MICCAI FLARE21 challenge — Improving fast and low GPU memory abdominal multi-organ segmentation with deep supervision

Pierre-Henri Conze
IMT Atlantique, LaTIM UMR 1101, Inserm
Technopôle Brest-Iroise, Brest, France
pierre-henri.conze@imt-atlantique.fr

Abstract

This digest addresses multi-organ segmentation from abdominal multi-center, multi-phase, multi-vendor and multi-disease CT examinations using deep learning (DL). We describe the strategy followed for the fast and low GPU memory abdominal organ segmentation (FLARE) challenge, organized in conjunction with the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) 2021. To deal with clinical requirements and obtain a low GPU memory but still efficient DL model, we build a U-Net architecture based on a lightweight network from the VGG family as encoder: `VGG-13`. The decoder is designed in a symmetrical fashion to keep exploiting forward skip connections. According to challenge rules, encoder and decoder weights are randomly initialized, without relying on any pre-training strategy. To improve the gradient flow and encourage extracting discriminative features, our model leverages multi-stage deep supervision. Employed for automatic liver, kidneys, spleen and pancreas delineation, our pipeline reaches promising results and offers new perspectives for abdominal image interpretation and decision making in clinical practice.

1. Introduction

The recent development of non-invasive imaging technologies has opened new horizons in studying abdominal structures. Segmentation has become a crucial task in abdominal image analysis with many applications such as computer-assisted diagnosis, surgery planning, image-guided intervention or radiotherapy [1]. In particular, the precise delineation from Computed Tomography (CT) images of abdominal solid visceral organs including liver, kidneys, spleen and pancreas for localization, volume assessment or follow-up purposes has critical importance. However, the analysis of abdominal imaging datasets is challenging and time-consuming for clinicians since the abdomen is a complex body space. Robust and automatic abdominal multi-organ segmentation is required to guide image interpretation, facilitate decision making and improve patient care while avoiding manual delineation efforts.

In this area, many interactive, semi- and fully-automated methods have been proposed with diverse methodologies including statistical shape models [2], multi-atlas segmentation [3] or machine learning [4, 5]. More recently, outstanding performance has been reached in almost all medical image analysis tasks using deep learning (DL) [6]. Despite the large variability in organ shape, size, location and texture, abdominal multi-organ segmentation has naturally benefited from this massive trend [7, 8, 9]. Compared to conventional machine learning, the need for hand-crafted features no longer remains necessary. In particular, huge efforts have been devoted to automatic segmentation based on variants of Fully Convolutional Networks (FCN) [10]. Recent architectures comprise a regular FCN to extract multi-scale features, followed by an up-sampling branch that enables to recover the input resolution through up-convolutions [6]. In the medical image processing community, U-Net [11] is one of the most well-known approach among existing convolutional encoder-decoders. Able to learn from relatively small datasets, U-Net and its derivatives are the most likely to automatically infer high-level knowledge involved by radiologists when interpreting abdominal images.

Despite intensive developments in DL, it remains difficult to judge the effectiveness of deep networks for abdominal multi-organ delineation since they are mainly assessed on one single organ only or relatively small and private datasets. Their robustness to segment abdominal organs from multiple centers, phases, vendors and/or disease cases and manage the strong variability between subjects is therefore under-investigated. Rather than organ or center-specific strategies, the development of more comprehensive and generic computational models is needed [12, 13]. Few challenges including the fast and low GPU memory abdominal organ segmentation (FLARE) challenge[1], orga-

