Improving Loss Functions for Automatic Semantic Segmentation of Kidney and Kidney Tumor

Mahira Jalisha
1712803642
mahira.jalisha@northsouth.edu
North South University

Ibtiaj Mahmud Diljar
1712184642
ibtiaj.diljar@northsouth.edu
North South University

Mahdi Mohammad Shibli
1712784042
mahdi.shibli@northsouth.edu
North South University

Md. Navid Bin Islam
1712404642
navid.islam@northsouth.edu
North South University

Javeria Fazal
1712157642
javeria.fazal@northsouth.edu
North South University

***Abstract*—Automatic segmentation of 3D medical images is a challenging yet increasingly crucial task. Limited availability of data and computational power is a fundamental hurdle but beyond that, segmenting medical images is incredibly challenging as it requires more accuracy and better generality. The role of the loss function is pivotal in overcoming these challenges. In this paper, we experimented with different loss functions that are predominantly used in 3D segmentation tasks to improve both the kidney and tumor dice score. Furthermore, we proposed improving on some of the loss functions using log cosh and weights. We focused on improving the segmentation of rarer classes and thus consequently improving the overall segmentation score.**

***Index Terms*—3d U-Net, kits19, Dice Soft, Weighted Dice Soft, Semantic segmentation**

1. Introduction

Kidney tumors are one of the significant health issues of current times. Each year there are more than 400,000 cases. The most common treatment for kidney tumors is surgery. To perform a surgery, the surgeon needs a precise detection of the tumors. Semantic segmentation of the kidneys and the tumors is a good first step towards improving the treatment outcome. It fast-forwards the processing steps and helps the attempts to relate tumor morphology to surgical work [1].

Because of the publicly accessible databases, semantic segmentation is one of the hottest topics in medical image computation. Semantic segmentation of kidneys and kidney tumors helps to characterize the tissue, but it imposes an immense manual effort burden than most nephelometry scores. Reliable automatic semantic kidneys and kidney tumors would open the path to full automation of different nephelometry scores and kidney tumor morphology studies on unparalleled scales. Because of the scarcity of labialized public data of kidney tumors, only a handful of sophisticated algorithms specialize in kidney tumor segmentation [2]. The kidney tumor segmentation challenge tries to solve this by providing high-quality CT scans for training and testing.

The kits19 Grand Challenge [3] dataset comprises 300 Computed Tomography(CT) scan images, of which 210 cases are segmented and are intended for training and validation while the rest of the 90 cases are reserved for testing.

A large portion of successful medical image segmentation algorithms is based on revisions of the U-Net architecture. But we can not use it for 3D images with multiple layers or slices. Also, this model is not capable of handling a large amount of data. For segmenting 3D images, medical, 3D variants of U-Net are used [4]. It is an enhanced version of the basic architecture with the capability of automatic segmentation of 3d volumetric data from a sparse annotation.

Because of the 3D U-Nets' huge success, we picked this architecture from the implementation of MIScnn framework [5] and focused on implementing different loss functions and finding out the effects of different loss functions on the dice score.

2. Literature Review

According to the work of Sudre et al [6], the dice score coefficient which is mainly used for measuring segmentation performances can be used as a loss function which will help to improve performance for highly imbalanced datasets. The generalised Dice score which was proposed at Crum et al [7] was used by Li as a loss function for training deep convolutional networks. This can be expressed as:

$$GDL = 1 - 2\frac{\sum_{l=1}^{2} w1 \sum_{n} r_{ln} p_{ln}}{\sum_{l=1}^{2} w1 \sum_{n} r_{ln} + p_{ln}} \tag{1}$$

They found that with mildly imbalanced data, most of the loss function worked well. But, as the rate of imbalance increased, generalised Dice loss or GDL function was more definitive. As medical image datasets are more imbalanced in most cases, GDL would be a good loss function while working with them.

