MEDICAL IMAGE SEGMENTATION WITH VISION TRANSFORMERS.

PROJECT REPORT

Tobias Höppe
thoppe@kth.se

Agnieszka Miszkurka
agnmis@kth.se

Kenza Bouzid
bouzid@kth.se

May 24, 2021

ABSTRACT

Rapid advances in the field of medical imaging are revolutionizing medicine. Disease diagnosis and treatment planning is becoming more and more accurate thanks to computer-aided diagnosis (CAD) in which Medical Image Segmentation plays a crucial role. Given the increased interest in self-attention mechanisms in computer vision and their ability to overcome convolution intrinsic locality, J. Chen et al. proposes TransUNet [2], the first medical image segmentation framework using Vision Transformers (ViT) [3] as a strong encoder in a U-shaped architecture U-Net [13]. TransUNet achieves state-of-the-art results compared to various architectures: V-Net [10], U-Net [13], AttnUnet [17], and DARR [4].

In this paper, we reproduce a subset of the work of [2], as a hybrid CNN-Transformer architecture able to leverage both detailed high-resolution spatial information from CNN features and the global context encoded by Transformers. We build 4 different architectures (ViT_None, ViT_CUP, R50-ViT_CUP and TransUnet). Each architecture represents a baseline for the following one. All experiments are conducted on synapse multi-organ segmentation dataset. We first attempted to reproduce the results of [2] with the exact same settings. Then, we fine tune the models by experimenting with different learning schedules and optimizers (cyclic learning [14], Adam [8], SGD [7]) and test the necessity of fine tuning the weights of the encoder. Finally, we can report a best dice score of **77,31%**.

Code and demo are available at : `https://github.com/KenzaB27/TransUnet`

1 Introduction

Medical image segmentation plays an essential role in computer-aided diagnosis (CAD) systems that help improve the sensitivity and specificity of lesion detection and diagnostic radiology research. With the development and increasing use of medical imaging modalities (X-ray, CT, MRI, Ultrasound, Microscopy, PET, Endoscopy, OCT, and many more) the tools for automating the information extraction from these images becomes as important as the modalities itself. With the improvement of hardware, Deep Learning methods became more feasible for these tasks and started to take over from pure mathematical and hand designed models in the 2000s. Nowadays, most of the frequently used methods are based on Deep Learning. The U-Net [13] is the basis for the most common architectures in medical image segmentation. It uses CNNs for feature extraction (encoding) and expanding (decoding). But in addition, it uses skip connections between the encoder and decoder as it improves localization accuracy. As U-Net, most of the Deep learning architectures are solely based on CNNs for encoding. The main disadvantage of CNNs is that they have limitations in modeling explicit long-range relations. Therefore, if images contain structural information with large variations in shape and texture per image, CNNs tend to perform poorly. To overcome this shortcoming, in [2] a Visions Transformer Network (ViT) [3] was proposed for encoding. Transformer Networks, introduced originally for sequence to sequence tasks by [15], have become the de-facto standard in NLP, but have recently been adapted for image classification [3] and semantic segmentation [18]. Transformers solely rely on attention mechanisms and are therefore powerful in modeling global context and show state of the art results when trained on large data-sets.

In this Project, we are going to reproduce the experiments done in [2] on the Synapse multi-organ segmentation dataset [1]. First we will reproduce the experiments with the same hyperparameters on all four models introduced in [2], namely ViT_None, ViT_CUP, R50-ViT_CUP and TransUNet (see Section 4). We will see that the naive use of a transformer alone does not give satisfactory results, but by using embedded features from a pre-trained Res-Net-50 [5] can help the Network preserve low-level visual cues. Also, we will conduct further experiments on TransUNet and R50-ViT_CUP by using different optimizer and cyclic learning [14] and show that it is not necessary to fine-tune the pre-trianed models used in the architectures to obtain state of the art results when the decoder is powerful enough.

