Chapter 3
2D-Based Coarse-to-Fine Approaches for Small Target Segmentation in Abdominal CT Scans

Yuyin Zhou, Qihang Yu, Yan Wang, Lingxi Xie, Wei Shen, Elliot K. Fishman and Alan L. Yuille

Abstract Deep neural networks have been widely adopted for automatic organ segmentation from abdominal CT scans. However, the segmentation accuracy of small organs (e.g., *pancreas*) or neoplasms (e.g., *pancreatic cyst*) is sometimes below satisfaction, arguably because deep networks are easily disrupted by the complex and variable background regions which occupy a large fraction of the input volume. In this chapter, we propose two coarse-to-fine mechanisms which use prediction from the first (coarse) stage to shrink the input region for the second (fine) stage. More specifically, the two stages in the first method are trained individually in a step-wise manner, so that the entire input region and the region cropped according to the bounding box are treated separately. While the second method inserts a saliency transformation module between the two stages so that the segmentation probability map from the previous iteration can be repeatedly converted as spatial weights to the current iteration. In training, it allows joint optimization over the deep networks. In testing, it propagates multi-stage visual information throughout iterations to improve

Y. Zhou and Q. Yu contributed equally to this work.

Y. Zhou · Q. Yu · Y. Wang · L. Xie · W. Shen · A. L. Yuille (✉)
Johns Hopkins University, 3400 N Charles St, Baltimore, MD 21218, USA
e-mail: alan.l.yuille@gmail.com

Y. Zhou
e-mail: zhouyuyiner@gmail.com

Q. Yu
e-mail: yucornetto@gmail.com

Y. Wang
e-mail: wyanny.9@gmail.com

L. Xie
e-mail: 198808xc@gmail.com

W. Shen
e-mail: shenwei1231@gmail.com

E. K. Fishman
Johns Hopkins University School of Medicine, 733 N Broadway, Baltimore, MD 21205, USA
e-mail: efishman@jhmi.edu

L. Lu et al. (eds.), *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, Advances in Computer Vision and Pattern Recognition,
https://doi.org/10.1007/978-3-030-13969-8_3

segmentation accuracy. Experiments are performed on several CT datasets, including NIH pancreas, JHMI multi-organ, and JHMI pancreatic cyst dataset. Our proposed approach gives strong results in terms of DSC.

3.1 Introduction

This chapter focuses on small organs (e.g., the *pancreas*) and neoplasms (e.g., *pancreatic cyst*) segmentation from abdominal CT scans, which is an important prerequisite for enabling computers to assist human doctors for clinical purposes. This problem falls into the research area named *medical imaging analysis*. Recently, great progress has been brought to this field by the fast development of deep learning, especially convolutional neural networks [18, 29]. Many conventional methods, such as the graph-based segmentation approaches [1] or those based on handcrafted local features [45], have been replaced by deep segmentation networks, which typically produce higher segmentation accuracy [34, 35, 43, 44, 51].

Segmenting tiny organs, blood vessels, or neoplasms from a CT scan is often challenging. As the target often occupies a *small part* of input data (e.g., less than 1.5% in a 2D image, see Fig. 3.1), deep segmentation networks such as FCN [29] and DeepLab [5] can be easily confused by the background region, which may contain complicated and variable contents. This motivates us to propose *coarse-to-fine* approaches, in which the coarse stage provides a rough localization and the fine stage performs accurate segmentation.

We propose two coarse-to-fine approaches in this chapter. In the first approach, we use the predicted segmentation mask to shrink the input region. With a relatively smaller input region (e.g., a bounding box defined by the mask), it is straightforward to achieve more accurate segmentation. At the training stage, we fix the input regions generated from the ground-truth annotation, and train two deep segmentation networks, i.e., a coarse-scaled one and a fine-scaled one, to deal with the entire input region and the region cropped according to the bounding box, respectively. At the testing stage, the network parameters remain unchanged, and the coarse-scaled

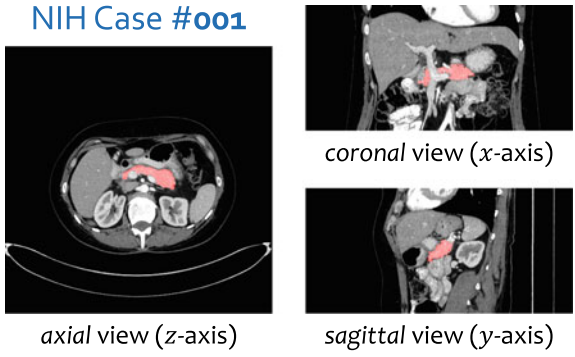

Fig. 3.1 A typical example from the NIH *pancreas* segmentation dataset [35] (best viewed in color). We highlight the *pancreas* in red seen from three different viewpoints. It is a relatively small organ with irregular shape and boundary

network was first used to obtain the rough position of the small target, and the fine-scaled network was executed several times and the segmentation mask was updated iteratively until convergence. The iterative process can be formulated as a fixed-point model [23]. This approach can be further extended to segment *pancreatic cyst*, which lays the foundation of early diagnosis of pancreatic cancer, where we first find the pancreas by a coarse-to-fine algorithm, then we localize and segment the cyst based on the predicted pancreas mask by a separate coarse-to-fine segmentation approach. Intuitively, the pancreatic cyst is often closely related to the pancreas, and thus segmenting the pancreas (relatively easier) may assist the localization and segmentation of the cyst. A deep supervision [21] strategy is introduced into the original segmentation network, leading to a joint objective function taking both the pancreas and the cyst into consideration.

In order to embed *consistency* between training and testing flowcharts, which is to say, in the training phase to minimize a global energy function in coarse and fine stages simultaneously, in our second approach, we propose a Recurrent Saliency Transformation Network (RSTN). The chief innovation is to relate the coarse and fine stages with a saliency transformation module, which repeatedly transforms the segmentation probability map from previous iterations as spatial priors in the current iteration. This brings us twofold advantages over the first method. First, in the training phase, the coarse-scaled and fine-scaled networks are optimized jointly, so that the segmentation ability of each of them gets improved. Second, in the testing phase, the segmentation mask of each iteration is preserved and propagated throughout iterations, enabling multi-stage visual cues to be incorporated toward more accurate segmentation. To capture the relationship between the pancreas and its internal cysts, we also extend this approach to segment pancreas and cyst by two RSTN modules, which observes strong results. To the best of our knowledge, this idea was not studied in the computer vision community, as it requires making use of some special properties of CT scans.

We perform experiments on three CT datasets for small target segmentation. We show the superiority of our approaches on the NIH *pancreas* segmentation dataset [35], JHMI multi-organ dataset, and JHMI pancreatic cyst dataset, which guarantees its efficiency and reliability in real clinical applications.

This chapter summarizes our previous works [48, 52, 53] and provides more experimental results. The remainder of this chapter is organized as follows. Section 3.2 briefly reviews related work, Sect. 3.3 describes the proposed step-wise coarse-to-fine approach, and Sect. 3.4 presents our proposed end-to-end coarse-to-fine approach. After experiments are shown in Sects. 3.5 and 3.6, we draw our conclusions in Sect. 3.8.

3.2 Related Work

Computer-aided diagnosis (CAD) is an important technique which can assist human doctors in many clinical scenarios. An important prerequisite of CAD is medical imaging analysis. As a popular and cheap way of medical imaging, contrast-enhanced

computed tomography (CECT) produces detailed images of internal organs, bones, soft tissues and blood vessels. It is of great value to automatically segment organs and/or soft tissues from these CT volumes for further diagnosis [2, 13, 42, 52]. To capture specific properties of different organs, researchers often design individualized algorithms for each of them. Typical examples include the the liver [15, 27], the *spleen* [28], the *kidneys* [1, 25], the *lungs* [16], the *pancreas* [6, 45], etc. Small organs (e.g., the *pancreas*) are often more difficult to segment, partly due to their low contrast and large anatomical variability in size and (most often irregular) shape, as well as the complicated and unpredictable background contents. In particular, the internal neoplasms such as cysts [7] and tumors [49] can further change the anatomical property of the pancreas, making it even more difficult to recognize both targets.

Compared to the papers cited above which used conventional approaches for segmentation, the progress of deep learning brought more powerful and efficient solutions. In particular, convolutional neural networks have been widely applied to a wide range of vision tasks, such as image classification [14, 18, 39], object detection [10, 33, 41], and semantic segmentation [5, 29]. Recurrent neural networks, as a related class of networks, were first designed to process sequential data [11, 38, 40], and later generalized to image classification [24] and scene labeling [32] tasks. In the area of medical imaging analysis, in particular organ segmentation, these techniques have been shown to significantly outperform conventional approaches, e.g., segmenting the *liver* [8], the *lung* [12], or the *pancreas* [3, 36, 37]. Note that medical images differ from natural images in that data appear in a volumetric form. To deal with these data, researchers either slice an 3D volume into 2D slices (as in this work), or train an 3D network directly [17, 30, 31, 47]. In the latter case, limited GPU memory often leads to patch-based training and testing strategies. The tradeoff between 2D and 3D approaches is discussed in [20].

By comparison to the entire CT volume, the organs and neoplasm considered in this chapter often occupy a relatively small area. As deep segmentation networks such as FCN [29] are less accurate in depicting small targets, researchers proposed two types of ideas to improve detection and/or segmentation performance. The first type involved rescaling the image so that the target becomes comparable to the training samples [46], and the second one considered to focus on a subregion of the image for each target to obtain higher accuracy in detection [4]. The coarse-to-fine idea was also well studied in the computer vision area for saliency detection [19] or semantic segmentation [22, 26]. This chapter focuses on presenting two coarse-to-fine frameworks for medical image segmentation.

