Deep Learning-Based Automatic Segmentation of 3D Medical Images

Sithembiso Ntanzi, *Student, UKZN,* Serestina Viriri, *Supervisor, UKZN,*

***Abstract*— There have been significant breakthroughs in developing models for segmenting 3D medical images, with many promising results attributed to the incorporation of Vision Transformers (ViT). However, the fundamental mechanism of transformers, known as self-attention, has quadratic complexity, which significantly increases computational requirements, especially in the case of 3D medical images. In this paper, we investigate the UNETR++ model and propose a Voxel-Focused Attention Mechanism inspired by TransNeXt Pixel-focused Attention. The core component of UNETR++ is the Efficient-Paired Attention (EPA) block, which learns from two interdependent branches: spatial and channel attention. For spatial attention, we incorporated the Voxel-Focused Attention Mechanism, which has linear complexity with respect to input sequence length, rather than projecting the keys and values into lower dimensions. This effectively reduces the model's parameter count while maintaining competitive performance and inference speed. On the Synapse dataset, the enhanced UNETR++ model contains 21.42M parameters, a 50% reduction from the original 42.96M, while achieving a competitive Dice score of 86.72%. The implementation is available on GitHub**

***Index Terms*—Volumetric Medical Image Segmentation, Efficient Attention, Hybrid Architecture, 3D Sliding Window**

INTRODUCTION

Volumetric medical imaging devices, such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), have had a significant impact on the medical field [1], [2]. Segmenting the 3D medical images produced by these devices is crucial, as it aids in quantitative analysis, guiding surgical procedures, and improving patient intervention outcomes. Classical image segmentation methods, such as Computer-aided diagnosis (CADx), which use tools like thresholding and the paintbrush, are often inefficient and time-consuming [3], [4].

Advancements in deep learning have led to the development of more efficient and accurate segmentation models. Generally, deep learning segmentation methods for 3D medical images can be divided into three categories: convolution-based, transformer-based, and hybrid. Convolution-based models rely solely on convolutional neural networks (CNNs) while transformer-based rely solely on transformers. Hybrid models, particularly for 3D medical imaging, combine CNNs and Transformers, and typically follow a U-shaped architecture with an encoder and a decoder. The adoption of Transformers has been driven by the success of Vision Transformers (ViTs) in various visual tasks, such as classification, segmentation, restoration, synthesis, registration, and object detection, where they often outperform fully CNN-based techniques [5], [6], [7], [8], [9], [10], [11].

In this paper, we investigated UNETR++ [12], which features a hybrid hierarchical encoder-decoder architecture that incorporates both Transformer and Convolution methods. The main component of UNETR++ is the Efficient-Paired Attention (EPA) block, which consists of two interdependent branches: the spatial branch and the channel branch. Spatial attention is applied to the spatial branch, while channel attention is applied to the channel branch. The fundamental component of Transformers is the self-attention mechanism [13], which scales quadratically with the input sequence length. This significantly impacts the computation of attention in medical volumetric images due to their large input sequences. To address this, the spatial attention in the original EPA block projects the keys and values into a lower dimension, making the computation of self-attention linear. For channel attention, the computation is performed on the channel dimension, which is relatively smaller compared to the spatial dimension, so no projection is applied to the channel branch. The weights for the keys and queries are shared between both branches, helping capture interdependent features from both the spatial and channel branches, while the values have different weights.

We experimented with full and semi voxel-focused attention (VFA) on the spatial branch to compute the attention weights. The performance difference was not significant when evaluated on the ACDC dataset: the full-VFA achieved an average Dice score of 92.9, while

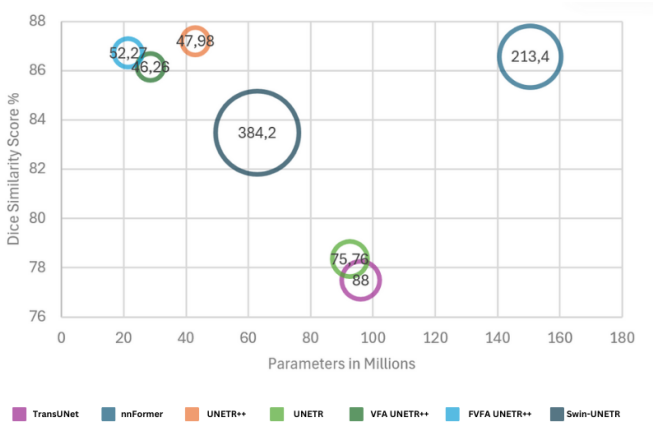

Fig. 1: Model parameters, Dice similarity score, and FLOP size. The circle size represents the FLOP size.

the semi-VFA scored 92.7. The full-VFA had fewer trainable parameters but a longer inference time compared to the semi-VFA. The experiments were conducted on three benchmarks: Synapse [14], ACDC [15], and BRaTs [16]. The enhanced UNETR++ model incorporating the VFA achieved comparable results across these datasets while having significantly fewer parameters. Figure 1 presents a visual comparison of the Dice similarity score, model parameters, and FLOPs for state-of-the-art models on the Synapse dataset.

I. Literature Review and Related Work

A. Convolution-based Segmentation Methods

Convolutional Neural Networks (CNNs) were among the first deep learning techniques to be adopted to segment medical images. U-Net [17] was one of the pioneering CNN-based models that demonstrated excellent performance in segmenting medical images. Since then, various U-Net variants have been developed to enhance performance for different medical imaging modalities and segmentation tasks [18], [19], [20], [21]. For volumetric medical image segmentation, some U-Net variants employ a strategy that avoids using the full 3D volumetric images, a technique commonly referred to as 2.5D segmentation [22]. In contrast, 3D methods utilise the entire volumetric data. 3D U-Net [20] is an example of a U-Net variant that processes the complete volume of the images. nnU-Net [23] provides a highly flexible framework due to its ability to configure the architecture, making it capable of handling both 2D and 3D images and adaptable to different data preprocessing requirements. Liu et al. [24] introduced ConvNeXt, a Convolutional Neural Network derived from ResNet [18], to revitalise CNNs as Vision Transformers began outperforming CNNs across multiple vision tasks. Some models have adopted architectural features from ConvNeXt. MedNeXt [25], for instance, is fully based on ConvNeXt for the segmentation of 3D medical images. Furthermore, some methods have explored the use of dilated depthwise convolutions and large kernels to capture more contextual information [26].

