Chapter 4
Volumetric Medical Image Segmentation: A 3D Deep Coarse-to-Fine Framework and Its Adversarial Examples

Yingwei Li, Zhuotun Zhu, Yuyin Zhou, Yingda Xia, Wei Shen, Elliot K. Fishman and Alan L. Yuille

Abstract Although deep neural networks have been a dominant method for many 2D vision tasks, it is still challenging to apply them to 3D tasks, such as medical image segmentation, due to the limited amount of annotated 3D data and limited computational resources. In this chapter, by rethinking the strategy to apply 3D Convolutional Neural Networks to segment medical images, we propose a novel 3D-based coarse-to-fine framework to efficiently tackle these challenges. The proposed 3D-based framework outperforms their 2D counterparts by a large margin since it can leverage the rich spatial information along all three axes. We further analyze the threat of adversarial attacks on the proposed framework and show how to defend against the attack. We conduct experiments on three datasets, the NIH pancreas dataset, the JHMI pancreas dataset and the JHMI pathological cyst dataset, where the first two and the last one contain healthy and pathological pancreases, respectively, and

Y. Li and Z. Zhu contribute equally and are ordered alphabetically. The first part of this work appeared as a conference paper [48], in which Zhuotun Zhu, Yingda Xia, and Wei Shen made contributions to. The second part was contributed by Yingwei Li, Yuyin Zhou, and Wei Shen. Elliot K. Fishman and Alan L. Yuille oversaw the entire project.

Y. Li · Z. Zhu · Y. Zhou · Y. Xia · W. Shen · A. L. Yuille (✉)
Johns Hopkins University, 3400 N Charles St, Baltimore, MD 21218, USA
e-mail: ayuille1@jhu.edu

Y. Li
e-mail: yingwei.li@jhu.edu

Z. Zhu
e-mail: ztzhu@jhu.edu

Y. Zhou
e-mail: yzhou103@jhu.edu

Y. Xia
e-mail: yxia25@jhu.edu

W. Shen
e-mail: wshen10@jhu.edu

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_4

achieve the current state of the art in terms of Dice-Sørensen Coefficient (DSC) on all of them. Especially, on the NIH pancreas dataset, we outperform the previous best by an average of over 2%, and the worst case is improved by 7% to reach almost 70%, which indicates the reliability of our framework in clinical applications.

4.1 Introduction

Driven by the huge demands for computer-aided diagnosis systems, automatic organ segmentation from medical images, such as computed tomography (CT) and magnetic resonance imaging (MRI), has become an active research topic in both the medical image processing and computer vision communities. It is a prerequisite step for many clinical applications, such as diabetes inspection, organic cancer diagnosis, and surgical planning. Therefore, it is well worth exploring automatic segmentation systems to accelerate the computer-aided diagnosis in medical image analysis.

In this chapter, we focus on pancreas segmentation from CT scans, one of the most challenging organ segmentation problems [31, 46]. As shown in Fig. 4.1, the main difficulties stem from three parts: (1) the small size of the pancreas in the whole abdominal CT volume; (2) the large variations in texture, location, shape, and size of the pancreas; (3) the abnormalities, like pancreatic cysts, can alter the appearance of pancreases a lot.

Following the rapid development of deep neural networks [17, 35] and their successes in many computer vision tasks, such as semantic segmentation [4, 21], edge detection [33, 34, 42], and 3D shape retrieval [7, 47], many deep learning-based methods have been proposed for pancreas segmentation and have achieved considerable progress [31, 32, 46]. However, these methods are based on 2D fully convolutional networks (FCNs) [21], which perform segmentation slice by slice while CT volumes are indeed 3D data. Although these 2D methods use strategies

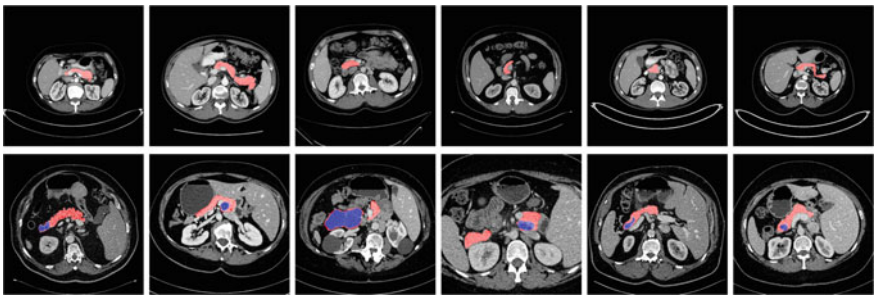

Fig. 4.1 An illustration of normal pancreases on NIH dataset [31] and abnormal cystic pancreases on JHMI dataset [45] shown in the first and second rows, respectively. Normal pancreas regions are masked as red and abnormal pancreas regions are marked as blue. The pancreas usually occupies a small region in a whole CT scan. Best viewed in color

to fuse the output from different 2D views to obtain 3D segmentation results, they inevitably lose some 3D context, which is important for capturing the discriminative features of the pancreas with respect to background regions.

An obstacle to train 3D deep segmentation networks is that it suffers from the "out of memory" problem. 2D FCNs can accept a whole 2D slice as input, but 3D FCNs cannot be fed a whole 3D volume due to the limited GPU memory size. A common solution is to train 3D FCNs from small sub-volumes and test them in a sliding-window manner [1, 3, 5, 24, 43], i.e., performing 3D segmentation on densely and uniformly sampled sub-volumes one by one. Usually, these neighboring sampled sub-volumes overlap with each other to improve the robustness of the final 3D results. It is worth noting that the overlap size is a trade-off between the segmentation accuracy and the time cost. Setting a larger/smaller overlap size generally leads to a better/worse segmentation accuracy but takes more/less time during testing.

To address these issues, we propose a concise and effective framework to train 3D deep networks for pancreas segmentation, which can simultaneously achieve high segmentation accuracy and low time cost. Our framework is formulated into a coarse-to-fine manner. In the training stage, we first train a 3D FCN from the sub-volumes sampled from an entire CT volume. We call this ***ResDSN Coarse*** model, which aims at obtaining the rough location of the target pancreas from the whole CT volume by making full use of the overall 3D context. Then, we train another 3D FCN from the sub-volumes sampled only from the ground truth bounding boxes of the target pancreas. We call this the ***ResDSN Fine*** model, which can refine the segmentation based on the coarse result. In the testing stage, we first apply the coarse model in the sliding-window manner to a whole CT volume to extract the most probable location of the pancreas. Since we only need a rough location for the target pancreas in this step, the overlap size is set to a small value. Afterward, we apply the fine model in the sliding-window manner to the coarse pancreas region, but by setting a larger overlap size. Thus, we can efficiently obtain a fine segmentation result and we call the coarse-to-fine framework by ***ResDSN C2F***.

Note that, the meaning of "coarse-to-fine" in our framework is twofold. First, it means the input region of interests (RoIs) for the ***ResDSN Coarse*** model and the ***ResDSN Fine*** model are different, i.e., a whole CT volume for the former one and a rough bounding box of the target pancreas for the latter one. We refer to this as coarse-to-fine RoIs, which is designed to achieve better segmentation performance. The coarse step removes a large amount of the unrelated background region, then with a relatively smaller region to be sampled as input, the fine step can much more easily learn cues which distinguish the pancreas from the local background, i.e., exploit local context which makes it easier to obtain a more accurate segmentation result. Second, it means the overlap sizes used for the ***ResDSN Coarse*** model and the ***ResDSN Fine*** model during inference are different, i.e., small and large overlap sizes for them, respectively. We refer to this as coarse-to-fine overlap sizes, which is designed for efficient 3D inference.

Recently, it is increasingly realized that deep networks are vulnerable to adversarial examples, i.e., inputs that are almost indistinguishable from natural data which are imperceptible to a human, but yet classified incorrectly by the network [10, 37, 41].

