Evaluation of multislice inputs to convolutional neural networks for medical image segmentation

Minh H. Vu
Department of Radiation Sciences, Umeå University, Umeå, Sweden

Guus Grimbergen
Department of Biomedical Engineering, Eindhoven University of Technology, Eindhoven 5612 AZ, the Netherlands

Tufve Nyholm and Tommy Löfstedt[a)]
Department of Radiation Sciences, Umeå University, Umeå, Sweden

(Received 3 February 2020; revised 9 June 2020; accepted for publication 7 July 2020; published 10 November 2020)

Purpose: When using convolutional neural networks (CNNs) for segmentation of organs and lesions in medical images, the conventional approach is to work with inputs and outputs either as single slice [two-dimensional (2D)] or whole volumes [three-dimensional (3D)]. One common alternative, in this study denoted as pseudo-3D, is to use a stack of adjacent slices as input and produce a prediction for at least the central slice. This approach gives the network the possibility to capture 3D spatial information, with only a minor additional computational cost.
Methods: In this study, we systematically evaluate the segmentation performance and computational costs of this pseudo-3D approach as a function of the number of input slices, and compare the results to conventional end-to-end 2D and 3D CNNs, and to triplanar orthogonal 2D CNNs. The standard pseudo-3D method regards the neighboring slices as multiple input image channels. We additionally design and evaluate a novel, simple approach where the input stack is a volumetric input that is repeatably convolved in 3D to obtain a 2D feature map. This 2D map is in turn fed into a standard 2D network. We conducted experiments using two different CNN backbone architectures and on eight diverse data sets covering different anatomical regions, imaging modalities, and segmentation tasks.
Results: We found that while both pseudo-3D methods can process a large number of slices at once and still be computationally much more efficient than fully 3D CNNs, a significant improvement over a regular 2D CNN was only observed with two of the eight data sets. triplanar networks had the poorest performance of all the evaluated models. An analysis of the structural properties of the segmentation masks revealed no relations to the segmentation performance with respect to the number of input slices. A post hoc rank sum test which combined all metrics and data sets yielded that only our newly proposed pseudo-3D method with an input size of 13 slices outperformed almost all methods.
Conclusion: In the general case, multislice inputs appear not to improve segmentation results over using 2D or 3D CNNs. For the particular case of 13 input slices, the proposed novel pseudo-3D method does appear to have a slight advantage across all data sets compared to all other methods evaluated in this work. [https://doi.org/10.1002/mp.14391]

Key words: convolutional neural network deep learning medical image segmentation multislice

1. INTRODUCTION

Segmentation of organs and pathologies are common activities for radiologists and routine work for radiation oncologists. Nowadays, the manual annotation of such regions of interest is aided by various software toolkits for image enhancement, automated contouring, and structure analysis in all fields on image-guided radiotherapy.[1–3] Over the recent years, deep learning (DL) has emerged as a very powerful concept in the field of medical image analysis. The ability to train complex neural networks by example to independently perform a vast spectrum of annotation tasks has proven itself a promising method to produce segmentations of organs and lesions with expert-level accuracy.[4,5]

For both organ segmentation and lesion segmentation, the most common DL model is the convolutional neural network (CNN). Whereas the classic approach of segmenting 3D medical volumes by CNNs consists of training on and predicting the individual 2D slices independently, the interest has shifted in recent years toward full 3D convolutions in vo1umetric neural networks.[5–9] Volumetric convolution kernels have the advantage of taking interslice context into account, thus preserving more of the spatial information than what is possible when using 2D convolutions within slices. However, volumetric operations require a much larger amount of computational resources. For medical image applications, the lack of sufficient Graphical Processing Unit (GPU) memory to fit entire volumes at once requires in

almost all cases a patch-based approach, reduced input sizes, and/or small batch sizes and therefore longer training times.

1.A. Related work

In terms of fully connected, end-to-end 3D networks, studies often attempt to compensate for the small patch size that can maximally fit into the GPU memory at once by creating more efficient architectures or utilizing postprocessing methods. The original U-Net by Ronneberger et al. (2015),[10] an architecture which was, at that time, and still is, a popular and powerful network for semantic medical image segmentation, was first reintroduced as a 3D variant by Çiçek et al. (2016).[8] The 3D U-Net was used by Vu et al. (2019a,b)[11,12] in a cascaded approach where a first coarse prediction was used to generate a candidate region in which a second, finer-grained prediction was performed; this proved to be an effective way of reducing the amount of input data for the final prediction. V-Net by Milletari et al. (2016)[7] extended the network of Çiçek et al. (2016)[8] by adding residual connections to the 3D U-Net.

Li et al. (2017)[13] reduced the computational cost required for a fully connected 3D CNN by replacing the deconvolution steps in the upsampling phase with dilated convolutions to preserve the spatial resolution of the feature maps. VoxResNet[14] is a very deep residual network that was trained on small 3D patches. The resulting output probability map was combined with the original multimodal volumes into a second VoxResNet to obtain a more accurate output. A related approach from Yu et al. (2017b)[15] extended this architecture by implementing long residual connections between residual blocks, in addition to the short connections within the residual blocks. The same group proposed another densely connected architecture called DenseVoxNet,[16] where each layer had access to the feature maps of all its preceding layers, decreasing the number of parameters and possibly avoiding to learn redundant feature maps.

Lu et al. (2017)[17]used a graph cut model to refine the output of their coarse 3D CNN. A 3D network composed of two separate convolutional pathways, at low and high resolution, was introduced by Kamnitsas et al. (2017).[18] For improvement, the resulting segmentation was, in turn, postprocessed by a Conditional Random Field. A variant of this multiscale feature extraction during convolution was used by Lian et al. (2018),[19] who used this procedure in the encoding phase of their U-Net-like 3D CNN. Ren et al. (2018)[20] exploited the small size of regions of interest in the head and neck area (i.e., the optic nerves and chiasm) to build an interleaved combination of small-input, shallow CNNs trained at different scales and in different regions. Feng et al. (2019)[21] used a two-step procedure: A first 3D U-Net was used to localize thoracic organs in a substantially downsampled volume, and crop them to a bounding box around each organ. Then, individual 3D U-Nets were trained to segment each organ inside its subvolume at the original resolution. Another example of 3D convolutions applied only on a small region of interest is from the work of Anirudh et al. (2016),[22] who randomly sampled subvolumes in lung images for which the centroid pixel intensity was above a certain intensity threshold, to classify the subvolume as containing a lung nodule or not.

While these studies have shown that 3D CNNs are worth the effort, alternative approaches have been investigated to involve volumetric context to improve segmentation while avoiding 3D convolutions altogether. One of the more common methods, usually called 2.5D, is to use CNNs that combine triplanar 2D CNNs from intersecting orthogonal patches.[23–30] This can be a computationally efficient way to incorporate more 3D spatial information, and these studies all present promising results. However, this method is limited in the volumetric information it can encompass at once, since it employs only three orthogonal planes to provide spatial information for a single voxel.

We, therefore, investigate a method that uses a volumetric input but is still largely 2D based with only a small number of 3D operations. Instead of a method that takes a single 2D slice as input, and outputs the 2D segmentation of that slice, one can also incorporate neighboring slices to provide a 3D context to enhance segmentation performance. A common approach to this is to include neighboring slices to a central slice as multiple input image channels. Novikov et al. (2018)[31] included the preceding and succeeding axial slice for vertebrae and liver segmentation. Such a three-slice input was also used by Kitrungrotsakul et al. (2019b)[32] for the detection of mitotic cells in 4D data (spatial + temporal). This was a cascaded approach where a first detection step with a three-slice input produced results for these three slices. In the second step, they reduced the number of false positives where for each slice the time-frame before and after was included. In a deep CNN for liver segmentation, Han (2017)[33] used five neighboring slices. Ghavami et al. (2018)[34] compared incorporating three, five, and seven slices for prostate segmentation from ultrasound images. While their method produced promising segmentation results, no significant difference was found between these three input sizes. In a recent paper, Ganaye et al. (2019)[35] employed a seven-slice input producing an output for the three central slices, which the authors refer to as 2.5D. This model was used to evaluate a loss function that penalized anatomically unrealistic transitions between adjacent slices. The authors did not report a significant improvement between the baseline 2D and 2.5D models, but the 2.5D model did outperform in terms of Hausdorff Distance when the non-adjacency loss was employed.

The large number of studies that employ multislice inputs for semantic medical image segmentation implies that the general consensus is that such multislice methods can improve the results. However, to the authors' best knowledge, there have as of yet been no systemic evaluation whether this indeed holds true and under what circumstances.

1.B. Contributions

This work provides, for the first time, a systematic investigation of using multiple adjacent slices as input to predict the

central slice in that subset. The investigation is performed on the task of segmentation in medical images. We will henceforth refer to any method based on this principle as *pseudo-3D*. We compare the segmentation performance of a range of input multislice sizes ($d \in \{3,5,\ldots,13\}$) to conventional end-to-end 2D and fully 3D input-output CNNs. Since using triplanar, orthogonally intersecting 2D slices, is another popular approach to multislice inputs, this method is also included in the comparison, and shall be referred to as the *triplanar method*. For pseudo-3D, we employ the common approach from the literature where each neighboring slice is put as a separate channel in the input, and we will refer to this method as *the channel-based method*. Further, we introduce a second pseudo-3D method that appears to have not been proposed in the literature before. This novel pseudo-3D method consists of two main components: A transition block that transforms a d-slice volume input into a single-slice (i.e., 2D) feature map by using 3D convolutions, and this feature map is then followed by a standard 2D convolutional network, such as the U-Net[10] or the SegNet,[36] that produces the final segmentation labels. This method shall be referred to as *the proposed method*.

