SEMANTIC SEGMENTATION OF THE PROSTATE IN 3D MRI

Christian Eschen s123656
s123656@student.dtu.dk

Technical University of Denmark

ABSTRACT

Segmentation of the prostate in 3D MRI is of high clinical demand since it has high diagnostic value. However, there is a lack of accurate prostate segmentation methods.
Semantic segmentation of the prostate in 3D MRI is a challenging task since there is a large variation of prostate shapes, there are different parameters used for acquisition of T2-weighted MRI of the prostate and the amount of provided training data is limited. The segmentation problem can be described as a pixel level boundary detection problem.
In this paper we investigate different convolutional neural network architectures for segmentation of the prostate. First, we find use of 3D context information necessary for high performance. We find that use of dense connected fully covolutional neural networks has high segmentation performance. This is caused by the fact that dense convolutional neural networks has deep supervision properties, thus the architecture mitigates vanishing gradients. We show state of the art on the Promise-12 challenge with a dice coefficient equal 0.898 on the validation set using 3D fully dense convolutional neural networks.

Index Terms— Semantic segmentation, fully convolutional neural networks, Promise-12 challenge

1. INTRODUCTION

Identification and segmentation of anatomic structures and pathology in medical images is useful for diagnosis of diseases, research, evaluation of treatment responds and monitoring diseases in medicine. Manual segmentaion of medical images like Magnetic Resonance Images (MRI) is a challenging and time consuming task, especially because MRI is usually acquired as multi-slices yielding a 3D volume. In practice there is inter-observer variance between different manually segmented images from different observers, since observers tend to over-segment or under-segment the images. Furthermore, there is intra observer variance since annotators tends to have difficulty reproducing the same segmented images. Ground truth medical image segmentation is thus special hard to obtain and expensive. Compared to other image segmentation tasks, segmentation of medical images is a challenging task. Therefore, there is a need for fast and accurate automatic image segmentation tools for medical use.
In recent years convolutional neural networks (CNN) has shown impressive results in different computer vision tasks including object detection [1] and semantic segmentation [2]. State of the art performance in computer vision semantic segmentation benchmarks like Pascal VOC [3] and Cityscapes [4] are governed by fully convolutional neural networks inspired by the architecture proposed by Long et al. in 2015 [2].
However, a slightly more relevant benchmark dataset for the task of segmenting MR images is the Promise-12 challenge dataset [5], which consists of T2weighted MR images and ground truth segmentations of the prostate. The images originates from different scanners with different protocols, however, all images are T2-weighted. Segmentation of the prostate is of high clinical demand since it has diagnostic value regarding benign as well as malignant prostatic hypertrophy and prostatitis. The shape and the size of the prostates in the MRI are varying, and thus segmentation of the prostate is a relative challenging task.
Semantic segmentation is a computer vision task, in which each pixel $\mathbf{x}$ is labeled as a category $\mathbf{y}$. State of the art semantic segmentation network architectures relies on [2] using fully convolutional neural networks. These so-called fully convolutional neural networks (FCNN) consists entirely of convolutional layers and these are quite efficient compared to sliding window approaches as in [6], in which a patch-window classifies the center pixel of the given patch window.

Regarding medical image segmentation state of the art methods relies on similar structures. Challenges in segmentation of medical imaging compared to segmentation in computer vision with natural images includes:

1. Ground truth data is incomplete and expensive to obtain. In order to obtain ground truth images, multiple annotations from the same annotator are obtained and annotations from different annotators are obtained as well. Since annotators either tend to over-segment (high sensitivity but low specificity) or under-segment (low sensitivity but high specificity), ground truth images can be modelled from imperfect annotations by modelling human errors [7]. Alternatively, it is also

possible to construct ground truth images using majority voting. However, access to this kind of data can be limited or the data might be imperfect.

2. Segmentation performance of convolutional neural networks tends to degrades when applied to a new database. This is caused by fact that MRI has countless imaging protocol as discussed [8]. In MRI one can adjust eg. echo time receiving an image with bright new contrast features. Manual annotation of each possible domain of image protocol is expensive and in practice unfeasible .

3. Convolutional neural networks is a framework, in which the input $\boldsymbol{x}$ is modelled to the output y using deterministic models. State of the art methods relies on pixel-wise classification, in which no prior information about the underlying shape is introduced. In order to introduce shape priors a generative neural networks framework with convolutional autoencoders can be used as proposed [9].

4. In MRI the region of interest often occupies only a small region compared to the full 3D volume. Convenient choice of cost function is thus important for successful training of convolutional neural networks. Different strategies has been suggested to overcome this problem, including weighted cross entropy loss function [10] and training with a dice loss [11]. Alternatively, one can train in batches with uniform distribution of the the presented classes.

5. Typical MR images consist of 3D volumes with dimension like 512x512x40 yielding millions of voxels per volumr. Compared to images in computer vision benchmark datasets like *VOC Pascal* medical images are significant larger. Since the computational cost for calculating convolutions is dependent of the spatial feature-map size it is expensive to train a neural network with large input size. Therefore, the best performing algorithms are using only sub volumes for training.

.

