

# Technical report KiTS 2019

Adel Zakirov

Independent researcher

**Abstract.** This technical report on my solution for KiTS 2019 Challenge. I build an end-to-end pipeline for kidney and tumor segmentation. My pipeline consists of three modules: (1) coarse segmentation step, (2) fine kidney segmentation, (3) fine tumor candidates segmentation on the kidney(s), (4) choosing between candidates. All 4 steps are based on 3D sem-seg model with some modifications in their architecture, training policy, data representation.

**Keywords:** Segmentation · Kidney · Tumor.

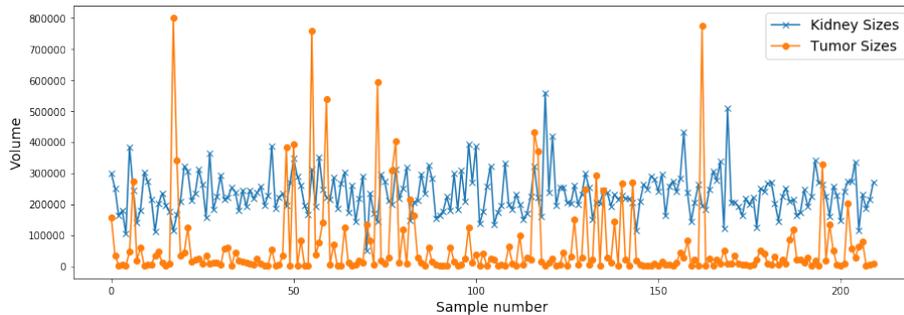
## 1 Introduction

KiTS 2019 Challenge represents a semantic segmentation task, which is very well investigated in the deep learning community as well as in medical image analysis society. Nevertheless, the task is not that trivial. First of all, the dataset [1] is three dimensional and consist of 210 training samples. Three dimensions and limited data amount makes it hard to simply use well known deep neural networks for semantic segmentation without any tricks. Fine-tuning is also a problem here, because of the small amount of pretrained models available. Moreover, datasets used to train these models are way too small, comparing to ImageNet [2], which is used as a pretrain dataset in 2D tasks. One will consider the task as slice-wise 2D semantic segmentation, but by my experience 3D models, in the end, get better accuracy scores and work more robust. I tried both approaches and came up with an end-to-end solution, that consist of 3D models only. Initial idea was to build some kind of ensemble of 3D and multiple (axial, sagittal and coronal) 2D models, but 2D models performed worst during validation even with some tricks that I used. All methods, techniques, and models that I tried but do not use in the final pipeline are described in section 4.

## 2 Dataset

Dataset is described in [1]. Great effort has been put into samples collection and annotation. Nevertheless, I want to give some bits of advice, that might help in future annotations. First of all, in my experience, I found ITK-Snap tool [3] the most convenient for 3D annotations in CT images. It has 3D brush, and 3 different views, that makes it easy and fast to use. Secondly, for tumor segmentation, it is better to crop ROI from the 3D sample. Skipping all of the irrelevant

volumetric information and focusing only on potential tumor position will make it much easier for annotators to make a pixel-perfect manual segmentation of tumor. I also believe that images from more CT tool manufacturers, as well as a more different field of view, should be presented in the dataset to make it as representative and possible. This is the best way to make general, accurate and robust model. I hope these remarks will help in the future researches. One of the open questions is a tumor on both kidneys simultaneously or multiple tumors on one kidney (if these cases are physically possible). I am not a specialist, but it would be better to include such cases in the dataset. Moreover, different tumor sizes needed in the dataset. Figure 1 represents the kidney and tumor sizes for each sample in the training set. It is seen that tumor size can be very different from sample to sample, and the dataset is not balanced in that way.



**Fig. 1.** Kidney and Tumor volumes of training samples.

In my training setup I used 11 samples as a validation set, which is approximately 5% of dataset, and other as a train set. This cases are: case\_00085, case\_00121, case\_00138, case\_00189, case\_00195, case\_00069, case\_00109, case\_00171, case\_00158, case\_00077, case\_00128. I tried to stratify the dataset by tumor sizes, still, there might be a better way of splitting.

### 3 Methods

First of all, I would like to mention that I worked with the interpolated set, provided by authors, but my pipeline should fit original scales too. I consider the task as a 3D semantic segmentation problem. My pipeline consists of four modules: (1) coarse segmentation step, (2) fine kidney segmentation, (3) fine tumor candidates segmentation on the kidney(s), (4) choosing between candidates. All 4 steps are based on 3D sem-seg model with some modifications in their architecture, training policy, data representation.