The main contributions of our work are:

1. We systematically compare the segmentation performance of 2D, pseudo-3D (with varying input size, d), 3D, and triplanar approaches.
2. We introduce a novel pseudo-3D method, using a transition block that transforms a multislice subvolume into a 2D feature map that can be processed by a 2D network. This method is compared to the channel-based pseudo-3D method.
3. We compare the computational efficiency of triplanar, fully 2D and 3D CNNs to the pseudo-3D methods in terms of graphical memory use, number of model parameters, floating point operations (FLOPs), training time, and prediction time.
4. We conduct all experiments on a diverse range of data sets, covering a broad range of data set sizes, imaging modalities, segmentation tasks, and body regions.

2. PROPOSED METHOD

The underlying concept of the pseudo-3D methods is similar to that of standard slice-by-slice predictions using 2D CNNs, but the input is now a subvolume with an odd number of slices, d, extracted from the whole volume with a total of D slices. The output of the model is compared to the ground truth of the central slice. If $d = 1$, the method is equivalent to a 2D CNN. Fig. 1 shows an illustration of the proposed method. In this study, the number of slices in the input subvolume ranged from $d = 3$ to $d = 13$. In order to isolate the contribution of using multislice inputs, this work did not include multislice outputs — where the multiple outputs for each slice are usually aggregated using, for example, means or medians.

Let the input volume be of width W, height H, depth d, and have C *channels*. A common way of utilizing depth

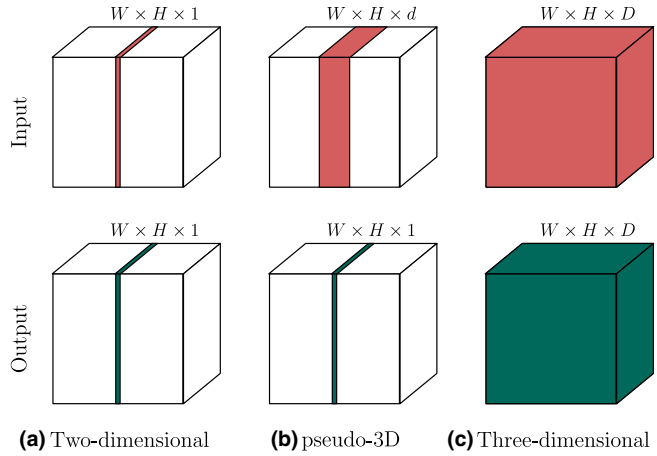


FIG. 1. A comparison of 2D, pseudo-3D, and 3D approaches. With a 2D network, the volume is segmented with a single slice input and output. Pseudo-3D uses multiple adjacent slices as input to produce an output of the central slice from the input. 3D approaches take in the whole volume at once and return a prediction for the whole volume as well. In the figure, the W, H, and D are the original width, height, and depth of the input volume, respectively. [Color figure can be viewed at wileyonlinelibrary.com]

information to train with regard to the central slice is as follows: group the channel and depth dimension together as one, and consider the input to be of shape $W \times H \times (d{\cdot}C)$, that is with $d{\cdot}C$ channels. By incorporating the slices in the channel dimension, the multislice input can be processed by a regular 2D network. As was mentioned in Section 1.B, this method is denoted here as the *channel-based method*.

The channel-based method is compared to a novel pseudo-3D approach denoted *the proposed method*. Consider the input to be of shape $W \times H \times d \times C$. This is fed through a transition block with $L = \lfloor \frac{d}{2} \rfloor$ layers (where $\lfloor \cdot \rfloor$ is the floor function). In each layer, a 3D convolution with a kernel of size $3 \times 3 \times 3$ is applied to the volume within the image, after it has been padded in the width (W) and depth (H) dimensions, but not in the depth (d) dimension. Thus, after each layer in the transition block, the depth of the image is reduced by 2 slices, while the width and height stay the same size. After the final convolution, the depth dimension is removed. Hence, the shapes change as

$$\begin{aligned} W \times H \times d \times C &\rightarrow W \times H \times (d-2) \times C \\ &\rightarrow W \times H \times (d-4) \times C \\ &\rightarrow \cdots \\ &\rightarrow W \times H \times 3 \times C \\ &\rightarrow W \times H \times 1 \times C \\ &\rightarrow W \times H \times C. \end{aligned}$$

In both the proposed method and the channel-based method, the output layer of the network is the segmentation mask, with an output shape of $W \times H \times 1$. Hence, it produces a single segmentation slice, corresponding to the central slice of the input subvolume. Fig. 1 shows an illustration of this.

The network architectures that were evaluated in this work was the U-Net[10] and the SegNet,[36] two popular variants of

encoder–decoder architectures that have been successful in semantic medical image segmentation. An illustration of both pseudo-3D methods, with U-Net as the main network architecture, is given in Fig. 2. Another illustration of the networks with the SegNet backbone can be seen in Fig. 1 in the Supplementary Material.

We evaluate the two pseudo-3D methods for $d \in \{3, 5, 7, 9, 11, 13\}$, and compare them to the corresponding conventional end-to-end 2D and 3D networks, all with the U-Net or SegNet architectures. Additionally, we employed the triplanar CNN method by Prasoon et al. (2013).[23] For each 2D slice, separate CNNs are trained and their outputs fused to predicts the output of the single centroid. This yields a total of 15 different experiments for each data set (six input sizes for the two pseudo-3D methods, plus 2D and 3D methods, all with two network architectures, and the triplanar network with its distinctive architecture.) Apart from the segmentation performance, the computational cost is also evaluated across experiments in terms of the number of network parameters, the maximum

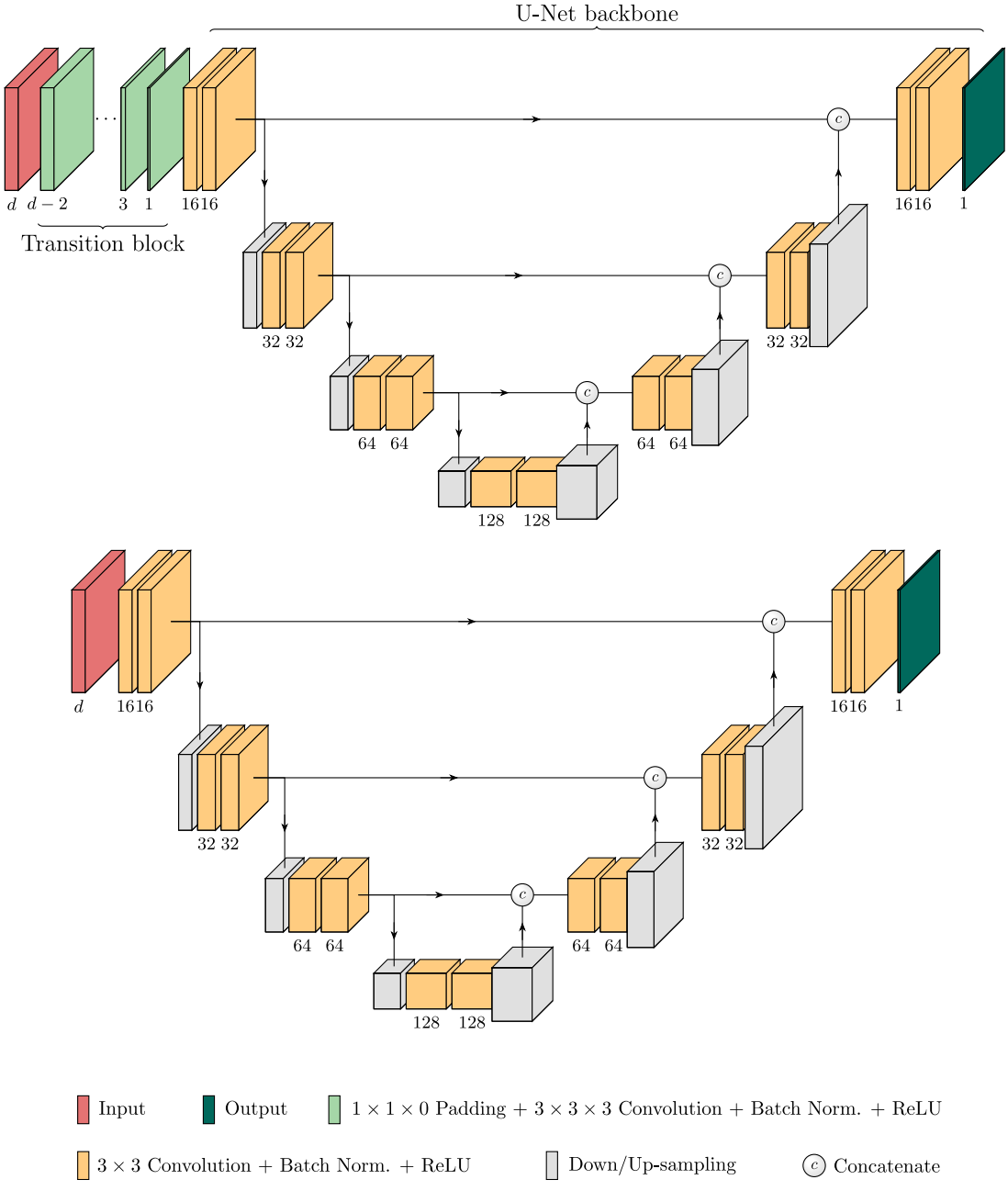


FIG. 2. The proposed methods illustrated with the U-Net backbone. The output is the prediction for the central slice of the input. The numbers in the transition block indicate the depth and in the backbone the number of filters. Top: The proposed method where the transition block uses 3D convolutions and 2D padding to iteratively reduce the input from depth d to 1, while the width and height remain. Bottom: The channel-based method, where neighboring slices are input as separate channels, and the input can be fed into a 2D CNN right away. [Color figure can be viewed at wileyonlinelibrary.com]

required amount of GPU memory, the number of FLOps, the training time per epoch, and the prediction time per sample.

