View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Liverpool Repository

TransBridge: A lightweight transformer for left ventricle segmentation in echocardiography

Kaizhong Deng[1], Yanda Meng[2], Dongxu Gao[2], Joshua Bridge[2], Yaochun Shen[1], Gregory Lip[3], Yitian Zhao[4], and Yalin Zheng[2](✉)

[1] Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK

[2] Department of Eye and Vision Science, University of Liverpool, Liverpool, UK
yalin.zheng@liverpool.ac.uk

[3] Department of Cardiovascular & Metabolic Medicine, University of Liverpool, Liverpool, UK

[4] Cixi Institute of Biomedical Engineering, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo, China

Abstract. Echocardiography is an essential diagnostic method to assess cardiac functions. However, manually labelling the left ventricle region on echocardiography images is time-consuming and subject to observer bias. Therefore, it is vital to develop a high-performance and efficient automatic assessment tool. Inspired by the success of the transformer structure in vision tasks, we develop a lightweight model named 'TransBridge' for segmentation tasks. This hybrid framework combines a convolutional neural network (CNN) encoder-decoder structure and a transformer structure. The transformer layers bridge the CNN encoder and decoder to fuse the multi-level features extracted by the CNN encoder, to build global and inter-level dependencies. A new patch embedding layer has been implemented using the dense patch division method and shuffled group convolution to reduce the excessive parameter number in the embedding layer and the size of the token sequence. The model is evaluated on the EchoNet-Dynamic dataset for the left ventricle segmentation task. The experimental results show that the total number of parameters is reduced by 78.7% compared to CoTr [22] and the Dice coefficient reaches 91.4%, proving the structure's effectiveness.

Keywords: Echocardiography · Left ventricle segmentation · Lightweight Transformer model · Parameter efficiency

1 Introduction

Cardiovascular disease has one of the highest mortality and morbidity rates worldwide. Echocardiography imaging is essential for evaluating cardiac functions in clinical practice, such as left ventricular ejection fraction [16]. The left ventricular ejection fraction assessment is usually performed by comparing the left ventricular volume at end-systolic and end-diastolic frames. Manual annotation of the left ventricular region is a time-consuming and human-dependent step, resulting in high inter-observer variance and limited precision [8,11]. Hence, it is vital to develop an automatic segmentation algorithm of the left ventricle in echocardiographic images. Some machine learning methods have been proposed, such as Structured Random Forest [9] and dynamic appearance model [7]. However, they are either based on hand-crafted features or not sufficiently robust. Recent research interest moves to the deep learning methods that will avoid hand-crafted features and are robust enough. Several models using distinct network structures have shown promising performance [17,10,12], while [12] provides a comprehensive review of the recent methods. One of the limitations of these methods is the large model size that is not efficient to use.

Related Works The development of deep learning methods and approaches [19,14,2,15,3] has led to improvements in biomedical image segmentation tasks. For example, U-Net [19] uses encoder-decoder architecture with the skip-connection to extract features from multiple scales and recover them to the original scale. It has been shown that the U-Net reaches good accuracy on left ventricle segmentation [10]. The residual connection in ResNet [5] improves the accuracy of the CNN by constructing a clean identity mapping path to ease optimization [6], and ResUNet [21] employs this technique in the U-Net structure. DeepLabV3 [2] uses dilated convolutions to increase the receptive field so that the model can catch dependency at a longer distance. It has been shown that DeepLabV3 can reach a remarkable performance on the left ventricle segmentation task [17]. In a recent study, the transformer model is introduced to break through the limitation of locality from convolution operators to build the global dependency. The Vision Transformer [4] is a pure Transformer model in image recognition tasks with state-of-the-art performance. The transformer model combined with CNN structure has also shown great potential in the image segmentation task [24,1,22]. However, the drawback of introducing transformer structures is the significant increase in the number of parameters. Therefore, it is necessary to design a lightweight transformer model to utilize its high performance on vision tasks.For example, works on reducing parameter number in CNNs and transformers by applying shuffle algorithm have been proposed in [23,13]. The Sandwich parameter sharing the transformer encoder structure has also been discussed [18]. Therefore, building an efficient and training-friendly model should also be a crucial criterion of the deep learning model.

