MS-UMLP: Medical Image Segmentation via Multi-Scale U-shape MLP-Mixer

Bin Xie[1], Hao Tang[2], Dawen Cai[3], and Yan Yan[4]

[1] Department of Computer Science, Illinois Institute of Technology, USA
[2] School of Computer Science, Peking University, China
[3] Department of Cell and Developmental Biology, University of Michigan, USA
[4] Department of Computer Science, University of Illinois Chicago, USA
bxie9@hawk.iit.edu, haotang@pku.edu.cn,
dwcai@umich.edu, yyan55@uic.edu

Abstract. With the emergence and rapid development of Transformers, medical image segmentation has also been revolutionized by Transformers due to their ability to encode long-range dependencies. Despite their advantages, Transformers also come with some drawbacks, such as larger models being built, resulting in more parameters being introduced. In some cases, several times the parameters may only result in marginal improvements. Additionally, medical segmentation images typically consist of multiple classes, with significant differences in size among classes and minimal differences within each class, which can be addressed via a multiple-scale model. In this paper, we proposed a novel Multi-Scale U-shape MLP-Mixer network named MS-UMLP, which aims to achieve multiple-scale receptive fields while using fewer parameters. Unlike the prevailing transformer-based trend of building models with more parameters, our MS-UMLP adopts dimension-wise multi-scale MLP-Mixer blocks via redesigning MLP-Mixer [22] to reduce model parameters and computational complexity, retain the ability to exploit long-term dependencies, and provide the ability to capture the different scale information in each block. Extensive experiments show that our MS-UMLP not only has the least number of parameters (only 48% parameters of a pure convolutional network) but also outperforms existing methods on the popular ACDC [4] and Synapse [15] medical image segmentation datasets.

1 Introduction

Medical image segmentation has achieved numerous and remarkable progress with convolution-based models [1, 3, 20] for the past few years. For example, U-Net [20] consists of a symmetric convolution-based encoder-decoder with skip connections that combine shallow, low-level, and fine-grained feature maps to deep, semantic, and coarse-grained feature maps. Afterward, its variants [2, 27] and 3D version [9, 24] have achieved great success in a variety of medical imaging applications. Particularly, nnUNet [14] proposes a robust and self-adapting framework based on 2D and 3D U-Nets for medical image segmentation and achieves great performance. By employing a hierarchical U-shaped structure through the stacking of convolutional layers to leverage multi-scale abilities,

convolution-based models exhibit significant capability in achieving exceptional performance across various medical image segmentation tasks.

Although the U-Net-based networks have been the dominant architectures for many years due to their ability to exploit local-term dependencies and feature extraction, they are inevitably subjected to the limitation of exploitation of long-term dependencies. Recently, with the emergence and rapid development of Transformers, medical image segmentation has also been revolutionized by Transformers due to their ability to encode long-range dependencies. Despite their advantages, Transformers also come with some drawbacks, in which larger models are built, resulting in more parameters being introduced. Unfortunately, several times the parameters usually bring marginal improvement, such as nnFormer [26] utilizes 5× parameters but achieves 0.3% improvement of DSC compared to a pure convolutional network (nnUNet [14]) shown in Fig 1. A natural question arises spontaneously: Is there only one way that building bigger models by Transformers can bring improvement? Specifically, can we design a model with a multi-scale receptive field while using fewer parameters?

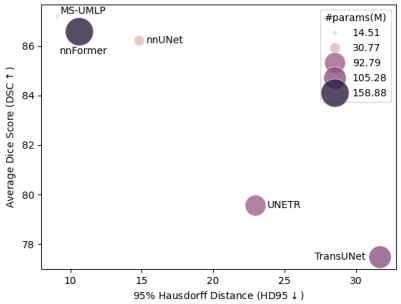

Fig. 1: Performance *vs.* model efficiency on Synapse.

Fortunately, MLP-Mixer [22] brings us a new perspective. MLP-Mixer has been proposed and attracted lots of attention due to cross-patch and cross-channel communication to capture long-term dependencies by *token-mixing* and *channel-mixing*, which only consists of simple multilayer perceptrons (MLPs). In this work, we utilize the idea of *token-mixing* and *channel-mixing* to redeem the limitation of capturing long-term dependencies. However, MLP-Mixer has suffered from quadratic model parameters and computational complexity due to MLPs, so it is not easily adapted to medical image analysis containing 4D information. Meanwhile, the MLP-Mixer crops input images into several non-overlapping patches, and these patches have a relatively large size, such as 16×16 in ViT [10] and MLP-Mixer [22]. Therefore, it is inevitably hard to capture local-term dependencies. The local-term dependencies are significantly important for medical image segmentation since the image segmentation task usually operates at a pixel level. We should encode more local-term dependencies and retain a high image resolution. Meanwhile, the MLP-Mixer does not change the shape among mixer layers. A hierarchical structure achieves a better performance in the segmentation tasks due to the ability to exploit different scale information.

