**CONVTRANSSEG: A MULTI-RESOLUTION CONVOLUTION-TRANSFORMER NETWORK FOR MEDICAL IMAGE SEGMENTATION**

Zhendi Gong  
School of Computer Science  
University of Nottingham  
Nottingham, NG8 1BB  
psxzg6@nottingham.ac.uk

Andrew P. French  
School of Computer Science  
University of Nottingham  
Nottingham, NG8 1BB

Guoping Qiu  
School of Computer Science  
University of Nottingham  
Nottingham, NG8 1BB

Xin Chen  
School of Computer Science  
University of Nottingham  
Nottingham, NG8 1BB  
xin.chen@nottingham.ac.uk

October 14, 2022

**ABSTRACT**

Convolutional neural networks (CNNs) achieved the state-of-the-art performance in medical image segmentation due to their ability to extract highly complex feature representations. However, it is argued in recent studies that traditional CNNs lack the intelligence to capture long-term dependencies of different image regions. Following the success of applying Transformer models on natural language processing tasks, the medical image segmentation field has also witnessed growing interest in utilizing Transformers, due to their ability to capture long-range contextual information. However, unlike CNNs, Transformers lack the ability to learn local feature representations. Thus, to fully utilize the advantages of both CNNs and Transformers, we propose a hybrid encoder-decoder segmentation model (ConvTransSeg). It consists of a multi-layer CNN as the encoder for feature learning and the corresponding multi-level Transformer as the decoder for segmentation prediction. The encoder and decoder are interconnected in a multi-resolution manner. We compared our method with many other state-of-the-art hybrid CNN and Transformer segmentation models on binary and multiple class image segmentation tasks using several public medical image datasets, including skin lesion, polyp, cell and brain tissue. The experimental results show that our method achieves overall the best performance in terms of Dice coefficient and average symmetric surface distance measures with low model complexity and memory consumption. In contrast to most Transformer-based methods that we compared, our method does not require the use of pre-trained models to achieve similar or better performance. The code is freely available for research purposes on Github: (the link will be added upon acceptance).

**Keywords** Medical Image Segmentation · Convolutional Neural Network · Transformer
1 Introduction

Accurate image segmentation is essential for many clinical applications, such as computer-aided diagnosis, treatment planning and image-guided surgery [1]. Conventional methods require detailed domain knowledge from human and task specific designs, such as active contours [2], atlas-based methods [3] and graph cuts [4]. These methods do not require a large number of training examples. However, due to complex parameter settings and poor robustness on different imaging conditions (e.g. intensity variations, presence of noise, etc.), these methods were outperformed by deep-learning-based methods, particularly convolutional neural networks (CNNs) [5].

In CNNs, convolutional filters are applied to local image regions in a hierarchical manner, which makes them effective in learning multi-resolution image features and leads to their success in many computer vision tasks (e.g. object recognition and segmentation). With the growth of computing ability, CNNs have become the backbone for most medical image analysis tasks, including image segmentation [6]. One of the most widely used CNNs in medical image segmentation is U-Net [7], due to its effective learning capability from a smaller number of training examples compared to other CNN models (e.g., Fully Convolutional Network [8]). U-Net consists of an encoder to extract multi-resolution image features, a decoder to upsample the feature maps for segmentation prediction and skip connections between the encoder and decoder for a more efficient model learning.

Many efforts have been made to improve the vanilla U-Net. The U-Net variants include: Residual U-Net (ResUNet) [9], which applied the residual network mechanism [10] to the U-net for handling vanishing gradients problem; U-Net++ [11], which applied more complex skip connections to utilize multi-scale information better; 3D U-Net [12] and V-Net [13], which extended the U-Net to 3D segmentation tasks, such as MRI volumes segmentation. Although the image segmentation field witnessed the success of CNNs, such methods still suffer from the weak ability of capturing long-range regional interactions, which is important in certain applications. With the lack of global contextual information of the whole image, CNNs may produce suboptimal results (shown in section 1).

More recently, inspired by the work of Vaswani et al. [14], to perform the machine translation task in natural language processing. The basic building block in a Transformer is self-attention, which is used to capture long-term dependencies between the input tokens (e.g. words). Specifically, self-attention aims to calculate the weighted sum of the values, in which the weight for each value is computed by a function of a query-key pair, where values (V), queries (Q) and keys (K) are all input vectors. For instance, when the self-attention is applied in machine translation, V, Q and K are the embedding vectors for the input words. Within the self-attention layer, all positions are connected with a number of operations [14], whereas a convolutional layer in CNN does not connect all paired positions between the input and output. Additionally, they used a multi-head attention (MHA) block instead of a single attention function based on the finding that it is beneficial to firstly project V, Q and K by different learnable linear layers several times in parallel attention functions. The model is able to manage information from various representation subspaces concurrently with multi-head attention. As the attention block can be taught to focus on specific regions within a context, it becomes a crucial part of neural transduction models for many techniques [23-25].

