

# nnFormer: Interleaved Transformer for Volumetric Segmentation

Hong-Yu Zhou<sup>\*1,2</sup>, Jiansen Guo<sup>\*1</sup>, Yinghao Zhang<sup>\*1</sup>, Lequan Yu<sup>3</sup>,  
Liansheng Wang<sup>1</sup>, and Yizhou Yu<sup>2</sup>

<sup>1</sup>Department of Computer Science, Xiamen University

<sup>2</sup>Department of Computer Science, The University of Hong Kong

<sup>3</sup>Department of Statistics and Actuarial Science, The University of Hong Kong  
whuzhouhongyu@gmail.com, {jsguo,zhangyinghao}@stu.xmu.edu.cn,  
lqyu@hku.hk, lswang@xmu.edu.cn, yizhouy@acm.org

**Abstract.** Transformers, the default model of choices in natural language processing, have drawn scant attention from the medical imaging community. Given the ability to exploit long-term dependencies, transformers are promising to help atypical convolutional neural networks (convnets) to overcome its inherent shortcomings of spatial inductive bias. However, most of recently proposed transformer-based segmentation approaches simply treated transformers as assisted modules to help encode global context into convolutional representations without investigating how to optimally combine self-attention (i.e., the core of transformers) with convolution. To address this issue, in this paper, we introduce nnFormer (i.e., not-another transFormer), a powerful segmentation model with an interleaved architecture based on empirical combination of self-attention and convolution. In practice, nnFormer learns volumetric representations from 3D local volumes. Compared to the naive voxel-level self-attention implementation, such volume-based operations help to reduce the computational complexity by approximate 98% and 99.5% on Synapse and ACDC datasets, respectively. In comparison to prior-art network configurations, nnFormer achieves tremendous improvements over previous transformer-based methods on two commonly used datasets Synapse and ACDC. For instance, nnFormer outperforms Swin-UNet by over 7 percents on Synapse. Even when compared to nnUNet, currently the best performing fully-convolutional medical segmentation network, nnFormer still provides slightly better performance on Synapse and ACDC. Codes and models are available at <https://github.com/282857341/nnFormer>.

**Keywords:** First keyword · Second keyword · Another keyword.

## 1 Introduction

Transformers [33], which are the de-facto choices for natural language processing (NLP) problems, have recently been widely exploited in vision-based applications [8, 20]. The core idea behind is to apply the self-attention mechanism

---

<sup>\*</sup> This work is done when Hong-Yu Zhou was visiting Xiamen University.  
First three authors contributed equally.

to capture long-range dependencies. Compared to convnets (i.e., convolutional neural networks [14]), transformers relax the inductive bias of locality, making them more capable of dealing with non-local interactions [42]. It has also been investigated that the prediction errors of transformers are more consistent with those of humans than convnets [31].

Given the fact that transformers are naturally more advantageous than convnets, there are a number of approaches trying to apply transformers to the field of medical image analysis. [6] first time proposed TransUNet to explore the potential of transformers in the context of medical image segmentation. The overall architecture of TransUNet is similar to that of U-Net [27], where convnets act as feature extractors and transformers help encode the global context. In fact, a major feature of TransUNet and most of its followers [4, 5, 32, 41] is to treat convnets as main bodies, on top of which transformers are further applied to capture long-term dependencies. However, such characteristic may cause a problem: advantages of transformers are not fully exploited. In other words, we believe one- or two-layer transformers are not enough to entangle long-term dependencies with convolutional representations that often contain precise spatial information and provide hierarchical concepts.

To address the above issue, some researchers [3, 12, 18] started to use transformers as the main stem of segmentation models. [12] first time introduced a convolution-free segmentation model by forwarding flattened image representations to transformers, whose outputs are then reorganized into 3D tensors to align with segmentation masks. Recently, Swin Transformer [20] showed that by referring to the feature pyramids used in convnets, transformers can learn hierarchical object concepts at different scales by applying appropriate down-sampling to feature maps. Inspired by this idea, Swin-UNet [3] utilized hierarchical transformer blocks to construct the encoder and decoder within a U-Net like architecture, based on which DS-TransUNet [18] added one more encoder to accept different-sized inputs. Both Swin-UNet and DS-TransUNet have achieved consistent improvements over TransUNet. Nonetheless, they did not explore how to appropriately combine convolution and self-attention for building an optimal medical segmentation network.

The main contribution of nnFormer (i.e., **not-another transFormer**) is its hybrid stem where convolution and self-attention are interleaved to give full play to their strengths. Figure 1 presents the effects of different components used in the encoder of nnFormer. Firstly, we put a light-weight convolutional embedding layer ahead of transformer blocks. In comparison to directly flattening raw pixels and applying 1D pre-processing in [12], the convolutional embedding layer encodes precise (i.e., pixel-level) spatial information and provide low-level yet high-resolution 3D features. After the embedding block, transformer and convolutional down-sampling blocks are interleaved to fully entangle long-term dependencies with high-level and hierarchical object concepts at various scales, which helps improve the generalization ability and robustness of learned representations.

The other contribution of nnFormer lies in proposing a computational-efficient way to capture inter-slice dependencies. To be specific, nnFormer introduces

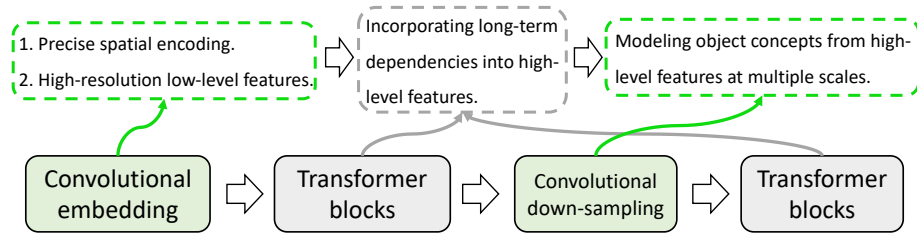


Fig. 1: Overview of the interleaved stem used in the encoder of nnFormer.

volume-based multi-head self-attention (V-MSA) to learn representations on 3D local volumes, which are then aggregated to produce whole-volumetric predictions. Compared to the naive multi-head self-attention (MSA) [33], V-MSA is able to reduce the computational complexity by about 98% and 99.5% in transformer blocks on Synapse and ACDC datasets, respectively.