2 Related work

In [13] a new U-Net Neural Network architecture for biomedical image-segmentation was proposed, which consisted of encoder and decoder, with additional skip connections between these parts to maintain local information. This enabled the Network to detect what and where objects are located in the image. Several extensions where made to this architecture, such as using attention gate (AG) [11]. The AGs help the Network to learn to focus on target structures of varying shapes and sizes and suppress irrelevant regions in an input image while highlighting salient features useful for a specific task. Also, to achieve better encoding, pre-trained Networks can be used as encoders. A common choice is the ResNetv2 [5] and using its intermediate results for the skip connections. Recently, in [2], in addition a pre-trained Vision-Transformer Network [3] was attached to the output of the ResNet as a further encoder. This method was able to achieve state of the art results on several data-sets. To work directly on 3D images, the V-Net was introduced by [10].

Transformer Networks were first introduced by [15] as a sequence transduction model which is solely based on attention mechanisms. The computational costs of this Network is lower when having large data-sets, as computations can be parallelized, but it does not scale well on the length of input sequence and was therefore not feasible to train on images. But in [3], images were divided into patches that were fed to the network which made the computational cost feasible and yielded in new state of the art results for image classification.

3 Data

Our experiments were conducted on Synapse multi-organ segmentation dataset. We use 30 abdominal CT (computed tomography) scans with 3779 axial contrast-enhanced abdominal clinical CT images in total. Voxel spatial resolution of each volume is ([0.54 ~0.54] × [0.98 ~0.98] × [2.5 ~5.0])mm^3. Each scan delineates multiple organs: aorta, gallbladder, left kidney, right kidney, liver, pancreas, spleen and stomach. Those organs together with none label constitute our nine target classes. We obtained preprocessed dataset from the authors of [2].

The training data is in npz (numpy) format and contains 2D images and labels both of size 512×512. There are 2211 slices (2D images). We converted each image to have 3 channels by repeating it 3 times. Next, both images and labels sizes were reduced to 224×224 using spline interpolation of order 3. Data is augmented with either random horizontal or vertical flip, rotation by 90 degrees or rotation from -20 to 20 degrees. Labels are cast to integer and one hot encoded.

[1] `https://www.synapse.org/#!Synapse:syn3193805/wiki/217789`

The test dataset is in h5 format, i.e. one sample is a 3D image, since evaluation metrics should be computed for each volume separately. It contains 12 black and white volumes and corresponding labels. It was also transformed to the same shape as training data.

To speed up the training process, we converted the data from numpy format to TFRecords. It is TensorFlow's binary storage format which takes up less space. TensorFlow is optimised to process them efficiently. With this improvement, training was around 4-5 times quicker.

4 Methods

The goal of our project was to reproduce results of [2] by predicting pixel-wise labelmap with size $H \times W$ for a given image (CT scan) of size $H \times W \times C$. Following the methods outlined in the paper, we implemented four model architectures which we will describe in this section: ViT_None, ViT_CUP, R50-ViT_CUP and TransUNet. In this order, each architecture acts as a baseline for the following one. Also, we briefly introduce the idea of the Vision Transformer Network, which builds the backbone for the encoder. The entire model divided into its individual parts can be seen in Figure 1.

4.1 Vision Transformer

In order to make use of Vision Transformer, we have to sequentialize the images before, unlike when using only CNNs. This is done by dividing each image into patches of size $P \times P$ and then flatten them. Therefore we will get an input sequence $x_1, ..., x_N$ with size $x_i \in \mathbb{R}^{P^2C}$ and $N = \frac{HW}{P^2}$. Before feeding this sequence into the Multihead Self-Attention (MSA) layers, we map the flattened image patches into a latent D-dimensional embedding space by a trainable linear projection $E \in \mathbb{R}^{(P^2C)D}$ (this is done by a single convolution layer with linear activation function). Also, we do add trainable positional embeddings $E_{pos} \in \mathbb{R}^{ND}$ to obtain encoded spatial information. With the matrix $X = [x_1, ..., x_N]$, we get