Based on the architecture of encoders in Transformer, Dosovitskiy et al. [15] proposed a Vision Transformer (ViT) for image classification tasks. Different from other...
works applying attention mechanism in vision tasks, the image-specific inductive bias was not introduced in their work, they directly used a standard Transformer encoder from the work of Vaswani et al. [14]. They firstly divided the input image into several contiguous patches with the same size. Additionally, to apply the same attention layers in Transformer, they flattened the 2D image patches into 1D vectors as the input tokens. Meanwhile, they projected the vectors to a subspace with learnable linear layers. In other words, V, Q and K are represented by the embedding vectors of image patches in their method. However, the success of ViT is highly dependent on pre-trained models and large training datasets. The limited availability of medical data restricts the application of ViT in the field of medical image processing. After Transformer and ViT were proposed, the combination of CNN and attention mechanisms to achieve medical image segmentation has been an active research area. We divide the recently proposed methods into four categories, dependent on the relative locations of the Transformer and CNN: Transformer in encoder, Transformer in decoder, Fused CNN and Transformer. Besides, some methods also applied pure Transformer models to perform medical image segmentation; we also presented some of these methods.

2.1 Transformer in encoder

Hatamizadeh et al. [26] proposed UNETR for 3D medical image segmentation. They firstly divided the 3D volume into 3D image patches and input them into the Transformer encoder. The multi-scale outputs were further processed by CNN encoder-decoder architecture, where skip connections were applied at different feature scales. Another work is from Wu et al. [18], who proposed a feature adaptive transformer, named FAT-net, which applied a dual encoder by using both CNN and Transformer to perform feature learning. Additionally, they applied a feature adaptive module in the skip connection to perform feature fusion and a memory-efficient decoder to further process the output feature maps for segmentation prediction. Additionally, Li et al. [33] proposed X-Net, which contains a CNN branch and a Transformer branch. The CNN branch takes the original image as input and performs reconstruction, and the Transformer branch firstly encodes the divided image patches and then applies convolutional layers from the CNN branch to decode the output. Shen et al. [28] proposed a model called COTR-Net to perform automated kidney tumor segmentation, where Transformer layers are inserted after each CNN block in the encoder of a U-Net-like model.

2.2 Transformer as a bridge

TransUNet [16] used an attention module consisted of 12 Transformer layers to capture the long-term relationship of the hidden features extracted by a CNN. The decoder applied CNN-based up-sampling layers to process the output from the attention module with skip connections applied between the encoder and decoder. TransAttUNet [29] applied both Transformer Self Attention and Global Spatial Attention between the encoder and decoder. Additionally, they applied multi-scale skip connections to enhance the performance of the original U-shaped architecture. Similarly, Ji et al. [30] proposed a Multi-Compound Transformer which applied two separate Transformer modules between the encoder and decoder. Specifically, they used a Transformer self-attention module to construct the multi-scale context and applied a proxy-embedding in the Transformer cross-attention module to handle feature representations and category dependencies. Additionally, Zhang et al. [51] applied Transformer blocks as a bridge between the CNN-based encoder and decoder to perform cell segmentation. After the decoder, they combined different losses on the cell body and edge to achieve better performance on learning separate information, which can provide accurate information of the edges and support local consistency. Different from the above methods that apply Transformer blocks as a bridge to connect the encoder and decoder, AttUnet [22] applied attention gates (AG) to filter the encoded features through the skip connections. They aimed to reduce the false-positive predictions for small objects using the feature selectivity offered by AG.

2.3 Transformer in decoder

Different from the aforementioned methods, Transformers can also be applied after the CNN encoder as a decoder to capture the global context information of the CNN encoded features. However, there are very limited works in this category. Li et al. [33] proposed SegTran for medical image segmentation. They applied a pre-trained ResNet in the encoder to extract the feature maps and used the squeeze-and-expansion Transformer in the decoder, with a squeeze block to learn attention matrix and an expansion block to learn varied representations. Another work is from Li et al. [34], who proposed a window attention up-sampling module in the classic U-Net decoder to replace the original convolutional up-sampling method. Specifically, the different resolution levels in the decoder are connected by the window attention decoder and bilinear up-sampling, which is used as a residual connection.

2.4 Fused CNN and Transformer

Li et al. [35] proposed a U-Net like group Transformer network GT U-Net. It inherits the architecture of the traditional U-Net [7], but grouped convolutions were applied instead of the original convolution layers. Additionally, they applied self-attention based Transformer blocks between each pair of group convolutions to learn the global dependencies of different feature representations. In another work, named as TransFuse [36], a convolutional branch and a Transformer branch are fused parallelly by their proposed BiFusion module. Specifically, the Transformer branch takes image patches as input then reshapes and up-samples the output to different scales. Meanwhile, the output features with different resolution from a CNN branch, which takes the whole image as input, are fused...
with the outputs from a Transformer branch. Furthermore, Gao, et al. proposed UTNet [37], which applies Transformer blocks after the convolutional blocks both in CNN encoder and decoder. Additionally, skip connections are used between encoder and decoder at each resolution level.

