

# A 2D U-Net for Automated Kidney and Renal Tumor Segmentation

Joseph Chen<sup>1</sup> and Benson Jin<sup>2</sup>

Independent Researchers jchen42703@gmail.com<sup>1</sup>  
jinb2@bxscience.edu<sup>2</sup>

**Abstract.** Kidney and renal tumor segmentation are critical aspects to the diagnosis process. However, segmentation is a time consuming and tedious task, especially for volume segmentation. To help with this issue, we test a simple two-dimensional U-Net architecture for automating the segmentation process for both regions of interests. In doing so, we found that the vanilla U-Net was able to achieve a local tumor-kidney test dice of 0.91 and tumor-only dice of 0.25 and leaderboard scores of 0.85 and 0.22.

**Keywords:** segmentation · kidney · convolutional neural networks.

## 1 Introduction

Kidney cancer is among one of the ten most common cancers in both men and women (American Cancer Society, 2018). The segmentation—the pixel-wise masking of region of interests in images—of the kidney plays a key role in the diagnosis process. However, segmenting medical images, such as computed tomography (CT), manually is a time-consuming process. The reason is medical images are often high-resolution and volumetric, as stacks of 2D images, and radiologists often must accurately produce segmentation masks for each slice or 2D image during the diagnosis and treatment process.

As such, the development of tools to automate the segmentation process or support radiologists' segmentation speeds up the process and indirectly, speeds up the diagnosis process. Convolutional neural networks, such as the U-Net, are an example of such tools.

### 1.1 Related Works

Yu et al. presented a novel architecture, Crossbar-Net for the automated segmentation of kidney tumors in CT scans. Their approach was characterized by using orthogonal patches with two convolutional neural network sub-models in a boosting training approach. Their high dice scores, ranging from 0.81 to 0.88, indicate that convolutional neural networks are effective algorithms for automating kidney tumor segmentation, and likely for kidney segmentation, as well.

Similarly, Ronneberger et al. proposed a convolutional neural network architecture, the U-Net, for medical image segmentation. It was successful due to it aggregating low-level information into higher level layers through skip connections. Similarly, we test a recursive version of this architecture in this study.

Building on the original U-Net neural network, Isensee et al. created the nnU-Net framework for automatic general medical image segmentation. Their framework centered around utilizing vanilla U-Nets rather than ones with add-ons, such as residual connections or pyramid pooling. They were able to achieve state-of-the-art results, placing first in the 2018 Medical Segmentation Decathlon and second in the 2018 MICCAI Brain Tumor Segmentation Challenge (BraTS). Furthermore, the researchers suggested that for highly anisotropic datasets, a 2D U-Net would be favorable. Likewise, the interpolated version of this challenge's dataset is highly anisotropic across the z-spacing. Therefore, we chose to test a 2D U-Net as opposed to its 3D counterpart.

## 2 Materials

The neural networks and training were done using tensorflow's keras package. batchgenerators (<https://github.com/MIC-DKFZ/batchgenerators>) was utilized for the data augmentation. Training was strictly done in Google's Colaboratory, using their K80 and T4 GPUs.



*Localization Pathway* This pathway describes the upsampling sequence. The output from the previous layer are upsampled, doubling the input size using bilinear interpolation. Then, a skip connection is made with the output of the last sub-block on the same level, and the concatenated output is passed through two sub-blocks, as described in the context pathway. This sequence of upsampling and convolutions is done for 5 upsampling operations. At the final layers, after the last upsampling-convolution sequence, the network is topped off by a 1x1 convolution. The softmax is calculated in the loss function.

*Filters* The filters were automatically set based on the user-specified starting filters, the number of filters for each convolution at the top layers (1st and last blocks). The rule is:

$$f' = f_0 * 2^l \quad (1)$$

where  $f_0$  is the number of starting filters and  $f'$  is the number of filters at level,  $l$ . In this case, the number of starting filters is 20, done for six levels, or five pools.