In the experiment section, we compare nnFormer with a wide range of baseline segmentation approaches. The proposed nnFormer surpasses Swin-UNet by over 7 percents in the task of multi-organ segmentation on Synapse. When performing automated cardiac diagnosis on ACDC dataset, nnFormer outperforms Swin-UNet by nearly 2 percents in average. Considering the average dice score on ACDC is over 90 percents, we believe 2-percent improvements on ACDC are as impressive as the 7-percent improvements on Synapse.

## 2 Related work

In this section, we mainly review methodologies that resort to transformers to improve segmentation results of medical images. Since most of them employ hybrid architectures of convolution and self-attention [33], we divide them into two categories based on whether the majority of the stem is convolutional or transformer-based.

**Convolution-based stem.** TransUNet [6] first time applied transformer to improve the segmentation results of medical images. TransUNet treats the convnet as a feature extractor to generate a feature map for the input slice. Patch embedding is then applied to patches of feature maps in the bottleneck instead of raw images in ViT [8]. Concurrently, similar to TransUNet, Li *et al.* [16] proposed to use a squeezed attention block to regularize the self-attention modules of transformers and an expansion block to learn diversified representations for fundus images, which are all implemented in the bottleneck within convnets. TransFuse [41] introduced a BiFusion module to fuse features from the shallow convnet-based encoder and transformer-based segmentation network to make final predictions on 2D images. Compared to TransUNet, TransFuse mainly applied the self-attention mechanism to the input embedding layer to improve segmentation models on 2D images. Yun *et al.* [39] employed transformers to incorporate spectral information,

which are entangled with spectral information encoded by convolutional features to address the problem of hyperspectral pathology. Xu *et al.* [38] extensively studied the trade-off between transformers and convnets and proposed a more efficient encoder named LeViT-UNet. Li *et al.* [17] presented a new up-sampling approach and incorporated it into the decoder of UNet to model long-term dependencies and global information for better reconstruction results. TransClaw U-Net [4] utilized transformers in UNet with more convolutional feature pyramids. TransAttUNet [5] explored the feasibility of applying transformer self attention with convolutional global spatial attention. Xie *et al.* [37] adopted transformers to capture long-term dependencies of multi-scale convolutional features from different layers of convnets. TransBTS [34] first utilized 3D convnets to extract the volumetric spatial features and down-sample the input 3D images to produce hierarchical representations. The outputs of the encoder in TransBTS are then reshaped into a vector (i.e. token) and fed into transformers for global feature modeling, after which an ordinary convolutional decoder is appended to up-sample feature maps for the goal of reconstruction. Different from these approaches that directly employ convnets as feature extractors, our nnFormer functionally relies on convolutional and transformer-based blocks, which are interleaved to take advantages of each other.

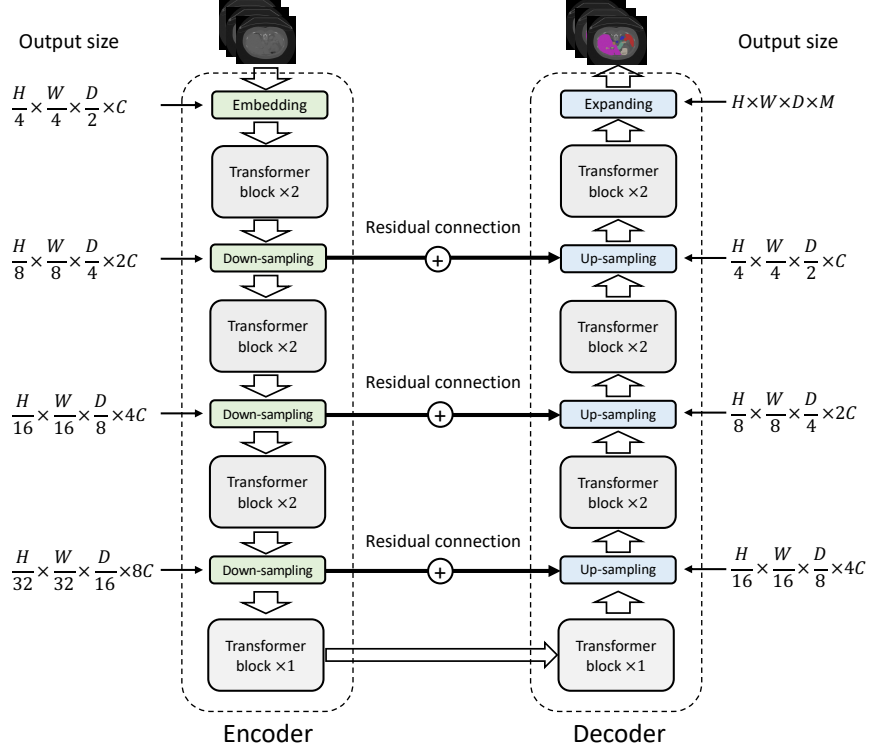
**Transformer-based stem.** Valanarasu *et al.* [32] proposed a gated axial-attention model (i.e., MedT) which extends the existing convnet architectures by introducing an additional control mechanism in the self-attention. Karimi *et al.* [12] removed the convolutional operations and built a 3D segmentation model based on transformers. The main idea is to first split the local volume block into 3D patches, which are then flattened and embedded to 1D sequences and passed to a ViT-like backbone to extract representations. Swin-UNet [3] built a U-shape transformer-based segmentation model on top of transformer blocks in [20], where observable improvements were achieved. DS-TransUNet [18] further extended Swin-UNet by adding one more encoder to handle multi-scale inputs and introduced a fusion module to effectively establish global dependencies between features of different scales through the self-attention mechanism. Compared to these transformer-based stems, nnFormer inherits the superiority of convolution in encoding precise spatial information and producing hierarchical representations that help model object concepts at various scales.

### 3 Method

#### 3.1 Overview

The overall architecture of nnFormer is presented in Figure 2, which maintains a similar U shape as that of U-Net [27] and mainly consists of two branches, i.e., the encoder and decoder. Concretely, the encoder involves one embedding block, seven transformer blocks and three down-sampling blocks. Symmetrically, the decoder branch includes seven transformer blocks, three up-sampling blocks and one patch expanding block for making final predictions. Inspired by U-Net [27],

a.



b.