This problem is even more serious for medical learning systems, as they may cause incorrect decisions, which could mislead human doctors. Adversarial examples may be only a small subset of the space of all medical images, so it is possible that they will only rarely occur in real datasets. But, even so, they could potentially have major errors. Analyzing them can help medical imaging researchers to understand more about their deep network-based model, with the ultimate goal of increasing robustness. In this chapter, we generate 3D adversarial examples by the gradient-based methods [10, 18] and investigate the threat of these 3D adversarial examples on our framework. We also show how to defend against these adversarial examples.

The contributions of this chapter can be summarized into two aspects: (1) A novel 3D deep network-based framework which leverages the rich spatial information for medical image segmentation, which achieves the state-of-the-art performance with relatively low time cost on segmenting both normal and abnormal pancreases; (2) A systematic analysis about the threat of 3D adversarial examples on our framework as well as the adversarial defense methods.

The first part of this work appeared as a conference paper [48], in which Zhuotun Zhu, Yingda Xia, and Wei Shen made contributions to. The second part was contributed by Yingwei Li, Yuyin Zhou, and Wei Shen. Elliot K. Fishman and Alan L. Yuille oversaw the entire project. This chapter extends the previous work [48] by including the analysis about the 3D adversarial attacks and defenses for our framework and more experimental results.

4.2 Related Work

4.2.1 Deep Learning-Based Medical Image Segmentation

The medical image analysis community is facing a revolution brought by the fast development of deep networks [17, 35]. Deep convolutional neural networks (CNNs) based methods have dominated the research area of volumetric medical image segmentation in the last few years. Generally speaking, CNN-based methods for volumetric medical image segmentation can be divided into two major categories: 2D CNNs based and 3D CNNs based.

4.2.1.1 2D CNNs for Medical Image Segmentation

2D CNNs based methods [12, 25, 29, 31, 32, 39, 40] performed volumetric segmentation slice by slice from different views, and then fused the 2D segmentation results to obtain a 3D Volumetric Segmentation result. In the early stage, the 2D segmentation-based models were trained from image patches and tested in a patch by patch manner [31], which is time consuming. Since the introduction of fully convolution networks (FCNs) [21], almost all the 2D segmentation methods are built

upon 2D FCNs to perform holistic slice segmentation during both training and testing. Havaei et al. [12] proposed a two-pathway FCN architecture, which exploited both local features as well as more global contextual features simultaneously by the two pathways. Roth et al. [32] performed pancreas segmentation by a holistic learning approach, which first segment pancreas regions by holistically nested networks [42] and then refine them by the boundary maps obtained by robust spatial aggregation using random forest. The U-Net [29] is one of the most popular FCN architectures for medical image segmentation, which is a encoder–decoder network, but with an additional short connection between encoder and decoder paths. Based on the fact that a pancreas only takes up a small fraction of the whole scan, Zhou et al. [46] proposed to find the rough pancreas region and then learn an FCN-based fixed-point model to refine the pancreas region iteratively. Their method is also based on a coarse-to-fine framework, but it only considered coarse-to-fine RoIs. Besides coarse-to-fine RoIs, our coarse-to-fine method also takes coarse-to-fine overlap sizes into account, which is designed specifically for efficient 3D inference.

4.2.1.2 3D CNNs for Medical Image Segmentation

Although 2D CNNs based methods achieved considerable progress, they are not optimal for medical image segmentation, as they cannot make full use of the 3D context encoded in volumetric data. Several 3D CNNs based segmentation methods have been proposed. The 3D U-Net [5] extended the previous 2D U-Net architecture [29] by replacing all 2D operations with their 3D counterparts. Based on the architecture of the 3D U-Net, the V-Net [24] introduced residual structures [13] (short term skip connection) into each stage of the network. Chen et al. [3] proposed a deep voxel-wise residual network for 3D brain segmentation. Both I2I-3D [23] and 3D-DSN [6] included auxiliary supervision via side outputs into their 3D deep networks. Despite the success of 3D CNNs as a technique for segmenting the target organs, such as prostate [24] and kidney [5], very few techniques have been developed for leveraging 3D spatial information on the challenging pancreas segmentation. Gibson et al. [8] proposed the DenseVNet which is, however, constrained to have shallow encoders due to the computationally demanding dense connections. Roth et al. [30] extended 3D U-Net to segment the pancreas, while obtaining good results, this method has the following shortcomings, (1) the input of their networks is fixed to $120 \times 120 \times 120$, which is very computationally demanding due to this large volume size, (2) the rough pancreas bounding box is resampled to a fixed size as their networks input, which loses information and flexibility, and cannot deal with the intrinsic large variations of pancreas in shape and size. Therefore, we propose our 3D coarse-to-fine framework that works on both normal and abnormal CT data to ensure both low computation cost and high pancreas segmentation accuracy.

4.2.2 Adversarial Attacks and Defenses for Medical Image Segmentation Networks

Deep learning has become increasingly adopted within the medical imaging community for a wide range of tasks including classification, segmentation, detection, *etc*. Though achieving tremendous success in various problems, CNNs have been demonstrated to be extremely vulnerable to adversarial examples, i.e., images which are crafted by human-imperceptible perturbations [10, 37, 41]. Xie et al. [41] were the first to make adversarial examples for semantic segmentation, which is directly related to medical image segmentation. Paschali et.al. [27] used the code from Xie et al. [41] and showed that state-of-the-art networks such as Inception [36] and UNet [28] are still extremely susceptible to adversarial examples for skin lesion classification and whole brain segmentation. It was also demonstrated that adversarial examples are superior in pushing a network to its limits and evaluating its robustness in [27]. Additionally, Huang et.al. [14] pointed out that the robustness of deep learning-based reconstruction techniques for limited angle tomography remains a concern due to its vulnerability to adversarial examples. This makes the robustness of neural networks for clinical applications an important unresolved issue.

To alleviate such adversarial effects for clinical applications, we investigate the application of adversarial training [37] for improving the robustness of deep learning algorithms in the medical area. Adversarial training was first proposed by Szegedy et.al. [37] to increase robustness by augmenting training data with adversarial examples. Madry et.al. [22] further validated that adversarially trained models can be robust against white-box attacks, i.e., with knowledge of the model parameters. Note that clinical applications of deep learning require a high level of safety and security [14]. Our experiments empirically demonstrate that adversarial training can be greatly beneficial for improving the robustness of 3D deep learning-based models against adversarial examples.

4.3 Method

4.3.1 A 3D Coarse-to-Fine Framework for Medical Image Segmentation

In this section, we elaborate our 3D coarse-to-fine framework, which includes a ***coarse*** stage and a ***fine*** stage afterward. We first formulate a segmentation model that can be generalized to both ***coarse*** stage and ***fine*** stage. Later in Sects. 4.3.1.1 and 4.3.1.2, we will customize the segmentation model to these two stages, separately.

We denote a 3D CT scan volume by $\mathbf{X}$. This is associated with a human-labeled per-voxel annotation $\mathbf{Y}$, where both $\mathbf{X}$ and $\mathbf{Y}$ have size $W \times H \times D$, which corresponds to axial, sagittal and coronal views, separately. The ground truth segmentation mask

$\mathbf{Y}$ has a binary value y_i, $i = 1, \ldots, WHD$, at each spatial location i where $y_i = 1$ indicates that x_i is a pancreas voxel. Denote a segmentation model by $\mathbb{M} : \mathbf{P} = \mathbf{f}(\mathbf{X}; \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ indicates model parameters and $\mathbf{P}$ means the binary prediction volume. Specifically in a neural network with L layers and parameters $\boldsymbol{\Theta} = \{\mathcal{W}, \mathcal{B}\}$, $\mathcal{W}$ is a set of weights and $\mathcal{B}$ is a set of biases, where $\mathcal{W} = \{\mathbf{W}^1, \mathbf{W}^2, \ldots, \mathbf{W}^L\}$ and $\mathcal{B} = \{\mathbf{B}^1, \mathbf{B}^2, \ldots, \mathbf{B}^L\}$. Given that $p(y_i|x_i; \boldsymbol{\Theta})$ represents the predicted probability of a voxel x_i being what is the labeled class at the final layer of the output, the negative log-likelihood loss can be formulated as

$$\mathcal{L} = \mathcal{L}(\mathbf{X}; \boldsymbol{\Theta}) = -\sum_{x_i \in \mathbf{X}} \log(p(y_i|x_i; \boldsymbol{\Theta})). \tag{4.1}$$

It is also known as the cross-entropy loss in our binary segmentation setting. By thresholding $p(y_i|x_i; \boldsymbol{\Theta})$, we can obtain the binary segmentation mask $\mathbf{P}$.

