2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI) | 978-1-6654-7358-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISBI53787.2023.10230496

VIG-UNET: VISION GRAPH NEURAL NETWORKS FOR MEDICAL IMAGE SEGMENTATION

Juntao Jiang[1], Xiyu Chen[2], Guanzhong Tian[3] and Yong Liu [1]**

1. College of Control Science and Engineering, Zhejiang University, Hangzhou, China
2. Polytechnic Institute, Zhejiang University, Hangzhou, China
3. Ningbo Innovation Center, Zhejiang University, Ningbo, China

ABSTRACT

Deep neural networks have been widely used in medical image analysis and medical image segmentation is one of the most important tasks. U-shaped neural networks with encoder-decoder are prevailing and have succeeded greatly in various segmentation tasks. While CNNs treat an image as a grid of pixels in Euclidean space and Transformers recognize an image as a sequence of patches, graph-based representation is more generalized and can construct connections for each part of an image. In this paper, we propose a novel ViG-UNet, a graph neural network-based U-shaped architecture with the encoder, the decoder, the bottleneck, and skip connections. The downsampling and upsampling modules are also carefully designed. The experimental results on ISIC 2016, ISIC 2017 and Kvasir-SEG datasets demonstrate that our proposed architecture outperforms most existing classic and state-of-the-art U-shaped networks.

***Index Terms*—** Medical image segmentation, ViG-UNet, Graph neural networks, Encoder-decoder

1. INTRODUCTION

Recent years have witnessed the rise of deep learning and its broader applications in computer vision tasks. As one of the most heated topics of computer vision applied in medical scenarios, image segmentation, identifying the pixels of organs or lesions from the background, plays a crucial role in computer-aided diagnosis and treatment, improving efficiency and accuracy.

Currently, medical image segmentation methods based on deep learning mainly use fully convolutional neural networks (FCN) with U-shaped encoder-decoder architecture such as U-Net [1] and its variants. Composed of a symmetric encoder-decoder with skip connections, U-Net uses convolutional layers and downsampling modules for feature extraction, while convolutional layers and upsampling modules for pixel-level semantic classification. The skip connection operation can maintain spatial information from a high-resolution feature, which may be lost in downsampling. Following this work and based on a fully convolutional structure, a lot of U-Net's variants like Attention-UNet [2], UNet++ [3] and so on, have been proposed and achieved great success. Recently, as Transformer-based methods like ViT [4] achieved good results in image recognition tasks, thanks to their capability of enhancing global understanding of images, extracting information from the inputs and their interrelations, the Transformer-based medical image segmentation models such as Trans-UNet [5] and Swin-UNet [6] also have been proposed and showed competitive performance.

While CNNs treat an image as a grid of pixels in Euclidean space and Transformer recognizes an image as a sequence of patches, graph-based representation can be more generalized and reflect the relationship of each part in an image. Since the graph neural network (GNN) [7] was first proposed, the techniques for processing graphs have been researched a lot. A series of spatial-based GCNs [8, 9] and spectral-based GCNs [10, 11, 12, 13] are widely proposed and applied. In recent work, Han et al.[14] proposed a Vision GNN (ViG), which splits the image into many blocks regarded as nodes and constructs a graph representation by connecting the nearest neighbors, then uses GNNs to process it. It contains Grapher modules with graph convolution to aggregate and update graph information and Feed-forward Networks (FFNs) modules with fully connected networks for node feature transformation, which performed well in image recognition tasks. ViG-S has achieved 0.804 Top-1 accuracy and ViG-B has achieved 0.823 on ImageNet [15].

Motivated by the success of ViG model, we propose a ViG-UNet to utilize the powerful functions of ViG for 2D medical image segmentation in this work. The graph-based representation can also be effective in segmentation tasks. ViG-UNet is a GNN-based U-shaped architecture consisting of the encoder, the bottleneck and the decoder, with skip connections. We do comparison experiments on ISIC 2016 [16], ISIC 2017 [17] and Kvair-SEG [18] datasets.The results show that the proposed model outperformed most existing classic and state-of-the-art methods. The code will be released at *https://github.com/juntaoJianggavin/ViG-UNet*.

*Corresponding authors

Authorized licensed use limited to: Zhejiang University. Downloaded on October 12,2023 at 06:04:37 UTC from IEEE Xplore. Restrictions apply.