$$Z_0 = XE + E_{pos}. \tag{1}$$

These embedding are now fed into the MSA layers in which three vectors are going to be extracted from each input patch (namely the Query (q), Key (k) and Value (v)). With these Vectors, a score (z_l) is computed after each layer l for each patch by using the Query and Key of the other patches in the sequence. In matrix multiplication we have

$$Z_l = \sigma\left(\frac{QK^T}{\sqrt{d_k}}V\right), \tag{2}$$

where d_k is the dimensionality and σ the softmax function. Finally, Z_l is fed into a MLP. In this architecture we also use layer normalization [1] (denoted as LN) and residual connections. Therefore the final output of an MSA layer is computed as follows:

$$Z_l^{(1)} = MSA(LN(Z_{l-1})) + Z_{l-1}, \tag{3}$$

$$Z_l = MLP(LN(Z_l^{(1)})) + Z_l^{(1)}. \tag{4}$$

The architecture can be seen in Figure 1a. In our model, 12 MSA layers are used and each layer consists of 12 self-attention heads.

4.2 Upsampling

After receiving the encodings from the transformer $Z_l \in \mathbb{R}^{\frac{HW}{P^2} \times D}$, we need to upsample them again to the full resolution mask $Y \in \mathbb{R}^{H \times W \times K}$ (where K denotes the number of classes). To recover the spatial order, the size of the encoded feature should first be reshaped from $\mathbb{R}^{\frac{H \times W}{P^2} \times D}$ to $\mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D}$. Now, there are two methods for upsampling.

Bilinear Upsampling. Here we simply use a convolution layer with a 1×1 kernel to reduce the channel size from D to the number of classes K. Then, the encoded image $Z \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times K}$ is upsampled bilinear to the full mask size $Y \in \mathbb{R}^{H \times W \times K}$. The combination of the Vision Transformer as encoder and bilinear upsampling is called **ViT_None**.

Cascaded Upsampler (CUP). The CUP upsampling consist of several upsampling steps using convolution layers. After reshaping, the encoded image is upsampled in each block by a 2× bilinear upsampling and a 3×3 convolution layer with a ReLU activation function to reduce the channel size. In total we use three of these blocks including the final segmentation layer (the 1×1 convolution from bilinear upsampling) to obtain the final mask. The combination of the Vision Transformer as encoder and Cascaded upsampling is called **ViT_CUP**.

4.3 Hybrid CNN-Transformer as Encoder

When using only a Vision Transformer as encoder, the results are decent but cannot reach state of the art. Therefore, in [2], a ResNet50v2 is proposed as an additional encoder. The image is fed into the ResNet50v2 and the embedded output $\hat{Z}_0 \in \mathbb{R}^{P \times P \times \hat{D}}$ will be flattened to $\hat{Z}_0 \in \mathbb{R}^{P^2 \hat{D}}$ and fed into the embedding layer described in Section 4.1 (note, that usually $\hat{D} \neq D$). The ResNet50v2 used here differs from the original ResNet50v2 as Group normalization [16] is used instead of Batch normalization [6] and weight standardization [12] for each convolution layer. When using ResNet50v2 as an additional encoder and CUP for decoding, we refer to this model as **R50-ViT_CUP**. The intermediate embeddings obtained with the ResNet50v2 can be used for skip connections to the decoder to increase localization. Finally, we obtain the **TransUnet** architecture by using additional skip connections from the ResNet50v2 that we concatenate to the inputs of each CUP block.

