Advances in Computer Vision and Pattern Recognition

Le Lu
Xiaosong Wang
Gustavo Carneiro
Lin Yang *Editors*

Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics

Springer

Advances in Computer Vision and Pattern Recognition

More information about this series at http://www.springer.com/series/4205

Le Lu · Xiaosong Wang ·
Gustavo Carneiro · Lin Yang
Editors

Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics

Springer

Editors
Le Lu
Bethesda Research Lab
PAII Inc.
Bethesda, MD, USA

Department of Computer Science
Johns Hopkins University
Baltimore, MD, USA

Gustavo Carneiro
School of Computer Science
University of Adelaide
Adelaide, SA, Australia

Xiaosong Wang
Nvidia Corporation
Bethesda, MD, USA

Lin Yang
Department of Biomedical Engineering
University of Florida
Gainesville, FL, USA

ISSN 2191-6586 ISSN 2191-6594 (electronic)
Advances in Computer Vision and Pattern Recognition
ISBN 978-3-030-13968-1 ISBN 978-3-030-13969-8 (eBook)
https://doi.org/10.1007/978-3-030-13969-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book is the second edition of a series documenting how deep learning and deep neural networks are being successfully employed within medical image computing. Looking back to the first edition, published 2 years ago, one can observe how much this field has grown. As observed later, to better represent the state-of-the-art work, *which surrounds the three core research problems of (1) medical image semantic segmentation; (2) anatomical or pathological structure detection and localization; (3) large-scale clinical imaging semantics (often using embedded natural language text and ontology knowledge) or deep image–text data mining in the "big data, weak label" theme*; tremendous amounts of intriguing and impactful methodological and application developments are needed to be timely reflected.

The first edition of this series focused more on demonstrating the validity of using deep learning, especially deep convolutional neural networks, on a range of key important problems within medical imaging computing. These included anatomical organ segmentation and high-performance computer-aided disease detection and diagnosis. The technical contributions in this edition clearly demonstrate more capable methodological developments, such as novel deep learning algorithms beyond convolutional neural networks, evaluations using much larger clinical imaging datasets, and demonstrations of more robust and accurate empirical performance. These all contribute toward more real and practical solutions for radiology and pathology applications. In particular, many works focus on exploiting truly large-scale and messy clinical databases, which can allow for more generalizable deep learning solutions, but at the cost of much greater technical challenges. These include frequent label noise, annotation inconsistency, and weak supervisory signals, e.g., unstructured radiologist text reports.

Going forward, we do expect the development of deep learning methods to continue steadily along with a broader adoption of these solutions into clinical practice. For example, the flagship conferences in this domain, e.g., IEEE Computer Vision and Pattern Recognition and International Conference on Medical Image Computing and Computer-Assisted Intervention, have grown rapidly by roughly 30% each year. Although there are surely many more technical and clinical challenges lying ahead, it has never been more exciting to contribute to these research efforts as now.

Organization and Features

This book covers a range of topics in medical imaging from traditional tasks, e.g., segmentation, detection, and localization, to large-scale radiological data mining for deep learning purposes. In the following, we give a brief overview of the contents of this book.

Part I discusses several recent works on segmentation tasks. Chapter 1 presents a recurrent neural network to address the problem of spatial discontinuity in segmenting the pancreas across adjacent image slices. The network takes outputs of the convolutional neural network and refines the segmentation by improving the shape smoothness. Chapter 2 discusses a recurrent neural network-based method to accurately segment the perimysium in hematoxylin- and eosin-stained whole-slide specimens of muscle biopsies. This will help provide early diagnosis of many muscular inflammation diseases. Chapter 3 introduces two coarse-to-fine mechanisms for segmenting small organs, e.g., the pancreas, in which predictions from the first (coarse) stage are used as the initial region for the second (fine) stage. Chapter 4 further extends the coarse-to-fine framework of pancreas segmentation using volumetric image data. It also presents an analysis of the threat of adversarial attacks on the proposed framework and shows how to defend against the attack. Chapter 5 presents an unsupervised domain adaptation method using adversarial learning in medical image segmentation tasks from two different perspectives: feature- and pixel-level adaptation. This may help train more generalizable deep learning solutions.

Part II exhibits some state-of-the-art solutions to disease/lesion detection and localization problems. Chapter 6 introduces two state-of-the-art glaucoma detection methods based on two different network architectures by either utilizing the segmentation information or generating the segmentation of the optic disk and cup. Chapter 7 presents a unified network for simultaneous disease recognition and localization in chest X-ray images. Particularly, patches of images are extracted in a multiple instance learning framework for more accurate disease localization. Chapter 8 introduces a searching strategy that gradually focuses on lesions by progressively transforming a bounding volume until a breast lesion is detected, which not only accelerates the inference time but also increases the detection accuracy. Chapter 9 discusses the influence of abnormal conditions in the images to accurately identify vertebrae locations. An automatic vertebra labeling approach is presented using a deep image-to-image network with message passing and sparsity regularization techniques. Chapter 10 discusses a solution for learning 3D convolution kernels by transferring convolutional features learned from 2D images to 3D anisotropic volumes. This is applied to two medical image analysis applications: lesion detection from a digital breast tomosynthesis volume and liver and liver tumor segmentation from a computed tomography volume.

Part III explores a variety of problems and applications in medical imaging, e.g., image quality control and restoration, image retrieval, image registration, and tumor growth prediction. Chapter 11 presents an introduction of automated histopathology image analysis and then discusses several recent deep hashing techniques and their applications on content-based histopathology image retrieval. Chapter 12 introduces a two-stream deep convolutional network method for prognostic tumor growth modeling via volumetric medical imaging observations. Experiments demonstrate that it outperforms the state-of-the-art mathematical model-based approach in both accuracy and efficiency. Chapter 13 proposes a deep spatiotemporal convolutional neural network to restore low-quality CT perfusion images that are acquired at reduced tube current, dosage, and spatial resolution. Chapter 14 introduces a new CT image denoising method based on generative adversarial networks with Wasserstein distance and perceptual similarity for low-dose CT data. The proposed method is capable of not only reducing the image noise level but also retaining the critical information at the same time. Chapter 15 discusses an automatic framework to check the quality of cardiac magnetic resonance (CMR) images (the coverage of left ventricle from CMR images) by using Fisher-discriminative and dataset-invariant 3D convolutional neural networks. Image-acquisition parameters are not considered in the model such as imaging device, magnetic field strength, variations in protocol execution. Chapter 16 introduces a learning-based image registration method that employs deep neural networks to estimate plausible registrations. Particularly, agent-based methods for medical image registration are reviewed together with two applications on rigid-body 3D/3D and 2D/3D registrations. Chapter 17 explores and compares the use of deep learning methods in comparison with conventional machine learning classifiers as well as their ensembles to analyze fMRI scans.

Part IV covers hot topics in large-scale data mining and data synthesis in clinical settings. Chapter 18 introduces a feasible solution to mine existing imaging informatics in a hospital environment using both natural language processing and image processing techniques so as to build large-scale datasets to facilitate data-hungry deep learning paradigms. Chapter 19 further discusses how to mine textual data (radiological reports) using deep learning-based natural language processing techniques and exploits a image–text embedding network for automatic free-text report generation. Chapter 20 outlines a system to extract the clinical annotations collected in radiologists' daily work routine and convert them into machine learnable formats. Sample applications of such extracted data are discussed, e.g, lesion retrieval, clustering, and classification. Chapter 21 explores the problem of synthesizing high-resolution images corresponding to one MRI modality from a low-resolution image of another MRI modality of the same subject. A cross-modality dictionary learning scheme is discussed together with a patch-based globally redundant model based on sparse representations.

Bethesda, USA Le Lu
Bethesda, USA Xiaosong Wang
Adelaide, Australia Gustavo Carneiro
Gainesville, USA Lin Yang

Contents

Part III Various Applications

Part I
Segmentation

Chapter 1
Pancreas Segmentation in CT and MRI via Task-Specific Network Design and Recurrent Neural Contextual Learning

Jinzheng Cai, Le Lu, Fuyong Xing and Lin Yang

Abstract Automatic pancreas segmentation in radiology images, e.g., computed tomography (CT), and magnetic resonance imaging (MRI), is frequently required by computer-aided screening, diagnosis, and quantitative assessment. Yet, pancreas is a challenging abdominal organ to segment due to the high inter-patient anatomical variability in both shape and volume metrics. Recently, convolutional neural networks (CNN) have demonstrated promising performance on accurate segmentation of pancreas. However, the CNN-based method often suffers from segmentation discontinuity for reasons such as noisy image quality and blurry pancreatic boundary. In this chapter, we first discuss the CNN configurations and training objectives that lead to the state-of-the-art performance on pancreas segmentation. We then present a recurrent neural network (RNN) to address the problem of segmentation spatial inconsistency across adjacent image slices. The RNN takes outputs of the CNN and refines the segmentation by improving the shape smoothness.

J. Cai · L. Yang (✉)
J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida,
Gainesville, FL, USA
e-mail: lin.yang@bme.ufl.edu

J. Cai
e-mail: jimmycai@ufl.edu

L. Lu
PAII Inc., Bethesda Research Lab, 6720B Rockledge Drive Ste 410, Bethesda, MD 20817, USA
e-mail: le.lu@paii-labs.com; lelu@cs.jhu.edu

Johns Hopkins University, Baltimore, MD, USA

F. Xing
Department of Biostatistics and Informatics, University of Colorado Anschutz Medical Campus,
Aurora, CO, USA
e-mail: fuyong.xing@ucdenver.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_1

1.1 Introduction

Detecting unusual volume changes and monitoring abnormal growths in pancreas using medical images is a critical yet challenging task for computer-aided diagnosis (CAD). This would require to delineate pancreas from its surrounding tissues in radiology images, e.g., computed tomography (CT), and magnetic resonance imaging (MRI) scans. The accurate segmentation of pancreas delivers more reliable and quantitative representations than the cross section diameter measurement and it may facilitate the producing of segmentation-based biomarkers, such as volumetric measurements and 3D shape/surface signatures. Moreover, automated rapid and accurate segmentation of pancreas on the scale of processing thousands of image scans may facilitate new protocols, findings, and insights for clinical trials. On the other hand, manual pancreas segmentation is very expensive and sometimes even intractable on the dataset at a very large scale. To fulfill this practical and important demand, many efforts have been investigated to significantly boost the segmentation performance in both CT and MRI modalities.

One major group on the automatic pancreas segmentation in CT images is based on top-down multi-atlas registration and label fusion (MALF) [11, 17, 26, 27]. Due to challenges from the high deformable shape and vague boundaries of the pancreas in CT scans from various patients, the reported segmentation accuracy (measured in Dice Similarity Coefficient or DSC) is limited in the range from 69.6 ± 16.7% [27] to 78.5 ± 14.0% [11, 17] under leave-one-patient-out (LOO) evaluation protocol. On the other hand, bottom-up deep CNN-based pancreas segmentation work [2, 8, 19–21, 30] have revealed promising results and steady performance improvements, e.g. from 71.8 ± 10.7% [19], 78.0 ± 8.2% [20], to 81.3 ± 6.3% [21] evaluated using the same NIH 82-patient CT dataset [6, 19] under fourfold cross-validation (CV).

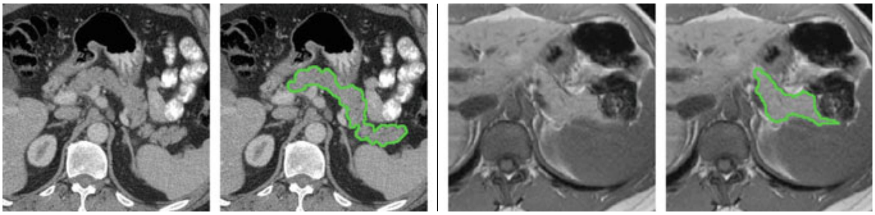
Deep learning-based approaches appear to demonstrate noticeably higher segmentation accuracies and numerically more stable results (significantly lower in standard deviation, or std). References [20, 21] are built upon the fully convolutional neural network (FCN) architecture [13] and its variant [28]. However, [20, 21] both rely on post-processing with random forest to further refine CNN's outputs, which cannot propagate errors back to the CNN model. Similarly, for pancreas segmentation on a 79-patient MRI dataset, [2] achieves 76.1 ± 8.7% in DSC, where graph-based result fusion is applied. Therefore, an end-to-end trainable deep learning model for pancreas segmentation may be more desirable to achieve superior results. Additionally, deep CNN-based bottom-up pancreas segmentation methods also have significant advantages on run-time computational efficiency, such as 2$\sim$4 h [11] versus 2$\sim$3 m [21] to process a new segmentation case.

1.2 Convolutional Neural Network for Pancreas Segmentation

With the goal of obtaining accurate segmentation for objects with complex shape outlines, many deep learning approaches have been proposed [4, 13, 18, 28, 29] to report good performances. Specifically, some of these models regularize deep learning output with appearance constraint that image pixels sharing similar color and location statistics would probably come from the same object category and this leads to conditional random field (CRF) post-processing that presented in [4, 29]. On the other hand, some methods propose to learn localized deep learning features. For instance, deep supervision is proposed in [12, 28] forcing all convolutional layers to learn effective and consistent low-level representations, e.g., edge and object boundary. Meanwhile, U-Net architecture [18] makes full use of low-level convolutional feature maps by projecting them back to the original image size. The dedicated backward propagation combines convolutional layer features of multiple scales, thereby boosting the accuracy of object segmentation.

1.2.1 Design of Network Architecture

Delineating the pancreas boundary from its surrounding anatomies in CT/MRI scans is challenging due to its complex visual appearance and ambiguous outlines. For example, Fig. 1.1 displays an example of the pancreas in CT and MRI images, where the pancreas shares similar intensity values with other soft tissues and its boundary is blurry where touching with other abdominal organs. These natures of pancreas segmentation inspire us to combine virtues from both of the holistically nested network (HNN) [28] and U-Net [18] and we name the combination P-Net as it is designed task specifically for pancreas segmentation. In Fig. 1.2, we visually depict the network architecture of the proposed P-Net and the most correlated networks, i.e. HNN and U-Net. The P-Net inherits the deep supervisions from HNN and the skip connections from U-Net for feature multi-scale aggregation.



(a) CT image (b) ground truth (c) MRI image (d) ground truth

Fig. 1.1 Examples of pancreas images in CT and MRI. The ground truth pancreas boundaries are presented in (**b**) and (**d**) delineated in green. Best viewed in color

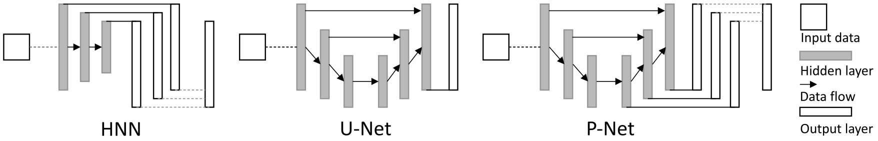


Fig. 1.2 Network architecture of several popular CNN models for pancreas segmentation

1.2.2 Design of Model Training Strategy

Apart from the network architecture, the proposed task-specific design also includes a strategy to train the P-Net from scratch. This is because the CT and MRI modalities demonstrate very different image statistics from natural images. Thus, a direct transfer of ImageNet [7] pretrained neural network to the medical domain could be suboptimal. Similarly, to transfer the model from other medical tasks can also be problematic for P-Net to achieve the optimal pancreas segmentation. During model training, we also observe gradient-vanishing problem occurs when fine-tune the network from pretrained models. Specifically, the top layers of the network will quickly capture the hierarchical (conceptual) appearance of pancreas but leaving lower layers not well tuned as the magnitudes of gradients (backpropagated from the training error) fastly decreased. To circumvent this problem, we propose to initialize P-Net from scratch and train the network layer by layer.

To present the training algorithm of P-Net in formal, we denote the number of steps for layer-by-layer training to be K. Then, the corresponding convolutional and deconvolutional layers can be represented as $\{C_1, \ldots, C_K\}$ and $\{D_1, \ldots, D_K\}$. We also denote standard network operations as up-sampling to be $Up(\#)$, concatenation to be $Concat([\cdot; \cdot])$, and dimension reduction to be $R(\#)$. For representation clarity, we use $\circ$ to denote the composition of two transformations and use $\prod$ for multiple transformations. We drop pooling operations between convolutional layers for simplicity.

First, we define the forward propagation as a combination of convolutional layers

$$F_k = \prod_{i=1}^{K+1-k} C_i. \tag{1.1}$$

Then in P-Net, the feature map is up-scaled step-by-step util it restores the size of the original input image. Specifically, when k equals 1, we have

$$B_k = F_k \circ D_k, \tag{1.2}$$

otherwise, the feature map process can be represented as

$$B_k = Concat([B_{k-1} \circ Up(2);\ F_k]) \ \circ\ D_k. \tag{1.3}$$

At each scale (or step), the segmentation output is

$$\hat{Y}_k = B_k \circ Up\ (2^{K-k}) \circ R(1) \circ Sigmoid, \tag{1.4}$$

where the feature map B_k is first up-scaled by a factor of 2^{K-k} such that restores the size of the original input image. Its channel is then be reduced to 1 via $R(1)$, and then it passes through a sigmoid activation function *Sigmoid* to produce the segmentation output at scale (or step) k as $\hat{Y}_k$.

The output $\hat{Y}_k$ is a probability map on which the segmentation loss is measured as

$$\mathcal{L}_k = H(\hat{Y}_k, Y), \tag{1.5}$$

where Y is the ground truth label map and $H(\cdot, \cdot)$ is the loss function, e.g. cross-entropy loss. Thus, each unit module B_k has its own gradient back propagation path that starts at the loss $\mathcal{L}_k$ and ends at the input image. It introduces deep supervision to the bottom CNN layers and enables us to train P-Net from swallow to deep. More specifically, the training of P-Net starts at $k = 1$ and increase k by 1 when $\mathcal{L}_k$-plot converges. The final segmentation result $\hat{Y}$ is a weighted combination of the side outputs as

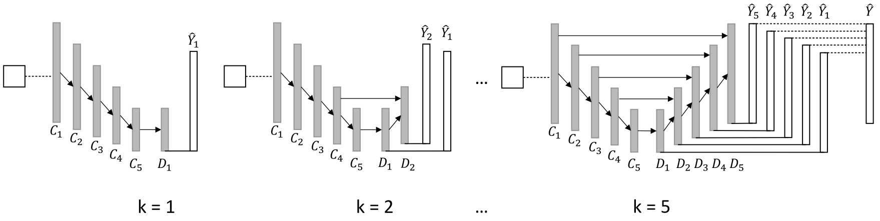
$$\hat{Y} = \sum_{i=1}^{K} \omega_i \hat{Y}_i, \tag{1.6}$$

and the overall objective for P-Net is,

$$\mathcal{L} = H(\hat{Y}, Y) + \sum_{i=1}^{K} \mathcal{L}_i, \tag{1.7}$$

where $K = 5$ delivers the best for pancreas segmentation. We summarize the training procedure in Algorithm 1 and visually depict the semantic illustration of P-Net structures at $k = 1$, $k = 2$, and $k = 5$, respectively, in Fig. 1.3.

Discussion: Although the network architecture of P-Net can be extended to process 3D inputs [5], we maintain the current 2D architecture in model training and inference because the 3D version can be computationally expensive while gaining



$\hat{Y}_1$
C_1 C_2 C_3 C_4 C_5 D_1
k = 1
$\hat{Y}_2$ $\hat{Y}_1$
C_1 C_2 C_3 C_4 C_5 D_1 D_2
k = 2
...
$\hat{Y}_5$ $\hat{Y}_4$ $\hat{Y}_3$ $\hat{Y}_2$ $\hat{Y}_1$ $\hat{Y}$
C_1 C_2 C_3 C_4 C_5 D_1 D_2 D_3 D_4 D_5
k = 5

Fig. 1.3 Semantic illustration of the P-Net training algorithm

no significant improvement in performance [30]. As a compromise, P-Net takes 3-connected slices as its input when given the segmentation ground truth mask of the middle slice. As explained in Sect. 1.3, P-Net is transformed into a lightweighted 3D model with RNN stacked to the end of it, which allows our model to capture 3D imaging information with minor extra computational loads. This is in a similar spirit to employ RNN to regulate, process and aggregate CNN activations for video classification and understanding [16].

```
Result: $\hat{Y}, \hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_K$
$K$=5, convolutions: $\{C_1, \ldots, C_K\}$, deconvolutions: $\{D_1, \ldots, D_K\}$;
Define: $F_k = \prod_{i=1}^{K+1-k} C_i$;
for $k$=1:K do
    if $k$==1 then
        $B_k = F_k \circ D_k$;
    else
        $B_k = Concat([B_{k-1} \circ Up(2); F_k]) \circ D_k$;
    end
    $\hat{Y}_k = B_k \circ Up(2^{K-k}) \circ R(1) \circ Sigmoid$;
    $\mathcal{L}_k = H(\hat{Y}_k, Y)$;
    Optimize $\sum_{i=1}^{k} \mathcal{L}_i$ until converge;
end
$\hat{Y} = \sum_{i=1}^{K} \omega_i \hat{Y}_i$;
$\mathcal{L} = H(\hat{Y}, Y)$;
Optimize $\mathcal{L} + \sum_{i=1}^{K} \mathcal{L}_i$ until converge;
```

Algorithm 1: The training algorithm of P-Net.

1.2.3 Design of Loss Functions

Loss functions compare the segmented volumetric image (e.g., $\hat{Y}$) with ground truth annotation (i.e., Y) and produces segmentation errors for model updating. Cross-entropy loss is one of the most popular loss functions that widely used for foreground–background segmentation. It is defined as

$$\mathcal{L}_{ce} = -\frac{1}{|Y|} \sum_{j \in Y} [y_j \log \hat{y}_j + (1 - y_j) \log(1 - \hat{y}_j)], \tag{1.8}$$

where $|Y|$ is the cardinality (or size) of Y representing the number of voxels in the volumetric image. It can be observed in the formulation of $\mathcal{L}_{ce}$ that errors from every voxel are equally treated. However, it is common in medical volumes that the anatomy of interest occupies only a very small region of the image. Thus, a direct use of $\mathcal{L}_{ce}$ will probably result in the foreground regions to miss or only partially detect. To automatically balance the loss between foreground/background classes, a class-balanced cross-entropy is designed to remedy this problem. It is defined as

$$\mathcal{L}_{cbce} = -\frac{\beta}{|Y_+|}\sum_{j\in Y_+}\log \hat{y}_j - \frac{1-\beta}{|Y_-|}\sum_{j\in Y_-}\log(1-\hat{y}_j), \tag{1.9}$$

where a class-balancing weight β is introduced on a per-voxel term basis. Specifically, we define $\beta = |Y_-|/|Y|$ and $1-\beta = |Y_+|/|Y|$, where Y_+ and Y_- denote the foreground and background ground truth label sets, respectively.

Apart from $\mathcal{L}_{ce}$ and $\mathcal{L}_{cbce}$, many work directly optimize evaluation metrics, e.g. Jaccard index and Dice score. In terms of advantages, the Jaccard loss makes procedures of model training and testing consistent and helps to generate threshold-free probability maps. It is defined as

$$\begin{aligned}\mathcal{L}_{jac} = \mathcal{L}(\hat{Y}, Y) &= 1 - \frac{|Y_+ \bigcap \hat{Y}_+|}{|Y_+ \bigcup \hat{Y}_+|} \\ &= 1 - \frac{\sum_{j\in Y_+}(y_j \wedge \hat{y}_j)}{\sum_{j\in Y_-}(y_j \vee \hat{y}_j)} = 1 - \frac{\sum_{j\in Y_+}(1 \wedge \hat{y}_j)}{|Y_+| + \sum_{j\in Y_-}(0 \vee \hat{y}_j)}.\end{aligned} \tag{1.10}$$

Practically, $\hat{y}_j$ can be relaxed to the value of foreground probability $\in [0, 1]$ and $\mathcal{L}_{jac}$ is then be approximated by

$$\tilde{\mathcal{L}}_{jac} = 1 - \frac{\sum_{f\in Y_+}\min(1, \hat{y}_f)}{|Y_+| + \sum_{b\in Y_-}\max(0, \hat{y}_b)} = 1 - \frac{\sum_{f\in Y_+}\hat{y}_f}{|Y_+| + \sum_{b\in Y_-}\hat{y}_b}. \tag{1.11}$$

The model is then updated by gradient flows as

$$\frac{\partial \tilde{\mathcal{L}}_{jac}}{\partial \hat{y}_j} = \begin{cases} -\frac{1}{|Y_+|+\sum_{b\in Y_-}\hat{y}_b}, \text{ for } j \in Y_+ \\ \\ \frac{\sum_{f\in Y_+}\hat{y}_f}{(|Y_+|+\sum_{b\in Y_-}\hat{y}_b)^2}, \text{ for } j \in Y_- \end{cases} \tag{1.12}$$

Since the inequality $(\sum_{j\in Y_+}\hat{y}_j) < (|Y_+| + \sum_{j\in Y_-}\hat{y}_j)$ holds, the Jaccard loss $\mathcal{L}_{jac}$ assigns greater gradients to foreground pixels than the background ones, which intrinsically balances the foreground and background classes. It empirically works better than the cross-entropy loss $\mathcal{L}_{ce}$ and classed balanced cross-entropy loss $\mathcal{L}_{cbce}$ when segmenting small-sized objects, e.g., pancreas in CT/MRI images.

1.2.4 Experimental Results

1.2.4.1 Experimental Setup

Datasets and evaluation metrics We use two fully annotated pancreas datasets to validate the presented methods. The first one is the NIH-CT dataset [6, 19, 20] that is publicly available and contains 82 abdominal contrast-enhanced 3D CT scans. We organize an in-house MRI dataset [2] that consists of 79 abdominal T1-weighted MRI scans. We treat the CT and MRI datasets as two independent groups and repeat the experiment on both of them. Results from both groups are evaluated to validate the generalization of segmentation methods.

In image preprocessing, we use simple morphological operations to find the abdomen area and have it extracted from the whole image volume. To generate images for training, we use ground truth mask to allocate pancreas and then crop a 256×256 sub-image centered at the target pancreas region. The cropped image patch is then fed for model training. In the data inference phase, we scan testing images with the 256×256 scanning window and fuse outputs together to generate the final segmentation result.

Following the evaluation strategy in [2, 19, 20], we conduct *fourfold cross-validation (CV)* for the reported segmentation result. The set of used evaluation metrics includes the Dice similarity coefficient (DSC): $DSC = 2(|Y_+ \cap \hat{Y}_+|)/(|Y_+| + |\hat{Y}_+|)$, Jaccard index (JI): $JI = (|Y_+ \cap \hat{Y}_+|)/(|Y_+ \cup \hat{Y}_+|)$, foreground precision: $P = |\hat{Y}_+ \cap Y_+|/|\hat{Y}_+|$ and foreground recall: $R = |\hat{Y}_+ \cap Y_+|/|Y_+|$.

Network Implementation We implement HNN [20], U-Net [18], and the introduced P-Net for comparison. Especially, lower layers of HNN are transferred from ImageNet [7] pretrained VGG-16 model [24], and the U-Net is initiated from a ssTEM [18] pretrained model. We note that ssTEM has a very different image statistics from CT and MRI images. Thus, the HNN and U-Net are two baseline methods that fine-tuned from other domains and the proposed P-Net is first initialized with Xavier initialization [9] and then trained from scratch.

Hyperparameters are determined via model selection with the training set. Specifically, the training dataset is first split into a training subset for network parameter training and a validation subset for hyperparameter selection. Denote the training accuracy as Acc_t after model selection, we then combine the training and validation subsets together to further fine-tune the network until its performance on the validation subset converges to Acc_t. Also validated from the validation subset, we observe the P-Net architecture, which contains 5 unit modules with 64 output channels in each convolution/deconvolution layer produces the best empirical performance and meanwhile holds a compact model size. The learning rate (1e-4) together with other hyperparameters are all fixed for all sets so that changes observed in experiment results can be traced back to factors of interest.

1.2.4.2 CNN Comparison

Table 1.1 presents segmentation results of different CNN architectures that trained with $\mathcal{L}_{jac}$. Without loss of generality, we set the output threshold for all CNN outputs to 0.5. P-Net achieved the best performance on both of the CT and MRI datasets. Specifically, it marginally outperformed the HNN baseline by 3.7% and 4.8% Dice scores in CT and MRI segmentation, respectively. In comparison with the U-Net

Table 1.1 Comparison to different CNN architectures including P-Net, HNN [28] and U-Net [18]. Specifically, the P-Net is trained from scratch while HNN and U-Net are fine-tuned from pretrained models

	Models	DSC (%)	JI (%)	Precision	Recall
CT	HNN [28]	79.6 ± 7.7 [41.9, 88.0]	66.7 ± 9.40 [26.5, 78.6]	83.4 ± 6.5 [62.0, 94.9]	77.4 ± 11.6 [28.3, 92.6]
	U-Net [18]	79.7 ± 7.6 [43.4, 89.3]	66.8 ± 9.60 [27.7, 80.7]	81.3 ± 7.5 [49.6, 97.0]	79.2 ± 11.3 [38.6, 94.1]
	P-Net	**83.3 ± 5.6** **[59.0, 91.0]**	**71.8 ± 7.70** **[41.8, 83.5]**	**84.5 ± 6.2** **[60.7, 96.7]**	**82.8 ± 8.37** **[56.4, 94.6]**
MRI	HNN [28]	75.9 ± 10.1 [33.0, 86.8]	62.1 ± 11.3 [19.8, 76.6]	**84.4 ± 6.4** **[61.0, 93.5]**	70.6 ± 13.3 [20.7, 88.2]
	U-Net [18]	79.9 ± 7.30 [54.8, 90.5]	67.1 ± 9.50 [37.7, 82.6]	83.7 ± 6.9 [64.6, 94.6]	77.3 ± 10.3 [46.1, 94.8]
	P-Net	**80.7 ± 7.40** **[48.8, 90.5]**	**68.2 ± 9.64** **[32.3, 82.7]**	84.3 ± 7.6 [55.8, 95.8]	**78.3 ± 10.2** **[38.6, 95.0]**

baseline, P-Net presented 3.6% and 0.8% Dice scores improvements in CT and MRI segmentation, respectively.

1.2.4.3 Loss Function Comparison

Table 1.2 presents comparison results of the three losses, i.e., the cross-entropy loss $\mathcal{L}_{ce}$, the class-balanced cross-entropy loss $\mathcal{L}_{cbce}$ [28], and the Jaccard loss $\mathcal{L}_{jac}$, under fourfold cross-validation with the same P-Net segmentation model. On the CT dataset, $\mathcal{L}_{jac}$ outperformed $\mathcal{L}_{ce}$ and $\mathcal{L}_{cbce}$ by 0.5% and 0.2% Dice scores, respectively. On the MRI dataset, also achieved the best performance referring to the Dice score and Jaccard index. We then evaluate the stability of segmentation performance with various thresholds. The CNN network usually outputs probabilistic segmentation maps instead of binary masks and an appropriate probability threshold is required to obtain the final binarized segmentation outcomes. However, it is often nontrivial to find the optimal probability threshold in practice. Figure 1.4 visually depicts results of our analysis that the probability output maps from the Jaccard loss $\mathcal{L}_{jac}$ delivered the steadiest segmentation results referring to different output thresholds. Empirically, the Naïve $\mathcal{L}_{ce}$ assigns same penalties on positive and negative pixels so that the probability threshold should be around 0.5. Meanwhile, $\mathcal{L}_{cbce}$ gives a higher penalty on positive pixels (due to its scarcity) making the resulted optimal threshold at a relatively higher value. By contrast, $\mathcal{L}_{jac}$ pushes the foreground pixels to the probability of 1.0 while remaining to be strongly discriminative against the background pixels. Thus, the plateau around the optimal segmentation performance of $\mathcal{L}_{jac}$ would be much wider than $\mathcal{L}_{ce}$ and $\mathcal{L}_{cbce}$ so that it could perform stably in a wide range of thresholds, i.e., [0.05, 0.95] in our experiments.

Table 1.2 Comparision of loss functions: $\mathcal{L}_{ce}$, $\mathcal{L}_{cbce}$, and $\mathcal{L}_{jac}$ with P-Net

	Loss	mean ± stdv. [min, max]	
		Dice score (%)	Jaccard index (%)
CT	$\mathcal{L}_{ce}$	83.5 ± 5.6 [59.3, 91.1]	72.0 ± 7.70 [42.2, 83.6]
	$\mathcal{L}_{cbce}$	83.2 ± 5.7 [57.2, 90.3]	71.6 ± 7.80 [40.1, 82.4]
	$\mathcal{L}_{jac}$	**83.7 ± 5.4 [58.4, 90.4]**	**72.3 ± 7.50 [41.3, 82.4]**
MRI	$\mathcal{L}_{ce}$	80.0 ± 7.60 [50.7, 89.9]	67.3 ± 9.80 [34.0, 81.6]
	$\mathcal{L}_{cbce}$	**80.2 ± 7.20 [53.6, 90.5]**	**67.6 ± 9.50 [36.6, 82.7]**
	$\mathcal{L}_{jac}$	**80.2 ± 7.90 [51.2, 90.1]**	**67.6 ± 10.3 [34.4, 82.0]**

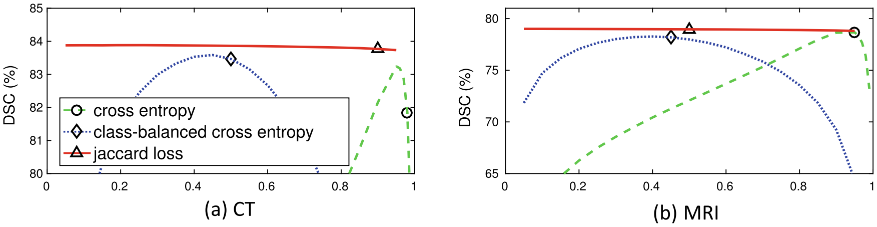


Fig. 1.4 Plot of the threshold versus Dice score (DSC): the proposed jaccard loss $\mathcal{L}_{jac}$ performs the steadiest across thresholds in the range of [0.05, 0.95] comparing to the cross-entropy loss $\mathcal{L}_{ce}$ and the class-balanced cross-entropy loss $\mathcal{L}_{cbce}$. The threshold that selected from validation dataset is marked as ○, ◇, and △ for losses $\mathcal{L}_{ce}$, $\mathcal{L}_{cbce}$, and $\mathcal{L}_{jac}$, respectively

1.3 Recurrent Neural Network for Contextual Learning

Previous work [2, 20, 30] perform deep 2D CNN segmentation on CT (or MRI) axial slices independently, not taking the correlation between neighboring images into consideration. Organ segmentation in 3D volumes can also be performed by directly taking cropped 3D sub-volumes as input to 3D CNNs [10, 14, 15]. However, even at the expense of being computationally expensive and prone-to-overfitting [30], the result of very high segmentation accuracy has not been reported for complexly shaped organs [14], or small anatomical structures [10]. Despite more demanding memory requirement, 3D CNN approaches deserve more investigation for future work. On the other hand, [3, 25] use hybrid CNN-RNN architectures to process/segment sliced CT (or MRI) images in sequence and present a promising direction to process CT and MRI segmentations. However, these methods do not apply spatial shape continuity constrain or regularization to enforce the segmentation consistency among successive slices. Thus, in this chapter, we present our own research for regulating pancreas segmentation across 3D slices with recurrent neural network.

1.4 Recurrent Neural Network

As discussed above, the P-Net processes pancreas segmentation with individual 2D image slices, delivering remarkable performance on the tested CT and MRI datasets. However, as shown in the first row of Fig. 1.5, the transition among the resulting CNN pancreas segmentation regions in the consecutive slices may not be smooth which often implies boarder failures of segmentations. Adjacent CT/MRI slices are expected to be correlated with each other thus segmentation results from successive slices need to be constrained for shape continuity. The model for 3D object segmentation is required to be able to detect and recover abnormally lost part inside slices (see $\hat{Y}_\tau$ in Fig. 1.5).

To achieve this, we concatenate a recurrent neural network (RNN) subnetwork to the output end of P-Net for modeling inter-slice shape continuity and regularization. The RNN is originally designed for sequential data analysis and thus naturally meets the need of processing the ordered image slices. Specifically, we slice the CT (or MRI) volume into an ordered sequence of 2D images and process to learn the segmentation shape continuity among neighboring image slices with a typical RNN architecture, the long short-term memory (LSTM) unit. However, the standard LSTM requires vectorized input, which will sacrifice the spatial information encoded in the output of CNN. To circumvent such problem, we utilize the convolutional-LSTM (C-LSTM) model [23] to preserve the 2D image segmentation layout by CNN. As shown in Fig. 1.5, H_τ and C_τ are the hidden state and cell output of C-LSTM in respective at the τth slice. The current cell output C_τ is computed based on both of the former cell hidden state $H_{\tau-1}$ and the current CNN output $\hat{Y}_\tau$. Then, H_τ will be calculated from C_τ and used to produce the next cell output $C_{\tau+1}$. Contextual information is propagated from slice τ to $\tau + 1$ through convolutional operations.

The strategy of the C-LSTM based context continuity learning is built upon an intuitive conditional probability assumption. Segmentation results of the former image slices are encoded in the cell hidden state $H_{\tau-1}$. Values of C_τ is decided by taking $H_{\tau-1}$ and $\hat{Y}_\tau$ together into consideration. If position p_i in $\hat{Y}_\tau$ is predicted as pancreatic tissue by the CNN model (e.g. P-Net), and the same position in $H_{\tau-1}$ are also encoded as pancreatic tissue, then with high confidence that position p_i should be a pancreatic pixel in C_τ, and vice versa. As a result, C-LSTM not only recovers missing segmentation parts but also outputs more confident probability maps than the original CNN subnetwork.

Formally, the C-LSTM unit is formulated,

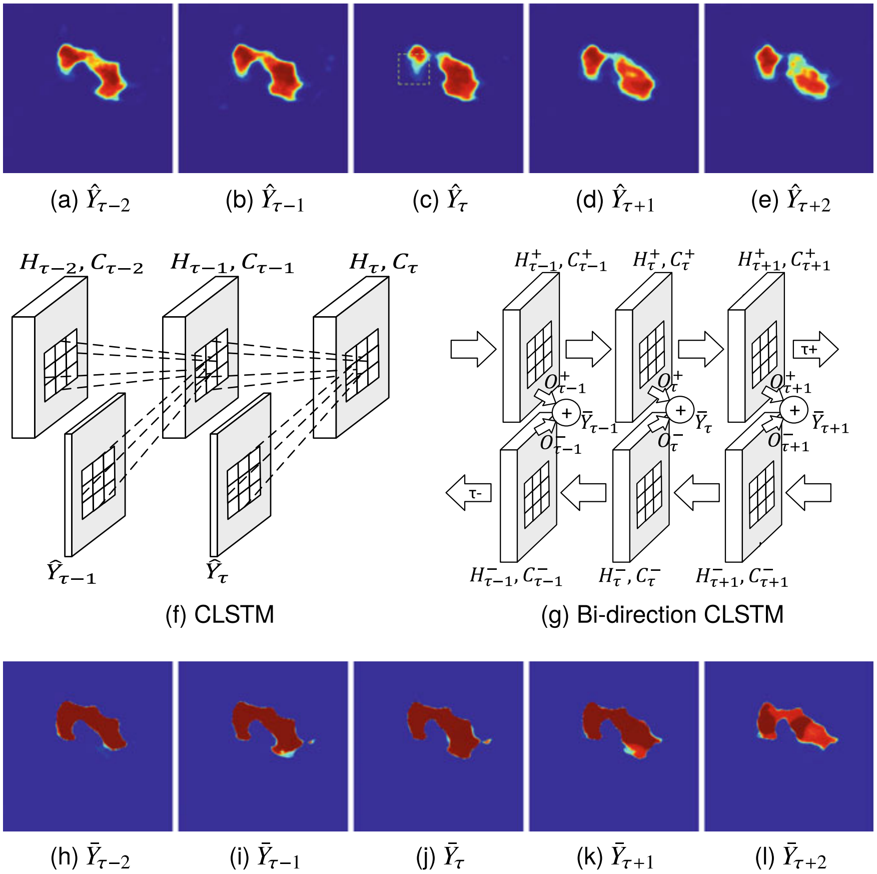
$$i_\tau = \sigma(W_{yi} * \hat{Y}_\tau + W_{hi} * h_{\tau-1} + W_{ci} \odot c_{\tau-1} + b_i), \tag{1.13}$$

$$f_\tau = \sigma(W_{yf} * \hat{Y}_\tau + W_{hf} * h_{\tau-1} + W_{cf} \odot c_{\tau-1} + b_f), \tag{1.14}$$

$$c_\tau = f_\tau \odot c_{\tau-1} + i_\tau \odot \tanh(W_{yc} * \hat{Y}_\tau + W_{hc} * h_{\tau-1} + b_c), \tag{1.15}$$

$$o_\tau = \sigma(W_{yo} * \hat{Y}_\tau + W_{ho} * h_{\tau-1} + W_{co} \odot c_\tau + b_o), \tag{1.16}$$

$$h_\tau = o_\tau \odot \tanh(c_\tau), \tag{1.17}$$


(a) $\hat{Y}_{\tau-2}$ (b) $\hat{Y}_{\tau-1}$ (c) $\hat{Y}_{\tau}$ (d) $\hat{Y}_{\tau+1}$ (e) $\hat{Y}_{\tau+2}$

(f) CLSTM (g) Bi-direction CLSTM

(h) $\bar{Y}_{\tau-2}$ (i) $\bar{Y}_{\tau-1}$ (j) $\bar{Y}_{\tau}$ (k) $\bar{Y}_{\tau+1}$ (l) $\bar{Y}_{\tau+2}$

Fig. 1.5 The main construction units of the proposed RNN model and its input/output segmentation sequence. The sequence of CNN outputs is shown in the first row (**a**–**e**), is taken as the input of the bidirectional C-LSTM (**g**), which is an RNN architecture composed of two layers of C-LSTM (**f**) working in opposite directions. The third row (**h**–**l**) presents the corresponding output sequence, which is sharp and clean. Note that the missing pancreatic part in $\hat{Y}_{\tau}$ (**c**), in the green dashed box, is recovered by shape continuity modeling in $\bar{Y}_{\tau}$ (**j**). For visual clarity, we omit the input $\hat{Y}_{(\cdot)}$ in the bidirectional CLSTM (**g**), which is same as in (**f**)

where $*$ represents convolution operation, and $\odot$ denotes the Hadamard product. Gates i_τ, f_τ, o_τ are the input, forget, and output, respectively, following the original definition of C-LSTM. $W_{(\cdot)}$, and $b_{(\cdot)}$ are weights and bias in the corresponding C-LSTM unit that needs model optimization. Finally, $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid and hyperbolic tangent activation function, respectively.

1.4.1 Bidirectional Contextual Regularization

Next, we have the contextual learning extended to a bidirectional. For pancreas, as well as other organs, its shape in the current slice is constrained by slices from not only its former slices but also the followings. The contextual information to the input could be doubled if the shape regularization is taken in both directions leading to a further improvement. Two layers of C-LSTM are stacked working in two opposite directions as shown in Fig. 1.5. Then, outputs of the two layers, one in the τ^--direction and the other in the τ^+-direction, are combined as the final segmentation output,

$$\bar{Y}_\tau = \sum_{i \in \{\tau^-, \tau^+\}} \lambda^i o_\tau^i, \tag{1.18}$$

where i represents the τ^- and τ^+ directions, and λ^i is the learned weights when combining CLSTM outputs from both directions. Thus, the bidirectional design of shape continuity modeling permits to explicitly enforce the pancreas segmentation to be spatial smooth and higher order inter-slice regularized.

Lastly, we define the objective of contextual learning based on Jaccard loss as

$$\mathcal{L}_{rnn} = \sum_{\tau=1}^{T} \mathcal{L}_{jac}(\bar{Y}_\tau, Y), \tag{1.19}$$

where T is the length of image sequence processed in each unit, and we empirically set $T = 5$ in our experiments.

1.4.2 Experimental Results

Given outputs of P-Net as the best CNN-based segmentation results, the bidirectional RNN (BiRNN) subnetwork is then stacked to the output end of P-Net and trained end to end. In each direction, a one-layer CLSTM is implemented with one hidden state and 3 × 3 convolution filter kernels [23]. Particularly, the number of hidden state is set to 1 since our shape continuity learning is inherently simplified by processing only the output probability maps of CNN subnetwork. CNN output $\hat{Y}_\tau \in R^{d_1^1 \times d_1^2 \times 1}$, where d_1^1 and d_1^2 are the width and height of the input image, provides a highly compacted representation of the input CT/MRI image for shape learning. Thus, BiRNN with the hidden state $H_\tau \in R^{d_1^1 \times d_1^2 \times 1}$ is sufficient to model and capture the shape continuity regularization among CNN outputs. We notice that BiRNN cannot converge stably during model training when a larger hidden state is used. In addition, we attempt to employ BiRNN on the feature maps from CNN's intermediate layers. However, this causes the model training process failed to converge. Thus, we mainly focus on the current design of BiRNN, which emphasizes to learn the inter-slice shape continuity among the successive segmentation outputs of P-Net.

Table 1.3 Evaluate pancreas segmentation on the CT dataset. BiRNN refines output of P-Net providing better performance in both volume measurement (DSC) and surface reconstruction (HD)

Method	HD (mm)	DSC (%)
P-Net	0.61 ± 0.53 [0.15, 3.48]	83.3 ± 5.6 [59.0, 91.0]
BiRNN	**0.54 ± 0.53 [0.12, 3.78]**	**83.7 ± 5.1 [59.0, 91.0]**

We model the segmentation shape continuity as a higher order inter-slice regularization among the CT/MRI axial slices. The average physical slice thickness in CT and MRI are 2 mm and 7 mm, respectively. Thus, slight shape change occurs between two correct segmented slices. Given the measured Hausdorff distance (HD) of neighboring slices in ground truth, the mean ± standard deviation of shape changes in CT and MRI are 0.35 ± 0.94 mm and 2.69 ± 4.45 mm, respectively. The drastic shape changes in MRI volumes indicates that successive MRI image slices are actually more independent, so that in our implementation, shape continuity learning brings marginal but consistent performance improvement. The improvement in CT images is more evident. Specifically, we detect abnormal shape changes in the outputs of CNN and have them refined by BiRNN. We define abnormal shape change occurs between two neighboring CT when $HD\ (\hat{Y}_\tau, \hat{Y}_{\tau-1}) > 0.5$ mm, which is decided basing on the shape change statics in the CT dataset.

Table 1.3 illustrates performance with and without shape continuity learning, where BiRNN boost volume segmentation (i.e., DSC) by 0.4%. More importantly, the error for pancreatic surface reconstruction (i.e., HD) drops from 0.61 to 0.54 mm, improved by 11.5%. Figure 1.7 further shows the segmentation performance difference statistics, with or without contextual learning in subject-wise. In particular, those cases with low DSC scores are greatly improved by BiRNN.

Finally, Fig. 1.6 displays examples of output probability maps from all of the comparative methods, i.e., HNN [28], U-Net [18], P-Net and *P-Net+BiRNN*, where the latter one delivers the sharpest and clearest output on both CT and MRI datasets. More specifically, P-Net presents detailed results that recover the major part of the pancreas, where both HNN and U-Net suffer from significant lower segmentation recall. When observing the BiRNN outputs for CT and MRI, we find detailed pancreas parts in CT have been recovered via shape continuity learning and regularization, while in MRI, the BiRNN only outputs probability map with the same shape in P-Net's output, which is optimal when the inter-slice shape changes drastically in the MRI dataset. Thus, BiRNN would help refine pancreas segmentation with a smoothed surface in the situation that slice thickness of the 3D scans is reasonably small, e.g., <2 mm.

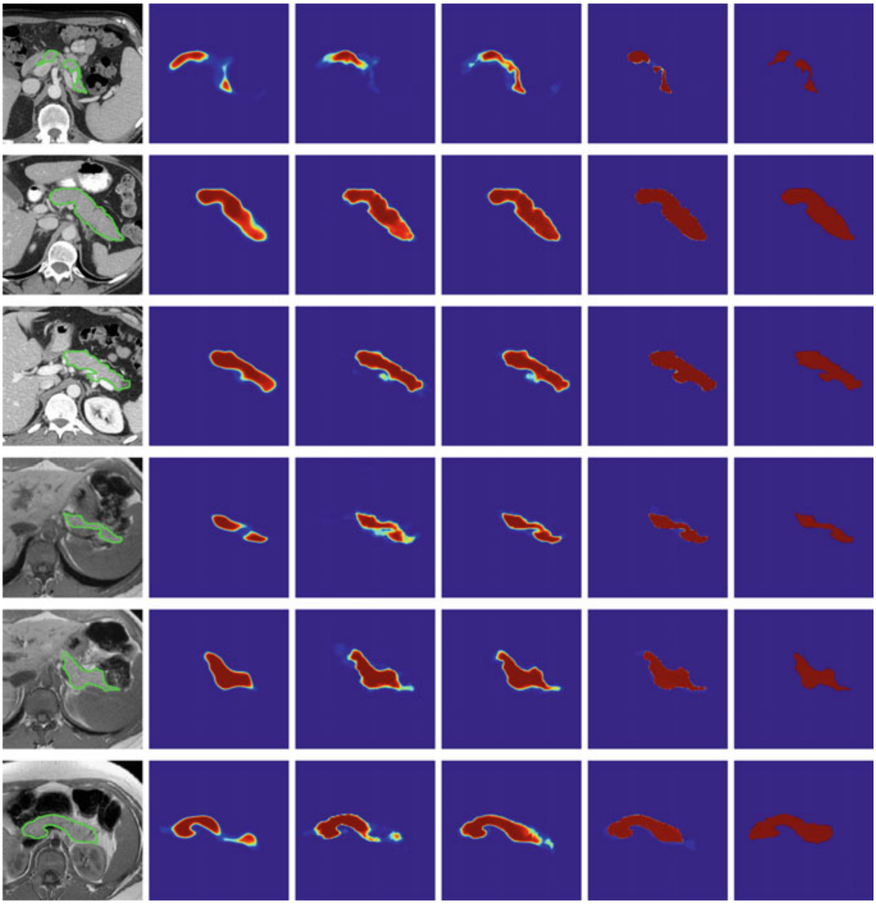


Fig. 1.6 Examples of output probability map: columns from left to right are the input CT/MRI image and results from HNN [28], U-Net [18], P-Net, and the full CNN-RNN (*P-Net+BiRNN*) model, and the ground truth. The CNN-RNN model delivers the most clear probability maps which preserve detailed pancreatic boundaries

1.5 State-of-the-Art Methods for Pancreas Segmentation

We compare selected state-of-the-art methods for pancreas segmentation. Dice score and Jaccard index are computed and reported in Table 1.4 under fourfold CV. The method *P-Net+BiRNN* performs the best on the CT dataset and P-Net achieves the best result on the MRI dataset. We notice that the current implementation of FCN 3D [22] is not as effective as its 2D segmentation counterparts, where *P-Net+BiRNN* outperforms FCN 3D by a large margin of 6.9% Dice score. The problem of segmenting 3D CT/MRI image volumes within a single inference is much more complex than the 2D CNN approaches where further network architecture exploration as well as

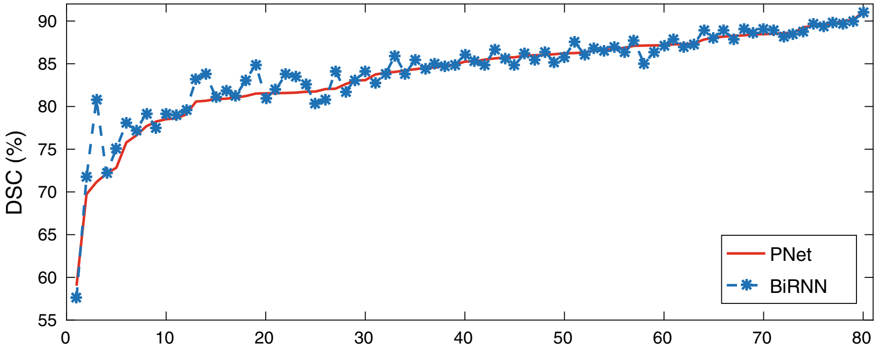


Fig. 1.7 Comparison of P-Net and *P-Net+ BiRNN* outputs for all 80 NIH-CT scans and the scans are sorted left to right using Dice scores of P-Net. Small fluctuations among the well- segmented cases (on the top right) possibly result from model updating, which can be omitted as noise

Table 1.4 Performance of the state-of-the-art methods for segmentation under fourfold CV. We show Dice score and Jaccard index in the form of *mean* ± *standard dev. [worst case, best case]*. The best results on CT and MRI are highlighted in bold

	Method	Dice score (%)	Jaccard index (%)
CT	3D FCN [22]	76.8 ± 9.4 [43.7, 89.4]	
	Roth et al. [20]	78.0 ± 8.2 [34.1, 88.6]	
	Roth et al. [21]	81.3 ± 6.3 [50.6, 88.9]	68.8 ± 8.12 [33.9, 80.1]
	Coarse-to-Fine [30]	82.3 ± 5.6 [62.4, 90.8]	
	CNN+RNN [1]	82.4 ± 6.7 [60.0, 90.1]	70.6 ± 9.00 [42.9, 81.9]
	P-Net	83.3 ± 5.6 [59.0, 91.0]	71.8 ± 7.70 [41.8, 83.5]
	P-Net+BiRNN	**83.7 ± 5.1 [59.0, 91.0]**	**72.3 ± 7.04 [41.8, 83.5]**
MRI	Graph-Fusion [2]	76.1 ± 8.70 [47.4, 87.1]	
	CNN+RNN [1]	80.5 ± 6.70 [59.1, 89.4]	67.9 ± 8.90 [41.9, 80.9]
	P-Net	**80.7 ± 7.40 [48.8, 90.5]**	**68.2 ± 9.64 [32.3, 82.7]**

more training images are typically required. This is also referred as *curse of dimensionality* in [19, 30]. In this scenario, we would argue that 2D network architectures may still be optimal for pancreas segmentation with large inter-slice thicknesses. We also note that our intuition of developing CNN-RNN combination is orthogonal to the principles of *coarse-to-fine* pancreas location and detection [21, 30]. Better performance may be achievable with the combination of both methodologies. Figure 1.8 visually depicts examples of reconstructed 3D segmentation results from the CT dataset.

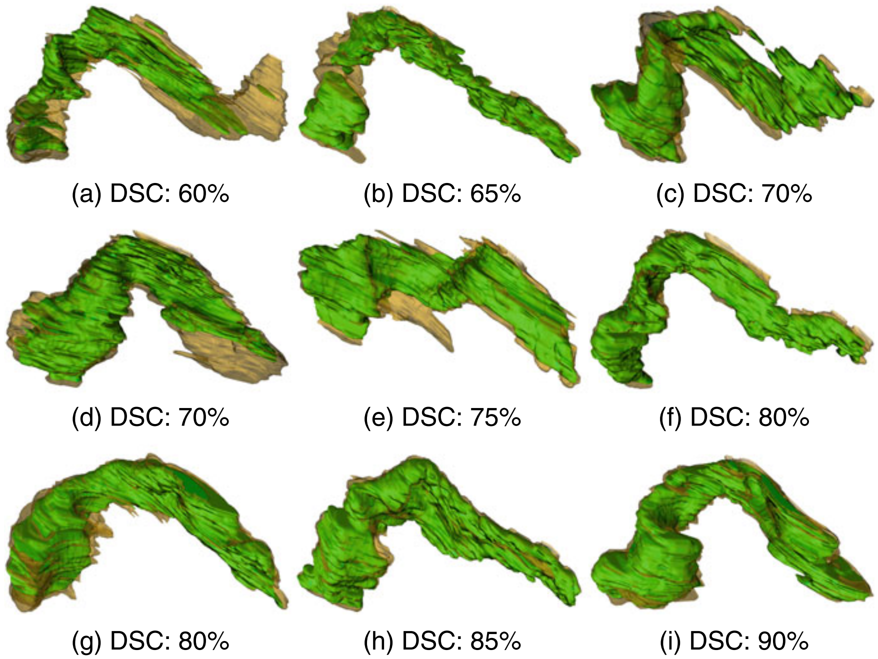

(a) DSC: 60%
(b) DSC: 65%
(c) DSC: 70%
(d) DSC: 70%
(e) DSC: 75%
(f) DSC: 80%
(g) DSC: 80%
(h) DSC: 85%
(i) DSC: 90%

Fig. 1.8 3D visualization of pancreas segmentation results where human annotation shown in yellow and computerized segmentation displayed in green. The DSC are 90%, 75%, and 60% for three examples from left to right, respectively

1.6 Summary

In this chapter, we present a novel CNN-RNN architecture for pancreas segmentation in CT and MRI scans via our tailor-made CNN module (P-Net) followed by a bidirectional C-LSTM (BiRNN). It is presented to regularize the segmentation results on individual image slices. The shape continuity regularization permits to enforce the pancreas segmentation spatial smoothness explicitly in the axial direction, in analogy to comprehending into videos by parsing and aggregating successive frames [16]. This may also share some similarity to the human doctor's way of reading radiology images. Combined with the proposed Jaccard loss function for model training to generate the threshold-free segmentation results, our quantitative pancreas segmentation result outperforms the previous state-of-the-art approaches [2, 20, 21, 30] on both CT and MRI datasets, with noticeable margins. Although the discussion focuses on pancreas segmentation in this chapter, the approaches would be generalizable to other organ segmentations in medical image analysis.

References

1. 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: MICCAI, pp 674–682. Springer
2. Cai J, Lu L, Zhang Z, Xing F, Yang L, Yin Q (2016) Pancreas segmentation in MRI using graph-based decision fusion on convolutional neural networks. In: MICCAI, pp 442–450. Springer
3. Chen J, Yang L, Zhang Y, Alber MS, Chen DZ (2016) Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In: NIPS, pp 3036–3044
4. Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI 40(4):834–848
5. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O (2016) 3d u-net: learning dense volumetric segmentation from sparse annotation. In: MICCAI, pp 424–432. Springer
6. Clark KW, Vendt BA, Smith KE, Freymann JB, Kirby JS, Koppel P, Moore SM, Phillips SR, Maffitt DR, Pringle M, Tarbox L, Prior FW (2013) The cancer imaging archive (TCIA): maintaining and operating a public information repository. J Digit Imaging 26(6):1045–1057
7. Deng J, Dong W, Socher R, Li L, Li K, Li F (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), 20–25 June 2009, Miami, Florida, USA, pp 248–255
8. Farag A, Lu L, Roth HR, Liu J, Turkbey E, Summers RM (2017) A bottom-up approach for pancreas segmentation using cascaded superpixels and (deep) image patch labeling. TMI 26(1):386–399
9. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: AISTATS, pp 249–256
10. Kamnitsas K, Ledig C, Newcombe VFJ, Simpson JP, Kane AD, Menon DK, Rueckert D, Glocker B (2017) Efficient multi-scale 3d CNN with fully connected CRF for accurate brain lesion segmentation. MIA 36:61–78
11. Karasawa K, Oda M, Kitasaka T, Misawa K, Fujiwara M, Chu C, Zheng G, Rueckert D, Mori K (2017) Multi-atlas pancreas segmentation: Atlas selection based on vessel structure. MIA 39:18–28
12. Lee C, Xie S, Gallagher PW, Zhang Z, Tu Z (2015) Deeply-supervised nets. In: AISTATS
13. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR, pp 3431–3440
14. Merkow J, Marsden A, Kriegman DJ, Tu Z (2016) Dense volume-to-volume vascular boundary detection. In: MICCAI, pp 371–379
15. Milletari F, Navab N, Ahmadi S (2016) V-net: fully convolutional neural networks for volumetric medical image segmentation. In: International conference on 3D vision, pp 565–571
16. Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: deep networks for video classification. In: ICCV, pp 4694–4702
17. Oda M, Shimizu N, Karasawa K, Nimura Y, Kitasaka T, Misawa K, Rueckert D, Mori K (2016) Regression forest-based atlas localization and direction specific atlas generation for pancreas segmentation. In: MICCAI, pp 556–563. Springer
18. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: MICCAI, pp 234–241
19. 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, pp 556–564. Springer
20. Roth HR, Lu L, Farag A, Sohn A, Summers RM (2016) Spatial aggregation of holistically-nested networks for automated pancreas segmentation. In: MICCAI, pp 450–451. Springer
21. Roth HR, Lu L, Lay N, Harrison AP, Farag A, Summers RM (2018) Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation. MIA 45:94–107

22. Rotha HR, Odaa H, Zhoub X, Shimizua N, Yanga Y, Hayashia Y, Odaa M, Fujiwarac M, Misawad K, Moria K (2018) An application of cascaded 3D fully convolutional networks for medical image segmentation. ArXiv e-prints
23. Shi X, Chen Z, Wang H, Yeung D, Wong W, Woo W (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: NIPS, pp 802–810
24. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 1–14
25. Stollenga MF, Byeon W, Liwicki M, Schmidhuber J (2015) Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In: NIPS, pp 2998–3006
26. Tong T, Wolz R, Wang Z, Gao Q, Misawa K, Fujiwara M, Mori K, Hajnal JV, Rueckert D (2015) Discriminative dictionary learning for abdominal multi-organ segmentation. MIA 23(1):92–104
27. Wolz R, Chu C, Misawa K, Fujiwara M, Mori K, Rueckert D (2013) Automated abdominal multi-organ segmentation with subject-specific atlas generation. TMI 32(9):1723–1730
28. Xie S, Tu Z (2015) Holistically-nested edge detection. In: ICCV, pp 1395–1403
29. Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, Du D, Huang C, Torr PH (2015) Conditional random fields as recurrent neural networks. In: ICCV, pp 1529–1537
30. Zhou Y, Xie L, Shen W, Fishman E, Yuille AL (2016) Pancreas segmentation in abdominal CT scan: a coarse-to-fine approach. http://arxiv.org/abs/1612.08230

Chapter 2
Deep Learning for Muscle Pathology Image Analysis

Yuanpu Xie, Fujun Liu, Fuyong Xing and Lin Yang

Abstract Inflammatory myopathy (IM) is a kind of heterogeneous disease that relates to disorders of muscle functionalities. The identification of IM subtypes is critical to guide effective patient treatment since each subtype requires distinct therapy. Image analysis of hematoxylin and eosin (H&E)-stained whole-slide specimens of muscle biopsies are considered as a gold standard for effective IM diagnosis. Accurate segmentation of perimysium plays an important role in early diagnosis of many muscle inflammation diseases. However, it remains as a challenging task due to the complex appearance of the perimysium morphology and its ambiguity to the background area. The muscle perimysium also exhibits strong structure spanned in the entire tissue, which makes it difficult for current local patch-based methods to capture this long-range context information. In this book chapter, we propose a novel spatial clockwork recurrent neural network (spatial CW-RNN) to address those issues. Besides perimysium segmentation, we also introduce a fully automatic whole-slide image analysis framework for IM subtype classification using deep convolutional neural networks (DCNNs).

Y. Xie and F. Liu contributed equally to this work.

Y. Xie
J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida,
Gainesville, FL 32611, USA
e-mail: shampool@ufl.edu

F. Liu
Department of Computer and Information Science and Engineering, University of Florida,
Gainesville, FL 32611, USA
e-mail: fujunliu@ufl.edu

F. Xing · L. Yang (✉)
Department of Biostatistics and Informatics, University of Colorado Anschutz Medical Campus,
Aurora, CO 80045, USA
e-mail: lin.yang@bme.ufl.edu

F. Xing
e-mail: fuyong.xing@ucdenver.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_2

2.1 Introduction

Many important morphological properties, such as the distribution of muscle fibers and their nuclei with respect to the perimysium, are important biomarkers for early diagnosis of many muscle diseases [1]. To compute these spatial morphological parameters, accurate and efficient segmentation of perimysium is an essential prerequisite. However, muscle perimysium often shares similar appearances to other structures in the muscle, such as endomysium, epimysium, and blood vessels. The large variations in staining intensity, global structure, and morphology further complicate the automated segmentation task.

Inflammatory myopathies (IM) are a set of heterogeneous diseases that relate to disorders of muscle functionalities [2]. Based on distinct clinicopathological features, IM can be classified into three major subtypes: dermatomyositis (DM), polymyositis (PM), and inclusion body myositis (IBM). The identification of IM subtypes is very important to guide effective patient treatment since each subtype has a different prognosis and responds to distinct therapy [3].

In clinic practice, the diagnosis of IM subtypes usually consists of two steps: (1) A pathologist conducts magnetic resonance imaging (MRI) screening to select regions of muscle tissues displaying relevant histological features and retrieve a biopsy; (2) The digitalized image of the muscle biopsy is achieved using the whole-slide imaging (WSI) technology and analyzed by a pathologist for subtype diagnosis. Currently, the diagnosis of whole-slide image is still a manual or, at best, a semiautomated process, which is labor intensive and prone to errors, leading to high interobserver variability. In this work, we propose a fully automatic whole-slide image analysis framework for IM subtype classification using deep convolutional neural networks (DCNNs) [4–9].

2.2 Muscle Perimysium Segmentation

Enormous efforts have been devoted to utilizing RNN on computer vision tasks. Francesco [10] applies GRU [11] to sweep the images as one chain-structured data but along four different directions to model the context information. Some pioneering works [12, 13] that exploit the potentials of multidimensional RNN in semantic image segmentation have also achieved promising results. However, 2D plain RNN [12] suffers from the exploding or the vanishing gradient problem for large images, and 2D LSTM [13] contains much more parameters than 2D RNN, which makes it inefficient at the runtime, and sometimes over-fit can happen especially when the amount of training data is limited.

In this chapter, we propose an 2D spatial clockwork RNN which extends the applicability of chain-structured CW-RNN [14] to 2D image domain for efficient perimysium segmentation. Our model directly exploits the 2D structure of images and encodes the global context information among local image patches. Different from [12, 13], our model contains a much smaller number of parameters, which makes it computationally efficient and suitable for medical image segmentation with limited training data. In our algorithm, instead of conducting an inefficient patchwise classification, we integrate the structured regression [15] into the proposed algorithm. This allows us to use nonoverlapping stride in both training and testing stages. Extensive experimental results demonstrate the effectiveness and efficiency of our proposed model. To the best of our knowledge, this is the first work to propose an 2D spatial CW-RNN that achieves promising results on biomedical image segmentation.

2.2.1 Recurrent Neural Network

The recurrent neural network (RNN) is one type of neural network that is equipped with recurrent connections, which enable the network to memorize past input patterns. For the simple RNN (SRNN), at each time step, its current hidden state $\boldsymbol{h}^t$ is a nonlinear transformation of the current input $\boldsymbol{x}^t$ and the hidden state $\boldsymbol{h}^{t-1}$ from the last step. The output $\boldsymbol{o}^t$ is directly connected to $\boldsymbol{h}^t$. Mathematically, those relationships can be expressed by the following equations:

$$\boldsymbol{h}^t = f(\boldsymbol{W}\boldsymbol{x}^t + \boldsymbol{U}\boldsymbol{h}^{t-1} + \boldsymbol{b}_h), \tag{2.1}$$

$$\boldsymbol{o}^t = g(\boldsymbol{V}\boldsymbol{h}^t + \boldsymbol{b}_o), \tag{2.2}$$

where $f(.)$, $g(.)$ represent the nonlinear activation functions, $\boldsymbol{W}$ and $\boldsymbol{U}$ are weight matrices connecting input units to hidden units, and hidden units to themselves, respectively. $\boldsymbol{V}$ is the weight matrix connecting hidden units to the output units. $\boldsymbol{b}_h$ and $\boldsymbol{b}_o$ represents the bias terms for the hidden and output layer, respectively.

SRNN is usually trained with a discriminative objective function using the backpropagation through time (BPTT) algorithm [16]. However, the fact that the computed gradients of SRNN are either exploding or vanishing when T becomes large hinders the SRNN from learning long-term temporal dependencies. Instead of introducing gated connections [11, 17] to complicate the model, clockwork RNN (CW-RNN) [14] addresses the long-term dependency issue by using a clever trick. Specifically, the hidden units $\boldsymbol{h}$ are partitioned into M modules ($\boldsymbol{h}^m$ for $i = 1, \ldots, M$), each is of size k and associated with a clock (or temporal) period $T_i \in \{T_1, \ldots, T_M\}$. The total length of the hidden units is $hid = M \times k$. At each time step t, the neurons in module i will be updated only when t satisfies $(t \ mod \ T_i) = 0$. Units corresponding to slower rates are thus capable of preserving long-term information. In addition, connections between hidden units are restricted that faster modules can only receive information from slower ones and not vice versa, this mechanism further reduces the total number of active weights.

2.2.1.1 Spatial Clockwork RNN

Since there are no existing sequences presented in static images, the aforementioned CW-RNN is not directly applicable to our application. Although we can reshape the tensor containing all the image patches into a chain-structured representation, this type of simplification are problematic due to the fact that interactions between image patches are beyond chain. To ameliorate this problem, we extend the CW-RNN to a two-dimensional domain, in which current state can receive information from its predecessors in both row and column directions. We denote the time step (r, c) for each local patch as the coordinate of this image patch at the image grid. In order for the spatial CW-RNN to process the image, all image patches need to be sorted to an acyclic sequence. One of the possible sorting strategies is $(x_1, x_2) < (x_1', x_2')$ if $\exists\, i \in \{1, 2\}$ such that $x_i < x_i'$ and $x_j = x_j', \forall j \neq i$. Without loss of generality, we define for every current step (r, c), we define its predecessor as the local patches that are processed by model before it reaches the current position.

Specifically, we maintain one sub-hidden state for both row and column dimension, denoted as $\widehat{h}$ and $\widetilde{h}$, which are composed together as the hidden states $H = [\widehat{h},\ \widetilde{h}]$. Denote respectively the weights matrix connecting the current hidden states to its row and column predecessor as $\widehat{U}$ and $\widetilde{U}$, which are split into four $hid \times hid$ block matrices; W connecting the input units to hidden units is partitioned into 2 $input\_dim \times hid$ blocks columns; the bias b^h is also evenly separated into 2 groups: $\widehat{U} = \begin{pmatrix} \widehat{U}^{(1,1)} & \widehat{U}^{(1,2)} \\ \widehat{U}^{(2,1)} & \widehat{U}^{(2,2)} \end{pmatrix}$, $\widetilde{U} = \begin{pmatrix} \widetilde{U}^{(1,1)} & \widetilde{U}^{(1,2)} \\ \widetilde{U}^{(2,1)} & \widetilde{U}^{(2,2)} \end{pmatrix}$, $W = \begin{pmatrix} W^1 & W^2 \end{pmatrix}$, and $b = \begin{pmatrix} b^1 & b^2 \end{pmatrix}$. Each block matrix $\widehat{U}^{(m,n)}$, $\widetilde{U}^{(m,n)}$ are further partitioned into $M \times M$ smaller block matrices with the same size $k \times k$. W^m and b^m is partitioned into M blocks columns as well; $\widehat{U}^{(m,n)} = \begin{pmatrix} \widehat{U}^{(m,n)}_{(1,1)} & \cdots & \widehat{U}^{(m,n)}_{(1,M)} \\ \vdots & \cdots & \vdots \\ \widehat{U}^{(m,n)}_{(M,1)} & \cdots & \widehat{U}^{(m,n)}_{(M,M)} \end{pmatrix}$, $\widetilde{U}^{(m,n)} = \begin{pmatrix} \widetilde{U}^{(m,n)}_{(1,1)} & \cdots & \widetilde{U}^{(m,n)}_{(1,M)} \\ \vdots & \cdots & \vdots \\ \widetilde{U}^{(m,n)}_{(M,1)} & \cdots & \widetilde{U}^{(m,n)}_{(M,M)} \end{pmatrix}$, $W^m = \begin{pmatrix} W_1^m & \cdots & W_M^m \end{pmatrix}$ and $b^m = \begin{pmatrix} b_1^m & \cdots & b_M^m \end{pmatrix}$. Recall that each sub-hidden state $\widehat{h}$ and $\widetilde{h}$ is partitioned into M module, each runs at specific temporal rate. Denote $i, j \in \{1, \ldots, M\}$ as the modules index, $u \in \{1, 2\}$ matrix identifier, and (r, c) as the time step. For brief narrative, we define the following general matrix placeholders: $\mathcal{H}_i^{(r,c)} = \begin{pmatrix} \widehat{h}_i^{(r,c)} & \widetilde{h}_i^{(r,c)} \end{pmatrix}$, $\widehat{\mathcal{U}}_{ij}^u = \begin{pmatrix} \widehat{U}_{ij}^{(u,1)} \\ \widehat{U}_{ij}^{(u,2)} \end{pmatrix}$, and $\widetilde{\mathcal{U}}_{ij}^u = \begin{pmatrix} \widetilde{U}_{ij}^{(u,1)} \\ \widetilde{U}_{ij}^{(u,2)} \end{pmatrix}$. The updating rule for the ith module of $\widehat{h}$ (similar case for $\widetilde{h}$) at time step (r, c) is given as:

$$\widehat{h}_i^{(r,c)} = \begin{cases} f(x^{(r,c)} W_i^1 + \sum_{j=i}^{M} \left(\mathcal{H}_j^{(r-1,c)} \widehat{\mathcal{U}}_{ij}^1 + \mathcal{H}_j^{(r,c-1)} \widetilde{\mathcal{U}}_{ij}^1 + b_i^1\right)) & \text{if } (r \; mod \; T_i) = 0, \\ \widehat{h}_i^{(r-1,c)} & \text{otherwise.} \end{cases} \tag{2.3}$$

$$\widetilde{h}_i^{(r,c)} = \begin{cases} f(x^{(r,c)} W_i^2 + \sum_{j=i}^{M} \left(\mathcal{H}_j^{(r-1,c)} \widehat{\mathcal{U}}_{ij}^2 + \mathcal{H}_j^{(r,c-1)} \widetilde{\mathcal{U}}_{ij}^2 + b_i^2\right)) & \text{if } (c \; mod \; T_i) = 0, \\ \widetilde{h}_i^{(r,c-1)} & \text{otherwise.} \end{cases} \tag{2.4}$$

Note that the aforementioned method only considers the 4-connected neighborhood, namely, every patch only receives information from its left, right, up and lower adjacent patches. But it is trivial to extend our method to 8-connected neighborhood. Both of the two cases are evaluated in the experiment part.

2.2.1.2 Structured Prediction

Due to the temporal dependency property of spatial CW-RNN, each local patch only receives context information from the region spanned by its predecessors. However, in 2D images, each local patch is surrounded by both its predecessors and postdecessors, thus we want the model to be aware of such bidirectional information from its predecessors and postdecessors at each time step.

To this end, we sweep the input image (or feature map) from four different corners (upper left, lower left, upper right, lower right) to the opposite corners. For each local image patch, activations from four directional sweepings are concatenated together as the full-context representation, which is fed to the successive layers to produce the final prediction output. The illustration of this process is shown in Fig. 2.1.

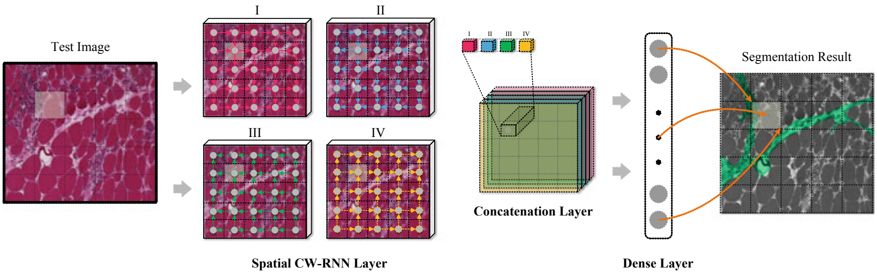


Fig. 2.1 One exemplar architecture. Spatial CW-RNN and dense layer represents the proposed spatial clockwork RNN and fully connected layer, respectively. Sweepings in different directions are illustrated in spatial CW-RNN layer using colorful arrows. Activations from four sweepings are concatenated together in the concatenation layer as the global context information for each local patch. The mapping between the output of dense layer and the predicted mask (overlaid on the original image) for each local patch is illustrated using brown arrows

Now, we omit the module index i in $\mathcal{H}^{(r,c)}$ and define $\mathcal{H}_{\searrow}^{(r,c)}$, $\mathcal{H}_{\swarrow}^{(r,c)}$, $\mathcal{H}_{\nearrow}^{(r,c)}$ and $\mathcal{H}_{\nwarrow}^{(r,c)}$ as the total hidden activations (containing all the modules) for each directional sweeping at time step (r, c). The output $O^{(r,c)}$ after applying one dense layer to those concatenated features can be computed as

$$O^{(r,c)} = f(\sum_{d'} \mathcal{H}_{d'}^{(r,c)} W_{d'} + b), \tag{2.5}$$

where $d' \in \{\searrow, \swarrow, \nearrow, \nwarrow\}$ denotes different sweeping direction. Please note that dense layer is applied individually across all the time step, and local patches corresponding to different time steps share the same weights $W_{d'}$.

Given a set of training data $\{(X_i, Y_i)\}_{i=1}^{N}$, where N is the total number of training data, X_i is the ith training image and Y_i is the corresponding mask label. Let R_i and C_i denote the total number of local patches in row and column dimension for the ith pair of training data. Denote Θ as the model's parameter, and ψ as our model. The objective function defined on $\{(X_i, Y_i)\}$ is given by

$$\mathcal{L}(\psi(X_i; \Theta), Y_i) = \frac{1}{2} \sum_{r=1}^{R_i} \sum_{c=1}^{C_i} \left\| Y_i^{(r,c)} - O_i^{(r,c)} \right\|_2^2, \tag{2.6}$$

where both of $Y_i^{(r,c)}$ and $O_i^{(r,c)}$ are reshaped into a vector to computed the loss.

Our proposed spatial CW-RNN is inherently capable of capturing semantic information in the entire image. Meanwhile, it is totally end-to-end trainable and can be optimized using standard BPTT algorithm [16]. It takes an input image with any size and produces the result mask with the same size as the input.

2.2.1.3 Experimental Results and Discussion

Dataset and Implementation Details: The proposed spatial CW-RNN has been extensively evaluated using 348 H&E-stained skeletal muscle microscopy images (each image roughly contains 300 × 600 pixels). All the images are manually annotated and double checked by two neuromuscular pathologists. In total, 150 images are chosen for testing and the rest for training. Both qualitative and quantitative experiments are reported. Perimysium in skeletal muscle images often exhibits large variation in morphology and image intensity. The inhomogeneous background noise along with the strong similarity with the thin gap among muscle fibers further complicate the segmentation task.

The detailed architecture of our method is summarized in Table.2.1. The first layer is a dense layer, and the next four spatial CW-RNN layers are used to sweep the input feature map in four different directions. The activations of those four spatial CW-RNN layers are then concatenated together as the whole context-aware feature and is then fed to the next consecutive dense layers to get the final output. The *sigmoid* activation function is used in the output layer while the ReLu activation

Table 2.1 The network architecture. D represents the fully connected layer applied individually to every time step. S represents Spatial CW-RNN, where the arrow indicates the sweeping direction. The *Inputs* row specifies the layer ID of each layer's inputs. Layer 7 takes the concatenation of the output from layer 3, 4, 5, and 6 as input

Layer ID	1	2	3	4	5	6	7	8
Layer	Input	Dense	$S_{\searrow}$	$S_{\swarrow}$	$S_{\nearrow}$	$S_{\nwarrow}$	Dense	Dense
Size	300	100	384	384	384	384	100	100
Inputs	–	1	2	2	2	2	[3, 4, 5, 6]	7

function is used in the other layers. The size of the nonoverlapping patches is set to $10 \times 10 \times 3$.

The model is trained using RMSprop optimizer with a learning rate of 0.003. To prevent over-fitting, we set the dropout rate to 0.4 and weight decay to $1e - 7$ across all layers. M and k in Section are 4 and 48, respectively. Time period is set to exponential series: $T_i = 2^{i-1}$.

Evaluation metrics: Denote m_{ij} as the number of pixels of class i labeled as class j, $t_i = \sum_j m_{ij}$ as the number of pixels of class i. We report six commonly used metrics based on variations of region intersection over union (IU) and pixel accuracy that is commonly used in semantic image segmentation and scene parsing evaluation. For the evaluation metrics, we do not distinguish the perimysium region and the background, instead, we treat them uniformly as pixel prediction. For all the comparison methods, we binarize the prediction mask by using a threshold that maximizes the average F_1 score for all the methods.

(1) Mean accuracy (MA): $(1/2) \sum_i m_{ii}/t_i$.
(2) Average IU (AIU): $(1/2) \sum_i (m_{ii}/(t_i + \sum_j m_{ji} - m_{ii}))$.
(3) Weighted IU (WIU) : $(1/\sum_i t_i) \sum_i (t_i m_{ii}/(t_i + \sum_j m_{ji} - m_{ii}))$.
(4) Precision (P), recall (R), and F_1 score.

Comparison with Other Works: We compare our method with several variations of other deep learning based frameworks, e.g., multi-layer perception (MLP), convolutional neural network (CNN). The detailed performance comparison is given in Table 2.2. SCW-RNN(4) and SCW-RNN(8) denote the proposed method for 4- and 8-connected neighborhood, respectively. CNN-nips is the famous architecture utilized in [18] to segment neuronal membranes, which consists of four convolutional layers and four max-pooling layers followed by two fully connected layer. This network uses a large input window size (95 × 95) to capture the context information. We also compare the proposed method with U-NET [19], an end-to-end CNN architecture. To demonstrate the proposed model's capability of handling spatial context information, a plain MLP network that shares similar architecture to our model, denoted as MLP-10 are considered for comparison as well. We also try a larger window size (48 × 48) for MLP network, denoted as MLP-48.

As we show in Table 2.2, both versions of the proposed method, SCW-RNN(4) and SCW-RNN(8), achieve the best overall performance compared with others. It

Table 2.2 Quantitative comparative results of muscle perimysium segmentation results. T represents the average running time (measured in second)

	P	R	F_1 score	MA	AIU	WIU	T
MLP-10	0.768	0.803	0.776	0.883	0.787	0.447	7.36
MLP-48	0.805	0.82	0.804	0.897	0.811	0.453	22.14
U-NET [19]	0.764	0.792	0.761	0.869	0.774	0.442	1.7
CNN-nips [18]	0.834	0.855	0.84	0.916	0.841	0.463	319.8
SCW-RNN(4)	0.854	0.843	0.842	0.909	0.842	0.462	2.6
SCW-RNN(8)	0.836	0.866	0.845	0.918	0.844	0.462	3.6

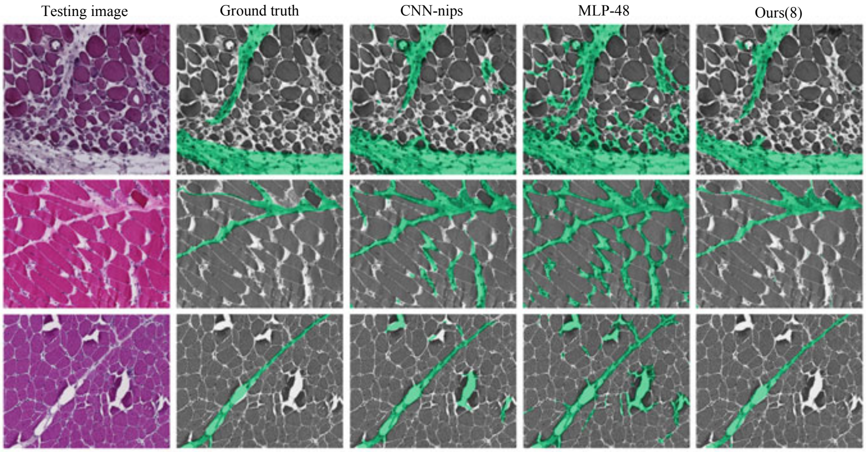


Fig. 2.2 Perymisum segmentation results on three challenging skeleton muscle images, which show strong global structure and demonstrates a lot of appearance similarity between perimysium (true positive) and endo/epimysium (false positive). Comparing with other methods, our results show much better global consistency because it can capture global spatial configurations

is obvious that the utilization of more spatial context information in SCW-RNN(8) leads to performance improvement than SCW-RNN(4), especially in terms of recall and F_1 score. MLP-10, which does not consider such spatial context information across local patches, produces a lot of false positive evidenced by the low precision and F_1 score. MLP-48, which has a larger receptive field outperforms MLP-10 with a large margin. CNN-nips, which uses a really large window size (95 × 95), achieves comparative results as ours, but its running time is almost 100 times slower than our method. Although for certain architecture, fast scanning can be utilized to remove redundant computations of convolution operation, it is not applicable to our case, which conducts patch-wise normalization. U-NET [19], which does not invoke patch-based testing, is also very efficient, but it produces a much lower F_1 score and AIU than ours, one of the possible reasons is that we do not apply aggressive data augmentation in all of our experimental settings.

For quantitative comparison, some challenging images with segmentation results overlaid on the original image are shown in Fig. 2.2. It can be observed that our method produces the most accurate results with much better global consistency. This further provides evidences that the spatial CW-RNN has strong capability to learn the global context information, which is the key to differentiate perimysium from endomysium, epimysium, and blood vessels.

2.3 WSI Inflammatory Muscle Disease Subtype Classification

In this section, we will discuss a fully automatic whole-slide image analysis framework for IM subtype classification using deep convolutional neural networks (DCNNs). The proposed framework consists of several key steps: (1) Automatic localization of diagnose-relevant image regions of interest (ROIs) using prior knowledge; (2) Image region classification with a DCNN and whole-slide image classification with a bag of words (BoW) algorithm, which aggregates patch level predictions into whole-slide image-level classification; (3) A two-level result visualization mechanism that can help interpret the diagnosis procedure and establish appropriate confidence of decision making. An overview of the proposed framework is shown in Fig. 2.3.

2.3.1 *Methodology*

In this section, we begin with introducing the automatic muscle image ROI generation algorithm via inflammatory cell detection, and then present a deep learning framework for ROI classification and class-discriminative image region visualization. Finally, we discuss how to compute a WSI image level representation via aggregating ROI level features for image classification.

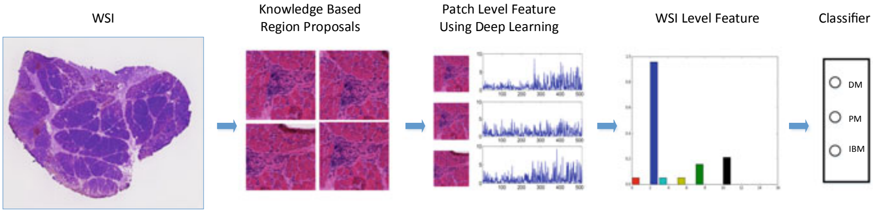

WSI
Knowledge Based Region Proposals
Patch Level Feature Using Deep Learning
WSI Level Feature
Classifier
DM
PM
IBM

Fig. 2.3 An overview of the proposed WSI muscle disease classification framework

2.3.1.1 Inflammatory Cell Detection via Regression

In muscle image analysis, ROIs are the image regions that contain dense inflammatory cells, it is important to achieve accurate and robust inflammatory cell detections.

Denote $\mathcal{T} = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$ the training data, where x represents a training image and $y = h * G$ is the corresponding cell center probability map, where $h(i, j) = 1$ if there is a human cell center annotation at pixel location (i, j), otherwise $h(i, j) = 0$. G denotes a Gaussian kernel with standard deviation σ. Let o denote the output, and then the loss function can be defined as

$$\mathcal{L}(y, o) = \frac{1}{2}\sum_{i=1}^{h}\sum_{j=1}^{w}(y^{ij} - o^{ij})^2, \tag{2.7}$$

where h and w denotes the height and width of x, respectively. Please refer to [20] for more details of the architecture.

2.3.1.2 Inflammatory Cell Detection on Whole Slide Image

In the testing stage, the trained FCN model is applied on WSI images using a two-level sliding window method. In level I, the WSI is divided into 5,000 × 5,000 nonoverlapping slide tiles, and each time one slide tile is loaded into memory for processing; In level II, each $5,000 \times 5,000$ slide tile is further divided into 500 × 500 nonoverlapping image patch tiles, and the image patch tiles are passed to the trained FCN model in a batch mode with a batch size of 20. Both training and testing are conducted in a machine equipped with an Intel i7 processor and a single Tesla K40c GPU. It takes 0.5 s with CPU to load a 5,000 × 5,000 slide tile into memory and 4 s with GPU to process 20 image patch tiles of size 500 × 500 in batch mode. in total, it takes around 80 s for inflammatory cell detection on a 10,000 × 10,000 WSI image.

For WSI image x, let o denote the inflammatory detection result. For ROI proposal generation, o is divided into a set of 128 × 128 nonoverlapping tiles, and the weight for ith tile is calculated as

$$w_i = \sum_{j \in p_i} o_j, \tag{2.8}$$

where p_i denotes all pixel coordinates inside grid tile i, and o_j denotes the FCN output value at pixel coordinate j. All image tiles are ranked by the calculated inflammatory cell density weight w in a nonincreasing order, and the top-k tiles are used to select ROIs for diagnosis analysis. For each selected tile, a 1024 × 1024 image patch around the tile center is cropped as a ROI for following diagnosis.

2.3.1.3 ROI Classification

For each WSI image, the selected ROI patches are assigned the same diagnosis label, i.e., DM, PM, or IBM. In this work, we classify muscle image patches using the ResNet [9] architecture. There are three reasons for our choice: (1) The short connections in ResNet make the network easy to train, which is essential for medical image classification since the available labeled data is scarce; (2) ResNet can achieve competitive accuracy as other deep learning architectures with fewer parameters; (3) The last layer of ResNet is a global average pooling layer (GAP) which provides a direct link between the last convolutional layer and final classification units. The GAP layers have been used to localize discriminative image regions for classification [21].

2.3.1.4 Discriminative Image Region Localization

In this framework, the GAP layer in ResNet is used to localize the discriminative image regions. Denote $\mathbf{F} \in \mathbf{R}^{n\times k\times k}$ the activations of the last convolutional map, and $\mathbf{W} \in \mathbf{R}^{C\times n}$ the weights of output layer. For class c, the activation map $M_c \in \mathbf{R}^{k\times k}$ can computed as

$$M_c(x, y) = f(\sum_{i=1}^{n} \mathbf{W}_{c,i} * \mathbf{F}_i(x, y)), \tag{2.9}$$

where $f(\cdot)$ is ReLU function [5].

Several examples of generated class activation mappings (CAMs) on testing patches can be seen in Fig. 2.4. As we can see, the network puts more weight on regions with dense inflammatory cells, and this is expected since the main difference of different muscle disease categories lies in the microenvironment of these regions. The visualization can help users establish confidence in the trained model since their results are interpretable.

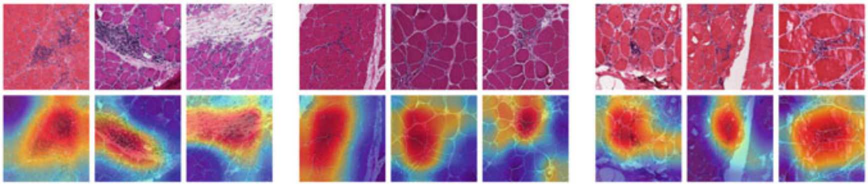

(a) DM (b) PM (c) IBM

Fig. 2.4 Examples of the generated CAMs for several sample testing patches of three classes

2.3.1.5 WSI Image Classification

Given a WSI image S, and a set of D-dimensional ROI descriptors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ from S, we aim to aggregate individual ROI representations $\mathbf{X}$ into a single feature vector, $\mathbf{f}$, as S's representation. One of the most simple yet effective local descriptor aggregating methods is the BoW model [22]. Given a learned codebook with M entries, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M] \in \mathbb{R}^{D\times M}$, $\mathbf{f} \in \mathbf{M}$ is computed via assigning each ROI descriptor $\mathbf{x}_i$ to the closest dictionary entry $\mathbf{B}_j$. The final feature representation is computed using ℓ_2 normalization, $\mathbf{f} = \mathbf{f}/\|\mathbf{f}\|_2$.

2.4 Experimental Results

2.4.1 Dataset

We validate the proposed algorithm on an H&E-stained muscle whole-slide image dataset, which comprises of 74 individuals (36 DM, 24 PM, and 14 IBM). The whole slide images, which are captured at a 40X objective, and the diagnosis labels are prepared by the Medical College of Wisconsin Neuromuscular Laboratory (MCWNL). A threefold cross-validation is used through the experiments. In each time, twofold are used for training with the rest for testing. All reported results are the average of the testing accuracies over threefolds.

2.4.2 Implementation Details

Patch level: The top 30 regions that contain the most dense inflammatory cells are cropped from each WSI as ROIs. The 1,024 × 1,024 patches are resized to 256 × 256 for deep model training and testing. During training, two data argumentation strategies are used: (1) Randomly cropped 224 × 224 patches from the original 256 × 256 patch and (2) Horizontal flipping with a probability of 0.5. The optimization is driven by stochastic gradient descent with momentum 0.95, weight decay 0.005, batch size 16. The initial learning rate is set as $1e{-}4$, and decreases by a ratio of 0.1 for every 10 epochs. The training is stopped after 60 epochs. Here, an 18-layer ResNet is used and the detailed implementation can be found in the open-source code.[1]

While it is common to use the activations of the last hidden layer in CNN models as feature representations, the deep learning predictions (softmax outputs) are found to be effective as patch level representations in [23]. Here both features are studied, and the t-SNE visualization [24] of both types of features on the testing set are

[1] https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py.

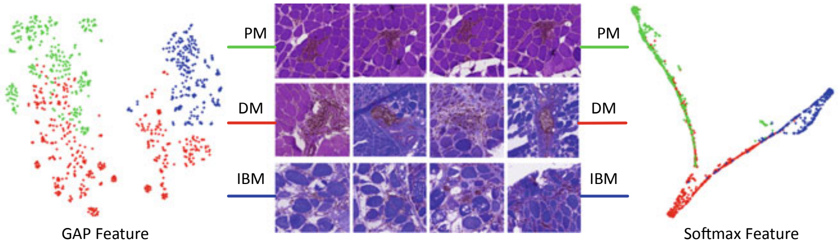


Fig. 2.5 t-SNE visualization of GAP and softmax layer representations in the CNN model for three muscle disease classes

shown in Fig. 2.5. We can see that: (1) Most ROI patches on testing set are separable regarding different muscle disease classes in both feature spaces; (2) ROI patches of DM class exhibit more diversity. In this work, the softmax outputs are used as feature representations due to its more concise clustering patterns, as shown in Fig. 2.5.

