Automatic Liver Segmentation Using an Adversarial Image-to-Image Network

Dong Yang[1], Daguang Xu[2(✉)], S. Kevin Zhou[2], Bogdan Georgescu[2], Mingqing Chen[2], Sasa Grbic[2], Dimitris Metaxas[1], and Dorin Comaniciu[2]

[1] Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA
[2] Medical Imaging Technologies, Siemens Healthcare Technology Center, Princeton, NJ 08540, USA
daguang.xu@siemens-healthineers.com

Abstract. Automatic liver segmentation in 3D medical images is essential in many clinical applications, such as pathological diagnosis of hepatic diseases, surgical planning, and postoperative assessment. However, it is still a very challenging task due to the complex background, fuzzy boundary, and various appearance of liver. In this paper, we propose an automatic and efficient algorithm to segment liver from 3D CT volumes. A deep image-to-image network (DI2IN) is first deployed to generate the liver segmentation, employing a convolutional encoder-decoder architecture combined with multi-level feature concatenation and deep supervision. Then an adversarial network is utilized during training process to discriminate the output of DI2IN from ground truth, which further boosts the performance of DI2IN. The proposed method is trained on an annotated dataset of 1000 CT volumes with various different scanning protocols (e.g., contrast and non-contrast, various resolution and position) and large variations in populations (e.g., ages and pathology). Our approach outperforms the state-of-the-art solutions in terms of segmentation accuracy and computing efficiency.

1 Introduction

Accurate liver segmentation from three dimensional (3D) medical images, e.g. computed tomography (CT) or magnetic resonance imaging (MRI) is essential in many clinical applications, such as pathological diagnosis of hepatic diseases, surgical planning, and postoperative assessment. However, automatic liver segmentation is still a highly challenging task due to the complex background, fuzzy boundary, and various appearance of liver in medical images.

To date, several methods have been proposed for automatic liver segmentation from 3D CT scans. Generally, they can be categorized into non-learning-based and learning-based approaches. Non-learning-based approaches usually

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-66179-7_58) contains supplementary material, which is available to authorized users.

M. Descoteaux et al. (Eds.): MICCAI 2017, Part III, LNCS 10435, pp. 507–515, 2017.
DOI: 10.1007/978-3-319-66179-7_58

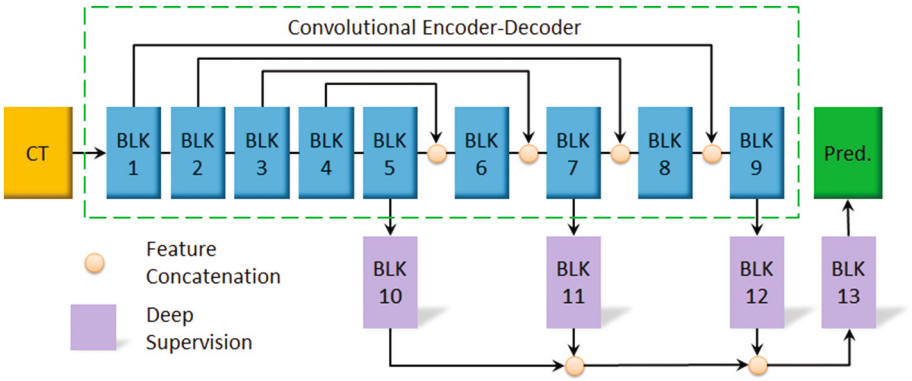

Fig. 1. Proposed deep image-to-image network (DI2IN). The front part is a convolutional encoder-decoder network with feature concatenation, and the backend is deep supervision network through multi-level. Blocks inside DI2IN consist of convolutional and upscaling layers.

rely on the statistical distribution of the intensity, including atlas-based [1], active shape model (ASM)-based [2], level-set-based [3], and graph-cut-based [4] methods, etc. On the other hand, learning-based approaches take the advantage of hand-crafted features to train the classifiers to achieve good segmentation. For example, in [5], the proposed hierarchical framework applies marginal space learning with steerable features to handle the complicated texture pattern near the liver boundary.