3. EXPERIMENTS

We here present the data sets the experiments were conducted on, as well as the encompassing information and parameters used in the experiments.

3.A. Materials

To test the generalizing capabilities of the methods, we ran experiments on eight different data sets, covering a variety of modalities, data set sizes, segmentation tasks, and body areas. Six of the data sets are publicly available, as they were part of segmentation challenges. On top of those, we further used two in-house data sets collected at the University Hospital of Umeå, Umeå, Sweden.

3.A.1. Umeå Pelvic region organs

An in-house data set containing computed tomography (CT) images of the pelvis region from 1244 patients that underwent radiotherapy for prostate cancer at the University Hospital of Umeå, Umeå, Sweden. We denote this data set Umeå Pelvic Region Organs (U-PRO). The delineated structures include the prostate (in most cases annotated as the clinical or gross target volume) and some organs at risk, among them the bladder and rectum. The individual structure masks were merged into a single multilabel truth image, with pixel value 1 for the prostate, 2 for the bladder, and 3 for the rectum (see Fig. 3). Patients without the complete set of structures were excluded, resulting in a final data set containing 1148 patients.

3.A.2. Umeå head and neck database

An in-house data set contains CT images of the head and neck region of 110 patients. This data set comprises the patients from the University Hospital of Umeå, Umeå, Sweden, that participated in the ARTSCAN study.[37] We denote this data set Umeå head and neck database (U-HAND). For each CT image, manual annotations of the target volumes and various organs at risk were provided. The organ structures that were included with this data were the bilateral submandibular glands, bilateral parotid glands, larynx, and medulla oblongata (see Fig. 4). After removal of faulty CT volumes where the slice spacing changed within a volume and excluding patients in which not all of the six aforementioned structures were present, the final data set contained 73 patients.

3.A.3. Brain tumors in multimodal magnetic resonance imaging challenge 2019

The Brain Tumors in Multimodal Magnetic Resonance Imaging Challenge 2019 (BraTS19)[38,39] was part of the MICCAI 2019 conference. It contains multimodal preoperative magnetic resonance imaging (MRI) data of 285 patients with pathologically confirmed high grade glioma (HGG) (n = 210) or low grade glioma (LGG) (n = 75) from 19 different institutes. For each patient, T1-weighted (T1w), post-contrast T1-weighted (T1c), T2-weighted (T2w), and T2 fluid attenuated inversion recovery (FLAIR) scans were available, acquired with different protocols and various scanners at 3 T.

Manual segmentations were carried out by one to four raters and approved by neuroradiologists. The necrotic and non-enhancing tumor core, peritumoral edema, and contrast-enhancing tumor were assigned labels 1, 2, and 4 respectively (see Fig. 5). The images were co-registered to the same anatomical template, interpolated to a uniform voxel size and skull-stripped.

3.A.4. Kidney tumor segmentation challenge 2019

The data set for the Kidney Tumor Segmentation Challenge 2019 (KiTS19) challenge,[40] part of the MICCAI 2019 conference, contains preoperative CT data from 210

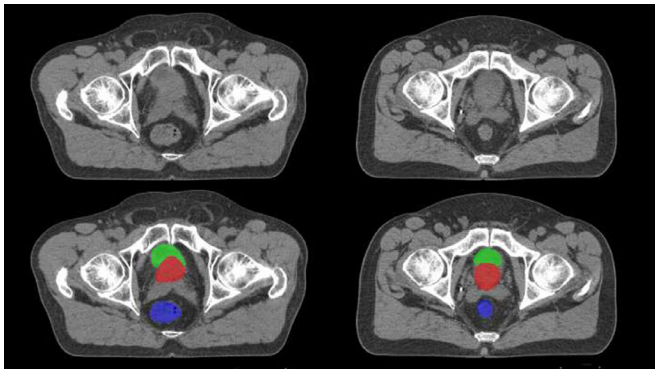


FIG. 3. Umeå Pelvic Region Organs data set. From top to bottom: images and ground truth images of the prostate (red), bladder (green), and rectum (blue). [Color figure can be viewed at wileyonlinelibrary.com]

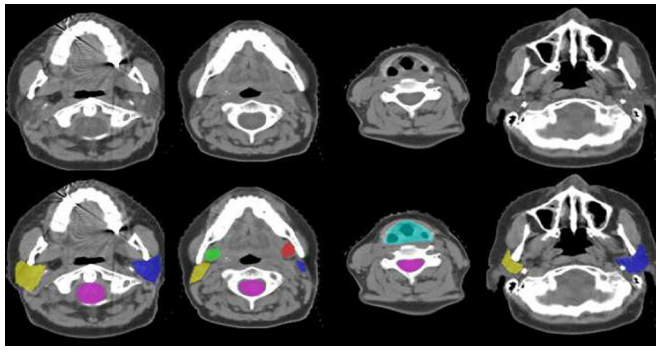


FIG. 4. Umeå Head and Neck Database. From top to bottom: images and ground truth images at different slices of the left and right submandibular glands (red and green), left and right parotid glands (dark blue and yellow), larynx (light blue), and medulla oblongata (pink). [Color figure can be viewed at wileyonlinelibrary.com]

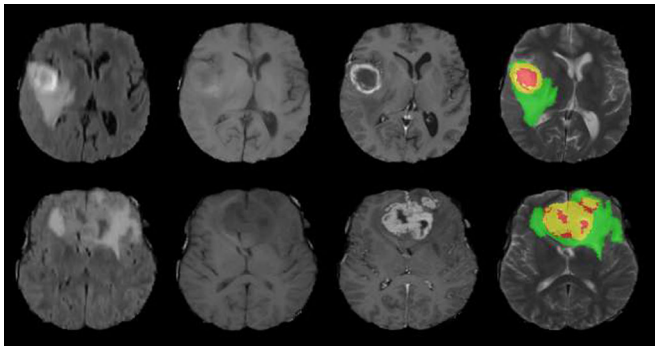


FIG. 5. Manual expert annotation of two patients with HGG from the Brain Tumors in Multimodal Magnetic Resonance Imaging Challenge 2019 data set. Shown are image patches with the tumor structures that are annotated in the different modalities. The image patches show (from left to right): (1) the whole tumor visible in fluid attenuated inversion recovery, (2-3) the enhancing and tumor structures visible in T1w and T1c, respectively, and (4) the final labels visible in T2w. The segmentations are combined to generate the final labels of the tumor structures: the necrotic and non-enhancing tumor core (NCR/NET — label 1, red), the peritumoral edema (ED — label 2, green) and the GD-enhancing tumor (ET — label 4, yellow). [Color figure can be viewed at wileyonlinelibrary.com]

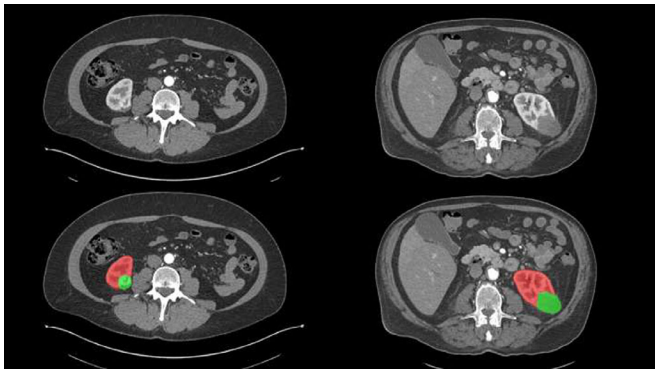


FIG. 6. Kidney Tumor Segmentation Challenge 2019 data set. From top to bottom: images and ground truth images of the kidney (red) and kidney tumor (green). [Color figure can be viewed at wileyonlinelibrary.com]

randomly selected kidney cancer patients that underwent radical nephrectomy at the University of Minnesota Medical Center between 2010 and 2018. Medical students annotated under supervision the contours of the whole kidney including any tumors and cysts (label 1), and contours of only the tumor component excluding all kidney tissue (label 2) (see Fig. 6). Afterward, voxels with a radiodensity of less than −30 HU were excluded from the kidney contours, as they were most likely perinephric fat.

3.A.5. Internet brain segmentation repository

The Internet Brain Segmentation Repository (IBSR18) data set[41] is a publicly available data set with 18 T1w MRI volumes, and is commonly used as a standard data set for tissue quantification and segmentation evaluation. Whole-brain segmentations of cerebrospinal fluid (CSF), gray matter, and white matter were included with their respective labels 1, 2, and 3 (see Fig. 7).

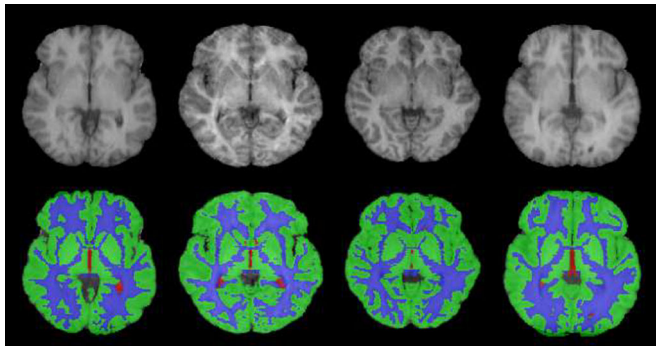


FIG. 7. Internet Brain Segmentation Repository data set. Axial slices of three patients with the ground truth of the cerebrospinal fluid (red), white matter (green), and gray matter (blue). [Color figure can be viewed at wileyonlinelibrary.com]

3.A.6. Heart segmentation decathlon

The Heart Segmentation Decathlon (D-HEART) data set was part of the Medical Segmentation Decathlon*, a large, open-source collection of data sets spanning many anatomies and segmentation tasks.[42] The D-HEART data set, originally part of the Left Artial Segmentation Challenge,[43] includes 30 gated cardiac MRI images of the entire cardiac phase. The data set is provided with expert annotations of the left atrium appendage, mitral plane, and portal vein end points (see Fig. 8).