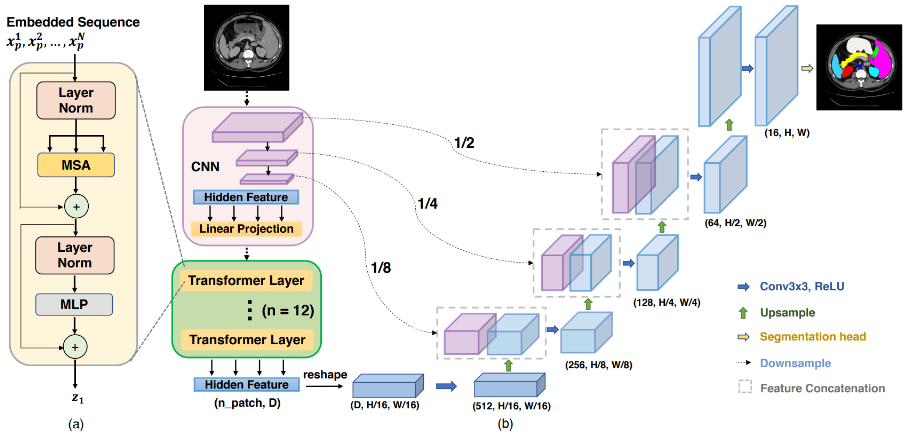


Figure 1: The architecture of the TransUNet with all its individual parts. When removing the skip connections, Res-Net50, CUP we obtain R50-ViT_CUP, ViT_CUP and ViT_None respectively (image taken from [2])

4.4 Loss Function and Evaluation Metrics

Loss Function The loss function is the weighted sum of categorical crossentropy and dice score denoted as CCE-DICE. As suggested in the original code, we set $\lambda = \mu = 0.5$

$$\begin{aligned} L_{cce-dice} &= \lambda L_{dice} + \mu L_{cce} \\ &= \lambda(1 - DSC) + \mu(-\sum_{i=0}^{9} y_i log\hat{y}_i), \end{aligned} \tag{5}$$

where DSC is the Dice Score or Sørensen–Dice coefficient: a common metric for pixel segmentation defined as 2 × the area of Overlap divided by the total number of pixels in both images:

$$DSC = \frac{2 \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid}. \tag{6}$$

5 Experiments

In this section we will introduce our conducted experiments. First, we will present the performance with the original hyperparameters on all four models as chosen in [2]. In addition, we experimented with the learning rate and the optimizer to increase performance. Throughout experiments, the "ViT" in the model name was replaced with "B16" to denote that "Base" version with 16x16 patch size of Vision Transformer is used (as described in [3]). Training was performed on google colab platform using NVIDIA V100 Tensor Core or Tesla P100 GPUs. The average training time was about 1 hour 45 min for all architectures thanks to the use of TfRecords that speeds up all I/O operations.

5.1 TransUNet paper parameters

First, we wanted to reproduce results of the main experiment in the TransUNet paper i.e. segmentation of 8 organs in Synapse mutliorgan dataset on 224x224 images. Since the codebase for the paper is publicly available, we were able to check which parameters authors used and get as close as possible to the original implementation. We used polynomial decay learning with decay at every update step, from 0.01 to 1e-6. As an optimiser, SGD [7] with momentum 0.9 was utilized. We used L2 regularisation for all layers.

Our implementation achieved the following results: 1% more, 2% less, 4% less and 3% less in the average dice score than in TransUNet paper for B16_None, B16_CUP, R50-B16_CUP and TransUNet respectively.

We tried to set all parameters to be the same as in original paper. However, there may be some differences which stem from differences in TensorFlow and PyTorch which caused the difference in our results. As an example, PyTorch SGD optimizer takes L2 regularisation parameter. In TensorFlow addons there is SGDW optimizer with this parameter but it uses Decoupled Weight Decay Regularization [9] which is different from L2 regularisation. Therefore, we added L2 loss to all the layers separately.

Model	**Average**	Aorta	Gallbladder	Kidney(L)	Kidney(R)	Liver	Pancreas	Spleen	Stomach
B16_None	**62.76**	47.33	38.69	70.35	68.38	89.91	41.44	79.70	66.30
B16_CUP	**65.98**	65.93	32.45	77.34	70.97	91.27	38.37	82.35	69.13
R50-B16_CUP	**67.87**	68.71	43.04	76.30	73.85	91.60	38.92	80.76	69.81
TransUNet	**74.76**	85.31	47.16	84.52	80.24	93.56	51.65	84.80	70.85

Table 1: Results for model trained with original parameters.