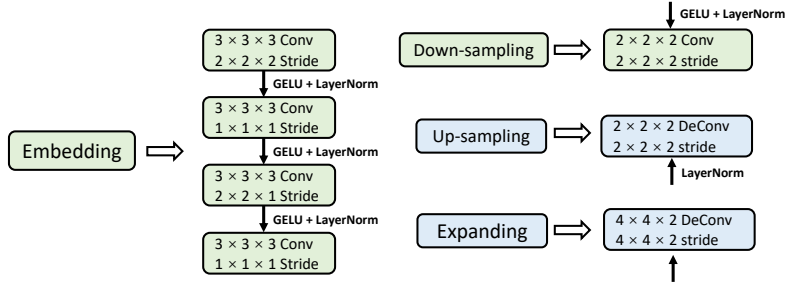


Fig. 2: Architecture of nnFormer. In (a), we show the overall architecture of nnFormer. In (b), we present more details of the embedding, down-sampling, up-sampling and the last expanding blocks.  $M$  denotes the number of segmentation classes. Note that the displayed architecture is applied to images from Synapse dataset. In practice, the architecture may slightly vary depending on the input patch size.

we add long residual connections between corresponding feature pyramids of the encoder and decoder in a symmetrical manner, which helps to recover fine-grained details in the prediction.

### 3.2 Encoder

The input of nnFormer is a 3D patch  $\mathcal{X} \in \mathcal{R}^{H \times W \times D}$  (usually randomly cropped from the original image), where  $H$ ,  $W$  and  $D$  denote the height, width and depth of each input patch, respectively.

**Embedding block.** The embedding block is responsible for transforming each input scan  $\mathcal{X}$  into a high-dimensional tensor  $\mathcal{X}_e \in \mathcal{R}^{\frac{H}{4} \times \frac{W}{4} \times \frac{D}{2} \times C}$ , where  $\frac{H}{4} \times \frac{W}{4} \times \frac{D}{2}$  represents the number of the patch tokens and  $C$  represents the sequence length. In practice, we set  $C$  to 192 and 96 on Synapse and ACDC datasets, respectively. Different from ViT [8] and Swin Transformer [20] that use large convolutional kernels in the embedding block to extract features, we found that applying SUCCESSIVE convolutional layers with SMALL convolutional kernels bring more benefits in the initial stage, which could be explained from two perspectives, i.e., i) *why applying successive convolutional layers* and ii) *why using small-sized kernels*. For i), we use convolutional layers in the embedding block because they encode pixel-level spatial information, more precisely than patch-wise positional encoding used in transformers. For ii), compared to large-sized kernels, small kernel sizes help reduce computational complexity while providing equal-sized receptive field. As shown in Figure 2b, the embedding block consists of four convolutional layers whose kernel size is 3. After each convolutional layer (except the last one), one GELU [10] and one layer normalization (i.e., LayerNorm) [1] layers are appended. In practice, depending on the size of input patch, strides of convolution in the embedding block may accordingly vary.

**Transformer block.** After the embedding block, we pass the high-dimensional tensor  $\mathcal{X}_e$  to interleaved transformer blocks. The main point behind is to fully entangle the captured long-term dependencies with hierarchical object concepts at various scales provided by following down-sampling convolution and high-resolution spatial information encoded by the initial embedding block. Compared to Swin Transformer [20], we employ a hierarchical way to conduct self-attention but compute self-attention within 3D local volumes (i.e., V-MSA, volume-based multi-head self-attention) instead of 2D local windows.

Supposing  $\mathcal{X}_t \in \mathcal{R}^{L \times C}$  being the input of transformer blocks, it would be first reshaped to  $\hat{\mathcal{X}}_t \in \mathcal{R}^{N_V \times N_T \times C}$ , where  $N_V$  is the number of 3D local volumes and  $N_T = S_H \times S_W \times S_D$  denotes the number of patch tokens in each volume.  $\{S_H, S_W, S_D\}$  stand for the size of local volume. In nnFormer, to adapt to various shape of MRI/CT scans, we design  $\{S_H, S_W, S_D\}$  to make it cover all patch tokens of the output of the last transformer block in the encoder. The intuition behind is that it may not be desirable to brute-force pad the data in order to satisfy fixed  $\{S_H, S_W, S_D\}$ . Thus, the size of the input cropped patch  $\mathcal{X}$  needs to adaptively adjusted in order to accord with the size of local

volumes. In practice, we set  $\{S_H, S_W, S_D\}$  on Synapse and ACDC to  $\{4, 4, 4\}$  and  $\{5, 5, 3\}$ , respectively.

We follow [20] to conduct two successive transformer blocks, where the main difference lies in that our computation is built on top of 3D volumes instead of 2D windows. The computational procedure can be summarized as follows:

$$\begin{aligned}\hat{\mathcal{X}}_t^l &= \text{V-MSA} \left( \text{LayerNorm} \left( \mathcal{X}_t^{l-1} \right) \right) + \mathcal{X}_t^{l-1}, \\ \mathcal{X}_t^l &= \text{MLP} \left( \text{LayerNorm} \left( \hat{\mathcal{X}}_t^l \right) \right) + \hat{\mathcal{X}}_t^l, \\ \hat{\mathcal{X}}_t^{l+1} &= \text{SV-MSA} \left( \text{LayerNorm} \left( \mathcal{X}_t^l \right) \right) + \mathcal{X}_t^l, \\ \mathcal{X}_t^{l+1} &= \text{MLP} \left( \text{LayerNorm} \left( \hat{\mathcal{X}}_t^{l+1} \right) \right) + \hat{\mathcal{X}}_t^{l+1}.\end{aligned}\tag{1}$$

Here,  $l$  stands for the layer index. MLP is an abbreviation for multi-layer perceptron. V-MSA and SV-MSA denote the volume-based multi-head self-attention and its shifted version. The computational complexity of V-MSA on a volume of  $h \times w \times d$  patches is:

$$\Omega(\text{V-MSA}) = 4hwdC^2 + 2S_H S_W S_D hwdC.\tag{2}$$

Compared to the complexity of the naive multi-head self-attention (MSA) [33] used in ViT [8], i.e.,

$$\Omega(\text{MSA}) = 4hwdC^2 + 2(hwd)^2C.\tag{3}$$

V-MSA reduces the computational complexity by approximate 98% and 99.5% on Synapse and ACDC datasets, respectively.