### 3.1 Coarse segmentation

Initial 3D CT volumes are too big to analyze them straight forward. Moreover, there is too much background information, that can affect models performances. So the first step I make is cropping the ROI for kidney+tumor segmentation. I consider this task as a binary segmentation, where all the tumor values in the masks are set to 1, making one class with kidney mask. I rescale the images to 3.4x3.4x3.4 scale, so each dimension becomes smaller and each image can fit into GPU memory. For the training, I cut 128x128x128 pp volume randomly from each training sample on each epoch. Validation images remain the same, except padding to 16 divisible sizes. I am making simple preprocessing before training: all the HU values lower than -1024 and higher than 1500 are being masked with those numbers respectively. Padding is done with -1024 values for images and with zeros for annotations (masks). While training random flip (on all directions), random rotation in the range of (-10, 10) angles performed with probability 0.5 for each sample in training set. The model architecture is classical Vnet [4], loss function is soft Jaccard (SJ), score metric is Intersection over Union (IoU), learning rate (lr) changes according to Cosine Annealing LR policy [5] from 1e-3 to 1e-7 values, that I got from lr finder [6], optimizer is Adam (worked better than SGD). The model trained in this setup achieved 0.95 IoU on the validation set and gives a good ROI for further steps.

### 3.2 Fine kidney and tumor segmentation

The next step is more or less the same, except the scale of the images is now set to original and ROI is cropped using bounding box, calculated from coarse models predictions. The data manipulation while training is the same as in the previous step, but the model and training policy is different. Since this step form input for next, tumor segmentation, step, I wanted to make the model as accurate as possible. I have come up with an ensemble of two models. First one is Vnet based model with Densenet 169 [7] encoder, the second one is also Vnet based model with SE ResNext101 32x4d [8] encoder. The padding factor on validation is now 32 instead of 16 because of the model architectures. The ensemble of these two models gives me 0.97 IoU. The overall training policy is similar to coarse model training, except Densenet based model used binary cross-entropy as a loss function, to get more diversity for ensembling. The last procedure here is taking two (or one if there is only one component) largest connected components, from the predictions. I havent met a case, where model outputs more than 2 components, but still, it can be possible during testing, so I added this postprocessing in the pipeline.

### 3.3 Fine tumor segmentation

My experiments showed that the results are better if the tumor segmentation model is trained on the single kidney, containing tumor, cropped from the original image and background values masked out to -1024. All the kidney values are

set to 0 and tumor values to 1. The setup is the same as for fine kidney segmentation step. Nevertheless, there are test time augmentations (TTA) performed on the validation and testing sets. I use 12 augmentations, that are combinations of 90-degree rotations and flips on all dimensions. From the output of the model, I calculate all the connected components and name them candidates to be a tumor on the given kidney. From the model I also now the probability map (output before converting to logits). So the next step is to calculate the mean probability for each candidate and select only the one, that have probabilities higher than 0.8. This number is empirical and more research on this part is needed.

Single model without TTA gives 0.77 IoU on the validation while adding TTA gives 0.79 IoU. The ensemble of two models, both with TTA, gives 0.81 IoU on validation.

Altogether, the whole pipeline gives me 0.92 IoU score defined by KiTS 2019 Challenge on my validation set.

## 4 What did not worked?

In this section, I will briefly describe what else I tried and what did not work in my experiments.

- Multiclass segmentation - models are training slowly and not that accurate.
- 2D segmentation - the scores are lower and there are a lot of false positive. Also, the shapes of the reconstructed 3D objects are not that smooth. Even with pretrained networks, pretrained encoders and some good augmentations.
- Tumor segmentation without maskout - not working at all. For some reason, maskout helped me dramatically in this step.
- Lighter encoders like resnet50, resnet18, or classical Vnet/3D-Unet architectures gives lower scores on validation. This might be fixed with the bigger dataset.
- Kidney only segmentation (without tumor) - low scores.
- SGD - it feels like SGD is a better option for my lr policy, but still, experiments show the opposite. Even training on more epochs is not helping.
- Binary classification for tumor detection for step 3 - I tried different resnets and densenets to train the classifier, that answers if the given volume contains a tumor in it. I got only 0.72 F-score, which is not good enough

## References

1. Nicholas Heller, Niranjana Sathianathan, Arveen Kalapara, Edward Walczak, Keenan Moore, Heather Kaluzniak, Joel Rosenberg, Paul Blake, Zachary Rengel, Makinna Oestreich, Joshua Dean, Michael Tradewell, Aneri Shah, Resha Tejpaul, Zachary Edgerton, Matthew Peterson, Shaneabbas Raza, Subodh Regmi, Nikolaos Papanikolopoulos and Christopher Weight. The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes, 2019; arXiv:1904.00445.

2. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014; arXiv:1409.0575.
3. Paul A. Yushkevich and Joseph Piven and Cody Hazlett, Heather and Gimpel Smith, Rachel and Sean Ho and James C. Gee and Guido Gerig. User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability; *Neuroimage*, 2006, 31, 3, 1116–1128
4. Fausto Milletari, Nassir Navab and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, 2016; arXiv:1606.04797.
5. Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts, 2016; arXiv:1608.03983.
6. Leslie N. Smith. Cyclical Learning Rates for Training Neural Networks, 2015; arXiv:1506.01186.
7. Gao Huang, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger. Densely Connected Convolutional Networks, 2016; arXiv:1608.06993.
8. Saining Xie, Ross Girshick, Piotr Dollr, Zhuowen Tu and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks, 2016; arXiv:1611.05431.