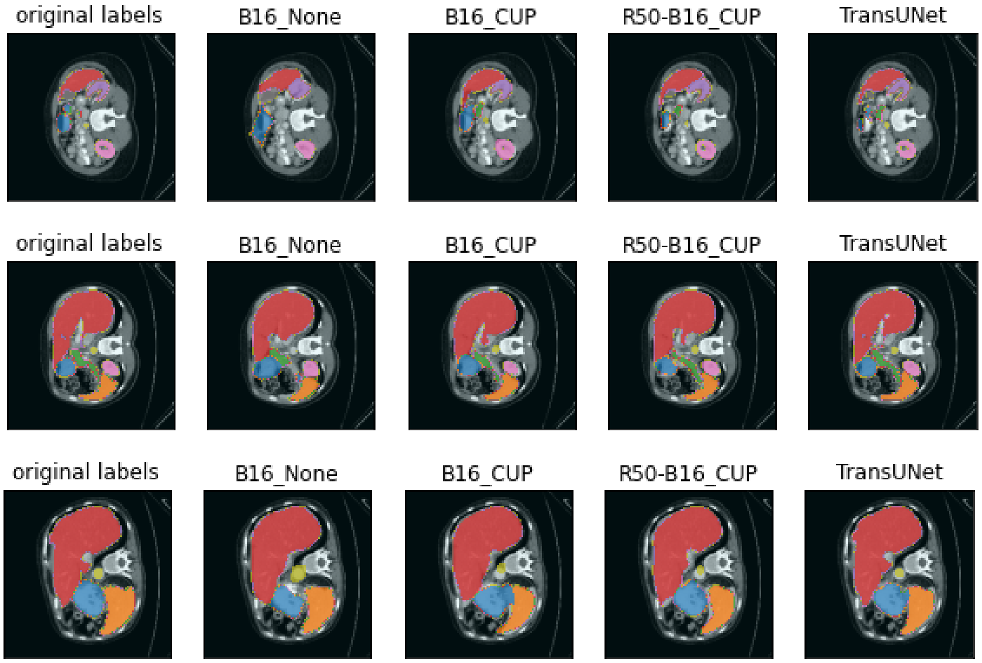


Figure 2: Segmentation results for models with original TransUNet parameters

5.2 Adam optimizer

In the experiments above, we used SGD with a polynomial decaying learning rate. The Adam optimizer [8] does adapt the learning rates individually for different parameters from estimates of first and second moments of the gradients. This often leads to better performance in Neural Networks.

We can see in Table 2 that changing the optimizer to Adam increased the performance of R50-B16_CUP by around 2%. However, results of TransUNet trained with Adam are almost the same as for SGD optimizer. In Figure 4 in the Appendix, we can see how the loss developed over the training process for networks with SGD and Adam optimizers. Validation loss for Adam is more unstable however the gap between validation and training loss is smaller than for SGD.

Model	**Average**	Aorta	Gallbladder	Kidney(L)	Kidney(R)	Liver	Pancreas	Spleen	Stomach
R50-B16_CUP	**68.32**	73.79	43.77	78.51	74.28	91.67	32.77	83.90	67.84
TransUNet	**74.47**	84.65	50.42	83.72	78.84	93.88	49.78	85.74	68.74

Table 2: Results for model trained using Adam optimiser.