We also add some auxiliary layers to the neural network, which produces side outputs under deep supervision [20]. These auxiliary layers form a branch network and facilitate feature learning at lower layer of the mainstream network. Each branch network shares the weights of the first d layers from the mainstream network, which is denoted by $\boldsymbol{\Theta}_d = \{\mathcal{W}_d, \mathcal{B}_d\}$ and has its own weights $\widehat{\boldsymbol{\Theta}}_d$ to output the per-voxel prediction. Similarly, the loss of an auxiliary network can be formulated as

$$\mathcal{L}_d(\mathbf{X}; \boldsymbol{\Theta}_d, \widehat{\boldsymbol{\Theta}}_d) = \sum_{x_i \in \mathbf{X}} -\log(p(y_i|x_i; \boldsymbol{\Theta}_d, \widehat{\boldsymbol{\Theta}}_d)), \tag{4.2}$$

which is abbreviated as $\mathcal{L}_d$. Finally, stochastic gradient descent is applied to minimize the negative log-likelihood, which is given by the following regularized objective function:

$$\mathcal{L}_{overall} = \mathcal{L} + \sum_{d \in \mathcal{D}} \xi_d \mathcal{L}_d + \lambda \left(\|\boldsymbol{\Theta}\|^2 + \sum_{d \in \mathcal{D}} \|\widehat{\boldsymbol{\Theta}}_d\| \right)^2, \tag{4.3}$$

where $\mathcal{D}$ is a set of branch networks for auxiliary supervisions, ξ_d balances the importance of each auxiliary network, and l_2 regularization is added to the objective to prevent the networks from overfitting. For notational simplicity, we keep a segmentation model that is obtained from the overall function described in Eq. 4.3 denoted by $\mathbb{M} : \mathbf{P} = \mathbf{f}(\mathbf{X}; \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ includes parameters of the mainstream network and auxiliary networks.

4.3.1.1 Coarse Stage

In the ***coarse*** stage, the input of "ResDSN Coarse" is sampled from the whole CT scan volume denoted by $\mathbf{X}^{\mathrm{C}}$, on which the ***coarse*** segmentation model $\mathbb{M}^{\mathrm{C}} : \mathbf{P}^{\mathrm{C}} = \mathbf{f}^{\mathrm{C}}(\mathbf{X}^{\mathrm{C}}; \boldsymbol{\Theta}^{\mathrm{C}})$ is trained on. All the C superscripts depict the ***coarse*** stage. The goal of this stage is to efficiently produce a rough binary segmentation $\mathbf{P}^{\mathrm{C}}$ from the complex

background, which can get rid of regions that are segmented as non-pancreas with a high confidence to obtain an approximate pancreas volume. Based on this approximate pancreas volume, we can crop from the original input $\mathbf{X}^{\mathrm{C}}$ with a rectangular cube derived from $\mathbf{P}^{\mathrm{C}}$ to obtain a smaller 3D image space $\mathbf{X}^{\mathrm{F}}$, which is surrounded by simplified and less variable context compared with $\mathbf{X}^{\mathrm{C}}$. The mathematic definition of $\mathbf{X}^{\mathrm{F}}$ is formulated as

$$\mathbf{X}^{\mathrm{F}} = \mathrm{Crop}[\mathbf{X}^{\mathrm{C}} \otimes \mathbf{P}^{\mathrm{C}}; \mathbf{P}^{\mathrm{C}}, m], \tag{4.4}$$

where $\otimes$ means an element-wise product. The function $\mathrm{Crop}[\mathbf{X}; \mathbf{P}, m]$ denotes cropping $\mathbf{X}$ via a rectangular cube that covers all the 1's voxels of a binary volume $\mathbf{P}$ added by a padding margin m along three axes. Given $\mathbf{P}$, the functional constraint imposed on $\mathbf{X}$ is that they have exactly the same dimensionality in 3D space. The padding parameter m is empirically determined in experiments, where it is used to better segment the boundary voxels of pancreas during the ***fine*** stage. The Crop operation acts as a dimensionality reduction to facilitate the fine segmentation, which is crucial to cut down the consuming time of segmentation. It is well worth noting that the 3D locations of the rectangular cube which specifies where to crop $\mathbf{X}^{\mathrm{F}}$ from $\mathbf{X}^{\mathrm{C}}$ is recorded to map the ***fine*** segmentation results back their positions in the full CT scan.

4.3.1.2 Fine Stage

In the ***fine*** stage, the input of the ConvNet is sampled from the cropped volume $\mathbf{X}^{\mathrm{F}}$, on which we train the ***fine*** segmentation model $\mathbb{M}^{\mathrm{F}} : \mathbf{P}^{\mathrm{F}} = \mathbf{f}^{\mathrm{F}}\left(\mathbf{X}^{\mathrm{F}}; \boldsymbol{\Theta}^{\mathrm{F}}\right)$, where the F superscripts indicate the ***fine*** stage. The goal of this stage is to refine the coarse segmentation results from previous stage. In practice, $\mathbf{P}^{\mathrm{F}}$ has the same volumetric size of $\mathbf{X}^{\mathrm{F}}$, which is smaller than the original size of $\mathbf{X}^{\mathrm{C}}$.

4.3.1.3 Coarse-to-Fine Segmentation

Our segmentation task is to give a volumetric prediction on every voxel of $\mathbf{X}^{\mathrm{C}}$, so we need to map the $\mathbf{P}^{\mathrm{F}}$ back to exactly the same size of $\mathbf{X}^{\mathrm{C}}$ given by

$$\mathbf{P}^{\mathrm{C2F}} = \mathrm{DeCrop}[\mathbf{P}^{\mathrm{F}} \odot \mathbf{P}^{\mathrm{C}}; \mathbf{X}^{\mathrm{F}}, \mathbf{X}^{\mathrm{C}}], \tag{4.5}$$

where $\mathbf{P}^{\mathrm{C2F}}$ denotes the final volumetric segmentation, and $\odot$ means an element-wise replacement, and DeCrop operation defined on $\mathbf{P}^{\mathrm{F}}$, $\mathbf{P}^{\mathrm{C}}$, $\mathbf{X}^{\mathrm{F}}$ and $\mathbf{X}^{\mathrm{C}}$ is to replace a predefined rectangular cube inside $\mathbf{P}^{\mathrm{C}}$ by $\mathbf{P}^{\mathrm{F}}$, where the replacement locations are given by the definition of cropping $\mathbf{X}^{\mathrm{F}}$ from $\mathbf{X}^{\mathrm{C}}$ given in Eq. 4.4.

All in all, our entire 3D-based coarse-to-fine segmentation framework during testing is illustrated in Fig. 4.2.

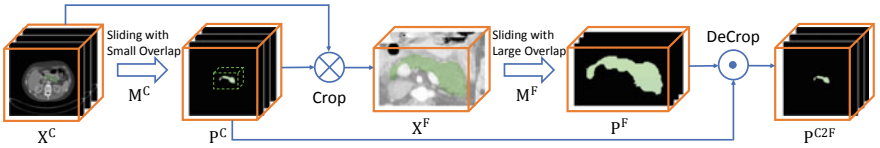

Fig. 4.2 Flowchart of the proposed 3D coarse-to-fine segmentation system in the testing phase. We first apply "ResDSN Coarse" with a small overlapped sliding window to obtain a rough pancreas region and then use the "ResDSN Fine" model to refine the results with a large overlapped sliding window. Best viewed in color

4.3.1.4 Network Architecture

As shown in Fig. 4.3, we provide an illustration of our convolutional network architecture. Inspired by V-Net [24], 3D U-Net [5], and VoxResNet [3], we have an encoder path followed by a decoder path each with four resolution steps. The left part of network acts as a feature extractor to learn higher and higher level of representations while the right part of network decompresses compact features into finer and finer resolution to predict the per-voxel segmentation. The padding and stride of each layer (Conv, Pooling, DeConv) are carefully designed to make sure the densely predicted output is the same size as the input.