To address those limitations, we propose a novel multi-scale MLP-based U-shape MLP-Mixer architecture for the medical image segmentation task. Specifically, we utilize an MLP-based hierarchical structure as our backbone, which consists of dimension-wise multi-scale MLP-Mixer (dwMSMLP) blocks and convolutional layers for down-sampling and up-sampling. The dwMSMLP consists

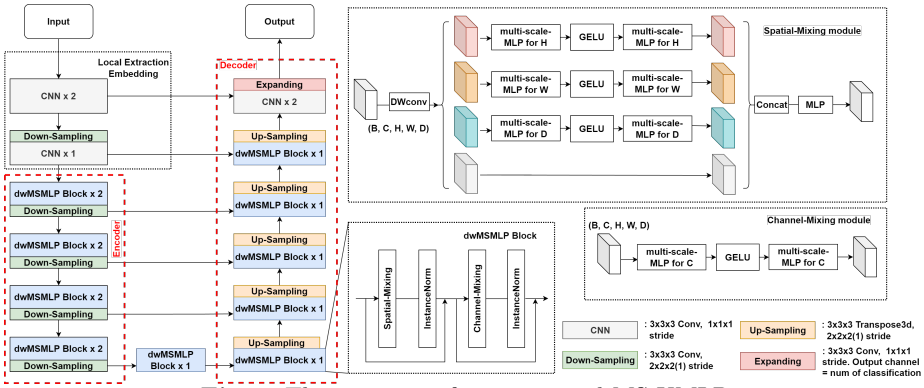

Fig. 2: The overview of our proposed MS-UMLP

of the redesigned spatial-mixing block and channel-mixing block. The spatial-mixing block firstly adopts one depth-wise convolutional layer to capture local dependencies and then decomposes the output with 3D spatial dimensions into three branches to process one single spatial dimension, respectively. Meanwhile, an Identity layer is added for skip connection parallel. Next, each of the three branches will be built with two multi-scale MLP blocks and a GELU [13] activation in the middle. Finally, it concatenates the outputs of three branches and the Identity layer in the channel dimension and then applies an MLP layer on the channel dimension to adjust the size of the channel dimension to the same as the input (from 4C to C). The channel-mixing block consists of two multi-scale MLP blocks worked on the channel dimension and a GELU [13] activation in the middle. In this way, the dwMSMLP can learn different scale information on the spatial and channel dimensions, exploiting the imbalance among multiple classes. Meanwhile, our model reduces lots of parameters and computational complexity via decomposing the 3D spatial dimensions into three single dimensions to process and retain the ability to exploit long-term dependencies.

We replace the non-overlapping patch embedding with the redesigned local extraction embedding, which is built via several successive convolutional layers with a small kernel size to extract local-term dependencies. Meanwhile, it retains a high resolution that contains more spatial information. The non-overlapping patch embedding in ViT [10] and MLP-Mixer [22] inevitably lose some local information, which is important to the decoder to recover pixel-level information in segmentation tasks. Meanwhile, it retains a high resolution that contains more spatial information.

To summarize, the contributions of this paper are:

– We propose a novel multi-scale MLP-based U-shape MLP-Mixer framework for medical image segmentation. To the best of our knowledge, the proposed MS-UMLP is the first MLP-based image segmentation framework;
– We propose a novel dimension-wise multi-scale MLP-Mixer (dwMSMLP) block. For spatial mixing, our dwMSMLP block decomposes an image with three spatial dimensions into every single spatial dimension and adopts multiple scale MLPs in each spatial dimension to learn different scale spatial information. For

channel mixing, our dwMSMLP block also applies for multiple scales MLPs on the channel dimension to learn different scale channel information.
– We conduct extensive experiments on two challenging ACDC [4] and Synapse [15] datasets. The results demonstrate that MS-UMLP not only has the least number of parameters (only 48% parameters of a pure convolutional network), but also achieves state-of-the-art results. The source code and pre-trained models will be made publicly available.

2 Related Work

CNNs for Medical Image Segmentation. Convolutional Neural Networks (CNNs), especially an encoder-decoder network U-Net [20] and its variants, have been demonstrated to achieve excellent performance and play an important role in medical image segmentation. U-Net++ [27] introduces nested and dense skip connections to reduce the semantic gap between the encoder and decoder. Attention U-Net [19] designs an attention gate mechanism that learns to focus on target structures of different shapes and sizes. 3D U-Net [9] modifies 2D U-Net into a 3D version. The V-Net [18] utilizes the residual blocks as the basic convolutional block and a dice loss to optimize the network, which plays an important role in segmentation tasks. The nnUNet [14] is a self-adapting framework for U-Net-based medical image segmentation. Although it only utilizes the basis of 2D and 3D vanilla U-Nets, it involves many tricks for preprocessing, analysis of the attributes of datasets, setting up excellent training strategies, and postprocessing. Particularly, it analyzes datasets to generate an exact architecture and strategies for data augmentation. In this way, it achieves great performance in many segmentation tasks.
MLP-Mixer in Vision Models. MLP-Mixer [22] has been proved that simply multi-layered perceptrons (MLPs) can achieve a high-quality vision results by two types of MLP layers: *channel-mixing* MLPs and *token-mixing* MLPs. The architecture of MLP-Mixers accepts a sequence of patches, i.e., "$patches \times channels$", generated by rearrangement and linear projection. The channel-mixing MLPs build relationships among different channels by operating on each token independently. The token-mixing MLPs build relationships among different spatial locations with tokens by operating on each channel independently. In this way, it learns token-level global information. The MLP-Mixer has achieved state-of-the-art performance in vision tasks such as image generation [6], image classification [17, 23]. For instance, Res-MLP [23] proposes an affine transform layer to achieve a deeper architecture and higher accuracy than MLP-Mixer. gMLP [17] designs a spatial gating unit to increase communication among spatial locations.