[1] https://flare.grand-challenge.org

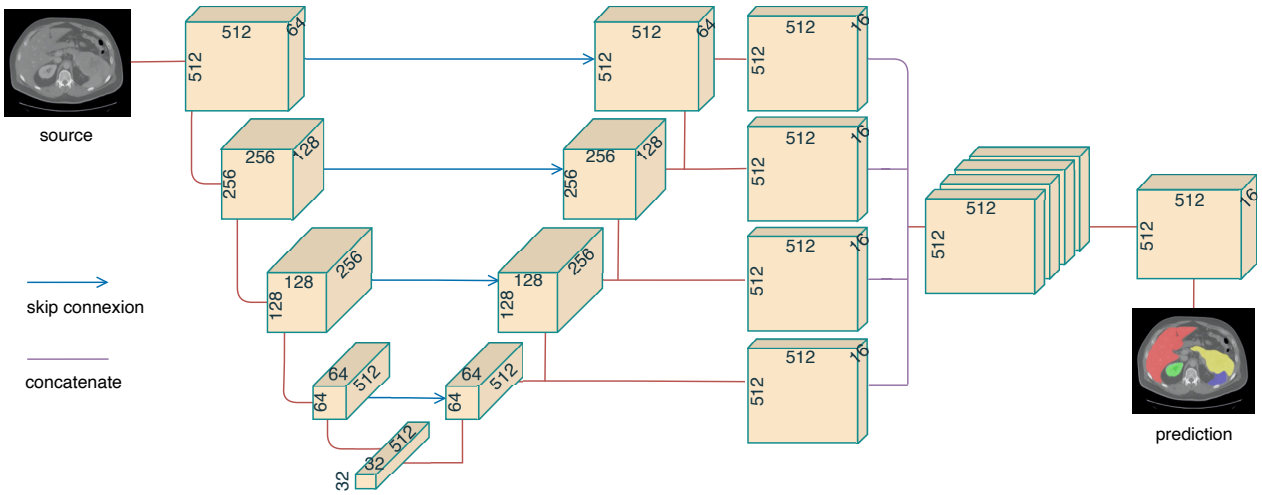

Figure 1. Proposed convolutional encoder-decoder architecture with forward skip connections (red arrows) and deep supervision.

nized in conjunction with the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) 2021, has been proposed to motivate further work on this perspective by making available a large dataset to segment liver, kidneys, spleen and pancreas from diverse CT examinations. Based on this unique dataset, we aim at developing a fast and low GPU memory abdominal multi-organ DL architecture that fits real clinical practice and requirements in terms of both accuracy and efficiency.

2. Method

To deal with clinical requirements and reach a low GPU memory but still efficient DL model, we build a U-Net architecture based on a lightweight `VGG-13` network from the VGG family [14] as encoder. The decoder branch is constructed in a similar fashion to obtain a symmetrical construction while keeping long-range shortcuts [15, 9]. According to the challenge rules which prevent relying on any pre-training scheme, weights of both encoder and decoder branches are randomly initialized. Neverthess, to further boost the performance, our model illustrated in Fig.1 leverages multi-stage deep supervision [16, 17].

2.1. Pre-processing

Intensity normalization is used as pre-processing step. Thus, each CT volume is clipped to the [1, 99] percentiles of the intensity values. In addition, a z-score normalization is applied based on the mean and standard deviation of the intensity values among the whole training dataset. Neither cropping nor resampling is employed.

2.2. Proposed method

Network architecture. Our model consists of a encoder-decoder architecture with forward skip connections from the encoder stages to the corresponding decoder stages. Contrary to standard U-Net [11], we employ an alternative simple but effective `VGG-13` encoder with batch normalization layers (`torchvision.models.vgg13_bn`).

To avoid large GPU memory consumption, we designed a 2D multi-class segmentation model with $C = 5$ classes dealing with background (`bg`), liver (`li`), kidneys (`ki`), spleen (`sp`) and pancreas (`pa`). The network independently processes axial slices to produce 2D segmentation masks which are then stacked together to recover 3D volumes. To exploit spatial relationships between abdominal structures, the model learns to simultaneoulsy delineate the multiple organs instead of relying on several organ-specific models.

The basic layer pattern consists of sequential layers including 3×3 convolutional layers (`conv`) with 1×1 stride and 1×1 padding followed by batch normalization (`BN`) and Rectified Linear Unit (`ReLU`) activation. Such pattern is repeated twice and followed by 2×2 max pooling (`MP`). The encoder comprises a sequence of 4 `[conv, BN, ReLU]x2 + MP` patterns (Fig.2). The first convolutional layer generates 64 channels. The number of channels doubles after each `MP` layer until it reaches 512. Compared to `VGG-13` [14], top layers including fully-connected layers and softmax are omitted. The fifth `[conv, BN, ReLU]x2` pattern from original `VGG-13` serves as central part to separate contracting and expanding paths.