SV-MSA displaces the 3D local volume used in V-MSA by  $(\lfloor \frac{S_H}{2} \rfloor, \lfloor \frac{S_W}{2} \rfloor, \lfloor \frac{S_D}{2} \rfloor)$  to introduce more interactions between different local volumes.

The query-key-value (QKV) attention [33] in each 3D local volume can be computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} + B \right) V,\tag{4}$$

where  $Q, K, V \in \mathcal{R}^{N_T \times d_k}$  denote the query, key and value matrices of dimension  $d_k$ .  $B \in \mathcal{R}^{N_T}$  is the relative position encoding. In practice, we first initialize a smaller-sized position matrix  $\hat{B} \in \mathcal{R}^{(2S_H-1) \times (2S_W-1) \times (2S_D-1)}$  and take corresponding values from  $\hat{B}$  to build a larger position matrix  $B$ .

It is worth noting that the last transformer block of the encoder (the down-most block in Figure 2) only employs V-MSA as we found introducing SV-MSA would deteriorate the overall segmentation results.

**Down-sampling block.** We found that by replacing the neighboring concatenation operation in [20] with direct strided convolution, nnFormer provides more improvements to volumetric segmentation. The intuition behind is that convolutional down-sampling produces hierarchical representations that help model

object concepts at multiple scales. As displayed in Figure 2b, in most cases, the down-sampling block involves a strided convolution operation where the stride is set to 2 in all dimensions. However, in practice, the stride with respect to specific dimension (refer to Table 1b) is set to 1 as the number of slices is limited in this dimension and over-down-sampling (i.e., a large stride) might be harmful.

### 3.3 Decoder

Architectures of transformer blocks of the decoder are highly symmetrical to those of the encoder. In contrast to the down-sampling blocks, we employ strided deconvolution to up-sample low-resolution feature maps to high-resolution ones, which in turn are merged with representations from the encoder via long-range residual connections to capture both semantic and fine-grained information. Similar to up-sampling blocks, the last patch expanding block also takes the deconvolutional operation to produce final predictions.

## 4 Experiments

To fairly compare nnFormer with previous Transformer-based architectures, we conduct experiments on Synapse [13] and Automatic Cardiac Diagnosis Challenge (ACDC) [2] datasets. For each experiment, we repeat it for three times and report their average results.

**Synapse for multi-organ CT segmentation.** This dataset includes 30 cases of abdominal CT scans. Following the split used in [6], 18 cases are extracted to build the training set while the rest 12 cases are used for testing. We report the model performance evaluated with the average Dice Similarity Coefficient (DSC) on 8 abdominal organs, which are aorta, gallbladder, spleen, left kidney, right kidney, liver, pancreas and stomach.

**ACDC for automated cardiac diagnosis.** ACDC involves 100 patients, with the cavity of the right ventricle, the myocardium of the left ventricle and the cavity of the left ventricle to be segmented. Each case’s labels involve left ventricle (LV), right ventricle (RV) and myocardium (MYO). The dataset is split into 70 training samples, 10 validation samples and 20 testing samples.

### 4.1 Implementation details

We run all experiments based on Python 3.6, PyTorch 1.8.1 and Ubuntu 18.04. All training procedures have been performed on a single NVIDIA 2080 GPU with 11GB memory. The initial learning rate is set to 0.01 and we employ a “poly” decay strategy as described in Equation 5. The default optimizer is SGD where we set the momentum to 0.99. The weight decay is set to  $3e-5$ . We utilize both cross entropy loss and dice loss by simply summing them up. The number of



Table 1: Network configurations of our nnFormer and nnUNet on Synapse and ACDC. We only report the stride of down-sampling (i.e. str. of down-samp.) as the stride of up-sampling can be easily inferred according to symmetrical down-sampling operations. Note that the network configuration of nnUNet is automatically determined based on pre-defined hand-crafted rules (for self-adaptation).

(a) Synapse		
	nnFormer	nnUNet
Target spacing	[0.76, 0.76, 3]	[0.76, 0.76, 3]
Median image shape	$512 \times 512 \times 148$	$512 \times 512 \times 148$
Crop size	$128 \times 128 \times 64$	$192 \times 192 \times 48$
Batch size	2	2
Str. of Down-samp.	[2, 2, 2], [2, 2, 1], [2, 2, 2], [2, 2, 1], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]	[2, 2, 2], [2, 2, 1]
(b) ACDC		
	nnFormer	nnUNet
Target spacing	[1.52, 1.52, 6.35]	[1.52, 1.52, 6.35]
Median image shape	$246 \times 213 \times 13$	$246 \times 213 \times 13$
Crop size	$160 \times 160 \times 14$	$256 \times 224 \times 14$
Batch size	4	4
Str. of Down-samp.	[2, 2, 1], [2, 2, 1], [2, 2, 1], [2, 2, 1], [2, 2, 1], [2, 2, 2], [2, 2, 2], [2, 2, 2]	[2, 2, 1], [2, 2, 1]

training epochs (i.e., max\_epoch in Equation 5) is 1000 and one epoch contains 250 iterations. The numbers of heads of multi-head self-attention used in different encoder stages are [6, 12, 24, 48] and [3, 6, 12, 24] on Synapse and ACDC, respectively.

$$\text{lr} = \text{initial\_lr} \times \left(1 - \frac{\text{epoch\_id}}{\text{max\_epoch}}\right)^{0.9}. \quad (5)$$

**Pre-processing and augmentation strategies.** All images will be first re-sampled to the same target spacing. Augmentations such as rotation, scaling, gaussian noise, gaussian blur, brightness and contrast adjust, simulation of low resolution, gamma augmentation and mirroring are applied in the given order during the training process.

**Deep supervision.** We also add deep supervision during the training stage. Specifically, the output of each stage in the decoder is passed to the final expanding block, where cross entropy loss and dice loss would be applied. In practice, given the prediction of one typical stage, we down-sample the ground truth segmentation mask to match the prediction’s resolution. Thus, the final training objective

function is the sum of all losses at three resolutions:

$$\mathcal{L}_{all} = \alpha_1 \mathcal{L}_{\{H, W, D\}} + \alpha_2 \mathcal{L}_{\{\frac{H}{4}, \frac{W}{4}, \frac{D}{2}\}} + \alpha_3 \mathcal{L}_{\{\frac{H}{8}, \frac{W}{8}, \frac{D}{4}\}}. \quad (6)$$

Here,  $\alpha_{\{1, 2, 3\}}$  denote the magnitude factors for losses in different resolutions. In practice,  $\alpha_{\{1, 2, 3\}}$  halve with each decrease in resolution, leading to  $\alpha_2 = \frac{\alpha_1}{2}$  and  $\alpha_3 = \frac{\alpha_1}{4}$ . Finally, all weight factors are normalized to 1.