Until recently, deep learning has been shown to achieve superior performance in various challenging tasks, such as classification, segmentation, and detection. Several automatic liver segmentation approaches based on convolutional neural network (CNN) have been proposed. Dou et al. [6] demonstrated a fully convolutional network (FCN) with deep supervision, which can perform end-to-end learning and inference. The output of FCN is refined with a fully connected conditional random field (CRF) approach. Similarly, Christ et al. [7] proposed cascaded FCNs followed by CRF refinement. Lu et al. [8] used a FCN with graph-cut based refinement. Although these methods demonstrated good performance, they all used pre-defined refinement approaches. For example, both CRF and graph-cut methods are limited to the use of pairwise models, and time-consuming as well. They may cause serious leakage at boundary regions with low contrast, which is common in liver segmentation.

Meanwhile, Generative Adversarial Network (GAN) [9] has emerged as a powerful framework in various tasks. It consists of two parts: generator and discriminator. The generator tries to produce the output that is close to the real samples, while the discriminator attempts to distinguish between real and generated samples. Inspired by [10], we propose an automatic liver segmentation approach using an adversarial image-to-image network (DI2IN-AN). A deep image-to-image network (DI2IN) is served as the generator to produce the liver

segmentation. It employs a convolutional encoder-decoder architecture combined with multi-level feature concatenation and deep supervision. Our network tries to optimize a conventional multi-class cross-entropy loss together with an adversarial term that aims to distinguish between the output of DI2IN and ground truth. Ideally, the discriminator pushes the generator's output towards the distribution of ground truth, so that it has the potential to enhance generator's performance by refining its output. Since the discriminator is usually a CNN which takes the joint configuration of many input variables, it embeds the higher-order potentials into the network (the geometric difference between prediction and ground truth is represented by the trainable network model instead of heuristic hints). The proposed method also achieves higher computing efficiency since the discriminator does not need to be executed at inference.

All previous liver segmentation approaches were trained using dozens of volumes which did not take the full advantage of CNN. In contrast, our network leverages the knowledge of an annotated dataset of 1000+ CT volumes with various different scanning protocols (e.g., contrast and non-contrast, various resolution and position) and large variations in populations (e.g., ages and pathology). To the best of our knowledge, our experiment is the first time that more than 1000 annotated 3D CT volumes are adopted in liver segmentation tasks. The experimental result shows that training with such a large dataset significantly improves the performance and enhances the robustness of the network.

2 Methodology

2.1 Deep Image-to-Image Network (DI2IN) for Liver Segmentation

In this section, we present a deep image-to-image network (DI2IN), which is a multi-layer convolutional neural network (CNN), for the liver segmentation. The segmentation task is defined as the voxel-wise binary classification.

DI2IN takes the entire 3D CT volumes as input, and outputs the probability maps that indicate how likely voxels belong to the liver region. As shown in Fig. 1, the main structure of DI2IN is designed following a symmetric way as a convolutional encoder-decoder. All blocks in DI2IN consist of 3D convolutional and bilinear upscaling layers. The details of the network is described in Fig. 3.

In the encoder part of DI2IN, only the convolution layers are used in all blocks. In order to increase the receptive field of neurons and lower the GPU memory consumption, we set stride as 2 at some layers and reduce the size of feature maps. Moreover, larger receptive field covers more contextual information and helps to preserve liver shape information in the prediction. The decoder of DI2IN consists of convolutional and bilinear upscaling layers. To enble end-to-end prediction and training, the upscaling layers are implemented as bilinear interpolation to enlarge the activation maps. All convolutional kernels are $3 \times 3 \times 3$. The upscaling factor in decoder is 2 for x, y, z dimension. The leaky rectified linear unit (Leaky ReLU) and batch normalization are adopted in all convolutional layers for proper gradient back-propagation.

