Pancreas Segmentation in Abdominal CT Scan: A Coarse-to-Fine Approach

Article · December 2016

DOI: 10.48550/arXiv.1612.08230

5 authors, including:

Yuyin Zhou
Johns Hopkins University
91 PUBLICATIONS **5,560** CITATIONS

SEE PROFILE

Wei Shen
Shanghai University
119 PUBLICATIONS **7,568** CITATIONS

SEE PROFILE

Elliot K Fishman
Johns Hopkins Medicine
2,123 PUBLICATIONS **71,958** CITATIONS

SEE PROFILE

Pancreas Segmentation in Abdominal CT Scan: A Coarse-to-Fine Approach

Yuyin Zhou[1], Lingxi Xie[2], Wei Shen[3], Elliot Fishman[4] and Alan Yuille[5]

[1,2,5]Center for Imaging Science, the Johns Hopkins University
[3]School of Communications and Information Engineering, Shanghai University
[4]Radiology and Radiological Science, the Johns Hopkins Medical Institutions
[1]shirley3452010@hotmail.com [2]198808xc@gmail.com [3]wei.shen@t.shu.edu.cn
[4]alan.l.yuille@gmail.com [5]efishman@jhmi.edu

Abstract. Deep neural networks have been widely adopted for automatic organ segmentation from CT-scanned images. However, the segmentation accuracy on some small organs (*e.g.*, the pancreas) is sometimes below satisfaction, arguably because deep networks are easily distracted by the complex and variable background region which occupies a large fraction of the input volume.

In this paper, we propose a coarse-to-fine approach to deal with this problem. We train two deep neural networks using different regions of the input volume. The first one, the *coarse-scaled* model, takes the entire volume as its input. It is used for roughly locating the spatial position of the pancreas. The second one, the *fine-scaled* model, only sees a small input region covering the pancreas, thus eliminating the background noise and providing more accurate segmentation especially around the boundary areas. At the testing stage, we first use the coarse-scaled model to roughly locate the pancreas, then adopt the fine-scaled model to refine the initial segmentation in an iterative manner to obtain increasingly better segmentation. We evaluate our algorithm on the NIH pancreas segmentation dataset with 82 volumes, and outperform the state-of-the-art [18] by more than 4%, measured by the Dice-Sørensen Coefficient (DSC). In addition, we report 62.43% DSC for our worst case, significantly better than 34.11% reported in [18].

Keywords: Pancreas Segmentation, A Coarse-to-Fine Approach

1 Introduction

In recent years, due to the fast development of deep neural networks, we have witnessed rapid progress in both medical image analysis and computer-aided diagnosis. This paper focuses on a specific task in this research field, namely to automatically segment small organs (*e.g.*, the pancreas) from CT-scanned images. This is a difficult task, as the pancreas often suffers high variety in shape, size and location [19], while occupying only a very small fraction (*e.g.*, $< 0.5\%$) of the entire CT volume. In such cases, deep neural networks can be distracted by the background region, which occupies a large fraction of the input volume

and may include complex and variable contents. Consequently, the segmentation result becomes less accurate especially around the boundary areas.

To deal with this problem, we propose a coarse-to-fine approach for pancreas segmentation. Starting with a baseline approach, *i.e.*, a *coarse-scaled* model which takes the entire volume as the input, we train an additional *fine-scaled* model on a small region covering the pancreas, so as to alleviate the background noise. We find that the latter model works much better, especially in detecting details on the boundary areas. At the testing stage, we first use the coarse-scaled model to roughly detect the pancreas, and then apply the fine-scaled model to refine the initial segmentation in an iterative manner. On a modern GPU, we need around 3 minutes to process a CT volume during the testing stage. This is comparable to [18], but we report higher accuracy.

We train and evaluate our algorithm on the NIH pancreas segmentation dataset [19]. Following [19], we partition the 82 samples into 4 folds and evaluate by cross-validation. Compared to recently published work [18], our average segmentation accuracy, measured by the Dice-Sørensen Coefficient (DSC), increases from 78.01% to 82.37%. Meanwhile, we report 62.43% DSC on the worst case, which guarantees reasonable performance on the particularly challenging test samples. In comparison, [18] reports 34.11% DSC on the worse case and [19] reports 23.99%. Some previous methods [4][26] report even lower numbers. We point out that, although our algorithm is only tested on a pancreas dataset, the approach is directly generalizable to other organ segmentation tasks, especially for those small organs such as the spleen or the duodenum.

The remainder of this paper is organized as follows. Section 2 briefly introduces related work, and Section 3 describes the algorithm. Experiments are shown in Section 4, and we conclude our work in Section 5.

