Semi-Supervised Abdomen Extraction and Organ Segmentation in CT Images

Rabin Adhikari[0000−0001−5019−2205]⋆, Safal Thapaliya[0000−0002−4463−6700]⋆, and Bishesh Khanal[0000−0002−2775−4748]

NepAl Applied Mathematics and Informatics Institute for research (NAAMII)
{rabin.adhikari,safal.thapaliya,bishesh.khanal}@naamii.org.np

Abstract. State-of-the-art supervised deep learning models perform well in segmenting abdominal organs from CT scan images when the test dataset has a similar distribution to the training dataset. The lack of generalization of deep learning models means that one still needs to label new data and retrain the models for new datasets in a different environment. However, expert annotation of multiple organs in volumetric scans is time-consuming and often prohibitively expensive. Semi-supervised methods that can leverage unlabeled data together with few labeled data could be an attractive solution, but existing semi-supervised multiple organs semantic segmentation methods do not perform well. Moreover, CT scans containing abdominal organs come with a great variety in the real clinical world with diverse resolutions and fields of view ranging from only the abdominal region to the whole body CT scan. In this paper, we propose a two-stage approach where abdomen region segmentation extracts the abdominal region, which is then fed as input to the abdominal organs segmentation network; both stages use the UNet architecture-based network. To leverage unlabeled data, we use FixMatchSeg, which adapts the semi-supervised classification method, FixMatch, to segmentation tasks. FixMatchSeg uses standard supervised loss with labeled examples and an unsupervised loss with pseudo labeling and consistency regularization that can leverage unlabeled samples. Our model improved the mean dice score from 31.54% to 52.1% on the validation set when utilizing 2,000 unlabeled training images over the 50 labeled images.

Keywords: Semi-Supervised · 3D U-Net · Abdomen Region Segmentor · Organ Segmentation · Easy Implementation

1 Introduction

Abdominal organ segmentation from CT scan images is an important medical imaging task with several clinical applications, such as aiding clinicians in diagnosing diseases, surgeons in therapy planning, and researchers in quantitative analysis for population-based studies. In recent years, supervised deep learning segmentation methods have performed very well on curated datasets where the

⋆ Equal Contribution

test sets do not have a significant distribution shift compared to the training set [9]. However, one still needs to label a large number of data and train again, even for the same task, if the model is intended to be deployed for test data that has domain shift such as population demography, scanners, a field of view of input images, etc. For multiple abdominal organs segmentation in 3D scans, manual annotation is tedious, time-consuming, and expensive. Thus, semi-supervised methods that leverage several unlabeled samples together with a few labeled instances can be an attractive solution to develop powerful segmentation models for new datasets in new environments. FLARE 2022 challenge[1] is focused on pushing the frontiers of semi-supervised segmentation methods. Additionally, the challenge is interested in building models that are not too big and can run relatively fast with an economical computing budget.

There are various ways to leverage unlabeled samples in semi-supervised segmentation models such as consistency regularization methods, proxy-label or pseudo labeling methods, generative models, and graph-based techniques [17]. Among these approaches, consistency regularization and pseudo labeling are common, widely used, and have shown promising results in both natural images and medical images [1,2,3,5,12,13]. A simple but effective semi-supervised classification approach, introduced in FixMatch [20], combines consistency regularization with pseudo-labeling. Upretee et al. [22] show with FixMatchSeg that FixMatch can be adapted to semantic segmentation tasks, providing promising results in medical images as well. Hence, we use FixMatchSeg as the semi-supervised approach for segmenting the abdominal region and abdominal organs.

The challenge provides 50 labeled and 2,000 unlabeled CT images. The task is to segment 13 abdominal organs, and a separate blind test set that is not available to the participants is used for the evaluation. One significant difficulty we saw in the dataset was the distribution shift due to the additional field of view in the unlabeled and validation set compared to that of the labeled set. A majority of the labeled images contain only the abdominal region, but the unlabeled and validation sets have images with the coverage extending up to the whole body. Similarly, the image size, spacing, and voxel anisotropy of the images varied very widely, as can be seen in Table 1.