2. RELATED WORKS

Recent advances in computer vision is utilized in medical image analysis. In [10] the U-Net was proposed, which is a fully convolutional neural network used for segmentation of neurons in electron microscopy images. The architecture of U-Net was extended to 3D data in [12] known as U-Net 3D. The architecture from U-Net inspired the authors in [11] to propose the V-Net. The V-Net is similar to the U-Net but utilizes residual connections as in [13]. The V-Net implements a dice-coefficient loss layer in order to circumvent the class imbalance problem. Furthermore, in V-Net the input consist of large cropped sub volumes of size 128x128x64. The authors of V-net reports a dice coefficient of 82,39 on the test set in the Promise-12 challenge. In [14] the authors used 3D convolution with mixed residual connections and showed state of the art on the Promise-12 challenge in 2016 with a dice score 86,93. Furthermore, the authors of [14] claims that use of deep supervision [15] and residual skip connections [13] is important for convergence speed and prevention of overfitting. The authors of [14] claims that segmentation of the prostate in MRI-T2 weighted images is a boundary detection problem, thus it is sufficient to take inputs consisting of sub volumes with size equal 64x64x16. The authors of [14] is currently ranked as the fifth best performing algorithm (May 2018). The top fourth methods on the Promise-12 challenge reports dice coefficient of 89.18, (WHU-CS-sigma-RPI method 2), 87.21 (Philips DL-MBS), 87.19 (AutoDenseSeg) and 87.04 (WHU-CS-sigma-RPI), respectively (May 2018). The best performing algorithm (WHU-CS-sigma-RPI method 2) is using residual connections with attention modules with input size 96x96x16. However, the AutoDenseSeg and WHU-CS-sigma-RPI reports use of dense-nets with input size equal 64x64x16.

3. SEMANTIC SEGMENTATION

Semantic segmentation is the process of labelling each pixel in an image. As previous stated in Section 1 state of the art deep learning network employs the architecture from [2], in which FCNN are utilized. Regarding medical image segmentation the so-called U-Net architecture [10] is of certain interest, in which an encoder-decoder structure is used. In this convolutional neural network architecture pooling layers are used in the decoder part to incorporate translation in-variance, achieve abstraction from original representation and reduce feature size. In the decoder path un-pooling layers (transposed convolution) are used to resize the feature maps to original scale and concatenate the feature maps with feature maps of the same size from the encoding path. The U-Net model was used for cell segmentation of neurons on electron microscopy imaging. The U-Net architecture has been adapted for 3D image segmentation in [12]. Current state of the art neural networks for semantic segmentation uses either residual networks [13] or densely connected neural networks [16]. Recently, the authors of the so-called DeepLabv3+ [17] uses fully convolutional neural networks with encoder-decoder with atrous depthwise seperable convolution for semantic image segmenation. The authors of [17] reports new state of the art on both Pascal VOC [3] (April 2018) and City-scapes [4] (April 2018).

In this section we briefly introduce the concepts and building blocks relevant for this paper.

3.1. Residual block

The residual block proposed by [13] uses an identity mapping of input features by forward passing the input and summing it after a sequence of convolution, ReLU activation and Batch Normalization layers:

$$x_l = H_l(x_l) + x_{(l-1)} \tag{1}$$

In which H is a function with the operations convolution, ReLU and Batch Normalization repeated 2 or 3 times. x_l is the next layer and x_{l-1} is the previous layer. The invention of the residual block has made it possible to train very deep convolutional neural networks without problems regarding vanishing gradients.

3.2. Dense block

An alternative to the residual block has been proposed by [16], in which input features are concatenated instead of added. This alternative block is known as the dense block. In this way features might be reused and gradients can flow directly from previous layers. The number of output feature maps k is referred to as the growth rate, which is normally is increased linearly.

$$x_l = H_l([x_{l-1}, x_{l-2}, ..., x_0]) \tag{2}$$

Here $[...]$ represents concatenation of layers. The function H is now composition of Batch Normalization (BN), ReLU, convolution and dropout.

3.3. Tiramisu

Motivated by the work on the dense block [16] a convolutional neural network for semantic segmentaion has been proposed by [18]. In the Tiramisu convolutional neural network the dense block is employed in an encoder decoder fashion like U-Net [10]. In the encoding path of this network dense blocks are applied before max pooling, and features are forwarded before the dense blocks by skip connections with concatenation. In the decoding path transpose convolutional layers are utilized before the dense blocks. Furthermore, long skip connections of the same size are applied to concatenate feature maps from the encoding path to the decoding path. The extensive use of skip connections by concatenation is hypothesized to enforce feature reuse and enforce deep supervision for training. In the Tiramisu convolutional neural network there is two important hyperparameters. The first one is the growth rate, which is the number of output feature maps. The second hyperparameter is the numbers of layers pr. block. The numbers of layers per block is often increases in the encoding path and decreases in the decoding path. The authors of dense fully convolutional neural network (Tiramisu) [18] shows state of art on the CamVid benchmark dataset (February 2017) with use of fewer parameters compared to other competitors. The original implementation in Theano of the Tiramisu model can be found on Github [1].

3.4. DeepLabv3+

In [17] it has been shown that usage of atrous convolution and Xception blocks in an encoder decoder scheme it is possible to obtain state of art performance on CityScapes [4] (April 2018) and Pascal VOC [3] (April 2018).