To get a symmetrical construction while still using forward skip connections, the decoder branch is extended in the same fashion as the encoder by adding batch normalization layers and more features channels [9] (Fig.2). Additionally, feature maps as outputs of each intermediate decoder blocs are upsampled using bilinear interpolation to the size of the input image. In the same spirit as in [16, 17], a convolutional operation with 3×3 kernel is applied to create 16 feature maps at each level (Fig.2). These maps

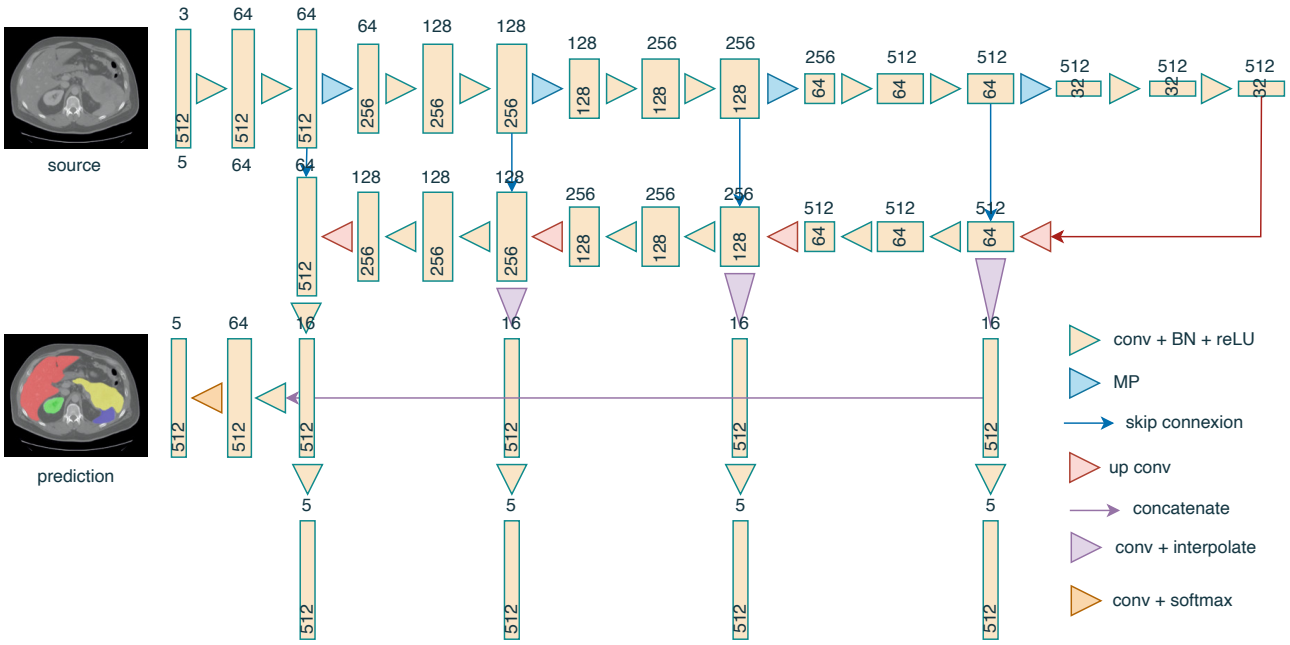

Figure 2. Detailed convolutional encoder-decoder architecture. The overall loss function is the weighted sum of losses estimated at different decoder levels involving supervision.

then go through deep supervision modules to improve the gradient flow and encourage learning more useful representations [17]. After having performed the concatenation of intermediate outputs (Fig.1), two convolutional layers including a final 3×3 one with softmax activation achieves pixel-wise multi-label segmentation.

Loss function. Our network is trained with the cross-entropy loss function $\mathcal{L}_{ce}$ defined below:

$$\mathcal{L}_{ce} = \frac{1}{N} \sum_{c=1}^{C} \sum_{i=1}^{N} g_i^c \, \texttt{log} \, p_i^c \tag{1}$$

where N is the number of pixels in the axial slices. p_i^c and g_i^c denote respectively the predicted probability and ground truth at pixel i for class label $c \in \{\texttt{bg}, \texttt{li}, \texttt{ki}, \texttt{sp}, \texttt{pa}\}$.

The overall loss function $\mathcal{L}$ is the weighted sum of the cross-entroy losses estimated at different decoder levels involving supervision:

$$\mathcal{L} = \sum_{j=1}^{M} w_j \mathcal{L}_{ce}^j + w_f \mathcal{L}_{ce}^f \tag{2}$$

where w_j and $\mathcal{L}_{ce}^j$ denote the weight and loss for the points of supervision at level j of the decoder. Following the `VGG-13` architecture [14], $M = 4$ intermediate decoder levels are considered. w_f and $\mathcal{L}_{ce}^f$ correspond to the weight and loss computed at the final network output (f stands for *final*). As in [17], we set $w_1 = 0.8$, $w_2 = 0.7$, $w_3 = 0.6$, $w_4 = 0.5$ and $w_f = 1$. Note that level $j = 1$ is closer to the network ending part than level $j > 1$.