The encoder subnetwork on the left is divided into different steps that work on different resolutions. Each step consists of one–two convolutions, where each convolution is composed of $3 \times 3 \times 3$ convolution followed by a batch normalization (BN [15]) and a rectified linear unit (ReLU [26]) to reach better convergence, and then a max-pooling layer with a kernel size of $2 \times 2 \times 2$ and strides of two to reduce

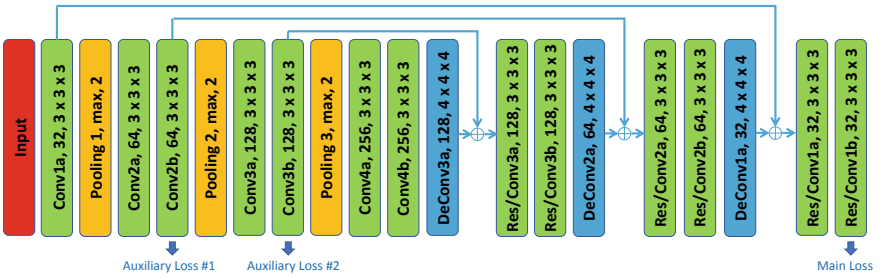

Fig. 4.3 Illustration of our 3D convolutional neural network for volumetric segmentation. The encoder path is the path between "Conv1a" and "Conv4b" while the decoder path is the one between "DeConv3a" and "Res/Conv1b". Each convolution or deconvolution layer consists of one convolution followed by a BatchNorm and a ReLU. To clarify, "Conv1a, 32, $3 \times 3 \times 3$" means the convolution operation with 32 channels and a kernel size of $3 \times 3 \times 3$. "Pooling 1, max, 2" means the max-pooling operation with kernel size of $2 \times 2 \times 2$ and a stride of two. Long residual connections are illustrated by the blue concrete lines. Blocks with the same color mean the same operations. Best viewed in color

resolutions and learn more compact features. The downsampling operation implemented by max pooling can reduce the size of the intermediate feature maps while increasing the size of the receptive fields. Having fewer size of activations makes it possible to double the number of channels during feature aggregation given the limited computational resource.

The decoder subnetwork on the right is composed of several steps that operate on different resolutions as well. Each step has two convolutions with each one followed by a BatchNorm and a ReLU, and afterward, a deconvolution with a kernel size of $4 \times 4 \times 4$ and strides of two are connected to expand the feature maps and finally predict the segmentation mask at the last layer. The upsampling operation that is carried out by deconvolution enlarges the resolution between each step, which increases the size of the intermediate activations so that we need to halve the number of channels due to the limited memory of the GPU card.

Apart from the left and right subnetworks, we impose a residual connection [13] to bridge shortcut connections of features between low-level layers and high-level layers. During the forward phase, the low-level cues extracted by networks are directly added to the high-level cues, which can help elaborate the fine-scaled segmentation, e.g., small parts close to the boundary which may be ignored during the feature aggregation due to the large size of receptive field at high-level layers. As for the backward phase, the supervision cues at high-level layers can be backpropagated through the shortcut way via the residual connections. This type of mechanism can prevent networks from gradient vanishing and exploding [9], which hampers network convergence during training.

We have one mainstream loss layer connected from "Res/Conv1b" and another two auxiliary loss layers connected from "Conv2b" and "Conv3b" to the ground truth label, respectively. For the mainstream loss in "Res/Conv1b" at the last layer which has the same size of data flow as one of the inputs, a $1 \times 1 \times 1$ convolution is followed to reduce the number of channels to the number of label classes which is 2 in our case. As for the two auxiliary loss layers, deconvolution layers are connected to upsample feature maps to be the same as the input.

The deep supervision imposed by auxiliary losses provides robustness to hyperparameters choice, in that the low-level layers are guided by the direct segmentation loss, leading to faster convergence rate. Throughout this work, we have two auxiliary branches where the default parameters are $\xi_1 = 0.2$ and $\xi_2 = 0.4$ in Eq. 4.3 to control the importance of deep supervisions compared with the major supervision from the mainstream loss for all segmentation networks.

As shown in Table 4.1, we give the detailed comparisons of network configurations in terms of four aspects: long residual connection, short residual connection, deep supervision, and loss function. Our backbone network architecture, named as "ResDSN", is proposed with different strategies in terms of combinations of long residual connection and short residual connection compared with VoxResNet [3], 3D HED [23], 3D DSN [6], and MixedResNet [44]. In this table, we also depict "FResDSN" and "SResDSN", where "FResDSN" and "SResDSN" are similar to MixedResNet [44] and VoxResNet [3], respectively. As confirmed by our quantitative experiments in Sect. 4.4.1.5, instead of adding short residual connections to

Table 4.1 Configurations comparison of different 3D segmentation networks on medical image analysis. For all the abbreviated phrases, "Long Res" means long residual connection, "Short Res" means short residual connection, "Deep Super" means deep supervision implemented by auxiliary loss layers, "Concat" means concatenation, "DSC" means Dice-Sørensen Coefficient and "CE" means cross-entropy. For residual connection, it has two types: concatenation ("Concat") or element-wise sum ("Sum")

Method	Long Res	Short Res	Deep Super	Loss
ResDSN (Ours)	Sum	No	Yes	CE
FResDSN	Sum	Sum	Yes	CE
SResDSN	No	Sum	Yes	CE
3D U-Net [5]	Concat	No	No	CE
V-Net [24]	Concat	Sum	No	DSC
VoxResNet [3]	No	Sum	Yes	CE
MixedResNet [44]	Sum	Sum	Yes	CE
3D DSN [6]	No	No	Yes	CE
3D HED [23]	Concat	No	Yes	CE

the network, e.g., "FResDSN" and "SResDSN", we only choose the long residual element-wise sum, which can be more computationally efficient while even performing better than the "FResDSN" architecture which is equipped with both long and short residual connections. Moreover, ResDSN has noticeable differences with respect to the V-Net [24] and 3D U-Net [5]. On the one hand, compared with 3D U-Net and V-Net which concatenate the lower level local features to higher level global features, we adopt the element-wise sum between these features, which outputs less number of channels for efficient computation. On the other hand, we introduce deep supervision via auxiliary losses into the network to yield better convergence.

4.3.2 3D Adversarial Examples

In this section, we discuss how to generate 3D adversarial examples for our segmentation framework as well as the defense method. We follow the notations defined in Sect. 4.3.1, i.e., $\mathbf{X}$ denotes a 3D CT scan volume, $\mathbf{Y}^{\text{true}}$ denotes the corresponding ground truth label, and $\mathcal{L}(\mathbf{X}; \boldsymbol{\Theta})$ denotes the network loss function. To generate the adversarial example, the goal is to maximize the loss $\mathcal{L}(\mathbf{X} + \mathbf{r}; \boldsymbol{\Theta})$ for the image $\mathbf{X}$, under the constraint that the generated adversarial example $\mathbf{X}^{\text{adv}} = \mathbf{X} + \mathbf{r}$ should look visually similar to the original image $\mathbf{X}$ and the corresponding predicted label $\mathbf{Y}^{\text{adv}} \neq \mathbf{Y}^{\text{true}}$. By imposing additional constraints such as $||\mathbf{r}||_\infty \leq \epsilon$, we can restrict the perturbation to be small enough to be imperceptible to humans.