3.3 A Step-Wise Coarse-to-Fine Approach for Medical Image Segmentation

We investigate the problem of segmenting an organ from abdominal CT scans. Let an CT image be a 3D volume $\mathbf{X}$ of size $W \times H \times L$ which is annotated with a binary ground-truth segmentation $\mathbf{Y}$ where $y_i = 1$ indicates a foreground voxel. The goal

of our work is to produce a binary output volume $\mathbf{Z}$ of the same dimension. Denote $\mathcal{Y}$ and $\mathcal{Z}$ as the set of foreground voxels in the ground-truth and prediction, i.e., $\mathcal{Y} = \{i \mid y_i = 1\}$ and $\mathcal{Z} = \{i \mid z_i = 1\}$. The accuracy of segmentation is evaluated by the Dice-Sørensen coefficient (DSC): $\text{DSC}(\mathcal{Y}, \mathcal{Z}) = \frac{2\times|\mathcal{Y}\cap\mathcal{Z}|}{|\mathcal{Y}|+|\mathcal{Z}|}$. This metric falls in the range of [0, 1] with 1 implying perfect segmentation.

3.3.1 Deep Segmentation Networks

Consider a segmentation model $\mathbb{M} : \mathbf{Z} = \mathbf{f}(\mathbf{X}; \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ denotes the model parameters, and the loss function is written as $\mathcal{L}(\mathbf{Z}, \mathbf{Y})$. In the context of a deep segmentation network, we optimize $\mathcal{L}$ with respect to the network weights $\boldsymbol{\Theta}$ by gradient backpropagation. As the foreground region is often very small, we follow [31] to design a DSC-loss layer to prevent the model from being heavily biased toward the background class. We slightly modify the DSC of two voxel sets $\mathcal{A}$ and $\mathcal{B}$, $\text{DSC}(\mathcal{A}, \mathcal{B}) = \frac{2\times|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}|+|\mathcal{B}|}$, into a loss function between the ground-truth mask $\mathbf{Y}$ and the predicted mask $\mathbf{Z}$, i.e., $\mathcal{L}(\mathbf{Z}, \mathbf{Y}) = 1 - \frac{2\times\sum_i z_i y_i}{\sum_i z_i + \sum_i y_i}$. Note that this is a "soft" definition of DSC, and it is equivalent to the original form if all z_i's are either 0 or 1. The gradient computation is straightforward: $\frac{\partial \mathcal{L}(\mathbf{Z},\mathbf{Y})}{\partial z_j} = -2 \times \frac{y_j(\sum_i z_i + \sum_i y_i) - \sum_i z_i y_i}{(\sum_i z_i + \sum_i y_i)^2}$.

We train 2D deep networks for 3D segmentation.[1] Each 3D volume $\mathbf{X}$ is sliced along three axes, the *coronal*, *sagittal* and *axial* views, and these 2D slices are denoted by $\mathbf{X}_{\text{C},w}$ $(w = 1, 2, \ldots, W)$, $\mathbf{X}_{\text{S},h}$ $(h = 1, 2, \ldots, H)$ and $\mathbf{X}_{\text{A},l}$ $(l = 1, 2, \ldots, L)$, where the subscripts C, S and A stand for *coronal*, *sagittal* and *axial*, respectively. On each axis, an individual 2D-FCN [29] on a 16-layer VGGNet [39] is trained We train three 2D-FCN models $\mathbb{M}_\text{C}$, $\mathbb{M}_\text{S}$ and $\mathbb{M}_\text{A}$ to perform segmentation through three views individually (images from three views are quite different). In testing, the segmentation results from three views are fused via majority voting. Both multi-slice segmentation (3 neighboring slices are combined as a basic unit in training and testing) and multi-axis fusion (majority voting over three axes) is performed to incorporate pseudo-3D information into segmentation.

3.3.2 Fixed-Point Optimization

The organs and neoplasms investigated in this chapter (e.g., the *pancreas*) are relatively small. In each 2D slice, the fraction of the foreground pixels is often smaller than 1.5%. It was observed [35] that deep segmentation networks such as FCN [29] produce less satisfying results when detecting small organs, arguably because the network is easily disrupted by the varying contents in the background regions. Much

[1] Please see Sect. 3.5.3.2 for the comparison to 3D networks.

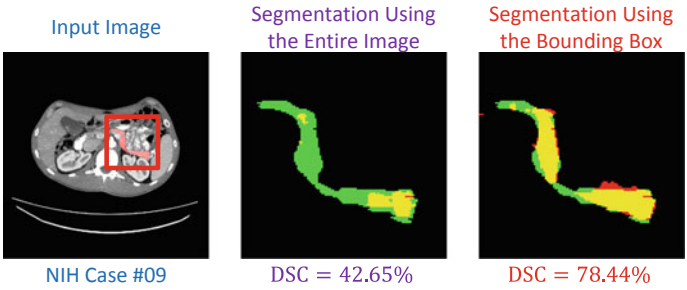

Fig. 3.2 Segmentation results with different input regions (best viewed in color), either using the entire image or the bounding box (the red frame). Red, green and yellow indicate the prediction, ground-truth, and overlapped pixels, respectively

more accurate segmentation can be obtained by using a smaller input region around the region of interest. A typical example is shown in Fig. 3.2.

This inspires us to make use of the predicted segmentation mask to shrink the input region. We introduce a transformation function $r(\mathbf{X}, \mathbf{Z}^\star)$ which generates the input region given the current segmentation $\mathbf{Z}^\star$. We rewrite the model as $\mathbf{Z} = \mathbf{f}(r(\mathbf{X}, \mathbf{Z}^\star); \boldsymbol{\Theta})$, and the loss function is $\mathcal{L}(\mathbf{f}(r(\mathbf{X}, \mathbf{Z}^\star); \boldsymbol{\Theta}), \mathbf{Y})$. Note that the segmentation mask ($\mathbf{Z}$ or $\mathbf{Z}^\star$) appears in both the input and output of $\mathbf{Z} = \mathbf{f}(r(\mathbf{X}, \mathbf{Z}^\star); \boldsymbol{\Theta})$. This is a fixed-point model, and we apply the approach described in [23] for optimization, i.e., finding a steady-state solution for $\mathbf{Z}$.

In training, the ground-truth annotation $\mathbf{Y}$ is used as the input mask $\mathbf{Z}^\star$. We train two sets of models (each set contains three models for different views) to deal with different input sizes. The *coarse-scaled* models are trained on those slices on which the pancreas occupies at least 100 pixels (approximately 25 mm^2 in an 2D slice, our approach is not sensitive to this parameter) so as to prevent the model from being heavily impacted by the background. For the *fine-scaled* models, we crop each slice according to the minimal 2D box covering the pancreas, add a frame around it, and fill it up with the original image data. The top, bottom, left and right margins of the frame are random integers sampled from $\{0, 1, \ldots, 60\}$. This strategy, known as data augmentation, helps to regularize the network and prevent over-fitting.

We initialize both networks using the FCN-8s model [29] pretrained on the PascalVOC image segmentation task. The coarse-scaled model is fine-tuned with a learning rate of 10^{-5} for 80,000 iterations, and the fine-scaled model undergoes 60,000 iterations with a learning rate of 10^{-4}. Each mini-batch contains one training sample (an 2D image sliced from an 3D volume).

In testing, we use an iterative process to find a steady-state solution for $\mathbf{Z} = \mathbf{f}(r(\mathbf{X}, \mathbf{Z}^\star); \boldsymbol{\Theta})$. At the beginning, $\mathbf{Z}^\star$ is initialized as the entire 3D volume, and we compute the *coarse* segmentation $\mathbf{Z}^{(0)}$ using the *coarse-scaled* models. In each of the following T iterations, we slice the predicted mask $\mathbf{Z}^{(t-1)}$, find the smallest 2D box to cover all predicted foreground pixels in each slice, add a 30-pixel-wide frame around it (this is the mean value of the random distribution used in training), and use the

Algorithm 1 Fixed-Point Model for Segmentation

Input: the testing volume $\mathbf{X}$, coarse-scaled models $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$, fine-scaled models $\mathbb{M}_{\mathrm{C}}^{\mathrm{F}}$, $\mathbb{M}_{\mathrm{S}}^{\mathrm{F}}$ and $\mathbb{M}_{\mathrm{A}}^{\mathrm{F}}$, threshold R, maximal rounds in iteration T.
Initialization: using $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$ to generate $\mathbf{Z}^{(0)}$ from $\mathbf{X}$;
for $t = 1, 2, \ldots, T$ **do**
  Using $\mathbb{M}_{\mathrm{C}}^{\mathrm{F}}$, $\mathbb{M}_{\mathrm{S}}^{\mathrm{F}}$ and $\mathbb{M}_{\mathrm{A}}^{\mathrm{F}}$ to generate $\mathbf{Z}^{(t)}$ from $\mathbf{Z}^{(t-1)}$;
  if $\mathrm{DSC}\big(\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\big) \geqslant R$ **then**
    break;
  end if
end for
Output: the final segmentation $\mathbf{Z}^{\star} = \mathbf{Z}^{(t)}$.

fine-scaled models to compute $\mathbf{Z}^{(t)}$. The iteration terminates when a fixed number of iterations T is reached, or the the similarity between successive segmentation results ($\mathbf{Z}^{(t-1)}$ and $\mathbf{Z}^{(t)}$) is larger than a given threshold R. The similarity is defined as the inter-iteration DSC, namely $d^{(t)} = \mathrm{DSC}\big(\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\big) = \frac{2\times\sum_i z_i^{(t-1)} z_i^{(t)}}{\sum_i z_i^{(t-1)} + \sum_i z_i^{(t)}}$. The testing stage is illustrated in Fig. 3.3 and described in Algorithm 1.