Our Contributions In our works, the patch embedding before the Transformer structures are re-designed using the shuffling layer and group convolutions to reduce the excessive parameter number and token numbers. Sandwich parameter sharing was used to minimize the transformer parameters [18]. We propose the TransBridge, a lightweight hybrid model using the transformer and the CNN structure for left ventricle segmentation in echocardiography.

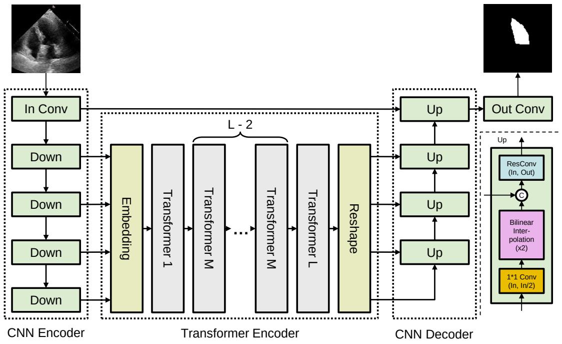


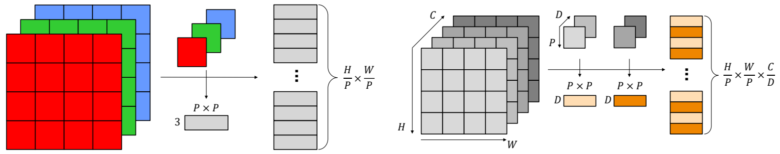
Fig. 1. TransBridge: Downsampling block in CNN Encoder and upsampling block in CNN decoder. The transformer bridges the CNN encoder and decoder to model inter-feature level dependency. The Sandwich parameter sharing mechanism allows parameters shared in all the middle layers except for the beginning and the end.

2 Methods

CNN Encoder The CNN encoder is used to extract features efficiently to obtain high abstract level features, saving time for the transformer encoder to focus its attention on low-level features. The CNN encoder adopts the U-Net encoder structure that cascades convolution layers and downsamples the resulting features between each block [19], shown in Fig. 1. The downsampling layer comprises a max-pooling operation to downsample the feature map size and a residual double convolution block. The residual block contains two BN-ReLU-Conv layers and a 1x1 Conv for identity mapping. In addition, the Pre-activation residual block can result in easier training [6]. Assuming the input image is of size of (H, W), the extracted feature maps can be expressed as $\{x\}_l \in \mathbb{R}^{(C\times l)\times\frac{H}{l}\times\frac{W}{l}}, 1 \leq l \leq L$. Considering the efficiency of the model, the feature maps from the first CNN encoder layer are not used for the transformer encoder layer but directly skip-connect to the CNN decoder layer at the same level to retain the low-level features and reduce the cost of attention.

Patch Division In the transformer-based vision task, such as ViT [4] and SeTr [24], the input of the transformer encoder layers is embedded patch sequence. In the embedding layer, shown in Fig. 2(a), the input image $x \in \mathbb{R}^{C\times H\times W}$ is equally divided into patches. Every patch is flattened to a 1-dimensional vector so that the patch sequence becomes $p \in \mathbb{R}^{N\times D_0}$, where the number of patches is represented as $N = \frac{H}{P} \times \frac{W}{P}$ and the vector size is represented as $D_0 = C \times P \times P$.

In order to embed the multi-channel feature maps, channels are split into several groups and treated independently. The patch is divided with a fixed size of (P, P), and D channels in each group are attributed to the patch, as shown in Fig. 2(b). Therefore, the sequence of a patch of feature maps is in the


$\frac{H}{P} \times \frac{W}{P}$
$P \times P$
3
C
D
P
H
W
$P \times P$
D
$P \times P$
D
$\frac{H}{P} \times \frac{W}{P} \times \frac{C}{D}$

(a) Patch division for RGB image (b) Patch division for multi-channel feature maps

Fig. 2. Patch division: The division method in TransBridge is designed for multi-channel feature maps as in Fig. 2(b). The total C channels are divided into several D channel groups. Then, channels in each group are treated independently. Finally, those D-channeled patches are flattened into vectors and concatenated.

form of a sequence of dense patches $p \in \mathbb{R}^{\frac{C}{D} \times \frac{H}{P} \times \frac{W}{P} \times D \times P \times P}$. After flattening the dense patches to vector, the feature map $x \in \mathbb{R}^{C \times H \times W}$ is transformed into a dense flattened patch sequence of $z_d \in \mathbb{R}^{M \times D_d}$, where the vector length is $D_d = D \times P \times P$, and the total number is $M = \frac{C}{D} \times \frac{H}{P} \times \frac{W}{P}$. As feature maps from the different levels have different channel sizes and spatial dimension, the number of token M is different in each level. However, the input size of the patch D_d is the same among all sequence so that the token sequence is $\{z_d\}_l \in \mathrm{R}^{M_l \times D_d}$and the l denotes the level of features.