In order to further improve the performance of DI2IN, we adopt several mainstream technologies with the necessary changes [6,11,12]. First, we use the feature layer concatenation in DI2IN. Fast bridges are built directly from the encoder layers to the decoder layers. The bridges pass the information from the encoder forward and then concatenate it with the decoder feature layers. The combined feature is used as the input for the next convolution layer. Following the steps above to explicitly combine advanced and low-level features, DI2IN benefits from local and global contextual information. The deep supervision of the neural network during end-to-end training is shown to achieve good boundary detection and segmentation results. In the network, we introduced a more complex deep supervision scheme to improve performance. Several branches are separated from layers of the decoder section of main DI2IN. With the appropriate upscaling and convolution operations, the output size of each channel for all branches matches the size of the input image (Upscaling factors are 16, 4, 1 in block 10, 11, 12 respectively). By calculating the loss item l_i with the same ground truth data, the supervision is enforced at the end of each branch i. In order to further utilize the results of different branches, the final output is determined by the convolution operations of all branches with the leaky ReLU. During training, we apply binary cross entropy loss to each voxel of the output layers. The total loss l_{total} is the weighted combination of loss terms for all output layers, including the final output layer and the output layers for all branches, as follows:

$$l_{total} = \sum_i w_i \cdot l_i + w_{final} \cdot l_{final}$$

2.2 Network Improvement with Adversarial Training

We adopt the prevailing idea of the generative adversarial networks to boost the performance of DI2IN. The proposed scheme is shown in Fig. 2. An adversarial network is adopted to capture the high-order appearance information, which distinguishes between the ground truth and the output from DI2IN. In order to guide the generator to better prediction, the adversarial network provides an extra loss function for updating the parameters of generator during training. The purpose of the extra loss is to make the prediction as close as possible to the ground truth labeling. We adopt the binary cross-entropy loss for training of the

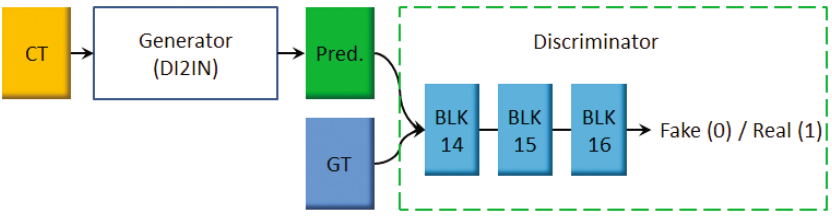

Fig. 2. Proposed adversarial training scheme. The generator produces the segmentation prediction, and discriminator classifies the prediction and ground truth during training.

adversarial network. D and G represent the discriminator and generator (DI2IN, in the context), respectively. For the discriminator $D\left(Y;\theta^D\right)$, the ground truth label Y_{gt} is assigned as one, and the prediction $Y_{pred} = G\left(X;\theta^G\right)$ is assigned as zero where X is the input CT volumes. The structure of discriminator network D is shown in Fig. 3. The following objective function is used in training the adversarial network:

$$\begin{aligned} l_D &= -\mathbb{E}_{y\sim p_{gt}} \log\left(D\left(y;\theta^D\right)\right) - \mathbb{E}_{y'\sim p_{pred}} \log\left(1 - D\left(y';\theta^D\right)\right) \\ &= -\mathbb{E}_{y\sim p_{gt}} \log\left(D\left(y;\theta^D\right)\right) - \mathbb{E}_{x\sim p_{data}} \log\left(1 - D\left(G\left(x;\theta^G\right);\theta^D\right)\right) \end{aligned} \tag{1}$$

During the training of network D, the gradient of loss l_D is propagated back to update the parameters of the generator network (DI2IN). At this stage, the loss for G has two components shown in the Eq. 2. The first component is the conventional segmentation loss l_b: voxel-wise binary cross-entropy between the prediction and ground truth. Minimizing the second loss component enables the discriminator D to confuse the ground truth with the prediction from G.

$$\begin{aligned} l_G &= \mathbb{E}_{y\sim p_{pred}, y'\sim p_{gt}}\left[l_{seg}\left(y,y'\right)\right] + \lambda\mathbb{E}_{y\sim p_{pred}} \log\left(1 - D\left(y;\theta^D\right)\right) \\ &= \mathbb{E}_{y\sim p_{pred}, y'\sim p_{gt}}\left[l_{seg}\left(y,y'\right)\right] + \lambda\mathbb{E}_{x\sim p_{data}} \log\left(1 - D\left(G\left(x;\theta^G\right);\theta^D\right)\right) \end{aligned} \tag{2}$$

