Bridging the Gap Between 2D and 3D Organ Segmentation with Volumetric Fusion Net

Yingda Xia[1], Lingxi Xie[1(✉)], Fengze Liu[1], Zhuotun Zhu[1],
Elliot K. Fishman[2], and Alan L. Yuille[1]

[1] The Johns Hopkins University, Baltimore, MD 21218, USA
`198808xc@gmail.com`
[2] The Johns Hopkins University School of Medicine, Baltimore, MD 21287, USA

Abstract. There has been a debate on whether to use 2D or 3D deep neural networks for volumetric organ segmentation. Both 2D and 3D models have their advantages and disadvantages. In this paper, we present an alternative framework, which trains 2D networks on different viewpoints for segmentation, and builds a 3D **Volumetric Fusion Net** (VFN) to fuse the 2D segmentation results. VFN is relatively shallow and contains much fewer parameters than most 3D networks, making our framework more efficient at integrating 3D information for segmentation. We train and test the segmentation and fusion modules individually, and propose a novel strategy, named *cross-cross-augmentation*, to make full use of the limited training data. We evaluate our framework on several challenging abdominal organs, and verify its superiority in segmentation accuracy and stability over existing 2D and 3D approaches.

1 Introduction

With the increasing requirement of fine-scaled medical care, computer-assisted diagnosis (CAD) has attracted more and more attention in the past decade. An important prerequisite of CAD is an intelligent system to process and analyze medical data, such as CT and MRI scans. In the area of medical imaging analysis, organ segmentation is a traditional and fundamental topic [2]. Researchers often designed a specific system for each organ to capture its properties. In comparison to large organs (*e.g.*, the liver, the kidneys, the stomach, *etc.*), small organs such as the pancreas are more difficult to segment, which is partly caused by their highly variable geometric properties [9].

In recent years, with the arrival of the deep learning era [6], powerful models such as convolutional neural networks [7] have been transferred from natural image segmentation to organ segmentation. But there is a difference. Organ segmentation requires dealing with volumetric data, and two types of solutions have been proposed. The first one trains 2D networks from three orthogonal planes and fusing the segmentation results [9,17,18], and the second one suggests training a 3D network directly [4,8,19]. But 3D networks are more computationally

expensive yet less stable when trained from scratch, and it is difficult to find a pre-trained model for medical purposes. In the scenario of limited training data, fine-tuning a pre-trained 2D network [7] is a safer choice [14].

This paper presents an alternative framework, which trains 2D segmentation models and uses a light-weighted 3D network, named **Volumetric Fusion Net** (VFN), in order to fuse 2D segmentation at a late stage. A similar idea is studied before based on either the EM algorithm [1] or pre-defined operations in a 2D scenario [16], but we propose instead to construct generalized linear operations (convolution) and allow them to be learned from training data. Because it is built on top of reasonable 2D segmentation results, VFN is relatively shallow and does not use fully-connected layers (which contribute a large fraction of network parameters) to improve its discriminative ability. In the training process, we first optimize 2D segmentation networks on different viewpoints individually (this strategy was studied in [12,13,18]), and then use the validation set to train VFN. When the amount of training data is limited, we suggest a *cross-cross-augmentation* strategy to enable reusing the data to train both 2D segmentation and 3D fusion networks.

We first apply our system to a public dataset for pancreas segmentation [9]. Based on the state-of-the-art 2D segmentation approaches [17,18], VFN produces a consistent accuracy gain and outperforms other fusion methods, including majority voting and statistical fusion [1]. In comparison to 3D networks such as [19], our framework achieves comparable segmentation accuracy using fewer computational resources, *e.g.*, using 10% parameters and being 3× faster at the testing stage (it only adds 10% computation beyond the 2D baselines). We also generalize our framework to other small organs such as the adrenal glands and the duodenum, and verify its favorable performance.