2 Related Work

Contrast-enhanced computed tomography (CECT) is a popular way of producing detailed images of internal organs, bones, soft tissues and blood vessels. Based on this technology, computer-aided diagnosis (CAD) is widely studied as a tool to assist physicians. Automatic segmentation is an important prerequisite of a CAD system [25][8]. The difficulty mainly comes from the high anatomical variability and/or the small volume of the target organs. Indeed researchers sometimes design a specific segmentation algorithm for each organ [1][19][23].

Over the past few years, the rapid development of deep learning has led to a revolution to medical image analysis. The most successful methods are based on the deep convolutional neural network (CNN), a hierarchical model which is learning complicated data distributions. In the computer vision field, deep learning has been widely used for image classification [11][24][20], object detection [7] and instance segmentation [13]. These neural network methods can be transferred to medical image analysis. But there is one challenge. CT-scanned data take the form of 3D volumes, and not 2D images. There are basically two types of solution, *i.e.*, using a 2D-based network to process 3D data, or training

a 3D-based network directly. A straightforward idea borrowed from computer vision is to cut the 3D volume into slices (2D images) and train a 2D model on these. These models can be trained in a patch-based manner [5][19], or directly applied to the full images [18]. To consider and exploit the underlying relationship between neighboring slices, 3D-based networks are also widely studied. Due to their heavier computational overhead, 3D models are often trained in a patch-based manner [17][10][14][15]. A more efficient solution is to integrate 3D cues to 2D models, for example by using a tri-planar structure (three orthogonal planes intersecting at the point to be classified) [16] or interpreting the 3D volume as a sequence of 2D image slices and applying recurrent neural networks [22][3]. A detailed discussion about 2D and 3D models can be found in [6][12].

In this paper, we choose fully-convolutional network (FCN) [13], a 2D segmentation model based on a pre-trained deep neural network [20], as the baseline model. Deeper and/or more sophisticated networks can lead to higher segmentation accuracy [2][15], but they are often more computationally expensive and risk over-fitting [14]. As we focus on segmenting small organs (*e.g.*, the pancreas), the standard loss function computed per pixel/voxel may cause the model be heavily biased towards predicting a pixel/voxel to fall on background. Solutions include performing class-balancing [18], or directly designing a new loss function which is consistent to the evaluation metric [15]. The latter often works better.

3 The Proposed Approach

3.1 The Baseline Framework

We use the 2D fully-convolutional network (FCN) [13] as our baseline. FCN inherits the down-sampling process of popular networks for image classification, and then applies an up-sampling process named deconvolution to restore the output to the original size. Throughout this paper, we specify the down-sampling layers using a pre-trained 16-layer VGGNet [20]. We do not use deeper network structures like [9], which produce slightly higher accuracy at the expense of heavier computation and a higher risk of over-fitting.

Let a CT-scanned image be a 3D volume $\mathbf{X}$ of size $W \times H \times L$, where W, H and L denote the width, height and length, respectively. Each volume is annotated with a ground-truth segmentation $\mathbf{Y}$ of the same dimensionality, where $Y_i = 1$ indicates a foreground voxel. FCN is a 2D segmentation model $\mathbb{M} : \mathbf{O} = \mathbf{f}(\mathbf{I}; \boldsymbol{\Theta})$, where $\mathbf{I}$ denotes the input image, $\boldsymbol{\Theta}$ the weights of the network, and $\mathbf{O}$ the output segmentation result which has the same spatial resolution as $\mathbf{I}$. To fit the 3D volume $\mathbf{X}$ into a 2D network $\mathbb{M}$, we cut it into a set of 2D slices. This process can be performed along three axes, *i.e.*, in the coronal, sagittal and axial views. We denote these 2D slices as $\mathbf{x}_{\mathrm{C},w}$ $(w = 1, 2, \ldots, W)$, $\mathbf{x}_{\mathrm{S},h}$ $(h = 1, 2, \ldots, H)$ and $\mathbf{x}_{\mathrm{A},l}$ $(l = 1, 2, \ldots, L)$, where the subscripts C, S and A stand for "coronal", "sagittal" and "axial", respectively. We train three FCN models $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$ to perform segmentation through three views individually and integrate these cues at the testing stage.

Simply cutting a 3D volume into 2D slices ignores rich information, *e.g.*, the correlation of structure between the neighboring slices. Motivated by this, we propose a 3-slice-segmentation model, which makes predictions on 3 successive slices simultaneously. In this case, each input image is composed of 3 successive slices, *e.g.*, the w-th 2D image on the coronal view contains 3 slices (channels), *i.e.*, $\mathbf{x}_{\mathrm{C},w-1}$, $\mathbf{x}_{\mathrm{C},w}$ and $\mathbf{x}_{\mathrm{C},w+1}$. Thus, each 2D slice can appear in at most 3 input images. Applying this modification only slightly increases the number of network parameters, but provides us with the opportunity of incorporating visual cues on the neighboring slices during prediction. We can certainly process more slices at the same time, but in practice, this does not yield much accuracy gain.

