KiTS challenge: VNet with attention gates and deep supervision

Alzbeta Tureckova[1[0000−0002−5566−7393]], Tomas Turecek[1[0000−0001−8872−3278]], Zuzana Kominkova Oplatkova[1[0000−0001−8050−162X]], and Antonio Rodríguez-Sánchez[2]

[1] Faculty of Applied Informatics, Tomas Bata University in Zlin, nam. T.G.Masaryka 5555, 760 01 Zlin, Czech Republic
{tureckova,turecek,oplatkova}@utb.cz
ailab.fai.utb.cz

[2] Department of Computer Science, University of Innsbruck, Technikerstr. 21a, 6020 Innsbruck, Austria
Antonio.Rodriguez-Sanchez@uibk.ac.at
iis.uibk.ac.at

Abstract. This paper presents the 3D fully convolutional neural network extended by attention gates and deep supervision layers. The model is able to automatically segment the kidney and kidney-tumor from arterial phase abdominal computed tomography (CT) scans. It was trained on the dataset proposed by the Kidney Tumor Segmentation Challange 2019. The best solution reaches the dice score $96,43 \pm 1,06$ and $79,94 \pm 5,33$ for kidney and kidney-tumor labels, respectively. The implementation of the proposed methodology using PyTorch is publicly available at github.com/tureckova/Abdomen-CT-Image-Segmentation.

Keywords: Medical Image Segmentation · CNN · Attention Gates · Deep Supervision.

1 Introduction

Deep learning techniques, especially convolutional neural networks occupy the main research interest in the area of medical image segmentation nowadays and outperform other techniques usually by a large margin. A very popular convolution neural network architecture used in medical imaging is the encoder-decoder structure with the skip connections at each image resolution level. The basic principle for segmentation in 2D biomedical images was presented by [8] for the firs time. The network architecture was named U-Net. The U-Net traditionally uses the max-pooling to downsample the image in the encoder part and upsampling in the decoder part of the structure. The work of [6] extended the model for volumetric medical image segmentation and replaced the max-pooling and upsampling by convolution operations, creating a fully convolutional neural network named V-Net.

Kidney cancer is one of the ten most common cancers in human beings [1]. The goal of Kidney Tumor Segmentation Challenge[3] is to accelerate the development of reliable kidney and kidney-tumor semantic segmentation method. Automatic segmentation of kidney tumors is a challenging problem due to a hight morphological heterogeneity. Some papers dealing with the kidney and tumor segmentation exists in the literature, usually utilizing private clinical datasets. For example, recent work by [10] introduces a 3D fully convolutional neural network with pyramid pooling module for kidney-tumor segmentation. The average dice scores for kidney and renal tumor obtained in this work are equal to 93,1 and 80,2, respectively. The model was trained with the ROI (region of interest) containing the kidneys with tumor lesion. On the contrary, we present a methodology that processes the whole CT image; therefore, no ROI cropping before the training is needed.

2 Methodology

This section describes the proposed methodology in detail and is divided into five subsections. First describes the dataset, second deals with data preprocessing, third explains the model architecture, forth defines the training procedure, and finally, the last subsection comments the inference. Our methodology implemented using PyTorch is publicly available at github.com/tureckova/Abdomen-CT-Image-Segmentation.

2.1 KiTS Dataset

The dataset features a collection of multi-phase CT scans, segmentation masks, and comprehensive clinical outcomes for 300 patients who underwent nephrectomy for kidney tumors at the University of Minnesota Medical Center between 2010 and 2018 [3]. Seventy percent (210) of these patients were selected at random as the training set for the 2019 MICCAI KiTS Kidney Tumor Segmentation Challenge[3]and have been released publicly. We perform a five fold cross-validation during training: 42 images were used for validation and 168 images for training.

2.2 Data preprocessing

The models were trained with the patient images resampled to the median voxel spacing provided by challenge organizers. Beside we did following normalization:

- the dataset is normalized by clipping to the [0.5, 99.5] percentiles of the intensity values occurring within the segmentation masks,
- z-score normalization is done based on the mean and standard deviation of all intensity values occurring within the segmentation masks.