2.5 Pure Transformer

Karimi et al. [38] proposed a convolution-free method for 3D medical image segmentation. They flattened the divided 3D image patches from the original volume to 1D vectors and input them to the Transformer mechanism. After a set of attention functions, they pick the output of the center patch and reshape it as the segmentation prediction on that patch. Furthermore, inspired by the U-Net architecture, Cao et al. [39] proposed Swin-Unet to perform medical image segmentation using the Swin Transformer [40] as the basic block. Different from the regular Transformer block used in ViT, the Swin Transformer uses shifted windows to perform self-attention computation, which has a higher computational efficiency. Swin-Unet applied these Swin Transformer in a U-shape architecture with skip connections between the encoder and decoder. MISSFormer [41] has a similar architecture with Swin-Unet with Swin Transformer and regular skip connections replaced by an enhanced Transformer block and an enhanced Transformer context bridge.

2.6 Contributions

It can be concluded from the above related works that the mainstream idea is to use a Transformer’s attention mechanism to capture the long-range relationships of image patches either using their raw values or the feature representations derived from CNNs. The main novelties and contributions of our work are summarized as below:

1. Our proposed image segmentation method is a hybrid model which applies convolutions as local image feature extractors in an encoder and attention layers in a decoder for segmentation prediction. Compared to the works [33][34] that also apply the Transformer as decoder, our method has certain advantages. Specifically, Li, et al. [33] only used attention layers to replace the original up-sampling layers in the decoder of U-Net. Our method explores the feasibility of utilizing a pure Transformer in the entire decoder, since we aim to fully use the features of Transformer to capture the long-term dependencies between the local features of different image patches. Additionally, although SegTran [33] used a pure Transformer in the decoder, their Transformer layers only used the final output of a CNN encoder without multi-resolution information. We propose a new approach to connect a CNN-based encoder and a pure Transformer-based decoder in multi-resolution that are seamlessly connected in their corresponding levels, including dimensionality reduction and reshaping operations. No such efficient connection between CNN encoder and Transformer decoder has been proposed in the literature.

2. Many Transformer-based methods require pre-training, since Transformers lack the ability to work well on limited datasets. However, our method does not require any pretraining, even when training on a small dataset, and can still achieve similar or better performance compared to the models with pretraining. Without the need for pretraining, the architecture of our model allows us to use different image sizes as the input as an additional bonus.

3 Method

3.1 Overall model structure

The overall model structure of our method is depicted in Fig 1. Our method contains four main components, including a CNN-based encoder (CE), a Transformer-based decoder (TD), and skip connections and a bridge between CE and TD. The CE follows a similar design of the encoder in ResUNet [9], where each convolutional block is a residual convolutional block (ResConv). A max pooling layer is applied after each ResConv block to down-sample the learned feature maps into lower image resolutions. The TD is also multi-resolution. At each level, n cascaded self-attention blocks (Trans blocks) are designed to take the transformed CNN-features from the corresponding level as the input. Additionally, a linear layer (Linear or Up-sample Linear) is applied to project the features of the current level to the same vector dimension of the higher resolution level, which is then added to the higher dimensional feature vector as the input to TD. Moreover, skip connections that contain linear down-sampling (Down-sample Linear) and reshaping (Flatten & Reshape) operators, and a bridge that contains flattening (Flatten) and a positional embedding (POS) are applied to connect the CE and TD in the corresponding levels. Finally, a reshaping (Reshape) and convolutional layer (CONV) in TD are applied to convert the output of the final Transformer blocks to multi-class segmentation predictions.

Although the input image size is flexible in our method, to assist the understanding of the model structure, an input color image (3 RGB channels) with size of 256x256 is used as an example to illustrate the dimensions of data at each of the building blocks in a 4-level architecture as shown in Fig 1. The justifications and working principles of the proposed design are provided in the following subsections.

3.2 CNN encoder

A residual block based CNN [9] is adopted as the encoder for feature extraction in our method (Fig 1, CNN Encoder-CE). As shown in Fig 2(a), each ResConv block consists
of two convolutional layers with the kernel size of 3×3 using zero padding, each followed by a batch normalization layer and a ReLU activation layer. The first convolutional layer in each ResConv block doubles the number of feature channels of the input data, except for the block in the first level, where the first convolutional layer changes the number of feature channels of the input data to C_{base} (e.g. C_{base} = 64 in Level 0 in Fig 1). The second convolutional layer in each ResConv block has the same number of feature channels as the first convolutional layer. Additionally, a single convolutional layer is added as the skip connection between the output and the input of ResConv. Specifically in the CNN Encoder, the input image X_{in} \in \mathbb{R}^{W \times H \times C_{in}} (W, H, and C_{in}) indicate the width, height, and the number of channels respectively of the input image) is processed to generate X_{out}^{CE} \in \mathbb{R}^{(W/2^l) \times (H/2^l) \times (2^l C_{base})} in each level i (i = 0, 1, 2, ..., l) indicates the depth of CE).