2   Our Approach

2.1   Framework: Fusing 2D Segmentation into a 3D Volume

We denote an input CT volume by $\mathbf{X}$. This is a $W \times H \times L$ volume, where W, H and L are the numbers of voxels along the *coronal*, *sagittal* and *axial* directions, respectively. The i-th voxel of $\mathbf{X}$, x_i, is the intensity (Hounsfield Unit, HU) at the corresponding position, $i = (1, 1, 1), \ldots, (W, H, L)$. The ground-truth segmentation of an organ is denoted by $\mathbf{Y}^\star$, which has the same dimensionality as $\mathbf{X}$. If the i-th voxel belongs to the target organ, we set $y_i^\star = 1$, otherwise $y_i^\star = 0$. The goal of organ segmentation is to design a function $\mathbf{g}(\cdot)$, so that $\mathbf{Y} = \mathbf{g}(\mathbf{X})$, with all $y_i \in \{0, 1\}$, is close to $\mathbf{Y}^\star$. We measure the similarity between $\mathbf{Y}$ and $\mathbf{Y}^\star$ by the Dice-Sørensen coefficient (DSC): $\mathrm{DSC}(\mathbf{Y}, \mathbf{Y}^\star) = \frac{2 \times |\mathcal{Y} \cap \mathcal{Y}^\star|}{|\mathcal{Y}| + |\mathcal{Y}^\star|}$, where $\mathcal{Y}^\star = \{i \mid y_i^\star = 1\}$ and $\mathcal{Y} = \{i \mid y_i = 1\}$ are the sets of foreground voxels.

There are, in general, two ways to design $\mathbf{g}(\cdot)$. The first one trains a 3D model to deal with volumetric data directly [4,8], and the second one works by cutting the 3D volume into slices, and using 2D networks for segmentation. Both 2D and 3D approaches have their advantages and disadvantages. We appreciate the

ability of 3D networks to take volumetric cues into consideration (radiologists also exploit 3D information to make decisions), but, as shown in Sect. 3.2, 3D networks are sometimes less stable, arguably because we need to train all weights from scratch, while the 2D networks can be initialized with pre-trained models from the computer vision literature [7]. On the other hand, processing volumetric data (*e.g.*, 3D convolution) often requires heavier computation in both training and testing (*e.g.*, requiring $3\times$ testing time, see Table 1).

In mathematical terms, let $\mathbf{X}_l^A$, $l = 1, 2, \ldots, L$ be a 2D slice (of $W \times H$) along the *axial* view, and $\mathbf{Y}_l^A = \mathbf{s}^A(\mathbf{X}_l^A)$ be the segmentation score map for $\mathbf{X}_l^A$. $\mathbf{s}^A(\cdot)$ can be a 2D segmentation network such as FCN [7], or a multi-stage system such as a coarse-to-fine framework [18]. Stacking all $\mathbf{Y}_l^A$'s yields a 3D volume $\mathbf{Y}^A = \mathbf{s}^A(\mathbf{X})$. This slicing-and-stacking process can be performed along each axis independently. Due to the large image variation in different views, we train three segmentation models, denoted by $\mathbf{s}^C(\cdot)$, $\mathbf{s}^S(\cdot)$ and $\mathbf{s}^A(\cdot)$, respectively. Finally, a fusion function $\mathbf{f}[\cdot]$ integrates them into the final prediction:

$$\mathbf{Y} = \mathbf{f}\big[\mathbf{X}, \mathbf{Y}^C, \mathbf{Y}^S, \mathbf{Y}^A\big] = \mathbf{f}\big[\mathbf{X}, \mathbf{s}^C(\mathbf{X}), \mathbf{s}^S(\mathbf{X}), \mathbf{s}^A(\mathbf{X})\big]. \tag{1}$$

Note that we allow the image $\mathbf{X}$ to be incorporated. This is related to the idea known as auto-contexts [15] in computer vision. As we shall see in experiments, adding $\mathbf{X}$ improves the quality of fusion considerably. Our goal is to equip $\mathbf{f}[\cdot]$ with partial abilities of 3D networks, *e.g.*, learning simple, local 3D patterns.

2.2   Volumetric Fusion Net

The VFN approach is built upon the 2D segmentation volumes from three orthogonal (*coronal*, *sagittal* and *axial*) planes. Powered by state-of-the-art deep networks, these results are generally accurate (*e.g.*, an average DSC of over 82% [18] on the NIH pancreas segmentation dataset [9]). But, as shown in Fig. 2, some *local* errors still occur because 2 out of 3 views fail to detect the target. Our assumption is that these errors can be recovered by learning and exploiting the 3D image patterns in its surrounding region.