Aurelio et al [8] proposed a way of using the weighted cross-entropy loss function with the imbalanced datasets. The cost sensitive loss function was defined as,

$$j(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y_i \log(\hat{y}_i)\lambda - (1-y_i)\log(1-\hat{y}_i)(1-\lambda)] \tag{2}$$

Their function worked better or was in par with other functions on various imbalanced datasets.

In medical images, having less false negatives is more appreciated than having less false positives. But, segmentation of imbalanced dataset can have precision but may lack in recall which may result in false negatives. Salehi et al [9] proposed another loss function for achieving better trade-off between precision and recall in model training. They used the tversky index as a loss function and defined it as

$$T(\alpha,\beta) = \frac{\sum_{i=1}^{N} p_{0i}g_{0i}}{\sum_{i=1}^{N} p_{0i}g_{0i} + \alpha\sum_{i=1}^{N} p_{0i}g_{1i} + \beta\sum_{i=1}^{N} p_{1i}g_{0i}} \tag{3}$$

Here, they didn't need to balance the weights for training the model. Also, they used the α and β hyperparameters to determine the trade-off between precision and recall. By putting more emphasis on recall rather than precision made the loss function to work better on imbalanced data. The best result was achieved on β = 0.7, and had better results than the dice coefficient which can be expressed as α = β = 0.5.

Li et al [10] proposed a generalised Focal loss or GFL for precise learning of bounding boxes for dense object detectors. This loss function generalizes the discrete formulation of the original focal loss to it's continuous version. They expressed it as,

$$\begin{aligned}\text{GFL}(p_{y_l}, p_{y_r}) = &-|y-(y_l p_{y_l} + y_r p_{y_r})|^{\beta} \\ &((y_r - y)log(p_{y_l}) + (y - y_l)log(p_{y_r}))\end{aligned} \tag{4}$$

This GFL can be used as both Quality focal loss (QFL) and Distribution focal loss (DFL). QFL provides a better standard for localization and classification. On the other hand, DFL brings precision for bounding box estimation.

The work of Abraham et al [11] shows the use of focal loss with tversky index and created the tversky focal loss function. They used the baseline tversky loss function and improved on it.

$$TI_c = \frac{\sum_{i=1}^{N} p_{ic}g_{ic} + \epsilon}{\sum_{i=1}^{N} p_{ic}g_{ic} + \alpha\sum_{i=1}^{N} p_{i\bar{c}}g_{ic} + \beta\sum_{i=1}^{N} p_{ic}g_{i\bar{c}} + \epsilon} \tag{5}$$

$$FTL_c = \sum_{c}(1 - TI_c)^{1/\gamma} \tag{6}$$

This function had better recall on imbalanced datasets than the dice loss function or the basic tversky loss function.

The Log-cosh approach proposed by Jadon [12] is a variant of Dice Loss and inspired by a regression-based problem for smoothing the curve. Variations can be employed for the skewed dataset. According to the study, hyperbolic functions are used in terms of non-linearities such as the tanh layer. They are tractable as well as easily differentiable.

3. Proposed Methodologies

This section gives an overview of the different sections of our work. It mainly discusses the theories, techniques, and step by step workflow of the work

3.1. Loss functions

We applied log cosh to the focal tversky loss, in the hopes of making the curve smoother and improving the results. In addition, we experimented with the dice loss by adding weights to the classes, to improve on the harder examples.

3.1.1. Dice loss. Dice loss [6] is a standard loss function for 3D medical image segmentation. It is modified from the dice coefficient which is a metric to measure the segmentation by taking into account the overlap of prediction and ground truth. The dice coefficient is defined as -

$$\text{DSC} = 2\frac{|X \cap Y|}{|X| + |Y|} \tag{7}$$

While the dice loss is defined as -

$$\begin{aligned}DL_2 = 1 &- \frac{\sum_{n=1}^{N} p_n r_n + \epsilon}{\sum_{n=1}^{N} p_n + r_n + \epsilon} \\ &- \frac{\sum_{n=1}^{N}(1-p_n)(1-r_n) + \epsilon}{\sum_{n=1}^{N} 2 - p_n - r_n + \epsilon}\end{aligned} \tag{8}$$

We will apply the dice loss function on our dataset and compare the results with the Log cosh dice loss function, to see what changes this brings and if there is any scope of improvements.