4.3.2.1 Attack Methods

As for 3D adversarial attacking, we mainly adopt the gradient-based methods. They are as follows:

- **Fast Gradient Sign Method (FGSM)**: FGSM [10] is the first member in this attack family, which finds the adversarial perturbations in the direction of the loss gradient $\nabla_{\mathbf{X}}\mathcal{L}(\mathbf{X};\boldsymbol{\Theta})$. The update equation is

$$\mathbf{X}^{\text{adv}} = \mathbf{X} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{X}}\mathcal{L}(\mathbf{X};\boldsymbol{\Theta})). \tag{4.6}$$

- **Iterative Fast Gradient Sign Method (I-FGSM)**: An extended iterative version of FGSM [18], which can be expressed as

$$\begin{aligned} \mathbf{X}_0^{\text{adv}} &= \mathbf{X} \\ \mathbf{X}_{n+1}^{\text{adv}} &= \text{Clip}_{\mathbf{X}}^{\epsilon}\left\{\mathbf{X}_n^{\text{adv}} + \alpha \cdot \text{sign}(\nabla_{\mathbf{X}}\mathcal{L}(\mathbf{X}_n^{adv};\boldsymbol{\Theta}))\right\}, \end{aligned} \tag{4.7}$$

  where $\text{Clip}_{\mathbf{X}}^{\epsilon}$ indicates the resulting image are clipped within the ϵ-ball of the original image $\mathbf{X}$, n is the iteration number and α is the step size.

4.3.2.2 Defending Against 3D Adversarial Examples

Following [22], defending against adversarial examples can be expressed as a saddle point problem, which comprises of an inner maximization problem and an outer minimization problem. More precisely, our objective for defending against 3D adversarial examples is formulated as follows:

$$\min_{\Theta} \rho(\boldsymbol{\Theta}), \quad \text{where} \quad \rho(\boldsymbol{\Theta}) = \mathbb{E}_{(\mathbf{X})\sim\mathcal{D}}\left[\max_{\mathbf{r}\in\mathcal{S}} \mathcal{L}(\mathbf{X}+\mathbf{r};\boldsymbol{\Theta})\right]. \tag{4.8}$$

$\mathcal{S}$ and $\mathcal{D}$ denote the set of allowed perturbations and the data distribution, respectively.

4.4 Experiments

In this section, we demonstrate our experimental results, which consists of two parts. In the first part, we show the performance of our framework on pancreas segmentation. We first describe in detail how we conduct training and testing on the ***coarse*** and ***fine*** stages, respectively. Then we give the comparison results on three pancreas datasets: the NIH pancreas dataset [31], the JHMI pathological cyst dataset [45], and the JHMI pancreas dataset. In the second part, we discuss the adversarial attack and defense results on our framework.

4.4.1 Pancreas Segmentation

4.4.1.1 Network Training and Testing

All our experiments were run on a desktop equipped with the NVIDIA TITAN X (Pascal) GPU and deep neural networks were implemented based on the CAFFE [16] platform customized to support 3D operations for all necessary layers, e.g., "convolution", "deconvolution" and "pooling", etc. For the data preprocessing, we simply truncated the raw intensity values to be in $[-100, 240]$ and then normalized each raw CT case to have zero mean and unit variance to decrease the data variance caused by the physical processes [11] of medical images. As for the data augmentation in the training phase, unlike sophisticated processing used by others, e.g., elastic deformation [24, 29], we utilized simple but effective augmentations on all training patches, i.e., rotation (90°, 180°, and 270°) and flip in all three axes (axial, sagittal and coronal), to increase the number of 3D training samples which can alleviate the scarce of CT scans with expensive human annotations. Note that different CT cases have different physical resolutions, but we keep their resolutions unchanged. The input size of all our networks is denoted by $W_I \times H_I \times D_I$, where $W_I = H_I = D_I = 64$.

For the ***coarse*** stage, we randomly sampled $64 \times 64 \times 64$ sub-volumes from the whole CT scan in the training phase. In this case, a sub-volume can either cover a portion of pancreas voxels or be cropped from regions with non-pancreas voxels at all, which acts as a hard negative mining to reduce the false positive. In the testing phase, a sliding window was carried out to the whole CT volume with a ***coarse*** stepsize that has small overlaps within each neighboring sub-volume. Specifically, for a testing volume with a size of $W \times H \times D$, we have a total number of $(\lfloor \frac{W}{W_I} \rfloor + n) \times (\lfloor \frac{H}{H_I} \rfloor + n) \times (\lfloor \frac{D}{D_I} \rfloor + n)$ sub-volumes to be fed into the network and then combined to obtain the final prediction, where n is a parameter to control the sliding overlaps that a larger n results in a larger overlap and vice versa. In the ***coarse*** stage for the low time cost concern, we set $n = 6$ to efficiently locate the rough region of pancreas $\mathbf{X}^{\mathrm{F}}$ defined in Eq. 4.4 from the whole CT scan $\mathbf{X}^{\mathrm{C}}$.

For the ***fine*** stage, we randomly cropped $64 \times 64 \times 64$ sub-volumes constrained to be from the pancreas regions defined by ground truth labels during training. In this case, a training sub-volume was assured to cover pancreatic voxels, which was specifically designed to be capable of segmentation refinement. In the testing phase, we only applied the sliding window on $\mathbf{X}^{\mathrm{F}}$ with a size of $W_F \times H_F \times D_F$. The total number of sub-volumes to be tested is $(\lfloor \frac{W_F}{W_I} \rfloor + n) \times (\lfloor \frac{H_F}{H_I} \rfloor + n) \times (\lfloor \frac{D_F}{D_I} \rfloor + n)$. In the ***fine*** stage for the high accuracy performance concern, we set $n = 12$ to accurately estimate the pancreatic mask $\mathbf{P}^{\mathrm{F}}$ from the rough segmentation volume $\mathbf{X}^{\mathrm{F}}$. In the end, we mapped the $\mathbf{P}^{\mathrm{F}}$ back to $\mathbf{P}^{\mathrm{C}}$ to obtain $\mathbf{P}^{\mathrm{C2F}}$ for the final pancreas segmentation as given in Eq. 4.5, where the mapping location is given by the cropped location of $\mathbf{X}^{\mathrm{F}}$ from $\mathbf{X}^{\mathrm{C}}$.

After we get the final binary segmentation mask, we denote $\mathcal{P}$ and $\mathcal{Y}$ to be the set of pancreas voxels in the prediction and ground truth, separately, i.e., $\mathcal{P} = \{i \,|\, p_i = 1\}$ and $\mathcal{Y} = \{i \,|\, y_i = 1\}$. The evaluation metric is defined by the Dice-Sørensen Coef-

ficient (DSC) formulated as $\mathrm{DSC}(\mathcal{P}, \mathcal{Y}) = \frac{2\times|\mathcal{P}\cap\mathcal{Y}|}{|\mathcal{P}|+|\mathcal{Y}|}$. This evaluation measurement ranges in [0, 1] where 1 means a perfect prediction.

4.4.1.2 NIH Pancreas Dataset

We conduct experiments on the NIH pancreas segmentation dataset [31], which contains 82 contrast-enhanced abdominal CT volumes provided by an experienced radiologist. The size of CT volumes is $512 \times 512 \times D$, where $D \in [181, 466]$ and their spatial resolutions are $w \times h \times d$, where $d = 1.0\,\mathrm{mm}$ and $w = h$ that ranges from 0.5 to 1.0 mm. Data preprocessing and data augmentation were described in Sect. 4.4.1.1. Note that we did not normalize the spatial resolution into the same one since we wanted to impose the networks to learn to deal with the variations between different volumetric cases. Following the training protocol [31], we perform fourfold cross-validation in a random split from 82 patients for training and testing folds, where each testing fold has 21, 21, 20, and 20 cases, respectively. We trained networks illustrated in Fig. 4.3 by SGD optimizer with a 16 mini-batch, a 0.9 momentum, a base learning rate to be 0.01 via polynomial decay (the power is 0.9) in a total of 80,000 iterations, and the weight decay 0.0005. Both training networks in the ***coarse*** and ***fine*** stages shared the same training parameter settings except that they took a $64 \times 64 \times 64$ input sampled from different underlying distributions described in Sect. 4.4.1.1, which included the details of testing settings as well. We average the score map of overlapped regions from the sliding window and throw away small isolated predictions whose portions are smaller than 0.2 of the total prediction, which can remove small false positives. For DSC evaluation, we report the average with standard deviation, max and min statistics over all 82 testing cases as shown in Table 4.2.

