Semantic Segmentation of Kidney Tumor using Convolutional Neural Networks

Laura Daza, Catalina Gómez, and Pablo Arbeláez

Universidad de los Andes, Bogotá, Colombia
{la.daza10, c.gomez10, pa.arbelaez}@uniandes.edu.co

Abstract. We present a fully automatic method for segmentation of kidney tumors in CT volumetric data based on DeepLab v3+, the state-of-the-art model in semantic segmentation in natural images. We adapt the architecture to process medical data and reduce the computational complexity to allow training 3D models. We evaluate our approach on the Kidney Tumor Segmentation Challenge 2019 dataset, and define a validation set to experiment with the model's parameters. In our validation set, we report a dice score of XX for the kidney class and YY for the tumor class.

Keywords: Kidney tumor · Semantic Segmentation · CT scans.

1 Introduction

Kidney cancer is the 12th most common cancer worldwide, with over 400,000 new cases diagnosed in 2018 [1]. Although it is one of the most common cause of death from cancer, the survival rates are relatively high in developed countries, but low in lower income countries where cancer is often detected at later stages. The standard tests to diagnose kidney cancer are ultrasound scans, cystoscopy or CT scans of the urinary system (CT urogram) [2]. The treatment options depend on certain factors including the size of the tumor and where it is located in the kidney and if has spread to another part of the body. The automatic segmentation and localization of tumors in CT scans can contribute to the diagnosis, and thus, to select the most appropriate treatment during early stages of the tumor.

Deep learning models, such as Fully Convolutional Networks [3], have been widely used for segmentation tasks in the medical image domain. For instance, U-Net [4] is an extension of the FCN, designed for cell segmentation in light microscopy images. An important modification of U-Net is that the expansive (upsampling) path has a large number of feature channels (similar to the contracting path), and they merge high resolution features from the contracting path with the upsampled outputs to refine the final prediction.

The drawbacks of traditional segmentation methods include the reduction of the original resolution at the encoding phase and objects at different scales. DeepLab [5] addresses these challenges by replacing pooling operations with dilated convolutions. To capture multi-scale features, they apply several parallel atrous convolutions with different dilation rates, which they called Atrous Spatial Pyramid Pooling (ASPP). Atrous convolution generalizes the standard convolution operation allowing to adjust the fielter's field of view to incorporate multi-scale information. Furthermore, in their more recent version [6], they define the atrous separable convolution to reduce the computation complexity, based on the depthwise separable convolution (DSC). For the decoder, instead of using a one-step interpolation to recover the original size, they concatenate the upsampled features from the encoder with low-level features from the network backbone, followed by 3×3 convolutions to refine the features and a final bilinear upsampling.

In this paper, we present an adaptation of DeepLab to segment kidneys and tumors in multimodal and three-dimensional data. We extend the ideas developed for the inherent challenges of semantic segmentation in natural images, to accurately segment tumors at different scales within or protruding from the kidneys. In addition, to deal with the class imbalance created by the small tumor size compared to the kidneys and background, we propose a combination of two loss functions to penalize wrong and missed predictions. Our 3D model consists of a single network that can be trained end-to-end with less than 560k parameters.

2 Method

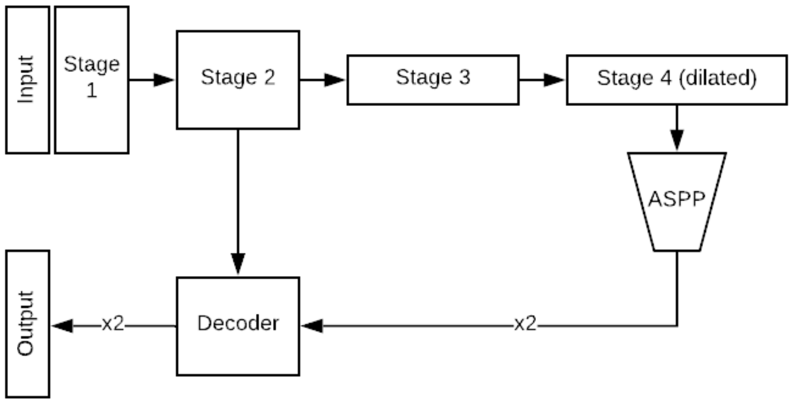

Fig. 1. Overview of the architecture. The method has an encoder to extract features from the input image and a decoder to reconstruct the high resolution segmentations.