WSI image level: The BoW method is used to aggregate ROI level features, which are the predictions of the trained deep learning model. For BoW aggregating, the dictionary **B** is constructed using k-means clustering algorithm on training sets, with k set as 16. For WSI image classification, a logistic regression model is trained. Since the result of k-means clustering varies with different initializations, all experiments are repeated for 20 times and the average result is reported for robust analysis.

2.4.3 Evaluation of Different WSI Frameworks

In this section, we compare the proposed WSI analysis framework to two other existing frameworks:

- Training with all image tiles (WSI FM-I). In this framework, the WSI images are divided into a set of nonoverlapping image tiles with size 1024 × 1024. All image tiles from one WSI are assigned the same diagnose label (DM, PM, or IBM). The activations of the GAP layer are used as the image tile's representations.
- Training with sampling (WSI FM-II). This is proposed in [23] to train CNN with noisy labels. At the beginning of CNN training, all image tiles are used. Then after every five epochs, the training data is regenerated. For each WSI image, 30 image tiles are selected using weighted reservoir sampling, where the weight is calculated as the classification score of the current model. Other configurations are the same as those in WSI FM-I.

A detailed comparison of muscle disease classification accuracies using different WSI frameworks is provided in Fig. 2.6. Note that, for each method, classification results using both GAP and Softmax output features are reported for comparison. We can observe that: (1) The proposed WSI analysis framework significantly outperforms existing WSI methods on muscle disease classification. This also proves that the

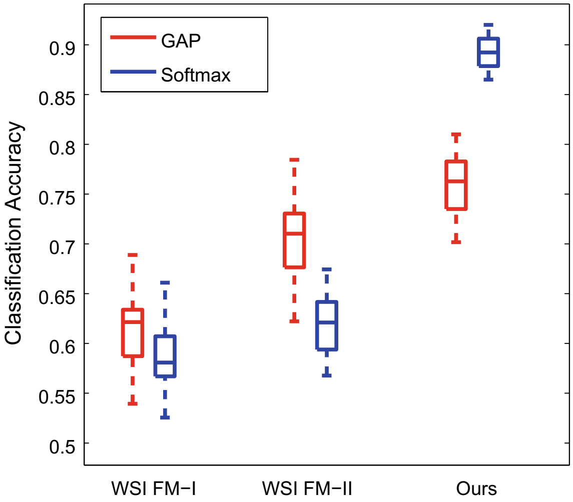


Fig. 2.6 Comparison of muscle disease classification accuracies using different WSI frameworks

Table 2.3 Classification accuracies of different whole-slide image analysis frameworks

WSI Methods	Aggregating	Class. Acc.
WSI FM-I	MV	0.6339
	BoW (GAP feat.)	0.6889
WSI FM-II [23]	MV	0.6739
	BoW (GAP feat.)	0.7844
Ours	MV	0.8650
	BoW (Softmax feat.)	**0.9200**

proposed ROI generation algorithm is effective in finding diagnostically relevant image regions. (2) For WSI FM-I and WSI FM-II, GAP feature produces better WSI classification result; For our method, softmax output feature delivers higher WSI classification accuracy. It is known that CNN features in deeper layers have more discriminative power than those in previous layers. However, in WSI FM-I and FM-II, a large portion of local image patches do not contain diagnostic relevant information, and these noises might have misled the CNN training and the softmax features become less indicative for classification in such cases.

Another simple yet effective way for WSI level classification is to use majority voting of ROI patches. The WSI classification accuracies of both majority voting and BoW are listed in Table 2.3. For BoW, only the best results are reported. Compared with existing WSI analysis methods, our framework using either majority voting or BoW provides significantly better WSI classification performance.

Table 2.4 Comparison of different training methods using three deep learning architectures: VGG, Inception, and ResNet

Architecture	Training method	MV	BoW	
			Mean	Std
VGG [7]	Scratch	0.5389	0.5747	0.0296
	Finetune Class	0.8378	0.8045	0.0142
	Finetune All	0.7833	0.8443	0.0178
Inception [8]	Scratch	0.6339	0.5640	0.0295
	Finetune Class	0.7422	0.7305	0.0242
	Finetune All	0.8517	0.8179	0.0249
ResNet [9]	Scratch	0.6467	0.5715	0.0329
	Finetune Class	0.8372	0.8075	0.0214
	Finetune All	0.8650	0.8903	0.0210

2.4.4 Evaluation of Different Training Methods

In our implementation, the ROI classification model is fine-tuned from a pretrained ResNet model[2] on our muscle dataset. For comparative analysis, two other training methods are also investigated: (1) Training from scratch and (2) Freezing all model parameters in the feature layers and only fine-tuning those in classification layer(s). To this end, we also test these different training methods using two other popular deep learning architectures: VGG [7] and GoogleNet Inception V3 [8]. The detailed comparison is illustrated in Table 2.4. Note that in Table 2.4, classification results of both majority voting and BoW are reported. For BoW, we report the mean classification accuracy and the standard deviations.

From Table 2.4, we can see that: (1) Fine-tuning-based methods achieve much better classification results than training from scratch, and fine-tuning all layers gives better results than only fine-tuning the classification layer(s); and (2) The 18-layer ResNet (ResNet-18) used in our implementation delivers better classification accuracy than VGG and GoogleNet Inception V3, which have 2 and 10 times more parameters than ResNet-18, respectively.

2.4.5 Evaluation of Different Number of ROIs

In this section, we investigate the effects of the number of ROIs on WSI image classification. Here, we repeat above experiments using three different number of ROIs: 10, 20, and 30. The detailed comparative results are listed in Table 2.5. For different numbers of ROIs, both majority voting and BoW produce satisfying classification results on three different deep learning architectures; (2) the ResNet model works better than VGG and Inception models.

[2] https://download.pytorch.org/models/resnet18-5c106cde.pth.

Table 2.5 Comparison of WSI classification accuracies using different number of ROIs on three deep learning architectures

Architecture	# of ROIs	MV	BoW	
			Mean	Std
VGG [7]	10	0.8917	0.8360	0.0168
	20	0.8244	0.8506	0.0193
	30	0.7833	0.8443	0.0178
Inception [8]	10	0.8517	0.7857	0.0181
	20	0.8111	0.8292	0.0204
	30	0.8517	0.8179	0.0249
ResNet [9]	10	0.9194	0.8895	0.0166
	20	0.8644	0.8377	0.0180
	30	0.8650	0.8903	0.0210

2.4.6 *Diagnosis Interpretation and Visualization*

In our framework, the diagnosis results can be properly visualized such that the insight of the working system can be interpreted. To this end, we present a two-level visualization mechanism: Level (I) The ROIs that the proposed system uses to make decisions; Level (II) The discriminative image regions in each ROI responsible for ROI classification.

2.4.6.1 Prediction Explanation

Currently, most deep learning methods work in a *black box* fashion in medical applications, which potentially have both high *impacts* and *risks*. Proper visualization will help pathologists or medical doctors better understand the working principle of the automatic system, with providing visual evidence. One example can be seen in Fig. 2.7. where the proposed system makes the correct prediction of a DM WSI. On the left, the ROIs used are overlayed on the original whole-slide image; On the right, the discriminative image regions within each ROI are highlighted. As we can see, the proposed system can achieve decent transparency while making diagnostic predictions.

2.4.6.2 Understand Failure Cases

For medical application researchers, proper visualization can also help identify and understand failure cases. Figure 2.8 shows that a DM WSI is falsely predicted as PM by the proposed system. By checking the WSI carefully, we find that this is a very difficult case which has mixed clinic features of both DM and PM. For these cases, a medical doctor might require more information to make a treatment plan [3].

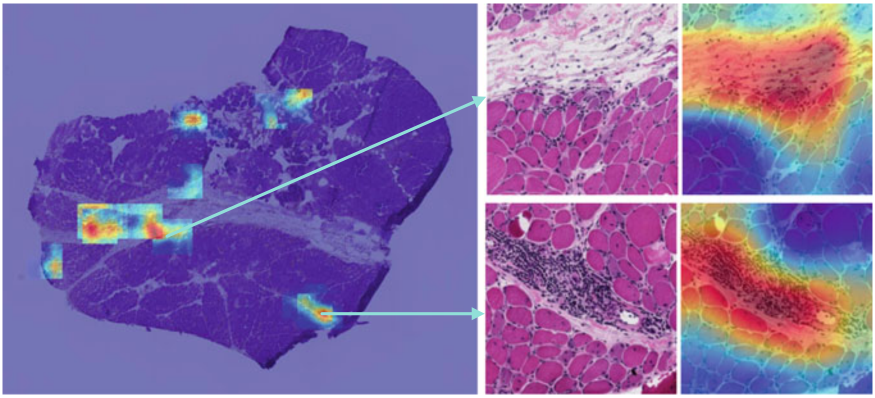


Fig. 2.7 An example of the two-level visualization mechanism proposed in our framework. The proposed system makes the correct prediction of a DM WSI. Left: The ROIs the proposed system uses to make predictions; Right: Sample results of the discriminative image regions in each ROI responsible for ROI classifications

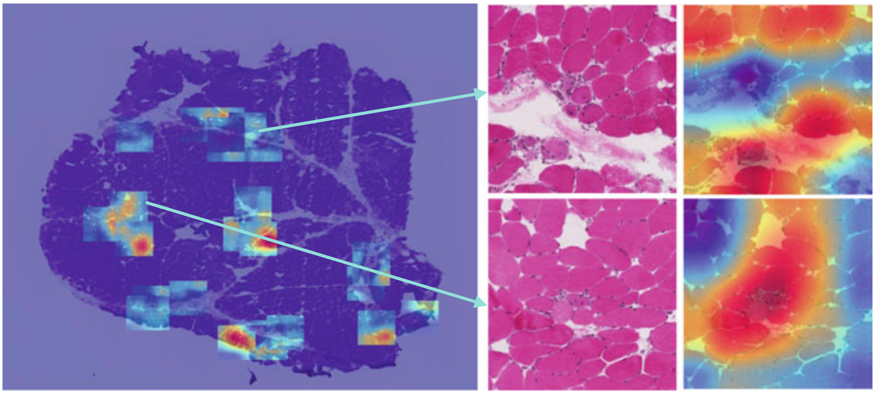


Fig. 2.8 An example of the two-level visualization mechanism proposed in our framework. In this example, the proposed system predicts a DM WSI into PM. Left: The ROIs the proposed system use to make predictions; Right: Sample results of the discriminative image regions in each ROI responsible for ROI classifications

We argue that a proper visualization or interpretation mechanism should be one of the core components of a computer-aided diagnosis (CAD) system. The idea presented in our framework can be generalized to many other medical applications.

2.5 Summary

In this chapter, we talk about the application of deep learning in muscle perimysium segmentation and whole-slide image inflammatory muscle disease classification. First of all, we introduce a formulation of the novel 2D spatial clockwork recurrent neural network and pave the way to utilize RNN architecture to process 2D biomedical image data. The proposed spatial CW-RNN is totally end-to-end trainable and capable of encoding the global context information into the features of each local image patch, which tremendously improves the performance.

Second, we also present a WSI inflammatory muscle disease classification framework. The proposed method can be directly applied on whole-slide images, and does not require manual ROI annotations from pathologists. Extensive experiments demonstrate that the proposed system can provide robust and accurate WSI diagnosis predictions. Furthermore, we show that the diagnosis results of the proposed machine learning system can be properly visualized and are interpretable. The visualization can not only help the intelligent system researchers identify and understand the failure cases, but can also establish appropriate confidence in diagnosis. We advocate that this type of machine learning systems that can present visual evidences along with the interpretable diagnosis should receive more research interest in the future.

References

1. Dalakas MC, Hohlfeld R (2003) Polymyositis and dermatomyositis 362:971–982
2. Dalakas MC (2002) Muscle biopsy findings in inflammatory myopathies. Rheum Dis Clin N Am 28(4):779–798
3. Dalakas MC (2015) Inflammatory muscle diseases. N Engl J Med 372(18):1734–1747
4. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
5. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
6. Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35(8):1915–1929
7. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
8. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
10. Visin F, Kastner K, Courville AC, Bengio Y, Matteucci M, Cho K (2015) Reseg: a recurrent neural network for object segmentation, vol 1. arXiv:1511.07053
11. Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches, vol 1. arXiv:1409.1259
12. Graves A, Fernández S, Schmidhuber J (2007) Multi-dimensional recurrent neural networks. In: 17th International Conference on Artificial Neural Networks, pp 549–558
13. Byeon W, Breuel TM, Raue F, Liwicki M (2015) Scene labeling with LSTM recurrent neural networks. In: IEEE conference on computer vision and pattern recognition, pp 3547–3555
14. Koutník J, Greff K, Gomez F, Schmidhuber J (2014) A clockwork RNN. In: Proceedings of the 31st international conference on machine learning, vol 32, pp 1863–1871
15. Xie Y, Xing F, Kong X, Su H, Yang L (2015) Beyond classification: Structured regression for robust cell detection using convolutional neural network. In: MICCAI, vol 9351, pp 358–365
16. Werbos PJ (1990) Backpropagation through time: what it does and how to do it. Proc IEEE 78(10):1550–1560

17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Ciresan D, Giusti A, Gambardella LM, Schmidhuber J (2012) Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in neural information processing systems, pp 2843–2851
19. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: MICCAI, pp 234–241
20. Xie W, Noble JA, Zisserman A (2015) Microscopy cell counting with fully convolutional regression networks. In: MICCAI 1st workshop on deep learning in medical image analysis
21. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
22. Sivic J, Zisserman A et al (2003) Video google: a text retrieval approach to object matching in videos
23. Hou L, Samaras D, Kurc TM, Gao Y, Davis JE, Saltz JH (2016) Patch-based convolutional neural network for whole slide tissue image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2424–2433
24. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(Nov):2579–2605

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

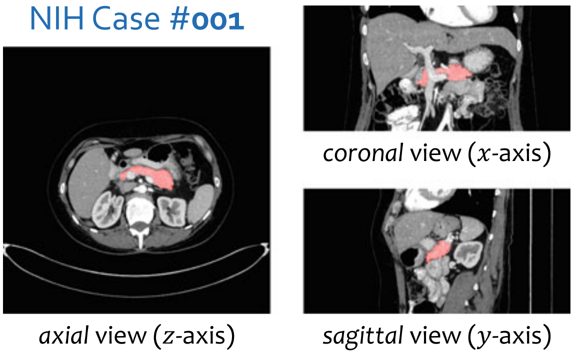


Fig. 3.1 A typical example from the NIH *pancreas* segmentation dataset [35] (best viewed in color). We highlight the *pancreas* in red seen from three different viewpoints. It is a relatively small organ with irregular shape and boundary

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): $\mathrm{DSC}(\mathcal{Y}, \mathcal{Z}) = \frac{2\times|\mathcal{Y}\cap\mathcal{Z}|}{|\mathcal{Y}|+|\mathcal{Z}|}$. 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 $\mathcal{B}$, $\mathrm{DSC}(\mathcal{A}, \mathcal{B}) = \frac{2\times|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}|+|\mathcal{B}|}$, 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: $\frac{\partial\mathcal{L}(\mathbf{Z},\mathbf{Y})}{\partial z_j} = -2 \times \frac{y_j\left(\sum_i z_i + \sum_i y_i\right) - \sum_i z_i y_i}{\left(\sum_i z_i + \sum_i y_i\right)^2}$.

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 $\mathbf{X}_{\mathrm{C},w}$ $(w = 1, 2, \ldots, W)$, $\mathbf{X}_{\mathrm{S},h}$ $(h = 1, 2, \ldots, H)$ and $\mathbf{X}_{\mathrm{A},l}$ $(l = 1, 2, \ldots, L)$, where the subscripts C, S and A stand for *coronal*, *sagittal* and *axial*, respectively. On each axis, an individual 2D-FCN [29] on a 16-layer VGGNet [39] is trained We train three 2D-FCN models $\mathbb{M}_{\mathrm{C}}$, $\mathbb{M}_{\mathrm{S}}$ and $\mathbb{M}_{\mathrm{A}}$ 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.

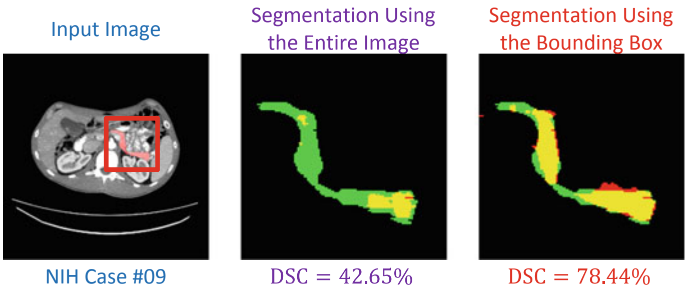

Input Image
Segmentation Using the Entire Image
Segmentation Using the Bounding Box
NIH Case #09
DSC = 42.65%
DSC = 78.44%

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.

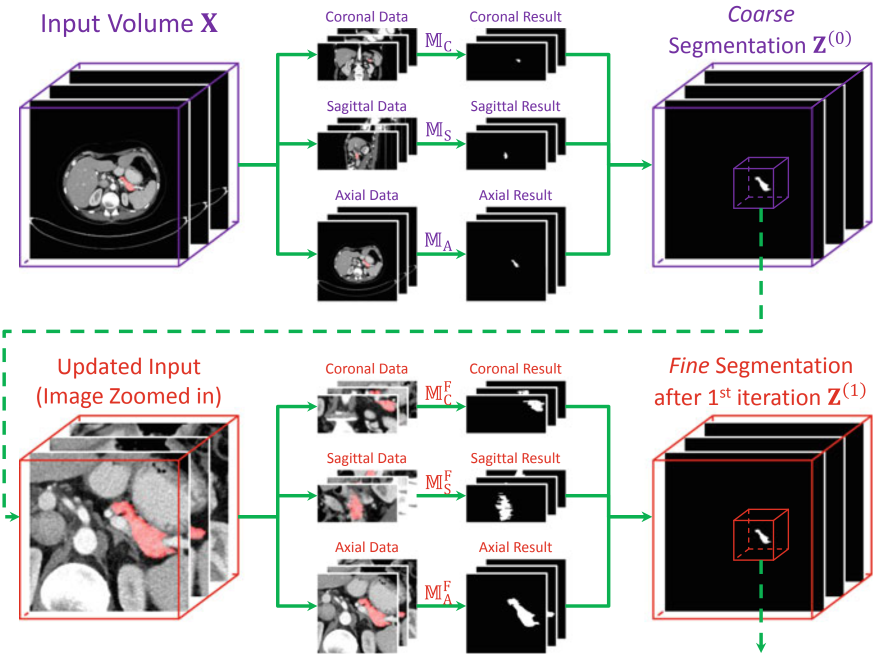


Fig. 3.3 Illustration of the testing process (best viewed in color). Only one iteration is shown here. In practice, there are at most 10 iterations

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 $\mathcal{C}^\star = \left\{i \mid C_i^\star = 1\right\}$. Frequently, a large fraction of $\mathcal{C}^\star$ falls within $\mathcal{P}^\star$ (e.g., $|\mathcal{P}^\star \cap \mathcal{C}^\star| \,/\, |\mathcal{C}^\star| > 95\%$ 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

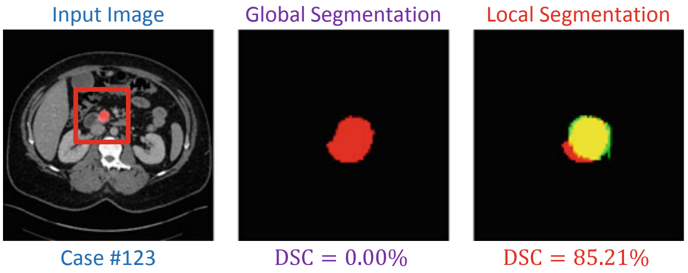


Fig. 3.4 A relatively difficult case in pancreatic cyst segmentation and the results produced by different input regions, namely using the entire image and the region around the ground-truth pancreas mask (best viewed in color). The cystic, predicted and overlapping regions are marked by red, green, 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.

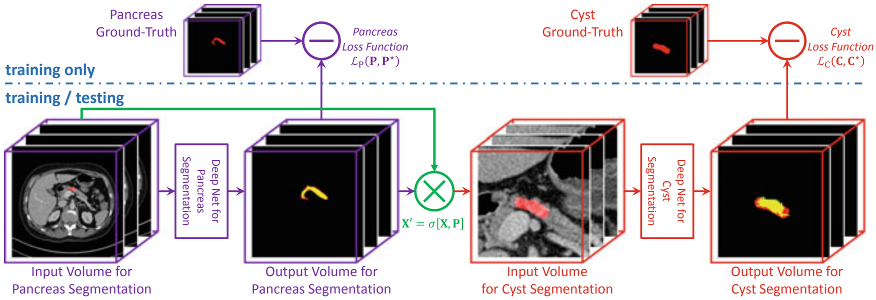

Pancreas Ground-Truth
Pancreas Loss Function $\mathcal{L}_\mathrm{P}(\mathbf{P},\mathbf{P}^\star)$
Cyst Ground-Truth
Cyst Loss Function $\mathcal{L}_\mathrm{C}(\mathbf{C},\mathbf{C}^\star)$
training only
training / testing
Deep Net for Pancreas Segmentation
$\mathbf{X}' = \sigma[\mathbf{X},\mathbf{P}]$
Deep Net for Cyst Segmentation
Input Volume for Pancreas Segmentation
Output Volume for Pancreas Segmentation
Input Volume for Cyst Segmentation
Output Volume for Cyst 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

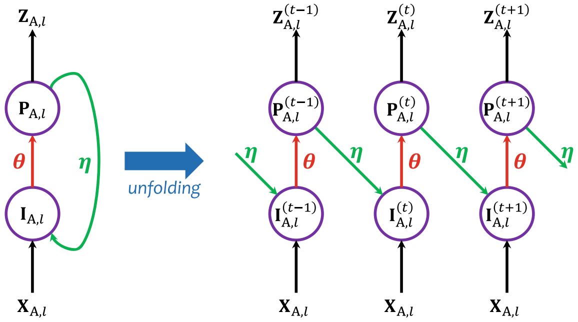

$\mathbf{Z}_{\mathrm{A},l}$
$\mathbf{P}_{\mathrm{A},l}$
$\boldsymbol{\theta}$
$\boldsymbol{\eta}$
$\mathbf{I}_{\mathrm{A},l}$
$\mathbf{X}_{\mathrm{A},l}$
unfolding
$\mathbf{Z}_{\mathrm{A},l}^{(t-1)}$
$\mathbf{Z}_{\mathrm{A},l}^{(t)}$
$\mathbf{Z}_{\mathrm{A},l}^{(t+1)}$
$\mathbf{P}_{\mathrm{A},l}^{(t-1)}$
$\mathbf{P}_{\mathrm{A},l}^{(t)}$
$\mathbf{P}_{\mathrm{A},l}^{(t+1)}$
$\mathbf{I}_{\mathrm{A},l}^{(t-1)}$
$\mathbf{I}_{\mathrm{A},l}^{(t)}$
$\mathbf{I}_{\mathrm{A},l}^{(t+1)}$
$\mathbf{X}_{\mathrm{A},l}$