Length Shortening The length of the token sequence is shortened before patch embedding. The token length is crucial because the complexity of the transformer encoder layer is sensitive to the sequence length. In this design, a shuffling layer and 1x1 group convolution are applied to shorten the length. First, in the shuffling layer, as shown in Fig. 3, all the four token sequences are divided into G groups individually through the channel dimension and all divided sequences from different feature levels are shuffled according to the group number to rearrange the group division so that each new group contains an element from each level. Next, sequences are concatenated through the channel dimension and conduct a 1x1 convolution in a group of G to compress the channel number to N to shorten the total sequence length.

Transformer Encoder Before feeding into the transformer encoder, patch and positional embedding are required to pre-process the patch sequence. A trainable linear layer projects the token vector from its length D_d to the hidden size D_h of the transformer encoder to obtain the patch embedding as shown in Eq.(1). Next, a trainable positional embedding layer is added to the patch embedding to retain the spatial information that the transformer encoder layer cannot model.

$$z_0 = \left[z_h^1 E; z_h^2 E; \ldots ; z_h^N E\right] + E_{pos}, E \in R^{D_d \times D_h}, E_{pos} \in R^{N \times D_h} \tag{1}$$

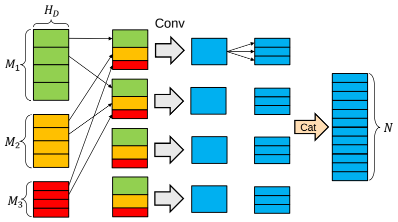


Fig. 3. Shuffling layer and group convolution: tokens from different feature channels are firstly split into groups and shuffled; A group convolution is applied to the grouped tokens to shorten the token sequence length from $M_1 + M_2 + M_3$ to N. All the tokens are concatenated together at the final stage. For the demonstration purpose, the level of features L is set to 3 and the group number G is 4.

Every single layer of the transformer encoder consists of Multihead Self-Attention (MSA) blocks and Multi-Layer Perceptron (MLP) blocks, shown in Eq.(2) and Eq.(3). A residual connection bypasses each block to form an identity mapping and a layer normalization operator is inserted in the front of each block. In addition, to increase the parameter efficiency, the parameter is shared in the Sandwich mode [18], which shares the parameters of all $L - 2$ middle layers, except the beginning and the ending layer in this L layer transformer encoder.

$$z'_l = MSA\left(LN\left(z_{l-1}\right)\right) + z_{l-1} \tag{2}$$

$$z_l = MLP\left(LN\left(z'_{l-1}\right)\right) + z'_{l-1} \tag{3}$$

The token sequence will be expanded and rearranged by reversing the length compression and patch division back to feature maps with the original dimension. During the rearranging, the shuffling process is not applied because the channel dimension has already been mixed.

CNN Decoder The CNN decoder absorbs feature maps from the transformer encoder and recovers them to the original size. For example, in the upsampling block, the feature maps from the previous decoder layer use 1x1 convolutions to match the channel numbers to half of the desired input channel number. Then its height and width are doubled by bilinear interpolation. Next, the resulted planes are concatenated with the feature maps from the transformer encoders to feed into a residual block to refine the feature maps. Finally, the output block will fuse the resulted planes into a one-dimensional segmentation map to output it as the final prediction.