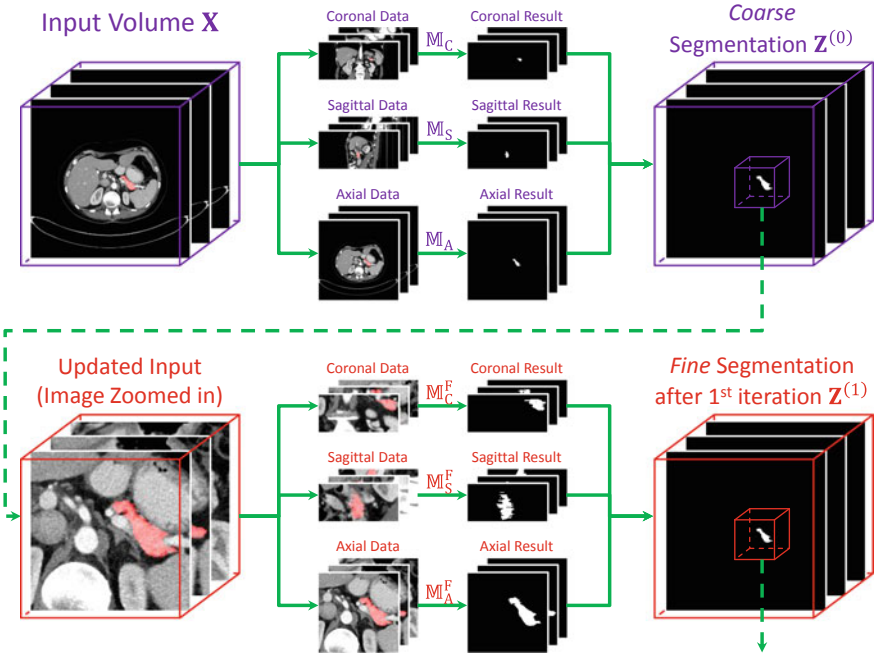

Fig. 3.3 Illustration of the testing process (best viewed in color). Only one iteration is shown here. In practice, there are at most 10 iterations

3.3.3 Application to Pancreatic Cyst Segmentation

3.3.3.1 Formulation

Let the 3D CT-scanned volume $\mathbf{X}$ annotated with ground-truth pancreas segmentation $\mathbf{P}^\star$ and cyst segmentation $\mathbf{C}^\star$, and both of them are of the same dimensionality as $\mathbf{X}$. $P_i^\star = 1$ and $C_i^\star = 1$ indicate a foreground voxel of pancreas and cyst, respectively. Denote a cyst segmentation model as $\mathbb{M} : \mathbf{C} = \mathbf{f}(\mathbf{X}; \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ denotes the model parameters. The loss function can be written as $\mathcal{L}(\mathbf{C}, \mathbf{C}^\star)$. In a regular deep neural network such as our baseline, the fully convolutional network (FCN) [29], we optimize $\mathcal{L}$ with respect to the network weights $\boldsymbol{\Theta}$ via gradient backpropagation. To deal with small targets, we also follow [31] to compute the DSC-loss function: $\mathcal{L}(\mathbf{C}, \mathbf{C}^\star) = \frac{2\times\sum_i C_i C_i^\star}{\sum_i C_i + \sum_i C_i^\star}$. The gradient $\frac{\partial \mathcal{L}(\mathbf{C},\mathbf{C}^\star)}{\partial \mathbf{C}}$ can be easily computed.

The pancreas is a small organ, and the pancreatic cyst is even smaller. In our newly collected dataset, the fraction of the cyst, relative to the entire volume, is often much smaller than 0.1%. In a very challenging case, the cyst only occupies 0.0015% of the volume, or around 1.5% of the pancreas. This largely increases the difficulty of segmentation or even localization. Figure 3.4 shows a representative example where cyst segmentation fails completely when we take the entire 2D slice as the input.

To deal with this problem, we note that the location of the pancreatic cyst is highly relevant to the pancreas. Denote the set of voxels of the pancreas as $\mathcal{P}^\star = \left\{i \mid P_i^\star = 1\right\}$, and similarly, the set of cyst voxels as $C^\star = \left\{i \mid C_i^\star = 1\right\}$. Frequently, a large fraction of $C^\star$ falls within $\mathcal{P}^\star$ (e.g., $|\mathcal{P}^\star \cap C^\star| / |C^\star| > 95\%$ in 121 out of 131 cases in our dataset). Starting from the pancreas mask increases the chance of accurately segmenting the cyst. Figure 3.4 shows an example of using the ground-truth pancreas mask to recover the failure case of cyst segmentation.

This inspires us to perform cyst segmentation based on the pancreas region, which is relatively easy to detect. To this end, we introduce the pancreas mask $\mathbf{P}$ as an

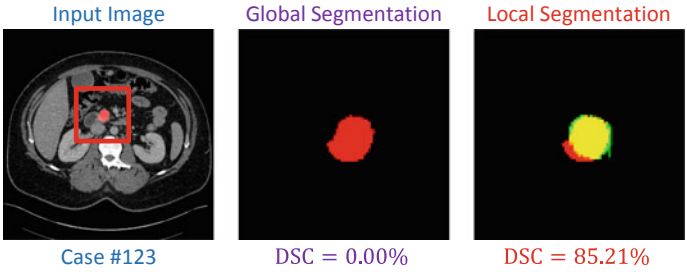

Fig. 3.4 A relatively difficult case in pancreatic cyst segmentation and the results produced by different input regions, namely using the entire image and the region around the ground-truth pancreas mask (best viewed in color). The cystic, predicted and overlapping regions are marked by red, green, and yellow, respectively. For better visualization, the right two figures are zoomed in with respect to the red frame

explicit variable of our approach, and append another term to the loss function to jointly optimize both pancreas and cyst segmentation networks. Mathematically, let the pancreas segmentation model be $\mathbb{M}_{\mathrm{P}} : \mathbf{P} = \mathbf{f}_{\mathrm{P}}(\mathbf{X}; \boldsymbol{\Theta}_{\mathrm{P}})$, and the corresponding loss term be $\mathcal{L}_{\mathrm{P}}(\mathbf{P}, \mathbf{P}^{\star})$. Based on $\mathbf{P}$, we create a smaller input region by applying a transformation $\mathbf{X}' = \sigma[\mathbf{X}, \mathbf{P}]$, and feed $\mathbf{X}'$ to the next stage. Thus, the cyst segmentation model can be written as $\mathbb{M}_{\mathrm{C}} : \mathbf{C} = \mathbf{f}_{\mathrm{C}}\big(\mathbf{X}'; \boldsymbol{\Theta}_{\mathrm{C}}\big)$, and we have the corresponding loss them $\mathcal{L}_{\mathrm{C}}(\mathbf{C}, \mathbf{C}^{\star})$. To optimize both $\boldsymbol{\Theta}_{\mathrm{P}}$ and $\boldsymbol{\Theta}_{\mathrm{C}}$, we consider the following loss function:

$$\mathcal{L}\big(\mathbf{P}, \mathbf{P}^{\star}, \mathbf{C}, \mathbf{C}^{\star}\big) = \lambda \mathcal{L}_{\mathrm{P}}\big(\mathbf{P}, \mathbf{P}^{\star}\big) + (1 - \lambda)\, \mathcal{L}_{\mathrm{C}}\big(\mathbf{C}, \mathbf{C}^{\star}\big), \tag{3.1}$$

where λ is the balancing parameter defining the weight between either terms.

3.3.3.2 Optimization

We use gradient descent for optimization, which involves computing the gradients over $\boldsymbol{\Theta}_{\mathrm{P}}$ and $\boldsymbol{\Theta}_{\mathrm{C}}$. Among these, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{\mathrm{C}}} = \frac{\partial \mathcal{L}_{\mathrm{C}}}{\partial \boldsymbol{\Theta}_{\mathrm{C}}}$, and thus we can compute it via standard backpropagation in a deep neural network. On the other hand, $\boldsymbol{\Theta}_{\mathrm{P}}$ is involved in both loss terms, and applying the chain rule yields:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{\mathrm{P}}} = \frac{\partial \mathcal{L}_{\mathrm{P}}}{\partial \boldsymbol{\Theta}_{\mathrm{P}}} + \frac{\partial \mathcal{L}_{\mathrm{C}}}{\partial \mathbf{X}'} \cdot \frac{\partial \mathbf{X}'}{\partial \mathbf{P}} \cdot \frac{\partial \mathbf{P}}{\partial \boldsymbol{\Theta}_{\mathrm{P}}}. \tag{3.2}$$

The second term on the right-hand side depends on the definition of $\mathbf{X}' = \sigma[\mathbf{X}, \mathbf{P}]$. In practice, we define a simple transformation to simplify the computation. The intensity value (directly related to the Hounsfield units in CT scan) of each voxel is either preserved or set as 0, and the criterion is whether there exists a nearby voxel which is likely to fall within the pancreas region:

$$X'_i = X_i \times \mathbb{I}\big\{\exists j \mid P_j > 0.5 \wedge |i - j| < t\big\}, \tag{3.3}$$

where t is the threshold which is the farthest distance from a cyst voxel to the pancreas volume. We set $t = 15$ in practice, and our approach is not sensitive to this parameter. With this formulation, i.e., $\frac{\partial X'_i}{\partial P_j} = 0$ almost everywhere. Thus, we have $\frac{\partial \mathbf{X}'}{\partial \mathbf{P}} = \mathbf{0}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{\mathrm{P}}} = \frac{\partial \mathcal{L}_{\mathrm{P}}}{\partial \boldsymbol{\Theta}_{\mathrm{P}}}$. This allows us to factorize the optimization into two stages in both training and testing. Since $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{\mathrm{P}}}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_{\mathrm{C}}}$ are individually optimized, the balancing parameter λ in Eq. (3.1) can be ignored. The overall framework is illustrated in Fig. 3.5. In training, we directly set $\mathbf{X}' = \sigma[\mathbf{X}, \mathbf{P}^{\star}]$, so that the cyst segmentation model $\mathbb{M}_{\mathrm{C}}$ receives more reliable supervision. In testing, starting from $\mathbf{X}$, we compute $\mathbf{P}$, $\mathbf{X}'$ and $\mathbf{C}$ orderly. Dealing with two stages individually reduces the computational overheads. It is also possible to formulate the second stage as multi-label segmentation.