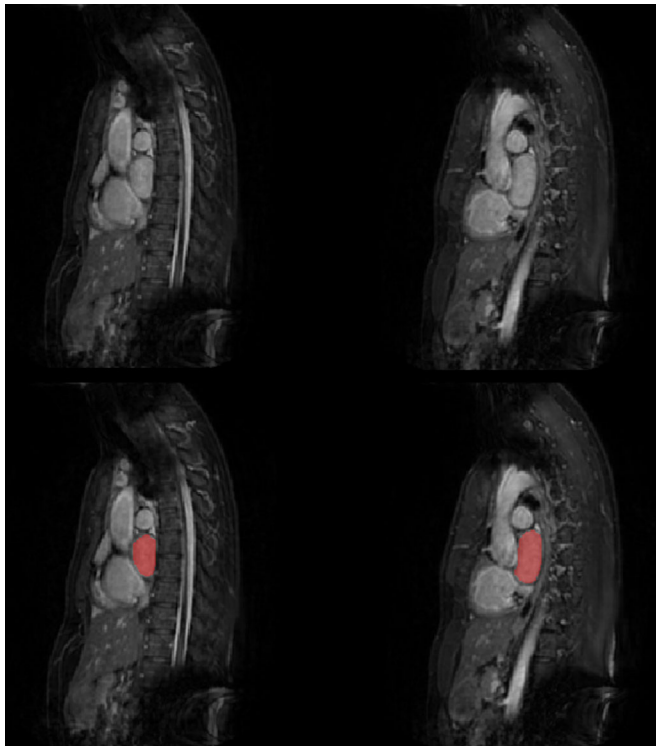


FIG. 8. Heart Segmentation Decathlon data set. Axial slices of two patients with the ground truth of the heart (red). [Color figure can be viewed at wileyonlinelibrary.com]

*http://medicaldecathlon.com/

3.A.7. Spleen segmentation decathlon

The Spleen Segmentation Decathlon (D-SPLEEN) data set[44] was also part of the Segmentation Decathlon. It consists of 61 CT scans in which the spleen was annotated semi-automatically, originally part of a study on splenic volume change due to chemotherapy in patients with liver metastases (see Fig. 9).

3.A.8. Hippocampus segmentation decathlon

Like the two previous data sets, the Hippocampus Segmentation Decathlon (D-HIPPO) is again provided in the Segmentation Decathlon. 195 T1w MRI volumes of healthy subjects and subjects with non-affective psychotic disorders obtained, and subsequently underwent manual segmentation of the hippocampus (see Fig. 10).

3.B. Preprocessing

Due to the diverse range of data sets, it must be ensured that the training data is as similar as possible across experiments in order to achieve a fair comparison.

The MRI data sets were N4ITK bias field corrected[45] and normalized to zero-mean and unit variance. The CT data sets were normalized by clipping each case to the range [−1000,2000], subtracting 500 and dividing by 1500. The BraTS19 volumes were cropped around the center to a resolution of 160 × 192 × 128, to increase processing speed. This last step was skipped for the IBSR18 data set because of the much smaller amount of data samples.

Most other data sets all had a varying matrix size, slice count, and resolution, so a preprocessing pipeline (see Fig. 11) was set up to transform these data sets to a uniform resolution and voxel size.

First, the data were resampled to an equal voxel size within the same set. The volumes were then zero-padded to the size of the single largest volume from that set after resampling. In order to increase processing speed and lower the memory consumption, the volumes were thereafter downsampled to uniform resolution across all samples in the same data set. An example of this method pipeline is shown in Fig. 11.

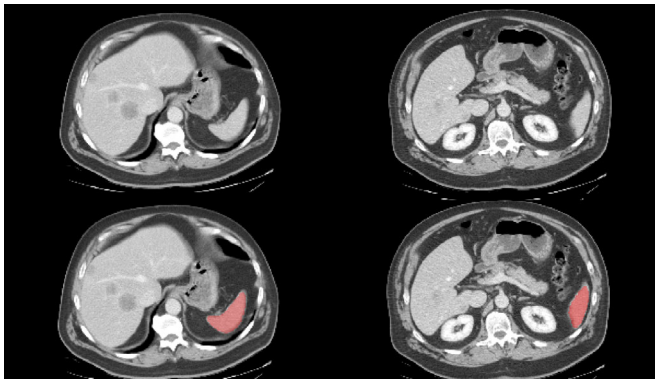


FIG. 9. Spleen Segmentation Decathlon data set Axial slices of two patients with the ground truth of the spleen (red). [Color figure can be viewed at wileyonlinelibrary.com]

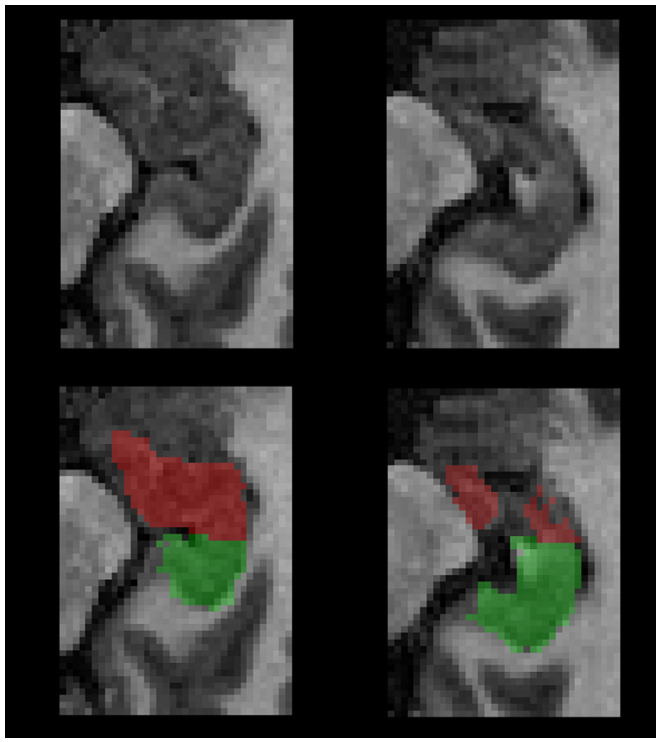


FIG. 10. Hippocampus Segmentation Decathlon data set. Axial slices of two patients with the ground truth of the hippocampus head (red) and body (green). [Color figure can be viewed at wileyonlinelibrary.com]

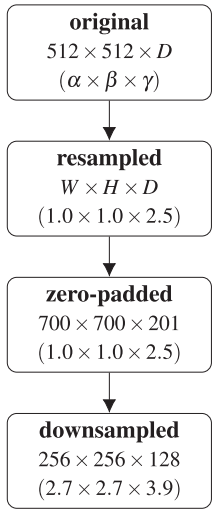


FIG. 11. Preprocessing pipeline as applied on the U-PRO data set. Given are the resolutions and in parentheses the voxel dimensions in mm. *W*, *H*, and *D* each denote that the volume shape is varied in width, height, and/or depth, respectively. α, β, and γ each denote that the voxel spacing is varied in width, height, and/or depth, respectively.

3.C. Training details

Our method was implemented in Keras 2.2.4[†] using TensorFlow 1.12.0[†] as the backend. The experiments were trained on either a desktop computer with an NVIDIA RTX 2080 Ti GPU, or the NVIDIA Tesla V100 GPUs from the High Performance Computer Center North (HPC2N) at Umeå University, Sweden. Depending on the model, the convergence speed, and the data set size, a single experiment took from minutes to multiple days to complete.

3.C.1. Experimental setup

For the 3D experiments, the BraTS19 data set was the only data where the whole volumes could be fed into the network at once because of constraints in GPU memory. For the other data sets, we resorted to a patch-based approach where the input size would be $256 \times 256 \times 32$, the largest size possible for our available hardware.

In all experiments, we employed the Adam optimizer[46] with an initial learning rate of $1 \cdot 10^{-4}$. If the validation loss did not improve after a certain number of epochs, we used a patience callback that dropped the learning rate by a factor of 0.2 and an early stopping callback that terminated the experiment. Because of the differences in data set sizes, these callbacks had to be determined from initial exploratory experiments for each separate data set to ensure experiments did not run for too long or too short. The patience callbacks were set to five epochs for the BraTS19, KiTS19, and U-PRO experiments, six epochs for the U-HAND data set, and ten epochs for the IBSR18 and Segmentation Decathlon (D-HEART, D-SPLEEN and D-HIPPO) data sets. The early stopping callbacks were set to 11 epochs for U-PRO data, 12 epochs for BraTS19 and KiTS19 data, 14 for U-HAND data, and 25 epochs for IBSR18 and the Segmentation Decathlon. The maximum number of epochs an experiment could run for, regardless of any changes in the validation loss, was set to 100 for the U-HAND and U-PRO data and 200 for the other data sets. Batch normalization and an L_2 norm regularization, with parameter $1 \cdot 10^{-5}$, were applied to all convolutional layers, both in the transition block and in the main network. The rectified linear unit (ReLU) function was used as the intermediate activate function. The activation function of the final layer was the softmax function. Each data set was split into 80% training and 20% test set, and with the training set, in turn, being split into 80% for training and 20% for validation.

As loss function, we employed a combination of the dice similarity coefficient (DSC) and categorical cross-entropy (CE). The DSC is typically defined as

$$D(U, V) = \frac{2 \cdot |U \cap V|}{|U| + |V|} \tag{1}$$

with U the output segmentation and V its ground truth. However, a differentiable version of Eq. (1), the so-called soft DSC, was used. The soft DSC is defined as

$$\mathcal{L}_{DSC}(u, v) = \frac{-2 \sum_i u_i v_i}{\sum_i u_i + \sum_i v_i + \epsilon}, \tag{2}$$

where for each label i, the u is the SOFTMAX output of the network and v is a one-hot encoding of the ground truth segmentation map. The ε is a small constant added to avoid division by zero.