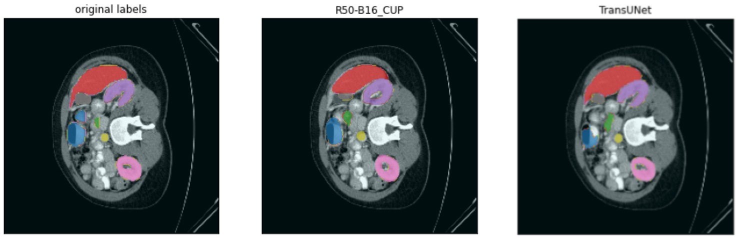


Figure 3: Segmentation results for R50-B16_CUP and TransUNet trained with Adam optimiser

5.3 Cyclical learning rate

In our experiments with the original parameters, we often observed fast convergence and just very little decrease in loss after the first 50 epochs. Therefore, we decided to use a cyclic learning rate [14] on R50-B16_CUP and TransUNet to explore several minimas in the loss function during training. We decided to have 25 cycles. We set minimum learning rate for 1e-5 and maximum to 1e-2. We used triangular cyclic learning rate.

As we can see in Table 3, this leads to a slight performance increase compared to the original setting. However, the R50-B16_CUP model trained with Adam still outperforms the one trained with a cyclic learning rate. Loss plot for R50-B16_CUP shown in Figure 4. We can see increase of the loss (when learning rate is maximal) followed by decrease (when learning rate is being reduced) especially during the first 50 epochs of training. Visualisation of models predictions can be found in the Appendix (Figure 6). Most evident improvement compared to original model can be observed for the stomach in the first CT scan (blue class) which was almost non-existent in case of polynomial learning rate decay.

Model	**Average**	Aorta	Gallbladder	Kidney (L)	Kidney (R)	Liver	Pancreas	Spleen	Stomach
R50-B16_CUP	**67.47**	69.06	41.83	76.63	74.08	91.11	39.34	82.61	65.08
TransUNet	**76.37**	85.26	51.01	86.64	81.47	93.76	52.77	87.77	72.28

Table 3: Results for model trained using cyclical learning rate

5.4 Without fine-tuning pretrained models

TransUNet's encoder consitsts of ResNet50v2 and ViT B16 pretrained on ImageNet dataset. Since samples in ImageNet dataset are substantially different from medical data, encoder layers were fine-tuned during training process. We performed ablation study, in which instead of fine tuning encoder layers, we froze them and only trained CUP. Surprisingly, the average dice score achieved by this network was higher than for the fine-tuned model (Table 4). We can observe 1-3% difference in dice score metric for most of the organs but over 12% improvement for gallbladder. Example segmentation results on test set can be found in Appendix (Figure 7). Both models seem to perform similarly well, with original model omitting some parts of the stomach (blue) in the first row and non-fined-tuned model in the third.

TransUNet	**Average**	Aorta	Gallbladder	Kidney (L)	Kidney (R)	Liver	Pancreas	Spleen	Stomach
Original	**74.76**	85.31	47.16	84.52	80.24	93.56	51.65	84.80	70.85
Frozen R50 and ViT	**77.31**	84.72	59.29	85.81	83.59	93.65	50.46	87.40	73.59

Table 4: Results for original TransUNet and TransUNet with Resnet50v2 and ViT B16 frozen.

6 Conclusion

Overall, we can see that the use of Vision Transformers for encoding can achieve state of the art results in biomedical image segmentation. The very strong global modeling from Vision Transformers can increase performance in segmentation, but lacks detailed information in the embedding. This is why it is necessary to use additional embeddings from CNNs to get performances of over 70%. However, the use of ResNet alone as additional encoder did only bring little increase in performance, but using skip connections from the intermediate embeddings of the ResNet achieved the final state of the art results. The most interesting finding is that the TransUNet without training the encoder performed best. However, we could not observe that behaviour for the other models, as the decoding might be not powerful enough to extract all necessary information from the embeddings of not trained models if skip connections are not used. Therefore the role of fine tuning the encoder is not entirely clear. Hence, we would suggest running experiments on additional datasets and using a more powerful decoder, as we could see that modifications on the decoder brought high increase in performance.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

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

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[7] J. Kiefer and J. Wolfowitz. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3):462 – 466, 1952.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[10] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation, 2016.

[11] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas, 2018.