3.1.2. Log Cosh Dice Loss. As Log cosh dice loss [12] has proven to be a good loss function for segmentation tasks, we aim to apply this loss function to create a baseline model. The function is defined as -

$$cosh(DL) = \frac{e^{DL} + e^{-DL}}{2} \tag{9}$$

Here, $cosh(DL)$ is the average of e^{DL} & e^{-DL}

$$log(cosh(DL)) = log[\frac{e^{DL} + e^{-DL}}{2}] \tag{10}$$

Log cosh is widely used in regression based problems for smoothing the curve, thus it is applied on the non-convex dice loss in the hopes of achieving an optimal result. We will compare the results with the focal tversky loss and observe how it performs on our dataset which has a small number of test samples and is highly imbalanced.

3.1.3. Focal Tversky. The Focal Tversky loss function [11] is a good option for highly imbalanced dataset as it prioritizes hard examples. Our dataset has a very limited number of tumor slices, a good portion of which only contains small instances of tumor, thus making them hard to predict or segment. As we plan to reduce the size of our images, the problem will be much more prominent. Thus we will apply the focal tversky loss and check how it performs on the tumor segmentation. Thus we will apply the equation 6 and check how it performs.

The tversky loss [9] adds weights to the False negatives and false positives, thus it can be viewed as generalized dice loss. When α = β = 0.5 the tversky loss becomes equivalent to the dice loss. For our experiments we will set the values of α to 0.7 and the value of β to 0.3 as it has been found through various experiments that it gives better results than other probable combinations.

With the addition of focal loss [10] with the tversky loss, we will have one more hyperparameter to tune, which will be the value of γ. By increasing or decreasing the value of γ, we can control how much weight to add to the rarer classes. We can tune the value of γ to find the best fit for our data.

3.1.4. Log Cosh Focal Tversky. We would like to further experiment on the Focal tversky loss function by applying log cosh to smoothen the curve. The function can be defined by -

$$cosh(FTL) = \frac{e^{FTL} + e^{-FTL}}{2} \tag{11}$$

$$log(cosh(FTL)) = log[\frac{e^{FTL} + e^{-FTL}}{2}] \tag{12}$$

3.1.5. Weighted Dice Soft. Considering the importance and recurrent use of dice soft in medical image segmentation tasks and the positive results obtained from adding weights to the rarer classes as seen in focal tversky, its justifiable to add weights to the dice score of each class as it would accentuate the rarer classes. This should produce better segmentation results compared to the regular dice loss.

$$WDSC = \frac{\sum_{i=1}^{N} w_i DSC_i}{\sum_{i=1}^{N} w_i} \tag{13}$$

We will experiment with the weights and compare the results to see what works best for our data.

3.2. 3D Unet and MIScnn

We will use 3D Unet with different loss functions for our experiment as it is a standard network model for 3D medical image segmentation. We will be using the standard 3D unet from MIScnn [5] which is an API specially designed for medical image segmentation. The model has 22,587,171 parameters, of which 22,581,283 are trainable. As our work is focused on improving the loss functions we did not bring any changes to the standard 3D U-net model.

4. Experiments

We performed data analysis and pre-processing on our data to prepare it for training. We applied heavy data augmentation before training it on various loss functions.

4.1. Dataset

The kits19 dataset comprises 300 CT scans, of which 210 are used for training, and the rest 90 are reserved for testing. The sample shapes are depth * height * width, where the depth varies while the height and width are fixed at 512. The images and the ground truth labels are provided in the NIFTI format with shape (num_slices, height, width). The number of slices is given from an axial view, where the slice index increases from superior to inferior. The slice thickness ranges from 1mm to 5mm. The average number of slices is 217, while the median is 110 approximately. The segmentation.nii.gz files contain the segmentations where 0 represents the background, 1 represents the kidney, and 2 represents the tumor. The patients layed upward during the scan, so the height-width origins lie to the patients' left anterior [3].