Atrous convolution is an extension of ordinary convolution. In general one would like to increase the receptive field of the convolutional kernel, however, this has a computational cost. Atrous convolution, also known as dilated convolution, uses a atrous rate when applying the convolutional kernel. In this way the kernel "skips" pixels at atrous rate r when sampling the input image:

$$\mathbf{a}[i] = \sum_k \mathbf{x}[i + r \cdot k]\mathbf{w}[k] \tag{3}$$

In this $\mathbf{x}$ is the input, $\mathbf{y}$ is the output, r is the atrous rate, k is the index in the input and $\mathbf{w}$ is the kernel weights.

Another important feature of the [17] is the use of Xception blocks, which is the building blocks in the Xception network [19]. The Xception blocks is an extreme version of the Inception block introduced in [20]. The Xception network has outperformed the Inception network on ImageNet. The Xception block consist of depthwise separable convolutions with residual connections invented by [13]. The depthwise separable convolution is more computational efficient since it has fever multiplication operations compared to standard convolution. In [21] it is shown that the depthwise separable convolution is 8-9 times computational cheaper compared to standard convolution. The depthwise separable convolution has two components 1) First, it applies depthwise convolution, which applies the convolutional kernels separate through each input channel. 2) Next, a point wise convolution with kernel 1x1 is applied to perform a linear combination of the channels. Unfortunately, this operation is currently only implemented for 2D problems. The DeepLabv3+ utilizes depthwise separable convolutions and atrous convolution in a encoder decoder structure. A Keras implementation of DeepLabv3+ used for this project is found on Github [2].

4. METHODS

We choose to compare the performance of a U-Net 3D architecture, DeepLabv3+ 2D and Tiramisu 3D. The input for the models consist of different sizes, since we also seek to investigate impact of the input size. The comparison is conducted on the Promise-12 segmentation challenge.

[1] https://github.com/SimJeg/FC-DenseNet

[2] https://github.com/bonlime/keras-deeplab-v3-plus

4.1. Data augmentation and inference

For training sub volumes were uniform sampled from background and prostate with equal probability. The size of the sub volumes were different for each investigated model. The sub volume size for the U-Net model was 256x256x16, the sub volumes size for the DeepLabv3+ model was 256x256x13 and finally the sub volume sizes for the Tiramisu 3D was 64x64x16. The sub volumes were z-score normalized before feeding to the GPU. For inference a sliding overlapping strategy was used with sub-volumes with stride equal half of the input size. The average of the probabilities of the overlapping sub-volumes yielded the final prediction. All implementations was implemented in Keras.

4.2. U-Net architecture

Inspired by the work of [10] we employed a encoder decoder fully convolutional neural network.

We investigated a 3D convolutional neural network with input size 256x256x16. The idea of using large input size is that it gives less redundancy in training and the network can learn global context with respect to the input size. However, it has a drawback since we are only able to train 1.9 million parameters due to memory limitations.

Inspired by the work of [10] we employed a encoder decoder convolutional neural networks. In general the architecture consist of building blocks containing the following operations in the following order; Batch Normalization, 3D convolution with 3x3x3 kernels, ReLU activation, dropout with $p = 0.2$, 3D convolution with kernel 3x3x3, ReLU activation and a max pooling layer. The network has three downsampling blocks and three upsampling blocks. Weight decay was utilized at each convolutional layer with w equal 0.0001. All convolution was performed with padding. The model is presented in Figure 1. In order to train the network a binary cross entropy was utilized and the Adam optimizer was chosen with learning rate equal 0.001 and step decay per epoch equal 0.0001. At each iteration random crops of size 256x256x16 was sampled with equal probability from each class. The batch size was 14 on 2 Titan X GPU's (7 at each GPU).

4.3. DeepLabv3+ architecture

Motivated by the impressive performance of DeepLabv3+ [17] on CityScapes and VOC Pascal we experimented with the DeepLabv3+ architecture on the Promise-12 dataset. Unfortunately, Tensorflow does currently not support 3D seperable convolution, and thus we employed 2D network architecture using the z-stack axis information as input features. The input for the DeepLabv3+ architecture is 256x256x13, in which 256x256 represents rows and columns, and 13 represents slices stacked as input features. For each iteration sub volumes of size 256x256x13 was uniform random sampled from each classes. The architecture in the Deeplabv3+ consist of a encoder-decoder structure. The decoder structure consist of an entry flow part, an middle flow part and a exit flow part similar to [19]. The implementation of the DeepLabv3+ follows the original implementation.

In the flow entry part two times convolution 2D 3x3, Batch Normalization and ReLU is implemented. The final part of the flow entry block consist of three times Xception blocks, in which the first two has convolution with stride 2, yielding a spatial dimension reduction with factor 4.

In the middle flow block 16 times Xception blocks are implemented each with atrous rate equal 2.

The exit flow part starts with two times Xception blocks with atrous rate equal 2 and 4, respectively. Next, convolution 2D, Batch Normalization and ReLU is implemented before 3 times depthwise seperable convoluton with atrous rate equal 12, 24 and 36, respectively. All feature maps from these depthwise seperable convolution with atrous rate 12, 24 and 36 are concatenated.

The decoding path starts with average pooling, convolution 2D, Batch Normalization followed by bilinear upsampling. The skip connection list is now concatenated to the upsampling part. Next, convolution 2D, Batch Normalization and bilinear upsampling is performed again. After the bilinear upsampling, convolution 2D, Batch Normalization and ReLU is utilized, and the input from the bilinear upsampling is concatenated to the current feature maps. Finally, two times depthwise seperable convoltion are implemented before convolution 2D 1x1 with sigmoid activation. The model is presented in Figure 2. All convolutions use padding and weight decay is implemented at each convolutional layers with $w = 0.0001$. To train the model a binary cross entropy loss is used with Adam as optimizer with learning rate equal 0.0001 and learning rate decay equal 0.5 for each 50 epoch. The training was carried out asynchronously on two Titan X with batch size equal 3 on each for 1000 epochs.