Table 1: Min and Max (top & bottom) image size in pixels and pixel spacing in mm for various image sets. The unlabeled and validation set images have remarkably wide variation compared to the training set images.

Images	**Labeled**	**Unlabeled**	**Validation Set**
Image Size (px)	[512, 512, 71]	[512, 79, 24]	[512, 512, 82]
	[512, 512, 113]	[796, 768, 1059]	[512, 512, 1338]
Pixel Spacing (mm)	[0.65, 0.65, 2.5]	[0.35, 0.35, 0.45]	[0.62, 0.62, 0.8]
	[0.98, 0.98, 5.0]	[1.37, 3.0, 8.0]	[1.52, 1.52, 5.05]

[1] https://flare22.grand-challenge.org/

We propose two key ideas to leverage the unlabeled data and tackle the data diversity and the domain shift. To leverage the unlabeled data, we use FixMatchSeg [22]. For handling the diversity in size and spacing, and the presence of non-abdominal regions in unlabeled and validation data, we propose a two-stage approach: input image fed to an abdomen region segmentation network and an organ segmentation network that takes the output region extracted by the region segmentation network as input. Such a two-stage approach has been successfully used in pancreas segmentation from CT images [24] and in last year's winning FLARE submission [23].

2 Method

2.1 Preprocessing

All the input images first go through the following steps:

Set to RAI orientation: Since the majority of images were found to be in RAI orientation, we automatically set the orientation of all the images to RAI by extracting the metadata from the images and transforming them in the required direction using Monai[2] library.

Resample input images: We resample the input images of dimensions $[x, y, z]$ to the target size of $[x', y', z'] = [128, \geq 128, \geq 64]$ in axes orthogonal to Sagittal, Coronal and Axial planes respectively. Using the scaling factor $f = \frac{128}{x}$, resampled input images to the size $[128, \max(fy, 128), \max(fz, 64)]$ is fed to the Abdomen Region Segmentor. The region extracted from the Abdomen Region Segmentor is then similarly resampled to the size of $[192, \geq 192, \geq 96]$ and fed to the Abdomen Organs Segmentor.

Histogram Normalization: For each input sample, we perform histogram normalization followed by the intensity normalization to the range $[-1, 1]$ using `HistogramNormalize`[3] in Monai.

2.2 UNet based Semi-Supervised Abdominal Region and Organs Segmentation

Two-stage segmentation: Figure 1 shows the pipeline of our two-stage approach to first segment the abdomen region and then abdominal organs. The Abdomen Region Segmentor (ARS) is trained for a binary segmentation task where the foreground is obtained as the convex hull of all the target abdominal organs. The Abdomen Region Segmentor first finds the largest connected component from the ARS foreground predictions, and then the smallest box that

[2] https://monai.io/
[3] https://docs.monai.io/en/stable/transforms.html#histogramnormalize