**Pre-trained model weights.** Pre-training can be vastly important to provide generalized and transferable representations for downstream tasks. Given the fact that most operations in nnFormer operate on 1D sequences, we explore the possibility of transferring pre-trained weights on natural images to the medical imaging field. More concretely, we aim to reap the benefit of pre-trained weights of MLP layers and QKV attention on ImageNet pre-training. To this goal, we align channel numbers of transformer blocks to those of pre-trained models so that we load the weights of MLP layers and QKV attention. Besides, considering architectures of the encoder and decoder are highly symmetrical, we propose *symmetrical initialization* to reuse the pre-trained weights of the encoder in the decoder. Specifically, transformer blocks with the same input and output resolution are initialized using the same set of model weights (i.e., symmetrical transformer blocks of the encoder and decoder in Figure 2).

**Network configurations.** In Table 1, we display network configurations of experiments on Synapse and ACDC. Compared to nnUNet, in nnFormer, better segmentation results can be achieved with smaller-sized input patches.

## 4.2 Experiments on Synapse

As shown in Table 2, we make experiments on Synapse and to compare our nnFormer against a variety of both transformer- and convnet-based baselines. The major evaluation metric is dice score.

Apart from nnUNet, the best performing convnet-based method is DualNorm-UNet [36] that achieves an average dice score of 80.37. In comparison, WAD reports the best transformer-based results whose average is 80.30, slightly lower than DualNorm-UNet. Our nnFormer is able to outperform WAD and DualNorm-UNet by over 8 percents and 7 percents in average, respectively, which are quite impressive improvements on Synapse. Besides, we found that the performance of nnUNet are severely underestimated. When being carefully tuned, nnUNet reaches an average dice score of 86.99, which is much better than DualNorm-UNet and WAD but still worse than proposed nnFormer.

## 4.3 Experiments on ACDC

Table 3 presents the experimental results on ACDC, where the overall performance of transformer-based baselines are better than those of convnet-based ones. The underlying reason is that images from ACDC have many fewer slices on  $z$ -axis

Table 2: Experiments on Synapse (dice score in %). Best results are bolded.

Methods	Average	Aotra	Gallbladder	Kidnery(L)	Kidnery(R)	Liver	Pancreas	Spleen	Stomach
VNet [22]	68.81	75.34	51.87	77.10	80.75	87.84	40.04	80.56	56.98
DARR [9]	69.77	74.74	53.77	72.31	73.24	94.08	54.18	89.90	45.96
R50 U-Net [27]	74.68	87.74	63.66	80.60	78.19	93.74	56.90	85.87	74.16
U-Net [27]	76.85	89.07	69.72	77.77	68.60	93.43	53.98	86.67	75.58
R50 Att-UNet [28]	75.57	55.92	63.91	79.20	72.71	93.56	49.37	87.19	74.95
Att-UNet [23]	77.77	89.55	68.88	77.98	71.11	93.57	58.04	87.30	75.75
VIT None [8]	61.50	44.38	39.59	67.46	62.94	89.21	43.14	75.45	68.78
VIT CUP [8]	67.86	70.19	45.10	74.70	67.40	91.32	42.00	81.75	70.44
R50 VIT CUP [8]	71.29	73.73	55.13	75.80	72.20	91.51	45.99	81.99	73.95
R50-Deeplabv3+ [7]	75.73	86.18	60.42	81.18	75.27	92.86	51.06	88.69	70.19
DualNorm-UNet [36]	80.37	86.52	55.51	88.64	86.29	95.64	55.91	94.62	79.8
CGNET [35]	75.08	83.48	65.32	77.91	72.04	91.92	57.37	85.47	67.15
ContextNet [25]	71.17	79.92	51.17	77.58	72.04	91.74	43.78	86.65	66.51
DABNet [15]	74.91	85.01	56.89	77.84	72.45	93.05	54.39	88.23	71.45
EDANet [21]	75.43	84.35	62.31	76.16	71.65	93.20	53.19	85.47	77.12
ENet [24]	77.63	85.13	64.91	81.10	77.26	93.37	57.83	87.03	74.41
FPENet [19]	68.67	78.98	56.35	74.54	64.36	90.86	40.60	78.30	65.35
FSSNet [40]	74.59	82.87	64.06	78.03	69.63	92.52	53.10	85.65	70.86
SQNet [30]	73.76	83.55	61.17	76.87	69.40	91.53	56.55	85.82	65.24
FastSCNN [26]	70.53	77.79	55.96	73.61	67.38	91.68	44.54	84.51	68.76
TransUNet [6]	77.48	87.23	63.16	81.87	77.02	94.08	55.86	85.08	75.62
SwinUNet [3]	79.13	85.47	66.53	83.28	79.61	94.29	56.58	90.66	76.6
TransClaw U-Net [4]	78.09	85.87	61.38	84.83	79.36	94.28	57.65	87.74	73.55
LeViT-UNet-384s [38]	78.53	87.33	62.23	84.61	80.25	93.11	59.07	88.86	72.76
WAD [17]	80.30	87.73	69.93	83.95	79.78	93.95	61.02	88.86	77.16
nnUNet (3D) [11]	86.99	<b>93.01</b>	<b>71.77</b>	85.57	<b>88.18</b>	<b>97.23</b>	<b>83.01</b>	91.86	85.25
nnFormer	<b>87.40</b>	92.04	71.09	<b>87.64</b>	87.34	96.53	82.49	<b>92.91</b>	<b>89.17</b>

(i.e., the spacing on  $z$ -axis is quite large in Figure 1b), which is the exactly the case where transformer has more advantages as they are designed to deal with 2D inputs with less interaction on  $z$ -axis. From Table 3, we can see that the best transformer model is LeViT-UNet-384s, which average dice is slightly higher than SwinUNet but much higher than convnet-based Dual-Attn. In contrast, nnFormer surpasses LeViT-UNet-384s by nearly 1.5 percents on average, again displaying its advantages over transformer-based baselines.