B. Transformer-based Segmentation Methods

Vision Transformers (ViTs) have demonstrated exceptional results across multiple visual tasks, outperforming CNNs. The self-attention mechanism is the key component of Transformers, enabling them to capture global context by encoding the relationships between image patches [13]. However, the standard self-attention mechanism has a quadratic computational complexity with respect to the input sequence length, which in this case refers to image patches. In recent years, several approaches have been proposed to reduce this complexity. Guo et al. [27]. addressed the issue by introducing small, external learnable shared memory units for the keys and values, resulting in a mechanism with linear complexity, which they named External Attention. TransNeXt [11] attempts to reduce the complexity of the self-attention mechanism using a pixel-focused attention mechanism inspired by the focal perception mode of biological vision. In recent years, fully transformer-based designs have also been explored. Huang et al. introduced LightViT [28], a convolution-free design incorporating learnable tokens to capture global dependencies and a bi-dimensional attention module to aggregate global information across both channel and spatial dimensions. LightViT has demonstrated good performance on various 2D visual tasks. Karimi et al. [29]. proposed a convolution-free design specifically for 3D medical image segmentation. In this approach, 3D images are broken down into patches, flattened, embedded into a 1D representation, and passed through the network to predict the segmentation map.

C. Hybrid Segmentation Methods

Due to the strong inductive spatial bias of CNNs, some research incorporates CNNs to help capture local context. Hybrid designs leverage the strengths of both CNNs and Transformers to capture local and global dependencies, respectively. TransUNet [30] is the first hybrid framework for medical image segmentation, consisting of a CNN-Transformer encoder. The Transformer within the encoder encodes global context from high-resolution spatial CNN features. The decoder is entirely

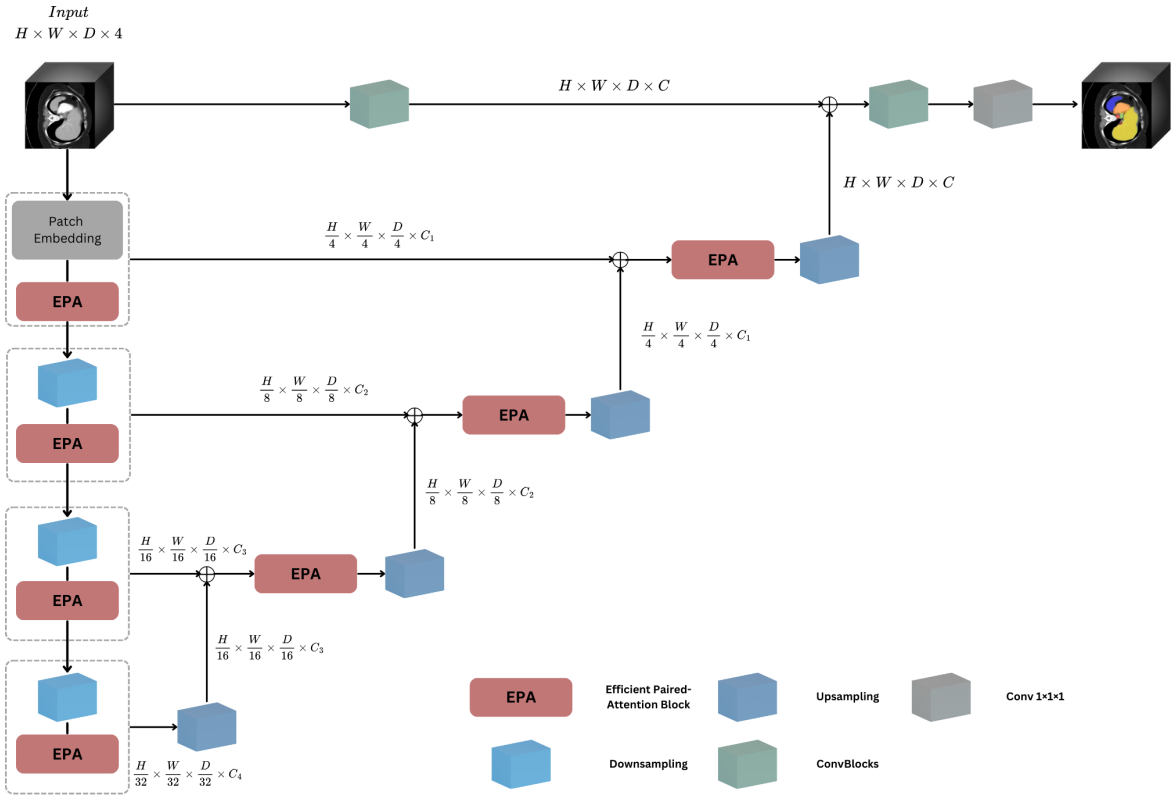

Fig. 2: Overview of the Enhanced UNETR++ Architecture: The overall structure of the architecture remains unchanged, with enhancements applied only to the Efficient Paired-Attention (EPA) block.

CNN-based and upsamples the encoded features. UNETR [31] uses a Transformer as the encoder and a CNN-based decoder to capture localised information. The encoder and decoder are connected via skip connections at different resolutions. nnFormer [32] has a hierarchical feature representation and is rooted in the SwinUNet [33] architecture. nnFormer adapts the Swin Transformer's shifted window approach to 3D segmentation. It divides the input into 3D patches and employs local and global volume-based self-attention mechanisms. Liu et al. [34]. introduced a hierarchical encoder-decoder model with skip connections (SCANeXt) for 3D medical image segmentation. While the hierarchical encoder-decoder structure is similar to that of UNETR++, SCANeXt replaces the EPA block with the dual attention and depthwise convolution (DADC) block. The DADC block utilises dual attention and depthwise convolution, inspired by ConvNeXt. The dual attention mechanism captures global context, while the depthwise convolution block extracts multiscale features.

II. Methods and Techniques

This section outlines the architecture of the enhanced UNETR++ model, which incorporates the Voxel-Focused Attention (VFA) mechanism. The overall architecture remains unchanged; only the EPA block in the spatial branch has been enhanced to calculate attention using VFA.

A. Overall Architecture

The UNETR++ architecture features a hierarchical encoder-decoder structure connected by skip connections, drawing inspiration from the UNETR architecture introduced by Hatamizadeh et al [31]. It comprises various components, including ConvBlock, Downsampling, Upsampling, 1×1×1 Convolution, and EPA blocks as shown in Figure 2.

In this hierarchical design, the Downsampling layers progressively reduce the resolution of the feature maps by a factor of 2 at each stage through a non-overlapping convolution operation. The encoder consists of four stages. The first stage employs patch embedding to divide the image volume into non-overlapping 3D patches, projecting them into the channel dimension. The size of the patches can be represented as $\frac{D}{P_1}, \frac{H}{P_2}, \frac{W}{P_3}$, where (P_1, P_2, P_3) is the patch resolution. The corresponding sequence length can be expressed as $N = \frac{D}{P_1} \times \frac{H}{P_2} \times \frac{W}{P_3}$. At each subsequent stage, the Downsampling layer is applied, followed by the EPA block.