3 Methodology

3.1 Framework Overview

The overview of our proposed MS-UMLP architecture is illustrated in Figure 2. Firstly, we utilize the Local Extraction Embedding layer which consists of suc-

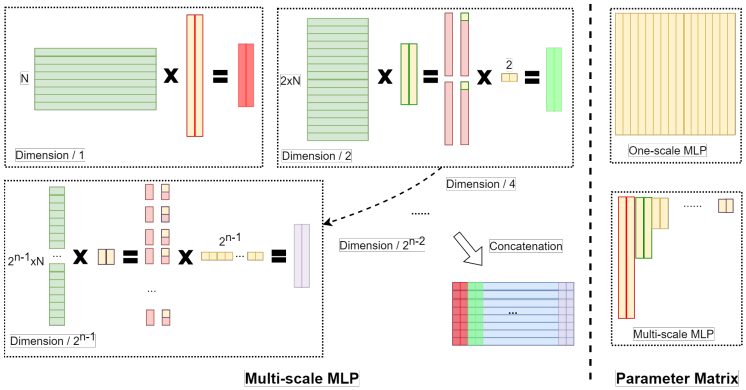

Fig. 3: The overview of the multi-scale MLP.

cessive and small kernel convolutional layers. In this way, our model can learn local-term dependencies, increase the channel dimension size, and retain a high resolution that contains more spatial information compared to the patch embedding in ViT [10] and MLP-Mixer [22]. The proposed encoder consists of several identical stages. Each stage consists of two dwMSMLP (dimension-wise multi-scale MLP-Mixer) blocks and one convolution-based down-sampling that has a small kernel (3×3×3) and stride 2 (the stride for depth dimension maybe 1 due to usually small size of depth dimension in medical images) to decrease two times for spatial dimensions. After the encoder, there are two dwMSMLP blocks as the bottleneck. The proposed decoder also consists of several identical stages. Each stage consists of one dwMSMLP block and one convolution-based up-sampling that the kernel is set to (2×2×2) and the stride 2 (the stride for depth dimension maybe 1) to increase two times for spatial dimensions. Meanwhile, we introduce skip connections from the encoder to the decoder. Finally, the expanding step consists of one convolutional layer to set the channel dimension to the number of segmentation classes.

3.2 Local Extraction Embedding Layer

The non-overlapping patch embedding in ViT [10] or MLP-Mixer [22] uses a large kernel convolutional layer to extract features, which inevitably lose some local information or pixel-wise information. However, the local information or pixel-wise information is significant for medical image segmentation since it needs to be predicted at a pixel level. To fix this limitation, we utilize a Local Extraction Embedding layer, which consists of four successive convolutional layers with a small size kernel to extract local features and maintain a high resolution that contains more spatial information. In the ablation study, we demonstrate that the Local Extraction Embedding layer has a better performance compared to a large non-overlapping patch embedding in the medical image segmentation.

The Local Extraction Embedding layer consists of four convolutional layers with kernel size (3×3×3). The third convolution is for down-sampling, in which the stride is set to (2, 2, 2) or (2, 2, 1) when the number of depth dimensions is too

small. The strides of the rest of the convolutions are set to (1, 1, 1). Specifically, given an image $X \in \mathbb{R}^{C_0 \times H \times W \times D}$ with a spatial resolution of $H \times W \times D$ and C_0 number of channels. Our goal is to predict the corresponding pixel-wise segmentation with size $M \times H \times W \times D$, where M is the number of classes of a segmentation task. Therefore, the output of the embedding $X_E \in \mathbb{R}^{2C_{base} \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2^{(1)}}}$, where C_{base} is set to 32 in our experiments.

3.3 Dimension-wise Multi-scale MLP-Mixer Block

The proposed dwMSMLP block is illustrated at the right of Figure 2, consisting of two main components: The Spatial-Mixing module and the Channel-Mixing module, which are responsible for encoding spatial information and channel information, respectively. Specifically, the dwMSMLP block takes $X \in \mathbb{R}^{C \times \frac{H}{2^m} \times \frac{W}{2^m} \times \frac{D}{r}}$ as an input, where m is the number of down-sampling, r is a number in range $[1, 2^m]$, and C is equal to $Min(C_{base} \times 2^{m-1}, 320)$. Thus, the dwMSMLP block can be formulated as follows,

$$\begin{aligned} U &= X + \text{IN}(\text{Spatial-Mixing}(X)), \\ Y &= U + \text{IN}(\text{Channel-Mixing}(U)), \end{aligned} \tag{1}$$

where $\text{IN}(\cdot)$ is instance normalization and Y is the output of the whole dwMSMLP block. The Spatial-Mixing$(\cdot)$ is the function of the Spatial-Mixing. The Channel-Mixing$(\cdot)$ is the function of the Channel-Mixing.

Multi-scale MLP for the Spatial Mixing The Spatial-Mixing module 1) firstly adopts a depth-wise convolutional layer with a small size kernel to exploit local-term dependencies; 2) then connects with four branches that the first three branches mix only one single dimension of 3D spatial dimensions via two multi-scale MLP blocks with a GELU activation in the middle, and the last branch is an Identity layer for a skip connection; 3) Next, the outputs of all branches are concatenated in the channel dimension; 4) Finally, an MLP layer working on the channel dimension is utilized to adjust the size of the channel dimension to the same as the input X (from 4C to C). When we process the spatial mixing, our model involves multiple scales MLPs to learn different scale spatial information at a dimension-wise level. The multi-scale method is a benefit for the segmentation of multiple classes with variant sizes.

Multi-scale MLP for the Channel Mixing The Channel-Mixing module consists of two multi-scale MLP blocks working on the channel dimension and a GELU activation in the middle. When processing on the channel dimension, the existing methods almost utilize a one-scale and channel-wise operation. In this paper, we involve a multi-scale MLP to learn different scale information on the channel. In the Ablation Study, we demonstrate the multi-scale method for the channel dimension can bring a huge improvement.