Regarding other choices, majority voting obviously cannot take image patterns into consideration. The STAPLE algorithm [1], while being effective in multi-atlas registration, does not have a strong ability of fitting image patterns from training data. We shall see in experiments that STAPLE is unable to improve segmentation accuracy over majority voting.

Motivated by the need to learn *local* patterns, we equip VFN with a small input region (64^3) and a shallow structure, so that each neuron has a small receptive field (the largest region seen by an output neuron is 50^3). In comparison, in the 3D network VNet [8], these numbers are 128^3 and 551^3, respectively. This brings twofold benefits. First, we can sample more patches from the training data, and the number of parameters is much less, and so the risk of over-fitting is alleviated. Second, VFN is more computationally efficient than 3D networks, *e.g.*, adding 2D segmentation, it needs only half the testing time of [19].

The architecture of VFN is shown in Fig. 1. It has three down-sampling stages and three up-sampling stages. Each down-sampling stage is composed of two
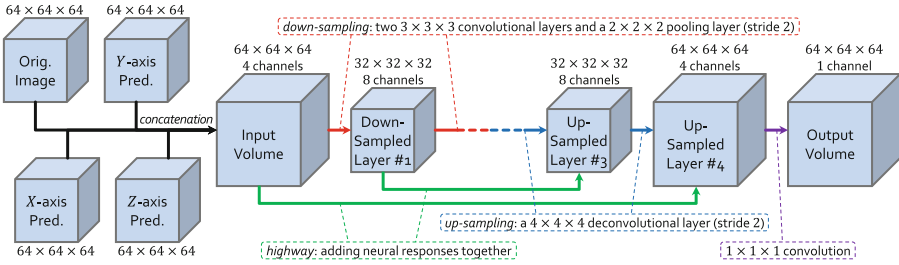
Fig. 1. The network structure of VFN. We only display one down-sampling and one up-sampling stages, but there are 3 of each. Each down-sampling stage shrinks the spatial resolution by 1/2 and doubles the number of channels. We build 3 highway connections (2 are shown). We perform batch normalization and ReLU activation after each convolutional and deconvolutional layer.

$3 \times 3 \times 3$ convolutional layers and a $2 \times 2 \times 2$ max-pooling layer with a stride of 2, and each up-sampling stage is implemented by a single $4 \times 4 \times 4$ deconvolutional layer with a stride of 2. Following other 3D networks [8,19], we also build a few residual connections [5] between hidden layers of the same scale. For our problem, this enables the network to preserve a large fraction of 2D network predictions (which are generally of good quality) and focus on refining them (note that if all weights in convolution are identical, then VFN is approximately equivalent to majority voting). Experiments show that these highway connections lead to faster convergence and higher accuracy. A final convolution of a $1 \times 1 \times 1$ kernel reduces the number of channels to 1.

The input layer of VFN consists of 4 channels, 1 for the original image and 3 for 2D segmentations from different viewpoints. The input values in each channel are normalized into $[0, 1]$. By this we provide equally-weighted information from the original image and 2D multi-view segmentation results, so that VFN can fuse them at an early stage and learn from data automatically. We verify in experiments that image information is important – training a VFN without this input channel shrinks the average accuracy gain by half.

2.3 Training and Testing VFN

We train VFN from scratch, *i.e.*, all weights in convolution are initialized as random white noises. Note that setting all weights as 1 mimics majority voting, and we find that both ways of initialization lead to similar testing performance. All $64 \times 64 \times 64$ volumes are sampled from the region-of-interest (ROI) of each training case, defined as the bounding box covering all foreground voxels padded by 32 pixels in each dimension. We introduce data augmentation by performing random 90°-rotation and flip in 3D space (each cube has 24 variants). We use a Dice loss to avoid background bias (a voxel is more likely to be predicted as background, due to the majority of background voxels in training). We train VFN for 30,000 iterations with a mini-batch size of 16. We start with a learning

Table 1. Comparison of segmentation accuracy (DSC, %) and testing time (in minutes) between our approach and the state-of-the-arts on the NIH dataset [9]. Both [18] and [17] are reimplemented by ourselves, and the default fusion is majority voting.