We did an 80-20 split on the dataset. We used case_0000 to case_00165 for training and case_00166 to case_00209 for validation. We excluded case ID's 15, 23, 37, 68, 125, 133. Case ID 15 and 37 were confirmed to be faulty by the challenge organizers and 23, 68, 125 and 133 were excluded by Fabian et al [13]. They excluded these cases because they were not sure of the correctness of annotations of these cases.

4.2. Pre-Processing

Due to hardware limitations we resized our data to a shape of (depth*256*256) and saved the resized samples in NIFTI format. We applied z-score normalization to avoid any issues with outliers and to prevent a single feature from dominating other features.

Due to the quantitative nature of CT scans, when examining the same organ the intensity values are expected to be identical even when they originate from different scanners and hospitals. Owing to this property, pixel intensities are clipped to some organ specific value range. For this reason we adopt a similar approach and clip the pixel intensities from -79 to 304.

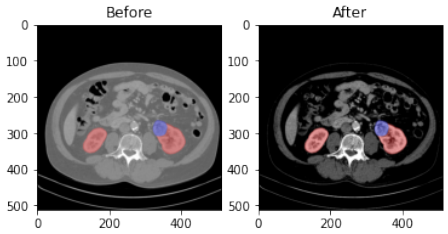

Figure 1. Before and after clipping an image

Furthermore, we resampled to a voxel spacing of 3.22*1.62*1.62 as the dataset is inhomogeneous and convolutional neural networks can't interpret voxel spacings natively. We tried to optimize the tradeoff between the amount of contextual information in the networks patch size vs the detail retained in the image, yet refrained from using a patch size of 80*160*160 like [13] due to hardware limitations, and instead used a smaller patch size of 80*80*80. Finally, we settled on a patch overlap of 40*40*40.

4.3. Data Augmentation

We applied an ample amount of Data Augmentation using the batch generators [14] framework to all the patches. We used scaling, rotation, elastic deformation, adjusted brightness and contrast, changed γ values and added gaussian noise. We set cycles to 2. We mirrored the images to increase the size of the dataset.

4.4. Training

Before training we made slight changes to the MIScnn framework as some of the functionalities were not suitable for our work. We changed the number of threads used during the creation of pickle files to 1 as the system was going into a deadlock. To monitor and log the training, we introduced new callbacks. We used unit batch size and a batch queue size of 1.

We initialized our network with a learning rate of 0.0001. During training we reduced the learning rate (minimum learning rate limit set to 0.00001) when the loss did not improve for more than five epochs. Training was stopped when the performance did not improve for 20 epochs. Weights which achieved the best performance during training were saved and compared with other weights on the validation set to ensure a better result.

Although we used different loss functions, our metrics to evaluate the network were the same during training. The metrics we used to monitor the performance of our model besides the loss function are the dice soft, dice cross-entropy, kidney dice, tumor dice and the background segmentation score. We logged the training to analyze how each loss deals with the three classes.

4.5. Loss Functions

We used the log cosh dice loss to create a baseline model, as it showed better performance than the dice loss and was on par with the Focal Tversky loss according to [12].

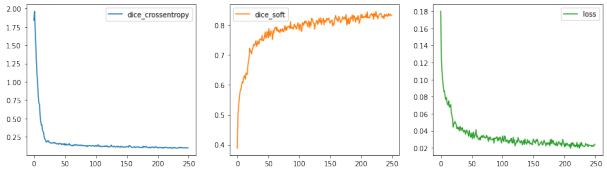

Figure 2. Training of logcosh Dice Loss

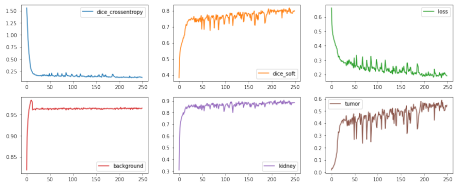

Figure 3. Training of Focal Tversky Loss

To check how well the log cosh dice loss works in comparison with the dice loss we trained our model with the dice loss and compared the results. As log cosh gave a sensitivity and specificity score close to that of Focal tversky [12], which is a loss function known for focusing on the hard cases, we trained another model with focal tversky loss to compare the tumor dice score.

