN model for brain tumour segmentation," *Healthcare Technol. Lett.*, vol. 8, no. 2, pp. 31–36, Apr. 2021, doi: 10.1049/htl2.12005.

[44] N. Micallef, D. Seychell, and C. J. Bajada, "Exploring the U-Net++ model for automatic brain tumor segmentation," *IEEE Access*, vol. 9, pp. 125523–125539, 2021, doi: 10.1109/ACCESS.2021.3111131.

[45] E. Gryska, I. Björkman-Burtscher, A. S. Jakola, T. Dunås, J. Schneiderman, and R. A. Heckemann, "Deep learning for automatic brain tumour segmentation on MRI: Evaluation of recommended reporting criteria via a reproduction and replication study," *BMJ Open*, vol. 12, no. 7, Jul. 2022, Art. no. e059000, doi: 10.1136/bmjopen-2021-059000.

[46] R. Mehta et al., "QU-BraTS: MICCAI BraTS 2020 challenge on quantifying uncertainty in brain tumor segmentation—Analysis of ranking scores and benchmarking results," Dec. 2021, *arXiv:2112.10074*.

[47] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Tech. Rep., 1974.

[48] P. Munro, "Backpropagation," in *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2011, p. 73, doi: 10.1007/978-0-387-30164-8_51.

[49] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[50] N. Ketkar, "Stochastic gradient descent," in *Deep Learning With Python*. Berkeley, CA, USA: Apress, 2017, pp. 113–132, doi: 10.1007/978-1-4842-2766-4_8.

[51] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 661–670, doi: 10.1145/2623330.2623612.

[52] B. H. Menze, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Trans. Med. Imag.*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015, doi: 10.1109/TMI.2014.2377694.

[53] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features," *Sci. Data*, vol. 4, no. 1, Dec. 2017, Art. no. 170117, doi: 10.1038/sdata.2017.117.

[54] S. Bakas et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge," Nov. 2018, *arXiv:1811.02629*.

[55] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim, ''Deep learning in medical imaging: General overview,'' *Korean J. Radiol.*, vol. 18, no. 4, p. 570, 2017, doi: 10.3348/kjr.2017.18.4.570.

[56] A. S. Lundervold and A. Lundervold, ''An overview of deep learning in medical imaging focusing on MRI,'' *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, May 2019, doi: 10.1016/j.zemedi.2018.11.002.

[57] B. J. Erickson, ''Basic artificial intelligence techniques,'' *Radiol. Clinics North Amer.*, vol. 59, no. 6, pp. 933–940, Nov. 2021, doi: 10.1016/j.rcl.2021.06.004.

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' 2017, *arXiv:1706.03762*.

[59] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, ''Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,'' in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2017, pp. 240–248, doi: 10.1007/978-3-319-67558-9_28.

[60] M. A. Rahman and Y. Wang, ''Optimizing intersection-over-union in deep neural networks for image segmentation,'' in *Proc. Int. Symp. Vis. Comput.*, 2016, pp. 234–244, doi: 10.1007/978-3-319-50835-1_22.

[61] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, ''Comparing images using the Hausdorff distance,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993, doi: 10.1109/34.232073.

[62] A. F. M. Alkarkhi, ''The observed significance level (P-value) procedure,'' in *Applications of Hypothesis Testing for Environmental Science*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 79–119, doi: 10.1016/B978-0-12-824301-5.00010-1.

[63] P. E. McKight and J. Najab, ''Kruskal–Wallis test,'' in *The Corsini Encyclopedia of Psychology*. Hoboken, NJ, USA: Wiley, Jan. 2010, doi: 10.1002/9780470479216.corpsy0491.

[64] E. Ostertagová, O. Ostertag, and J. Kováč, ''Methodology and application of the Kruskal–Wallis test,'' *Appl. Mech. Mater.*, vol. 611, pp. 115–120, Aug. 2014, doi: 10.4028/www.scientific.net/AMM.611.115.

[65] J. M. J. Valanarasu, P. Oza, I. Hacihaliloglu, and V. M. Patel, ''Medical transformer: Gated axial-attention for medical image segmentation,'' in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2021, pp. 36–46, doi: 10.1007/978-3-030-87193-2_4.

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[67] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

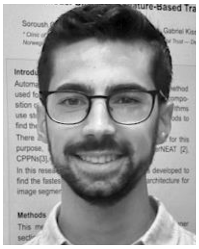
[68] M. Aljabri and M. AlGhamdi, ''A review on the use of deep learning for medical images segmentation,'' *Neurocomputing*, vol. 506, pp. 311–335, Sep. 2022, doi: 10.1016/j.neucom.2022.07.070.

[69] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 1, Feb. 2015, pp. 448–456.

[70] I. Goodfellow, Y. Bengio, and A. Courville, ''Optimization for training deep models,'' in *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, pp. 313–317. [Online]. Available: http://www.deeplearningbook.org

**ANDRÉS ANAYA-ISAZA** received the degree in systems engineering from Universidad Cooperativa de Colombia, in 2009, the first master's degree in computer engineering from Atlantic International University, in 2012, with a specialty in algorithmics, and the second master's degree in systems and computer engineering, with specialization in machine learning from the Technological University of Pereira, Colombia, in 2017. He is currently pursuing the Ph.D. degree in bioengineering and artificial intelligence with Pontificia Universidad Javeriana. He is also the Vice President of research and innovation with Indigo Technologies.

**LEONEL MERA-JIMÉNEZ** was born in El Tambo, Nariño, Colombia, in April 1990. He received the degree in physical engineering from Universidad del Cauca, Colombia, in 2015, and the master's degree in engineering, emphasizing bioengineering, from the University of Antioquia, Colombia, in 2021. He is currently a Machine Learning Researcher with Indigo Technologies. His main research interests include computer vision, machine learning techniques, deep learning, image processing, and biomedical imaging applications.



**ALVARO FERNANDEZ-QUILEZ** was born in Barcelona, Spain. He received the degree in telecommunications engineering from UPC, Barcelona, the master's degree in computational biomedical engineering from UPF, Barcelona, and the Ph.D. degree in AI and medical imaging from the University of Stavanger, Norway, in collaboration with Stavanger University Hospital (SUS). He is currently developing his research as a Postdoctoral Researcher of AI applied to medical imaging with SUS and the University of Stavanger.