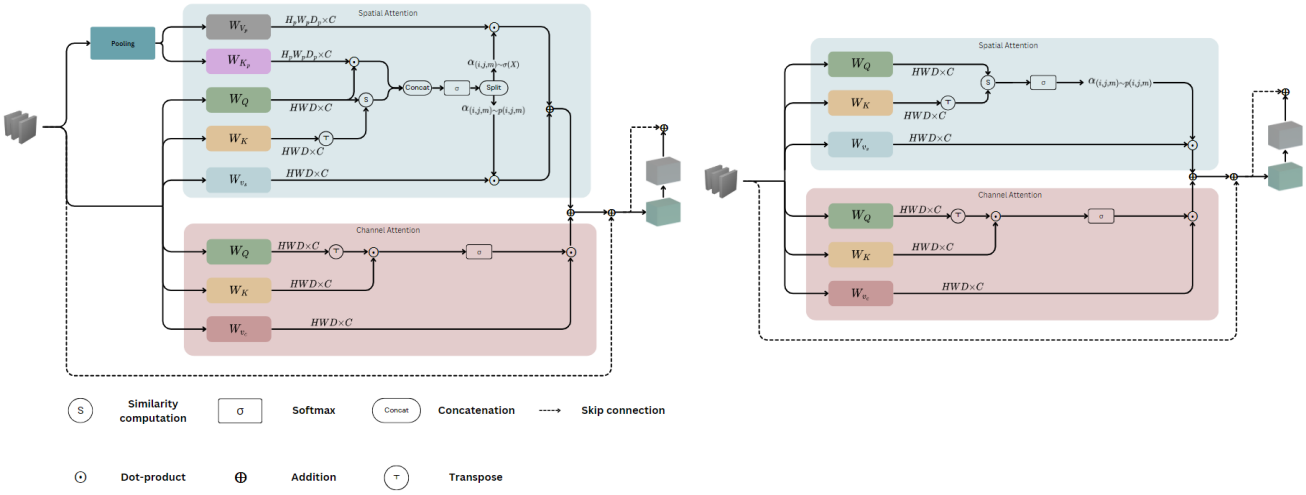

Fig. 3: Overview of the Enhanced Efficient Paired-Attention Block with the spatial branch at the top and the channel branch at the bottom. **Left:** The spatial branch is enhanced to use the Full Voxel-Focused Attention mechanism. **Right:** The spatial branch is enhanced to use the Semi Voxel-Focused Attention mechanism.

The decoder also consists of four stages. Each stage includes an EPA block (except for the last stage) and an Upsampling layer that performs deconvolution, increasing the resolution of the feature maps by a factor of 2 while reducing the channel dimension by the same factor. The encoder and decoder stages are linked through skip connections, which help recover spatial information lost during the downsampling operations.

In the final stage of the decoder, the output is fused with the convolution feature maps to restore spatial information and enhance feature representation. This output is then processed through a 3×3×3 ConvBlock and a 1×1×1 Convolution Block to generate the final prediction mask.

B. Voxel-Focused Attention (VFA)

Voxel-Focused Attention (VFA) is inspired by the Pixel-Focused Attention (PFA) mechanism introduced by Dai Shi [11]. The PFA mechanism aims to replicate the behaviour of biological foveal vision by using a query-centred sliding window for pixel-wise attention along with pooling attention. We extended this mechanism to operate voxel-wise for 3D medical image segmentation.

1) Approaches to VFA: We explored two approaches to implementing VFA:

- **Approach 1 - VFA**: This approach considers only the query-centred sliding window voxel-wise attention, without incorporating the pooling attention.
- **Approach 2 - Full VFA (FVFA)**: This approach extends the sliding window voxel-wise attention by including the pooling attention operation to capture additional contextual information.

Given a patch $X \in \mathbb{R}^{C\times H\times W\times D}$, we define a set of voxels within the sliding window as $v(i,j,m)$. The size of the window is expressed as:

$$|v(i,j,m)| = k \times k \times k, \tag{1}$$

where k denotes the size of the window along each dimension. Using this definition, the voxel-wise attention for Approach 1 can be represented as:

$$S_{(i,j,m)\sim p(i,j,m)} = (Q_{(i,j,m)} + QE) \cdot K^{\top}_{p(i,j,m)} \tag{2}$$

$$\alpha_{(i,j,m)\sim p(i,j,m)} = \text{Softmax}\left(\frac{S_{(i,j,m)\sim\sigma(i,j,m}}{\sqrt{d}} + B_{(i,j,m)\sim\rho(i,j,m)}\right) \tag{3}$$

$$\text{VFA}(X(i,j,m)) = \alpha_{(i,j,m)\sim p(i,j,m)} \cdot V_{(i,j,m)} \tag{4}$$

where:

- $S_{(i,j,m)\sim p(i,j,m)}$: Query-key similarity score
- $Q_{(i,j,m)}$: Query vector at location (i,j,m)
- QE: Query embedding
- $K_{p(i,j,m)}$: Key vector at position $p(i,j,m)$
- $\alpha_{(i,j,m)\sim p(i,j,m)}$: Attention weights
- d: Dimension of the query and key vectors
- $B_{(i,j,m)\sim\rho(i,j,m)}$: Learnable position bias
- $V_{(i,j,m)}$: Value vector

For Approach 2, we first define a set of voxels obtained from pooling the feature map as $\sigma(X)$. Given the pooled size as $H_p \times W_p \times D_p$, the voxel set size can be expressed as:

$$|\sigma(X)| = H_p \times W_p \times D_p. \tag{5}$$

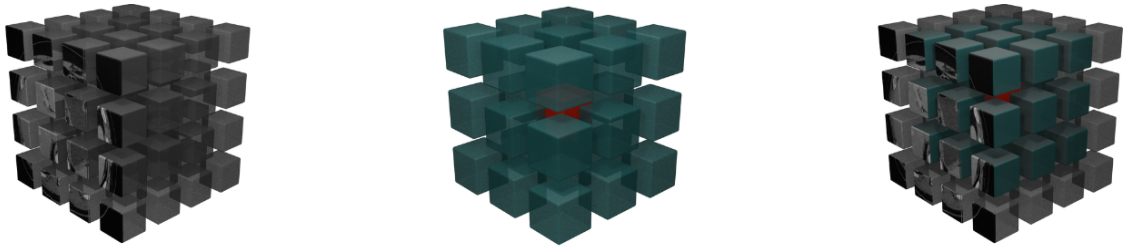

Fig. 4: Illustration of the 3D Sliding Window. The left side shows a 4 × 4 × 4 input feature. The middle image shows a 3 × 3 × 3 sliding window, with the centre voxel highlighted in red to indicate the query voxel, while the surrounding teal voxels represent key voxels, demonstrating the query-centred nature of the similarity computation. The right side presents an application of the 3D sliding window on the input feature.