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 $\mathrm{DSC}\left\{\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\right\} \geqslant$ thr
  return $\mathbf{Z} \leftarrow \mathbf{Z}^{(t)}$.
```

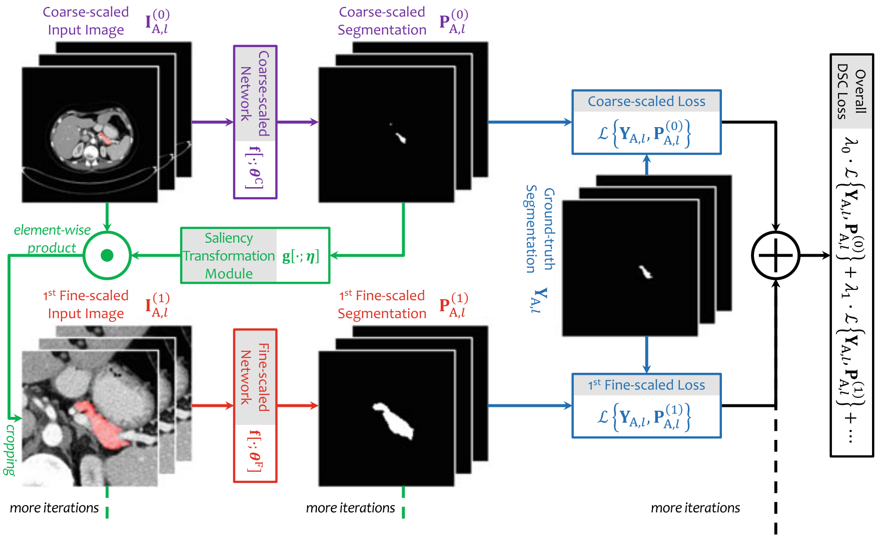

Coarse-scaled Input Image $\mathbf{I}_{A,l}^{(0)}$
Coarse-scaled Network $\mathbf{f}[\cdot;\boldsymbol{\theta}^{C}]$
Coarse-scaled Segmentation $\mathbf{P}_{A,l}^{(0)}$
Coarse-scaled Loss $\mathcal{L}\{\mathbf{Y}_{A,l},\mathbf{P}_{A,l}^{(0)}\}$
element-wise product
Saliency Transformation Module $\mathbf{g}[\cdot;\boldsymbol{\eta}]$
Ground-truth Segmentation $\mathbf{Y}_{A,l}$
1st Fine-scaled Input Image $\mathbf{I}_{A,l}^{(1)}$
Fine-scaled Network $\mathbf{f}[\cdot;\boldsymbol{\theta}^{F}]$
1st Fine-scaled Segmentation $\mathbf{P}_{A,l}^{(1)}$
1st Fine-scaled Loss $\mathcal{L}\{\mathbf{Y}_{A,l},\mathbf{P}_{A,l}^{(1)}\}$
cropping
more iterations
more iterations
more iterations
Overall DSC Loss
$\lambda_0\cdot\mathcal{L}\{\mathbf{Y}_{A,l},\mathbf{P}_{A,l}^{(0)}\}+\lambda_1\cdot\mathcal{L}\{\mathbf{Y}_{A,l},\mathbf{P}_{A,l}^{(1)}\}+\ldots$

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 $\mathbf{Y}_{A,l}$ is not available, the probability map $\mathbf{P}_{A,l}^{(t)}$ 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

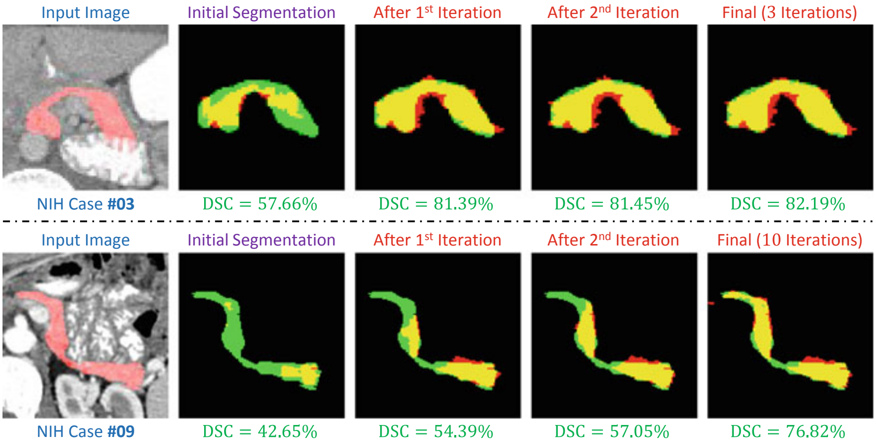


Fig. 3.8 Examples of segmentation results throughout the iteration process (best viewed in color). We only show a small region covering the pancreas in the axial view. The terminating condition is $d^{(t)} \geqslant 0.95$. Red, green and yellow indicate the prediction, ground-truth and overlapped regions,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

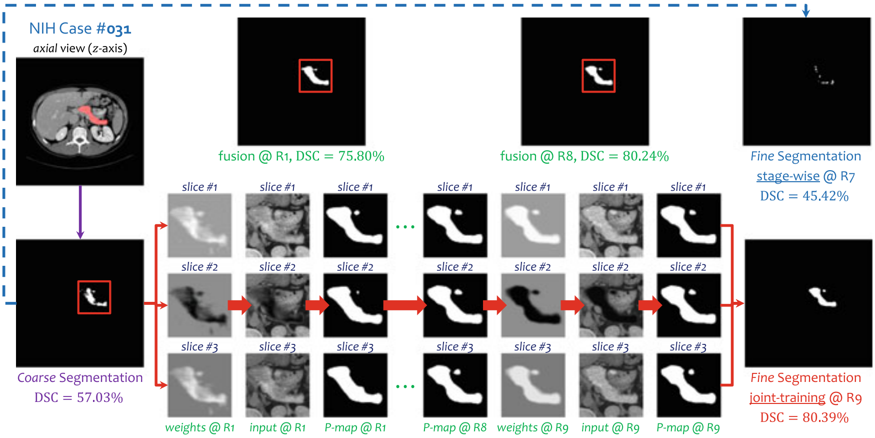


Fig. 3.9 Visualization of how recurrent saliency transformation works in coarse-to-fine segmentation (best viewed in color). Segmentation accuracy is largely improved by making use of the probability map from the previous iteration to help the current iteration. Note that three weight maps capture different 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: $d^{(t)} = \mathrm{DSC}\left\{\mathbf{Z}^{(t-1)}, \mathbf{Z}^{(t)}\right\} = \frac{2\times\sum_i Z_i^{(t-1)} Z_i^{(t)}}{\sum_i Z_i^{(t-1)} + Z_i^{(t)}}$.

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 ± 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

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

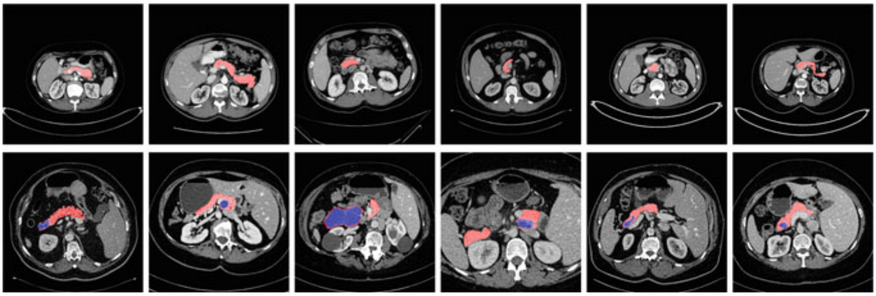


Fig. 4.1 An illustration of normal pancreases on NIH dataset [31] and abnormal cystic pancreases on JHMI dataset [45] shown in the first and second rows, respectively. Normal pancreas regions are masked as red and abnormal pancreas regions are marked as blue. The pancreas usually occupies a small region 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 voxelwise 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 $\mathbb{M}^{\mathrm{C}} : \mathbf{P}^{\mathrm{C}} = \mathbf{f}^{\mathrm{C}}\left(\mathbf{X}^{\mathrm{C}}; \boldsymbol{\Theta}^{\mathrm{C}}\right)$ 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 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 $\mathbb{M}^{\mathrm{F}} : \mathbf{P}^{\mathrm{F}} = \mathbf{f}^{\mathrm{F}}\big(\mathbf{X}^{\mathrm{F}}; \boldsymbol{\Theta}^{\mathrm{F}}\big)$, 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.

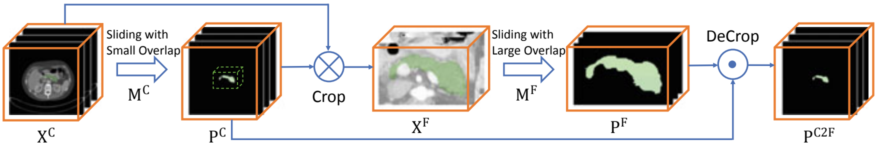


Fig. 4.2 Flowchart of the proposed 3D coarse-to-fine segmentation system in the testing phase. We first apply "ResDSN Coarse" with a small overlapped sliding window to obtain a rough pancreas region and then use the "ResDSN Fine" model to refine the results with a large overlapped sliding window. Best 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 × 3 × 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 × 2 × 2 and strides of two to reduce

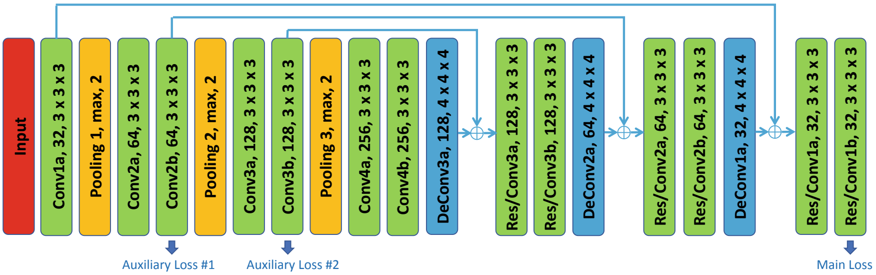


Fig. 4.3 Illustration of our 3D convolutional neural network for volumetric segmentation. The encoder path is the path between "Conv1a" and "Conv4b" while the decoder path is the one between "DeConv3a" and "Res/Conv1b". Each convolution or deconvolution layer consists of one convolution followed by a BatchNorm and a ReLU. To clarify, "Conv1a, 32, 3 × 3 × 3" means the convolution operation with 32 channels and a kernel size of 3 × 3 × 3. "Pooling 1, max, 2" means the max-pooling operation with kernel size of 2 × 2 × 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

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

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

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

  where $\operatorname{Clip}_{\mathbf{X}}^{\epsilon}$ 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 Coefficient (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 $d = 1.0$ 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

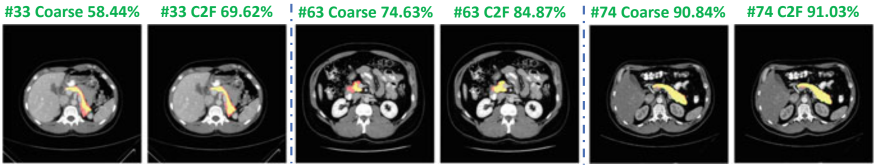


Fig. 4.4 Examples of segmentation results reported by "ResDSN Coarse" and "ResDSN C2F" on a same slice in the axial view from NIH case #33, #63 and #74, respectively. Numbers after "Coarse" or "C2F" mean testing DSC. Red, green, and yellow indicates the ground truth, prediction, and overlapped regions, 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 $\nabla_{\mathbf{X}}\mathcal{L}(\mathbf{X}; \mathbf{\Theta})$ 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 $\mathcal{L}(\mathbf{X_{adv}}; \mathbf{\Theta})$ 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

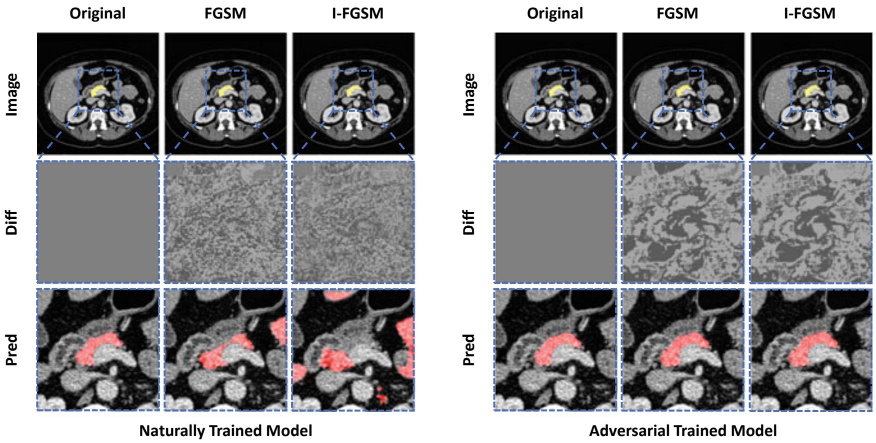


Fig. 4.5 Qualitative comparison of adversarial examples and their effects on model predictions. Note that the added perturbation is effectively imperceptible to the human eye, and the difference between the original image and the adversarial image has been magnified by 5 × (values shifted by 128) for 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

Chapter 5
Unsupervised Domain Adaptation of ConvNets for Medical Image Segmentation via Adversarial Learning

Qi Dou, Cheng Chen, Cheng Ouyang, Hao Chen and Pheng Ann Heng

Abstract Deep convolutional networks (ConvNets) have achieved the state-of-the-art performance and become the de facto standard for solving a wide variety of medical image analysis tasks. However, the learned models tend to present degraded performance when being applied to a new target domain, which is different from the source domain where the model is trained on. This chapter presents unsupervised domain adaptation methods using adversarial learning, to generalize the ConvNets for medical image segmentation tasks. Specifically, we present solutions from two different perspectives, i.e., feature-level adaptation and pixel-level adaptation. The first is to utilize feature alignment in latent space, and has been applied to cross-modality (MRI/CT) cardiac image segmentation. The second is to use image-to-image transformation in appearance space, and has been applied to cross-cohort X-ray images for lung segmentation. Experimental results have validated the effectiveness of these unsupervised domain adaptation methods with promising performance on the challenging task.

Q. Dou (✉) · C. Chen · H. Chen · P. A. Heng
Department of Computer Science and Engineering, CUHK, Hong Kong, Hong Kong
e-mail: dqcarren@gmail.com

C. Chen
e-mail: cchen@cse.cuhk.edu.hk

H. Chen
e-mail: hchen@cse.cuhk.edu.hk

P. A. Heng
e-mail: pheng@cse.cuhk.edu.hk

C. Ouyang
Department of Electrical Engineering and Computer Science, University of Michigan, Michigan, USA
e-mail: couy@umich.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_5

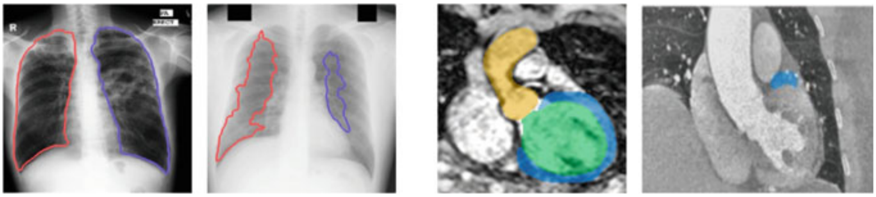
5.1 Introduction

Deep convolutional networks (ConvNets) have made wide success in a variety of automatic medical image analysis tasks, such as anatomical structure segmentation [1, 2], lesion detection [3, 4], cancer diagnosis [5, 6], attributing to the network's learned highly representative features. In typical practice, the deep ConvNets are trained and tested on datasets where all the images come from the same dataset, i.e., samples are drawn from the same data distribution. However, it has been frequently observed that domain shift can bring about performance degradation. The ConvNets tend to present poor results when being applied to new target data, which are acquired using different protocols, scanners, or modalities [7, 8]. It is crucial to close the performance gap, for large-scale study or deployment of deep learning models in real-world clinical practice.

Domain adaptation has been a long-standing topic in machine learning. It is a very common challenge to investigate the generalization capability of the learning systems. In medical imaging, some traditional automatic methods also suffer from similar poor generalization problem. For example, Philipsen et al. [9] have studied the influence of data distribution variations across chest radiography datasets on segmentation methods based on k-nearest neighbor classification and active shape modeling. In recent years, the study of adapting ConvNets have gradually attracted more attention. In the concept of domain adaptation, the domain of labeled training data is termed as *source domain*, and the unseen test data is termed as *target domain*. One straightforward solution is transfer learning, i.e., fine-tuning the ConvNets learned on source domain with extra labeled data from the target domain. Remarkably, Ghafoorian et al. [7] investigated on the number of fine-tuned layers to reduce the required amount of annotations for brain lesion segmentation across MRI datasets. However, the way of supervised transfer learning (STL) still relies on extra labeled data, which is quite expensive or sometimes even infeasible to obtain in the medical field.

Instead, the unsupervised domain adaptation (UDA) methods are more appealing and feasible, since these scenarios transfer knowledge across domains without using additional target domain labels. Generally speaking, existing literatures tackle the unsupervised domain adaptation task based on adversarial learning [10] from two directions: (1) feature-level adaptation with latent space alignment; (2) pixel-level adaptation with image-to-image translation. More specifically, for feature-level adaptation, the source and target inputs are mapped into a shared latent feature space, such that a classifier learned based on this common space can work for both domains. For pixel-level adaptation, the images from target domain are transformed into the appearance of source domain, such that ConvNets trained on source domain can be used for target images, or vice versa. Detailed literatures within these two solution directions are described in the next section.

In this chapter, we focus on demonstrating how to conduct unsupervised domain adaptation of ConvNets on medical image segmentation tasks, with two case studies as illustrated in Fig. 5.1. One is using feature space alignment for adapting ConvNets



(a) Chest X-ray images

(b) Cardiac MRI/CT images

Fig. 5.1 Illustration of performance degradation of deep learning models on medical images. **a** ConvNet trained on source chest X-ray images can perform well on source data (left) but get poor results on unseen target data (right). **b** ConvNet trained on cardiac MRI images (left) receives a complete failure when tested on cardiac CT images (right)

between different modalities of images (i.e., CT and MRI) for cardiac segmentation. The other is employing pixel space transformation for adapting ConvNets between different cohorts of chest X-ray images for lung segmentation. Our works related to this chapter have been published in [11, 12].

5.2 Related Works

Domain adaptation aims to recover the performance degradation caused by any distribution change occurred after learning a classifier. For deep learning models, this situation also applies, and it has been an active and fruitful research topic in recent investigations of deep neural networks. In this section, we review the literatures of unsupervised domain adaptation methods proposed from two different perspectives, i.e., feature-level adaptation and pixel-level adaptation.

5.2.1 Feature-Level Adaptation

One group of prior studies on unsupervised domain adaptation focused on aligning the distributions between domains in the feature space, by minimizing measures of distance between features extracted from the source and target domains. Pioneer works tried to minimize the distance between domain statistics. For example, the maximum mean discrepancy (MMD) was minimized together with a task-specific loss to learn the domain-invariant and semantic-meaningful features in [13, 14]. The correlations of layer activations between the domains were aligned in the study of [15]. Later on, [16] pioneered adversarial feature adaptation where a domain discriminator aims to classify the source and target representations while a feature generator competes with the discriminator to produce domain-invariant features. The [17] introduced a more flexible adversarial learning method with untied weight sharing, which helps effective learning in the presence of larger domain shifts. Recent studies [18, 19] proposed to apply the adversarial learning in other lower dimensional spaces instead of the high-dimensional feature space for more effective feature alignment.

Effectiveness of the adversarial framework for feature adaptation has also been validated in medical applications. Kamnitsas et al. [20] made the earliest attempt to align feature distributions in cross-protocol MRI images with adversarial loss. The adversarial framework was further extended to cross-modality cardiac segmentation in [11, 21]. Most recently, the adversarial loss was combined with a shape prior to improve domain adaptation performance for left atrium segmentation across ultrasound datasets [22]. In [23], the adaptation for whole-slide images was achieved through the adversarial training between domains along with a Siamese architecture on the target domain to add a regularization. Dong et al. [24] discriminated segmentation predictions of the heart on both source and target X-rays from those ground truth masks, based on the assumption that segmentation masks should be domain independent. Zhang et al. [25] proposed multi-view adversarial training for dataset-invariant left and right-ventricular coverage estimation in cardiac MRI.

5.2.2 *Pixel-Level Adaptation*

With the success of generative adversarial networks (GANs) [10] and its powerful extensions such as CycleGAN [26] for producing realistic images, there exists lines of researches performing adaptation in pixel-level through image-to-image transformation. Some methods first trained a ConvNet in source domain, and then transformed the target images into source-like ones, such that the transformed image can be tested using the pretrained source model [12, 27, 28]. Inversely, other methods tried to transform the source images into the appearance of target images [29–31]. The transformed target-like images were then used to train a task model which could perform well in the target domain. For pixel-level adaptation, it is important that the structural contents of original images are well preserved in the generated images. For example, Shrivastava et al. [29] used an L1 reconstruction loss to ensure the contents similarity between the generated target images and original images. Bousmalis et al. [30] proposed a content similarity loss to force the generated image to preserve original contents.

In the field of medical image analysis using deep learning, pixel-level domain adaptation has been more and more frequently explored to generalize learned models across domains. Zhao et al. [32] combined the annotated vessel structures with target image style to generate target-like retinal fundus data, then used the synthetic dataset to train a target domain model. Some CycleGAN-based methods have been proposed to tackle the cross-cohort or cross-modality domain shift. For the X-ray segmentation, both [12, 28] translated target X-ray images to resemble the source images, and directly applied the established source model to segment the generated source-like images. In [33], a two-stage approach was proposed to first translate CT images to appear like MRI using CycleGAN, and then used both generated MRI and a few real MRI for semi-supervised tumor segmentation. In [34], an end-to-end synthetic segmentation network was applied for MRI and CT images adaptation, which combined CycleGAN with a segmentation network.

5.3 Feature-Level Adaptation with Latent Space Alignment

In this section, we present a feature-level unsupervised domain adaptation framework with adversarial learning, applied to cross-modality cardiac image segmentations. To transfer the established ConvNet from source domain (MRI) to target domain (CT), we design a plug-and-play domain adaptation module (DAM) which implicitly maps the target input data to the feature space of source domain. Furthermore, we construct a discriminator which is also a ConvNet termed as domain critic module (DCM) to differentiate the feature distributions of two domains. Adversarial loss is derived to train the entire domain adaptation framework in an unsupervised manner, by placing the DAM and DCM into a minimax two-player game. Figure 5.2 presents overview of our method. The details of network architecture, adaptation method, adversarial loss, training strategies, and experimental results are elaborated in the followings.

5.3.1 Method

5.3.1.1 ConvNet Segmenter Architecture

Given a set of N^s labeled samples $\{x_i^s, y_i^s\}_{i=1}^{N^s}$ from the source domain X^s, we conduct supervised learning to establish a mapping from the input image to the label space Y^s. In our setting, the x_i^s represents the sample (pixel or patch) of medical images and y_i^s is the category of anatomical structures. For the ease of denotation, we omit the index i in the following, and directly use x^s and y^s to represent the samples and labels from the source domain.

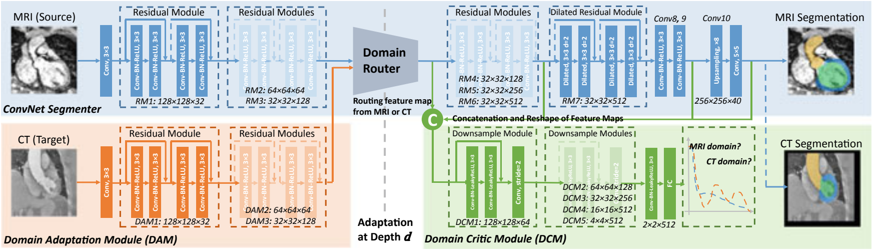


Fig. 5.2 Our proposed feature-level adaptation framework for cross-modality domain adaptation. The DAM and DCM are optimized via adversarial learning. During inference, the domain router is used for routing feature maps of different domains

A segmentation ConvNet is established to implicitly learn the mapping M^s from input to the label space. The backbone of our segmenter is residual network for pixel-wise prediction of biomedical images. We employ the dilated residual blocks [35] to extract representative features from a large receptive field while preserving the spatial acuity of feature maps. This is for the considerations of our network design for feature space alignment, because short cut connections are not expected in our model. More specifically, the image is first input to a Conv layer, then forwarded to three residual modules (termed as RM, each consisting of two stacked residual blocks) and downsampled by a factor of 8. Next, another three RMs and one dilated RM are stacked to form a deep network. To enlarge receptive field for extracting global semantic features, four dilated convolutional layers are used in RM7 with a dilation factor of 2. For dense predictions in our segmentation task, we conduct upsamling at layer Conv10, which is followed by 5×5 convolutions to smooth out the feature maps. Finally, a softmax layer is used for probability predictions of the pixels.

The segmentation ConvNet is optimized with labeled data from the source domain by minimizing the hybrid loss $\mathcal{L}_{\text{seg}}$ composed of the multi-class cross-entropy loss and the Dice coefficient loss [36]. Formally, we denote $y^s_{i,c}$ for binary label regarding class $c \in C$ in sample x^s_i, its probability prediction is $\hat{p}^s_{i,c}$, and the label prediction is $\hat{y}^s_{i,c}$, the source domain segmenter loss function is as follows:

$$\mathcal{L}_{\text{seg}} = -\sum_{i=1}^{N^s} \sum_{c \in C} w^s_c \cdot y^s_{i,c} \log(\hat{p}^s_{i,c}) - \lambda \sum_{c \in C} \frac{\sum_{i=1}^{N^s} 2 y^s_{i,c} \hat{y}^s_{i,c}}{\sum_{i=1}^{N^s} y^s_{i,c} y^s_{i,c} + \sum_{i=1}^{N^s} \hat{y}^s_{i,c} \hat{y}^s_{i,c}}, \quad (5.1)$$

where the first term is the cross-entropy loss for pixel-wise classification, with w^s_c being a weighting factor to cope with the issue of class imbalance. The second term is the Dice loss for multiple cardiac structures, which is commonly employed in biomedical image segmentation problems. We combine the two complementary loss functions to tackle the challenging cardiac segmentation task. In practice, we also tried to use only one type of loss, but the performance was not quite high.

5.3.1.2 Plug-and-Play Domain Adaptation Module

After obtaining the ConvNet learned on the source domain, our goal is to generalize it to a target domain. In transfer learning, the last several layers of the network are usually fine-tuned for a new task with new label space. The supporting assumption is that early layers in the network extract low-level features (such as edge filters and color blobs) which are common for vision tasks. Those upper layers are more task-specific and learn high-level features for the classifier [37, 38]. In this case, labeled data from target domain are required to supervise the learning process. Differently, we use unlabeled data from the target domain, given that labeling dataset is time consuming and expensive. This is critical in clinical practice where radiologists are willing to perform image computing on cross-modality data with as less extra annotation cost as possible. Hence, we propose to adapt the ConvNet with unsupervised learning.

In our segmenter, the source domain mapping M^s is layer-wise feature extractors composing stacked transformations of $\{M^s_{l_1}, \ldots, M^s_{l_n}\}$, with the l denoting the network layer index. Formally, the predictions of labels are obtained by

$$\hat{y}^s = M^s(x^s) = M^s_{l_1:l_n}(x^s) = M^s_{l_n} \circ \cdots \circ M^s_{l_1}(x^s). \tag{5.2}$$

For domain adaptation, the source and target domains share the same label space, i.e., we segment the same anatomical structures from medical MRI/CT data. Our hypothesis is that the distribution changes between the cross-modality domains are primarily low-level characteristics (e.g., gray scale values) rather than high-level (e.g., geometric structures). The higher layers (such as $M^s_{l_n}$) are closely in correlation with the class labels which can be shared across different domains. In this regard, we propose to reuse the feature extractors learned in higher layers of the ConvNet, whereas the earlier layers are updated to conduct distribution mappings in feature space for our unsupervised domain adaptation.

To perform segmentation on target images x^t, we propose a domain adaptation module $\mathcal{M}$ that maps x^t to the feature space of the source domain. We denote the adaptation depth by d, i.e., the layers earlier than and including l_d are replaced by DAM when processing the target domain images. In the meanwhile, the source model's upper layers are frozen during domain adaptation learning and reused for target inference. Formally, the predictions for target domain is

$$\hat{y}^t = M^s_{l_{d+1}:l_n} \circ \mathcal{M}(x^t) = M^s_{l_n} \circ \cdots \circ M^s_{l_{d+1}} \circ \mathcal{M}(x^t), \tag{5.3}$$

where $\mathcal{M}(x^t) = \mathcal{M}_{l_1:l_d}(x^t) = \mathcal{M}_{l_d} \circ \cdots \circ \mathcal{M}_{l_1}(x^t)$ represents the DAM which is also a stacked ConvNet. Overall, we form a flexible plug-and-play domain adaptation framework. During the test inference, the DAM directly replaces the early d layers of the model trained on source domain. The images of target domain are processed and mapped to deep learning feature space of source domain via the DAM. These adapted features are robust to the cross-modality domain shift, and can be mapped to the label space using those high-level layers established on source domain. In practice, the ConvNet configuration of the DAM is identical to $\{M^s_{l_1}, \ldots, M^s_{l_d}\}$. We initialize the DAM with trained source domain model and fine-tune the parameters in an unsupervised manner with adversarial loss.

5.3.1.3 Learning with Adversarial Loss

We propose to employ adversarial loss to train our domain adaptation framework in an unsupervised manner. The spirit of adversarial training roots in GAN, where a generator model and a discriminator model form a minimax two-player game. The generator learns to capture the real data distribution; and the discriminator estimates the probability that a sample comes from the real training data rather than the generated data. These two models are alternatively optimized and compete with each other, until the generator can produce real-like samples that the discriminator fails to differentiate. For our problem, we train the DAM, aiming that the ConvNet can generate source-like feature maps from target input. Hence, the ConvNet is equivalent to a generator from GAN's perspective.

Considering that accurate segmentations come from high-level semantic features, which in turn rely on fine patterns extracted by early layers, we propose to align multiple levels of feature maps between source and target domains (see Fig. 5.2). In practice, we select several layers from the frozen higher layers, and refer their corresponding feature maps as the set of $F_H(\cdot)$ where $H = \{k, \ldots, q\}$ being the set of selected layer indices. Similarly, we denote the selected feature maps of DAM by $\mathcal{M}_A(\cdot)$ with the A being the selected layer set. In this way, the feature space of target domain is $(\mathcal{M}_A(x^t), F_H(x^t))$ and the $(M_A^s(x^s), F_H(x^s))$ is their counterpart for source domain. Given the distribution of $(\mathcal{M}_A(x^t), F_H(x^t)) \sim \mathbb{P}_g$, and that of $(M_A^s(x^s), F_H(x^s)) \sim \mathbb{P}_s$, the distance between these two domain distributions which needs to be minimized is represented as $W(\mathbb{P}_s, \mathbb{P}_g)$. For stabilized training, we employ the Wassertein distance [39] between the two distributions as follows:

$$W(\mathbb{P}_s, \mathbb{P}_g) = \inf_{\gamma \sim \prod(\mathbb{P}_s, \mathbb{P}_g)} \mathbb{E}_{(\mathrm{x},\mathrm{y}) \sim \gamma}[\|\mathrm{x} - \mathrm{y}\|], \tag{5.4}$$

where $\prod(\mathbb{P}_s, \mathbb{P}_g)$ represents the set of all joint distributions $\gamma(\mathrm{x}, \mathrm{y})$ whose marginals are respectively $\mathbb{P}_s$ and $\mathbb{P}_g$.

In adversarial learning, the DAM is pitted against an adversary: a discriminative model that implicitly estimates the $W(\mathbb{P}_s, \mathbb{P}_g)$. We refer our discriminator as domain critic module and denote it by $\mathcal{D}$. Specifically, our constructed DCM consists of several stacked residual blocks, as illustrated in Fig. 5.2. In each block, the number of feature maps is doubled until it reaches 512, while their sizes are decreased. We concatenate the multiple levels of feature maps as input to the DCM. This discriminator would differentiate the complicated feature space between the source and target domains. In this way, our domain adaptation approach not only removes source-specific patterns in the beginning but also disallows their recovery at higher layers [20]. In unsupervised learning, we jointly optimize the generator $\mathcal{M}$ (DAM) and the discriminator $\mathcal{D}$ (DCM) via adversarial loss. Specifically, with X^t being target set, the loss for learning the DAM is

$$\min_{\mathcal{M}} \mathcal{L}_{\mathcal{M}}(X^t, \mathcal{D}) = -\mathbb{E}_{(\mathcal{M}_A(x^t), F_H(x^t)) \sim \mathbb{P}_g}[\mathcal{D}(\mathcal{M}_A(x^t), F_H(x^t))]. \tag{5.5}$$

Then, with the X^s representing the set of source images, the DCM is optimized via

$$\begin{aligned} &\min_{\mathcal{D}} \mathcal{L}_{\mathcal{D}}(X^s, X^t, \mathcal{M}) = \\ &\mathbb{E}_{(\mathcal{M}_A(x^t), F_H(x^t)) \sim \mathbb{P}_g}[\mathcal{D}(\mathcal{M}_A(x^t), F_H(x^t))] - \\ &\mathbb{E}_{(M_A^s(x^s), F_H(x^s)) \sim \mathbb{P}_s}[\mathcal{D}(M_A^s(x^s), F_H(x^s))], \; s.t. \; \|\mathcal{D}\|_{L \leq K}, \end{aligned} \tag{5.6}$$

where K is a constant that applies Lipschitz constraint to $\mathcal{D}$.

During the alternative updating of $\mathcal{M}$ and $\mathcal{D}$, the DCM outputs a more precise estimation of $W(\mathbb{P}_s, \mathbb{P}_g)$ between distributions of the feature space from both domains. The updated DAM is more effective to generate source-like feature maps for conducting cross-modality domain adaptation.

5.3.1.4 Training Strategies

In our setting, the source domain is biomedical cardiac MRI images and the target domain is CT data. All the volumetric MRI and CT images were resampled to the voxel spacing of $1 \times 1 \times 1$ mm^3 and cropped into the size of $256 \times 256 \times 256$ centering at the heart region. In preprocessing, we conducted intensity standardization for each domain, respectively. Augmentations of rotation, zooming, and affine transformations were employed to combat over fitting. To leverage the spatial information existing in volumetric data, we sampled consecutive three slices along the coronal plane and input them to three channels. The label of the intermediate slice is utilized as the ground truth when training the 2D networks.

We first trained the segmenter on the source domain data in supervised manner with stochastic gradient descent. The Adam optimizer was employed with parameters as batch size of 5, learning rate of 1×10^{-3} and a stepped decay rate of 0.95 every 1500 iterations. After that, we alternatively optimized the DAM and DCM with the adversarial loss for unsupervised domain adaptation. Following the heuristic rules of training WGAN [39], we updated the DAM every 20 times when updating the DCM. In adversarial learning, we utilized the RMSProp optimizer with a learning rate of 3×10^{-4} and a stepped decay rate of 0.98 every 100 joint updates, with weight clipping for the discriminator being 0.03.

5.3.2 Experimental Results

5.3.2.1 Dataset and Evaluation Metrics

We validated our proposed unsupervised domain adaptation method on the public dataset of *MICCAI 2017 Multi-Modality Whole Heart Segmentation* for cross-modality cardiac segmentation in MRI and CT images [40]. This dataset consists of unpaired 20 MRI and 20 CT images from 40 patients. The MRI and CT data were acquired in different clinical centers. The cardiac structures of the images were manually annotated by radiologists for both MRI and CT images. Our ConvNet segmenter aimed to automatically segment four cardiac structures including the ascending aorta (AA), the left atrium blood cavity (LA-blood), the left ventricle blood cavity (LV-blood), and the myocardium of the left ventricle (LV-myo). For each modality, we randomly split the dataset into training (16 subjects) and testing (4 subjects) sets, which were fixed throughout all experiments.

For evaluation, we employed two commonly used metrics to quantitatively evaluate the segmentation performance of automatic methods [41]. The DICE coefficient ([%]) was employed to assess the agreement between the predicted segmentation and ground truth for cardiac structures. We also calculated the average surface distance (ASD[voxel]) to measure the segmentation performance from the perspective of the boundary. A higher Dice and lower ASD indicate better segmentation performance. Both metrics are presented in the format of *mean*±*std*, which shows the average performance as well as the cross-subject variations of the results (Table 5.1).

5.3.2.2 Experimental Settings

We employed the MRI images as the source domain and the CT dataset as the target domain. We demonstrated the effectiveness of the proposed unsupervised cross-modality domain adaptation method with extensive experiments. We designed several experiment settings: (1) training and testing the ConvNet segmenter on source domain (referred as *Seg-MRI*); (2) training the segmenter from scratch on annotated target domain data (referred as *Seg-CT*); (3) fine-tuning the source domain segmenter with annotated target domain data, i.e., the supervised transfer learning (referred as *Seg-CT-STL*); (4) directly testing the source domain segmenter on target domain data (referred as *Seg-CT-noDA*); (5) our proposed unsupervised domain adaptation method (referred as *Seg-CT-UDA*). We also compared with a previous state-of-the-art heart segmentation method using ConvNets [42]. Last but not least, we conducted ablation studies to observe how the adaptation depth would affect the performance.

5.3.2.3 Results of UDA on Cross-Modality Cardiac Images

Table 5.1 reports the comparison results of different methods, where we can see that the proposed unsupervised domain adaptation method is effective by mapping the feature space of the target CT domain to that of the source MRI domain. Qualitative results of the segmentations for CT images are presented in Fig. 5.3.

In the experiment setting *Seg-MRI*, we first evaluate the performance of the source domain model, which serves as the basis for subsequent domain adaptation procedures. Compared with [42], our ConvNet segmenter reached promising performance with exceeding Dice on LV-blood and LV-myo, as well as comparable Dice on AA and LA-blood. With this standard segmenter network architecture, we conducted following experiments to validate the effectiveness of our unsupervised domain adaptation framework.

To experimentally explore the potential upper bounds of the segmentation accuracy of the cardiac structures from CT data, we implemented two different settings, i.e., the *Seg-CT* and *Seg-CT-STL*. Generally, the segmenter fine-tuned from *Seg-MRI* achieved higher Dice and lower ASD than the model trained from scratch, proving the effectiveness of supervised transfer learning for adapting an established network

Table 5.1 Quantitative comparison of segmentation performance on cardiac structures between different methods. (Note: the - means that the results were not reported by that method.)

Methods	AA		LA-blood		LV-blood		LV-myo	
	Dice	ASD	Dice	ASD	Dice	ASD	Dice	ASD
DL-MR [42]	76.6 ± 13.8	–	81.1 ± 13.8	–	87.7 ± 7.7	–	75.2 ± 12.1	–
DL-CT [42]	91.1 ± 18.4	–	92.4 ± 3.6	–	92.4 ± 3.3	–	87.2 ± 3.9	–
Seg-MRI	75.9 ± 5.5	12.9 ± 8.4	78.8 ± 6.8	16.0 ± 8.1	90.3 ± 1.3	2.0 ± 0.2	75.5 ± 3.6	2.6 ± 1.4
Seg-CT	81.3 ± 24.4	2.1 ± 1.1	89.1 ± 3.0	10.6 ± 6.9	88.8 ± 3.7	21.3 ± 8.8	73.3 ± 5.9	42.8 ± 16.4
Seg-CT-STL	78.3 ± 2.8	2.9 ± 2.0	89.7 ± 3.6	7.6 ± 6.7	91.6 ± 2.2	4.9 ± 3.2	85.2 ± 3.3	5.9 ± 3.8
Seg-CT-noDA	19.7 ± 2.0	31.2 ± 17.5	25.7 ± 17.2	8.7 ± 3.3	0.8 ± 1.3	N/A	11.1 ± 14.4	31.0 ± 37.6
Seg-CT-UDA ($d = 13$)	63.9 ± 15.4	**13.9 ± 5.6**	54.7 ± 13.2	16.6 ± 6.8	35.1 ± 26.1	**18.4 ± 5.1**	35.4 ± 18.4	**14.2 ± 5.3**
Seg-CT-UDA ($d = 21$)	**74.8 ± 6.2**	27.5 ± 7.6	51.1 ± 11.2	20.1 ± 4.5	**57.2 ± 12.4**	29.5 ± 11.7	**47.8 ± 5.8**	31.2 ± 10.1
Seg-CT-UDA ($d = 31$)	71.9 ± 0.5	25.8 ± 12.5	**55.2 ± 22.9**	**15.2 ± 8.2**	39.2 ± 21.8	21.2 ± 3.9	34.3 ± 19.1	24.7 ± 10.5

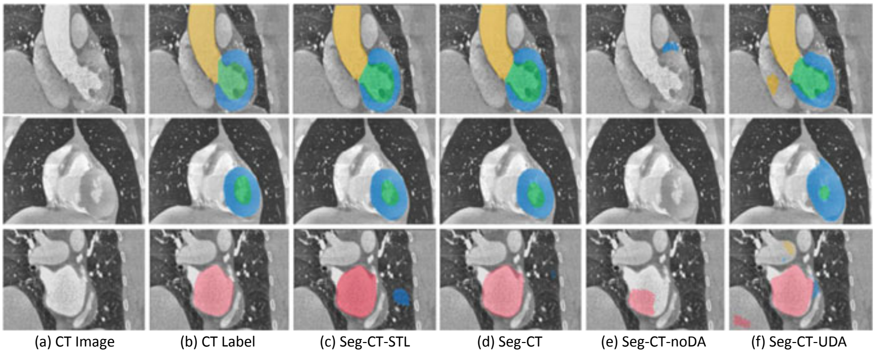


Fig. 5.3 Results of different methods for CT image segmentations. Each row presents one typical example, from left to right: **a** raw CT slices **b** ground truth labels **c** supervised transfer learning **d** ConvNets trained from scratch **e** directly applying MRI segmenter on CT data **f** our unsupervised cross-modality domain adaptation results. The structures of AA, LA-blood, LV-blood, and LV-myo are indicated by yellow, red, green, and blue colors, respectively (best viewed in color)

to a related target domain using additional annotations. Meanwhile, these results are comparable to [42] on most of the four cardiac structures.

To demonstrate the severe domain shift inherent in cross-modality biomedical images, we directly applied the segmenter trained on MRI domain to the CT data without any domain adaptation procedure. Unsurprisingly, the network of *Seg-MRI* completely failed on CT images, with average Dice of merely 14.3% across the structures. As shown in Table 5.1, the *Seg-CT-noDA* only got a Dice of 0.8% for the LV-blood. The model did not even output any correct predictions for two of the four testing subjects on the structure of LV-blood (please refer to (e) in Fig. 5.3). This demonstrates that although the cardiac MRI and CT images share similar high-level representations and identical label space, the significant difference in their low-level characteristics makes it extremely difficult for MRI segmenter to extract effective features for CT.

With our proposed unsupervised domain adaptation method, a great improvement of the segmentation performance on the target CT data was achieved compared with the *Seg-CT-noDA*. More specifically, our *Seg-CT-UDA (d = 21)* model has increased the average Dice across four cardiac structures by 43.4%. As presented in Fig. 5.3, the predicted segmentation masks from *Seg-CT-UDA* can successfully localize the cardiac structures and further capture their anatomical shapes. The performance on segmenting AA is even close to that of *Seg-CT-STL*. This reflects that the distinct geometric pattern and the clear boundary of the AA have been successfully captured by the DCM. In turn, it supervises the DAM to generate similar activation patterns as the source feature space via adversarial learning. Looking at the other three cardiac structures (i.e., LA-blood, LV-blood, and LV-myo), the *Seg-CT-UDA* performances are not as high as that of AA. The reason is that these anatomical structures are more challenging, given that they come with either relatively irregular geometrics or limited intensity contrast with surrounding tissues. The deficiency focused on the unclear boundaries between neighboring structures or noise predictions on relatively homogeneous tissues away from the ROI. This is responsible for the high ASDs of *Seg-CT-UDA*, where boundaries are corrupted by noisy outputs. Nevertheless, by mapping the feature space of target domain to that of the source domain, we obtained greatly improved and promising segmentations against *Seg-CT-noDA* with zero data annotation effort.

5.3.2.4 Ablation Study on Adaptation Depth

We conduct ablation experiments to study the adaptation depth d, which is an important hyperparameter in our framework to determine how many layers to be replaced during the plug-and-play domain adaptation procedure. Intuitively, a shallower DAM (i.e., smaller d) might be less capable of learning effective feature mapping function $\mathcal{M}$ across domains than a deeper DAM (i.e., larger d). This is due to the insufficient capacity of parameters in shallow DAM, as well as the huge domain shift in feature distributions. Conversely, with an increase in adaptation depth d, DAM becomes more powerful for feature mappings, but training a deeper DAM solely with adversarial gradients would be more challenging.

To experimentally demonstrate how the performance would be affected by d and search for an optimal d, we repeated the experiments with domain adaptation from MRI to CT by varying the $d = \{13, 21, 31\}$, while maintaining all the other settings the same. Viewing the examples in Fig. 5.4, *Seg-CT-UDA (d=21)* model obtained an approaching ground truth segmentation mask for ascending aorta. The other two models also produced inspiring results capturing the geometry and boundary characteristics of *AA*, validating the effectiveness of our unsupervised domain adaptation method. From Table 5.1, we can observe that DAM with a middle-level of adaptation depth ($d = 21$) achieved the highest Dice on three of the four cardiac structures, exceeding the other two models by a significant margin. For the LA-blood, the three adaptation depths reached comparable segmentation Dice and ASD, and the $d = 31$ model was the best. Notably, the model of *Seg-CT-UDA* ($d = 31$) overall demonstrated superiority over the model with adaptation depth $d = 13$. This shows that enabling more layers learnable helps to improve the domain adaptation performance on cross-modality segmentations.

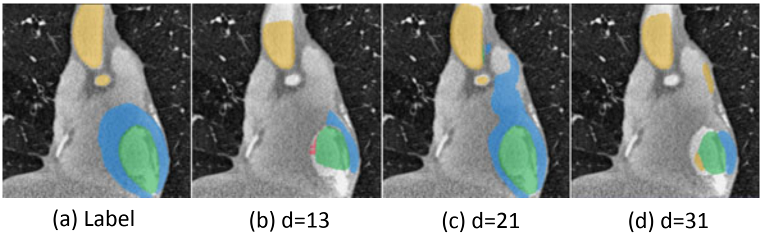


Fig. 5.4 Comparison of results using *Seg-CT-UDA* with different adaptation depths (colors are the same with Fig. 5.3)

5.4 Pixel-Level Adaptation with Image-to-Image Translation

In this section, we present a pixel-level unsupervised domain adaptation framework with generative adversarial network, applied to cross-cohort X-ray lung segmentation. Different from feature-level adaptation method described in the last section, this pixel-level adaptation method detaches the segmentation ConvNets from the domain adaptation process. Given a test image, our framework conducts image-to-image transformation to generate a source-like image which is directly forwarded to the established source ConvNet. To enhance the preservation of structural information during image transformation, we improve CycleGAN with a novel semantic-aware loss by embedding a nested adversarial learning in semantic label space. Our method is named as *SeUDA*, standing for semantic-aware unsupervised domain adaptation, and Fig. 5.5 presents overview of it. Details of network configurations, adversarial losses and experimental results will be presented in the followings.

5.4.1 *Method*

With a set of the source domain images $x^s \in \mathcal{X}^s$ and corresponding labels $y^s \in \mathcal{Y}$, we train a ConvNet, denoted by f^s, to segment the input images. For a new set of the target domain images $x^t \in \mathcal{X}^t$, we aim to adapt the appearance of x^t to source image space $\mathcal{X}^s$, so that the established f^s can be directly generalized to the transformed image.

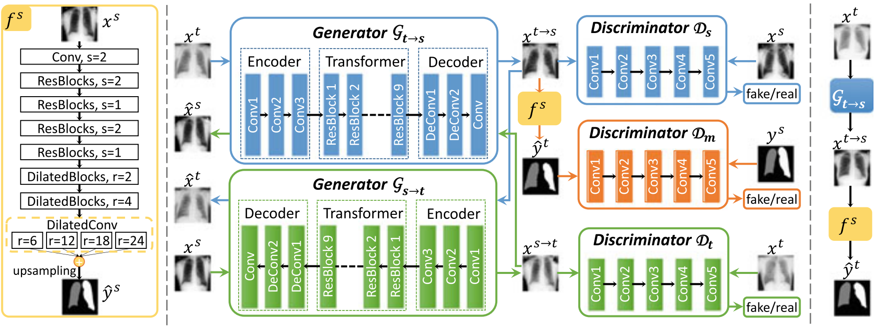


Fig. 5.5 The overview of our unsupervised domain adaptation framework. Left: the segmentation DNN learned on source domain; Middle: the *SeUDA* where the paired generator and discriminator are indicated with the same color, the blue/green arrows illustrate the data flows from original images (x^t/x^s) to transformed images ($x^{t\to s}/x^{s\to t}$) then back to reconstructed images ($\hat{x}^t/\hat{x}^s$) in cycle-consistency loss, the orange part is the discriminator for the semantic-aware adversarial learning; Right: the inference process of *SeUDA* given a new target image for testing

5.4.1.1 ConvNet Segmenter Architecture

To establish a state-of-the-art segmentation network, we make complementary use of the residual connection, dilated convolution and multi-scale feature fusion. The backbone of our segmenter is modified ResNet-101. We replace the standard convolutional layers in the high-level residual blocks with the dilated convolutions. To leverage features with multi-scale receptive fields, we replace the last fully connected layer with four parallel 3×3 dilated convolutional branches, with a dilation rate of $\{6, 12, 18, 24\}$, respectively. An upsampling layer is added in the end to produce dense predictions for the segmentation task. We start with 32 feature maps in the first layer and double the number of feature maps when the spatial size is halved or the dilation convolutions are utilized. The segmenter is optimized by minimizing the pixel-wise multi-class cross-entropy loss of the prediction $f^s(x^s)$ and ground truth y^s with standard stochastic gradient descent.

5.4.1.2 Image Transformation with Semantic-Aware CycleGAN

With the source domain model f^s which maps the source input space $\mathcal{X}^s$ to the semantic label space $\mathcal{Y}$, our goal is to make it generally applicable to new target images. Given that annotating medical data is quite expensive, we conduct the domain adaptation in an unsupervised manner. Specifically, we map the target images toward the source image space. The generated new image $x^{t \to s}$ appears to be drawn from $\mathcal{X}^s$ while the content and semantic structures remain unchanged. In this way, we can directly apply the well-established model f^s on $x^{t \to s}$ without retraining and get the segmentation result for x^t.

To achieve this, we use generative adversarial networks [10], which have made a wide success for pixel-to-pixel image translation, by constructing a generator $\mathcal{G}_{t \to s}$ and a discriminator $\mathcal{D}_s$. The generator aims to produce realistic transformed image $x^{t \to s} = \mathcal{G}_{t \to s}(x^t)$. The discriminator competes with the generator by trying to distinguish between the fake generated data $x^{t \to s}$ and the real source data x^s. The GAN corresponds to a minimax two-player game and is optimized via the following objective:

$$\mathcal{L}_{\text{GAN}}(\mathcal{G}_{t \to s}, \mathcal{D}_s) = \mathbb{E}_{x^s}[\log \mathcal{D}_s(x^s)] + \mathbb{E}_{x^t}[\log(1 - \mathcal{D}_s(\mathcal{G}_{t \to s}(x^t)))], \tag{5.7}$$

where the discriminator tries to maximize this objective to correctly classify the $x^{t \to s}$ and x^s, while the generator tries to minimize $\log(1 - \mathcal{D}_s(\mathcal{G}_{t \to s}(x^t)))$ to learn the data distribution mapping from $\mathcal{X}^t$ to $\mathcal{X}^s$.

Cycle-consistency adversarial learning. To achieve domain adaptation with image transformation, it is crucial that the detailed contents in the original x^t are well preserved in the generated $x^{t \to s}$. Inspired by the CycleGAN [26], we employ the cycle-consistency loss during the adversarial learning to maintain the contents with clinical clues of the target images.

We build a reverse source-to-target generator $\mathcal{G}_{s\to t}$ and a target discriminator $\mathcal{D}_t$, to bring the transformed image back to the original image. This pair of models are trained with a same way GAN loss $\mathcal{L}_{\text{GAN}}(\mathcal{G}_{s\to t}, \mathcal{D}_t)$ following the Eq. (5.7). In this regard, we derive the cycle-consistency loss which encourages $\mathcal{G}_{s\to t}(\mathcal{G}_{t\to s}(x^t)) \approx x^t$ and $\mathcal{G}_{t\to s}(\mathcal{G}_{s\to t}(x^s)) \approx x^s$ in the transformation:

$$\mathcal{L}_{\text{cyc}}(\mathcal{G}_{t\to s}, \mathcal{G}_{s\to t}) = \mathbb{E}_{x^t}[||\mathcal{G}_{s\to t}(\mathcal{G}_{t\to s}(x^t)) - x^t||_1] + \mathbb{E}_{x^s}[||\mathcal{G}_{t\to s}(\mathcal{G}_{s\to t}(x^s)) - x^s||_1], \tag{5.8}$$

where the L1-Norm is employed for reducing blurs in the generated images. This loss imposes the pixel-level penalty on the distance between the cyclic transformation result and the input image.

Semantic-aware adversarial learning. The image quality of $x^{t\to s}$ and the stability of $\mathcal{G}_{t\to s}$ are crucial for the effectiveness of our method, since we apply the established f^s to $x^{t\to s}$ which is obtained by inputting x^t to $\mathcal{G}_{t\to s}$. Therefore, besides the cycle-consistency loss which composes both generators and constraints the cyclic input–output consistency, we further try to explicitly enhance the intermediate transformation result $x^{t\to s}$. Specifically, for our segmentation domain adaptation task, we design a novel semantic-aware loss which aims to prevent the semantic distortion during the image transformation.

In our unsupervised learning scenario, we establish a nested adversarial learning module by adding another new discriminator $\mathcal{D}_m$ into the system. It distinguishes between the source domain ground truth lung mask y^s and the predicted lung mask $f^s(x^{t\to s})$ obtained by applying the segmenter on the source-like transformed image. Our underlying hypothesis is that the shape of anatomical structure is consistent across multicenter medical images. The prediction of $f^s(x^{t\to s})$ should follow the regular semantic structures of the lung to fool the $\mathcal{D}_m$, otherwise, the generator $\mathcal{G}_{t\to s}$ would be penalized by the semantic-aware loss:

$$\mathcal{L}_{\text{sem}}(\mathcal{G}_{t\to s}, \mathcal{D}_m) = \mathbb{E}_{y^s}[\log \mathcal{D}_m(y^s)] + \mathbb{E}_{x^t}[\log(1 - \mathcal{D}_m(f^s(\mathcal{G}_{t\to s}(x^t))))]. \tag{5.9}$$

This loss imposes an explicit constraint on the intermediate result of the cyclic transformation. Its gradients can assist the update of the generator $\mathcal{G}_{t\to s}$, which benefits the stability of the entire adversarial learning procedure.

5.4.1.3 Learning Procedure and Implementation Details

We follow the practice of [26] to configure the generators and discriminators. Specifically, both generators have the same architecture consisting of an encoder (three convolutions), a transformer (nine residual blocks), and a decoder (two deconvolutions and one convolution). All the three discriminators process 70 × 70 patches and produce real/fake predictions via 3 stride-2 and 2 stride-1 convolutional layers. The overall objective for the generators and discriminators is as follows:

$$\begin{aligned}\mathcal{L}(\mathcal{G}_{s\to t}, \mathcal{G}_{t\to s}, \mathcal{D}_s, \mathcal{D}_t, \mathcal{D}_m) = \; & \mathcal{L}_{GAN}(\mathcal{G}_{s\to t}, \mathcal{D}_t) + \alpha\mathcal{L}_{GAN}(\mathcal{G}_{t\to s}, \mathcal{D}_s) + \\ & \beta\mathcal{L}_{\text{cyc}}(\mathcal{G}_{t\to s}, \mathcal{G}_{s\to t}) + \lambda\mathcal{L}_{\text{sem}}(\mathcal{G}_{t\to s}, \mathcal{D}_m),\end{aligned} \tag{5.10}$$

where the $\{\alpha, \beta, \lambda\}$ denote trade-off hyperparameters adjusting the importance of each component, which is empirically set to be $\{0.5, 10, 0.5\}$ in our experiments. The entire framework is optimized to obtain

$$\mathcal{G}^*_{s\to t}, \mathcal{G}^*_{t\to s} = \arg \min_{\substack{\mathcal{G}_{s\to t} \\ \mathcal{G}_{t\to s}}} \max_{\mathcal{D}_s, \mathcal{D}_t, \mathcal{D}_m} \mathcal{L}(\mathcal{G}_{s\to t}, \mathcal{G}_{t\to s}, \mathcal{D}_s, \mathcal{D}_t, \mathcal{D}_m). \tag{5.11}$$

The generators $\{\mathcal{G}_{t\to s}, \mathcal{G}_{s\to t}\}$ and discriminators $\{\mathcal{D}_s, \mathcal{D}_t, \mathcal{D}_m\}$ are optimized altogether and updated successively. Note that the segmenter f^s is not updated in the process of image transformation. In practice, when training the generative adversarial networks, we followed the strategies of [26] for reducing model oscillation. Specifically, the negative log likelihood in $\mathcal{L}_{GAN}$ was replaced by a least-square loss to stabilize the training. The discriminator loss was calculated using one image from a collection of fifty previously generated images rather than the one produced in the latest training step. We used the Adam optimizer with an initial learning rate of 0.002, which was linearly decayed every 100 epochs. We implemented our proposed framework on the TensorFlow platform using an Nvidia Titan Xp GPU.

5.4.2 Experimental Results

5.4.2.1 Datasets and Evaluation Metrics

Our unsupervised domain adaptation method was validated on lung segmentations using two public Chest X-ray datasets, i.e., the Montgomery set (138 cases) [43] and the JSRT set (247 cases) [44]. Both the datasets are typical X-ray scans collected in clinical practice, but their image distributions are quite different in terms of the disease type, intensity, and contrast (see the first and fourth columns in Fig. 5.6a). The ground truth masks of left and right lungs are provided in both datasets. We randomly split each dataset into 7:1:2 for training, validation and test sets. All the images were resized to 512×512, and rescaled to $[0, 255]$. The prediction masks were post-processed with the largest connected-component selection and hole filling.

To quantitatively evaluate our method, we utilized four common segmentation measurements, i.e., the Dice coefficient ([%]), recall ([%]), precision ([%]) and average surface distance (ASD)([mm]). The first three metrics are measured based on the pixel-wise classification accuracy. The ASD assesses the model performance at boundaries and a lower value indicates better segmentation performance.

Table 5.2 Quantitative evaluation results of pixel-level domain adaptation methods for right/left lung segmentations from chest X-ray images

Methods	Right lung				Left lung			
	Dice	Recall	Precision	ASD	Dice	Recall	Precision	ASD
S-test	95.98	97.98	94.23	2.23	95.23	96.56	94.01	2.45
T-noDA	82.29	98.40	73.38	10.68	76.65	95.06	69.15	11.40
T-HistM [45]	90.05	92.96	88.05	5.72	91.03	94.35	88.45	4.66
T-FeatDA [20]	94.85	93.66	96.42	3.26	92.93	91.67	94.46	3.80
T-STL [7]	96.91	98.47	95.46	1.93	95.84	97.48	94.29	2.20
CyUDA	94.09	96.31	92.28	3.88	91.59	92.28	91.70	4.57
SeUDA (Ours)	95.59	96.55	94.77	2.85	93.42	92.40	94.70	3.51

5.4.2.2 Experimental Settings

In our experiments, the source domain is the Montgomery set and the target domain is the JSRT set. We first established the segmenter on source training data independently. Next, we test the segmenter under various settings: (1) testing on source domain (*S-test*); (2) directly testing on target data (*T-noDA*); (3) using histogram matching to adjust target images before testing (*T-HistM*); (4) aligning target features with the source domain as proposed in [20] (*T-FeatDA*); (5) fine-tuning the model on labeled target data before testing on JSRT (*T-STL*); In addition, we investigated the performance of our proposed domain adaptation method with and w/o the semantic-aware loss, i.e., *SeUDA* and *CyUDA*.

5.4.2.3 Results of UDA on Cross-Cohort Chest X-Ray Images

The comparison results of different methods are listed in Table 5.2. We can see that when directly applying the learned source domain segmenter to target data (*T-noDA*), the model performance significantly degraded, indicating that domain shift would severely impede the generalization performance of DNNs. Specifically, the average Dice over both lungs dropped from 95.61 to 79.47%, and the average ASD increased from 2.34 to 11.04 mm.

With our proposed method, we find a remarkable improvement by applying the source segmenter on transformed target images. Compared with *T-noDA*, our *SeUDA* increased the average Dice by 15.04%. Meanwhile, the ASDs for both lungs were reduced significantly. Also, our method outperforms the UDA baseline histogram matching *T-HistM* with the average dice increased by 3.97% and average ASD decreased from 5.19 to 3.18 mm. Compared with the feature-level domain adaptation method *T-FeatDA*, our *SeUDA* can not only obtain higher segmentation performance,

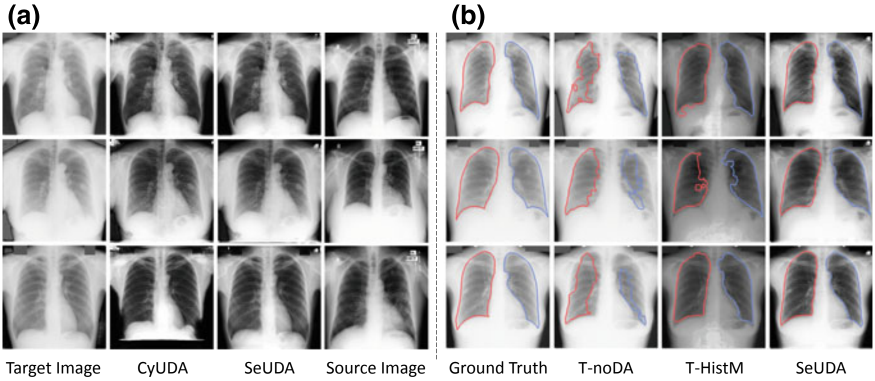


Fig. 5.6 Typical results for the image transformation and lung segmentation. **a** Visualization of image transformation results, from left to right, are the target images in JSRT set, *CyUDA* transformation results, *SeUDA* transformation results, and the nearest neighbor of $x^{t \to s}$ got from source set; each row corresponds to one patient. **b** Comparison of segmentation results between the ground truth, *T-noDA*, *T-HistM*, and our proposed *SeUDA*; each row corresponds to one patient

but also provide intuitive visualization of how the adaptation is achieved. Notably, the performance of our unsupervised *SeUDA* is even comparable to the upper bound of supervised *T-STL*. In Table 5.2, the gaps of Dice are marginal, i.e., 1.32% for right lung and 2.42% for left lung.

The typical transformed target images can be visualized in Fig. 5.6a, demonstrating that *SeUDA* has successfully adapted the appearance of target data to look similar to source images. In addition, the positions, contents, semantic structures, and clinical clues are well preserved after transformation. In Fig. 5.6b, we can observe that without domain adaptation, the predicted lung masks are quite cluttered. With histogram matching, appreciable improvements are obtained but the transformed images cannot mimic the source images very well. With our *SeUDA*, the lung areas are accurately segmented attributing to the good target-to-source appearance transformation.

5.4.2.4 Effectiveness of Semantic-Aware Loss with Ablation Study

We conduct ablation experiments to investigate the contribution of our novel semantic-aware loss designed for segmentation domain adaptation. We implemented *CyUDA* by removing the semantic-aware loss from the *SeUDA*. One notorious problem of GANs is that their training would be unstable and sensitive to initialization states [30, 46]. In this study, we measured the standard deviation (std) of the *CyUDA* and *SeUDA* by running each model for 10 times under different initializations but with the same hyperparameters. We observed significant lower variability on the segmentation performance across the 10 *SeUDA* models than the 10 *CyUDA* models, i.e., Dice std: 0.25 versus 2.03%, ASD std: 0.16 versus 1.19 mm. Qualitatively, we observe that the *CyUDA* transformed images may suffer from distorted lung boundaries in some cases, see the third row in Fig. 5.6a. In contrast, adding the semantic-aware loss, the transformed images consistently present a high quality. This reveals that the novel semantic-aware loss contributes to stabilize the image transformation process and prevent the distortion in structural contents, and hence contributes to boost the performance of segmentation domain adaptation.

5.5 Discussion

This chapter introduces how to tackle domain adaptation problem in medical imaging from two different perspectives. This is an essential and urgent topic to study the generalization capability and robustness of ConvNets, given that deep learning nowadays has become the state of the art for solving image recognition tasks. Resolving this issue will help to promote deep learning studies based on large-scale real-world clinical dataset composing inhomogeneous images [47].

Fine-tuning the ConvNets with a set of new labeled images from the target domain can improve the model's performance on target data. However, this straightforward supervised solution still requires extra efforts from clinicians for constructing the annotated fine-tune dataset. Unsupervised domain adaptation methods are more appealing and practical in the long-run, though it is technically challenging at current stage. Basically, the UDA requires to model and map the underlying distributions of different domains, either in latent feature space or appearance pixel space. The insights of adversarial networks fit well into this scope, as which can implicitly learn how to model, transform, and discriminate the data distributions via highly nonlinear networks. This forms the basis of the situation that adversarial learning has been frequently investigated for unsupervised domain adaptation tasks.

Feature-level adaptation and pixel-level adaptation are two independent ways to conduct unsupervised domain adaptation, with ideas from different perspectives. Feature-level adaptation aims to transform different data sources into a shared latent space with domain-invariant features, such that a shared classifier can be established in this common space. The advantage is that the classifier is learned in a high-quality homogeneous feature space, with reduced confounding factors from scanner effects. The disadvantage is that the obtained domain-invariant features are unclear for interpretation and intuitive visualization. Pixel-level adaptation aims to transform the image appearance from one domain to the other, and use the transformed images to train or test a model. The advantage for this stream of solution is that we can directly assess the quality of domain adaptation by observing the transformed images. The disadvantage is that there may still exist a domain gap between the synthetic images and real images. It is worth noting that these two independent manners of matching across domains can be complementary to each other. Jointly taking advantage of both is feasible and have good potential to present more appealing performance to narrow the domain gap.

5.6 Conclusion

In conclusion, this chapter presents unsupervised domain adaptation methods for medical image segmentation using adversarial learning. Solutions from two different perspectives are presented, i.e., feature-level adaptation and pixel-level adaptation. The feature-level adaptation method has been validated on cross-modality (MRI/CT) cardiac image segmentation. The pixel-level adaptation method has been validated on cross-cohort X-ray images for lung segmentation. Both application scenarios of unsupervised domain adaptation have demonstrated highly promising results on generalizing the ConvNets to the unseen target domain. The proposed frameworks are general and can be extended to other similar scenarios in medical image computing with domain shift issues.

Acknowledgements This work was supported by a research grant supported by the Hong Kong Innovation and Technology Commission under ITSP Tier 2 Platform Scheme (Project No. ITS/426/17FP) and a research grant from the Hong Kong Research Grants Council General Research Fund (Project No. 14225616).

References

1. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) MICCAI 2015. LNCS. Springer, Munich, Germany, pp 234–241
2. 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: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 556–564
3. Roth HR, Lu L, Seff A, Cherry KM, Hoffman J, Wang S, Liu J, Turkbey E, Summers RM (2014) A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 520–527
4. Dou Q, Chen H, Yueming J, Huangjing L, Jing Q, Heng P (2017) Automated pulmonary nodule detection via 3d convnets with online sample filtering and hybrid-loss residual learning. In: MICCAI, pp 630–638
5. Bejnordi BE, Veta M, Van Diest PJ, Van Ginneken B, Karssemeijer N, Litjens G, Van Der Laak JA, Hermsen M, Manson QF, Balkenhol M et al (2017) Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. Jama 318(22):2199–2210
6. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115
7. Ghafoorian M, Mehrtash A, Kapur T, Karssemeijer N, Marchiori E, Pesteie M, Guttmann CR, de Leeuw FE, Tempany CM, van Ginneken B et al (2017) Transfer learning for domain adaptation in mri: application in brain lesion segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 516–524
8. Gibson E, Hu Y, Ghavami N, Ahmed HU, Moore C, Emberton M, Huisman HJ, Barratt DC (2018) Inter-site variability in prostate segmentation accuracy using deep learning. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 506–514

9. Philipsen RH, Maduskar P, Hogeweg L, Melendez J, Sánchez CI, van Ginneken B (2015) Localized energy-based normalization of medical images: application to chest radiography. IEEE Trans Med Imaging 34(9):1965–1975
10. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B et al (2014) Generative adversarial nets. In: Conference on neural information processing systems (NIPS), pp 2672–2680
11. Dou Q, Ouyang C, Chen C, Chen H, Heng PA (2018) Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. arXiv:180410916
12. Chen C, Dou Q, Chen H, Heng PA (2018) Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest x-ray segmentation. arXiv:180600600
13. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: maximizing for domain invariance. arXiv:14123474
14. Long M, Cao Y, Wang J, Jordan MI (2015) Learning transferable features with deep adaptation networks. In: International conference on machine learning (ICML), pp 97–105
15. Sun B, Saenko K (2016) Deep coral: correlation alignment for deep domain adaptation. In: European conference on computer vision (ECCV) workshops, pp 443–450
16. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2030–2096
17. Tzeng E, Hoffman J, Saenko K, Darrell T (2017) Adversarial discriminative domain adaptation. In: CVPR, pp 2962–2971
18. Tsai Y, Hung W, Schulter S, Sohn K, Yang M, Chandraker M (2018) Learning to adapt structured output space for semantic segmentation. In: IEEE conference on computer vision and pattern recognition. CVPR, pp 7472–7481
19. Sankaranarayanan S, Balaji Y, Jain A, Lim SN, Chellappa R (2018) Learning from synthetic data: addressing domain shift for semantic segmentation. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 3752–3761
20. Kamnitsas K et al (2017) Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In: IPMI. Springer, Berlin, pp 597–609
21. Joyce T, Chartsias A, Tsaftaris SA (2018) Deep multi-class segmentation without ground-truth labels. In: International conference on medical imaging with deep learning (MIDL)
22. Degel MA, Navab N, Albarqouni S (2018) Domain and geometry agnostic cnns for left atrium segmentation in 3d ultrasound. In: MICCAI, pp 630–637
23. Ren J, Hacihaliloglu I, Singer EA, Foran DJ, Qi X (2018) Adversarial domain adaptation for classification of prostate histopathology whole-slide images. In: MICCAI, pp 201–209
24. Dong N, Kampffmeyer M, Liang X, Wang Z, Dai W, Xing E (2018) Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio. In: MICCAI. Springer, Berlin, pp 544–552
25. Zhang L, Pereañez M, Piechnik SK, Neubauer S, Petersen SE, Frangi AF (2018) Multi-input and dataset-invariant adversarial learning (mdal) for left and right-ventricular coverage estimation in cardiac mri. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 481–489
26. Zhu J, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV, pp 2242–2251
27. Russo P, Carlucci FM, Tommasi T, Caputo B (2018) From source to target and back: Symmetric bi-directional adaptive GAN. In: IEEE conference on computer vision and pattern recognition. CVPR, pp 8099–8108
28. Zhang Y, Miao S, Mansi T, Liao R (2018) Task driven generative modeling for unsupervised domain adaptation: application to x-ray image segmentation. In: International conference on medical image computing and computer-assisted intervention (MICCAI), pp 599–607
29. Shrivastava A, Pfister T, Tuzel O, Susskind J, Wang W, Webb R (2017) Learning from simulated and unsupervised images through adversarial training. In: ieee conference on computer vision and pattern recognition. CVPR, pp 2242–2251
30. Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D (2017) Unsupervised pixel-level domain adaptation with generative adversarial networks. In: IEEE conference on computer vision and pattern recognition. CVPR, pp 95–104

31. Hoffman J, Tzeng E, Park T, Zhu J, Isola P, Saenko K, Efros AA, Darrell T (2018) Cycada: Cycle-consistent adversarial domain adaptation. In: International conference on machine learning (ICML), pp 1994–2003
32. Zhao H, Li H, Maurer-Stroh S, Guo Y, Deng Q, Cheng L (2018) Supervised segmentation of un-annotated retinal fundus images by synthesis. IEEE Trans Med Imaging
33. Jiang J, Hu YC, Tyagi N, Zhang P, Rimner A, Mageras GS, Deasy JO, Veeraraghavan H (2018) Tumor-aware, adversarial domain adaptation from ct to mri for lung cancer segmentation. In: MICCAI. Springer, Berlin, pp 777–785
34. Huo Y, Xu Z, Moon H, Bao S, Assad A, Moyo TK, Savona MR, Abramson RG, Landman BA (2018) Synseg-net: synthetic segmentation without target modality ground truth. IEEE Trans Med Imaging
35. Yu F, Koltun V, Funkhouser T (2017) Dilated residual networks. In: CVPR, pp 636–644
36. Milletari F, Navab N, Ahmadi SA (2016) V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). IEEE, pp 565–571
37. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: ECCV. Springer, Berlin, pp 818–833
38. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: NIPS, pp 3320–3328
39. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein gan. arXiv:170107875
40. Zhuang X, Shen J (2016) Multi-scale patch and multi-modality atlases for whole heart segmentation of mri. Med Image Anal 31:77–87
41. 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. Med Image Anal 41:40–54
42. Payer C, Štern D, Bischof H, Urschler M (2017) Multi-label whole heart segmentation using cnns and anatomical label configurations, pp 190–198
43. Jaeger S, Candemir S, Antani S, Wáng YXJ, Lu PX, Thoma G (2014) Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. Quant Imaging Med Surg 4(6):475
44. Shiraishi J, Katsuragawa S, Ikezoe J, Matsumoto T, Kobayashi T, Komatsu Ki, Matsui M, Fujita H, Kodera Y, Doi K (2000) Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. Am J Roentgenol 174(1):71–74
45. Wang L, Lai HM, Barker GJ, Miller DH, Tofts PS (1998) Correction for variations in mri scanner sensitivity in brain studies with histogram matching. Magn Reson Med 39(2):322–327
46. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. In: Advances in neural information processing systems, pp 2234–2242
47. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Computer vision and pattern recognition (CVPR), pp 3462–3471

Part II
Detection and Localization

Chapter 6
Glaucoma Detection Based on Deep Learning Network in Fundus Image

Huazhu Fu, Jun Cheng, Yanwu Xu and Jiang Liu

Abstract Glaucoma is a chronic eye disease that leads to irreversible vision loss. In this chapter, we introduce two state-of-the-art glaucoma detection methods based on deep learning technique. The first is the multi-label segmentation network, named M-Net, which solves the optic disc and optic cup segmentation jointly. M-Net contains a multi-scale U-shape convolutional network with the side-output layer to learn discriminative representations and produces segmentation probability map. Then the vertical cup to disc ratio (CDR) is calculated based on segmented optic disc and cup to assess the glaucoma risk. The second network is the disc-aware ensemble network, named DENet, which integrates the deep hierarchical context of the global fundus image and the local optic disc region. Four deep streams on different levels and modules are, respectively, considered as global image stream, segmentation-guided network, local disc region stream, and disc polar transformation stream. The DENet produces the glaucoma detection result from the image directly without segmentation. Finally, we compare two deep learning methods with other related methods on several glaucoma detection datasets.

H. Fu (✉)
Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates
e-mail: huazhufu@gmail.com

J. Cheng · J. Liu
Cixi Institute of Biomedical Engineering, Chinese Academy of Sciences, Ningbo, China
e-mail: sam.j.cheng@gmail.com

J. Liu
e-mail: jimmyliu@nimte.ac.cn

Y. Xu
Baidu Inc., Beijing, China
e-mail: ywxu@ieee.org

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_6

6.1 Introduction

Glaucoma is the leading cause of irreversible blindness worldwide [37]. Vision loss from glaucoma cannot be reversed, early detection, thus, is essential to preserve vision and life quality. Clinically, there are three main examinations practiced to screen glaucoma: intraocular pressure (IOP) measurement, function-based visual field test, and optic nerve head (ONH) assessment. IOP is an important risk factor but not specific enough to be an effective detection tool for a great number of glaucoma patients with normal tension. Function-based visual field testing requires specialized equipment, which is not widely present in the healthcare clinics. Moreover, the early glaucoma often does not have visual symptoms. By contrast, ONH assessment is a convenient way to screen glaucoma, and is widely performed by trained glaucoma specialists [23, 33]. In the fundus image, vertical cup to disc ratio (CDR) as one clinical parameter is well accepted and commonly used by clinicians [24], which is calculated by the ratio of vertical cup diameter (VCD) to vertical disc diameter (VDD), as shown in Fig. 6.1. In general, a larger CDR suggests a higher risk of glaucoma and vice versa. However, this manual ONH assessment is time consuming and costly, and it is also not suitable for large-scale screening. Thus, automated ONH assessment methods are required.

For automatic screening, segmentation-based method has been proposed first for CDR measurement, which segments the main structure (e.g., optic disc (OD) and optic cup (OC)), and then calculates the CDR value to identify the glaucoma cases [2, 6, 25]. Some automated methods measure the disc and cup from 3D optical coherence tomography (OCT) imaging [14, 15, 17]. But, OCT is not easily available due to its high cost, the fundus image is still referred to by most clinicians. For example, a superpixel segmentation method is proposed in [6], which utilizes hand-crafted features of superpixel level to extract the OD and OC region. In [8], a CDR assessment using fundus image is proposed, where a sparse dissimilarity constrained coding approach is employed to consider both the dissimilarity constraint and the sparsity constraint from a set of reference discs with known CDRs. The reconstruction coefficients are used to compute the CDR for the testing disc. These segmentation-based methods rely on the pixel level training data, and are easily affected by pathological regions and low- contrast quality. In contrast with segmentation-based method, the learning-based methods provide a direct way to screen the glaucoma from fundus image by using the various visual features with a learned classifier [3, 10, 34]. The visual features identify a wider set of image properties, some of which are unrelated to what clinicians seem to recognize as relevant. For example, Noronha et al. [34] proposed an automated glaucoma diagnosis method based on higher order spectra cumulants extracted from Radon transform applied on fundus images. Besides, Acharya et al. [1] proposed a screening method using various features extracted from Gabor transform applied on digital fundus images. The extracted visual features could explore more image relevant information, and have more representation capacity than clinical measurements. However, most existing methods, both segmentation-based and learning-based methods, are based on hand-crafted features, which lack suffi-

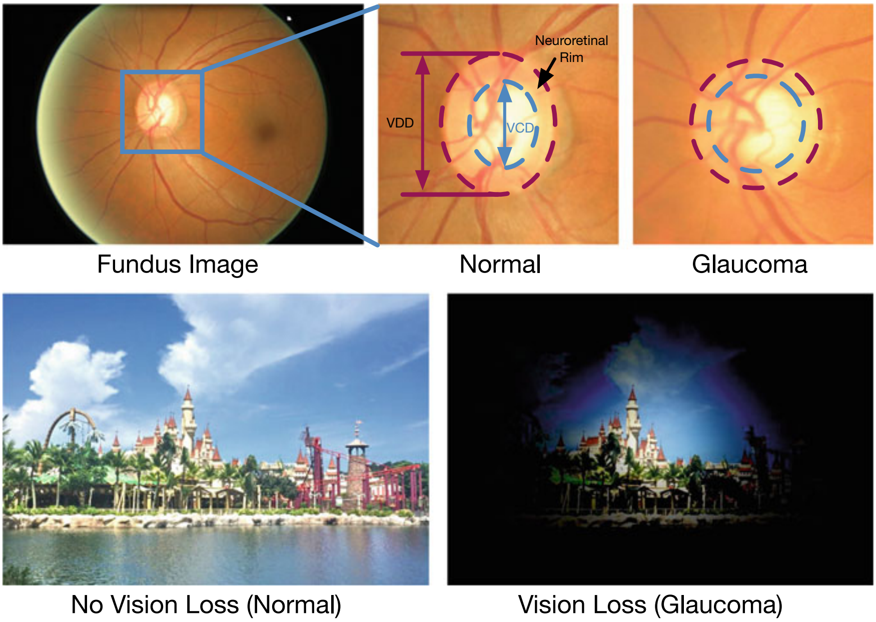


Fig. 6.1 Top: the whole fundus image and zoom in normal/glaucoma disc regions, where the vertical cup to disc ratio (CDR) is calculated by the ratio of vertical cup diameter (VCD) to vertical disc diameter (VDD). Bottom: the visual fields with normal and glaucoma cases

ciently discriminative representations and are easily affected by pathological regions and low-contrast quality.

Deep learning techniques have been recently demonstrated to yield highly representations that have aided in many computer vision tasks [5, 21, 29]. For example, Convolutional Neural Networks (CNNs) have obtained the significant improvements in image classification [26] and object segmentation [36]. For ocular image, the deep learning system also obtained the high sensitivity and specificity for detecting referable diabetic retinopathy [19]. In works [16, 18, 22, 22, 32], the deep learning systems achieved the state-of-the-art performances on fundus segmentation tasks.

In this chapter, we introduce two deep learning-based methods for glaucoma detection in fundus images.[1] The first deep network is a multi-label segmentation network [12], named M-Net, which solves the optic disc and cup segmentation jointly into a one-stage framework. In M-Net, the multi-scale layer builds an image pyramid to fed as a multi-level inputs, while the side-output layer works as the early classifier to predict the companion local prediction maps for training the early layer. Finally, a multi-label loss function is utilized to guarantee segmenting optic disc and cup jointly. For improving the segmentation result further, it also employs a polar transformation to transfer the original image to the polar coordinate system. The M-Net generates

[1] Project Page: http://hzfu.github.io/proj_glaucoma_fundus.html.

the OD and OC segmentation maps, and then calculates CDR based on segmented optic disc and cup to assess the glaucoma risk.

The second network is a disc-aware ensemble network [13], called DENet, which utilizes four deep streams on different levels and modules. The first is the global image stream, which represents the global image structure on image level and runs as a classifier on the fundus image directly. The second steam is a segmentation-guided network, which localizes the disc region from the fundus image and embeds the disc-segmentation representation to detect glaucoma. The third steam works on local disc region to predict the probability from the disc region level. The last stream focuses on disc region with polar transformation, which enlarges the disc and cup structure with the geometry operation and improves the performance. Finally, all the outputs of these four streams are fused to produce the final glaucoma screening result.

6.2 M-Net: Multi-label Segmentation Network

Figure 6.2 shows the framework of the M-Net, including multi-label segmentation network and the image polar transformation. In this method, the disc region is first localized by using the automatic disc detection method, and then the original fundus image is transferred into polar coordinate system based on the polar transformation. After that, the polar transferred image is fed into the M-Net to predict the multi-label probability maps for OD and OC regions. Finally, the inverse polar transformation recovers the segmentation map back to the Cartesian coordinate. The architecture of M-Net is an end-to-end multi-label deep network, which consists of three main parts. The first is a multi-scale U-shape convolutional network used to construct an image multi-scale input and produce a rich hierarchical feature. The second part is side-output layer that provides deeply supervision. Finally, a multi-label loss function is utilized to segment OD and OC regions jointly.

6.2.1 Multi-scale U-Shape Network

The U-Net [35] is modified as the main architecture of M-Net, which is an efficient fully convolutional neural network for the image segmentation. Similar to the U-Net architecture, M-Net consists of the encoder path (left side) and decoder path (right side). Each encoder path learns a filter bank to produce a set of encoder feature maps, and the element-wise rectified-linear nonlinearity (ReLU) activation function is employed. The decoder path also utilizes the convolution layer to output decoder feature map. The skip connections transfer the corresponding feature map from encoder path and concatenate them to up-sampled decoder feature maps.

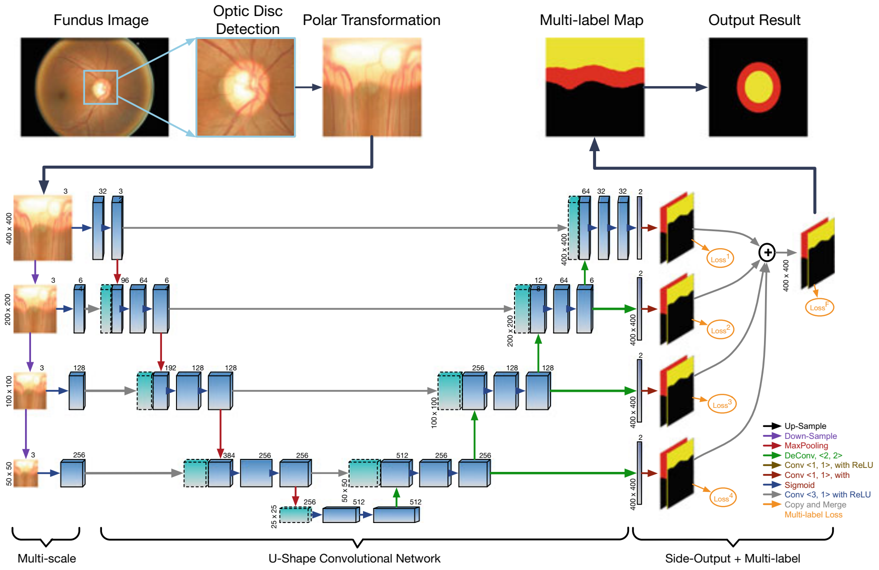


Fig. 6.2 The framework of M-Net, which mainly includes fundus polar transformation and segmentation network. After optic disc localization, the polar transformation transfer the original fundus image into the polar coordinate system based on the detected disc center. Then M-Net produces the multi-label prediction maps for disc and cup regions. The M-Net network includes multi-scale U-shape network, side-output layer, and multi-label loss. The (De)Convolutional layer parameters are denoted as "(De)Conv <kernel size, stride>"

The multi-scale input or image pyramid has been demonstrated to improve the quality of segmentation effectively. Different from the other works, which fed the multi-scale images to multi-scream networks separately and combine the final output map in the last layer [30, 31], the M-Net employs the average pooling layer to downsample the image naturally and construct a multi-scale input in the encoder path.

Finally, the high dimensional feature representation at the output of the final decoder layer is fed to a trainable multi-label classifier. In M-Net, the final classifier utilizes 1×1 convolutional layer with *Sigmoid* activation to produce the probability map. For multi-label segmentation, the output is a K channel probability map, where K is the class number ($K = 2$ for OD and OC in this work). The predicted probability map corresponds to the class with maximum probability at each pixel.

The multi-scale U-shape Network has the following advantages: (1) integrating multi-scale inputs into the decoder layers to avoid the large growth of parameters; (2) increasing the network width of decoder path.

6.2.2 Side-Output Layer

M-Net also uses the side-output layer, which acts as a early classifier to provide a companion local output map [28]. Let $\mathbf{W}$ denote the parameters of all the standard convolutional layers, and there are M side-output layers in the network, where the corresponding weights are denoted as $\mathbf{w} = (\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(M)})$. The objective function of the side-output layer is given as

$$L_{side}(\mathbf{W}, \mathbf{w}) = \sum_{m=1}^{M} \alpha_m L_{side}^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}), \tag{6.1}$$

where α_m is the loss function fusion weight for each side-output layer ($\alpha_m = 0.25$ in M-Net), M is the side-output number, and $L_{side}^{(m)}(,)$ denotes the multi-label loss of the m-th side-output layer. An average layer is used directly to combine all side-output maps as the final prediction map. The main advantages of the side-output layer are: (1) the side-output layer can be treated as a special short connection between the loss and early layer, which could relieve gradient vanishing problem and help the early layer training; (2) the multi-scale output fusion could produce a higher performance; (3) the side-output layer supervises the output map of each scale to output the better local result.

6.2.3 Multi-label Loss Function

In M-Net, the OD and OC segmentation are formulated as a multi-label problem. The existing segmentation method usually belongs to the multi-class setting, which labels each instance to one unique label. By contrast, multi-label focuses an independent binary classifier for each class, and predicts each instance to multiple binary labels. Especially for OD and OC segmentation, the disc region overlays the cup pixels, which means the pixel marked as cup also has the label as disc. Moreover, for the glaucoma cases, the disc pixels excluded cup region is as a ring shape, which makes the disc label extremely imbalance to background label under the multi-class setting. Thus, multi-label method, considering OD and OC as two independent regions, is more suitable for addressing these issues. In M-Net, a multi-label loss function based on Dice coefficient is given, which is defined as

$$L_s = 1 - \sum_{k}^{K} \frac{2w_k \sum_i^N p_{(k,i)} g_{(k,i)}}{\sum_i^N p_{(k,i)}^2 + \sum_i^N g_{(k,i)}^2}, \tag{6.2}$$

where N is the pixel number, $g_{(k,i)} \in \{0, 1\}$ and $p_{(k,i)} \in [0, 1]$ are ground truth and the predicted probability for class k, respectively. K is the class number, and $\sum_k w_k = 1$ are the class weights. The multi-label loss function in Eq. (6.2) is equivalent to the traditional Dice coefficient by setting $K = 1$. Note that the Dice loss function indicates the foreground mask overlapping ratio, and can deals with the imbalance issue in the pixels of foreground (i.e., OD or OC) region and background. Under the multi-label setting, the pixel can be labeled as OD or/and OC independently. Thus, the imbalance issue does not exist between OD and OC. w_k in Eq. (6.2) is the trade-off weight to control the contribution of OD and OC. For glaucoma screening, both the OD and OC are important, thus it sets $w_k = 0.5$. The multi-label loss function L_s can be differentiated yielding the gradient as

$$\frac{\partial L_s}{\partial p_{(k,i)}} = \sum_k^K 2w_k \left[-\frac{g_{(k,i)}}{\sum_i^N p_{(k,i)}^2 + \sum_i^N g_{(k,i)}^2} + \frac{2p_{(k,i)} \sum_i^N p_{(k,i)} g_{(k,i)}}{(\sum_i^N p_{(k,i)}^2 + \sum_i^N g_{(k,i)}^2)^2} \right]. \tag{6.3}$$

This loss is efficiently integrated into backpropagation via standard stochastic gradient descent.

6.2.4 Polar Transformation

In M-Net, the pixel-wise polar transformation is applied to transfer the original image to the polar coordinate. Let $p(u, v)$ denote the point on the original Cartesian plane, and its corresponding point on polar coordinate system is denoted $p'(\theta, r)$, as shown in Fig. 6.3, where r and θ are the radius and directional angle of the original point p, respectively. Three parameters are utilized to control the polar transformation: the disc center $O(u_o, v_o)$, the polar radius R, and the polar angle ϕ. The polar transformation could be formulated as

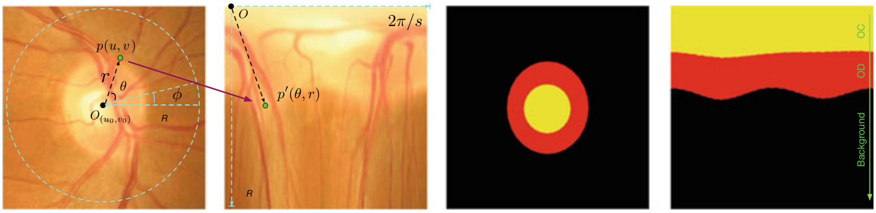
$$\begin{cases} u = u_o + r\cos(\theta + \phi), \\ v = v_o + r\sin(\theta + \phi), \end{cases} \tag{6.4}$$

and the inverse polar transformation is

$$\begin{cases} r = \sqrt{(u - u_o)^2 + (v - v_o)^2}, \\ \theta = \tan^{-1}(\dfrac{v - v_o}{u - u_o}) - \phi. \end{cases}$$

The height and width of polar image are set as the polar radius R and discretization $2\pi/s$, where s is the stride. The disc polar transformation provides a pixel-wise representation of the original image in the polar coordinate system, which has the following properties:

(1) Spatial Constraint: In the fundus image, a useful geometry constraint is that the OC should be within the OD region, as shown in Fig. 6.3a. But this redial relationship is hard to implement in the Cartesian coordinate. By contrast, the polar transformation transfers this redial relationship to a spatial relationship, where the regions of



(a) Fundus Image on Cartesian Coordinate

(b) Fundus Image on Polar Coordinate

(c) Ground truth on Cartesian Coordinate

(d) Ground truth on Polar Coordinate

Fig. 6.3 Illustration of the mapping from Cartesian coordinate system (**a**) to the polar coordinate system (**b**) by using the polar transformation. The point $p(u, v)$ in Cartesian coordinate corresponds to the point $p'(\theta, r)$ in polar coordinate. (**c**) and (**d**) are the corresponding ground truth, where yellow, red, and black regions denote the optic cup, optic disc, and background, respectively

cup, disc, and background appear the ordered layer structure, as shown in Fig. 6.3d. This layer-like spatial structure is convenient to use, especially some layer-based segmentation methods [11, 27] can be employed as the post-processing.

(2) Equivalent Augmentation: Since the polar transformation is a pixel-wise mapping, the polar transformation is equivalent to data augmentation. For example, moving the expansion center $O(u_o, v_o)$ is equivalent to the drift cropping transformations on polar coordinate. Using different transformation radius R is same as augmenting with the various scaling factor. Thus the data augmentation for deep learning could be done well during the polar transformation with various parameters.

(3) Balancing Cup Proportion: In the fundus image, the distribution of OC and background pixels is heavily imbalanced. Even in the cropped ROI, the cup region still accounts for a low proportion. Using Fig. 6.3c as an example, the cup region only occupies about 4%. This extremely imbalance proportion easily leads the bias and overfitting in training the deep model. The polar transformation flat the image based on OD center, that could enlarge the cup region by using interpolation and increase the OC proportion. As shown in Fig. 6.3d, the ratio of cup region increases to 23.4% over the ROI, which is more balanced than that in original fundus image, which could help avoid the overfitting during the model training and improve the OC segmentation further.

6.3 DENet: Disc-Aware Ensemble Network

The second glaucoma screening method is Disc-aware Ensemble Network (DENet), which detects the glaucoma from the fundus image directly by using the classification way. DENet takes into account both global and local levels of fundus image information, as shown in Fig. 6.4. The global image level provides the coarse structure representation on the whole image, while the local disc region is utilized to learn a fine representation around the disc region.

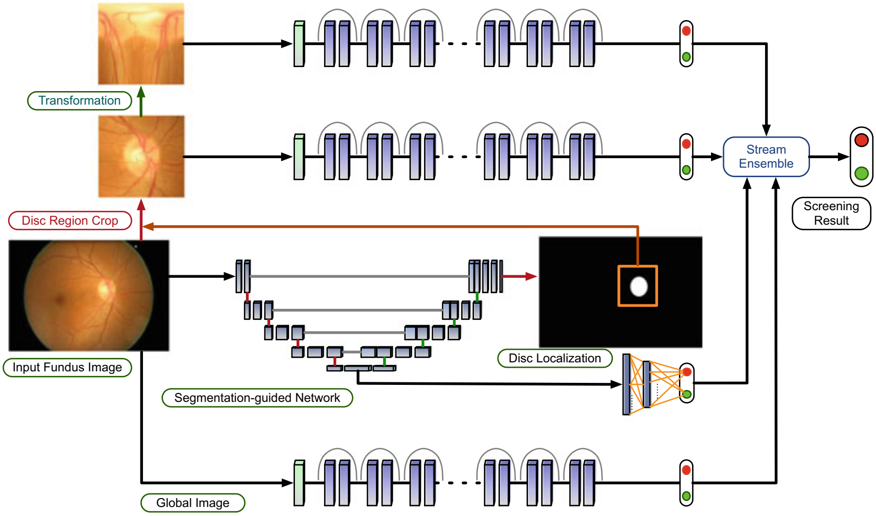


Fig. 6.4 Architecture of DENet, which contains four streams: global image stream produces the result based on the whole fundus image; the segmentation-guided network detects the optic disc region and predicts a probability embedded the disc-segmentation representation; disc region stream works on disc region cropped by disc-segmentation map from segmentation-guided network; disc polar stream transfers the disc region image into the polar coordinate system. Then these four stream outputs are fused as the final glaucoma screening result

6.3.1 Global Fundus Image Level

In the DENet, two streams are employed on the global fundus image level. The first stream is a standard classification network by using Residual Network (ResNet) [20], which employs the shortcut connection on a Convolutional Neural Network to handle the vanishing gradient problem. DENet utilizes the ResNet-50 as the backbone network to learn the global representation on the whole fundus image, which consists of five down-sampling blocks, followed by a global max-pooling layer and a fully connected (FC) layer for glaucoma screening. The input image of this stream is resized to 224×224 to enable use of pretrained model in [20] as initialization for this network.

The second global level stream is the segmentation-guided network, which detects the optic disc region and produces a screening result based on the disc-segmentation representation. As shown in Fig. 6.4, the main architecture of the segmentation-guided network is adapted by the U-shape convolutional network (U-Net) [35]. Similar to the original U-Net architecture, DENet consists of the encoder path (left side) and decoder path (right side). A new branch is extended from the saddle layer of the U-shape network, where the size scale is the smallest (i.e., 40×40) and the number of channels is the highest (i.e., 512D). The extended branch acts as an implicit vector with average pooling and flatten layers. Next, it connects two fully connected layers to produce a glaucoma classification probability. This pipeline embeds the segmentation-guided representation through the convolutional filters on decoder path of the U-shape network. The input image of this stream is resized to 640×640, which guarantees that the image has enough details to localize disc region accurately.

In the global fundus image level networks, two loss functions are employed. The first one is the binary cross entropy loss function for glaucoma detection layer. The other is the Dice coefficient for assessing disc segmentation [9], which is defined as

$$L_{Dice} = 1 - \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}, \tag{6.5}$$

where N is the pixel number, $p_i \in [0, 1]$ and $g_i \in \{0, 1\}$ denote predicted probability and binary ground truth label for disc region, respectively. The Dice coefficient loss function can be differentiated yielding the gradient as

$$\frac{\partial L_{Dice}}{\partial p_i} = \frac{4p_i \sum_i^N p_i g_i - 2g_i\left(\sum_i^N p_i^2 + \sum_i^N g_i^2\right)}{\left(\sum_i^N p_i^2 + \sum_i^N g_i^2\right)^2}. \tag{6.6}$$

These two losses are efficiently integrated into backpropagation via standard stochastic gradient descent (SGD).

Note that DENet uses two phases for training the segmentation-guided model. First, the U-Net for disc detection is trained by pixel-level disc data with Dice coefficient loss. Then the parameters of CNN layers are frozen and the fully connected layers for the classification task are trained by using glaucoma detection data. The separate phases train the segmentation-guided model instead of the multitask-based single-stage training, with the following reasons: (1) Using the disc-segmentation representation on screening could add diversity of the proposed network. (2) The pixel-level label data for disc segmentation is more expensive than image-level label data for glaucoma detection. The separate training stages could employ different training datasets and configuration (e.g., different batch sizes and image numbers). (3) The extracted disc region by network influences the follow-up stream, thus the accuracy of disc detection is more important than classification branch.

6.3.2 Optic Disc Region Level

The second level in DENet is based on the local optic disc region, which is cropped based on the previous segmentation-guided network. The disc region preserves more detailed information with high resolution and it is benefited to learn a fine representation. Two local streams are employed to learn representations on the local disc region. The first one is a standard classification network based on ResNet [20] on the original local disc region, as shown in Fig. 6.4, while the other stream focuses on the disc polar transformation.

6.4 Experiments

6.4.1 Implementation

M-Net and DENet are both implemented with Python based on Keras with Tensorflow backend. M-Net employs stochastic gradient descent (SGD) for optimizing, and uses a gradually decreasing learning rate from 0.0001 and a momentum of 0.9. The transformation radius R is set to $R = 400$, and the directional angles are divided into 400 bins, thus the size of transferred polar image is 400×400. A two-channel probability map is produced, and the fixed threshold 0.5 is used to get binary mask for OD and OC region. Finally, the fitted ellipse fit on largest connected region in OD/OC mask is used as the final segmentation result.

TheDENet is implemented with Python based on Keras with Tensorflow backend. The four streams in DENet system are trained separately according to different data augmentation strategies, which dues to three reasons: (1) The disc local streams are based on the disc detection result of global image stream. (2) The separate training stage is convenient to add a new stream into the ensemble network. (3) The separate training stage could employ different training datasets and configuration for different stream. For the global image stream and segmentation-guided network, the data augmentation is done on the training set by random rotations (0/90/180/270 degrees) and random flips. For the local disc region scream, the data augmentation is done by random rotations (0/90/180/270 degrees), crop drift (±20 pixels) and random flips. For the disc polar region, we tune the polar transformation parameters to control the data augmentation, by polar angle ($\phi = 0/90/180/270$ degrees), polar center drift ($C_{(u_o,v_o)} \pm 20$), and polar radius ($R = 400 \times \{0.8, 1\}$). During training, we employ stochastic gradient descent (SGD) for optimizing the deep model. We use a gradually decreasing learning rate starting from 0.0001 and a momentum of 0.9. ResNet-50 stream employs pretrained parameters based on ImageNet as initialization, and all the layers are fine-tuned during the training.

6.4.2 Segmentation Evaluation

We first evaluate the OD and OC segmentation of M-Net. The ORIGA dataset [40] is used, which contains 650 fundus images with 168 glaucomatous eyes and 482 normal eyes. The 650 images with manual ground truth boundaries are divided into 325 training images (including 73 glaucoma cases) and 325 testing images (including 95 glaucomas) as same as that in [8, 38]. The overlapping error (E) and balanced accuracy (A) is utilized to evaluate the segmentation performance as

$$E = 1 - \frac{Area(S \bigcap G)}{Area(S \bigcup G)},\ A = \frac{1}{2}(Sen + Spe), \tag{6.7}$$

Table 6.1 Performance comparisons (%) of the different methods on ORIGA Dataset. (PT: Polar Transformation)

Method	E_{disc}	A_{disc}	E_{cup}	A_{cup}	δ_E
R-Bend [25]	0.129	–	0.395	–	0.154
ASM [39]	0.148	–	0.313	–	0.107
Superpixel [6]	0.102	0.964	0.264	0.918	0.077
LRR [38]	–	–	0.244	–	0.078
U-Net [35]	0.115	0.959	0.287	0.901	0.102
Joint U-Net	0.108	0.961	0.285	0.913	0.083
M-Net	0.083	0.972	0.256	0.914	0.078
Joint U-Net + PT	0.072	0.979	0.251	0.914	0.074
M-Net + PT	**0.071**	**0.983**	**0.230**	**0.930**	**0.071**

with

$$Sen = \frac{TP}{TP + FN}, \ Spe = \frac{TN}{TN + FP}, \tag{6.8}$$

where S and G denote the segmented mask and the manual ground truth, respectively. TP and TN denote the number of true positives and true negatives, respectively, and FP and FN denote the number of false positives and false negatives, respectively. Moreover, we also calculate the absolute CDR error δ_E as evaluation metric, which is defined as $\delta_E = |CDR_S - CDR_G|$, where CDR_G and CDR_S denote the manual CDR from trained clinician and by the segmented result, respectively.

We compare the M-Net with the several state-of-the-art segmentation methods, including relevant-vessel bends (R-Bend) method in [25], active shape model (ASM) method in [39], superpixel-based classification (Superpixel) method in [6], quadratic divergence regularized SVM (QDSVM) method in [7], and low-rank superpixel representation (LRR) method in [38]. Additional, we compare with the U-Net [35]. We report two outputs of U-Net, the original U-Net for segmenting OC and OD separately and U-Net utilized the multi-label loss function (Joint U-Net) for segmenting OC and OD jointly. We also provide segmentation results with/without the polar transformation (PT). The performances are shown in Table 6.1.

R-Bend [25] provides a parameterization technique based on vessel bends, and ASM [39] employs the circular Hough transform initialization to segment the OD and OC regions. These bottom-up methods segment the OD and OC separately, which do not perform well on the ORIGA dataset. Superpixel method [6] utilizes superpixel-level classifier to identify the OD and OC regions, which obtains a better result than the bottom-up methods [25, 39]. The methods LRR [38] and QDSVM [7] obtain good performances. But, they only focus on individual OD or OC region segmentation, and could not calculate the CDR value for detecting glaucoma. Joint U-Net with the multi-label loss constrains the mutual relation of OD and OC regions, and produces a better result than that in pure U-Net [35]. The M-Net with multi-scale

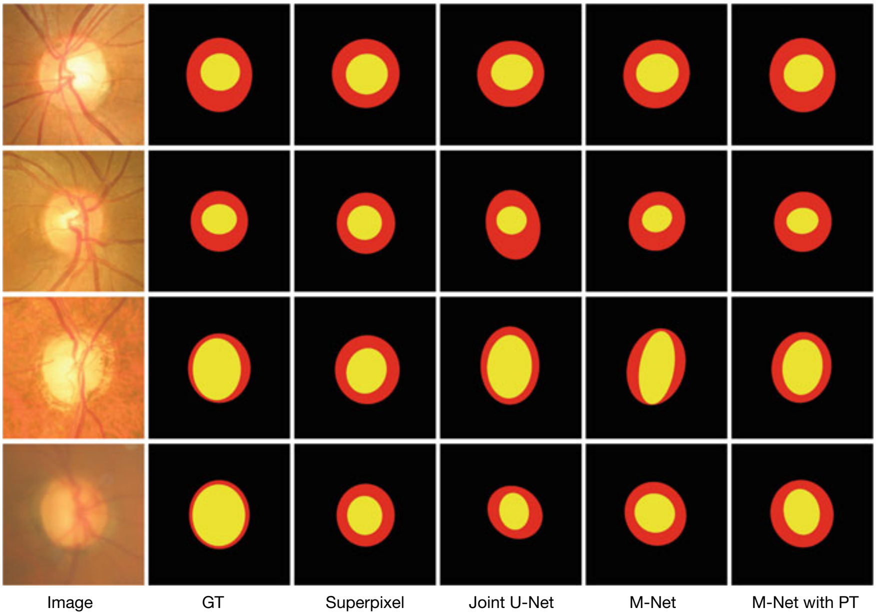


Fig. 6.5 The visual results of OD and OC segmentation, where the yellow and red region denote the cup and disc regions, respectively. From the left to right: fundus image, ground truth (GT), Joint U-Net, the M-Net, and M-Net with polar transformation (PT). The last row shows the failed case

input and side-output layers obtains the higher score than single-scale network and superpixel method [6], which demonstrates that the multi-scale input and side-output layers are useful to guide the early layer training.

The polar transformation as one contribution of the M-Net work solves the imbalance issue of OC and OD region. One major advantage is that the polar transformation augments the proportion of cup region, and makes the area of the disc/cup and background more balance. The balanced region helps avoid the overfitting during the model training and improves the segmentation performance further. In Table 6.1, polar transformation reduces about 0.03 in Joint U-Net and 0.02 in M-Net on E_{cup} scores. Note that the performance of Joint U-Net with PT is slightly better than that in M-Net without PT. It shows that the gains of the polar transformation may be higher than that using multi-scale input and side-output layers. Finally, the M-Net with PT obtains the best performance, and outperforms other state-of-the-art methods.

Figure 6.5 shows some visual examples of the segmentation results, where the first two rows are normal eyes and the rest rows are glaucoma cases. For the superpixel method [6], the segmented OC is smaller than ground truth in glaucoma cases, which may cause an underestimated CDR. The deep learning methods (e.g., Joint U-Net and M-Net) obtain more accurate cup boundary, but it easily generates a larger OD. By contrast, the M-Net with PT can effectively and accurately segment OD and OC regions. The last row in Fig. 6.5 gives a challenging case for segmentation, where the image is blurred and has low contrast for identifying the OC boundary. For this case, all the methods fail to produce accurate OC segmentation. This issue could potentially be addressed in future work through the use of more powerful network or additional image enhancement preprocessing.

6.4.3 Glaucoma Screening Evaluation

In these experiments, we use three glaucoma screening datasets to evaluate the glaucoma screening performances. The first one is the ORIGA dataset, which is used in segmentation experiment, including 168 glaucomatous eyes and 482 normal eyes. The second is the Singapore Chinese Eye Study (SCES) dataset, which consists of 1676 images with 46 glaucoma cases. The third dataset is a population-based study conducted, which includes a total of 5783 eye images with 113 glaucomatous eyes and 5670 normal eyes. Because only the ORIGA dataset has the optic disc and cup ground truth, we use all the 650 images in ORIGA dataset for network training including disc and cup segmentation and glaucoma screening. Then we employ the SCES and new collected datasets for screening test. The resolution of the fundus image in these three datasets are 3072 × 2048, and the size of cropped disc region is 800 × 800.

For evaluation, the Receiver Operating Characteristic (ROC) curves and the area under ROC curve (AUC) are reported. Moreover, three evaluation criteria are also employed to measure performances, as Sensitivity (Sen), Specificity (Spe), and Balanced Accuracy (BAcc):

$$\text{Sen} = \frac{TP}{TP + FN}, \text{Spe} = \frac{TN}{TN + FP}, \text{BAcc} = \frac{\text{Sen} + \text{Spe}}{2},$$

where TP and TN denotes the number of true positives and true negatives, respectively, and FP and FN denotes the number of false positives and false negatives, respectively. Tuning the diagnostic thresholds could obtain a series of criteria scores, and then we use the threshold with highest B-Accuracy score as the final threshold to report the performance. We compare M-Net and DENet with four state-of-the-art glaucoma screening baselines: wavelet-based feature method (Wavelet) in [10], Gabor transformation method (Gabor) in [1], Glaucoma risk index method (GRI) in [3], and superpixel-based classification method (Superpixel) in [6]. We also additionally provide the intraocular pressure (IOP) measurement result as the clinical baseline. The experiment results are reported in Table 6.2 and Figs. 6.6 and 6.7.

From the glaucoma screening results, the IOP performs poorly with 0.66 AUC on the SECS dataset and 0.6233 AUC on the new collected dataset. The energy property of the wavelet transformed image is utilized in [10], which does not provide enough discriminative capability for glaucoma screening. By contrast, SRI in [3] fuses multiple image features (e.g., intensity value, FFT coefficient, and B-spline coefficient) and obtains the better scores than Wavelet in [10]. The non-deep learning method,

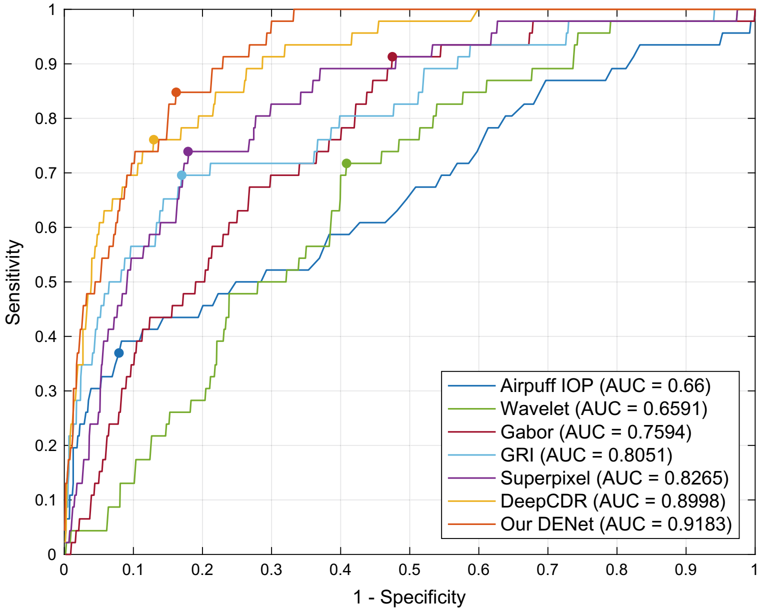


Fig. 6.6 The ROC curves with AUC scores for glaucoma screening on SCES dataset

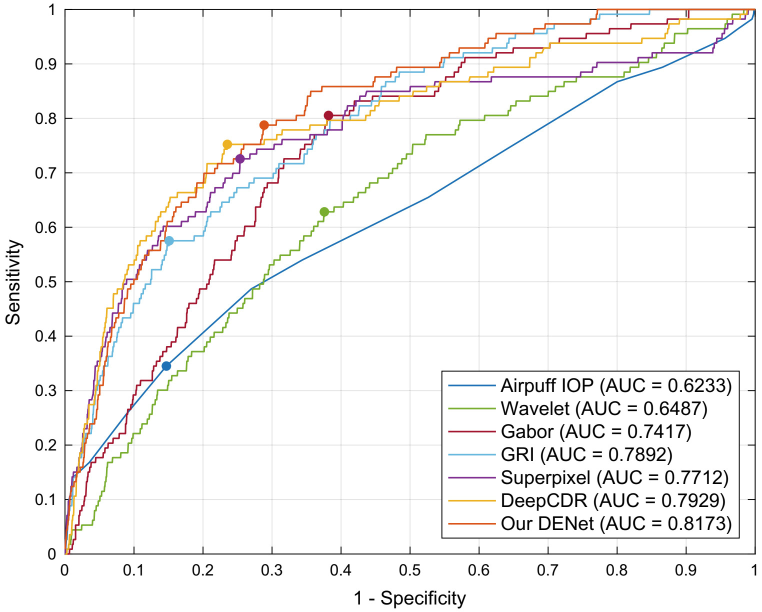


Fig. 6.7 The ROC curves with AUC scores for glaucoma screening on new collected dataset

Table 6.2 Performance comparisons of the different methods on datasets

	SCES dataset				New collected dataset			
Method	AUC	B-Acc	Sen	Spe	AUC	B-Acc	Sen	Spe
Airpuff IOP	0.6600	0.6452	0.3696	0.9209	0.6233	0.5991	0.3451	0.8531
Wavelet [10]	0.6591	0.6544	0.7174	0.5914	0.6487	0.6262	0.6283	0.6242
Gabor [1]	0.7594	0.7191	0.9130	0.5252	0.7417	0.7117	0.8053	0.6182
GRI [3]	0.8051	0.7629	0.6957	0.8301	0.7892	0.7121	0.5752	0.8490
Superpixel [6]	0.8265	0.7800	0.7391	0.8209	0.7712	0.7360	0.7257	0.7464
M-Net	0.8998	0.8157	0.7609	0.8706	0.7929	0.7585	0.7522	0.7647
DENet	0.9183	0.8429	0.8478	0.8380	0.8173	0.7495	0.7876	0.7115

Superpixel [6], produces a competitive performance on the SCES dataset (0.8265 AUC) and the new collected dataset (0.7712 AUC), which is better than IOP. In the deep learning method, M-Net, achieves the higher performances on both datasets, which demonstrates the capability of deep learning technique. The DENet obtains the best performances on the SCES dataset (0.9183 AUC) and the new collected dataset (0.8173 AUC). It can be seen that without extracting clinical parameters (e.g., CDR), the visual features could be used for glaucoma screening. One possible reason is that the clinical parameters are based on what information clinicians currently observe, while visual features deeply represent a wider set of image properties, some of which may be unrelated with what clinicians defined explicitly. Hence, visual features gain more latent image representations, and more discriminative information than clinical parameters. Moreover, the DENet also outperforms other deep learning based methods. For example, the deep learning method in [4] provides a glaucoma screening system by using CNN feature on the disc region directly, which obtained 0.898 AUC on the SCES dataset. The DENet is comparable to that of the deep system [4], and is also able to localize the disc region from the whole fundus image.

6.4.4 *REFUGE Challenge*

Retinal Fundus Glaucoma Challenge[2] (REFUGE) in conjunction with MICCAI 2018 conference is a glaucoma challenge, which consists of two tasks, namely optic cup/disc segmentation and glaucoma classification. We load the model parameters of MNet and DENet trained on ORIGA dataset directly without any fine-tuning, and test them on the REFUGE training set. The results are shown in Fig. 6.8.

From the result, the M-Net obtains the better performance than DENet. One of the main reasons for that is the REFUGE images are centered at macula, which is different from the disc-center fundus image in ORIGA dataset. The effect of different views on

[2] https://refuge.grand-challenge.org.

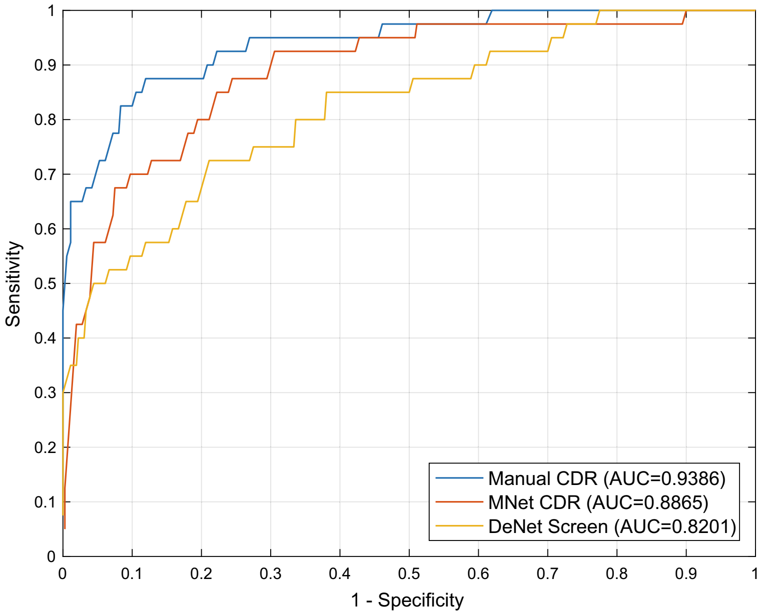


Fig. 6.8 The ROC curves with AUC scores for glaucoma screening on REFUGE training set

segmentation-based method is much less than that in learning-based method. Based on this observation, we can conclude that when there has enough training data with the similar image distribution, the learning-based method for glaucoma screening could obtain the better performance. And if the training data has the different distribution with the target data, the segmentation-based method appears better robustness.

6.5 Conclusion

In this chapter, we have evaluated two kinds of deep learning based method for automated glaucoma screening, M-Net and DENet. The M-Net is a segmentation-based method, which solves the OD and OC segmentation jointly into a one-stage multi-label framework. The DENet is a learning-based method, which integrates four deep streams on different levels and modules and predicts glaucoma from the fundus image directly. The experiments on several glaucoma datasets show that the two methods obtain the satisfied performances. The codes of both M-Net and DENet details are available.[3]

[3] Project Page: http://hzfu.github.io/proj_glaucoma_fundus.html.

References

1. Acharya UR, Ng EY, Eugene LWJ, Noronha KP, Min LC, Nayak KP, Bhandary SV (2015) Decision support system for the glaucoma using Gabor transformation. Biomed Signal Process Control 15:18–26
2. Almazroa A, Burman R, Raahemifar K, Lakshminarayanan V (2015) Optic disc and optic cup segmentation methodologies for glaucoma image detection: a survey. J Ophthalmol 2015
3. Bock R, Meier J, Nyul LG, Hornegger J, Michelson G (2010) Glaucoma risk index: automated glaucoma detection from color fundus images. Med. Image Anal 14(3):471–481
4. Chen X, Xu Y, Yan S, Wing D, Wong T, Liu J (2015) Automatic feature learning for glaucoma detection based on deep learning. In: Proceedings of MICCAI, pp 669–677
5. Cheng J, Li Z, Gu Z, Fu H, Wong DWK, Liu J (2018) Structure-preserving guided retinal image filtering and its application for optic disc analysis. IEEE Trans Med Imaging 37(11):2536–2546
6. Cheng J, Liu J, Xu Y, Yin F, Wong D, Tan N, Tao D, Cheng CY, Aung T, Wong T (2013) Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. IEEE Trans Med Imaging 32(6):1019–1032
7. Cheng J, Tao D, Wong DWK, Liu J (2017) Quadratic divergence regularized SVM for optic disc segmentation. Biomed Opt Express 8(5):2687–2696
8. Cheng J, Zhang Z, Tao D, Wong D, Liu J, Baskaran M, Aung T, Wong T (2017) Similarity regularized sparse group lasso for cup to disc ratio computation. Biomed Opt Express 8(8):1192–1205
9. Crum WR, Camara O, Hill DLG (2006) Generalized overlap measures for evaluation and validation in medical image analysis. IEEE Trans Med Imaging 25(11):1451–1461
10. Dua S, Rajendra Acharya U, Chowriappa P, Vinitha Sree S (2012) Wavelet-based energy features for glaucomatous image classification. IEEE Trans Inform Technol Biomed 16(1):80–87
11. Dufour PA, Ceklic L, Abdillahi H, Schröder S, De Dzanet S, Wolf-Schnurrbusch U, Kowal J (2013) Graph-based multi-surface segmentation of oct data using trained hard and soft constraints. IEEE Trans Med Imaging 32(3):531–543
12. Fu H, Cheng J, Xu Y, Wong D, Liu J, Cao X (2018) Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. IEEE Transa Med Imaging 37(7):1597–1605
13. Fu H, Cheng J, Xu Y, Zhang C, Wong DWK, Liu J, Cao X (2018) Disc-aware ensemble network for glaucoma screening from fundus image. IEEE Trans Med Imaging 37(11):2493–2501
14. Fu H, Xu D, Lin S, Wong DWK, Liu J (2015) Automatic optic disc detection in OCT slices via low-rank reconstruction. IEEE Trans Biomed Eng 62(4):1151–1158
15. Fu H, Xu Y, Lin S, Wong D, Mani B, Mahesh M, Aung T, Liu J (2018) Multi-context deep network for angle-closure glaucoma screening in anterior segment OCT. In: International conference on medical image computing and computer assisted intervention (MICCAI), pp 356–363
16. Fu H, Xu Y, Lin S, Wong DWK, Liu J (2016) DeepVessel: retinal vessel segmentation via deep learning and conditional random field. In: Proceedings of MICCAI, pp 132–139
17. Fu H, Xu Y, Lin S, Zhang X, Wong D, Liu J, Frangi A (2017) Segmentation and quantification for angle-closure glaucoma assessment in anterior segment OCT. IEEE Trans Med Imaging 36(9):1930–1938
18. Fu H, Xu Y, Wong D, Liu J (2016) Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. In: Proceedings of ISBI, pp 698–701
19. Gulshan V, Peng L, Coram M et al (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. J Am Med Assoc 304(6):649–656
20. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of CVPR, pp 770–778
21. Liu H, Wong DWK, Fu H, Xu Y, Liu J (2018) DeepAMD: detect early age-related macular degeneration by applying deep learning in a multiple instance learning framework. In: Asian conference on computer vision (ACCV)

22. Jiang Y, Xia H, Xu Y, Cheng J, Fu H, Duan L, Meng Z, Liu J (2018) Optic disc and cup segmentation with blood vessel removal from fundus images for glaucoma detection. In: IEEE engineering in medicine and biology conference (EMBC)
23. Jonas J, Budde W, Panda-Jonas S (1999) Ophthalmoscopic evaluation of the optic nerve head. Surv Ophthalmol 43(4):293–320
24. Jonas JB, Bergua A, Schmitz-Valckenberg P, Papastathopoulos KI, Budde WM (2000) Ranking of optic disc variables for detection of glaucomatous optic nerve damage. Invest Ophthalmol Vis Sci 41(7):1764–1773
25. Joshi GD, Sivaswamy J, Krishnadas SR (2011) Optic disk and cup segmentation from monocular colour retinal images for glaucoma assessment. IEEE Trans Med Imaging 30(6):1192–1205
26. Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of NIPS, pp 1097–1105
27. Lang A, Carass A, Hauser M, Sotirchos ES, Calabresi PA, Ying HS, Prince JL (2013) Retinal layer segmentation of macular oct images using boundary classification. Biomed Opt Express 4(7):1133–1152
28. Lee C, Xie S, Gallagher P, Zhang Z, Tu Z (2015) Deeply-supervised nets. In: International conference on artificial intelligence and statistics
29. Li C, Guo J, Porikli F, Fu H, Pang Y (2018) A cascaded convolutional neural network for single image dehazing. IEEE Access 6:24877–24887
30. Li G, Yu Y (2016) Visual saliency detection based on multiscale deep cnn features. IEEE Trans Image Process 25(11):5012–5024
31. Liu Y, Cheng MM, Hu X, Wang K, Bai X (2017) Richer convolutional features for edge detection. In: Proceedings of CVPR
32. Maninis K, Pont-Tuset J, Arbelaez P, Gool LV (2016) Deep retinal image understanding. In: Proceedings of MICCAI, pp 140–148
33. Morgan JE, Sheen NJL, North RV, Choong Y, Ansari E (2005) Digital imaging of the optic nerve head: monoscopic and stereoscopic analysis. Br J Ophthalmol 89(7):879–884
34. Noronha KP, Acharya UR, Nayak KP, Martis RJ, Bhandary SV (2014) Automated classification of glaucoma stages using higher order cumulant features. Biomed Signal Process Control 10(1):174–183
35. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of MICCAI, pp 234–241
36. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651
37. Tham YC, Li X, Wong TY, Quigley HA, Aung T, Cheng CY (2014) Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis. Ophthalmology 121(11):2081–2090
38. Xu Y, Duan L, Lin S, Chen X, Wong D, Wong T, Liu J (2014) Optic cup segmentation for glaucoma detection using low-rank superpixel representation. In: Proceedings of MICCAI
39. Yin F, Liu J, Ong SH, Sun Y, Wong DWK, Tan NM, Cheung C, Baskaran M, Aung T, Wong TY (2011) Model-based optic nerve head segmentation on retinal fundus images. In: Proceedings of EMBC, pp 2626–2629
40. Zhang Z, Yin F, Liu J, Wong W, Tan N, Lee B, Cheng J, Wong T (2010) ORIGA(-light): an online retinal fundus image database for glaucoma analysis and research. In: Proceedings of EMBC, pp 3065–3068

Chapter 7
Thoracic Disease Identification and Localization with Limited Supervision

Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li and Li Fei-Fei

Abstract Accurate identification and localization of abnormalities from radiology images play an integral part in clinical diagnosis and treatment planning. Building a highly accurate prediction model for these tasks usually requires a large number of images manually annotated with labels and finding sites of abnormalities. In reality, however, such annotated data are expensive to acquire, especially the ones with location annotations. We need methods that can work well with only a small amount of location annotations. To address this challenge, we present a unified approach that simultaneously performs disease identification and localization through the same underlying model for all images. We demonstrate that our approach can effectively leverage both class information as well as limited location annotation, and significantly outperforms the comparative reference baseline in both classification and localization tasks.

Z. Li (✉)
Department of Electrical Engineering and Computer Science, Syracuse University,
900 South Crouse Ave., Syracuse, NY 13210, USA
e-mail: zli89@syr.edu

M. Han
PAII Inc., Palo Alto Research Lab, 3000 El Camino Real, 5 Palo Alto Square, Ste 150,
Palo Alto, CA 94306, USA
e-mail: hanmei613@paii-labs.com

C. Wang · Y. Xue · W. Wei · L.-J. Li · L. Fei-Fei
Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
e-mail: chongw@google.com

Y. Xue
e-mail: yuanxue@google.com

W. Wei
e-mail: wewei@google.com

L.-J. Li
e-mail: lijiali@google.com

L. Fei-Fei
e-mail: feifeili@google.com

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_7

7.1 Introduction

Automatic image analysis is becoming an increasingly important technique to support clinical diagnosis and treatment planning. It is usually formulated as a classification problem where medical imaging abnormalities are identified as different clinical conditions [4, 26, 27, 29, 35]. In clinical practice, visual evidence that supports the classification result, such as spatial localization [2] or segmentation [36, 39] of sites of abnormalities is an indispensable part of clinical diagnosis which provides interpretation and insights. Therefore, it is of vital importance that the image analysis method is able to provide both classification results and the associated visual evidence with high accuracy.

Figure 7.1 is an overview of our approach. We focus on chest X-ray image analysis. Our goal is to both classify the clinical conditions and identify the abnormality locations. A chest X-ray image might contain multiple sites of abnormalities with monotonous and homogeneous image features. This often leads to the inaccurate classification of clinical conditions. It is also difficult to identify the sites of abnormalities because of their variances in the size and location. For example, as shown in Fig. 7.2, the presentation of "Atelectasis" (alveoli are deflated down) is usually limited to local regions of a lung [11] but possible to appear anywhere on both sides

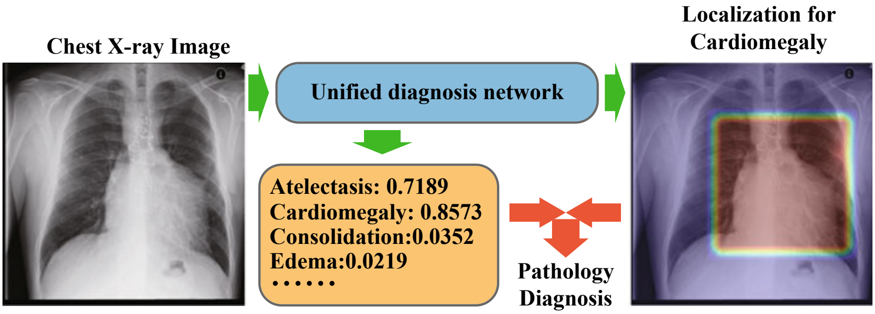


Fig. 7.1 Overview of our chest X-ray image analysis network for thoracic disease diagnosis. The network reads chest X-ray images and produces prediction scores and localization for the diseases

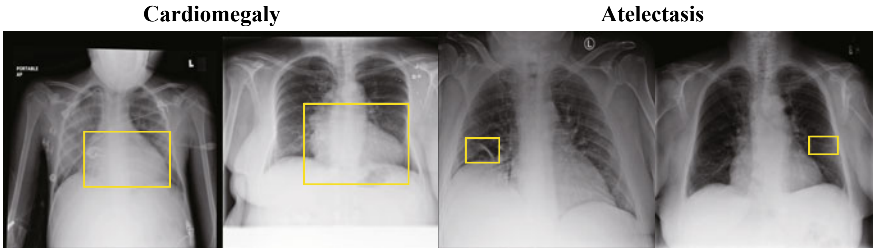


Fig. 7.2 Examples of chest X-ray images with the disease bounding box. The disease regions are annotated in the yellow bounding boxes by radiologists

of lungs; while “Cardiomegaly” (enlarged heart) always covers half of the chest and is always around the heart.

The lack of large-scale datasets also stalls the advancement of automatic chest X-ray diagnosis. Wang et al. provides one of the largest publicly available chest x-ray datasets with disease labels[1] along with a small subset with region-level annotations (bounding boxes) for evaluation [30].[2] As we know, the localization annotation is much more informative than just a single disease label to improve the model performance as demonstrated in [20]. However, getting detailed disease localization annotation can be difficult and expensive. Thus, designing models that can work well with only a small amount of localization annotation is a crucial step for the success of clinical applications.

In this chapter, we present a unified approach that simultaneously improves disease identification and localization with only a small amount of X-ray images containing disease location information. Figure 7.1 demonstrates an example of the output of our model. Unlike the standard object detection task in computer vision, we do not strictly predict bounding boxes. Instead, we produce regions that indicate the diseases, which aligns with the purpose of visualizing and interpreting the disease better. First, we apply an CNN to the input image so that the model learns the information of the entire image and implicitly encodes both the class and location information for the disease [23]. We then slice the image into a patch grid to capture the local information of the disease. For an image with bounding box annotation, the learning task becomes a fully supervised problem since the disease label for each patch can be determined by the overlap between the patch and the bounding box. For an image with only a disease label, the task is formulated as a multiple-instance learning (MIL) problem [3]—at least one patch in the image belongs to that disease. If there is no disease in the image, all patches have to be disease-free. In this way, we have unified the disease identification and localization into the same underlying prediction model but with two different loss functions.

