

Segmentation of Kidney Tumor*

Navdeep Dahiya¹ and Alok Sharma²

¹ Georgia Institute of Technology, Atlanta GA 30332, USA ndahiya3@gatech.edu

² Independent Researcher aloksharma.3105@gmail.com

Abstract. Medical Image Segmentation is a routine task in various clinical settings. There is a great interest in understanding the morphology of different human organs healthy or diseased. Until fairly recently, this task was done manually by clinical experts requiring years of difficult training tailored to specific medical fields. In this article we present a fully autonomous Machine Learning based method to segment Kidney and Tumor from human abdominal CT scans. Manually annotated data (210 datasets) were provided as a part of the Kits2109 MICCAI grand-challenge.

Keywords: Medical Image Segmentation · Deep Learning · Convolutional Neural Networks.

1 Background

Medical Image Segmentation is a routine task in various clinical settings. There is a great interest in understanding the morphology of different human organs healthy or diseased. Until fairly recently, this task was done manually by clinical experts requiring years of difficult training tailored to specific medical fields. The task of manual image segmentation is still very tedious and time consuming.

With the advent of machine learning and deep learning methods, there is great hope of being able to develop automatic image segmentation methods for different kinds of clinical applications and modalities. A common roadblock in the application of Machine Learning methods is the lack of manually annotated training data especially in Medical Imaging. In this regard, the Kits19, Kidney Tumor Segmentation challenge hosted on MICCAI grand-challenges website has provided expertly annotated data for Kidney and Tumor segmentation.

There are more than 400,000 new cases of kidney cancer each year [1], and surgery is its most common treatment [2]. Due to the wide variety in kidney and kidney tumor morphology, there is currently great interest in how tumor morphology relates to surgical outcomes, [3,4] as well as in developing advanced surgical planning techniques [5]. Hence the objective of the MICCAI challenge was to allow machine learning and computer vision practitioners to develop automatic Kidney and Tumor Segmentation tools.

This short manuscript describes our attempt at developing a fully automatic Deep Learning based method for Kidney and Tumor segmentation in arterial phase abdominal CT scans.

* [MICCAI Grand Challenges](#)

2 Data Preprocessing

The kits2019 MICCAI challenge organizers provided a total of 300 abdominal CT scans of patients who underwent partial or radical nephrectomy for one or more kidney tumors at the University of Minnesota Medical Center between 2010 and 2018. Out of the total 300, 210 cases were released during the competition phase for training algorithms and remaining 90 cases were withheld for testing phase. For each training case ground truth manual segmentation labels with one label for kidney and another for tumor or renal cysts were provided. These labels were created by medical students under the supervision of clinical chair, Dr. Christopher Weight.

Using this training data we trained fully convolutional neural networks (CNNs) for image segmentation. We divided the available training data into three cohorts of train/test/valid, as is standard practice in machine learning community. All the algorithms we trained/tested were based on 2D CNNs, Hence one of the first preprocessing steps for us was to convert the training and validation data sets from 3D volumes to single images by extracting individual slices of each dataset. Another aspect of data preprocessing is the dynamic range of the data. Typical machine learning toolkits are designed to work with natural grayscale or color image which are typically saved in 8 bit unsigned integer format. However, medical images typically have much higher dynamic range and hence we saved the individual slices in NUMPY compressed file format as 32 bit floating point images to preserve the original dynamic range of images.

A very basic yet key aspect of successfully training Deep Learning based models is data standardization. A key mathematical technique involved in such algorithms is Gradient Descent based optimization strategy. This algorithm can be very sensitive to the numerical range of input data and if problems occur, can lead to what's called vanishing and/or exploding gradients in the training process. We believe we suffered from this problem during our training process. Using the same CNNs and setting that we have used for successfully segmenting other kinds of Medical Imaging data, we could not train a network to be able to segment kidney/tumor data. Eventually, we applied a data preprocessing technique called feature wise mean normalization. This technique, in general, is used to bring the numerical values of the features being trained on similar scale. This is achieved by transforming the initial data, by subtracting the mean and dividing by the standard deviation of each feature. In case of 2D images we considered each pixel location as a unique feature. Hence we used all the available training data to calculate a mean "slice" and a standard deviation "slice". That is we calculated the mean and standard deviation of each pixel of a 512x512 grid. We used this mean and std to normalize each training image before feeding it to our neural network. This trick helped to get meaningful results from the training process.

We believe that we had to do this process because there were several training cases with extremely different dynamic ranges. Inclusion of these datasets in the training process would "throw-off" the gradient descent process leading to non-convergent or divergent behavior of the neural networks. Another possible

strategy to deal with such a scenario would have been to ignore the cases with divergent dynamic range. However, such a solution would not be ideal as we can still encounter such cases during testing phase. Our aim was to develop robust algorithms capable of dealing with real-world data which cannot always be perfect and as we expect. Figure 1, shows an example slice of case_00166 with extremely high dynamic range of -6986 to 18326 compared to normal range of -1024 to 3000. We believe this dynamic range variation prevents typical deep learning based algorithms from being effective in learning to segment images. Figure 2, shows a mean slice which is subtracted from all training/testing data before the forward pass of the training process.