3 Experiments

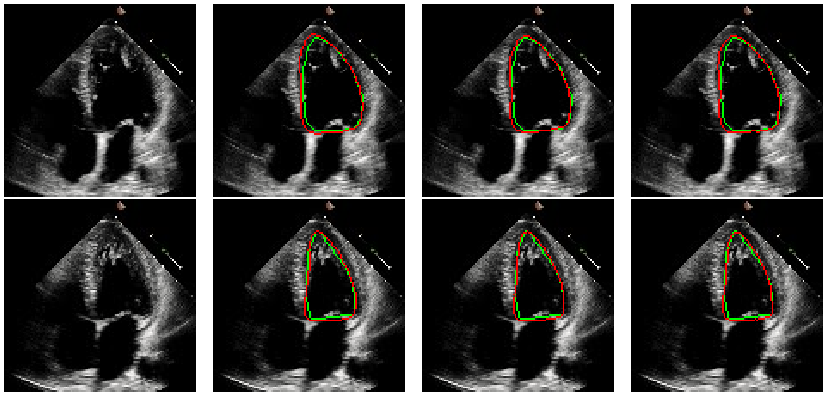
Dataset EchoNet-Dynamic dataset is a large public dataset with apical four-chamber two-dimensional echocardiographs [17]. For each video, an end-systole and an end-diastole frame were selected for the analysis. Expert sonographers and cardiologists annotate the left ventricle region during the standard clinical workflow. Among the 20,048 images, 14,920 images were used for training, 2,576 images for validating, and 2,552 images for testing. The end-systolic and end-diastolic frame of the same subject were placed in the same group. All the images were resized to 112*112 pixels and converted to grayscale. The training set was shuffled in each epoch to avoid any specific class distribution in each batch.

Implementation details The proposed model was implemented with two scales: Base ('TransBridge-B') and Large ('TransBridge-L'). To better compare with the TransBridge, the CoTr [22] was implemented with the original Vaswani Transformer instead of the Deformable Transformer and built in the base scale. The differences between the two scales of the TransBridge are CNN channel number, transformer hidden size, and transformer MLP intermediate layer size, shown in Table 1. The number of CNN feature levels fed to the Transformer, L, is set to 4. The patch size was set to (7, 7), and the grouping factor G was set to 8 so that there were at least two groups in each feature level for the shuffling. The transformer encoder layer has six layers and is split into four heads in the self-attention layer. The parameter number of TransBridge-B has been reduced by 78.7% compared with the CoTr model. Meanwhile, the number of parameters of the embedding layer has been reduced from 12.07M in CoTr to 0.17M in TransBridge, which is 1.4% of the normal embedding layer. The UNet and the ResUNet have also been implemented as references. The ResUNet has the CNN structure but without the transformer encoder layer in TransBridge.

Table 1. The configurations of the evaluated models

Method	Total	Embedding Layer	CNN Structure	Transformer Structure	
	Param	Param	L1 Channel Number	Hidden size	MLP size
CoTr	16.39M	12.07M	16	256	256
TransBridge-B	3.49M	0.17M	16	256	256
TransBridge-L	11.3M	0.23M	32	392	512
UNet	7.25M	-	32	-	-
ResUNet	7.6M	-	32	-	-

The model was trained on an Nvidia Tesla P100 GPU with a batch size of 8. The running GPU memory of our model can be limited to approximately 2GB. All the models were trained with an RMSprop optimizer with learning rate of 1e-4, momentum of 0.9, and a weight decay of 1e-8 for 15 epochs. Each epoch contains 20 steps, and each step has 93 iterations. The learning rate dropped to 10% of its original value if there is no further improvement in 10 steps. Binary cross-entropy loss is used to train the model and the Dice loss is used for validation.



(a) Original image (b) CoTr (c) TransBridge-B (d) TransBridge-L

Fig. 4. Segmentation Results Visualization: The ground truth is labeled with a green line, while the segmentation boundary from each model is in red.

4 Results

Comparison between models The performance of the TransBridge in two scales is compared with other methods on the left ventricle segmentation task. The segmentation are divided into two groups based on the heart contraction stage, either end-systolic or end-diastolic. Dice coefficient and Hausdorff distance are used to evaluate the segmentation quality.

Table 2. Comparing the segmentation results of: TransBridge-B (ours), TransBridge-L (ours), CoTr [22] are trained on the dataset. In addition, the results of the UNet and DeepLabV3 are cited from [10] and [17] respectively.