We evaluate the model on the aforementioned chest X-ray image dataset provided in [30]. Our quantitative results show that the proposed model achieves significant accuracy improvement over the published state of the art on both disease identification and localization, despite the limited number of bounding box annotations of a very small subset of the data. In addition, our qualitative results reveal a strong correspondence between the radiologist’s annotations and detected disease regions, which might produce further interpretation and insights of the diseases.

[1] While abnormalities, findings, clinical conditions, and diseases have distinct meanings in the medical domain, here, we simply refer to them as diseases and disease labels for the focused discussion in computer vision.

[2] The method proposed in [30] did not use the bounding box information for localization training.

7.2 Related Work

Object detection. Following the R-CNN work [9], recent progresses has focused on processing all regions with only one shared CNN [8, 12], and on eliminating explicit region proposal methods by directly predicting the bounding boxes. In [24], Ren et al. developed a region proposal network (RPN) that regresses from anchors to regions of interest (ROIs). However, these approaches could not be easily used for images without enough annotated bounding boxes. To make the network process images much faster, Redmon et al. proposed a grid-based object detection network, YOLO, where an image is partitioned into $S \times S$ grid cells, each of which is responsible to predict the coordinates and confidence scores of B bounding boxes [23]. The classification and bounding box prediction are formulated into one loss function to learn jointly. A step forward, Liu et al. partitioned the image into multiple grids with different sizes proposing a multi-box detector overcoming the weakness in YOLO and achieved better performance [21]. Similarly, these approaches are not applicable for the images without bounding boxes annotation. Even so, we still adopt the idea of handling an image as a group of grid cells and treat each patch as a classification target.

Medical disease diagnosis. Zhang et al. proposed a dual-attention model using images and optional texts to make accurate prediction [34]. In [35], Zhang et al. proposed an image-to-text model to establish a direct mapping from medical images to diagnostic reports. Both models were evaluated on a dataset of bladder cancer images and corresponding diagnostic reports. Wang et al. took advantage of a large-scale chest X-ray dataset to formulate the disease diagnosis problem as multi-label classification, using class-specific image feature transformation [30]. They also applied a thresholding method to the feature map visualization [33] for each class and derived the bounding box for each disease. Their qualitative results showed that the model usually generated much larger bounding box than the ground truth. Hwang et al. [16] proposed a self-transfer learning framework to learn localization from the globally pooled class-specific feature maps supervised by image labels. These works have the same essence with class activation mapping [37] which handles natural images. The location annotation information was not directly formulated into the loss function in the none of these works. Feature map pooling-based localization did not effectively capture the precise disease regions.

Multiple instance learning. In multiple instance learning (MIL), an input is a labeled bag (e.g., an image) with many instances (e.g., image patches) [3]. The label is assigned at the bag level. Wu et al. assumed each image as a dual-instance example, including its object proposals and possible text annotations [31]. The framework achieved convincing performance in vision tasks including classification and image annotation. In medical imaging domain, Yan et al. utilized a deep MIL framework for body part recognition [32]. Hou et al. first trained an CNN on image patches and then an image-level decision fusion model by patch-level prediction histograms to generate the image-level labels [15]. By ranking the patches and defining three types of losses for different schemes, Zhu et al. proposed an end-to-end deep multi-instance network to achieve mass classification for whole mammogram [38]. We are building an end-to-end unified model to make great use of both image-level labels and bounding box annotations effectively.

7.3 Model

Given images with disease labels and limited bounding box information, we aim to design a unified model that simultaneously produces disease identification and localization. We have formulated two tasks into the same underlying prediction model so that (1) it can be jointly trained end to end and (2) two tasks can be mutually beneficial. The proposed architecture is summarized in Fig. 7.3.

7.3.1 Image Model

Convolutional neural network. As shown in Fig. 7.3a, we use the residual neural network (ResNet) architecture [13] given its dominant performance in ILSVRC competitions [25]. Our framework can be easily extended to any other advanced CNN models. The recent version of pre-act-ResNet [14] is used (we call it ResNet-v2 interchangeably in this chapter). After removing the final classification layer and global pooling layer, an input image with shape $h \times w \times c$ produces a feature tensor with

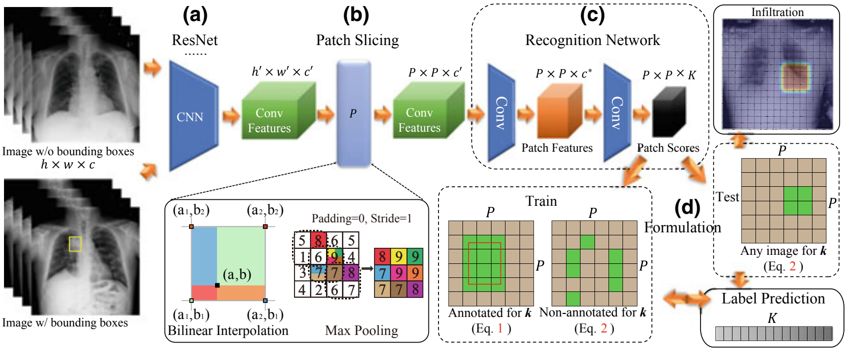


Fig. 7.3 Model overview. **a** The input image is first processed by a CNN. **b** The patch slicing layer resizes the convolutional features from the CNN using max-pooling or bilinear interpolation. **c** These regions are then passed to a fully convolutional recognition network. **d** During training, we use multi-instance learning assumption to formulate two types of images; during testing, the model predicts both labels and class-specific localizations. The red frame represents the ground truth bounding box. The green cells represent patches with positive labels, and brown is negative. Please note during training, for unannotated images, we assume there is at least one positive patch and the green cells shown in the figure are not deterministic

shape $h' \times w' \times c'$ where h, w, and c are the height, width, and number of channels of the input image respectively while $h' = \frac{h}{32}$, $w' = \frac{w}{32}$, $c' = 2048$. The output of this network encodes the images into a set of abstracted feature maps.

Patch slicing. Our model divides the input image into $P \times P$ patch grid, and for each patch, we predict K binary class probabilities, where K is the number of possible disease types. As the CNN gives c' input feature maps with the size of $h' \times w'$, we down/up sample the input feature maps to $P \times P$ through a patch slicing layer shown in Fig. 7.3b. Please note that P is an adjustable hyperparameter. In this way, a node in the same spatial location across all the feature maps corresponds to one patch of the input image. We upsample the feature maps If their sizes are smaller than expected patch grid size. Otherwise, we downsample them.

Upsampling. We use a simple bilinear interpolation to upsample the feature maps to the desired patch grid size. As interpolation is, in essence, a fractionally stridden convolution, it can be performed in-network for end-to-end learning and is fast and effective [22]. A deconvolution layer [33] is not necessary to cope with this simple task.

Downsampling. The bilinear interpolation makes sense for downsampling only if the scaling factor is close to 1. We use max-pooling to downsample the feature maps. In general cases, the spatial size of the output volume is a function of the input width/height (w), the filter (receptive field) size (f), the stride (s), and the amount of zero padding used (p) on the border. The output width/height (o) can be obtained by $\frac{w-f+2p}{s} + 1$. To simplify the architecture, we set $p = 0$ and $s = 1$, so that $f = w - o + 1$.

Fully convolutional recognition network. We follow [22] to use fully convolution layers as the recognition network. Its structure is shown in Fig. 7.3c. The c' resized feature maps are first convolved by 3×3 filters into a smaller set of feature maps with c^* channels, followed by batch normalization [17] and rectified linear units (ReLU) [10]. Note that the batch normalization also regularizes the model. We set $c^* = 512$ to represent patch features. The abstracted feature maps are then passed through a 1×1 convolution layer to generate a set of $P \times P$ final predictions with K channels. Each channel gives prediction scores for one class among all the patches, and the prediction for each class is normalized by a logistic function (sigmoid function) to [0, 1]. The final output of our network is the $P \times P \times K$ tensor of predictions. The image-level label prediction for each class in K is calculated across $P \times P$ scores, which is described in Sect. 7.3.2.

7.3.2 Loss Function

Multi-label classification. Multiple disease types can be often identified in one chest X-ray image and these disease types are not mutually exclusive. Therefore, we define a binary classifier for each class/disease type in our model. The binary classifier outputs the class probability. Note that the binary classifier is not applied to the entire image, but to all small patches. We will show how this can translate to image-level labeling below.

Joint formulation of localization and classification. Since we intend to build K binary classifiers, we will exemplify just one of them, for example, class k. Note that K binary classifiers will use the same features and only differ in their last logistic regression layers. The ith image x_i is partitioned into a set $\mathscr{M}$ of patches equally, $x_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$, where $m = |\mathscr{M}| = P \times P$

Images with annotated bounding boxes. As shown in Fig. 7.3d, suppose an image is annotated with class k and a bounding box. We denote n be the number of patches covered by the bounding box, where $n < m$. Let this set be $\mathscr{N}$. Each patch in the set $\mathscr{N}$ as positive for class k and each patch outside the bounding box as negative. Note that if a patch is covered partially by the bounding box of class k, we still consider it a positive patch for class k. The bounding box information is not lost. For the jth patch in the ith image, let p_{ij}^k be the foreground probability for class k. Since all patches have their labels, the probability of an image being positive for class k is defined as

$$p(y_k|x_i, \text{bbox}_i^k) = \prod_{j \in \mathscr{N}} p_{ij}^k \cdot \prod_{j \in \mathscr{M} \setminus \mathscr{N}} (1 - p_{ij}^k), \tag{7.1}$$

where y_k is the kth network output denoting whether an image is a positive example of class k. For example, for a class other than k, this image is treated as the negative sample without a bounding box. We define a patch as positive to class k when it is overlapped with a ground truth box, and negative otherwise.

Images without annotated bounding boxes. If the ith image is labeled as class k without any bounding box, we know that there must be at least one patch classified as k to make this image a positive example of class k. Therefore, the probability of this image being positive for class k is defined as the image-level score,[3]

$$p(y_k|x_i) = 1 - \prod_{j \in \mathscr{M}} (1 - p_{ij}^k). \tag{7.2}$$

At test time, we calculate $p(y_k|x_i)$ by Eq. 7.2 as the prediction probability for class k.

Combined loss function. Note that $p(y_k|x_i, \text{bbox}_i^k)$ and $p(y_k|x_i)$ are the image-level probabilities. The loss function for class k can be expressed as minimizing the negative log-likelihood of all observations as follows:

[3] Later on, we notice a similar definition [19] for this multi-instance problem. We argue that our formulation is in a different context of solving classification and localization in a unified way for images with limited bounding box annotation. Yet, this related work can be viewed as a successful validation of our multi-instance learning based formulation.

$$\begin{aligned}\mathcal{L}_k = &- \lambda_{\text{bbox}} \sum_i \eta_i p(y_k^*|x_i, \text{bbox}_i^k) \log(p(y_k|x_i, \text{bbox}_i^k)) \\ &- \lambda_{\text{bbox}} \sum_i \eta_i (1 - p(y_k^*|x_i, \text{bbox}_i^k)) \log(1 - p(y_k|x_i, \text{bbox}_i^k)) \\ &- \sum_i (1 - \eta_i) p(y_k^*|x_i) \log(p(y_k|x_i)) \\ &- \sum_i (1 - \eta_i)(1 - p(y_k^*|x_i)) \log(1 - p(y_k|x_i)),\end{aligned} \tag{7.3}$$

where i is the index of a data sample, η_i is 1 when the ith sample is annotated with bounding boxes, otherwise 0. λ_{bbox} is the factor balancing the contributions from annotated and unannotated samples. $p(y_k^*|x_i) \in \{0, 1\}$ and $p(y_k^*|x_i, \text{bbox}_i^k) \in \{0, 1\}$ are the observed probabilities for class k. Obviously, $p(y_k^*|x_i, \text{bbox}_i^k) \equiv 1$, thus Eq. 7.3 can be rewritten as follows:

$$\begin{aligned}\mathcal{L}_k = &- \lambda_{\text{bbox}} \sum_i \eta_i \log(p(y_k|x_i, \text{bbox}_i^k)) \\ &- \sum_i (1 - \eta_i) p(y_k^*|x_i) \log(p(y_k|x_i)) \\ &- \sum_i (1 - \eta_i)(1 - p(y_k^*|x_i)) \log(1 - p(y_k|x_i)).\end{aligned} \tag{7.4}$$

In this way, the training is strongly supervised (per patch) by the given bounding box; it is also supervised by the image-level labels if the bounding boxes are not available.

To enable end-to-end training across all classes, we sum up the class-wise loss to define the total loss as

$$\mathcal{L} = \sum_k \mathcal{L}_k. \tag{7.5}$$

7.3.3 Localization Generation

The full model predicts a probability score for each patch in the input image. We define a score threshold T_s to distinguish the activated patches against the nonactivated ones. If the probability score p_{ij}^k is larger than T_s, we consider the jth patch in the ith image belongs to the localization for class k. We set $T_s = 0.5$ in this chapter. Please note that we do not predict strict bounding boxes for the regions of disease—the combined patches representing the localization information can be a non-rectangular shape.

7.3.4 Training

We use ResNet-v2-50 as the image model and select the patch slicing size from $\{12, 16, 20\}$. The model is pre-trained on the ImageNet 1000-class dataset [6] with Inception [28] preprocessing method where the image is normalized to $[-1, 1]$ and resized to 299×299. We initialize the CNN with the weights from the pre-trained model, which helps the model converge faster than training from scratch. During training, we also fine-tune the image model, as we believe the feature distribution of medical images differs from that of natural images. We set the batch size as 5 to load the entire model to the GPU, train the model with 500k iterations of minibatch, and decay the learning rate by 0.1 from 0.001 every 10 epochs of training data. We add L2 regularization to the loss function to prevent overfitting. We optimize the model by Adam [18] method with asynchronous training on 5 Nvidia P100 GPUs. The model is implemented in TensorFlow [1].

Smoothing the image-level scores In Eqs. 7.1 and 7.2, the notation $\prod$ denotes the product of a sequence of probability terms ($[0, 1]$), which often leads to the a product value of 0 due to the computational underflow if $m = |\mathscr{M}|$ is large. The log loss in Eq. 7.3 mitigates this for Eq. 7.1, but does not help Eq. 7.2, since the log function cannot directly affect its product term. To mitigate this effect, we normalize the patch scores p_{ij}^k and $1 - p_{ij}^k$ from $[0, 1]$ to $[0.98, 1]$ to make sure the image-level scores $p(y_k|x_i, \text{bbox}_i^k)$ and $p(y_k|x_i)$ smoothly varies within the range of $[0, 1]$. Since we are thresholding the image-level scores in the experiments, we found this normalization works quite well.

Notation p_{ij}^k and $1 - p_{ij}^k$ represent a patch's (the jth patch of image i) positive and negative probabilities for class k. Their values are always in $[0, 1]$. We consider the problem of numerical underflow as follows. The product terms ($\prod$) in Eqs. 7.1 and 7.2 can quickly go to 0 when many of the terms in the product is small due to the limited precision of float numbers. The log loss in Eq. 7.3 mitigates this for Eq. 7.1, but does not help Eq. 7.2, since the log function can not directly affect its product term. This effectively renders Eq. 7.2 as a constant value of 1, making it irrelevant on updating the network parameters. (The contribution of the gradient from Eq. 7.2 will be close to 0.) Similar things happen at test time. To do binary classification for an image, we determine its label by thresholding the image-level score (Eq. 7.2). It is impossible to find a threshold in $[0, 1]$ to distinguish the image-level scores when the score (Eq. 7.2) is a constant of 1; all the images will be labeled the same.

Fortunately, if we can make sure that the image-level scores $p(y_k|x_i, \text{bbox}_i^k)$'s and $p(y_k|x_i)$ spread out in $[0, 1]$ instead of congregating at 1, we then can find an appropriate threshold for the binary classification. To this end, we normalize p_{ij}^k and $1 - p_{ij}^k$ from $[0, 1]$ to $[0.98, 1]$. The reason of such choice is as follows. In the actual system, we often use single-precision floating-point number to represent real numbers. It can represent a real number as accurate as 7 decimal digits [5]. If the number of patches in an image, $m = 16 \times 16$, a real number $p \in [0, 1]$ should be larger than around 0.94 (by obtaining p from $p^{256} \geq 10^{-7}$) to make sure that the p^m varies smoothly in $[0, 1]$ with respect to p changes in $[0.94, 1]$. To be a bit more conservative, we set 0.98 as our lower limit in our experiment. This method enables valid and efficient training and testing of our method. And in the evaluation, the number of thresholds can be finite to calculate the AUC scores, as the image-level probability score is well represented using the values in [0, 1]. A downside of our approach is that a normalized patch-level probability score does not necessarily reflect the meaning of probability anymore.

More weights on images with bounding boxes. In Eq. 7.4, the parameter λ_{bbox} weighs the contribution from the images with annotated bounding boxes. Since the amount of such images is limited, and if we treat them equally with the images without bounding boxes, it often leads to worse performance. We thus increase the weight for images with bounding boxes to $\lambda_{\text{bbox}} = 5$ by cross-validation.

7.4 Experiments

NIH Chest X-ray dataset [30] consists of 112, 120 frontal-view X-ray images with 14 disease labels (each image can have multi-labels). These labels are obtained by analyzing the associated radiology reports. The disease labels are expected to have an accuracy of above 90% [30]. We take the provided labels as ground truth for training and evaluation in this chapter. Meanwhile, the dataset also contains 984 labeled bounding boxes for 880 images by board-certified radiologists. Note that the provided bounding boxes correspond to only eight types of disease instances. We separate the images with provided bounding boxes from the entire dataset. Hence we have two sets of images called "annotated" (880 images) and "unannotated" (111, 240 images).

We resize the original 3-channel images from 1024×1024 to 512×512 pixels for fast processing. The pixel values in each channel are normalized to $[-1, 1]$. We do not apply any data augmentation techniques.

7.4.1 Disease Identification

We conduct a fivefold cross-validation. For each fold, we have done two experiments. In the first one, we train the model using 70% of bounding box annotated and 70% unannotated images to compare the results with the reference model [30] (Table 7.1). To our knowledge, the reference model has the published state-of-the-art performance of disease identification on this dataset. In the second experiment, we explore two data ratio factors of annotated and unannotated images to demonstrate the effectiveness of the supervision provided by the bounding boxes (Fig. 7.4). We decrease the amount of images without bounding boxes from 80 to 0% by a step of 20%. And then for each of those settings, we train our model by adding 80% or none of bounding box annotated images. For both experiments, the model is always evaluated on the fixed 20% annotated and unannotated images for this fold.

Table 7.1 AUC scores comparison with the reference baseline model. Results are rounded to two decimal digits for table readability. Bold values denote better results. The results for the reference baseline are obtained from the latest update of [30]

Disease	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis
Baseline	0.70	0.81	0.70	0.81	0.76	0.83	**0.79**
Ours	**0.80** ± 0.00	**0.87** ± 0.01	**0.80** ± 0.01	**0.88** ± 0.01	**0.87** ± 0.00	**0.91** ± 0.01	0.78 ± 0.02
Disease	Hernia	Infiltration	Mass	Nodule	Pleural thickening	Pneumonia	Pneumothorax
Baseline	**0.87**	0.66	0.69	0.67	0.68	0.66	0.80
Ours	0.77 ± 0.03	**0.70** ± 0.01	**0.83** ± 0.01	**0.75** ± 0.01	**0.79** ± 0.01	**0.67** ± 0.01	**0.87** ± 0.01

Evaluation metrics. We use AUC scores (the area under the ROC[4] curve) to measure the performance of our model [7]. A higher AUC score implies a better classifier.

Comparison with the reference model. Table 7.1 gives the AUC scores for all the classes. We compare our results with the reference baseline fairly: we, as the reference, use ImageNet pretrained ResNet-50,[5] after which a convolution layer follows; both works use 70% images for training and 20% for testing, and we also conduct a fivefold cross-validation to show the robustness of our model.

Compared to the reference model, our proposed model achieves better AUC scores for most diseases. The overall improvement is remarkable and the standard errors are small. The large objects, such as "Cardiomegaly", "Emphysema", and "Pneumothorax", are as well recognized as the reference model. Nevertheless, for small objects like "Mass" and "Nodule", the performance is significantly improved. Because our model slices the image into small patches and uses bounding boxes to supervise the training process, the patch containing small object stands out of all the patches to represent the complete image. For "Hernia", there are only 227 (0.2%) samples in the dataset. These samples are not annotated with bounding boxes. Thus, the standard error is relatively larger than other diseases.

Bounding box supervision improves classification performances. We consider using 0% annotated images as our own baseline (right groups in Fig. 7.4). We use 80% annotated images (left groups in Fig. 7.4) to compare with the our own baseline. We plot the mean performance for the cross-validation in Fig. 7.4, the standard errors are not plotted but similar to the numbers reported in Table 7.1. The number of 80% annotated images is just 704, which is quite small compared to the number of 20% unannotated images (22, 248). We observe in Fig. 7.4 that for almost all the disease types, using 80% annotated images to train the model improves the prediction performance (by comparing the bars with the same color in two groups for the same disease). For some disease types, the absolute improvement is significant (>5%). We believe that this is because all the disease classifiers share the same underlying image model; a better trained image model using eight disease annotations can improve all

[4] Here ROC is the Receiver Operating Characteristic, which measures the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings (200 thresholds in this chapter).

[5] Using ResNet-v2 [14] shows marginal performance difference for our network compared to ResNet-v1 [13] used in the reference baseline.

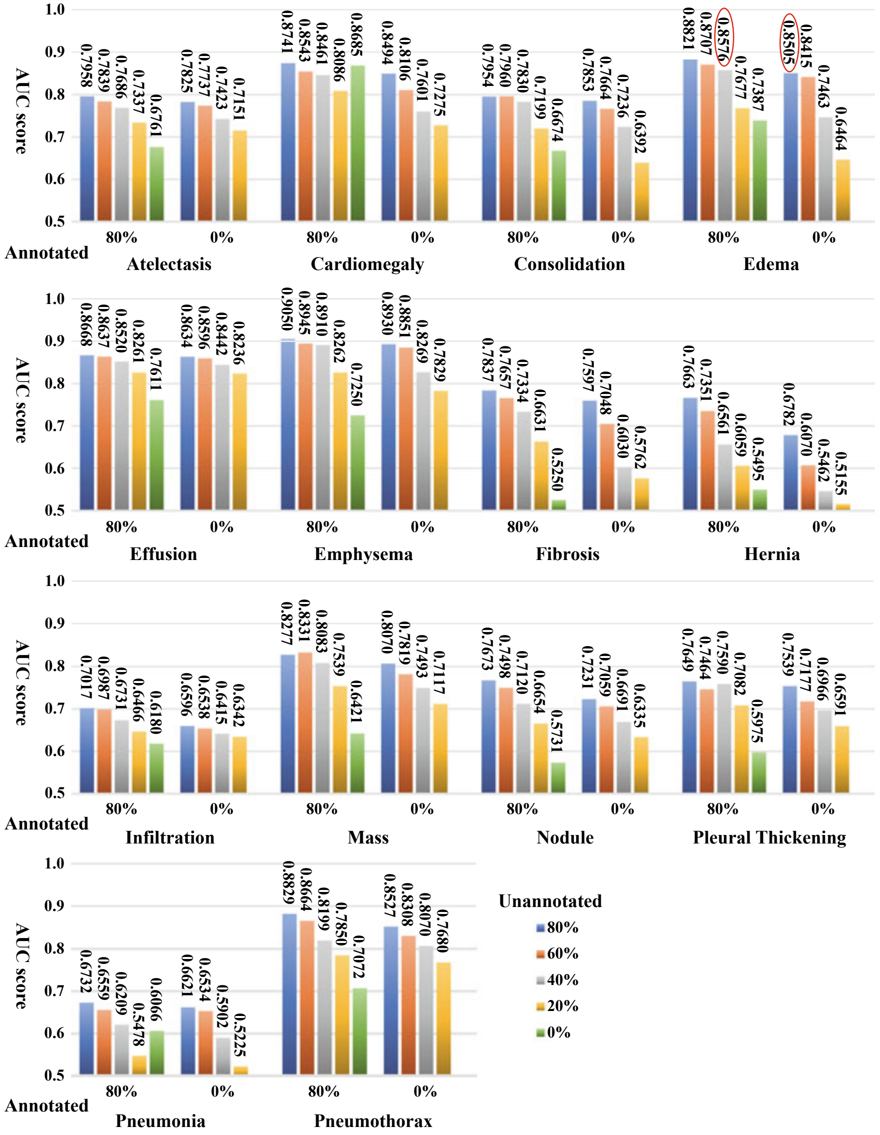


Fig. 7.4 AUC scores for models trained using different data combinations. Training set: annotated samples,{left: 80% (704 images), right: 0% (baseline, 0 images)} for each disease type; unannotated samples, {80% (88, 892), 60% (66, 744), 40% (44, 496), 20% (22, 248), 0% (0)} from left to right for each disease type. The evaluation set is 20% annotated and unannotated samples which are not included in the training set. No result for 0% annotated and 0% unannotated images. Using 80% annotated images and certain amount of unannotated images improves the AUC score compared to using the same amount of unannotated images (same colored bars in two groups for the same disease), as the joint model benefits from the strong supervision of the tiny set of bounding box annotations

14 classifiers' performance. Specifically, some diseases, annotated and unannotated, share similar visual features. For example, "Consolidation" and "Edema" both appear as fluid accumulation in the lungs, but only "Consolidation" is annotated. The feature sharing enables supervision for "Consolidation" to improve "Edema" performance as well.

Bounding box supervision reduces the demand of the training images. Importantly, it requires less unannotated images to achieve the similar AUC scores by using a small set of annotated images for training. As denoted with red circles in Fig. 7.4, taking "Edema" as an example, using 40% (44, 496) unannotated images with 80% (704) annotated images (45, 200 in total) outperforms the performance of using only 80% (88, 892) unannotated images.

Discussion. Generally, decreasing the amount of unannotated images (from left to right in each bar group) will degrade AUC scores accordingly in both groups of 0% and 80% annotated images. Yet as we decrease the amount of unannotated images, using annotated images for training gives smaller AUC degradation or even improvement. For example, we compare the "Cardiomegaly" AUC degradation for two pairs of experiments: {annotated: 80%, unannotated: 80 and 20%} and {annotated: 0%, unannotated: 80 and 20%}. The AUC degradation for the first group is just 0.07 while that for the second group is 0.12 (accuracy degradation from blue to yellow bar).

When the amount of unannotated images is reduced to 0%, the performance is significantly degraded. Because under this circumstance, the training set only contains positive samples for eight disease types and lacks the positive samples of the other six. Interestingly, "Cardiomegaly" achieves the second best score (AUC = 0.8685, the second green bar in Fig. 7.4) when only annotated images are trained. The possible reason is that the location of cardiomegaly is always fixed to the heart covering a large area of the image and the feature distributions for enlarged hearts are similar to normal ones. Without unannotated samples, the model easily distinguishes the enlarged hearts from normal ones given supervision from bounding boxes. When the model sees hearts without annotation, the enlarged ones are disguised and fail to be recognized. As more unannotated samples are trained, the enlarged hearts are recognized again by image-level supervision (AUC from 0.8086 to 0.8741).

7.4.2 Disease Localization

Similarly, we conduct a fivefold cross-validation. For each fold, we have done three experiments. In the first experiment, we investigate the importance of bounding box supervision by using all the unannotated images and increasing the amount of annotated images from 0 to 80% by the step of 20% (Fig. 7.5). In the second one, we fix the amount of annotated images to 80% and increase the amount of unannotated images from 0 to 100% by the step of 20% to observe whether unannotated images are able to help annotated images to improve the performance (Fig. 7.7). At last, we train the model with 80% annotated images and half (50%) unannotated images to

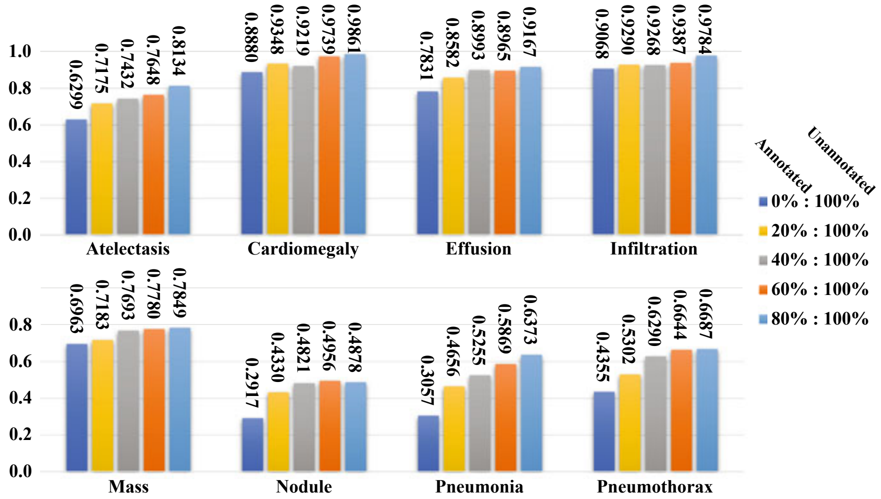


Fig. 7.5 Disease localization accuracy using IoR where T(IoR) = 0.1. Training set: annotated samples, {0% (0), 20% (176), 40% (352), 60% (528), 80% (704)} from left to right for each disease type; unannotated samples, 100% (111, 240 images). The evaluation set is 20% annotated samples which are not included in the training set. For each disease, the accuracy is increased from left to right, as we increase the amount of annotated samples, because more annotated samples bring more bounding box supervision to the joint model

compare localization accuracy with the reference baseline [30] (Table 7.2). For each experiment, the model is always evaluated on the fixed 20% annotated images for this fold.

Evaluation metrics. We evaluate the detected regions (which can be non-rectangular and discrete) against the annotated ground truth (GT) bounding boxes, using two types of measurement: intersection over union ratio (IoU) and intersection over the detected region (IoR).[6] The localization results are only calculated for those eight disease types with ground truth provided. We define a correct localization when either IoU > T(IoU) or IoR > T(IoR), where T(*) is the threshold.

Bounding box supervision is necessary for localization. We present the experiments shown in Fig. 7.5. The threshold is set as tolerable as T(IoR) = 0.1 to show the training data combination effect on the accuracy. Even though the amount of the complete set of unannotated images is dominant compared with the evaluation set (111, 240 versus 176), without annotated images (the most left bar in each group), the model fails to generate accurate localization for most disease types. Because in this situation, the model is only supervised by image-level labels and optimized using probabilistic approximation from patch-level predictions. As we increase the amount of annotated images gradually from 0 to 80% by the step of 20% (from left to right in each group), the localization accuracy for each type is increased accordingly. We can

[6] Note that we treat discrete detected regions as one prediction region, thus IoR is analogous to intersection over the detected bounding box area ratio (IoBB).

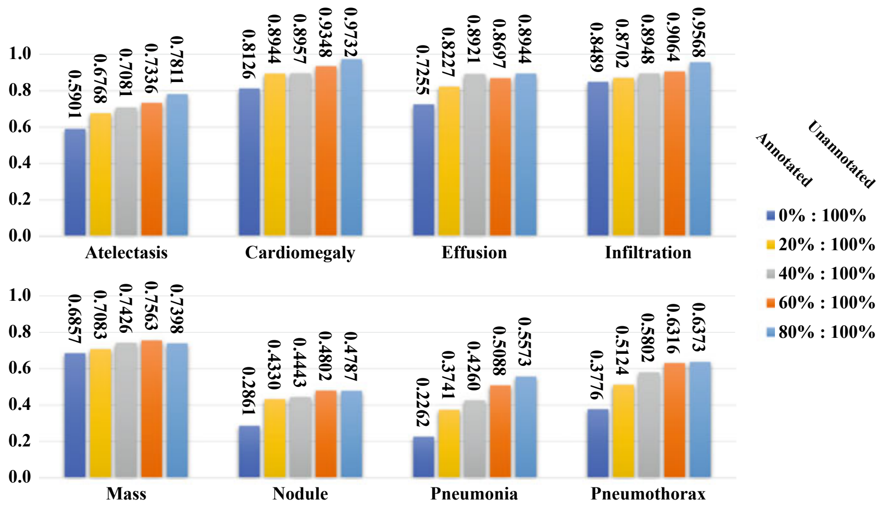


Fig. 7.6 Disease localization accuracy using IoU where T(IoU) = 0.1. Training set: annotated samples, {0% (0), 20% (176), 40% (352), 60% (528), 80% (704)} from left to right for each disease type; unannotated samples, 100% (111, 240 images). The evaluation set is 20% annotated samples which are not included in the training set. For each disease, the accuracy is increased from left to right, as we increase the amount of annotated samples, because more annotated samples bring more bounding box supervision to the joint model

see the necessity of bounding box supervision by observing the localization accuracy increase. Therefore, the bounding box is necessary to provide accurate localization results and the accuracy is positively proportional to the amount of annotated images. We have similar observations when T(*) varies.

Similarly, we investigate the importance of bounding box supervision by using all the unannotated images and increasing the amount of annotated images from 0 to 80% by the step of 20% (Fig. 7.6). without annotated images (the most left bar in each group), the model is only supervised by image-level labels and optimized using probabilistic approximation from patch-level predictions. The results by unannotated images only are not able to generate accurate localization of disease. As we increase the amount of annotated images gradually from 0% to 80% by the step of 20% (from left to right in each group), the localization accuracy for each type is increased accordingly.

More unannotated data does not always mean better results for localization. In Fig. 7.7, when we fix the amount of annotated images and increase the amount of unannotated ones for training (from left to right in each group), the localization accuracy does not increase accordingly. Some disease types achieve very high accuracy (even highest) without any unannotated images (the most left bar in each group), such as “Pneumonia” and “Cardiomegaly”. Similarly as described in the discussion of Sect. 7.4.1, unannotated images and too many negative samples degrade the localization performance for these diseases. All disease types experience an accuracy

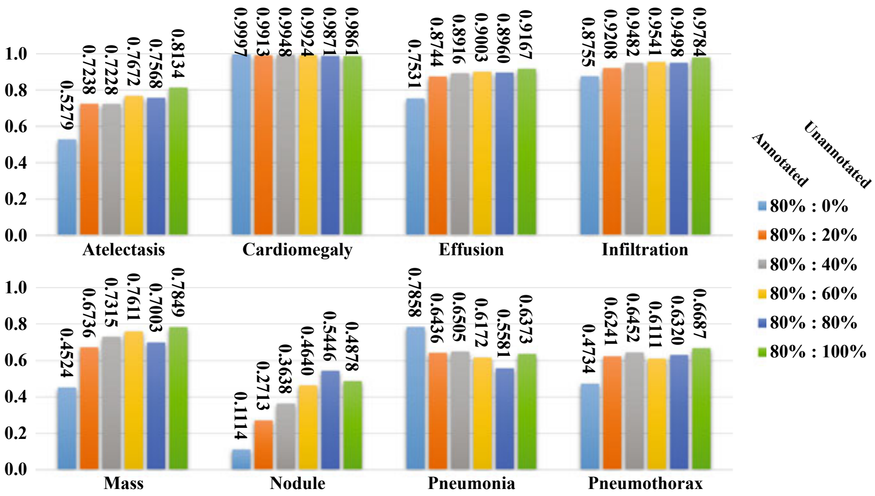


Fig. 7.7 Disease localization accuracy using IoR where T(IoR) = 0.1. Training set: annotated samples, 80% (704 images); unannotated samples, {0% (0), 20% (22, 248), 40% (44, 496), 60% (66, 744), 80% (88, 892), 100% (111, 240)} from left to right for each disease type. The evaluation set is 20% annotated samples which are not included in the training set. Using annotated samples only can produce a model which localizes some diseases. As the amount of unannotated samples increases in the training set, the localization accuracy is improved and all diseases can be localized. The joint formulation for both types of samples enables unannotated samples to improve the performance with weak supervision

increase, a peak score, and then an accuracy fall (from orange to green bar in each group). Therefore, with bounding box supervision, unannotated images will help to achieve better results in some cases and it is not necessary to use all of them.

Similarly, we fix the amount of annotated images to 80% and increase the amount of unannotated images from 0% to 100% by the step of 20% to observe whether unannotated images are able to help annotated images to improve the performance (Fig. 7.8). For some diseases, it achieves the best accuracy without any unannotated images. For most diseases, the accuracy experience an accuracy increase, a peak score, and then an accuracy fall (from orange to green bar in each group) as we increase the amount of unannotated images. A possible explanation is that too many unannotated images overwhelm the strong supervision from the small set of annotated images. A possible remedy is to lower the weight of unannotated images during training.

Comparison with the reference model. In each fold, we use 80% annotated images and 50% unannotated images to train the model and evaluate the other 20% annotated images in each fold. Since we use fivefold cross-validation, the complete set of annotated images has been evaluated to make a relatively fair comparison with the reference model. In Table 7.2, we compare our localization accuracy under varying T(IoU) with respect to the reference model in [30]. Our model predicts accurate disease regions, not only for the easy tasks like "Cardiomegaly" but also for the hard

Table 7.2 Disease localization accuracy comparison using IoU where T(IoU) = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7}.The bold values denote the best results. Note that we round the results to two decimal digits for table readability. Using different thresholds, our model outperforms the reference baseline in most cases and remains capability of localizing diseases when the threshold is big

T(IoU)	Model	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pneumothorax
0.1	Ref.	0.69	0.94	0.66	0.71	0.40	0.14	**0.63**	0.38
	Ours	**0.71** ± 0.05	**0.98** ± 0.02	**0.87** ± 0.03	**0.92** ± 0.05	**0.71** ± 0.10	**0.40** ± 0.10	0.60 ± 0.11	**0.63** ± 0.09
0.2	Ref.	0.47	0.68	0.45	0.48	0.26	0.05	0.35	0.23
	Ours	**0.53** ± 0.05	**0.97** ± 0.02	**0.76** ± 0.04	**0.83** ± 0.06	**0.59** ± 0.10	**0.29** ± 0.10	**0.50** ± 0.12	**0.51** ± 0.08
0.3	Ref.	0.24	0.46	0.30	0.28	0.15	0.04	0.17	0.13
	Ours	**0.36** ± 0.08	**0.94** ± 0.01	**0.56** ± 0.04	**0.66** ± 0.07	**0.45** ± 0.08	**0.17** ± 0.10	**0.39** ± 0.09	**0.44** ± 0.10
0.4	Ref.	0.09	0.28	0.20	0.12	0.07	0.01	0.08	0.07
	Ours	**0.25** ± 0.07	**0.88** ± 0.06	**0.37** ± 0.06	**0.50** ± 0.05	**0.33** ± 0.08	**0.11** ± 0.02	**0.26** ± 0.07	**0.29** ± 0.06
0.5	Ref.	0.05	0.18	0.11	0.07	0.01	0.01	0.03	0.03
	Ours	**0.14** ± 0.05	**0.84** ± 0.06	**0.22** ± 0.06	**0.30** ± 0.03	**0.22** ± 0.05	**0.07** ± 0.01	**0.17** ± 0.03	**0.19** ± 0.05
0.6	Ref.	0.02	0.08	0.05	0.02	0.00	0.01	0.02	0.03
	Ours	**0.07** ± 0.03	**0.73** ± 0.06	**0.15** ± 0.06	**0.18** ± 0.03	**0.16** ± 0.06	**0.03** ± 0.03	**0.10** ± 0.03	**0.12** ± 0.02
0.7	Ref.	0.01	0.03	0.02	0.00	0.00	0.00	0.01	0.02
	Ours	**0.04** ± 0.01	**0.52** ± 0.05	**0.07** ± 0.03	**0.09** ± 0.02	**0.11** ± 0.06	**0.01** ± 0.00	**0.05** ± 0.03	**0.05** ± 0.03

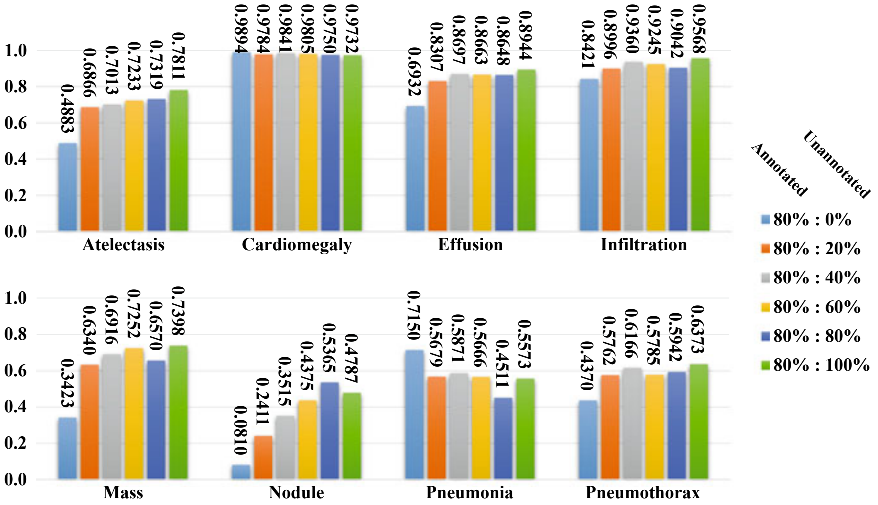


Fig. 7.8 Disease localization accuracy using IoU where T(IoU) = 0.1. Training set: annotated samples, 80% (704 images); unannotated samples, {0% (0), 20% (22, 248), 40% (44, 496), 60% (66, 744), 80% (88, 892), 100% (111, 240)} from left to right for each disease type. The evaluation set is 20% annotated samples which are not included in the training set. Using annotated samples only can produce a model which localizes some diseases. As the amount of unannotated samples increases in the training set, the localization accuracy is improved and all diseases can be localized. The joint formulation for both types of samples enables unannotated samples to improve the performance with weak supervision

ones like "Mass" and "Nodule" which have very small regions. When the threshold increases, our model maintains a large accuracy lead over the reference model. For example, when evaluated by T(IoU) = 0.6, our "Cardiomegaly" accuracy is still 73.42% while the reference model achieves only 16.03%; our "Mass" accuracy is 14.92% while the reference model fails to detect any "Mass" (0% accuracy). In clinical practice, a specialist expects as accurate localization as possible so that a higher threshold is preferred. Hence, our model outperforms the reference model with a significant improvement with less training data. Please note that as we consider discrete regions as one predicted region, the detected area and its union with GT bboxs are usually larger than the reference work which generates multiple bounding boxes. Thus for some disease types like "Pneumonia", when the threshold is small, our result is not as good as the reference.

Similarly, we use 80% annotated images and 50% unannotated images to train the model and evaluate on the other 20% annotated images in each fold. Comparing with the reference model [30], our model achieves higher localization accuracy for various T(IoR) as shown in Table 7.3.

Table 7.3 Disease localization accuracy comparison using IoR where T(IoR) = {0.1, 0.25, 0.5, 0.75, 0.9}. The bold values denote the best results. Note that we round the results to two decimal digits for table readability. Using different thresholds, our model outperforms the reference baseline in most cases and remains capability of localizing diseases when the threshold is big. The results for the reference baseline are obtained from the latest update of [30]

T(IoR)	Model	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pneumothorax
0.1	Ref.	0.62	**1.00**	0.80	0.91	0.59	0.15	**0.86**	0.52
	Ours	**0.77** ± 0.06	0.99 ± 0.01	**0.91** ± 0.04	**0.95** ± 0.05	**0.75** ± 0.08	**0.40** ± 0.11	0.69 ± 0.09	**0.68** ± 0.10
0.25	Ref.	0.39	0.99	0.63	0.80	0.46	0.05	**0.71**	0.34
	Ours	**0.57** ± 0.09	**0.99** ± 0.01	**0.79** ± 0.02	**0.88** ± 0.06	**0.57** ± 0.07	**0.25** ± 0.10	0.62 ± 0.05	**0.61** ± 0.07
0.5	Ref.	0.19	0.95	0.42	**0.65**	0.31	0.00	0.48	0.27
	Ours	**0.35** ± 0.04	**0.98** ± 0.02	**0.52** ± 0.03	0.62 ± 0.08	**0.40** ± 0.06	**0.11** ± 0.04	**0.49** ± 0.08	**0.43** ± 0.10
0.75	Ref.	0.09	0.82	0.23	0.44	0.16	0.00	0.29	0.17
	Ours	**0.20** ± 0.04	**0.87** ± 0.05	**0.34** ± 0.06	**0.46** ± 0.07	**0.29** ± 0.06	**0.07** ± 0.04	**0.43** ± 0.06	**0.30** ± 0.07
0.9	Ref.	0.07	**0.65**	0.14	**0.36**	0.09	0.00	0.23	0.12
	Ours	**0.15** ± 0.03	0.59 ± 0.04	**0.23** ± 0.05	0.32 ± 0.07	**0.22** ± 0.05	**0.06** ± 0.03	**0.34** ± 0.04	**0.22** ± 0.05

7.4.3 Qualitative Results

Figure 7.9 shows exemplary localization results of the unified diagnosis model. The localization enables the explainability of chest X-ray images. It is intuitive to see that our model produces accurate localization for the diseases compared with the given ground truth bounding boxes. Please note for “Infiltration” (third and fourth images in the third row of Fig. 7.9), both sides of lungs for this patient is infiltrated. Since the dataset only has one bounding box for one disease per image, it misses annotating other bounding boxes for the same disease. Our model gives the remedy. Even though the extra region decreases the IoR/IoU score in the evaluation, but in clinical practice, it provides the specialist with suspicious candidate regions for further examination. When the localization results have no ground truth bounding boxes to compare with, there is also a strong consistency between our results and radiological signs. For example, our model localizes the enlarged heart region (first and second images in the second row) which implies “Cardiomegaly”, and the lung peripheries is highlighted (fifth and sixth images in the second row) implying “Fibrosis” which is in

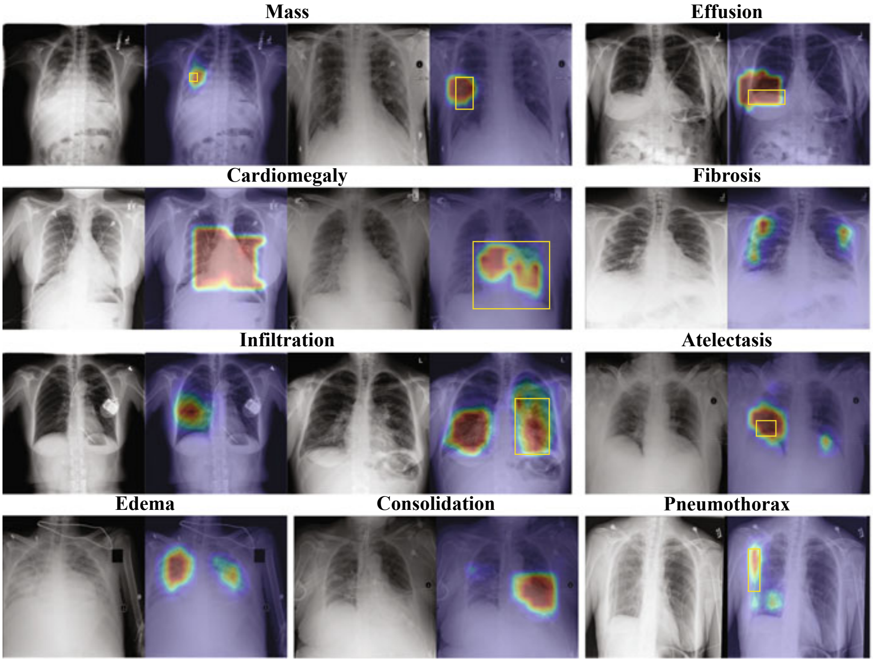


Fig. 7.9 Example localization visualization on the test images. The visualization is generated by rendering the final output tensor as heatmaps and overlapping on the original images. We list some thoracic diseases as examples. The left image in each pair is the original chest X-ray image and the right one is the localization result. All examples are positive for corresponding labels. We also plot the ground truth bounding boxes in yellow on the results when they are provided in the dataset

accordance with the radiological sign of the net-like shadowing of lung peripheries. The "Edema" (first and second images in the fourth row) and "Consolidation" (third and fourth images in the fourth row) are accurately marked by our model. "Edema" always appears in an area that is full of small liquid effusions as the example shows. "Consolidation" is usually a region of compressible lung tissue that has filled with the liquid which appears as a big white area. The model successfully distinguishes both diseases which are caused by similar reason.

7.5 Conclusion

We propose a unified model that jointly models disease identification and localization with limited localization annotation data. This is achieved through the same underlying prediction model for both tasks. Quantitative and qualitative results demonstrate that our method significantly outperforms the state-of-the-art algorithm.

References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. Software available from https://www.tensorflow.org/
2. Akselrod-Ballin A, Karlinsky L, Alpert S, Hasoul S, Ben-Ari R, Barkan E (2016) A region based convolutional network for tumor detection and classification in breast mammography. In: International workshop on large-scale annotation of biomedical data and expert label synthesis. Springer, Berlin, pp 197–205
3. Babenko B. Multiple instance learning: algorithms and applications
4. Chen X, Xu Y, Wong DWK, Wong TY, Liu J (2015) Glaucoma detection based on deep convolutional neural network. In: 2015 37th Annual International Conference of the IEEE Engineering in medicine and biology society (EMBC). IEEE, pp 715–718
5. IEEE Standards Committee et al. 754-2008 ieee standard for floating-point arithmetic. *IEEE Computer Society Std*, 2008, 2008
6. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition. CVPR 2009. IEEE, pp 248–255
7. Fawcett T (2006) An introduction to roc analysis. Pattern Recognit Lett 27(8):861–874
8. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
9. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
10. Glorot X, Bordes A, Bengio Y (20111) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 315–323
11. Gylys BA, Wedding ME (2017) Medical terminology systems: a body systems approach. FA Davis

12. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European conference on computer vision. Springer, Berlin, pp 346–361
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
14. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: European conference on computer vision. Springer, Berlin, pp 630–645
15. Hou L, Samaras D, Kurc TM, Gao Y, Davis JE, Saltz JH (2016) Patch-based convolutional neural network for whole slide tissue image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2424–2433
16. Hwang S, Kim H-E (2016) Self-transfer learning for fully weakly supervised object localization. arXiv:1602.01625
17. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, p 448–456
18. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
19. Liao F, Liang M, Li Z, Hu X, Song S (2017) Evaluate the malignancy of pulmonary nodules using the 3d deep leaky noisy-or network. arXiv:1711.08324
20. Liu C, Mao J, Sha F, Yuille AL (2017) Attention correctness in neural image captioning. In: AAAI, pp 4176–4182
21. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, Berlin, pp 21–37
22. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
23. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
24. 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, pp 91–99
25. Russakovsky O, Deng J, Hao S, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Via 115(3):211–252
26. Shi J, Zheng X, Li Y, Zhang Q, Ying S (2017) Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of alzheimer's disease. IEEE J Biomed Health Inform
27. Shin H-C, Roberts K, Lu L, Demner-Fushman D, Yao J, Summers RM (2016) Learning to read chest x-rays: recurrent neural cascade model for automated image annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2497–2506
28. Szegedy C, Liu W, Jia, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
29. Wang J, Ding H, Azamian F, Zhou B, Iribarren C, Molloi S, Baldi P (2017) Detecting cardiovascular disease from mammograms with deep learning. IEEE Trans Med Imaging
30. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3462–3471
31. Wu J, Yu Y, Huang C, Yu K (2015) Deep multiple instance learning for image classification and auto-annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3460–3469
32. Yan Z, Zhan Y, Peng Z, Liao S, Shinagawa Y, Zhang S, Metaxas DN, Zhou XS (2016) Multi-instance deep learning: discover discriminative local anatomies for bodypart recognition. IEEE Trans Med Imaging 35(5):1332–1343
33. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, Berlin, pp 818–833

34. Zhang Z, Chen P, Sapkota M, Yang L (2017) Tandemnet: distilling knowledge from medical images using diagnostic reports as optional semantic references. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 320–328
35. Zhang Z, Xie Y, Xing F, McGough M, Yang L (2017) Mdnet: a semantically and visually interpretable medical image diagnosis network. arXiv:1707.02485
36. Zhao L, Jia K (2016) Multiscale cnns for brain tumor segmentation and diagnosis. Comput Math Methods Med 2016
37. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
38. Zhu W, Lou Q, Vang YS, Xie X (2017) Deep multi-instance networks with sparse label assignment for whole mammogram classification. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 603–611
39. Zilly J, Buhmann JM, Mahapatra D (2017) Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation. Comput Med Imaging Graph 55:28–41

Chapter 8
Deep Reinforcement Learning for Detecting Breast Lesions from DCE-MRI

Gabriel Maicas, Andrew P. Bradley, Jacinto C. Nascimento, Ian Reid and Gustavo Carneiro

Abstract We present a detection model that is capable of accelerating the inference time of lesion detection from breast dynamically contrast-enhanced magnetic resonance images (DCE-MRI) at state-of-the-art accuracy. In contrast to previous methods based on computationally expensive exhaustive search strategies, our method reduces the inference time with a search approach that gradually focuses on lesions by progressively transforming a bounding volume until the lesion is detected. Such detection model is trained with reinforcement learning and is modeled by a deep Q-network (DQN) that iteratively outputs the next transformation to the current bounding volume. We evaluate our proposed approach in a breast MRI data set containing the T1-weighted and the first DCE-MRI subtraction volume from 117 patients and a total of 142 lesions. Results show that our proposed reinforcement learning based detection model reaches a true positive rate (TPR) of 0.8 at around three false positive detections and a speedup of at least 1.78 times compared to baselines methods.

G. Maicas (✉) · I. Reid · G. Carneiro
School of Computer Science, Australian Institute for Machine Learning,
The University of Adelaide, Adelaide, SA, Australia
e-mail: gabriel.maicas@adelaide.edu.au

I. Reid
e-mail: ian.reid@adelaide.edu.au

G. Carneiro
e-mail: gustavo.carneiro@adelaide.edu.au

A. P. Bradley
Science and Engineering Faculty, Queensland University of Technology, Brisbane, QLD, Australia
e-mail: a6.bradley@qut.edu.au

J. C. Nascimento
Institute for Systems and Robotics, Instituto Superior Tecnico, Lisbon, Portugal
e-mail: jan@isr.ist.utl.pt

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_8

8.1 Introduction

Breast cancer is among the most diagnosed cancers in the last few years [1–3], causing a large number of casualties [4, 5]. To reduce the number of deaths due to breast cancer, early detection from screening exams [6] has gained importance because the treatment of smaller tumors is correlated with higher survival rates [7, 8]. Due to its reduced cost and noninvasiveness, mammography is the most commonly imaging modality used to assess patients in screening programs. However, mammography has unsatisfactory results for patients with dense breasts because of its large number of false positive detections [9–11]. Such high number of false positives is a relevant limitation because patients that are affected tend to be within the high-risk group, which are recommended to be included in the screening process from an early age. Due to their younger age, breasts tend to be denser and mammography is not a suitable modality [9–11]. Thus, it is recommended that the screening for these patients includes dynamically contrast-enhanced magnetic resonance images (DCE-MRI) [12–14], which has been shown to increase the sensitivity and specificity of breast cancer detection.

Interpreting DCE-MRI volumes is a time-consuming and laborious task due to the number of volumes and their high dimensionality, yielding high interobserver variability [15] and errors [16]. Aiming to reduce volume interpretation time, variability, and mistakes, computer-aided diagnosis (CAD) systems are being developed to assist radiologists by providing a second opinion. There is evidence that CAD systems can improve sensitivity and specificity in such framework [17, 18].

The pipeline of CAD systems for breast cancer diagnosis can follow a pre-hoc or a post-hoc framework [19]. In a pre-hoc system (i.e., those that follow a pre-hoc pipeline), suspicious regions of interest (ROIs) are first localized in the input volume. Then, a classifier distinguishes lesions between malignant and nonmalignant (where nonmalignant lesions can be classified into benign or false positive). The diagnosis is finally produced by combining such individual lesion classification results. On the other hand, post-hoc systems (i.e., those that follow a post-hoc pipeline) perform diagnosis by first classifying the whole input volume into negative (normal or benign findings) or positive (malignant findings). Only for the positive cases, the method then localizes the malignant lesions. See Fig. 8.1 for a graphical explanation of the pipeline of pre-hoc and post-hoc approaches.

Pre-hoc and post-hoc approaches differ not only in the order of each of the stages involved to perform diagnosis, but in the type of annotations required at its training stage. While pre-hoc systems require a strongly labeled training set (i.e., the voxel-wise annotation of each lesion), post-hoc approaches only require weakly labeled training sets (i.e., volume-level annotation). Although this seems an advantage in favor of post-hoc approaches, the effect of relying on weakly annotated data sets results in a reduced lesion localization performance accuracy compared to approaches trained with strongly annotated data sets [19]. This lack of precision to highlight the malignant lesions is an important limitation since it can prevent such CAD systems from its adoption in clinical practice. Therefore, the vast majority of CAD systems

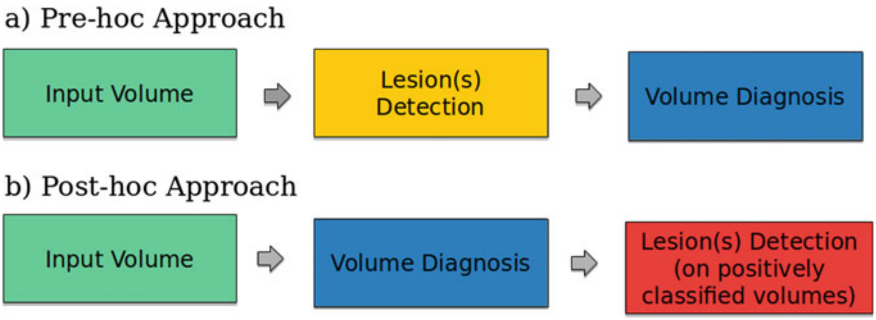


Fig. 8.1 Pre-hoc and post-hoc approaches for breast cancer diagnosis from DCE-MRI. In a pre-hoc pipeline, lesions are initially localized in the image and then classified to decide about the volume diagnosis as positive or negative. In a post-hoc pipeline, the diagnosis is initially computed and, only for positive cases, malignant lesions are localized

are based on the pre-hoc framework given that their higher true positive and true negative detections are more effective at helping radiologists improve their diagnosis performance.

Initial works on fully automated lesion detection methods were based on thresholding the input intensity volume [20]. These methods were improved by two types of approaches that can capture more of the appearance and shape variability present in lesions: (1) exhaustive search based on hand-designed features [21] or (2) unsupervised clustering followed by structured learning [22]. Both approaches can capture more the variability than thresholding methods, but are inefficient in terms of running time complexity given that they need the voxel-wise analysis of the entire volume.

Recently developed deep learning methodologies by the computer vision community [23–25] can help to reduce the inference time for lesion detection and can additionally be trained end-to-end for diagnosis [26]. However, the training of these methods relies on the availability of large strongly labeled data sets, which are not available for many problems in medical image analysis, and in particular for lesion detection from breast DCE-MRI. As a consequence, the medical image analysis community has focused on independently improving the lesion detection and lesion classification stages. In this work, we focus on reducing the large inference time required by the expensive search methods used for the lesion detection stage of pre-hoc systems.

To address the limitations derived from the large inference time and the need for large training sets, we propose a reinforcement learning method based on deep Q-network (DQN) [27] for detecting breast masses from DCE-MRI. The DQN models a policy that indicates how to sequentially transform a large bounding volume such that after each of the transformation the lesion is better focused, and eventually be detected. See Fig. 8.2 for an example of how our proposed method, which is faster than previous detection approaches [21, 22], detects a lesion in a breast DCE-MRI volume. Compared to recently proposed deep learning methods that can detect lesions and perform diagnosis, our approach can be trained with small training sets.

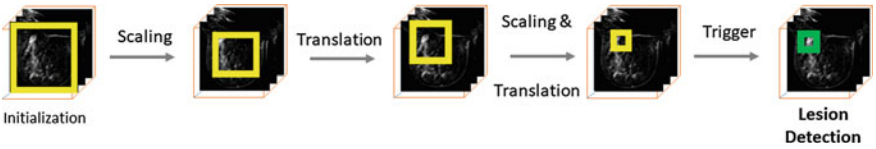


Fig. 8.2 Reinforcement learning based detection model to detect lesions from breast DCE-MRI. The system begins by analyzing a large percentage of the input volume and sequentially applies several transformations to the bounding volume until the lesions are tightly targeted

This advantage stems from the fact that the underlying model can be trained with a potentially infinite number of bounding volumes from the training volumes.

We evaluate the accuracy of our method on a breast DCE-MRI data set containing 117 patients, where 45 are used for training, 13 for validation, and 59 for testing. The number of lesions present in the training, validation, and testing sets are 57, 15, and 69, respectively. Our results show that our proposed method achieves comparable to the state-of-the-art detection results at the same time that it is capable of significantly reducing the processing times by a factor of at least 1.78 times compared with traditional approaches [21, 22]. More precisely, it can produce a true positive detection rate of 0.8 at 3.2 false positives per volume.

8.2 Literature Review

The first fully automated methods for breast lesion detection from DCE-MRI were based on hand-designed features. The method by Vignati et al. [20] detected lesion candidates by thresholding a normalized input volume. Region and kinetic features were used to reduce the number of false positive detections, but their method still yielded a relatively high number of false positives. Aiming to capture the larger variation present in the lesions, Renz et al. [28] extended [20] by increasing the number of hand-crafted features used to refine candidates. However, due to the large enhancement (appearance) variability of lesions, further research required focus on better search strategies.

Candidate proposal shifted to more complex strategies such as exhaustive search [21], unsupervised clustering [22], or saliency detection [29]. Two main issues arose with these search methods: the suboptimality of the hand-crafted features and the large processing times required for the exhaustive search or clustering analysis. Both of these limitations were addressed with deep learning methods [30]. For example, Maicas et al. [31] proposed a multi-scale exhaustive search based on deep learning that searched for optimal features to classify bounding volumes at different scales. However, lesion candidate proposal remained a computationally costly process. Recently developed object detectors [24, 25, 32] can help to overcome such large inference time for lesion detection [26]. However, these methods require large strongly labeled data sets, which are usually not available for breast MRI analysis.

Aiming to reduce the inference time of traditional approaches but still use optimal features, two alternatives can overcome the need for large training sets of deep learning methods: (1) the rapid computation of lesion segmentation maps using a U-net structure [33] as proposed by [34] and (2) a reinforcement learning based detection. Caicedo and Lazebnik [35] proposed to localize objects in an image by using reinforcement learning to train a model that sequentially selects actions to transform a large initial bounding box until the object is tightly focused. Their idea has been shown to work relatively well for detecting objects that present consistent shapes, background, and appearance in computer vision [35] and medical image analysis communities [36]. However, lesions in breast DCE-MRI lack the consistency in the terms mentioned above, and their small size together with the small training sets makes the U-net approach challenging.

In this chapter, we focus on developing a detection method based on deep reinforcement learning that can speed up the inference time while keeping a state-of-the-art accuracy for lesion detection from breast DCE-MRI. We base our work on the approach proposed by Caicedo and Lazebnik [35] than can help to overcome the lack of the training data by designing an embedding function that can be trained with a large number of training samples. Using such embedding function, the algorithm decides at each time step how to optimally transform an initial large bounding volume to precisely focus a lesion. The algorithm achieves detection results comparable to the state of the art at running times that are at least 1.78 times faster than baseline approaches.

8.3 Methods

In this section, we introduce the data set in Sect. 8.3.1, the method to detect lesions from breast DCE-MRI volumes in Sect. 8.3.2, and the training and inference stages of our method in Sects. 8.3.3 and 8.3.4, respectively.

8.3.1 Data Set

Let $\mathcal{D} = \left\{ \left(\mathbf{b}_i, \mathbf{x}_i, \mathbf{t}_i, \{\mathbf{s}_i^{(j)}\}_{j=1}^{M} \right)_i \right\}_{i \in \{1,\ldots,|\mathcal{D}|\}}$ be the breast MRI data set. For the ith patient, $\mathbf{b}_i \in \{\text{left, right}\}$ indicates the left or right breast, $\mathbf{x}_i$ indicates the first subtraction volume computed by subtracting the first T1-weighted post-contrast DCE-MRI volume from the pre-contrast T1-weighted volume, $\mathbf{t}_i$ is the T1-weighted volume, and $\{\mathbf{s}_i^{(j)}\}_{j=1}^{M}$ is the voxel-wise annotation of the jth lesion (out of M total lesions) present in the volume $\mathbf{b}_i$. Due to the increasing interest in ultrafast MRI protocols to reduce costs [37] and acquisition time [38], our method performs the analysis using only the first subtraction image. We randomly divide the data set $\mathcal{D}$ into training, validation, and testing sets in a patient-wise manner, such that a patient belongs to only one of sets and the testing set contains approximately half of the lesions present in $\mathcal{D}$.

8.3.2 Detection Method

Our proposed detection method receives as input a breast DCE-MRI volume and outputs several bounding volumes that contain each of the lesions present in the input volume. The method is initialized with several large bounding volumes that are individually transformed by applying a sequence of actions that are decided by a policy π. To decide the transformation at each inference step, the policy π receives an embedding of the current bounding volume and outputs the transformation to be applied. In this case, we model π as a deep Q-network (DQN) [27], which is a neural network whose outputs indicate how optimal is each of the possible transformations that can be applied to the current bounding volume.

The input embedding to the DQN is represented by $\mathbf{x}(\mathbf{v})$ (with $\mathbf{v} \in \mathbb{R}^9$ denoting the three 3-D coordinates of the current bounding volume), and outputs the visual representation $\mathbf{o}$ to be used by the DQN, as follows:

$$\mathbf{o} = f(\mathbf{x}(\mathbf{v}), \theta_{\mathbf{f}}), \tag{8.1}$$

where $\theta_{\mathbf{f}}$ are the parameters of the embedding function $f(.)$. Note that the function $f(.)$ that computes the representation of the current observation must be able to do so for bounding volumes extracted from many sizes (scales) and locations. This fact allows our method to be trained using a large number of samples, extracted from a relatively small training set. More precisely, the function $f(.)$ is a 3D ResNet [32] that is trained to classify whether the input volume contains a lesion.

The output of the DQN indicates how optimal is each of the possible transformations to optimally transform the current bounding volume $\mathbf{x}(\mathbf{v})$ until it finds a lesion. The transformation is selected from the following set of actions: $\mathcal{A} = \{l_x^+, l_x^-, l_y^+, l_y^-, l_z^+, l_z^-, s^+, s^-, w\}$, where l indicates the translation of the current bounding volume in the positive l^+ or negative l^- direction in each of the x, y, z axes, s indicates the scaling the current bounding volume by shrinking s^+ or enlarging s^-, and w indicates the trigger action that terminates the search for lesions.

The training process targets the learning of the optimal weights of the policy π, which represents the transformation to be applied at each inference step to optimally find the lesions. During inference, we exploit the learnt policy π, where given the current bounding volume $\mathbf{x}(\mathbf{v})$ at each time step, we apply the transformation given by the policy π. The policy is exploited until the optimal action is the trigger or a maximum number of transformations are applied (no lesion detected). See Fig. 8.3 for a diagram of the method.

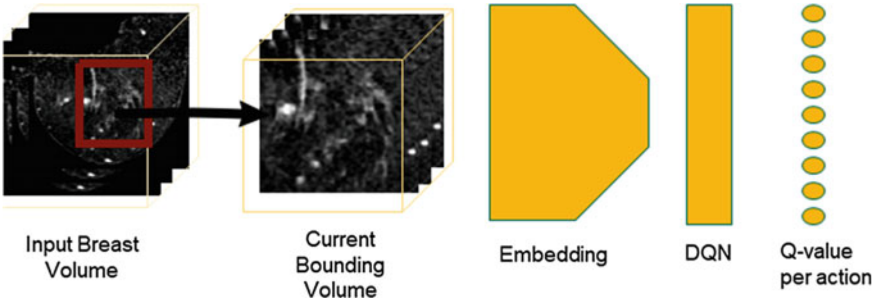


Fig. 8.3 Diagram for our method at each time step. For a given observation, we obtain its representation, which is the input to the Q-network that will output how optimal is each of the possible actions (transformations)

8.3.3 Training

The aim of the training phase is to learn the optimal weights $\boldsymbol{\theta}$ of the DQN representing the policy π that indicates how to transform a bounding volume until it targets a lesion. During M epochs and for each volume in the training set in each of the epochs, we place a large centered bounding volume as the initial observation. During a sequence of time steps, an action will be selected from the set of actions $\mathcal{A}$.

The selection of the actions depends on the epoch number and follows a modified ϵ-greedy strategy: with probability ϵ the method does exploration and with probability $1-\epsilon$ the method exploits the current policy. During exploration, with probability κ a random action from $\mathcal{A}$ is chosen, and with probability $1-\kappa$ a random action from the subset of $\mathcal{A}$ that will increase the Dice coefficient from the current to the transformed bounding volume is chosen. At the beginning of the training phase, we set $\epsilon = 1$, and we linearly decrease the value of ϵ until it reaches 0.1 at the end of the training phase. The rationale behind this balance of exploration–exploitation is that at the beginning of the training phase, the policy does not store any information on how to transform the initial $\mathbf{x}(\mathbf{v})$ to detect lesions. Thus, the actions are selected randomly to *explore* which actions are best. As the training phase progresses, the actions are selected according to the current policy, allowing to fine-tune the learnt knowledge in π.

After an action is applied to transform the current bounding volume from $\mathbf{x}(\mathbf{v}_t)$ to $\mathbf{x}(\mathbf{v}_{t+1})$, a reward will be yielded indicating how accurate the selected action was in order to achieve the final goal of lesion detection. The reward for the trigger action that terminates the search is defined as follows:

$$r(\mathbf{o}_t, a, \mathbf{o}_{t+1}) = \begin{cases} +\eta & \text{if} \quad Dice(\mathbf{x}(\mathbf{v})_{t+1}, \mathbf{s}) \geq \tau_w \\ -\eta & \text{otherwise} \end{cases}, \tag{8.2}$$

where η is decided experimentally, $Dice(.,.)$ measures the Dice coefficient between the final observation $\mathbf{x}(\mathbf{v})_{t+1}$ and the ground truth lesion, τ_w establishes a criteria for the minimum Dice score required to consider a lesion detected, and $\mathbf{o}_t$ and $\mathbf{o}_{t+1}$ indicate the embedding of $\mathbf{x}(\mathbf{v}_t)$ and $\mathbf{x}(\mathbf{v}_{t+1})$.

For the remaining of the actions $a \in \mathcal{A}$, the reward is defined by

$$r(\mathbf{o}_t, a, \mathbf{o}_{t+1}) = sign(Dice(\mathbf{x}(\mathbf{v})_{t+1}, \mathbf{s}) - Dice(\mathbf{x}(\mathbf{v})_t, \mathbf{s})), \tag{8.3}$$

where the $sign(.)$ function is valued +1 if the parameter is larger than 0 or -1 otherwise. The rationale for the different reward function depending on whether the action is the trigger is experimental: we found that a higher reward to terminate the search encourages the algorithm to improve its performance for lesion detection. In addition, the rationale behind the reward for the translation and scaling actions is intuitive: a positive reward if the action improved the localization of the lesion, a negative reward otherwise. Note that the use of the quantification of the reward improves the learning phase of the algorithm [35].

The aim of the training process is to learn which action should be applied at each time step of the sequence of actions that will optimally lead from an initial bounding volume to the detection of a lesion. This is achieved by maximizing the sum of discounted future rewards obtained during the sequence of applied transformations:

$$R_T = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}, \tag{8.4}$$

where $\gamma \in (0, 1)$ encourages the algorithm to detect the lesions in the least possible number of steps and $r_{t'}$ is the reward obtained at time step t' after applying an action.

Let $Q(\mathbf{o}, a)$ represent the action value function that can be interpreted as the expected sum of discounted future rewards yielded after selecting action a as the next transformation to be applied to $\mathbf{o}$:

$$Q(\mathbf{o}, a) = \mathbb{E}[R_t | \mathbf{o}_t = \mathbf{o}, a_t = a, \pi]. \tag{8.5}$$

In this context, the optimal action value function represented by $Q^{\star}(\mathbf{o}, a)$ is the maximum of the expected sum of discounted future rewards:

$$Q^{\star}(\mathbf{o}, a) = \max_{\pi} \mathbb{E}[R_t | \mathbf{o}_t = \mathbf{o}, a_t = a, \pi]. \tag{8.6}$$

Note that $Q^{\star}(\mathbf{o}, a)$ indicates which is the action a that would lead to the highest sum of future rewards, and therefore the optimal action that should be next chosen in order to maximize the sum of discounted future rewards.

Thus, the goal of the learning process becomes the estimation of $Q^{\star}(\mathbf{o}, a)$. In problems where the observation–action space is relatively small, the optimal Q-value $Q^{\star}(\mathbf{o}, a)$ can be estimated using the *Bellman* equation and the Q-Learning algorithm [39]:

$$Q_{i+1}(\mathbf{o}_t, a_t) = \mathbb{E}_{\mathbf{o}_{t+1}}\Big[r_t + \gamma \max_{a_{t+1}} Q(\mathbf{o}_{t+1}, a_{t+1})|\mathbf{o}_t, a_t\Big], \tag{8.7}$$

where t and $t + 1$ indicate the time (iteration) step.

However, the large observation–action space in our lesion detection task demands the use of a function approximator. We use the deep Q-network (DQN) [27] as a function approximator to estimate $Q(\mathbf{o}, a)$ for a given observation–action pair. The DQN can be trained by minimizing a loss that represents the mean squared error of the Bellman equation [27]:

$$L(\boldsymbol{\theta}_t) = \mathbb{E}_{(\mathbf{o}_t, a_t, r_t, \mathbf{o}_{t+1}) \sim U(\mathcal{E})}\Big[\Big(r_t + \gamma \max_{a_{t+1}} Q(\mathbf{o}_{t+1}, a_{t+1}; \boldsymbol{\theta}_t^-) - Q(\mathbf{o}_t, a_t; \boldsymbol{\theta}_t)\Big)^2\Big], \tag{8.8}$$

where $U(\mathcal{E})$ is a batch of experiences uniformly sampled from an experience replay memory $\mathcal{E}_t = \{e_1, \ldots, e_t\}$ and used for any update of the network. This experience replay memory stores past tuples $e_t = \{\mathbf{o}_t, a_t, r_t, \mathbf{o}_{t+1}\}$ that represent past experiences of selecting an action a_t to transform observation $\mathbf{o}_t$ to observation $\mathbf{o}_{t+1}$ yielding the reward r_t. The loss in (8.8) is computed using the reference of a target value defined by the target network with weights $\boldsymbol{\theta}_t^-$ and similar architecture as the DQN. The weights $\boldsymbol{\theta}_t^-$ of the target network are constant during each epoch and are updated with the values of the DQN at the end of every epoch. Note that the goal of the target network is to stabilize the training of the deep Q-network [27].

8.3.4 Inference

During inference, we exploit the optimal policy $\theta^\star$ learned as explained in Sect. 8.3.3 along with the embedding function $\mathbf{x}(\mathbf{v})$. We apply a sequence of actions that will transform initial bounding volumes covering different large portions of the input volume until a lesion is targeted (or a maximum number of steps are reached). At each time step t, we choose the optimal action:

$$a_t^\star = \arg\max_{a_t} Q(\mathbf{o}_t, a_t; \boldsymbol{\theta}^\star). \tag{8.9}$$

We process each initialization individually according to the learnt policy, and thus according to each of the actions that yield the largest expected sum of future rewards.

8.4 Experiments

In this section, we describe the data set and the experiments performed to assess our reinforcement learning based detection model. We also present the results obtained for the experiments.

8.4.1 Data Set

We evaluate our methods on a breast MRI data set of 117 patients [22]. We form the training, validation, and testing sets by splitting the data set in a patient-wise manner [19, 31]. The training set contains 45 patients and 57 lesions, the validation set contains 13 patients and 15 lesions, and the testing set contains 59 patients and 69 lesions. There are 38, 11, and 46 malignant and 19, 4, and 23 benign lesions in the training, validation, and testing sets, respectively. All lesions have been confirmed with a biopsy. Note that although patients suffer from at least one lesion, not all breasts contain lesions.

For every patient, there is a T1-weighted anatomical volume acquired without fat suppression, and a DCE-MRI dynamic sequence composed of the first subtraction volume. The procedure to acquire the dynamic sequence is as follows: (1) an initial pre-contrast T1-weighted volume is obtained with fat suppression before a contrast agent is injected into the patient, (2) the first post-contrast T1-weighted volume is acquired with fat suppression 45 seconds after the contrast agent was injected, and (3) the first subtraction volume is formed by subtracting the pre-contrast volume from the post-contrast volume. The acquisition matrices are 512 × 512 for the T1-weighted anatomical volume and 360 × 360 for the dynamic sequence with a slice thickness of 1 mm. All images were acquired with a 1.5 Tesla GE Signa HDxt scanner.

Preprocessing

We exclusively use the T1-weighted volume of each patient to extract the corresponding left and right breast regions. We employ Haytons' method [40] as in previous work on this data set [19, 22] to remove the pectoral muscle and automatically separate the breast region from the chest wall. It is important to note that the pectoral muscle is removed to reduce the number of false positive detections. Finally, as our method operates breast-wisely, the breast region is split into the left and right breasts and resized into a volume of size 100 × 100 × 50.

8.4.2 Experimental Setup

We evaluate the performance of our proposed reinforcement learning based detection method according to (1) the free response operating curve (FROC) that compares the true positive rate (TPR) in terms of the number of false positive detections and (2) the inference time employed per patient in an Intel Core i7 PC with 12 GB of RAM and a GPU Nvidia Titan X 12 GB. Note that a detected region is considered a true positive detection only if its Dice coefficient with respect to a lesion is equal or greater than 0.2 [19, 41].

The embedding function $f(.)$ is a 3D ResNet [32] of input size $100 \times 100 \times 50$ and is composed of five ResBlocks [42] (with an initial convolutional layer before each of them), two convolutional layers, and a fully connected layer. The input size to $f(.)$ is selected such that every lesion is visible in the input volume. The embedding $\mathbf{o}$ is the input to the last convolutional layer of $f(.)$ and is of size 2304. We train $f(.)$ by randomly selecting 8K positive and 8K negative bounding volumes at different locations and scales from the training data. A bounding volume is considered positive if its Dice coefficient with any lesion is larger than 0.6. We set this threshold higher compared to the criterion to decide whether a detection is positive (at least a Dice coefficient with respect to a lesion of 0.2) to assure that bounding volumes used for training the embedding function contain a relatively large portion of the lesion. Note that the parameters θ_f of $f(.)$ are not updated during the training of the DQN.

The policy π that models the DQN is a two-layer perceptron with 512 units each layer and nine output units corresponding to the nine Q-values, that is, $|\mathcal{A}|$. In order to train it, we choose a learning rate of 10^{-6} and use Adam optimizer [43]. The training of the DQN uses the loss function described in (8.8) with $\gamma = 0.9$ and is computed using a mini-batch of 100 experiences uniformly sampled from the experience replay memory $\mathcal{E}$, containing a maximum of 10K experiences. During training, the exploration–exploitation balance is guided by ϵ that is initialized at 1 and linearly decreases until 0.1 in 300 epochs. During exploration a random action or an action that will increase the Dice coefficient of the current bounding volume is selected with probability $\kappa = 0.5$. We empirically set the reward for the trigger action as $\eta = 10$ if $\tau_w = 0.2$ or $\eta = -10$ otherwise. The inference phase of our method uses 13 initial bounding volumes. The first one covers 75% of the whole input volume and is centered. Other eight initializations are of size $50 \times 50 \times 25$ and are placed at each of the eight corners of the input volume. Finally, four other initializations of size $50 \times 50 \times 25$ are placed at the centered intersections of the previous eight initializations of size $50 \times 50 \times 25$. During training and inference phases, a maximum of 20 transformations can be applied to detect a lesion, otherwise, no lesion is considered detected.

8.4.3 Experimental Results

Figure 8.4 and Table 8.1 present the results achieved by our proposed reinforcement learning based detection model. We compare our method against two baselines that are run on the same data set. We use the detection method proposed in [31] as a representative for exhaustive search methods. In [31], the authors propose cascade of three deep learning classifiers at different scales, where at each scale, the algorithm refines the detections obtained in the previous step. We also compare our results against another baseline proposed in [22], consisting of an unsupervised clustering method (mean-shift) to build a graph of contiguous regions to then apply structure learning to detect which of those clusters represent lesions. Note that in [22], the authors evaluate their method in the same data set as the one used in the work

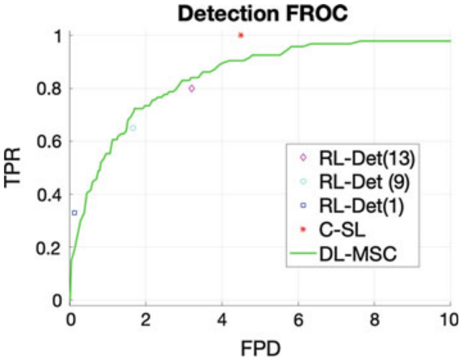


Fig. 8.4 FROC curve for the detection results of our method (labeled as RL-Det) and two baselines. Since we have detections, but not the probability of detections for each of the bounding volumes, we run our model with a different number of initializations (1, 9, and 13—indicated between parenthesis in the graph) to obtain its FROC representation. The FROC curve for the multi-scale exhaustive search based on deep learning method (labeled as DL-MSC) is represented by the green curve. The unsupervised clustering followed by structure learning baseline (labeled as C-SL) is plotted as a red point

Table 8.1 Detection results in the **test set** for our proposed reinforcement learning based detection method (row 1) in terms of the true positive rate (TPR), average number of false positive detections (FPD), and average inference time per patient. We compare our method against two baselines: (1) a multi-scale exhaustive search based on deep learning (row 2) [31] and (2) an unsupervised clustering followed by structure learning method (row 3) [22]

	Inference time	TPR	FPD
Reinforcement learning detection (Ours)	**92 ± 21 s**	**0.8**	**3.2**
Deep learning multi-scale cascade [31]	164 ± 137 s	0.8	2.8
Clustering + structure learning [22]	≈3600 s	1.0	4.5

presented in this chapter. However, the patient-wise partition of the data is different from ours.

Using the paired t-test, we test the significance of the difference in terms of the inference time per patient between our proposed reinforcement learning based detection model and the multi-scale exhaustive search based on deep learning approach in [31], obtaining $p \leq 9 \times 10^{-5}$. Given such a small p-value, we conclude the significant difference between the inference time of both methods. We also present qualitative results in Fig. 8.5.

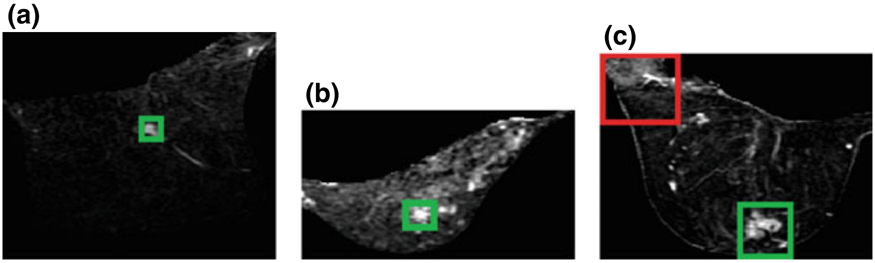


Fig. 8.5 Example of detections by our reinforcement learning based detection model. Images **a** and **b** show a true positive detection. Image **c** shows a true positive detection (green box and a false positive detection (red box)

8.5 Discussion

We observe from the results in Fig. 8.4 and Table 8.1 that our proposed reinforcement learning based detection model achieves comparable to state-of-the-art lesion detection results in a significantly reduced inference time compared to the baselines presented. More precisely, our method is 1.78 times faster than the quicker of the baselines [31], which is the multi-scale exhaustive search based on deep learning. We also noted that 90% of the inference time is spent resizing the current bounding volume to the input size (100 × 100 × 50) of the embedding function. Therefore, the bottleneck of our reinforcement learning based detection model lies in this resizing function, which could be optimized, and thus greatly reducing the inference time.

According to the results presented in Table 8.1, we observe that the variability of our method in terms of inference time is smaller compared to the multi-scale exhaustive search based on deep learning [31]. We believe this reduced variation is due to the ability of our reinforcement learning based detection model to focus on the most salient regions of a given bounding volume, whereas the cascade approach is not as effective at disregarding noisy bounding volumes that advance to the next detection scale to be analyzed at a higher resolution, increasing analysis time. Therefore, our model seems to be more robust to noisy inputs.

Compared to the unsupervised clustering followed by the structure learning baseline [22], we observe that our method does not achieve a TPR of 1 (Fig. 8.4). However, we believe that the faster inference time makes our approach more suitable to be deployed in clinical practice.

8.6 Conclusion and Future Work

In this chapter, we have presented a novel detection method that can increase the efficiency of the lesion detection stage in pre-hoc systems for breast screening from MRI. Our model is capable of achieving comparable to state-of-the-art lesion detection results from breast MRI 1.78 times faster than current exhaustive search methods. This speedup is achieved by transforming the initial large bounding volumes with a sequence of actions until the lesion is detected. We based our detection method on the DQN trained with reinforcement learning. The DQN receives as input an embedding of the current bounding volume and outputs the next action that should be applied to optimally find a lesion.

Future work includes the addition of a larger set of actions to change the aspect ratio of the initialization. We expect that small lesions can be better detected by deforming the cube into volumes that capture different sizes in different axes. By changing the bounding volume aspect ratio, we expect that detected lesions could be found with higher precision. Such increased flexibility of the bounding volume would allow us to consider a detection to be positive with a higher minimum Dice coefficient than 0.2, improving the quality of the detection (and probably the quality of the final diagnosis of a breast lesion). To achieve a similar TPR with a higher minimum Dice coefficient than 0.2, it might also be necessary to improve the quality of the embedding function. For example, we suggest that the criteria to consider a volume positive in the training of the embedding could be a minimum Dice coefficient of 0.8. Finally, we leave for the future the optimization of the implementation of the presented algorithm, which could possibly lead to a reduced inference time as well as the comparison with a breast lesion detector based on the U-net [34].

Acknowledgements This work was partially supported by the Australian Research Council project (DP180103232). IR acknowledges the Australian Research Council: ARC Centre for Robotic Vision (CE140100016) and Laureate Fellowship (FL130100102). We would like to thank Nvidia for the donation of a Titan Xp used in the development of this work.

References

1. Siegel RL, Miller KD, Jemal A (2019) Cancer statistics, 2019. CA Cancer J Clin 69(1):7–34
2. Siegel RL, Miller KD, Jemal A (2017) Cancer statistics, 2017. CA Cancer J Clin 67(1):7–30
3. Siegel RL, Miller KD, Jemal A (2016) Cancer statistics, 2016. CA: Cancer J Clin 66(1):7–30
4. Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A (2018) Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin 68(6):394–424
5. Smith RA, DeSantis CE (2018) Breast cancer epidemiology. Breast imaging
6. Lauby-Secretan B, Scoccianti C, Loomis D, Benbrahim-Tallaa L, Bouvard V, Bianchini F, Straif K (2015) Breast cancer screening-viewpoint of the IARC working group. N Engl J Med 372(24):2353–2358
7. Carter CL, Allen C, Henson DE (1989) Relation of tumor size, lymph node status, and survival in 24,740 breast cancer cases. Cancer 63(1):181–187
8. Park JH, Anderson WF, Gail MH (2015) Improvements in US breast cancer survival and proportion explained by tumor size and estrogen-receptor status. J Clin Oncol 33(26):2870
9. Siu AL (2016) Screening for breast cancer: US preventive services task force recommendation statement. Ann Intern Med 164(4):279–296
10. Kuhl CK, Strobel K, Bieling H, Leutner C, Schild HH, Schrading S (2017) Supplemental breast MR imaging screening of women with average risk of breast cancer. Radiology 283(2):361–370

11. Weigel S, Heindel W, Heidrich J, Hense HW, Heidinger O (2017) Digital mammography screening: sensitivity of the programme dependent on breast density. Eur Radiol 27(7):2744–2751
12. Saslow D, Boetes C, Burke W, Harms S, Leach MO, Lehman CD, Morris E, Pisano E, Schnall M, Sener S et al (2007) American cancer society guidelines for breast screening with MRI as an adjunct to mammography. CA Cancer J Clin 57(2):75–89
13. Mainiero MB, Moy L, Baron P, Didwania AD, Green ED, Heller SL, Holbrook AI, Lee SJ, Lewin AA, Lourenco AP et al (2017) ACR appropriateness criteria® breast cancer screening. J Am CollE Radiol 14(11):S383–S390
14. Seely J, Alhassan T (2018) Screening for breast cancer in 2018-what should we be doing today? Curr Oncol 25(Suppl 1):S115
15. Grimm LJ, Anderson AL, Baker JA, Johnson KS, Walsh R, Yoon SC, Ghate SV (2015) Interobserver variability between breast imagers using the fifth edition of the BI-RADS MRI lexicon. Am J Roentgenol 204(5):1120–1124
16. Yamaguchi K, Schacht D, Newstead GM, Bradbury AR, Verp MS, Olopade OI, Abe H (2013) Breast cancer detected on an incident (second or subsequent) round of screening MRI: MRI features of false-negative cases. Am J Roentgenol 201(5):1155–1163
17. Vreemann S, Gubern-Merida A, Lardenoije S, Bult P, Karssemeijer N, Pinker K, Mann R (2018) The frequency of missed breast cancers in women participating in a high-risk MRI screening program. Breast Cancer Res Treat 169(2):323–331
18. Meeuwis C, van de Ven SM, Stapper G, Gallardo AMF, van den Bosch MA, Willem PTM, Veldhuis WB (2010) Computer-aided detection (CAD) for breast MRI: evaluation of efficacy at 3.0 T. Eur Radiol 20(3):522–528
19. Maicas G, Bradley AP, Nascimento JC, Reid I, Carneiro G (2018) Pre and post-hoc diagnosis and interpretation of malignancy from breast DCE-MRI. arXiv:1809.09404
20. Vignati A, Giannini V, De Luca M, Morra L, Persano D, Carbonaro LA, Bertotto I, Martincich L, Regge D, Bert A et al (2011) Performance of a fully automatic lesion detection system for breast DCE-MRI. J Magn Reson Imaging 34(6):1341–1351
21. Gubern-Mérida A, Martí R, Melendez J, Hauth JL, Mann RM, Karssemeijer N, Platel B (2015) Automated localization of breast cancer in DCE-MRI. Med Image Anal 20(1):265–274
22. McClymont D, Mehnert A, Trakic A, Kennedy D, Crozier S (2014) Fully automatic lesion segmentation in breast MRI using mean-shift and graph-cuts on a region adjacency graph. J Magn Reson Imaging 39(4):795–804
23. 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, pp 91–99
24. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision, pp 21–37. Springer
25. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
26. Ribli D, Horváth A, Unger Z, Pollner P, Csabai I (2018) Detecting and classifying lesions in mammograms with deep learning. Sci Rep 8(1):4165
27. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529
28. Renz DM, Böttcher J, Diekmann F, Poellinger A, Maurer MH, Pfeil A, Streitparth F, Collettini F, Bick U, Hamm B et al (2012) Detection and classification of contrast-enhancing masses by a fully automatic computer-assisted diagnosis system for breast MRI. J Magn Reson Imaging 35(5):1077–1088
29. Amit G, Hadad O, Alpert S, Tlusty T, Gur Y, Ben-Ari R, Hashoul S (2017) Hybrid mass detection in breast MRI combining unsupervised saliency analysis and deep learning. In: International conference on medical image computing and computer-assisted intervention, pp 594–602. Springer

30. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
31. Maicas G, Carneiro G, Bradley AP (2017) Globally optimal breast mass segmentation from DCE-MRI using deep semantic segmentation as shape prior. In: 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017), pp 305–309. IEEE
32. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
33. 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, pp 234–241. Springer
34. Dalmış MU, Vreemann S, Kooi T, Mann RM, Karssemeijer N, Gubern-Mérida A (2018) Fully automated detection of breast cancer in screening MRI using convolutional neural networks. J Med Imaging 5(1):014502
35. Caicedo JC, Lazebnik S (2015) Active object localization with deep reinforcement learning. In: Proceedings of the IEEE international conference on computer vision, pp 2488–2496
36. Ghesu FC, Georgescu B, Mansi T, Neumann D, Hornegger J, Comaniciu D (2016) An artificial agent for anatomical landmark detection in medical images. In: International conference on medical image computing and computer-assisted intervention, pp 229–237. Springer
37. Mann RM, Kuhl CK, Moy L (2019) Contrast-enhanced MRI for breast cancer screening. J Magn Reson Imaging
38. van Zelst JC, Vreemann S, Witt HJ, Gubern-Merida A, Dorrius MD, Duvivier K, Lardenoije-Broker S, Lobbes MB, Loo C, Veldhuis W et al (2018) Multireader study on the diagnostic accuracy of ultrafast breast magnetic resonance imaging for breast cancer screening. Investig Radiol 53(10):579–586
39. Sutton RS, Barto AG (1998) Introduction to reinforcement learning, vol 135. MIT press, Cambridge (1998)
40. Hayton P, Brady M, Tarassenko L, Moore N (1997) Analysis of dynamic MR breast images using a model of contrast enhancement. Med Image Anal 1(3):207–224
41. Dhungel N, Carneiro G, Bradley AP (2015) Automated mass detection in mammograms using cascaded deep learning and random forests. In: 2015 International conference on digital image computing: techniques and applications (DICTA), pp 1–8. IEEE
42. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. In: European conference on computer vision, pp 646–661. Springer
43. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980

Chapter 9
Automatic Vertebra Labeling in Large-Scale Medical Images Using Deep Image-to-Image Network with Message Passing and Sparsity Regularization

Dong Yang, Tao Xiong and Daguang Xu

Abstract Efficient and accurate vertebra labeling in medical images is important for longitudinal assessment, pathological diagnosis, and clinical treatment of the spinal diseases. In practice, the abnormal conditions in the images increase the difficulties to accurately identify the vertebrae locations. Such conditions include uncommon spinal curvature, bright imaging artifacts caused by metal implants, and limited field of the imaging view, etc. In this chapter, we propose an automatic vertebra localization and labeling method with high accuracy and efficiency for medical images. First, we introduce a deep image-to-image network (DI2IN) which generates the probability maps for vertebral centroids. The DI2IN adopts multiple prevailing techniques, including feature concatenation and deep supervision, to boost its performance. Second, a message-passing scheme is used to evolve the probability maps from DI2IN within multiple iterations, according to the spatial relationship of vertebrae. Finally, the locations of vertebra are refined and constrained with a learned sparse representation. We evaluate the proposed method on two categories of public databases, 3D CT volumes, and 2D X-ray scans, under various pathologies. The experimental results show that our method outperforms other state-of-the-art methods in terms of localization accuracy. In order to further boost the performance, we add 1000 extra 3D CT volumes with expert annotation when training the DI2IN for CT images. The results justify that large databases can improve the generalization capability and the

Dong Yang and Tao Xiong contributed equally to this work.

D. Yang (✉)
Department of Computer Science, Rutgers University, New Brunswick, NJ, USA
e-mail: don.yang.mech@gmail.com

T. Xiong (✉)
Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, USA
e-mail: tao.xiong@jhu.edu

D. Xu (✉)
NVIDIA, Santa Clara, CA, USA
e-mail: cathy.xudg@gmail.com

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_9

performance of the deep neural networks. To the best of our knowledge, it is the first time that more than 1000 3D CT volumes are utilized for the anatomical landmark detection and the overall identification rate reaches 90% in spine labeling.

9.1 Introduction

Automatic and accurate landmark positioning and identification, e.g., for human spine detection and labeling, have been developed as key tools in 2D or 3D medical imaging, such as computed tomography (CT), magnetic resonance imaging (MRI), X-ray, etc. General clinical tasks such as pathological diagnosis, surgical planning [22], and postoperative assessment can benefit from such locate-and-name tool. Specific applications in human vertebrae detection and labeling include vertebrae segmentation [11, 32], fracture detection [7], tumor detection, registration [2, 26], statistical shape analysis [3, 17], etc. However, designing such an automatic and accurate vertebrae detection and labeling framework faces multiple challenges such as pathological conditions, image artifacts, and limited field-of-view (FOV), as shown in Fig. 9.1. Pathological conditions can arise from spinal curvature, fractures, deformity, and degeneration, of which spinal shapes are significantly different compared to normal anatomy. Image artifacts such as surgical metal implants change the image intensity distribution and greatly alter the appearance of vertebrae. Furthermore, limited FOVs given by spine-focused scans also add difficulty to the localization and identification of each vertebra due to the repetitive nature of these vertebrae and the lack of global spatial and contextual information. In order to address these challenges, an accurate and efficient spine localization algorithm is required for the potential clinical usage.

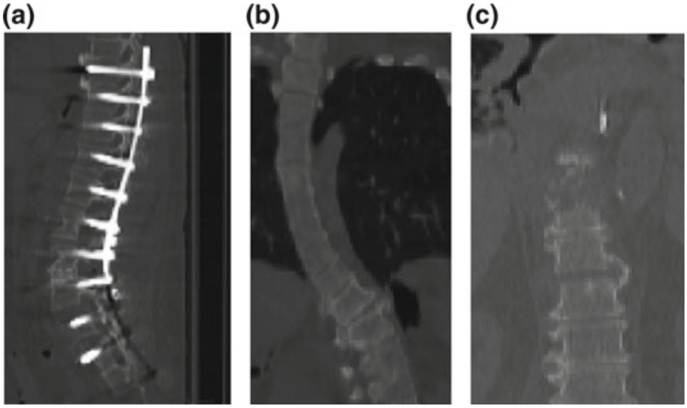


Fig. 9.1 Demonstration of uncommon conditions in CT scans. **a** Surgical metal implants **b** Spine curvature **c** Limited FOV

To meet the requirements of both accuracy and efficiency, many approaches have been presented in the recent decade. Generally, they can be divided into two categories: conventional machine learning based approaches and deep neural network based approaches. Schmidt et al. [21] proposed an efficient method for part-based localization of spine detection which incorporates contextual shape information into a probabilistic graphic model. Features for detecting parts are learned from the training database and detected by a multi-class classifier followed by a graphical model. Their method is evaluated on an MRI database and demonstrates robust detection even when some of vertebrae are missing in the image. Glocker et al. [8] presents an algorithm based on the regression forests and probabilistic graphical models. This two-stage approach is quantitatively evaluated on 200 CT scans, which achieves an identification rate of 81%. Furthermore, Glocker et al. [9] extends this vertebrae localization approach to address the challenge in pathological spine CT scans. Their approach is built on the supervised classification forests and evaluated on a challenging database of 224 pathological spine CT scans. It obtains an overall mean localization error of less than 9 mm with an identification rate of 70%, which outperforms state of the art on pathological cases at that moment. Recently, deep neural networks (DNN) have been achieving great progress in solving low-level computer vision tasks such as image classification, scene segmentation, and object detection. DNN has been highlighted in the research of landmark detection in medical imaging and demonstrated its outstanding performance compared to the conventional approaches. Chen et al. [4] proposed a joint learning model with convolutional neural networks (J-CNN) to effectively localize and identify the vertebrae. This approach, which is composed of a random forest classifier, a J-CNN and a shape regression model, improved the identification rate (85%) with a large margin with smaller localization errors in the same challenging database [9]. Suzani et al. [25] presented a fast automatic vertebrae detection and localization approach using deep learning. Their approach first extracts intensity-based features from the voxels in the CT scans, then applied a deep neural network on these features to regress the distance between the center of vertebrae and the reference voxels. It achieves a higher detection rate with faster inference but suffers from a larger mean error compared to other approaches [4, 9]. While most approaches are conducted on CT scans, Sun et al. [24] proposed the method of structured support vector regression for spinal angle estimation and landmark detection in 2D X-ray images. Their method has strong dependence on the hand-crafted features.

In order to take the advantage of deep neural networks and overcome the limitations in vertebrae detection, we propose an effective and automatic approach, as shown in Fig. 9.2, with the following contributions.

(a) Deep Image-to-Image Network for Voxel-Wise Regression

Compared to the approaches that require hand-crafted features from input images, the proposed deep image-to-image network (DI2IN) performs directly on the 2D X-ray images or 3D CT volumes and generates the multichannel probability maps which are associated with different vertebrae. The probability map itself explicitly indicates the location and type of vertebra. Additionally, the proposed DI2IN does

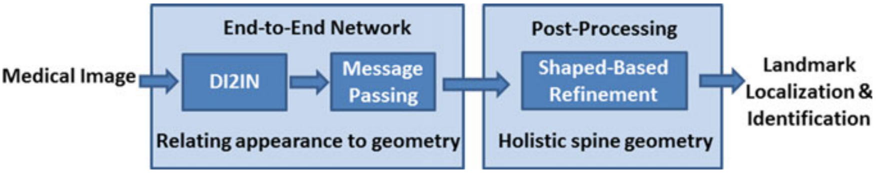


Fig. 9.2 Proposed method which consists of three major components: deep image-to-image network (DI2IN), message passing, and shape-based refinement

not adopt any classifier to coarsely remove outliers in preprocessing. By building the DI2IN in a fully convolutional manner, it is significantly efficient in terms of computation time, which sets it apart from the sliding window approaches.

(b) Response Enhancement with Message Passing

Although the proposed DI2IN usually provides high confident probability maps, sometimes it produces few false positives due to the similar appearance of vertebrae. The anatomical structure of spine provides a strong geometric prior for vertebral centroids. In order to fully explore such prior, we introduce a message-passing scheme which can communicate information of the neighborhood in space. At first, the chain-structured graph is constructed based on the prior on vertebra structure. The graph connection directly defines the neighborhood of each vertebra. Second, for the neighboring centroids, we learn the convolutional kernels between the probability maps. At inference, the probability maps from previous step are further convoluted with the learned kernels to help refine the prediction of neighbors' probability maps. The messages are passed via the convolution operations between neighbors. After a few iterations of message passing, the probability maps converge to a stable state. The probability maps of vertebrae are enhanced, and the issues, such as missing response or false positive response, are well compensated.

(c) Joint Refinement using Shape-Based Dictionaries

Given the coordinates of vertebrae, which are the outputs of DI2IN and message passing, we present a joint refinement approach using dictionary learning and sparse representation. In detail, we first construct a shape-based dictionary in the refinement, which embeds the holistic structure of the spine. Instead of learning a shape regression model [4] or Hidden Markov Model [8] to fit the spinal shape, the shape-based dictionary is simply built from the coordinates of spines in the training samples. The refinement can be formulated as an $\ell 1$-norm optimization problem and solved by the sparse coding approach in a predefined subspace. This optimization aims to find the best sparse representation of the coordinates with respect to the dictionary. By taking the regularity of the spine shape into account, ambiguous predictions and false positives are removed. Finally, the coordinates from all directions are jointly refined, which leads to further improvement in performance.

In the previous published version of this chapter [30], we validated our proposed method in a large-scale CT database. In this chapter version, we extend our work with more analysis, results, and implementation details. Several typical failure cases are well studied and solved with sufficient explanation. In addition, we validate our method in another large-scale database, 2D chest X-ray scans, which is also challenging due to similar imaging appearance. The experimental results show that our method has large potentials for any general applications of the anatomical landmark location.

The remainder of this chapter is organized as follows: In Sect. 9.2, we present the details of the proposed approach for vertebrae localization and identification, which consists of three subsections. In Sect. 9.3, we evaluate the proposed approach on both 2D X-ray and 3D CT databases. Our results are compared with other state-of-the-art works. Section 9.4 presents the conclusion.

9.2 Methodology

9.2.1 The Deep Image-to-Image Network (DI2IN) for Spinal Centroid Localization

In this section, we present a deep image-to-image network (DI2IN) model, which is a multilayer fully convolutional neural network [1, 13] for localization of the vertebral centroids. Figure 9.3 shows the configuration of 3D DI2IN used in the 3D CT images

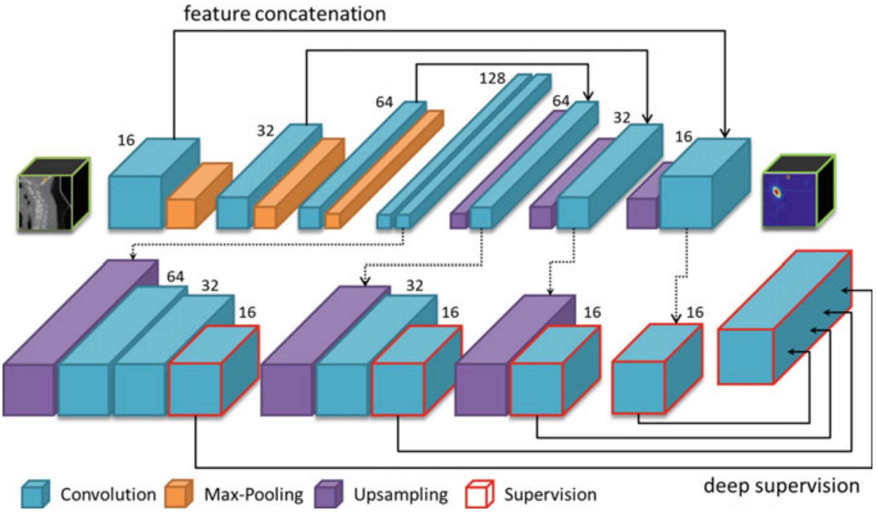


Fig. 9.3 Proposed deep image-to-image network (DI2IN) used in 3D CT images experiments. The front part is a convolutional encoder–decoder network with feature concatenation, while the backend is a multi-level deep supervision network. Numbers next to convolutional layers are the channel numbers

experiments. The 2D DI2IN used in X-ray experiments has similar structure except all layers are 2D-based. As can be seen, the deployment of DI2IN is symmetric and can be considered as a convolutional encoder–decoder model. DI2IN follows the end-to-end learning fashion, which also guarantees the efficiency at inference. For such purpose, the multichannel ground truth data is specially designed using the coordinates of vertebral centroids. The 3D Gaussian distribution $I_{\text{gt}} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\|\mathbf{x}-\mu\|^2/2\sigma^2}$ is defined around the positions of the vertebrae in each channel. Vector $\mathbf{x} \in \mathbb{R}^3$ denotes the voxel coordinate inside the volume, and vector μ is the ground truth position of each vertebra. Variance σ^2 is predefined, which controls the size of the Gaussian distribution. The prediction of each channel $I_{\text{prediction}}$ is corresponding to the unique vertebral centroid. It shares the same size with the input image. Thus, the whole learning problem is transformed into a multichannel voxel-wise regression. In the training process, we use the square loss of $\left\| I_{\text{gt}} - I_{\text{prediction}} \right\|^2$ in the output layer of each voxel. The reason that we define the centroid localization as a regression task instead of classification is that the highly unbalanced labeling of voxels is unavoidable in the classification approach, which may cause misleading classification accuracy.

The encoder part of the proposed network uses convolution, rectified linear unit (ReLU), and maximum pooling layers. The pooling layer is vital because it helps to increase the receptive field of neurons, while reducing the GPU memory consumption at the same time. With larger receptive field, each neuron in different levels considers richer contextual information, and therefore the relative spatial positions of the vertebral centroid are better understood. The decoder section consists of convolution, ReLU, and upsampling layers. The upsampling layer is implemented as the bilinear interpolation to amplify and densify the activation. It enables the voxel-wise end-to-end training scheme. In Fig. 9.3, the convolution filter size is $1 \times 1 \times 1$ at the final output layer (1×1 for 2D images), and $3 \times 3 \times 3$ for other convolution layers (3×3 for 2D images). The filter size of the maximum pooling layers is $2 \times 2 \times 2$ (2×2 for 2D images). The stride number in the convolution layer is set to one, so that each channel remains the same size. The stride number in the pooling layer is set to two which downsamples the size of feature maps by two in each dimension. The number of channels in each layer is illustrated next to the convolution layers in Fig. 9.3. In the upsampling layers, the input feature maps are upsampled by two in all directions. The network takes a 3D CT image (volume) or 2D X-ray scans as input and directly outputs probability maps associated with vertebral centroids within different channels. Our framework computes the probability maps and the center of gravity positions, which are more efficient than the methods of classification or regression methods in [4, 25].

Our DI2IN has adopted several popular techniques. We use feature concatenation (skip connection) in the DI2IN, which is similar to the references [14, 18]. The shortcut bridge is built directly from the encoder layers to the decoder layers. It forwards the feature maps of the encoder, then concatenates them to the corresponding layers of the decoder. The outcome of concatenation is used as input of the following convolution layers. Based on the design, the high- and low-level features are clearly combined to gain the benefits of local and global information into the network. In [29], the deep supervision in neural network depth monitoring enables excellent boundary detection and segmentation results. In this work, we introduce a more complex deep supervision method to improve the performance. Multiple branches are separated from the middle layers of the decoder in the master network. They upsample each input feature map to the same size of the image, followed by several convolution layers to match the channel number of ground truth data. The supervision happens at the end of each branch i and shares the same ground truth data in order to calculate the loss item l_i. The final output is determined by another convolution of the concatenation of all branches' outputs and the decoder output. The total loss l_{total} is a sum of the loss from all branches and that from the final output, as shown in the following equation:

$$l_{\text{total}} = \sum_i l_i + l_{\text{final}}. \tag{9.1}$$

9.2.2 Probability Map Enhancement with Message Passing

Given an input image I, the DI2IN usually generates a probability map $P\left(v_i|I\right)$ for the centroid coordinate v_i of vertebra i with a high confidence. The location with highest probability shall be marked as the prediction of v_i. However, the probability maps from DI2IN are not always perfect, which may result in errors in the vertebra location prediction. In the worst-case scenario, there are no clear responses in the corresponding probability maps for few vertebrae because the imaging appearance of those vertebrae is very similar. In order to handle the issue of the missing response and reduce the false positive response, we propose a message-passing scheme to enhance the probability maps from the DI2IN utilizing the spatial relationship of vertebrae (Fig. 9.4).

The concept of message-passing algorithm, also known as belief propagation, has been brought up on the graphical models for decades [27]. It is used to compute marginal distribution of each unobserved nodes (sum-product algorithm) or infer the mode of joint distribution (max-product algorithm). The algorithm has been prevailing in the field of computer vision for many applications [12, 15, 19]. The key idea is to pass mutual information between neighboring nodes for multiple iterations until convergence and enable the model to reach the global optimization. Similarly, we introduce a chain-structured graph based on the geometry of spine. Each node i represents a vertebral centroid, and has at most two neighboring nodes (vertebrae). We propose the following formulation to update the probability map $P\left(v_i|I\right)$ at the tth iteration of message passing.

$$P_{t+1}\left(v_i|I\right) = \frac{\alpha \cdot \frac{\sum_{j\in\partial i} m_{j\to i}}{|\partial i|} + P_t\left(v_i|I\right)}{Z} \tag{9.2}$$

$$= \frac{\alpha \cdot \frac{\sum_{j\in\partial i} P_t\left(v_j|I\right) * k\left(v_i|v_j\right)}{|\partial i|} + P_t\left(v_i|I\right)}{Z}, \tag{9.3}$$

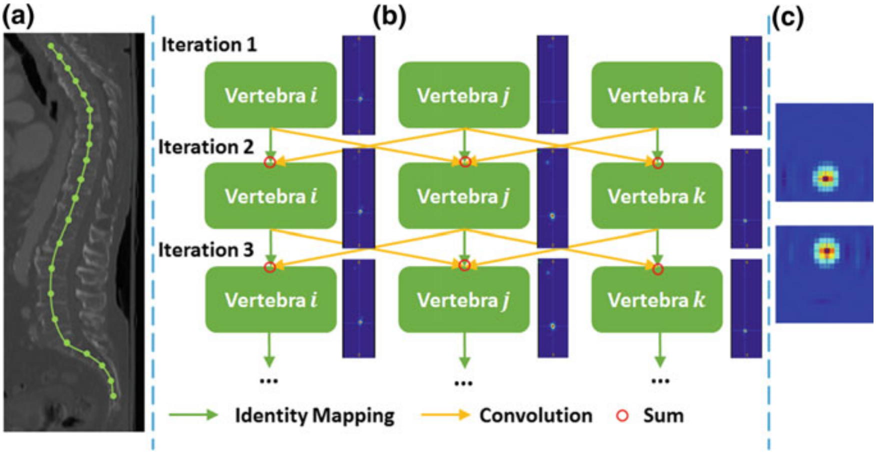


Fig. 9.4 **a** The chain-structure model for vertebra centroids shown in CT image; **b** Several iterations of message passing (landmarks represent vertebra centers): the neighbors' centroid probability maps help compensating the missing response of centroids. **c** Sample appearance of the learned kernels

where ∂i denotes the neighbors of node i in the graph, which is also corresponding to the adjacent vertebrae. α is a constant to adjust the summation weights between the passed messages and the previous probability map. Z is another constant for normalization. The message $m_{j \to i}$ is passed from node j to its neighboring node i, defined as $P_t\left(v_j|I\right) * k\left(v_i|v_j\right)$. $*$ denotes the convolution operation and the kernel $k\left(v_i|v_j\right)$ is learned from the ground truth Gaussian distributions of i and j. The convolution using the kernels actually shifts the probability map $P\left(v_i|I\right)$ toward $P\left(v_j|I\right)$. If DI2IN provides a confident response at the correct location of vertebra i, its message would be strong as well around the ground truth location of vertebra j after convoluting with the learned kernel. The messages from all neighbors are aggregated to enhance the response. After several iterations of message passing, the probability maps will converge to a stable state and the issue of the missing response would be compensated. The locations of vertebrae are determined at the peak positions of the enhanced probability maps at the moment. The underlying assumption of message passing is that DI2IN has given the correct and confident prediction for most vertebrae, which has already been proved in the experiments. Another advantage of the scheme is that it enables the end-to-end training (or fine-tuning) together with DI2IN for better optimization when the iteration number is fixed.

Several recent works have applied the similar message-passing schemes in different applications of the landmark detection. Chu et al. [5] introduced a similar message-passing method for human pose estimation (or body joint detection). However, the effectiveness of their implicit passing method may not be clear because it is conducted between feature maps of different landmarks. Our message passing is directly applied between the probability maps of vertebrae. It is more intuitive to understand how the kernel works and justify the quality of messages. Yang et al. [31] also proposed an analogous message-passing method for human pose estimation. They used the hand-crafted features, which usually have limitation on generalization, to describe the spatial relationship of landmarks. Our method uses the learnable kernels to describe the geometric relationship of vertebrae. The convolution kernels enable the pair-wise communication between vertebrae. Payer et al. [16] brought up a one-time message passing method for the anatomical landmark detection. Their passing scheme used dot product for message aggregation and mainly for outlier removal. But in our framework, the missing response is the major issue instead of noisy probability maps, then the dot product is not applicable for our passing scheme.

9.2.3 Joint Refinement Using Shape-Based Dictionaries

Given the probability maps generated by DI2IN and message-passing enhancement, it may still generate some outliers or false positives. For example, even though the DI2IN followed by message-passing enhancement outputs quite clear and reasonable probability maps, there is still false positive as shown in Fig. 9.5. This might arise from the low-resolution scans, image artifacts, or lack of global contextual information. In order to overcome these limitations, localization refinement has been introduced in many works [4, 8]. In [8], a hidden Markov model (HMM) with hidden states is

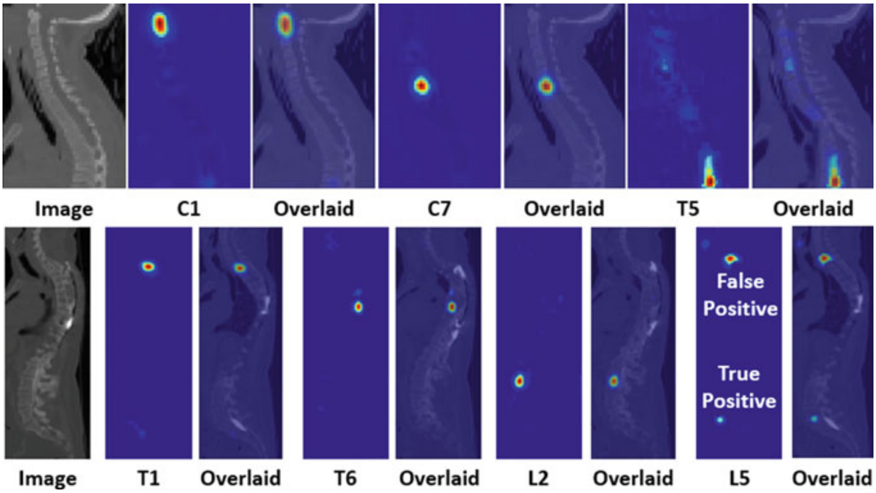


Fig. 9.5 Demonstration of two prediction examples in CT images. Only one representative slice is shown for demonstration purpose. Left: CT image. Middle: Output of one channel from the network. Right: Overlaid display. The most predicted responses are close to ground truth location. In the second row, a false positive response exists remotely besides the response at the correct location

defined for vertebrae location, appearance likelihoods, and inter-vertebra shape priors, which could yield a refined localization based on several thousands of candidate locations from the forest prediction. In [4], a quadratic polynomial curve is proposed to refine the coordinate in the vertical axis. By optimizing an energy function, the parameters for the shape regression model are learned to refine the coordinates of vertebrae. However, this model assumes the shape of the spine could be represented by a quadratic form. In addition, only coordinates in the vertical axis (head to foot direction) are refined.

Inspired by dictionary learning and sparse representation [20, 28], we design a joint refinement using a shape-based dictionary. For illustration purpose, we are using 3D representation in this section which is used in 3D CT experiments. Given a predefined shape-based dictionary, the coordinates are refined jointly in all x, y, and z axes. The refinement itself can be formulated as an ℓ_1-norm optimization and solved by the sparse coding approach. In detail, given the shape-based dictionary $\mathbf{D} \in \mathbb{R}^{M\times N}$ and the coordinate prediction $\mathbf{v} \in \mathbb{R}^N$, we propose a joint refinement algorithm as shown in Algorithm 1 to solve the sparse coefficient vector $\mathbf{a} \in \mathbb{R}^M$. Then the refined coordinate vector is defined as $\hat{\mathbf{v}} = \mathbf{Da}$. Specifically, the shape-based dictionary $\mathbf{D}$ is simply built by the coordinates of vertebrae in training samples. For example, the notation $\mathbf{D}_z$ indicates the shape-based dictionary associated with vertical axis or z direction. $\mathbf{d}_{z,i} \in \mathbb{R}^M$, which is a column of $\mathbf{D}_z$, is defined as $[z_{i,1}\ z_{i,2} \dots z_{i,26}]^T$. For instance, $z_{i,1}$ denotes the vertical ground truth coordinate of ith sample corresponding to vertebrae C_1. The $\mathbf{D}_x$ and $\mathbf{D}_y$ denote the dictionaries associated with x and y directions, respectively. They are both build in the same manner as $\mathbf{D}_z$. Similarly, $\mathbf{v_z}$, defined as $[v_{z,1}\ v_{z,2} \dots v_{z,26}]$, is the vertical coordinate of prediction. $\mathbf{v}_x$ and $\mathbf{v}_y$ are defined in the same manner.

In order to address the challenges such as outliers and limited FOV in spinal scans, we define the original space ϕ_0 and a subspace ϕ_1 in proposed refinement approach. The original space denotes a set which contains all indexes of 26 vertebrae. In our case, ϕ_0 contains the indexes from 1 to 26 which are corresponding to vertebra C_1 to S_2. Compared to the original space ϕ_0, the subspace ϕ_1 denotes a subset which only contains the partial indexes of ϕ_0. Based on the subspace ϕ_1, we define sub-dictionary $\mathbf{D}_{\phi_1}$ and sub-coordinate vector $\mathbf{v}_{\phi_1}$. Intuitively, $\mathbf{D}_{z,\phi_1}$ indicates the sub-dictionary associated with axis z, which is also simply a sub-matrix of $\mathbf{D}_{z,\phi_0}$. Basically, the optimization problem is solved based on the subspace ϕ_1 instead of the original space ϕ_0.

The details are demonstrated in Algorithm 1. Taking the shape regularity into account, we first find the maximum descending subsequence in the coordinate prediction $\mathbf{v}_z$ via dynamic programming. The reason we choose the vertical axis z to determine the maximum subsequence instead of $\mathbf{v}_x$ and $\mathbf{v}_y$ is the vertical axis of the human spine naturally demonstrates the most robust geometric shape compared to x- and y-axes. Based on the subspace ϕ_1 generated in Step 1, we further remove the indexes of neighboring vertebrae of which distance is too large or too small. Given the subspace ϕ_1, we define the sub-dictionary and sub-coordinate vector for each axis, respectively. Then, the ℓ_1 norm problem in Step 5 is optimized for x, y, and z individually based on the same subspace ϕ_1. Finally, all coordinates are refined based

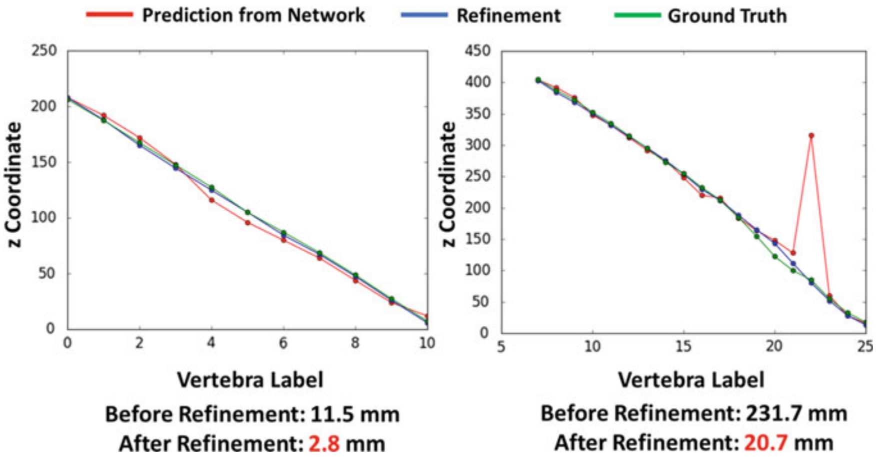


Fig. 9.6 Maximum errors of vertebra localization before and after the joint shape-based refinement in 3D CT experiments

on the original space ϕ_0 (i.e., $\mathbf{D}_{z,\phi_0}$ and $\mathbf{v}_{z,\phi_0}$). Intuitively, we remove the ambiguous outliers from the preliminary prediction and then jointly refine the coordinates without these outliers. Based on the subspace, we optimize the refinement problem to find the best sparse combination in the shape-based sub-dictionary. By taking the advantage of the original shape-based dictionary, all coordinates are refined jointly as shown in Fig. 9.6.

Algorithm 1 Joint Refinement using Shape-Based Dictionary