Architecture of Multi-scale MLP Block The proposed multi-scale MLP block is illustrated in Figure 3. The number of different scales MLPs n for a certain dimension can be expressed as

$$n = \begin{cases} 2, & dim < 24 \\ 3, & 24 \leq dim < 256 \\ 4, & dim \geq 256 \end{cases} \tag{2}$$

After determining n for a certain dimension of input, our multi-scale MLP block reshapes the processing dimension of the input n times. Each time reshapes the processing dimension divided by 2^i (i=0, 1, ..., n−1) from the spacial dimension to the rest of the dimensions. Then an MLP layer will be adopted. The number of the output feature, named f_{out}, of the MLP layer is equal to the input feature divided by n in the processing dimension, which $f_{out}(i)$ can be expressed as,

$$\begin{cases} dim/n + dim\%n, & i = 0 \\ dim/n, & i > 0 \end{cases} \tag{3}$$

where % is the MOD function. Next, there is an MLP layer to adjust the shape from 2^i to 1. After processing the n branches, all outputs will be concatenated in the processing dimension, which is the same as the input. The multi-scale MLP can be expressed as,

$$\begin{cases} X_i = \text{Reshape}_\text{i}(X), & i = 0, 1, ...n-1, \\ H_i = X_i W_i, & W_i \in \mathbb{R}^{dim/2^i \times f_{out}(i)} \\ C = \text{Concat}(H_1, H_2, ...H_n), & \\ S = \text{Reshape}_{\text{adjust}}(C), & \\ O = SW, & W \in \mathbb{R}^{2^i \times 1} \end{cases} \tag{4}$$

where $\text{Reshape}_\text{i}(\cdot)$ represents the reshape operation that reshapes from $X \in \mathbb{R}^{B \times C \times H \times W \times D}$ to $X_i \in \mathbb{R}^{B \times rest \times 2^i \times dim/2^i}$ where dim is one of (C, H, W, D). $\text{Concat}(\cdot)$ is a concatenation function in the operating dimension, $\text{Reshape}_{\text{adjust}}(\cdot)$ reshape the operating dimension into its original location so that $S \in \mathbb{R}^{B \times C \times H \times W \times D \times 2^i}$. Finally, we utilize an MLP layer W to adjust the dimension as the same as the input.

Unlike a full-connected layer in Mixer-MLP that only has one scale MLP layer, our multi-scale MLP block has multiple scale MLPs to learn different scale information. At the bottom right of Figure 3, it illustrates the parameter matrix of our multi-scale MLP not only learns different scale information but also has a smaller number of parameters than a one-scale MLP layer. Since medical images usually have large differences in size between classes and little difference in sizes within classes, Our model has great advantages in managing such situations.

Relation to MLP-Mixer. Unlike the token-mixing of MLP-Mixer [22] that conducts a linear projection with respect to the multiplication of all the tokens along the spatial dimension, which increases model parameters and computational complexity, we propose a dimension-wise mixing that separately processes the height, width, depth, and channel dimensions. For the mixing of every single

Method	Average		Aotra ↑	Gallbladder ↑	Kidnery(L) ↑	Kidnery(R) ↑	Liver ↑	Pancreas ↑	Spleen ↑	Stomach ↑
	DSC ↑	HD95 ↓								
VNet [18]	68.81	-	75.34	51.87	77.10	80.75	87.84	40.04	80.56	56.98
DARR [11]	69.77	-	74.74	53.77	72.31	73.24	94.08	54.18	89.90	45.96
R50-U-Net [20]	74.68	36.87	87.74	63.66	80.60	78.19	93.74	56.90	85.87	74.16
U-Net [20]	74.99	27.57	83.17	58.74	80.40	73.36	93.13	45.43	83.90	66.59
R50-AttnUNet [21]	75.57	36.97	55.92	63.91	79.20	72.71	93.56	49.37	87.19	74.95
VIT-CUP [10]	67.86	36.11	70.19	45.10	74.70	67.40	91.32	42.00	81.75	70.44
R50-VIT-CUP [10]	71.29	32.87	73.73	55.13	75.80	72.20	91.51	45.99	81.99	73.95
TransUNet [8]	77.48	31.69	87.23	63.16	81.87	77.02	94.08	55.86	85.08	75.62
SwinUNet [5]	79.13	21.55	85.47	66.53	83.28	79.61	94.29	56.58	90.66	76.6
TransClaw-U-Net [7]	78.09	26.38	85.87	61.38	84.83	79.36	94.28	57.65	87.74	73.55
LeVit-UNet-384s [25]	78.53	16.84	87.33	62.23	84.61	80.25	93.11	59.07	88.86	72.76
WAD [16]	80.30	23.12	87.73	69.93	83.95	79.78	93.95	61.02	88.86	77.16
UNETR [12]	79.56	22.97	89.99	60.56	85.66	84.80	94.46	59.25	87.81	73.99
nnUNet [14]	86.21	14.82	92.39	71.71	86.07	**91.46**	95.84	82.92	90.31	79.01
nnFormer [26]	86.57	10.63	**92.40**	70.17	**86.57**	86.25	96.84	**83.35**	90.51	**86.83**
MS-UMLP (Ours)	**87.19**	**9.09**	92.26	**74.20**	86.19	89.15	**96.98**	82.45	**93.00**	83.31

Table 1: Quantitative evaluation with state-of-the-art methods on the Synapse dataset (DSC in % and HD95 in mm).

dimension, it can learn the patterns and information in a certain dimension, which is more efficient. Meanwhile, we redesign the only one-scale linear projection into a multi-scale linear projection. In this way, our model can learn different scale information in one block and decrease the number of parameters compared to a one-scale linear projection. The ability to exploit multiple-scale information is greatly effective in performing medical image segmentation since medical images usually have large differences in size between classes and little difference in sizes within classes.

