

Project Title: Kidney Tumor Segmentation Challenge

Team

Mohamed Ahmed <mohahme@kth.se>

Rohit Saluja <rsaluja@kth.se>

Introduction

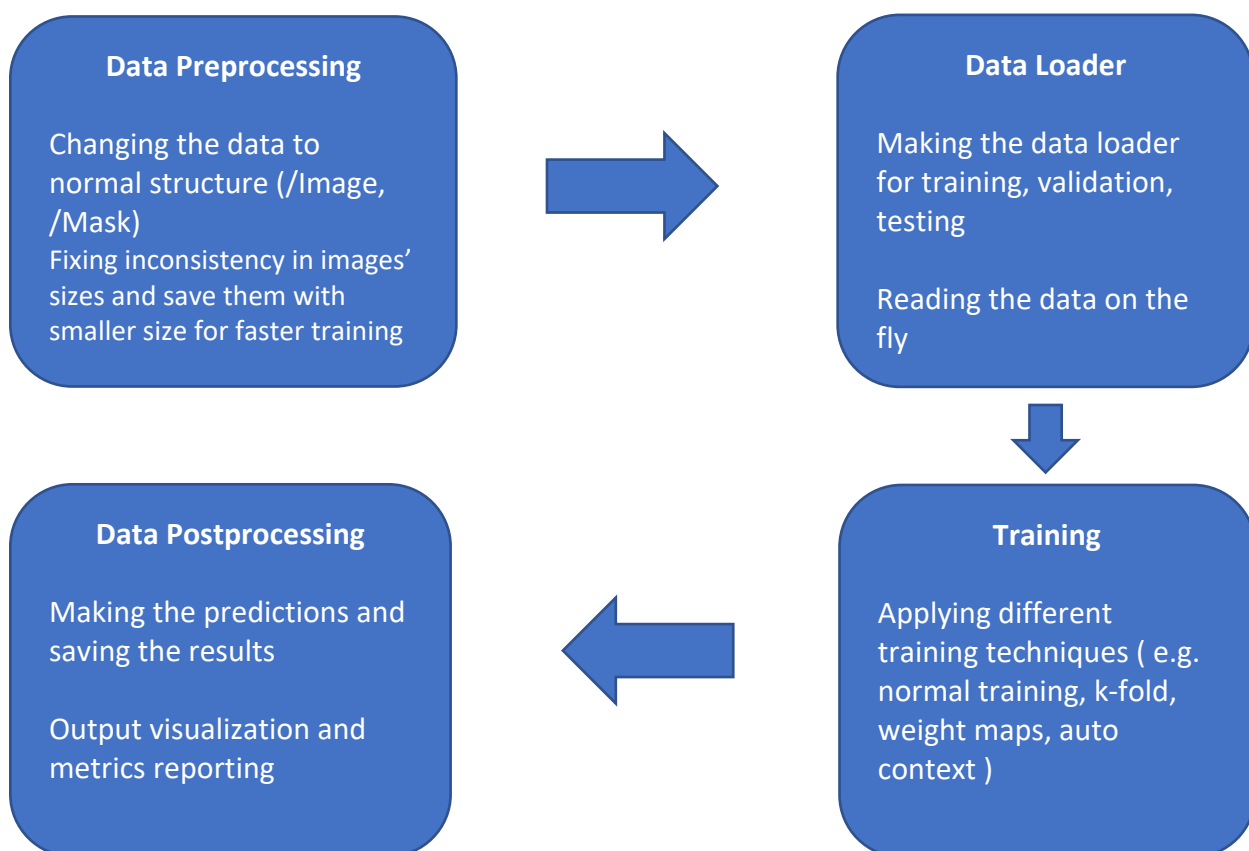
The challenge that our team picked for the project was first announced on March 15, 2019 and was held in conjunction with MICCAI 2019 in Shenzhen, China [1]. The motivation and the objective of the challenge are as follows.

Each year there are more than 400,000 thousand cases of kidney cancer and the most common way of treating them is by doing a surgery [2]. There is a lot of interest in finding out how a tumor morphology [3] is related to the surgical outcome and also in developing advanced surgical planning techniques [4]. Due to these reasons, exploring automatic semantic segmentation of a kidney tumor becomes an interesting challenge that can have a very fruitful impact.