The original FCN model uses a loss function to sum up the cross-entropy loss at each voxel. For a training datum $(\mathbf{x}, \mathbf{y})$ and prediction $\mathbf{p} \doteq \mathbf{f}(\mathbf{x}; \boldsymbol{\Theta})$, the loss is computed as $\mathcal{L}_{\mathrm{voxel}} = \sum_i \left(-y_i \log p_i - (1-y_i)\log(1-p_i)\right)$, where i sums through all the voxels in $\mathbf{y}$ and $\mathbf{p}$. However, as the pancreas often occupies a very small fraction of each slice, the voxel-wise loss is heavily biased to predicting a voxel as non-target (*e.g.*, a simple model guesses that all voxels are background gets $> 98\%$ voxel-wise accuracy), which results in terrible performance as measured by the Dice-Sørensen Coefficient (DSC). To avoid this, we follow [15] to directly design a DSC-loss layer. The DSC between the ground-truth set $\mathcal{G}$ and the prediction set $\mathcal{A}$ is defined by $\mathrm{DSC}(\mathcal{A}, \mathcal{G}) = \frac{2\times|\mathcal{A}\cap\mathcal{G}|}{|\mathcal{A}|+|\mathcal{G}|}$. Suppose the ground-truth annotation of each voxel is $y_i \in \{0, 1\}$ where 1 indicates the target, and the prediction value of each voxel is $p_i \in [0, 1]$, the DSC-loss function can be computed as $\mathcal{L}_{\mathrm{DSC}} = 1 - \mathrm{DSC}(\mathbf{p}, \mathbf{y}) = 1 - \frac{2\times\sum_i p_i y_i}{\sum_i p_i + \sum_i y_i}$. Note that this function is equivalent to the original DSC function if $p_i \in \{0, 1\}$ for all i. The gradient computation is straightforward: $\frac{\partial \mathcal{L}_{\mathrm{DSC}}}{p_j} = -2 \times \frac{y_j\left(\sum_i p_i + \sum_i y_i\right) - \sum_i p_i y_i}{\left(\sum_i p_i + \sum_i y_i\right)^2}$.

3.2 The Fine-Tuning Approach

We focus on segmenting small organs (*e.g.*, the pancreas), which often occupy a very small part (*e.g.*, $< 0.5\%$) of the CT volume. It was observed [19][21] that deep segmentation networks such as FCN produce less satisfying results in these scenarios, arguably because the network is easily distracted by the varying contents in background regions, *e.g.*, numerical instability caused by different vessel-contrast in the CECT scanning. In other words, a considerable number of neurons in the deep network are trained to detect the rough position of the pancreas, and thus the ability to precisely capture its shape and appearance becomes weak. Figure 1 shows the comparison of segmentation results when the input is either the entire image or a small area around the pancreas. We see that the latter strategy leads to more accurate segmentation.

Motivated by this, we propose a fine-tuning approach to polish the initial segmentation. The idea is straightforward. In addition to the coarse-scaled deep network which sees the entire image, we train an additional fine-scaled deep network which only sees a small region covering the pancreas. In the training phase, this is done by taking the ground-truth annotation of each slice, and

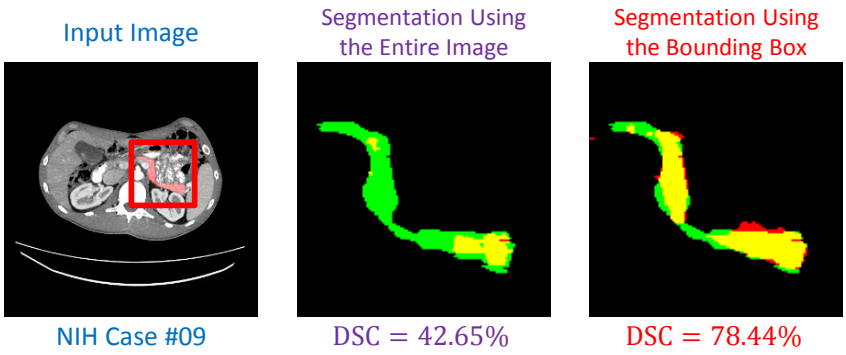

Fig. 1. The comparison of segmentations with different input regions (best viewed in color PDF), namely using the entire image or the bounding box (marked by the red frame). The right images show the segmentations within the frame, in which red, green and yellow indicate the prediction, ground-truth and overlapped regions, respectively.

add a small frame around the minimal bounding box covering the segmentation mask. Note that, the input image size may differ from case to case. In the testing phase, we first feed the entire image to the coarse-scaled deep network and obtain the initial segmentation. Based on this, we estimate the bounding box, add a frame of a fixed width, crop the image region accordingly, and feed it to the fine-scaled models. Note that this fine-tuning process can be performed in an iterative manner at the testing stage without training new models. In practice, this process often terminates very quickly (*e.g.* in 2–3 iterations).