Following suggestions in [9], we replace $-\log\left(1 - D\left(G\left(x\right)\right)\right)$ with $\log\left(D\left(G\left(X\right)\right)\right)$. In another word, we would like to maximize the probability that prediction to be the ground truth in Eq. 2, instead of minimizing the probability that prediction not to be the generated label map. Such replacement provides strong gradient during training of G and speed up the training process in practice.

$$l_G = \mathbb{E}_{y\sim p_{pred}, y'\sim p_{gt}}\left[l_{seg}\left(y,y'\right)\right] - \lambda\mathbb{E}_{x\sim p_{data}} \log D\left(G\left(x;\theta^G\right);\theta^D\right) \tag{3}$$

The generator and discriminator are trained alternatively for several times shown in Algorithm 1, until the discriminator is not able to easily distinguish between ground truth label and the output of DI2IN. After the training process, the adversarial network is no longer required at inference. The generator itself can provide high quality segmentation results and its performance is improved.

3 Experiments

Most public dataset for liver segmentation only consists of tens of cases. For example, the MICCAI-SLiver07 [13] dataset only contains 20 CT volumes for training and 10 CT volumes for testing. All the data are contrast enhanced. Such a small dataset is not suitable to show the power of CNN: it has been well known that neural network trained with more labelled data can usually achieve much better performance. Thus, in this paper, we collected more than 1000 CT volumes. The liver of each volume was delineated by human experts. These data covers large variations in populations, contrast phases, scanning ranges, pathologies, and field of view (FOV), etc. The inter-slice distance varies from

Algorithm 1. Adversarial training of generator and discriminator.

Input : pre-trained generator (DI2IN) with weights θ_0^G
Output: updated generator weights θ_1^G

for *number of training iterations* **do**
  for k_D *steps* **do**
    sample a mini-batch of training images $x \sim p_{data}$;
    generate prediction y_{pred} for x with $G\left(x;\theta_0^G\right)$;
    $\theta^D \leftarrow$ propagate back the stochastic gradient $\nabla l_D\left(y_{gt}, y_{pred}\right)$;
  end
  for k_G *steps* **do**
    sample a mini-batch of training images $x' \sim p_{data}$;
    generate y'_{pred} for x' with $G\left(x';\theta_0^G\right)$ and compute $D\left(G\left(x'\right)\right)$;
    $\theta_1^G \leftarrow$ propagate back the stochastic gradient $\nabla l_G\left(y'_{gt}, y'_{pred}\right)$;
  end
  $\theta_0^G \leftarrow \theta_1^G$
end

0.5 mm to 7.0 mm. All scans covers the abdominal regions but may extend to head and feet. Tumor can be found in multiple cases. The volumes may also have various other disease. For example, pleural effusion, which brights the lung region and changes the pattern of upper boundary of liver. Then we collected additional 50 volumes from clinical sites for the independent testing. The livers of these data were also annotated by human experts for the purpose of evaluation. We down-sampled the dataset into 3.0 mm resolution isotropically to speed up the processing and lower the consumption of computing memory without loss of accuracy. Training DI2IN from scratch takes 200 iterations using stochastic gradient descent with a batch size of 4 samples. The learning rate is 0.01 in the beginning and divided by 10 after 100 iterations. In the adversarial training (DI2IN-AN), we set λ to 0.01, and the number of training iterations is 100. For training D, k_D is 10 and the mini-batch size is 8. For training G, k_G is 1 and the mini-batch size is 4. Less training iterations are required for G than that for D because G is pre-trained before adversarial training. w_i is set as 1 in the loss.

Table 1 compares the performance of five different methods. The first method, the hierarchical, learning-based algorithm proposed in [5], was trained using 400 CT volumes. More training data did not show performance improvement for this method. For comparison purpose, the DI2IN network, which is similar to deep learning based algorithms proposed in [6–8] without post-processing steps, and the DI2IN-AN were trained using the same 400 cases. Both the DI2IN network and the DI2IN-AN were also trained using all 1000+ CT volumes. The average symmetric surface distance (ASD) and dice coefficients are computed for all methods on the test data. As shown in Table 1, DI2IN-AN achieves the best performance in both evaluation metrics. All deep learning based algorithms outperform the classic learning based algorithm with the hand-craft features, which shows the power of CNN. The results show that more training data enhances the