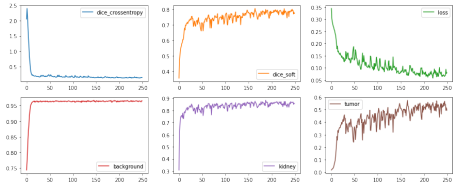

Figure 4. Training of logcosh Focal Tversky

For the configuration of the tversky loss, fixing the value of α to 0.7 made the value of β 0.3. Then we tuned the value

of γ and saved the best model for each value of γ. We then compared the evaluation results to pick the best models from them and recorded the γ value. During this process, we also checked how the model performs when we apply focal loss to (1 - tversky index) instead of the tversky index. Since tversky is a modification of dice loss, we applied log cosh to focal tversky to see if the score improves.

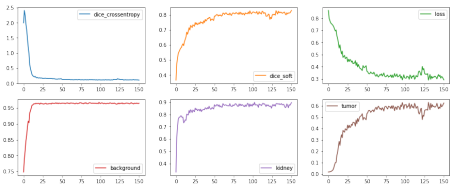

Figure 5. Training of Weighted Dice Loss

As dice loss is a standard for 3D medical image segmentation, we tried to improve on this loss by applying weights to each class.We noticed that although the trained models gave a decent dice score during training, the trained weights couldn't hold up the performance when it came to the validation set. Analysis of the train log revealed that the network segments the easier classes well, and as we are taking the average of the three classes, the score of the easier classes outweighs the harder ones. So while we were getting good background and kidney segmentation, the tumor segmentation was lacking. As seen in the focal tversky loss and in the weighted cross-entropy loss, adding weights to the rarer or harder classes improves the segmentation of those classes. So it seemed justifiable to apply weights to the underrepresented classes to improve their learning. By understanding how the training proceeds from the previous training sessions we applied different weights to each class and compared the results. We also applied log cosh to see if the results improved

5. Results

We used the evaluation metric provided in the kits19 website to evaluate our prediction. The metric is defined as-

$$s = \frac{1}{38}\sum_{i=0}^{37}\frac{1}{2}\Big(\frac{2 * n^i_{t,tp}}{2 * n^i_{t,tp} + n^i_{t,fp} + n^i_{t,fn}} + \frac{2 * n^i_{k,tp}}{2 * n^i_{k,tp} + n^i_{k,fp} + n^i_{k,fn}}\Big) \tag{14}$$

Using this metric we obtained an overall score of 0.854 in segmenting both the kidney and tumor together on our baseline model where we used log cosh dice loss. The kidney and tumor dice score was 0.870 and 0.407 respectively.

Our expectation turned out to be inaccurate as the dice loss gave a better overall score along with a better kidney dice and tumor dice. The overall score turned out to be 0.872, while the kidney score was 0.909 and the tumor dice was 0.494.

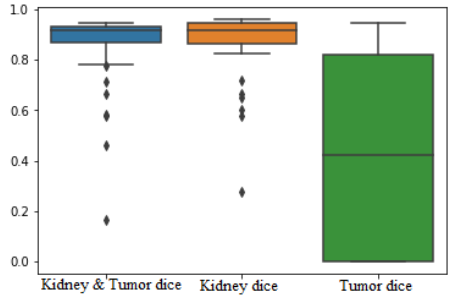

Figure 6. Performance of log cosh Dice Loss

Loss Function	Tumor + Kidney	kidney	Tumor
Dice Loss	0.872	**0.909**	0.494
Focal Tversky (1/γ = .75)	0.885	0.870	**0.535**
Focal Tversky (1/γ = 5)	0.825	0.758	0.455
Log cosh Dice Loss	0.854	0.870	0.407
Weighted Log cosh Dice Loss	0.881	0.850	0.515
Log cosh Focal Tversky	0.887	0.876	0.338
Weighted Dice Loss	**0.910**	0.902	0.521