We illustrate the overall flowchart of our approach in Figure 2. We denote the coarse-scaled models trained on the coronal, sagittal and axial views as $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$, respectively, and three fine-scaled models as $\mathbb{M}_{\mathrm{C}}^{\mathrm{F}}$, $\mathbb{M}_{\mathrm{S}}^{\mathrm{F}}$ and $\mathbb{M}_{\mathrm{A}}^{\mathrm{F}}$, where the superscript F stands for "fine-tuning". When a testing volume $\mathbf{X}$ is given, we first send it through three coarse-scaled models, and fuse their results into the initial segmentation $\mathbf{P}^{(0)} = \frac{1}{3}\left[\mathbf{P}_{\mathrm{C}}^{(0)} + \mathbf{P}_{\mathrm{S}}^{(0)} + \mathbf{P}_{\mathrm{A}}^{(0)}\right]$, where $\mathbf{P}_{\mathrm{C}}^{(0)}$, $\mathbf{P}_{\mathrm{S}}^{(0)}$ and $\mathbf{P}_{\mathrm{A}}^{(0)}$ are the probabilistic outputs from three different views. We compute the initial segmentation mask $\mathcal{S}^{(0)}$ by thresholding $\mathbf{P}^{(0)}$ with 0.5. Then the fine-tuning process is performed iteratively. At the t-th iteration, the segmentation result of the previous iteration, *i.e.*, $\mathcal{S}^{(t-1)}$ is used to estimate the current 3D bounding box, and the volume within the 3D bounding box is cropped, framed and denoted as $\mathbf{X}^{(t)}$, and fed into the fine-scaled models. The results of three fine-scaled models are fused, *i.e.*, $\mathbf{P}^{(t)} = \frac{1}{3}\left[\mathbf{P}_{\mathrm{C}}^{(t)} + \mathbf{P}_{\mathrm{S}}^{(t)} + \mathbf{P}_{\mathrm{A}}^{(t)}\right]$, and the segmentation mask $\mathcal{S}^{(t)}$ is updated. This process terminates when the maximal number of iterations T is reached, or the the similarity between successive segmentation results is sufficiently high, *e.g.*, when $\mathrm{DSC}\big(\mathcal{S}^{(t-1)}, \mathcal{S}^{(t)}\big) > 0.95$.

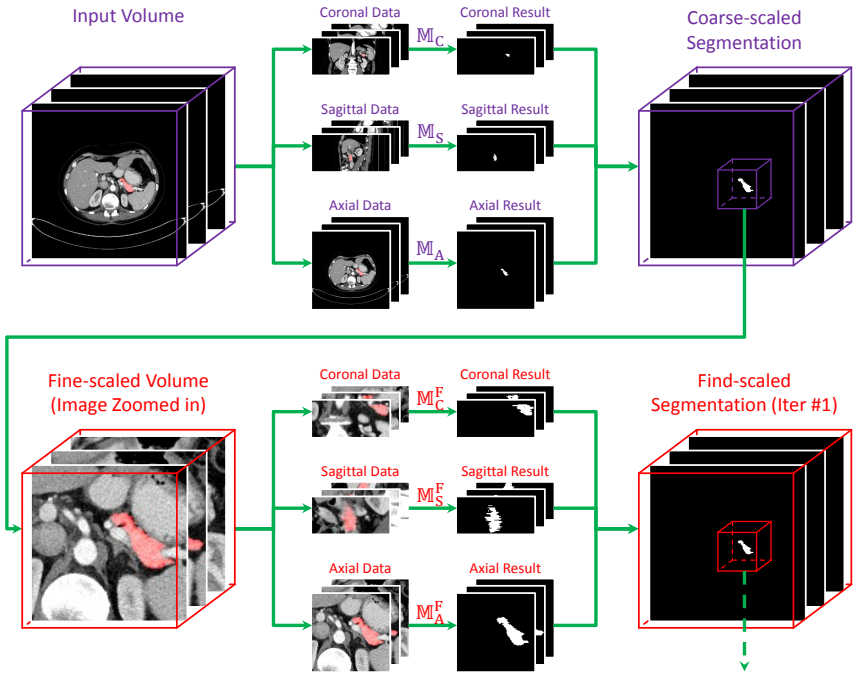

Fig. 2. The flowchart of coarse-scaled segmentation (above) and fine-tuning (below). For simplicity, we only illustrate one iteration in the fine-tuning process. In practice, there are at most 10 iterations in total.