2. METHODS

2.1. Architecture Overview

ViG-UNet is a U-shape model with symmetrical architectures, whose architecture can be seen in Figure 1. It consists of structures of the encoder, the bottleneck, the decoder and skip-connections. The basic modules of ViG-UNet are the stem block, Grapher Modules, Feed-forward Networks (FFNs), downsampling and upsampling modules. The detailed settings of ViG-UNet can be seen in Table 1, where D means feature dimension, E means the numbers of convolutional layers in FFNs, K means the number of neighbors in GCNs, $H \times W$ means the output image size.

2.2. Stem Block, Upsampling, Downsampling and Skip-Connections

In the stem block, two convolutional layers are used with stride 1 and stride 2, respectively. The output features have height and width equal to $\frac{H}{2}$ and $\frac{W}{2}$, where H, W are the original height and width of the input image. And the position embedding is added. We used a convolutional layer with stride 2 for downsampling operation and a bilinear for upsampling operation with the scale factor 2 following with a convolutional layer. The output of each FFN in the encoder is added to the output of the FFN in the decoder.

2.3. Grapher Module

Vision GNN first builds an image's graph structure by dividing it into N patches, converting them into feature vectors, and then recognizing them as a set of nodes $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$. A K nearest neighbors method is used to find K nearest neighbors $\mathcal{N}(v_i)$ for each node v_i. An edge e_{ji} is added from v_j to v_i for all $v_j \in \mathcal{N}(v_i)$. In this way, a graph representation of an image $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is obtained, where $e_{ji} \in \mathcal{E}$. For the constructed graph representation $\mathcal{G} = G(X)$ and the input feature x_i, the aggregation operation calculates the representation of a node by aggregating features of neighboring nodes. Then the update operation merge the aggregated feature. The updated feature $\mathbf{x}_i'$ can be represented as:

$$\mathbf{x}_i' = h\left(\mathbf{x}_i, g\left(\mathbf{x}_i, \mathcal{N}\left(\mathbf{x}_i\right); W_{aggregate}\right); W_{\text{update}}\right), \quad (1)$$

where $W_{aggregate}$ and W_{update} are the learnable weights of the aggregation and update operations.

$$g(\cdot) = \mathbf{x}_i'' = \left[\mathbf{x}_i, \max\left(\{\mathbf{x}_j - \mathbf{x}_i \mid j \in \mathcal{N}\left(\mathbf{x}_i\right)\}\right)\right], \quad (2)$$

and

$$h(\cdot) = \mathbf{x}_i' = \mathbf{x}_i'' W_{\text{update}} + b_h, \quad (3)$$

b_h is the bias. And following the design of original ViG networks, the $g(\cdot)$ operation uses the max-relative graph convolution [19].

Table 1. Detailed settings of ViG-UNet

Module	Output size	Architecture
Stem	$\frac{H}{2} \times \frac{W}{2}$	Conv×2
Grapher + FFN	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} D=32 \\ E=2 \\ K=9 \end{bmatrix}$
Downsampling	$\frac{H}{4} \times \frac{W}{4}$	Conv
Grapher + FFN	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} D=64 \\ E=2 \\ K=9 \end{bmatrix}$
Downsampling	$\frac{H}{8} \times \frac{W}{8}$	Conv
Grapher + FFN	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} D=128 \\ E=2 \\ K=9 \end{bmatrix}$
Downsampling	$\frac{H}{16} \times \frac{W}{16}$	Conv
Grapher + FFN	$\frac{H}{16} \times \frac{W}{16}$	$\begin{bmatrix} D=256 \\ E=2 \\ K=9 \end{bmatrix}$
Downsampling	$\frac{H}{32} \times \frac{W}{32}$	Conv
Grapher ×2	$\frac{H}{32} \times \frac{W}{32}$	$\begin{bmatrix} D=512 \\ K=9 \end{bmatrix} \times 2$
Upsampling	$\frac{H}{16} \times \frac{W}{16}$	bilinear + Conv
Grapher + FFN	$\frac{H}{16} \times \frac{W}{16}$	$\begin{bmatrix} D=256 \\ E=2 \\ K=9 \end{bmatrix}$
Upsampling	$\frac{H}{8} \times \frac{W}{8}$	bilinear + Conv
Grapher + FFN	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} D=128 \\ E=2 \\ K=9 \end{bmatrix}$
Upsampling	$\frac{H}{4} \times \frac{W}{4}$	bilinear + Conv
Grapher + FFN	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} D=64 \\ E=2 \\ K=9 \end{bmatrix}$
Upsampling	$\frac{H}{2} \times \frac{W}{2}$	bilinear + Conv
Grapher + FFN	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} D=32 \\ E=2 \\ K=9 \end{bmatrix}$
Final Layer	$H \times W$	bilinear + Conv

Authorized licensed use limited to: Zhejiang University. Downloaded on October 12,2023 at 06:04:37 UTC from IEEE Xplore. Restrictions apply.