#### 4.4 Ablation study

In this section, we present the significance of the embedding block and convolutional down-sampling blocks. Besides, we also investigate the influences of adding more transformer blocks to the encoder and employing natural images based pre-training.

**Significance of the embedding block.** To investigate the influence of our embedding block, we replace it with patch-wise convolution, as shown in Table 4. Patch-wise convolution [8, 20] only contains one convolutional layer that has large kernel size and stride. For instance, on Synapse, both the kernel size and convolutional stride are set to [4, 4, 2]. We can see that the successive convolutional

Table 3: Experiments on ACDC (dice score in %). Best results are bolded.

Methods	Average	RV	Myo	LV
R50-U-Net [27]	87.55	87.10	80.63	94.92
R50-Attn UNet [28]	86.75	87.58	79.20	93.47
VIT-CUP [8]	81.45	81.46	70.71	92.18
R50-VIT-CUP [8]	87.57	86.07	81.88	94.75
TransUNet [6]	89.71	88.86	84.54	95.73
SwinUNet [3]	90.00	88.55	85.62	<b>95.83</b>
LeViT-UNet-384s [38]	90.32	89.55	87.64	93.76
nnUNet (3D) [11]	91.59	<b>90.25</b>	89.10	95.41
nnFormer	<b>91.78</b>	90.22	<b>89.53</b>	95.59

Table 4: Investigation of the embedding block on Synapse. **Patch-wise convolution** consists of only one convolutional layer with large kernel size and stride.

	Average	Aotra	Gallbladder	Kidnery(L)	Kidnery(R)	Liver	Pancreas	Spleen	Stomach
Patch-wise convolution	84.63	88.84	65.33	86.97	85.98	95.58	77.30	91.83	85.15
Ours	<b>87.40</b>	<b>92.04</b>	<b>71.09</b>	<b>87.64</b>	<b>87.34</b>	<b>96.53</b>	<b>82.49</b>	<b>92.91</b>	<b>89.17</b>

layers of the embedding block surpasses the atypical patch-wise convolution by nearly 3 percents in average. Actually, similar phenomena were first time observed in [29], where small-sized kernels are found to be more effective than large ones.

**Influences of convolutional down-sampling blocks.** In Table 5, we report the results of using neighboring concatenation [20] in nnFormer to replace convolutional down-sampling blocks. The main operation of neighboring concatenation is to concatenate neighboring patches of features maps in the channel dimension, after which a fully-connected layer is added to reduce the number of channels. From Table 5, we can see that convolutional down-sampling blocks provide over 3 percents improvements over neighboring concatenation in average, demonstrating that applying interleaved convolutional down-sampling blocks are more helpful to build hierarchical object concepts at various scales.

**Investigation of adding more transformer blocks.** We add one more SV-MSA block to the last encoder stage and report its results in Table 6. Generally speaking, more transformer blocks help nnFormer to encode more long-term dependencies into representations. Somewhat interestingly, from Table 6, we can find that capturing more long-term dependencies may not be an optimal choice for nnFormer. For instance, although introducing more transformer blocks achieve a higher segmentation dice score on Gallbladder (the most difficult organ on Synapse), it deteriorates the segmentation performance on other organs. We will explore the reason behind in the future.

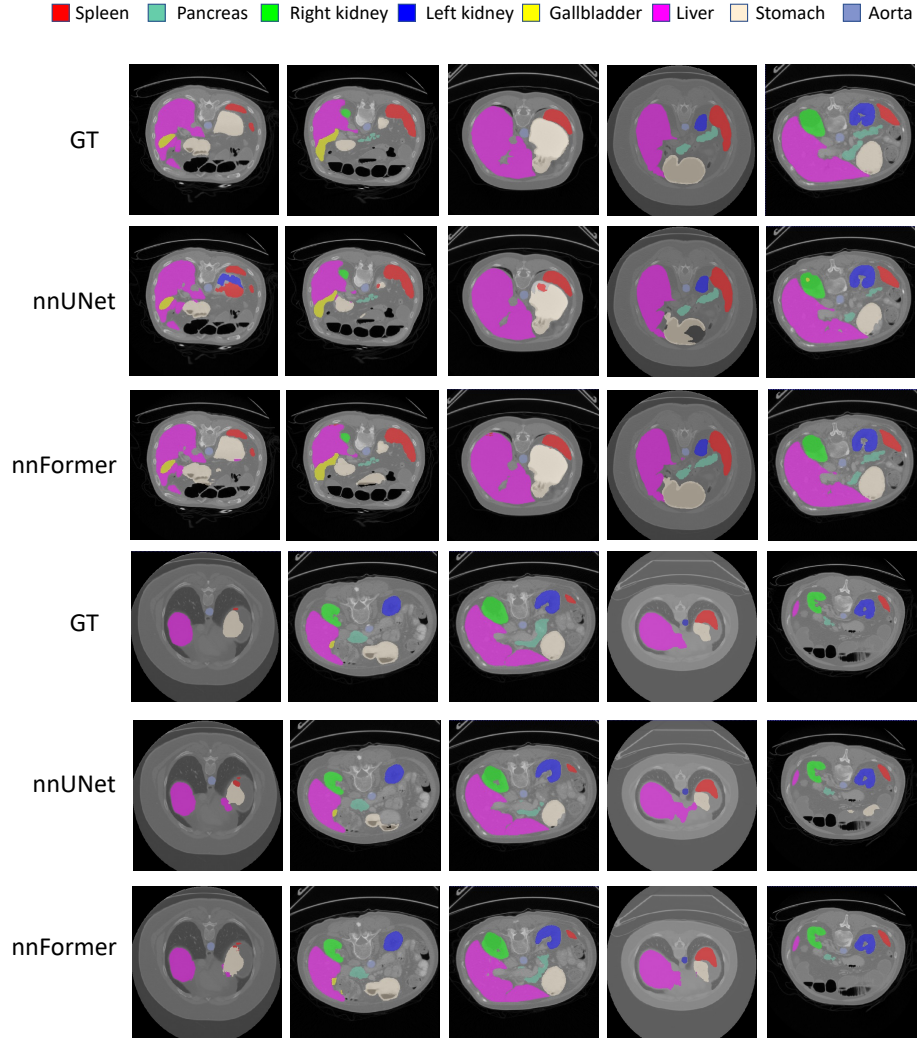


Fig. 3: Segmentation results of some hard samples on Synapse.