The DSC is a good objective for segmentation, as it directly represents the degree of overlap between structures. However, for unbalanced data sets with small structures and where the vast majority of pixels are background, it may converge to poorly generalizing local minima, since misclassifying only a few pixels can lead to large deviations in DSC. A common way[47,48] to resolve this is to combine the DSC loss with the CE loss, defined as

$$\mathcal{L}_{CE}(u, v) = -\sum_i u_i \cdot \log(v_i), \tag{3}$$

and we did this as well. Hence, the final loss function was

$$\mathcal{L}(u, v) = \alpha \mathcal{L}_{dice}(u, v) + (1 - \alpha)\mathcal{L}_{CE}(u, v). \tag{4}$$

with α being the trade-off weighting factors for both losses. For our purposes, we set $\alpha = 0.5$.

3.C.2. Data augmentation

In order to artificially increase the data set size and to diversify the data, we employ various common methods for on-the-fly data augmentation: flipping along the horizontal axis, rotation within a range of -1 to 1 degrees, shear images within the range of -0.05 to 0.05, zoom with a factor between 0.9 and 1.1, and adding small elastic deformations as described in Simard et al. (2003).[49] The data augmentation implementation we used was based on Dong et al. (2017).[50]

The images in the KiTS19 data are asymmetric along the x-axis because of the liver; therefore, vertical flipping was not applied on that data set as it would result in anatomically unrealistic images (see Table I).

3.C.3. Evaluation

In order to ensure a fair comparison and to investigate the variability of the results within experiments, we used fivefold cross-validation in each experiment (except for the U-PRO). Due to its much larger size, the experiments on the U-PRO data set were run only once.

To compare the computational cost of our proposed models to the corresponding 2D and 3D, and triplanar CNN models, we extracted the number of trainable parameters, the maximum amount of GPU memory used, the number of FLOps, training time per epoch, and prediction time per sample.

For a complete evaluation of the segmentation performance, we employed several metrics for segmentation accuracy. First, the conventional DSC as defined in Eq. (1).

[†]https://keras.io
[‡]https://tensorflow.org

TABLE I. Data sets and augmentation techniques in this study.

Material/data set	BraTS19	KiTS19	IBSR18	U-HAND	U-PRO	D-HEART	D-SPLEEN	D-HIPPO
type	MRI	CT	MRI	CT	CT	MRI	CT	MRI
#modalities	4	1	1	1	1	1	1	1
#classes	3	2	3	6	3	1	1	2
#patients	335	210	18	73	1 148	20	41	263
Train	268	168	15-16	59	734	16	32-33	208-211
Val	67	42	2-3	14	184	4	8-9	52-55
Test	67	42	2-3	14	230	4	8-9	52-55
Original shape	240-240-155	512-512-*D*	256-128-256	*W-H-D*	512-512-*D*	320-320-*D*	512-512-*D*	*W-H-D*
Original voxel size (in mm)	1.0-1.0-1.0	α-β-γ	1.0-1.0-1.0	α-β-γ	α-β-γ	1.3-1.3-1.4	α-β-γ	1.0-1.0-1.0
Preprocessed shape	160-192-128	256-256-128	256-128-256	256-256-64	256-256-128	256-256-96	256-256-128	64-64-64
Preprocessed voxel size (in mm)	1.0-1.0-1.0	2.3-2.3-2.3	1.0-1.0-1.0	1.3-1.0-5.8	2.7-2.7-3.9	1.6-1.6-1.8	2.0-2.0-5.8	1.0-1.0-1.0
augmentation								
Flip left-right	✓	×	✓	✓	✓	✓	✓	✓
Elastic transform	✓	✓	✓	✓	✓	✓	✓	✓
Rotation	✓	✓	✓	✓	✓	✓	✓	✓
Shear	✓	✓	✓	✓	✓	✓	✓	✓
Xoom	✓	✓	✓	✓	✓	✓	✓	✓
training								
#epochs	200	200	200	100	100	200	200	200
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
Learning rate drop	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	$2 \cdot 10^{-1}$
Patience	5	5	10	6	5	10	10	10
Early-stopping	12	12	25	14	11	25	25	25

W, *H*, and *D* each denote that the volume shape is varied in width, height, and/or depth, respectively. α, β, and γ each denote that the voxel spacing is varied in width, height, and/or depth, respectively.

Second, the 95th percentile of the Hausdorff distance (HD95), where the 95th percentile of the Hausdorff distance (HD95) is defined as:

$$H(U, V) = \max\{d(U, V), d(V, U)\} \tag{5}$$

where

$$d(U, V) = \max_{u \in U} \min_{v \in V} \|u - v\|_2 \tag{6}$$

where $\|u - v\|_2$ is the ℓ_2 norm, or the Euclidean distance between points u and v on the boundaries of output segmentation U and ground truth V. In other words, the HD is the largest distance in the set of distances between the closest points of two objects. Common practice is to use the 95th percentile to avoid outliers from noisy boundaries.

Third, the average symmetric surface distance (ASSD) is also computed. This metric is closely related to the HD95, but instead of the 95th percentile, we compute the average closest distance.

Finally, we use the relative absolute volume difference (RAVD): the total volume difference of the segmentation to the reference is divided by the total volume of the reference. The result is multiplied by 100. This signed number is reported in the tables in the Results section, so one can recognize under-segmentations by negative values and over-segmentation by positive values. To obtain a single score value, the absolute value is taken. Note that the perfect value of 0 can also be obtained for a non-perfect segmentation, as long as the volume of that segmentation is equal to the volume of the reference.

3.D. Feature-based regression analysis

In an attempt to connect behaviour with d to differences in data set properties, a feature-based regression analysis was performed. We computed features of the structures (ground truth masks) that describe each mask's structural properties: structure depth (i.e., the average number of consecutive slices a structure is present in), structure size relative to the total volume, and average structural interslice spatial displacement. The extracted feature values for all data sets and their respective structures can be found in the Supplementary Material Table 1–8. We then used multiple different regression methods including Ridge regression, Lasso, Elastic Net, and Bayesian ARD regression, and then utilized the Bootstrap (with 1000 rounds) to find a mean regression vector for each. For more details about the feature extraction and regression analysis, see sections 1.1 and 1.2 of the Supplementary Material.

3.E. Statistical tests

For further analysis of model performance, we performed a Friedman test of equivalence between the methods, which reported significant differences on a 5 % level, and followed

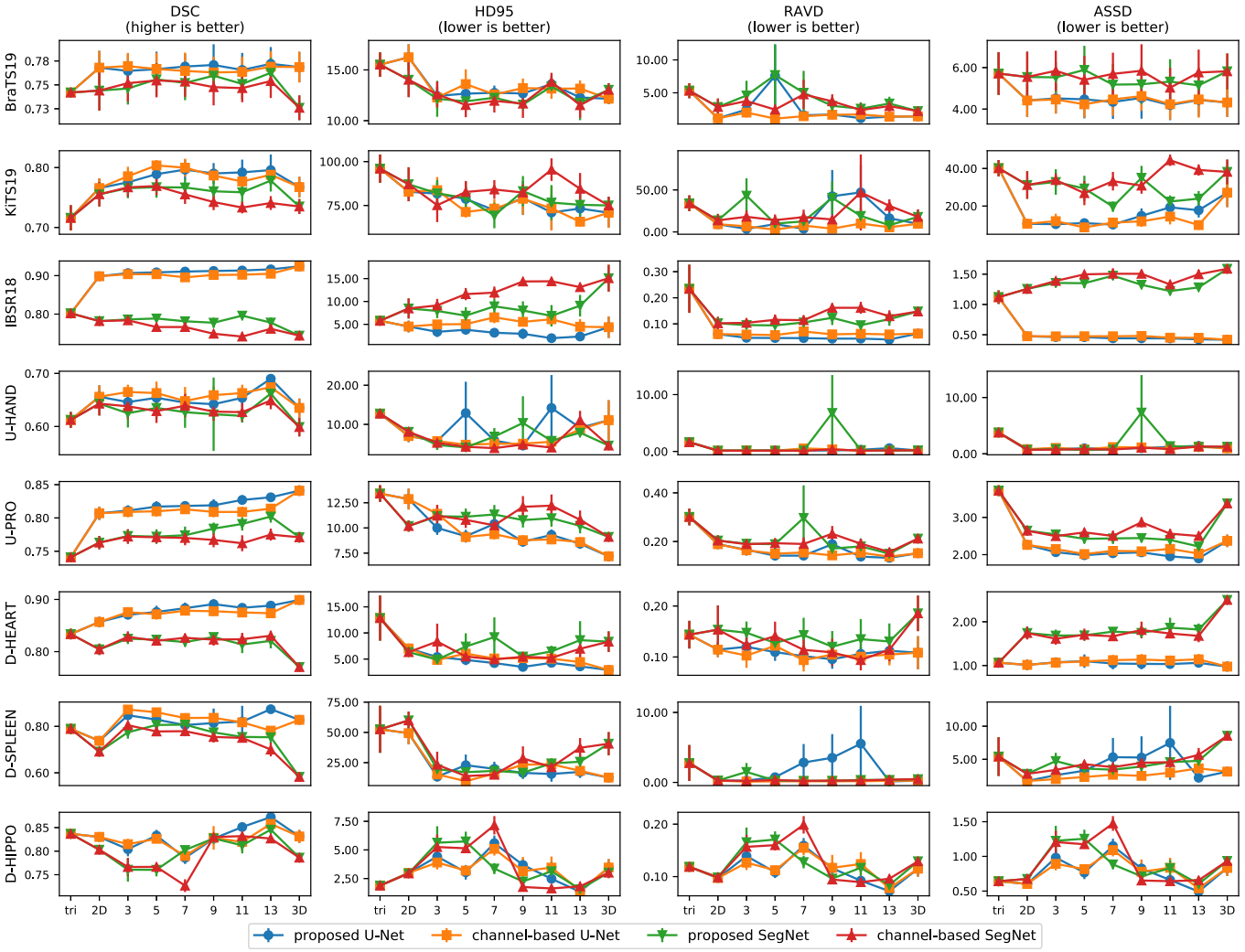


FIG. 12. Mean and standard deviation of 5 runs on all data sets in terms of DSC, HD95, RAVD, and ASSD. The reader should note that the best possible score for DSC is one, and is zero for the other metrics. HD95 and ASSD are given in mm. DSC and RAVD are dimensionless. [Color figure can be viewed at wileyonlinelibrary.com]