TABLE 1. SCORE OF DIFFERENT LOSS FUNCTIONS

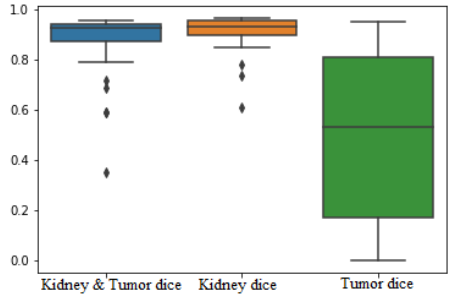

Figure 7. Performance of Dice Loss

From our experiments with the focal tversky loss we achieved the best results for 1/ γ = 0.75. The kidney and tumor segmentation score for this version was 0.885. This version also gave the best tumor dice with a score of 0.535. The kidney score was 0.870.

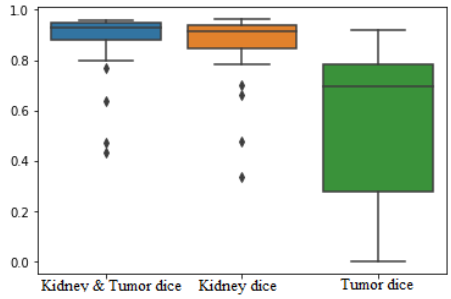

Figure 8. Performance of Focal Tversky Loss

Similar to the log cosh dice loss, the log cosh focal tversky did not not improve on the score achieved from

the focal tversky loss. The kidney and tumor segmentation score, kidney segmentation score and tumor segmentation score was 0.887, 0.876 and 0.338 respectively.

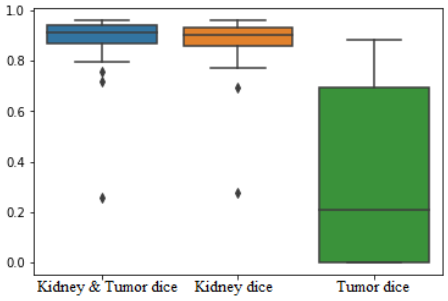

Figure 9. Performance of log cosh Focal Tversky

The weighted dice loss gave the best result, compared to the other scores, with an overall score of 0.910. The kidney and tumor dice was 0.902 and 0.521 respectively. Both the scores are close to the best scores we achieved in the respective criteria.

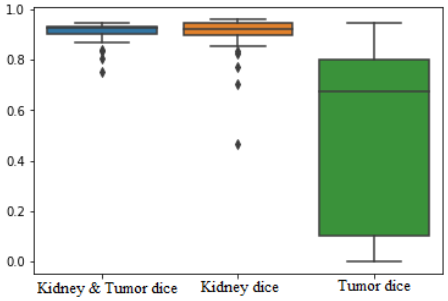

Figure 10. Performance of Weighted Dice Loss

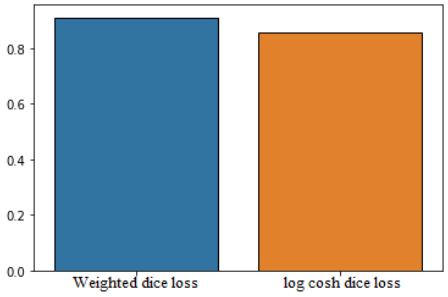

Figure 11. Comparison of the kidney and tumor dice of our best model with the baseline model

Comparing our baseline model with our overall best model we see a significant improvement in the tumor dice and a minor improvement in the kidney dice.

As seen in fig. 13, the log cosh dice loss which was used for our baseline model failed to detect the tumor. It only detected half of the kidney segmentation of one kidney.