Method	Hausdorff distance					Dice (in %)				
	end-systolic		end-diastolic		Average	end-systolic		end-diastolic		Average
	Mean	Std	Mean	Std		Mean	Std	Mean	Std	
UNet [10]	-	-	-	-	7.3	-	-	-	-	89.6
DeepLabV3 [17]	-	-	-	-	-	90.3	-	92.7	-	91.5
UNet	6.506	5.977	6.017	4.405	6.262	82.50	0.078	87.63	0.054	85.07
ResUNet	4.175	5.403	3.725	5.403	3.950	91.17	0.048	93.51	0.034	92.34
CoTr	4.699	5.838	4.201	3.652	4.450	89.87	0.061	92.71	0.042	91.29
TransBridge-B	4.633	5.853	4.184	3.757	4.409	90.01	0.057	92.76	0.037	91.39
TransBridge-L	4.411	5.528	3.959	3.346	4.185	90.24	0.058	93.04	0.035	91.64

The testing results are shown in Table 2. Comparing the TransBridge-B and TransBridge-L with CoTr, improvements are made on the Dice coefficient (91.69% and 91.39% vs. 91.29%) and Hausdorff distance (4.185 and 4.409 vs. 4.450). In particular, TransBridge-B has only 21.3% parameters of CoTr, so it

Table 3. Ablation test with each structure configuration

The first layer skip connection	CNN-block	Sandwich sharing	Dice (in %)
No	Conv	No	90.7
No	ResConv	Yes	90.7
Yes	Conv	No	89.9
Yes	ResConv	No	90.2
Yes	ResConv	Yes	91.0

is more lightweight and efficient. Meanwhile, UNet, ResUNet, and DeepLabV3 have also been compared with the TransBridge models. In previous work [10], the UNet is evaluated on a small dataset with 1000 images. When training on this larger dataset, the large variance on features makes it difficult to perform as well as in the smaller dataset, and the further increment on its width cannot contribute to better accuracy. However, after introducing the Residual block, the accuracy of ResUNet has improved compared to UNet, exceeding the performance of DeepLabV3 and TransBridge. The reason for this might be that for this specific dataset, the image size is relatively small and the LV geometry is simple to segment, so there is no need for complicated models.

Ablation test In the ablation test, the model is trained until early converged, and it takes no more than five epochs for the validation loss to converge with tolerance less than 0.001. The results show that almost all the design changes can improve the overall performance, shown in Table 3. Sandwich sharing can make the most significant progress. Using the residual block instead of simple convolutions cannot make sufficient progress but it can avoid gradient vanishing. The skip connection of the first layer introduce low-level features, and its effect on the overall performance might depend on the presence of the other two features.

5 Discussion

The proposed TransBridge shows excellent potential for the left ventricle segmentation task. This lightweight design reduces the parameter by 78.7% while achieving a Dice score of 91.4%. In addition, the group and shuffling embedding can facilitate the information exchange in different feature levels and channels with fewer parameters. However, compared to the pure CNN structure, the transformer is not easy to train and attain competitive performance. It is sensitive to the dataset and hyperparameters, demanding extensive large-scale empirical trials to achieve the best performance [20]. Therefore, more sophisticated hyper-parameter tuning could further enhance the performance of the model.

6 Conclusion

This paper has proposed TransBridge, an efficient lightweight model that combines the CNN and transformer architecture for the LV segmentation task. The proposed shuffling layer and group convolution for patch embedding significantly reduces the total number of parameters by 78.7% and efficiently utilizes the transformer's power to cooperate with CNN. The model has been evaluated on the largest public echocardiography dataset, and the results confirm its effectiveness. In the future, the proposed model can be used as a powerful tool to support the management of cardiovascular diseases.

References

1. 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 (Feb 2021)
2. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017)
3. Chen, X., Williams, B.M., Vallabhaneni, S.R., Czanner, G., Williams, R., Zheng, Y.: Learning active contour models for medical image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11632–11640 (2019)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (Oct 2020)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 630–645. Springer International Publishing, Cham (2016)
7. Huang, X., Dione, D.P., Compas, C.B., Papademetris, X., Lin, B.A., Bregasi, A., Sinusas, A.J., Staib, L.H., Duncan, J.S.: Contour tracking in echocardiographic sequences via sparse representation and dictionary learning. Medical Image Analysis **18**(2), 253–271 (2014). https://doi.org/10.1016/j.media.2013.10.012, https://www.sciencedirect.com/science/article/pii/S1361841513001564
8. Lang, R.M., Badano, L.P., Mor-Avi, V., Afilalo, J., Armstrong, A., Ernande, L., Flachskampf, F.A., Foster, E., Goldstein, S.A., Kuznetsova, T., Lancellotti, P., Muraru, D., Picard, M.H., Rietzschel, E.R., Rudski, L., Spencer, K.T., Tsang, W., Voigt, J.U.: Recommendations for Cardiac Chamber Quantification by Echocardiography in Adults: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging. European Heart Journal - Cardiovascular Imaging **16**(3), 233–271 (02 2015). https://doi.org/10.1093/ehjci/jev014
9. Leclerc, S., Grenier, T., Espinosa, F., Bernard, O.: A fully automatic and multi-structural segmentation of the left ventricle and the myocardium on highly heterogeneous 2D echocardiographic data. In: 2017 IEEE International Ultrasonics Symposium (IUS). pp. 1–4 (2017). https://doi.org/10.1109/ULTSYM.2017.8092797
10. Leclerc, S., Smistad, E., Grenier, T., Lartizien, C., Ostvik, A., Espinosa, F., Jodoin, P.M., Lovstakken, L., Bernard, O.: Deep learning applied to multi-structure segmentation in 2D echocardiography: A preliminary investigation of the required