### 3.3 Loss Function

The loss function is sparse version of the loss function in the nnU-Net paper, where the loss is simply a sum of the cross entropy and the mean soft dice loss. This was chosen as the loss function because of how experimentally, it converged much more smoothly and quickly than vanilla soft dice loss or tversky loss. Additionally, since it contained soft dice loss as part of the calculation, it helped the U-Net to learn to deal with the class imbalance, circumventing their habit to only predict the background class. The only differences from the loss in Isensee et. al.'s nnU-Net are softmax is computed in the loss functions themselves and they accept logits and non-one-hot encoded masks as labels. The differentiable dice loss is defined as:

$$L_{dice} = \frac{1}{C} \sum_{n=0}^C \frac{2 * \sum_i p_{i,j} g_{i,j}}{\sum_i p_{i,j} + \sum_i g_{i,j}} \quad (2)$$

where  $C$  is the number of classes,  $p_{i,j}$  is the softmaxed prediction, and  $g$  is the one-hot encoded groundtruth array. The final loss is calculated as:

$$L = L_{dice} + L_{CE} \quad (3)$$

### 3.4 Training

The model was trained using the Adam optimizer, with an initial learning rate of 0.0001. It was trained for 1920 iterations, with a batch size of 7. Whenever the validation loss did not drop by at least 0.005 for 30 epochs, the learning rate was lowered by a factor of 0.8, with a minimum learning rate of 0.000001.

*Positive Class Sampling* For each batch, five out of the seven sampled slices were slices containing at least one non-background pixel. The other two slices were randomly sampled.

*Data Augmentation* The slices were then randomly cropped to 224x224 images, randomly scaled in the range (0.6, 1.25) with a probability of 0.1, and horizontally and vertically flipped half the time.

### 3.5 Inference

The inference and post-processing was nearly identical to Isensee's nnU-Net approach. For instance, the 224x224 predictions had to be aggregated across overlapping patches into the original volume size. The only difference was that the test time augmentation only included mirroring, excluding Gaussian noising.

<i>Model</i>	<i>Average Precision</i>	<i>Average Recall</i>	<i>TK-Dice</i>	<i>TU-Dice</i>	<i>Parameters</i>
2D U-Net	(0.914, 0.334)	(0.883, 0.238)	0.91	0.25	12M
Actual Results	—	—	0.858	0.224	12M

**Table 1:** The dataset was randomly divided in the [0.8, 0.1, 0.1] train/validation/test splits, and the metrics are calculated based on the local test set. TK is the dice score for tumor and kidney. TU is the dice score for just tumor. The precision and recall scores refer to the (kidney, tumor).

## 4 Discussion

The results on the initial test set didn't translate as well for the actual test set. This may be due to how small the test set we used in the local experiments, which lead the results to not translate well. Ensembling on multiple folds, as well as more extensive test-time augmentation, would help with us overfitting on our test set.

However, in both cases, the common theme was that the networks were much better at kidney segmentation versus tumor segmentation. It often missed tumors by a couple slices. This may be due to the model lacking enough complexity to handle both the kidney and tumor segmentation. Future works should look towards 30+ starting filters in the Recursive U-Net because local experiments with just binary tumor segmentation have demonstrated that it seems to be the bare minimum to get the network to properly learn. This can easily be done through even smaller batch sizes with accumulated gradients. It could have also been due to insufficient data, which can be addressed through more data augmentation.

Although other experiments were performed, most were not worth mentioning. The only result that was interesting was how the sigmoid counterpart of the pipeline actually performed better. However, we decided not to submit it, because we did not know why it did perform better.

## Bibliography

- American Cancer Society (2018). Key Statistics About Kidney Cancer. <https://www.cancer.org/cancer/kidney-cancer/about/key-statistics.html>. pages
- Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P. F., Kohl, S., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S. J., and Maier-Hein, K. H. (2018). nnu-net: Self-adapting framework for u-net-based medical image segmentation. *CoRR*, abs/1809.10486. pages
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597. pages
- Yu, Q., Shi, Y., Sun, J., Gao, Y., Dai, Y., and Zhu, J. (2018). Crossbar-net: A novel convolutional network for kidney tumor segmentation in CT images. *CoRR*, abs/1804.10484. pages