```
Require: The dictionary $\mathbf{D}_{x,\phi_0}$, $\mathbf{D}_{y,\phi_0}$, and $\mathbf{D}_{z,\phi_0} \in \mathbb{R}^{M \times N}$, the predicted coordinates vector $\mathbf{v}_x$, $\mathbf{v}_y$, and $\mathbf{v}_z$, the error threshold $\epsilon_1$ and $\epsilon_2$, and the coefficient $\lambda$. $M$ and $N$ indicate the number of landmarks and size of items in dictionary, respectively.
1: Given the predicted coordinates $\mathbf{v}_z$ from the DI2IN and message passing, the maximum descending subsequence is found via dynamic programming.
2: Add the indexes associated with the maximum descending subsequence into the set $\phi_1$.
3: Remove the pair of neighboring indexes if $|\mathbf{v}_{z,i} - \mathbf{v}_{z,j}| \leq \epsilon_1$ or $|\mathbf{v}_{z,i} - \mathbf{v}_{z,j}| \geq \epsilon_2$, where $i, j \in \phi_1$ and $|i - j| = 1$.
4: Based on the subspace $\phi_1$, define the sub-dictionary $\mathbf{D}_{x,\phi_1}$, $\mathbf{D}_{y,\phi_1}$, and $\mathbf{D}_{z,\phi_1}$ and the sub-coordinate predictions $\mathbf{v}_{x,\phi_1}$, $\mathbf{v}_{y,\phi_1}$, and $\mathbf{v}_{z,\phi_1}$.
5: Solve the optimization problem below by $\ell_1$ norm recovery for the vertical axis $z$:

   $$\min_{\mathbf{a}_z} \frac{1}{2} ||\mathbf{v}_{z,\phi_1} - \mathbf{D}_{z,\phi_1}\mathbf{a}_z||_2^2 + \lambda||\mathbf{a}_z||_1.$$