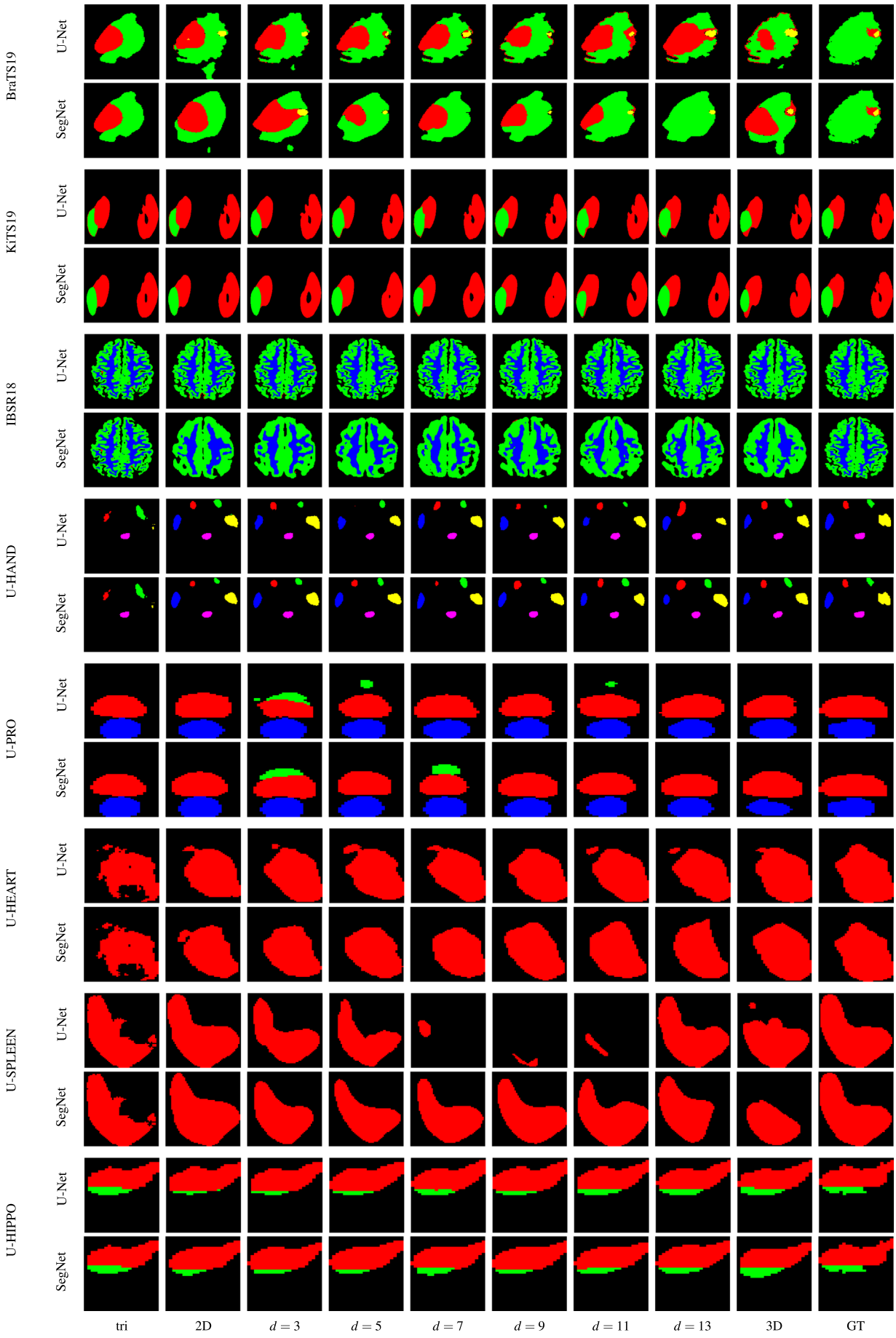
it up by a Nemenyi post hoc test (as proposed by Demsar et al. (2006)[51]) on all evaluated metrics by utilizing all predictions from eight data sets. These results are shown in Tables 21–22 for DSC and HD95 in the Supplementary Material, respectively.

4. RESULTS

The segmentation performances in terms of the various metrics of all models are illustrated in Fig. 12. For each data set, the mean value over all samples for each metric are plotted (with point-wise standard deviation bars) as a function of the input size, and are given for the triplanar, 2D, pseudo-3D with $d = 3$ through $d = 13$, and 3D models, and for the U-Net and SegNet backbones. These results in terms of DSC, HD95, RAVD, and ASSD are tabulated in Tables 17–20 in the Supplementary Material, respectively, along with summaries of the experiment setups per data set.

Randomly selected example segmentations are illustrated in Fig. 13. For each data set, a prediction from each model is given, along with the respective ground truth. For data sets with small structures, segmentations are cropped for ease of viewing. We chose to omit examples for the channel-based pseudo-3D models because of their high level of similarity to

FIG. 13. Qualitative results of proposed method on all data sets. From top to bottom: (i) BraTS19 tumor structures: the necrotic and non-enhancing tumor core (NCR/NET — label 1, red), the peritumoral edema (ED — label 2, green) and the GD-enhancing tumor (ET — label 4, yellow); (ii) KiTS19 class structure: the kidney (red) and kidney tumor (green); (iii) IBSR18 class structure: cerebrospinal fluid (red), white matter (green) and gray matter (blue); (iv) U-HAND class structure: left and right submandibular glands (red and green), left and right parotid glands (dark blue and yellow), larynx (light blue), and medulla oblongata (pink); (v) U-PRO class structure: prostate (red), bladder (green) and rectum (blue); (vi) D-HEART: heart (red); (vii) D-SPLEEN: spleen (red); (viii) D-HIPPO class structure: hippocampus head (red) and body (green). From left to right: triplanar, 2D, d = 3,5,7,9,11,13, 3D, and ground truth (GT). [Color figure can be viewed at wileyonlinelibrary.com]


BraTS19
KiTS19
IBSR18
U-HAND
U-PRO
U-HEART
U-SPLEEN
U-HIPPO
U-Net
SegNet
tri
2D
d = 3
d = 5
d = 7
d = 9
d = 11
d = 13
3D
GT

the proposed method. Segmentations with the channel-based method, along with additional exemplary segmentations, can be found in Figs. 4–6 in section 6 of the Supplementary Material.

The computational costs of the models used for BraTS19 experiments are presented in Table II. The number of model parameters, graphical memory use, and FLOps are only dependent on the model type, and therefore the corresponding columns in Table II are equal for all other data sets. The same variables are shown for the other data sets in Tables 9–15 in section 2 of the Supplementary Material, where the only differences are in the training and inference times due to the different numbers of samples; these two parameters scale with the data set size.

The results of the statistical tests on all evaluated metrics (see Section 2.E) are shown in Tables 21–24 in the Supplementary Material.

5. DISCUSSION

This study evaluated the inclusion of neighboring spatial context as an input of CNNs for medical image segmentation. Such pseudo-3D methods with a multislice input and single-slice output are commonly implemented by regarding the adjacent slices as additional channels to the central slice. Aside from this approach, we also proposed an alternative pseudo-3D method, based upon multiple preliminary 3D convolution steps before processing by a 2D CNN. Across eight different data sets and using U-Net and SegNet CNN backbones, we compared both these pseudo-3D methods, for an input size $d = 3$ up to $d = 13$, to (1) end-to-end 2D and 3D CNNs with respectively single slice and whole volume inputs and outputs and (2) triplanar orthogonal input CNNs, another common approach to multislice inputs. Additionally, we evaluated a number of computational parameters to get a sense of each model's hardware requirement and load.

5.A. Computational costs

As seen in Table II, the computational costs are in line with what would be expected. The transition block adds a relatively small amount of extra parameters on top of the main 2D network, and the required amount of GPU memory and FLOps scale accordingly with d. Since the input is still the same size as for the channel-based method, the training times per epoch are largely similar. One advantage of the fully 3D CNNs demonstrated in these results, is that prediction time is significantly faster because samples can be processed all at once instead of slice by slice. The triplanar network, which has to be trained and inferred on a voxel-by-voxel basis, is computationally very inefficient. While it uses by far the lowest number of FLOps, its training and prediction times are the highest of all models.

The high computational cost of end-to-end 3D convolution is also demonstrated in Table II. The memory footprint is almost 35 times larger than the 2D U-Net; over 16 GB is required to train on the complete volumes, which is at or above the limit of most modern commodity GPUs. Both pseudo-3D methods use less than 5 % of the GPU memory consumed by the end-to-end 3D network, even at $d = 13$. It can thus be concluded that both pseudo-3D methods are computationally very efficient ways of including more interslice information, with the proposed method being slightly more expensive in terms of the GPU memory consumption compared to the channel-based method.

TABLE II. Architecture comparison.