Our method is based on DeepLab v3+ [6], the current state-of-the-art in semantic segmentation. The method is composed of an encoder to extract features form the image in different stages, and a decoder to recover the resolution lost during the previous step. We adapt the architecture to analyze medical images by extending it to 3D, which allows it to leverage on the rich information provided by CT scans. However, processing volumetric data is highly expensive in terms of memory and computational power. Therefore, we also reduce the size of the model by decreasing the number of layers and feature maps produced by each one. An overview of the architecture can be seen in figure 1.

2.1 Architecture

For the encoder we use an Xception [7] backbone with four stages. The first three stages are composed of two residual blocks with N depthwise separable convolutions, and are separated using convolutions with $stride = 2$. The last stage has two dilated separable convolutions to increase the receptive field without loosing resolution. After the final stage, we use an ASPP with dropout 0.5 to obtain the features that will be sent to the decoder. In the ASPP, the dilations used are $[1, 1, 2, 3]$ and every layer produces 32 feature maps.

The decoder receives the final output and features extracted in the second stage of the encoder. The former are upsampled to match the size of the latter, and then both outputs are concatenated and combined using two convolutional layers that produce 48 feature maps each, with dropouts 0.5 and 0.1, respectively. Finally, the output is upsampled back to the original resolution and a $1 \times 1 \times 1$ layer is used to produce the final segmentation.

Throughout the network, all the convolutional layers, except the one to produce the output, are followed by a batch normalization and a ReLU operation.

Input	Conv (8), 1x1x1
Stage 1	DSC (16) x 2 DSC (16) x 2, $s = 2$
Stage 2	DSC (32) x 2* DSC (32) x 3, stride 2
Stage 3	DSC (32) x 3 DSC (64) x 2
Stage 4	DSC (96) x 1, dilation 2 DSC (128) x 1, dilation 2*

Table 1. Encoder architecture. The values in parenthesis represent the number of feature maps produced. The blocks with * produce the outputs that will be sent to the decoder.

Fig. 2. Depthwise Separable Convolution (DSC) block configuration. If the block has stride $s > 1$, one DSC layer with $stride = s$ is included before the residual connection

2.2 Loss Functions

We use the Tversky loss [8], which is a generalization of Dice loss that weights the false positives (FP) and false negatives (FN). However, with this loss we can obtain a good recall, but a very low accuracy. For this reason, we also use the Focal loss [9] with $\gamma = 2$ to focus on examples that are hard to classify. This loss takes into account the imbalance between foreground and background voxels inherent to the problem of medical images segmentation. Our final loss is a weighted sum of both metrics.

2.3 Data Pre- and Post-processing

Following the procedure suggested by nnU-net [10], the images are normalized using the mean and standard deviation of all foreground pixels of the training dataset. Also, the range of intensities are limited between the 0.05 and 99.5 percentile of the foreground values. To reduce the memory usage during training, if the CT scans had more than 200 slices, the image was sliced along the z-axis making sure that we keep all the foreground pixels. For the final results, we refine the network's predictions applying morphological operations. We define a threshold to determine if the voxels of a predicted connected component should be assigned to the closest category.

2.4 Implementation Details

We interpolated the raw data provided by the challenge, which has different voxel spacing, by first calculating the median spacing of the entire dataset and then accommodating all the scans to match those values using trilinear interpolation. Afterwards, we normalized the images following the procedure explained in the previous section. With these images we pretraied a model from scratch using randomly extracted patches of size $48 \times 48 \times 48$ from the images, making sure that the central voxel has equal probabilities of belonging to the different categories. The batch size was set to 39 patches per worker and was collected from 13 patients. The model was trained for a maximum of 300 epochs or until the learning rate became lower than $1e^{-8}$. We used Adam optimizer with an initial learning rate of $1e^{-3}$ that was multiplied by 0.1 when the loss had not lowered for 20 epochs. We trained the model using 2 Titan Xp GPUs.