The Voxel-Focused Attention for Approach 2 can be expressed as:

$$S_{(i,j,m)\sim p(i,j,m)} = (Q_{(i,j,m)} + \text{QE}) \cdot K^{\top}_{p(i,j,m)} \tag{6}$$

$$S_{(i,j,m)\sim \sigma(X)} = (Q_{(i,j,m)} + \text{QE}) \cdot K^{\top}_{\sigma(X)} \tag{7}$$

$$B_{(i,j,m)} = Concat(B_{(i,j,m)\sim\rho(i,j,m)}, \textbf{log-CPB}(\Delta_{(i,j,m)\sim\sigma(X)})) \tag{8}$$

$$\alpha_{(i,j,m)} = \text{Softmax}\Big(\tau \log(N) \cdot \text{Concat}\left[S_{(i,j,m)\sim p(i,j,m)}, S_{(i,j,m)\sim\sigma(X)}\right] + B_{(i,j,m)}\Big) \tag{9}$$

$$\alpha_{(i,j,m)\sim p(i,j,m)}, \alpha_{(i,j,m)\sim\sigma(X)} = \text{split}(\alpha_{(i,j,m)}) \tag{10}$$

$$\text{FVFA}(X(i,j,m)) = (\alpha_{(i,j,m)\sim p(i,j,m)} + Q_{(i,j,m)} \cdot T) \cdot V_{(i,j,m)} + \alpha_{(i,j,m)\sim\sigma(X)} \cdot V_{\sigma(X)} \tag{11}$$

where:

- $S_{(i,j,m)\sim p(i,j,m)}$: Query-key similarity score
- $S_{(i,j,m)\sim\sigma(X)}$: Pooled query-key similarity score
- $\alpha_{(i,j,m)}$: Attention weights
- QE: Query embedding
- $\sigma(X)$: Pooled feature map
- log-CPB: Log-spaced continuous position bias
- $\Delta_{(i,j,m)\sim\sigma(X)}$: Relative coordinates
- T: Learnable tokens

3D Padding Mask: Similar to the pixel-focused attention similarity computation in TransNeXt [11], the padding mask is used to prevent zero-padding voxels at the edges of the feature map from influencing the softmax operation. This is achieved by setting the results of any similarity computations involving these voxels to $-\infty$.

Motivated by the exceptional performance of aggregated attention mechanisms demonstrated by TransNeXt [11], several key features were incorporated to further enhance the model's effectiveness. These include the **Learnable Query Embedding (QE)**, which enables vision-language models to perform cross-attention between textual queries and visual keys, aiding tasks such as Visual Question Answering.

To enhance voxel-focused attention across multi-scale image inputs, **position bias** is calculated differently on two paths. On the pooling feature path, a **log-spaced continuous position bias (log-CPB)** is computed with a 2-layer multilayer perceptron (MLP) using relative coordinates $\Delta_{(i,j,m)\sim\sigma(X)}$ between $Q_{i,j,m}$ and $K_{i,j,m}$. On the sliding window path, a learnable position bias $B_{(i,j,m)\sim\rho(i,j,m)}$ is directly applied.

Additionally, **length-scaled cosine attention** is introduced to improve training stability and moderate attention weights by using cosine similarity instead of dot product attention. This method incorporates a learnable scaling factor, λ, that adjusts based on input sequence length, with λ expressed as $\lambda = \tau \log(N)$, where τ is a learnable variable and N represents the number of significant keys, excluding masked tokens.

Lastly, **positional attention**, or Query-Learnable-Value (QLV) attention, replaces static key-value pairs with learnable keys that adapt to each query. This dynamic approach improves positional information and enhances locality modelling, providing greater robustness than static methods. It introduces learnable tokens T for each attention head to generate adaptive position bias.

C. Efficient Paired-Attention Block

The EPA block can be broken into two branches: the spatial branch and the channel branch. Given an input volume $\boldsymbol{X}$, the branches can be computed as follows:

$$\boldsymbol{X}_s = \text{SA}(\boldsymbol{Q}_{\text{shared}}, \boldsymbol{K}_{\text{shared}}, \boldsymbol{V}_{\text{spatial}})$$

$$\boldsymbol{X}_c = \text{CA}(\boldsymbol{Q}_{\text{shared}}, \boldsymbol{K}_{\text{shared}}, \boldsymbol{V}_{\text{channel}})$$

where $\boldsymbol{X}_s$ denotes the spatial attention and $\boldsymbol{X}_c$ denotes the channel attention. SA is the spatial attention module, which is performed by the VFA operation, and CA is the channel attention module. $\boldsymbol{Q}_{\text{shared}}$ and $\boldsymbol{K}_{\text{shared}}$ are matrices for queries and keys that are shared between the branches. $\boldsymbol{V}_{\text{channel}}$ and $\boldsymbol{V}_{\text{spatial}}$ are matrices for the channel value and spatial value, respectively. Linear layers are used to generate these matrices as follows:

$$\begin{aligned}
\boldsymbol{Q}_{\text{shared}} &= \boldsymbol{W}_Q \cdot \boldsymbol{X} \\
\boldsymbol{K}_{\text{shared}} &= \boldsymbol{W}_K \cdot \boldsymbol{X} \\
\boldsymbol{V}_{\text{spatial}} &= \boldsymbol{W}_{V_s} \cdot \boldsymbol{X} \\
\boldsymbol{V}_{\text{channel}} &= \boldsymbol{W}_{V_c} \cdot \boldsymbol{X}
\end{aligned}$$

where the weight matrices $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_{V_s}$, and $\boldsymbol{W}_{V_c}$ are learnable parameters used to linearly project the input volume $\boldsymbol{X}$ into query, key, and value matrices for the spatial and channel branches. For Full Voxel-Focused Attention (FVFA), additional query and value matrices are added and utilised in the pooling path of FVFA, as shown in Figure 3. They can be expressed as follows:

$$\begin{aligned}
\boldsymbol{K}_{\text{pool}} &= \boldsymbol{W}_{K_p} \cdot \boldsymbol{\sigma}(\boldsymbol{X}) \\
\boldsymbol{V}_{\text{pool}} &= \boldsymbol{W}_{V_p} \cdot \boldsymbol{\sigma}(\boldsymbol{X})
\end{aligned}$$

where $\boldsymbol{W}_{K_p}$ and $\boldsymbol{W}_{V_p}$ are learnable weight matrices used to linearly project the pooled input volume $\boldsymbol{\sigma}(\boldsymbol{X})$.

The spatial branch performs spatial attention using Voxel-Focused Attention (VFA) or Full Voxel-Focused Attention (FVFA). The original UNETR++ architecture projects the spatial information into a lower dimension so that the attention computation is linear with respect to the input sequence. Both FVFA and VFA also achieve linear computation. Algorithm 1 shows the overall process that volumetric medical images undergo to produce segmentations using VFA and FVFA UNETR++.

III. Experimental Results and Discussion

A. Experimental setup

To evaluate the performance and efficiency of the enhanced UNETR++ model, three datasets were used: Synapse, ACDC, and BraTS.