Approach	Average	Min	1/4-Q	Med	3/4-Q	Max	Time (m)
Roth *et al.* [9]	71.42 ± 10.11	23.99	–	–	–	86.29	6–8
Roth *et al.* [10]	78.01 ± 8.20	34.11	–	–	–	88.65	2–3
Roth *et al.* [11]	81.27 ± 6.27	50.69	–	–	–	88.96	2–3
Cai *et al.* [3]	82.4 ± 6.7	60.0	–	–	–	90.1	N/A
Zhu *et al.* [19]	84.59 ± 4.86	69.62	–	–	–	91.45	4.1
Zhou *et al.* [18]	82.50 ± 6.14	56.33	81.63	84.11	86.28	89.98	0.9
[18] + NLS	82.25 ± 6.57	56.86	81.54	83.96	86.14	89.94	1.1
[18] + VFN	$\mathbf{84.06} \pm 5.63$	62.93	81.98	85.69	87.62	91.28	1.0
Yu *et al.* [17]	84.48 ± 5.03	62.23	82.50	85.66	87.82	91.17	1.3
[17] + NLS	84.47 ± 5.03	62.22	82.42	85.59	87.78	91.17	1.5
[17] + VFN	$\mathbf{84.63} \pm 5.07$	61.58	82.42	85.84	88.37	91.57	1.4

rate of 0.01, and divide it by 10 after 20,000 and 25,000 iterations, respectively. The entire training process requires approximately 6 h in a Titan-X-Pascal GPU. In the testing process, we use a sliding window with a stride of 32 in the ROI region (the minimal 3D box covering all foreground voxels of multi-plane 2D segmentation fused by majority voting). For an average pancreas in the NIH dataset [9], testing VFN takes around 5 s.

An important issue in optimizing VFN is to construct the training data. Note that we cannot reuse the data used for training segmentation networks to train VFN, because this will result in the input channels contain very accurate segmentation, which limits VFN from learning meaningful local patterns and generalizing to the testing scenarios. So, we further split the training set into two subsets, one for training the 2D segmentation networks and the other for training VFN with the testing segmentation results.

However, under most circumstances, the amount of training data is limited. For example, in the NIH pancreas segmentation dataset, each fold in cross-validation has only 60 training cases. Partitioning it into two subsets harms the accuracy of both 2D segmentation and fusion. To avoid this, we suggest a **cross-cross-augmentation** (CCA) strategy, described as follows. Suppose we split data into K folds for cross-validation, and the k_1-th fold is left for testing. For all $k_2 \neq k_1$, we train 2D segmentation models on the folds in $\{1, 2, \ldots, K\} \setminus \{k_1, k_2\}$, and test on the k_2-th fold to generate training data for the VFN. In this way, all data are used for training both the segmentation model and the VFN. The price is that a total of $K(K-1)/2$ extra segmentation models need to be trained, which is more costly than training K models in a standard cross-validation. In practice, this strategy improves the average segmentation accuracy by ∼1% in each fold. Note that we perform CCA only on the NIH dataset due to the limited

amount of data – in our own dataset, we perform standard training/testing split, requiring <10% extra training time and ignorable extra testing time.
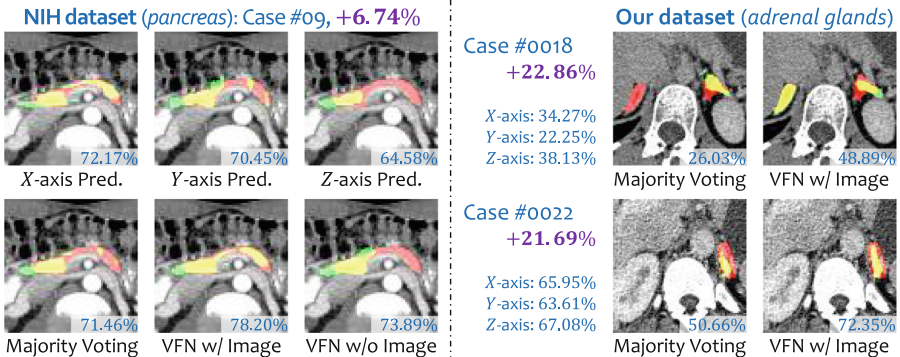
3 Experiments

3.1 The NIH Pancreas Segmentation Dataset

We first evaluate our approach on the NIH pancreas segmentation dataset [9] containing 82 abdominal CT volumes. The width and height of each volume are both 512, and the number of slices along the *axial* axis varies from 181 to 466. We split the dataset into 4 folds of approximately the same size, and apply cross-cross-augmentation (see Sect. 2.3) to improve segmentation accuracy.