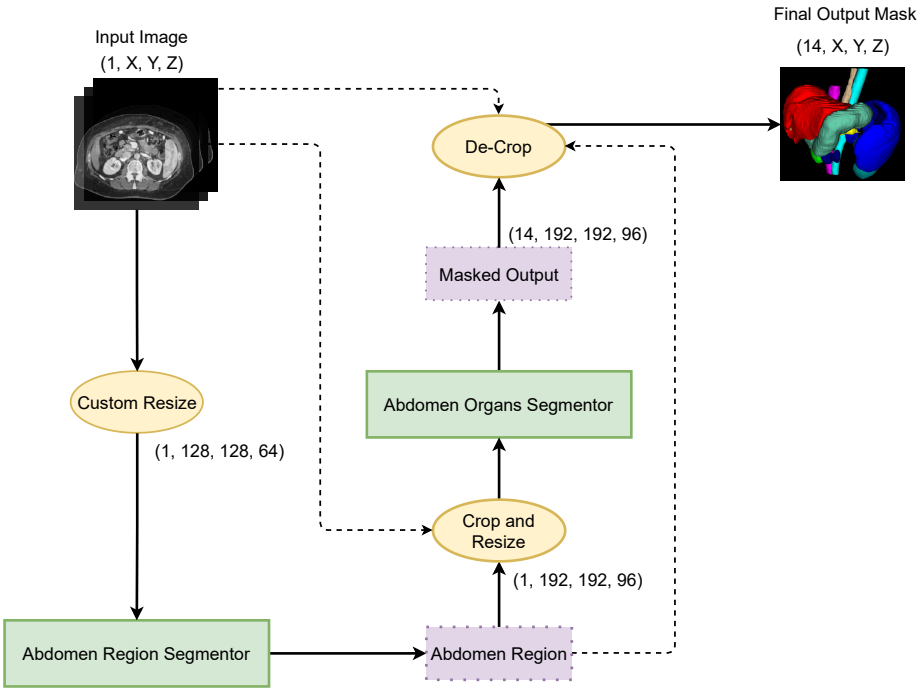


Fig. 1: The two-stage approach for abdominal organs segmentation. The Abdomen Region Segmentor (ARS) is first trained on a binary segmentation task where the foreground is the convex hull of all the abdominal organs. The abdomen region extracted from the ARS output is then fed as input to Abdomen Organs Segmentor after cropping and resizing.

includes the largest foreground component is extracted as the abdomen region. The abdomen region is then fed as input to Abdomen Organs Segmentor after cropping and resizing.

UNet with Dice-CE compound loss: We use UNet [18] with instance normalization [21] from Monai[4] for both the networks. We used anisotropic kernels of size (5, 5, 3) as the last axis has a lower resolution, hence the bigger physical receptive field can be achieved even with a lower kernel size. Since compound loss functions have proven to be robust in various medical image segmentation tasks [14], we used the weighted sum of soft Dice loss [16] without background (weight 1) and cross-entropy loss (weight 0.25) for training both the models.

FixMatchSeg to leverage unlabeled images: We implemented FixMatchSeg [22] to leverage the unlabeled examples for both the segmentation networks.

[4] https://docs.monai.io/en/stable/networks.html#unet

The segmentation model's prediction on a weakly augmented version of an input image is used as ground truth (pseudo-label) and the loss is computed with the model's prediction on a strongly augmented version of the same image. The input sample with pseudo-label is used in training only when the average of the pixel-wise max of the softmax prediction is higher than a certain threshold, that is only the samples where the model is confident enough. The transformations for weak and strong augmentations and the hyperparameters including the confidence threshold for pseudo-label are described in subsection 3.2.

Improving inference speed and reducing resource consumption: We JIT-compile our models[5] to convert them to torchscripts. TorchScript stores its definitions in a graph rather than dynamically. It combines numerous operators (kernels) into a single one, making inference quicker. Additionally, to make inference faster, we use `torch.jit.optimize_for_inference`[6].

2.3 Post-processing

We use the following post-processing steps in the provided order.

a. We use a sliding window inference with 25% overlap between consecutive windows. When combining the multiple predictions in the overlap regions, we weight the confidence of each prediction using a Gaussian kernel ($\sigma = 0.125$) with its mean placed at the window center. The assumption for using such a kernel is that the network makes better predictions for the organs in the central region of the input image.
b. We use `KeepLargestConnectedComponent`[7] from Monai to keep the largest connected component with connectivity set to 1.

3 Experiments

3.1 Dataset and evaluation measures[8]

The FLARE 2022 dataset is curated from more than 20 medical groups under the license permission, including MSD [19], KiTS [8,7], AbdomenCT-1K [15], and TCIA [4]. The training set includes 50 labeled CT scans with pancreas disease; 2000 unlabeled CT scans with liver, kidney, spleen, or pancreas diseases. The validation set includes 50 CT scans with liver, kidney, spleen, or pancreas diseases. The testing set includes 200 CT scans where 100 cases have liver, kidney, spleen, or pancreas diseases and the other 100 cases have uterine corpus endometrial,

[5] https://pytorch.org/docs/stable/generated/torch.jit.script.html
[6] https://pytorch.org/docs/stable/generated/torch.jit.optimize_for_inference.html
[7] https://docs.monai.io/en/stable/transforms.html#keeplargestconnectedcomponent
[8] This section, as suggested in the challenge guideline, is copied from the template provided by the challenge organizers.

urothelial bladder, stomach, sarcomas, or ovarian diseases. The metadata does not contain disease, demography, and center or scanner information.