The data

The challenge dataset contains arterial phase abdominal CT scans of 300 unique kidney patients along with their semantic segmentations. 210 of these were used for model training and validation, 90 were held out by the challenge organizers for testing.

Solution Pipeline



Tools Used

- Numpy, Nibabel, OS, Skimage, SimpleITK
- Implemented two pipelines, one with TensorFlow (keras) and one using Pytorch.
- In the pipeline with TensorFlow, we have experimented with multiple configurations.
- The pipeline built with Pytorch is basic vanilla pipeline. The motivation behind using both TensorFlow and Pytorch was to get a hands-on experience of building end to end pipelines with both the libraries. This aligns with the motivation of the course.

Models and Configurations

Mainly, we used U-Net for the segmentation task, experimented with different hyperparameters (e.g. batch norm, dropout, number of base features, number of output classes), and more advanced techniques like weight maps and autocontext.

For more information about how to configure a task, please read Keras/README.md.

Competition Metric

We have used two metrics to evaluate the performance of our model, (Normal) Dice Coefficient and custom Dice Coefficient which was provided by the competition organizers.

$$S = \frac{1}{90} \sum_{i=0}^{89} \frac{1}{2} \left(\frac{2 * n_{t,tp}^{(i)}}{2 * n_{t,tp}^{(i)} + n_{t,fp}^{(i)} + n_{t,fn}^{(i)}} + \frac{2 * n_{k,tp}^{(i)}}{2 * n_{k,tp}^{(i)} + n_{k,fp}^{(i)} + n_{k,fn}^{(i)}} \right)$$

Where the i th test case has the following confusion matrices

	Predicted Kidney or Tumor	Predicted Background
Labeled Kidney or Tumor	$n_{k,tp}^{(i)}$	$n_{k,fn}^{(i)}$
Labeled Background	$n_{k,fp}^{(i)}$	

	Predicted Tumor	Predicted Not Tumor
Labeled Tumor	$n_{t,tp}^{(i)}$	$n_{t,fn}^{(i)}$
Labeled Not Tumor	$n_{t,fp}^{(i)}$	

Figure 1: The confusion metrics used in the competition

Challenges Faced & Results

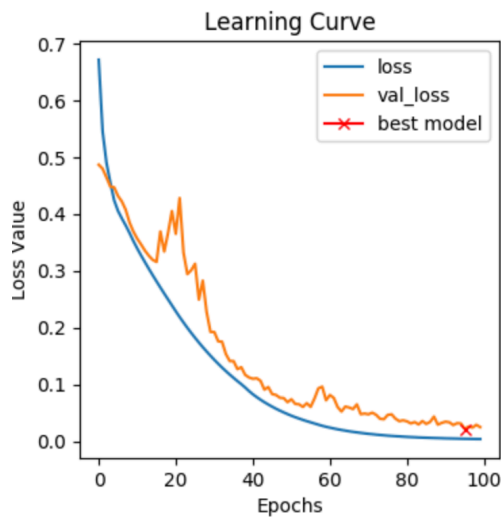


Figure 2: Dice Loss (Normal)

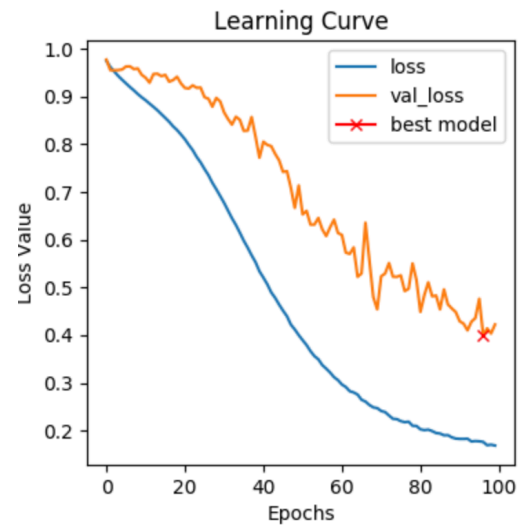


Figure 3: Using Dice Loss from the competition (100 epochs)