### 3.3 Skip connections and bridge

The output of the deepest level in CE X_{out}^{CE} \in \mathbb{R}^{(W/2^l) \times (H/2^l) \times (2^l C_{base})} is flattened spatially to \mathbb{R}^{P_l \times (2^l C_{base})} and added to a learnable positional embedding that has the same dimension, where P_l = (W/2^l) \times (H/2^l). This process aims to learn the positional information of the input feature vectors (tokens), before being input to the TD. In other words, the input to TD in level l can be considered as the feature embeddings of P_l image patches, each with the size of 2^l \times 2^l.

Instead of only using the lowest feature resolution at level l, the skip connections between CE and TD aim to help the Transformer layers learn the CNN features from different resolution levels. However, if the output of CE is flattened and input to TD directly in all levels, the whole model would be extremely large to be computed (e.g. it would require W×H tokens in level 0). To solve such a problem, we fix the number of input tokens to P_l in the TD for all levels. Thus, the TD can be considered as always processing the feature vectors of 2^l \times 2^l patches of the input image. Subsequently, a learnable linear layer is added (the “Down-sample linear” shown in Fig 1) to down-sample the feature channels of the CNN outputs before reshaping them to the desired feature dimensions. Specifically, in level i (i < l), the output from CE X_{out}^{CE} \in \mathbb{R}^{(W/2^l) \times (H/2^l) \times (2^l C_{base})} is first linearly downsampled to \mathbb{R}^{(W/2^l) \times (H/2^l) \times (2^l C_{base}/m)}, where m indicates the down-sampling factor that is consistent for all levels m=4 in the example shown in Fig 1. Subsequently, this three-dimensional tensor at each level is reshaped to P_l (e.g. P_l = 1024 in Fig 1) by d_l (e.g. d_0 = 1024,
As the example shown in Fig 1, the feature map from CE has the dimension of $64 \times 64 \times 256$ in level 2. The down-sampling linear layer in level 2 is then firstly reduce the dimension of the feature channels from 256 to 64 for dimensionality reduction purpose. The first two dimensions $64 \times 64$ can be considered as $32 \times 32 (1024)$ image patches, each with a size of $2 \times 2$. The “Flatten & Reshape” process then converts this $64 \times 64 \times 64$ 3D feature to a 2D $1024 \times 256$ feature representation. 1024 is the number of image patches, and each has the length of $64 \times 64 \times 2 \times 2$ (concatenating the feature vectors in a $2 \times 2$ image region). The resolution increases, with a fixed number of image patches (1024), the sizes of image patches increase hence $d_i$ also increases. The output of the “Flatten & Reshape” process in level $i$ is then added to the output feature vectors from TD in level $i + 1$ to form the input to TD in level $i$. The Transformer block is shown in Fig 2 (b), which was originally introduced by Vaswani et al. [14]. The block contains $n$ Transformer blocks to process the output features from the bridge and skip connections, and is followed by a linear layer to project the feature vector from the current level $i$ to the $i - 1$ level. The Transformer block make no changes to the dimension of the feature vector, while the linear layers change the feature vectors’ dimension to the dimension of the TD input in lower level. The Transformer block is shown in Fig 2(b), which was originally introduced by Vaswani et al. [14]. The block contains a multi-head self-attention mechanism (MHA) and a position-wise fully connected feed-forward network (FFN). Additionally, a normalization layer is added before MHA and FFN, and a dropout layer is added after MHA and FFN. Furthermore, a skip connection is used between the input and the dropout layer. Thus, the calculation within a Transformer block is expressed as below:

$$X_{out1} = X_{TransIn} + \text{Dropout}(\text{MHA}(\text{Norm}(X_{TransIn}), \text{Norm}(X_{TransIn})))$$ (1)

$$X_{out} = X_{out1} + \text{Dropout}(\text{FFN}(\text{Norm}(X_{out1})))$$ (2)

where $X_{TransIn}$ and $X_{out}$ are the input and output of the block. Instead of performing a single attention mechanism, the MHA calculates the scaled dot-product attention (SDPA) several times in parallel, as illustrated in Fig 3. It takes $Q$, $K$, and $V$ as input, and outputs the weighted sum of $V$. Since we use self-attention in the whole method, the matrices $Q$, $K$, and $V$ are the input feature vectors to the TD, which is identical to each other and are consistently updated during model training. As illustrated in Fig 3, they are linearly mapped to $R^{n_i \times d_i}$ from $R^{n_i \times d_i \times d_i}$ and reshaped to $R^{n_i \times d_i \times b}$ before being input to SDPA. $n_i$ indicates the number of tokens ($P_i$ as described