First of all, our overall coarse-to-fine framework outperforms previous state of the art by nearly 2.2% (Cai et al. [2] and Zhou et al. [46]) in terms of average DSC, which is a large improvement. The lower standard deviation of DSC shows that our method

Table 4.2 Evaluation of different methods on the NIH dataset. Our proposed framework achieves state of the art by a large margin compared with previous state of the arts

Method	Mean DSC (%)	Max DSC (%)	Min DSC (%)
ResDSN C2F (Ours)	**84.59 ± 4.86**	**91.45**	**69.62**
ResDSN Coarse (Ours)	83.18 ± 6.02	91.33	58.44
Cai et al. [2]	82.4 ± 6.7	90.1	60.0
Zhou et al. [46]	82.37 ± 5.68	90.85	62.43
Dou et al. [6]	82.25 ± 5.91	90.32	62.53
Roth et al. [32]	78.01 ± 8.20	88.65	34.11
Yu et al. [43]	71.96 ± 15.34	89.27	0

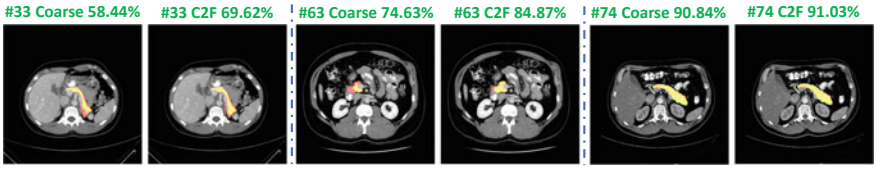

Fig. 4.4 Examples of segmentation results reported by "ResDSN Coarse" and "ResDSN C2F" on a same slice in the axial view from NIH case #33, #63 and #74, respectively. Numbers after "Coarse" or "C2F" mean testing DSC. Red, green, and yellow indicates the ground truth, prediction, and overlapped regions, respectively. Best viewed in color

is the most stable and robust across all different CT cases. Although the enhancement of max DSC of our framework is small due to saturation, the improvement of the min DSC over the second best (Dou et al. [6]) is from 62.53 to 69.62%, which is a more than 7% advancement. The worst case almost reaches 70%, which is a reasonable and acceptable segmentation result. After coarse-to-fine, the segmentation result of the worst case is improved by more than 11% after the 3D-based refinement from the 3D-based coarse result. The overall average DSC was also improved by 1.41%, which proves the effectiveness of our framework.[1]

As shown in Fig. 4.4, we report the segmentation results by "ResDSN Coarse" and "ResDSN C2F" on the same slice for comparison. Note that yellow regions are the correctly predicted pancreas. For the NIH case #33, which is the min DSC case reported by both "ResDSN Coarse" and "ResDSN C2F", the "ResDSN C2F" successfully predict more correct pancreas regions at the bottom, which is obviously missed by "ResDSN Coarse". If the coarse segmentation is bad, e.g., case #33 and #63, our 3D coarse-to-fine can significantly improve the segmentation results by as much as 10% in DSC. However, if the coarse segmentation is already very good, e.g., case #74, our proposed method cannot improve too much. We conclude that our proposed "ResDSN C2F" shows its advancement over 2D methods by aggregating rich spatial information and is more powerful than other 3D methods on the challenging pancreas segmentation task.

4.4.1.3 JHMI Pathological Cyst Dataset

We verified our proposed idea on the JHMI pathological cyst dataset [45] of abdominal CT scans as well. Different from the NIH pancreas dataset, which only contains healthy pancreas, this dataset includes pathological cysts where some can be or can become cancerous. The pancreatic cancer stage largely influences the morphology of the pancreas [19] that makes this dataset extremely challenging for considering the large variants.

[1]The results are reported by our runs using the same cross-validation splits where the code is available from their GitHub: https://github.com/yulequan/HeartSeg.

Table 4.3 Evaluations on the JHMI pathological pancreas

Method	Mean DSC (%)
ResDSN C2F (Ours)	**80.56 ± 13.36**
ResDSN Coarse (Ours)	77.96 ± 13.36
Zhou et al. [45]	79.23 ± 9.72

This dataset has a total number of 131 contrast-enhanced abdominal CT volumes with human-labeled pancreas annotations. The size of CT volumes is $512 \times 512 \times D$, where $D \in [358, 1121]$ that spans a wider variety of thickness than one of the NIH dataset. Following the training protocol [45], we conducted fourfold cross-validation on this dataset where each testing fold has 33, 33, 32, and 33 cases, respectively. We trained networks illustrated in Fig. 4.3 in both the ***coarse*** and ***fine*** stage with the same training settings as on the NIH except that we trained a total of 300,000 iterations on this pathological dataset since a pancreas with cysts is more difficult to segment than a normal case. In the testing phase, we vote the prediction map of overlapped regions from the sliding window and ignore small isolated pancreas predictions whose portions are smaller than 0.05 of the total prediction. As shown in Table 4.3, we compare our framework with only one available published results on this dataset. "ResDSN C2F" achieves an average 80.56% DSC that consistently outperforms the 2D based coarse-to-fine method [45], which confirms the advantage of leveraging the rich spatial information along three axes. What's more, the "ResDSN C2F" improves the "ResDSN Coarse" by 2.60% in terms of the mean DSC, which is a remarkable improvement that proves the effectiveness of the proposed 3D coarse-to-fine framework. Both [45] and our method have multiple failure cases whose testing DSC is 0, which indicates the segmentation of pathological organs is a more tough task. Due to these failure cases, we observe a large deviation on this pathological pancreas dataset compared with results on the NIH healthy pancreas dataset.

4.4.1.4 JHMI Pancreas Dataset

In order to further validate the superiority of our 3D model, We also evaluate our approach on a large high-quality dataset collected by the radiologists in our team. This dataset contains 305 contrast-enhanced abdominal CT volumes, and each of them is manually labeled with pancreas masks. Each CT volume consists of $319 \sim 1051$ slices of 512×512 pixels, and have voxel spatial resolution of $([0.523 \sim 0.977] \times [0.523 \sim 0.977] \times 0.5)\,\text{mm}^3$. Following the training protocol [31], we perform fourfold cross-validation in a random split from all patients for training and testing folds, where each testing fold has 77, 76, 76, and 76 cases, respectively. We demonstrate the superiority of our 3D model[2] by comparing with the 2D baseline [46] (see Table 4.4).

[2]The coarse model is used for comparison since it is the basis of our framework.

Table 4.4 Evaluations on the JHMI pancreas dataset

Method	Mean DSC (%)	Max DSC (%)	Min DSC (%)
ResDSN Coarse (Ours)	**87.84 ± 7.27**	**95.27**	0.07
Zhou et al. [45]	84.99 ± 7.42	93.45	**3.76**

4.4.1.5 Ablation Study

In this section, we conduct the ablation studies about residual connection, time efficiency and deep supervision to further investigate the effectiveness and efficiency of our proposed framework for pancreas segmentation.

Residual Connection

We discuss how different combinations of residual connections contribute to the pancreas segmentation task on the NIH dataset. All the residual connections are implemented in the element-wise sum and they shared exactly the same deep supervision connections, cross-validation splits, data input, training, and testing settings except that the residual structure is different from each other. As given in Table 4.5, we compare four configurations of residual connections of 3D-based networks only in the ***coarse*** stage. The major differences between our backbone network "ResDSN" with respect to "FResDSN", "SResDSN" and "DSN" are depicted in Table 4.1. "ResDSN" outperforms other network architectures in terms of average DSC and a small standard deviation even though the network is not as sophisticated as "FResDSN", which is the reason we adopt "ResDSN" for efficiency concerns in the ***coarse*** stage.