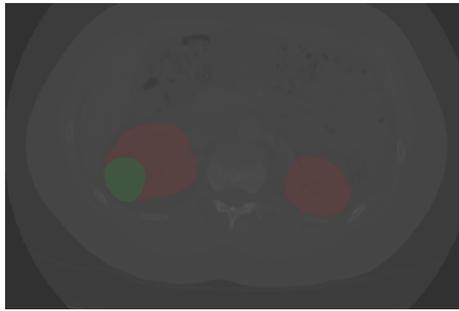


Fig. 1. A single slice of case_00166 with extremely high dynamic range.

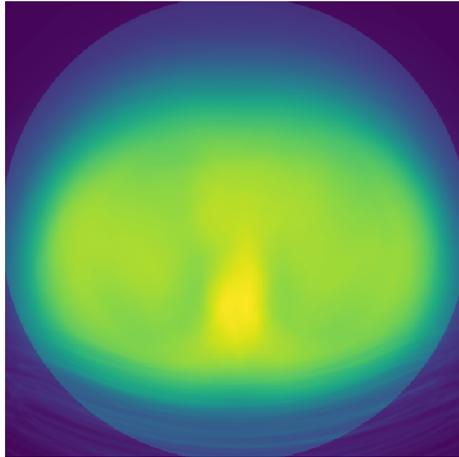


Fig. 2. A mean slice computed using all training data.

3 Network Architecture

3.1 ENet (Efficient Neural Network) CNN

We decided to use ENet [6] architecture to solve this semantic segmentation problem. ENet has been designed specifically for semantic segmentation with limited computation resources. Figure 3 shows the architecture of ENet The

| Name | Type | Output size |
|--|--------------|----------------------------|
| initial | | $16 \times 256 \times 256$ |
| bottleneck1.0 | downsampling | $64 \times 128 \times 128$ |
| $4 \times$ bottleneck1.x | | $64 \times 128 \times 128$ |
| bottleneck2.0 | downsampling | $128 \times 64 \times 64$ |
| bottleneck2.1 | | $128 \times 64 \times 64$ |
| bottleneck2.2 | dilated 2 | $128 \times 64 \times 64$ |
| bottleneck2.3 | asymmetric 5 | $128 \times 64 \times 64$ |
| bottleneck2.4 | dilated 4 | $128 \times 64 \times 64$ |
| bottleneck2.5 | | $128 \times 64 \times 64$ |
| bottleneck2.6 | dilated 8 | $128 \times 64 \times 64$ |
| bottleneck2.7 | asymmetric 5 | $128 \times 64 \times 64$ |
| bottleneck2.8 | dilated 16 | $128 \times 64 \times 64$ |
| <i>Repeat section 2, without bottleneck2.0</i> | | |
| bottleneck4.0 | upsampling | $64 \times 128 \times 128$ |
| bottleneck4.1 | | $64 \times 128 \times 128$ |
| bottleneck4.2 | | $64 \times 128 \times 128$ |
| bottleneck5.0 | upsampling | $16 \times 256 \times 256$ |
| bottleneck5.1 | | $16 \times 256 \times 256$ |
| fullconv | | $C \times 512 \times 512$ |

Fig. 3. ENet architecture. Output sizes are given for an example input of 512×512 .

initial stage contains a single block, that is presented in Figure 4. Stage 1 consists of 5 bottleneck blocks, while stage 2 and 3 have the same structure, with the exception that stage 3 does not downsample the input at the beginning (we omit the 0th bottleneck). These three first stages are the encoder. Stage 4 and 5 belong to the decoder

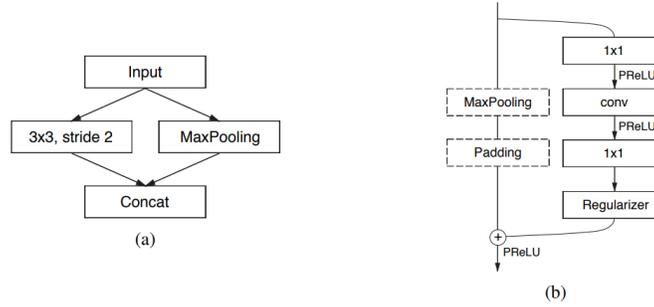


Fig. 4. (a) ENet initial block. MaxPooling is performed with non-overlapping 2x2 windows, and the convolution has 13 filters, which sums up to 16 feature maps after concatenation. (b) ENet bottleneck module. conv is either a regular, dilated, or full convolution (also known as deconvolution) with 3x3 filters, or a 5x5 convolution decomposed into two asymmetric ones.