- **Synapse**: This dataset consists of 30 abdominal CT scans with annotations for 8 target organs: spleen, right kidney, left kidney, gallbladder, liver, stomach, aorta, and pancreas.
- **ACDC**: The ACDC dataset contains 100 patients' cardiac MRI images, with annotations for 3 anatomical structures: right ventricle (RV), left ventricle (LV), and myocardium (MYO).

Algorithm 1 VFA and FVFA UNETR++ Segmentation Pseudocode

Input: 3D Medical Image Volume
Output: Segmented 3D Volume
1. Preprocessing:
- Resample the input volume to a target spacing.
- Apply data augmentations:
  - Rotation, scaling, Gaussian noise, Gaussian blur
  - Brightness and contrast adjustment, low-resolution simulation
  - Gamma adjustment, mirroring

2. Encoder:
- Split the input volume into non-overlapping 3D patches.
- Embed patches and add positional embeddings for spatial information.
- Pass embedded patches through multiple Transformer layers:
  - For each Transformer layer:
    - Perform layer normalisation.
    - Apply VFA or FVFA on the spatial dimension.
    - Apply channel attention mechanism.
    - Apply convolution.
    - Return features for each layer.

3. Decoder:
- Initialize the decoder with feature maps from the last Transformer layer.
- For each decoder stage (in reverse order of encoder stages):
  - Apply an EPA block with VFA or FVFA spatial attention.
  - Upsample feature maps from the previous layer.
  - Concatenate upsampled features with corresponding encoder features (skip connections).
- Repeat until reaching the original input resolution.

4. Output Layer:
- Apply a final 1x1x1 convolution to reduce channels to the number of segmentation classes.
- Apply a convolutional block to generate the final segmentation mask.

5. Postprocessing.
6. Return segmented 3D volume.

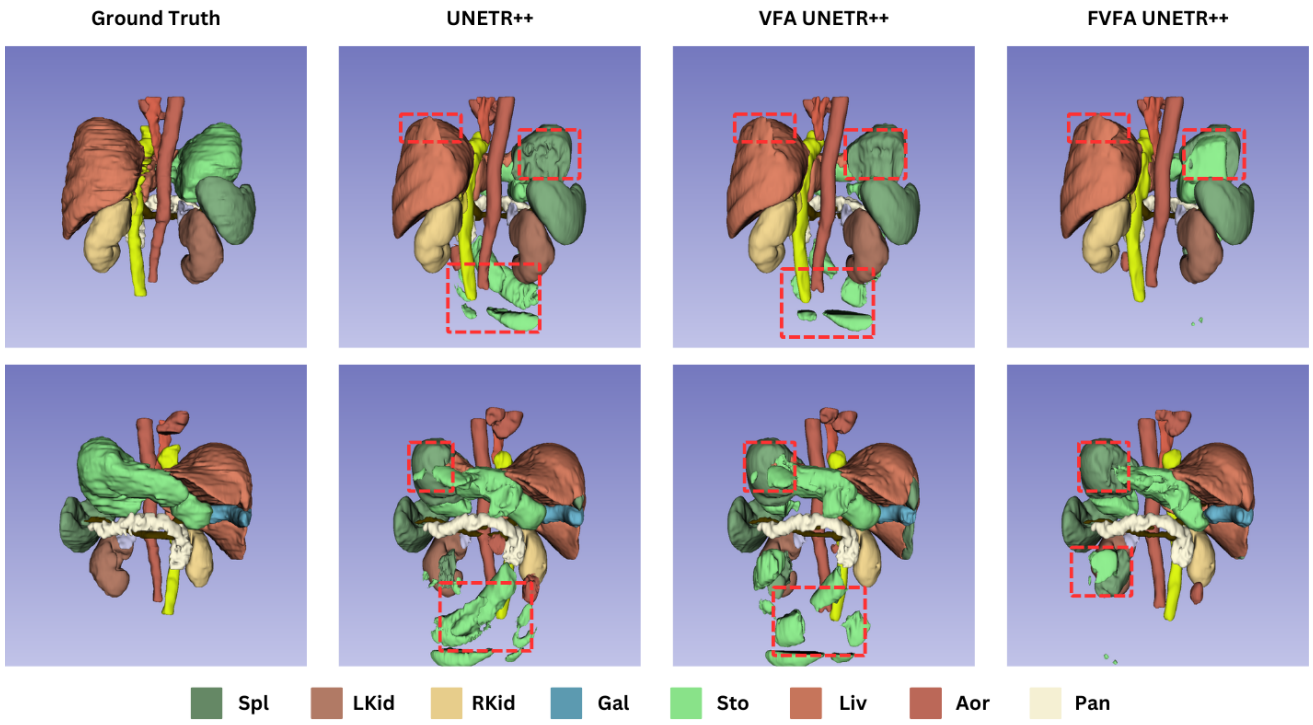

Fig. 5: Qualitative comparison of segmentation performance among UNETR++, VFA UNETR++, and FVFA UNETR++ on the Synapse dataset. The dashed red boxes highlight segmentation errors.

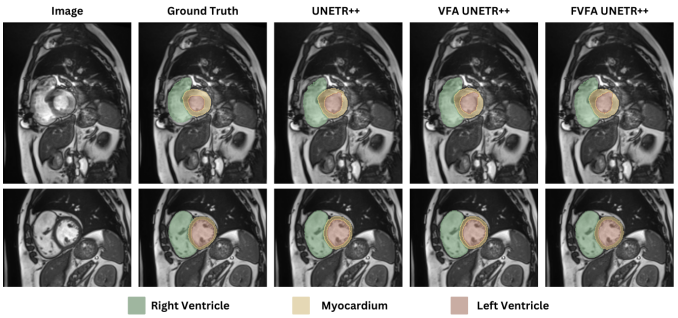

Fig. 6: Qualitative comparison of segmentation performance among UNETR++, VFA UNETR++, and FVFA UNETR++ on the ACDC dataset.

- **BraTS**: This dataset includes 484 MRI images with four modalities (FLAIR, T1w, T1gd, and T2w). The dataset is annotated for peritumoral oedema, GD-enhancing tumour, and necrotic/non-enhancing tumour core.

All images are first resampled to a uniform target spacing, followed by a series of augmentations, including rotation, scaling, Gaussian noise, Gaussian blur, brightness and contrast adjustment, low-resolution simulation, gamma augmentation, and mirroring, applied in sequence during training as in nnFormer [32]. To ensure consistent experimental conditions, the same preprocessing techniques used in the original UNETR++ experiment were applied, with training, validation, and test samples kept consistent across the original UNETR++ and nnFormer models, allowing for a fair comparison of model performance.

The data splits for each dataset were as follows:

- **Synapse**: 18 samples were used for training, and 12 samples were used for evaluation.
- **ACDC**: The dataset was split into a 70:10:20 ratio for training, validation, and testing, respectively.
- **BraTS**: The data was split into an 80:5:15 ratio for training, validation, and testing, respectively.