4.4. Tiramisu architecture

As mentioned in Section 2 the top scores on the Promise-12 leaderboard uses deep dense connected convolutional neural networks. In order to replicate similar results we employed a Tiramisu inspired 3D convolutional neural network. We experiment with different number of layers pr dense block with both 3 and 4 pooling layers. The growth rate described in section 3 is set to 16. The network starts with a 3D convolutional layer with 48 filters. The downsampling path consist of dense blocks each followed by max pooling. Next, the bottleneck dense block is implemented. The decoding path of the network consist of dense blocks followed by transpose convolution. Long skip connections from the encoding path to the decoding path by concatenating are performed. Finally, convolution with 1x1x1 is used before sigmoid activation. All convolutions use padding and weight decay is implemented

at each convolutional layers with $w = 0.0001$. To train the model a binary cross entropy loss is used with Adam as optimizer. The learning rate is equal 0.0001 with decay equal 0.0001. Training was carried out asynchronously on 4 times Titan X with batch size 2 on each GPU.

5. EXPERIMENTS

5.1. U-Net 3D architecture results

The binary cross entropy loss for the U-Net 3D architecture is presented in Figure 4.

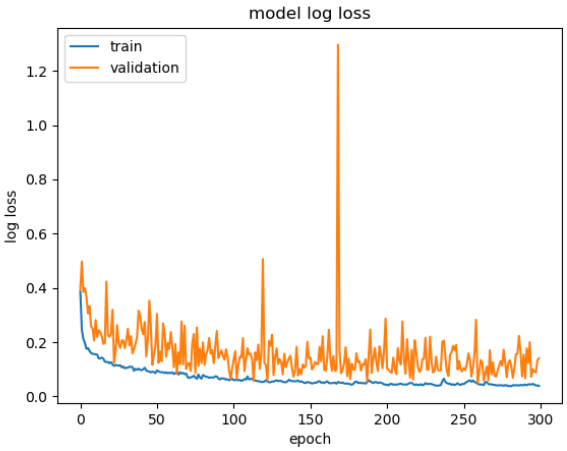


Fig. 4: Binary cross entropy log loss for U-Net 3D architecture.

The inference was accomplished by overlapping sliding sub volume strategy of size 256x256x16 with stride 128x128x8. The probability for each overlapping prediction was averaged. The U-Net achieved a dice coefficient of 0.72 on the validation set. The final prediction is presented in the first row in Figure 9

5.2. DeepLabv3+ 2D architecture results

The binary cross entropy loss for the DeepLabv3+ is presented in Figure 5. The inference was accomplished by overlapping sliding patches strategy of size 256x256 with stride 128x128. This was repeated for all slices in the volume. The probability for each overlapping prediction was averaged. The DeepLabv3+ achieved a dice coefficient of 0.79 on the validation set. The final prediction is presented in second row in Figure 9.

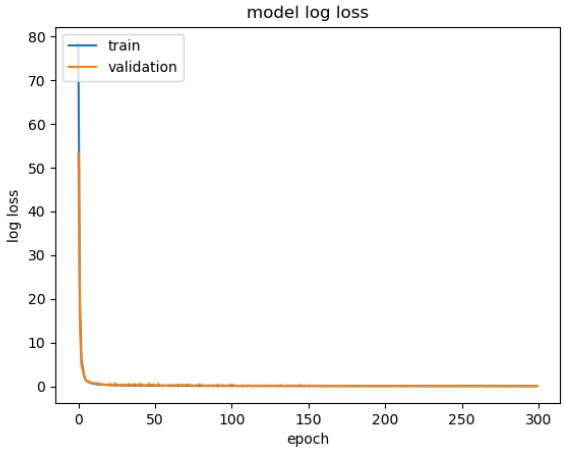


Fig. 5: Binary cross entropy log loss for DeepLabv3+

5.3. Tiramisu 3D architecture results

As mentioned in section 4.4 several experiments with the 3D Tiramisu model was carried out. The inference for all Tiramisu models was accomplished by overlapping sliding patches strategy of size 64x64x16 with stride 32x32x8. The probability for each overlapping prediction was averaged.
In Figure 6 the binary cross entropy loss is presented for the Tiramisu model with 4 pooling layers and 9 dense blocks with the following numbers of convolutional layers: $9, 10, 12, 15, 17, 20, 17, 15, 12$. This model is denoted as "Tiramisu3D p4a".
In Figure 7 the binary cross entropy is presented for the Tiramisu 3D with 3 pooling layers and 7 dense blocks with the 7 dense blocks with the following numbers of convolutional layers $7, 8, 9, 12, 15, 16, 17$. This model is denoted as "Tiramisu3D p3a".
In Figure the binary cross entropy is presented for the Tiramisu 3D with 3 pooling layers and 7 dense blocks with the 7 dense blocks with the following numbers of convolutional layers $12, 12, 12, 12, 12, 12, 12$. This model is denoted as "Tiramisu3D p4b".

The final prediction using model *Tiramisu3D p4 a* is presented in the third row in Figure 9. This prediction yielded an validation dice coefficient of 0.898.