Block	Layer	*s*	*f*	Block	Layer	*s*	*f*	Block	Layer	*s*	*f*	Block	Layer	*s*	*f*
1	Conv.	1	16	6	Up.	2	-	10	Conv.	1	8	13	Conv.	1	1
	Conv.	2	16		Conv.	1	128		Conv.	1	8	14	Conv.	1	16
2	Conv.	1	32	7	Up.	2	-		Conv.	1	8		Conv.	2	16
	Conv.	2	32		Conv.	1	64		Conv.	1	1	15	Conv.	1	32
3	Conv.	1	64	8	Up.	2	-	11	Up.	4	-		Conv.	1	32
	Conv.	2	64		Conv.	1	32		Conv.	1	8		Conv.	2	32
4	Conv.	1	128	9	Up.	2	-		Conv.	1	8	16	Conv.	1	64
	Conv.	2	128		Conv.	1	16		Conv.	1	1		Conv.	1	64
5	Conv.	1	256		Up.	16	-	12	Up.	1	-		Conv.	1	64
	Conv.	1	256		Conv.	1	8		Conv.	1	1		Conv.	1	1

Fig. 3. Parametric setting of blocks in neural network. *s* stands for the stride, *f* is filter number. *Conv.* is convolution, and *Up.* is bilinear upscaling.

performance of both DI2IN and DI2IN-AN. Take DI2IN for example, training with 1000+ labelled data improves the mean ASD by 0.23 mm and the max ASD by 3.84 mm compared to training with 400 labelled data. Table 1 also shows that the adversarial structure can further boost the performance of DI2IN. The maximum ASD error is also reduced. Typical test samples are provided in Fig. 4. We also tried CRF and graph cut to refine the output of DI2IN. However, the results became worse, since a large portion of testing data had no contrast and the boundary of liver bottom at many locations was very fuzzy. CRF and graph cut both suffer from serious leakage in these situations. Using an NVIDIA TITAN X GPU and the Theano/Lasagne library, the run time of our algorithm is less than one second, which is significantly faster than most of the current approaches. For example, it requires 1.5 min for one case in [6]. More experimental results can be found in the supplementary material.

Our proposed DI2IN has clear advantages comparing with other prevailing methods. First, previous studies show that DI2IN, which incorporates the encoder-decoder structure, skip connections, and deep supervision scheme within one framework, has better structure design than U-Net or deep supervised network (DSN) for 3D volumetric datasets [6,12]. DI2IN is a different design from other prevailing networks, but it gathers the merits of them. Second, the CNN-based methods (no upsampling or deconvolution) are often time-consuming at inference, and their performance is sensitive to the selection of training sample. We examined the aforementioned networks with internal implementation, and DI2IN achieved better performance (20% improvement in terms of average symmetric surface distance).

Table 1. Comparison of five methods on 50 unseen CT data.

Method	ASD (mm)				Dice			
	Mean	Std	Max	Median	Mean	Std	Min	Median
Ling *et al.* (400) [5]	2.89	5.10	37.63	2.01	0.92	0.11	0.20	0.95
DI2IN (400)	2.25	1.28	10.06	2.0	0.94	0.03	0.79	0.94
DI2IN-AN (400)	2.00	0.95	7.82	1.80	0.94	0.02	0.85	0.95
DI2IN (1000)	2.15	0.81	6.51	1.95	0.94	0.02	0.87	0.95
DI2IN-AN (1000)	**1.90**	0.74	**6.32**	**1.74**	**0.95**	0.02	**0.88**	0.95