Starting from this weights, we trained a model on the interpolated data provided by the challenge using patches of size $44 \times 44 \times 44$. This change in size is made because the new images are smaller than the original ones, hence some images should be padded in order to maintain the size of the patches and this would introduce noise to the model.

In the inference stage, the patches are extracted uniformly and their outer voxels are discarded after the processing. This procedure is done to alleviate the noise introduced by the padded convolutions. Lastly, the whole volume is reconstructed using the segmented patches.

3 Experiments

3.1 Experimental Framework

Dataset We train the model on the Kidney Tumor Segmentation Challenge (KiTS) 2019 dataset, which contains CT scans of 210 kidney cancer patients for training and 90 for model evaluation [11]. Each patient has its corresponding semantic segmentation annotation with three labels: background, kidney and tumor. We train our method on a subset of the training set that has 147 patients and validate it on the remaining 63.

Evaluation We measure the performance of out method using standard Dice coefficient over each label, Recall and Precision. The results reported are calculated over our validation set.

3.2 Ablation Experiments

Architecture We test the effect of changing the size of the architecture by adding an extra layer to the final stage; increasing the number of channels produced by every layer of the ASPP module; and by altering the decoder to receive outputs from each stage of the encoder and combine them progressively using 1 or 2 layers. The results are shown in table 2. The results show that increasing the size of the overall architecture does not represent an increase in the accuracy.

	Dice		Recall		Precision	
	Kidney	Tumor	Kidney	Tumor	Kidney	Tumor
Bigger encoder	75.41	14.89	89.39	41.88	62.12	9.8
Bigger ASPP	70.7	12.64	83.10	34.21	56.6	8.16
Decoder every stage 1 convolution	78.83	16.81	93.06	40.67	66.39	11.36
Decoder every stage 2 convolutions	75.206	13.85	89.12	43.69	62.01	8.99
Proposed method	75.6	14.03	82.95	33.6	61.83	8.86

Table 2. Effect of changing the architecture.

Loss We test the effect of changing the weights of the loss function. Table 3 shows that the best overall Dice score is obtained when the Focal loss is multiplied by 0.5.

Postprocessing With our postprocessing, the results from the best method obtained improve by a large margin, as shown in table 4.

	Dice		Recall		Precision	
	Kidney	Tumor	Kidney	Tumor	Kidney	Tumor
Tversky + 5*Focal	86.41	23.55	89.28	41.55	77.01	16.08
Tversky + 2*Focal	87.25	24.97	87.56	35.66	78.30	17.43
Tversky + 0.5*Focal	86.72	24.43	89.68	41.40	77.48	16.84
Tversky + 0.2*Focal	86.41	25.49	88.42	41.97	76.98	17.69

Table 3. Effect of changing the loss function.

	Dice		Recall		Precision	
	Kidney	Tumor	Kidney	Tumor	Kidney	Tumor
Final score	87.23	38.41	84.26	41.06	78.99	29.21

Table 4. Best method after postprocessing.

References

[1] World Cancer Research Fund. *Kidney Cancer. How diet, nutrition and physical activity affect kidney cancer risk.* 2018. URL: https://www.wcrf.org/dietandcancer/kidney-cancer.

[2] CANCER RESEARCH UK. *Kidney Cancer.* 2018. URL: https://www.cancerresearchuk.org/about-cancer/kidney-cancer?_ga=2.15857462.1657942749.1564427608-1396970486.1564427608.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 3431–3440.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[5] Liang-Chieh Chen et al. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[6] Liang-Chieh Chen et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV).* 2018, pp. 801–818.

[7] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *CoRR* abs/1610.02357 (2016). arXiv: 1610.02357. URL: http://arxiv.org/abs/1610.02357.

[8] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks". In: *CoRR* abs/1706.05721 (2017). arXiv: 1706.05721. URL: http://arxiv.org/abs/1706.05721.

[9] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision.* 2017, pp. 2980–2988.

[10] Fabian Isensee et al. "nnU-Net: Breaking the Spell on Successful Medical Image Segmentation". In: *CoRR* abs/1904.08128 (2019). arXiv: `1904.08128`. URL: `http://arxiv.org/abs/1904.08128`.

[11] Nicholas Heller et al. "The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes". In: *arXiv preprint arXiv:1904.00445* (2019).