Number of model parameters. The number of trainable parameters is 30,146,777 (around 115Mb), much less than the 41,268,192 parameters employed in nnU-Net [18].

Number of flops. The number of flops estimated using the fvcore library is 208095739904, to be compared with 590861472000 when using nnU-Net [18].

2.3. Post-processing

As post-processing, we keep the largest connected segmented areas for voxels respectively labeled as liver, spleen and pancreas. The two largest connected components are kept for voxels labeled as `ki` to get final kidney contours. No ensembling method is used.

3. Dataset and evaluation metrics

3.1. Dataset

Data. The dataset used of FLARE2021 is adapted from MSD [19] (Liver [20], Spleen, Pancreas), NIH Pancreas [21, 22, 23], KiTS [24, 25], and Nanjing University under the license permission. For more detail information of the dataset, please refer to the challenge website and [26].

Training / validation / testing split. The total number of cases is 511. An approximate 70%/10%/20% as train/validation/testing split is employed resulting in 361

Table 1. Data splits of FLARE2021.

Data Split	Center	Phase	# Num.
Training (361 cases)	The National Institutes of Health Clinical Center	portal venous phase	80
	Memorial Sloan Kettering Cancer Center	portal venous phase	281
Validation (50 cases)	Memorial Sloan Kettering Cancer Center	portal venous phase	5
	University of Minnesota	late arterial phase	25
	7 Medical Centers	various phases	20
Testing (100 cases)	Memorial Sloan Kettering Cancer Center	portal venous phase	5
	University of Minnesota	late arterial phase	25
	7 Medical Centers	various phases	20
	Nanjing University	various phases	50

Table 2. Environments and requirements.

Windows/Ubuntu version	Ubuntu 20.04 LTS
CPU	Intel Xeon W-2104 CPU@3.20GHz
RAM	16×4GB, 2.67MT/s
GPU	Nvidia 1080Ti
CUDA version	11.1
Programming language	Python3.7
Deep learning framework	pytorch (torch 1.6.0, torchvision 0.7.0)
Specification of dependencies	scikit-image, nibabel, torch
code is publicly available at	https://github.com/conze/Flare21IMTAtlantique

training cases, 50 validation cases, and 100 testing cases. The detail information is presented in Table 1.

3.2. Evaluation metrics

Both model accuracy and efficiency are evaluated throught the following evaluation metrics:

- Dice Similarity Coefficient (DSC)
- Normalized Surface Distance (NSD)
- running time
- maximum used GPU memory

4. Implementation details

4.1. Environments and requirements

The environments and requirements of the baseline method are shown in Table 2.

4.2. Training protocols

The training of the proposed method is displayed in Table 3. Among the 361 training cases, 10 examinations are randomly extracted to act as validation data. Instead of processing data in a 3D fashion using 3D patches, our model independently processes 2D axial slices. Image size taken for axial slices is 512×512.

Table 3. Training protocols.

Data augmentation methods	rotations (-20,20), translations (0.2,0.2), scaling (0.8,1.2), shears (-20,20)
Initialization of the network	normal initialization
Patch sampling strategy	None (full axial slices)
Batch size	4
Patch size	512×512 (full axial slices)
Total epochs	100
Optimizer	Adam
Initial learning rate	10^{-5}
Learning rate decay schedule	no decay
Stopping criteria, and optimal model selection criteria	Stopping criterion is reaching the maximum number of epoch. Optimal model selected on a validation set of 10 exams extracted from the training set
Training time	17 hours
CO_2eq	/

4.3. Testing protocols

For inference purposes, the same pre- and post-processing steps used during training are employed. Since our model is in 2D, the inference process is based on 2D 512×512 axial slices and produces 2D segmentation masks which are then stacked together to recover 3D volumes.

5. Results

5.1. Quantitative results on validation set

Table 4. Quantitative results on validation set.

Organ	DSC (%)	NSD (%)
Liver	94.58±4.340	74.81±13.79
Kidneys	81.34±19.09	73.47±16.63
Spleen	89.72±15.28	78.38±19.50
Pancreas	64.45±23.57	50.40±20.24

Table 4 illustrates the results on validation cases. The obtained DSC scores for liver, kidneys, spleen and pancreas are respectively 94.6%, 81.3%, 89.7% and 64.5%. The DSC