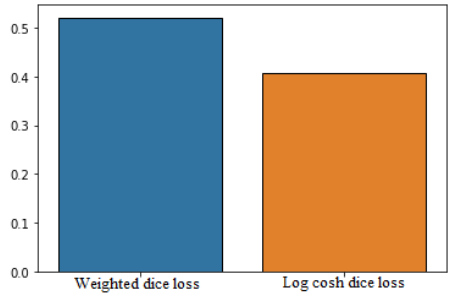

Figure 12. Comparison of the tumor dice of our best model with the baseline model

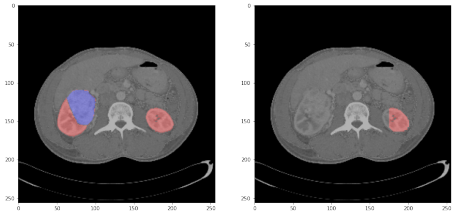

Figure 13. Comparison of the log cosh dice loss with the ground truth

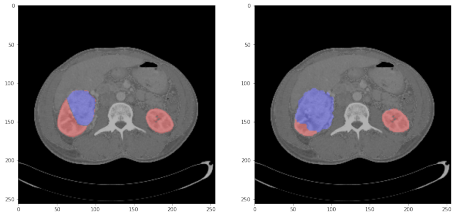

Figure 14. Comparison of the weighted dice with the ground truth

In contrast to fig. 13, in fig. 14, although we can see some regions with false positive cases, the model was able to predict the kidney accurately while predicting the entire tumor.

6. Conclusion

In this paper, we proposed a different loss function of a 3D-Unet model specially designed for kidney and tumor segmentation. Our approach gave a better kidney dice score while providing a decent tumor dice. Due to hardware limitations, we may not have achieved the best results in terms of segmentation, but our approach shows promising results and is on par with the state of the art approaches. Experimental results and comparisons with other methods indicate that our method achieves a very competitive performance with an average dice coefficient equal to 0.902 for kidney segmentation and 0.521 for tumor. Anyone can use this strategy with careful optimization to translate for various segmentation applications.

References

[1] A. Taha, P. Lo, J. Li, and T. Zhao, "Kid-Net: Convolution networks for kidney vessels segmentation from ct-volumes," 2018.

[2] Q. Yu, Y. Shi, J. Sun, Y. Gao, J. Zhu, and Y. Dai, "Crossbar-Net: A novel convolutional network for kidney tumor segmentation in CT images," 2018.

[3] N. Heller, N. Sathianathen, A. Kalapara, E. Walczak, K. Moore, H. Kaluzniak, J. Rosenberg, P. Blake, Z. Rengel, M. Oestreich, J. Dean, M. Tradewell, A. Shah, R. Tejpaul, Z. Edgerton, M. Peterson, S. Raza, S. Regmi, N. Papanikolopoulos, and C. Weight, "The KiTS19 challenge data: 300 Kidney tumor cases with clinical context, Ct semantic segmentations, and surgical outcomes," 2019.

[4] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation BT - Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2016.

[5] D. Müller and F. Kramer, "MIScnn: A Framework for Medical Image Segmentation with Convolutional Neural Networks and Deep Learning," *arXiv*, 2019.

[6] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10553 LNCS, pp. 240–248, 2017.

[7] W. R. Crum, O. Camara, and D. L. Hill, "Generalized overlap measures for evaluation and validation in medical image analysis," *IEEE Transactions on Medical Imaging*, vol. 25, no. 11, pp. 1451–1461, 2006.

[8] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, "Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function," *Neural Processing Letters*, 2019.

[9] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10541 LNCS, pp. 379–387, 2017.

[10] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," *arXiv*, pp. 1–14, 2020.

[11] N. Abraham and N. M. Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2019-April, pp. 683–687, 2019.

[12] S. Jadon, "A survey of loss functions for semantic segmentation," pp. 1–7, 2020.

[13] F. Isensee and K. H. Maier-Hein, "An attempt at beating the 3D U-Net," *arXiv*, pp. 1–8, 2019.

[14] F. Isensee, P. Jäger, J. Wasserthal, D. Zimmerer, J. Petersen, S. Kohl, J. Schock, A. Klein, T. Roß, S. Wirkert, P. Neher, S. Dinkelacker, G. Köhler, and K. Maier-Hein, "batchgenerators - a python framework for data augmentation," Jan. 2020.