3.3 Implementation Details

In both coarse-scaled and fine-scaled models, we apply the FCN-8s configuration [13], which achieves the highest validation performance on the PascalVOC segmentation task. The down-sampling stage is directly borrowed from, and initialized by, a pre-trained 16-layer VGGNet. This is followed by several deconvolutional layers to up-sample the image to the original resolution. We train the network with a fixed learning rate of 10^{-5}. We run 80,000 and 50,000 mini-batches for the coarse-scaled and fine-scaled models, respectively. Each mini-batch contains only one training sample (a 2D datum $(\mathbf{x}, \mathbf{y})$).

In training the coarse-scaled models, we only select those 2D slices in which the pancreas occupies at least 1% pixels. This is to prevent the model from being heavily impacted by the noisy background contents. In training the fine-scaled models, we first use the ground-truth annotation to find the 3D bounding box of the pancreas. When each slice is generated within the box, we add a frame around it, and filled with the original image data. The top, bottom, left and right margins of the frame are random numbers uniformly sampled from $\{10, 11, \ldots, 20\}$. This strategy, known as data augmentation, helps to regularize the network and prevent it from over-fitting. In the testing stage, the input image

Method	View			Fusion
	Coronal	Sagittal	Axial	
1-slice-segmentation	64.60 ± 11.42	69.71 ± 11.06	71.54 ± 8.87	75.51 ± 9.87
3-slice-segmentation	$\mathbf{66.88} \pm 11.08$	$\mathbf{71.41} \pm 11.12$	$\mathbf{73.08} \pm 9.60$	$\mathbf{75.74} \pm 10.47$

Table 1. Baseline segmentation accuracy (measured by DSC, %). We test both the 1-slice-segmentation and 3-slice-segmentation models. We report accuracies obtained by individual models and the fused model (see Section 3.1 for details).

is first fed into the coarse-scaled models to obtain an initial segmentation. After that, we find the minimal 3D bounding box containing the estimated pancreas. After each slice is cropped according to the box, we add a fixed frame of 15 pixels wide around it, and send it to the fine-scaled model.

4 Experiments

4.1 Dataset and Evaluation

We evaluate our algorithms on the NIH pancreas segmentation dataset [19], which contains 82 contrast-enhanced abdominal CT volumes. These data are acquired using Philips and Siemens MDCT scanners (120kVp tube voltage, scanning about 70 seconds after intravenous contrast injection in portal-venous). The subjects are normals, *i.e.*, they are neither major abdominal pathologies nor pancreatic cancer lesions. Their ages range from 18 to 76 years with a mean age of 46.8±16.7. The resolution of each CT scan is $512 \times 512 \times L$, where $L \in [181, 466]$ is the number of sampling slices along the long axis of the body. The slice thickness varies from 0.5mm–1.0mm. Following the standard cross-validation strategy, we split the dataset into 4 fixed folds, each of which contains approximately the same number of samples. We use the leave-one-out evaluation method, *i.e.*, training the model on 3 out of 4 subsets and testing it on the remaining one. We measure the segmentation accuracy by computing the Dice-Sørensen Coefficient (DSC) for each sample. This is a similarity metric between the prediction voxel set $\mathcal{A}$ and the ground-truth set $\mathcal{G}$. The mathematical form is $\mathrm{DSC}(\mathcal{A}, \mathcal{G}) = \frac{2 \times |\mathcal{A} \cap \mathcal{G}|}{|\mathcal{A}| + |\mathcal{G}|}$. We report the average DSC score together with the standard deviation over 82 testing cases.

4.2 Baseline Results

We first evaluate the baseline (coarse-scaled) approach, *i.e.*, performing either 1-slice-segmentation or 3-slice-segmentation using the models $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$ trained from different views. We also fuse the these three models by averaging their results at the testing stage. The results are summarized in Table 1.

We observe that 3-slice-segmentation works better than 1-slice-segmentation in each single case, either when three views are considered individually or after

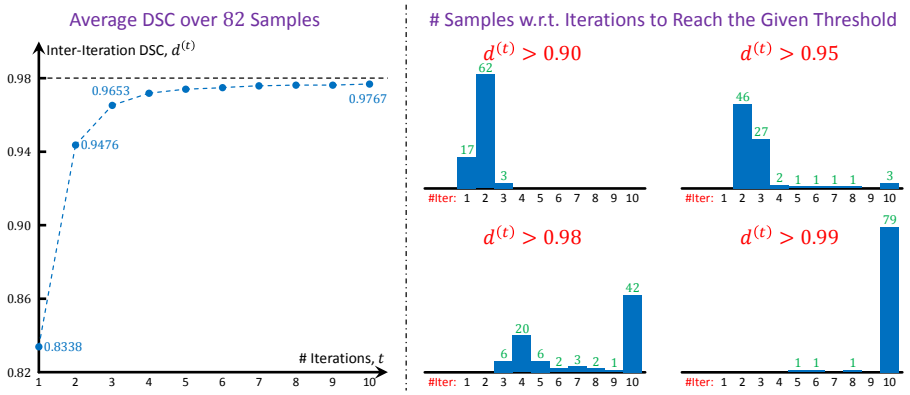

Fig. 3. Left: the average $d^{(t)} = \text{DSC}\left(\mathcal{S}^{(t-1)}, \mathcal{S}^{(t)}\right)$ over 82 samples with respect to t. Right: the histogram of samples that require a specific number of iterations in the fine-tuning process. 10 is the maximal number of iterations.