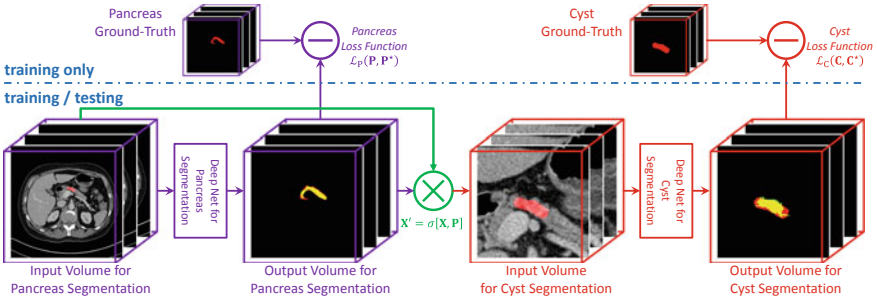

Fig. 3.5 The framework of our approach (best viewed in color). Two deep segmentation networks are stacked, and two loss functions are computed. The predicted pancreas mask is used in transforming the input image for cyst segmentation

3.4 An End-to-End Coarse-to-Fine Approach for Medical Image Segmentation

The step-wise coarse-to-fine approach is delicately designed for tiny target segmentation, but lacks global optimization of both the coarse and fine networks in the training stage. This motivates us to connect these two networks with a saliency transformation module, which leads to our end-to-end coarse-to-fine approach.

3.4.1 Recurrent Saliency Transformation Network

Following the step-wise coarse-to-fine approach, we also train an individual model for each of the three viewpoints. Without loss of generality, we consider a 2D slice along the *axial* view, denoted by $\mathbf{X}_{\mathrm{A},l}$. Our goal is to infer a binary segmentation mask $\mathbf{Z}_{\mathrm{A},l}$, which is achieved by first computing a *probability map* $\mathbf{P}_{\mathrm{A},l} = \mathbf{f}\big[\mathbf{X}_{\mathrm{A},l}; \boldsymbol{\theta}\big]$, where $\mathbf{f}[\cdot; \boldsymbol{\theta}]$ is a deep segmentation network with $\boldsymbol{\theta}$ being network parameters, and then binarizing $\mathbf{P}_{\mathrm{A},l}$ into $\mathbf{Z}_{\mathrm{A},l}$ using a fixed threshold of 0.5, i.e., $\mathbf{Z}_{\mathrm{A},l} = \mathbb{I}\big[\mathbf{P}_{\mathrm{A},l} \geqslant 0.5\big]$.

In order to assist segmentation with the probability map, we introduce $\mathbf{P}_{\mathrm{A},l}$ as a latent variable. We introduce a *saliency transformation* module, which takes the probability map to generate an updated input image, i.e., $\mathbf{I}_{\mathrm{A},l} = \mathbf{X}_{\mathrm{A},l} \odot \mathbf{g}\big(\mathbf{P}_{\mathrm{A},l}; \boldsymbol{\eta}\big)$, and uses the updated input $\mathbf{I}_{\mathrm{A},l}$ to replace $\mathbf{X}_{\mathrm{A},l}$. Here $\mathbf{g}[\cdot; \boldsymbol{\eta}]$ is the transformation function with parameters $\boldsymbol{\eta}$, and $\odot$ denotes element-wise product, i.e., the transformation function adds spatial weights to the original input image. Thus, the segmentation process becomes:

$$\mathbf{P}_{\mathrm{A},l} = \mathbf{f}\big[\mathbf{X}_{\mathrm{A},l} \odot \mathbf{g}\big(\mathbf{P}_{\mathrm{A},l}; \boldsymbol{\eta}\big); \boldsymbol{\theta}\big]. \tag{3.4}$$

This is a recurrent neural network. Note that the saliency transformation function $\mathbf{g}[\cdot, \boldsymbol{\eta}]$ needs to be differentiable so that the entire recurrent network can be optimized

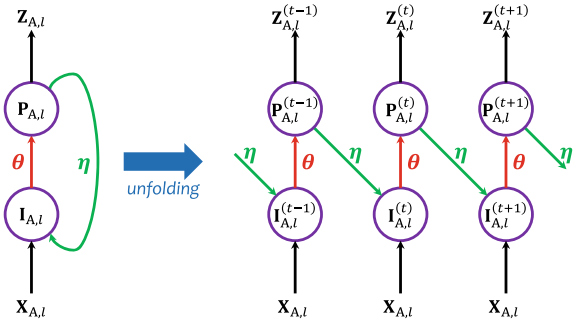

Fig. 3.6 We formulate our approach into a recurrent network, and unfold it for optimization and inference

in an end-to-end manner. As $\mathbf{X}_{\mathrm{A},l}$ and $\mathbf{P}_{\mathrm{A},l}$ share the same spatial dimensionality, we set $\mathbf{g}[\cdot, \boldsymbol{\eta}]$ to be a *size-preserved* convolution, which allows the weight added to each pixel to be determined by the segmentation probabilities in a small neighborhood around it. As we will show in the experimental section (see Fig. 3.9), the learned convolutional kernels are able to extract complementary information to help the next iteration.

To optimize Eq. (3.4), we unfold the recurrent network into a plain form (see Fig. 3.6). Given an input image $\mathbf{X}_{\mathrm{A},l}$ and an integer T which is the maximal number of iterations, we update $\mathbf{I}_{\mathrm{A},l}^{(t)}$ and $\mathbf{P}_{\mathrm{A},l}^{(t)}$, $t = 0, 1, \ldots, T$:

$$\mathbf{I}_{\mathrm{A},l}^{(t)} = \mathbf{X}_{\mathrm{A},l} \odot \mathbf{g}\left(\mathbf{P}_{\mathrm{A},l}^{(t-1)}; \boldsymbol{\eta}\right), \tag{3.5}$$

$$\mathbf{P}_{\mathrm{A},l}^{(t)} = \mathbf{f}\left[\mathbf{I}_{\mathrm{A},l}^{(t)}; \boldsymbol{\theta}\right]. \tag{3.6}$$

Note that the original input image $\mathbf{X}_{\mathrm{A},l}$ does not change, and the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ are shared by all iterations. At $t = 0$, we directly set $\mathbf{I}_{\mathrm{A},l}^{(0)} = \mathbf{X}_{\mathrm{A},l}$.

When segmentation masks $\mathbf{P}_{\mathrm{A},l}^{(t)}$ ($t = 0, 1, \ldots, T-1$) are available for reference, deep networks benefit considerably from a shrunk input region especially when the target organ is very small. Thus, we define a *cropping* function $\mathrm{Crop}\left[\cdot; \mathbf{P}_{\mathrm{A},l}^{(t)}\right]$, which takes $\mathbf{P}_{\mathrm{A},l}^{(t)}$ as the *reference map*, binarizes it into $\mathbf{Z}_{\mathrm{A},l}^{(t)} = \mathbb{I}\left[\mathbf{P}_{\mathrm{A},l}^{(t)} \geqslant 0.5\right]$, finds the minimal rectangle covering all the activated pixels, and adds a K-pixel-wide margin (padding) around it. We fix K to be 20; our algorithm is not sensitive to this parameter.

Finally note that $\mathbf{I}_{\mathrm{A},l}^{(0)}$, the original input (the entire 2D slice), is much larger than the cropped inputs $\mathbf{I}_{\mathrm{A},l}^{(t)}$ for $t > 0$. We train two FCNs to deal with such a major difference in input data. The first one is named the *coarse-scaled* segmentation network, which is used *only* in the first iteration. The second one, the *fine-scaled* segmentation network, takes the charge of all the remaining iterations. We denote their parameters by $\boldsymbol{\theta}^{\mathrm{C}}$ and $\boldsymbol{\theta}^{\mathrm{F}}$, respectively. These two FCNs are optimized jointly.