database size. In: 2018 IEEE International Ultrasonics Symposium (IUS). pp. 1–4 (2018). https://doi.org/10.1109/ULTSYM.2018.8580136

11. Leclerc, S., Smistad, E., Pedrosa, J., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., Berg, E.A.R., Jodoin, P.M., Grenier, T., Lartizien, C., D'hooge, J., Lovstakken, L., Bernard, O.: Deep learning for segmentation using an open large-scale dataset in 2D echocardiography. IEEE Transactions on Medical Imaging **38**(9), 2198–2210 (2019). https://doi.org/10.1109/TMI.2019.2900516
12. Li, M., Dong, S., Gao, Z., Feng, C., Xiong, H., Zheng, W., Ghista, D., Zhang, H., de Albuquerque, V.H.C.: Unified model for interpreting multi-view echocardiographic sequences without temporal information. Applied Soft Computing **88**, 106049 (2020). https://doi.org/10.1016/j.asoc.2019.106049, https://www.sciencedirect.com/science/article/pii/S1568494619308312
13. Mehta, S., Ghazvininejad, M., Iyer, S., Zettlemoyer, L., Hajishirzi, H.: Delight: Deep and light-weight transformer (Aug 2020)
14. Meng, Y., Meng, W., Gao, D., Zhao, Y., Yang, X., Huang, X., Zheng, Y.: Regression of instance boundary by aggregated CNN and GCN. In: European Conference on Computer Vision. pp. 190–207. Springer (2020)
15. Meng, Y., Wei, M., Gao, D., Zhao, Y., Yang, X., Huang, X., Zheng, Y.: Cnn-gcn aggregation enabled boundary regression for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 352–362. Springer (2020)
16. Oktay, O., Ferrante, E., Kamnitsas, K., Heinrich, M., Bai, W., Caballero, J., Cook, S.A., de Marvao, A., Dawes, T., O'Regan, D.P., Kainz, B., Glocker, B., Rueckert, D.: Anatomically constrained neural networks (ACNNs): Application to cardiac image enhancement and segmentation. IEEE Transactions on Medical Imaging **37**(2), 384–395 (2018). https://doi.org/10.1109/TMI.2017.2743464
17. Ouyang, D., He, B., Ghorbani, A., Yuan, N., Ebinger, J., Langlotz, C.P., Heidenreich, P.A., Harrington, R.A., Liang, D.H., Ashley, E.A., Zou, J.Y.: Video-based AI for beat-to-beat assessment of cardiac function. Nature **580**(7802), 252–256 (mar 2020). https://doi.org/10.1038/s41586-020-2145-8
18. Reid, M., Marrese-Taylor, E., Matsuo, Y.: Subformer: Exploring weight sharing for parameter efficiency in generative transformers (2021)
19. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015). pp. 234–241. Springer International Publishing, Cham (2015)
20. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers (2021)
21. Xiao, X., Lian, S., Luo, Z., Li, S.: Weighted Res-UNet for high-quality retina vessel segmentation (Oct 2018). https://doi.org/10.1109/ITME.2018.00080
22. Xie, Y., Zhang, J., Shen, C., Xia, Y.: CoTr: Efficiently bridging CNN and transformer for 3D medical image segmentation (Mar 2021)
23. Yang, Q.L.Z.Y.B.: SA-Net: Shuffle attention for deep convolutional neural networks (Jan 2021)
24. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6881–6890 (June 2021)