Because of memory restrictions, the model was trained on 3D image patches. We consider two different approaches:

[3] kits19.grand-challenge.org

Full Resolution - the original resolutions of images are used for the training and relatively small patches are chosen randomly during training. This way, the network has access to high resolution details, on the other hand, neglects context information.
Low Resolution - the patient image is downsampled by a factor of two until the median shape of the resampled data has less than four times the voxels that can be processed as an input patch. The patches are also chosen randomly during training. In this case, the model has more information about the context but lacks the high resolution details.

All the models were trained on the 11GB GPU with the batch size of two. The Table 1 shows the image shapes, training setups, and network topologies of trained models.

Table 1: An overview of image shapes, training setups, and network topologies of trained models.

	High Resolution	Low Resolution
num. images training	168	168
num. images validation	42	42
median patient shape	$511 \times 511 \times 136$	$247 \times 247 \times 127$
input patch size	$160 \times 160 \times 48$	$128 \times 128 \times 80$
num. downsampling per axis	5, 5, 3	5, 5, 4
batch size	2	2

2.3 Model architecture

Our architecture follows most closely the nnUNet [4] model design choices in the process of creating concrete encoder-decoder architecture. We use 30 feature maps in the highest layers (the number of feature maps doubles with each downsampling) and we downsample the image along each axis until the feature maps have size 8 or for a maximum of 5 times. The encoder part consists of context modules and decoder part is created by localization modules. Each module contain convolution layer, dropout layer, instance normalization layer and leakyReLU activation.

In addition to original encoder-decoder network architecture, we added attention gates from [7] in the top two model levels and appended the deep supervision layers presented by [5]. Both extensions are described in the next two subsections. Finally, we apply the instance normalization [9] and LeakyReLU activation function through the network. The structure of proposed network architecture is visualized in Figure 1.

Attention Gates Attention coefficients, $\alpha_{i,c} \in [0, 1]$ emphasizes salient image regions and significant features to preserve only relevant activations specific to

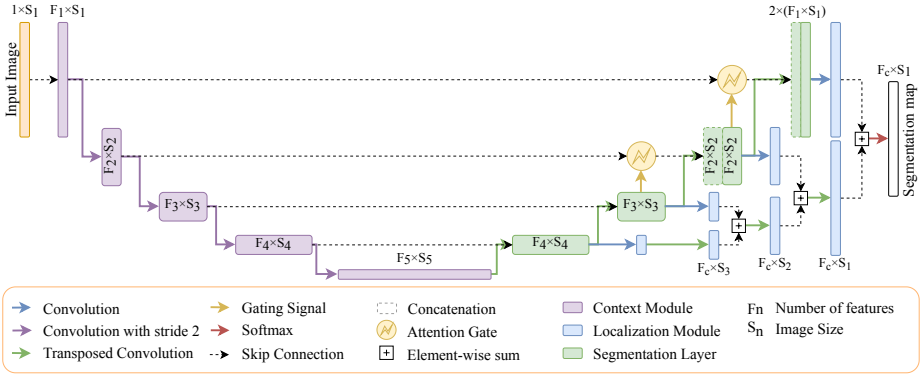

Fig. 1: A block diagram of the network architecture with attention gates and deep supervision.

the actual task. The output of AGs is the element-wise multiplication of input feature-maps and attention coefficients (1):

$$\hat{x}^l_{i,c} = x^l_{i,c} \cdot \alpha^l_{i,c} \tag{1}$$

where $x^l_{i,c}$ remarks pixel vector in layer l for the class c and $x^l_i \in \mathbb{R}^{F_l}$ where F_l corresponds to the number of feature-maps in layer l. Therefore, each AG learns to focus on a subset of target structures. The gating vector contains contextual information to reduce lower-level feature responses. The gate uses the additive attention. All the AG parameters can be trained with the standard back-propagation updates. For more information about the AG please refer to the original paper [7].

Deep Supervision The deep supervision [5] is the design where multiple segmentation maps are generated at different resolutions levels. The feature maps from each network level are transposed by $1 \times 1 \times 1$ convolutions to create secondary segmentation maps. These are then combined in the following way: First, the segmentation map with the lowest resolution is upsampled with bilinear interpolation to have the same size as the second-lowest resolution segmentation map. The element-wise sum of the two maps is then upsampled and added to the third-lowes segmentation map and so on until we reach the highest resolution level. For illustration please see Figure 1.