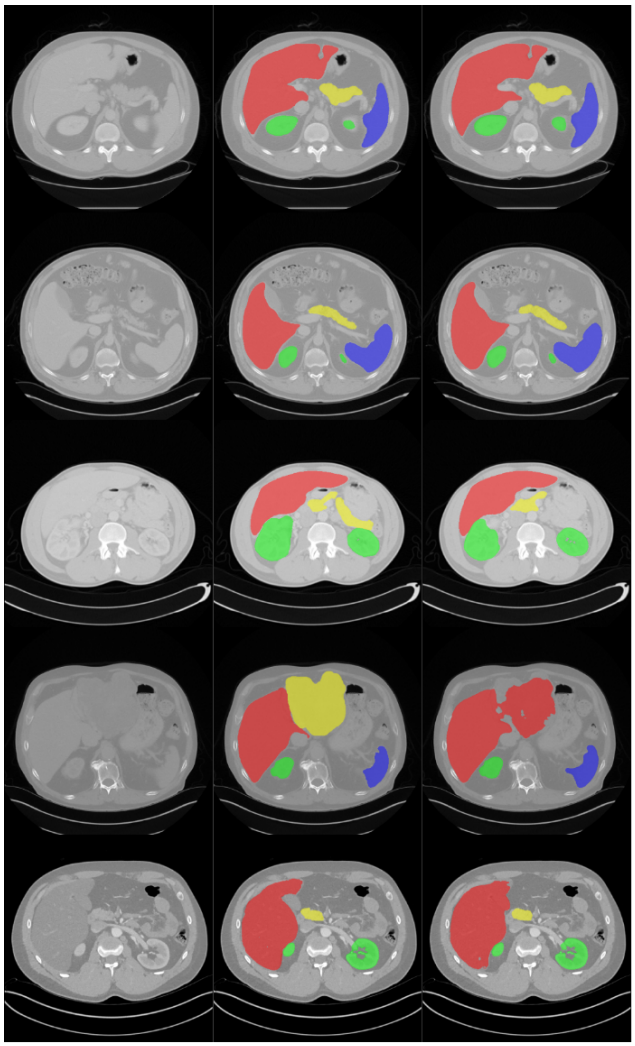

Figure 3. Qualitative results showing source images (column 1), ground truth (column 2) and predicted (column 3) abdominal multi-organ segmentation. Two cases of successful prediction are given in row 1-2, while rows 3-5 represent challenging cases.

score for liver indicates overall great segmentation results in terms of region overlap between ground truth and extracted areas. Nevertheless, the pancreas which obtains the lowest DSC still remains hard to extract. In terms of NSD scores, the proposed method achieves 74.8%, 73.5%, 78.4% and 50.4% for liver, kidney, spleen and pancreas respectively. Moreover, the performance for the pancreas is again the lowest. The spleen is the second best organ in terms of DSC and NSD. Finally, the vicinity between right kidney and liver as well as between left kidney and spleen seems to complicate the kidney contouring task.

5.2. Qualitative results

Figure 3 presents two simple (upper part) and three challenging (bottom part) examples from the validation set. Source axial slices, ground truth and predicted label maps are shown from left to right. Liver, kidneys, spleen and pancreas are respectively displayed in red, green, blue and yellow colors. The ability of our model to provide realistic abdominal organ contours appears encouraging given the low-computational constraints. Apart from under and over-segmentation issues, some organs seem not well delineated due to the presence of large lesions.

6. Discussion and conclusion

This work tackles abdominal multi-organ CT segmentation and describes the methodology employed for the fast and low GPU memory abdominal organ segmentation (FLARE) challenge. Standard pipelines are extended to lightweight convolutional encoder-decoders with deep supervision. Preliminary results suggests that further step forward can be achieved in abdominal image interpretation to avoid large model size and cost extensive computational resources in clinical practice. While many images containing healthy organs or organs with small lesions are accurately processed by our approach (especially for liver and spleen), the existence of large tumoral areas appears as a critical factor in term of delineation performance. In addition, the pancreas segmentation task should deserve further investigations to improve the capacity of DL models to handle the strong inter-patient anatomical variability in terms of size, shape, location and texture.

Acknowledgment

The author of this paper declares that the segmentation method they implemented for participation in the FLARE challenge has not used any pre-trained models nor additional datasets other than those provided by the organizers.