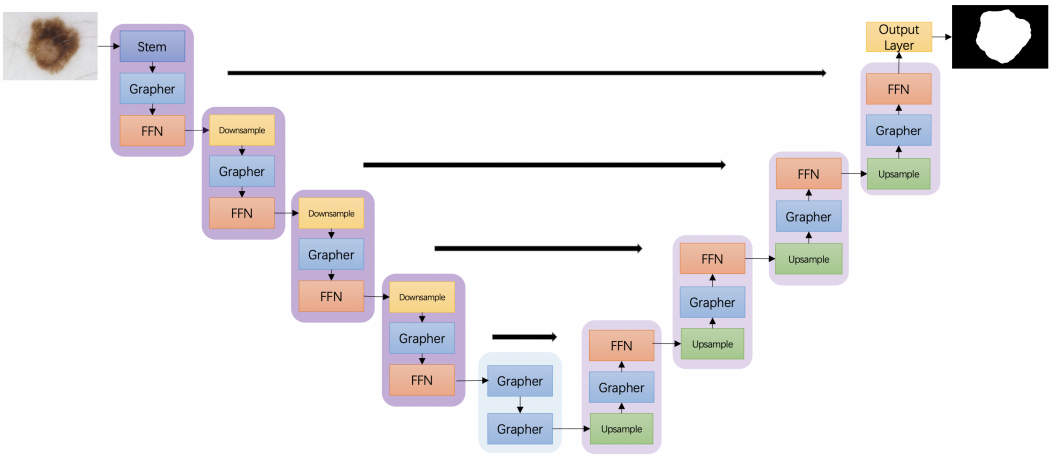

Fig. 1. The architecture of ViG-UNet: the basic modules are the stem block for visual embedding, Grapher Modules, Feed-forward Networks (FFNs) modules, downsampling modules in the encoder and upsampling modules in the decoder.

The aggregated feature $\mathbf{x}_i''$ is split into h heads, then each head is updated with different weights. Then the updated feature $\mathbf{x}_i'$ can be obtained by concatenating all the heads:

$$\begin{aligned}\mathbf{x}_i' = [&\mathbf{x}_{i\text{head}1}'' W_{\text{update}}^1 + b_{h1}, \mathbf{x}_{i\text{head}\,2}'' W_{\text{update}}^2 + b_{h2}, \\ &\mathbf{x}_{i\text{head}\,3}'' W_{\text{update}}^3 + b_{h3}, \cdots \mathbf{x}_{i\text{head}h}'' W_{\text{uphate}}^h + b_{hh}]\end{aligned} \tag{4}$$

where $\mathbf{x}_{i\text{head}1}'', \mathbf{x}_{i\text{head}2}'', \cdots, \mathbf{x}_{i\text{head}h}''$ represent the split heads from $\mathbf{x}_i'$, $W_{\text{update}}^1, W_{\text{update}}^2, \cdots, W_{\text{update}}^h$ represent different weights and $b_{h1}, b_{h2}, \cdots, b_{hh}$ represent different biases.

For the input feature X, the output feature Y after a Grapher module can be represented as:

$$X_1 = XW_{in} + b_{in}, \tag{5}$$

$$Y = \text{Droppath}(\text{GELU}(\text{GraphConv}(X_1)W_{\text{out}} + b_{out}) + X, \tag{6}$$

where Y has the same size as X, W_{in} and W_{out} are the weights. The activation function used is GELU [20]. b_{in} and b_{out} are biases. In the implementation, all $InputW + b$ are achieved by using a convolutional layer following a batch normalization operation. GraphConv means aggregating and updating the discussed graph-level processing. The Grapher module is with a shortcut structure. The Droppath [21] operation is used.