3.2 Loss Function

One of the main challenges in training neural networks is data imbalance, which is particularly problematic in medical imaging applications such as tumor segmentation where the number of tumor voxels is often much lower than the number of background voxels. Training with unbalanced data can lead to predictions that are severely biased towards high precision but low recall (sensitivity), which is undesired especially in medical applications where false negatives are much less tolerable than false positives. Several methods have been proposed to deal with this problem including balanced sampling, two step training, sample re-weighting, and similarity loss functions.

In our experiments we decided to use a generalized loss function called Tversky loss function [7] based on Tversky index to address the issue of data imbalance and achieve much better trade-off between precision and recall.

Tversky index is defined as shown in Figure 5, where in output of softmax layer of our model, p_{0i} is the probability of voxel i be a tumor and p_{1i} is the probability of voxel i be background. Also, g_{0i} is 1 for a tumor voxel and 0 for a background voxel and vice versa for the g_{1i} . The Tversky loss is simply 1 minus

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

Fig. 5. Tversky Index

the average Tversky index of a batch of input data. We use $\alpha = 0.3$ and $\beta = 0.7$ in all our experiments.

4 Implementation Details

Since class imbalance is severe in this dataset, we decided to go with a two-stage approach. In stage 1, we combine kidney with tumor as foreground and rest is background. The model is trained to segment kidney + tumor as foreground. In stage2, we train another model on a 256 x 256 image , cropped around centroid of kidney + tumor blob, to predict three classes - tumor, kidney and background. We believe this will help to better segment tumor since the class distribution will be less skewed.

We use the Adam optimiser with learn rate of $1e^{-03}$ and batchsize of 8. We augment our training data with random rotations, horizontal and vertical flip, shear and zoom. The training pipeline has been implemented in Keras with Tensorflow backend.

5 Inference

We run stage 1 model to segment kidney combined with tumor from background. We then use a 256 x 256 cropped version (as described above) in stage 2 model to predict kidney, tumor and background pixels. In post processing, we overlay this 256 x 256 mask onto our stage 1 mask at correct locations. This gives us our final segmentation mask. In both these stages of processing, we use the mean and standard deviation calculated using training data to normalize testing datasets as well.

6 Results

We are getting encouraging results from our two stage deep neural network based segmentation pipeline. On ou hold-out test dataset of 20 cases, we are getting an average Dice score for kidney as 0.94. Segmenting tumor is considerably difficult but we are getting an average Dice score of 0.46. One of our observations is that usually our tumor segmentation fails when the actual ground truth is very small. These are very difficult cases to segment and very hard to do with automatic algorithms.

Our network takes about 3 seconds to process a case with 100 slices. Figure 6 shows our preliminary predicted segmentation masks on a 2D slice from case_00235 test set.

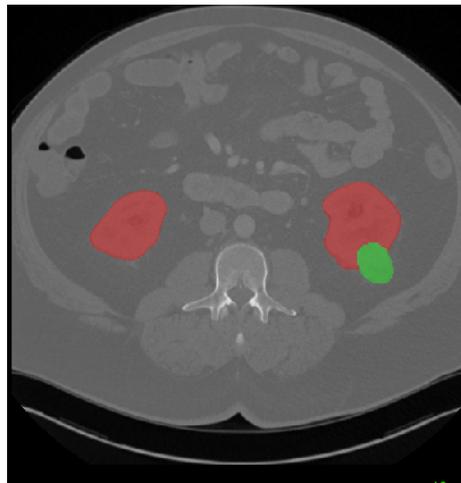


Fig. 6. Example predicted mask case_00235

References

1. Kidney Cancer Statistics, <http://www.wcrf.org/dietandcancer/cancer-trends/kidney-cancer-statistics>. Last accessed 12 Sept. 2018
2. Cancer Diagnosis and Treatment Statistics, <http://www.cancerresearchuk.org/health-professional/cancer-statistics/diagnosis-and-treatment>. Last accessed 26 Oct. 2017
3. Alexander, K., and Uzzo, R.G.: The RENAL nephrometry score: a comprehensive standardized system for quantitating renal tumor size, location and depth. *The Journal of urology*. **182**(3), 844–853 (2009)
4. Vincenzo, F., et. al.: Preoperative aspects and dimensions used for an anatomical (PADUA) classification of renal tumours in patients who are candidates for nephron-sparing surgery. *European urology*. **56**(5), 786–793 (2009)
5. Ahmed, T., et. al.: Kid-Net: Convolution Networks for Kidney Vessels Segmentation from CT-Volumes. arXiv preprint arXiv:1806.06769. (2018)
6. Paszke, A., et.al.: ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. arXiv:1606.02147 (2016)
7. Salehi, S., et. al. : Tversky loss function for image segmentation using 3D fully convolutional deep networks. arXiv:1706.05721 (2017)