their results are fused. This verifies that integrating information on the neighboring slices is useful for segmentation. In fact, this is an alternative way of introducing 3D cues into segmentation, which is less computationally expensive than 3D-based networks [17][15]. When the amount of training data is limited (in the NIH pancreas segmentation dataset, only about 60 samples are used for training), this strategy also alleviates the risk of over-fitting, since the number of parameters only increases by a little compared to the 1-slice-segmentation model, *a.k.a.*, the vanilla 2D FCN model.

In the meantime, we find that fusing the results from three views largely boosts the segmentation accuracy. This suggests that complementary information is captured by combining different views. We also observe that in fusion, the 1-slice-segmentation approach enjoys significantly more accuracy gain than the 3-slice-segmentation approach. We believe this is a marginal effect, since both fusion and 3-slice-segmentation take the advantage of 3D visual cues.

In the fine-tuning section, to take the advantage of 3D, we train 3-slice-segmentation models, and perform fusion at the end of each iteration.

4.3 Fine-Tuning Results

Next, we evaluate the fine-tuning approach. As illustrated in Section 3.2, fine-tuning is performed in an iterative manner. Since deep neural networks are often sensitive to small noise, it is often very difficult for the iteration process to reach a complete stop, instead, after a sufficient number of iterations, the segmentation result does not change significantly. To measure convergence, we define the *inter-iteration DSC* to be $d^{(t)} \doteq \text{DSC}\left(\mathcal{S}^{(t-1)}, \mathcal{S}^{(t)}\right)$, where $\mathcal{S}^{(t)}$ is the segmentation mask after the t-th iteration. Note that $d^{(t)} = 1$ implies perfect convergence which is unlikely to happen. We compute the average $d^{(t)}$ value

Method	Mean DSC	# Iterations	Max DSC	Min DSC
Coarse Segmentation	75.74 ± 10.47	–	88.12	39.99
After 1 Iteration	82.16 ± 6.29	1	**90.85**	54.39
After 2 Iterations	82.13 ± 6.30	2	90.77	57.05
After 3 Iterations	82.09 ± 6.17	3	90.78	58.39
After 5 Iterations	82.11 ± 6.09	5	90.75	62.40
After 10 Iterations	82.25 ± 5.73	10	90.76	61.73
After $d_t > 0.90$	82.13 ± 6.35	1.83 ± 0.47	**90.85**	54.39
After $d_t > 0.95$	**82.37** ± 5.68	2.89 ± 1.75	**90.85**	**62.43**
After $d_t > 0.98$	82.28 ± 5.71	7.39 ± 2.88	90.78	61.94
After $d_t > 0.99$	82.28 ± 5.72	9.87 ± 0.73	90.77	61.94
Best among All Iterations	82.65 ± 5.47	3.49 ± 2.92	90.85	63.02
Oracle Bounding Box	83.18 ± 4.81	–	91.03	65.10

Table 2. Fine-tuned segmentation accuracy (measured by DSC, %). We start from the 3-slice-segmentation results, and explore different terminating conditions, including a fixed number of iterations and a fixed threshold of inter-iteration DSC. The last two lines show two upper-bounds of our approach, *i.e.*, "Best of All Iterations" means that we choose the highest DSC value over 10 iterations, and "Oracle Bounding Box" corresponds to using the ground-truth segmentation in extracting the bounding box.

($t = 1, 2, \ldots, T$) over 82 testing samples. The results are shown in Figure 3. Based on these, we conclude that the fine-tuning approach is generally stable: after 10 iterations, the average $d^{(t)}$ value over all samples is 0.9767, the median of the $d^{(t)}$ values is 0.9794, and the minimum is 0.9362. We also record the number of iterations required for each sample to reach a given threshold of $d^{(t)}$. Not surprisingly, increasing the threshold leads to a larger number of iterations.

Now, we consider two types of conditions for terminating the iteration, more precisely, after a fixed number of iterations, or after the inter-iteration DSC reaches a fixed threshold. When using the latter condition, we set the maximal number of iterations to be 10, *i.e.*, after 10 iterations, the fine-tuning process is terminated even if the threshold is not reached. The results are summarized in Table 2. Our fine-tuning approach significantly boosts the baseline accuracy (by 6.63%). This is impressive given the relatively high baseline accuracy (76.15%).

