Organ segmentation using U-Net like models

Hidy Gábor

Supervisor: dr. András Lukács

2023. 5. 27.

1 Introduction

The subject of this sememester's project was medical image segmentation of abdonimal CT scan images. Here I will present the topics that I researched relating to this area, along with some experimental results.

2 Segmentation

2.1 Task

Consider an image as an $\mathbf{x} \in [0,1]^{L \times H \times W}$ tensor.[1] The task of segmentation is to find a *segmentation mask* $\mathbf{y} \in \{0,1\}^{C \times H \times W}$. In multiclass segmentation, $\mathbf{y}$ is usually restricted such that for all (i,j), exactly one of $y_{1ij} \dots y_{Cij}$ is 1. In practice, the output of a segmentation model will be a $\hat{\mathbf{y}} \in [0,1]^{C \times H \times W}$, where for each h and w, $\hat{y}_{0,h,w} + \ldots + \hat{y}_{C-1,h,w} = 1$. The final prediction then is obtained by taking the index c where $\hat{y}_{c,h,w}$ is maximal.

2.2 Metrics

In multiclass classification problems, measuring the accuracy (i. e. the percentage of accurately predicted datapoints) is frequently used, but it can be misleading in

[1] Here L denotes the channel size (eg. 3 for RGB images), following PyTorch's example and using the channel-first represenation.

medical image segmentation problems, since most pixels will belong to the background class, therefore the trivial model that always predicts the background will have high accuracy.

In binary segmentation problems, there are several well known metrics designed to provide an accurate measurement for imbalanced data, such as the Dice index, Jaccard index, or the average precision score (the area under the precision–recall curve), or HD95 (average modified – 95th percentile – Hausdorff distance). For multiclass segmentation, a mean value for these is frequently used. For example, the mean Dice index is obtained as follows: except for the background, we calculate the Dice index value for all classes in a "one versus all" manner, i. e. treating the given class as positive, and all other classes as negative, and calculating the binary Dice index. Then we take the average of the obtained $C - 1$ values.

3 U-Net

Most recent models for medical image segmentation use a U-Net [1] based architecture.

The base U-Net architecture – seen in Figure 1 – consists of an encoder (or "down") and a decoder (or "up") part. Both the encoder and decoder have several levels. Within one level, the spatial size of the tensor does not change. As is custom with other convolutional models, between levels all spatial dimensions are halved, and the channel size is doubled.

The main feature that distinguishes U-Net based networks from older fully connected segmentation nets is the presence of "lateral" residual connections – represented by red arrows in Figure 1. In the case of the base U-Net architecture, these concatenate the output of a "down" level to the input of the corresponding "up" level.

4 U-Net variants

Since the appearance of the original U-Net, there have been a multitude of segmentation models with similar architectures. Changes range from added skip connections