[12] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Micro-batch training with batch-channel normalization and weight standardization, 2020.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[14] Leslie N. Smith. Cyclical learning rates for training neural networks, 2017.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

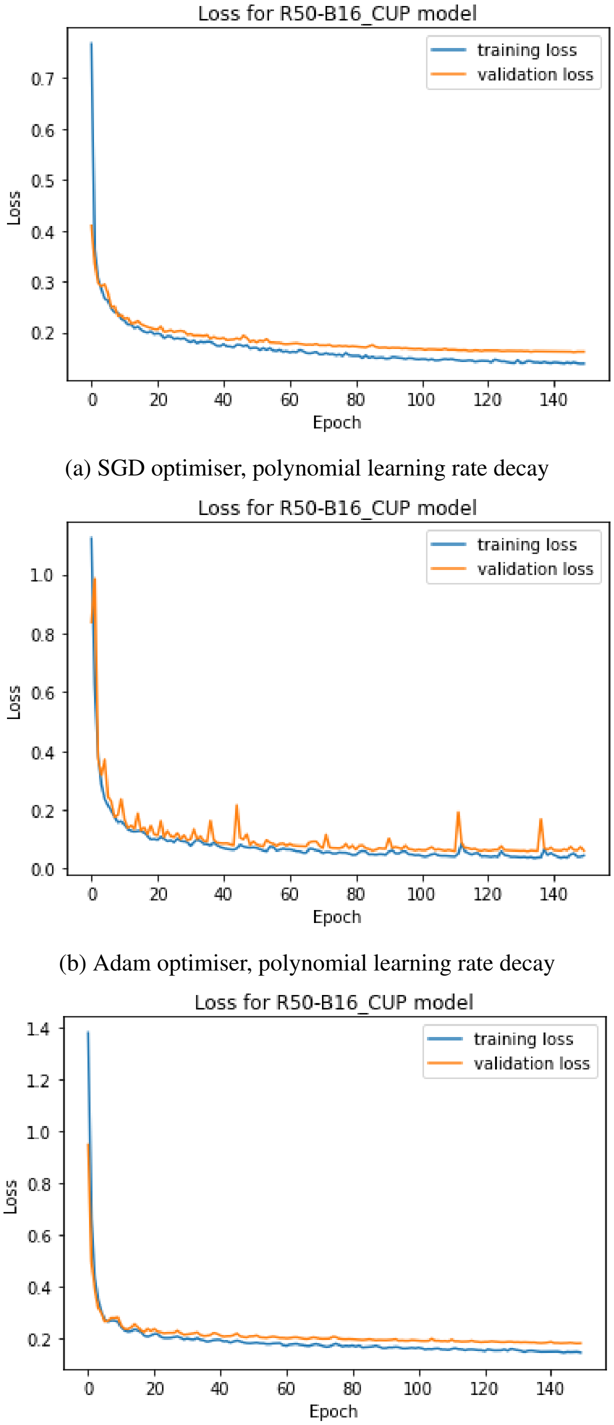
[16] Yuxin Wu and Kaiming He. Group normalization, 2018.

[17] Mou-Cheng Xu, Neil P. Oxtoby, Daniel C. Alexander, and Joseph Jacob. Learning to pay attention to mistakes, 2020.

[18] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *CoRR*, abs/2012.15840, 2020.