Time Efficiency

We discuss the time efficiency of the proposed coarse-to-fine framework with a smaller overlap in the ***coarse*** stage for the low consuming time concern while a larger one in the ***fine*** stage for the high prediction accuracy concern. The overlap size depends on how large we choose n defined in Sect. 4.4.1.1. We choose $n = 6$ during the coarse stage while $n = 12$ during the fine stage. Experimental results are shown in Table 4.6. "ResDSN Coarse" is the most efficient while the accuracy is the worst among three methods, which makes sense that we care more of the efficiency to obtain a rough pancreas segmentation. "ResDSN Fine" is to use a large overlap

Table 4.5 Evaluation of different residual connections on NIH

Method	Mean DSC (%)	Max DSC (%)	Min DSC (%)
ResDSN Coarse (Ours)	**83.18 ± 6.02**	91.33	58.44
FResDSN Coarse	83.11 ± 6.53	91.34	61.97
SResDSN Coarse	82.82 ± 5.97	90.33	62.43
DSN [6] Coarse	82.25 ± 5.91	90.32	62.53

Table 4.6 Average time cost in the testing phase, where n controls the overlap size of sliding windows during the inference

Method	Mean DSC (%)	n	Testing time (s)
ResDSN C2F (Ours)	**84.59 ± 4.86**	6 and 12	245
ResDSN coarse (Ours)	83.18 ± 6.02	6	**111**
ResDSN fine (Ours)	83.96 ± 5.65	12	382

Table 4.7 Ablation study of the deep supervision on NIH

Method	Mean DSC (%)	Max DSC (%)	Min DSC (%)
ResDSN C2F (Ours)	**84.59 ± 4.86**	91.45	**69.62**
Res C2F	84.06 ± 6.51	**91.72**	51.83

on an entire CT scan to do the segmentation which is the most time consuming. In our coarse-to-fine framework, we combine the two advantages together to propose "ResDSN C2F" which can achieve the best segmentation results while the average testing time cost for each case is reduced by 36% from 382 to 245s compared with "ResDSN Fine". In comparison, it takes an experienced board-certified Abdominal Radiologist 20 min for one case, which verifies the clinical use of our framework.

Deep Supervision

We discuss how effective of the auxiliary losses to demonstrate the impact of the deep supervision on our 3D coarse-to-fine framework. Basically, we train our mainstream networks without any auxiliary losses for both coarse and fine stages, denoted as "Res C2F", while keeping all other settings as the same, e.g., cross-validation splits, data preprocessing and post-processing. As shown in Table 4.7, "ResDSN C2F" outperforms "Res C2F" by 17.79% to a large extent on min DSC and 0.53% better on average DSC though it's a little bit worse on max DSC. We conclude that 3D coarse-to-fine with deep supervisions perform better and especially more stable on the pancreas segmentation.

4.4.2 Adversarial Attack and Defense

In spite of the success of 3D learning models such as our proposed ResDSN, the robustness of neural networks for clinical applications remains a concern. In this section, we first show that our well-trained 3D model can be easily led to failure under imperceptible adversarial perturbations (see Sect. 4.4.2.1), and then investigate how to improve the adversarial robustness by employing adversarial training (see Sect. 4.4.2.2). We evaluate our approach by performing standard fourfold cross-validation on the JHMI pancreas dataset since this dataset is the largest in scale and has the best quality (see Sect. 4.4.1.4).

4.4.2.1 Robustness Evaluation

To evaluate the robustness of our well-trained 3D model, we attack the ResDSN Coarse model following the methods in Sect. 4.3.2.1. For both attacking methods, i.e., FGSM and I-FGSM, we set $\epsilon = 0.03\Lambda$ so that the maximum perturbation can be small enough compared with the range of the truncated intensity value (Λ).[3] Specially in the case of I-FGSM, the total iteration number N and the step size α are set to be 5 and 0.01Λ, respectively. Following the test strategy in the coarse stage, we first compute the loss gradients of the $64 \times 64 \times 64$ sub-volumes[4] obtained by a sliding-window policy, and these gradients are then combined to calculate the final loss gradient map $\nabla_{\mathbf{X}}\mathcal{L}(\mathbf{X}; \boldsymbol{\Theta})$ of each whole CT volume. The combined approach is also similar as the testing method described in Sect. 4.4.1.3, i.e., taking the average of loss gradient if a voxel is in the overlapped region. According to Eqs. 4.6 and 4.7, the overall loss gradient can be used to generate adversarial examples which can then attack the 3D model for the purpose of robustness evaluation.

4.4.2.2 Defending Against Adversarial Attacks

To improve the adversarial robustness of our 3D segmentation model, we apply the adversarial training policy as described in Sect. 4.3.2.2. During each training iteration, $\mathbf{X_{adv}}$ is first randomly sampled in the ϵ-ball and then updated by I-FGSM so that $\mathcal{L}(\mathbf{X_{adv}}; \boldsymbol{\Theta})$ can be maximized. Afterward $\mathbf{X_{adv}}$ is fed to the model instead of $\mathbf{X}$ to update the parameter Θ. Note that we set the same maximum perturbation ϵ, iteration number N and step size α as in Sect. 4.4.2.1. Similar to the training process described in Sect. 4.4.1.2, our model is trained by SGD optimizer with a 128 mini-batch, a 0.9 momentum, a base learning rate to be 0.08 via polynomial decay (the power is 0.9) in a total of 10, 000 iterations, and the weight decay 0.0005.

4.4.2.3 Results and Discussion

All attack and defense results are summarized in Table 4.8. We can see that both attack methods, i.e., FGSM and I-FGSM, can successfully fool the well-trained 3D ResDSN into producing incorrect prediction maps. More specifically, the dramatic performance drop of I-FGSM, i.e., 85.83% (from 87.84 to 2.01%), suggests low adversarial robustness of the original model. Meanwhile, the maximum performance drop decreases from 85.83 to 13.11%, indicating that our adversarially trained model can largely alleviate the adversarial effect and hence improving the robustness of our 3D model. Note that our baseline with "Clean" training has 87.84% accuracy when

[3] Since the raw intensity values are to be in $[-100, 240]$ during preprocessing (see Sect. 4.4.1.1), here we set $\Lambda = 240 - (-100) = 340$ accordingly.

[4] For implementation simplicity and efficiency, we ignored the sub-volumes only containing the background class when generating adversarial examples.

Table 4.8 Comparative evaluation of the 3D segmentation model on clean (indicated by "Clean" in the table) and adversarial examples. Different attack methods, i.e., FGSM and I-FGSM, are used for generating the adversarial examples. We report the average accuracy and Dice overlap score along with the % maximum drop in performance on adversarial examples with respect to performance on clean data

Attack methods	Clean (%)	FGSM [10] (%)	I-FGSM [18] (%)	Drop (%)
ResDSN coarse	**87.84 ± 7.27**	42.68	2.01	85.83
Adversarially trained ResDSN coarse	79.09 ± 12.10	**67.58**	**65.98**	**13.11**

tested on clean images, whereas its counterpart with adversarial training obtains 79.09%. This trade-off between adversarial and clean training has been previously observed in [38]. We hope this trade-off can be better studied in future research.

We also show a qualitative example in Fig. 4.5. As can be observed from the illustration, adversarial attacks to naturally trained 3D ResDSN induces many false positives, which makes the corresponding outcomes noisy. On the contrary, the adversarially trained 3D model yields similar performances even after applying I-FGSM. More specifically, the original average Dice score of 3D ResDSN is 89.30%, and after applying adversarial attack the performance drops to 48.45 and 6.06% with FGSM