$\text{loss} = \alpha * \text{loss}_{CE} + \beta * \text{loss}_{Dice}$ (5)
4 Experiments and results

4.1 Datasets and evaluation metrics

To evaluate our proposed method and compare it to other state-of-the-art methods, five publicly available medical imaging datasets that cover four medical applications are used:

1. Skin lesion segmentation (ISIC): ISIC2018 [19] (binary-class), which was published by the International Skin Imaging Collaboration challenge, containing 2594 RGB dermoscopy images with different image sizes.

2. Cell segmentation: 2018 Data Science Bowl (Bowl): [https://www.kaggle.com/c/data-science-bowl-2018](https://www.kaggle.com/c/data-science-bowl-2018) (binary-class) and Panuke [21] (5 classes). There are 670 320 × 256 RGB images in 2018 Data Science Bowl and 7901 256 × 256 RGB images in Panuke. The 5 classes for the segmentation mask in Panuke are Neoplastic (Neo), Inflammatory (Inf), Connective (Con), Dead, and Epithelial (Epi) respectively.

3. Polyp segmentation (CVC): CVC-ClinicDB [20] (binary-class) contains 612 384 × 288 RGB colonoscopy images from 31 colonoscopy sequences.

4. Brain tissue segmentation (OASIS): OASIS-1 [22] (4 classes) contains 414 192 × 192 gray scale images, the 4 classes for the segmentation mask are Cortex, Subcortical-Gray-Matter (SGM), White-Matter (WM), and Cerebrospinal fluid (CSF) respectively.

Although our method can accept any size of input image (as long as it can be divided by 2), in our experiments, we reshaped all images to 224 × 224 since most of the methods for comparison used such resolution as the input size (due to they used pre-trained models). Specially, when we compared our model with UTnet [37], we reshaped the images to 256 × 256, since UTNet used the input size of 256 × 256, and the input size of 224 × 224 cannot fit in the model by their default parameter settings. The percentage of training, validation and testing sets are 70%, 10% and 20% respectively for all tasks. All models were trained on a 12GB GeForce GTX 1080 Ti GPU with Adam optimizer. Additionally, the learning rate was set to 0.0001 and the models were all trained for 50 epochs. During each training phase, the model with the least validation loss was saved for testing.

In model evaluation, Dice Coefficient (DC) and Average Symmetric Surface Distance (ASSD) are the main metrics, and we applied Wilcoxon Signed Rank Test (WSRT) to evaluate the statistical significance in difference between two methods. We also compared the number of learnable parameters (NP), the GPU memory usage (GM), and the average training time per step (TS) to evaluate the model complexity and training efficiency.

It is worth noting that, for the Pannuke dataset, some classes have large number of empty ground truth. Our initial results show that the model tend to consistently generate empty predictions on most images in such classes. Thus, we only calculated the loss on the prediction of the class whose ground truth was not empty in the training process for all methods to avoid model over-fitting caused by unbalanced labels. Similarly, by including these empty classes in evaluation, it results in very high DC scores. It makes the measure being less sensitive to the errors in images with non-empty foreground. Hence, we also only calculated DC and ASSD on non-empty classes (in ground truth mask) in the evaluation.

4.2 Parameter tuning

Before comparing our method with other state-of-the-art methods, we tested the sensitivities on the choice of hyper-parameters in our method by performing parameter tuning. Specifically, we set \( l = 3 \) for the depth of CE and TD, so that each feature vector processed in TD is a representation of a 8 × 8 image patch. Additionally, we tested and compared the method performance of using 1, 2, 3, 4, 5 for \( n \) (number of Transformer blocks in TD), 32, 64 for \( C_{base} \) (number of output feature channels of the first convolutional block) and 2, 4, 8 for \( m \) (down-sampling factor). We selected ISIC2018 as a binary-class dataset and OASIS-1 as a multi-class dataset to evaluate the above parameter settings. Considering the trade-off between model performance and model complexity in both datasets, the optimal settings were \( n = 3, C_{base} = 64, \) and \( m = 8 \). These settings were then consistently used for evaluating all other datasets and compared to other methods.