A comparison between the model is presented in Tabel 1

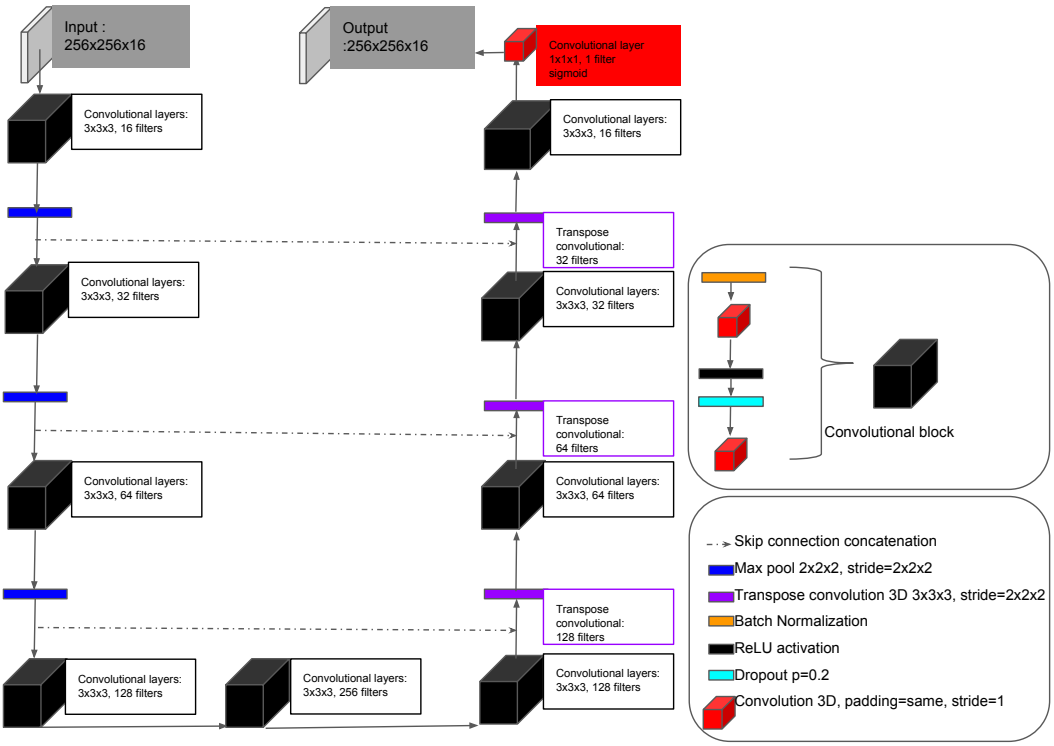


Fig. 1: Architecture for the U-Net 3D

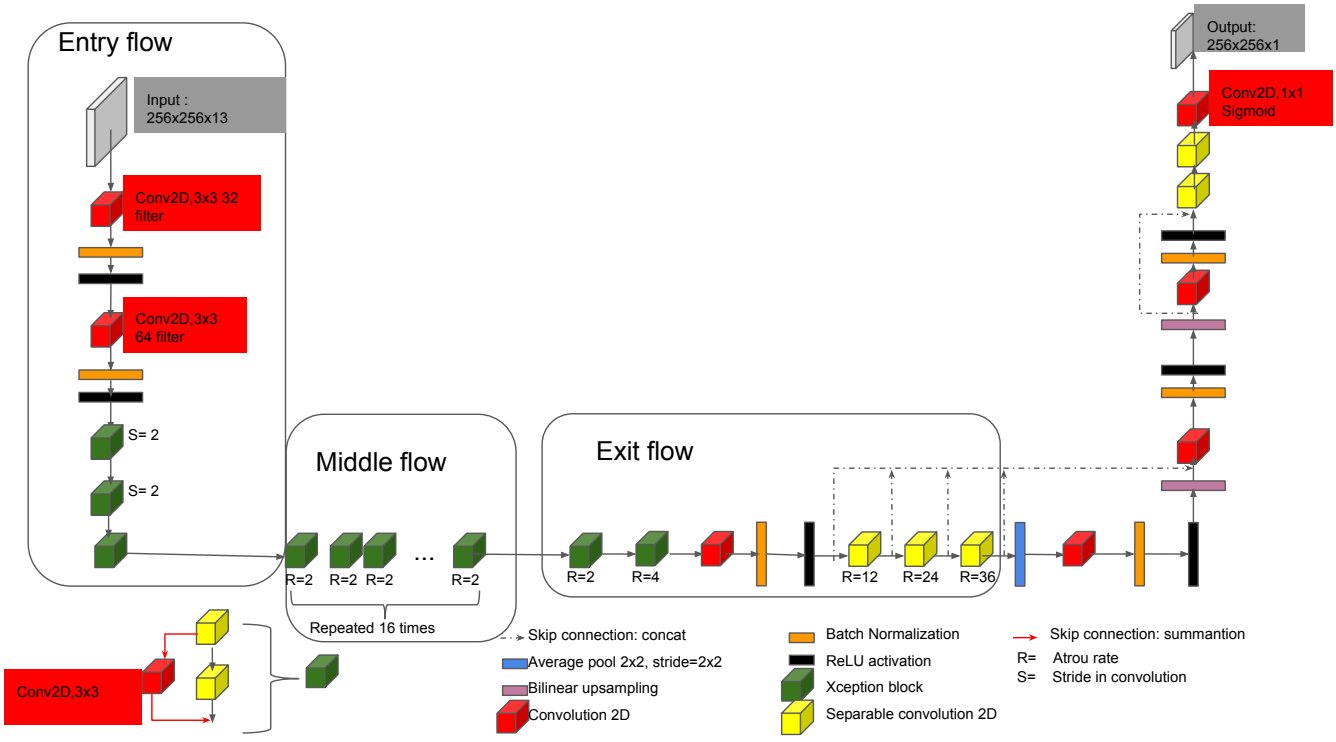


Fig. 2: Architecture for DeepLabv3+

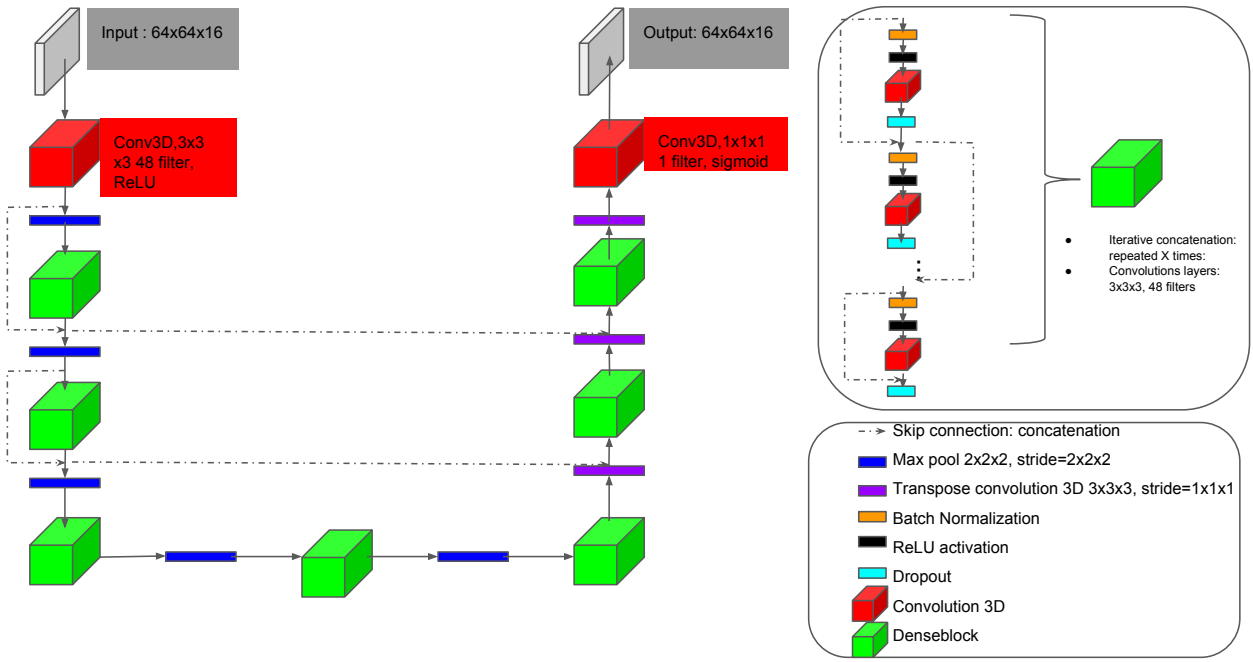


Fig. 3: Architecture for Tiramisu 3D

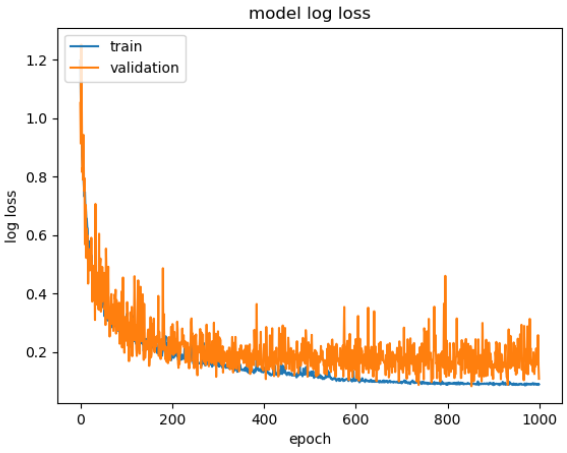


Fig. 6: Binary cross entropy log loss for Tiramisu 3D

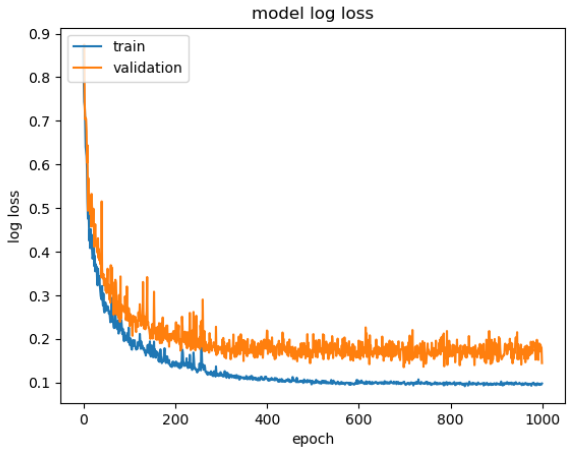


Fig. 7: Binary cross entropy log loss for Tiramisu 3D with 3 pooling layers and growthing and number of layers:

6. DISCUSSION

The U-Net model with larger input size converges faster since it has lesser redundancy, but it lack in performance with an dice score on 0.722. However, as compuational resources growth we will expect that whole volume input will be preferred in the future.