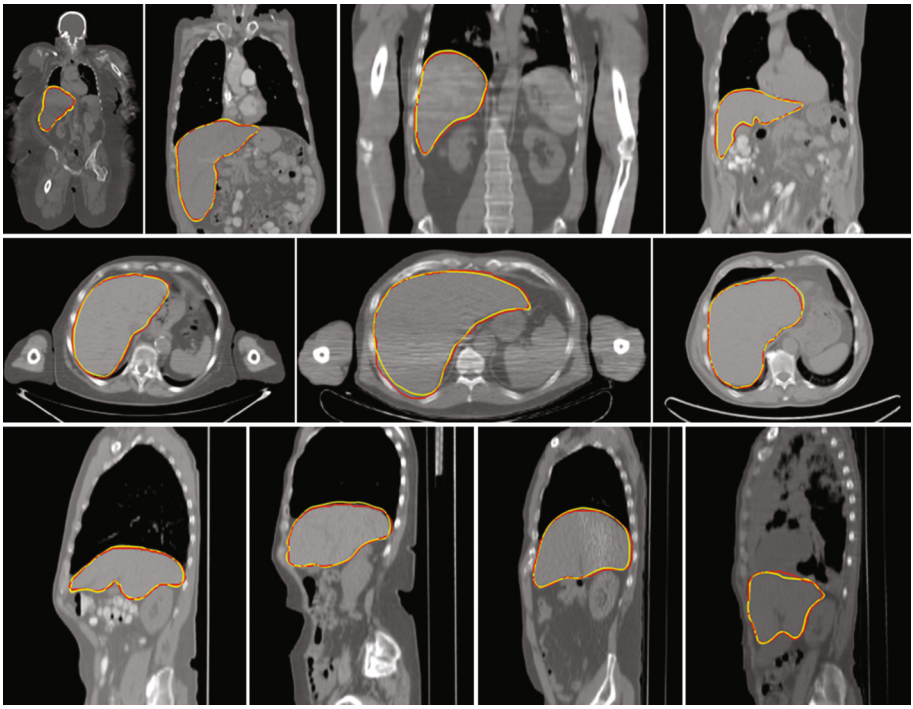

Fig. 4. Visual Results from different views. Yellow meshes are ground truth. Red ones are the prediction from DI2IN-AN.

4 Conclusion

In this paper, we proposed an automatic liver segmentation algorithm based on an adversarial image-to-image network. Our method achieves good segmentation quality as well as faster processing speed. The network is trained on an annotated dataset of 1000+ 3D CT volumes. We demonstrate that training with such a large dataset can improve the performance of CNN by a large margin.

References

1. Linguraru, M.G., Sandberg, J.K., Li, Z., Pura, J.A., Summers, R.M.: Atlas-based automated segmentation of spleen and liver using adaptive enhancement estimation. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 1001–1008. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04271-3_121
2. Kainmuller, D., Lange, T., Lamecker, H.: Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In: Proceedings of MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge, pp. 109–116 (2007)
3. Lee, J., Kim, N., Lee, H., Seo, J.B., Won, H.J., Shin, Y.M., Shin, Y.G., Kim, S.H.: Efficient liver segmentation using a level-set method with optimal detection of the initial liver boundary from level-set speed images. Comput. Methods Programs Biomed. **88**(1), 26–28 (2007)
4. Massoptier, L., Casciaro, S.: Fully automatic liver segmentation through graph-cut technique. In: Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2007), pp. 5243–5246 (2007)
5. Ling, H., Zhou, S.K., Zheng, Y., Georgescu, B., Suehling, M., Comaniciu, D.: Hierarchical, learning-based automatic liver segmentation. In: CVPR, pp. 1–8 (2008)
6. Dou, Q., Chen, H., Jin, Y., Yu, L., Qin, J., Heng, P.-A.: 3D deeply supervised network for automatic liver segmentation from CT volumes. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 149–157. Springer, Cham (2016). doi:10.1007/978-3-319-46723-8_18
7. Christ, P.F., et al.: Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 415–423. Springer, Cham (2016). doi:10.1007/978-3-319-46723-8_48
8. 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. **12**(2), 171–182 (2017)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS, pp. 2672–2680 (2014)
10. Luc, P., Couprie, C., Chintala, S.: Semantic Segmentation using Adversarial Networks. arXiv preprint. arXiv:1611.08408 (2016)
11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). doi:10.1007/978-3-319-24574-4_28
12. Merkow, J., Marsden, A., Kriegman, D., Tu, Z.: Dense volume-to-volume vascular boundary detection. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9902, pp. 371–379. Springer, Cham (2016). doi:10.1007/978-3-319-46726-9_43
13. Heimann, T., Van Ginneken, B., Styner, M.A., Arzhaeva, Y., Aurich, V., Bauer, C., Beck, A.: Comparison and evaluation of methods for liver segmentation from CT datasets. IEEE Trans. Med. Imaging **28**(8), 1251–1265 (2009)