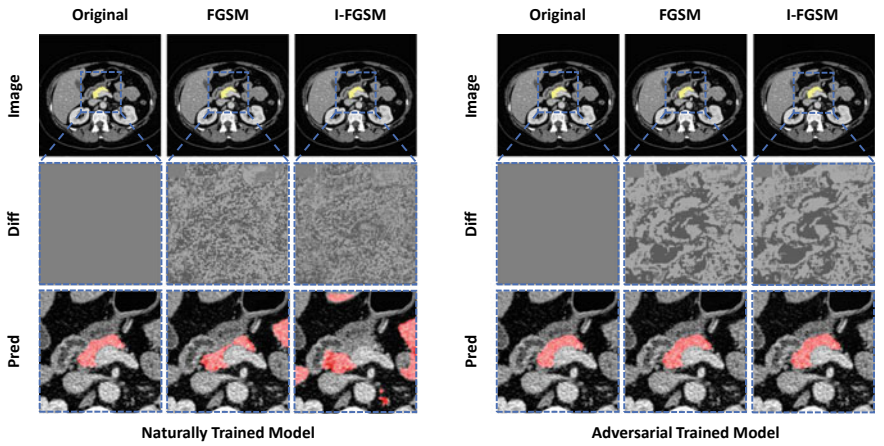

Fig. 4.5 Qualitative comparison of adversarial examples and their effects on model predictions. Note that the added perturbation is effectively imperceptible to the human eye, and the difference between the original image and the adversarial image has been magnified by 5 × (values shifted by 128) for a better visualization. Contrasting with prediction on original images, the crafted examples are able to successfully fool the models into generating incorrect segmentation maps. Meanwhile, adversarial training can effectively alleviate such negative influence of adversarial attacks, hence improving the performance to a reasonable level. Image differences and predictions are zoomed in from the axial view to better visualize the finer details

and I-FGSM, respectively. However, when applying the same attack methods to the adversarially trained model, the performance only drops from 86.41 to 80.32 and 79.56%, respectively. In other words, employing adversarial training decreases the performance drop from 83.24 to only 6.85%. This promising result clearly indicates that our adversarially trained model can largely improve the adversarial robustness.

4.5 Conclusion

In this chapter, we proposed a novel 3D network called "ResDSN" integrated with a coarse-to-fine framework to simultaneously achieve high segmentation accuracy and low time cost. The backbone network "ResDSN" is carefully designed to only have long residual connections for efficient inference. In addition, we also analyzed the threat of adversarial attacks on our framework and showed how to improve the robustness against the attack. Experimental evidence indicates that our adversarially trained model can largely improve adversarial robustness than naturally trained ones.

To our best knowledge, the proposed 3D coarse-to-fine framework is one of the first works to segment the challenging pancreas using 3D networks which leverage the rich spatial information to achieve the state of the art. We can naturally apply the proposed idea to other small organs, e.g., spleen, duodenum and gallbladder, etc, In the future, we will target on error causes that lead to inaccurate segmentation to make our framework more stable, and extend our 3D coarse-to-fine framework to cyst segmentation which can cause cancerous tumors, and the very important tumor segmentation [49] task.

References

1. Bui TD, Shin J, Moon T (2017) 3D densely convolution networks for volumetric segmentation. arXiv:1709.03199
2. 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
3. Chen H, Dou Q, Yu L, Qin J, Heng PA (2017) Voxresnet: deep voxelwise residual networks for brain segmentation from 3D MR images. NeuroImage
4. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2016) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv:1606.00915
5. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O (2016) 3D u-net: learning dense volumetric segmentation from sparse annotation. In: MICCAI
6. Dou Q, Yu L, Chen H, Jin Y, Yang X, Qin J, Heng PA (2017) 3D deeply supervised network for automated segmentation of volumetric medical images. MIA 41:40–54
7. Fang Y, Xie J, Dai G, Wang M, Zhu F, Xu T, Wong E (2015) 3D deep shape descriptor. In: CVPR
8. Gibson E, Giganti F, Hu Y, Bonmati E, Bandula S, Gurusamy K, Davidson B, Pereira SP, Clarkson MJ, Barratt DC (2018) Automatic multi-organ segmentation on abdominal ct with dense v-networks. TMI

9. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: AISTATS
10. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: ICLR
11. Gravel P, Beaudoin G, De Guise JA (2004) A method for modeling noise in medical images. TMI 23(10):1221–1232
12. Havaei M, Davy A, Warde-Farley D, Biard A, Courville AC, Bengio Y, Pal C, Jodoin P, Larochelle H (2017) Brain tumor segmentation with deep neural networks. MIA 35:18–31
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR
14. Huang Y, Würfl T, Breininger K, Liu L, Lauritsch G, Maier A (2018) Some investigations on robustness of deep learning in limited angle tomography. In: MICCAI
15. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML
16. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) CAFFE: convolutional architecture for fast feature embedding. MM
17. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: NIPS
18. Kurakin A, Goodfellow I, Bengio S (2017) Adversarial machine learning at scale. In: ICLR
19. Lasboo AA, Rezai P, Yaghmai V (2010) Morphological analysis of pancreatic cystic masses. Acad Radiol 17(3):348–351
20. Lee CY, Xie S, Gallagher P, Zhang Z, Tu Z (2015) Deeply-supervised nets. In: AISTATS
21. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR
22. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu, A (2018) Towards deep learning models resistant to adversarial attacks. In: ICLR
23. Merkow J, Marsden A, Kriegman D, Tu Z (2016) Dense volume-to-volume vascular boundary detection. In: MICCAI
24. Milletari F, Navab N, Ahmadi SA (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV
25. Moeskops P, Wolterink JM, van der Velden BHM, Gilhuijs KGA, Leiner T, Viergever MA, Isgum I (2017) Deep learning for multi-task medical image segmentation in multiple modalities. CoRR arXiv:1704.03379
26. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: ICML
27. Paschali M, Conjeti S, Navarro F, Navab N (2018) Generalizability versus robustness: adversarial examples for medical imaging. In: MICCAI
28. 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
29. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: MICCAI
30. Roth H, Oda M, Shimizu N, Oda H, Hayashi Y, Kitasaka T, Fujiwara M, Misawa K, Mori K (2018) Towards dense volumetric pancreas segmentation in CT using 3D fully convolutional networks. In: SPIE
31. Roth HR, Lu L, Farag A, Shin HC, Liu J, Turkbey EB, Summers RM (2015) Deeporgan: multi-level deep convolutional networks for automated pancreas segmentation. In: MICCAI
32. Roth HR, Lu L, Farag A, Sohn A, Summers RM (2016) Spatial aggregation of holistically-nested networks for automated pancreas segmentation. In: MICCAI
33. Shen W, Wang B, Jiang Y, Wang Y, Yuille AL (2017) Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection. In: ICCV, pp 2410–2419
34. Shen W, Wang X, Wang Y, Bai X, Zhang Z (2015) Deepcontour: a deep convolutional feature learned by positive-sharing loss for contour detection. In: CVPR
35. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556

36. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: CVPR
37. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: ICLR
38. Tsipras D, Santurkar S, Engstrom L, Turner A, Madry A (2018) Robustness may be at odds with accuracy, p 1. arXiv:1805.12152
39. Wang Y, Zhou Y, Shen W, Park S, Fishman EK, Yuille AL (2018) Abdominal multi-organ segmentation with organ-attention networks and statistical fusion. CoRR. arXiv:1804.08414
40. 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: Proceedings of MICCAI, pp. 434–442
41. Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A (2017) Adversarial examples for semantic segmentation and object detection. In: ICCV
42. Xie S, Tu Z (2015) Holistically-nested edge detection. In: ICCV
43. Yu L, Cheng JZ, Dou Q, Yang X, Chen H, Qin J, Heng PA (2017) Automatic 3D cardiovascular MR segmentation with densely-connected volumetric convnets. In: MICCAI
44. 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
45. Zhou Y, Xie L, Fishman EK, Yuille AL (2017) Deep supervision for pancreatic cyst segmentation in abdominal CT scans. In: MICCAI
46. Zhou Y, Xie L, Shen W, Wang Y, Fishman EK, Yuille AL (2017) A fixed-point model for pancreas segmentation in abdominal CT scans. In: MICCAI
47. Zhu Z, Wang X, Bai S, Yao C, Bai X (2016) Deep learning representation using autoencoder for 3D shape retrieval. Neurocomputing 204:41–50
48. Zhu Z, Xia Y, Shen W, Fishman EK, Yuille AL (2018) A 3d coarse-to-fine framework for volumetric medical image segmentation. In: International conference on 3D vision, pp 682–690
49. Zhu Z, Xia Y, Xie L, Fishman EK, Yuille AL (2018) Multi-scale coarse-to-fine segmentation for screening pancreatic ductal adenocarcinoma. arXiv:1807.02941