The evaluation measures consist of two accuracy measures: Dice Similarity Coefficient (DSC) and Normalized Surface Dice (NSD), and three running efficiency measures: running time, area under GPU memory-time curve, and area under CPU utilization-time curve.

3.2 Implementation details

Environment settings: The development environments and requirements are presented in Table 2.

Table 2: Development environments and requirements.

Linux version	Ubuntu 20.04.4 LTS
CPU	Intel® Core™ i9-9820X CPU @ 3.30GHz
RAM	16 × 4 GB; 3200 MT/s
GPU	NVIDIA GeForce GTX 1080 Ti 11GB and NVIDIA TITAN Xp 12 GB
CUDA version	11.7
Programming language	Python 3.8.10
Deep learning framework	PyTorch (Torch 1.12, Monai 0.9.0)
Specific dependencies	pytorch_lightning
Link to code	https://github.com/naamiinepal/flare-2022

Data Augmentation Strategy: FixMatchSeg needs weak and strong augmentation. The transformations for weak and strong augmentation were selected from the transformations used by nnU-Net [9,10]. The following augmentations, in order, were performed to the input images of the segmentor network.

Weak augmentation:

a. Randomly rotate by 15° in all directions with 0.9 probability with bilinear interpolation for image and nearest neighbor for the label with zero padding.
b. Randomly zoom, with zoom value sampled from a Uniform distribution in the range (0.8, 1.3) with 0.9 probability, with trilinear interpolation.
c. Randomly crop with size (128, 128, 64) for Abdomen Region Segmentor and (192, 192, 96) for Abdomen Organs Segmentor. If the input size for this step is less than the crop size, we zero pad instead of cropping.

Strong augmentation, with a probability of 0.9:

a. Gaussian Random Noise with mean $\mu = 0$ and $\sigma = 0.1$

b. Gaussian smoothing with σ sampled from the Uniform distribution from the range (0.25, 1.5).
c. Multiplying the intensity by a factor sampled from the Uniform distribution from the range (0.7, 1.3).
d. Gamma correction by a factor sampled from the Uniform distribution from the range (0.65, 1.5).
e. Simulating low resolution by scaling down the image by a factor sampled from the Uniform distribution from the range (0.5, 1) by using the nearest neighbor interpolation and upscaling to the same shape using trilinear interpolation.

Table 3: Hyperparamenters for Abdomen Region Segmentor

Network initialization	Kaiming Uniform [6]
Batch size*	4
Gradient Accumulation Batches	64
Patch size	$128 \times 128 \times 64$
Total epochs	100
Optimizer	Adam [11] with betas=(0.9, 0.999)
Initial learning rate (LR)	0.01
Lr decay schedule	ReduceLROnPlateu with the patience of 3 epochs
Training time	6.678 hours
Loss Function	Dice-CE Compound Loss
Number of model parameters	637,425
Number of flops	7.156G

* One patient's input is randomly cropped twice during the weak augmentation (see section 3.2) to give two input samples to the segmentors, which are sent in a single batch.

Table 4: Hyperparamenters for Abdomen Organs Segmentor

Network initialization	Kaiming Uniform [6]
Batch size*	2
Gradient Accumulation Batches	64
Patch size	$192 \times 192 \times 96$
Total epochs	50
Optimizer	Adam [11] with betas=(0.9, 0.999)
Initial learning rate (LR)	0.05
Lr decay schedule	ReduceLROnPlateu with the patience of 3 epochs
Training time	19.02 hours
Loss Function	Dice-CE Compound Loss
Number of model parameters	1.989M
Number of flops	42.345G

4 Results and discussion

4.1 Quantitative results on validation set

Table 5: Dice Similarity Coefficient (DSC) comparison between training the model with only annotated images and training it using unlabeled images. Here, the model's mean DSC after using unlabeled images is higher than the DSC when using solely annotated images.