Regarding different terminating conditions, we find that performing merely 1 iteration is enough to significantly boost the segmentation accuracy (+6.42%). This verifies our hypothesis, *i.e.*, training a fine-scaled model helps to depict small organs more accurately. The best performance comes from setting a proper threshold (*e.g.*, 0.95) of $d^{(t)}$. Using a large threshold (*e.g.*, 0.98 or 0.99) does not produce accuracy gain, but requires a larger number of iterations and, consequently, heavier computation at the testing stage. For all 82 testing samples, it takes on average less than 3 iterations to reach the threshold 0.95, which guarantees the efficiency of our approach. On a modern GPU, we need about 3 minutes on each testing sample, comparable to recent work [18], but we report much higher segmentation accuracy (see Table 3). Setting a threshold of $d^{(t)}$ is

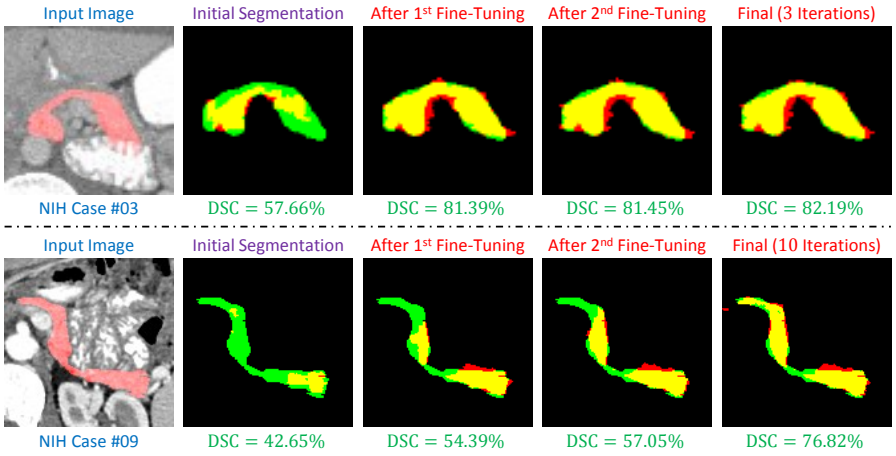

Fig. 4. Examples of segmentation results before and after fine-tuning (best viewed in color PDF). We only show a small region covering the pancreas in the axial view. The terminating condition is $d^{(t)} > 0.95$. Red, green and yellow indicate the prediction, ground-truth and overlapped regions, respectively.

Method	Mean DSC (%)	Max DSC (%)	Min DSC (%)
Roth *et.al*, MICCAI'2015 [19]	71.42 ± 10.11	86.29	23.99
Roth *et.al*, MICCAI'2016 [18]	78.01 ± 8.20	88.65	34.11
Ours, without Fine-Tuning	75.74 ± 10.47	88.12	39.99
Ours, with Fine-Tuning	$\mathbf{82.37} \pm 5.68$	**90.85**	**62.43**

Table 3. Comparison of our algorithm with the state-of-the-art approaches. Our coarse-to-fine approach achieves the best number under each performance statistics.

more efficient than using a fixed number of iterations, *e.g.*, if we fix the number of iterations to be 3, the average DSC goes down by nearly 0.3%.

As an diagnostic experiment, we use the ground-truth bounding box of the testing cases to generate the input volume for fine-tuning. This results in a 83.18% accuracy (no iteration is needed). In comparison, we report a comparable 82.37% accuracy, indicating that we are good at estimating the bounding box.

We show two challenging cases in Figure 4, and observe how fine-tuning gradually improves the segmentation accuracy by iterations.

4.4 Comparison to the State-of-the-Art

In Table 3, we compare our segmentation result with the state-of-the-art approaches. Using DSC as the evaluation metric, our approach outperforms the recent published work [18] significantly. The average accuracy over 82 samples increases remarkably from 78.01% to 82.37%, and the standard deviation decreases

from 8.20% to 5.68%, implying that fine-tuning generates more stable results. Especially, [18] reports 34.11% on the worse case, and this number is impressively boosted to 62.43% by our algorithm. We point out that these improvements are mainly owed to the fine-tuning approach. Without them, the average accuracy is 76.15%, and the accuracy on the worst case is merely 39.99%. Please refer to Figure 4 for an intuitive illustration of how fine-tuning improves the accuracy of challenging cases.

5 Conclusions

We present a coarse-to-fine approach for segmenting the pancreas from CT-scanned images. Our motivation is straightforward: deep networks such as FCN are not good at segmenting very small objects compared to the input image size. Therefore, we train an additional fine-scaled model to deal with small inputs. In the testing stage, the coarse-scaled model is used to roughly locate the pancreas, and an iterative process is performed on the fine-scaled model to refine the initial segmentation. In practice, this process often comes to an end after 2–3 iterations.