3.4 Discussion

Comparison with Network Parameters. We compare the network parameters between our dwMSMLP and the vanilla MLP-Mixer [22]. Given the input tensor $T_v \in \mathbb{R}^{(T_H \times T_W \times T_D) \times C}$, it has two dimensions with $(T_H \times T_W \times T_D)$ number of tokens and C number of channels for the vanilla MLP-Mixer, while the input tensor $T_{dw} \in \mathbb{R}^{C \times T_H \times T_W \times T_D}$ has four dimensions with a spatial resolution $T_H \times T_W \times T_D$ and C number of channels for our dwMSMLP.

For the vanilla MLP-Mixer, (i) the token-mixing module has two parameter matrices that have the same number of parameters. The number of parameters of the one dimension of each matrix is equal to $T_H \times T_W \times T_D$ and the other dimension is $e \times (T_H \times T_W \times T_D)$, where e is the expansion factor that we usually set to 4; (ii) The channel-mixing module also has two parameter matrices that have the same number of parameters. The number of parameters of the one dimension of each matrix is equal to C and the second dimension is $e \times C$. Therefore, the total number of matrix is $2 \times (4 \times T_H^2 \times T_W^2 \times T_D^2 + 4 \times C^2)$. Thus, the complexity of vanilla MLP-Mixer can be expressed as,

$$\Theta(\text{vMLP}) \in \Theta(T_H^2 \times T_W^2 \times T_D^2 \times C^2). \tag{5}$$

For our dwMSMLP, (i) the spatial-mixing has three parts built with parameters. The first part is a depth-wise convolution in which the number of parameters

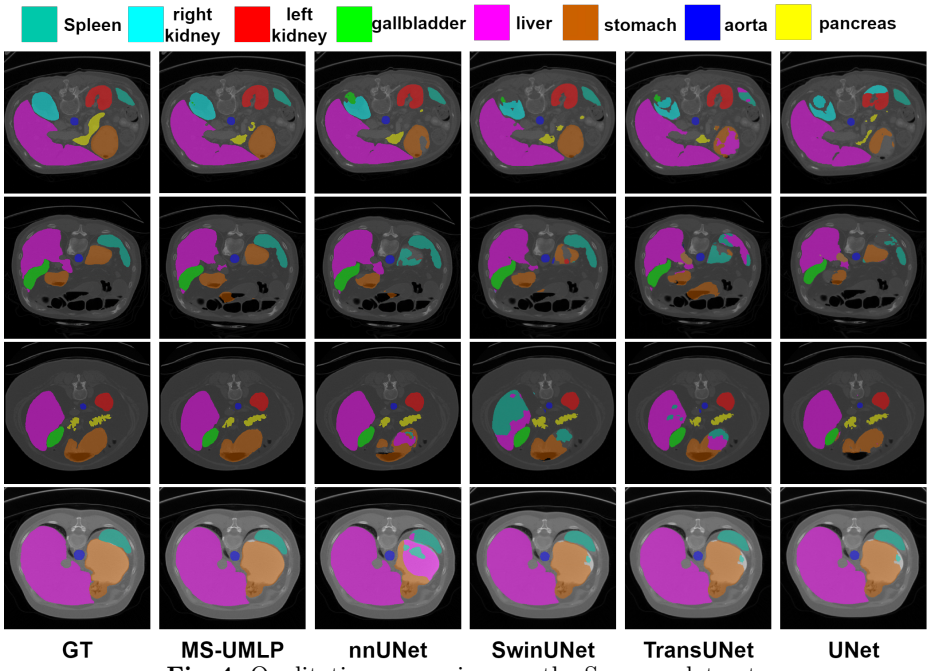

Fig. 4: Qualitative comparison on the Synapse dataset.

is equal to $3\times3\times3\times C$. The second part has four branches for mixing. The total number of parameters for mixing the height dimension via a multi-scale MLP can be expressed as,

$$\Theta(\text{MSMLP}_\text{H}) \in \Theta(2 \times \sum_{i}^{n} \frac{T_H}{2^i} \times \frac{T_H}{f_{out}{}^i} + 2^i) \leq O(2T_H^2). \tag{6}$$

The number of parameters for the width and depth dimensions is the same as the height dimension. Therefore, the parameter of this part is less than $2(T_H^2+T_W^2+T_D^2)$. Since we only take the mixing operation in every single dimension, it is not multiplication but a sum to calculate the number of parameters. The final part is for the fusion of the output of the four branches. We perform a linear projection in the channel dimension. Therefore, the number of parameters is equal to $(4\times C)\times C$; (ii) The channel-mixing module is the same as the multi-scale MLP for the height dimension. The number of parameters of the channel-mixing module is less than $2\times C^2$. Therefore, the total number of parameters of our dwMSMLP is less than $2T_H^2+2T_W^2+2T_D^2+6C^2+9C$, and the complexity of our dwMSMLP,

$$\begin{aligned} \Theta(\text{dwMSMLP}) &\geq O(min(T_H^2, T_W^2, T_D^2, C^2)), \\ \Theta(\text{dwMSMLP}) &\leq O(max(T_H^2, T_W^2, T_D^2, C^2)). \end{aligned} \tag{7}$$

Compared to the complexity of the vanilla MLP-Mixer, our dwMSMLP reduces almost the complexity of $\Theta(T^4)$.