2.4 Training

All models were trained in the five-fold cross-validation. The network is trained with a combination of dice (3) and cross-entropy (4) loss function (2):

$$L_{total} = L_{dice} + L_{crossEntropy}, \tag{2}$$

$$L_{dice} = -\frac{2}{|C|}\sum_{c\in C}\frac{\sum_{i\in I} u_i^c v_i^k}{\sum_{i\in I} u_i^c + \sum_{i\in I} v_i^c}, \tag{3}$$

$$L_{crossEntropy} = -\sum_{c\in C}\sum_{i\in I}(v_i^c \log(u_i^k)), \tag{4}$$

where u is the softmax output of the network and v is a one hot encoding of the ground truth segmentation map. Both u and v have shape $I \times C$ with $i \in I$ being the number of pixels in the training patch/batch and $c \in C$ being the classes.

The dice loss is computed for each class and each sample in the batch and averaged over the batch and over all classes. We use the Adam optimizer with an initial learning rate 3×10^{-5} and l_2 weight decay 3×10^{-5} for all experiments. An epoch is defined as the iteration over all training images. Whenever the exponential moving average of the training loss does not improve within the last 30 epochs the learning rate is dropped by a factor of 0.2. We train till the learning rate drops below 10^{-6} or 1000 epochs are exceeded.

Gradient updates are computed by standard backpropagation using a small batch size of 2 due to the memorz restrictions. All weights were initially filled with values according to the method presented by [2], using a normal distribution. Gating parameters are initialized so that attention gates pass through feature vectors at all spatial locations.

Data Augmentation Training of the deep convolutional neural networks from limited training data suffers from overfitting. To minimize this problem, we apply a large variety of data augmentation techniques: random rotations, random scaling, random elastic deformations, gamma correction augmentation, and mirroring. All the augmentation techniques were applied on the fly during training. Data augmentation was realized with a framework which is publically available at `github.com/MIC-DKFZ/batchgenerators`.

The patches are generated randomly during the training, but we force that minimally one of the samples in a batch contains at least one foreground class to enhance the stability of the network training.

2.5 Inference

According to the training, inference of the final segmentation mask is also made patch-wise. The output accuracy is known to decrease towards the borders of the predicted image. Therefore, we chose to overlap the patches by half the patch size and also weigh voxels close to the center higher than those close to the border, when aggregating predictions across patches. The weights are generated, so the center position in patch equals to one, and boundary pixels are zero, in between the values are fading according to Gaussian distribution with sigma equals one eight of patch size. To further increase the stability, we use test time data augmentation by mirroring all patches along all valid axes.

Table 2: Metrics scores from five-fold cross-validation.

	Kidney			Tumor		
Network	Precision	Recall	Dice	Precision	Recall	Dice
Low Res.	94.79±0.78	95.07±1.42	94.63±0.88	77.85±3.43	78.51±2.79	74.12±2.66
Full Res.	96.01±0.71	96.15±1.19	95.93±0.54	78.77±3.60	79.72±2.57	75.43±1.59
Assembly	96.54±1.06	96.63±1.35	96.43±1.06	82.71±2.80	83.39±8.21	79.94±5.33

3 Results

The results of the proposed methodology are given in Table 2. We could see that the full-resolution model variant performs better than the low-resolution variant. The highest score was achieved by the prediction assembled from both trained models. The softmax outputs of both networks were averaged, and only then the final segmentation map is produced. This assembly method was also used for test submission. The best solution reaches the dice score 96.43 ± 1.06 and 79.94 ± 5.33 for kidney and kidney-tumor labels, respectively. The precision and recall scores are also good being around 96 for kidney and 83 for tumor. The lower score and higher variation in tumor label segmentation correspond to the greater inter-variability of the tumor sizes, positions, and morphology structures.

Figure 2 shows the visualization of attention maps obtained from low-resolution model. We could see that the attention gate focuses on the kidney as expected. It directs the attention of the model on the whole organ in the topmost level, while in the second level it seems to concentrate preferably on kidney borders.