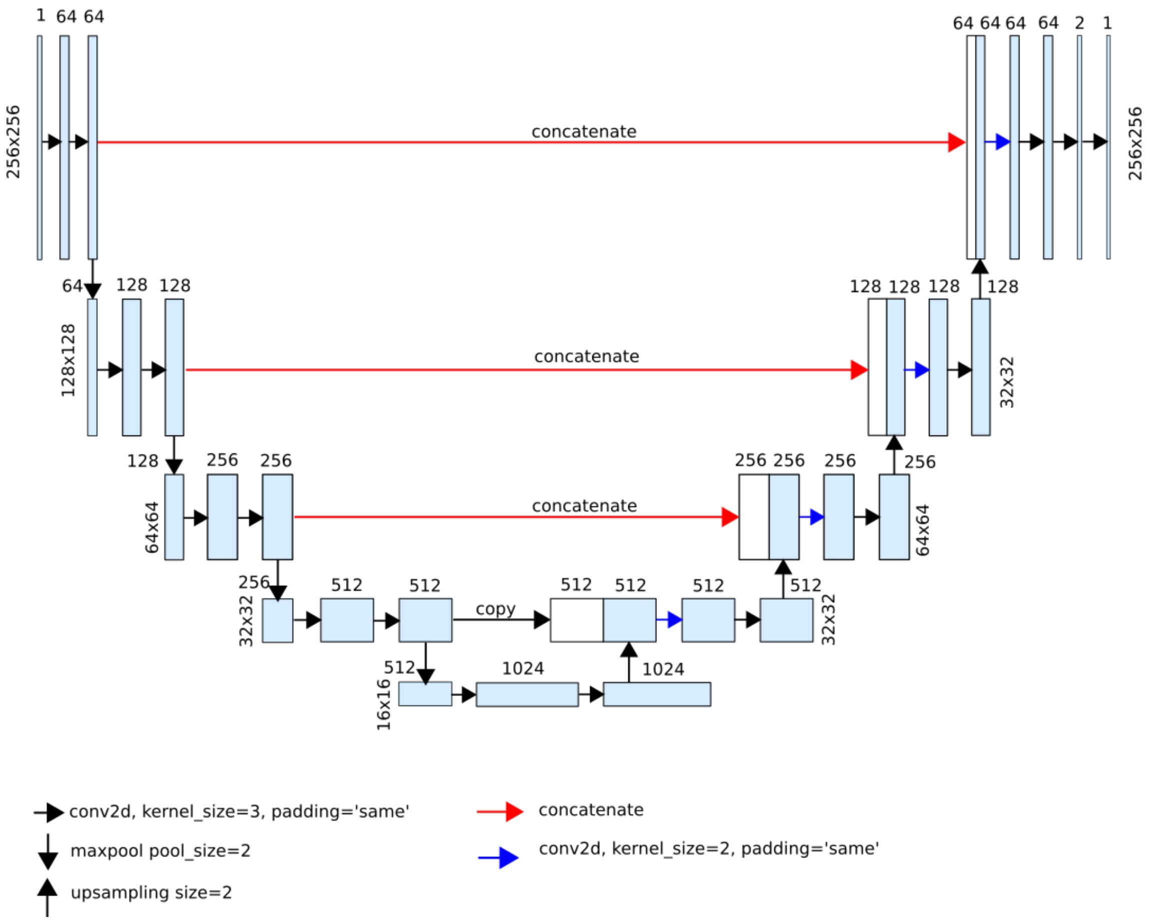

Figure 1: Base U-Net architecture

within levels, to wider or deeper architectures, to introducing attention mechanisms to the lateral residual connections, to fully transformer-based models.

For easier discussion, we have chosen a list of attributes with which any U-Net based segmentation model can be described.

We describe the models in the following way: it has a certain number of levels, described by its *depth*. On every level, there is a number of *basic blocks*, described by its *width*. The basic block is usually two or three layers, but theoretically it could be arbitrarily large. The input and output of a basic block might be connected by a *skip connection*. Between levels, there is a *downsampling* module in the down path, and an *upsampling* module in the up path. Between corresponding levels of the down and up paths, a “lateral” *residual connection* carries the information.

Before the first basic block, there might be an additional *stem* that performs preprocessing. After the final basic block, a *postprocessing* module will transform the output into the desired shape.

Other variants

Some U-Net variants cannot be described by configuring the above parameters. The most commonly used among these are U-Nets with other convolutional networks – e. g. a ResNet50 [2] – as encoders. Unlike the base U-Net encoder, ResNet50 does not have a constant width across levels. Its bottleneck blocks allow it to have a higher channel number than a U-Net encoder with the same parameter count.

Moreover, it only has four levels, while the base U-Net encoder has five. To deal with this, the decoders U-Nets with ResNet50 encoders have one more levels than their encoder, and their last level does not get a "lateral" residual signal, only the output of the previous level.

5 Data

The data used for testing was the Synapse multi-organ CT dataset, following [3]. The original dataset consists of 3D CT scan images, complete with segmentation masks for 13 different organs. As preprocessing, we only keep the masks for eight organs – aorta, gallbladder, left kidney, right kidney, liver, pancreas, spleen, stomach –, and we split the original samples into 2D slices. Out of sixteen samples, four were used for validation, and twelve for training. This resulted in approximately 1800 training records, and 400 validation records.

6 Experiments

6.1 Implementation details

Models were trained on 112×112 images, using a batch size of 24. The training ran for 150 epochs. Random rotations and flips were used on the training images. An SGD optimizer was used, with momentum 0.9 and weight decay 10^{-4}.