These consistent splits ensure a fair comparison of the models' performance across all datasets.

Evaluation Metrics: The Dice Similarity Coefficient (DSC) and 95% Hausdorff Distance (HD95) are the metrics used to measure the segmentation performance of the models. The **DSC** measures the overlap between two images: the prediction and the ground truth images. In the context of volumetric images, the DSC measures the overlap between the voxels of the segmentation prediction and the voxels of the ground truth. The DSC can be expressed as follows:

$$\text{DSC}(Y, P) = \frac{2 \cdot |Y \cap P|}{|Y| + |P|} \tag{12}$$

where Y and P denote ground truth voxels and predicted voxels, respectively.

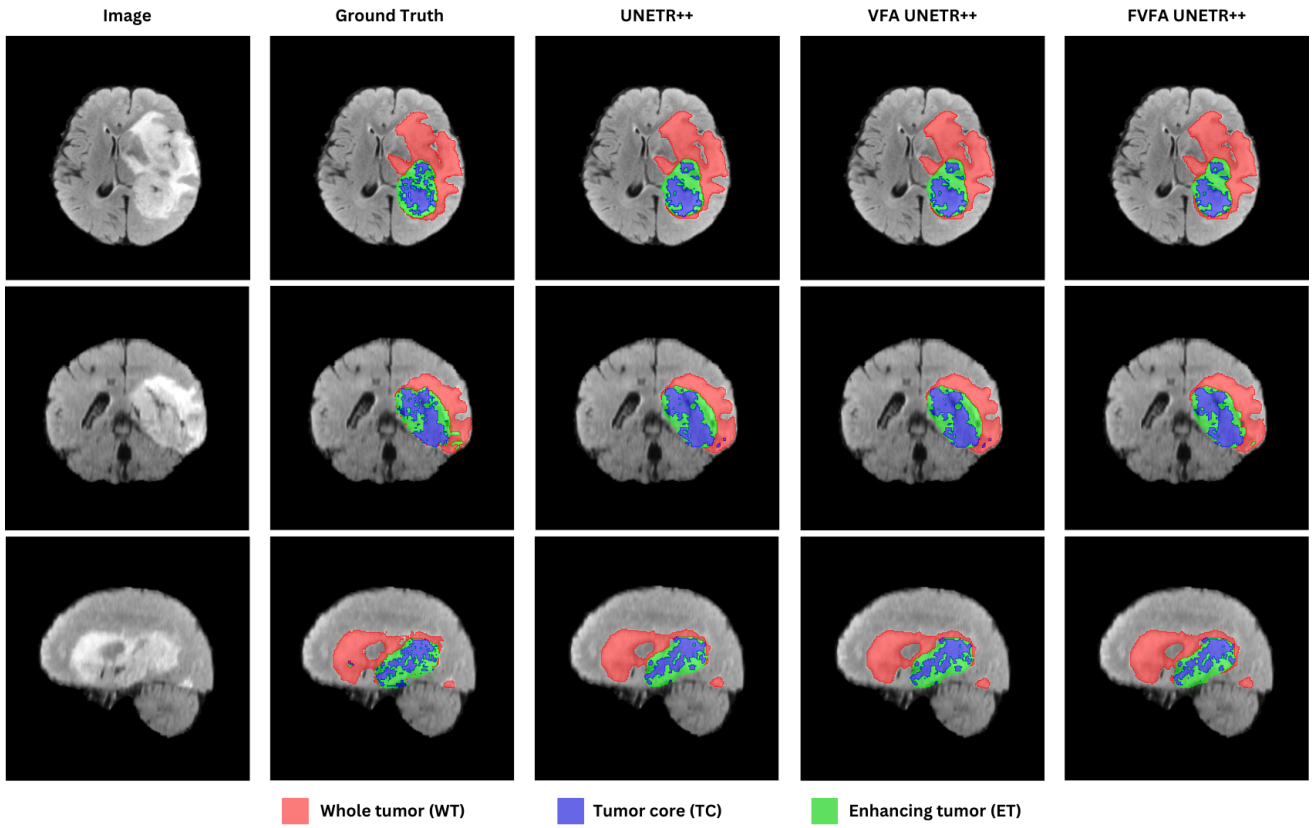

Fig. 7: Qualitative comparison of segmentation performance among UNETR++, VFA UNETR++, and FVFA UNETR++ on the BraTs dataset.

HD95 is a boundary-based metric that measures the 95th percentile of distances between the boundaries of the segmentation predictions and the boundaries of the ground truth. HD95 can be expressed as follows:

$$\text{HD95} = \max\left\{\sup_{a\in P} d(a, Y), \sup_{b\in Y} d(b, P)\right\} \tag{13}$$

where $d(a, Y)$ denotes the shortest distance from a point $a \in P$ to any point in Y, and $d(b, P)$ denotes the shortest distance from a point $b \in Y$ to any point in P. The boundaries of the prediction and ground truth are represented by P and Y, respectively.

Implementation Details: Both VFA UNETR++ and FVFA UNETR++ were implemented using Python version 3.8.19 and PyTorch 2.4.0, with the MONAI libraries also utilised. The models were trained on a single Nvidia V100 16GB (PCIe) GPU. For efficiency, a custom CUDA implementation was developed to compute the query-key similarity and aggregate the attention weights. The CUDA code was compiled using GCC version 9.2.0 and CUDA version 12.4. The Stochastic Gradient Descent (SGD) optimiser was used with an initial learning rate of 0.01, which was gradually decreased using a Polynomial Learning Rate Schedule at each epoch:

$$\text{lr} = \text{initial\_lr} \times \left(1 - \frac{\text{epoch\_id}}{\text{max\_epoch}}\right)^{0.9} \tag{14}$$

where:

- **lr**: Learning rate at the current epoch.
- **initial_lr**: Initial learning rate at the start of training.
- **epoch_id**: The current epoch number.
- **max_epoch**: The total number of epochs for training.
- **0.9**: The exponent that controls the decay rate.

The weight decay used across all datasets was set to 3e-5, and the default SGD momentum (0.99) was applied. The hyperparameters were kept consistent with those of UNETR++ to ensure a fair comparison. Each dataset was trained for 1000 epochs with varying input resolutions and patch resolutions: ACDC with an input resolution of 16 × 160 × 160 and a patch size of 1 × 4 × 4, Synapse with an input resolution of 64 × 128 × 128 and a patch size of 2 × 4 × 4, and BraTS with an input resolution of 128 × 128 × 128 and a patch size of 4 × 4 × 4. Data augmentation was performed in line with the methods used in UNETR, nnFormer, and the original UNETR++.