Appendices

A Loss plots for R50-B15_CUP



(a) SGD optimiser, polynomial learning rate decay

(b) Adam optimiser, polynomial learning rate decay

(c) SGD optimiser, cyclical learning rate

Figure 4: Loss plots

B Segmentation results for ablation studies

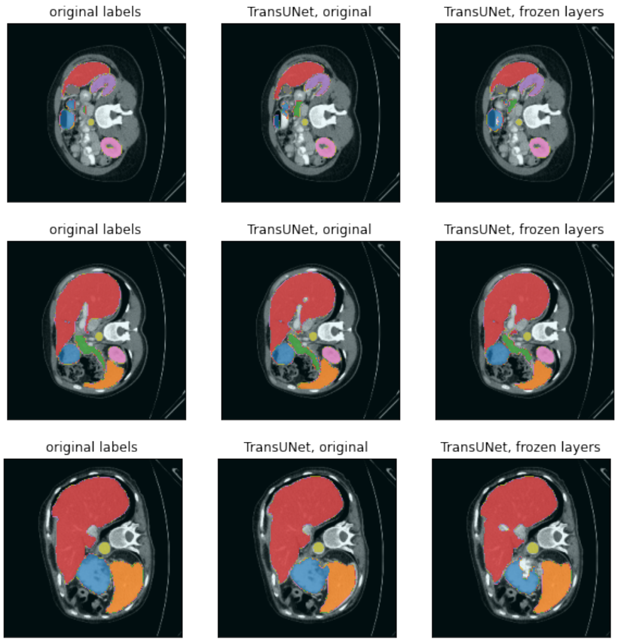


Figure 5: Segmentation results for original TransUNet and TransUNet with Resnet50v2 and ViT B16 frozen.

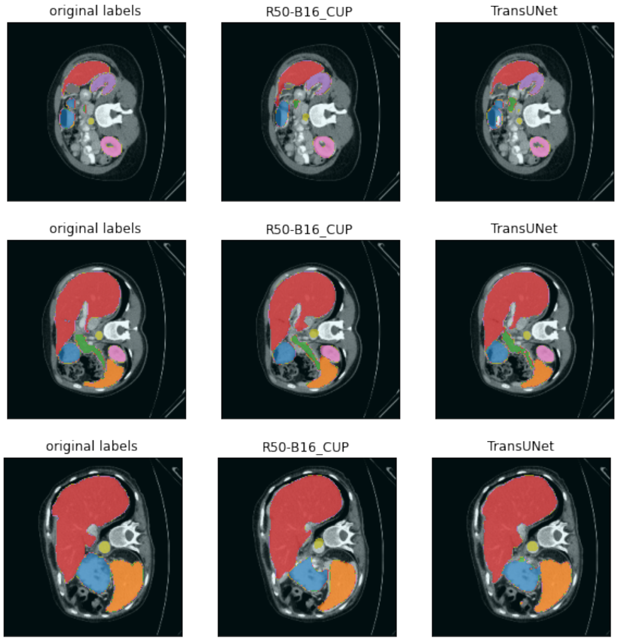


Figure 6: Segmentation results for R50-B16_CUP and TransUNet trained with cyclical learning rate

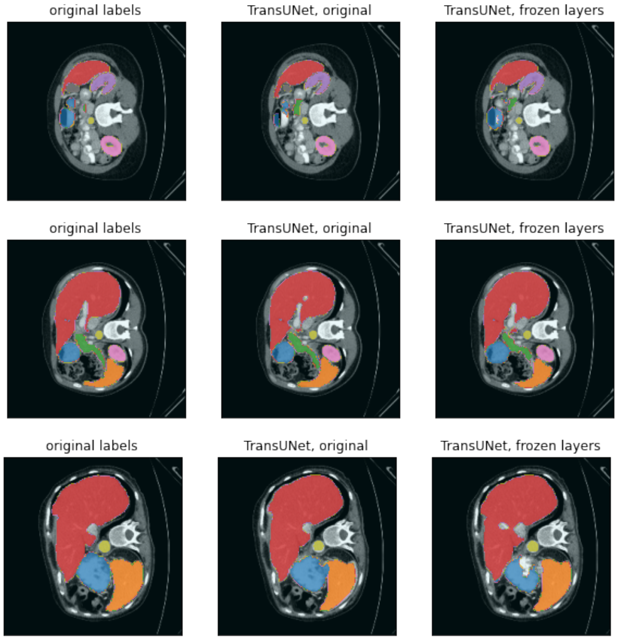


Figure 7: Segmentation results for original TransUNet and TransUNet with Resnet50v2 and ViT B16 frozen.