Model	#slices	#params (k)	Memory (MB)	FLOPs (M)	t per epoch (s)	p per sample (s)
Triplanar	3	203	535	0.405	554	36.15
2D	1	493	467	2.450	49	10.17
Proposed	3	495	497	2.463	73	10.46
	5	502	519	2.497	88	11.11
	7	509	541	2.532	109	11.43
	9	516	563	2.567	156	12.15
	11	523	586	2.602	204	12.46
	13	530	601	2.637	241	14.73
Channel-based	3	493	485	2.451	72	10.33
	5	493	497	2.453	82	10.49
	7	493	510	2.454	101	11.01
	9	493	523	2.454	138	11.17
	11	494	534	2.457	190	11.33
	13	495	541	2.459	249	12.39
3D	128	1 461	16 335	7.306	370	2.36

Experiments on U-Net architecture and multimodal BraTS19 data set. Patch shape was set at $160 \times 192 \times d$ where d is the number of slices. Here, t and p denote the training time per epoch and prediction time per sample, respectively.

5.B. Quantitative analysis

As can be seen in Fig. 12, overall, all experiments managed to produce acceptable segmentation results, even for data sets with complex structures such as the BraTS19 images, or with organs that can be hard to visually distinguish, such as in the BraTS19 set. One obvious similarity between these data sets is that using a U-Net backbone outperforms the SegNet in nearly every case. Regarding the behavior as a function of input size d, the results in Fig. 12 are inconclusive for almost all data sets.

In the plots from Fig. 12, there does not seem to be a large additional benefit by adding more slices as input over an end-to-end 2D approach. The largest improvements from 2D to pseudo-3D are in terms of HD95, but this does not hold for all data sets. For most data sets and metrics, however, the variance is either too high or the rate of improvement is too low to draw any strong conclusions. For these cases, it would be doubtful if the accessory downsides, for example, increased training time, are worth the at most marginal improvements in segmentation performance.

The only data sets in this study where most metrics do seem to significantly improve with d are the U-PRO and D-HEART sets. This improvement is most visible in terms of DSC and HD95. As more slices are being included in the input volume, the segmentation performance approaches that of a fully 3D network, and the proposed method outperforms the channel-based method by an increasing margin. While the overall improvement when going from 2D to pseudo-3D with $d = 13$ is arguably low, we can regard the U-PRO and D-HEART cases as demonstrations of the possibility that pseudo-3D models can improve the segmentation performance over 2D methods.

In almost all cases, triplanar 2D networks perform worse than all other models. The inefficiency of voxel-by-voxel training seems to produce numerous false positives, outweighing the additional spatial information. Moreover, the number of extracted patches is much greater than 2D, pseudo-3D, and 3D, making it quite storage-inefficient. While there are other, and perhaps more efficient implementations of triplanar CNNs than the one by Prasoon et al. (2013),[23] we do not consider this method to compete with the other models.

Fully 3D CNNs seems to produce equal or worse results compared to their 2D and pseudo-3D counterparts in most cases, mostly in terms of DSC and RAVD. Again, the only exception seems to be in the U-PRO and D-HEART results. This could be explained by the much higher number of parameters in the 3D CNNs, which makes them more prone to overfitting. There are examples in the literature of 3D U-Nets outperforming other methods, including studies on data sets that are also part of this study, e.g. Myronenko (2018)[52] for BraTS19 and Isensee and Maier-Hein (2019)[53] for KiTS19. However, apart from extensive hyperparameter optimization, these studies modify the standard U-Net to either fewer layers or numerous residual blocks which might induce a regularizing effect, thereby achieving better performances than reported in this study.

There does not seem to be a straight-forward explanation as to why some data sets perform better than other data sets. In an attempt to connect performance behaviour with d to differences in data set properties, a feature-based regression analysis was performed. However, we found no significant agreements between models that could connect one of these data set features to any performance metrics with respect to d. For more details about the feature extraction and regression analysis, see sections 1.1 and 1.2 of the Supplementary Material.

From Table 21 in the Supplementary Material we see that: (1) we again see that networks with a U-Net backbone appear to outperform those with a SegNet backbone, (2) the proposed model at $d = 13$ with a U-Net backbone significantly outperforms almost all other models, followed by the 3D U-Net, and (3) the proposed model with a SegNet backbone at $d = 13$ is also the best-performing model among those tested SegNet networks. Regarding the HD95, presented in Table 22 in the Supplementary Material, the best two performing methods are 3D U-Net and the proposed pseudo-3D method with a U-Net backbone ($d = 13$); while the two best performing methods are the proposed and the channel-based U-Nets at $d = 11$ when RAVD is evaluated (see Table 23 in the Supplementary Material). What is interesting in Table 24 in the Supplementary Material is that the proposed U-Net models outperform the rest (including 3D U-Net) in terms of ASSD. Another conclusion that we can make is that the triplanar CNN performs the worst in all evaluated metrics.

The reader should note that $d = 13$ might be the optimal value when all data sets and all metrics are regarded, but this statement is not supported in individual data sets. In Fig. 12 we observe that, for example.. the BraTS19 set does not perform better for any particular value of d. This, along with the substantial longer training times as d increases, still leaves ambiguity for the added value of using multislice inputs.

A possible follow-up study might be to investigate whether it was the multislice outputs (e.g., producing segmentations for all input slices) in pseudo-3D methods that improved the results in other studies. While this was out of the scope of this work, aggregating multiple outputs may be the main reason why pseudo-3D methods sometimes improve the segmentation performances. Based on our conclusions that using multislice inputs does not seem to improve the results on their own, the added benefit might only come into play from the aggregation of multiple outputs. In this case, using something like Bayesian dropout could prove just as beneficial. A study that analyses multislice outputs in a similar fashion as in this study could also include an investigation into loss functions that penalize anatomically unrealistic transitions between adjacent slices, as proposed in e.g. Ganaye et al. (2019).[35]

One caveat that is often glossed over when bench-marking DL segmentation methods in medical imaging across data sets is the preprocessing step of resampling to uniform voxel size. While a uniform resolution is crucial for neural network inputs, very heterogeneous data sets might render multislice inputs not useful if the interpolated slices are highly correlated. In other words, an input of multiple slices after interpolation might correspond to inputting a single slice at the

original resolution multiple times. A better case might be made if only natively isotropic samples are included, but for this study, this approach was not employed since it would drastically reduce the total number of samples. However, we do recommend this strategy for possible follow-up studies that might further investigate our findings and whether highly interpolated data sets do actually perform worse than natively isotropic data, in a multislice input setting.

5.C. Qualitative analysis

It is important to emphasize that the images in Fig. 13 are randomly selected single slices from thousands of samples and are therefore presented purely for illustrative purposes, and might not always be a representation of the overall segmentation performance of a particular data set. However, some remarks can be made that can be related to the quantitative results in Fig 12. The relatively large variance in segmentation performance between experiments of the BraTS19 data is demonstrated in Fig. 13; as seen, the predictions can differ quite drastically within the same model and with varying d. This reflects the metrics of the BraTS19 set presented in Fig. 12.

It also appears that the U-Net is better at capturing fine structural details, while the SegNet segmentations seem to be coarser and simpler. This becomes particularly noticeable in data sets with complex structures, such as the gray matter-white matter border in the IBSR18 images (Fig. 13). This in turn results in an overall large difference in mean DSC between U-Net and SegNet. When the ground truth structures are more coarsely shaped, such as in the U-HAND set, the SegNet can keep up much better with the U-Net performance.

5.D. Effect of the loss function

In an earlier stage of this project, we employed a different experimental setup with a pure DSC loss function. However, these initial experiments proved this loss not to be sufficient for all data sets. Particularly the KiTS19 and U-HAND data sets yielded unacceptably unstable results which, even with exactly equal hyperparameters, could either result in fairly accurate segmentations or complete failure. Investigation of the DSCs of individual structures demonstrated that in these failed experiments, multiple structures did not improve beyond a DSC on the order of 0.1. After adapting the loss function to include also the CE term [see Eq. (4)], the results improved substantially for all data sets. Performance details for each run using the pure DSC and final loss function can be seen in Fig. 3 and Table 16 in section 5 of the Supplementary Material.

6. CONCLUSION

This study systematically evaluated pseudo-3D CNNs, where a stack of adjacent slices is used as input for a prediction on the central slice. The hypothesis underlying this approach is that the added neighboring spatial information would improve segmentation performance, with only a small amount of added computational cost compared to an end-to-end 2D CNN. However, whether or not this is actually a sensible approach had not previously been evaluated in the literature.

Aside from the conventional method, where the multiple slices are input as multiple channels, we introduced here a novel pseudo-3D method where a subvolume is repeatably convolved in 3D to obtain a final 2D feature map. This 2D feature map is then in turn fed into a standard 2D network.

We investigated the segmentation performance in terms of multiple performance metrics and the computational cost for a large range of input sizes, for the U-Net and SegNet backbone architectures, and for five diverse data sets covering different anatomical regions, imaging modalities, and segmentation tasks. While pseudo-3D networks can have a large input image size and still be computationally less costly than fully 3D CNNs by a large factor, a significant improvement from using multiple input slices was only observed for an input size of 13 slices. We also observed that triplanar CNNs performed generally worse than the other models and were computationally much more inefficient compared to pseudo-3D and conventional 2D and 3D CNNs.

In the general case, multislice inputs appear not to improve segmentation results over using 2D or 3D CNNs. For the particular case of 13 input slices, the proposed novel pseudo-3D method does appear to have a slight advantage across all data sets compared to all other methods evaluated in this work.

ACKNOWLEDGMENTS

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the HPC2N in Umeå, Sweden. We are grateful for the financial support obtained from the Cancer Research Fund in Northern Sweden, Karin and Krister Olsson, Umeå University, The Västerbotten regional county, and Vinnova, the Swedish innovation agency.

[a]Author to whom correspondence should be addressed. Electronic mail: tommy.lofstedt@umu.se.