Results are summarized in Table 1. We use two recent 2D segmentation approaches as our baseline, and compare VFN with two other fusion approaches, namely majority voting and non-local STAPLE (NLS) [1]. The latter was verified more effective than its former local version. We measure segmentation accuracy using DSC and report the average accuracy over 82 cases. Based on [18], VFN improves majority voting significantly by an average of 1.69%. The improvement over 82 cases is consistent (the student's t-test reports a p-value of 6.9×10^{-7}), although the standard deviation over 82 cases is relatively large – this is mainly caused by the difference in difficulties from case to case. Figure 2 shows an example on which VFN produces a significant accuracy gain. VFN does not improve [17] significantly, arguably because [17] has almost reached the human-level agreement (we invited a radiologist to segment this dataset individually, and she achieves an average accuracy of ∼86%). Note that the other approaches without CCA used both the training and validation folds for training, and so all numbers are comparable in Table 1.

Due to our analysis in Sect. 2.2, NLS does not produce any accuracy gain over either [18] and [17]. NLS is effective in multi-atlas registration, where the

NIH dataset (*pancreas*): Case #09, **+6.74**%

72.17% — *X*-axis Pred.
70.45% — *Y*-axis Pred.
64.58% — *Z*-axis Pred.
71.46% — Majority Voting
78.20% — VFN w/ Image
73.89% — VFN w/o Image

Our dataset (*adrenal glands*)

Case #0018 +**22.86**%
X-axis: 34.27%
Y-axis: 22.25%
Z-axis: 38.13%
26.03% — Majority Voting
48.89% — VFN w/ Image

Case #0022 +**21.69**%
X-axis: 65.95%
Y-axis: 63.61%
Z-axis: 67.08%
50.66% — Majority Voting
72.35% — VFN w/ Image

Fig. 2. Two typical examples, each with the original image, segmentation results from three viewpoints, and different fusion results. In each label map, red, green and yellow indicate ground-truth, prediction and overlap, respectively.

labels come from different images and the annotation is relatively accurate [1]. But in our problem, segmentation results from 2D networks can be noisy, thus recovering these errors requires learning local image patterns from training data, which is what VFN does to outperform NLS.

To reveal the importance of image information, we train a VFN without the image channel in the input layer. Based on [18], this version produces approximately half of the improvement (1.69%) by the full model. We show an example in Fig. 2, in which the right part of the pancreas is missing in both *sagittal* and *axial* planes, but the high confidence in the *coronal* plane and the continuity of image intensities suggest its presence in the final segmentation.

3.2   Our Multi-organ Dataset

The radiologists in our team collected a dataset with 300 high-resolution CT scans. These scans were performed on some potential renal donors. Four experts in abdominal anatomy annotated 11 abdominal organs, taking 3−4 h for each scan, and all annotations were verified by an experienced board certified Abdominal Radiologist. Except for the *pancreas*, we choose several challenging targets, including the *adrenal glands*, the *duodenum*, and the *gallbladder* (easy cases such as the *liver* and the *kidneys* are not considered). We use 150 cases for training 2D segmentation models, 100 cases for training VFN, and test on the remaining 50 cases. The data split is random but identical for different organs.

Results are shown in Table 2. Again, our approach consistently improves 2D segmentation, which demonstrates the transferability of our methodology. In *pancreas*, based on [17], we obtain a *p*-value of 2.7×10^{-5} over 50 testing cases. In *adrenal glands*, although the average accuracy gains are not large, the improvement is significant in some badly segmented cases, *e.g.*, Fig. 2 shows two examples with more than 20% accuracy boosts. Refining bad segmentations makes our segmentation results more reliable. By contrast, the 3D network [19] produces unstable performance ([19] was designed for pancreas segmentation, thus works reasonably well in *pancreas*), which is mainly caused by the limited training data especially for small organs such as *adrenal glands* and *gallbladder*.

Table 2. Comparison of segmentation accuracy (DSC, %) on our multi-organ dataset. The baseline for [18] and [17] is majority voting. The numbers of [17] are different from those in their original paper, because we are using a different dataset.