6: Solve the same optimization problem in Step 3 for $\mathbf{v}_{x,\phi_1}$ and $\mathbf{v}_{y,\phi_1}$, respectively.
7: Return the refined coordinate vectors $\hat{\mathbf{v}_x} = \mathbf{D}_{x,\phi_0}\mathbf{a}_x$, $\hat{\mathbf{v}_y} = \mathbf{D}_{y,\phi_0}\mathbf{a}_y$ and $\hat{\mathbf{v}_z} = \mathbf{D}_{z,\phi_0}\mathbf{a}_z$.
```

9.3 Experiments

In this section, we evaluate the performance of the proposed approach on two different and large databases. The first one has been introduced in [9] which contains 302 spine-focused 3D CT scans with various pathologies. These unusual appearances include abnormal curvature, fractures, and bright visual artifacts such as surgical implants in postoperative cases. In addition, the FOV of each 3D CT scan varies greatly in terms of vertical cropping. The whole spine is visible only in a few samples. Generally, most of the 3D CT scan contain 5–15 vertebrae. In particular, in order to boost the performance of our approach and validate that DNN favors more training data, we further introduce extra 1000+ 3D CT scans in our experiments. The second database consists of 1000+ 2D X-ray scans described in [6, 10, 23]. The ground truth of each database is marked on the center of each vertebra. The location and label of each ground truth are manually annotated by clinical experts. It should be noted that there is no overlap between the training and testing samples.

For 3D CT scans, there are two different settings that have been adopted in previous works [4, 9, 25]. The first setting uses 112 scans as training samples and another 112 scans as testing samples in [9, 25]. The second setting uses overall 242 scans as training samples and the other 60 scans as testing samples in [4, 9]. In order to fairly compare to other state-of-the-art works [4, 9, 25], we follow the same training and testing configurations, which are denoted as Set 1 and Set 2 in Tables 9.1 and 9.2, respectively. For 2D X-Ray scans, we adopt 1170 images as training samples and 50 images as testing samples.

Tables 9.1 and 9.2 summarize the quantitative results in terms of localization mean error, identification rate defined by [8] on Set 1 and Set 2, and other metrics. We compare our approach to other results reported in [4, 9, 25] on the 3D CT scans. In detail, “DI2IN”, “MP”, and “S” denote the deep image-to-image network, message passing, and shape-based refinement, respectively. “1000” indicates this model is trained with additional 1000 scans and evaluated on the same testing samples. In order to show the improvement of the performance, we list the results after each step for comparison.

Overall, our approach outperforms the state-of-art approaches [4, 9] by 13% and 6% on the same evaluation settings, respectively. For Set 1, the DI2IN itself improves the Id. Rates by a margin of 6% compared to the approach in [9]. Message passing and shape-based refinement further increase the Id. Rates to 77% and 80%, respectively. In addition, we have demonstrated that extra 1000 samples boost the performance to 83%. Similarly, the proposed approach also demonstrates better performance in Set 2 compared to [4, 9, 25]. Our approach has achieved Id. Rates of 85% and a localization mean error of 8.6 mm, which is better than the state-of-the-art work [4]. Taking advantage of extra 1000 samples, the Id. Rates have achieved 90%. Furthermore, other metrics such as standard deviation (Std), median (Med), and maximum (Max) also intuitively demonstrate the efficiency of our approach. For example, the maximum errors in both sets are significantly reduced to 42.3 and 37.9 mm. Figure 9.6 intuitively illustrates the refinement of proposed shape-based refinement in vertical direction.

Table 9.1 Comparison of localization errors in *mm* and identification rates among different methods for Set 1

Region	Method	Set 1				
		Mean	Std	Id.Rates (%)	Med	Max
All	Glocker et al. [9]	12.4	11.2	70	8.8	–
	Suzani et al. [25]	18.2	11.4	–	–	–
	Chen et al. [4]	–	–	–	–	–
	DI2IN	13.5	32.0	76	6.7	396.9
	DI2IN+MP	11.7	19.7	77	6.8	396.9
	DI2IN+MP+S	9.1	7.0	80	7.1	**42.3**
	DI2IN+1000	10.6	21.5	80	**5.5**	430.4
	DI2IN+MP+1000	9.4	16.2	82	6.0	430.4
	DI2IN+MP+S+1000	**8.5**	**7.7**	**83**	6.2	59.6
Cervical	Glocker et al. [9]	7.0	4.7	80	–	–
	Suzani et al. [25]	17.1	8.7	–	–	–
	Chen et al. [4]	–	–	–	–	–
	DI2IN+MP+S	6.6	3.9	83	–	–
	DI2IN+MP+S+1000	**5.8**	**3.9**	**88**	–	–
Thoracic	Glocker et al. [9]	13.8	11.8	62	–	–
	Suzani et al. [25]	17.2	11.8	–	–	–
	Chen et al. [4]	–	–	–	–	–
	DI2IN+MP+S	9.9	7.5	74	–	–
	DI2IN+MP+S+1000	**9.5**	**8.5**	**78**	–	–
Lumbar	Glocker et al. [9]	14.3	12.3	75	–	–
	Suzani et al. [25]	20.3	12.2	–	–	–
	Chen et al. [4]	–	–	–	–	–
	DI2IN+MP+S	10.9	9.1	80	–	–
	DI2IN+MP+S+1000	**9.9**	**9.1**	**84**	–	–

As shown in Fig. 9.6, the shape-based refinement takes the shape regularity of spine into account and removes the false positive coordinates. Specifically, the maximum error is significantly reduced.

Additionally, in order to demonstrate the robustness of our approach, we extend our experiments into a 2D X-ray database for training and evaluation. For 2D X-ray scans, the database [6, 10, 23] is randomly divided into two parts: 1170 scans for training samples and 50 scans for testing samples. It is the first time by our knowledge to evaluate such an approach on 2D X-ray scan for human vertebrae localization and identification task. We conducted experiments using the input images with two different resolutions: 0.70 and 0.35 mm. They are both resampled from the original database. Due to four times larger input and output data size, the DI2IN used in 0.35 mm experiment has less number of filters in the convolution layers comparing to

Table 9.2 Comparison of localization errors in mm and identification rates among different methods for Set 2

Region	Method	Set 2				
		Mean	Std	Id.Rates (%)	Med	Max
All	Glocker et al. [9]	13.2	17.8	74	–	–
	Suzani et al. [25]	–	–	–	–	–
	Chen et al. [4]	8.8	13.0	84	–	–
	DI2IN	13.6	37.5	76	5.9	410.6
	DI2IN+MP	10.2	13.9	78	5.7	153.1
	DI2IN+MP+S	8.6	7.8	85	5.2	75.1
	DI2IN+1000	7.1	11.8	87	4.3	235.9
	DI2IN+MP+1000	6.9	8.3	89	4.6	108.7
	DI2IN+MP+S+1000	**6.4**	**5.9**	**90**	**4.5**	**37.9**
Cervical	Glocker et al. [9]	6.8	10.0	89	–	–
	Suzani et al. [25]	–	–	–	–	–
	Chen et al. [4]	**5.1**	8.2	92	–	–
	DI2IN+MP+S	5.6	**4.0**	92	–	–
	DI2IN+MP+S+1000	5.2	4.4	**93**	–	–
Thoracic	Glocker et al. [9]	17.4	22.3	62	–	–
	Suzani et al. [25]	–	–	–	–	–
	Chen et al. [4]	11.4	16.5	76	–	–
	DI2IN+MP+S	9.2	7.9	81	–	–
	DI2IN+MP+S+1000	**6.7**	**6.2**	**88**	–	–
Lumbar	Glocker et al. [9]	13.0	12.5	80	–	–
	Suzani et al. [25]	–	–	–	–	–
	Chen et al. [4]	8.4	8.6	88	–	–
	DI2IN+MP+S	11.0	10.8	83	–	–
	DI2IN+MP+S+1000	**7.1**	**7.3**	**90**	–	–

the network in 0.70 mm experiment, as well as smaller batch size in training. Tables 9.3 and 9.4 demonstrate the performance of each step using our approach in terms of localization error and identification rates on input images with 0.70 and 0.35 mm resolution, respectively. Because most of vertebrae in X-ray scans belong to the thoracic region ($T_1 - T_{12}$), we only present the overall results instead of showing results in individual region. In detail, the DI2IN itself achieves a localization error of 8.4 and 7.8 mm and an identification rate of 80% and 82% on 0.70 and 0.35 mm resolution, respectively. We also introduce message passing scheme and shape-based refinement to evaluate the performance. The quality of performance is further improved compared to the DI2IN itself. The identification rate is also greatly improved after the introduction of message passing and shape-based refinement. Overall, the identification rate has been significantly increased by the message passing and refinement

Table 9.3 Comparison of localization errors in *mm* and identification rates among different methods for 0.70 mm X-ray set

Region	Method	0.7 mm				
		Mean	Std	Id.Rates (%)	Med	Max
All	DI2IN	8.4	14.7	80	3.7	283.4
	DI2IN+MP	7.7	9.6	82	3.7	45.9
	DI2IN+MP+S	**7.1**	**9.2**	**88**	**4.2**	**44.2**

Table 9.4 Comparison of localization errors in *mm* and identification rates among different methods for 0.35 mm X-ray set

Region	Method	0.35 mm				
		Mean	Std	Id.Rates (%)	Med	Max
All	DI2IN	7.8	12.1	82	3.1	114.0
	DI2IN+MP	7.4	9.8	84	3.6	57.9
	DI2IN+MP+S	**6.4**	**7.8**	**91**	**3.0**	**46.2**

and finally reached 91% on higher resolution settings. Our experiment demonstrates the proposed approach is able to achieve better performance on higher resolution database. Given more memory allocation and model capacity, our approach could further improve the quality of landmark detection.

Although our approach has achieved high identification rates on various pathological cases in both 3D CT scans and 2D X-ray scans, there are still some challenging cases. As shown in Fig. 9.7, the proposed approach occasionally fails to refine the coordinates which are jointly offset. This limitation might arise from special pathological cases, limited FOV and low-resolution input images. In our approach, the underlying assumption is that majority of the vertebra probability maps are confident and well distributed around the true locations, which is guaranteed by the powerful DI2IN. In order to address this limitation, more sophisticated network will be further studied in the future. From Fig. 9.8, we can see that vertebrae in thoracic region are comparatively harder to locate because those vertebrae share similar imaging appearance.

All experiments are conducted on a high-performance cluster equipped with an Intel 3.5 GHz CPU as well as a 12 GB NVIDIA Titan X GPU. In order to alleviate the pressure of memory, experiments on 3D CT scans and X-ray scans are conducted on a resolution of 4 mm, 0.7 mm, and 0.35 mm, respectively. The size of convolutional kernel in message passing is 23 × 23 × 23 for 3D volume and 49 × 49 for 2D images. The evaluation time of our approach is around three seconds per 3D CT case on average using GPU. In order to extract valid information from noisy probability maps, the response maps of DI2IN are compared to a heuristic threshold in an element-wise manner. Only channels with strong response are considered as valid outputs. Then

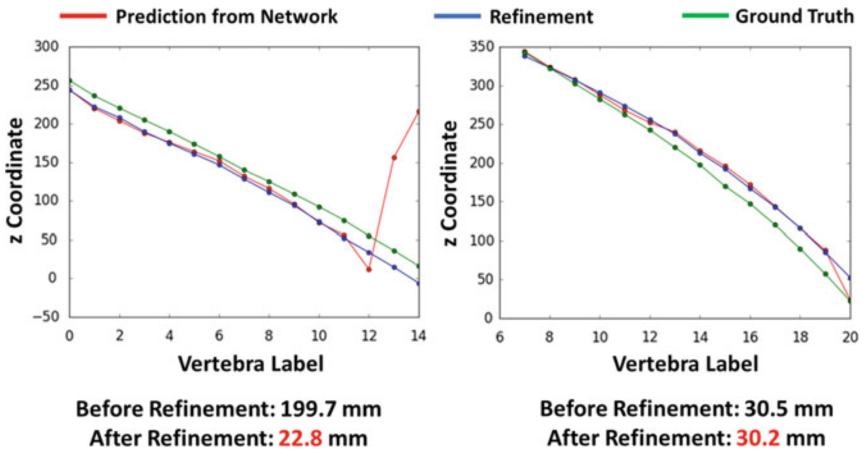


Fig. 9.7 Maximum errors of vertebra localization in challenging CT cases before and after the message passing and shape-based network refinement

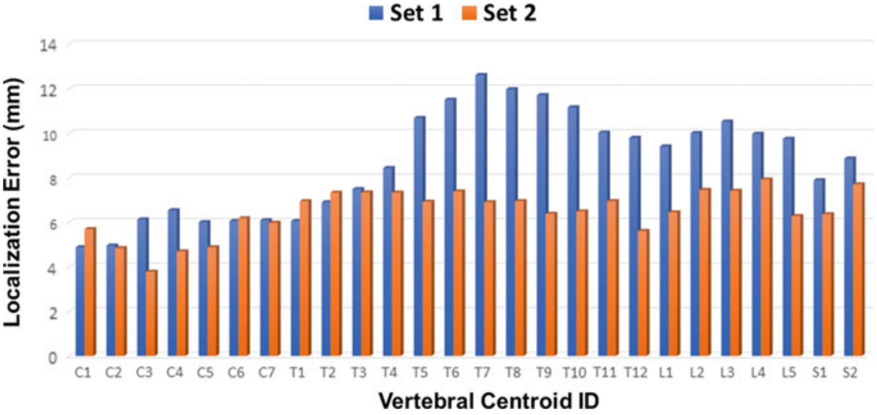


Fig. 9.8 Average localization errors (in *mm*) of the testing database Set 1 and Set 2 using the proposed methods with extra 1000 training volumes (line "DI2IN+MP+S+1000" in Tables 9.1 and 9.2). "C" is for cervical vertebrae, "T" is for thoracic vertebrae, "L" is for thoracic vertebrae, and "S" is for sacral vertebrae

vertebra centroids associated with these channels are identified to be present in the image. The vertebrae associated with other probability maps are identified as non-presented in the image. Therefore, we are able to localize and identify all vertebrae simultaneously in an efficient way.

9.4 Conclusion

We proposed and validated a novel method for vertebral labeling in medical images. The experimental results in both 3D CT volumes and 2D X-ray images show that the proposed method is effective and efficient comparing with the state-of-the-art methods. In addition, the extra 1000+ training data in 3D CT experiments evidently boost the performance of the proposed DI2IN, which further acknowledges the importance of large database for deep neural networks.

Acknowledgements We thank Dr. David Liu, Dr. Kevin Zhou, and Dr. Mingqing Chen who provided insight and expertise that greatly assisted this research. We acknowledge the data which has been provided by the Department of Radiology at University of Washington.

References

1. Badrinarayanan V, Kendall A, Cipolla R (2015) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. arXiv:1511.00561
2. Benameur S, Mignotte M, Parent S, Labelle H, Skalli W, de Guise J (2003) 3d/2d registration and segmentation of scoliotic vertebrae using statistical models. Comput Med Imaging Graph 27(5):321–337
3. Boisvert J, Cheriet F, Pennec X, Labelle H, Ayache N (2008) Geometric variability of the scoliotic spine using statistics on articulated shape models. IEEE Trans Med Imaging 27(4):557–568
4. Chen H, Shen C, Qin J, Ni D, Shi L, Cheng JC, Heng PA (2015) Automatic localization and identification of vertebrae in spine ct via a joint learning model with deep neural networks. In: International conference on medical image computing and computer-assisted intervention, pp 515–522. Springer International Publishing
5. Chu X, Ouyang W, Li H, Wang X (2016) Structured feature learning for pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4715–4723
6. Demner-Fushman D, Kohli MD, Rosenman MB, Shooshan SE, Rodriguez L, Antani S, Thoma GR, McDonald CJ (2015) Preparing a collection of radiology examinations for distribution and retrieval. J Am Med Inform Assoc 23(2):304–310
7. Genant HK, Wu CY, van Kuijk C, Nevitt MC (1993) Vertebral fracture assessment using a semiquantitative technique. J Bone Miner Res 8(9):1137–1148
8. Glocker B, Feulner J, Criminisi A, Haynor D, Konukoglu E (2012) Automatic localization and identification of vertebrae in arbitrary field-of-view ct scans. In: Medical image computing and computer-assisted intervention-MICCAI, pp 590–598
9. Glocker B, Zikic D, Konukoglu E, Haynor DR, Criminisi A (2013) Vertebrae localization in pathological spine ct via dense classification from sparse annotations. In: International conference on medical image computing and computer-assisted intervention, pp 262–270. Springer
10. Jaeger S, Candemir S, Antani S, Wáng YXJ, Lu PX, Thoma G (2014) Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. Quant Imaging Med Surg 4(6):475–477
11. Klinder T, Ostermann J, Ehm M, Franz A, Kneser R, Lorenz C (2009) Automated model-based vertebra detection, identification, and segmentation in ct images. Med Image Anal 13(3):471–482

12. Komodakis N, Paragios N, Tziritas G (2007) Mrf optimization via dual decomposition: message-passing revisited. In: 2007 IEEE 11th international conference on computer vision, ICCV 2007, pp 1–8. IEEE
13. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
14. Milletari F, Navab N, Ahmadi SA (2016) V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp 565–571. IEEE
15. Nowozin S, Lampert CH et al (2011) Structured learning and prediction in computer vision. Foundations and Trends® in Computer Graphics and Vision **6**(3–4):185–365
16. Payer C, Stern D, Bischof H, Urschler M (2016) Regressing heatmaps for multiple landmark localization using cnns. In: MICCAI 2, pp 230–238
17. Roberts M, Cootes T, Adams J (2005) Vertebral shape: automatic measurement with dynamically sequenced active appearance models. In: Medical image computing and computer-assisted intervention-MICCAI 2005, pp 733–740
18. 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, pp. 234–241. Springer
19. Ross S, Munoz D, Hebert M, Bagnell JA (2011) Learning message-passing inference machines for structured prediction. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR), pp 2737–2744. IEEE
20. Rubinstein R, Bruckstein AM, Elad M (2010) Dictionaries for sparse representation modeling. Proc IEEE 98(6):1045–1057
21. Schmidt S, Kappes J, Bergtholdt M, Pekar V, Dries S, Bystrov D, Schnörr C (2007) Spine detection and labeling using a parts-based graphical model. In: Information processing in medical imaging, pp 122–133. Springer
22. Schwarzenbach O, Berlemann U, Jost B, Visarius H, Arm E, Langlotz F, Nolte LP, Ozdoba C (1997) Accuracy of computer-assisted pedicle screw placement: an in vivo computed tomography analysis. Spine 22(4):452–458
23. Shiraishi J, Katsuragawa S, Ikezoe J, Matsumoto T, Kobayashi T, Komatsu KI., Matsui M, Fujita H, Kodera Y, Doi K (2000) Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. Am J Roentgenol **174**(1):71–74
24. Sun H, Zhen X, Bailey C, Rasoulinejad P, Yin Y, Li S (2017) Direct estimation of spinal cobb angles by structured multi-output regression. In: International conference on information processing in medical imaging, pp 529–540. Springer
25. Suzani A, Seitel A, Liu Y, Fels S, Rohling RN, Abolmaesumi P (2015) Fast automatic vertebrae detection and localization in pathological ct scans-a deep learning approach. In: International conference on medical image computing and computer-assisted intervention, pp 678–686. Springer
26. Tomazevic D, Likar B, Slivnik T, Pernus F (2003) 3-d/2-d registration of ct and mr to x-ray images. IEEE Trans Med Imaging 22(11):1407–1416
27. Wainwright MJ, Jordan MI et al (2008) Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning **1**(1–2):1–305
28. Wright J, Ma Y, Mairal J, Sapiro G, Huang TS, Yan S (2010) Sparse representation for computer vision and pattern recognition. Proc IEEE 98(6):1031–1044
29. Xie S, Tu Z (2015) Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision, pp 1395–1403
30. Yang D, Xiong T, Xu D, Huang Q, Liu D, Zhou SK, Xu Z, Park J, Chen M, Tran TD et al (2017) Automatic vertebra labeling in large-scale 3d ct using deep image-to-image network with message passing and sparsity regularization. In: International conference on information processing in medical imaging, pp 633–644. Springer

31. Yang W, Ouyang W, Li H, Wang X (2016) End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3073–3082
32. Yao J, Burns JE, Forsberg D, Seitel A, Rasoulian A, Abolmaesumi P, Hammernik K, Urschler M, Ibragimov B, Korez R et al (2016) A multi-center milestone study of clinical vertebral ct segmentation. Comput Med Imaging Graph 49:16–28

Chapter 10
Anisotropic Hybrid Network for Cross-Dimension Transferable Feature Learning in 3D Medical Images

Siqi Liu, Daguang Xu, S. Kevin Zhou, Sasa Grbic, Weidong Cai and Dorin Comaniciu

Abstract While deep convolutional neural networks (CNN) have been successfully applied for 2D image analysis, it is still challenging to apply them to 3D anisotropic volumes, especially when the within-slice resolution is much higher than the between-slice resolution and when the amount of 3D volumes is relatively small. On one hand, direct learning of CNN with 3D convolution kernels suffers from the lack of data and likely ends up with poor generalization; insufficient GPU memory limits the model size or representational power. On the other hand, applying 2D CNN with generalizable features to 2D slices ignores between-slice information. Coupling 2D network with LSTM to further handle the between-slice information is not optimal due to the difficulty in LSTM learning. To overcome the above challenges, 3D anisotropic hybrid network (AH-Net) transfers convolutional features learned from 2D images to 3D anisotropic volumes. Such a transfer inherits the desired strong generalization capability for within-slice information while naturally exploiting between-slice information for more effective modeling. We show the effectiveness of the 3D AH-Net on two example medical image analysis applications, namely, lesion detection from a digital breast tomosynthesis volume, and liver, and liver tumor segmentation from a computed tomography volume.

S. Liu (✉) · S. Grbic · D. Comaniciu
Digital Services, Digital Technology and Innovation, Siemens Healthineers,
Princeton, NJ, USA
e-mail: lsqshr@gmail.com

D. Xu
NVIDIA Corporation, Santa Clara, USA

S. K. Zhou
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

W. Cai
School of Information Technologies, University of Sydney, Darlington, NSW, Australia

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_10

10.1 Introduction

3D volumetric images (or volumes) are widely used for clinical diagnosis, surgical planning, and biomedical research. The 3D context information provided by such volumetric images are important for visualizing and analyzing the object of interest. However, given the added dimension, it is more time-consuming and sometimes harder to interpret 3D volumes than 2D images by machines. Many previous studies use convolutional neural networks (CNN) to extract the representation of structural patterns of interests in human or animal body tissues.

Due to the special imaging settings, many imaging modalities come with anisotropic voxels, meaning not all the three dimensions have equal resolutions. For example, in the 3D volumes of digital breast tomosynthesis (DBT), and sometimes computed tomography (CT), the image resolution in xy plane/slice (or within-slice resolution) is more than ten times higher than that of the z resolution (or between-slice resolution). Thus, the xy slices preserve much more information than the z dimension. In DBT images, only the spatial information within the xy-plane can be guaranteed. However, the 3D context between xy slices, even with a slight misalignment, still carries meaningful information for analysis (Fig. 10.1).

Directly applying 3D CNN to such images remains a challenging task due to the following reasons: (1) It may be hard for a small 3 × 3 × 3 kernel to learn useful features from anisotropic voxels, because of the different information density along each dimension. (2) Theoretically more features are needed in 3D networks compared to 2D networks. The capability of 3D networks is bounded by the GPU memory, constraining both the width and depth of the networks. (3) Unlike 2D computer vision tasks which nowadays can make use of the backbone networks pretrained

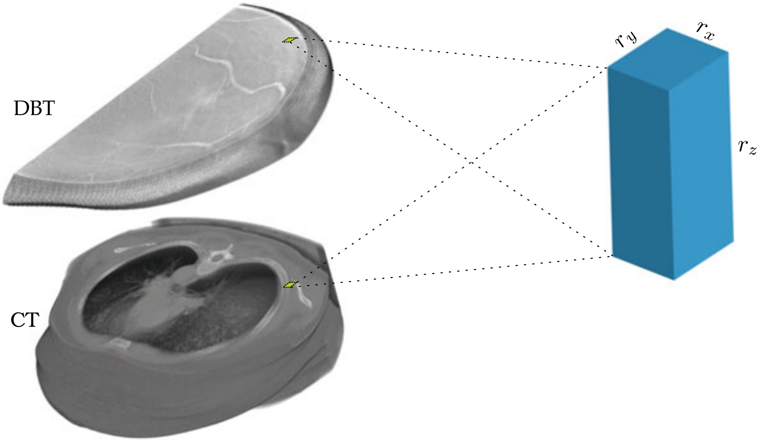


Fig. 10.1 The example anisotropic volumes of DBT and CT are shown in the left column. Such volumes contain voxels with much higher within-slice resolution $r_x \times r_y$ than the between-slice resolution r_z

using millions of 2D images [21], 3D tasks mostly have to train from scratch, and hence suffer from the lack of large 3D datasets. In addition, the high data variations make the 3D networks harder to be trained. Also, 3D CNNs trained on such small image datasets with relatively small 3D context are hard to generalize to unseen data.

Besides the traditional 3D networks built with $1 \times 1 \times 1$ and $3 \times 3 \times 3$ kernels, there are other methods for learning representations from anisotropic voxels. Some studies process 2D slices separately with 2D networks [14]. To make better use of the 3D context, more than one image slice is used as the input for 2D networks [12, 24]. The 2D slices can also be viewed sequentially by combining a fully convolutional network (FCN) architecture with convolutional LSTM to view the adjacent image slices as a time series to distill the 3D context from a sequence of abstracted 2D context [5]. There are also a few studies using anisotropic convolutional kernels to distribute more learning capability on the xy-plane than on the z-axis [2, 11, 22].

In this chapter, we present the 3D anisotropic hybrid network (AH-Net) [15] to learn informative features from images with anisotropic resolution. To obtain the 3D AH-Net, we first train a 2D fully convolutional ResNet [17] which is initialized with pretrained weights and uses multiple 2D image slices as inputs. The feature encoder of such a 2D network is then transformed into a 3D network by extending the 2D kernel with one added dimension. Then we add a feature decoder subnetwork to extract the 3D context. The feature decoder consists of anisotropic convolutional blocks with $3 \times 3 \times 1$ and $1 \times 1 \times 3$ convolutions. Different anisotropic convolutional blocks are combined with dense connections [8]. Similar to U-Net [20], we use skip connections between the feature encoder and the decoder. A pyramid volumetric pooling module [25] is stacked at the end of the network before the final output layer for extracting multiscale features. Since AH-Net can make use of 2D networks pretrained with large 2D general image datasets such as ImageNet [21], it is easier to train as well as to generalize. The anisotropic convolutional blocks enable it to exploit the 3D context. With end-to-end inference as a 3D network, AH-Net runs much faster than the conventional multichannel 2D networks regarding the GPU time required for processing each 3D volume.

10.2 Related Work

It is hard for conventional 3D neural networks with isotropic $3 \times 3 \times 3$ kernels to extract robust representations from 3D volumes with anisotropic resolution. The most intuitive approach is to resample the images to isotropic resolutions [16]. This would work when the difference between the three dimensions are small, and the spatial information between different slices is accurate. When the z resolution is much smaller than the xy resolution, the majority of voxels added by image resampling are redundant, thus introducing unnecessary extra computational cost. It may also result in a loss of information if downsampling happens in the xy-direction.

Instead of using 3D networks, some studies deal with the voxel anisotropy using 2D networks. DeepEM3D-Net [24] has only two 3D convolution layers to integrate

3D information in the early stages and performs 2D convolution for the rest of the following layers in an FCN. The input to DeepEM3D-Net is a stack of 2D image slices. The resultant 3D segmentation is obtained by concatenating the 2D output slices. HDenseNet [12] applies 2D networks on all image slices at first. Then a 3D DenseUNet is applied to the concatenated 3D output volume to obtain the final result. Different from our proposed network, HDenseNet does not have shared convolutions between the 2D and 3D networks. Also, we use anisotropic 3D convolutional blocks to replace the isotropic 3D convolutions.

A bidirectional convolutional LSTM (BDC-LSTM) and an FCN model are combined to view slices as a time series [5]. BDC-LSTM is trained to exploit the 3D contexts by applying a series of 2D convolutions on the xy-plane in a recurrent fashion to interpret 3D contexts while propagating contextual information in the z-direction. The FCN model is used for extracting the initial 2D feature maps which are used as the inputs to BDC-LSTM. The final output is obtained from the BDC-LSTM model with a softmax layer. Though the idea of fusing the 2D features to maintain the between-slice consistency is similar to our proposed method, we believe this can be achieved with stacked anisotropic convolution blocks, which are easier to train and to generalize than the convolutional LSTM.

Some studies use 3D convolutional kernels with anisotropic sizes to distribute more learning capability to the xy-plane. For example, $9 \times 9 \times 5$ convolutions are used in [2]. However, large convolution kernels would bring higher computational cost. Two more recent studies [11, 18, 22] use small kernels to simulate the large anisotropic kernels. The convolution modules in [11] starts with a $3 \times 1 \times 1$ convolution, followed by two $3 \times 3 \times 3$ convolutions. Similar to our work, all the isotropic convolutions are replaced by $3 \times 3 \times 1$ and $1 \times 1 \times 3$ convolutions in [18, 22]. Several possible designs of combining the $3 \times 3 \times 1$ and $1 \times 1 \times 3$ kernels are discussed in a recent paper [18] that focuses on video learning. Our network is different to the ones in [18, 22] since we use the anisotropic 3D convolutions only in the feature decoder while the encoder is locked with pretrained weights transferred from a 2D network. It allows the proposed AH-Net to use any 2D fully convolutional networks pretrained on large-scale datasets for initializing the encoder network. In [23], the authors show that the network with pretrained network could be significantly helpful to train 3D models for volumetric segmentation.

10.3 Anisotropic Hybrid Network

The AH-Net consists of a feature encoder and a feature decoder. The encoder, transformed from a 2D network, is designed for extracting the deep representations from 2D slices with high resolution. The decoder built with densely connected blocks of anisotropic convolutions is responsible for exploiting the 3D context and maintaining the between-slice consistency. The network training is performed in two stages: the encoder is learned, then the 3D decoder is added and fine-tuned with the encoder parameters locked. To perform end-to-end hard-voxel mining, we use the focal loss (FL) originally designed for object detection [13].

10.3.1 Learning a Multichannel 2D Feature Encoder

We train a 2D multichannel global convolutional network (MC-GCN) similar to the architecture proposed in [17] to extract the 2D within-slice features at different resolutions, as shown in Fig. 10.2. In this chapter, we choose the ResNet50 model [7] as the backbone network which is initialized by pretraining with the ImageNet images [21], although other pretrained networks would work similarly. The network is then fine-tuned with 2D image slices extracted from the 3D volumes. The input

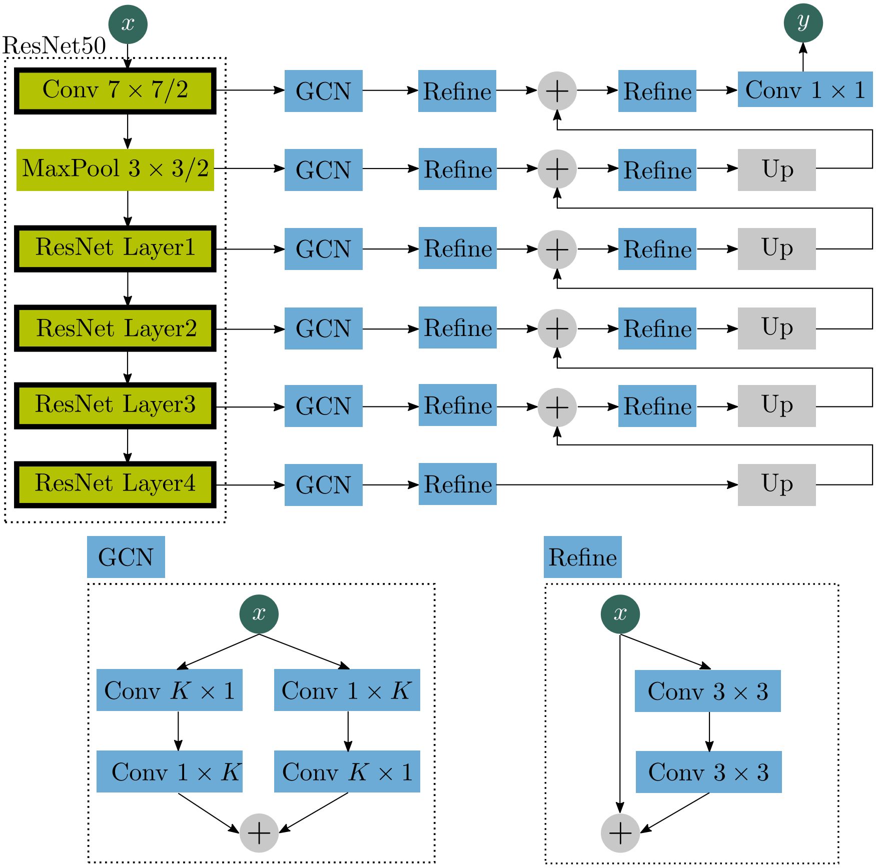


Fig. 10.2 The network architecture for pretraining the 2D encoder network multichannel global convolutional neural network (MC-GCN). The ResNet50 is used as the backbone network, initialized with ImageNet images. The global convolutional network modules and refinement modules [17] are added to the encoder network to increase the receptive field during the pretraining as well as to increase the output response map to the original resolution. Conv $K \times K/S$ represents a convolution layer with the kernel size K and the stride size S in each dimension. The upsampling module (Up) consists of a Conv 1×1 projection layer and a bilinear upsampling layer

to this network is three neighboring slices (treated as RGB channels). Thus, the entire architecture of the ResNet50 remains unchanged. The multichannel 2D input could enable the 2D network to fuse the between-slice context at an early stage. A decoder is added to accompany the encoder to upscale the response map to the original resolution. We choose the decoder architecture with the global convolutional networks (GCN) and refinement blocks [17]. The GCN module simulates a large $K \times K$ convolutional kernel by decomposing it into two 1D kernels ($1 \times K$ and $K \times 1$). Two branches containing the 1D kernels permuted in different orders are merged by summation. The output of each GCN module contains the same number of output maps as the final outputs. The large kernels simulated by GCNs ensure that the network has a large receptive field at each feature resolution. Each refinement block contains two 3×3 convolutions with a ReLU activation in the middle. The input of each refinement block is also added to its output to form a residual connection. At the end of each encoder resolution level, the features are fed into GCN modules with the kernel sizes of 63, 31, 15, 9, 7, 5, respectively. The output features are fed into a refinement block and summed with the features upsampled from a lower resolution level. The summed features are fed into another refinement block and upsampled with a 1×1 convolution and a bilinear upsampling layer. The final output has the same resolution as the image input. The decoder has only a small number of parameters with little computational cost. The lightweight decoder makes the encoder features easier to be transferred to the 3D AH-Net since majority of the feature learning relies on the encoder network.

10.3.2 Transferring the Learned 2D Net to 3D AH-Net

The architecture of the proposed 3D anisotropic hybrid network (AH-Net) is shown in Fig. 10.3. After the 2D MC-GCN network converges, we extract the parameters of its encoder and transfer them to the corresponding encoder layers of AH-Net. The decoder part of the 2D MC-GCN is discarded and instead, we design a new decoder for the AH-Net that consists of multiple levels of densely connected blocks, followed by a pyramid volumetric pooling module. The parameters of the new decoder are randomly initialized. The input and output of AH-Net are now 3D patches, similar to other conventional 3D CNN. The transformation of convolution tensors from 2D to 3D is illustrated in Fig. 10.4, which aims to perform 2D convolutions on 3D volumes slice by slice in the encoder part of AH-Net.

10.3.2.1 Notations

A 2D convolutional tensor is denoted by $T^i_{n \times m \times h \times w}$, where n, m, h, and w, respectively, represent the number of output channels, the number of input channels, the height, and width of the ith convolution layer. Similarly, a 3D weight tensor is denoted by $T^i_{n \times m \times h \times w \times d}$ where d is the filter depth. We use $P^{(b,a,c,d)}(T_{a \times b \times c \times d})$ to denote

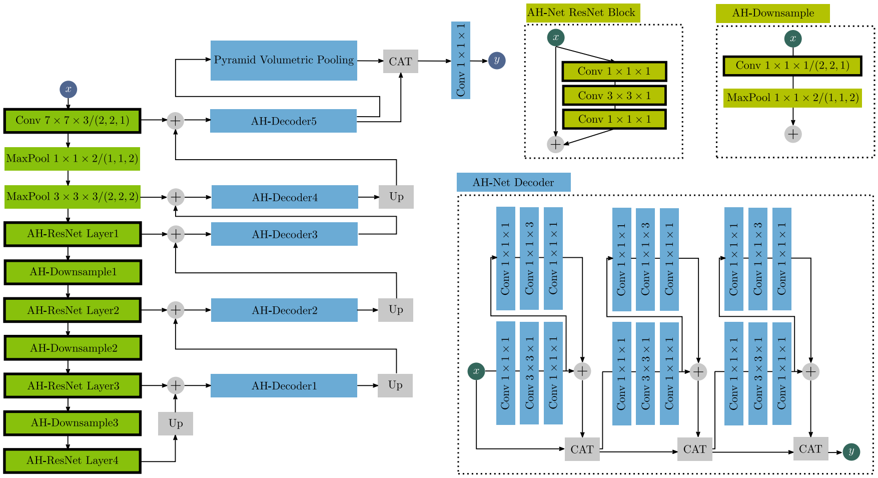
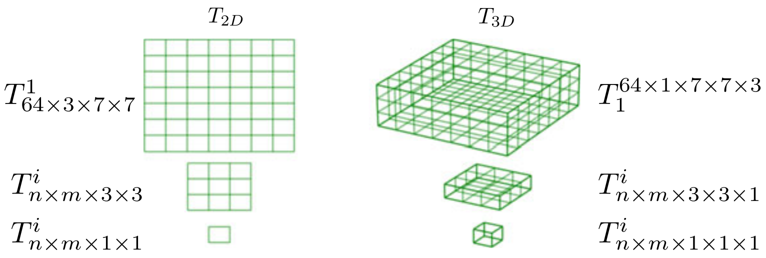


Fig. 10.3 The architecture of 3D AH-Net. The feature encoder with AH-ResNet blocks is transferred from the pretrained 2D network with 1 × 1 × 1 and 3 × 3 × 1 convolutions. The features are then processed with the AH-Net decoders which are designed with 3 × 3 × 1 and 1 × 1 × 3 convolutional blocks. Feature summation is used instead of concatenation as in [4] to support more feature maps with less memory consumption. The pyramid pooling [25] is used for extracting the multiscale feature responses. We hide the batch normalization [9] and ReLu layers for brevity. The weights of the blocks with black borders are transformed from the 2D MC-GCN



T_{2D}
T_{3D}
$T^1_{64\times3\times7\times7}$
$T_1^{64\times1\times7\times7\times3}$
$T^i_{n\times m\times3\times3}$
$T^i_{n\times m\times3\times3\times1}$
$T^i_{n\times m\times1\times1}$
$T^i_{n\times m\times1\times1\times1}$

Fig. 10.4 Transforming the 2D convolutional weight tensor T^{2D} to 3D T^{3D}, where m and n are the number of features and channels of a layer, respectively. The first layer weight tensor $T^1_{64\times3\times7\times7}$ is transformed to $T^1_{64\times1\times7\times7\times3}$. The other convolutional kernels are transformed by adding an extra dimension

the dimension permutation of a tensor $T_{a\times b\times c\times d}$, resulting in a new tensor $T_{b\times a\times c\times d}$ with the first and second dimensions switched. $P^{(a,*,b,c,d)}(T_{a\times b\times c\times d})$ adds an identity dimension between the first and second dimensions of the tensor $T_{a\times b\times c\times d}$ and gives $T_{a\times1\times b\times c\times d}$. We define a convolutional layer as Conv $K_x \times K_y \times K_z/(S_x, S_y, S_z)$, where K_x, K_y, and K_z are the kernel sizes; S_x, S_y, and S_z are the stride step size in each direction. Max pooling layers are denoted by MaxPool $K_x \times K_y \times K_z/(S_x, S_y, S_z)$. The stride is omitted when a layer has a stride size of one in all dimensions.

10.3.2.2 Input Layer Transform

The input layer of the 2D MC-GCN contains a convolutional weight tensor $T^1_{64\times 3\times 7\times 7}$ inherited from its ResNet50 backbone network. The 2D convolutional tensor $T^1_{64\times 3\times 7\times 7}$ is transformed into 3D as

$$P^{(1,*,3,4,2)}(T^1_{64\times 3\times 7\times 7}) = T^1_{64\times 1\times 7\times 7\times 3} \tag{10.1}$$

in order to form a 3D convolution kernel that convolves three neighboring slices. To keep the output consistent with the 2D network, we only apply stride 2 convolutions on the xy-plane and stride 1 on the third dimension. This results in the input layer Conv $7 \times 7 \times 3/(2, 2, 1)$. To downsample the z-dimension, we use a MaxPool $1 \times 1 \times 2/(1, 1, 2)$ to fuse every pair of the neighboring slices. An additional MaxPool $3 \times 3 \times 3/(2, 2, 2)$ is used to keep the feature resolution consistent with the 2D network.

10.3.2.3 ResNet Block Transform

All the 2D convolutional tensors $T^i_{n\times m\times 1\times 1}$ and $T^i_{n\times m\times 3\times 3}$ in the ResNet50 encoder are transformed as

$$P^{(1,2,3,4,*)}(T^i_{n\times m\times 1\times 1}) = T^i_{n\times m\times 1\times 1\times 1} \tag{10.2}$$

and

$$P^{(1,2,3,4,*)}(T^i_{n\times m\times 3\times 3}) = T^i_{n\times m\times 3\times 3\times 1}. \tag{10.3}$$

In this way, all the ResNet Conv $3 \times 3 \times 1$ blocks as shown in Fig. 10.3 only perform 2D slice-wise convolutions on the 3D volume within the xy-plane. The original downsampling between ResNet blocks is performed with Conv $1 \times 1/(2, 2)$. However, in a 3D volume, a Conv $1 \times 1 \times 1/(2, 2, 2)$ skips a slice for every step on the z-dimension. This would miss important information when the image only has a small number of slices along the z-dimension, especially for detection tasks. We therefore use a Conv $1 \times 1 \times 1/(2, 2, 1)$ followed by a MaxPool $1 \times 1 \times 2/(1, 1, 2)$ to downsample the 3D feature maps between the ResNet blocks as shown in the AH-Downsample block in Fig. 10.3. This MaxPooling simply takes the maximum response along the z-direction between two neighboring slices. Unlike the previous studies that avoided downsampling along the z-direction [11], we find it important for allowing the use of large and deep networks on 3D data with limited GPU memory.

10.3.3 Anisotropic Hybrid Decoder

Accompanying to the transformed encoder, an anisotropic 3D decoder subnetwork is added to exploit the 3D anisotropic image context. In the decoder, anisotropic convolutional blocks with Conv 1 × 1 × 1, Conv 3 × 3 × 1, and Conv 1 × 1 × 3 are used. The features are passed into an xy bottleneck block at first with a Conv 3 × 3 × 1 surrounded by two layers of Conv 1 × 1 × 1. The output is then forwarded to another bottleneck block with a Conv 1 × 1 × 3 in the middle and summed with itself before forwarding to the next block. This anisotropic convolution block decomposes a 3D convolution into 2D and 1D convolutions. It receives the inputs from the previous layers using a 2D convolution at first, preserving the detailed 2D features. Conv 1 × 1 × 3 mainly fuses the within-slice features to keep the z-dimension output consistent.

Three anisotropic convolutional blocks are connected as the densely connected neural network [8] using feature concatenation for each resolution of encoded features. Similar to LinkNet [4], the features received from each resolution of the encoder are first projected to match the number of features of the higher encoder feature resolution using a Conv 1 × 1 × 1. They are then upsampled using the 3D tri-linear interpolation and summed with the encoder features from a higher resolution. The summed features are forwarded to the decoder blocks in the next resolution.

At the end of the decoder network, we add a pyramid volumetric pooling module [25] to obtain multi-scaled features. The output features of the last decoder block are first downsampled using four different Maxpooling layers, namely, MaxPool 64 × 64 × 1, MaxPool 32 × 32 × 1, MaxPool 16 × 16 × 1, and MaxPool 8 × 8 × 1 to obtain a feature map pyramid. Conv 1 × 1 × 1 layers are used to project each resolution in the feature pyramid to a single response channel. The response channels are then interpolated to the original size and concatenated with the features before downsampling. The final outputs are obtained by applying a Conv 1 × 1 × 1 projection layer on the concatenated features.

Training AH-Net using the same learning rate on both the pretrained encoder and the randomly initialized decoder would make the network difficult to optimize. To train the 3D AH-Net, all the transferred parameters are locked at first. Only the decoder parameters are fine-tuned in the optimization. All the parameters can be then fine-tuned altogether afterward to the entire AH-Net jointly. Though it is optional to unlock all the parameters for fine-tuning afterward, we did not observe better performance.

10.4 Experimental Results

To demonstrate the efficacy and efficiency of the proposed 3D AH-net, we conduct two experiments, namely, lesion detection from a digital breast tomosynthesis (DBT) volume and liver tumor segmentation from a computed tomography (CT) volume. We use ADAM [10] to optimize all the compared networks with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We use the initial learning-rate 0.0005 to fine-tune the 2D multichannel GCN. Then, the learning rate is increased to 0.001 to fine-tune the AH-Net after the 2D network is transferred. We find that 3D networks need a larger learning-rate to converge within a reasonable amount of time. All the networks are implemented in Pytorch (http://pytorch.org).

10.4.1 Breast Lesion Detection from DBT

We use an in-house database containing 2809 3D DBT volumes acquired from 12 different sites globally. DBT is an advanced form of mammography, which uses low-dose X-rays to image the breast. Different from 2D mammography that superimposes 3D information into one 2D image, DBT creates 3D pictures of the breast tissue, and hence allows radiologists to read these pictures and detect breast cancer more easily, especially in dense breast tissues. The xy-plane of DBT images has a high spatial resolution of $0.085\,\text{mm} \times 0.085\,\text{mm}$ which is much larger than the z-dimension of 1 mm. The structures in the z-dimension are not only is compressed during the imaging process, but the 3D volumetric information also has large variations due to imaging artifacts.

We have experienced radiologists annotate and validate the lesions in DBT volumes, which might contain zero to several lesions. Each lesion is approximately annotated with a 3D bounding box. To train the proposed networks as lesion detection networks, we generate 3D multivariant Gaussian heatmaps that have the same sizes as the original images as

$$f(\mathbf{p}) = \sum_{\mu_\mathbf{i}, \Sigma_\mathbf{i}} \frac{\mathbf{exp}(-\frac{1}{2}(\mathbf{p} - \mu_\mathbf{i})^\mathbf{T} \Sigma_\mathbf{i} (\mathbf{p} - \mu_\mathbf{i}))}{\sqrt{\mathbf{det}(2\pi \Sigma_\mathbf{i})}}, \tag{10.4}$$

where $\mathbf{p}$ is a 3D coordinate x, y, z; $\mu_\mathbf{i}$ is the center coordinate of each lesion 3D bounding box; and $\Sigma_\mathbf{i}$ is the covariant matrix of the i-th Gaussian determined by the height, width, and depth of the 3D bounding box. Please note that we do not directly predict the bounding box coordinates as the general object detection methods such as Faster RCNN [19] because it is sometimes challenging to define the exact boundary of a breast lesion. Also, the voxel-wise confidence maps of lesion presence could be more helpful for clinical decision support than bounding boxes.

We randomly split the database into the training and the testing sets as described in Table 10.1. A volume or a 3D patch is considered positive if at least one lesion is annotated by the radiologist. We ensure the images from the same patient could only be found either in the training or the testing set. For training, we extract $256 \times 256 \times 32$ 3D patches. 70% of the training patches are sampled as positives with at least one lesion included, considering the balance between the voxels within and without

Table 10.1 The numbers of volumes (#Volumes), lesion-positive volumes (#Positive) and lesions (#Lesions) in the evaluated DBT dataset

	#Volumes	#Positives	#Lesions
Train	2678	1111	1375
Test	131	58	72

Table 10.2 The number of convolutional layers (#Conv Layers) and model float parameters (#Parameters), respectively, in 2D-UNet, 3D-UNet, ResNet50, GCN, and AH-Net. ResNet50 is shown here as a reference to be compared with GCN with a simple decoder added

Network	#Conv Layers	#Parameters
2D-UNet	15	28,254,528
3D-UNet	15	5,298,768
*ResNet50	53	23,507,904
GCN	94	23,576,758
AH-Net	123	27,085,500

Table 10.3 The GPU inference time (ms) of different networks on a 384 × 256 × 64 volume computed by averaging 1000 inferences with a NVIDIA GTX 1080Ti

	2D U-Net	3D U-Net	MC-GCN	3D AH-Net
ms	699.3	2.3	775.2	17.7

a breast lesion. The patches are sampled online asynchronously with the network training to form the mini-batches.

Along with the proposed networks, we also train 2D and 3D U-Nets with the identical architecture and parameters [3, 20] as two baseline comparisons. The 2D U-Net is also trained with input having three input channels. The 3D U-Net is trained with the same patch sampling strategies as the AH-Net. All the networks are trained till convergence then the L2 loss function is replaced with the Focal Loss [13] for hard-voxel mining. The number of convolutional layers and parameters is shown in Table 10.2. Using 2D networks, such as the MC-GCN and the 2D U-Net, to process 3D volumes involves repeatedly feeding duplicated images slices. Thus, they could be slower than the 3D networks when they are used for processing 3D volumes. We measure the GPU inference time of four networks by forwarding a 3D DBT volume of size 384 × 256 × 64 1000 times on an NVIDIA GTX 1080Ti GPU. The time spent on operations such as volume slicing is not included in the timing. The mean GPU time (ms) is shown in Table 10.3. The GPU inference of AH-Net is 43 times faster than MC-GCN though AH-Net has more parameters. The speed gain could be brought mostly by avoiding repetitive convolutions on the same slices required by multichannel 2D networks.

Non-maximal suppression is performed on the network output map to obtain the lesion locations. The network responses at the local maximal voxels are considered

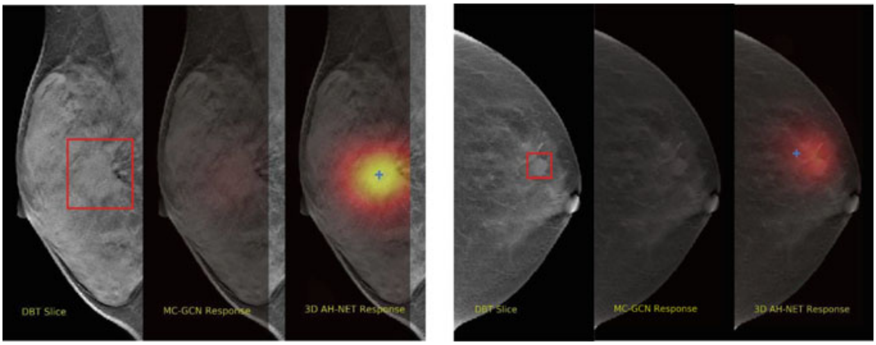


Fig. 10.5 Two example cases that AH-Net could detect the lesions that MC-GCN missed using the identical encoder weights

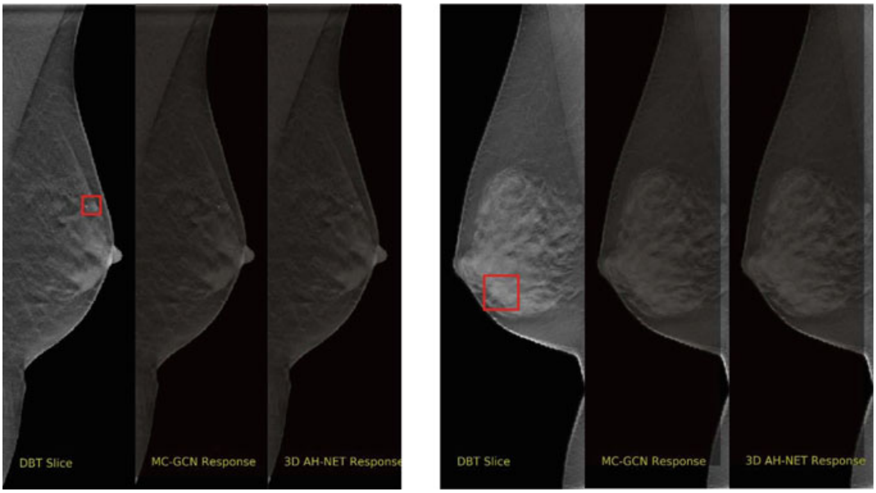


Fig. 10.6 Two example breast lesions that neither MC-GCN nor AH-Net was able to detect

as the confidence scores of the cancerous findings. Figure 10.5 shows some visual comparisons of the network's output on two example cases that AH-Net could detect the lesions missed by MC-GCN. Figure 10.6 shows two example cases with lesions surrounded by dense breast tissues that neither MC-GCN nor AH-Net was able to detect.

By altering a threshold to filter the response values, we can control the balance between the false positive rate (FPR) and true positive rate (TPR). The lesion detected by the network is considered a true positive finding if the maximal point resides in a 3D bounding box annotated by the radiologist. Similarly, if a bounding box contains a maximal point, we consider it is detected by the network. The maximal points are otherwise considered as false positive findings. We evaluate the lesion detection performance by plotting the free response operating characteristic (FROC)

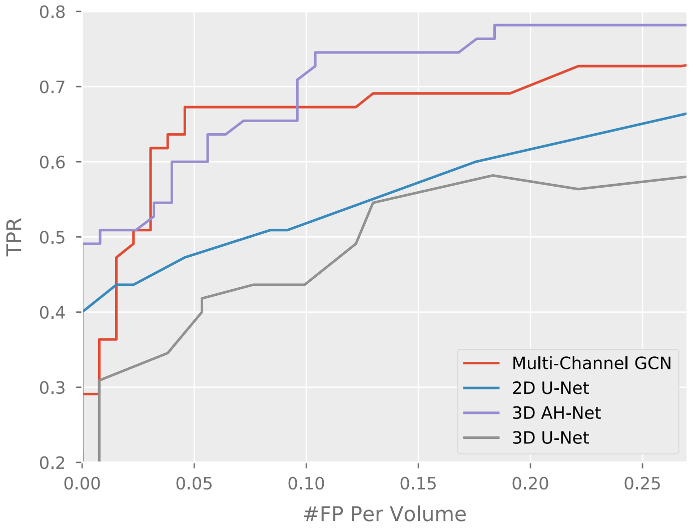


Fig. 10.7 The free response operating characteristic (FROC) curves regarding the lesion detection performance

curves, which measures the true positive rate (TPR) against the number of false positive (#FP) allowed per volume. TPR represents the percentage of lesions that have been successfully detected by the network. FPR represents the percentage of lesions that the network predicted that are false positives. As shown in Fig. 10.7, the proposed AH-Net outperforms both the 2D and 3D U-Net with large margins. Compared to the performance of the 2D network (multichannel GCN), the 3D AH-Net generates higher TPR for a majority of thresholds, except the region around 0.05 per volume false positives. It is noticeable that AH-Net also obtains nearly 50% TPR even when only 0.01 false positive findings are allowed per volume. Interestingly, the performance of 3D-UNet is slightly worse than that of 2D-UNet, though the DBT volumes have three dimensions. This might be caused by the anisotropic resolution of DBT images and the limited number of parameters constrained by the GPU memory. The FROC numbers are summarized in Table 10.4.

Table 10.4 The quantitative metrics of the compared networks on the DBT dataset. True positive rate (TPR) sampled at five different numbers of false positive (FP) findings allowed are shown in the first five columns

	FP = 0.01	FP = 0.05	FP = 0.10	FP = 0.15	FP = 0.20	FP = 0.25
2D U-Net	0.4238	0.4767	0.5181	0.5723	0.6166	0.6506
3D U-Net	0.2448	0.3877	0.4381	0.5592	0.5738	0.5733
GCN	0.3385	**0.6727**	0.6727	0.6909	0.7018	0.7272
AH-Net	**0.4931**	0.6000	**0.7272**	**0.7454**	**0.7818**	**0.7818**

10.4.2 Liver and Liver Tumor Segmentation from CT

The second evaluation dataset was obtained from the liver lesion segmentation challenge in MICCAI 2017 (lits-challenge.com), which contains 131 training and 70 testing 3D contrast-enhanced abdominal CT scans. Liver lesions are one of the commonest cancer worldwide. It is estimated that 28 920 people will die of liver lesion and 40 710 new cases will be diagnosed in 2017 [1]. Automatic segmentation of liver and lesion is challenging due to the heterogeneous and diffusive appearance of both liver and lesions. Also, the number, shape, and location of the lesions vary a lot among different volumes. The data and ground-truth masks were provided by various clinical sites around the world. The ground-truth masks contain both liver and lesion labels. Most CT scans consist of anisotropic resolution: the between-slice resolution ranges from 0.45 to 6.0 mm while the within-slice resolution varies from 0.55 to 1.0 mm. All scans cover the abdominal regions but may extend to head and feet. Other than the liver lesion, other diseases may also exist in these data, which further increases the task difficulty.

In preprocessing, the abdominal regions are truncated from the CT scans using the liver center biomarker detected by a reinforcement learning based algorithm [6]. While this step makes the network concentrate on the targeting region, its accuracy is not critical as we choose a relatively large crop region which usually ranges from the middle of the lung to the top of the pelvis. The image intensity is truncated to the range of $[-125, 225]$ HU based on the intensity distribution of liver and lesion in the training data. Due to the limited number of training data, we applied random rotation (within ±20 degree in the xy-plane), random scaling (within ±0.2 in all directions), and random mirror (within xy-plane) to reduce overfitting.

We first train the MC-GCN with pretrained ResNet50 as the backbone network. The input size of stacked 2D slices is 512 × 512 with three channels. After convergence, the weights of the encoder part of MC-GCN are transformed to the corre-

Table 10.5 The liver lesion segmentation (LITS) challenge results with the dice global (DG) and dice per case (DPC). Please refer to the challenge leaderboard for the latest results (lits-challenge.com/#results)

	Lesion		Liver	
Method	DG	DPC	DG	DPC
leHealth	0.794	**0.702**	0.964	0.961
H-DenseNet [12]	0.829	0.686	0.965	0.961
hans.meine	0.796	0.676	0.963	0.960
medical	0.783	0.661	0.951	0.951
deepX	0.820	0.657	0.967	**0.963**
superAI	0.814	0.674	–	–
GCN	0.788	0.593	0.963	0.951
3D AH-Net	**0.834**	0.634	**0.970**	**0.963**

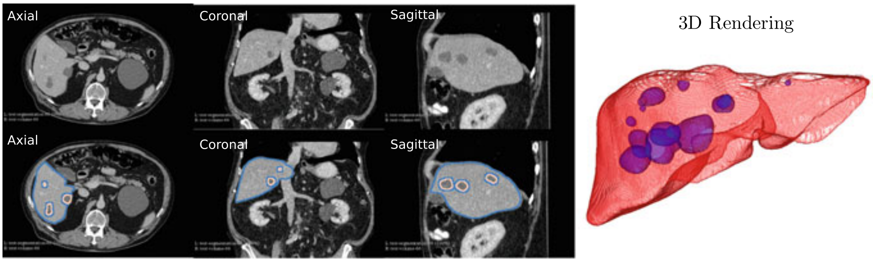
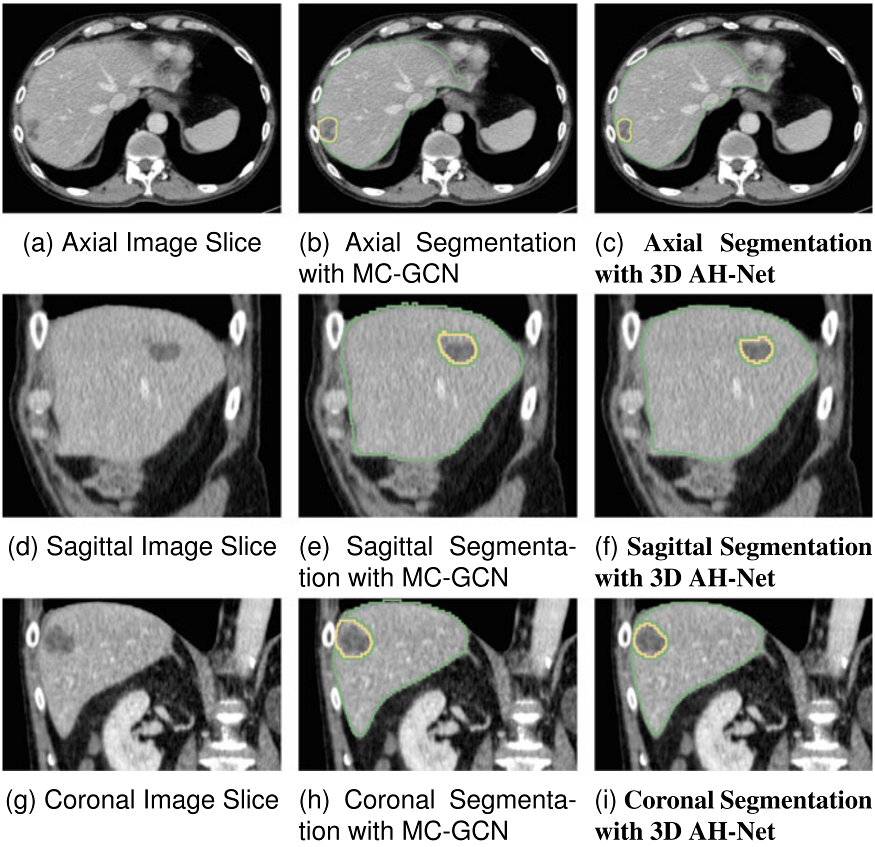


Fig. 10.8 The example liver lesion segmentation results from 3D AH-Net. The segmented contours of liver (blue) and liver lesion (pink) are overlaid on three slices viewed from different orientations (Axial, Coronal and Sagittal). The segmentations are rendered in 3D on the right



(a) Axial Image Slice

(b) Axial Segmentation with MC-GCN

(c) **Axial Segmentation with 3D AH-Net**

(d) Sagittal Image Slice

(e) Sagittal Segmentation with MC-GCN

(f) **Sagittal Segmentation with 3D AH-Net**

(g) Coronal Image Slice

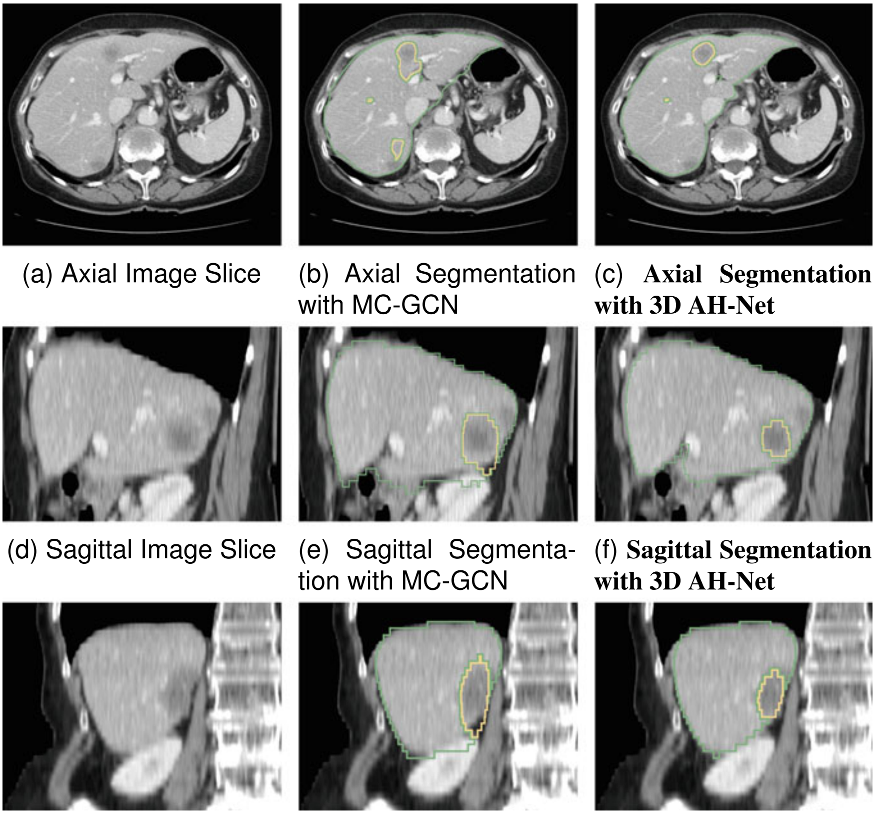
(h) Coronal Segmentation with MC-GCN

(i) **Coronal Segmentation with 3D AH-Net**

Fig. 10.9 Multi-view slices of the example test CT volume 1 of the LITS challenge

sponding layers of a 3D AH-Net, which is then fine-tuned using 3D patches with size $192 \times 192 \times 64$. The weights of other layers are randomly initialized. In the training of both networks, the cross-entropy loss is used at the beginning until convergence, which is then replaced by the Focal Loss for hard-voxel mining [13].

The performance of AH-Net is listed in Table 10.5, together with other six top-ranked submissions retrieved from the LITS challenge leaderboard. These submissions employ various types of neural network architectures: 2D, 3D, 2D–3D hybrid, and model fusion. Two evaluation metrics are adopted: (1) Dice Global (DG) which is the dice score combining all the volumes into one; (2) dice per case (DPC) which is the average of the dice scores of every single case. The dice score between two masks is defined as $DICE(A,B) = 2|A \cap B|/(|A| + |B|)$. Our results achieve the state-of-the-art performance in three of the four metrics, including the dice global score of the lesions, dice global, and dice per case score of the livers, which prove the effective-



(a) Axial Image Slice

(b) Axial Segmentation with MC-GCN

(c) **Axial Segmentation with 3D AH-Net**

(d) Sagittal Image Slice

(e) Sagittal Segmentation with MC-GCN

(f) **Sagittal Segmentation with 3D AH-Net**

(g) Coronal Image Slice

(h) Coronal Segmentation with MC-GCN

(i) **Coronal Segmentation with 3D AH-Net**

Fig. 10.10 Multi-view slices of the example test CT volume 2 of the LITS challenge

ness of AH-Net for segmenting 3D images with diverse anisotropic resolution. Some example visual results are shown in Fig. 10.8. In Figs. 10.9 and 10.10, we visually compare the results from MC-GCN and AH-Net on two different volumes acquired from the LITS challenge. AH-Net generated less false positive areas in the upper and the lower boundaries of both the lesion and liver.

10.5 Conclusion

In this chapter, we propose the 3D anisotropic hybrid network (3D AH-Net) which is capable of transferring the convolutional features of 2D images to 3D volumes with anisotropic resolution. By evaluating the proposed methods on both a large-scale in-house DBT dataset and a highly competitive open challenge dataset of CT segmentation, we show our network could obtain the state-of-the-art results. AH-Net generalizes better than the traditional 3D networks, such as 3D U-Net [3] due to the features transferred from a 2D network and the anisotropic convolution blocks. The GPU inference of AH-Net is also much faster than piling the results from a 2D network. Though AH-Net is designed for anisotropic volumes, we believe it could also be applied to volumes with resolution closed to being isotropic, such as CT and MRI.

Disclaimer: This feature is based on research, and is not commercially available. Due to regulatory reasons, its future availability cannot be guaranteed.

References

1. American Cancer Society (2017) Cancer facts and figures 2017. American Cancer Society
2. Brosch T, Tang LYW, Yoo Y, Li DKB, 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. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O (2016) 3D U-Net: learning dense volumetric segmentation from sparse annotation. arXiv e-prints arXiv:1606.06650
4. Chaurasia A, Culurciello E (2017) LinkNet: exploiting encoder representations for efficient semantic segmentation. arXiv e-prints arXiv:1707.03718
5. Chen J, Yang L, Zhang Y, Alber M, Chen DZ (2016) Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation. arXiv e-prints arXiv:1609.01006
6. Ghesu FC, Georgescu B, Grbic S, Maier AK, Hornegger J, Comaniciu D (2017) Robust multi-scale anatomical landmark detection in incomplete 3d-ct data. In: MICCAI
7. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv e-prints arXiv:1512.03385
8. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2261–2269
9. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv e-prints arXiv:1502.03167

10. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv e-prints arXiv:1412.6980
11. Lee K, Zung J, Li P, Jain V, Seung HS (2017) Superhuman accuracy on the SNEMI3D connectomics challenge. arXiv e-prints arXiv:1706.00120
12. Li X, Chen H, Qi X, Dou Q, Fu CW, Heng PA (2017) H-DenseUNet: hybrid densely connected unet for liver and tumor segmentation from CT volumes. arXiv e-prints arXiv:1709.07330
13. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. arXiv e-prints arXiv:1708.02002
14. Liu F, Zhou Z, Jang H, Samsonov A, Zhao G, Kijowski R (2017) Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging. Magn Reson Med 79(4):2379–2391
15. Liu S, Xu D, Zhou SK, Pauly O, Grbic S, Mertelmeier T, Wicklein J, Jerebko A, Cai W, Comaniciu D (2018) 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G (eds) Medical Image Computing and Computer Assisted Intervention - MICCAI 2018. Springer International Publishing, Cham, pp 851–858
16. Moeskops P, Viergever MA, Mendrik AM, de Vries LS, Benders MJNL, Isgum I (2016) Automatic segmentation of MR brain images with a convolutional neural network. IEEE Trans Med Imaging 35(5):1252–1261
17. Peng C, Zhang X, Yu G, Luo G, Sun J (2017) Large kernel matters improve semantic segmentation by global convolutional network. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 1743–1751
18. Qiu Z, Yao T, Mei T (2017) Learning spatio-temporal representation with pseudo-3d residual networks. In: 2017 IEEE international conference on computer vision (ICCV), pp 5534–5542
19. Ren S, He K, Girshick RB, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39:1137–1149
20. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: MICCAI
21. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
22. Wang G, Li W, Ourselin S, Vercauteren T (2017) Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In: BrainLes workshop at MICCAI 2017
23. Xia Y, Liu F, Yang D, Cai J, Yu L, Zhu Z, Xu D, Yuille A, Roth H (2018) 3D semi-supervised learning with uncertainty-aware multi-view co-training. arXiv e-prints arXiv:1811.12506
24. Zeng T, Wu B, Ji S (2017) DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. Bioinformatics 33(16):2555–2562
25. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6230–6239

Part III
Various Applications

Chapter 11
Deep Hashing and Its Application for Histopathology Image Analysis

Xiaoshuang Shi and Lin Yang

Abstract Content-based image retrieval (CBIR) has attracted considerable attention for histopathology image analysis because it can provide more clinical evidence to support the diagnosis. Hashing is an important tool in CBIR due to the significant gain in both computation and storage. Because of the tremendous success of deep learning, deep hashing simultaneously learning powerful feature representations and binary codes has achieved promising performance on microscopic images. This chapter presents several popular deep hashing techniques and their applications on histopathology images. It starts introducing the automated histopathology image analysis and explaining the reasons why deep hashing is a significant and urgent need for data analysis in histopathology images. Then, it specifically discusses three popular deep hashing techniques and mainly introduces pairwise-based deep hashing. Finally, it presents their applications on histopathology image analysis.

11.1 Introduction

Histopathology images play a significant role in early disease detection and grading, such as lung, breast, and brain cancers [1–4]. However, manual assessment is laborious, expensive, time-consuming, and error prone due to high-resolution of images and subjective assessment of pathologists. To reduce the workload of pathologists and improve the objectivity of image analysis, computer-aided diagnosis (CAD) systems including image processing and modern machine learning techniques have been widely applied to histopathology image computing. Generally, CAD systems can be roughly classified into two categories: classifier-based CAD and content-based image retrieval (CBIR) [5–7]. Compared to classifier-based CAD that directly

X. Shi · L. Yang (✉)
J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida,
Gainesville, FL, USA
e-mail: lin.yang@bme.ufl.edu

X. Shi
e-mail: xsshi2015@ufl.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_11

provides diagnosis results or grading scores, CBIR can not only be utilized to classify query images but also retrieve and visualize images with the most similar morphological profiles [8, 9]. Therefore, CBIR techniques attract considerable attention for histopathology image analysis [10–14].

Although traditional CBIR systems have exhibited their advantages on providing pathologists with diagnosis support in visualizing relevant images and diagnosis information, most of them are suitable for disease diagnosis with only tens or hundreds of images and fail to tackle large-scale data sets due to the computational efficiency and storage costs. However, the large number of annotated medical images might reduce the semantic gap (the difference between image and label (disease) description) between images and diagnosis information with modern data-driven methods [15]. To handle large-scale image data, hashing-based retrieval methods have become attractive [10, 13, 16] because hashing can encode the high-dimensional data into compact binary codes with maintaining the similarity among neighbors [17, 18], leading to significant gains in both computation and storage [19–21].

Based on whether employing semantic information, hashing methods can be classified into two groups: (i) unsupervised hashing that aims to explore the intrinsic structure of data to maintain the similarity among neighbors without any semantic information and (ii) supervised hashing that utilizes semantic information to produce binary codes. Due to the semantic gap, supervised hashing is more preferred in histopathology image analysis. Because hand-crafted features cannot optimally represent the image content and maintain the semantic similarity, and deep learning [22, 23] can automatically learn powerful features from large-scale raw image data, deep hashing [24–27], which utilizes deep learning architectures to simultaneously learn feature representations and binary codes of images, achieves better retrieval and classification accuracy than traditional hashing methods. In this chapter, we will focus on several popular supervised deep hashing techniques and their applications to histopathology image analysis.

11.2 Deep Hashing

Based on the usage of semantic information, supervised deep hashing methods can be roughly grouped into three categories: pointwise, multiwise, and pairwise.

11.2.1 Pointwise-Based Hashing

Pointwise-based hashing formulates the searching problem into a classification problem to learn binary codes. Their objective functions are usually developed on the basis of a regression model. One of the most popular objective functions is [20]

$$\begin{aligned} \min_{\mathbf{B},\mathbf{W},F} \textstyle\sum_{i=1}^{n} L(\mathbf{y}_i, \mathbf{W}\mathbf{b}_i) + \lambda \left\|\mathbf{W}\right\|_F^2 + \upsilon \textstyle\sum_{i=1}^{n} \left\|\mathbf{b}_i - F(\mathbf{x}_i)\right\|_2^2 , \\ s.t.\ \mathbf{b}_i \in \{-1, 1\}^m , \end{aligned} \tag{11.1}$$

where $\mathbf{b}_i$ is the binary vector of the sample $\mathbf{x}_i$, $\mathbf{y}_i \in \mathbb{R}^c$ is the one-hot vector maintaining the semantic information of $\mathbf{x}_i$, $\mathbf{W} \in \mathbb{R}^{c \times m}$ is one projection matrix, $L(\cdot)$ is a loss function (least squares, hinge loss, or others), and $F(\cdot)$ represents a nonlinear mapping function, c and m denote the number of classes and binary codes, respectively.

One early and popular pointwise-based deep hashing method is deep learning of binary hash codes (DLBHC) [24], which simultaneously learns image representation and binary codes in a pointwise manner. Because DLBHC utilizes a relaxation strategy, sigmoid function, to generate binary codes, it might generate accumulated quantization errors between relaxed and discrete binary matrices, thereby decreasing the retrieval performance. To address this issue, deep convolutional hashing (DCH) [26] proposes a novel loss function to jointly learn image features and discrete binary codes in a pointwise manner. It is [26]

$$\begin{aligned} \min_{\mathbf{W},\mathbf{c}} E = \textstyle\sum_{n=1}^{M_b} log \frac{e^{a_{y_n}^L}}{\sum_j e^{a_j^L}} + \frac{\gamma}{2} \textstyle\sum_{n=1}^{M_b} \left\|\mathbf{b}_n - \tilde{\mathbf{b}}_n\right\|_2^2 \\ s.t.\ \tilde{\mathbf{b}}_n = tanh(\mathbf{a}^{L-1}), \mathbf{b}_n = sgn(\tilde{\mathbf{b}}_n), \end{aligned} \tag{11.2}$$

where L is the number of layers in the network, M_b is the batch size, $\mathbf{W}$ and $\mathbf{c}$ stand for the parameters of the network, y_n represents the class membership of $\mathbf{x}_n$, $\mathbf{a}^{L-1}$ is the output of the L-1-th layer, a_j^L denotes the linear activation of the j-th neuron in the output layer, $\mathbf{b}_n \in \{-1, 1\}^m$ is a binary vector, and m is the number of bits. In Eq. (11.2), the first term is to maximize the classification accuracy and the regularization term is to reduce the accumulated error between relaxed and discrete binary vectors.

11.2.2 Multiwise-Based Hashing

Multiwise (ranking)-based hashing aims to learn hash functions to project originally high-dimensional data into a binary space and meanwhile maximizes the agreement of similarity orders over more than two items. Several popular algorithms are triplet ranking hashing (TRH) [28] that proposes a triplet ranking loss function based on the pairwise hinge loss, ranking supervision hashing (RSH) [29] that incorporates the ranking triplet information into a listwise matrix to learn binary codes, ranking preserving hashing (RPH) [30] that directly optimizes normalized discounted cumulative gain (NDCG) [31] to learn binary codes with high ranking accuracy. These algorithms cannot learn features and binary codes simultaneously, and later deep hashing approaches: network in network hashing (NINH) [25], bit-scalable deep hashing (DRSCH) [32], and triplet-based deep binary embedding (TDBE) [33] utilize the triplet ranking loss function to simultaneously learn feature representations and binary codes in order to preserve the similarity orders among originally high-dimensional data. For example, given training triplets of images in form of I, I^+, I^-, where I is more similar to I^+ than I^-, to find a mapping $\mathcal{F}(\cdot)$ to generate binary codes such that $\mathcal{F}(I)$ is more similar to $\mathcal{F}(I^+)$ than $\mathcal{F}(I^-)$, the triplet ranking hinge loss is defined as [25] follows:

$$\begin{gathered} \tilde{l}_{triplet}(\mathcal{F}(I), \mathcal{F}(I^+), \mathcal{F}(I^-)) \\ = max\left(0, 1 - \left(\left\|\mathcal{F}(I) - \mathcal{F}(I^-)\right\|_H - \left\|\mathcal{F}(I) - \mathcal{F}(I^+)\right\|_H\right)\right) \\ s.t.\ \mathcal{F}(I), \mathcal{F}(I^+), \mathcal{F}(I^-) \in \{0, 1\}^m, \end{gathered} \tag{11.3}$$

where $\|\cdot\|_H$ denotes Hamming distance.

Deep semantic ranking based hashing (DSRH) [34], adding an adaptive weight into a triplet hinge loss function, has been proposed to handle the multi-label retrieval problem. These deep hashing approaches adopt relaxation strategies to learn binary codes. To further improve the retrieval performance, discrete semantic ranking hashing (DSeRH) [35] directly learns binary codes with preserving similarity orders among samples.

11.2.3 Pairwise-Based Hashing

Pairwise-based hashing is to utilize the elementwise product of two binary vectors to preserve the Hamming affinity of data pairs. Among pointwise-, multiwise-, and pairwise-based deep hashing, pointwise-based hashing learns binary codes by formulating the searching into a classification problem, it usually neglects the similarity relationship among neighbors. Multiwise-based hashing is time-consuming to construct the triplet loss for large-scale training data, and it is usually difficult to converge due to its harder optimization problem than that of pointwise- and pairwise-based hashing. Therefore, pairwise-based hashing is more widely applied to pathology image analysis than pointwise- and multiwise-based hashing. In this subsection, we introduce several popular traditional and deep pairwise-based hashing methods.

11.2.3.1 Traditional Pairwise-Based Hashing

Several popular traditional pairwise-based hashing methods are spectral hashing (SH) [17] maps the original high-dimensional data into a low-dimensional Hamming space with preserving the similarity among neighbors for approximate nearest neighbors (ANN) search. Inspired by SH, many variants have been developed, such as multidimensional spectral hashing (MDSH) [36], elastic embedding (EE) [37], anchor graph hashing (AGH) [19], joint kernel graph hashing (JKGH) [12], etc. The minimal loss hashing (MLH) [38] utilizes a least-squares model to preserve the relations of similar pairs and a hinge loss to maintain the relationship of dissimilar pairs. Semi-supervised hashing (SSH) [16] leverages the Hamming distance between pairs and kernel-based supervised hashing (KSH) [19] extends it by using kernels to explore the nonlinear structure hidden in the data. Compared with linear algorithms, nonlinear pairwise-based hashing algorithms like binary reconstruction embedding (BRE) [39] and KSH often generate more effective binary codes because of the usage of the nonlinear structure hidden in the data. Additionally, many previous pairwise-based hashing methods usually relax non-differentiable discrete vectors into differential continuous ones and then learn binary codes by thresholding, thereby generating accumulated quantization errors between discrete and continuous matrices. To address this problem, many discrete hashing methods have been proposed, such as kernel-based supervised discrete hashing (KSDH) [18], column sampling based discrete supervised hashing (COSIDISH) [40], asymmetric discrete graph hashing (ADGH) [21], etc.

In the following, we briefly review four popular pairwise-based hashing algorithms: SH [17], BRE [39], KSH [19], and KSDH [18]. The basic SH formulation requires the codes in each hash bit to be balanced so that each hash bit has the maximum information, and different bits are mutually uncorrelated so that the redundancy among these bits is minimized. Formally, assuming $\mathbf{h}_i = [h_1(x), h_2(x), \ldots, h_m(x)]$, computed by m hashing functions $\{h(\cdot)\}_{i=1}^{m}$, represents the binary codes of the data point $\mathbf{x}_i$, the formulation is written as

$$\begin{aligned} &\min_{\mathbf{H}} Tr\left\{\mathbf{HLH}^T\right\}, \\ &s.t. \mathbf{H} \in \{-1, 1\}^{m \times n}, \mathbf{H1}_n = 0, \mathbf{HH}^T = n\mathbf{I}_m, \end{aligned} \tag{11.4}$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is a column vector with all elements being one and $\mathbf{I}_m \in \mathbb{R}^{m \times m}$ is an identity matrix. The constraint $\mathbf{H1}_n = 0$ aims to maximize the information of each hash bit and the constraint $\mathbf{HH}^T = n\mathbf{I}_m$ is to minimize the redundancy among different hash bits. $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is a Laplacian matrix, $\mathbf{D}$ is a diagonal matrix with the i-th element $d_{ii} = \sum_{j=1}^{n} s_{ij}$. $\mathbf{S} \in \mathbb{R}^{n \times n}$ is an affinity matrix to characterize the weight of any two data points, and it is defined as follows:

$$s_{ij} = \begin{cases} e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\tau^2}} & \{x_i, x_j\} \in \mathcal{M} \\ 0 & otherwise, \end{cases} \tag{11.5}$$

where τ is a regularization coefficient and $\mathcal{M}$ represents the the similar pairs (neighbors in terms of a metric distance or sharing the same label) set. It is difficult to solve the NP-hard problem in Eq. (11.4). Usually, spectral relaxation (or symmetric relaxation) is used to relax the discrete matrix $\mathbf{H}$ into a continuous matrix followed by thresholding to compute the final discrete matrix $\mathbf{H}$. However, the spectral relaxation usually generates accumulated quantization errors between the discrete matrix $\mathbf{H}$ and its relaxed continuous matrix, thereby significantly decreasing the retrieval accuracy, especially for large-scale training data [19, 20].

Similar to SH, the goal of BRE is also to project the original high-dimensional data into a low-dimensional Hamming space, taking advantage of fast nearest neighbor routines. Specifically, suppose the matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ represents n

original data points, and $\mathbf{x}_i$ $(1 \leq i \leq n)$ can be represented by a set of binary codes $[h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \ldots, h_m(\mathbf{x}_i)]$, where the k-th hashing function is defined as

$$h_k(\mathbf{x}) = sgn\left(\sum_{i=1}^{n_s} a_{ki}\kappa(\mathbf{x}_{ki}, \mathbf{x})\right), \tag{11.6}$$

where $\kappa(\cdot)$ is a kernel function, $\mathbf{x}_{ki} \in \mathbf{X}_k$ that represents the anchors selected from training data for learning the hashing function h_k, n_s is the number of anchors, and $a_{ki} \in \mathbf{A}$ that is a projection matrix. Note that in this paper we define that $sgn(\mathbf{x}) \in \{-1, 1\}$, and thus $h_k(\mathbf{x}) \in \{-1, 1\}$. Given any two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, BRE aims to minimize the difference between their Euclidean distance d_E and Hamming distance d_H. The formulation is

$$\min_{\mathbf{A}} \sum_{(i,j)\in\mathcal{N}} (d_E(\mathbf{x}_i, \mathbf{x}_j) - d_H(\mathbf{x}_i, \mathbf{x}_j))^2, \tag{11.7}$$

where $d_E(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2$, $d_H(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{4m}\sum_{k=1}^{m}\left\|h_k(\mathbf{x}_i) - h_k(\mathbf{x}_j)\right\|_2^2$, and $\mathcal{N}$ represents the pair set of training data. To remove the scale of data points, usually the original data point x is normalized to be unit vector so that $d_E \in [0, 1]$. Equation (11.7) can be easily extended to a supervised scenario with setting the distance of the same label pairs to be zero and different label pairs to be a large positive value. It is difficult to solve the optimization problem in Eq. (11.7) due to the non-differential $sgn(\cdot)$ function. Although the coordinate-descent algorithm [39] can solve the problem in Eq. (11.7) with preserving the discrete constraint, it usually consumes high training costs for large-scale training data.

KSH aims to look for m hashing functions to project the data $\mathbf{X} \in \mathbb{R}^{n\times d}$ into a Hamming space, in order to obtain the compact representation of each data point and preserve the similarity of pairs. For n points $\mathbf{X}$, the similar pairs (neighbors in terms of a distance or sharing the same label) are collected in the set $\mathcal{M}$ and the dissimilar pairs (non-neighbors or with different labels) are collected in the set $\mathcal{C}$. Let $\phi : \mathbb{R}^d \mapsto T$ be a kernel mapping from the original space to the kernel space, where T is a reproducing kernel Hilbert space (RKHS) with a kernel function $\kappa(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T\phi(\mathbf{y})$. With n_s points selected from $\mathbf{X}$ and a projection matrix $\mathbf{A} \in \mathbb{R}^{m\times n_s}$, the k-th $(1 \leq k \leq m)$ hashing function of KSH is defined as follows:

$$h_k(\mathbf{x}) = sgn\left(\sum_{j=1}^{n_s} \kappa(\mathbf{x}_j, \mathbf{x})a_{jk} - b_k) = sgn(\mathbf{a}_k\bar{\kappa}(\mathbf{x})\right), \tag{11.8}$$

where $\mathbf{x} \in \mathbf{X}$ and $b_k = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n_s}\kappa(\mathbf{x}_j, \mathbf{x}_i)a_{jk}$. Equation (11.8) implies a balanced hashing function constraint that is $\sum_{i=1}^{n} h_k(\mathbf{x}_i) = 0$. Based on Eq. (11.8), $h_k(\mathbf{x}) \in \{-1, 1\}$, KSH attempts to learn the projection matrix $\mathbf{A} \in \mathbb{R}^{m\times n_s}$ such that $h_k(\mathbf{x}_i) = h_k(\mathbf{x}_j)$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$, and $h_k(x_i) \neq h_k(x_j)$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$. Let the m-bit hash code of each point $\mathbf{x}$ be $code_m(\mathbf{x}) = [h_1, h_2, \ldots, h_m]$. Then, if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$,

$code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) = m$; otherwise, $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) = -m$, where $\circ$ represents the code inner product. In order to obtain the hashing function, the pairwise label matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ is defined as follows:

$$s_{ij} = \begin{cases} 1 & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ -1 & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ 0 & otherwise. \end{cases} \tag{11.9}$$

Because $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) \in [-m, m]$ and $s_{ij} \in [-1, 1]$, KSH learns the projection matrix by solving the following optimization model:

$$\min_{\mathbf{H} \in \{-1,1\}^{m \times n}} \left\| \mathbf{H}^T\mathbf{H} - r\mathbf{S} \right\|_F^2, \tag{11.10}$$

where $\mathbf{H} = sgn(\mathbf{A}\bar{\mathbf{K}}) = [code_m(x_1), \ldots, code_m(x_n)] \in \mathbb{R}^{m \times n}$ denotes the code matrix produced by hashing functions and $\bar{\mathbf{K}} \in \mathbb{R}^{n_s \times n}$ is a kernel matrix with zero mean. There is one implied condition: $\mathbf{H}\mathbf{1}_n = 0$ in Eq. (11.10). This condition maximizes the information of each bit.

To learn a discrete matrix and reduce the accumulated quantization error, KSDH utilizes an asymmetric relaxation to learn binary codes as follows:

$$\begin{gathered} \min_{\mathbf{A},\mathbf{H}} \left\| \mathbf{H}^T\mathbf{A}\bar{\mathbf{K}} - m\mathbf{S} \right\|_F^2, \\ s.t.\ \mathbf{A}\bar{\mathbf{K}}\bar{\mathbf{K}}^T\mathbf{A}^T = n\mathbf{I}_m,\ \ \mathbf{H} = sgn(\mathbf{A}\bar{\mathbf{K}}). \end{gathered} \tag{11.11}$$

Note that the hashing function $\mathbf{H}$ is preserved in the objective function. Usually, the smaller accumulated quantization error between $\mathbf{A}\bar{\mathbf{K}}$ and $sgn(\mathbf{A}\bar{\mathbf{K}})$, the smaller reconstruction error of the objective function. Ideally when $\mathbf{H}^T\mathbf{A}\bar{\mathbf{K}} = m\mathbf{S}$, it is easy to obtain $\mathbf{H} = \mathbf{A}\bar{\mathbf{K}}$. The constraint $\mathbf{A}\bar{\mathbf{K}}\bar{\mathbf{K}}^T\mathbf{A}^T = n\mathbf{I}_m$ is derived from the constraint $\mathbf{H}\mathbf{H}^T = n\mathbf{I}_m$, which enforces m bit hashing codes to be mutually uncorrelated such that the redundancy among these bits is minimized [17]. In addition, the constraint $\mathbf{A}\bar{\mathbf{K}}\bar{\mathbf{K}}^T\mathbf{A}^T = n\mathbf{I}_m$ can also reduce the redundancy among data points [41]. Since $Tr\left\{\bar{\mathbf{K}}^T\mathbf{A}^T\mathbf{H}\mathbf{H}^T\mathbf{A}\bar{\mathbf{K}}\right\}$ and $Tr\left\{\mathbf{W}^T\mathbf{W}\right\}$ are constants, the optimization problem in Eq. (11.11) is equivalent to the following optimization problem:

$$\begin{gathered} \max_{\mathbf{A},\mathbf{H}} Tr\left\{\mathbf{H}\mathbf{W}\bar{\mathbf{K}}^T\mathbf{A}^T\right\}, \\ s.t.\ \mathbf{A}\bar{\mathbf{K}}\bar{\mathbf{K}}^T\mathbf{A}^T = n\mathbf{I}_m, \mathbf{H} = sgn(\mathbf{A}\bar{\mathbf{K}}). \end{gathered} \tag{11.12}$$

11.2.3.2 Deep Pairwise-Based Hashing

Aforementioned traditional methods produce binary codes after obtaining features extracted by using GIST [42], HOG [43], SIFT [42], or convolutional neural network (CNN) [22], which might decrease their retrieval accuracy due to learning features and binary representations individually. To address this problem, deep hashing network (DHN) [44] and deep supervised pairwise hashing (DSPH) [45] simultaneously learn image representations and binary codes by maintaining the relationship of each pair. To take advantage of the pair labels, DSPH proposes an effective and efficiency model as follows:

$$\begin{gathered}\min_{\mathbf{B},\mathbf{W},\mathbf{V},\theta} J = -\sum_{s_{ij}\in S}(s_{ij}\Theta_{ij} - log(1+e^{\Theta_{ij}})) + \eta\sum_{i=1}^{n}\left\|\mathbf{b}_i - \mathbf{W}^T\phi(\mathbf{x}_i,\theta)\right\|_2^2, \\ s.t.\ \mathbf{u}_i = \mathbf{W}^T\phi(\mathbf{x}_i,\theta),\ \Theta_{ij} = \frac{1}{2}\mathbf{u}_i^T\mathbf{u}_j,\ \mathbf{b}_i \in \{-1,1\}^m,\end{gathered} \tag{11.13}$$

where $s_{ij} \in \{0, 1\}$ is the pairwise label to denote the similarity relationship between samples $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{B}$ represents binary codes, $\phi(\mathbf{x}_i, \theta)$ denotes the features extracted by CNN and θ stands for its parameters, and $\mathbf{W}$ is a projection matrix in the last layer of the deep hashing architecture.

Similar to DHN and DSPH, many other deep pairwise-based hashing methods have been developed to further improve their retrieval performance, like deep supervised hashing (DSH) [46] and HashNet [47]. Because these hashing methods focus on the interclass difference of images but ignore the relevance order of images within the same classes, pairwise-based deep ranking hashing (PDRH) [27] is proposed to simultaneously learn feature representations and binary codes by employing a convolutional neural network and a pairwise matrix to maintain the interclass difference and intraclass relevance among images. We introduce PDRH in the following.

Given data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n\times d}$, where n and d are the number of data points and dimensions, respectively. Suppose these data have c classes, with each class containing n_k ($\sum_{k=1}^{c} n_k = n$) data points. Let $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same class; otherwise, $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$, where $\mathcal{M}$ and $\mathcal{C}$ represent the neighbor-pair and non-neighbor-pair sets, respectively. Assume that a data point $\mathbf{x} \in \mathbb{R}^d$ belongs to the k-th class and it has different relevance to the data points in the same class, the relevance list can be written as follows:

$$r(\mathbf{x}, \mathbf{X}_k) = \left\{r_1^k, r_2^k, \ldots, r_{n_k}^k\right\}, \tag{11.14}$$

where $\mathbf{X}_k$ is a set containing all data points belonging to the k-th class, $r_j^k > 0$ represents the relevance of data point $\mathbf{x}_j^k$ to $\mathbf{x}$, and $r_j^k > r_l^k$ means that $\mathbf{x}$ is more similar to $\mathbf{x}_j^k$ than that to $\mathbf{x}_l^k$.

Hashing is to encode the high-dimensional data into a set of compact binary codes. Specifically, for a data point $\mathbf{x}$, its k-th hashing function is defined as follows:

$$h_k(\mathbf{x}) = sgn(f(\mathbf{x})\mathbf{a}_k + b_k), \tag{11.15}$$

where $sgn(\cdot)$ is a mapping function such that $sgn(f(\mathbf{x})\mathbf{a}_k + b_k) \in \{-1, 1\}$, $f(\mathbf{x}) \in \mathbb{R}^p$ is a row vector representing p features extracted from $\mathbf{x}$, $\mathbf{a}_k \in \mathbb{R}^p$ is a column vector to project the high-dimensional features into a low-dimensional space and b_k is a basis. In this chapter, we define $h_k(\mathbf{x}) = 1$ if $f(\mathbf{x})\mathbf{a}_k + b_k \geq 0$; otherwise, $h_k(\mathbf{x}) = -1$. Let m-bit hash codes of $\mathbf{x}$ be $code_m(\mathbf{x}) = [h_1, h_2, \ldots, h_m]$, and then it has $-m \leq code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) \leq m$. For three data points $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$, if $\mathbf{x}_i$ is more similar to $\mathbf{x}_j$ than $\mathbf{x}_k$, there exists $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) > code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_k)$. In order to distinguish data points belonging to different classes, let $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) > 0$ when $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ and $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) < 0$ when $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$. Considering the intraclass relevance of data points, a pairwise matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ to describe the relationship of data pairs is defined as follows:

$$s_{ij} = \begin{cases} r(\mathbf{x}_i, \mathbf{x}_j) & (x_i, x_j) \in \mathcal{M}, \\ -\gamma & (x_i, x_j) \in \mathcal{C}, \end{cases} \tag{11.16}$$

where $0 < r(\mathbf{x}_i, \mathbf{x}_j) \le m$ is the relevance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $0 < \gamma \le m$ is a constant. Although the largest γ can be m, empirically, we choose a relatively small γ in order to loose the constraint of hashing codes. In this paper, we set $\gamma = 1$.

Hashing aims to learn compact binary codes to preserve the relations among originally high-dimensional data. Because $code_m(\mathbf{x}) = [h_1, h_2, \ldots, h_m]$, $-m \le code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) \le m$, and $-\gamma \le s_{ij} \le r_{max}$, the objective function can be intuitively written as follows:

$$\min_{\mathbf{H}} \frac{1}{4} \left\| \frac{r_{max}}{m} \mathbf{H}\mathbf{H}^T - \mathbf{S} \right\|_F^2, \tag{11.17}$$

where r_{max} is the maximum element in $\mathbf{S}$, $\mathbf{H} = sgn(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b})$, $\mathbf{H} \in \{-1, 1\}^{n \times m}$, $\mathbf{A} \in \mathbb{R}^{p \times m}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{1}_n \in \mathbb{R}^n$ is a column vector with all elements being one.

Unfortunately, Eq. (11.17) is non-differential and thus it is difficult to directly solve. To learn the projection matrix $\mathbf{A}$, $\mathbf{H} = sgn(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}) \in \{-1, 1\}^{n \times m}$ is relaxed into $\mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}) \in [-1, 1]^{n \times m}$ based on the following observations: (1) $[-1, 1]^{n \times m}$ is the closest convex region to the non-convex region $\{-1, 1\}^{n \times m}$; (2) $\mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b})$ is differentiable with respect to $\mathbf{A}$ and $\mathbf{b}$, while $\mathbf{H} = sgn(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b})$ is non-differentiable due to the non-smooth function $sgn(\cdot)$. Then Eq. (11.17) can be reformulated as follows:

$$\begin{aligned} J_1 = \min_{\mathbf{A},\mathbf{b}} \tfrac{1}{4} \left\| \tfrac{r_{max}}{m} \mathbf{Y}\mathbf{Y}^T - \mathbf{S} \right\|_F^2, \\ s.t. \;\; \mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}), \end{aligned} \tag{11.18}$$

which is a pairwise loss function that preserves the semantic information into $\mathbf{Y}$. Because a large accumulated quantization error between $\mathbf{H}$ and the relaxed $\mathbf{Y}$ will decrease the retrieval accuracy, a quantization loss term $J_2 = \frac{\lambda_1}{2} \|\mathbf{H} - \mathbf{Y}\|_F^2$ is added into Eq. (11.18) to make the projection matrix $\mathbf{A}$ and the vector $\mathbf{b}$ reduce the accumulated error, where λ_1 is a weight coefficient. Furthermore, because the variance of the projection matrix $\mathbf{A}$ is important to obtain a robust and stable solution, a regularization loss term $J_3 = \frac{\lambda_2}{4} \left\| \mathbf{A}^T\mathbf{A} - \mathbf{I}_m \right\|_F^2$ is incorporated into Eq. (11.18), where $\mathbf{I}_m \in \mathbb{R}^{m \times m}$ is an identity matrix and λ_2 is a regularization coefficient. Therefore, Eq. (11.18) becomes

$$\begin{aligned} J = \min_{\mathbf{A},\mathbf{b}} \tfrac{1}{4} \left\| \tfrac{r_{max}}{m} \mathbf{Y}\mathbf{Y}^T - \mathbf{S} \right\|_F^2 + \tfrac{\lambda_1}{2} \|\mathbf{H} - \mathbf{Y}\|_F^2 + \tfrac{\lambda_2}{4} \left\| \mathbf{A}\mathbf{A}^T - \mathbf{I}_m \right\|_F^2, \\ s.t. \;\; \mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}), \mathbf{H} = sgn(\mathbf{Y}), \end{aligned} \tag{11.19}$$

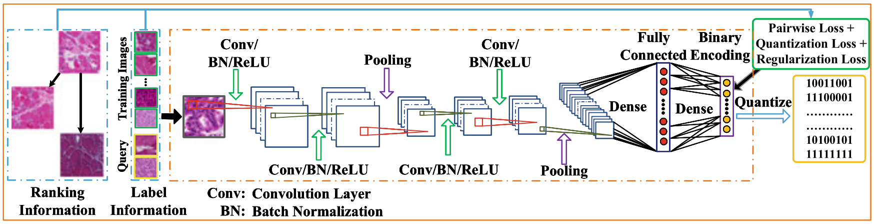


Fig. 11.1 The flowchart of PDRH. Binary encoding layer is used to encode the extracted features obtained from the fully connected layer into binary codes

which is the proposed objective function of PDRH. In Eq. (11.19), λ_1 is mainly used to balance the preserved semantic information in $\mathbf{Y}$ and the accumulated errors between $\mathbf{H}$ and $\mathbf{Y}$, i.e., the larger λ_1, the smaller accumulated errors yet the less preserved semantic information; λ_2 is to control the variance of $\mathbf{A}$, and a too large λ_2 might increase the accumulated error and decrease the preserved semantic information. For clarity, we present the flowchart of PDRH using one CNN architecture and Eq. (11.19) in Fig. (11.1).

Next, we will show the optimization procedure of the proposed objective function Eq. (11.19) embedded in a CNN architecture.

Suppose that the proposed network contains L layers with parameters $\left(\mathbf{A}^l, \mathbf{b}^l\right)_{l=1}^L$, where $\mathbf{A}^l$ denotes the weight connection between the $(l-1)$-th and l-th layers, and $\mathbf{b}^l$ represents the bias in the l-th layer. The previous $L-1$ layers are provided by a CNN architecture (please refer to Fig. 11.1). Equation (11.19) is the objective function of the L-th (binary encoding) layer after the $L-1$-th (fully connected) layer, and thus $\mathbf{A}^L = \mathbf{A}$ and $\mathbf{b}^L = \mathbf{b}$. The output of the l-th layer is

$$\mathbf{Z}^l = \sigma(\mathbf{Z}^{l-1}\mathbf{A}^l + \mathbf{1}_n\mathbf{b}^l), \tag{11.20}$$

where $\sigma(\cdot)$ represents the activation function. Hence, the parameters $f(\mathbf{X})$ and $\mathbf{Y}$ in Eq. (11.19) are equivalent to $\mathbf{Z}^{L-1}$ and $\mathbf{Z}^L$ in the network, respectively.

To calculate the gradients of the parameters $\left(\mathbf{A}^l, \mathbf{b}^l\right)_{l=1}^L$, we need to first calculate the partial derivatives $\frac{\partial J}{\partial \mathbf{A}^L}$ and $\frac{\partial J}{\partial \mathbf{b}^L}$ in the L-th layer as follows:

$$\frac{\partial J}{\partial \mathbf{A}^L} = \frac{\partial J_1}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{A}^L} + \frac{\partial J_2}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{A}^L} + \frac{\partial J_3}{\partial \mathbf{A}^L} = \mathbf{Z}^{L-1T}\Delta^L + \lambda_2\mathbf{A}^L(\mathbf{A}^{LT}\mathbf{A}^L - \mathbf{I}_m), \tag{11.21a}$$

$$\frac{\partial J}{\partial \mathbf{b}^L} = \frac{\partial J_1}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{b}^L} + \frac{\partial J_2}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{b}^L} = \frac{1}{n}\mathbf{1}_n\Delta^L, \tag{11.21b}$$

where $\Delta^L = (\frac{r_{max}}{m}(\frac{r_{max}}{m}\mathbf{Z}^L\mathbf{Z}^{LT} - \mathbf{S})\mathbf{Z}^L + \lambda_1(\mathbf{H} - \mathbf{Z}^L)) \odot (\mathbf{1}_n\mathbf{1}_m^T - \mathbf{Z}^L \odot \mathbf{Z}^L)$, and $\odot$ denotes elementwise multiplication.

The partial derivatives $\frac{\partial J}{\partial \mathbf{A}^l}$ and $\frac{\partial J}{\partial \mathbf{b}^l}$ in the l-th ($l < L$) layer are calculated as follows:

$$\frac{\partial J}{\partial \mathbf{A}^l} = \frac{\partial J}{\partial \mathbf{Z}^l}\frac{\partial \mathbf{Z}^l}{\partial \mathbf{A}^l} = \mathbf{Z}^{l-1T}\Delta^l, \tag{11.22a}$$

$$\frac{\partial J}{\partial \mathbf{b}^l} = \frac{\partial J}{\partial \mathbf{Z}^l}\frac{\partial \mathbf{Z}^l}{\partial \mathbf{b}^l} = \frac{1}{n}\mathbf{1}_n\Delta^l, \tag{11.22b}$$

where $\Delta^l = \Delta^{l+1}\mathbf{A}^{l+1T} \odot \sigma'(\mathbf{Z}^{l-1}\mathbf{A}^l + \mathbf{1}_n\mathbf{b}^l)$, and $\sigma'(\mathbf{Z}^{l-1}\mathbf{A}^l + \mathbf{1}_n\mathbf{b}^l)$ denotes the derivative of $\mathbf{Z}^l$.

The parameters $\left(\mathbf{A}^l, \mathbf{b}^l\right)_{l=1}^{L}$ are updated by using the gradient descent algorithm as follows:

$$\mathbf{A}^l = \mathbf{A}^l - \eta\frac{\partial J}{\partial \mathbf{A}^l}, \tag{11.23a}$$

$$\mathbf{b}^l = \mathbf{b}^l - \eta\frac{\partial J}{\partial \mathbf{b}^l}. \tag{11.23b}$$

11.3 Experimental Results and Discussion

Experiments on an image data set including histopathological skeletal muscle and lung cancer images are utilized to evaluate some popular hashing methods. All skeletal muscle and lung cancer images are stained with hematoxylin and eosin (H&E). The skeletal muscle images contain two major classes of idiopathic inflammatory myopathy (IIM), i.e., polymyositis (PM) and dermatomyositis (DM). The lung cancer images are with two types of diseases, i.e., adenocarcinoma (AC) and squamous cell carcinoma (SC). 5,256 (2,572 PM and 2,678 DM) skeletal muscle images corresponding to 41 individual subjects are collected and cropped from the Medical College of Wisconsin Neuromuscular Laboratory (MCWNL), and 2,904 (1,456 AC and 1,448 SC) lung cancer images of 42 patients are selected and cropped from The Cancer Genome Altas (TCGA). Here, all images are randomly partitioned into training and testing sets with a (approximate) ratio 3:1. Specifically, 6,128 images including 3,952 (2,010 PM and 1,942 DM) skeletal muscle images and 2,176 (1,092 AC and 1,084 SC) lung cancer images are utilized for training, and the remaining 2,032 (562 PM, 736 DM, 364 AC, and 364 SC) images are used for testing. In all experiments, the RGB raw images are directly used as input for all deep hashing methods, and they are wrapped to patches with a size of 128 × 128 before inputting to the learning pipeline. Moreover, for images in the same class, the Euclidean distance between any two images is calculated, and then all images are divided into eight subsets based on the distance. The relevance of the images in the subset with the smallest distance is eight and that with the largest distance is one. Note that in practice the relevance can be defined based on the applications.

Table 11.1 Configuration of the network for input images with the size 128 × 128 × 3. (Note that the batch normalization followed by the ReLU layer is in the middle of two convolution layers or the convolution and max-pooling layers.)

Type	Filter size/stride	Output size
Convolution	$3 \times 3 \times 3 \times 32/1$	$126 \times 126 \times 32$
Convolution	$3 \times 3 \times 32 \times 32/1$	$124 \times 124 \times 32$
Pool	$2 \times 2/2$	$62 \times 62 \times 32$
Convolution	$3 \times 3 \times 32 \times 64/1$	$60 \times 60 \times 64$
Convolution	$3 \times 3 \times 64 \times 64/1$	$58 \times 58 \times 64$
Pool	$2 \times 2/2$	$29 \times 29 \times 64$
FC	–	512
Binary	–	m

We show PDRH and eight non-deep hashing methods including six non-ranking hashing algorithms, spectral hashing (SH) [17], KSH [19], COSIDISH [40], SDH [20], KSDH [18], and ADGH [21], as well as two ranking hashing algorithms RSH [29] and RPH [30], and two popular deep hashing algorithms, convolutional neural network hashing (CNNH) [48] and deep learning of binary hash codes (DLBHC) [24]. For non-deep hashing algorithms, the holistic high-dimensional features are extracted from the whole image as the input, i.e., first detecting scale-invariant feature transform (SIFT) key points from the whole image and then employing SIFT to extract features around these key points. Then, these features are encoded into 2,000-dimensional histograms using the bag-of-words (BoW) method [6]. For comparison, the output of the fully connected layer of PDRH is also used as the input for the non-deep hashing methods, and these deep learning features are named as DLF. In PDRH, its three essential parameters are set as $\gamma = 1$, $\lambda_1 = r_{max}$ and $\lambda_2 = 0.1$. For fairness, all deep hashing methods utilize the architecture in Table 11.1. To evaluate the performance of hashing algorithms, we employ five main criterion: classification accuracy, MAP, PR-curve, REL, and NDCG. Given a set of queries, its MAP, REL, and NDCG scores are calculated as follows:

MAP is the mean of the average precision (AP) for each query. AP is defined as follows:

$$AP@q = \frac{\sum_{k=1}^{q} P(k)\delta(k)}{\sum_{k=1}^{q} \delta(k)}, \tag{11.24}$$

where q is the number of top returned samples, $P(k)$ is the precision at cut-off k in the list, $\delta(k) = 1$ if the sample ranked at k-th position is relevant; otherwise, $\delta(k) = 0$.

$$REL = \frac{1}{q}\sum_{i=1}^{q} r_i, \tag{11.25}$$

where r_i is the relevance of the ith nearest neighbors to the query image.

$$NDCG = \frac{1}{Z} \sum_{i=1}^{q} \frac{2^{r_i} - 1}{log(i+1)}, \tag{11.26}$$

where Z is a constant to normalize the maximum of NDCG to be one.

11.3.1 Experimental Results

To calculate the retrieval accuracy of various hashing algorithms, we first adopt each algorithm to encode training and query images into binary codes, and then utilize the binary codes to calculate the Hamming distance between query and training images. Next, we can select the nearest training images for each query image based on their Hamming distance. Finally, we can compute the retrieval accuracy of each query image based on the corrected selected training images. We calculate the average value of all query images based on the five criterion and report them in Tables 11.2, 11.3, and Figs. 11.2, 11.3, 11.4, 11.5.

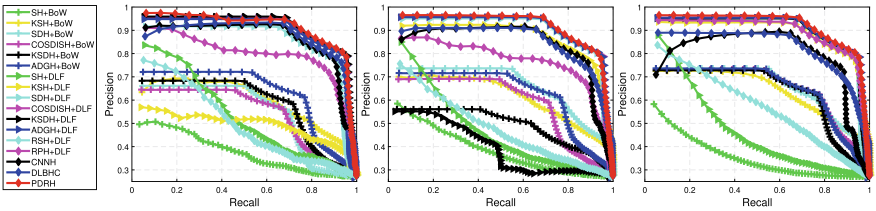
Table 11.2 shows the average retrieval accuracy of all query images, including classification accuracy and MAP of various methods, e.g., the non-deep algorithms SH, KSH, SDH, COSDISH, KSDH, ADGH, RSH, RPH, and deep hashing algorithms CNNH, DLBHC and PDRH, on 8-, 16-, and 32-bit hashing codes. As we can see, the non-deep hashing algorithms with DLF achieve significantly better performance than that with BoW. Additionally, COSIDISH, KSDH, and ADGH with DLF outperform the deep hashing algorithms CNNH and DLBHC. PDRH obtains higher accuracy (97.49%) and MAP (97.49 and 97.33%) than COSIDISH, KSDH, and ADGH with DLF at 8 bit, and they have similar performance (the difference is within 0.5%) at 16 and 32 bit.

Figure 11.2 displays the PR-curve of various algorithms at 8, 16, and 32 bit. It suggests that with DLF features, the non-deep algorithms, SH, KSH, SDH, COSDISH, KSDH, and ADGH outperform those with BoW features. At 8 bit, PDRH achieves better performance than the other algorithms when the recall is smaller than 0.3. At 16 and 32 bit, PDRH can obtain the best retrieval performance among all hashing algorithms. Figure 11.3 presents the precision of various algorithms at all 8, 16, and 32 bits using Hamming radius $r = 1, 2$ and 3. When $r = 1$ or 2, PDRH attains the best precision at 8 and 16 bit. When $r = 3$, PDRH outperforms other algorithms at 16 and 32 bit.

Table 11.3 presents the ranking performance (REL and NDCG) of the proposed method PDRH and the comparative algorithms with DLF features at 16 bit on 5, 10, and 50 returned neighbors. PDRH obtains the highest REL score 5.74, 5.74, and 5.76 on 5, 10, and 50 returned neighbors, respectively. Figure 11.4 shows the relevance of above algorithms on 5, 10, 20, 50, and 100 retrieved images at all the three bits. It further illustrates that PDRH achieves higher REL scores than the other algorithms on different returned neighbors or bits. Table 11.3 also demonstrates that with 16-bit hashing codes, PDRH, SH+DLF, and RSH+DLF obtain the best NDCG score than

Table 11.2 Retrieval performance (%) measured as classification accuracy and MAP with the top 100 and 500 returned neighbors, respectively

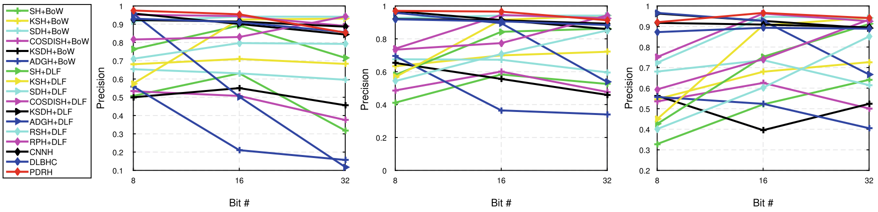
Method	Accuracy			MAP (Top 100)			MAP (Top 500)		
	8	16	32	8	16	32	8	16	32
SH [17]+BoW	49.31	63.48	69.73	54.86	65.16	65.38	55.22	57.95	56.16
KSH [19]+BoW	64.22	75.30	77.66	64.19	72.33	73.55	71.09	71.56	73.32
SDH [20]+BoW	66.14	73.67	73.67	66.14	73.67	73.67	66.14	73.67	73.67
COSDISH [40]+BoW	64.42	69.29	73.08	68.22	70.63	76.26	68.68	71.30	76.44
KSDH [18]+BoW	68.36	66.10	72.83	68.36	66.10	72.83	68.36	66.10	72.83
ADGH [21]+BoW	72.15	71.56	73.57	72.15	71.56	73.57	72.15	71.56	73.57
SH [17]+DLF	85.73	90.26	92.57	87.25	88.86	91.42	83.98	80.08	82.56
KSH [19]+DLF	57.48	92.86	93.65	57.02	91.90	94.40	56.96	93.22	94.16
SDH [20]+DLF	91.68	95.08	94.98	91.68	95.08	94.98	91.68	95.08	94.98
COSDISH [40]+DLF	96.56	**96.70**	**96.41**	96.69	**96.70**	**96.49**	96.70	**96.70**	**96.50**
KSDH [18]+DLF	96.90	96.15	96.11	96.90	96.15	96.11	96.90	96.15	96.11
ADGH [21]+DLF	96.56	**96.70**	**96.41**	96.69	**96.70**	**96.49**	96.70	**96.70**	**96.70**
RSH [29]+DLF	80.27	80.36	88.63	83.16	81.38	87.65	78.88	74.02	81.33
RPH [30]+DLF	92.22	85.29	94.24	91.56	88.01	95.06	92.39	88.55	94.61
CNNH [48]	92.86	92.32	66.19	88.48	82.57	67.14	92.81	90.01	85.03
DLBHC [24]	94.49	92.13	90.80	82.92	88.30	89.44	90.86	91.50	89.27
PDRH	**97.49**	**96.75**	**96.65**	**97.49**	**96.80**	**96.66**	**97.33**	**96.65**	**96.52**



@8-bit @16-bit @32-bit

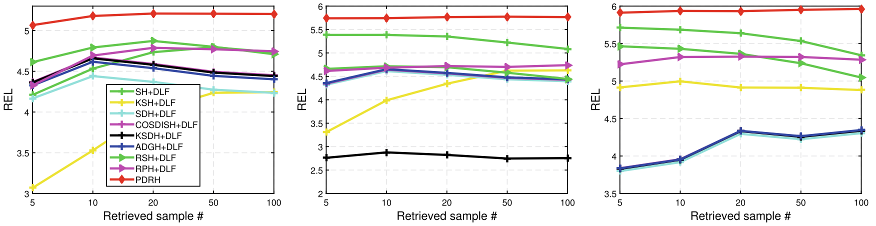
Fig. 11.2 PR-curve of various algorithms at different number of bits

the others when five images are retrieved. With 10 and 50 images returned, PDRH achieves the highest NDCG score 0.49 and 0.51, respectively. Figure 11.5 shows the NDCG score of various algorithms on 5, 10, 20, 50, and 100 retrieved images. At 8 bit, PDRH achieves slightly worse score than RSH+DLF, while it significantly outperforms the others. At 16 bit, PDRH and RSH+DLF achieve similar NDCG scores on 5, 10, and 20 retrieved images, and PDRH obtains the best score on 50 and 100 returned neighbors. At 32 bit, PDRH outperforms the others on 10, 20, 50, and 100 returned samples.



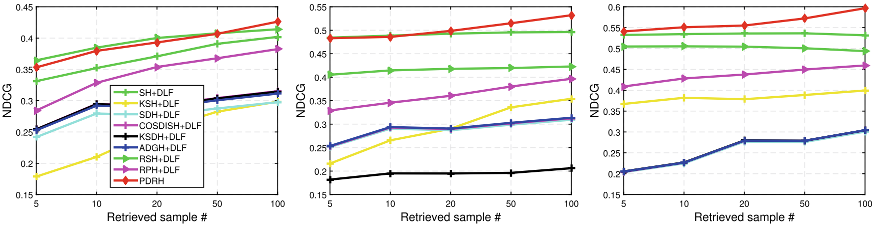
@r=1 @r=2 @r=3

Fig. 11.3 Precision versus bit using various algorithms on different Hamming radiuses (@r)



@8-bit @16-bit @32-bit

Fig. 11.4 REL with different number of retrieved images



@8-bit @16-bit @32-bit

Fig. 11.5 NDCG with different number of retrieved images

11.3.2 Discussion

Based on experimental results in Tables 11.2, 11.3, and Figs. 11.2, 11.3, 11.4, 11.5, we have the following observations:

- Non-deep hashing algorithms SH, KSH, SDH, COSDISH, KSDH, and ADGH with DLF obtain better retrieval performance including classification accuracy, MAP, and PR-curve than those with BoW features. PDRH containing a convolutional neural network has powerful ability to extract features from the original histopathology images with preserving the significant semantic information. By contrast, BoW explores the intrinsic structure to extract features with maintaining the significant information without any supervision, and thus it might neglect some significant semantic information.

Table 11.3 Ranking performance using REL and NDCG with 16-bit hashing codes on 5, 10, and 50 retrieved images

Method	REL			NDCG		
	5	10	50	5	10	50
SH [17]+DLF	5.38	5.38	5.22	**0.48**	0.48	0.49
KSH [19]+DLF	3.31	3.99	4.62	0.22	0.27	0.34
SDH [20]+DLF	4.32	4.61	4.44	0.25	0.29	0.30
COSDISH [40]+DLF	4.34	4.64	4.47	0.25	0.29	0.30
KSDH [18]+DLF	3.76	3.87	3.82	0.21	0.22	0.28
ADGH [21]+DLF	4.36	4.65	4.57	0.25	0.29	0.29
RSH [29]+DLF	5.38	5.39	5.22	**0.48**	0.48	0.49
RPH [30]+DLF	4.62	4.68	4.70	0.33	0.35	0.38
PDRH	**5.74**	**5.74**	**5.76**	**0.48**	**0.49**	**0.51**

- Although the non-deep hashing algorithms including the non-ranking and ranking hashing can perform well with DLF features, they usually achieve similar or inferior performance to PDRH. The main possible reason is that PDRH extracts features and learns binary representations simultaneously, leading to better and more stable solutions.
- The non-ranking hashing including KSH, SDH, COSDISH, KSDH, and ADGH can deliver fair retrieval performance, while their ranking performance, including REL and NDCG, is relatively poor. This is because their affinity matrices maintain the interclass difference of images, while they do not consider the similarity order of images within the same classes.
- Although the ranking hashing algorithms, RSH and RPH, can obtain better ranking performance than most of non-ranking hashing, they usually achieve worse retrieval performance. This can be attributed to the fact that RSH and RPH focus on the intraclass difference among images, but they do not emphasize the interclass difference among images.

11.4 Summary

This chapter presents three types of deep hashing techniques, especially pairwise-based deep hashing, and their applications on histopathology images. Specifically, it discusses pointwise-, multiwise-, and pairwise-based deep hashing and analyzes their differences. Additionally, it introduces one pairwise-based deep hashing method that can simultaneously extract features from histopathology images and learn their binary representations, with preserving the interclass difference for image classification and maintaining the intraclass relevance order in the same classes. Finally, this chapter shows experimental results of several popular hashing methods on a histopathology image data set including skeletal muscle and lung cancer images.

Currently, there are still some challenging problems required to address (i) designing a more general deep hashing model to handle more types of histopathology images; (ii) proposing fast and efficient deep hashing methods to retrieve the most relevant image patches to one type of diseases from whole-slide images, which usually contain billions pixels and provide an obstacle for many hashing methods; (iii) leveraging a small amount of labeled data and large-scale unlabeled data to achieve deserved retrieval accuracy, because most of current deep hashing methods require a large amount of labeled images, labeling which is laborious, extensive, and time-consuming for pathologists; (iv) designing a robust deep hashing model to tackle histopathology images with noisy labels, because it is error prone to annotate images due to the subjective assessment of pathologists.

References

1. Petushi S, Garcia FU, Haber MM, Katsinis C, Tozeren A (2006) Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. Bio Med Comput Med Imag 6(1):1
2. Yang L, Chen W, Meer P, Salaru G, Feldman MD, Foran DJ (2007) High throughput analysis of breast cancer specimens on the grid. In: Proceedings of the international conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 617–625
3. Xing F, Yang L (2016) Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. IEEE Rev Biomed Eng 9:234–263
4. Xing F, Xie Y, Yang L (2016) An automatic learning-based framework for robust nucleus segmentation. IEEE Trans Med Imag 35(2):550–566
5. Comaniciu D, Meer P, Foran DJ (1999) Image-guided decision support system for pathology. Mach Vis Appl 11(4):213–224
6. Caicedo JC, Cruz A, Gonzalez FA (2009) Histopathology image classification using bag of features and kernel functions. In: Proceedings of the conference on artificial intelligence in medicine in Europe. Springer, Berlin, pp 126–135
7. Basavanhally AN, Ganesan S, Agner S, Monaco JP, Feldman MD, Tomaszewski JE, Bhanot G, Madabhushi A (2010) Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology. IEEE Trans Biomed Eng 57(3):642–653
8. Zheng L, Wetzel AW, Gilbertson J, Becich MJ (2003) Design and analysis of a content-based pathology image retrieval system. IEEE Trans Inf Tech Biomed 7(4):249–255
9. Akakin HC, Gurcan MN (2012) Content-based microscopic image retrieval system for multi-image queries. IEEE Trans Inf Tech Biomed 16(4):758–769
10. Zhang X, Liu W, Dundar M, Badve S, Zhang S (2015) Towards large-scale histopathological image analysis: hashing-based image retrieval. IEEE Trans Med Imag 34(2):496–506
11. Zhang X, Su H, Yang L, Zhang S (2015) Fine-grained histopathological image analysis via robust segmentation and large-scale retrieval. In: Proceedings of the international conference on computer vision and pattern recognition, pp 5361–5368
12. Jiang M, Zhang S, Huang J, Yang L, Metaxas DN (2015) Joint kernel-based supervised hashing for scalable histopathological image analysis. In: Medical image computing and computer-assisted intervention, pp 366–373
13. Shi X, Xing F, Xu K, Xie Y, Su H, Yang L (2017) Supervised graph hashing for histopathology image retrieval and classification. Med Image Anal

14. Shi X, Xing F, Xie Y, Su H, Yang L (2017) Cell encoding for histopathology image classification. In: Medical image computing and computer-assisted intervention
15. Zhang S, Metaxas D (2016) Large-scale medical image analytics: recent methodologies, applications and future directions. Med Imag Anal 33:98–101
16. Wang J, Kumar S, Chang S (2012) Semi-supervised hashing for large-scale search. IEEE Trans Pattern Anal Mach Intell 34(12):2393–2406
17. Weiss Y, Torralba A, Fergus R (2009) Spectral hashing. In: Proceedings of the advances in neural information processing systems, pp 1753–1760
18. Shi X, Xing F, Cai J, Zhang Z, Xie Y, Yang L (2016) Kernel-based supervised discrete hashing for image retrieval. In: Proceedings of the European conference on computer vision, pp 419–433
19. Liu W, Wang J, Ji R, Jiang Y, Chang S (2012) Supervised hashing with kernels. In: Proceedings of the international conference on computer vision and pattern recognition, pp 2074–2081
20. Shen F, Shen C, Liu W, Tao Shen H (2015) Supervised discrete hashing. In: Proceedings of the international conference on computer vision and pattern recognition, pp 37–45
21. Shi X, Xing F, Xu K, Sapkota M, Yang L (2017) Asymmetric discrete graph hashing. In: Association for the advancement artificial intelligence
22. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
23. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings advances in neural information processing systems, pp 1097–1105
24. Lin K, Yang HF, Hsiao JH, Chen CS (2015) Deep learning of binary hash codes for fast image retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 27–35
25. Lai H, Pan Y, Liu Y, Yan S (2015) Simultaneous feature learning and hash coding with deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3270–3278
26. Sapkota M, Shi X, Xing F, Yang L (2018) Deep convolutional hashing for low dimensional binary embedding of histopathological images. IEEE J Biomed Health Inform
27. Shi X, Sapkota M, Xing F, Liu F, Cui L, Yang L (2018) Pairwise based deep ranking hashing for histopathology image classification and retrieval. Pattern Recog 81:14–22
28. Norouzi M, Fleet DJ, Salakhutdinov RR (2012) Hamming distance metric learning. In: Proceedings neural information processing systems, pp 1061–1069
29. Wang J, Liu W, Sun AX, Jiang YG (2013) Learning hash codes with listwise supervision. In: International conference on computer vision, pp 3032–3039
30. Wang Q, Zhang Z, Si L (2015) Ranking preserving hashing for fast similarity search. In: International joint conference on artificial intelligence, pp 3911–3917
31. Järvelin K, Kekäläinen J (2000) IR evaluation methods for retrieving highly relevant documents. In: Proceedings of the international ACM SIGIR conference on research and development in information retrieval. ACM, pp 41–48
32. Zhang R, Lin L, Zhang R, Zuo W, Zhang L (2015) Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. IEEE Trans Image Proc 24(12):4766–4779
33. Zhuang B, Lin G, Shen C, Reid I (2016) Fast training of triplet-based deep binary embedding networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5955–5964
34. Zhao F, Huang Y, Wang L, Tan T (2015) Deep semantic ranking based hashing for multi-label image retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1556–1564
35. Liu L, Shao L, Shen F, Yu M (2017) Discretely coding semantic rank orders for supervised image hashing. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1425–1434
36. Weiss Y, Fergus R, Torralba A (2012) Multidimensional spectral hashing. In: European conference on computer vision, pp 340–353

37. Carreira-Perpinán MA (2010) The elastic embedding algorithm for dimensionality reduction. In: International conference on machine learning, vol 10, pp 167–174
38. Norouzi M, Blei DM (2011) Minimal loss hashing for compact binary codes. In: International conference on machine learning, pp 353–360
39. Kulis B, Darrell T (2009) Learning to hash with binary reconstructive embeddings. In: Proceedings of the advances in neural information processing systems, pp 1042–1050
40. Kang WC, Li WJ, Zhou ZH (2016) Column sampling based discrete supervised hashing. In: Association for the advancement of artificial intelligence
41. Shi X, Guo Z, Nie F, Yang L, You J, Tao D (2016) Two-dimensional whitening reconstruction for enhancing robustness of principal component analysis. IEEE Trans Pattern Anal Mach Intell 38(10):2130–2136
42. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175
43. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the ieee international conference on computer vision and pattern recognition, pp 886–893
44. Zhu H, Long M, Wang J, Cao Y (2016) Deep hashing network for efficient similarity retrieval. In: Association for the advancement of artificial intelligence, pp 2415–2421
45. Li WJ, Wang S, Kang WC (2015) Feature learning based deep supervised hashing with pairwise labels. arXiv:1511.03855
46. Liu H, Wang R, Shan S, Chen X (2016) Deep supervised hashing for fast image retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2064–2072
47. Cao Z, Long M, Wang J, Yu PS (2017) Hashnet: deep learning to hash by continuation. In: Proceedings of the IEEE international conference on computer vision, pp 5609–5618
48. Xia R, Pan Y, Lai H, Liu C, Yan S (2014) Supervised hashing for image retrieval via image representation learning. In: Association for the advancement of artificial intelligence, vol 1, p 2

Chapter 12
Tumor Growth Prediction Using Convolutional Networks

Ling Zhang, Lu Le, Ronald M. Summers, Electron Kebebew and Jianhua Yao

Abstract Prognostic tumor growth modeling via volumetric medical imaging observations is a challenging yet important problem in precision and predictive medicine. It can potentially imply and lead to better outcomes of tumor treatment management and surgical planning. Traditionally, this problem is tackled through mathematical modeling. Recent advances of convolutional neural networks (ConvNets) have demonstrated higher accuracy and efficiency than conventional mathematical models can be achieved in predicting tumor growth. This indicates that deep learning based data-driven techniques may have great potentials on addressing such problem. In this chapter, we first introduce a statistical group learning approach to predict the pattern of tumor growth that incorporates both the population trend and personalized data, where deep ConvNet is used to model the voxel-wise spatiotemporal tumor progression. We then present a two-stream ConvNets which directly model and learn the two fundamental processes of tumor growth, i.e., cell invasion and mass effect, and

L. Zhang (✉)
Nvidia Corporation, Bethesda, MD 20814, USA
e-mail: lingz@nvidia.com

L. Le
PAII Inc., Bethesda Research Lab, 6720B Rockledge Drive, Ste 410, Bethesda, MD 20817, USA
e-mail: le.lu@paii-labs.com; lelu@cs.jhu.edu

Johns Hopkins University, Baltimore, MD, USA

R. M. Summers
Imaging Biomarkers and Computer-Aided Diagnosis Laboratory and Clinical Image Processing Service Radiology and Imaging Sciences Department, National Institutes of Health Clinical Center, Bethesda, MD 20892, USA

E. Kebebew
Endocrine Oncology Branch National Cancer Institute, National Institutes of Health, Bethesda, MD 20892, USA

J. Yao
Tencent Holdings Limited, Shenzhen 518057, China
e-mail: jianhuayao@tencent.com

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_12

predict the subsequent involvement regions of a tumor. Experiments on a longitudinal pancreatic tumor data set show that both approaches substantially outperform a state-of-the-art mathematical model-based approach in both accuracy and efficiency.

12.1 Introduction

The prediction of tumor growth is a very challenging task. It has long been viewed as a mathematical modeling problem [6, 11, 31]. Medical imaging data provide noninvasive and in vivo measurements of the tumor over time at a macroscopic level. For this reason, previous works on image-based tumor growth modeling are mainly based on the reaction–diffusion equations and on biomechanical models. Some previous tumor growth models [6, 11, 31] are derived from two or more longitudinal imaging studies of a specific patient over time. While these methods yield informative results, most previous tumor growth models are independently estimated from the target patient without considering the tumor growth pattern of population trend. Furthermore, the small number of model parameters (e.g., 5 in [31]) may be insufficient to represent the complex characteristics of tumor growth.

Aside from mathematical modeling methods, the combination of data-driven principles and statistical group learning may provide a potential solution to solve these problems by building a model based on both population trend and personalized clinical characteristics. The only pioneer study in this direction [20] attempts to model the glioma growth patterns in a classification-based framework. This model learns tumor growth patterns from selected features at the patient, tumor, and voxel levels, and achieves a prediction accuracy of 59.8%. However, this study only uses population trend of tumor growth without incorporating the history of the patient-specific tumor growth pattern, and is unable to predict tumor growth at different time points. Furthermore, this early study only employs hand-crafted low-level features. In fact, information describing tumor progression may potentially lie in the latent high-level feature space of tumor imaging, but this has yet to be investigated.

Deep neural networks [16] are high capacity trainable models with a large set of ($\sim$15 M) parameters. By optimizing the massive amount of network parameters using gradient backpropagation, the network can discover and represent intricate structures from raw data without any type of feature engineering. In particular, deep convolutional neural networks (ConvNets) have significantly improved performance in a variety of traditional medical imaging applications [9]. The basic idea of these applications is using deep learning to determine the current status of a pixel or an image (whether it belongs to object boundary/region, or certain category). The ConvNets have also been successfully used in prediction of future binary labels at image/patient level, such as survival prediction of patients with brain and lung cancer [22, 32]. More generally, in artificial intelligence community, ConvNet has shown its strong ability to predict the next status at image pixel level—as a key component in AlphaGo [19, 23], fully ConvNets are trained to predict the next move (position

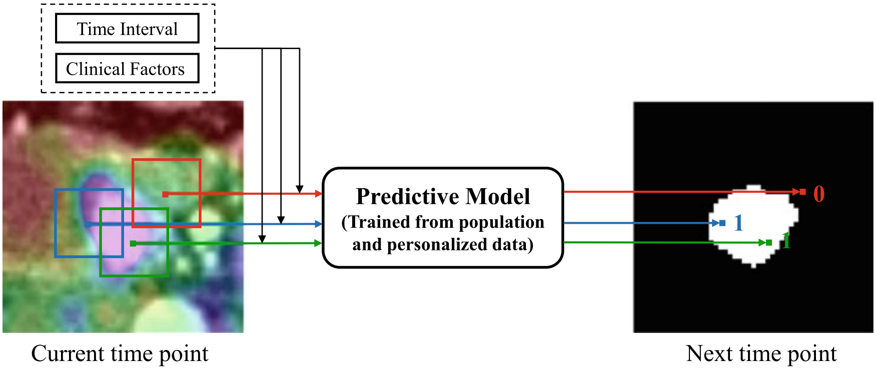


Fig. 12.1 Framework of the voxel-wise prediction of tumor growth using statistical group learning

of the 19 × 19 Go game board) of Go player, given the current board status, with an accuracy of 57%.

In this chapter, we are investigating whether deep ConvNets are capable of predicting the future status at the pixel/voxel level for medical problem. Our main objective is to design a deep learning predictive model to predict whether the voxels in the current time point will become tumor voxels or not at the next time point (cf. Fig. 12.1).

We first present a statistical group learning framework to predict tumor growth that incorporates tumor growth patterns derived from population trends and personalized clinical factors. Our hypothesis is that regions involved in future tumor progression are predictable by combining visual interpretations of the longitudinal multimodal imaging information with those from clinical factors. We then present a two-stream ConvNets which directly represent and learn the two fundamental processes of tumor growth (cell invasion and mass effect) from multi-model tumor imaging data at multiple time points. Quantitative experiments on a pancreatic tumor data set demonstrate that both methods substantially outperform a state-of-the-art model-based method [31] in both accuracy and efficiency.

12.2 Group Learning Approach for Tumor Growth Prediction

In the longitudinal pancreatic tumor data studied in this work, each patient has multimodal imaging data (dual-phase contrast-enhanced CT and FDG-PET [(2-[18F] Fluoro-2-deoxyglucose positron emission tomography)]) and clinical records at three time points spanning 3–4 years. We design an integrated training and personalization and prediction framework illustrated in Fig. 12.2. The imaging data scans of different modalities acquired at different time points are first registered, after which the

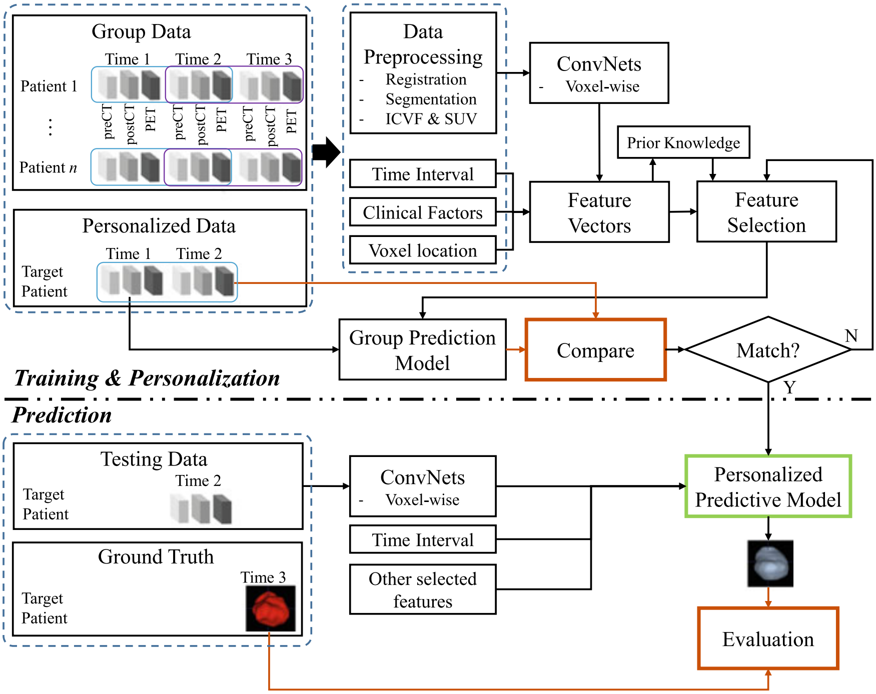


Fig. 12.2 Overview of the proposed learning method for predicting tumor growth. The upper part represents stages of model training (to learn population trend) and personalization and the lower part formulates the process of (unseen) data prediction

tumors are segmented. Intracellular volume fraction (ICVF) and standardized uptake value (SUV) [31] are also computed. In the training and personalization stage, all voxel-wise ConvNets- and location-based features, time intervals, and clinical factors are extracted from any pairs of two time points (time1/time2 and time2/time3) from group data (patient 1—patient n) and the pair of time1/time2 from personalized data (the target patient, denoted as patient $n + 1$). Next, feature selection, which takes prior knowledge into account, is used to rank these features from hybrid resources. The top m-ranked features ($m = 1, \ldots, M$) are employed to train SVM models on group data (to capture population trend). These SVM classifiers are then personalized via the time1/time2 pair of the target patient data to determine the optimal feature set and model parameters (personalization). In the prediction stage, given the data of the target patient at time2, the imaging and clinical features are fed into the predictive model to predict and estimate the voxel-wise tumor region at a future time3. Note that the testing data (i.e., for predicting time3 based on time2 of the target patient) has never been seen by the predictive model.

12.2.1 Image Processing and Patch Extraction

To establish the spatial–temporal relationship of tumor growth along different time points, the multi-model imaging data are registered based on mutual information and imaging data at different time points are aligned using the tumor center [31]. After that, three types of information (SUV, ICVF, and tumor mask, refer to the left panel in Fig. 12.3 as an example) related to tumor property are extracted from the multimodal images and used as a three-channel input to the invasion ConvNet model.

(1) The FDG-PET characterizes regions in the body which are more active and need more energy to maintain existing tumor cells and to create new tumor cells. This motivates us to use FDG-PET to measure metabolic rate and incorporate it in learning the tumor predictive model. SUV is a quantitative measurement of the metabolic rate [18]. To adapt to the ConvNets model, the SUV values from PET images are magnified by 100 followed by a cutting window [100 2600] and then transformed linearly to [0 255].

(2) Tumor grade is one of the most important prognosticators, and is determined by the proliferation rate of the neoplastic cells [2]. This motivates us to extract the underlying physiological parameter related to the cell number. ICVF is a representation of the normalized tumor cell density, and is computed from the registered dual-phase contrast-enhanced CT:

$$\text{ICVF} = 1 - \frac{HU_{post\_tumor} - HU_{pre\_tumor}}{E[HU_{post\_blood} - HU_{pre\_blood}]} \times (1 - Hct), \qquad (12.1)$$

where HU_{post_tumor}, HU_{pre_tumor}, HU_{post_blood}, and HU_{pre_blood} are the Hounsfield units of the post- and pre-contrast CT images at the segmented tumor and blood pool

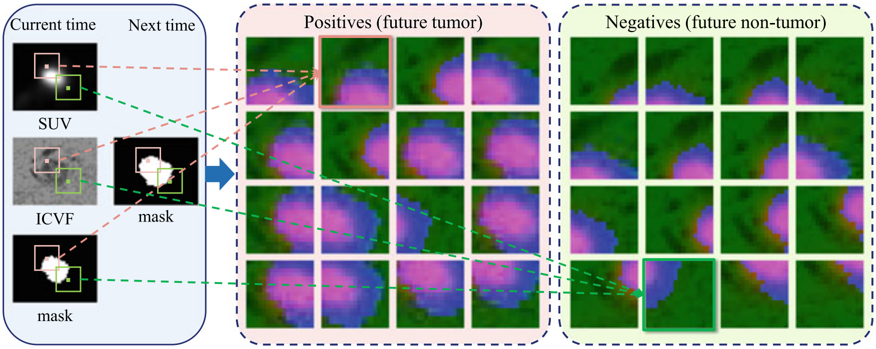


Fig. 12.3 Some examples of positive (center panel) and negative (right panel) training samples. In the left panel, the pink and green bounding boxes at the current time illustrate the cropping of a positive sample and a negative sample from multimodal imaging data. Each sample is a three-channel RGB image formed by the cropped SUV, ICVF, and mask at the current time. The label of each sample is determined by the location of corresponding bounding box center at the next time—inside tumor (pink): positive; outside tumor (green): negative

(aorta), respectively. $E[\bullet]$ represents the mean value. Hct is the hematocrit which can be obtained from blood samples, thus the ICVF of the tumor is computed using the ICVF of blood (Hct) as a reference. The resulting ICVF values are magnified by 100 (range between [0 100]) for ConvNets input.

(3) Tumor stage is another important prognosticator, and is determined by the size and extent of the tumor [2]. Previous studies have used the tumor mask/boundary to monitor the tumor morphological change and estimate model parameters [6, 12, 25]. In this study, following [31], the tumors are segmented by a semiautomatic level set algorithm with region competition [34] on the post-contrast CT image to form tumor masks with binary values (0 or 255).

As illustrated in Fig. 12.1, to train a ConvNet to distinguish between future tumor and future non-tumor voxels, image patches of size 17×17 voxels $\times 3$—centered at voxels near the tumor region at the current time point—are sampled from four channels of representations reflecting and modeling the tumor's physiological status. Patches centered inside or outside of tumor regions at the next time point are labeled as "1" and "0", serving as positive and negative training samples, respectively. This patch-based extraction method allows for embedding the context information surrounding the tumor voxel. The voxel (patch center) sampling range is restricted to a bounding box of ± 15 pixels centered at the tumor center, as the pancreatic tumors in our data set are <3 cm (≈ 30 pixels) in diameter and are slow growing. To avoid the classification bias toward the majority class (non-tumor) and to improve the accuracy and convergence rate during ConvNet training [15, 35], we create a roughly balanced training set by proportionally under-sampling the non-tumor patches. A few examples of positive and negative patches of SUV, ICVF, and mask encoded in three-channel RGB color images are shown in Fig. 12.3.

12.2.2 Learning a Voxel-Wise Deep Representation

We use AlexNet [15] as our network architecture. AlexNet contains five convolutional ($conv1$–$conv5$), three pooling ($pool1$, $pool2$, $pool5$), and two fully connected layers ($fc6$–$fc7$). This network is trained from scratch on all pairs of time points (time1/time2 and time2/time3) from the group data set. The training is terminated after a predetermined number of epochs, where the model with the lowest validation loss is selected as the final network.

The resulting ConvNet is then used to extract the high-level representation of voxels/patches. This is achieved by feeding the three-channel SUV-ICVF-mask image patches into the personalized ConvNet model, where the fc and the output layers can be treated as the learned deep features. Considering that the high-dimensional deep image features of the ConvNet fc layers may tend to overwhelm the low number tumor-level and patient-level features if combined directly, the outputs of the last layer with two nodes are regarded as the final extracted deep features.

12.2.3 Learning a Predictive Model with Multi-source Features

12.2.3.1 Feature Extraction and Selection

A general statistical learning concept is that cues from different sources can provide complementary information for learning a stronger classifier. Therefore, in addition to deep features, we extract three other types of features: (1) Time intervals between two imaging time points, with days as the time unit. (2) Tumor-level features—the Euclidean distance of the patch center toward its closest tumor surface within the 3D volume for each voxel. This distance value is positive if the patch center locates inside the current tumor region and negative otherwise. In addition, the tumor volume is calculated. (3) Patient-level features, including age, gender, height, and weight. The SVM RFE technique [10] is adopted to find the most informative features during the process of model training and personalization. Reflecting the significance of image-based features for assessing the growth of tumor [31], the two ConvNet-based features are found to be always selected by the SVM RFE model selection. Finally, time interval is used as a prior feature, as it is necessary for our task.

12.2.3.2 Predictive Model Training and Personalization, and Testing

Once the feature set has been fully ranked, the first m features ($m = [2, 3, \dots, 9]$) are each iteratively added to train a set of (seven) SVM classifiers until all features are included. In each iteration, the SVM classifier is trained on samples from the group data set, and then personalized on the samples of the personalization data set. The prediction accuracies are calculated and recorded for all classifiers, where the accuracy metric (ACC) is defined by $ACC = \frac{TP+TN}{TP+FP+FN+TN}$. The feature set and classifier that maximize the prediction ACC are selected.

To better personalize the predictive model from population trend to the target patient, we optimize an objective function which measures the agreement between the predicted tumor volume and its future ground-truth volume on the target patient. To do so, we first apply the predictive model to voxels in the searching neighborhood (tumor growth zone) of the personalization volume, and later threshold the classification outputs. The relative volume difference (RVD) between the predicted and ground-truth tumor volumes are computed. As in [31], the tumor growth zone is set as a bounding box surrounding the tumor, parametrized with the pixel distances N_x, N_y, and N_z to the tumor surface in the x-, y-, and z-directions, respectively.

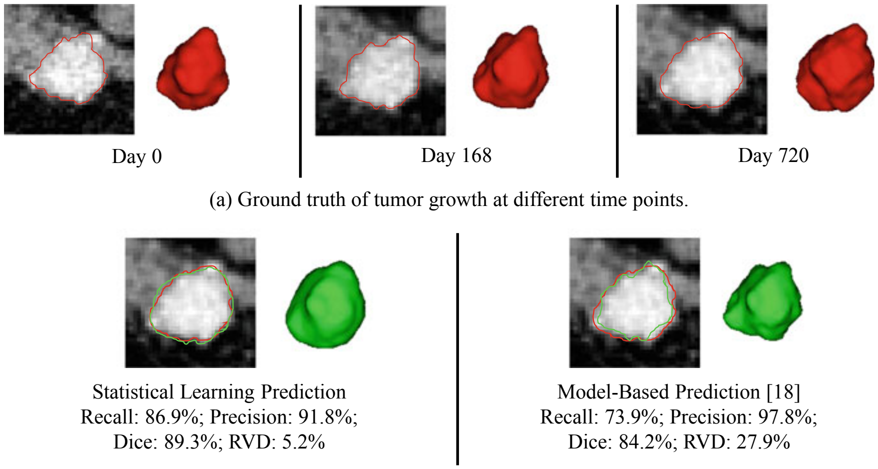
In the testing stage, given the data at time 2 of the target patient, the predictive model, along with its personalized model parameters, is applied to predict the label of every voxel in the growth zone at time 3.

12.2.4 Experiments and Results

Seven patients (five males and two female) with von Hippel–Lindau (VHL) disease, each with a pancreatic neuroendocrine tumor (PanNET), are studied in this section. The VHL-associated PanNETs are commonly found to be nonfunctioning with malignant (cancer) potential [14], and can often be recognized as well-demarcated and solid masses through imaging screens [29]. For the natural history of this kind of tumor, around 60% patients demonstrate nonlinear tumor growth, 20% stable, and 20% decreasing (over a median follow-up duration of 4 years) [27]. Treatments of PanNETs include active surveillance, surgical intervention, and medical treatment. Active surveillance is undertaken if a PanNET does not reach 3 cm in diameter or a tumor-doubling time <500 days; otherwise, the PanNET should be resected due to high risk of metastatic disease [14]. Medical treatment (e.g., everolimus) is for the intermediate-grade (PanNETs with radiologic documents of progression within the previous 12 months), advanced or metastatic disease [33]. Therefore, patient-specific prediction of spatial–temporal progression of PanNETs at earlier stage is desirable, as it will assist in making decision within different treatment strategies to better manage the treatment or surgical planning. In this data set, the average age, height, and weight of the patients at time 1 were 48.6 ± 13.9 years, 1.70 ± 0.13 m, and 88.1 ± 16.7 kg, respectively. The time interval between two time points is 418 ± 142 days (mean ± std.). This data set is obtained from [31].

The ConvNet is trained over 30 epochs. The initial learning rate is 0.001, and is decreased by a factor of 10 at every tenth epoch. Weight decay and momentum are set to 0.0005 and 0.9. A dropout ratio of 0.5 is used to regularize the *fc*6 and *fc*7 layers. Mini-batch size is 256. The image patch size s is set as 17 pixels due to the small size of the pancreatic tumors. To accommodate the Caffe framework used for our ConvNet, the original 17×17 image patches are upsampled to 256×256 patches via bilinear interpolation. A total of 36,520 positive and 41,999 negative image patches are extracted from seven patients. AlexNet is run on the Caffe platform [13], using a NVIDIA GeForce GTX TITAN Z GPU with 12 GB of memory. The SVM (LIBSVM library [4]) with linear kernel ($C = 1$) is used for both SVM RFE feature selection and SVM classifier training. The parameters for the tumor growth zone are set as $N_x = 3$, $N_y = 3$, and $N_z = 3$ for prediction speed concern, and we note that the prediction accuracy is not sensitive to variation of these parameters.

We evaluate the proposed method using a leave-one-out cross-validation, which not only facilitates comparison with the state-of-the-art model-based method [31] (tumor status at time1 and time2 already known, predict time3) but also more importantly enables learning both population trend and patient-specific tumor growth patterns. In each of the seven validations, six patients are used as the group training data to learn the population trend, the time1/time2 of the remaining patient is used as the personalization data set, and time2/time3 of the remaining patient as the testing set. We obtain the model's final performance values by averaging results from the seven cross-validation folds. The prediction performance is evaluated using measurements



(a) Ground truth of tumor growth at different time points.

Statistical Learning Prediction
Recall: 86.9%; Precision: 91.8%;
Dice: 89.3%; RVD: 5.2%

Model-Based Prediction [18]
Recall: 73.9%; Precision: 97.8%;
Dice: 84.2%; RVD: 27.9%

(b) Prediction at the third time point (Day 720).

Fig. 12.4 Comparison of the proposed learning-based tumor growth prediction to a state-of-the-art model-based prediction [31]. **a** Segmented (ground truth) tumor contours and volumes at different time points. **b** Prediction results at the third time point obtained by learning- and model-based techniques (red: ground-truth boundaries; green: predicted tumor boundaries)

Table 12.1 Performance comparison of our method with previous methods on testing set. Results are reported as mean ± std. [min, max]

	Recall (%)	Precision (%)	Dice (%)	RVD (%)
Ref. [31]	83.2 ± 8.8 [69.4, 91.1]	**86.9** ± 8.3 [74.0, **97.8**]	84.4 ± 4.0 [79.5, **92.0**]	13.9 ± 9.8 [3.6, 25.2]
Ours	**87.9** ± **5.0** [**81.4**, **94.4**]	86.0 ± **5.8** [**78.7**, 94.5]	**86.8** ± **3.6** [**81.8**, 91.3]	**7.9** ± **5.4** [**2.5**, **19.3**]

at the third time point by four metrics: recall, precision, Dice coefficient, and RVD (as defined in [31]).

$$\text{recall} = \frac{TPV}{V_{gt}}, \quad \text{precision} = \frac{TPV}{V_{pred}}, \quad \text{Dice} = \frac{2 \times TPV}{V_{gt} + V_{pred}}, \tag{12.2}$$

where TPV (true positive volume) is the overlapping volume between the predicted V_{pred} and the ground-truth tumor volume V_{gt}.

In the example shown in Fig. 12.4, our method achieves both a higher Dice coefficient and a lower RVD than the model-based method. Note that the perfect values for Dice and RVD are 100% and 0%, respectively. As indicated in Table 12.1, our method yields a higher Dice coefficient (86.8 ± 3.6% vs. 84.4 ± 4.0%), and especially a much lower RVD (7.9 ± 5.4% vs. 13.9 ± 9.8%) than the model-based method [31], and thus is far more effective in future tumor volume prediction. The model-based approach in [31] requires ~24 h for model personalization and ~21 s for simulation per patient, while our method merely requires 3.5 h for training and personalization and 4.8 ± 2.8 min for prediction per patient.

12.3 Convolutional Invasion and Expansion Networks for Tumor Growth Prediction

Cancer cells originate from the irreversible injuring of respiration of normal cells. Part of the injured cells could succeed in replacing the lost respiration energy by fermentation energy, but will therefore convert into undifferentiated and widely growing cells (cancer cells) [26]. Tumors develop from such abnormal cell/tissue growth, which is associated with cell invasion and mass effect [8]. Cell invasion is characterized by the migration and penetration of cohesive groups of tumor cells to surrounding tissues, and mass effect by the distension and outward pushing of tissues induced by tumor growth (Fig. 12.5).

We propose to use ConvNets to directly represent and learn the two fundamental processes of tumor growth (cell invasion and mass effect) from multi-model tumor imaging data at multiple time points. Our proposed ConvNet architectures are partially inspired by the mixture of policy and value networks for evaluating the next move/position in game of Go [23], as well as the integration of spatial and temporal networks for effectively recognizing action in videos [7, 24]. In addition to x- and y-direction optical flow magnitudes (i.e., two-channel image input) used in [7, 24], we add the flow orientation information to form a three-channel input, as the optical flow orientation is crucial to tumor growth estimation. In addition, we apply a personalization training step to our networks which is necessary and important to patient-specific tumor growth modeling [5, 18, 30, 31]. Furthermore, we focus on predicting future labels of tumor mask/segmentation, which is found to be substantially better than directly predicting and then segmenting future raw images [21].

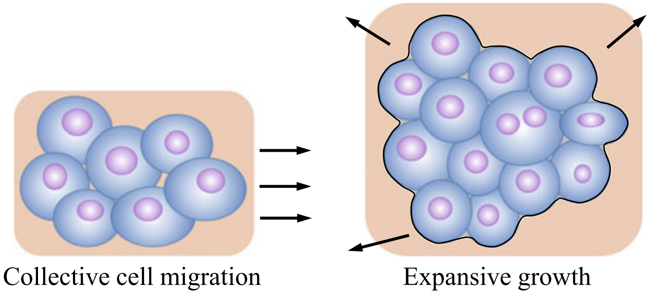


Fig. 12.5 The two fundamental processes of tumor growth: cell invasion and expansive growth of tumor cells

12.3.1 *Learning Invasion Network*

12.3.1.1 Image Processing and Patch Extraction

As detailed in Sect. 12.2.1, three types of information (SUV, ICVF, and tumor mask, refer to the left panel in Fig. 12.3 as an example) related to tumor property are extracted from the multimodal images and used as a three-channel input to the invasion ConvNet model.

12.3.1.2 Network Architecture

We use a six-layer ConvNet adapted from AlexNet [15], which includes four convolutional (*conv*) layers and one fully connected (*fc*) layer (cf. upper panel in Fig. 12.6). The inputs are of size 17 × 17 × 3 image patch stacks, where three refers to the tumor status channels of SUV, ICVF, and tumor mask. All *conv* layer filters are of size 3 × 3, with padding and stride of 1. The number of filters from *conv*1 to *conv*4 layers are 64, 128, 256, and 512, respectively. Max-pooling is performed over 3 × 3 spatial windows with stride 2 for *conv*1 and *conv*4 layers. Local response normalization is used for *conv*1 and *conv*2 layers using the same setting as [15]. The *fc*5 layer contains 256 rectifier units and applies "dropout" to reduce overfitting. All layers are equipped with the ReLU (rectified linear unit) activation function. The output layer is composed of two neurons corresponding to the classes future tumor or non-tumor, and applies a softmax loss function. The invasion ConvNet is trained on image patch–label pairs from scratch on all pairs of time points (time1/time2 and time2/time3) from the population data set.

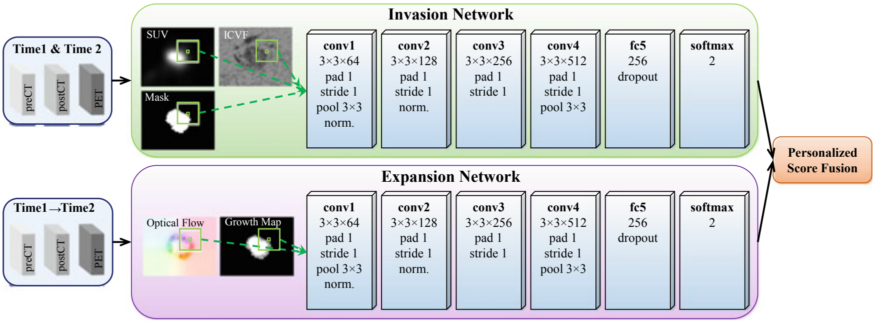


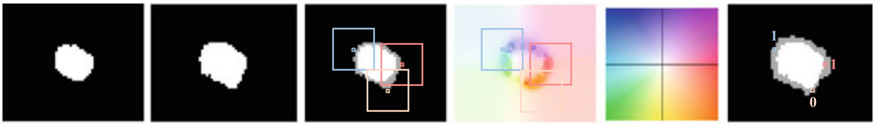
Fig. 12.6 ConvNet architecture for late fusion of the invasion and expansion networks for predicting tumor growth

12.3.2 Learning Expansion Network

12.3.2.1 Image Processing and Patch Extraction

Unlike the invasion network, which performs predictions from static images, the expansion network accounts for image motion information. Its input images, of size 17 × 17 × 4, capture expansion motion information between two time points. Three channels derive from a color-coded three-channel optical flow image, and the fourth from a tumor growth map between time1 and time2. Such images explicitly describe the past growing trend of tumor mass, as an image-based approximation of the underlying biomechanical force exerted by the growing tumor. These patches are sampled using the same restriction and balancing schemes applied for the invasion network (Sect. 12.3.1.1).

More specifically, for a pair of consecutive tumor mask images at time1 and time2 (Fig. 12.7a, b), we use the algorithm in [3] for optical flow estimation. The computed dense optical flow maps are a set of spatially coordinated displacement vector fields, which capture the displacement movements for all matched pairs of voxels from time1 to time2. By utilizing the color encoding scheme for flow visualization in [1, 17], the magnitude and orientation of the vector field can be formed as a three-channel color image (Fig. 12.7d). As depicted in the color coding map (Fig. 12.7e), the magnitude and orientation are represented by saturation and hue, respectively. This is a redundant but expressive visualization for explicitly capturing the motion dynamics of all corresponding voxels at different time points. Such a representation is also naturally fit for a ConvNet. The optical flow maps computed between raw CT image pairs may be noisy due to the inconsistent image appearance of tumors and surrounding tissues across two time points. Therefore, a binary tumor mask pair is used to estimate the optical flow as it provides the growing trend of tumor mass. It



(a) Tumor mask at time 1

(b) Tumor mask at time 2

(c) Tumor growth map (time1 → time2)

(d) 3-channel optical flow (time1 → time2)

(e) Flow field color coding

(f) Tumor growth map (time2 → time3)

Fig. 12.7 An example of color-coded optical flow image (**d**) generated based on the tumor mask pair at time 1 (**a**) and time 2 (**b**). The flow field color coding map is shown in (**e**), where hue indicates orientation and saturation indicates magnitude. In the tumor growth maps (**c**) and (**f**), white indicates the previous tumor region and gray indicates the newly grown tumor region. In **c** and **d**, three non-tumor voxels and their surrounding image patches are highlighted by three colors, which indicate the colors of these voxels in (**d**). The blue and red voxels indicate left and right growing trend and both become tumors at time 3 (**f**), while the pink voxel indicates very small motion and is still non-tumor at time 3 (**f**). Also note that although some voxels show tiny motion (e.g., lower left location) between time1 and time2, they grow faster from time2 to time3, indicating the nonlinear growth pattern of tumors

should be mentioned that both the expansion and shrink motion can be coded in the three-channel image.

However, such a representation of tumor growth motion has a potential limitation—both the voxels locate around the tumor center and at background have very small motion, which may confuse the ConvNet. Therefore, we additionally provide the past (time1 and time2) locations of tumor by adding a tumor growth map (Fig. 12.7c) as the 4th input channel. Specifically, voxels belong to the overlap region of time1 and time2, newly growing (expansion) region, shrink region, and background are assigned values of 255, 170, 85, and 0, respectively. This strategy implicitly indicates the probabilities of voxels to be tumor or not in the future.

12.3.2.2 Network Architecture

The expansion subnetwork has the same architecture as its invasion counterpart (cf. Sect. 12.3.1.2 and lower panel in Fig. 12.6), and is trained to learn from our motion-based representations and infer the future involvement regions of the tumor. This network is trained from scratch on different time point configurations ((time1→time2)/time3) of the population data set. In [28], optical flow is used to predict the future tumor position in a scan, and the future motion of a voxel is directly predicted by a linear combination of its past motions, which may be oversimplified. Our main difference is that the prediction is based on the nonlinear ConvNet learning of 2D motion and tumor growth maps where boundary/morphological information in a local region surrounding each voxel is maintained.

12.3.3 Fusing Invasion and Expansion Networks

To take advantage of the invasion–expansion information, we study a number of ways of fusing the invasion and expansion networks. Different fusion strategies result in significant different number of parameters in the networks.

12.3.3.1 Two-Stream Late Fusion

The two-stream architecture treats the appearance and motion cues separately and makes the prediction, respectively. The fusion is achieved by averaging decision/softmax scores of two subnetworks, as shown in Fig. 12.6. This method is denoted as *late fusion*. The invasion and expansion subnetworks are trained on all time–point pairs (time1/time2 and time2/time3) and triplets ((time1→time2)/time3) of the population data, respectively. Since they are trained independently, late fusion is not able to learn the voxel-wise correspondences between invasion and expansion features, i.e., registering appearance and motion cues. For example, what are the cell density and energy when a local voxel exhibits fast-growing trend? Late fusion doubles the number of network parameters compared to invasion or expansion subnetworks only.

12.3.3.2 One-Stream Early Fusion

In contrast to late fusion, we present an *early fusion* architecture, which directly stacking the three-channel invasion and four-channel expansion images as a seven-channel input to the ConvNet. The same network architecture as invasion/expansion network is used. Different from late fusion, early fusion can only be trained on time2/time3 pairs (without time1/time2 pairs) along with triplets ((time1→time2)/time3) of the population data. Therefore, less training samples can be used. Early fusion is able to establish voxel-wise correspondences. However, it leaves the correspondence to be defined by subsequent layers through learning. As a result, information in the motion image may not be able to be well captured by the network since there is more variability in the appearance images (i.e., SUV and ICVF). Early fusion keeps almost the same number of parameters as a single invasion or expansion network.

12.3.3.3 Two-Stream End-to-End Fusion

To jointly learn the nonlinear static and dynamic tumor information while allocating enough network capacity to both appearance and motion cues, we introduce a two-stream *end-to-end* fusion architecture. As shown in Fig. 12.8, the two subnetworks are connected by a fusion layer that adds a convolution on top of their *conv*4 layers. More specifically, the fusion layer first concatenates the two feature maps generated by *conv*4 (after ReLU4) and convolves the stacked data with 1 × 1 × 512 convolution filters with padding and stride of 1, then ReLU5 is attached and max-pooling 3 × 3 is performed. The outputs of the fusion layer are fed into a subsequent fully connected layer (*fc*5). As such, the fusion layer is able to learn correspondences of two compact

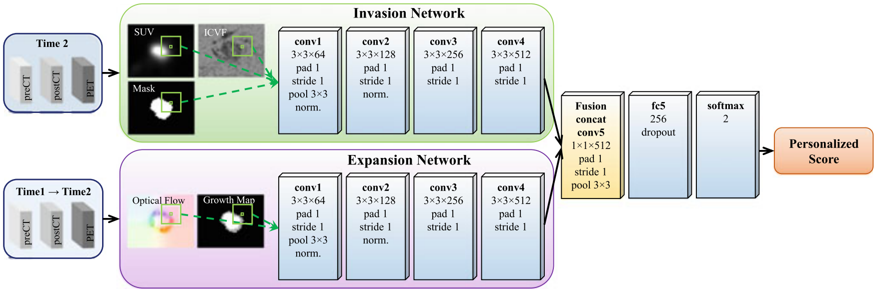

Invasion Network
Time 2
preCT
postCT
PET
SUV
ICVF
Mask
conv1 3×3×64 pad 1 stride 1 pool 3×3 norm.
conv2 3×3×128 pad 1 stride 1 norm.
conv3 3×3×256 pad 1 stride 1
conv4 3×3×512 pad 1 stride 1
Expansion Network
Time1 → Time2
preCT
postCT
PET
Optical Flow
Growth Map
conv1 3×3×64 pad 1 stride 1 pool 3×3 norm.
conv2 3×3×128 pad 1 stride 1 norm.
conv3 3×3×256 pad 1 stride 1
conv4 3×3×512 pad 1 stride 1
Fusion concat conv5 1×1×512 pad 1 stride 1 pool 3×3
fc5 256 dropout
softmax 2
Personalized Score

Fig. 12.8 Two-stream end-to-end fusion of the invasion and expansion networks for predicting tumor growth. The (convolution) fusion is after the *conv*4 (ReLU4) layer

feature maps that minimize a joint loss function. Fusion at ReLU4 instead of *fc* layer is because the spatial correspondences between invasion and expansion are already collapsed at the *fc* layer; fusion at the last *conv* layer has been demonstrated to have higher accuracy in compared to at earlier *conv* layers [7]. End-to-end framework is trained on the same time pairs and triplets as early fusion, without time1/time2 pairs compared to late fusion. End-to-end fusion removes nearly half of the parameters in the late fusion architecture as only one tower of *fc* layer is used after fusion.

12.3.4 Personalizing Invasion and Expansion Networks

Predictive model personalization is a key step of model-based tumor growth prediction [5, 18, 30, 31]. In statistical learning, model validation is a natural way to optimize the pretrained model. Particularly, given tumor status at time1 and time2 already known (predict time3), the model personalization includes two steps. In the first step, the invasion network is trained on population data and time1/time2 of the target patient is used as validation. Training is terminated after a predetermined number (30) of epochs, after which the model snapshot with the lowest validation loss on the target patient data is selected. Since there are no corresponding validation data sets for the expansion network, early fusion, and end-to-end fusion, their trainings are terminated after the empirical number of 20 epochs, in order to reduce the risk of overfitting.

To better personalize the invasion network to the target patient, we propose a second step that optimizes an objective function which measures the agreement between any predicted tumor volume and its corresponding future ground-truth volume on the target patient. This is achieved by directly applying the invasion network to voxels in a tumor growth zone in the personalization volume, and later thresholding the probability values of classification outputs to reach the best objective function. Dice coefficient measures the agreement between ground truth and predicted volumes, and is used as the objective function in this study:

$$\text{Dice} = \frac{2 \times \text{TPV}}{V_{pred} + V_{gt}}, \tag{12.3}$$

where TPV is the true positive volume—the overlapping volume between the predicted tumor volume V_{pred} and the ground-truth tumor volume V_{gt}. The tumor growth zone is set as a bounding box surrounding the tumor, with pixel distances N_x, N_y, and N_z from the tumor surface in the x-, y-, and z-directions, respectively. The personalized threshold of invasion network is also used for expansion network and the three fusion networks.

12.3.5 Predicting with Invasion and Expansion Networks

During testing, given the imaging data at time1 and time2 for the target patient, one of the frameworks, the personalized invasion network, expansion network, late fusion, early fusion, or end-to-end fusion could be applied to predict the scores for every voxels in the growth zone at the future time3. The static information from time2 serves as invasion information, while the motion/change information between time1 and time2 represents the expansion information. Late fusion and end-to-end fusion feed the static and motion information to invasion and expansion subnetworks, separately, while early fusion concatenates both static and motion information as input to a one-stream ConvNet.

12.3.6 Experimental Methods and Results

Ten patients (six males and four females) with VHL disease, each with a PanNET, are studied in this section. In this data set, each patient has three time points of contrast-enhanced CT and FDG-PET imaging spanning 3 to 4 years, with the time interval of 405 ± 133 days (average ± std.). The average age of the patients at time1 is 46.9 ± 13.2 years. The image pixel sizes range between $0.68 \times 0.68 \times 1$ mm^3 and $0.98 \times 0.98 \times 1$ mm^3 for CT and $2.65 \times 2.65 \times 1.5$ mm^3 and $4.25 \times 4.25 \times 3.27$ mm^3 for PET. The tumor growth information of all patients is shown in Table 12.2. Most tumors are slow growing, while two are more aggressive and two experience shrinkage. Some tumors keep a similar growing rate as their past trend, while others have varying growing rates.

Table 12.2 Tumor information at the first, second, and third time points of ten patients

Patient ID	1st–2nd		2nd–3rd		Size (cm^3, 3rd)
	Days	Growth (%)	Days	Growth (%)	
1	384	34.6	804	33.4	2.3
2	363	15.3	363	10.7	1.4
3	378	18.9	372	7.5	0.4
4	364	150.1	364	28.9	3.1
5	426	41.5	420	68.6	3.8
6	372	7.4	360	12.5	6.3
7	384	13.6	378	−3.9	1.6
8	168	18.7	552	18.7	3.2
9	363	16.9	525	34.7	0.3
10	196	−28.9	567	17.7	0.9

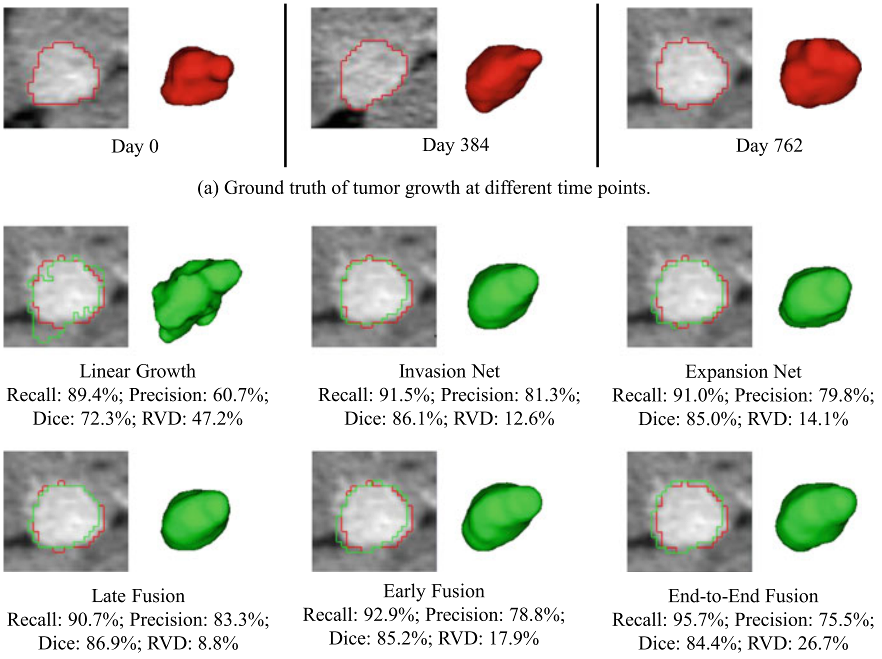
A total of 45,989 positive and 52,996 negative image patches are used for the invasion network in late fusion, and 23,448 positive and 25,896 negative image patches for both the invasion network and expansion network in other fusion (i.e., early and end to end), extracted from 10 patients. Each image patch is subtracted by the mean image patch over the training set. Data augmentation is not performed since we could not observe improvements in a pilot study. The following hyperparameters are used: initial learning rate—0.001, decreased by a factor of 10 at every tenth epoch; weight decay—0.0005; momentum—0.9; mini-batch size—512. We use an aggressive dropout ratio of 0.9 to improve generalization. Lower dropout ratios (e.g., 0.5) do not decrease performance significantly. The ConvNets are implemented using Caffe platform [13]. The parameters for tumor growth zone are set as $N_x = 3, N_y = 3$, and $N_z = 3$ for prediction speed concern. We observe that the prediction accuracy is not sensitive to the choice of these parameters, e.g., $N_{x|y|z} \geq 4$ results in similar performance. For the model personalization via Dice coefficient objective function, we vary the model thresholding values in the range of $[0.05, 0.95]$ with 0.05 intervals. The proposed method is tested on a DELL TOWER 7910 workstation with 2.40 GHz Xeon E5-2620 v3 CPU, 32 GB RAM, and a Nvidia TITAN X Pascal GPU of 12 GB of memory.

The proposed method is evaluated using leave-one-out cross-validation. In each of the ten evaluations, nine patients are used as the population training data to learn the population trend, the time1/time2 of the remaining patient is used as the personalization data set for invasion network, and time3 of the remaining patient as the to-be-predicted testing set. We obtain the model's final performance values by averaging results from the ten cross-validations. The number of parameters in each of the proposed network is reported, and the prediction performances are evaluated using measurements at the third time point by recall, precision, Dice coefficient (defined in Eq. 12.3), and relative volume difference (RVD) as defined in Eq. 12.2.

To establish a benchmark for comparisons, we implement a *linear growth* model that assumes that tumors would keep their past growing trend in the future. More specifically, we first compute the radial expansion/shrink distances on tumor boundaries between the first and second time points, and then expand/shrink the tumor boundary at the second time point to predict the third with the same radial distances. Furthermore, we compare the accuracy and efficiency of our method with two state-of-the-art tumor growth prediction methods [31, 36] which have been evaluated on a subset (seven patients, without patient 4, 7, 10 in Table 12.2) of the same data set.

Figure 12.9 shows the results of patient 7. In this case, the tumor demonstrates a nonlinear growth trend, and its size first increases from time1 to time2 but decreases a little bit from time2 to time3. Therefore, all the personalized predictive models overpredicted the tumor size (recall is higher than precision). However, our models especially the two-stream late fusion can still generate promising prediction result.

Table 12.3 presents the overall prediction performance on ten patients. Compared to the baseline linear growth method, all our methods show substantially higher performance. The performance of invasion and expansion networks is comparable. Fusion of the two networks can further improve the prediction accuracy, especially for the RVD measure. Two-stream late fusion achieves the highest mean values with


Day 0
Day 384
Day 762
Linear Growth
Recall: 89.4%; Precision: 60.7%; Dice: 72.3%; RVD: 47.2%
Invasion Net
Recall: 91.5%; Precision: 81.3%; Dice: 86.1%; RVD: 12.6%
Expansion Net
Recall: 91.0%; Precision: 79.8%; Dice: 85.0%; RVD: 14.1%
Late Fusion
Recall: 90.7%; Precision: 83.3%; Dice: 86.9%; RVD: 8.8%
Early Fusion
Recall: 92.9%; Precision: 78.8%; Dice: 85.2%; RVD: 17.9%
End-to-End Fusion
Recall: 95.7%; Precision: 75.5%; Dice: 84.4%; RVD: 26.7%

(a) Ground truth of tumor growth at different time points.

(b) Prediction at the third time point (Day 762).

Fig. 12.9 An example (patient 7) shows the tumor growth prediction by our individual and fusion networks. **a** The segmented (ground truth) tumor contours and volumes at different time points. **b** The prediction results at the third time point, with red and green, represent ground truth and predicted tumor boundaries, respectively

Table 12.3 Overall performance on ten patients—baseline linear predictive model, invasion network, expansion network, early fusion, late fusion, and end-to-end fusion. Results are estimated by the recall, precision, Dice coefficient, and relative volume difference (RVD), and are reported as mean ± std. [min, max]. The numbers of parameters for each model are provided

	Recall (%)	Precision (%)	Dice (%)	RVD (%)	#parameters
Linear	84.5 ± **7.0** [**73.3**, **97.3**]	69.5 ± 8.0 [60.7, 82.3]	75.9 ± 5.4 [69.5, 85.0]	23.1 ± 18.5 [5.9, 58.8]	–
Invasion	86.9 ± 9.4 [63.7, 97.0]	83.3 ± 5.6 [74.7, 90.2]	84.6 ± 5.1 [73.0, 90.4]	11.5 ± 11.3 [2.3, 30.0]	8.11M
Expansion	**87.6** ± 8.6 [68.3, 96.5]	82.9 ± 7.6 [76.5, **97.2**]	84.8 ± 5.4 [73.2, 91.1]	13.8 ± **6.3** [1.0, 23.5]	8.11M
Early fusion	86.4 ± 7.9 [66.6, 94.8]	84.7 ± 5.8 [77.0, 92.7]	85.2 ± 5.2 [73.9, 90.6]	9.2 ± 7.3 [2.4, **19.6**]	8.11M
Late fusion	86.9 ± 8.8 [64.0, 95.5]	**85.5** ± **4.9** [**78.6**, 91.3]	**85.9** ± 5.6 [72.8, **91.7**]	**8.1** ± 8.3 [1.0, 24.2]	16.22M
End-to-end	87.5 ± 8.1 [70.0, 96.9]	84.1 ± 5.6 [75.5, 91.3]	85.5 ± **4.8** [**76.5**, 91.5]	9.0 ± 10.1 [**0.3**, 26.7]	10.18M

Table 12.4 Comparison of performance on seven patients—baseline linear predictive model, state-of-the-art model-based [31], statistical group learning [36], and our models. Results are estimated by the recall, precision, Dice coefficient, and relative volume difference (RVD), and are reported as mean ± std. [min, max]. EG-IM-FEM* has higher performance than EG-IM, but it has some issues mentioned by the authors (Sect. VI in [31])

	Recall (%)	Precision (%)	Dice (%)	RVD (%)
Linear	84.3 ± **3.4** [78.4, 88.2]	72.6 ± 7.7 [64.3, 82.1]	77.3 ± 5.9 [72.3, 85.1]	16.7 ± 10.8 [5.2, 34.3]
EG-IM [31]	83.2 ± 8.8 [69.4, 91.1]	**86.9** ± 8.3 [74.0, **97.8**]	84.4 ± 4.0 [79.5, **92.0**]	13.9 ± 9.8 [3.6, 25.2]
EG-IM-FEM* [31]	86.8 ± 5.8 [77.6, 96.1]	86.3 ± 8.2 [72.7, 96.5]	86.1 ± **3.2** [**82.8**, 91.7]	10.8 ± 11.2 [2.3, 32.3]
Group learning [36]	87.9 ± 5.0 [81.4, 94.4]	86.0 ± 5.8 [**78.7**, 94.5]	**86.8** ± 3.6 [81.8, 91.3]	7.9 ± **5.4** [2.5, 19.3]
Invasion	88.1 ± 4.6 [81.4, 94.3]	84.4 ± 5.6 [75.0, 90.2]	86.1 ± 3.6 [80.8, 90.4]	6.6 ± 8.5 [2.3, 25.8]
Expansion	**90.1** ± 6.3 [79.1, 96.5]	81.9 ± 6.9 [76.5, 96.9]	85.5 ± 3.8 [78.7, 90.5]	14.2 ± 7.6 [1.0, 23.5]
Early fusion	88.2 ± 4.2 [81.9, 94.8]	85.2 ± 6.5 [77.0, 92.7]	86.5 ± 4.0 [80.7, 90.6]	7.5 ± 6.1 [2.5, **19.0**]
Late fusion	89.1 ± 4.3 [**83.4**, 95.5]	84.9 ± **5.2** [78.6, 93.3]	**86.8** ± 3.4 [81.8, 91.5]	**6.6** ± 7.1 [1.0, 21.5]
End-to-end	88.8 ± 5.9 [79.1, **96.9**]	84.8 ± 5.6 [77.8, 91.3]	86.6 ± 4.4 [80.5, 91.5]	**6.6** ± 8.3 [**0.4**, 24.4]

Dice coefficient of 85.9 ± 5.6% and RVD of 8.1 ± 8.3%, but requires nearly twice of the model parameters compared to early fusion. End-to-end fusion has the second highest accuracy with much less network parameters than late fusion. Nevertheless, this suggests that the mechanism of fusion ConvNets leverages the complementary relationship between static and dynamic tumor information.

Table 12.4 compares our methods with two state-of-the-art methods [31, 36] on a subset (seven patients) of our data. Out of ten patients, three patients (patient 4, 7, and 10 in Table 12.2) with aggressive and shrink tumors are not included in the experiment. As a result, the performances on seven patients (Table 12.4) are better than that on ten patients (Table 12.3). Our single network can already achieve better accuracy than the model-based method (i.e., EG-IM) [31], especially the invasion network has a much lower/better RVD than [31]. This demonstrates the high effectiveness of ConvNets (learning invasion information) in future tumor volume estimation. Network fusions further improve the accuracy and achieve comparable performance with the group learning method [36], which benefits results from integrating the deep features, hand-crafted features, and clinical factors into an SVM-based learning framework. Again, the two-stream late fusion performs the best among the proposed three fusion architectures, with Dice coefficient of 86.8 ± 3.4% and RVD of 6.6 ± 7.1%.

The proposed two-stream late fusion ConvNets (our other architectures are even faster) requires ∼5 min for training and personalization, and 15 s for prediction per patient, on average—significantly faster than the model-based approach in [31] (∼24 h—model personalization; 21 s—simulation), and group learning method in [36] (∼3.5 h—model training and personalization; 4.8 min—prediction).

12.4 Summary

In this chapter, we have demonstrated that (1) our statistical group learning method, which incorporates tumor growth patterns from a population trend and a specific patient, deep image confidence features, and time interval and clinical factors in a robust predictive model, is an effective approach for tumor growth prediction; (2) deep ConvNets can effectively represent and learn both cell invasion and mass effect in tumor growth prediction. Composite images encoding static and dynamic tumor information are fed into our ConvNet architectures to predict the future involvement region of pancreatic tumors. Both methods surpass the state-of-the-art mathematical model-based method [31] in both speed and accuracy. The invasion and expansion networks alone predict the tumor growth at higher accuracies than [31], and our proposed fusion architectures further improve the prediction accuracy. Two-stream end-to-end fusion might be a trade-off between accuracy and generalization compared with early and late fusions.

References

1. Baker S, Scharstein D, Lewis J, Roth S, Black MJ, Szeliski R (2011) A database and evaluation methodology for optical flow. IJCV 92(1):1–31
2. Bosman FT, Carneiro F, Hruban RH, Theise ND et al (2010) WHO classification of tumours of the digestive system, 4th edn. World Health Organization, Geneva
3. Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: ECCV, pp 25–36. Springer, Berlin
4. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27
5. Chen X, Summers RM, Yao J (2013) Kidney tumor growth prediction by coupling reaction-diffusion and biomechanical model. IEEE Trans Biomed Eng 60(1):169–173
6. Clatz O, Sermesant M, Bondiau PY, Delingette H, Warfield SK, Malandain G, Ayache N (2005) Realistic simulation of the 3D growth of brain tumors in MR images coupling diffusion with biomechanical deformation. TMI 24(10):1334–1346
7. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: CVPR, pp 1933–1941
8. Friedl P, Locker J, Sahai E, Segall JE (2012) Classifying collective cancer cell invasion. Nat Cell Biol 14(8):777–783
9. Greenspan H, van Ginneken B, Summers RM (2016) Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. TMI 35(5):1153–1159
10. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422

11. Hogea C, Davatzikos C, Biros G (2007) Modeling glioma growth and mass effect in 3D MR images of the brain. In: MICCAI, pp 642–650
12. Hogea C, Davatzikos C, Biros G (2008) An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects. J Math Biol 56(6):793–825
13. Jia Y (2013) Caffe: an open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/
14. Keutgen XM, Hammel P, Choyke PL, Libutti SK, Jonasch E, Kebebew E (2016) Evaluation and management of pancreatic lesions in patients with von hippel-lindau disease. Nat Rev Clin Oncol 13(9):537–549
15. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: NIPS, pp 1097–1105
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
17. Liu C (2009) Beyond pixels: exploring new representations and applications for motion analysis. PhD thesis, Massachusetts Institute of Technology
18. Liu Y, Sadowski S, Weisbrod A, Kebebew E, Summers R, Yao J (2014) Patient specific tumor growth prediction using multimodal images. Med Image Anal 18(3):555–566
19. Maddison CJ, Huang A, Sutskever I, Silver D (2015) Move evaluation in go using deep convolutional neural networks. In: ICLR
20. Morris M, Greiner R, Sander J, Murtha A, Schmidt M (2006) Learning a classification-based glioma growth model using MRI data. J Comput 1(7):21–31
21. Neverova N, Luc P, Couprie C, Verbeek J, LeCun Y (2017) Predicting deeper into the future of semantic segmentation. In: ICCV
22. Nie D, Zhang H, Adeli E, Liu L, Shen D (2016) 3D deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In: International conference on medical image computing and computer-assisted intervention, pp 212–220. Springer
23. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489
24. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: NIPS, pp 568–576
25. Swanson KR, Alvord E, Murray J (2000) A quantitative model for differential motility of gliomas in grey and white matter. Cell Prolif 33(5):317–329
26. Warburg O (1956) On the origin of cancer. Science 123(3191):309–314
27. Weisbrod AB, Kitano M, Thomas F, Williams D, Gulati N, Gesuwan K, Liu Y, Venzon D, Turkbey I, Choyke P et al (2014) Assessment of tumor growth in pancreatic neuroendocrine tumors in von hippel lindau syndrome. J Am CollE Surg 218(2):163–169
28. Weizman L, Ben-Sira L, Joskowicz L, Aizenstein O, Shofty B, Constantini S, Ben-Bashat D (2012) Prediction of brain MR scans in longitudinal tumor follow-up studies. In: MICCAI, pp 179–187. Springer
29. Wolfgang CL, Herman JM, Laheru DA, Klein AP, Erdek MA, Fishman EK, Hruban RH (2013) Recent progress in pancreatic cancer. CA Cancer J Clin 63(5):318–348
30. Wong KC, Summers RM, Kebebew E, Yao J (2015) Tumor growth prediction with reaction-diffusion and hyperelastic biomechanical model by physiological data fusion. Med Image Anal 25(1):72–85
31. Wong KCL, Summers RM, Kebebew E, Yao J (2017) Pancreatic tumor growth prediction with elastic-growth decomposition, image-derived motion, and FDM-FEM coupling. TMI 36(1):111–123
32. Yao J, Wang S, Zhu X, Huang J (2016) Imaging biomarker discovery for lung cancer survival prediction. In: MICCAI, pp 649–657. Springer
33. Yao JC, Shah MH, Ito T, Bohas CL, Wolin EM, Van Cutsem E, Hobday TJ, Okusaka T, Capdevila J, De Vries EG et al (2011) Everolimus for advanced pancreatic neuroendocrine tumors. N Engl J Med 364(6):514–523
34. Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G (2006) User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. NeuroImage 31(3):1116–1128

35. Zhang L, Lu L, Nogues I, Summers R, Liu S, Yao J (2017) Deeppap: deep convolutional networks for cervical cell classification. IEEE J Biomed Health Inform 21(6):1633–1643
36. Zhang L, Lu L, Summers RM, Kebebew E, Yao J (2017) Personalized pancreatic tumor growth prediction via group learning. In: MICCAI, pp 424–432

Chapter 13
Deep Spatial-Temporal Convolutional Neural Networks for Medical Image Restoration

Yao Xiao, Skylar Stolte, Peng Liu, Yun Liang, Pina Sanelli, Ajay Gupta, Jana Ivanidze and Ruogu Fang

Abstract Computed tomography perfusion (CTP) facilitates low-cost diagnosis and treatment of acute stroke. Cine scanning allows users to visualize brain anatomy and blood flow in virtually live time. However, effective visualization exposes patients to radiocontrast pharmaceuticals and extended scan times. Higher radiation dosage exposes patients to potential risks including hair loss, cataract formation, and cancer. To alleviate these risks, radiation dosage can be reduced along with tube current and/or X-ray radiation exposure time. However, resulting images may lack sufficient information or be affected by noise and/or artifacts. In this chapter, we propose a deep spatial-temporal convolutional neural network to preserve CTP image quality at reduced tube current, low spatial resolution, and shorter exposure time. This network structure extracts multi-directional features from low-dose and low-resolution patches at different cross sections of the spatial-temporal data and reconstructs high-quality CT volumes. We assess the performance of the network concerning image restoration at different tube currents and multiple resolution scales. The results indicate the ability of our network in restoring high-quality scans from data captured at as low as 21% of the standard radiation dose. The proposed network achieves an average improvement of 7% in perfusion maps compared to the state-of-the-art method.

Y. Xiao · S. Stolte · P. Liu · Y. Liang · R. Fang (✉)
J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, Gainesville, FL, USA
e-mail: ruogu.fang@bme.ufl.edu

P. Sanelli
Department of Radiology, Northwell Health, New York, NY, USA

Weill Cornell Medical College, New York, NY, USA

A. Gupta · J. Ivanidze
Department of Radiology, Weill Cornell Medical College, New York, NY, USA

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_13

13.1 Introduction

Acute stroke is responsible for high rates of death and chronic disability. More than 795,000 people suffer from stroke per year in the United States. Approximately 140,000 of these individuals lose their lives, which amounts to approximately 5% of all deaths [28]. Stroke afflicts individuals of all ages, but it increases in prevalence with age. In 2009, two-thirds of inpatients who were being treated for stroke were older than 65 [10]. The annual cost for stroke-related care in the United States is estimated at about 34 billion dollars [1].

Acute stroke mandates rapid diagnosis and treatment so that patients can receive optimal care. Disability 3 months after a stroke is less prevalent in individuals who received emergency room care within three hours of their initial symptoms [4]. Further, identification of the specific type of stroke, hemorrhagic or ischemic, must be performed quickly to ensure proper treatment as soon as possible. Hemorrhagic stroke is caused as a fragile blood vessel ruptures, whereas ischemic stroke is caused by thrombosis or embolism. Computed Tomography (CT) scanning provides rapid evaluation of the brain and cerebral vasculature; its high detail is particularly useful in triage of patients with ischemic or hemorrhagic stroke. Clinicians may thus optimize treatment based on the different needs of these patients; for example, ischemic stroke often requires further imaging of brain tissue hemodynamics. Therefore, these patients are more likely to benefit from CT Perfusion (CTP) to further guide treatment such as thrombolytic therapy. CTP rapidly sequences cerebrovascular physiology; it hence enables physicians to monitor blood flow actively.

Clear visualization of brain anatomy and blood flow necessitates radiocontrast agent injection and repeated CT scans. This comes at the cost of extensive X-ray radiation exposure to the patient; for instance, a cerebral CTP scan that lasts 40 seconds exposes a patient suffering from acute stroke to radiation that is equivalent to a year of exposure from natural surroundings [17, 23]. The risk associated with such a procedure can be particularly expensive—comparatively, chest X-ray equates to approximately ten days of natural exposure. Acquiring data on the entire brain by CTP/CT Angiography (CTA) requires a mean dose of 6.8 mSv [24], which is significantly higher than the radiation acquired from natural background sources. Annually, natural surroundings expose individuals to approximately 2.4 mSv [5]. Repeated cerebral scanning accumulates the radiation exposure to patients; hence, it influences the development of health risks such as hair loss/epilation [25], cataract formation [6], and cancer [7, 13]. Due to the convenience and high visualization that CT scans provide, they are being used with increasing prevalence in medical care. Use in the United States equates to approximately 80 million CT scans per year. Therefore, there is an increasing need for solutions that reduce radiation dose while preserving vital image information.

Researchers have approached radiation dose reduction using many different methods; they involve two primary focuses: optimizing CT systems and reducing contrast dose. Optimizing CT systems consists of shortening temporal sampling frequency and reducing radiation sources like tube current and the number of beams and receptors. However, lowering radiation exposure using these methods will increase image noise and artifacts. Hence, radiologists must balance image quality and patient risk for effective diagnostic and therapeutic potential. In this paper, we propose a deep spatial-temporal convolutional neural network for image restoration. As such, we aim to reduce CTP radiation exposure to patients and maintain high image quality.

Two main components compose this network, including the following: super-resolution denoising nets (SRDN) and multi-directional conjunction to jointly address image super-resolution (SR) and denoising. This work offers contributions which are fivefold: (1) SRDN extracts spatial and temporal features from CTP image patches; it can thus use this cross section information to represent spatial and temporal details simultaneously. (2) SRDN performs image SR and denoising conjointly. It effectively handles CTP images with multi-level noise and multi-scale resolution. (3) We achieve high performance on 3D spatiotemporal CTP data through the integration of multiple SRDNs based on different cross sections into one multi-directional network. (4) Through extensive experiments, we demonstrate the ability of the proposed network in low-dose CTP images restoration. The proposed network addresses reduced radiation dose in images collected with low tube current, shorten exposure time, and poor spatial resolution. The experiment results provide comparable image quality and accuracy to images obtained at standard clinical doses. (5) We generate perfusion maps showcasing cerebral blood flow (CBF) and cerebral blood volume (CBV). These maps show that our method provides comparable results to the state-of-the-art method.

Our proposed network can restore images using three types of limited data at the same time, including the following: data using low tube current, decreased temporal sampling rate, and poor spatial resolution. Previously, no work had simultaneously addressed these data limitations within one deep learning structure. Our network yields an average of 21% improvement in peak signal-to-noise ratio (PSNR) compared to low-dose scans using state-of-the-art approaches, despite its use of 20% to 40% lower tube currents. Therefore, a deep learning approach has high potential in reducing CTP radiation exposure.

13.2 Related Work

Researchers have devoted themselves to seeking solutions for reducing the radiation exposure of CTP imaging. The development of low-dose CTP protocols is essential to reduce the health risks of excessive radiation exposure to the patients. Below, we summarize several methods of accomplishing this by adjusting different acquisition parameters such as tube current, temporal sampling frequency, and spatial resolution. However, those parameters are meticulously related to the quality of the reconstructed CTP images, especially for generating perfusion maps for treatment decisions. In this section, we introduce several dose reduction approaches and the methods for restoring the quality of the reconstructed CTP images; the later of this section combines image processing strategies and deep learning approaches.

13.2.1 Radiation Dose Reduction

There are three types of radiation dose reduction approaches for CTP imaging, namely the tube current, spatial resolution, and X-ray exposure time. The radiation dose and the tube current is linear related; for example, in order to reduce 50% radiation dose, we will need to lower the tube current to 50% of the original value. However, CTP sequences scanned by low tube currents will have high image noise and artifacts, as image noise and the square root of tube current have an inverse proportional relationship. Methods for reducing the image noise and artifacts is necessary for radiation dose reduction and image quality preservation. Current studies demonstrate that it is possible and effective to maintain image quality at reduced tube currents [14, 18]. In [29], a study of the pediatric abdomen, pelvis, and chest CT examinations demonstrate that a 50% dose reduction can still maintain diagnostic quality. Similar to the decrements of the tube current, the reduction in X-ray exposure time regarding the scanning time intervals between frames will reduce the radiation correspondingly, as the time for patients undergoing radiation exposure is reduced. Low-dose CT scans result in low spatial resolution. Thus, recovering spatial resolution from these scans could allow the overall X-ray radiation exposure to patients to be reduced.

13.2.2 Image Restoration

Image restoration is the operation of estimating the clean and original images from the low-quality observations by reducing the noise and recovering the resolution loss. The process of image restoration can be performed in two domains: the frequency domain and the image domain. The most conventional technique for image restoration is deconvolution, which is an inverse process for generating low-quality images. It operates in the frequency domain after applying the Fourier transform and the point spread function and undoes the resolution loss caused by the degrading factors. The consequence of the deconvolution by direct inversion of the point spread function is shown as a poor matrix condition number, which amplifies noise and creates imperfect reconstructed images.

In recent years, deep learning methods emerging flourishingly in various computer vision tasks, including image classification [11] and object detection [9], and have dramatically improved the performance of these systems. These approaches have also achieved significant improvement in image restoration [3, 27]. Convolutional neural network (CNN), one of the most esteemed deep learning architectures, shows promising results for image-based problems. CNN structures are usually composed of several convolutional layers with activation layers, followed by one or more fully connected layers. The purpose of CNN architecture design is for utilizing image structures via local connections, weights sharing, and nonlinearity. Another benefit of CNN is that they are easier to train and have fewer parameters than fully connected networks with the same number of hidden units.

Image Super-Resolution aims at restoring HR images from the observed LR images. By using different portions of the LR images, or separate images, SR methods intend to approximate what the HR image may look alike. Deep learning models are recently widely used for image SR problems [8, 15, 19, 21]. However, most of the SR frameworks focus on 2D images, as involving the temporal dimension is more challenging, especially in CTP imaging.

Image Denoising aims at recovering the clean image from an observed noisy image. One of the main challenges for image denoising is to identify the noise and remove it accurately. Deep learning-based methods [16, 30] have shown many advantages in learning the mapping of the low-quality images to the high-quality images, which they accomplish using multi-layer CNNs that are trained on tens of thousands of samples. However, the paired training data are usually scarce in the medical field. Hence, an effective learning-based model is desired.

13.2.3 Spatial-Temporal Architecture

In our previous work, we proposed STAR [26] for low-dose CTP image super-resolution. It preserves image quality at reduced scanning time and leading to radiation reduction to only one-third of the original level. STAR is an image-based dose reduction approach that focuses on super-resolution only. Through this work, we found that features extracted from both spatial and temporal directions improve SR performance. The integration of multiple single-directional networks (SDNs) can boost the performance of SR for the spatiotemporal CTP data. The experimental results show that the proposed basic model of SDN improves both spatial and temporal resolution, while the multi-directional conjoint network further enhances the SR results—comparing favorably with only temporal or only spatial SR. However, this work only addresses the low spatial and temporal resolutions, missing the important noise issue in low-dose CTP.

In this chapter, we propose a deep spatial-temporal network for CTP image restoration and radiation reduction. This network structure integrates several SRDNs at different cross sections for both image super-resolution and denoising simultaneously. The structure of the proposed spatial-temporal network is explained in Sect. 13.3. In Sect. 13.4, we provide the experiment platform setup and describe the data acquisition method and the preprocessing procedures. In Sect. 13.5, we detail the experiments and results. Finally, Sect. 13.6 concludes the paper.

13.3 Methodology

In this section, we propose our deep learning approach for low-dose CTP image restoration. In Sect. 13.3.1, we introduce how the spatial-temporal patches were generated. Then, in Sect. 13.3.2, we describe our deep spatial-temporal network structure with multiple cross-sectional inputs.

13.3.1 Spatial-Temporal Patches

The data acquired from one CTP scanning is stored as a sequence of slices. We convert the sequence by concatenating the slices into a 3D matrix. The three dimensions of this matrix are X, Y, and T, where X and Y represent the spatial dimensions of the 2D slices and T indicates the temporal dimension of the sequence. The input 2D LR patches are then generated based on these three dimensions for the deep spatial-temporal network, including use in both image restoration tasks: image super-resolution and image denoising.

Our 2D patches include three directional combinations: $X \times Y$, $X \times T$, and $Y \times T$. In our super-resolution task, we create 2D LR patches through down-sampling on the spatial and the temporal directions. For example, we simulate two times longer scanning intervals by removing every other pixel along the T direction. In our denoising task, we add spectrum Gaussian noise on the entire CTP volume to model images that are produced with lower tube current. A complete description of noise simulation and data preprocessing can be found in Sect. 13.4.3. Our resulting noise models represent 2D patches along each of the three cross sections used. On the combined denoising and SR task, we begin with the noisy images and create LR patches as described above. Finally, the respective input patches are feed into the proposed network structure—the relevant original CTP volume slices serve as ground truth. During training, the network learns the spatial-temporal details that map the LR and/or noisy patches to their original patches. By using these features, the network generates HR and/or denoised output images during testing.

13.3.2 Deep Spatial-Temporal Network

Our proposed deep spatial-temporal network is a convolutional-based end-to-end architecture for image restoration. The network structure is shown in Fig. 13.1. As described in Sect. 13.3.1, we utilize three cross sections as input. Each cross section input is passed through an SRDN network and combined by a conjoint layer for the final high-quality output. The conjoint layer is a simple average function that calculates the mean value of the outputs from the three SRDN pathways. The combination of the various features extracted from different directions of the CTP

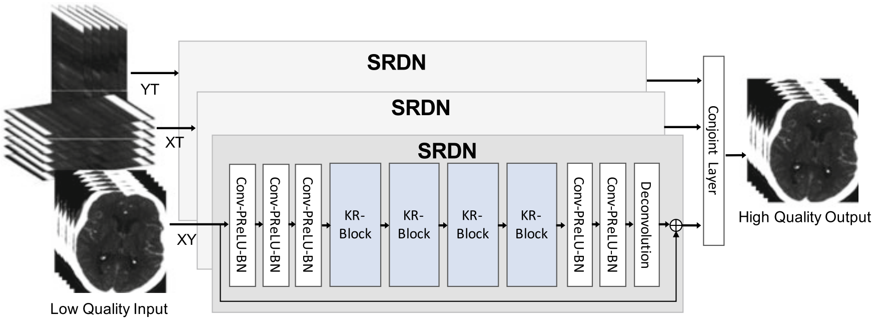
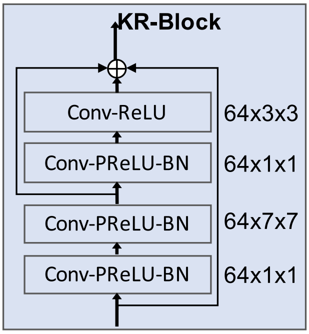


Fig. 13.1 Deep spatial-temporal network architecture. It takes low-quality inputs from three cross sections: XY, XT, and YT. Each cross section goes through one SRDN, and the outputs of each SRDN meet in a conjoint layer to provide the final high-quality outputs of the deep spatial-temporal network

Fig. 13.2 Kernel regulation block (KR-block) architecture. It comprises two 1 × 1 convolution components for computation reduction, one 7 × 7 convolution layer, and one 3 × 3 convolution module for regularizing the features extracted by the preceding large size kernels



volume is beneficial to enhance the capability of the network inference and generality. Since multi-directional inputs provide different perspectives of the 3D data, they cannot merely be regarded as feeding more training data into multi-networks. Instead, they complement each other to encode the sparse features through the network.

As mentioned before, there is an SRDN that corresponds to each cross section input. SRDN is an end-to-end structure that learns from pair-wise LR/noisy patches with their original images and outputs high-quality CT images based on low-quality input images while testing. The main functional part of SRDN is built by stacking four modularized Kernel Regulation Blocks (KR-Block). The structure of KR-Block is shown in Fig. 13.2. The design of our KR-Block is inspired by GoogLeNet [22], which has a combination of kernels of varying sizes. Specifically, KR-Block comprises two 1 × 1 convolutional layers, one 7 × 7 convolutional layer, and one 3 × 3 convolutional layer for regulating the features extracted by the 7 × 7 convolutional layer. The combination of large and small filters is to balance the extraction of subtle and edge features. Moreover, the KR-Block is embedded with skip connections, which allow reference to the feature mapping from previous layers and boost the network performance.

Several KR-blocks are cascaded to perform feature regulation, mapping, and transformation. Residual learning is performed by skip connections, which connect the outputs of two adjacent KR-blocks. The use of skip connections between KR-blocks leads to faster and more stable training. Using a shortcut between input and the end of the network is to allow the original input information better assisted final image reconstruction. It is because the input data contains significant real pixel information that can be taken as a prior, thus relaxing the network inference difficulty. The convolutional layers before the last layer have 128 filters with a size of 3 × 3. We utilize a deconvolutional layer with a filter size of 3 × 3 as our last layer.

13.4 Platform and Data Acquisition

13.4.1 Computational Platform

The proposed network structure is building on top of deep learning framework Caffe [12]. All experiments are conducted by a GPU workstation that contains four NVIDIA PASCAL xp GPUs. We use MATLAB (Version R2016b) for data preprocessing and post-analysis, as it is an efficient programming language for matrix-based image processing.

13.4.2 Datasets

We evaluate the proposed method on 23 stroke patients' CTP sequences. All CTP sequences are scanned using the same acute stroke protocol for patients from August 2007 to June 2010 using GE Lightspeed or Pro-16 scanners (General Electric Medical Systems, Milwaukee, WI). The scanners are in cine 4i scanning mode and perform 45 s acquisitions at one rotation per second using 80 kVp and 190 mAs. Approximately 45 mL of non-ionic iodinated contrast was administered intravenously at 5 mL/s using a power injector with a 5 s delay. The thickness of the brain region at the z-axis is 20 mm for each sequence. Each sequence has four slices along the z-axis, where each slice is 5 mm thick (cross-plane resolution). The brain region has 0.43 spatial resolution (in-plane resolution) on the xy-plane. The slices within one CTP sequence are intensity normalized and co-registered over time. The entire volume size of one patient is 512 × 512 × 4 × 119, where 512 is the height and width of each CT slice, 4 is the number of slices on the z-axis, and 119 is the number of frames in the CTP sequence. In this paper, we only select one slice along the z-axis; thus the size of resulting the CTP volume is 512 × 512 × 119, denoted as $X \times Y \times T$.

We randomly split the patients into three groups: 12 patients for training, 4 patients for validation, and 7 patients for testing. As each patient has 119 slices, the training, validation, and testing sets resulted in 1428, 476, and 833 images in XY cross section (the spatial direction), respectfully. We only maintain brain regions in the images for the other two cross sections, XT and YT, or about 300 pixels for the X and the Y directions. Therefore for these cross sections, we estimate that we have 3600 images for training, 1200 for validation, and 2100 for testing. We use the patch-based method in this paper, so the images are further cropped into patches of size 41 × 41 with a stride of 21, which resulted in 822,528 and 274,176 patches in XY cross section and 75,600 patches and 25,200 patches in XT and YT cross sections, respectively, for training and validation.

13.4.3 Low Radiation Dose Simulation and Data Preprocessing

Below, we explain three methods that are commonly used to simulate low radiation dose CTP images, including the following: decreasing the tube current, shortening the time that the patient is exposed to X-ray radiation, and lowering the spatial resolution of the imaging system.

- **Low Tube Current** We simulated low-dose conditions by adding spatially correlated noise, specifically Gaussian noise, to the original high-dose CT images. Our procedure followed that described in [2], and the original volumes were scanned at tube current $I_0 = 190\,\text{mAs}$. Accordingly, tube current I and noise standard deviation σ follow an inverse relationship, in which the Gaussian noise level is modified to model the desired tube current using the equation:

  $$\sigma = K \times \sqrt{\frac{1}{I} - \frac{1}{I_0}} \tag{13.1}$$

  where $K = 103.09\,\text{mA}^{\frac{1}{2}}$ is computed based on phantom studies. We model three noise levels in our study using images generated at corresponding tube currents: 40, 60, and 80 mAs.
- **Shorter X-ray Exposure Time** To simulate shortened exposure of patients to X-ray radiation, we remove frames from the original CTP volumes at specified intervals. We down-sample the original CTP volumes at two times shorter S_2 and three times shorter S_3, then compare these volumes to that at the original sampling rate. For example, we generate a temporal sampling rate that is two times shorter by using a down-sampled version of the original CTP volume which has had every other frame removed. Finally, we use bicubic interpolation to restore all volumes to the original volume size.
- **Low Spatial Resolution** We likewise model CT images with a low spatial resolution by decreasing the spatial sampling rate that is used to observe the image volume. For instance, we sample every other pixel grid-wise in the high radiation dose volumes to yield images that are down-sampled to half of the original spatial

sampling rate. Our low-resolution (LR) images represent two scales S_i: two times down-sampled S_2, and three times down-sampled S_3. Similar to the temporal case, we return the LR images to the original image size using the bicubic method.

The inputs for our proposed network include the following directional cross sections: XY, XT, and YT. We detail the image patches that we use at these cross sections in Sect. 13.3.1, which we preprocess accordingly: (1) We add Gaussian noise or apply spatial/temporal down-sampling to the original CT volumes individually to denoising or super-resolution cases, respectively. (2) We add the noise and then apply the down-sampling in sequence for combined denoising and super-resolution tasks.

13.5 Experiments and Results

We train the proposed network using low-quality images from 12 patients' different cross sections. Then, we use seven patients to test the performance of our model and report the average results. In our study, we test each of the cross sections, including the following: spatial only (XY), temporal only (XT and YT), and the combination of spatial and temporal dimensions. For the XT and YT cross sections, we concatenate the 2D images into 3D volumes and calculate the performance based on the spatial (XY) direction.

13.5.1 Evaluation Metrics

We evaluate the experiment performance according to structural similarity (SSIM) index and PSNR. SSIM compares two images using terms for luminance $l(x, y)$, $s(x, y)$, contrast $c(x, y)$, and structure $s(x, y)$, where x and y are two images. We calculate SSIM based on the following equations:

$$\mathit{SSIM(x,y)} = [l(x, y) \cdot c(x, y) \cdot s(x, y)] \tag{13.2}$$

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, s(x, y) = \frac{2\sigma_x y + c_3}{\sigma_x^2 + \sigma_y^2 + c_3} \tag{13.3}$$

where μ_x, μ_y, σ_x, σ_y, σ_{xy} are the local means, standard deviations, and cross-covariance for images x and y. The value of c_1, c_2, and c_3 are set as 6.5025, 58.5225, and 29.26125.

PSNR relates the maximum intensity in the ground truth image, I_{max} , and the power of the corrupting noise σ. The corrupting noise defines the root mean square error (MSE) between the enhanced and ground truth images, and it plays a role in representation fidelity.

$$\mathit{PSNR} = 20 \log_{10} \frac{I_{max}}{\sigma} \tag{13.4}$$

13.5.2 Spatial-Temporal Super-Resolution and Denoising

We run denoising and SR experiments by training our network on low-dose and low-resolution inputs. Figure 13.3 shows the PSNR comparison among multi-scale expected patch log likelihood (MS-EPLL) [20], spatiotemporal architecture for super-resolution (STAR) [26], and the proposed network. Our results display the average performance for each method over the output of seven patients' 833 slices. The STAR and our proposed methods both include spatial SR (XY direction) and temporal SR (XT and YT directions) separately and jointly. In training and testing stages, we down-sample the noisy images and rescale them back to size using bicubic interpolation. The resulting LR image has fewer discontinuities and is smoother along edges, but image artifacts are also enlarged.

Figure 13.3 compares the denoising performance at three levels of tube current (40, 60, and 80 mAs) and the SR performance at two down-sampling scales (down-sampled to 1/2 and to 1/3). For the SR experiments, we down-sample along the spatial dimension and the temporal dimension for different spatial-temporal cross sections. In this figure, we highlight the highest PSNR value of the results from our proposed network under different down-sampling rates and noise levels. STAR and the proposed network both outperform the MS-EPLL method. The conjoint model of the proposed network achieves the highest performance for two times down-sampling at all noise levels and the proposed method's temporal only model performs best for three times down-sampling at all noise level cases. Our proposed network performs superiorly in low-dose CT images that also have poor spatial and temporal resolutions. The proposed network structure yields an average improvement of 8.08

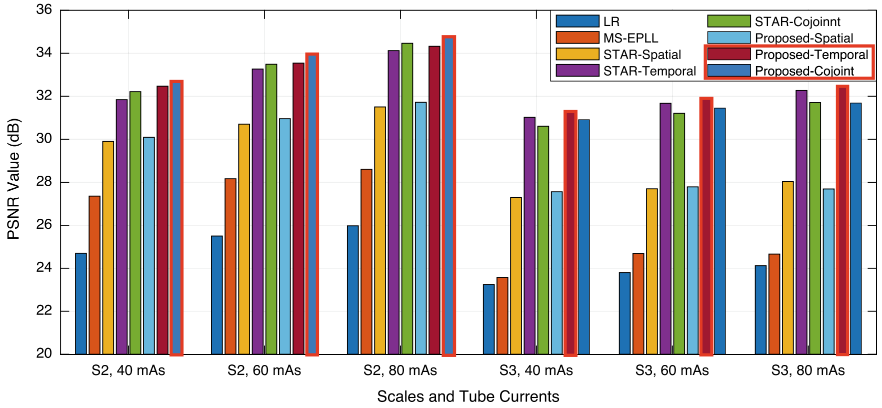
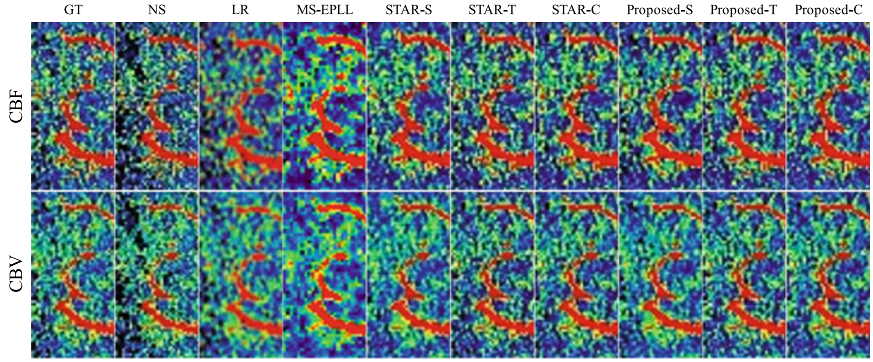


Fig. 13.3 Average PSNR comparison of seven patients' 833 CTP slices for different conditions. In this figure, three methods are compared: MS-EPLL [20], STAR [26], and the proposed network. The conditions include three types of tube current (40, 60, 80 mAs) and two kinds of SR scales (S2: down-sampling to 1/2, S3: down-sampling to 1/3). LR means the low-resolution inputs after down-sampling from the noise image. The best results are highlighted for different scenarios. mAs is the unit for tube current–time product



PSNR	CBF	15.2819	12.4629	12.5136	16.2096	16.5549	16.3982	15.5352	16.7103	**18.5072**
	CBV	16.1098	13.4545	13.0900	17.5337	18.6950	18.8577	17.6539	18.8597	**19.1962**
SSIM	CBF	0.6906	0.3435	0.5284	0.7491	0.7570	0.7574	0.6961	0.7652	**0.8454**
	CBV	0.7691	0.5342	0.6142	0.8314	0.8570	0.8703	0.8304	0.8681	**0.8807**

Fig. 13.4 Performance comparison of perfusion maps (CBF and CBV) regions for patient #21 when reducing the tube current to 80 mAs with down-sample rate of two (two times low spatial and two times shorter exposure time). The notation for each column is as follows: GT: the ground truth images, NS: the simulated low-dose images by adding spectrum Gaussian noise, LR: the LR inputs that are further down-sampled on NS images, MS-EPLL: the restoration results from MS-EPLL method (spatial only), STAR-S: the reconstruction result of STAR, the spatial only model, STAR-T: the reconstruction result of STAR, the temporal only model, STAR-C: the reconstruction result of STAR, the conjoint model (spatial+temporal), Proposed-S: the reconstruction result of the proposed network structure, the spatial only model, Proposed-T: the reconstruction result of the proposed network, the temporal only model, Proposed-C: the reconstruction result of the proposed network, the conjoint model (spatial+temporal). The table below is the quantitative evaluation regarding PSNR and SSIM corresponded to the images above. The best performance among all methods is highlighted with bold font

dB compared to the LR inputs and 4 dB compared to the MS-EPLL method. The proposed network achieves an average 1.1% improvement for the joint denoising and SR task than the state-of-the-art method (STAR's conjoint model). These results indicate that the proposed network performs well on the low tube current and low-resolution images. The performance particularly holds for temporal direction; thus, the radiation exposure time can be reduced as CTP volumes contain more information that is related to the down-sampled slices to reconstruct CT frames.

Doctors use perfusion maps, as opposed to CTP frames, to diagnose patients due to their ability to show hemodynamic changes during blood flow. Therefore, we emphasize performance in perfusion map generation for optimal clinical relevance. Figure 13.4 compares the region of interest on the perfusion maps of patient #21 when different models are used. From left to right, different columns represent the following: GT: the ground truth images, NS: the simulated low-dose images by adding spectrum Gaussian noise, LR: the LR inputs that are further down-sampled on NS images, MS-EPLL: the restoration results from MS-EPLL method (spatial only), STAR-S: the reconstruction result of STAR, the spatial only model, STAR-T: the reconstruction result of STAR, the temporal only model, STAR-C: the reconstruction result of STAR, the conjoint model (spatial + temporal), Proposed-S: the reconstruction result of the proposed network structure, the spatial only model, Proposed-T: the reconstruction result of the proposed network, the temporal only model, Proposed-C: the reconstruction result of the proposed network, the conjoint model (spatial+temporal).

Figure 13.4 shows that the proposed network generates perfusion maps that are significantly closer to the ground truth image as compared to MS-EPLL and STAR models. Our network's spatiotemporal structure preserves details in the region of interest (ROI). We calculate the PSNR and SSIM values for the perfusion maps, as listed in the form below the figure. On average, our proposed network yields a 45% higher PSNR value over the LR inputs and a 7% improvement over the STAR-conjoint method (state-of-the-art). MS-EPLL fails to address super-low-dose cases, as it produces perfusion maps that are of lower quality than the LR images. In SSIM comparisons, the proposed network's conjoint model achieves the highest value in all the cases as well as an average 6% SSIM value than the state-of-the-art model. Thus, these results support that our method is more robust for joint denoising and SR issues.

13.6 Conclusion

In this paper, we propose a novel multi-directional deep spatial-temporal framework to restore low radiation dose CTP images for diagnosis and treatment of stroke. It outputs high-resolution and low-noise images based on inputs that are scanned at lower tube current, shorter times of radiation exposure, and lower spatial resolution. Our framework jointly addresses image denoising and super-resolution tasks. With proper training, the CNN-based SRDN handles prior and data fidelity terms through a sequence of filter-based KR-Blocks. Each component within this block offers a distinct ability to deal with image noise and resolution problems simultaneously.

Our proposed network achieves superior reconstruction results for a mix of low-resolution and noise conditions based on feature extraction for both the spatial and temporal domains. As such, the network successfully reconstructs high-quality CT volumes based on low-dose and low-resolution patch inputs. The results of our experiments support that our framework can maintain diagnostic image quality while CT scanning conditions are reduced from the commercial standard as follows: tube current, X-ray radiation exposure time, and spatial resolution reduced to 21%, 1/3, and 1/3, respectively. Therefore, our approach effectively and efficiently reduces the radiation exposure in CTP imaging. In future work, we will extend our methodology to reduce radiation dose in multimodal imaging. Specifically, we will holistically combine low-dose non-contrast CT, CTA, and CTP images.

Acknowledgements This work is partially supported by the National Science Foundation under Grant No. IIS-1564892, the University of Florida Informatics Institute SEED Funds and the UF Clinical and Translational Science Institute, which is supported in part by the NIH National Center for Advancing Translational Sciences under award number UL1 TR001427. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

1. Benjamin EJ, Blaha MJ, Chiuve SE, Cushman M, Das SR, Deo R, Floyd J, Fornage M, Gillespie C, Isasi C et al (2017) Heart disease and stroke statistics-2017 update: a report from the american heart association. Circulation 135(10):e146–e603
2. Britten A, Crotty M, Kiremidjian H, Grundy A, Adam E (2004) The addition of computer simulated noise to investigate radiation dose and image quality in images with spatial correlation of statistical noise: an example application to X-ray CT of the brain. Br J Radiol 77(916):323–328
3. Burger HC, Schuler CJ, Harmeling S (2012) Image denoising: can plain neural networks compete with BM3D? In: IEEE conference on computer vision and pattern recognition. IEEE, pp 2392–2399
4. Centers for Disease Control and Prevention (2008) Awareness of stroke warning symptoms-13 States and the District of Columbia, 2005. Morb Mortal Wkly Rep 57(18):481
5. Cho G, Kim JH, Park TS, Cho K (2017) Proposing a simple radiation scale for the public: Radiation index. Nucl Eng Technol 49(3):598–608
6. Chodick G, Bekiroglu N, Hauptmann M, Alexander BH, Freedman DM, Doody MM, Cheung LC, Simon SL, Weinstock RM, Bouville A et al (2008) Risk of cataract after exposure to low doses of ionizing radiation: a 20-year prospective cohort study among US radiologic technologists. Am J Epidemiol 168(6):620–631
7. de González AB, Mahesh M, Kim K-P, Bhargavan M, Lewis R, Mettler F, Land C (2009) Projected cancer risks from computed tomographic scans performed in the United States in 2007. Arch Intern Med 169(22):2071–2077
8. Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: European conference on computer vision. Springer, pp 184–199
9. Erhan D, Szegedy C, Toshev A, Anguelov D (2014) Scalable object detection using deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2147–2154
10. Hall MJ, Levant S, DeFrances CJ (2012) Hospitalization for stroke in US hospitals, 1989–2009. Diabetes 18(23):23
11. Erhan D, Szegedy C, Toshev A, Anguelov D (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
12. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. ACM, pp 675–678
13. Journy NM, Lee C, Harbron RW, McHugh K, Pearce MS, de González AB (2017) Projected cancer risks potentially related to past, current, and future practices in paediatric CT in the United Kingdom, 1990–2020. Br J Cancer 116(1):109
14. Juluru K, Shih J, Raj A, Comunale J, Delaney H, Greenberg E, Hermann C, Liu Y, Hoelscher A, Al-Khori N et al (2013) Effects of increased image noise on image quality and quantitative interpretation in brain CT perfusion. Am J Neuroradiol 34(8):1506–1512
15. Kim J, Kwon Lee J, Mu Lee K (2016) Accurate image super-resolution using very deep convolutional networks. In: The IEEE conference on computer vision and pattern recognition, June 2016
16. Mao XJ, Shen C, Yang YB (2016) Image restoration using convolutional auto-encoders with symmetric skip connections. arXiv:1606.08921

17. Mettler FA Jr, Bhargavan M et al (2009) Radiologic and nuclear medicine studies in the United States and Worldwide: frequency, radiation dose, and comparison with other radiation sources-1950-2007 1. Radiology 253(2):520–531
18. Murphy A, So A, Lee T-Y, Symons S, Jakubovic R, Zhang L, Aviv RI (2014) Low dose CT perfusion in acute ischemic stroke. Neuroradiology 56(12):1055–1062
19. Oktay O, Bai W, Lee M et al (2016) Multi-input cardiac image super-resolution using convolutional neural networks. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 246–254
20. Papyan V, Elad M (2016) Multi-scale patch-based image restoration. IEEE Trans Image Process 25(1):249–261
21. Shi W et al (2016) Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1874–1883
22. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
23. Takei Y, Miyazaki O, Matsubara K, Shimada Y, Muramatsu Y, Akahane K, Fujii K, Suzuki S, Koshida K (2016) Nationwide survey of radiation exposure during pediatric computed tomography examinations and proposal of age-based diagnostic reference levels for Japan. Pediatr Radiol 46(2):280–285
24. Thierfelder KM, Sommer WH, Baumann AB, Klotz E, Meinel FG, Strobl FF, Nikolaou K, Reiser MF, von Baumgarten L (2013) Whole-brain CT perfusion: reliability and reproducibility of volumetric perfusion deficit assessment in patients with acute ischemic stroke. Neuroradiology 55(7):827–835
25. Wintermark M, Lev M (2010) FDA investigates the safety of brain perfusion CT. Am J Neuroradiol 31(1):2–3
26. Xiao Y, Gupta A, Sanelli PC, Fang R (2017) STAR: spatio-temporal architecture for super-resolution in low-dose CT perfusion. In: International workshop on machine learning in medical imaging. Springer, pp 97–105
27. Xie J, Xu L, Chen E (2012) Image denoising and inpainting with deep neural networks. In: Advances in neural information processing systems, pp 341–349
28. Yang Q, Tong X, Schieb L, Vaughan A, Gillespie C, Wiltz JL, King SC, Odom E, Merritt R, Hong Y et al (2017) Vital signs: recent trends in stroke death rates-United States, 2000–2015. Morb Mortal Wkly Rep 66(35):933–939
29. Yu L, Fletcher JG, Shiung M, Thomas KB, Matsumoto JM, Zingula SN, McCollough CH (2015) Radiation dose reduction in pediatric body CT using iterative reconstruction and a novel image-based denoising method. Am J Roentgenol 205(5):1026–1037
30. Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 26(7):3142–3155

Chapter 14
Generative Low-Dose CT Image Denoising

Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K. Kalra, Yi Zhang, Ling Sun and Ge Wang

Abstract The continuous development and extensive use of CT in medical practice have raised a public concern over the associated radiation dose to patients. Reducing the radiation dose may lead to increased noise and artifacts, which can adversely affect radiologists' judgment and confidence. Hence, advanced image reconstruction from low-dose CT data is needed to improve the diagnostic performance, which

Q. Yang · P. Yan (✉) · G. Wang
Department of Biomedical Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
e-mail: yanp2@rpi.edu

Q. Yang
e-mail: yangq4@rpi.edu

G. Wang
e-mail: wangg6@rpi.edu

Y. Zhang · H. Yu
Department of Electrical and Computer Engineering, University of Massachusetts Lowell, Lowell, MA 01854, USA
e-mail: yanbo_zhang@uml.edu

H. Yu
e-mail: hengyong-yu@ieee.org

Y. Shi · X. Mou
Institute of Image Processing and Pattern Recognition, Xian Jiaotong University, Xian 710049, Shaanxi, China
e-mail: xqmou@mail.xjtu.edu.cn

M. K. Kalra
Department of Radiology, Harvard Medical School, Massachusetts General Hospital, Boston, MA, USA
e-mail: mkalra@mgh.harvard.edu

Y. Zhang
College of Computer Science, Sichuan University, Chengdu 610065, China
e-mail: yzhang@scu.edu.cn

L. Sun
Department of Radiology, West China Hospital, Huaxi MR Research Center (HMRRC), Sichuan University, Chengdu 610041, China
e-mail: 251834489@qq.com

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_14

is a challenging problem due to its ill-posed nature. Over the past years, various low-dose CT methods have produced impressive results. However, most of the algorithms developed for this application, including the recently popularized deep learning techniques, aim for minimizing the mean squared error (MSE) between a denoised CT image and the ground truth under generic penalties. Although the peak signal-to-noise ratio (PSNR) is improved, MSE- or weighted-MSE-based methods can compromise the visibility of important structural details after aggressive denoising. This paper introduces a new CT image denoising method based on the generative adversarial network (GAN) with Wasserstein distance and perceptual similarity. The Wasserstein distance is a key concept of the optimal transport theory, and promises to improve the performance of GAN. The perceptual loss suppresses noise by comparing the perceptual features of a denoised output against those of the ground truth in an established feature space, while the GAN focuses more on migrating the data noise distribution from strong to weak statistically. Therefore, our proposed method transfers our knowledge of visual perception to the image denoising task and is capable of not only reducing the image noise level but also trying to keep the critical information at the same time. Promising results have been obtained in our experiments with clinical CT images.

14.1 Introduction

X-ray computed tomography (CT) is one of the most important imaging modalities in modern hospitals and clinics. However, there is a potential radiation risk to the patient since X-rays could cause genetic damage and induce cancer in a probability related to the radiation dose [6, 11]. Lowering the radiation dose increases the noise and artifacts in reconstructed images, which can compromise diagnostic information. Hence, extensive efforts have been made to design better image reconstruction or image processing methods for low-dose CT (LDCT). These methods generally fall into three categories: (a) sinogram filtration before reconstruction [30, 41, 42], (b) iterative reconstruction [5, 20], and (c) image post-processing after reconstruction [10, 16, 29].

Over the past decade, researchers were dedicated to developing new iterative algorithms (IR) for LDCT image reconstruction. Generally, those algorithms optimize an objective function that incorporates an accurate system model [12, 27], a statistical noise model [15, 33, 43], and prior information in the image domain. Popular image priors include total variation (TV) and its variants [28, 35, 38], as well as dictionary learning [45, 47]. These iterative reconstruction algorithms greatly improved image quality but they may still lose some details and suffer from remaining artifacts. Also, they require a high computational cost, which is a bottleneck in practical applications.

On the other hand, sinogram pre-filtration and image post-processing are computationally efficient compared to iterative reconstruction. Noise characteristic was well modeled in the sinogram domain for sinogram-domain filtration. However, sinogram data of commercial scanners are not readily available to users, and these methods may suffer from resolution loss and edge blurring. Sinogram data need to be carefully processed; otherwise, artifacts may be induced in the reconstructed images.

Differently from sinogram denoising, image post-processing directly operates on an image. Many efforts were made in the image domain to reduce LDCT noise and suppress artifacts. For example, the nonlocal means (NLM) method was adapted for CT image denoising [29]. Inspired by compressed sensing methods, an adapted K-SVD method was proposed [10] to reduce artifacts in CT images. The block-matching 3D (BM3D) algorithm was used for image restoration in several CT imaging tasks [16, 23]. With such image post-processing, image quality improvement was clear but oversmoothing and/or residual errors were often observed in the processed images. These issues are difficult to address, given the nonuniform distribution of CT image noise.

The recent explosive development of deep neural networks suggests new thinking and huge potential for the medical imaging field [39, 40]. As an example, the LDCT denoising problem can be solved using deep learning techniques. Specifically, the convolutional neural network (CNN) for image super-resolution [14] was recently adapted for low-dose CT image denoising [8], with a significant performance gain. Then, more complex networks were proposed to handle the LDCT denoising problem such as the RED-CNN in [9] and the wavelet network in [24]. The wavelet network adopted the shortcut connections introduced by the U-net [34] directly and the RED-CNN [27] replaced the pooling/unpooling layers of U-net with convolution/deconvolution pairs.

Despite the impressive denoising results with these innovative network structures, they fall into a category of an end-to-end network that typically uses the mean squared error (MSE) between the network output and the ground truth as the loss function. As revealed by the recent work [22, 26], this per-pixel MSE is often associated with oversmoothed edges and loss of details. As an algorithm tries to minimize per-pixel MSE, it overlooks subtle image textures/signatures critical for human perception. It is reasonable to assume that CT images distribute over some manifolds. From that point of view, the MSE-based approach tends to take the mean of high-resolution patches using the Euclidean distance rather than the geodesic distance. Therefore, in addition to the blurring effect, artifacts are also possible such as nonuniform biases.

To tackle the above problems, here we propose to use a generative adversarial network (WGAN) [4] with the Wasserstein distance as the discrepancy measure between distributions and a perceptual loss that computes the difference between images in an established feature space [22, 26].

The use of WGAN is to encourage that denoised CT images share the same distribution as that of normal-dose CT (NDCT) images. In the GAN framework, a generative network G and a discriminator network D are coupled tightly and trained simultaneously. While the G network is trained to produce realistic images $G(z)$ from a random vector z, the D network is trained to discriminate between real and generated images [17, 18]. GANs have been used in many applications such as single image super-resolution [22], art creation [7, 48], and image transformation [21]. In the field of medical imaging, Nie et al. [31] proposed to use GAN to estimate CT image from its corresponding MR image. Wolterink et al. [44] are the first to apply

GAN network for cardiac CT image denoising. And Yu et al. [46] used GAN network to handle the de-aliasing problem for fast CS-MRI. Promising results were achieved in these works. We will discuss and compare the results of those two networks in Sect. 14.3 since the proposed network is closely related with their works.

Despite its success in these areas, GANs still suffer from a remarkable difficulty in training [3, 17]. In the original GAN [18], D and G are trained by solving the following minimax problem:

$$\min_{G} \max_{D} L_{\mathrm{GAN}}(D, G) = \mathbb{E}_{\boldsymbol{x} \sim P_r}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim P_z}[\log (1 - D(G(\boldsymbol{z})))], \qquad (14.1)$$

where $\mathbb{E}(\cdot)$ denotes the expectation operator, P_r and P_z are the real data distribution and the noisy data distribution. The generator G transforms a noisy sample to mimic a real sample, which defines a data distribution, denoted by P_g. When D is trained to become an optimal discriminator for a fixed G, the minimization search for G is equivalent to minimizing the Jensen–Shannon (JS) divergence of P_r and P_g, which will lead to vanished gradient on the generator G [3] and G will stop updating as the training continues.

Consequently, Arjovsky et al. [4] proposed to use the *Earth-Mover* (EM) distance or Wasserstein metric between the generated image samples and real data for GAN, which is referred to as WGAN, because the EM distance is continuous and differentiable almost everywhere under some mild assumptions while neither KL nor JS divergence is. After that, an improved WGAN with *gradient penalty* was proposed [19] to accelerate the convergence.

The rationale behind the perceptual loss is twofold. First, when a person compares two images, the perception is not performed pixel-by-pixel. Human vision actually extracts and compares features from images [32]. Therefore, instead of using pixel-wise MSE, we employ another pretrained deep CNN (the famous VGG [36]) for feature extraction and compare the denoised output against the ground truth in terms of the extracted features. Second, from a mathematical point of view, CT images are not uniformly distributed in a high-dimensional Euclidean space. They reside more likely in a low-dimensional manifold. With MSE, we are not measuring the intrinsic similarity between the images, but just their superficial differences in the brute-force Euclidean distance. By comparing images according to their intrinsic structures, we should project them onto a manifold and calculate the geodesic distance instead. Therefore, the use of the perceptual loss for WGAN should facilitate producing results with not only lower noise but also sharper details.

In particular, we treat the LDCT denoising problem as a transformation from LDCT to NDCT images. WGAN provides a good distance estimation between the denoised LDCT and NDCT image distributions. Meanwhile, the VGG-based perceptual loss tends to keep the image content after denoising. The rest of this paper is organized as follows. The proposed method is described in Sect. 14.2. The experiments and results are presented in Sect. 14.3. Finally, relevant issues are discussed and a conclusion is drawn in Sect. 14.4.

14.2 Methods

14.2.1 Noise Reduction Model

Let $\boldsymbol{z} \in \mathbb{R}^{N \times N}$ denote a LDCT image and $\boldsymbol{x} \in \mathbb{R}^{N \times N}$ denote the corresponding NDCT image. The goal of the denoising process is to seek a function G that maps LDCT $\boldsymbol{z}$ to NDCT $\boldsymbol{x}$:

$$G : \boldsymbol{z} \to \boldsymbol{x}. \tag{14.2}$$

On the other hand, we can also take $\boldsymbol{z}$ as a sample from the LDCT image distribution P_L and $\boldsymbol{x}$ from the NDCT distribution or the real distribution P_r. The denoising function G maps samples from P_L into a certain distribution P_g. By varying the function G, we aim to change P_g to make it close to P_r. In this way, we treat the denoising operator as moving one data distribution to another.

Typically, noise in X-ray photon measurements can be simply modeled as the combination of Poisson quantum noise and Gaussian electronic noise. On the contrary, in the reconstructed images, the noise model is usually complicated and nonuniformly distributed across the whole image. Thus, there is no clear clue that indicates how data distributions of NDCT and LDCT images are related to each other, which makes it difficult to denoise LDCT images using traditional methods. However, this uncertainty of noise model can be ignored in deep learning denoising because a deep neural network itself can efficiently learn high-level features and a representation of data distribution from modest-sized image patches through a neural network.

14.2.2 WGAN

Compared to the original GAN network, WGAN uses the Wasserstein distance instead of the JS divergence to compare data distributions. It solves the following minimax problem to obtain both D and G [19]:

$$\begin{aligned}\min_G \max_D L_{\text{WGAN}}(D, G) = & -\mathbb{E}_{\boldsymbol{x}}[D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}}[D(G(\boldsymbol{z}))] \\ & + \lambda \mathbb{E}_{\hat{\boldsymbol{x}}}[(||\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})||_2 - 1)^2],\end{aligned} \tag{14.3}$$

where the first two terms perform a Wasserstein distance estimation, the last term is the gradient penalty term for network regularization, $\hat{\boldsymbol{x}}$ is uniformly sampled along straight lines connecting pairs of generated and real samples, and λ is a constant weighting parameter. Compared to the original GAN, WGAN removes the log function in the losses and also drops the last sigmoid layer in the implementation of the discriminator D. Specifically, the networks D and G are trained alternatively by fixing one and updating the other.

14.2.3 Perceptual Loss

While the WGAN network encourages that the generator transforms the data distribution from high-noise to a low-noise version, another part of the loss function is added for the network to keep image details or information content. Typically, a mean squared error (MSE) loss function is used, which tries to minimize the pixel-wise error between a denoised patch $G(\boldsymbol{z})$ and a NDCT image patch $\boldsymbol{x}$ as [8, 9]

$$L_{\mathrm{MSE}}(G) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})}\left[\frac{1}{N^2}||G\,(\boldsymbol{z}) - \boldsymbol{x}||_F^2\right], \tag{14.4}$$

where $||\cdot||_F$ denotes the Frobenius norm. However, the MSE loss can potentially generate blurry images and cause the distortion or loss of details. Thus, instead of using a MSE measure, we apply a perceptual loss function defined in a feature space

$$L_{\mathrm{Perceptual}}(G) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})}\left[\frac{1}{whd}||\phi(G(\boldsymbol{z})) - \phi(\boldsymbol{x})||_F^2\right], \tag{14.5}$$

where ϕ is a feature extractor, and w, h, and d stand for the width, height, and depth of the feature space, respectively. In our implementation, we adopt the well-known pretrained VGG-19 network [36] as the feature extractor. Since the pretrained VGG network takes color images as input while CT images are in grayscale, we duplicated the CT images to make RGB channels before they are fed into the VGG network. The VGG-19 network contains 16 convolutional layers followed by three fully connected layers. The output of the sixteenth convolutional layer is the feature extracted by the VGG network and used in the perceptual loss function,

$$L_{\mathrm{VGG}}(G) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})}\left[\frac{1}{whd}||VGG(G(\boldsymbol{z})) - VGG(\boldsymbol{x})||_F^2\right]. \tag{14.6}$$

For convenience, we call the perceptual loss computed by VGG network *VGG loss*.

Combining Eqs. (14.3) and (14.6) together, we get the overall joint loss function expressed as

$$\min_G \max_D L_{\mathrm{WGAN}}(D, G) + \lambda_1 L_{\mathrm{VGG}}(G), \tag{14.7}$$

where λ_1 is a weighting parameter to control the trade-off between the WGAN adversarial loss and the VGG perceptual loss.

14.2.4 Network Structures

The overall view of the proposed network structure is shown in Fig. 14.1. For convenience, we name this network WGAN-VGG. It consists of three parts. The first

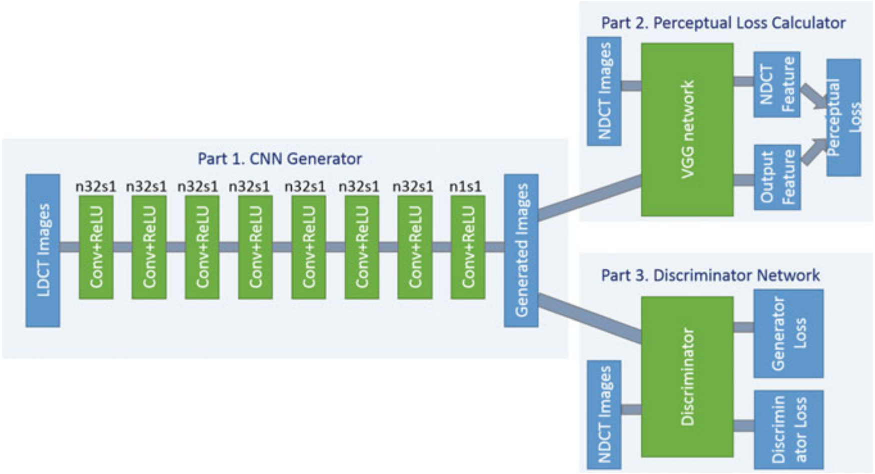


Fig. 14.1 The overall structure of the proposed WGAN-VGG network. In Part 1, n stands for the number of convolutional kernels and s for convolutional stride. So, $n32s1$ means the convolutional layer has 32 kernels with stride 1

part is the generator G, which is a convolutional neural network (CNN) of eight convolutional layers. Following the common practice in the deep learning community [37], small 3×3 kernels were used in each convolutional layer. Due to the stacking structure, such a network can cover a large enough receptive field efficiently. Each of the first seven hidden layers of G has 32 filters. The last layer generates only one feature map with a single 3×3 filter, which is also the output of G. We use rectified linear unit (ReLU) as the activation function.

The second part of the network is the perceptual loss calculator, which is realized by the pretrained VGG network [36]. A denoised output image $G(z)$ from the generator G and the ground-truth image $\boldsymbol{x}$ are fed into the pretrained VGG network for feature extraction. Then, the objective loss is computed using the extracted features from a specified layer according to Eq. (14.6). The reconstruction error is then backpropagated to update the weights of G only, while keeping the VGG parameters intact.

The third part of the network is the discriminator D. As shown in Fig. 14.2, D has six convolutional layers with the structure inspired by others' work [22, 26, 36]. The first two convolutional layers have 64 filters, then followed by two convolutional layers of 128 filters, and the last two convolutional layers have 256 filters. Following the same logic as in G, all the convolutional layers in D have a small 3×3 kernel size. After the six convolutional layers, there are two fully connected layers, of which the first has 1024 outputs and the other has a single output. Following the practice in [4], there is no sigmoid cross-entropy layer at the end of D.

The network is trained using image patches and applied on entire images. The details are provided in Sect. 14.3 on experiments.

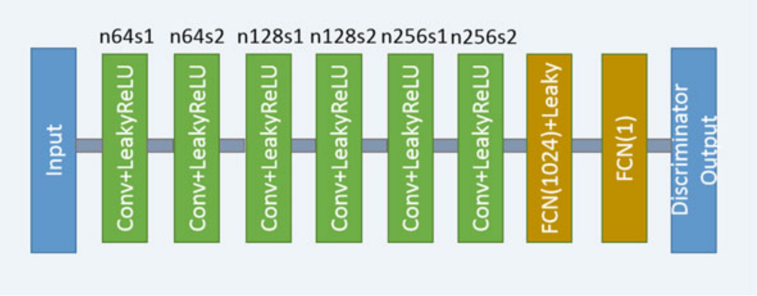


Fig. 14.2 The structure of the discriminator network. *n* and *s* have the same meaning as in Fig. 14.1

Table 14.1 Summary of all trained networks: their loss functions and trainable networks

Network	Loss
CNN-MSE	$\min_G L_{\text{MSE}}(G)$
WGAN-MSE	$\min_G \max_D L_{\text{WGAN}}(G,D) + \lambda_2 L_{\text{MSE}}(G)$
CNN-VGG	$\min_G L_{\text{VGG}}(G)$
WGAN-VGG	$\min_G \max_D L_{\text{WGAN}}(G,D) + \lambda_1 L_{\text{VGG}}(G)$
WGAN	$\min_G \max_D L_{\text{WGAN}}(G,D)$
GAN	$\min_G \max_D L_{\text{GAN}}(G,D)$

14.2.5 Other Networks

For comparison, we also trained four other networks.

- CNN-MSE with only MSE loss,
- CNN-VGG with only VGG loss,
- WGAN-MSE with MSE loss in the WGAN framework,
- WGAN with no other additive losses, and
- Original GAN.

All the trained networks are summarized in Table 14.1.

14.3 Experiments

14.3.1 Experimental Datasets

We used a real clinical dataset authorized for "*the 2016 NIH-AAPM-Mayo Clinic Low Dose CT Grand Challenge*" by Mayo Clinic for the training and evaluation of the proposed networks [1]. The dataset contains 10 anonymous patients' normal-dose abdominal CT images and simulated quarter-dose CT images. In our experiments, we randomly extracted 100,096 pairs of image patches from 4,000 CT images as our training inputs and labels. The patch size is 64 × 64. Also, we extracted 5,056 pairs of patches from another 2,000 images for validation. When choosing the image patches, we excluded image patches that were mostly air. For comparison, we implemented a state-of-the-art 3D dictionary learning reconstruction technique as a representative IR algorithm [45, 47]. The dictionary learning reconstruction was performed from the LDCT projection data provided by Mayo Clinic.

14.3.2 Network Training

In our experiments, all the networks were optimized using Adam algorithm [25]. The optimization procedure for WGAN-VGG network is shown in Fig. 14.3. The mini-batch size was 128. The hyperparameters for Adam were set as $\alpha = 1e-5$, $\beta_1 = 0.5$, $\beta_2 = 0.9$, and we chose $\lambda = 10$ as suggested in [19], $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ according to our experimental experience. The optimization processes for WGAN-MSE and WGAN are similar except that line 12 was changed to the corresponding loss function, and for CNN-MSE and CNN-VGG, lines 2–10 were removed and line 12 was changed according to their loss functions.

The networks were implemented in Python with the Tensorflow library [2]. A NVIDIA Titan XP GPU was used in this study.

14.3.3 Network Convergence

To visualize the convergence of the networks, we calculated the MSE loss and VGG loss over the 5,056 image patches for validation according to Eqs. (14.4) and (14.6) after each epoch. Figure 14.4 shows the averaged MSE and VGG losses, respectively, versus the number of epochs for the five networks. Even though these two loss functions were not used at the same time for a given network, we still want to see how their values change during the training. In the two figures, both the MSE and VGG losses decreased initially, which indicates that the two metrics are positively correlated. However, the loss values of the networks in terms of MSE are increasing in the following order: CNN-MSE<WGAN-MSE<WGAN-VGG<CNN-VGG (Fig. 14.4a), yet the VGG loss is in the opposite order (Fig. 14.4b). The MSE and VGG losses of GAN network are oscillating in the converging process. WGAN-VGG and CNN-VGG have very close VGG loss values, while their MSE losses are quite different. On the other hand, WGAN perturbed the convergence as measured by MSE but smoothly converged in terms of VGG loss. These observations suggest that the two metrics have different focuses when being used by the networks. The difference