Organ	Without Unlabeled Images	With Unlabeled Images
Aorta	0.6482	0.7786
Duodenum	0.1459	0.3345
Esophagus	0.3137	0.5225
Gallbladder	0.2187	0.3603
Inferior Vena Cava	0.5403	0.634
Left Adrenal Gland	0.0400	0.3521
Left Kidney	0.3381	0.5147
Liver	0.5907	0.8453
Pancreas	0.2032	0.4003
Right Adrenal Gland	0.0644	0.3406
Right Kidney	0.3334	0.493
Spleen	0.5068	0.6993
Stomach	0.1562	0.4978
Average	0.3154	0.521

Table 5 shows that our method leverages the unlabeled data, improving the mean dice score to 0.521 from 0.3154 when not using the unlabeled data. However, the overall dice score is not high compared to the leaderboard which achieved 0.9064 on the validation set. From our preliminary qualitative analysis, it seems that the abdomen localization could not be accurately done when the input images had whole-body CT scans or regions outside the abdomen. In particular, a large majority of the labeled examples did not have a field of view outside the abdomen regions but the unlabeled and validation set had regions outside the abdomen. It seems that the model was unable to train properly due to this domain shift.

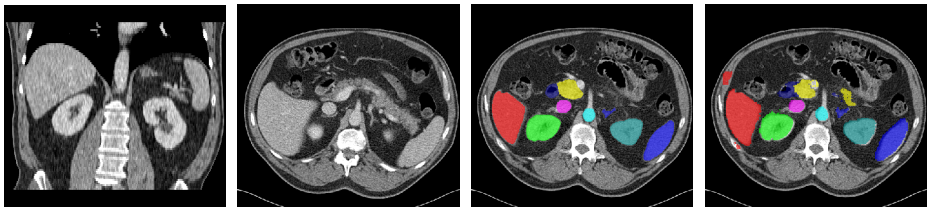
4.2 Qualitative results on validation set

The segmentation findings for two successfully segmented images (case 21 and case 35) and two difficult-to-segment images (case 42 and case 44) are shown in Figure 2 and Figure 3, respectively.

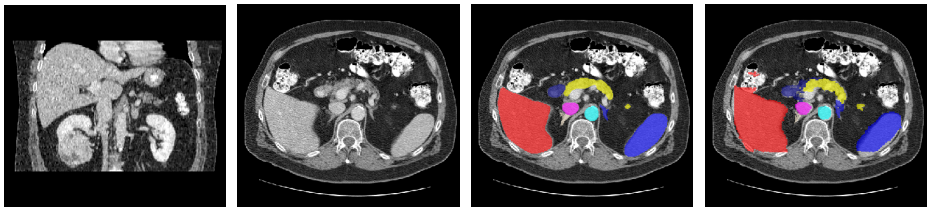
Some images also have some organs missing (case 31 has the left kidney missing and case 44 has the right kidney missing). Our model is not robust against such cases.

4.3 Segmentation efficiency results

The model validation was performed by running a docker container on a Ubuntu 20.04 desktop with the following specifications.

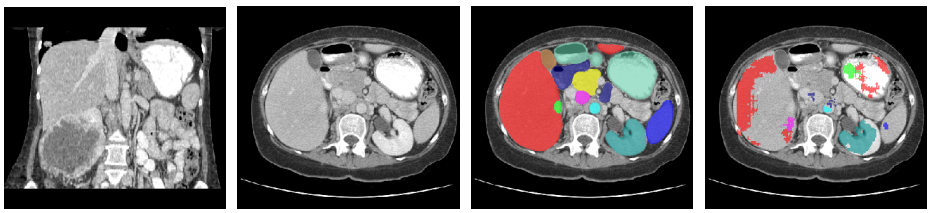


(a) a coronal slice; C21 (b) an axial slice (c) ground truth (d) model segmentation