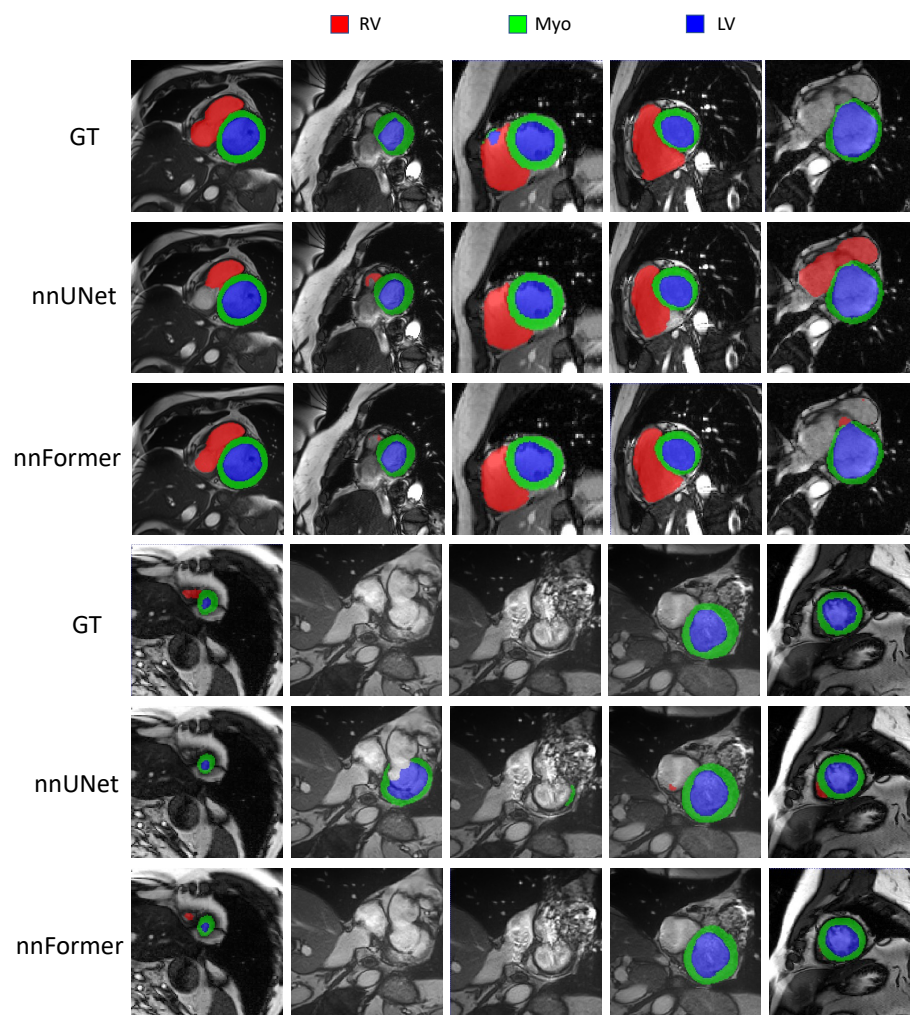


Fig. 4: Segmentation results of some hard samples on ACDC.

Table 5: Investigation of the convolutional down-sampling blocks.

	Average	Aotra	Gallbladder	Kidnery(L)	Kidnery(R)	Liver	Pancreas	Spleen	Stomach
Neighboring concatenation	84.30	88.00	67.60	87.52	87.38	95.31	80.63	85.29	82.69
Ours	<b>87.40</b>	<b>92.04</b>	<b>71.09</b>	<b>87.64</b>	<b>87.34</b>	<b>96.53</b>	<b>82.49</b>	<b>92.91</b>	<b>89.17</b>

Table 6: Investigation of adding more transformer blocks.

	Average	Aotra	Gallbladder	Kidnery(L)	Kidnery(R)	Liver	Pancreas	Spleen	Stomach
More transformer blocks	85.98	89.02	<b>71.74</b>	86.76	87.06	96.37	82.30	89.04	85.51
Ours	<b>87.40</b>	<b>92.04</b>	71.09	<b>87.64</b>	<b>87.34</b>	<b>96.53</b>	<b>82.49</b>	<b>92.91</b>	<b>89.17</b>

Table 7: Benefits of using pre-trained weights on natural images.

	Average	Aotra	Gallbladder	Kidnery(L)	Kidnery(R)	Liver	Pancreas	Spleen	Stomach
No pre-training	84.34	90.15	69.00	86.34	87.48	95.93	80.97	85.23	79.67
Ours	<b>87.40</b>	<b>92.04</b>	71.09	<b>87.64</b>	<b>87.34</b>	<b>96.53</b>	<b>82.49</b>	<b>92.91</b>	<b>89.17</b>

**Influences of using pre-trained models on natural images.** In Table 7, we show that it is crucial to make use of pre-trained weights on natural images, where removing pre-training deteriorates the overall segmentation performance by over 3 percents. The underlying reason is that Synapse does not have enough labeled 3D scans to fully realize the potential of nnFormer.

#### 4.5 Visualization

In Figures 3 and 4, we compare the segmentation results of nnUNet and nnFormer in some hard samples. On Synapse, it seems that nnFormer has quite apparent advantages on stomach where nnUNet often fails to generate an integrated delineation mask. Meanwhile, compared nnUNet, nnFormer has the ability to reduce false positive predictions of spleen, which are also consistent with the performance reported in Table 2.

Similar phenomena can also be observed in Figure 4, where nnFormer can greatly reduce false positive predictions of right ventricle (RV) and myocardium (MYO), especially myocardium. These segmentation results help to verify the fact that nnFormer can produce more robust and discriminative representations than nnUNet.

## 5 Conclusion

In this paper, we present a new medical image segmentation network named nnFormer. nnFormer is constructed on top of an interleaved stem of convolution and self-attention, where convolution helps encode precise spatial information into high-resolution low-level features and build hierarchical object concepts at multiple scales. On the other hand, self-attention in transformer blocks entangles long-term dependencies with convolutional representations to capture global

context. Based on such hybrid architecture, nnFormer achieves tremendous progress over previous transformer-based segmentation methodologies. Even when compared to nnUNet, currently the best performing segmentation network, nnFormer still provides consistent yet observable improvements. In the future, we hope nnFormer could draw more attention from the medical imaging community to make efforts on developing more efficient segmentation models.