REFERENCES

1. Rangayyan RM, Ayres FJ, Desautels JL. A review of computer-aided diagnosis of breast cancer: toward the detection of subtle signs. *J Franklin Inst*. 2007;344:312–348.
2. Bauer S, Wiest R, Nolte L-P, Reyes M. A survey of MRI-based medical image analysis for brain tumor studies. *Phys Med Biol*. 2013;58:R97.
3. Dolz J, Kirişli H, Fechter T. Interactive contour delineation of organs at risk in radiotherapy: clinical evaluation on NSCLC patients. *Med Phys*. 2016;43:2569–2580.
4. Shen D, Wu G, Suk H-I. Deep learning in medical image analysis. *Annu Rev Biomed Eng*. 2017;19:221–248.
5. Litjens G, Kooi T, Bejnordi BE. A survey on deep learning in medical image analysis. *Med Image Anal*. 2017;42:60–88.
6. Sahiner B, Pezeshk A, Hadjiiski LM, et al. Deep learning in medical imaging and radiation therapy. *Med Phys*. 2019;46:e1–e36.
7. Milletari F, Navab N, Ahmadi S-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation, in 2016 Fourth International Conference on 3D Vision (3DV). IEEE; 2016:565–571.

8. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O. 3D U-Net: learning dense volumetric segmentation from sparse annotation, in International conference on medical image computing and computer-assisted intervention, Springer; 2016:424–432.
9. Dou Q, Yu L, Chen H, et al. 3D deeply supervised network for automated segmentation of volumetric medical images. *Med Image Anal*. 2017;41:40–54.
10. 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. Springer; 2015:234–241.
11. Vu MH, Grimbergen G, Simkó A, Nyholm T, Löfstedt T, End-to-End Cascaded U-Nets with a Localization Network for Kidney Tumor Segmentation, arXiv preprint arXiv:1910.07521 2019.
12. Vu MH, Nyholm T, Löfstedt T. TuNet: End-to-end Hierarchical Brain Tumor Segmentation using Cascaded Networks, arXiv preprint arXiv:1910.05338 2019.
13. Li W, Wang G, Fidon L, Ourselin S, Cardoso MJ, Vercauteren T. On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task, in International Conference on Information Processing in Medical Imaging in International Conference on Information Processing in Medical Imaging. Springer; 2017:348–360.
14. Chen H, Dou Q, Yu L, Qin J, Heng P-A. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage*. 2018;170:446–455.
15. Yu L, Yang X, Chen H, Qin J, Heng PA. Volumetric ConvNets with mixed residual connections for automated prostate segmentation from 3D MR images, in Thirty-first AAAI conference on artificial intelligence, 2017.
16. Yu L, Cheng J-Z, Dou Q, et al. Automatic 3D Cardiovascular MR Segmentation with Densely-Connected Volumetric ConvNets, in Descoteaux M, Maier-Hein L, Franz A, Jannin P, Collins DL, Duchesne S, eds. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2017*, Cham: Springer International Publishing; 2017:287–295.
17. Lu F, Wu F, Hu P, Peng Z, Kong D. Automatic 3D liver location and segmentation via convolutional neural network and graph cut. *Int J Comput Assist Radiol Surg*. 2017;12:171-182.
18. Kamnitsas K, Ledig C, Newcombe VF, et al. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med Image Anal*. 2017;36:61–78.
19. Lian C, Zhang J, Liu M, et al. Multi-channel multi-scale fully convolutional network for 3D perivascular spaces segmentation in 7T MR images. *Med Image Anal*. 2018;46:106–117.
20. Ren X, Xiang L, Nie D, et al. Interleaved 3D-CNNs for joint segmentation of small-volume structures in head and neck CT images. *Med Phys*. 2018;45:2063–2075.
21. Feng X, Qing K, Tustison NJ, Meyer CH, Chen Q. Deep convolutional neural network for segmentation of thoracic organs-at-risk using cropped 3D images. *Med Phys*. 2019;46:2169–2180.
22. Anirudh R, Thiagarajan JJ, Bremer T, Kim H. Lung nodule detection using 3D convolutional neural networks trained on weakly labeled data, in Medical Imaging 2016: Computer-Aided Diagnosis, International Society for Optics and Photonics, 2016;9785:978532.
23. Prasoon A, Petersen K, Igel C, Lauze F, Dam E, Nielsen M. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network, in International conference on medical image computing and computer-assisted intervention. Springer; 2013: 246–253.
24. Roth HR, Lu L, Seff A, et al. A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations, in International conference on medical image computing and computer-assisted intervention. Springer; 2014:520–527.
25. de Brebisson A, Montana G. Deep neural networks for anatomical brain segmentation, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015:20–28.
26. Yang D, Zhang S, Yan Z, Tan C, Li K, Metaxas D. Automated anatomical landmark detection ondistal femur surface using convolutional neural network, in 2015 IEEE 12th international symposium on biomedical imaging (ISBI). IEEE; 2015:17–21.
27. Lyksborg M, Puonti O, Agn M, Larsen R. An ensemble of 2D convolutional neural networks for tumor segmentation, in Scandinavian Conference on Image Analysis. Springer; 2015:201–211.
28. Mlynarski P, Delingette H, Criminisi A, Ayache N. 3D convolutional neural networks for tumor segmentation using long-range 2D context. *Comput Med Imaging Graph*. 2019;73:60–72.
29. Kitrungrotsakul T, Han X-H, Iwamoto Y, et al. VesselNet: A deep convolutional neural network with multi pathways for robust hepatic vessel segmentation. *Comput Med Imaging Graph*. 2019;75:74–83.
30. Geng Y, Ren Y, Hou R, Han S, Rubin GD, Lo JY, *2.5 D CNN model for detecting lung disease using weak supervision*, in Medical Imaging 2019: Computer-Aided Diagnosis, International Society for Optics and Photonics, 2019;10950:109503O.
31. Novikov AA, Major D, Wimmer M, Lenis D, Buhler K. Deep sequential segmentation of organs in volumetric medical scans. *IEEE Trans Med Imaging*. 2018;38:1207–1215.
32. Kitrungrotsakul T, Iwamoto Y, Han X-H, et al. A Cascade of CNN and LSTM Network with 3D Anchors for Mitotic Cell Detection in 4D Microscopic Image, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); IEEE; 2019:1239–1243.
33. Han X. Automatic liver lesion segmentation using a deep convolutional neural network method, arXiv preprint arXiv:1704.07239; 2017.
34. Ghavami N, Hu Y, Bonmati E, et al. Integration of spatial information in convolutional neural networks for automatic segmentation of intraoperative transrectal ultrasound images. *J Med Imaging*. 2018;6:011003.
35. Ganaye P-A, Sdika M, Triggs B, Benoit-Cattin H. Removing segmentation inconsistencies with semi-supervised non-adjacency constraint. *Med Image Anal*. 2019;58:101551.
36. Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell*. 2017;39:2481–2495.
37. Zackrisson B, Nilsson P, Kjellén E, et al. Two-year results from a Swedish study on conventional versus accelerated radiotherapy in head and neck squamous cell carcinoma-the ARTSCAN study. *Radiother Oncol*. 2011;100:41–48.
38. Menze BH, Jakab A, Bauer S, et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans Med Imaging*. 2014;34:1993–2024.
39. Bakas S, Akbari H, Sotiras A, et al. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Sci data*. 2017;4:170117.
40. Heller N. The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes. arXiv preprint arXiv:1904.00445; 2019.
41. Cocosco CA, Kollokian V, Kwan RK-S, Pike GB, Evans AC, Brainweb: Online interface to a 3D MRI simulated brain database, in NeuroImage, Citeseer. 1997.
42. Simpson AL, Antonelli M, Bakas S, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms, arXiv preprint arXiv:1902.09063; 2019.
43. Tobon-Gomez C, Geers AJ, Peters J, et al. Benchmark for algorithms segmenting the left atrium from 3D CT and MRI datasets. *IEEE Trans Med Imaging*. 2015;34:1460–1473.
44. Simpson AL, Leal JN, Pugalenthi A, et al. Chemotherapy-induced splenic volume increase is independently associated with major complications after hepatic resection for metastatic colorectal cancer. *J Am Coll Surg*. 2015;220:271–280.
45. Tustison NJ, Avants BB, Cook PA, et al. N4ITK: improved N3 bias correction. *IEEE Trans Med Imaging*. 2010;29:1310.
46. Kingma DP, Ba J, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980; 2014.
47. Roy AG, Conjeti S, Karri SPK, et al. ReLayNet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks. *Biomed Opt Express*. 2017;8: 3627–3642.
48. Wong KCL, Moradi M, Tang H, Syeda-Mahmood T. 3D Segmentation with Exponential Logarithmic Loss for Highly Unbalanced Object Sizes, in Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G, eds. *Medical Image Computing and Computer Assisted*

Intervention - MICCAI 2018. Cham: Springer International Publishing, 2018:612–619.
49. Simard PYBest practices for convolutional neural networks applied to visual document analysis., in Icdar, 2003;3.
50. Dong H, Supratak A, Mai L, et al. TensorLayer: A Versatile Library for Efficient Deep Learning Development. ACM Multimedia. 2017.
51. Demšar J. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res*. 2006;7:1–30.
52. Myronenko A. 3D MRI brain tumor segmentation using autoencoder regularization, in International MICCAI Brainlesion Workshop. Springer; 2018:311–320.
53. Isensee F, Maier-Hein KH. An attempt at beating the 3D U-Net, arXiv preprint arXiv:1908.02182 2019.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Data S1. Supplementary Material for "Evaluation of Multi-Slice Inputs to Convolutional Neural Networks for Medical Image Segmentation"

[Correction added on November 16, 2020, after first online publication: The supporting information was corrected .]