References

[1] R. M. Summers, "Progress in fully automated abdominal CT interpretation," *American Journal of Roentgenology*, vol. 207, no. 1, pp. 67–79, 2016. 1

[2] J. J. Cerrolaza *et al.*, "Automatic multi-resolution shape modeling of multi-organ structures," *Medical Image Analysis*, vol. 25, no. 1, pp. 11–21, 2015. 1

[3] Z. Xu *et al.*, "Efficient multi-atlas abdominal segmentation on clinically acquired CT with SIMPLE context learning," *Medical Image Analysis*, vol. 24, no. 1, pp. 18–27, 2015. 1

[4] R. Cuingnet *et al.*, "Automatic detection and segmentation of kidneys in 3D CT images using random forests," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2012, pp. 66–74. 1

[5] P.-H. Conze *et al.*, "Scale-adaptive supervoxel-based random forests for liver tumor segmentation in dynamic contrast-enhanced CT scans," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 2, pp. 223–233, 2017. 1

[6] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, 2017. 1

[7] H. R. Roth *et al.*, "Hierarchical 3D fully convolutional networks for multi-organ segmentation," *arXiv preprint arXiv:1704.06382*, 2017. 1

[8] E. Gibson *et al.*, "Automatic multi-organ segmentation on abdominal CT with dense V-networks," *IEEE Transactions on Medical Imaging*, vol. 37, no. 8, pp. 1822–1834, 2018. 1

[9] P.-H. Conze, A. E. Kavur, E. Cornec-Le Gall, N. S. Gezer, Y. Le Meur, M. A. Selver, and F. Rousseau, "Abdominal multi-organ segmentation with cascaded convolutional and adversarial deep networks," *Artificial Intelligence in Medicine*, vol. 117, 2021. 1, 2

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440. 1

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241. 1, 2

[12] J. J. Cerrolaza *et al.*, "Computational anatomy for multi-organ analysis in medical imaging: A review," *Medical Image Analysis*, vol. 56, pp. 44–67, 2019. 1

[13] A. E. Kavur *et al.*, "CHAOS challenge : Combined (CT-MR) healthy abdominal organ segmentation," *Medical Image Analysis*, vol. 69, 2021. 1

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[15] P.-H. Conze, S. Brochard, V. Burdin, F. T. Sheehan, and C. Pons, "Healthy versus pathological learning transferability in shoulder muscle MRI segmentation using deep convolutional encoder-decoders," *Computerized Medical Imaging and Graphics*, vol. 83, 2020. 2

[16] S. Xie and Z. Tu, "Holistically-nested edge detection," in *IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403. 2

[17] H. Dou, D. Karimi, C. K. Rollins, C. M. Ortinau, L. Vasung, C. Velasco-Annis, A. Ouaalam, X. Yang, D. Ni, and A. Gholipour, "A deep attentive convolutional neural network for automatic cortical plate segmentation in fetal MRI," *IEEE Transactions on Medical Imaging*, vol. 40, no. 4, pp. 1123–1133, 2020. 2, 3

[18] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021. 3

[19] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019. 3

[20] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser *et al.*, "The liver tumor segmentation benchmark (LITS)," *arXiv preprint arXiv:1901.04056*, 2019. 3

[21] H. Roth, A. Farag, E. Turkbey, L. Lu, J. Liu, and R. Summers, "Data from pancreas-CT. The Cancer Imaging Archive (2016)." 3

[22] H. R. Roth, L. Lu, A. Farag, H.-C. Shin, J. Liu, E. B. Turkbey, and R. M. Summers, "DeepOrgan: Multi-level deep convolutional networks for automated pancreas segmentation," in *International conference on medical image computing and computer-assisted intervention*, 2015, pp. 556–564. 3

[23] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The cancer imaging archive (TCIA): maintaining and operating a public information repository," *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013. 3

[24] N. Heller, F. Isensee, K. H. Maier-Hein, X. Hou, C. Xie, F. Li, Y. Nan, G. Mu, Z. Lin, M. Han *et al.*, "The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge," *Medical Image Analysis*, vol. 67, p. 101821, 2021. 3

[25] N. Heller, S. McSweeney, M. T. Peterson, S. Peterson, J. Rickman, B. Stai, R. Tejpaul, M. Oestreich, P. Blake, J. Rosenberg *et al.*, "An international challenge to use artificial intelligence to define the state-of-the-art in kidney and kidney tumor segmentation in ct imaging." *American Society of Clinical Oncology*, vol. 38, no. 6, pp. 626–626, 2020. 3

[26] J. Ma, Y. Zhang, S. Gu, C. Zhu, C. Ge, Y. Zhang, X. An, C. Wang, Q. Wang, X. Liu, S. Cao, Q. Zhang, S. Liu, Y. Wang, Y. Li, J. He, and X. Yang, "AbdomenCT-1K: Is abdominal organ segmentation a solved problem?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3