We evaluate our algorithm on the NIH pancreas segmentation dataset with 82 samples, and outperforms the state-of-the-art by more than 4%, measured by the Dice-Sørensen Coefficient (DSC). Most of the benefit of the fine-tuning approach comes from the first iteration. The remaining iterations only improve the segmentation accuracy by a little (about 0.3%). We believe that our algorithm can achieve even higher accuracy when more powerful deep networks are equipped. We point out that the idea of fine-tuning can be applied to other small organs, *e.g.*, the spleen. In the future, we will also explore the possibility of incorporating fine-tuning into an end-to-end training process.

Acknowledgements

This work is supported by the Lustgarten Foundation for Pancreatic Cancer research.

References

1. Al-Ayyoub, M., Alawad, D., Al-Darabsah, K., Aljarrah, I.: Automatic Detection and Classification of Brain Hemorrhages. WSEAS Transactions on Computers (2013)
2. Chen, H., Dou, Q., Yu, L., Heng, P.: VoxResNet: Deep Voxelwise Residual Networks for Volumetric Brain Segmentation. arXiv preprint arXiv:1608.05895 (2016)
3. Chen, J., Yang, L., Zhang, Y., Alber, M., Chen, D.: Combining Fully Convolutional and Recurrent Neural Networks for 3D Biomedical Image Segmentation. Advances In Neural Information Processing Systems (2016)
4. Chu, C., Oda, M., Kitasaka, T., Misawa, K., Fujiwara, M., Hayashi, Y., Nimura, Y., Rueckert, D., Mori, K.: Multi-organ Segmentation based on Spatially-Divided Probabilistic Atlas from 3D Abdominal CT Images. International Conference on Medical Image Computing and Computer-Assisted Intervention (2013)

5. Ciresan, D., Giusti, A., Gambardella, L., Schmidhuber, J.: Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. Advances in Neural Information Processing Systems (2012)
6. Elnakib, A., Gimelfarb, G., Suri, J., El-Baz, A.: Medical Image Segmentation: A Brief Survey. Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies (2011)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Computer Vision and Pattern Recognition (2014)
8. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P., Larochelle, H.: Brain Tumor Segmentation with Deep Neural Networks. Medical Image Analysis (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. Computer Vision and Pattern Recognition (2016)
10. Kamnitsas, K., Ledig, C., Newcombe, V., Simpson, J., Kane, A., Menon, D., Rueckert, D., Glocker, B.: Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation. arXiv preprint arXiv:1603.05959 (2016)
11. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (2012)
12. Lai, M.: Deep Learning for Medical Image Segmentation. arXiv preprint arXiv:1505.02000 (2015)
13. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. Computer Vision and Pattern Recognition (2015)
14. Merkow, J., Kriegman, D., Marsden, A., Tu, Z.: Dense Volume-to-Volume Vascular Boundary Detection. arXiv preprint arXiv:1605.08401 (2016)
15. Milletari, F., Navab, N., Ahmadi, S.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. arXiv preprint arXiv:1606.04797 (2016)
16. Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., Nielsen, M.: Deep Feature Learning for Knee Cartilage Segmentation Using a Triplanar Convolutional Neural Network. International Conference on Medical Image Computing and Computer-Assisted Intervention (2013)
17. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)
18. Roth, H., Lu, L., Farag, A., Sohn, A., Summers, R.: Spatial Aggregation of Holistically-Nested Networks for Automated Pancreas Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (2016)
19. Roth, H., Lu, L., Farag, A., Shin, H., Liu, J., Turkbey, E., Summers, R.: DeepOrgan: Multi-level Deep Convolutional Networks for Automated Pancreas Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)
20. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (2015)
21. Singh, S., Hoiem, D., Forsyth, D.: Learning to Localize Little Landmarks. Computer Vision and Pattern Recognition (2016)

22. Stollenga, M., Byeon, W., Liwicki, M., Schmidhuber, J.: Parallel Multi-Dimensional LSTM, with Application to Fast Biomedical Volumetric Image Segmentation. Advances in Neural Information Processing Systems (2015)
23. Subbanna, N., Precup, D., Arnold, D., Arbel, T.: IMaGe: Iterative Multilevel Probabilistic Graphical Model for Detection and Segmentation of Multiple Sclerosis Lesions in Brain MRI. International Conference on Information Processing in Medical Imaging (2015)
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going Deeper with Convolutions. Computer Vision and Pattern Recognition (2015)
25. Wang, D., Khosla, A., Gargeya, R., Irshad, H., Beck, A.: Deep Learning for Identifying Metastatic Breast Cancer. arXiv preprint arXiv:1606.05718 (2016)
26. Wang, Z., Bhatia, K., Glocker, B., Marvao, A., Dawes, T., Misawa, K., Mori, K., Rueckert, D.: Geodesic Patch-based Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (2014)