The DeepLabv3+ 2D yielded a validation dice score on 0.793 indicating a lot of information from the 3D context is needed to gain full contextual information. As mentioned be-

Table 1: Comparison of the performance of different model architectures

Model	U-Net 3D	DeepLabv3+	Tiramisu 3D p4a	Tiramisu 3D p3a	Tiramisu 3D p3b
Parameters	1,605,381	41,253,023	49,382,353	22,692,289	23,487,169
Validation dice coefficient	0.722	0.793	0.867	0.898	0.898

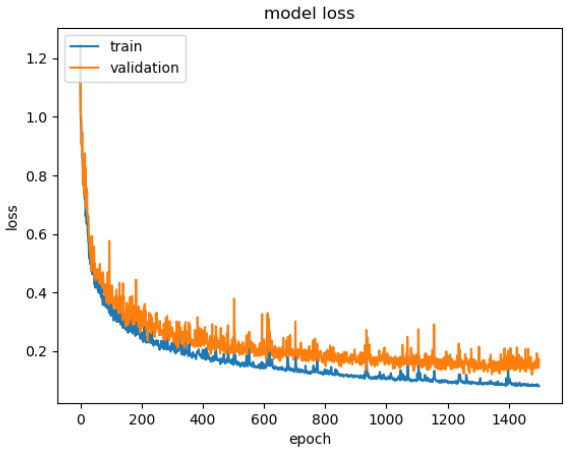


Fig. 8: Binary cross entropy log loss for Tiramisu 3D with 3 pooling layers and 12 number of layers per block

fore the depthwise seperable convolution is not implemented in deep learning frameworks like Tensorflow. Future work will reveal the performance of the DeepLabv3+ in 3D.

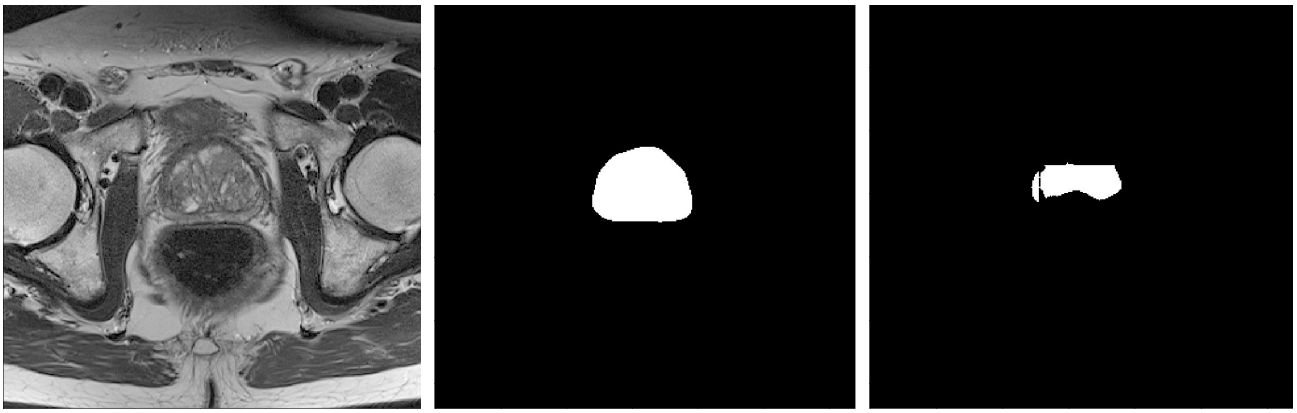
For training of the Tiramisu 3D we find that use of very small batch sizes (4-8) is difficult since the distribution of the classes is imbalanced. Sampling of all classes with equal probabilities mitigates this problem We find that the use of dense blocks in the Tiramisu 3D model has deep supervision properties, which is important for training deep convolutional neural networks in 3D since it helps the flow of gradients. Furthermore, we investigate different Tiramisu architectures. We find that with the given input we only need to pool 3 times, and that the number of layers pr dense block does not play an important rule. One can either increase the number of layers per dense blocks through the network or the number of layers can remain constant.

By comparing the different models in the results section we see that the best performing algorithm is the Tiramisu 3D p3a and the Tiramisu 3D p3b architectures both with an validation dice score equal 0.898. This suggest that in order to keep the model sufficient deep we need to train with sub volumes of relative small sizes (64x64x16).