We compute a DSC-loss term on each probability map $\mathbf{P}_{\mathrm{A},l}^{(t)}$, $t = 0, 1, \ldots, T$, and denote it by $\mathcal{L}\left\{\mathbf{Y}_{\mathrm{A},l}, \mathbf{P}_{\mathrm{A},l}^{(t)}\right\}$. Here, $\mathbf{Y}_{\mathrm{A},l}$ is the ground-truth segmentation mask, and $\mathcal{L}\{\mathbf{Y}, \mathbf{P}\} = 1 - \frac{2\times\sum_i Y_i P_i}{\sum_i Y_i + P_i}$ is based on the *soft* version of DSC [31]. Our goal is to minimize the overall loss:

$$\mathcal{L} = \sum_{t=0}^{T} \lambda_t \cdot \mathcal{L}\left\{\mathbf{Y}_{\mathrm{A},l}^{(t)}, \mathbf{Z}_{\mathrm{A},l}^{(t)}\right\}. \tag{3.7}$$

This leads to joint optimization over all iterations, which involves network parameters $\boldsymbol{\theta}^{\mathrm{C}}, \boldsymbol{\theta}^{\mathrm{F}}$, and transformation parameters $\boldsymbol{\eta}$. $\{\lambda_t\}_{t=0}^{T}$ controls the tradeoff among all loss terms. We set $2\lambda_0 = \lambda_1 = \cdots = \lambda_T = 2/(2T+1)$ so as to encourage accurate fine-scaled segmentation.

3.4.2 Training and Testing

The training phase is aimed at minimizing the loss function $\mathcal{L}$, defined in Eq. (3.7), which is differentiable with respect to all parameters. In the early training stages, the coarse-scaled network cannot generate reasonable probability maps. To prevent the fine-scaled network from being confused by inaccurate input regions, we use the ground-truth mask $\mathbf{Y}_{\mathrm{A},l}$ as the reference map. After a sufficient number of training, we resume using $\mathbf{P}_{\mathrm{A},l}^{(t)}$ instead of $\mathbf{Y}_{\mathrm{A},l}$. In Sect. 3.5.3.1, we will see that this "fine-tuning" strategy improves segmentation accuracy considerably.

Algorithm 2 The Testing Phase for RSTN

Require: input volume $\mathbf{X}$, viewpoint $\mathcal{V} = \{\mathrm{C}, \mathrm{S}, \mathrm{A}\}$
Require: parameters $\boldsymbol{\theta}_{\mathrm{v}}^{\mathrm{C}}$
Require: $\boldsymbol{\theta}_{\mathrm{v}}^{\mathrm{F}}$ and $\boldsymbol{\eta}_{\mathrm{v}}$, $\mathrm{v} \in \mathcal{V}$;
Require: max number of iterations T, threshold thr;
$t \leftarrow 0, \mathbf{I}_{\mathrm{v}}^{(0)}$
$\leftarrow \mathbf{X}$, $\mathrm{v} \in \mathcal{V}$;
$\mathbf{P}_{\mathrm{v},l}^{(0)} \leftarrow \mathbf{f}\left[\mathbf{I}_{\mathrm{v},l}^{(0)}; \boldsymbol{\theta}_{\mathrm{v}}^{\mathrm{C}}\right]$, $\mathrm{v} \in \mathcal{V}, \forall l$;
$\mathbf{P}^{(0)} = \frac{\mathbf{P}_{\mathrm{C}}^{(0)} + \mathbf{P}_{\mathrm{S}}^{(0)} + \mathbf{P}_{\mathrm{A}}^{(0)}}{3}$, $\mathbf{Z}^{(0)} = \mathbb{I}\left[\mathbf{P}^{(0)} \geqslant 0.5\right]$;
repeat
  $t \leftarrow t + 1$;
  $\mathbf{I}_{\mathrm{v},l}^{(t)} \leftarrow \mathbf{X}_{\mathrm{v},l} \odot \mathbf{g}\left(\mathbf{P}_{\mathrm{v},l}^{(t-1)}; \boldsymbol{\eta}\right)$, $\mathrm{v} \in \mathcal{V}, \forall l$;
  $\mathbf{P}_{\mathrm{v},l}^{(t)} \leftarrow \mathbf{f}\left[\mathrm{Crop}\left[\mathbf{I}_{\mathrm{v},l}^{(t)}; \mathbf{P}_{\mathrm{v},l}^{(t-1)}\right]; \boldsymbol{\theta}_{\mathrm{v}}^{\mathrm{F}}\right]$, $\mathrm{v} \in \mathcal{V}, \forall l$;
  $\mathbf{P}^{(t)} = \frac{\mathbf{P}_{\mathrm{C}}^{(t)} + \mathbf{P}_{\mathrm{S}}^{(t)} + \mathbf{P}_{\mathrm{A}}^{(t)}}{3}$, $\mathbf{Z}^{(t)} = \mathbb{I}\left[\mathbf{P}^{(t)} \geqslant 0.5\right]$;
until $t = T$ **or** $\mathrm{DSC}\left\{\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\right\} \geqslant \mathrm{thr}$
return $\mathbf{Z} \leftarrow \mathbf{Z}^{(t)}$.

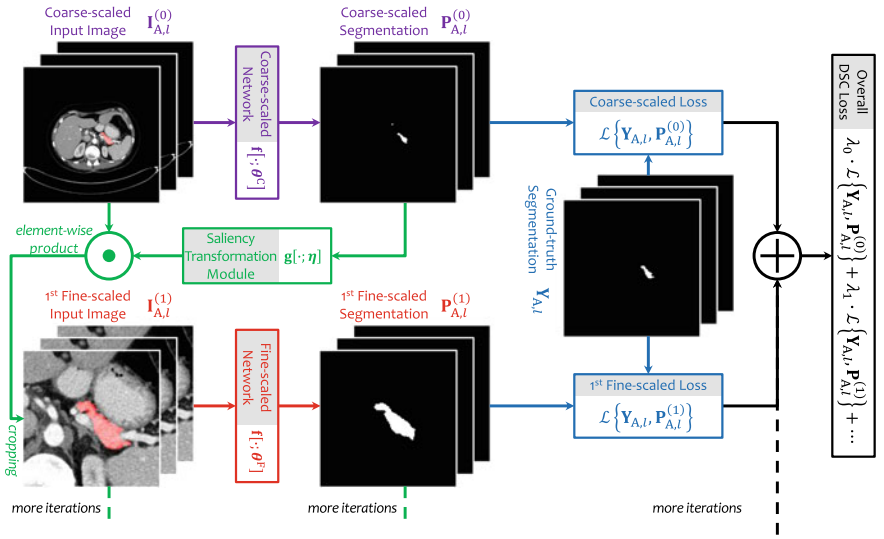

Fig. 3.7 Illustration of the training process (best viewed in color). We display an input image along the *axial* view which contains 3 neighboring slices. To save space, we only plot the coarse stage and the first iteration in the fine stage

Due to the limitation in GPU memory, in each mini-batch containing one training sample, we set T to be the maximal integer (not larger than 5) so that we can fit the entire framework into the GPU memory. The overall framework is illustrated in Fig. 3.7. As a side note, we find that setting $T \equiv 1$ also produces high accuracy, suggesting that major improvement is brought by joint optimization.

The testing phase follows the flowchart described in Algorithm 2. There are two minor differences from the training phase. First, as the ground-truth segmentation mask $\mathbf{Y}_{\mathrm{A},l}$ is not available, the probability map $\mathbf{P}^{(t)}_{\mathrm{A},l}$ is always taken as the reference map for image cropping. Second, the number of iterations is no longer limited by the GPU memory, as the intermediate outputs can be discarded on the way. In practice, we terminate our algorithm when the similarity of two consecutive predictions, measured by $\mathrm{DSC}\{\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\} = \frac{2\times\sum_i Z_i^{(t-1)} Z_i^{(t)}}{\sum_i Z_i^{(t-1)} + Z_i^{(t)}}$, reaches a threshold thr, or a fixed number (T) of iterations are executed. We will discuss these parameters in Sect. 3.5.3.3.

3.4.3 *Application to Pancreatic Cyst Segmentation*

We follow Sect. 3.3.3 to use a multi-stage approach, which first finds the regular organ (pancreas), and then locates the neoplasm (cyst) by referring to that organ. A four-stage strategy is also adopted, i.e., coarse-scaled and fine-scaled pancreas segmentation, as well as coarse-scaled and fine-scaled cyst segmentation. This can

be implemented by two RSTN modules, where the first RSTN segments the pancreas given the CT images while the second segments the pancreatic cyst given the pancreas-cropped region.

3.5 Pancreas Segmentation Experiments

3.5.1 Dataset and Evaluation

We evaluate our approach on the NIH *pancreas* segmentation dataset [35], which contains 82 contrast-enhanced abdominal CT volumes. The resolution of each scan is $512 \times 512 \times L$, where $L \in [181, 466]$ is the number of slices along the long axis of the body. The distance between neighboring voxels ranges from 0.5 to 1.0 mm.

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 apply cross-validation, i.e., training the models on 3 out of 4 subsets and testing them on the remaining one. We measure the segmentation accuracy by computing the Dice-Sørensen coefficient (DSC) for each sample, and report the average and standard deviation over all 82 cases.

3.5.2 Evaluation of the Step-Wise Coarse-to-Fine Approach

We initialize both networks using the FCN-8s model [29] pretrained on the PascalVOC image segmentation task. The coarse-scaled model is fine-tuned with a learning rate of 10^{-5} for 80,000 iterations, and the fine-scaled model undergoes 60,000 iterations with a learning rate of 10^{-4}. Each mini-batch contains one training sample (a 2D image sliced from a 3D volume).

We first evaluate the baseline (coarse-scaled) approach. Using the coarse-scaled models trained from three different views (i.e., $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$), we obtain $66.88\% \pm 11.08\%$, $71.41\% \pm 11.12\%$ and $73.08\% \pm 9.60\%$ average DSC, respectively. Fusing these three models via majority voting yields $75.74 \pm 10.47\%$, suggesting that complementary information is captured by different views. This is used as the starting point $\mathbf{Z}^{(0)}$ for the later iterations.