4.3 Ablation study

The two key novelties of our model design are the integrated multi-level feature fusion and the down-sampling linear layers in skip connections to reduce the model’s complexity. Firstly, to illustrate the benefit of utilizing the output feature maps from CE in multiresolution rather than only using the output of level \( l \), we evaluated our method with and without the skip connections (SC) between CE and TD. Secondly, we evaluated our method with and without the down-sampling linear layers (DSL). We choose to use the OASIS-1 dataset in these experiments, since it is a more generic multiclass dataset which includes both large and small objects of interest. We used smaller \( C_{base} \) and \( m \) values instead of the parameter settings that are concluded in section 4.2. Because the model (with SC without DSL) requires significantly larger memory that can not be accommodated by a 12 GB GPU. Thus, we changed \( C_{base} \) to 32 instead of 64, which decreases the number of parameters for all models so that the models can be trained on the same GPU. By setting \( m \) to 4 instead of 8, it ensures that the model with DSL will not lose much information by the down-sampling process.

The mean and standard deviation of DC and ASSD are reported in Table 1, and the evaluations of model complexity are reported in Table 2. Firstly, it is seen from Table 1 that compared to the model without SC, SC (with and without DSL) can increase the performance measured by both DC
Table 1: Evaluation of model performance on OASIS-1 dataset ($C_{base} = 32$, $m = 4$). Mean and Standard deviation of Dice coefficient (DC) and Average Symmetric Surface Distance (ASSD) are reported for different classes and the Average of all classes.

| SC | DSL | Cortex | SGM | WM | CSF | Mean |
|----|-----|--------|-----|----|-----|------|
|    |     | DC     |     |    |     |      |
| Yes | Yes | 0.904  | 0.941| 0.957| 0.929| 0.933|
|     |     | ±0.015 | ±0.014| ±0.006| ±0.032| ±0.009|
| Yes | No  | 0.906  | 0.939| 0.957| 0.919| 0.930|
|     |     | ±0.015 | ±0.016| ±0.006| ±0.036| ±0.010|
| No  |    | 0.878  | 0.924| 0.947| 0.901| 0.912|
|     |     | ±0.016 | ±0.016| ±0.006| ±0.035| ±0.010|

Table 2: Evaluation of model complexity ($C_{base} = 32$, $m = 4$) on OASIS-1 dataset. The mean value of number of parameters (NP), memory consumption (GM) and training time per step are reported.

| SC | DSL | NP (m) | GM (GB) | TS (s) |
|----|-----|--------|---------|--------|
| Yes | Yes | 11.84  | 2.41    | 0.126  |
| Yes | No  | 138.34 | 9.35    | 0.423  |
| No  |    | 11.83  | 2.39    | 0.129  |

and ASSD significantly in all classes ($p < 0.01$ measured by WSRT). Additionally, the use of DSL further increased the performance on the mean class results measured by DC significantly ($p < 0.01$ measured by WSRT) and by ASSD non-significantly ($p = 0.38$ measured by WSRT), compared to the model without DSL.

By achieving the best segmentation accuracy, the model with SC and DSL has very similar NP, GM and TS, compared to the model without SC and DSL as shown in Table 2. The model with SC but without DSL requires significantly larger NP, GM and longer TS, although it has a similar performance to the model with both SC and DSL. Both Table 1 and 2 show the necessity of SC and DSL in achieving good segmentation performance with lower model complexity, less memory consumption and quicker training speed.

4.4 Comparison to the state-of-the-art methods

Next, we compared the performance of our method ConvTransSeg (CTS) with the state-of-the-art methods. We chose one representative method with highly reported performance from each of the categories described in section 2. These methods are FAT-Net [18] from the category Transformer in encoder, SegTrans [33] from the category Transformer in decoder, TransUnet [16] from the category Transformer as a bridge, and U-Net [37] from the category Fused CNN and Transformer. SwinUnet [39] from the category Pure Transformer. Additionally, we also compared our model with the pure CNN method ResUnet [9]. For a fair comparison, data augmentation was not applied in the training process in any of the methods. Additionally, besides using the input size of $224 \times 224$, we also trained our model and ResUnet with input size of $256 \times 256$ for comparison with U-Net [37]. The mean and standard deviation of DC and ASSD are reported in section 4.4.1 to compare the segmentation accuracy of all methods. Additionally, NP, GM and TS are reported in section 4.4.2 to compare the models’ complexity and training efficiency. Finally, some qualitative results are shown in section 4.4.3 to demonstrate the advantages and drawbacks of each method visually.