## References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G., et al.: Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE transactions on medical imaging* **37**(11), 2514–2525 (2018)
3. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-Unet: Unet-like pure transformer for medical image segmentation. arXiv preprint arXiv:2105.05537 (2021)
4. Chang, Y., Menghan, H., Guangtao, Z., Xiao-Ping, Z.: TransClaw U-Net: Claw u-net with transformers for medical image segmentation. arXiv preprint arXiv:2107.05188 (2021)
5. Chen, B., Liu, Y., Zhang, Z., Lu, G., Zhang, D.: TransAttUnet: Multi-level attention-guided u-net with transformer for medical image segmentation. arXiv preprint arXiv:2107.05274 (2021)
6. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., et al.: TransUNet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(4), 834–848 (2018)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Fu, S., Lu, Y., Wang, Y., Zhou, Y., Shen, W., Fishman, E., Yuille, A.: Domain adaptive relational reasoning for 3d multi-organ segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 656–666. Springer (2020)
10. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
11. Isensee, F., Jäger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: Automated design of deep learning methods for biomedical image segmentation. arXiv preprint arXiv:1904.08128 (2019)
12. Karimi, D., Vasylechko, S., Gholipour, A.: Convolution-free medical image segmentation using transformers. arXiv preprint arXiv:2102.13645 (2021)
13. Landman, B., Xu, Z., Igelsias, J.E., Styner, M., Langerak, T., Klein, A.: Miccai multi-atlas labeling beyond the cranial vault—workshop and challenge. In: *Proc. MICCAI: Multi-Atlas Labeling Beyond Cranial Vault-Workshop Challenge* (2015)



14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
15. Li, G., Yun, I., Kim, J., Kim, J.: Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv preprint arXiv:1907.11357* (2019)
16. Li, S., Sui, X., Luo, X., Xu, X., Liu, Y., Goh, R.S.M.: Medical image segmentation using squeeze-and-expansion transformers. *arXiv preprint arXiv:2105.09511* (2021)
17. Li, Y., Cai, W., Gao, Y., Hu, X.: More than encoder: Introducing transformer decoder to upsample. *arXiv preprint arXiv:2106.10637* (2021)
18. Lin, A., Chen, B., Xu, J., Zhang, Z., Lu, G.: DS-TransUNet: Dual swin transformer u-net for medical image segmentation. *arXiv preprint arXiv:2106.06716* (2021)
19. Liu, M., Yin, H.: Feature pyramid encoding network for real-time semantic segmentation. *arXiv preprint arXiv:1909.08599* (2019)
20. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* (2021)
21. Lo, S.Y., Hang, H.M., Chan, S.W., Lin, J.J.: Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In: *MMAAsia '19: ACM Multimedia Asia* (2019)
22. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: *2016 fourth international conference on 3D vision (3DV)*. pp. 565–571. *IEEE* (2016)
23. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999* (2018)
24. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147* (2016)
25. Poudel, R.P., Bonde, U., Liwicki, S., Zach, C.: Contextnet: Exploring context and detail for semantic segmentation in real-time. *arXiv preprint arXiv:1805.04554* (2018)
26. Poudel, R.P., Liwicki, S., Cipolla, R.: Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502* (2019)
27. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. pp. 234–241. *Springer* (2015)
28. Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., Rueckert, D.: Attention gated networks: Learning to leverage salient regions in medical images. *Medical Image Analysis* **53**, 197–207 (2019). <https://doi.org/https://doi.org/10.1016/j.media.2019.01.012>, <https://www.sciencedirect.com/science/article/pii/S1361841518306133>
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
30. Trembl, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Hochreiter, S.: Speeding up semantic segmentation for autonomous driving. In: *NIPS 2016 Workshop - MLITS* (2016)
31. Tuli, S., Dasgupta, I., Grant, E., Griffiths, T.L.: Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197* (2021)
32. Valanarasu, J.M.J., Oza, P., Hacihaliloglu, I., Patel, V.M.: Medical Transformer: Gated axial-attention for medical image segmentation. *arXiv preprint arXiv:2102.10662* (2021)

33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is all you need. In: *Adv. in Neur. Info. Process. Sys.* pp. 5998–6008 (2017)
34. Wang, W., Chen, C., Ding, M., Li, J., Yu, H., Zha, S.: TransBTS: Multimodal brain tumor segmentation using transformer. *arXiv preprint arXiv:2103.04430* (2021)
35. Wu, T., Tang, S., Zhang, R., Cao, J., Zhang, Y.: Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Transactions on Image Processing* **30**, 1169–1179 (2021)
36. Xiao, J., Yu, L., Xing, L., Yuille, A., Zhou, Y.: Dualnorm-unet: Incorporating global and local statistics for robust medical image segmentation. *arXiv preprint arXiv:2103.15858* (2021)
37. Xie, Y., Zhang, J., Shen, C., Xia, Y.: CoTr: Efficiently bridging cnn and transformer for 3d medical image segmentation. *arXiv preprint arXiv:2103.03024* (2021)
38. Xu, G., Wu, X., Zhang, X., He, X.: LeViT-UNet: Make faster encoders with transformer for medical image segmentation. *arXiv preprint arXiv:2107.08623* (2021)
39. Yun, B., Wang, Y., Chen, J., Wang, H., Shen, W., Li, Q.: SpecTr: Spectral transformer for hyperspectral pathology image segmentation. *arXiv preprint arXiv:2103.03604* (2021)
40. Zhang, X., Chen, Z., Wu, Q., Cai, L., Lu, D., Li, X.: Fast semantic segmentation for scene perception. *IEEE Transactions on Industrial Informatics* (2018)
41. Zhang, Y., Liu, H., Hu, Q.: TransFuse: Fusing transformers and cnns for medical image segmentation. *arXiv preprint arXiv:2102.08005* (2021)
42. Zhou, H.Y., Lu, C., Yang, S., Yu, Y.: ConvNets vs. Transformers: Whose visual representations are more transferable? *arXiv preprint arXiv:2108.05305* (2021)