The loss was a Dice index based loss, calculated as

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{1}{C}\sum_{c=0}^{C-1} \mathcal{D}(\mathbf{y}_c, \hat{\mathbf{y}}_c), \quad \mathcal{D}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2\mathbf{y}\hat{\mathbf{y}} + \varepsilon}{\mathbf{y}\mathbf{y} + \hat{\mathbf{y}}\hat{\mathbf{y}} + \varepsilon},$$

where $\mathcal{L}$ is the Dice loss, $\mathbf{y}_c$ and $\hat{\mathbf{y}}_c$ is the c^{th} channel of the mask and prediction respectively, $\mathcal{D}$ is the soft binary Dice index, and ε is a smoothing term (10^{-5} in practice).

All models were variants of the original U-Net, as seen if Figure 1. A stem was added that was a 3×3 convolution, transforming the images from 3 to 64 channels. A softmax layer was applied to the output to obtain classification probabilites.

6.2 Instability

Experiments on the Synapse dataset showed high instability. Randomly initialised models converged to some constant output most of the time, and only on select occasions – about 1 out of 5 runs with a basic U-Net, and worse odds for other architectures – did it start producing sensible results.

6.3 Results

Weight initialisation

weight init	acc ↑	AUC ↑	AP ↑	DSC ↑	IoU ↑	HD95 ↓
He (fan out)	0.990	0.969	0.785	0.735	0.619	0.003
He (fan in)	0.990	0.979	0.776	0.731	0.612	0.003
Glorot	0.990	0.976	0.762	0.718	0.601	0.003
orthogonal	0.991	0.976	0.789	0.737	0.614	0.003
pretrained encoder	0.990	0.983	0.817	0.750	0.626	0.004

Table 1: Comparison of different weight initialisation techniques

To try and combat the instability issue, different weight initialisation techniques were tried. Table 1 shows the results. As seen, the Kaiming He initialisation scheme in its "fan out" mode is the best random initialisation technique. In most metrics, it is surpassed if instead of randomly initialised weights, we use an encode pretrained on Imagenet. (Using a pretrained encoder also helped with the instability issues.)

Pretraining length

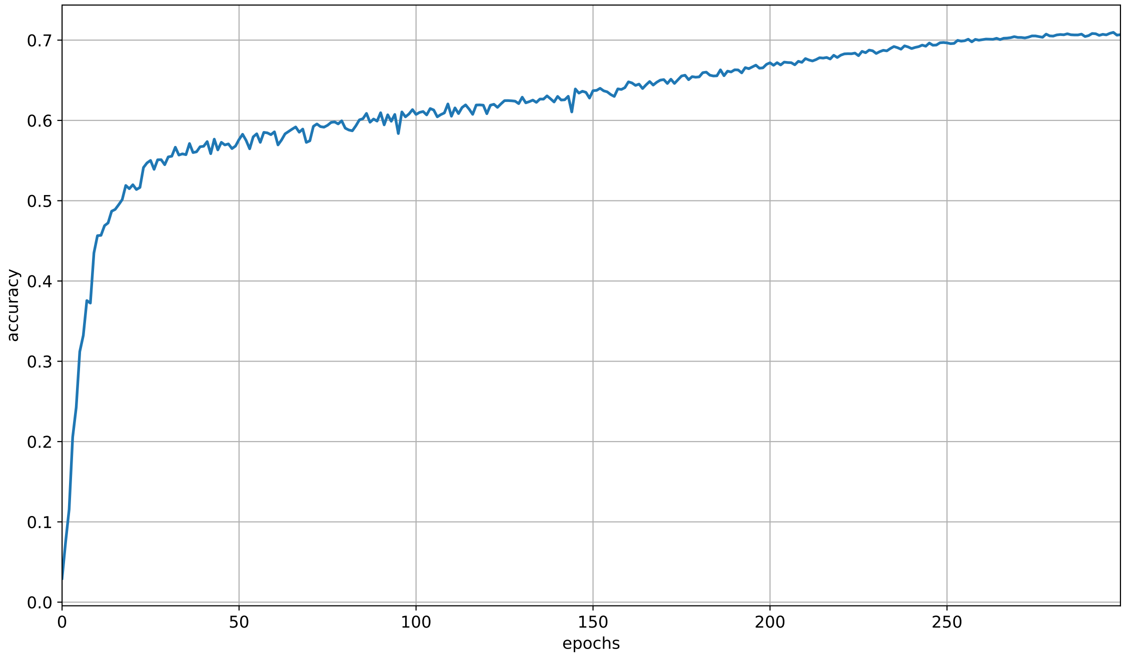

Figure 2: Accuracy of the U-Net encoder on the test set of Imagenet

Since pretraining on Imagenet [4] produced the most improvement, we experimented more with this approach. We were curious how much the performance of the pretrained model influenced the final scores. To measure this, we trained a U-Net encoder on Imagenet, and saved its weights at several checkpoints.