4.4.1 Segmentation accuracy

Table 3: Comparison of model performance on five public datasets. Mean and Standard deviation of Dice coefficient (DC) and Average Symmetric Surface Distance (ASSD) are reported. * indicates that the method used $256 \times 256$ as the input size, otherwise $224 \times 224$ was used. Numbers in bold are the best methods in each input size, † indicates that the result is significantly worse than the best results ($p < 0.01$ measured by WSRT).

| Model | ISIC | Bowel | Pannuke | CVC | OASIS |
|-------|------|-------|---------|-----|-------|
|       | DC   |       |         |     |       |
| ResUnet | 0.831| ±0.015| ±0.073 | ±0.100| ±0.225| ±0.011|
| SwinUnet | 0.877| 0.905| 0.516 | 0.792| 0.923| 0.031|
| FAT-Net | 0.884| 0.901| 0.495 | 0.894| 0.775| 0.031|
| SegTrans | 0.874| 0.895| 0.526 | 0.835| 0.906| 0.031|
| TransUnet | 0.894| 0.900| 0.500 | 0.872| 0.917| 0.031|
| CTS (ours) | 0.900| 0.917| 0.551| 0.900| 0.933| 0.031|
| ResUnets | 0.813| 0.919| 0.525| 0.781| 0.943| 0.031|
| ResUnets* | 0.917| 0.903| 0.487| 0.927| 0.951| 0.031|
| U-Net* | 0.879| 0.837| 0.553| 0.862| 0.936| 0.031|
| CTS (ours)* | 0.885| 0.925| 0.564| 0.809| 0.943| 0.031|

From the results reported in Table 3, it is seen that our method achieved the best mean DC and mean ASSD on four out of five datasets (except for OASIS) using both input image sizes. Our method ranked the second after
ResUNet and the third after ResUNet and SwinUnet on the OASIS dataset using DC and ASSD measurements respectively, when the input image size is $224 \times 224$. When a larger input image size ($256 \times 256$) is used, our method achieved the same performance as ResUnet and performed better than UT-net using DC measure, and it performed the best using ASSD measure. In conclusion, our method (CTS) is overall the best method when evaluated on different datasets with varying input image size based on DC and ASSD measures.

We also ran statistical tests (WSRT) in comparing each method to the best method for each dataset indicated by $\dagger$ in Table 3. It can be seen that our method is the best method in most datasets. Although it is not always significantly better ($p < 0.01$ by WSRT) than the second best method, the second best method is not always the same in these datasets.

### 4.4.2 Model complexity

Table 4 presents the measurements of NP, GM and TS for comparing the number of parameters, memory consumption and training speed respectively. It is seen that our method requires the least number of learning parameters compared to other Transformer based methods. Although the pure Transformer SwinUnet consumed the least GPU memory and had the quickest training speed, it performed worse than our method in 4 out of 5 datasets and similar performance on the remaining dataset (OASIS) in Table 3. More importantly, although our method is not the best method in terms of model complexity and training speed, pre-training is not needed in our method. The only Transformer based method that does not require pre-training is UUnet, but it has many more learnable parameters and performed worse in all datasets than our method.

| Model         | nP/m | GM/GB | TS (s) |
|---------------|------|-------|--------|
| ResUNet       | 13.04| 1.39  | 0.120  |
| SwinUnet      | 27.17| 0.94  | 0.110  |
| FAT-Net       | 28.76| 1.01  | 0.132  |
| SegTran       | 50.47| 0.96  | 0.131  |
| TransUnet     | 105.28| 2.46 | 0.174  |
| CTS (ours)    | 21.48| 2.21  | 0.139  |
| ResUNet*      | 13.04| 1.39  | 0.161  |
| UTnet*        | 57.46| 2.67  | 0.250  |
| CTS (ours)*   | 21.60| 3.03  | 0.161  |

### 4.4.3 Qualitative analysis

To find the dataset on which the performances of different methods differ the largest, we calculated the sum of pair-wise absolute difference of the DC values of different methods for each dataset reported in Table 3. The CVC-ClinicDB dataset was identified using such a measure, hence some examples were extracted from this dataset for qualitative comparison.

To visualize the performance differences between different methods, we firstly plot the distributions of DC values of all methods on CVC dataset in Fig 4. The plot illustrates that CTS (our method) and FAT-net have more high-quality segmentation results ($DC > 0.9$) than other methods. ResUnet and SegTran have more low-quality segmentation results ($DC < 0.1$) than other methods. SegTran also uses Transformer in decoder but different from our method, it reveals the strength of our proposed new model structure.

From both Table 3 and Fig 4, it can be seen that ResUNet performed much worse than all other Transformer-based methods on the CVC dataset. The CVC dataset is for polyp detection, which contains many challenging cases where the foreground and background share very similar image intensities and feature patterns. In this case, long range image dependencies and interactions play an important role in distinguishing foreground from background when they share similar features.