2.4. Feed-forward Network

Feed-forward Networks (FFNs) are used to help with the feature transformation capacity and relief the over-smoothing phenomenon after the Grapher module. The FFN can be represented as

$$Z = \text{Droppath}(\text{GELU}\left(YW_1 + b_1\right)W_2 + b_2) + Y, \tag{7}$$

where W_1 and W_2 are weights, b_1 and b_2 are biases. In the implementation, each feed-forward operation $InputW + b$ is achieved by using a convolutional layer following a batch normalization operation. The Droppath operation is used. The workflow of Grapher and FFN modules is shown in Figure2.

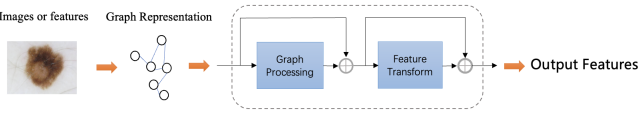

Fig. 2. The workflow of Grapher and FFN modules: graph processing and feature transformation are applied

3. EXPERIMENTS

3.1. Datasets

ISIC 2016 is a dataset of dermoscopic images of skin lesions. We used the dataset of lesion segmentation in this paper. There are 900 pairs of images and corresponding masks in the training set. In the testing set, there are 379 pairs. **ISIC 2017** is a dataset of dermoscopic images of skin lesions. We used the dataset of the lesion segmentation task, which contains images and corresponding masks. There are 2000 pairs in the training set, 150 in the validation set, and 600 in the testing set. The **Kvasir-SEG** dataset contains 1000 pairs of polyp images and masks. We split the dataset into training and testing sets with a ratio of 0.2 with a random state of 41.

3.2. Implementation Details

The experiments are all done on PG500-216(V-100) with 32 GB memory. The training and validation set of ISIC 2016 and

Authorized licensed use limited to: Zhejiang University. Downloaded on October 12,2023 at 06:04:37 UTC from IEEE Xplore. Restrictions apply.

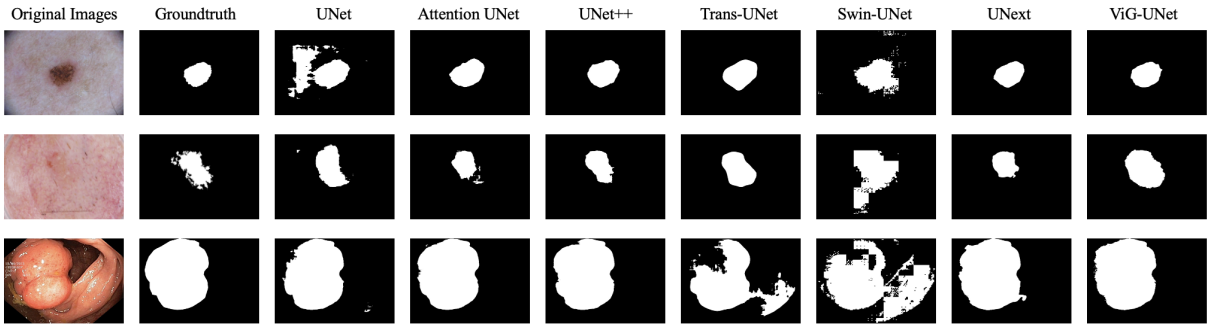

Fig. 3. The example segmentation experimental results of different methods on ISIC 2016, ISIC 2017 and Kvasir-SEG datasets

Kvair-SEG are split with a ratio of 0.2 with the random state 41. The total training epochs are 200 and the batch size is 4. The input images are all resized to 512×512. The optimizer used is ADAM [22]. The initial learning rate is 0.0001 and a CosineAnnealingLR [22] scheduler is used. The minimum learning rate is 0.00001. Only random rotation within 90 degree, flipping and normalization methods are used for augmentation. The evaluation metrics in validation are $mIOU$. A mixed loss combining binary cross entropy (BCE) loss and dice loss [23] is used in the training process:

$$\mathcal{L} = 0.5BCE(\hat{y}, y) + Dice(\hat{y}, y)$$

We implemented ViG-UNet and six other U-Net variants for comparison experiments. The pre-trained Swin-T model of 224×224 input size on ImageNet 1k is used for Swin-UNet, and the pre-trained model of ViT-B/16 on ImageNet 21k is used for Trans-UNet.