To apply the fixed-point model for segmentation, we first compute $d^{(t)}$ to observe the convergence of the iterations. After 10 iterations, the average $d^{(t)}$ value over all samples is 0.9767, the median is 0.9794, and the minimum is 0.9362. These numbers indicate that the iteration process is generally stable.

Now, we investigate the fixed-point model using the threshold $R = 0.95$ and the maximal number of iterations $T = 10$. The average DSC is boosted by 6.63%, which is impressive given the relatively high baseline (75.74%). This verifies our hypothesis, i.e., a fine-scaled model depicts a small organ more accurately.

Table 3.1 Segmentation accuracy (measured by DSC, %) reported by different approaches. We start from initial (coarse) segmentation $\mathbf{Z}^{(0)}$, 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 to generate the bounding box in testing. We also compare our results with the state-of-the-art [35, 36], demonstrating our advantage over all statistics

Method	Mean DSC	# iterations	Max DSC	Min DSC
Roth et al., MICCAI'2015 [35]	71.42 ± 10.11	–	86.29	23.99
Roth et al., MICCAI'2016 [36]	78.01 ± 8.20	–	88.65	34.11
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 $d_t > 0.99$	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

We also summarize the results generated by different terminating conditions in Table 3.1. We find that performing merely 1 iteration is enough to significantly boost the segmentation accuracy (+6.42%). However, more iterations help to improve the accuracy of the worst case, as for some challenging cases (e.g., Case #09, see Fig. 3.8), the missing parts in coarse segmentation are recovered gradually. The best average accuracy comes from setting $R = 0.95$. Using a larger threshold (e.g., 0.99) does not produce accuracy gain, but requires more iterations and, consequently, more computation at the testing stage. In average, it takes less than 3 iterations to reach the threshold 0.95. On a modern GPU, we need about 3 min on each testing sample, comparable to recent work [36], but we report much higher segmentation accuracy (82.37% vs. 78.01%).

As a diagnostic experiment, we use the ground-truth (oracle) bounding box of each testing case to generate the input volume. This results in an 83.18% average accuracy (no iteration is needed in this case). By comparison, we report a comparable 82.37% average accuracy, indicating that our approach has almost reached the upper bound of the current deep segmentation network.

We also compare our segmentation results with the state-of-the-art approaches. Using DSC as the evaluation metric, our approach outperforms the recent published

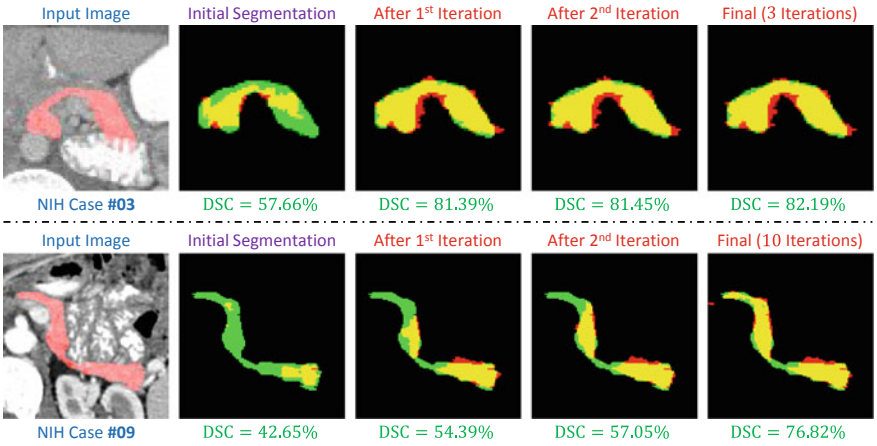

Fig. 3.8 Examples of segmentation results throughout the iteration process (best viewed in color). We only show a small region covering the pancreas in the axial view. The terminating condition is $d^{(t)} \geqslant 0.95$. Red, green and yellow indicate the prediction, ground-truth and overlapped regions, respectively

work [36] 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 our approach is more stable. We also implement a recently published coarse-to-fine approach [50], and get a 77.89% average accuracy. In particular, [36] reported 34.11% for the worst case (some previous work [6, 45] reported even lower numbers), and this number is boosted considerably to 62.43% by our approach. We point out that these improvements are mainly due to the fine-tuning iterations. Without it, the average accuracy is 75.74%, and the accuracy on the worst case is merely 39.99%. Figure 3.8 shows examples on how the segmentation quality is improved in two challenging cases.

3.5.3 Evaluation of the End-to-End Coarse-to-Fine Approach

3.5.3.1 Different Settings

We initialize the up-sampling layers in FCN-8s model [29] pretrained on PascalVOC [9] with random weights, set the learning rate to be 10^{-4} and run 80,000 iterations. Different options are evaluated, including using different kernel sizes in saliency transformation, and whether to fine-tune the models using the predicted segmentations as reference maps (see the description in Sect. 3.4.2). Quantitative results are summarized in Table 3.2.

Table 3.2 Accuracy (DSC, %) comparison of different settings of our approach. Please see the texts in Sect. 3.5.3.1 for detailed descriptions of these variants

Model	Average	Max	Min
3×3 kernels in saliency transformation (basic model)	83.47 ± 5.78	90.63	57.85
1×1 kernels in saliency transformation	82.85 ± 6.68	90.40	53.44
5×5 kernels in saliency transformation	83.64 ± 5.29	90.35	66.35
Two-layer saliency transformation (3×3 kernels)	83.93 ± 5.43	90.52	64.78
Fine-tuning with noisy data (3×3 kernels)	83.99 ± 5.09	90.57	65.05

As the saliency transformation module is implemented by a size-preserved convolution (see Sect. 3.4.1), the size of convolutional kernels determines the range that a pixel can use to judge its saliency. In general, a larger kernel size improves segmentation accuracy (3×3 works significantly better than 1×1), but we observe the marginal effect: the improvement of 5×5 over 3×3 is limited. As we use 7×7 kernels, the segmentation accuracy is slightly lower than that of 5×5. This may be caused by the larger number of parameters introduced to this module. Another way of increasing the receptive field size is to use two convolutional layers with 3×3 kernels. This strategy, while containing a smaller number of parameters, works even better than using one 5×5 layer. But, we do not add more layers, as the performance saturates while computational costs increase.

As described in Sect. 3.4.2, we fine-tune these models with images cropped from the coarse-scaled segmentation mask. This is to adjust the models to the testing phase, in which the ground-truth mask is unknown, so that the fine-scaled segmentation needs to start with, and be able to revise the coarse-scaled segmentation mask. We use a smaller learning rate (10^{-6}) and run another 40,000 iterations. This strategy not only reports 0.52% overall accuracy gain, but also alleviates over-fitting (see Sect. 3.5.3.3).

In summary, all these variants produce higher accuracy than our step-wise coarse-to-fine approach (82.37%), which verifies that the major contribution of our end-to-end approach comes from our recurrent framework which enables joint optimization. In the later experiments, we inherit the best variant learned from this section, including in a large-scale multi-organ dataset (see Sect. 3.6). That is to say, we use two 3×3 convolutional layers for saliency transformation, and fine-tune the models with coarse-scaled segmentation. This setting produces an average accuracy of 84.50%, as shown in Table 3.3.

3.5.3.2 Performance Comparison

We show that our end-to-end coarse-to-fine approach works better than the step-wise coarse-to-fine approach. As shown in Table 3.3, the average improvement over 82 cases is $2.13 \pm 2.67\%$. The standard deviations (5.68% of step-wise approach and

Table 3.3 Accuracy (DSC, %) comparison between our approach and the state of the art on the NIH *pancreas* segmentation dataset [35]

Approach	Average	Max	Min
Roth et al. [35]	71.42 ± 10.11	86.29	23.99
Roth et al. [36]	78.01 ± 8.20	88.65	34.11
Zhang et al. [50]	77.89 ± 8.52	89.17	43.67
Roth et al. [37]	81.27 ± 6.27	88.96	50.69
Cai et al. [3]	82.4 ± 6.7	90.1	60.0
Our step-wise approach	82.37 ± 5.68	90.85	62.43
Our end-to-end approach	**84.50** ± 4.97	**91.02**	**62.81**

4.97% of end-to-end approach) are mainly caused by the difference in scanning and labeling qualities. A case-by-case study reveals that our end-to-end approach reports higher accuracies on 67 out of 82 cases, with the largest advantage being +17.60% and the largest deficit being merely −3.85%. We analyze the sources of improvement in Sect. 3.5.3.3.

We briefly discuss the advantages and disadvantages of using 3D networks. 3D networks capture richer contextual information, but also require training more parameters. Our 2D approach makes use of 3D contexts more efficiently. At the end of each iteration, predictions from three views are fused, and thus the saliency transformation module carries these informations to the next iteration. We implement VNet [31], and obtain an average accuracy of 83.18% with an 3D *ground-truth* bounding box provided for each case. Without the ground-truth, a sliding-window process is required which is really slow—an average of 5 min on a Titan-X Pascal GPU. In comparison, our end-to-end approach needs 1.3 min, slower than our step-wise approach (0.9 min), but faster than other 2D approaches [35, 36] (2–3 min).