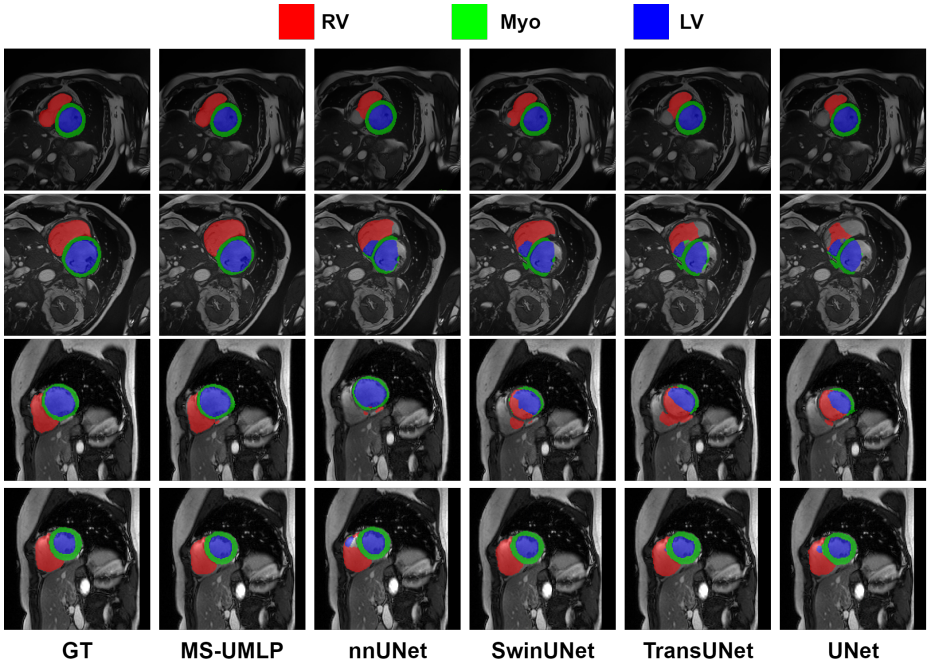

Fig. 5: Qualitative comparison on the ACDC dataset.

4 Experiments

Datasets and Evaluation Metrics. We use two publicly available datasets, Synapse multiorgan segmentation [15] and Automatic Cardiac Diagnosis Challenge (ACDC) [4]. (i) Synapse dataset consists of 30 cases of abdominal CT scans. Following the split strategies [8], we use a random split of 18 training cases and 12 cases for validation. We evaluate the model performance via the average Dice score (DSC) and the 95% Hausdorff Distance (HD95) on 8 abdominal organs (aorta, gallbladder, spleen, left kidney, right kidney, liver, pancreas, and stomach). (ii) ACDC dataset consists of 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. The labels involve the right ventricle (RV), myocardium (MYO), and left ventricle (LV). We use a random split of 70 training cases, 10 validation cases, and 20 testing cases. We evaluate the model performance via the average DSC.

Implementation Details. We utilize some data augmentations such as rotation, scaling, Gaussian noise, Gaussian blur, brightness, and contrast adjustment, simulation of low resolution, gamma augmentation, and mirroring. We set the initial learning rate to 0.01 and employ a "poly" decay strategy in Eq. (8).

$$lr(e) = init\_lr \times (1 - \frac{e}{\text{MAX\_EPOCH}})^{0.9}, \tag{8}$$

where e means the number of epochs, MAX_EPOCH means the maximum of epochs, set it to 1000 and each epoch includes 250 iterations. We utilize SGD

Method	Average ↑	RV ↑	Myo ↑	LV ↑
R50-U-Net [20]	87.55	87.10	80.63	94.92
VIT-CUP [10]	81.45	81.46	70.71	92.18
R50-VIT-CUP [10]	87.57	86.07	81.88	94.75
UNETR [12]	88.61	85.29	86.52	94.02
TransUNet [8]	89.71	88.86	84.54	95.73
SwinUNet [5]	90.00	88.55	85.62	**95.83**
LeViT-UNet-384s [25]	90.32	89.55	87.64	93.76
nnUNet [14]	91.20	89.30	89.09	95.20
MS-UMLP (Ours)	**91.81**	**90.17**	**89.81**	95.45

Table 2: Quantitative evaluation with state-of-the-art methods on the ACDC dataset (dice score in %).

as our optimizer and set the momentum to 0.99. The weighted decay is set to 3e-5. We utilize both cross-entropy loss and dice loss by simply summing them up as the loss function. We utilize instance normalization as our normalization layer. All experiments are conducted using single NVIDIA RTX A5000 GPUs with 24GB memory.

Deep Supervision. Our network is trained with deep supervision when training. Auxiliary lossees are added in the decoder to the last three stages (the three largest resolutions). For each deep supervision output, we downsample the ground truth segmentation mask for the loss computation with each deep supervision output. The final training objective is the sum of all resolutions loss:

$$\mathcal{L} = w_1 \cdot \mathcal{L}_1 + w_2 \cdot \mathcal{L}_2 + w_3 \cdot \mathcal{L}_3 \tag{9}$$

where the weights halve with each decrease in resolution (*i.e.,* $w_2 = \frac{1}{2} \cdot w_1; w_3 = \frac{1}{4} \cdot w_1$, etc), and all weight are normalized to sum to 1. Meanwhile, the resolution of $\mathcal{L}_1$ is equal to $2 \cdot \mathcal{L}_2$ and $4 \cdot \mathcal{L}_3$

4.1 Comparison with State-of-the-Art Methods

Results on Synapse Dataset. In Table 1, we present the quantitative experimental results on the Synapse dataset compared to several leading convolution-based methods (*i.e.*, VNet [20] and nnUNet [14]) and transformer-based methods (*i.e.*, TransUNet [8], SwinUNet [5], and LeViT-UNet-384s [25]). We observe that our proposed MS-UMLP framework achieves much better results than existing methods. Meanwhile, our model predicts well different size classes (*e.g.*, large-size 'Liver' label, middle-size 'Spleen' label, and small-size 'Aotra' label), which demonstrates our model is effective for multi-scale tasks. In Figure 4, we illustrate the qualitative results compared with several representative methods. These results also demonstrate that our MS-UMLP model can predict more accurately the large-size 'Liver' label, middle-size 'Spleen' label, and small-size 'Aotra' label. Meanwhile, our model achieves the best performance in HD95 and outperforms the second place by more than 1.5 mm. Therefore, the results demonstrate the effectiveness of our method.