```
Require: Set hyper-parameters, $\lambda = 10, \alpha = 1 \times 10^{-5}, \beta_1 = 0.5, \beta_2 = 0.9, \lambda_1 = 0.1, \lambda_2 = 0.1$,
Require: Set the number of total epochs, $N_{epoch} = 100$, the number of iteration for discriminator
    training, $N_D = 4$, the batch size $m = 128$, and image patch size of $80 \times 80$.
Require: Initial discriminator parameters $w_0$, initial generator parameters $\theta_0$
Require: Load VGG-19 network parameters
for $num\_epoch = 0, ..., N_{epoch}$ do
    for $t = 1, ..., N_D$ do
        Sample a batch of NDCT image patches $\{\boldsymbol{x}^{(i)}\}_{i=1}^m$, latent LDCT patches $\{\boldsymbol{z}^{(i)}\}_{i=1}^m$, and
        random numbers $\{\varepsilon^{(i)}\}_{i=1}^m \sim \text{Uniform}[0,1]$
        for $i = 1, ..., m$ do
            $\hat{\boldsymbol{x}}^{(i)} \leftarrow \varepsilon^{(i)}\boldsymbol{x}^{(i)} + (1-\varepsilon^{(i)})G(\boldsymbol{z}^{(i)})$
            $L^{(i)}(D) \leftarrow D(G(\boldsymbol{z}^{(i)})) - D(\boldsymbol{x}^{(i)}) + \lambda(||\nabla D(\hat{\boldsymbol{x}}^{(i)})||_2 - 1)^2$
        end for
    end for
    Update $D$: $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m}\sum_{i=1}^m L^{(i)}(D), w, \alpha, \beta_1, \beta_2)$
    Sample a batch of LDCT patches $\{\boldsymbol{z}^{(i)}\}_{i=1}^m$ and corresponding NDCT patches $\{\boldsymbol{x}^{(i)}\}_{i=1}^m$,
    for $i = 1, ..., m$ do
        $L^{(i)}(G) \leftarrow \lambda_1 L_{\text{VGG}}(\boldsymbol{z}^{(i)}, \boldsymbol{x}^{(i)}) - D(G(\boldsymbol{z}^{(i)}))$
    end for
    Update $G$, $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m}\sum_{i=1}^m L^{(i)}(G), w, \alpha, \beta_1, \beta_2)$
end for
```

Fig. 14.3 Optimization procedure of WGAN-VGG network

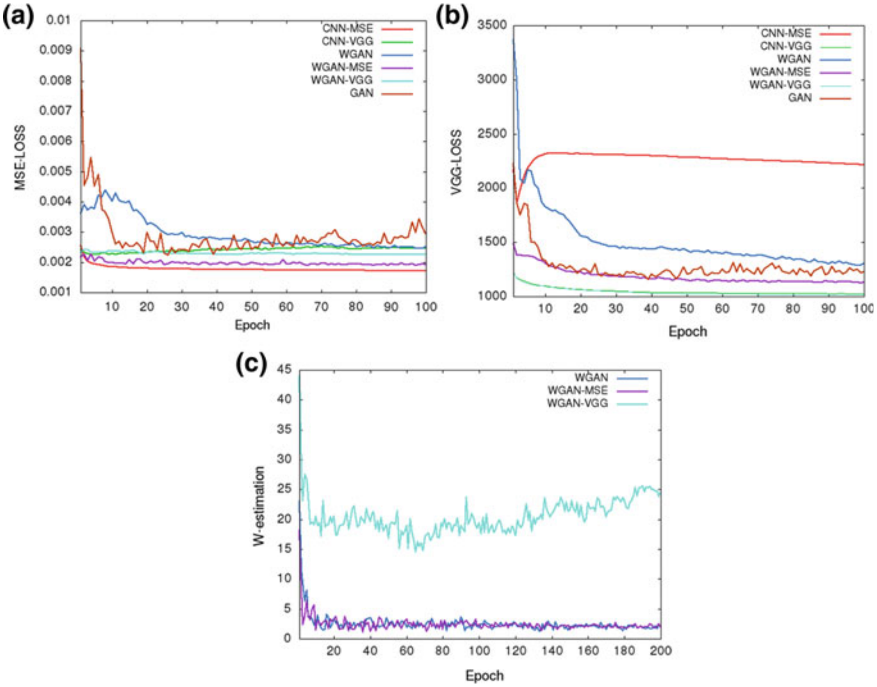


Fig. 14.4 Plots of validation loss versus the number of epochs during the training of the five networks. **a** MSE loss convergence, **b** VGG loss convergence, and **c** Wasserstein estimation convergence

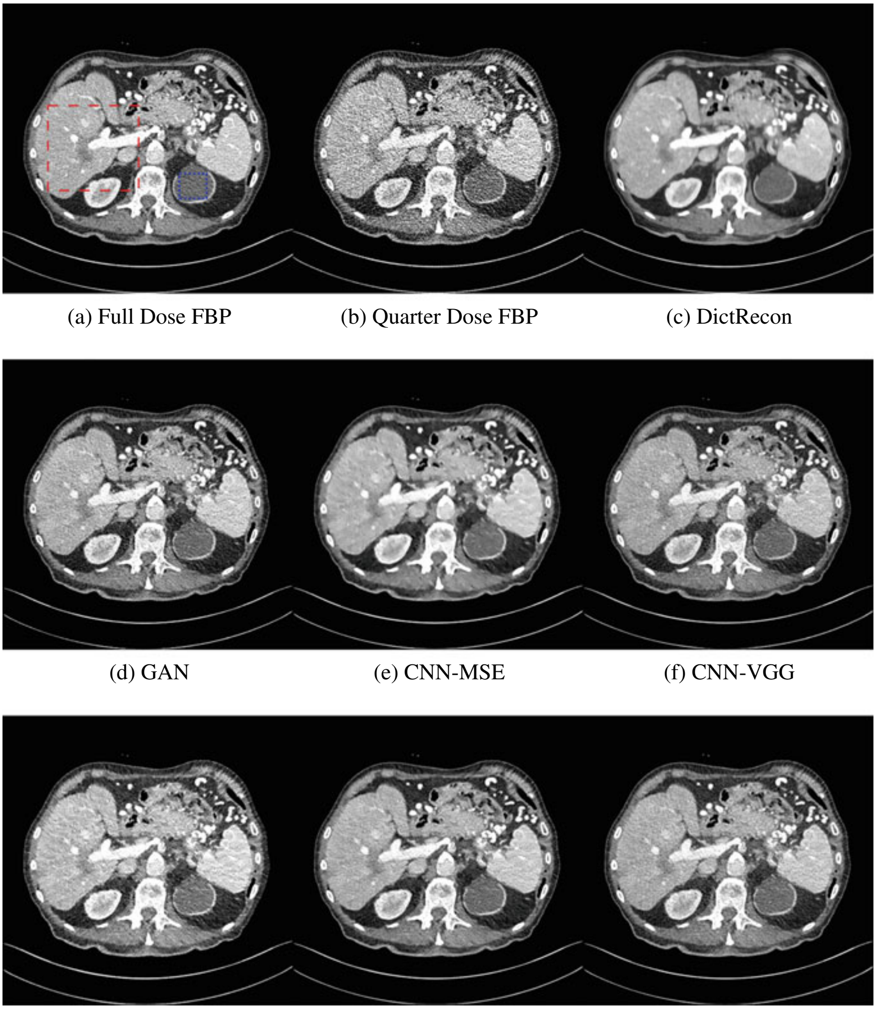
between MSE and VGG losses will be further revealed in the output images of the generators.

In order to show the convergence of WGAN part, we plotted the estimated Wasserstein values defined as $|-\mathbb{E}[D(\boldsymbol{x})]+\mathbb{E}[D(G(\boldsymbol{z}))]|$ in Eq. (14.3). It can be observed in Fig. 14.4c that increasing the number of epochs did reduce the W-distance, although the decay rate becomes smaller. For the WGAN-VGG curve, the introduction of VGG loss has helped to improve the perception/visibility at a cost of a compromised loss measure. For the WGAN and WGAN-MSE curves, we would like to note that what we computed is a surrogate for the W-distance which has not been normalized by the total number of pixels, and if we had done such a normalization the curves would have gone down closely to zero after 100 epochs.

14.3.4 Denoising Results

To show the denoising effect of the selected networks, we took two representative slices as shown in Figs. 14.5 and 14.7. And Figs. 14.6 and 14.8 are the zoomed regions-of-interest (ROIs) marked by the red rectangles in Figs. 14.5 and 14.7. All the networks demonstrated certain denoising capabilities. However, CNN-MSE blurred the images and introduced waxy artifacts as expected, which are easily observed in the zoomed ROIs in Figs. 14.6e and 14.8e. WGAN-MSE was able to improve the result of CNN-MSE by avoiding oversmooth but minor streak artifacts can still be observed especially compared to CNN-VGG and WGAN-VGG. Meanwhile, using WGAN or GAN alone generated stronger noise (Figs. 14.6g and 14.8g) than the other networks enhanced a few white structures in the WGAN/GAN generated images, which are originated from the low-dose streak artifact in LDCT images, while on the contrary the CNN-VGG and WGAN-VGG images are visually more similar to the NDCT images. This is because the VGG loss used in CNN-VGG and WGAN-VGG is computed in a feature space that is trained previously on a very large natural image dataset [13]. By using VGG loss, we transferred the knowledge of human perception that is embedded in VGG network to CT image quality evaluation. The performance of using WGAN or GAN alone is not acceptable because it only maps the data distribution from LDCT to NDCT but does not guarantee the image content correspondence. As for the lesion detection in these two slices, all the networks enhance the lesion visibility compared to the original noisy low-dose FBP images as noise is reduced by different approaches.

As for iterative reconstruction technique, the reconstruction results depend greatly on the choices of the regularization parameters. The implemented dictionary learning reconstruction (DictRecon) result gave the most aggressive noise reduction effect compared to the network outputs as a result of strong regularization. However, it oversmoothed some fine structures. For example, in Fig. 14.8, the vessel pointed by the green arrow was smeared out while it is easily identifiable in NDCT as well as WGAN-VGG images. Yet, as an iterative reconstruction method, DictRecon has its advantage over post-processing method. As pointed by the red arrow in Fig. 14.8,



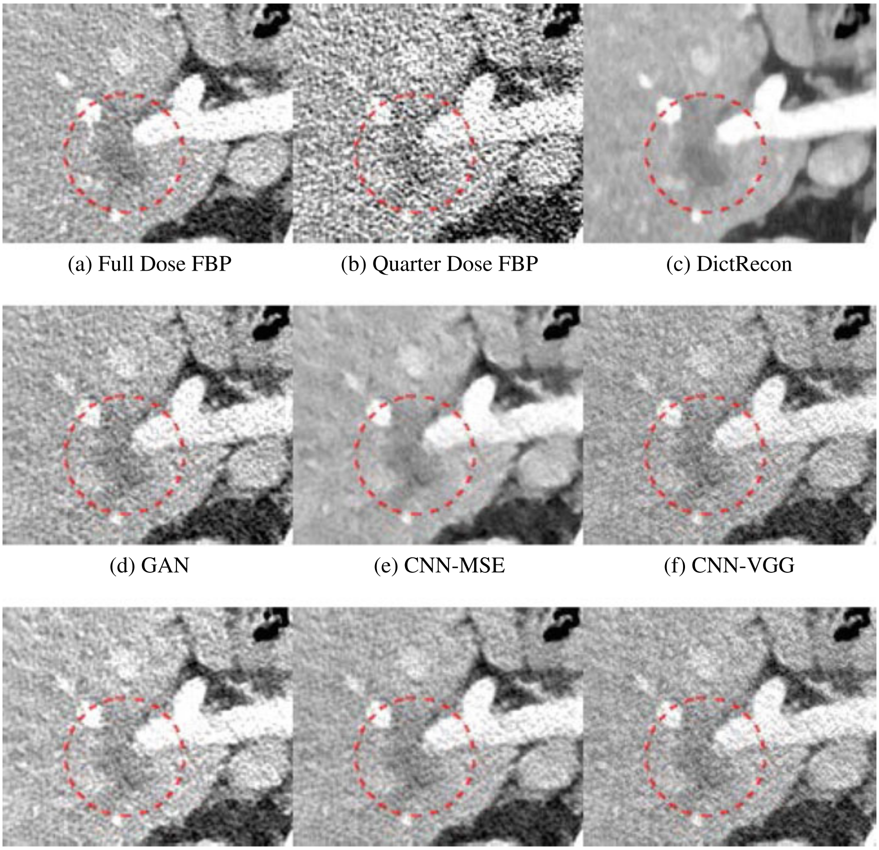
(a) Full Dose FBP (b) Quarter Dose FBP (c) DictRecon

(d) GAN (e) CNN-MSE (f) CNN-VGG

(g) WGAN (h) WGAN-MSE (i) WGAN-VGG

Fig. 14.5 Transverse CT images of the abdomen demonstrate a low attenuation liver lesion (in the red box) and a cystic lesion in the upper pole of the left kidney (in the blue box) This display window is [–160, 240]HU

there is a bright spot which can be seen in DictRecon and NDCT images, but is not observable in LDCT and network processed images. Since the WGAN-VGG image is generated from LDCT image, in which this bright spot is not easily observed, it is reasonable that we do not see the bright spot in the images processed by neural networks. In other words, we do not want the network to generate structure that does not exist in the original images. In short, the proposed WGAN-VGG network is a post-processing method and information that is lost during the FBP reconstruction cannot


(a) Full Dose FBP (b) Quarter Dose FBP (c) DictRecon

(d) GAN (e) CNN-MSE (f) CNN-VGG

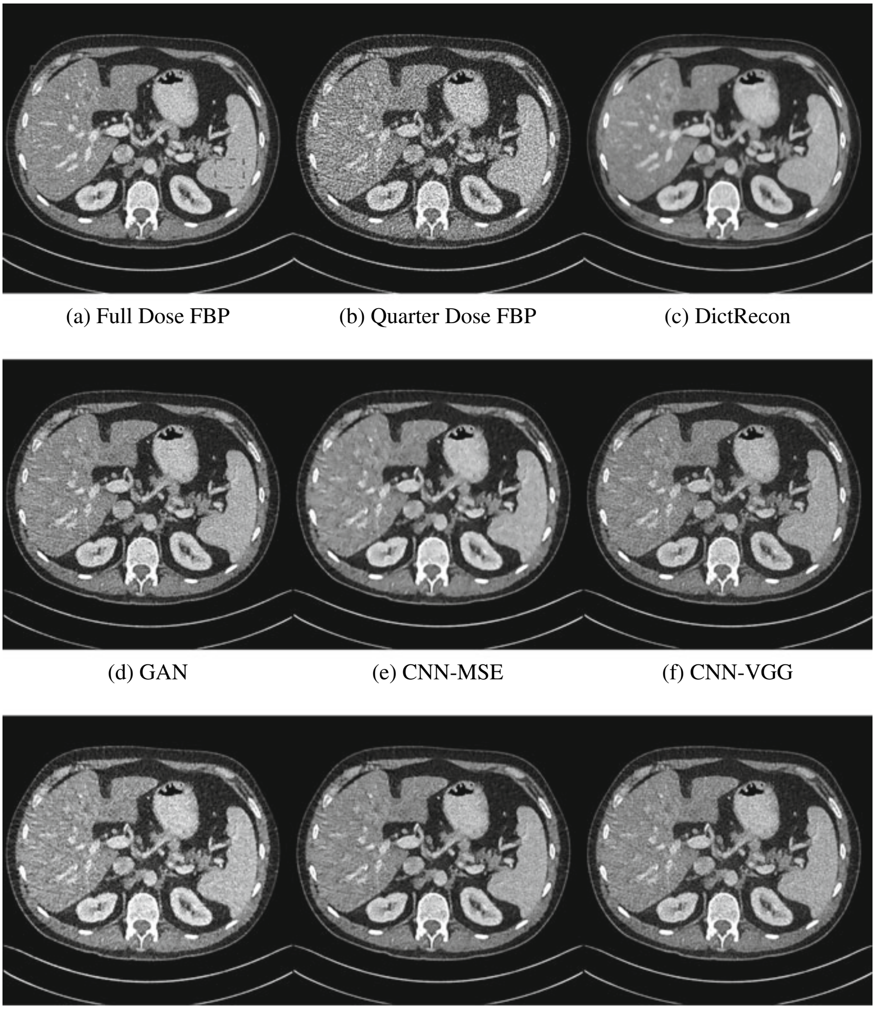
(g) WGAN (h) WGAN-MSE (i) WGAN-VGG

Fig. 14.6 Zoomed ROI of the red rectangle in Fig. 14.5. The low attenuation liver lesion within the dashed circle represents metastasis. The lesion is difficult to assess on quarter-dose FBP recon (**b**) due to high-noise content. This display window is [–160, 240]HU

easily be recovered, which is one limitation for all the post-processing methods. On the other hand, as an iterative reconstruction method, DictRecon algorithm generates images from raw data, which has more information than the post-processing methods.

14.3.5 Quantitative Analysis

For quantitative analysis, we calculated the peak-to-noise ratio (PSNR) and structural similarity (SSIM). The summary data are in Table 14.2. CNN-MSE ranks the first in terms of PSNR, while WGAN is the worst. Since PSNR is equivalent to the per-pixel loss, it is not surprising that CNN-MSE, which was trained to minimize MSE loss,



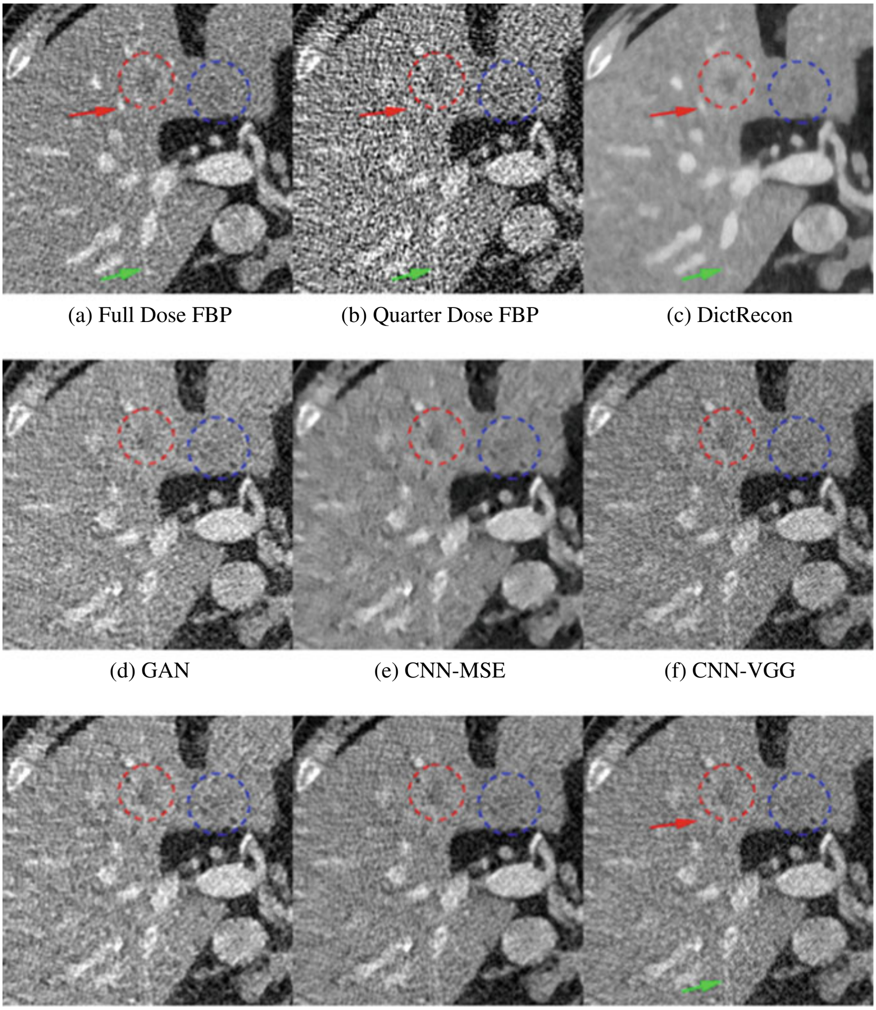
(a) Full Dose FBP (b) Quarter Dose FBP (c) DictRecon

(d) GAN (e) CNN-MSE (f) CNN-VGG

(g) WGAN (h) WGAN-MSE (i) WGAN-VGG

Fig. 14.7 Transverse CT images of the abdomen demonstrate small low attenuation liver lesions. The display window is [–160, 240]HU

outperformed the networks trained to minimize other feature-based loss. It is worth noting that these quantitative results are in decent agreement with Fig. 14.4, in which CNN-MSE has the smallest MSE loss and WGAN has the largest. The reason why WGAN ranks the worst in PSNR and SSIM is because it does not include either MSE or VGG regularization. DictRecon achieves the best SSIM and a high PSNR. However, it has the problem of image blurring and leads to blocky and waxy artifacts


(a) Full Dose FBP (b) Quarter Dose FBP (c) DictRecon

(d) GAN (e) CNN-MSE (f) CNN-VGG

(g) WGAN (h) WGAN-MSE (i) WGAN-VGG

Fig. 14.8 Zoomed ROI of the red rectangle in Fig. 14.7 demonstrates the two attenuation liver lesions in the red and blue circles. The display window is [–160, 240]HU

in the resultant images. This indicates that PSNR and SSIM may not be sufficient in evaluating image quality.

In the reviewing process, we found two papers using similar network structures. In [44], Wolterink et al. trained three networks, i.e., GAN, CNN-MSE, and GAN-MSE for cardiac CT denoising. Their quantitative PSNR results are consistent with our counterpart results. And Yu et al. [46] used GAN-VGG to handle the de-aliasing problem for fast CS-MRI. Their results are also consistent with ours. Interestingly, despite the high PSNRs obtained by MSE-based networks, the authors in the two

Table 14.2 Quantitative results associated with different network outputs for Figs. 14.5 and 14.7

	Figure 14.5		Figure 14.7	
	PSNR	SSIM	PSNR	SSIM
LDCT	19.7904	0.7496	18.4519	0.6471
CNN-MSE	24.4894	0.7966	23.2649	0.7022
WGAN-MSE	24.0637	0.8090	22.7255	0.7122
CNN-VGG	23.2322	0.7926	22.0950	0.6972
WGAN-VGG	23.3942	0.7923	22.1620	0.6949
WGAN	22.0168	0.7745	20.9051	0.6759
‘1 GAN	21.8676	0.7581	21.0042	0.6632
DictRecon	24.2516	0.8148	24.0992	0.7631

papers all claim that GAN and VGG loss based networks have better image quality and diagnostic information.

To gain more insight into the output images from different approaches, we inspect the statistical properties by calculating the mean CT numbers (Hounsfield Units) and standard deviations (SDs) of two flat regions in Figs. 14.5 and 14.7 (marked by the blue rectangles). In an ideal scenario, a noise reduction algorithm should achieve mean and SD to the gold standard as close as possible. In our experiments, the NDCT FBP images were used as gold standard because they have the best image quality in this dataset. As shown in Table 14.3, both CNN-MSE and DictRecon produced much smaller SDs compared to NDCT, which indicates they oversmoothed the images and supports our visual observation. On the contrary, WGAN produced the closest SDs yet smaller mean values, which means it can reduce noise to the same level as NDCT but it compromised the information content. On the other hand, the proposed WGAN-VGG has outperformed CNN-VGG, WGAN-MSE, and other selected methods in terms of mean CT numbers, SDs, and most importantly visual impression.

In addition, we performed a blind reader study on 10 groups of images. Each group contains the same image slice but processed by different methods. NDCT and LDCT images are also included for reference, which are the only two labeled images in each group. Two radiologists were asked to independently score each image in terms of noise suppression and artifact reduction on a five-point scale (1 = unacceptable and 5 = excellent), except for the NDCT and LDCT images, which are the references. In addition, they were asked to give an overall image quality score for all the images. The mean and standard deviation values of the scores from the two radiologists were then obtained as the final evaluation results, which are shown in Table 14.4. It can be seen that CNN-MSE and DictRecon give the best noise suppression scores while the proposed WGAN-VGG outperforms the other methods for artifact reduction and overall quality improvement. Also, *-VGG networks provide higher scores than *-MSE networks in terms of artifact reduction and overall quality but lower scores for noise suppression. This indicates that MSE loss based networks are good at noise suppression at a loss of image details, resulting in an image quality degradation for

Table 14.3 Statistical properties of the blue rectangle areas in Figs. 14.5 and 14.7. The values are in Hounsfield Unit (HU)

	Figure 14.5		Figure 14.7	
	Mean	SD	Mean	SD
NDCT	9	36	118	38
LDCT	11	74	118	66
CNN-MSE	12	18	120	15
WGAN-MSE	9	28	115	25
CNN-VGG	4	30	104	28
WGAN-VGG	9	31	111	29
WGAN	23	37	135	33
GAN	8	35	110	32
DictRecon	4	11	111	13

Table 14.4 Subjective quality scores (mean ± sd) for different algorithms

	NDCT	LDCT	CNN-MSE	CNN-VGG	WGAN-MSE	WGAN-VGG	WGAN	GAN	DictRecon
Noise suppression	–	–	4.35 ± 0.24	3.10 ± 0.23	3.55 ± 0.25	3.20 ± 0.25	2.90 ± 0.26	3.00 ± 0.21	**4.65 ± 0.20**
Artifact reduction	–	–	1.70 ± 0.28	2.85 ± 0.32	3.05 ± 0.27	**3.45 ± 0.25**	2.90 ± 0.28	3.05 ± 0.27	2.05 ± 0.27
Overall quality	3.95 ± 0.20	1.35 ± 0.16	2.15 ± 0.25	3.05 ± 0.20	3.30 ± 0.21	**3.70 ± 0.15**	3.05 ± 0.22	3.10 ± 0.21	2.05 ± 0.36

diagnosis. Meanwhile, the networks using WGAN give better overall image quality than the networks using CNN, which supports the use of WGAN for CT image denoising.

14.4 Discussions and Conclusion

The most important motivation for this paper is to approach the gold standard NDCT images as much as possible. As described above, the feasibility and merits of GAN have been investigated for this purpose with the Wasserstein distance and the VGG loss. The difference between using the MSE and VGG losses is rather significant. Despite the fact that networks with MSE would offer higher values for traditional figures of merit, VGG loss based networks seem desirable for better visual image quality with more details and less artifacts.

The experimental results have demonstrated that using WGAN helps to improve image quality and statistical properties. Comparing the images of CNN-MSE and

WGAN-MSE, we can see that the WGAN framework helped to avoid oversmoothing effect typically suffered by MSE-based image generators. Although CNN-VGG and WGAN-VGG visually share a similar result, the quantitative analysis shows WGAN-VGG enjoys higher PSNRs and more faithful statistical properties of denoised images relative to those of NDCT images. However, using WGAN/GAN alone reduced noise but at the expense of losing critical features. The resultant images do not show a strong noise reduction. Quantitatively, the associated PSNR and SSIM increased modestly compared to LDCT but they are much lower than what the other networks produced. Theoretically, WGAN/GAN network is based on generative model and may generate images that look natural yet cause a severe distortion for medical diagnostics. This is why an additive loss function such as MSE and VGG loss should be added to guarantee the image content remains the same.

It should be noted that the experimental data contain only one noise setting. Networks should be retrained or retuned for different data to adapt for different noise properties. Especially, networks with WGAN are trying to minimize the distance between two probability distributions. Thus, their trained parameters have to be adjusted for new datasets. Meanwhile, since the loss function of WGAN-VGG is a mixture of feature domain distance and the GAN adversarial loss, they should be carefully balanced for different datasets to reduce the amount of image content alternation.

The denoising network is a typical end-to-end operation, in which the input is an LDCT image while the target is an NDCT image. Although we have generated images visually similar to NDCT counterparts in the WGAN-VGG network, we recognize that these generated images are still not as good as NDCT images. Moreover, noise still exists in NDCT images. Thus, it is possible that VGG network has captured these noise features and kept them in the denoised images. This could be a common problem for all the denoising networks. How to outperform the so-called gold standard NDCT images is an interesting open question. Moreover, image post-denoising methods also suffer from the information loss during the FBP reconstruction process. These phenomena are observed in comparison with DictRecon result. A better way to incorporate the strong fitting capability of neural network and the data completeness of CT data is to design a network that maps directly from raw projection to the final CT images, which could be the next step of our work.

In conclusion, we have proposed a contemporary deep neural network that uses a WGAN framework with perceptual loss function for LDCT image denoising. Instead of focusing on the design of a complex network structure, we have dedicated our effort to combine synergistic loss functions that guide the denoising process so that the resultant denoised results are as close to the gold standard as possible. Our experiment results with real clinical images have shown that the proposed WGAN-VGG network can effectively solve the well-known oversmoothing problem and generate images with reduced noise and increased contrast for improved lesion detection. In the future, we plan to incorporate the WGAN-VGG network with more complicated generators such as the networks reported in [9, 24] and extend these networks for image reconstruction from raw data by making a neural network counterpart of the FBP process.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61671312, and in part by the National Institute of Biomedical Imaging and Bioengineering/National Institutes of Health under Grant R01 EB016977 and Grant U01 EB017140. The authors would also like to thank NVIDIA Corporation for the donation of the Titan Xp GPU used for this research.

References

1. AAPM (2017) Low dose CT grand challenge. http://www.aapm.org/GrandChallenge/LowDoseCT/
2. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M et al (2016) Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467
3. Arjovsky M, Bottou L (2017) Towards principled methods for training generative adversarial networks. In: NIPS 2016 workshop on adversarial training. In review for ICLR, vol 2016
4. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN. arXiv:1701.07875
5. Beister M, Kolditz D, Kalender WA (2012) Iterative reconstruction methods in x-ray CT. Phys Med: Eur J Med Phys 28(2):94–108
6. Brenner DJ, Hall EJ (2007) Computed tomography — an increasing source of radiation exposure. N Engl J Med 357(22):2277–2284
7. Brock A, Lim T, Ritchie JM, Weston N (2016) Neural photo editing with introspective adversarial networks. arXiv:1609.07093
8. Chen H, Zhang Y, Zhang W, Liao P, Li K, Zhou J, Wang G (2016) Low-dose CT denoising with convolutional neural network. arXiv:1610.00321
9. Chen H, Zhang Y, Kalra MK, Lin F, Chen Y, Liao P, Zhou J, Wang G (2017) Low-dose CT with a residual encoder-decoder convolutional neural network. IEEE Trans Med Imaging 36(12):2524–2535
10. Chen Y, Yin X, Shi L, Shu H, Luo L, Coatrieux JL, Toumoulin C (2013) Improving abdomen tumor low-dose CT images using a fast dictionary learning based processing. Phys Med Biol 58(16):5803
11. De Gonzalez AB, Darby S (2004) Risk of cancer from diagnostic x-rays: estimates for the UK and 14 other countries. Lancet 363(9406):345–351
12. De Man B, Basu S (2004) Distance-driven projection and backprojection in three dimensions. Phys Med Biol 49(11):2463
13. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
14. Dong C, Loy CC, He K, Tang X (2016) Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell 38(2):295–307
15. Elbakri IA, Fessler JA (2002) Statistical image reconstruction for polyenergetic x-ray computed tomography. IEEE Trans Med Imaging 21(2):89–99
16. Feruglio PF, Vinegoni C, Gros J, Sbarbati A, Weissleder R (2010) Block matching 3D random noise filtering for absorption optical projection tomography. Phys Med Biol 55(18):5401
17. Goodfellow I (2017) NIPS 2016 tutorial: generative adversarial networks. arXiv:1701.00160
18. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems, pp 2672–2680
19. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A (2017) Improved training of Wasserstein GANs. arXiv:1704.00028
20. Hara AK, Paden RG, Silva AC, Kujak JL, Lawder HJ, Pavlicek W (2009) Iterative reconstruction technique for reducing body radiation dose at CT: feasibility study. Am J Roentgenol 193(3):764–771

21. Isola P, Zhu JY, Zhou T, Efros AA (2016) Image-to-image translation with conditional adversarial networks. arXiv:1611.07004
22. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. arXiv:1603.08155
23. Kang D, Slomka P, Nakazato R, Woo J, Berman DS, Kuo CCJ, Dey D (2013) Image denoising of low-radiation dose coronary CT angiography by an adaptive block-matching 3D algorithm. In: SPIE medical imaging, international society for optics and photonics, pp 86,692G–86,692G
24. Kang E, Min J, Ye JC (2016) A deep convolutional neural network using directional wavelets for low-dose x-ray CT reconstruction. arXiv:1610.09736
25. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
26. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2016) Photo-realistic single image super-resolution using a generative adversarial network. arXiv:1609.04802
27. Lewitt RM (1990) Multidimensional digital image representations using generalized Kaiser–Bessel window functions. J Opt Soc Am A 7(10):1834–1846
28. Liu Y, Ma J, Fan Y, Liang Z (2012) Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction. Phys Med Biol 57(23):7923
29. Ma J, Huang J, Feng Q, Zhang H, Lu H, Liang Z, Chen W (2011) Low-dose computed tomography image restoration using previous normal-dose scan. Med Phys 38(10):5713–5731
30. Manduca A, Yu L, Trzasko JD, Khaylova N, Kofler JM, McCollough CM, Fletcher JG (2009) Projection space denoising with bilateral filtering and CT noise modeling for dose reduction in CT. Med Phys 36(11):4911–4919
31. Nie D, Trullo R, Petitjean C, Ruan S, Shen D (2016) Medical image synthesis with context-aware generative adversarial networks. arXiv:1612.05362
32. Nixon M, Aguado AS (2008) Feature extraction and image process, 2nd edn. Academic, New York
33. Ramani S, Fessler JA (2012) A splitting-based iterative algorithm for accelerated statistical x-ray CT reconstruction. IEEE Trans Med Imaging 31(3):677–688
34. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Proceedings of the international conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
35. Sidky EY, Pan X (2008) Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. Phys Med Biol 53(17):4777
36. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
37. Srinivas S, Sarvadevabhatla RK, Mopuri KR, Prabhu N, Kruthiventi SS, Babu RV (2016) A taxonomy of deep convolutional neural nets for computer vision. Front Robot AI 2:36
38. Tian Z, Jia X, Yuan K, Pan T, Jiang SB (2011) Low-dose CT reconstruction via edge-preserving total variation regularization. Phys Med Biol 56(18):5949
39. Wang G (2016) A perspective on deep imaging. IEEE Access 4:8914–8924
40. Wang G, Kalra M, Orton CG (2017) Machine learning will transform radiology significantly within the next 5 years. Med Phys 44(6):2041–2044
41. Wang J, Lu H, Li T, Liang Z (2005) Sinogram noise reduction for low-dose CT by statistics-based nonlinear filters. In: Medical imaging 2005: image processing, international society for optics and photonics, vol 5747, pp 2058–2067
42. Wang J, Li T, Lu H, Liang Z (2006) Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose x-ray computed tomography. IEEE Trans Med Imaging 25(10):1272–1283
43. Whiting BR, Massoumzadeh P, Earl OA, O'Sullivan JA, Snyder DL, Williamson JF (2006) Properties of preprocessed sinogram data in x-ray computed tomography. Med Phys 33(9):3290–3303
44. Wolterink JM, Leiner T, Viergever MA, Isgum I (2017) Generative adversarial networks for noise reduction in low-dose CT. IEEE Trans Med Imaging

45. Xu Q, Yu H, Mou X, Zhang L, Hsieh J, Wang G (2012) Low-dose x-ray CT reconstruction via dictionary learning. IEEE Trans Med Imaging 31(9):1682–1697
46. Yu S, Dong H, Yang G, Slabaugh G, Dragotti PL, Ye X, Liu F, Arridge S, Keegan J, Firmin D et al (2017) Deep de-aliasing for fast compressive sensing MRI. arXiv:1705.07137
47. Zhang Y, Mou X, Wang G, Yu H (2017) Tensor-based dictionary learning for spectral CT reconstruction. IEEE Trans Med Imaging 36(1):142–154
48. Zhu JY, Krähenbühl P, Shechtman E, Efros AA (2016) Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision. Springer, pp 597–613

Chapter 15
Image Quality Assessment for Population Cardiac Magnetic Resonance Imaging

Le Zhang, Marco Pereañez, Stefan K. Piechnik, Stefan Neubauer, Steffen E. Petersen and Alejandro F. Frangi

Abstract Cardiac magnetic resonance (CMR) images play a growing role in diagnostic imaging of cardiovascular diseases. MRI is arguably the most comprehensive imaging modality for noninvasive and nonionizing imaging of the heart and great vessels and, hence, most suited for population imaging cohorts. Ensuring full coverage of the left ventricle (LV) is a basic criterion of CMR image quality. Complete LV coverage, from base to apex, precedes accurate cardiac volume and functional assessment. Incomplete coverage of the LV is identified through visual inspection, which is time-consuming and usually done retrospectively in large imaging cohorts. In this chapter, we propose a novel automatic method to check the coverage of LV from CMR images by using Fisher discriminative and dataset invariance (FDDI) three-dimensional (3D) convolutional neural networks (CNN) independently of image-acquisition parameters, such as imaging device, magnetic field strength, variations in protocol execution, etc. The proposed model is trained on multiple cohorts of different provenance to

L. Zhang (✉)
Centre for Computational Imaging and Simulation Technologies in Biomedicine (CISTIB), University of Sheffield, Sheffield, UK
e-mail: le.zhang@sheffield.ac.uk

M. Pereañez · A. F. Frangi
Centre for Computational Imaging and Simulation Technologies in Biomedicine (CISTIB), University of Leeds, Leeds, UK
e-mail: m.pereanez@leeds.ac.uk

A. F. Frangi
e-mail: a.frangi@leeds.ac.uk

S. K. Piechnik · S. Neubauer
Division of Cardiovascular Medicine, Oxford Center for Clinical Magnetic Resonance Research (OCMR), University of Oxford, John Radcliffe Hospital, Oxford, UK
e-mail: stefan.piechnik@cardiov.ox.ac.uk

S. Neubauer
e-mail: stefan.neubauer@cardiov.ox.ac.uk

S. E. Petersen
Cardiovascular Medicine at the William Harvey Research Institute, Queen Mary University of London and Barts Heart Center, Barts Health NHS Trust, London, UK
e-mail: s.e.petersen@qmul.ac.uk

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_15

learn the appearance and identify missing basal and apical slices. To address this, a two-stage framework is proposed. First, the FDDI 3D CNN extracts high-level features in the common representation from different CMR datasets using adversarial approach; then these image features are used to detect missing basal and apical slices. Compared with the traditional 3D CNN strategy, the proposed FDDI 3D CNN can minimize the within-class scatter and maximize the between-class scatter, which can be adapted to other CMR image data for LV coverage assessment.

15.1 Introduction

Left Ventricular (LV) cardiac anatomy and function are widely used for diagnosis and monitoring disease progression in cardiology and to assess the patient's response to cardiac surgery and interventional procedures. Cardiac ultrasound (US) and cardiac magnetic resonance (CMR) imaging are arguably the most widespread techniques for clinical diagnostic imaging of the heart. For population imaging studies, however, CMR remains the modality of choice and provides a one-stop-shop access to cardiac anatomy and function noninvasively [24]. The quantification of LV anatomy and function from large population imaging studies or patient cohorts from large clinical trials requires automatic image quality assessment and image analysis tools. A basic criterion for cardiac image quality is LV coverage and detection of missing apical and basal CMR slices [15]. Due to insufficient radiographer's experience in planning a scan, natural cardiac muscle contraction, breathing motion, and imperfect triggering, CMR can display incomplete LV coverage, which hampers quantitative LV characterization and diagnostic accuracy [26]. For example, missing basal slices has an important impact on LV volume calculation and several derived LV functional measures like ejection fraction and cardiac output. Even if scout images are acquired to center the LV in view and minimize this problem, incomplete coverage can result at any point throughout the cardiac cycle due to patient breathing and cardiac motion. Automatic quality assessment is important in large-scale population imaging studies, where data is acquired across different imaging sites, by staff of variable experience and without systematic checks by experienced physicians for quality before images get stored for future analysis. In addition, data are acquired from subjects with a diverse constitution and, with strict time constraints on scanner availability [5, 31]. Image quality assessment is traditionally performed to radiographers who assure that patients do not leave the scanner without diagnostically interpretable data. However, there are limits to their human attention and, with CMR examinations becoming cheaper and increasingly commissioned, some centers may reach inadequate scanning loads to maintain consistent standards. Quality assessment is of particular importance in large-scale population imaging studies, where data is acquired across different imaging sites before core lab analysis, where large volumes of data may be stored unchecked by experienced staff before actual analysis is attempted [5, 31]. Automatic methods for these repetitive quality assurance tasks provide the required consistency and reliability.

Few guidelines exist, clinical or otherwise, that objectively establish what constitutes a good medical image and a good CMR study [8]. To ensure consistent quantification of CMR data, automatic assessment of complete LV coverage is the first step. LV coverage is still assessed by visual inspection of CMR image sequences, which is subjective, repetitive, error-prone, and time-consuming [1]. Automatic coverage assessment is required to promptly intervene and correct data acquisition, and/or discard images with incomplete LV coverage whose analysis would otherwise impair any aggregated statistics over the cohort. The most common causes of incomplete LV coverage are the lack of basal slices (no atrial chamber visible in end-systole, hence no certainty that the base of the heart is covered completely), and the lack of the apical slice (LV cavity still visible at end-systole) [15]. Although technological developments in MRI hardware and pulse sequences have led over the years to faster CMR acquisitions, full heart imaging and motion compensation remain challenging. In the UK Biobank's CMR protocol, for instance, incomplete heart coverage is the reason for flagging 4% of all CMR examinations as unreliable or non-analyzable image data [2]. While 4% may seem to be a small proportion, the challenge is to automatically sift through the entire database to identify and exclude those cases from further quantitative analysis. Methods for objective detection of basal and apical imaging planes are relevant, as their absence affects diagnostic accuracy, as well as anatomical and functional LV quantification.

In the field of video processing, automatic image quality assessment (AIQA) is a well-developed corpus of techniques usually concerned with detecting image distortions characteristic of multimedia communications [27, 32]. These distortions are generally very different to those affecting medical imagery. No-reference-based image quality assessment (NR-IQA) [10, 22] is relevant for medical imaging data since while it is easy to get access to abundant data sets of mixed quality, it is infeasible to collect data without some level of image degradation or artifacts. Usually, in practical CMR image processing applications, there are no perfect versions of the incomplete LV coverage images, and only the image to be assessed is available to us. While the assessment tries to highlight the differences of our assessed dataset regarding a hypothetical high-quality image [14]. The final image quality is estimated solely based upon characteristics of the assessed image.

In medical image analysis, it is sometimes convenient or necessary to infer an image in one modality from another for image quality assessment purposes. One major challenge of slice pose estimation for CMR comes from differences between data sources, which are tissue appearance and/or spatial resolution of images sourced from different physical acquisition principles or parameters. Such differences make it difficult to generalize algorithms trained on specific datasets to other data sources. This is problematic not only when the source and target datasets are different, but more so, when the target dataset contains no labels. In all such scenarios, it is highly desirable to learn a discriminative classifier or other predictor in the presence of a shift between training and test distributions, which is called *dataset invariance*. The general approach of achieving dataset adaptation has been explored under many facets. Among the existing cross-dataset learning works, dataset adaptation has been adopted for reidentification hoping labeled data from a source dataset can provide transferable

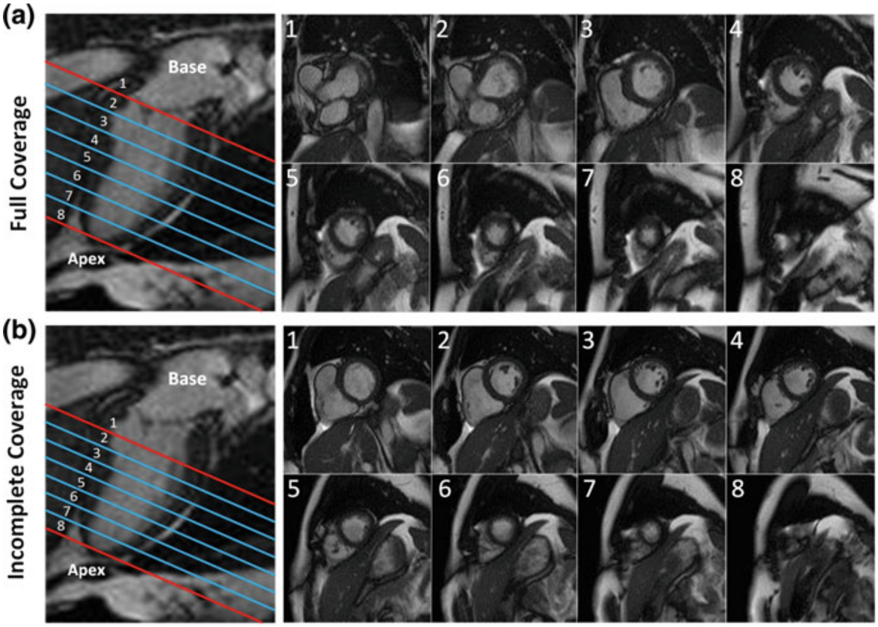


Fig. 15.1 *Top*: a typical two-chamber view cardiac MRI with eight slices fully covered from base to apex and SAX view volume with whole coverage (slice 1 is the basal slice); *bottom*: a typical two-chamber view cardiac MRI with eight slices incompletely covered from base to apex and SAX view volume with missing basal slice (slice 1 is not the basal slice). In each rectangle, from top to bottom, rows correspond to adjacent axial slices

identity-discriminative information for a target dataset. Reference [11] explored the possibility of generating multimodal images from single-modality imagery. References [18, 20] employed multitask metric learning models to benefit the target task. However, these works are focused mainly on linear assumptions.

In this chapter, we focus on the analysis of short axis (SA) cine MRI although the technique could be generalized to long axis images too. We aim to identify missing apical slices (MAS) and/or basal slices (MBS) in 3D cardiac MRI volumes (Fig. 15.1). In previous work, we used a 2D CNN constructed on single-slice images and processed them sequentially [34]. However, this solution ignores contextual information contained across slices providing inferior performance compared to a 3D analysis. We assume that 3D CNNs can easily and effectively deal with within-class variability and between-class similarity, which are important sources of detection error [3]. We seek to learn a feature representation that achieves reliable classification results even with small numbers of training data or iterations. Meanwhile, to deal with the problem where there is no labeled data for a target dataset, one hopes to transfer knowledge from a model trained on sufficient labeled data of a source dataset sharing the same feature space, but with a different marginal distribution, we present a dataset invariance model for any cross-dataset basal/apical slice estimation problem in CMR volumes. To solve these problems, we address incomplete LV coverage detection using a Fisher discriminative and dataset invariance (FDDI) 3D CNN, which utilizes 3D convolution kernels and exploits the spatial contextual information in volumetric data and integrate adversarial feature learning by building an end-to-end architecture of CNNs and transferring nonlinear representations from a labeled source dataset to a target dataset where labels are nonexistent. The proposed FDDI 3D CNN uses the Fisher discriminant criterion [33] on the fully connected layer to make the features discriminative and insensitive to geometric structural variations.

To the best of our knowledge, this is the first work tackling the problem of automatic detection of missing apical and basal slices in CMR and evaluating the proposed approach on a very extensive and challenging population imaging dataset. Besides introducing a novel FDDI 3D CNN architecture, we propose an effective cascaded detection strategy for incomplete coverage identification. The first stage classifies the image representations. We train two separate FDDI 3D CNN classifiers to detect the absence of basal and apical slices. The second stage is the quality verification. We combine the classification results from stage 1 to determine the kind of incomplete coverage found on the image.

15.2 Full LV Coverage Detection Method

15.2.1 Problem Formulation

We formulate our problem as two tasks:

(1) *Datasets invariance*: given a set of 3D images $\mathscr{V}^s = [\mathbf{V}_1^s, \dots, \mathbf{V}_N^s] \in \mathbb{R}^{m\times n\times z^s\times N^s}$ of modality $\mathscr{M}_s$ in the source dataset, and $\mathscr{V}^t = [\mathbf{V}_1^t, \dots, \mathbf{V}_N^t] \in \mathbb{R}^{m\times n\times z^t\times N^t}$ of modality $\mathscr{M}_t$ in the target dataset. m, n are the dimensions of axial view of the image, and z^s and z^t denotes the size of image along the z-axis, while N^s and N^t are the number of elements in source and target datasets, respectively. Our goal is to build mappings between the source (training-time) and the target (test-time) datasets, so that reducing the difference between the source and target dataset distributions.

(2) *Missing Slice Detection*: We use a vector $\mathbf{s}$ to represent pixel values in each slice. A 3D cardiac MRI volume $\mathbf{V}$ with full coverage with z slices can be described as

$$\mathbf{V} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_z]\,. \tag{15.1}$$

Each cardiac volume, $\mathbf{V} = [\mathbf{s}_p, \dots, \mathbf{s}_q], p \leq q \in [1, z]$, can have a different number of slices, or have the same number of slices but cover a different portion of the LV (Fig. 15.2).

To guarantee accurate cardiac volumetry and functional measurements [15], full LV coverage is a basic requirement [21]. To address this problem, we propose a two-

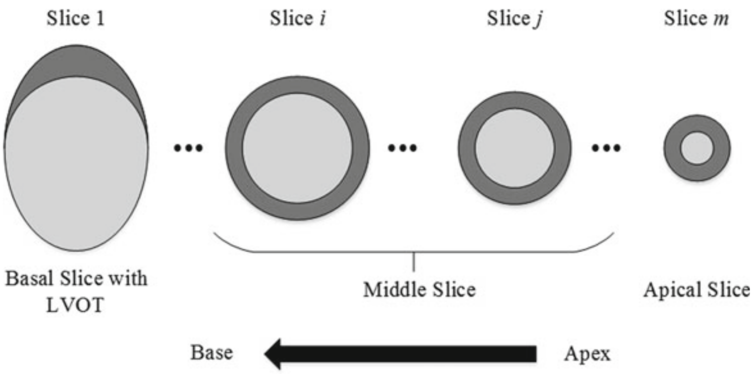


Fig. 15.2 Schematic LV shapes showing blood pool (light gray) and myocardium (dark gray) for different slices from apex to base. Slice 1 (left) shows LVOT, which identifies the basal slice

stage detection system that first computes intensity representations by an FDDI 3D CNN model and then detects missing slices based on the computed representations. In the first stage, we propose an FDDI 3D CNN to encode the spatial contextual information and hierarchically extract high-level features on multiple cohorts of different provenance to learn the appearance, which represent intensity representations. In the second stage, the missing basal and apical planes prediction is verified by the learned representations from the results of the first stage.

15.2.2 Dataset Invariance 3D Intensity Representations

Which 3D intensity representations? Our dataset invariance intensity representations are represented as a feature distribution matrix, which integrates information about the LV shape and size across dataset. We detect incomplete LV coverage by image classification using the distribution matrix. We define two classes—MAS (missing apical slice) and MBS (missing basal slice).

Given a particular describable visual representation, we can formalize our notion of 3D intensity representations based on Eq. (15.1). For example, if we are looking at the volume from base to apex, MAS and MBS can be formalized as

$$\begin{cases} \mathbf{V}_{MBS} = [\mathbf{s}_q, \ldots, \mathbf{s}_n], \\ \mathbf{V}_{MAS} = [\mathbf{s}_1, \ldots, \mathbf{s}_p], \end{cases} \tag{15.2}$$

where $p, q \in (1, n)$, $\mathbf{s}_1$ is the basal slice, and $\mathbf{s}_n$ is the apical slice. Our intensity representations classifiers can be thought of functions $f(\cdot)$ mapping 3D stacks $\mathbf{V}$ to real value p_i. Positive value of p_i indicates the presence or strength of the ith representation, while negative values indicate its absence. Considering our intensity representations, if we define $\mathbf{V}_1$ and $\mathbf{V}_2$ as the MBS and non-MBS samples, respectively, the representation function $f_{MBS}(\cdot)$ can map $\mathbf{V}_1$ to a positive value and $\mathbf{V}_2$ to a negative value. This is a binary classification function. Our 3D intensity representation classifiers are trained on the UK Biobank dataset as they provide reliable ground truth labels based on visual inspection and manual annotation.

Dataset Invariance: Inspired by adversarial learning (AL) [7] and dataset adaptation (DA) [30] for cross-dataset transfer, we propose a dataset-invariant adversarial learning model, which extends the DA formulation into an AL strategy, and performs them jointly in a unified framework. An overview of our method is depicted in Fig. 15.3a. Given a set of slices $\{\mathbf{x}_k^s\}_{k=1}^N$ with slice position labels $\{y_k^s\}_{k=1}^N$ for training, since dataset adversarial learning satisfies a dataset adaptation mechanism, to learn a model that can generalize well from one dataset to another, we minimize source and target representation distances through alternating *minimax* between two loss functions: one is the dataset discriminator loss

$$\begin{aligned}\mathscr{L}_d^k &= \mathscr{L}_d(G_{disc}(G_{conv}(\mathbf{V}_s;\mathbf{W}_d,\mathbf{b}_d),d_k)\\ &= -\sum_i \mathbb{1}\,[o_d = d_k]\log(G_{disc}(G_{conv}(\mathbf{V}_s;\mathbf{W}_f,\mathbf{b}_f);\mathbf{W}_d,\mathbf{b}_d),d_k),\end{aligned}\tag{15.3}$$

which classifies whether an image is drawn from the source or the target dataset. o_d indicates the output of the dataset classifier for the kth image, $\mathbf{W}_d, \mathbf{b}_d$ are the parameters used for the computation of the dataset prediction output of the network, which corresponds to the dataset invariance layers; $\mathbf{W}_f, \mathbf{b}_f$ are the representation parameters of the neural network feature extractor, which corresponds to the feature extraction layers; d_k denotes the dataset that the example slice k is drawn from. The other is the source and target mapping invariant loss

$$\begin{aligned}\mathscr{L}_f^k &= \mathscr{L}_f(G_{conf}(G_{conv}(\mathbf{V}_s;\mathbf{W}_f,\mathbf{b}_f);\mathbf{W}_d,\mathbf{b}_d),d_k)\\ &= -\sum_d \frac{1}{D}\log(G_{conf}(G_{conv}(\mathbf{V}_s;\mathbf{W}_f,\mathbf{b}_f);\mathbf{W}_d,\mathbf{b}_d),d_k),\end{aligned}\tag{15.4}$$

which is optimized with a constrained adversarial objective by computing the cross-entropy between the output predicted dataset labels, and a uniform distribution over dataset labels. D indicates the number of input channels. $G_{conv}(\cdot)$ is a convolution layer function that maps an example into a new representation; $G_{sigm}(\cdot)$ is a label prediction layer function; $G_{disc}(\cdot)$ and $G_{conf}(\cdot)$ are the dataset prediction and invariance layer functions.

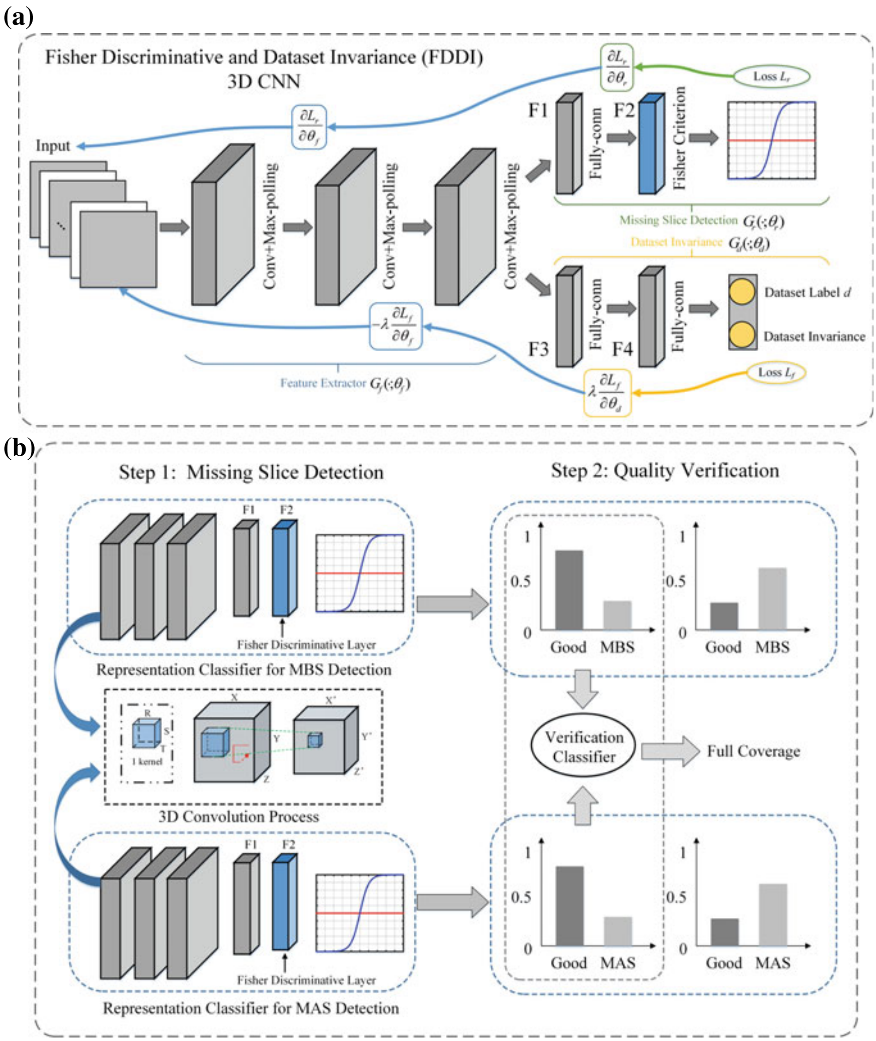

(a)
Fisher Discriminative and Dataset Invariance (FDDI)
3D CNN
Input
Conv+Max-polling
Conv+Max-polling
Conv+Max-polling
$\frac{\partial L_r}{\partial \theta_r}$
Loss L_r
$\frac{\partial L_r}{\partial \theta_f}$
F1
Fully-conn
F2
Fisher Criterion
Missing Slice Detection $G_r(\cdot;\theta_r)$
Dataset Invariance $G_d(\cdot;\theta_d)$
F3
Fully-conn
F4
Fully-conn
Dataset Label d
Dataset Invariance
$-\lambda\frac{\partial L_f}{\partial \theta_f}$
Loss L_f
$\lambda\frac{\partial L_f}{\partial \theta_d}$
Feature Extractor $G_f(\cdot;\theta_f)$
(b)
Step 1: Missing Slice Detection
Step 2: Quality Verification
F1
F2
Fisher Discriminative Layer
Representation Classifier for MBS Detection
1
0.5
0
Good
MBS
1
0.5
0
Good
MBS
R
S
T
1 kernel
X
Y
Z
X'
Y'
Z'
3D Convolution Process
Verification Classifier
Full Coverage
F1
F2
Fisher Discriminative Layer
Representation Classifier for MAS Detection
1
0.5
0
Good
MAS
1
0.5
0
Good
MAS

Fig. 15.3 The whole assessment framework. **a** The proposed architecture includes a deep feature extractor and a missing slice predictor (green) as the first branch, which together form a standard feed-forward architecture. Dataset Invariance is achieved by adding a dataset classifier (yellow) connected to the feature extractor as the second branch; **b** the framework of our LV coverage assessment process

15.2.3 Fisher Discriminative 3D CNN Model

In this subsection, we propose an FD3D CNN (shown in Fig. 15.6c) to extract high-level features, which represent 3D intensity representations. Our FD3D CNN model is designed by adding a new Fisher discriminative fully connected layer, F2, that uses the output of the previous layer, F1, as input. The new layer is then stacked onto a conventional 3D CNN. To maximize the inter-class distances between the learned features while minimizing the intra-class distances of the learned features, we train the newly added Fisher discriminative layer F2 on CNN features based on a Fisher discriminant criterion [33].

To process our main task of the missing slice detection, and boost the discriminative power of 3D CNN learned features, we impose a Fisher discrimination criterion [33] on them. Given the 3D input data $\mathbf{V}_i^t$ where i is the representation class, with $i = \{1, 2\}$, corresponding to MAS and MBS; the superscript t in $\mathbf{V}_i^t$ indicates whether the representation is positive or negative, i.e., $t = \{0, 1\}$; $\mathbf{V}_i^t = \left[\mathbf{v}_{i,1}^t, \mathbf{v}_{i,2}^t, \ldots, \mathbf{v}_{i,C}^t\right]$, $\mathbf{v}_{i,j}^t$ is the input data of jth sample from class i, for $j = 1, 2, \ldots, C$. We denote $\mathbf{F}_{i,j}^t$ to be the features in the fully connected layer of the 3D CNN for class i and jth sample. Using the Fisher criterion, discrimination is achieved by minimizing the within-class scatter of $\mathbf{F}^t$, denoted by $S_w(\mathbf{F}^t)$, and maximizing the between-class scatter of $\mathbf{F}^t$, denoted by $S_b(\mathbf{F}^t)$. $S_w(\mathbf{F}^t)$ and $S_b(\mathbf{F}^t)$ are defined as

$$S_w(\mathbf{F}^t) = \sum_{i=1}^{I} \sum_{\mathbf{F}_{i,j}^t \in t} (\mathbf{F}_{i,j}^t - \mathbf{m}_i^t)(\mathbf{F}_{i,j}^t - \mathbf{m}_i^t)^T, \tag{15.5}$$

$$S_b(\mathbf{F}^t) = \sum_{i=1}^{I} n_i (\mathbf{m}_i^t - \mathbf{m}_i)(\mathbf{m}_i^t - \mathbf{m}_i)^T, \tag{15.6}$$

where $\mathbf{m}_i^t$ and $\mathbf{m}^t$ are the mean vectors of $\mathbf{F}_i^t$ and $\mathbf{F}^t$, respectively, and n_i is the number of samples from class i. The Fisher discriminant regularization term $\Phi(\mathbf{F}^t)$ is defined as $\mathrm{tr}(S_w(\mathbf{F}^t)) - \mathrm{tr}(S_b(\mathbf{F}^t))$. To obtain a discriminative classification result with deep learned features, we propose to modify the objective function of DID3D CNN model by inserting a Fisher discriminant regularization term:

$$\begin{aligned} \mathbf{J}(\mathbf{W}, \mathbf{b}) = &\mathop{\arg\min}_{\mathbf{W},\mathbf{b}} \frac{1}{I} \sum_{i=1}^{I} \mathbb{1}\left[o_d = d_k\right] \log(G_{disc}(G_{conv}(\mathbf{V}_{i,j}^{s,t}; \mathbf{W}_f, \mathbf{b}_f); \mathbf{W}_y, \mathbf{b}_y), y_k) \\ &+ \frac{1}{2}\lambda\left(\|\mathbf{W}_f\|_2^2 + \|\mathbf{W}_y\|_2^2\right) + \frac{1}{2}\eta(tr(S_w(\mathbf{F}^t)) - tr(S_b(\mathbf{F}^t))), \end{aligned} \tag{15.7}$$

where $\mathbf{J}$ is our new cost function which can minimize the within-class scatter and maximize the between-class scatter. The output activation $a(\mathbf{V}_{i,j}^t; \mathbf{W}, \mathbf{b}) = 1/(1 + e^{-\mathbf{W}\mathbf{V}_{i,j}^t - \mathbf{b}})$ is typically restricted to the open interval $(0, 1)$ by using a logistic sigmoid, which is parametrized by $\mathbf{W}$ and $\mathbf{b}$ on the jth training sample. $\|\mathbf{W}\|_2^2$ is a penalty term to the loss function that prevents the weights from getting too large, and helps preventing overfitting. The weights in each layer can be adjusted toward the target classes and let the input data close to the corresponding classes when we have no large dataset or small number of iteration. Here, $\lambda, \eta \in [0, 1]$ are two trade-off parameters that control the relative importance of each term and usually chosen by experiments, which could be different based on different databases and network structures.

To obtain a dataset invariance and discriminative classification result with deep learned features, we propose to modify the objective function of 3D CNN model by combining Eqs. (15.4) and (15.7):

$$\begin{aligned}\mathbf{J}^*(\mathbf{W},\mathbf{b}) = &\arg\min_{\mathbf{W},\mathbf{b}} \frac{1}{I}\sum_{i=1}^{I} \mathbb{1}\left[o_d = d_k\right] \log(G_{disc}(G_{conv}(\mathbf{V}_{i,j}^{s,t}; \mathbf{W}_f, \mathbf{b}_f); \mathbf{W}_y, \mathbf{b}_y), y_k) \\ &- \sum_{d} \frac{1}{D} \log(G_{conf}(G_{conv}(\mathbf{V}_s; \mathbf{W}_f, \mathbf{b}_f); \mathbf{W}_d, \mathbf{b}_d), d_k) \\ &+ \frac{1}{2}\eta(tr(S_w(\mathbf{F}^t)) - tr(S_b(\mathbf{F}^t))) + \frac{1}{2}\lambda\left(\|\mathbf{W}_f\|_2^2 + \|\mathbf{W}_y\|_2^2 + \|\mathbf{W}_d\|_2^2\right). \end{aligned} \tag{15.8}$$

All the losses are readily implemented in standard deep learning frameworks, and after setting learning rates properly so that Eq. (15.3) only updates $\mathbf{W}_d$, $\mathbf{b}_d$ and Eq. (15.4) only updates $\mathbf{W}_f$, $\mathbf{b}_f$, the updates can be performed via standard back-propagation [16]. Together, these updates ensure that we learn a representation that is dataset-invariant. Our key problem is to calculate the error of the output units, which is consisted of the output errors from the two sub-functions $\mathbf{J}^*(\mathbf{W}, \mathbf{b})$. To update the parameters $\mathbf{W}_i^t$ and $\mathbf{b}_i^t$, we first calculate the error $\delta_i^{L,t}$ (L is the output layer) of the output layer with forward propagation, then we adopt the back-propagation method [12] to calculate the error $\delta_i^{l,t}(l < L)$ for other layers. The partial derivatives of the overall cost function $\mathbf{J}^*(\mathbf{W}, \mathbf{b})$ regarding $\mathbf{W}^t$ and $\mathbf{b}^t$ are

$$\frac{\partial \mathbf{J}^*(\mathbf{W},\mathbf{b})}{\partial W^{l,t}} = \sum_{t=0}^{C}\sum_{F^t \in t} \frac{\partial \mathbf{J}(\mathbf{W}^t,\mathbf{b}^t)}{\partial W^{l,t}} + \eta \sum_{t=0}^{C}\sum_{F^t \in t} \frac{\partial \Phi(\mathbf{F}^t)}{\partial W}, \tag{15.9}$$

$$\frac{\partial \mathbf{J}^*(\mathbf{W},\mathbf{b})}{\partial b^{l,t}} = \sum_{t=0}^{C}\sum_{F^t \in t} \frac{\partial \mathbf{J}(\mathbf{W}^t,\mathbf{b}^t)}{\partial b^{l,t}} + \eta \sum_{t=0}^{C}\sum_{F^t \in t} \frac{\partial \Phi(\mathbf{F}^t)}{\partial b}. \tag{15.10}$$

In this stage, we use the 3D CNN model with the architecture in Table 15.1. Algorithm 1 gives the pseudocode to train this new network. In our 3D CNN implementation, a rectifier linear unit (ReLU) [17] is utilized as nonlinear activation function in the C and F1 layers.

Algorithm 1: FDDI 3D CNN Training