Figure 2 shows the progress of the validation accuracy of the encoder. As can be seen, the encoder keeps improving over 300 epochs on Imagenet. However, when used as initial weights, the checkpoints did not produce significantly different results after the second epoch. Variance testing, and other experiments will need to be concocted to be more certain, but this suggests that models for medical image segmentation might only need pretraining on Imagenet for a few hours, isntead of weeks. (Although we should note that Synapse is a relatively large dataset. We have not yet investigated whether the same phenomenon holds for smaller datasets as well.)

Skip connections

Convolutional networks with skip connections have been successfully used for both classification and segmentation for a long time. Specifically residual U-Net based models are widely used. On the Synapse dataset itself, the state of the art Swin U-Net [3] model uses identity skip connections. For this reason, we have tried different versions of identity skip connections.

In the down pass, the channel size is doubled within a block. Skip connections require the input and output of the block to be the same shape. We tried two possible solutions: filling the missing channels with zeros, and repeating the input to double its channel size.

In the up pass, we also tried two possible configurations: at the end of the skip connection, either just add the value coming from the level below, or add both that and the value coming from the corresponding level in the up path.

However, no combination of these approaches resulted in improved performance; most models with skip connections introduced proved to be unstable, even when advanced training techniques, such as layer scaling or stochastic depth were used.

Downsampling and channel change

The final examined architectural feature was the downsampling module of the down pass. The original U-Net used max pooling for downsampling, and changed the channel of the input inside the basic block. We tried out to different approaches. For the first one, we simply replaced the max pooling layer with a 2×2 convolutional layer with stride 2. For the second one, we kept the convolutional layer for downsampling, but we changed the basic block, so it would not change the number of channels in the input; instead, we let the 2×2 convolution used for downsampling change the channel size. This resulted in a significant increase in the number of trainable parameters.

However, as can be seen in Table 2, neither architecture managed to outperform the parameterless max pooling downsampling.

downsampling	acc ↑	AUC ↑	AP ↑	DSC ↑	IoU ↑	HD95 ↓
maxpool	0.989	0.981	0.796	0.773	0.659	0.004
conv (channel change)	0.989	0.945	0.710	0.709	0.595	0.005
conv (no channel change)	0.988	0.945	0.710	0.709	0.599	0.004

Table 2: Result of different downsampling techniques

Other models

We also briefly experimented with a ResNet50 based variant. The model was an Attention U-Net [5], meaning its lateral residual connections did not simply concatenate the two signals, but performed an attention gate. Initial experiments produced worse results than the base U-Net. We plan to do further testing on these models.

Conclusions

We managed to produce results that are similar to those given in [3]. (Our scores tend to be slightly worse. This, however, can be accounted for by the fact that we only use part of their training set, as we preserve the other part for validation.) Figures 3 and 4 show the performance of the best model.

We investigated several architectural changes, but neither of them outperformed the base U-Net. Since we know that there are better architectures for Synapse – such as the Swin U-Net –, this suggests that choosing these modules via a line search is not productive, and that the base U-Net occupies a local optimum within the hyperparameter space of its attributes.