3.5.3.3 Diagnosis

Joint Optimization and Mutli-stage Cues

Our end-to-end approach enables joint training, which improves both the coarse and fine stages individually. We denote the two networks trained by our step-wise approach by $\mathbb{I}^{\mathrm{C}}$ and $\mathbb{I}^{\mathrm{F}}$, and similarly, those trained in our approach by $\mathbb{J}^{\mathrm{C}}$ and $\mathbb{J}^{\mathrm{F}}$, respectively. In the coarse stage, $\mathbb{I}^{\mathrm{C}}$ reports 75.74% and $\mathbb{J}^{\mathrm{C}}$ reports 78.23%. In the fine stage, applying $\mathbb{J}^{\mathrm{F}}$ on top of the output of $\mathbb{I}^{\mathrm{C}}$ gets 83.80%, which is considerably higher than 82.37% ($\mathbb{I}^{\mathrm{F}}$ on top of $\mathbb{I}^{\mathrm{C}}$) but lower than 84.50% ($\mathbb{J}^{\mathrm{F}}$ on top of $\mathbb{J}^{\mathrm{C}}$). Therefore, we conclude that both the coarse-scaled and fine-scaled networks benefit from joint optimization. A stronger coarse stage provides a better starting point, and a stronger fine stage improves the upper bound.

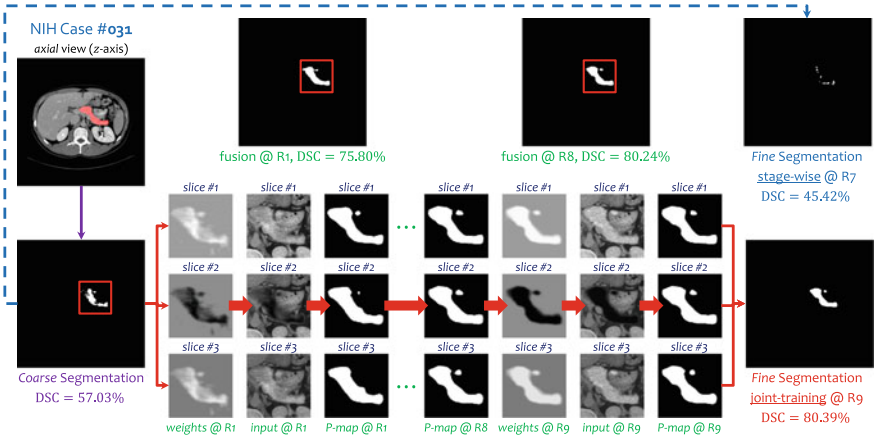

Fig. 3.9 Visualization of how recurrent saliency transformation works in coarse-to-fine segmentation (best viewed in color). Segmentation accuracy is largely improved by making use of the probability map from the previous iteration to help the current iteration. Note that three weight maps capture different visual cues, with two of them focused on the foreground region, and the remaining one focused on the background region

In Fig. 3.9, we visualize how the recurrent network assists segmentation by incorporating multi-stage visual cues. It is interesting to see that in saliency transformation, different channels deliver complementary information, i.e., two of them focus on the target organ, and the remaining one adds most weights to the background region. Similar phenomena happen in the models trained in different viewpoints and different folds. This reveals that except for foreground, background and boundary also contribute to visual recognition [54].

Convergence

We study convergence, which is a very important criterion to judge the reliability of our end-to-end approach. We choose the best model reporting an average accuracy of 84.50%, and record the inter-iteration DSC throughout the testing process: $d^{(t)} = \text{DSC}\{\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\} = \frac{2\times\sum_i Z_i^{(t-1)} Z_i^{(t)}}{\sum_i Z_i^{(t-1)} + Z_i^{(t)}}$.

After 1, 2, 3, 5, and 10 iterations, these numbers are 0.9037, 0.9677, 0.9814, 0.9908, and 0.9964 for our approach, and 0.8286, 0.9477, 0.9661, 0.9743, and 0.9774 for our step-wise approach, respectively. Each number reported by our end-to-end approach is considerably higher than that by the step-wise approach. The better convergence property provides us with the opportunity to set a more strict terminating condition, e.g., using thr $= 0.99$ rather than thr $= 0.95$.

When the threshold is increased from 0.95 to 0.99 in our end-to-end approach, 80 out of 82 cases converge (in an average of 5.22 iterations), and the average accuracy is improved from 83.93% to 84.50%. On a Titan-X Pascal GPU, one iteration takes 0.2 min, so using thr $= 0.99$ requires an average of 1.3 min in each testing case.

The Over-Fitting Issue

Finally, we investigate the over-fitting issue of our end-to-end approach by making use of *oracle* information in the testing process. We use the ground-truth bounding box *on each slice*, which is used to crop the input region in *every* iteration. Note that annotating a bounding box in each slice is expensive and thus not applicable in real-world clinical applications. This experiment is aimed at exploring the upper bound of our segmentation networks under perfect localization.

With oracle information provided, our best model reports 86.37%, which is considerably higher than the number (84.50%) without using oracle information. If we do not fine-tune the networks using coarse-scaled segmentation (see Table 3.2), the above numbers are 86.26% and 83.68%, respectively. This is to say, fine-tuning prevents our model from relying on the ground-truth mask. It not only improves the average accuracy, but also alleviates over-fitting (the disadvantage of our model against that with oracle information is decreased by 0.67%).

3.6 JHMI Multi-organ Segmentation Experiments

To verify that our approach can be applied to other organs, the radiologists in our team collect a large dataset which contains 200 CT scans, 11 abdominal organs and 5 blood vessels. This corpus took 4 full-time radiologists around 3 months to annotate. To the best of our knowledge, this dataset is larger and contains more organs than any public datasets. We choose 5 most challenging targets including the *pancreas* and a blood vessel, as well as two *kidneys* which are relatively easier. Other easy organs such as the *liver* are ignored. To the best of our knowledge, some of these organs were never investigated before, but they are important in diagnosing pancreatic diseases and detecting the pancreatic cancer at an early stage. We randomly partition the dataset into fourfold for cross-validation. Each organ is trained and tested individually. When a pixel is predicted as more than one organs, we choose the one with the largest confidence score.

Table 3.4 Comparison of coarse-scaled (**C**) and fine-scaled (**F**) segmentation by our step-wise approach and end-to-end approach on our JHMI multi-organ dataset. A fine-scaled accuracy is indicated by ♯ if it is lower than the coarse-scaled one. The *pancreas* segmentation accuracies are higher than those in Table 3.3, due to the increased number of training samples and the higher resolution in CT scans

Organ	Stepwise-**C**	Stepwise-**F**	End-to-end-**C**	End-to-end-**F**
adrenal g.	57.38	61.65	60.70	**63.76**
duodenum	67.42	69.39	71.40	**73.42**
gallbladder	82.57	♯82.12	87.08	**87.10**
inferior v.c.	71.77	♯71.15	79.12	**79.69**
kidney l.	92.56	92.78	96.08	**96.21**
kidney r.	94.98	95.39	95.80	**95.97**
pancreas	83.68	85.79	86.09	**87.60**

Results of our two approaches are summarized in Table 3.4. Our end-to-end approach performs generally better than the step-wise approach. It reports a 4.29% average improvement over 5 challenging organs (the *kidneys* excluded). For some organs, e.g., the *gallbladder*, we do not observe significant accuracy gain by iterations.

3.7 JHMI Pancreatic Cyst Segmentation Experiments

Finally, we evaluate our approach on a cyst dataset collected by the radiologists in our team. This dataset contains 131 contrast-enhanced abdominal CT volumes, and each of them is manually labeled with both pancreas and pancreatic cyst masks. The resolution of each CT scan is $512 \times 512 \times L$, where $L \in [358, 1121]$ is the number of sampling slices along the long axis of the body. The slice thickness varies from 0.5 to 1.0 mm. We split the dataset into 4 fixed folds, and each of them contains approximately the same number of samples. We apply cross- validation, i.e., training our approach on 3 out of 4 folds and testing it on the remaining one. The same as before, we measure the segmentation accuracy by computing the Dice-Sørensen Coefficient (DSC) for each 3D volume. We report the average DSC score together with other statistics over all 131 testing cases from 4 testing folds.

We report both pancreas and cyst segmentation results in Table 3.5, where we summarize the results of pancreas segmentation, pancreatic cyst segmentation without pancreas supervision (i.e., two-stage coarse-to-fine approach, w/o deep supervision), and pancreatic cyst segmentation with pancreas supervision (i.e., four-stage strategy, w/deep supervision). It is interesting to see that without deep supervision, our two approaches perform comparably with each other, but with deep supervision, end-to-end approach works better than the step-wise one. This is because, a much better pancreas segmentation result (i.e., 83.81% compared with 79.32%) provides more accurate contextual information for cyst segmentation. In addition, our

Table 3.5 Accuracy (DSC, %) comparison on different targets (*pancreas* or *cyst*) and different approaches. For *cyst* segmentation, w/o Deep Supervision means directly apply our coarse-to-fine approaches on cyst segmentation, given the whole CT image, while w/Deep Supervision means segmenting the *pancreas* first, and then segmenting the *cyst* in the input image cropped by the *pancreas* region

Target	Method	Average	Max	Min
pancreas	Step-wise	79.23 ± 9.72	93.82	**69.54**
pancreas	End-to-end	**83.81** ± 10.51	**94.34**	20.77
cyst	Step-wise, w/o deep supervision	60.46 ± 31.37	95.67	0.00
cyst	End-to-end, w/o deep supervision	**60.73** ± 32.46	**96.50**	0.00
cyst	Step-wise, w/deep supervision	63.44 ± 27.71	95.55	0.00
cyst	End-to-end, w/deep supervision	**67.19** ± 27.91	**96.05**	0.00

approaches yield even better results by adopting a stronger backbone, e.g., under the setting of Step-Wise, w/Deep Supervision, when we employ DeepLab [5] as the backbone network in the coarse stage for pancreas segmentation, we can even achieve $69.38 \pm 27.60\%$ in DSC for cyst segmentation.