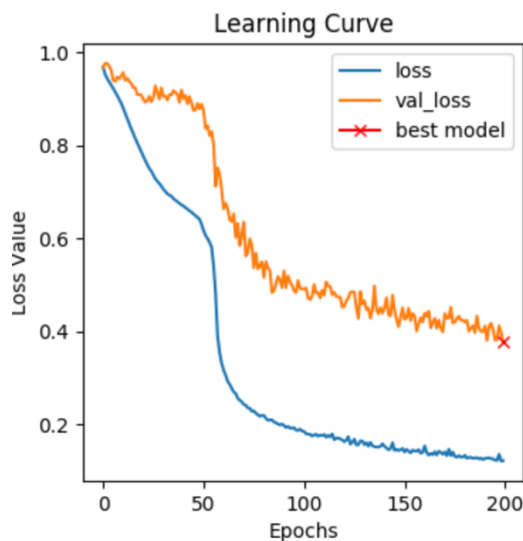


Figure 3: Using Dice Loss from the competition (200 epochs)

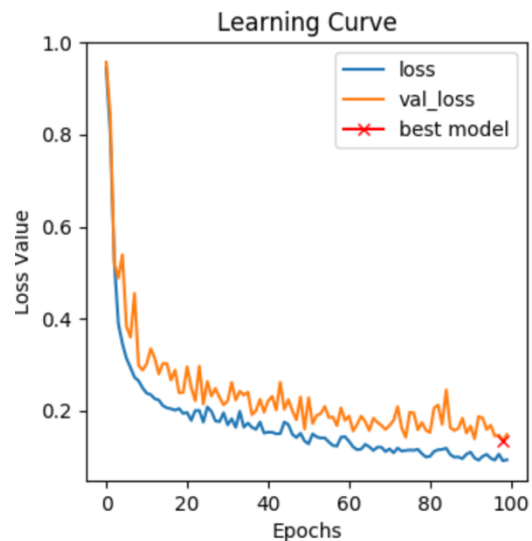


Figure 4: Using Dice Loss from the competition (100 epochs), increasing learning rate to .0001 from .00001

Initially, we found that a lot of input images don't have kidney or tumor class, so we decided to exclude them. In the beginning, the metric we used for performance evaluation was the normal dice loss over 100 epochs in order to get a benchmark and the model performed very well. Then we moved on to evaluate the model performance with the dice loss from the competition over 100 epochs, but the results were not as good as the normal dice loss.

We tried fixing this by doubling the number of epochs and increasing them to 200. The validation loss started saturating at around 0.4 which was still not good enough. In order

to further improve the performance, we increased the learning rate from .00001 to .0001 and evaluated the performance with the dice loss from the competition over 100 epochs. We were finally able to obtain good results with the custom dice loss from the competition; around .1 (Dice coefficient ~ .9).

We found how shuffling before training helps getting better performance. Also, we had a problem with the metric provided by the competition as they use Numpy for the calculation, but we had to change that to tensors, so the computation graph of Keras works. Furthermore, we tweaked this function by slicing kidney and tumor and flatten them then use them to calculate the loss to avoid gradient flow problem.

Results on the Testing data

When we did prediction on 90 cases provided as test set and submitted the results. That's what we got:

Mean Kidney Tumor Dice Loss	Kidney Dice Loss	Tumor Dice Loss
0.5229	0.8438	0.2020

As it can be observed above, the segmentation model is doing a great job at identifying the kidney but is performing poorly while detecting the tumor. We investigated this to find out the reason and it turns out that there is class imbalance in the training and validation dataset. The class distribution is as follows.

- Background class exists in 100% of the images.
- Kidney class exists in 32.66% of the images.
- Tumor class exists in 12.76% of the images.

Future Work

In order to improve the performance of the model at detecting the tumor, we would like to suggest the following solutions and would like to experiment with them in the future.

- Using images with tumor for training the model.
- Assigning more weight to the tumor class in the loss function while using the images of kidney and kidney with tumor.
- Experimenting with weight maps and auto context. However, we feel that this would not improve the model's ability to identify the tumor because the problem is mainly because of class imbalance.

References:

1. <https://kits19.grand-challenge.org/home/>
2. Kidney Cancer Statistics." World Cancer Research Fund, 12 Sept. 2018, www.wcrf.org/dietandcancer/cancer-trends/kidney-cancer-statistics
3. Kutikov, Alexander, and Robert G. Uzzo. "The RENAL nephrometry score: a comprehensive standardized system for quantitating renal tumor size, location and depth." *The Journal of urology* 182.3 (2009): 844-853.
4. Taha, Ahmed, et al. "Kid-Net: Convolution Networks for Kidney Vessels Segmentation from CT-Volumes." *arXiv preprint arXiv:1806.06769* (2018).