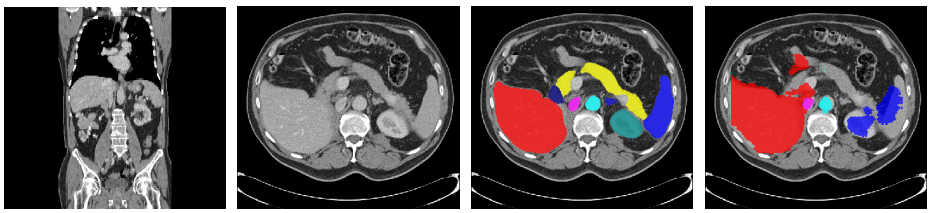


(e) a coronal slice; C35 (f) an axial slice (g) ground truth (h) model segmentation

Fig. 2: The well-segmented images with mean DSC 0.71(C21) and 0.70(C35) are shown above. They have smaller file sizes and solely include CT scans of the abdomen. The model performs well in these circumstances because the annotated images include CT scans restricted to the abdomen areas.



(a) a coronal slice; C42 (b) an axial slice (c) ground truth (d) model segmentation



(e) a coronal slice; C44 (f) an axial slice (g) ground truth (h) model segmentation

Fig. 3: The challenging examples with mean DSC 0.28(C42) and 0.23(C44) are shown above. These CT scans cover a wider area of the body, such as the entire body. Though in the case of patient 42, the image is restricted to the abdomen area, it contains a diseased right kidney whose size and intensity are unusual.

- CPU: Intel® Xeon(R) W-2133 CPU @ 3.60GHz × 12
- GPU: NVIDIA QUADRO RTX5000 (16G)
- RAM: 32G (Available memory 28G)
- Driver Version: 510.60.02
- CUDA Version: 11.6
- Docker version 20.10.13

The average running time, GPU memory usage, AUC of the GPU time, and CPU time of the model evaluation are given in Table 6. The running time, memory usage, and the AUC of both GPU and CPU curves were all observed on the 50 validation cases provided by the challenge's organizers. In the example with a file size of 463 MB and more than 1000 slices (case 50), the maximum values of running time, GPU memory, Area under GPU memory-time curve, and Area under CPU utilization-time curve were found to be 34.02 seconds, 6.231 GB, 144.879 GBs, and 789.77 %s, respectively.

Table 6: Average time of execution, GPU memory used, area under GPU memory-time curve, and area under CPU utilization-time curve for the 50 validation cases

Running Time	**Maximum GPU Memory**	**Area under GPU Memory-Time Curve**	**Area under CPU Utilization-Time Curve**
10.2456 s	4.3975 GB	27.275 GBs	256.7952 %s

4.4 Limitation and future work

While the model seems to perform satisfactorily on input CT scans with the field of view restricted to the abdominal regions only. The large majority of the labeled training images are of this type. However, the unlabeled training images and the validation set images have domain shifts with a much larger field of view extending up to the whole body scan. We believe that our method's pseudo-labeling approach could not provide good pseudo-labels on the unlabeled images with this domain shift when training the abdomen region segmenter. Poor abdomen region segmentation then naturally leads to poor segmentation of the organs as the organs segmenter will not get the right input. Similarly, our method does not address the missing organs and domain shift due to disease which makes it difficult to provide accurate segmentation of these cases.

A more detailed analysis of the failure cases and quantitative analysis such as the correlation of dice scores vs input field of view and abdomen region segmenter metrics will provide better insight into the issues with the current method. Since the two-stage approach can be fragile when the first stage performs poorly, exploring end-to-end coarse-to-fine models with networks taking both coarse and fine resolution images together is another interesting direction. Finally, semi-supervised methods other than FixMatchSeg and using self-supervision for pre-training can be explored.

5 Conclusion

FixMatchSeg was able to leverage the unlabeled data, improving the segmentation performance from 0.3154 mean DSC (only labeled data) to 0.521 DSC when adding the unlabeled data. The final segmentation performance has reasonably low compared to the leaderboard toppers. We believe this to be due to the inability of our method to handle the domain shift between the labeled and unlabeled images, and the validation-test images. However, a more detailed error and failure analysis are needed in the future together with the exploration of other semi-supervised methods.