To the best of our knowledge, pancreatic cyst segmentation has been little studied previously. A competitor is [7] published in 2016, which combines random walk and region growth for segmentation. However, it requires the user to annotate the region of interest (ROI) beforehand, and provide interactive annotations on foreground/background voxels throughout the segmentation process. In comparison, our approaches can be widely applied to automatic diagnosis, especially for the common users without professional knowledge in medicine.

3.8 Conclusions

This work is motivated by the difficulty of small target segmentation, which is required to focus on a local input region. Two coarse-to-fine approaches are proposed, namely, step-wise coarse-to-fine and end-to-end coarse-to-fine. Step-wise algorithm is formulated as a fixed-point model taking the segmentation mask as both input and output. End-to-end algorithm jointly optimize over two networks, and generally achieves better results compared with the step-wise one.

Our approaches are applied to three datasets for *pancreas* segmentation, multi-organ segmentation, and pancreatic cyst segmentation, and outperforms the baseline (the state-of-the-art) significantly. Confirmed by the radiologists in our team, these segmentation results are helpful to computer-assisted clinical diagnoses.

References

1. Ali A, Farag A, El-Baz A (2007) Graph cuts framework for kidney segmentation with prior shape constraints. In: International conference on medical image computing and computer-assisted intervention
2. Brosch T, Tang L, Yoo Y, Li D, Traboulsee A, Tam R (2016) Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple Sclerosis Lesion segmentation. IEEE Trans Med Imaging 35(5):1229–1239
3. Cai J, Lu L, Xie Y, Xing F, Yang L (2017) Improving deep pancreas segmentation in CT and MRI images via recurrent neural contextual learning and direct loss function. In: International conference on medical image computing and computer-assisted intervention
4. Chen H, Dou Q, Wang X, Qin J, Heng P (2016) Mitosis detection in breast cancer histology images via deep Cascaded networks. In: AAAI conference on artificial intelligence
5. Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille A (2015) Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: International conference on learning representations
6. Chu C, Oda M, Kitasaka T, Misawa K, Fujiwara M, Hayashi Y, Nimura Y, Rueckert D, Mori K (2013) Multi-organ segmentation based on spatially-divided probabilistic Atlas from 3D

abdominal CT images. In: International conference on medical image computing and computer-assisted intervention
7. Dmitriev K, Gutenko I, Nadeem S, Kaufman A (2016) Pancreas and cyst segmentation. In: Medical imaging 2016: image processing, vol 9784, pp 97842C
8. Dou Q, Chen H, Jin Y, Yu L, Qin J, Heng P (2016) 3D deeply supervised network for automatic liver segmentation from CT volumes. In: International conference on medical image computing and computer-assisted intervention
9. Everingham M, Van Gool L, Williams C, Winn J, Zisserman A (2010) The Pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
10. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Computer vision and pattern recognition
11. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: International conference on acoustics, speech and signal processing
12. Harrison A, Xu Z, George K, Lu L, Summers R, Mollura D (2017) Progressive and multi-path holistically nested neural networks for pathological lung segmentation from CT images. In: International conference on medical image computing and computer-assisted intervention
13. Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, Pal C, Jodoin P, Larochelle H (2017) Brain tumor segmentation with deep neural networks. In: Medical image analysis
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Computer vision and pattern recognition
15. Heimann T, Van Ginneken B, Styner M, Arzhaeva Y, Aurich V, Bauer C, Beck A, Becker C, Beichel R, Bekes G et al (2009) Comparison and evaluation of methods for liver segmentation from CT datasets. IEEE Trans Med Imaging 28(8):1251–1265
16. Hu S, Hoffman E, Reinhardt J (2001) Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images. IEEE Trans Med Imaging 20(6):490–498
17. Kamnitsas K, Ledig C, Newcombe V, Simpson J, Kane A, Menon D, Rueckert D, Glocker B (2017) Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med Image Anal 36:61–78
18. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems
19. Kuen J, Wang Z, Wang G (2016) Recurrent attentional networks for saliency detection. In: Computer vision and pattern recognition
20. Lai M (2015) Deep learning for medical image segmentation. arXiv:1505.02000
21. Lee C, Xie S, Gallagher P, Zhang Z, Tu Z (2015) Deeply-supervised nets. In: International conference on artificial intelligence and statistics
22. Li G, Xie Y, Lin L, Yu Y (2017) Instance-level salient object segmentation. In: Computer vision and pattern recognition
23. Li Q, Wang J, Wipf D, Tu Z (2013) Fixed-point model for structured labeling. In: International conference on machine learning
24. Liang M, Hu X (2015) Recurrent convolutional neural network for object recognition. In: Computer vision and pattern recognition
25. Lin D, Lei C, Hung S (2006) Computer-aided kidney segmentation on abdominal CT images. IEEE Trans Inf Technol Biomed 10(1):59–65
26. Lin G, Milan A, Shen C, Reid I (2017) RefineNet: multi-path refinement networks with identity mappings for high-resolution semantic segmentation. In: Computer vision and pattern recognition
27. Ling H, Zhou S, Zheng Y, Georgescu B, Suehling M, Comaniciu D (2008) Hierarchical, learning-based automatic liver segmentation. In: Computer vision and pattern recognition
28. Linguraru M, Sandberg J, Li Z, Shah F, Summers R (2010) Automated segmentation and quantification of liver and spleen from CT images using normalized probabilistic Atlases and enhancement estimation. Med Phys 37(2):771–783
29. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Computer vision and pattern recognition

30. Merkow J, Kriegman D, Marsden A, Tu Z (2016) Dense volume-to-volume vascular boundary detection. In: International conference on medical image computing and computer-assisted intervention
31. Milletari F, Navab N, Ahmadi S (2016) V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: International conference on 3d vision
32. Pinheiro P, Collobert R (2014) Recurrent convolutional neural networks for scene labeling. In: International conference on machine learning
33. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems
34. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention
35. Roth H, Lu L, Farag A, Shin H, Liu J, Turkbey E, Summers R (2015) DeepOrgan: multi-level deep convolutional networks for automated pancreas segmentation. In: International conference on medical image computing and computer-assisted intervention
36. Roth H, Lu L, Farag A, Sohn A, Summers R (2016) Spatial aggregation of holistically-nested networks for automated pancreas segmentation. In: International conference on medical image computing and computer-assisted intervention
37. Roth H, Lu L, Lay N, Harrison A, Farag A, Sohn A, Summers R (2017) Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation. arXiv:1702.00045
38. Shen W, Wang B, Jiang Y, Wang Y, Yuille A (2017) Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection. In: International Conference on Computer Vision
39. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations
40. Socher R, Lin C, Manning C, Ng A (2011) Parsing natural scenes and natural language with recursive neural networks. In: International conference on machine learning
41. Tang P, Wang X, Bai S, Shen W, Bai X, Liu W, Yuille AL (2018) PCL: proposal cluster learning for weakly supervised object detection. In: IEEE transaction on pattern analysis and machine intelligence
42. Wang D, Khosla A, Gargeya R, Irshad H, Beck A (2016) Deep learning for identifying metastatic breast cancer. arXiv:1606.05718
43. Wang Y, Zhou Y, Tang P, Shen W, Fishman EK, Yuille AL (2018) Training multi-organ segmentation networks with sample selection by relaxed upper confident bound. In: International conference on medical image computing and computer-assisted intervention
44. Wang Y, Zhou Y, Shen W, Park S, Fishman EK, Yuille AL (2018) Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation. arXiv:1804.08414
45. Wang Z, Bhatia K, Glocker B, Marvao A, Dawes T, Misawa K, Mori K, Rueckert D (2014) Geodesic patch-based segmentation. In: International conference on medical image computing and computer-assisted intervention
46. Xia F, Wang P, Chen L, Yuille A (2016) Zoom better to see clearer: human and object parsing with hierarchical auto-zoom net. In: European Conference on Computer Vision
47. Yu L, Yang X, Chen H, Qin J, Heng P (2017) Volumetric ConvNets with mixed residual connections for automated prostate segmentation from 3D MR images. In: AAAI Conference on Artificial Intelligence
48. Yu Q, Xie L, Wang Y, Zhou Y, Fishman E, Yuille A (2018) Recurrent saliency transformation network: incorporating multi-stage visual cues for small organ segmentation. In: Computer vision and patter recognition
49. Zhang L, Lu L, Summers RM, Kebebew E, Yao J (2017) Personalized pancreatic tumor growth prediction via group learning. In: International conference on medical image computing and computer-assisted intervention

50. Zhang Y, Ying M, Yang L, Ahuja A, Chen D (2016) Coarse-to-Fine stacked fully convolutional nets for lymph node segmentation in ultrasound images. In: IEEE international conference on bioinformatics and biomedicine
51. Zhou Y, Wang Y, Tang P, Bai S, Shen W, Fishman EK, Yuille AL (2019) Semi-supervised multi-organ segmentation via multi-planar co-training. In: IEEE winter conference on applications of computer vision
52. Zhou Y, Xie L, Fishman E, Yuille A (2017) Deep supervision for pancreatic cyst segmentation in abdominal CT scans. In: International conference on medical image computing and computer-assisted intervention
53. Zhou Y, Xie L, Shen W, Wang Y, Fishman E, Yuille A (2017) A fixed-point model for pancreas segmentation in abdominal CT scans. In: International conference on medical image computing and computer-assisted intervention
54. Zhu Z, Xie L, Yuille A (2017) Object recognition with and without objects. In: International joint conference on artificial intelligence