Approach	Adrenal g.	Duodenum	Gallbladder	Pancreas
Zhu *et al.* [19]	36.74 ± 25.14	68.80 ± 14.38	42.01 ± 29.47	85.25 ± 6.04
Zhou *et al.* [18]	66.09 ± 18.19	71.65 ± 13.15	90.39 ± 5.30	84.52 ± 6.23
[18] + VFN	$\mathbf{69.24} \pm 17.42$	$\mathbf{72.77} \pm 12.80$	$\mathbf{91.40} \pm 5.19$	$\mathbf{86.39} \pm 6.20$
Yu *et al.* [17]	71.40 ± 12.87	77.48 ± 8.70	91.81 ± 4.90	87.22 ± 5.90
[17] + VFN	$\mathbf{72.09} \pm 13.61$	$\mathbf{77.77} \pm 8.46$	$\mathbf{92.15} \pm 5.05$	$\mathbf{88.06} \pm 5.33$

Therefore, we conclude that 2D segmentation followed by 3D fusion is currently a very promising idea to bridge the gap between 2D and 3D segmentation approaches, particularly if there is limited training data.

4   Conclusions

In this paper, we discuss an important topic in medical imaging analysis, namely bridging the gap between 2D and 3D organ segmentation approaches. We propose to train more stable 2D segmentation networks, and then use a light-weighted 3D fusion module to fuse their results. In this way, we enjoy the benefits of exploiting 3D information to improve segmentation, as well as avoiding the risk of over-fitting caused by tuning 3D models (which have $10\times$ more parameters) on a limited amount of training data. We verify the effectiveness of our approach on two datasets, one of which contains several challenging organs.

Based on our work, a promising direction is to train the segmentation and fusion modules in a joint manner, so that the 2D networks can incorporate 3D information in the training process by learning from the back-propagated gradients of VFN. Another issue involves training VFN more efficiently, *e.g.*, using hard example mining. These topics are left for future research.

References

1. Asman, A.J., Landman, B.A.: Non-local statistical label fusion for multi-atlas segmentation. Med. Image Anal. **17**(2), 194–208 (2013)
2. Boykov, Y., Jolly, M.-P.: Interactive organ segmentation using graph cuts. In: Delp, S.L., DiGoia, A.M., Jaramaz, B. (eds.) MICCAI 2000. LNCS, vol. 1935, pp. 276–286. Springer, Heidelberg (2000). https://doi.org/10.1007/978-3-540-40899-4_28
3. Cai, J., Lu, L., Xie, Y., Xing, F., Yang, L.: Pancreas segmentation in MRI using graph-based decision fusion on convolutional neural networks. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10435, pp. 674–682. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66179-7_77
4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
7. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)

8. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016)
9. Roth, H.R., et al.: Deeporgan: multi-level deep convolutional networks for automated pancreas segmentation. In: MICCAI (2015)
10. Roth, H.R., Lu, L., Farag, A., Sohn, A., Summers, R.M.: Spatial aggregation of holistically-nested networks for automated pancreas segmentation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 451–459. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_52
11. Roth, H.R., et al.: Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation. arXiv:1702.00045 (2017)
12. Setio, A.A.A., et al.: Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. IEEE Trans. Med. Imaging **35**(5), 1160–1169 (2016)
13. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 945–953 (2015)
14. Tajbakhsh, N., et al.: Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE TMI **35**(5), 1299–1312 (2016)
15. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3D brain image segmentation. IEEE TPAMI **32**(10), 1744–1757 (2010)
16. Yang, H., Sun, J., Li, H., Wang, L., Xu, Z.: Deep fusion net for multi-atlas segmentation: application to cardiac MR images. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 521–528. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_60
17. Yu, Q., Xie, L., Wang, Y., Zhou, Y., Fishman, E.K., Yuille, A.L.: Recurrent saliency transformation network: incorporating multi-stage visual cues for small organ segmentation. arXiv:1709.04518 (2017)
18. Zhou, Y., Xie, L., Shen, W., Wang, Y., Fishman, E.K., Yuille, A.L.: A fixed-point model for pancreas segmentation in abdominal CT scans. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 693–701. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_79
19. Zhu, Z., Xia, Y., Shen, W., Fishman, E.K., Yuille, A.L.: A 3D coarse-to-fine framework for automatic pancreas segmentation. arXiv:1712.00201 (2017)