Methods	Params	FLOPs	Spl	RKid	LKid	Gal	Liv	Sto	Aor	Pan	Average	
											HD95 ↓	DSC ↑
U-Net [17]	-	-	86.67	68.60	77.77	69.72	93.43	75.58	89.07	53.98	-	76.85
TransUNet [30]	96.07	88.91	85.08	77.02	81.87	63.16	94.08	75.62	87.23	55.86	31.69	77.49
Swin-UNet [33]	-	-	90.66	79.61	83.28	66.53	94.29	76.60	85.47	56.58	21.55	79.13
UNETR [31]	92.49	75.76	85.00	84.52	85.60	56.30	94.57	78.00	89.60	60.47	18.59	78.35
MISSFormer [35]	-	-	91.92	82.00	85.21	68.65	94.41	80.81	89.06	65.67	18.20	81.96
Swin-UNETR [36]	62.83	384.2	**95.37**	86.26	86.99	66.54	95.72	77.01	91.12	68.80	10.55	83.48
nnFormer [32]	150.5	213.4	90.51	86.25	86.57	70.17	96.84	**86.83**	92.04	**83.35**	10.63	86.57
UNETR++ [12]	42.96	47.98	89.98	**90.29**	87.58	**73.20**	96.92	85.18	92.28	82.19	9.84	**87.20**
VFA UNETR++	28.6	**46.26**	88.68	87.77	**87.67**	69.61	**97.05**	84.55	**92.43**	81.49	10.49	86.16
FVFA UNETR++	**21.42**	52.27	89.56	87.39	87.33	72.96	96.71	85.56	92.56	81.68	**8.84**	86.72

TABLE I: Comparison of State-of-the-Art Methods on the Abdominal Multi-Organ Synapse Dataset, using Dice Similarity Coefficient (DSC) and Hausdorff Distance 95 (HD95) metrics. The target anatomical structures evaluated are: Spl (spleen), RKid (right kidney), LKid (left kidney), Gal (gallbladder), Liv (liver), Sto (stomach), Aor (aorta), and Pan (pancreas). Additional columns report the number of parameters (Params) and floating point operations per second (FLOPs). The best performance scores are highlighted in bold.

B. Comparison with state-of-the-art methods

1) Synapse dataset: Table I shows the segmentation performance on the Synapse dataset, including the number of parameters, FLOPs, Dice Score for each organ, and the average HD95 and Dice Score. The results are presented for 10 methods, which include both classical and state-of-the-art models. The results for UNETR++ were reproduced using the weights provided by the authors, while results for other methods were taken from their respective published papers. Both Voxel-Focused Attention (VFA) and Full Voxel-Focused Attention (FVFA) results are included. FVFA demonstrated competitive results in terms of Dice score while having fewer parameters than other models. However, FVFA has higher computational complexity (measured in FLOPs) and requires a longer training time compared to the original UNETR++ and VFA UNETR++. Figure 5: The qualitative comparison illustrates that FVFA UNETR++ achieves a better HD95 score, with fewer outliers, reduced voxel misclassification, and sharper boundary delineation.

2) ACDC dataset: Tables II and III present the segmentation results and computational efficiency of the ACDC dataset. FVFA UNETR++ achieves the highest performance while having the fewest parameters. However, it has a higher FLOP count and longer training time compared to the original UNETR++ and VFA UNETR++. FVFA UNETR++, VFA UNETR++, and UNETR++ outperform other state-of-the-art methods in terms of average DSC score. FVFA UNETR++ has the lowest number of parameters, with 41% fewer than UNETR++, while VFA has 35% fewer parameters compared to UNETR++. VFA also has lower FLOPs than the other three models. Figure 6 shows the qualitative comparison of segmentations produced by the three models. The differences between the segmentations are minimal, as the models have very close segmentation performance, with a maximum DSC difference of 0.19. However, VFA UNETR++ is more efficient overall due to improvements in the number of parameters and FLOPs.

Methods	RV	Myo	LV	Average (DSC)
TransUNet [30]	88.86	84.54	95.73	89.71
Swin-UNet [33]	88.55	85.62	95.83	90.00
UNETR [31]	85.29	86.52	94.02	86.61
MISSFormer [35]	86.36	85.75	91.59	87.90
nnFormer [32]	90.94	89.58	95.65	92.06
UNETR++ [12]	**91.89**	90.61	96.00	92.83
VFA UNETR++	91.45	90.59	96.10	92.71
FVFA UNETR++	91.75	**90.74**	**96.21**	**92.90**

TABLE II: State-of-the-art comparison on the ACDC dataset using Dice Similarity Coefficient scores.

Methods	Params (M)	FLOPs
UNETR++ [12]	66.8	43.71
VFA UNETR++	44.36	**42.4**
FVFA UNETR++	**39.03**	47.54

TABLE III: Number of parameters and FLOPs for UNETR++, VFA UNETR++, and FVFA UNETR++.

3) BraTs dataset: The experimental results for tumour segmentation are presented in Tables IV and V, which includes both segmentation performance and computational efficiency. UNETR++, VFA UNETR++, and FVFA UNETR++ demonstrate similar performance, with a difference of less than 0.4 DSC between them. VFA UNETR++ achieved the highest DSC score of 82.8, with the original UNETR++ trailing by only 0.1 DSC. FVFA UNETR++ had the lowest performance on the BraTS dataset compared to the other two models. All three models outperform other state-of-the-art models not only in performance but also in terms of the number of parameters, FLOPs, and memory usage (except FVFA UNETR++). FVFA UNETR++ has a higher memory usage than UNETR. FVFA has the lowest number of parameters compared to all the models (21.4), which is

Methods	Params	FLOPs	Mem	DSC (%)
UNETR [31]	92.5	153.5	3.3	81.2
SwinUNETR [36]	62.8	572.4	19.7	81.5
nnFormer [32]	149.6	421.5	12.6	82.3
UNETR++ [12]	42.6	**70.1**	**2.7**	82.7
VFA UNETR++	28.6	72.29	3.1	**82.8**
FVFA UNETR++	**21.4**	78.3	5.8	82.4

TABLE IV: Comparison of state-of-the-art methods on the BraTS dataset, including the number of parameters, FLOPs, Memory (GB), and Dice Similarity Coefficient Score.

Methods	Training Time	Inference Time	HD95
UNETR++ [12]	**244.14**	5.69	5.27
VFA UNETR++	245.21	**5.28**	**5.01**
FVFA UNETR++	248.58	7.94	5.08

TABLE V: Average HD95, training time, and inference time for the original, VFA, and FVFA UNETR++ in seconds.

a 50% reduction compared to the original UNETR++. Among the three models, UNETR++ has the shortest average epoch training time of 244.14 seconds, while FVFA UNETR++ has the longest, with a difference of only 4.44 seconds. VFA UNETR++ has the fastest inference time and the highest HD95 score. FVFA UNETR++ has the longest inference time of 7.94, which is 2.66 seconds longer than that of VFA UNETR++. Figure 7 shows the segmentations of the three models for qualitative analysis.