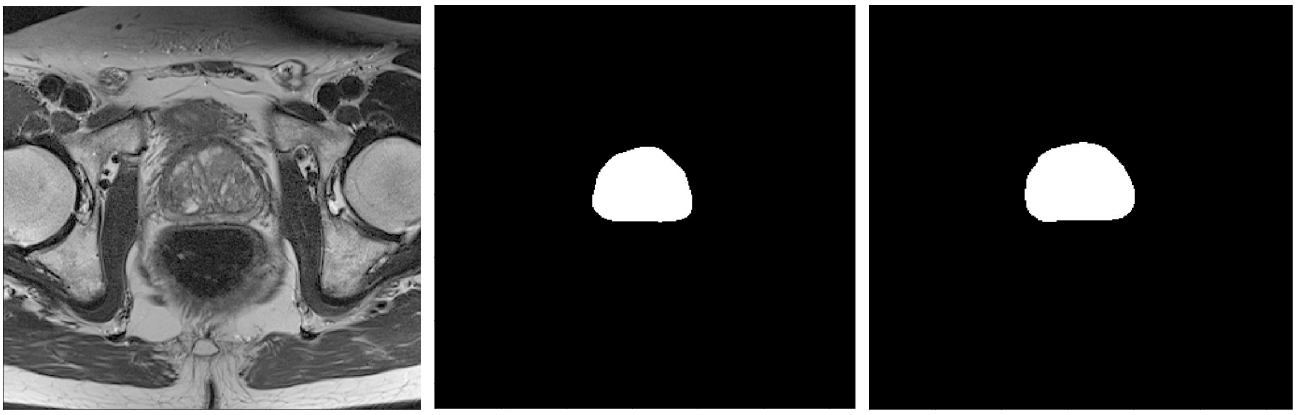
We assume that the network architecture of Tiramisu 3D can easily extent to other biomedical 3D segmentation challenges, in which the segmentation challenge can be described as a boundary detection problem.

7. CONCLUSION

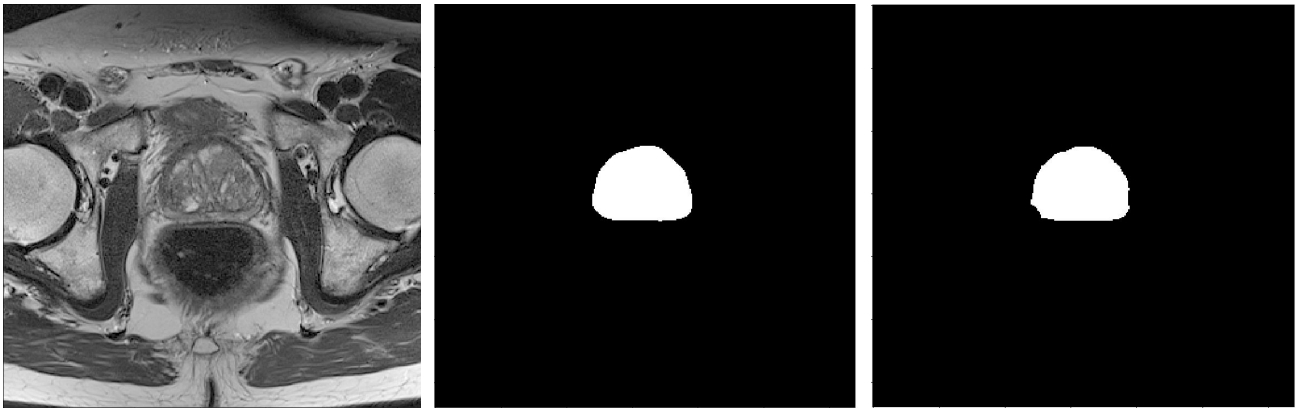
We present a new architecture for semantic segmentation of the prostate in 3D MRI T2-weighted images. We evaluated the performance of a U-Net 3D architecture, DeepLabv3v+ 2D architecture and several Tiramisu 3D architectures. We propose the Tiramisu 3D p3a architecture which shows state of the art on the Promise-12 challenge dataset. Futhermore, we have investigated the impact of input size, and found that small sub volumes of size eg. 64x64x16 is necessary for training a sufficient deep network architecture with the current computational resources. Furthermore, we have shown the performance of the DeepLabv3+ architecture in 2D. Future work will explore the DeepLabv3+ architecture in 3D when depthwise separable convolution is implemented in 3D Tensorflow.



From left to right: center slice of MRI scan, ground truth image and U-Net prediction



From left to right: center slice of MRI scan, ground truth image and Deeplabv3+ prediction



From left to right: center slice of MRI scan, ground truth image and Tiramisu prediction

Fig. 9: Model predictions on the validation set

8. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] 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.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[5] Geert Litjens, Robert Toth, Wendy van de Ven, Caroline Hoeks, Sjoerd Kerkstra, Bram van Ginneken, Graham Vincent, Gwenael Guillard, Neil Birbeck, Jindang Zhang, et al., "Evaluation of prostate segmentation algorithms for mri: the promise12 challenge," *Medical image analysis*, vol. 18, no. 2, pp. 359–373, 2014.

[6] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[7] Simon K Warfield, Kelly H Zou, and William M Wells, "Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation," *IEEE transactions on medical imaging*, vol. 23, no. 7, pp. 903–921, 2004.

[8] Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, et al., "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 597–609.

[9] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A Cook, Antonio de Marvao, Timothy Dawes, Declan P ORegan, et al., "Anatomically constrained neural networks (acnns): Application to cardiac image enhancement and segmentation," *IEEE transactions on medical imaging*, vol. 37, no. 2, pp. 384–395, 2018.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[11] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 565–571.

[12] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 424–432.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] Lequan Yu, Xin Yang, Hao Chen, Jing Qin, and Pheng-Ann Heng, "Volumetric convnets with mixed residual connections for automated prostate segmentation from 3d mr images.," in *AAAI*, 2017, pp. 66–72.

[15] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[16] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, vol. 1, p. 3.

[17] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv preprint arXiv:1802.02611*, 2018.

[18] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1175–1183.

[19] François Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, 2016.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al., "Going deeper with convolutions," Cvpr, 2015.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.