Furthermore, we selected some representative cases where the DC values of different methods have large standard deviations, as shown in Fig 5. From Fig 5(a), it can be seen that when the background is dark, ResUNet produced no predictions. Our method was influenced by the noise (the reflection part), which resulted in over-segmentation in the bottom part. The results from both TransUnet and SwinUnet were more similar to the ground truth mask. However arguably, it is a challenging case even for human annotators to produce a reliable ground truth. In Fig 5(b), the target object shares very similar image intensity with a large part of the background. Our method produced the best results compared to other methods. Additionally, Fig 5(c) represents a case of large region of interest. In this case, SegTran and FAT-net performed well and our method under-segmented the target with the lowest DC value. However, our method managed to detect one of the poly regions correctly. The example in Fig 5(d) can be
Figure 5: Some predictions on CVC-ClinicDB that show large difference between ResUNet and our method. The images from left to right is the original data, ground truth, predictions of our method, ResUNet, FAT-net, UT-net, SegTran, TransUnet, and SwinUnet. ‘Decoder’, ‘PureC’, ‘Encoder’, ‘Fusion’, ‘Bridge’, ‘PureT’ indicates the category Transformer in decoder, Pure CNN, Transformer in encoder, Fused CNN and Transformer, Transformer as a bridge, Pure Transformer respectively.

considered as the situation where the target is difficult to be distinguished from the background, where only ResUNet and our method can make some correct predictions in the target region. Although ResUNet has higher dice coefficient, our method captures the location of the target more correctly. Overall, SegTran (also using Transformer as decoder) performs the worst on most cases, which reveals the advantage of our new method of using Transformer as decoder.

5 Discussion and conclusions

In this work, we have proposed a hybrid model called ConvTransSeg that combines components of Transformer and CNN. By taking the advantages of both attention mechanism and CNN, our method can learn the long-term dependency of multi-resolution local features. We have evaluated our method and compared its segmentation accuracy, model complexity, and training efficiency with pure CNNs, pure Transformer, and other Transformer-CNN hybrid methods on five public datasets of different medical segmentation tasks. The results demonstrate that our method has the least number of learnable parameters with a quick convergence speed. More importantly, it is overall the best method for both DC and ASSD measures based on the results provided in section 4.4.1. We also qualitatively assessed some representative segmentation results of all compared methods on the CVC-ClinicDB dataset. It demonstrates the superiority of our method in segmenting challenging images that the target and background are sharing similar intensities and feature patterns. However, our method is not robust enough to handle noisy regions and tends to predict more false positives on the noisy regions (e.g. Fig 5(a)).

In conclusion, different from other Transformer-based methods, by applying CNN as encoder and utilizing the encoded features in multi-resolution, our Transformer-based decoder can learn from the local representations in different image scales (i.e. both large patches and small patches). Such multi-resolution features can provide rich information for the Transformer decoder to make decisions, even when the dataset has small number of images. Hence, unlike other Transformer-based methods, our method does not require any pre-training to achieve similar or better performance, and the input size of our method is flexible. Currently, there are very few works on applying Transformer to capture 3D information, due to the limitation of GPU memory. Thus, in future work, we will improve the attention mechanism used in our method and extend it to 3D images without consuming too much computational resource. In addition, we will modify our method to improve its robustness to better handle noisy image regions.

References

[1] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. arXiv preprint arXiv:2201.09873, 2022.

[2] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. Inter-
national journal of computer vision, 1(4):321–331, 1988.

[3] James C Gee, Martin Reivich, and Ruzena Bajcsy. Elastically deforming a three-dimensional atlas to match anatomical brain images. 1993.

[4] Yuri Boykov and Marie-Pierre Jolly. Interactive organ segmentation using graph cuts. In International conference on medical image computing and computer-assisted intervention, pages 276–286. Springer, 2000.

[5] S Kevin Zhou, Hayit Greenspan, Christos Davatzikos, James S Duncan, Bram Van Ginneken, Anant Madabhushi, Jerry L Prince, Daniel Rueckert, and Ronald M Summers. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. Proceedings of the IEEE, 109(5):820–838, 2021.

[6] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. Advances in Neural Information Processing Systems, 32, 2019.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[9] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters, 15(5):749–753, 2018.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[11] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In Deep learning in medical image analysis and multimodal learning for clinical decision support, pages 3–11. Springer, 2018.

[12] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In International conference on medical image computing and computer-assisted intervention, pages 424–432. Springer, 2016.

[13] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 fourth international conference on 3D vision (3DV), pages 565–571. IEEE, 2016.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[16] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306, 2021.

[17] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. Medical transformer: Gated axial-attention for medical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 36–46. Springer, 2021.

[18] Huisi Wu, Shihuai Chen, Guilian Chen, Wei Wang, Baiying Lei, and Zhenkun Wen. Fat-net: Feature adaptive transformers for automated skin lesion segmentation. Medical Image Analysis, 76:102327, 2022.

[19] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). arXiv preprint arXiv:1902.03368, 2019.

[20] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilarriño. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. Computerized medical imaging and graphics, 43:99–111, 2015.

[21] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benet, Ali Khuram, and Nasir Rajpoot. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In European congress on digital pathology, pages 11–19. Springer, 2019.

[22] Daniel S Marcus, Tracy H Wang, Jamie Parker, John G Csernansky, John C Morris, and Randy L Buckner. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. Journal of cognitive neuroscience, 19(9):1498–1507, 2007.