IV. Comparison of CUDA C++ and Native Python-only Implementations

The computational efficiency of the CUDA and Python-only implementations is presented in Tables VI and VII. Table VI shows results for FVFA UNETR++, while Table VII shows results for VFA UNETR++. The CUDA implementation outperforms the Python-only implementation in training time, inference speed, and memory usage for both VFA and FVFA UNETR++. For FVFA UNETR++ inference time, the CUDA implementation is 12.36% faster than the Python-only version. Similarly, the CUDA implementation for VFA UNETR++ achieves a 15.52% faster inference speed compared to the Python-only version. The training time difference is smaller, with the CUDA implementation being 5% faster for FVFA UNETR++ and 1% faster for VFA UNETR++. However, the Python-only implementation of VFA UNETR++ is faster than the CUDA implementation of FVFA UNETR++ for inference. In terms of memory efficiency, the CUDA implementation significantly reduces GPU memory usage, with FVFA UNETR++ using 22.67% less GPU memory and VFA UNETR++ using 26.19% less.

Implementations	Training Time (s)	Inference Time (s)	Mem
Python	261.15	9.06	7.5
CUDA	248.58	7.94	5.8

TABLE VI: Comparison of the computational efficiency between CUDA and Python-only implementations for the Full Voxel-Focused Attention (FVFA) UNETR++ model.

Implementations	Training Time (s)	Inference Time (s)	Mem
Python	248.37	6.25	4.2
CUDA	245.21	5.28	3.1

TABLE VII: Comparison of the computational efficiency between CUDA and Python-only implementations for the Voxel-Focused Attention (VFA) model.

V. Conclusion

This research propose two competitive models, the Voxel-Focused Attention UNETR++ (VFA UNETR++) and Full Voxel-Focused Attention UNETR++ (FVFA UNETR++), which extend pixel-focused attention and adapt aggregated attention from TransNeXt for 3D medical image segmentation. The experimental results demonstrated the competitiveness of the proposed models in terms of DSC, HD95, and computational efficiency. Both models effectively learn from the data while utilising fewer parameters, all while maintaining strong segmentation performance and computational efficiency.

References

[1] National Health Service, "Magnetic resonance imaging (mri) scan," 2022. Retrieved from Health A to Z.

[2] National Health Service, "Ct scan," 2023. Retrieved from Health A to Z.

[3] S. Padhi, S. Rup, S. Saxena, and F. Mohanty, "Mammogram segmentation methods: A brief review," in *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pp. 218–223, 2019.

[4] K. Doi, "Computer-aided diagnosis in medical imaging: Historical review, current status and future potential," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4, pp. 198–211, 2007. Computer-aided Diagnosis (CAD) and Image-guided Decision Support.

[5] Y. Chen, Z. Zhuang, and C. Chen, "Object detection method based on pvtv2," in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, pp. 730–734, 2023.

[6] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, p. 415–424, Mar. 2022.

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[8] J. Yang, C. Li, X. Dai, and J. Gao, "Focal modulation networks," in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 4203–4217, Curran Associates, Inc., 2022.

[9] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," 2022.

[10] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," 2023.

[11] D. Shi, "Transnext: Robust foveal visual perception for vision transformers," 2024.

[12] A. M. Shaker, M. Maaz, H. Rasheed, S. Khan, M.-H. Yang, and F. S. Khan, "Unetr++: Delving into efficient and accurate 3d medical image segmentation," *IEEE Transactions on Medical Imaging*, pp. 1–1, 2024.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[14] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. Gonzalez Ballester, G. Sanroma, S. Napel, S. Petersen, G. Tziritas, E. Grinias, M. Khened, V. A. Kollerathu, G. Krishnamurthi, M.-M. Rohe, X. Pennec, M. Sermesant, F. Isensee, P. Jager, K. H. Maier-Hein, P. M. Full, I. Wolf, S. Engelhardt, C. F. Baumgartner, L. M. Koch, J. M. Wolterink, I. Isgum, Y. Jang, Y. Hong, J. Patravali, S. Jain, O. Humbert, and P.-M. Jodoin, "Deep learning techniques for automatic MRI cardiac Multi-Structures segmentation and diagnosis: Is the problem solved?," *IEEE Trans Med Imaging*, vol. 37, pp. 2514–2525, May 2018.

[15] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Trans Med Imaging*, vol. 34, pp. 1993–2024, Dec. 2014.

[16] B. Landman, Z. Xu, J. Igelsias, M. Styner, T. Langerak, and A. Klein, "Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge," in *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, vol. 5, p. 12, 2015.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), (Cham), pp. 234–241, Springer International Publishing, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[19] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," 2020.

[20] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, eds.), (Cham), pp. 424–432, Springer International Publishing, 2016.

[21] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," 2018.

[22] C. Angermann and M. Haltmeier, "Random 2.5 d u-net for fully 3d segmentation," in *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting: First International Workshop, MLMECH 2019, and 8th Joint International Workshop, CVII-STENT 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1*, pp. 158–166, Springer, 2019.

[23] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, and K. H. Maier-Hein, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," 2018.

[24] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022.

[25] S. Roy, G. Koehler, C. Ulrich, M. Baumgartner, J. Petersen, F. Isensee, P. F. Jaeger, and K. H. Maier-Hein, "Mednext: transformer-driven scaling of convnets for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 405–415, Springer, 2023.

[26] H. Li, Y. Nan, and G. Yang, "Lkau-net: 3d large-kernel attention-based u-net for automatic mri brain tumor segmentation," in *Medical Image Understanding and Analysis* (G. Yang, A. Aviles-Rivero, M. Roberts, and C.-B. Schönlieb, eds.), (Cham), pp. 313–327, Springer International Publishing, 2022.

[27] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5436–5447, 2022.

[28] T. Huang, L. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Lightvit: Towards light-weight convolution-free vision transformers," *arXiv preprint arXiv:2207.05557*, 2022.

[29] D. Karimi, S. D. Vasylechko, and A. Gholipour, "Convolution-free medical image segmentation using transformers," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, eds.), (Cham), pp. 78–88, Springer International Publishing, 2021.

[30] 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," 2021.

[31] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 574–584, January 2022.

[32] H.-Y. Zhou, J. Guo, Y. Zhang, L. Yu, L. Wang, and Y. Yu, "nnformer: Interleaved transformer for volumetric segmentation," *arXiv preprint arXiv:2109.03201*, 2021.

[33] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*, pp. 205–218, Springer, 2022.

[34] Y. Liu, Z. Zhang, J. Yue, and W. Guo, "Scanext: Enhancing 3d medical image segmentation with dual attention network and depth-wise convolution," *Heliyon*, vol. 10, no. 5, 2024.

[35] X. Huang, Z. Deng, D. Li, and X. Yuan, "Missformer: An effective medical image segmentation transformer," *arXiv preprint arXiv:2109.07162*, 2021.

[36] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images," in *International MICCAI brainlesion workshop*, pp. 272–284, Springer, 2021.