Acknowledgments We declare that the segmentation method we implemented for participation in the FLARE 2022 challenge has not used any pre-trained models or additional datasets other than those provided by the organizers. The proposed solution is fully automatic without any manual intervention.

References

1. Bai, W., Oktay, O., Sinclair, M., Suzuki, H., Rajchl, M., Tarroni, G., Glocker, B., King, A., Matthews, P.M., Rueckert, D.: Semi-supervised learning for network-based cardiac mr image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 253–260. Springer (2017) 2
2. Baur, C., Albarqouni, S., Navab, N.: Semi-supervised deep learning for fully convolutional networks. In: International conference on medical image computing and computer-assisted intervention. pp. 311–319. Springer (2017) 2
3. Bortsova, G., Dubost, F., Hogeweg, L., Katramados, I., Bruijne, M.d.: Semi-supervised medical image segmentation via learning consistency under transformations. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 810–818. Springer (2019) 2
4. Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al.: The cancer imaging archive (tcia): maintaining and operating a public information repository. Journal of digital imaging **26**(6), 1045–1057 (2013) 5
5. Cui, W., Liu, Y., Li, Y., Guo, M., Li, Y., Li, X., Wang, T., Zeng, X., Ye, C.: Semi-supervised brain lesion segmentation with an adapted mean teacher model. In: International Conference on Information Processing in Medical Imaging. pp. 554–565. Springer (2019) 2
6. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015) 7
7. Heller, N., Isensee, F., Maier-Hein, K.H., Hou, X., Xie, C., Li, F., Nan, Y., Mu, G., Lin, Z., Han, M., 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 **67**, 101821 (2021) 5
8. Heller, N., McSweeney, S., Peterson, M.T., Peterson, S., Rickman, J., Stai, B., Tejpaul, R., Oestreich, M., Blake, P., Rosenberg, J., 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. (2020) 5

9. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021) 2, 6
10. Isensee, F., Jäger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: Automated design of deep learning methods for biomedical image segmentation. arXiv preprint arXiv:1904.08128 (2019) 6
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 7
12. Li, X., Yu, L., Chen, H., Fu, C.W., Xing, L., Heng, P.A.: Transformation-consistent self-ensembling model for semisupervised medical image segmentation. IEEE Transactions on Neural Networks and Learning Systems **32**(2), 523–534 (2020) 2
13. Li, Y., Chen, J., Xie, X., Ma, K., Zheng, Y.: Self-loop uncertainty: A novel pseudo-label for semi-supervised medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 614–623. Springer (2020) 2
14. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.L.: Loss odyssey in medical image segmentation. Medical Image Analysis **71**, 102035 (2021) 4
15. Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., Zhang, Y., An, X., Wang, C., Wang, Q., Liu, X., et al.: Abdomenct-1k: Is abdominal organ segmentation a solved problem. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 5
16. 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). pp. 565–571. IEEE (2016) 4
17. Ouali, Y., Hudelot, C., Tami, M.: An overview of deep semi-supervised learning. arXiv preprint arXiv:2006.05278 (2020) 2
18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 4
19. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019) 5
20. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems **33**, 596–608 (2020) 2
21. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016) 4
22. Upretee, P., Khanal, B.: Fixmatchseg: Fixing fixmatch for semi-supervised semantic segmentation. arXiv preprint arXiv:2208.00400 (2022) 2, 3, 4
23. Zhang, F., Wang, Y., Yang, H.: Efficient context-aware network for abdominal multi-organ segmentation. arXiv preprint arXiv:2109.10601 (2021) 3
24. Zhu, Z., Xia, Y., Shen, W., Fishman, E., Yuille, A.: A 3d coarse-to-fine framework for volumetric medical image segmentation. In: 2018 International conference on 3D vision (3DV). pp. 682–690. IEEE (2018) 3