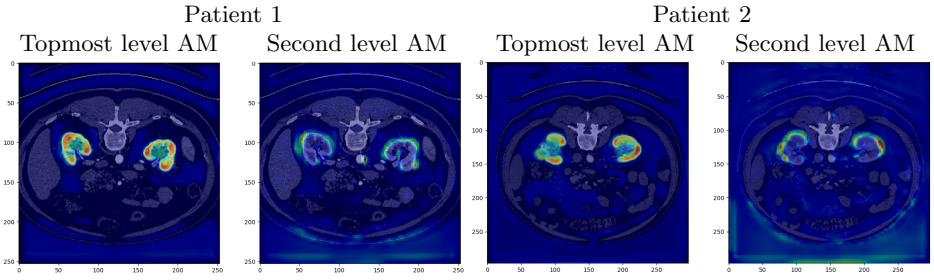

Fig. 2: Visualization of attention maps (AM) in low-resolution model on two randomly chosen patient images from the validation set. For each patient, the left picture shows the attention from the top most network layer, and the right picture shows the attention from the second network layer.

4 Conclusion

This paper describes the fully automatic methodology for kidney and kidney-tumor segmentation from computed tomography scans. The model is a deep fully convolutional network extended by attention gates and deep supervision. The attention gates help the model to highlight salient image regions and significant features and preserve only relevant activations specific to the actual task. To prove this, we show the activation maps from attention gates, where the activation obviously focus on the kidneys and its surroundings. The overall dice scores achieved by the best-proposed model are 96.43 ± 1.06 and 79.94 ± 5.33 for kidney and kidney-tumor label, respectively. Since the dataset is newly created, the results cannot be directly compared with the state-of-the-art. Nevertheless, the proposed method achieves higher dice scores than work [10] (achieving the dice score 93.1 for kidney, and 80.2 for tumor label), although our method does not require the cropping of the region with the kidney as [10] do.

Acknowledgments

This work was supported by Internal Grant Agency of Tomas Bata University under the Project no. IGA/CebiaTech/2019/002, by the Project no. LO1303 (MSMT-7778/2014), the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089. Further by resources of A.I. Lab (ailab.fai.utb.cz) and and IIS group at the University of Innsbruck (iis.uibk.ac.at). Moreover, access to the CERIT-SC computing and storage facilities provided by the CERIT-SC Center, presented under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CERIT Scientific Cloud LM2015085), is greatly appreciated.

References

1. Diagnosis, C., Statistics, T.: Stages — mesothelioma — cancer research uk (2017), `www.cancerresearchuk.org/health-professional/cancer-statistics/diagnosis-and-treatment`
2. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR **abs/1502.01852** (2015), `http://arxiv.org/abs/1502.01852`
3. Heller, N., Sathianathen, N., Kalapara, A., Walczak, E., Moore, K., Kaluzniak, H., Rosenberg, J., Blake, P., Rengel, Z., Oestreich, M., et al.: The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes. arXiv preprint arXiv:1904.00445 (2019)
4. Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P.F., Kohl, S., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S.J., Maier-Hein, K.H.: nnu-net: Self-adapting framework for u-net-based medical image segmentation. CoRR **abs/1809.10486** (2018), `http://arxiv.org/abs/1809.10486`
5. Kayalibay, B., Jensen, G., van der Smagt, P.: Cnn-based segmentation of medical imaging data. CoRR **abs/1701.03056** (2017), `http://arxiv.org/abs/1701.03056`

6. Milletari, F., Navab, N., Ahmadi, S.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. CoRR **abs/1606.04797** (2016), http://arxiv.org/abs/1606.04797
7. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M.C.H., Heinrich, M.P., Misawa, K., Mori, K., McDonagh, S.G., Hammerla, N.Y., Kainz, B., Glocker, B., Rueckert, D.: Attention u-net: Learning where to look for the pancreas. CoRR **abs/1804.03999** (2018), http://arxiv.org/abs/1804.03999
8. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
9. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. CoRR **abs/1607.08022** (2016), http://arxiv.org/abs/1607.08022
10. Yang, G., Li, G., Pan, T., Kong, Y., Wu, J., Shu, H., Luo, L., Dillenseger, J.L., Coatrieux, J.L., Tang, L., et al.: Automatic segmentation of kidney and renal tumor in ct images based on 3d fully convolutional neural network with pyramid pooling module. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3790–3795. IEEE (2018)