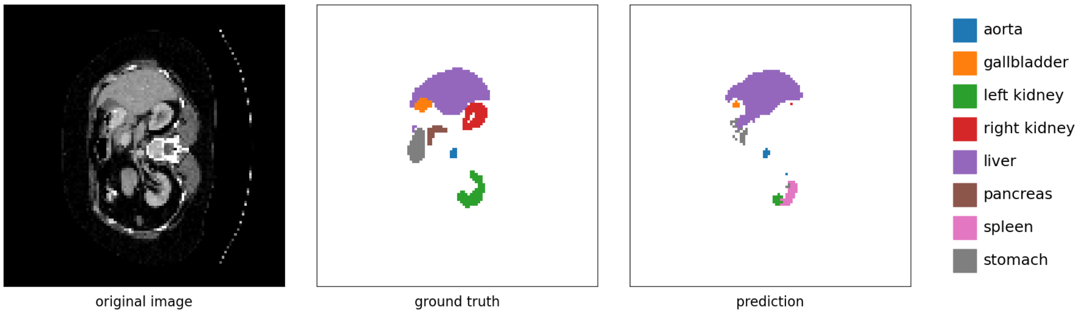

Figure 3: Example validation image, ground truth mask, and model prediction

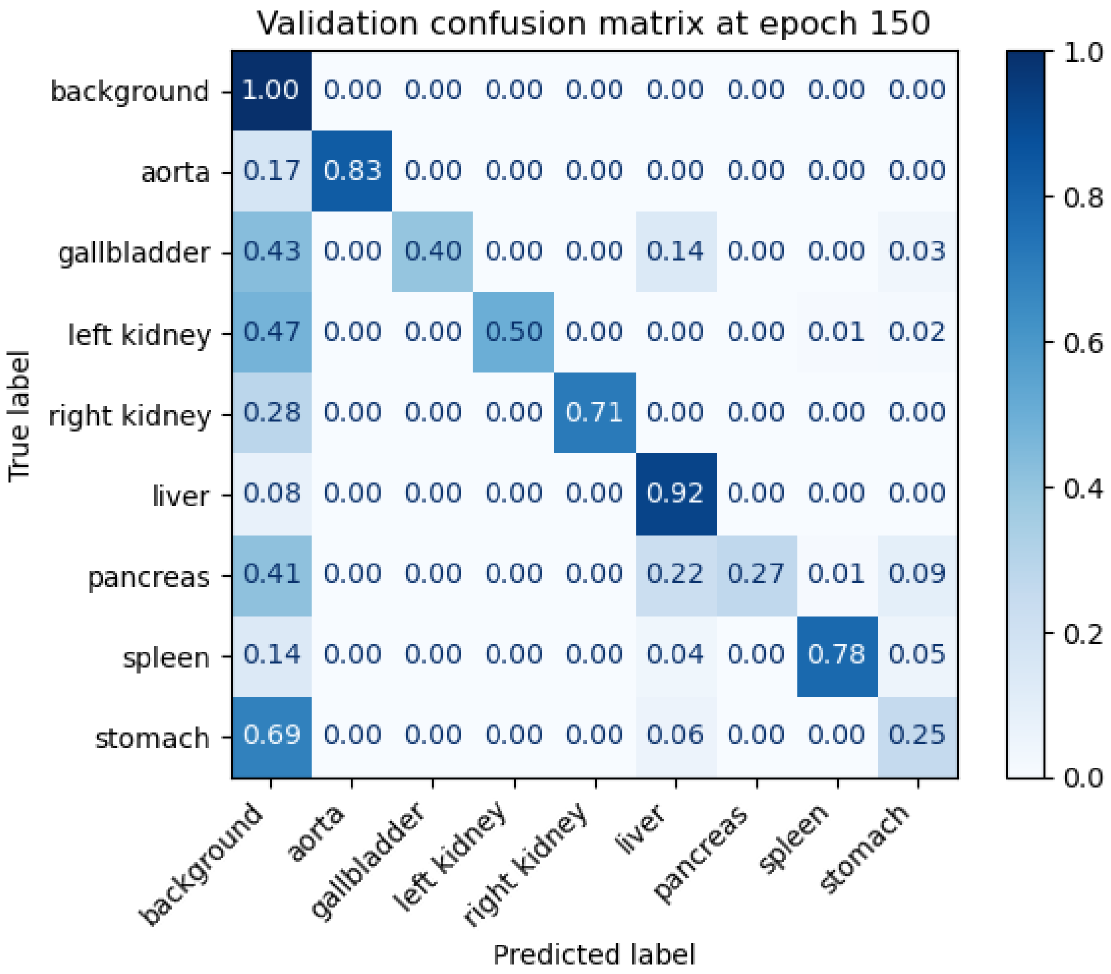

Figure 4: Example confusion matrix

7 Future work

We plan to investigate the effect of different pretraining methods – both on ImageNet and on domain-specific datasets – on model performance on Synapse, as well as other medical image datasets.

References

[1] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, November 2015, pp. 234–241.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] Hu Cao et al. Swin-Unet: Unet-like pure transformer for medical image segmentation. arXiv preprint, May 2021. arXiv:2105.05537v1 [eess.IV]

[4] O. Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **115**, 2015, pp. 211–252.

[5] O. Oktay et al. Attention U-Net: Learning where to look for the pancreas. arXiv preprint, April 2018. arXiv:1804.03999v3 [cs.CV]