Table 2. Comparison Experimental Results on ISIC 2016, ISIC 2017 and Kvasir SEG (using the $mIoU$ metric)

Methods	ISIC 2016	ISIC 2017	Kvasir SEG
UNet [1]	0.8209	0.6410	0.6913
Attention-UNet [2]	0.8325	0.6473	0.6946
UNet++ [3]	0.8343	0.6504	0.6906
Trans-UNet [5]	0.8481	0.7147	0.4943
Swin-UNet [6]	0.7559	0.6676	0.3405
UNext [24]	0.8397	0.7156	0.6996
ViG-UNet	**0.8558**	**0.7211**	**0.7104**

3.3. Results

The performances of different methods are shown in Table 2 and Table 3 with metrics of $mIoU$ and $mDice$. The example segmentation results of different methods are displayed in Figure 3. From these experiments, we can see that on all three datasets, our approach performs best. And we can expect that if we use pre-trained models of ViG on ImageNet, the performance may be better. For the Swin-UNet, it's strange but [24] also reports its low performance on small datasets. In conclusion, our method shows competitive performance compared to classical and state-of-the-art techniques.

Table 3. Comparison Experimental Results on ISIC 2016, ISIC 2017 and Kvasir SEG (using the $mDice$ metric)

Methods	ISIC 2016	ISIC 2017	Kvasir SEG
UNet	0.8984	0.7708	0.8023
Attention-UNet	0.9058	0.7739	0.8065
UNet++	0.9070	0.7768	0.8033
Trans-UNet	0.9158	0.8244	0.6439
Swin-UNet	0.8568	0.7914	0.4974
UNext	0.9103	0.8241	0.8122
ViG-UNet	**0.9206**	**0.8292**	**0.8188**

Table 4. Comparison of Parameters of Different models

Methods	Parameters
UNet	7.8M
Attention-UNet	8.7M
UNet++	9.2M
Trans-UNet	92.3M
Swin-UNet	27.3M
UNext	1.5M
ViG-UNet	0.7G

We also calculate the number of parameters by using fvcore Python package with (1, 3, 512, 512) input. Admittedly, our model is larger than others and needs more computational resources.

4. CONCLUSION

In this work, we propose a ViG-UNet for 2D medical image segmentation, which has a GNN-based U-shaped architecture consisting of the encoder, the bottleneck, and the decoder with skip connections. Experiments are done on the ISIC 2016, the ISIC 2017 and the Kvasir-SEG dataset, whose results show that our method is effective.

Authorized licensed use limited to: Zhejiang University. Downloaded on October 12,2023 at 06:04:37 UTC from IEEE Xplore. Restrictions apply.

5. ACKNOWLEDGEMENTS

This work was supported by a Grant from The National Natural Science Foundation of China (No. U21A20484).

6. REFERENCES

[1] Olaf Ronneberger et al., "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[2] Ozan Oktay et al., "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[3] Zongwei Zhou et al., "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.

[4] Alexey Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] Jieneng Chen et al., "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[6] Hu Cao et al., "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021.

[7] Marco Gori et al., "A new model for learning in graph domains," in *Proceedings. 2005 IEEE international joint conference on neural networks*, 2005, vol. 2, pp. 729–734.

[8] Alessio Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.

[9] Mathias Niepert et al., "Learning convolutional neural networks for graphs," in *International conference on machine learning*. PMLR, 2016, pp. 2014–2023.

[10] Joan Bruna et al., "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[11] Mikael Henaff et al., "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[13] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[14] Kai Han et al., "Vision gnn: An image is worth graph of nodes," *arXiv preprint arXiv:2206.00272*, 2022.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[16] David Gutman et al., "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1605.01397*, 2016.

[17] Noel CF Codella et al., "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)," in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018, pp. 168–172.

[18] Debesh Jha et al., "Kvasir-seg: A segmented polyp dataset," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 451–462.

[19] Guohao Li et al., "Deepgcns: Can gcns go as deep as cnns?," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9267–9276.

[20] Dan Hendrycks and Kevin Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," 2016.

[21] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[22] Ilya Loshchilov and Frank Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[23] Fausto Milletari et al., "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[24] Jeya Maria Jose Valanarasu and Vishal M Patel, "Unext: Mlp-based rapid medical image segmentation network," *arXiv preprint arXiv:2203.04967*, 2022.

Authorized licensed use limited to: Zhejiang University. Downloaded on October 12,2023 at 06:04:37 UTC from IEEE Xplore. Restrictions apply.