Networks	#param (M) ↓	Layer	#param (K) ↓
nnUNet [14]	30.77	Conv3d	27.68
UNETR [12]	92.79	MHSA	1.87×10^3
TransUNet [8]	105.28	MLP-Mixer	1.26×10^6
nnFormer [26]	158.88	dwMLP	11.42
MS-UMLP (Ours)	**14.51**	dwMSMLP	**9.03**

Table 3: Comparisons with *different networks* and *different layers* via parameters.

Results on ACDC Dataset. In Table 2, we provide the quantitative experimental results on ACDC dataset. Specifically, we compare the proposed MS-UMLP with several leading convolution-based methods (*i.e.*, R50-U-Net [20] and nnUNet [14]) and transformer-based methods (*i.e.*, TransUNet [8], SwinUNet [5], and LeViT-UNet-384s [25]). The results show that the proposed MLP-based MS-UMLP framework outperforms other existing baselines. In Figure 5, we provide the qualitative results compared with several state-of-the-art methods. As shown in Figure 5, our MS-UMLP model can predict more accurately on the 'RV' label. Meanwhile, the results demonstrate the effectiveness of our method.

4.2 Comparison with Network Parameters

We perform experiments to compare the number of parameters. Table 3 presents the comparisons among different networks with a shape (1, 96, 96, 96) tensor as the input and different layers with a shape (32, 14, 32, 28) tensor as the input. At the left of Table 3, it demonstrates that our MS-UMLP model utilizes the least number of parameters, even less than a pure 3D convolution-based network (nnUNet [14]) since we only utilize MLPs with a small size to build our MS-UMLP. Other transformer-based methods such as TransUNet [8], UNETR [12] and nnFormer [26] have 105.28M, 92.79M and 158.88M parameters, respectively. The results illustrate that transformer-based methods have a large number of parameters. At the right of Table 3, it demonstrates that our proposed dwMSMLP block has the least number of parameters. Compared to a 3D convolutional layer, our dwMSMLP block decreases almost three times the number of parameters. In conclusion, our MS-UMLP not only has the least number of parameters but also achieves the best performance.

4.3 Ablation Study

Variants of Network Architectures We also evaluate the effectiveness of several variants of network architectures.
(A) Baseline Models. The proposed MS-UMLP has 5 baselines (*i.e.*, S1, S2, S3, S4, S5) as shown in Table 4. (i) S1 adopts a non-overlapping patch embedding, 12 vanilla MLP-Mixer blocks as the encoder, and a CNN-based decoder. (ii) S2 utilizes a non-overlapping patch embedding, 12 dimension-wise MLP blocks (dwMLP) that are the same as our dwMSMLP except replacing

Method	DSC ↑	HD95 ↓
S1 Patch-EM + vMLP×12 + CNN-Decoder	80.79	34.49
S2 Patch-EM + dwMLP×12 + CNN-Decoder	81.85	24.93
S3 Patch-EM + U-shape dwMLP	83.57	15.46
S4 LE-Em + U-shape dwMLP	84.54	17.18
S5 LE-Em + U-shape dwMSMLP-Spatial	86.38	16.67
S6 Our Full Model	**87.19**	**9.09**

Table 4: The ablation studies of the proposed method on the Synapse dataset. The vMLP means the vanilla MLP-Mixer. The LE-Em means the local extraction embedding. The dwMLP means that we replace the multi-scale MLP with a one-scale MLP in our dwMSMLP. The U-shape means building a U-shape network. dwMSMLP-Spatial means only utilizing the multi-scale MLP on spatial dimensions

the multi-scale MLP with a one-scale MLP (a full-connected layer), and a CNN-based decoder. (iii) S3 utilizes a non-overlapping patch embedding and several dwMLP blocks to build a U-shape network, similar to our model. (iv) S4 utilizes a Local Extraction Embedding (LE-Embedding) block and several dwMLP blocks to build a U-shape network. (v) S5 utilizes a LE-Embedding block and several dwMSMLP-Spatial blocks that are the same as dwMSMLP except utilizing one-scale MLPs for the Channel-Mixing to build a U-shape network. (vi) S6 is our full model, named MS-UMLP, illustrated in Figure 2. Our model utilizes LE-Embedding and several dwMSMLP blocks that utilize multi-scale MLPs for the Spatial-Mixing and Channel-Mixing to build a U-shape network. The results of the ablation study are shown in Table 4.

(B) Effect of LE-Embedding. When we use a LE-Embedding to extract local information, the average DSC of S4 improves by 1.0% compared with S3. The result confirms the effectiveness of the proposed LE-Embedding.

Part (1)	Part (2)	Part (3)	DSC ↑	HD95 ↓
DWconv	Multi-scale MLP	Concat	**87.19**	**9.09**
		Sum	85.67	12.30
	One-scale MLP	Concat	84.54	17.18
		Sum	84.97	12.27
None	Multi-scale MLP	Concat	85.63	13.71
		Sum	85.45	18.75
	One-scale MLP	Concat	84.03	15.46
		Sum	83.19	13.61

Table 5: The ablation studies of dwMSMLP on Synapse.

(C) Effect of Dimension-wise MLP-Mixer. To reduce the model parameters and computational complexity, we replace the vanilla MLP-Mixer block with a dimension-wise MLP-Mixer block. S2 utilizes a dimension-wise one-scale MLP-Mixer block and improves 1.1% compared with S1 which utilizes a vanilla MLP-Mixer. Meanwhile, all dimension-wise MLP-Mixer blocks are better than S1. It demonstrates a dimension-wise MLP-Mixer block achieves better performance than the vanilla MLP-Mixer block.

(D) Effect of a Hierarchical Structure built by MLP-based blocks. S3, S4, S5, and S6 utilize a hierarchical structure built by dimension-wise MLP-Mixer blocks. The performance of these models improves by more than 2% com-

pared to S1 without a hierarchical structure and S2 with a CNN-based decoder, which demonstrates the effectiveness of the hierarchical structure built by MLP-based blocks.
(E) Effect of multi-scale MLP-Mixer for spatial dimensions. When we adopt the multi-scale MLP-Mixer for spatial dimensions, the performance has a huge improvement. The average DSC of S5 improves by 2% compared to S4, which confirms the benefits of the multi-scale MLP-Mixer.
(F) Effect of multi-scale MLP-Mixer for the channel dimension. S7 is our full model, MS-UMLP, utilizing a LE-Embedding and built by the dimension-wise multi-scale MLP-Mixer for the spatial dimensions and channel dimension as shown in Figure 2. Compared to S5, our model utilizes a multi-scale MLP-Mixer for the channel dimension, which brings 1% improvements. Therefore, the results demonstrate the effectiveness of our proposed MS-UMLP.

Variants of the Spatial-Mixing Module The variants of the Spatial-Mixing module focus on three parts at the upper right of Figure 2. Part 1 is the local information extraction performed with and without depth-wise convolution (DWconv). Part 2 is about utilizing one-scale MLP or multi-scale MLP during dimension-mixing. Part 3 is the fusion of the output of dimension-mixing.

The experimental results of the ablation studies are summarized in Table 5. Table 5 shows that involving a depth-wise convolution brings more than 1% improvement, the multi-scale MLP brings more than 1.5% improvement compared to the one-scale MLP, and the concatenation is better than the 'Sum' operation. We conclude that the depth-wise convolution, multi-scale MLP, and concatenation as the fusion of the output will improve the performance. Thus, we utilize the three parts in our dwMSMLP block.

5 Conclusion

In this paper, we propose a novel multi-scale MLP-based U-shape MLP-Mixer network (*i.e.*, MS-UMLP) for medical image segmentation. Particularly, we introduce a novel dimension-wise multi-scale MLP-Mixer (dwMSMLP) block to enhance the ability of local information extraction and mix dimension-wise multiple different scale information for the spatial and channel dimensions, which is suitable for dealing with medical image segmentation. By the decomposition of the 3D spatial dimensions into three single dimensions to process, our method extremely mitigates the cost of memory. Specifically, it costs less memory than a pure 3D convolutional layer. Extensive experiments on ACDC [4] and Synapse [15] datasets show that MS-UMLP not only has the least number of parameters (even only 48% parameters of a pure 3D convolution-based network) but also achieves state-of-the-art results.
Acknowledgements: This research is supported by NSF IIS-2309073 and NIH 1RF1MH133764-01. This article solely reflects the opinions and conclusions of its authors and not the funding agencies.

References

1. Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D.L., Erickson, B.J.: Deep learning for brain mri segmentation: state of the art and future directions. Journal of digital imaging **30**(4), 449–459 (2017)
2. Alom, M.Z., Yakopcic, C., Hasan, M., Taha, T.M., Asari, V.K.: Recurrent residual u-net for medical image segmentation. Journal of Medical Imaging **6**(1), 014006 (2019)
3. Avendi, M., Kheradvar, A., Jafarkhani, H.: A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri. Medical image analysis **30**, 108–119 (2016)
4. 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)
5. 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)
6. Cazenavette, G., De Guevara, M.L.: Mixergan: An mlp-based architecture for unpaired image-to-image translation. arXiv preprint arXiv:2105.14110 (2021)
7. 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)
8. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
9. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
11. 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)
12. Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R., Xu, D.: Unetr: Transformers for 3d medical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 574–584 (2022)
13. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
14. 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)
15. 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)

16. Li, Y., Cai, W., Gao, Y., Hu, X.: More than encoder: Introducing transformer decoder to upsample. arXiv preprint arXiv:2106.10637 (2021)
17. Liu, H., Dai, Z., So, D.R., Le, Q.V.: Pay attention to mlps. arXiv preprint arXiv:2105.08050 (2021)
18. 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)
19. 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)
20. 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)
21. 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)
22. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., et al.: Mlp-mixer: An all-mlp architecture for vision. arXiv preprint arXiv:2105.01601 (2021)
23. Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Joulin, A., Synnaeve, G., Verbeek, J., Jégou, H.: Resmlp: Feedforward networks for image classification with data-efficient training. arXiv preprint arXiv:2105.03404 (2021)
24. Xiao, X., Lian, S., Luo, Z., Li, S.: Weighted res-unet for high-quality retina vessel segmentation. In: 2018 9th international conference on information technology in medicine and education (ITME). pp. 327–331. IEEE (2018)
25. 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)
26. Zhou, H.Y., Guo, J., Zhang, Y., Yu, L., Wang, L., Yu, Y.: nnformer: Interleaved transformer for volumetric segmentation. arXiv preprint arXiv:2109.03201 (2021)
27. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, pp. 3–11. Springer (2018)