```
Input: the input–target pairs ($\mathbf{v}_{i,j}^{t}, \mathbf{y}^{t}$), corresponding $j$th pairs from class $i$, $t$ indicates the
        positive or negative sample; $\eta$.
Output: DID3D CNN weight and biases, respectively, $\mathbf{W} = [\mathbf{W}^{1,t}, \mathbf{W}^{2,t}, \dots, \mathbf{W}^{l,t}]$ and
        $\mathbf{b} = [\mathbf{b}^{1,t}, \mathbf{b}^{2,t}, \dots, \mathbf{b}^{l,t}]$.
begin
Initialize $\mathbf{W}_{i,j}^{t}$ and $\mathbf{b}_{i,j}^{t}$
while stopping criterion has not been met do
    (1) Calculate $\mathbf{W}_f$ and $\mathbf{W}_d$ for dataset invariance using the Eqs. (15.3) and (15.4) iteratively.
    (2) Calculate $\mathbf{W}_f$ and $\mathbf{W}_y$ for representation classifier optimization.
    (3) Update $\mathbf{W}_f$ and $\mathbf{W}_y$ using Fisher discriminant $\Phi(\mathbf{F}^t) = \mathrm{tr}(S_w(\mathbf{F}^t)) - \mathrm{tr}(S_b(\mathbf{F}^t))$.
    (4) Update $\mathbf{W}_{i,j}^{t}$ and $\mathbf{b}_{i,j}^{t}$ with Eqs. (15.9) and (15.10).
end
return $\mathbf{W}_{i,j}^{t}$ and $\mathbf{b}_{i,j}^{t}$ until the values of $\mathbf{J}^*(\mathbf{W}, \mathbf{b})$ in successive iterations are close enough or
the maximum number of iterations is reached.
end begin
```

Table 15.1 The architecture of the FDDI 3D CNN model

Layer	Kernel size	Stride	Output size	Feature volumes
Input	–	–	120 × 120 × 3	1
C1	7 × 7 × 2	1	114 × 114 × 2	16
M1	2 × 2 × 1	2	57 × 57 × 2	16
C2	13 × 13 × 2	1	45 × 45 × 1	16
M2	3 × 3 × 1	1	15 × 15 × 1	16
C3	10 × 10 × 1	1	6 × 6 × 1	64
M3	2 × 2 × 1	1	3 × 3 × 1	64
F1	–	1	1 × 1 × 1	256
F2	–	1	1 × 1 × 1	256
F3	–	1	1 × 1 × 1	256
F4	–	1	1 × 1 × 1	128

Note F1 and F2 belong to the missing slice detection branch and F2 is the Fisher discriminant layer; F3 and F4 belong to the dataset invariance branch

15.3 Materials and Metrics

15.3.1 CMR Acquisition Protocol and Annotation

UK Biobank CMR Protocol: UK Biobank's CMR acquisitions are performed on a clinical wide-bore 1.5T scanner (MAGNETOM Aera, Syngo Platform VD13A, Siemens Healthcare, Erlangen, Germany), and include piloting, sagittal, transverse, and coronal partial coverage of the chest and abdomen [25]. For measuring cardiac function, three long axis cines are acquired (viz. horizontal long axis—HLA, vertical long axis—VLA, and left ventricular outflow tract—LVOT in both sagittal and coronal views). In addition, a complete short axis (SA) stack is acquired. All acquisitions use balanced steady-state free precession (bSSFP) MRI sequences, attempting full coverage of the left ventricle (LV) and right ventricle (RV) [23]. In this work, we will focus on the SA bSSFP cine CMR data. To date, over 18,800 volunteers have been scanned. The voxel and matrix size of these CMR images are, respectively, $1.8 \times 1.8 \times 8.0\,\text{mm}^3$, and 208×187 with, approximately, 10 slices per volume. Each volumetric sequence contains about 50 cardiac phases.

Gold-standard image quality annotations: quality-scored cardiac MRI data is available for circa 5,000 volunteers of the UK Biobank (UKBB) imaging resource. Following visual inspection, manual annotation was carried out with a simple three-grade quality score [2]: (1) optimal quality for diagnosis, (2) sub-optimal quality yet analyzable, and (3) bad quality and diagnostically unusable. In 5,065 SA cine CMR from the same number of volunteers, 4,361 sequences correspond quality score 1, additional 527 sequences have a quality score 2, and the remaining 177 sequences have quality score 3. All datasets with optimal quality (score 1) had full coverage of the heart from base (LVOT existing) to apex (LV cavity still visible at end-systole). This data was used to construct the ground truth classes for our experiments. Note that having full coverage should not be confused with having the top/bottom slices corresponding exactly to base/apex.

15.3.2 Training and Testing Set Definitions

Training set: To create a training dataset for learning intensity representations, we extract the 3 topmost slices as negative samples for MBS detection and the three bottommost slices as negative samples for MAS detection. To create positive samples, we choose three-slice blocks each starting from the middle slice toward the base/apex for MBS/MAS detection training, respectively. We created the training set from images with optimal quality with exclusively full coverage.

(1) Dataset Invariance: We validated the proposed FDDI model on three target datasets: UKBB, DETERMINE[1] and MESA.[2] To prevent overfitting due to insufficient target data, and to improve the detection rate of our algorithm, we employ data augmentation techniques to artificially enlarge the target datasets. For this purpose, we chose a set of realistic rotations, scaling factors, and corresponding mirror images, and applied them to the MRI images. The set of rotations chosen were $-45°$ and $45°$, and the scaling factors 0.75 and 1.25. This increased the number of training samples by a factor of eight. After data augmentation, we had 2400, and 2384 sequences for DETERMINE and MESA datasets, and 845,000 stacks of 2D CMR slices from 3,380 sequences each with 50 cardiac phases, with quality score 1 for UKBB dataset. These augmented data is used for experiments in Sect. IV-A, B, and C. We set aside

[1] http://www.cardiacatlas.org/studies/determine/.

[2] http://www.cardiacatlas.org/studies/mesa/.

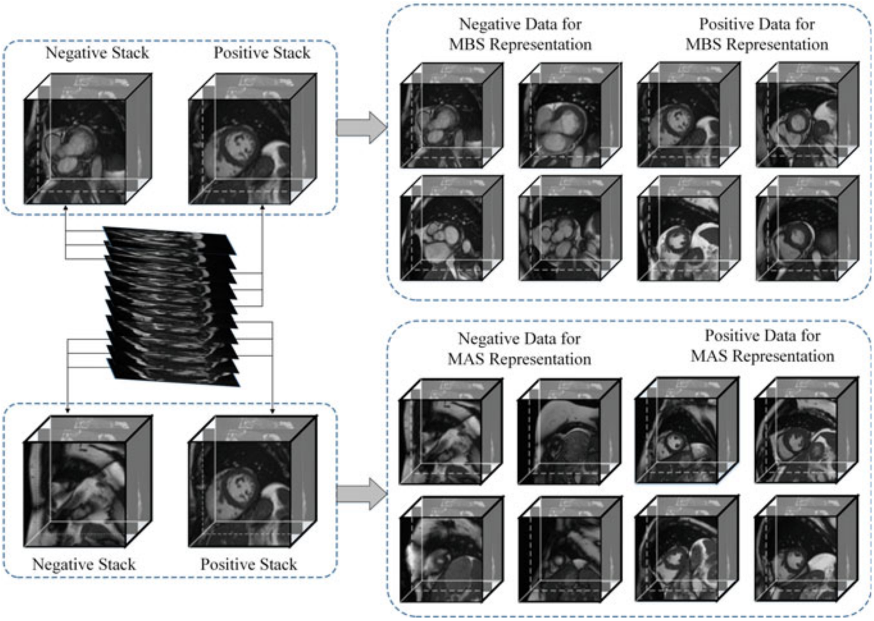

Negative Stack
Positive Stack
Negative Data for MBS Representation
Positive Data for MBS Representation
Negative Data for MAS Representation
Positive Data for MAS Representation
Negative Stack
Positive Stack

Fig. 15.4 The process of training data origination. *Top row*: 3D stacks extraction for missing basal slice detection; *bottom row*: 3D stacks extraction for missing apical slice detection

981 sequences and the data with quality score 2, 3 for later use in Sect. IV-D. To the best of our knowledge, this is the largest annotated dataset available to date for automatic CMR quality assessment (Fig. 15.4).

(2) Missing Slice Identification: We train using three-slice stacks (or triplets) to model 3D context. The average number of slices per image volume is around 10, and during training, we extract 4 triplets (two samples containing the base/apex and two samples without the base/apex). To maximize inter-class separation, it is wise to avoid intersection between training samples, for example, if we use four-slice stacks (for a 10-slice volume), there will be a 2-slice overlap between both the basal positive/negative examples and the apical region. By choosing the proposed slice triplets, we ensure there is no overlap and increase the discriminative power of the FDDI 3D CNN. Another important observation that supports the choice of slice triplets is that a CMR scan volume is not actually acquired at once but each slice is collected over several cardiac cycles leading to some degree of slice-to-slice misalignment. This effect is minimized when considering only slice triplets in contrast to, on the other extreme, using the full 3D volume.

Testing set: We extract every three adjacent slices from top to bottom for each volume and apply these triplets into the intensity representation classifiers. Data with known MBS/MAS are created by manually removing the 3 topmost/bottommost slices, respectively, from images with optimal quality as in the training set.

During training and testing, three-slice stacks are the input to the proposed FDDI 3D CNN. The scores of the output layer can be interpreted as likelihoods that the triplets correspond to normal LV volumes or to either missing basal/apical triplets. Our deep learning methods process images with 3D small blocks (120 × 120 × 3), which are cropped centered on the images to extract specific regions of interest. The parameter setting of block size is detailed in Sect. 15.4.1.

15.3.3 Learning Performance Metrics

To evaluate the learning process we use established classification metrics: precision, sensitivity and error rate as $\text{Precision} = TP/(TP + FP)$, $\text{Sensitivity} = TP/(TP + FN)$, $\text{Error Rate} = (FP + FN)/N$, where TP, FP, and FN are the numbers of the true positive, false positive and false negative samples, respectively, and N represents the number of subjects in the test set.

15.4 Experiments and Results

We experimented to characterize the performance of our FDDI 3D CNN learning framework. The error (cost) functions used in learning are in the range [0, 1]. In all experiments, the learning process was terminated when the Standard Deviation (SD) of the error function over the last five iterations was smaller than $\sigma = 0.01$.

15.4.1 Hyper Parameter Selection on UK Biobank

A 3D CNN demands a suitable receptive field (i.e., input size) to achieve fine discrimination. Automatically choosing an optimal size for the input model is time-consuming, which makes it best suited for discriminative learning. Specifically, we compared three block size configurations, i.e., 120 × 120 × 3 (which removes redundant background information based on the central point of original images), 180 × 180 × 3 (which is the original size as we extracted and resized from the UK Biobank), and 80 × 80 × 3, which mostly contains the LV part at the center. We tested sizes smaller than the original block size of the classification model because we wanted to validate whether a larger input block with more contextual information can enhance the discrimination capability of the model. The results under these settings are shown in Table 15.2. With block size 80 × 80 × 3, the MAS/MBS detection precision rate reached 89.01 and 88.36%. The detection performance was improved to the precision rate of 91.81 and 90.73% under block size 120 × 120 × 3, demonstrating that increasing contextual information can enhance the discrimination capability of 3D CNN. When non-block was employed, the detection precision rate

Table 15.2 Performance versus block size

Block size	Precision		Sensitivity	
	MAS (%)	MBS (%)	MAS (%)	MBS (%)
80 × 80 × 3	89.01	88.36	88.24	87.94
120 × 120 × 3	**91.81**	**90.73**	**90.92**	**90.25**
180 × 180 × 3	90.12	89.78	89.63	88.92

decreased to 90.12 and 89.78% for MAS/MBS detection. This may be because too much redundant contextual information clutters the actual LV signature, and hence degrades detection performance. Based on these experiments, we set the block size to 120 × 120 × 3, and then achieve optimal detection performance.

LeCun et al. [29] and Salah et al. [28] used CNN to recognize handwritten digital numbers with different numbers of training samples on the MNIST dataset. Their results illustrate that when reducing training samples, the recognition rate of the algorithm drops sharply. Differences between data source make it difficult to generalize algorithms trained on specific datasets to other data sources. To demonstrate the behavior of the fisher discriminant criterion, we experimented with different numbers of training samples by increasing the training data from 10,000 to 800,000 in multiples of 10,000 as in Fig. 15.5 and compared the performance of the conventional 3D CNN and Fisher discriminative 3D CNN, respectively, tested on 45,000 UKB samples. We found that the two networks achieve poor results with less than 20,000 samples. We used the *improvement*, which is defined as $(1 - ER_{\mathrm{D}}/ER_{\mathrm{T}}) \times 100$ to benchmark our method against traditional 3D CNNs, where ER_{D} and ER_{T} are the error rates of our Fisher discriminative 3D CNN and traditional 3D CNN, respectively. For 40,000 training samples, our method improves the conventional 3D CNN error rate by around 29.1% for MBS detection. When the number of training samples is larger than 40,000, our method can still improve the test result with better performance compared to traditional 3D CNN. The error rates of MAS/MBS representation learning are shown in Fig. 15.5. The data in the figure illustrates that our proposed method achieves comparable results with less training data compared to conventional 3D CNNs. We chose 80% of the 845,000 training samples and tested with the remaining 20% samples. The results are shown in Table 15.3. Even when trained with fewer iterations, our method can achieve better results compared to traditional 3D CNN.

With sufficient training samples and iterations, most machine learning methods can improve their accuracy at a higher computational cost. However, we usually want to obtain a trained network in the shortest time possible. This is especially important in population imaging as new datasets can become available and retraining might be required. It is also a desirable feature during algorithmic development as finding an optimal architecture may require multiple training procedures for different parameter settings. We illustrated that our Fisher discriminative 3D CNN has a better error

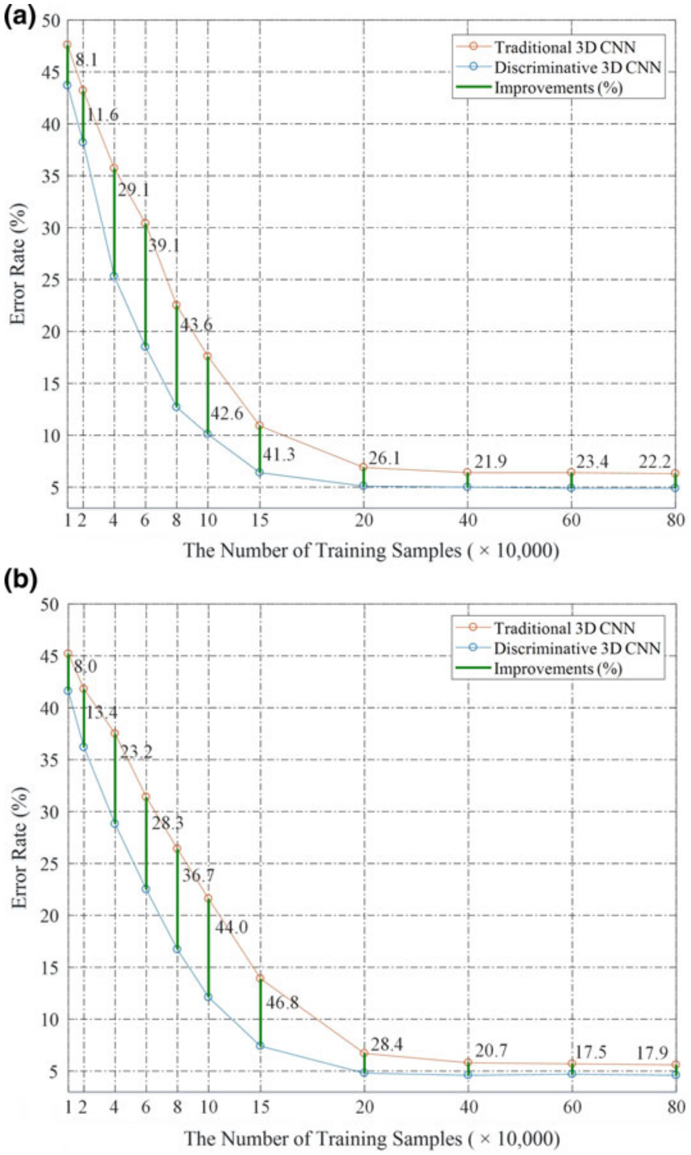
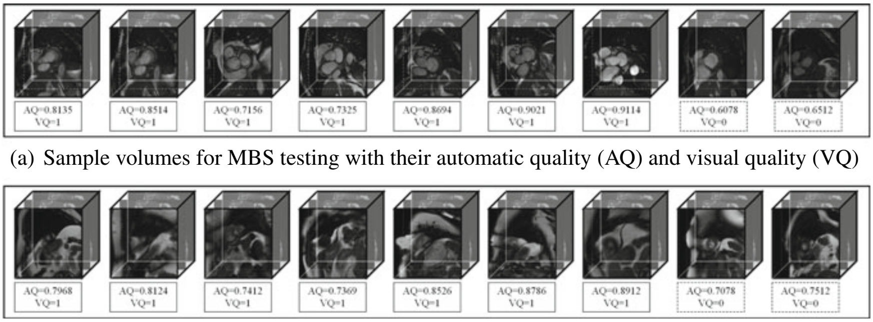


Fig. 15.5 Error rates and improvements for increasingly larger training sets: **a** MBS detection; **b** MAS detection

Table 15.3 Error rates versus learning epochs

Error rate (%)

Epochs	Traditional 3D CNN (MBS/MAS)	Discriminative 3D CNN (MBS/MAS)	Improvement (%) (MBS/MAS)
1	32.4/30.7	28.8/27.4	11.1/10.8
10	25.4/24.2	19.2/17.6	24.4/27.3
20	19.2/18.7	11.3/10.8	41.1/42.2
30	12.7/13.1	8.3/8.6	34.6/34.4
40	6.3/5.6	4.9/4.6	22.2/17.9



(a) Sample volumes for MBS testing with their automatic quality (AQ) and visual quality (VQ)

(b) Sample volumes for MAS testing with their automatic quality (AQ) and visual quality (VQ)

Fig. 15.6 Sample test volumes and their automatic quality (AQ) and visual quality (VQ) for MBS detection (top row) or MAS detection (bottom row) are shown. The left seven samples in each row show consistent between AQ and VQ, which means our algorithm gives the accurate prediction; the right two samples in each row show the wrong quality prediction

reduction performance as a function of the number of training samples and iterations than other competing techniques.

Typical classification results using the proposed fisher discriminative 3D CNN architecture are shown in Fig. 15.6. A few basal stacks (top row) and apical stacks (bottom row) in the test datasets with their automatic quality (AQ) or corresponding posterior probability values are shown. High score values on the stack correspond to the likelihood of being correct basal or apical triplets. The basal slices with existing LVOT indicate higher probability values of being correctly classified. This shows that the training has captured the LVOT as a prominent feature in the correctly positioned basal slices.

15.4.2 Dataset Adversarial Learning Performance

In a second experiment, we evaluated the generalization of the performance of our full LV coverage detection system on independent datasets. We assessed the sensitivity of our system to moderate changes in imaging conditions, scanner vendors, image resolution, etc. To this effect, we evaluate the performance of the missing slice detection task with and without dataset invariance (adaptation vs. non-adaptation), by transferring object intensity representation classifiers from the UKBB to MESA and DETERMINE. To fully evaluate the effectiveness of the proposed dataset adversarial learning method, we conduct comprehensive comparison of our approach with several state-of-the-art (related) approaches for cross datasets slice detection:

- **2D CNN**: Metric Classification with 2D CNN [34].
- **DI2D CNN**: 2D CNN with dataset invariance [6].
- **3D CNN**: Metric Classification with 3D CNN [13].
- **FD3D CNN**: Metric Classification with fisher discriminative 3D CNN.
- **FDDI 3D CNN**: Discriminative 3D CNN with dataset invariance.

To evaluate performance on MESA and DETERMINE, we manually generated annotations as follows: we checked one slice above and below the detected basal slice to confirm the slice is the basal and record true or false, ditto for apex. This dataset comprises 2400 cardiac MRI volumes after data augmentation. 6 to 12 SAX images were obtained from the atrioventricular ring to the apex. Gold-standard full LV coverage was obtained by an experienced reader (LZ) and checked visually by inspecting slices from base to apex. The original volumes are used for full LV coverage detection and triplets of top and bottom slices are used, respectively, as negative examples for MBS and MAS. Positive examples of MBS/MAS are obtained by manually removing the 3 topmost/bottommost slices. This dataset is used as test set on the different methods.

We compared our framework with non-adaption methods: 2D CNN, 3D CNN, and FD3D CNN, and the adaptation methods: DI2D CNN and FDDI 3D CNN. In particular, we chose the CNN architecture in [34] for conventional 2D CNN, and the 3D approach in [13] for 3D CNN and FD3D CNN metrics with non-adaptation, and the dataset adversarial learning architecture in [35] for both two adaption methods. For the methods with no adaptation, we train the networks only on UKBB dataset and test on the target datasets; for the methods with adaptation, we train the networks on both source and target datasets. Tables 15.4 and 15.5 show the precision and sensitivity for missing basal/apical slice of the adaptation and non-adaptation methods on MESA and DETERMINE, respectively. For both test datasets, the best improvements are the result of combining both of these features. For MESA the precision rate was increased by 29%, and for DETERMINE best improvements are of 22%. We also list the results obtained using handcrafted features [19]. The basal slice is identified by these steps: (1) Choose the middle slice image as the start image, and process each image sequentially in the basal direction. (2) Apply the optimal threshold method to convert the ROI to a binary image. (3) Identify the binary object with

Table 15.4 Cross-dataset performance comparison of different learning models with learned and handcrafted visual representations on MESA

Method	Precision (%)			Sensitivity (%)		
	MAS	MBS	$\overline{\mathrm{MBS} \vee \mathrm{MAS}}$	MAS	MBS	$\overline{\mathrm{MBS} \vee \mathrm{MAS}}$
FDDI 3D CNN	**88.37 ±0.31**	**89.14 ±0.34**	**88.59 ±0.38**	**87.95 ±0.41**	**89.03 ±0.38**	**88.42 ±0.40**
FD3D CNN	86.32 ±0.29	86.91 ±0.27	85.63 ±0.32	84.92 ±0.25	85.48 ±0.28	85.67 ±0.29
3D CNN	79.65 ±0.21	80.12 ±0.26	80.09 ±0.28	78.51 ±0.23	79.647 ±0.27	79.37 ±0.26
DI2D CNN	87.12 ±0.43	87.68 ±0.44	87.42 ±0.42	86.25 ±0.46	88.34 ±0.41	86.89 ±0.45
2D CNN	62.38 ±0.39	65.42 ±0.37	66.75 ±0.37	63.21 ±0.38	63.83 ±0.35	64.71 ±0.37
Lu et al. [19]	32.74 ±1.42	38.69 ±1.65	52.12 ±1.58	58.42 ±1.12	63.48 ±1.37	59.74 ±1.59

Table 15.5 Cross-dataset performance comparison of different learning models with learned and handcrafted visual representations on DETERMINE

Method	Precision (%)			Sensitivity (%)		
	MAS	MBS	$\overline{\mathrm{MBS} \vee \mathrm{MAS}}$	MAS	MBS	$\overline{\mathrm{MBS} \vee \mathrm{MAS}}$
FDDI 3D CNN	**89.64 ±0.28**	**89.78 ±0.29**	**89.42 ±0.26**	**88.36 ±0.28**	**89.63 ±0.30**	**89.79 ±0.29**
FD3D CNN	85.69 ±0.23	86.60 ±0.22	86.13 ±0.21	85.46 ±0.24	86.05 ±0.22	85.71 ±0.23
3D CNN	78.45 ±0.23	79.86 ±0.27	80.27 ±0.26	79.59 ±0.21	80.12 ±0.22	79.69 ±0.27
DI2D CNN	88.42 ±0.46	88.91 ±0.48	88.64 ±0.42	87.29 ±0.39	88.47 ±0.41	88.58 ±0.46
2D CNN	69.65 ±0.49	71.42 ±0.42	71.87 ±0.44	69.32 ±0.41	69.76 ±0.43	69.83 ±0.44
Lu et al. [19]	33.68 ±1.37	40.12 ±1.42	54.29 ±1.26	58.31 ±1.14	62.73 ±1.39	61.57 ±1.58

blood pool, which shows a shape of ellipse. (4) Calculate the length of the major axis L of the ellipse that has the same normalized second central moments as the binary object. (5) If the ratio of current to preceding L is larger than a predefined threshold (e.g., >1.2 in this work), then a basal slice is identified; otherwise, the basal slice is missing. We use the similar method to identify the apical slice. We process each image sequentially from base to apex. If the ratio of the current to the preceding L is smaller than a predefined threshold (e.g., <0.2 in this work), an apical slice is detected; otherwise, the apical slice is missing. We employed this feature extraction procedure for prediction. The proposed FDDI 3D CNN shows the best precision and sensitivity figures in each representation classifier, and the full LV coverage detection performance.

Table 15.6 Confusion matrix of the expert cardiologist (VQ1) and cardiac image expert's visual (VQ2) results. Gray numbers indicate number and ratio of correct estimates

(a) MESA

		VQ2			Correct
		MBS	MAS	$\overline{\text{MBS} \vee \text{MAS}}$	
VQ1	MBS	67	0	3	0.96
	MAS	0	65	2	0.97
	$\overline{\text{MBS} \vee \text{MAS}}$	3	1	59	0.94

(b) DETERMINE

		VQ2			Correct
		MBS	MAS	$\overline{\text{MBS} \vee \text{MAS}}$	
VQ1	MBS	68	0	2	0.97
	MAS	0	64	3	0.96
	$\overline{\text{MBS} \vee \text{MAS}}$	2	1	60	0.95

15.4.3 Intra-rater Agreement of Full LV Coverage Detection

To contextualize the results of the automatic full LV coverage assessment, we compared it to intra-rater full LV coverage detection rate by expert readers. Intra-rater agreement [9] of human experts was evaluated by reassessing a subset of 200 random CMR datasets. To ensure robustness of the results, the designed FDDI3D CNN was trained three times with 800,000 samples with random parameter initialization and was evaluated against visual quality assessment by an expert cardiologist on the MESA and DETERMINE datasets. The inter-observer agreement for full LV coverage was evaluated as the mean absolute error (MAE) between the automatic (AQ) and the expert's visual (VQ) results. The confusion matrix of the proposed network for MESA and DETERMINE, AQ versus VQ is presented in Table 15.6a, b. A few test samples with their corresponding VQ and AQ are depicted in Fig. 15.6.

According to the confusion matrix, among the 200 reassessment samples, there were only nine samples in MESA and eight in DETERMINE with inconsistent quality between AQ and VQ, most of which were further confirmed with the consultant cardiologist who graded the CMR to be outliers of the original labeling process. Distribution quality levels in this randomly selected subset was compared to the original data using Pearson's χ^2 goodness-of-fit test to confirm that it represents the original data distribution (p-value > 0.05). Reassessed samples demonstrated a high agreement with the original qualities (Cohen's $\kappa = 0.76$, p-value < 0.05).

15.4.4 Implementation Considerations

The experiments here reported were conducted using the ConvNet library [4] on an Intel Xeon E5-1620 v3 @3.50 GHz machine running Windows 10 with 32 GB RAM and Nvidia Quadro K620 GPU. The networks were optimized using gradient descent method [16] with these hyperparameters: learning rate $= 0.01$, momentum $= 0.9$, drop-out rate $= 0.1$. The trainable weights were randomly initialized from a Gaussian distribution ($\mu = 0$, $\sigma = 0.01$) and updated with standard back-propagation. The models converged in about 6 h when training with 800,000 volumes with size $120 \times 120 \times 3$. Testing was fast and could process each volume in 3 s.

15.5 Conclusion

In this paper, we tackled the problem of detecting incomplete LV coverage in large population imaging databases. We illustrated the concept by proposing a discriminative 3D CNN tested on CMR data from the UK Biobank. Our FDDI 3D CNN is proposed by adding a new Fisher discriminative fully connected layer into the network, which achieved a significant improvement in intensity representation. Learned representation classifiers are computed on the candidates to corresponding quality categories. We also validated our model by training with UKBB pilot datasets and cross-evaluating it in CMR data from Data Science Bowl Cardiac Challenge. The proposed model shows a high consistency with human perception and becomes superior compared to the state-of-the-art methods, showing its high potential. Our proposed FDDI 3D CNN can also be easily applied and boost the results for other detection and segmentation tasks in medical image analysis.

References

1. Attili A, Schuster A, Nagel E, Reiber J, van der Geest R (2010) Quantification in cardiac MRI: advances in image acquisition and processing. Int J Cardiovasc Imaging 26:27–40. https://doi.org/10.1007/s10554-009-9571-x
2. Carneiro G et al (eds) (2016) Towards the semantic enrichment of free-text annotation of image quality assessment for UK Biobank cardiac cine MRI scans. In: Deep learning and data labeling for medical applications, vol 10008. Springer, Berlin
3. Cheng G, Zhou P, Han J (2016) RIFD-CNN: rotation-invariant and Fisher discriminative convolutional neural networks for object detection. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2884–2893. https://doi.org/10.1109/CVPR.2016.315
4. Demyanov S (2017) ConvNet library for Matlab. https://github.com/sdemyanov/ConvNet. Accessed 15 Oct 2017
5. Ferreira P, Gatehouse P, Mohiaddin R, Firmin D (2013) Cardiovascular magnetic resonance artefacts. J Cardiovasc Magn Reson 15:41. https://doi.org/10.1186/1532-429X-15-41
6. Ganin Y, Ustinova E, Ajakan H, Germain P et al (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2030–2096
7. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D et al (2014) Generative adversarial nets. Advances in neural information processing systems, pp 2672–2680
8. van der Graaf A, Bhagirath P, Ghoerbien S, Götte M (2014) Cardiac magnetic resonance imaging: artefacts for clinicians. Neth Heart J 22:542–549. https://doi.org/10.1007/s12471-014-0623-z
9. Gwet KL (2008) Intrarater reliability. Wiley encyclopedia clinical trials. Wiley, Hoboken, pp 1–14
10. He L, Tao D, Li X, Gao X (2012) Sparse representation for blind image quality assessment. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1146–1153
11. Huang Y, Shao L, Frangi AF (2017) Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding. In: IEEE conference on CVPR, pp 6070–6079
12. Ionescu C, Vantzos O, Sminchisescu C (2015) Matrix backpropagation for deep networks with structured layers. In: IEEE conference on computer vision (ICCV), pp 2965–2973. https://doi.org/10.1109/ICCV.2015.339

13. Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
14. Kang L, Ye P, Li Y, Doermann D (2014) Convolutional neural networks for no-reference image quality assessment. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1733–1740. https://doi.org/10.1109/CVPR.2014.224
15. Klinke V, Muzzarelli S, Lauriers N, Locca D, Vincenti G, Monney P, Lu C, Nothnagel D, Pilz G, Lombardi M, van Rossum A, Wagner A, Bruder O, Mahrholdt H, Schwitter J (2013) Quality assessment of cardiovascular magnetic resonance in the setting of the European CMR registry: description and validation of standardized criteria. J Cardiovasc Magn Reson 15:55. https://doi.org/10.1186/1532-429X-15-55
16. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, pp 1097–1105
17. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
18. Lisanti G, Masi I, Bagdanov AD, Del Bimbo A (2015) Person re-identification by iterative re-weighted sparse ranking. IEEE TPAMI 37(8):1629–1642
19. Lu Y, Connelly K, Dick A, Wright G, Radau P (2011) Watershed segmentation of basal left ventricle for quantitation of cine cardiac MRI function. J Cardiovasc Magn Reson 13:1. https://doi.org/10.1186/1532-429X-13-S1-P4
20. Ma L, Yang X, Tao D (2014) Person re-identification over camera networks using multi-task distance metric learning. IEEE TIP 23(8):3656–3670
21. Marcus J, Götte M, DeWaal L, Stam M, Van der Geest R, Heethaar R, Van Rossum A (1999) The influence of through-plane motion on left ventricular volumes measured by magnetic resonance imaging: implications for image acquisition and analysis. J Cardiovasc Magn Reson 1:1–6. https://doi.org/10.3109/10976649909080828
22. Moorthy A, Bovik A (2011) Blind image quality assessment: from natural scene statistics to perceptual quality. IEEE Trans Image Process 20:3350–3364. https://doi.org/10.1109/TIP.2011.2147325
23. Petersen S, Matthews P, Francis J, Robson M, Zemrak F, Boubertakh R, Young A, Hudson S, Weale P, Garratt S, Collins R, Piechnik S, Neubauer S (2016) UK Biobank's cardiovascular magnetic resonance protocol. J Cardiovasc Magn Reson 18:8. https://doi.org/10.1186/s12968-016-0227-4
24. Petersen SE, Matthews PM, Bamberg F, Bluemke DA, Francis JM, Friedrich MG, Leeson P, Nagel E, Plein S, Rademakers FE et al (2013) Imaging in population science: cardiovascular magnetic resonance in 100,000 participants of UK Biobank-rationale, challenges and approaches. J Cardiovasc Magn Reson 15:46. https://doi.org/10.1186/1532-429X-15-46
25. Petersen SE, Matthews PM, Francis JM, Robson MD, Zemrak F, Boubertakh R, Young AA, Hudson S, Weale P, Garratt S et al (2016) UK Biobank's cardiovascular magnetic resonance protocol. J Cardiovasc Magn Reson 18(1):8
26. Pusey E, Lufkin R, Brown R, Solomon M, Stark D, Tarr R, Hanafee W (1986) Magnetic resonance imaging artifacts: mechanism and clinical significance. Radiographics 6:891–911. https://doi.org/10.1148/radiographics.6.5.3685515
27. Saad M, Bovik A, Charrier C (2012) Blind image quality assessment: a natural scene statistics approach in the DCT domain. IEEE Trans Image Process 21:3339–3352. https://doi.org/10.1109/TIP.2012.2191563
28. Salah A, Alpaydin E, Akarun L (2002) A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. IEEE Trans Pattern Anal Mach Intell 24:420–425. https://doi.org/10.1109/34.990146
29. Sermanet P, Chintala S, LeCun Y (2012) Convolutional neural networks applied to house numbers digit classification. In: International conference on pattern recognition (ICPR). IEEE, pp 3288–3291
30. Sharmanska V, Quadrianto N (2016) Learning from the mistakes of others: matching errors in cross-dataset learning. In: IEEE conference on CVPR, pp 3967–3975

31. Wang Z, Wu G, Sheikh H, Simoncelli E, Yang EH, Bovik A (2006) Quality-aware images. IEEE Trans Image Process 15:1680–1689. https://doi.org/10.1109/TIP.2005.864165
32. Xue W, Mou X, Zhang L, Bovik A, Feng X (2014) Blind image quality assessment using joint statistics of gradient magnitude and Laplacian features. IEEE Trans Image Process 23:4850–4862. https://doi.org/10.1109/TIP.2014.2355716
33. Yang M, Zhang L, Feng X, Zhang D (2014) Sparse representation based Fisher discrimination dictionary learning for image classification. Int J Comput Vis 109:209–232. https://doi.org/10.1007/s11263-014-0722-8
34. Zhang L, Gooya A, Dong B, Hua R, Petersen SE, Medrano-Gracia P, Frangi AF (2016) Automated quality assessment of cardiac MR images using convolutional neural networks. In: International workshop on simulation and synthesis in medical imaging (SASHIMI). Springer, pp 138–145. https://doi.org/10.1007/978-3-319-46630-9_14
35. Zhang L, Pereañez M, Piechnik SK, Neubauer S, Petersen SE, Frangi AF (2018) Multi-input and dataset-invariant adversarial learning (MDAL) for left and right-ventricular coverage estimation in cardiac MRI. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 481–489

Chapter 16
Agent-Based Methods for Medical Image Registration

Shun Miao and Rui Liao

Abstract Medical imaging registration is a critical step in a wide spectrum of medical applications from diagnosis to therapy and has been an extensively studied research field. Prior to the popularity of deep learning, image registration was commonly performed by optimizing an image matching metric as a cost function in search for the optimal registration. However, the optimization task is known to be challenging due to (1) the non-convex nature of the matching metric over the registration parameter space and (2) the lack of effective approaches for robust optimization. With the latest advance in deep learning and artificial intelligence, the field of medical image registration had a major paradigm shift, whereby learning-based image registration methods are developed to employ deep neural networks to analyze images in order to estimate plausible registrations. Among the latest advances in learning-based registration methods, agent-based methods have been shown to be effective in both 3-D/3-D and 2-D/3-D registrations with significant robustness advantage over conventional optimization-based methods. In this chapter, we give an overview of agent-based methods for medical image registration and its two applications on rigid-body 3-D/3-D and 2-D/3-D registrations.

16.1 Introduction

The goal of medical image registration is to recover correspondences between two medical images acquired from (1) different patients, (2) the same patient at different time, or (3) different modalities, e.g., fluoroscopy, computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), etc. The

S. Miao (✉) · R. Liao
Siemens Healthineers, 755 College Rd E, Princeton, NJ 08540, USA
e-mail: shun.miao@outlook.com

R. Liao
e-mail: rui.liao@siemens-heathineers.com

S. Miao
PAII Inc., 6720B Rockledge Drive, Ste 410, Bethesda, Maryland 20817, USA

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_16

images are brought into the same coordinate system via various transformation models, e.g., rigid-body, affine, parametric splines, and dense motion fields [25]. The aligned images could then provide complementary information for decision-making, enable longitudinal change analysis, or guide minimally invasive therapy [9, 16].

Based on the dimensionality of the images to be registered, medical image registration tasks mainly fall into two categories: (1) 3-D/3-D image registration, where the two images to be registered are both 3-D imaging modalities, and (2) 2-D/3-D image registration, which aims at registering an intraoperative 2-D image with a preoperative 3-D image. Since the two categories of medical image registration tasks have fundamental similarities, they can often be formulated and solved with the same framework. Among different image registration frameworks, parametric image registration is the most widely adopted one, which uses parametric transformation models to describe the registration, and searches the optimal parameters of the transformation model to spatially align the two images. It can be applied to solve both 3-D/3-D and 2-D/3-D image registration. The parametric transformation model is applied to one 3-D image to align it with the other image, which can be a 3-D or 2-D image for 3-D/3-D and 2-D/3-D registration, respectively.

As deep learning is rapidly revolutionizing artificial intelligence and resulting in new state of the art for various computer vision tasks, the field of medical image registration also had a major paradigm shift toward deep learning, whereby deep neural networks are employed to analyze the images in order to perform registrations. Among the latest advances in learning-based registration methods, the agent-based methods, first introduced in [15, 20], have been shown to be effective in parametric 3-D/3-D and 2-D/3-D registration with significant robustness advantage over conventional optimization-based methods. In this chapter, we describe the agent-based registration methods, demonstrate two use cases of them, one on 3-D/3-D registration and one on 2-D/3-D registration, and compare them with conventional optimization-based parametric image registration methods.

The remainder of the chapter is organized as follows: Sect. 16.2 gives an overview of the background of medical image registration and discusses related works. Section 16.3 describes the agent-based method for parametric image registration. In Sects. 16.4 and 16.5 , we apply the agent-based method on rigid-body 3-D/3-D and 2-D/3-D registration problems, respectively, and compare its results with state-of-the-art methods. Section 16.6 discusses the advantages and disadvantages of the agent-based method.

16.2 Background

In this section, we give an overview of parametric image registration and discuss the recent advance in deep learning-based image registration. We also give backgrounds of deep reinforcement learning (DRL) and the special Euclidean group SE(3), which are used by the agent-based methods described in Sect. 16.3.

16.2.1 Parametric Image Registration

Medical image registration has been an active research area for more than two decades [18, 23]. Parametric image registration is the most widely adopted 3-D/3-D image registration framework, which formulates image registration as a searching problem. It uses a parametric transformation model (e.g., rigid-body transformation) to describe the state of image registration, defines a generic matching metric to measure the similarity of the image pairs to be registered, and performs optimization to search for the optimal transformations to maximize the matching metric [26]. In 2-D/3-D registration, the same parametric image registration framework is also widely adopted, but is referred to as intensity-based registration, where the registration state is also described by a transformation of the 3-D image, and the similarity metric is calculated on the 2-D image and a 2-D projection image of the 3-D image, referred to as digitally reconstructed radiography (DRR).

Since parametric image registration has been conventionally solved using optimization techniques (therefore often referred to as optimization-based registration), the success of the registration heavily depends on the global optimization of the matching metric. Popular matching metrics are mostly based on low level measurements, e.g., sum of squared distance (SSD), mutual information (MI) [17], cross-correlation (CC) [11], gradient correlation (GC) [3], etc. These metrics compare images directly at the pixel level without understanding the higher level structures. As a result, on images with a low signal-to-noise ratio (SNR) and/or severe image artifacts, they often have numerous local maxima, which makes it extremely challenging to locate the global solution using mathematical optimization strategy, e.g., Powell's method, Nelder–Mead, BFGS, CMA-ES.

16.2.2 Image Registration Using Deep Learning

While deep learning has achieved state-of-the-art performance in image segmentation, image recognition, and image classification, deep learning-based image registration is still an emerging field of research. Unsupervised learning using deep learning was proposed in [28] to extract features for deformable registration. However, these features are extracted separately from the image pairs and therefore cannot be guaranteed to be optimal for registration purpose. A deep learning-based regression approach was presented in [21] to solve 2-D/3-D registration for device tracking from 2-D X-ray images. Optical flow estimation between 2-D RGB images has been proposed using convolutional neural network (CNN) via supervised learning in [4]. The agent-based registration methods described in this chapter were first introduced in [15, 20], where 3-D/3-D and 2-D/3-D registrations are formulated as Markov decision process (MDP) and is solved using DRL techniques with a single-agent and multi-agent setup, respectively.

16.2.3 Deep Reinforcement Learning

In reinforcement learning (RL) [10], the agent learns to perform certain tasks through a reward system, via successive trial and errors. While RL has been widely studied in game theory, control, operations research, robotics, etc., it is only with the recent breakthroughs in (DRL), which combine RL with deep learning, that it could be applied to more complex problems, reaching human-level performances (e.g., Atari game [22] and Go [27]). In [2] an active detection model for localizing objects in 2-D RGB images is trained using DRL. Similarly, a detection agent is trained using DRL for localizing landmarks in 3-D CT images in [5]. However, one of the main challenges in DRL is the training process, which can be extremely time-consuming. Guided policy search [14] and imitation learning [12] were proposed for more efficient RL via improved policy/data sampling, which however were not directly applicable to the end-to-end trainable DRL framework. The agent-based image registration method described in this chapter adopts the DRL framework, but it trains the agent via deep supervised learning (DSL) to avoid the need of heavy environment exploration in order to improve training efficiency.

16.2.4 Special Euclidean Group SE(3)

Special Euclidean group SE(3) is the set of 4 × 4 matrices corresponding to translations and rotations. The tangent space of SE(3) is described using the Lie algebra se(3), which has six generators corresponding to the derivatives of translation and rotation along/around each of the standard axes. An element of se(3) is then represented by multiples of the generators

$$\boldsymbol{\delta} = (\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^6 \tag{16.1}$$

$$\begin{aligned}\boldsymbol{\delta}_\times &= u_1 G_1 + u_2 G_2 + u_3 G_3 + \\ &\quad v_1 G_4 + v_2 G_5 + v_3 G_6 \in \mathrm{se}(3),\end{aligned} \tag{16.2}$$

where (G_1, G_2, G_3) are the translation generators, and (G_4, G_5, G_6) are the rotation generators. Matrix exponential and logarithm can be taken to convert elements between SE(3) and se(3).

$$T = \exp(\boldsymbol{\delta}_\times) \in \mathrm{SE}(3). \tag{16.3}$$

16.3 Agent-Based Image Registration

In this section, we describe the agent-based image registration formulation. We start with formulating image registration problems as an MDP in Sect. 16.3.1, followed by Sects. 16.3.2–16.3.4 defining the action space, reward system, and the agent's observation of the environment. In Sect. 16.3.5, we further show that with the reward system used, the neural network can be learned by DSL, which greatly reduces training complexity comparing to DRL. In Sect. 16.3.6, we describe a multi-agent extension of the agent-based registration method.

16.3.1 Image Registration as an MDP

In 3-D/3-D registration, there are a floating (or moving) 3-D image, denoted as $I_f : \mathbb{R}^3 \mapsto \mathbb{R}$, and a reference image, denoted as $I_r : \mathbb{R}^3 \mapsto \mathbb{R}$. The floating and reference images can be of different 3-D image modalities (e.g., CT, MRI, Ultrasound, etc.). The goal of rigid-body 3-D/3-D registration is to estimate the optimal rigid-body transformation $T : \mathbb{R}^3 \mapsto \mathbb{R}^3$ that spatially aligns the floating image with the reference image. In 2-D/3-D registration, there is also a floating 3-D image, $I_f : \mathbb{R}^3 \mapsto \mathbb{R}$, which is usually CT or cone-bean computed tomography (CBCT), and a reference 2-D X-ray image $I_r : \mathbb{R}^2 \mapsto \mathbb{R}$. The goal of rigid-body 2-D/3-D is to estimate the optimal rigid-body transformation T that spatially aligns the projection of the 3-D floating image with the 2-D reference image. The projection of CT or CBCT can be calculated by DRR.

Since both 3-D/3-D and 2-D/3-D image registrations aim to estimate the transformation T, agent-based registration methods cast the problem of finding T as an MDP, which is defined by a five-tuple $\{\mathcal{T}, \mathcal{A}, P_\cdot(\cdot), R_\cdot(\cdot), \gamma\}$, where $\mathcal{T}$ is the set of possible states (i.e., transformations in SE(3)), $\mathcal{A}$ is the set of actions (i.e., modification of the transformation), $P_A(T)$ is the state obtained by taking action A in state T, $R_A(T)$ is the reward received by taking action A in state T, and γ is the discount factor that controls the importance of future rewards. The agent chooses the action to perform following a policy, denoted as $\pi(\cdot)$, which is a function that takes the current environment state as input and returns an action. With the action space $\mathcal{A}$ and the reward scheme $R_\cdot(\cdot)$ defined (detailed definitions in Sects. 16.3.2 and 16.3.3), starting from a state T, the expected total reward that will be collected with a policy π can be written as

$$V^\pi(T) = \sum_t \gamma^{t-1} r_t, \tag{16.4}$$

where r_i are the rewards collected following the policy π. The core problem of MDP is to find the policy $\pi^*(\cdot)$ to maximize the expected total reward:

$$\pi^* = \arg\max_\pi V^\pi(T), \quad \forall T \in \mathcal{T}. \tag{16.5}$$

For convenience, RL algorithms introduce an action-value function, $Q(T, A)$, which is a function of a state-action pair and returns a real value. The optimal action-value function $Q^*(T, A)$ means the expected total reward received by an agent starting in T, picking action A, and behaving optimally afterward. $Q^*(T, A)$ is an indication for how good it is for an agent to pick action A while being in state T. The optimal policy π^* can then be obtained by maximizing the optimal action-value function $Q^*(T, A)$:

$$\pi^*(T) = \arg\max_{A} Q^*(T, A). \tag{16.6}$$

The main goal of the popular Q-learning algorithm is to estimate the optimal action-value function, from which the optimal policy can be derived. In DRL, with an assumption that the action space is discrete, the optimal action-value function $Q^*(T, A)$ is modeled by a neural network, referred to as Q-network. The Q-network takes the agent's observation of the environment (detailed definition in Sect. 16.3.4) and the state T as input, and outputs $N = |\mathcal{A}|$ scalars corresponding to the values of $Q^*(T, A)$ of each action in the discrete action space $\mathcal{A}$.

The agent-based image registration method described in this chapter adopts the DRL framework. The action space, reward mechanism, and the agent's observation of the environment are critical components and need to be designed carefully, which we will detail in the next sections.

16.3.2 Action Space

Action space defines possible actions that the agent is allowed to take during an MDP to alter the state. It affects the complexity of the optimal action-value function to be learned and therefore needs to be carefully defined. To this end, a concept of *agent coordinate system* is first introduced, which is the coordinate system where the actions are defined and performed. The agent coordinate system can be described by a transformation E, from the image coordinate system to the agent coordinate system. The registration transformation T can then be described in the agent coordinate system, written as $E \circ T$. Since the DRL framework requires discrete actions, the action space is defined as a set of small movements in the tangent space of SE(3) at $E \circ T$, parameterized by se(3). Specifically, the action space contains 12 actions of positive and negative movements along the 6 generators of se(3), which can be written as

$$\mathcal{A} = \{-\lambda_1 G_1, \lambda_1 G_1, \ldots, -\lambda_6 G_6, \lambda_6 G_6\}, \tag{16.7}$$

where λ_i is the step size for the action along the generator G_i. Performance of an action $A \in \mathcal{A}$ is represented as

$$P_A(T) = E^{-1} \circ \exp(A) \circ E \circ T. \tag{16.8}$$

Since the actions need relatively small step size in order to achieve high accuracy, $\lambda_{1,2,3}$ are set to be 1 to get a step size of 1 mm in translation, and $\lambda_{4,5,6}$ are set to be $\pi/180 = 0.0174$ to get a step size of 1° in rotation. Such small step sizes allow the agent to reach a relatively accurate registration (with up to 1 mm and 1° errors in translation and rotation).

16.3.3 Reward System

In a standard MDP, the optimization target is a long-term reward, i.e., an accumulation of discounted future reward, due to the difficulty of forging a reward system that directly associates the immediate reward with the long-term goal. For image registration, however, one can define a distance-based reward system such that the immediate reward is tied with the improvement of the registration. The reward scheme is defined as the reduction of the geodesic distance to the ground truth transformation described in the agent coordinate system:

$$R_A(T) = D(E \circ T, E \circ T_g) - D(E \circ T', E \circ T_g), \tag{16.9}$$

where T and T' are transformations before and after the action, T_g is the ground truth transformation, and $D(\cdot, \cdot)$ denotes the geodesic distance of two transformations on SE(3) [7]:

$$\begin{aligned} D(T_1, T_2) &= \| \log(T_2 \circ T_1^{-1}) \|_F \\ &= \left(2\|\boldsymbol{u}\|_2^2 + \|\boldsymbol{v}\|_2^2\right)^{\frac{1}{2}}, \end{aligned} \tag{16.10}$$

where $\log(\cdot)$ takes $T_2 \circ T_1^{-1} \in \mathrm{SE}(3)$ into se(3), $\boldsymbol{u}$ and $\boldsymbol{v}$ are rotation and translation coefficients of $\log(T_2 \circ T_1^{-1})$ as described in Eq. 16.1. Because the units for rotation and translation are radian and mm, the distance impact of rotation is too small comparing to translation. To balance the impacts of rotation and translation, the rotation coefficients $\boldsymbol{v}$ are scaled by $180/\pi$ (i.e., change the unit from radian to mm).

16.3.4 Agent Observation

In DRL, the agent draws an observation of the environment in the current state and feeds it into Q-network to estimate the optimal action-value function. For image registration, the observation needs to (1) contain visual cue of the alignment of the floating and reference images, and (2) have a fixed size since it's used as the input of a neural network. A region of interest (ROI) of fixed size extracted from overlapping areas of the two images meets these requirements and is therefore used as the agent's observation. The location and orientation of the ROI are defined based on the origin and the axes of the agent coordinate system.

The origin of the agent coordinate system is typically selected to be located on the object of interest in the reference image, for both 3-D/3-D and 2-D/3-D registrations,

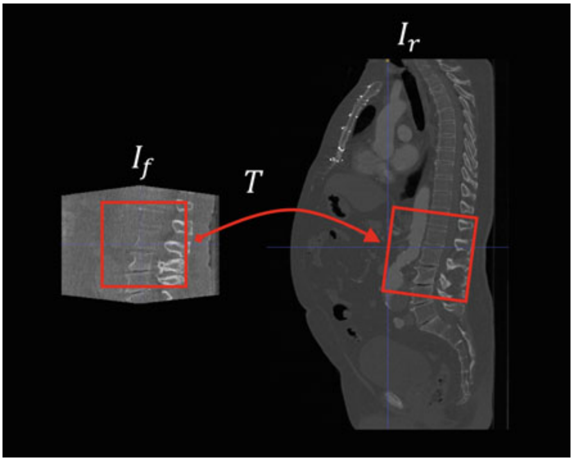


Fig. 16.1 Agent's observation in 3-D/3-D registration

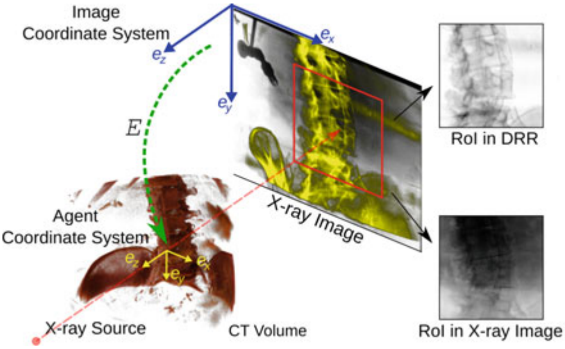


Fig. 16.2 Agent's observation in 2-D/3-D registration

in order to attend the agent to the important structures. However, the orientation of the agent coordinate system needs to be defined in slightly different ways in 3-D/3-D and 2-D/3-D registrations. In 3-D/3-D registration, the orientation of the agent coordinate system is simply aligned with the patient's orientation (left, anterior, and head). The ROI is then defined as a 3-D box centered on the origin and aligned with the three axes, and is extracted from both the reference and floating images as shown in Fig. 16.1. In 2-D/3-D registration, the orientation of the agent coordinate system is defined based on the X-ray projection geometry to make the action learnable (more discussions in Sect. 16.5). The ROI is defined as a 2-D box in the projected images, centered on the projection of the origin of the agent coordinate system and aligned with the projection of the axes of the agent coordinate system. The ROI is extracted from both the reference X-ray image and the DRR (i.e., floating image projected with transformation T), as shown in Fig. 16.2.

16.3.5 Learning Policy with Supervised Learning

The core problem in MDP is to find a policy that guides the decision process of the agent. In [22, 27], the policy learning process is formulated as a DRL problem, where the optimal action-value function $Q^*(T, A)$ is approximated by a neural network and learned following the Bellman equation as an iterative update. However, unguided exploration of the agent and iterative update of action-value function can result in a low training efficiency, as the agent has to try many combinations before reaching an effective policy. In agent-based registration, since the distance-based reward ties the immediate reward directly with the improvement of the registration, the optimal action-value function can be calculated following a greedy registration path where actions are taken to maximize the immediate reward. In this setup, the action-value function can be calculated recursively following the supervised greedy path:

$$Q(S_k, A_k) = R_{A_k}(S_k) + \gamma Q(S_{k+1}, A_{k+1}), \tag{16.11}$$

where

$$A_k = \arg\max_{A'} R_{A'}(S_k).$$

The agent terminates successfully when the distance to the ground truth transformation, $D(T_t, T_g)$, is less than 0.5. Upon successful termination, the agent receives a bonus reward (e.g., 10). Interestingly, it can be shown that if the agent is allowed to take continuous actions in the 6-D transformation parameter space with small steps (e.g., 1 mm and 1°), Eq. 16.11 is the optimal action-value function $Q^*(S, A)$ (readers are referred to [15] for proof).

As described in Sect. 16.3.1, a deep neural network is used to model $Q^*(S, A)$. The loss function used to train the network is

$$Loss = \sum_{k=1}^{M} \sum_{i=1}^{12} \|y_i - Q^*(T_k, A_i)\|_2, \tag{16.12}$$

where y_i is the ith ($i = 1 \ldots 12$) output of the network for the kth sample among M training samples. Since the ground truth value of $Q^*(S, A)$ can be pre-calculated, the network can be trained via DSL.

Using DSL to learn $Q^*(S, A)$ has two major advantages over using DRL. First, the ground truth $Q^*(S, A)$ is given analytically without iterative estimation so that the network could be trained much more efficiently and with a more stable convergence property. Second, calculating Eq. 16.11 does not require the exploration history of the agent, meaning that the state space and action space can be randomly sampled without memory replay, which lowers memory requirements. The advantage of using DSL is demonstrated in [15] in a 2-D/2-D registration toy problem, where the agent-based registration is used to register 2-D slices extracted from CT and CBCT. The

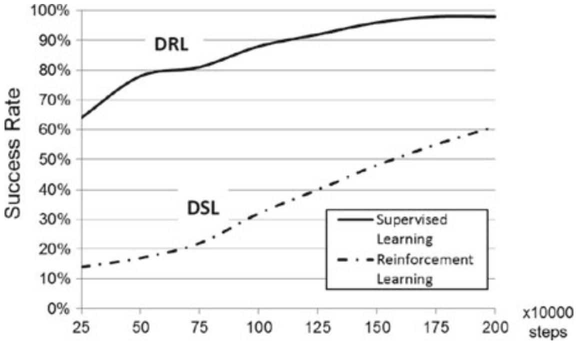


Fig. 16.3 Success rates achieved at different training steps for DSL and DRL

same network architecture and training meta-parameters are used for DSL and DRL. The training progress curve (Fig. 16.3) shows that DSL is significantly more efficient than DRL.

16.3.6 Multi-agent System

When a single agent is used for registration, the location of the ROI (i.e., the origin of the agent coordinate system) will need to be placed on the object of interest, which typically requires manual input or additional object detection steps. The selection of location can be avoided by using a multi-agent system introduced in [20], which provides auto attention and adaptively chooses the most reliable ROIs during registration. Specifically, scanning window can be used to create ROIs on the reference image, and each ROI is associated with one agent who observes it. If the scanning window using 1 pixel as its step size and the agents observing different ROIs follow the same policy, the network can be efficiently trained using a dilated fully convolutional network (FCN) (Please refer to [20] for details of the dilated FCN training).

During registration, the FCN is applied to produce a dense reward map, which contains estimated rewards for agents with scanning window ROIs from the input images. The reward map is denoted as $R_i(A)$, where i is the index of the agent and $A \in \mathcal{A}$ is the action. For every agent, the maximum reward is calculated and the action associated with it is selected:

$$\begin{aligned} \hat{R}_i &= \max_{A \in \mathcal{A}} R_i(A), \\ A_i &= \arg\max_{A \in \mathcal{A}} R_i(A). \end{aligned} \tag{16.13}$$

Since $\hat{R}_i$ is the anticipated reward of its selected action A_i, it represents the agent's confidence in the action. With the confidence map, one can design a mechanism to select agents with high confidence (e.g., in [20], agents are selected by thresholding with confidence scores). After the agents are selected, actions from the selected agents are further aggregated by L2 chordal mean to obtain the final actions:

$$\hat{A} = \arg \min_{A \in SE(3)} \sum_{i \in \mathcal{I}} \|A_i - A\|_F^2, \tag{16.14}$$

where $\mathcal{I}$ denotes the set of selected agents. The L2 chordal mean can be solved globally in close form as described in [7]. Intuitively, the aggregated action is the action with the smallest meaning distance to the selected actions on SE(3).

16.4 Agent-Based 3-D/3-D Image Registration

In this section, we introduce an application of the agent-based method on solving 3-D/3-D image registration problems. We first detail the implementation of the agent-based 3-D/3-D image registration in Sect. 16.4.1. This is followed by Sect. 16.4.2, which describes experiments and results on two 3-D/3-D registration tasks.

16.4.1 Implementation

A single-agent setting is used for 3-D/3-D registration, mainly because the memory consumption required by the multi-agent system is unaffordable with modern GPU RAM (i.e., 16 GB) on 3-D images. The ROI of the observation has a size of 64 × 64 × 64 voxels and is manually placed on the object of interest. For memory efficiency, the subtraction of the ROIs from the floating and reference images is used as the input of the network. The ROI size is relatively small comparing to the size of a typical CT image (e.g., 512 × 512 × 512), which is mainly due to the limitation of GPU memory. However, the size of the network input is of critical importance for practical use. For a combined robustness and accuracy, a hierarchical strategy based on attention is employed, where two separate networks are trained, both using 64 × 64 × 64 volumes as the input but with different resolutions and FOVs. The first network is trained for coarse alignment using down-sampled volumes with a lower resolution but larger FOV, helping the agent to gain global anatomical understanding and thus able to perform robust alignment of the object without being trapped into local optimum even when the initial displacement is large. The second CNN uses a high-resolution volume with a limited FOV and focuses on aligning the object as accurately as possible despite the limited FOV.

The same network architecture and meta-parameters for both coarse and fine registration. The network consists of 5 convolutional layers followed by three fully connected layers. The convolutional layers use 8, 32, 32, 128, 128 filters, all with 3 × 3 × 3 kernels. The first two convolutional layers are each followed by a max-pooling layer with stride 2. The three fully connected layers have 512, 512, and 64 activation neurons, and the output has 12 nodes corresponding to the 12

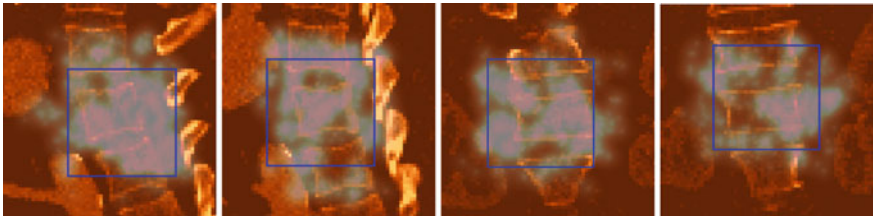


Fig. 16.4 Examples of saliency maps and attention of focus

actions. Each layer is followed by a nonlinear rectified layer, and batch normalization is applied to each layer. During training, each training pair is augmented 64,000 times, leading to more than 5M training data for each data-split. To train the CNN for coarse registration, rigid-body perturbation is randomly generated within [±30 mm, ±30 mm, ±30 mm, ±30°, ±30°, ±30°] for E2, and [±30 mm, ±30 mm, ±150 mm, ±30°, ±30°, ±30°] for E1 to cover the large FOV in the head-foot direction in spine CT. To train the CNN for refinement registration, rigid-body perturbation range is reduced to [±5 mm, ±5 mm, ±5 mm, ±5°, ±5°, ±5°].

The registration task is then performed as follows. First, the agent applies the first CNN to roughly align the object using N_1 (empirically set to 200) sequential actions. Then, a saliency map Ω is generated by computing the derivative of the sum of the network's outputs with respect to pixels of the input ROI via back-propagation. Ω indicates the importance of a given pixel in influencing the outcome of the CNN network in the first step of coarse registration. The most influencing pixels are selected via thresholding using 95th percentile. Their geometrical mean weighted by their importance is then calculated as the center of the ROI (marked by the blue rectangle box in Fig. 16.4) for the second step of refined registration. Finally, the ROI is extracted from the high-resolution volume, and starting from the final position obtained in the first step, the agent applies the second CNN with N_2 (empirically set to100) sequential actions.

Training of the neural network requires reference and floating image pairs with a random initial transformation T and a known ground truth transformation T_g. The initial transformation is obtained by randomly perturbing ground truth transformation T_g within a given range. Since aligned image pair with known T_g are not easily obtainable in the medical domain, a large number of initial transformations need to be generated for each aligned pair available to fully exploit the information. Denser sampling at transformations close to the ground truth transformation is also performed for finer training of the network close to the solution. Furthermore, each aligned pairs are also geometrically co-deformed by randomly generated affine transformations T_A:

$$T_A = I + \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 \\ c_{21} & c_{22} & c_{23} & 0 \\ c_{31} & c_{32} & c_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{16.15}$$

where I is the 4×4 identity matrix and all the elements in $[c_{ij}]_{i=1,2,3,j=1,2,3}$ for shearing are independently and sampled within a given range, to cover possible anatomical variations among patients in sizes and shapes.

16.4.2 Experiments and Results

The agent-based 3-D/3-D image registration method is evaluated on two 3-D medical image registration tasks: 1. (**Spine**) registering spine in CT and CBCT, where the main challenge is that CT has a much larger FOV than CBCT, leading to many local optima in the registration space due to the repetitive nature of the spine (Fig. 16.5b). 2. (**Cardiac**) registering heart in CT and CBCT, where the main challenge is the poor quality of CBCT with severe streaking artifacts and weak soft tissue contrast at the boundary of the object to be registered, i.e., the epicardium (Fig. 16.5c).

16.4.2.1 Experiment Setup

Registration accuracy of *Spine* is measured by 3-D target registration error (TRE) of manually annotated spine landmarks. Success rate is evaluated by TRE $\leq$ 10 mm. Registration accuracy of *Cardiac* is measured by the average mesh-to-mesh distance (MMD) of the segmented epicardium meshes. Success rate is evaluated by MMD $\leq$ 20 mm. Spine landmarks and epicardium segmentations are performed by experts. Iterative closest point registration [1] followed by visual inspection and manual editing whenever necessary is performed to provide the ground truth alignment.

Fivefold cross-validation is performed for both *Spine* and *Cardiac*. For each data-split, there are 82 pairs for training and 5 pairs for testing for *Spine*, and 92 pairs for training and 5 pairs for testing for *Cardiac*. Each test pair is randomly de-aligned

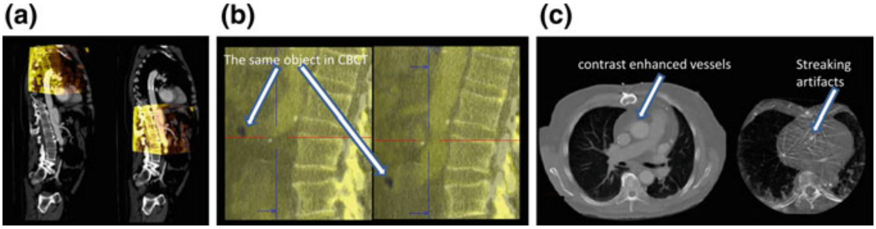


Fig. 16.5 **a** Overlay of spine CT (gray) and CBCT (yellow) volumes before (left) and after (right) registration, with large differences in FOVs. **b** Correct overlay of CT and CBCT (left) versus wrong overlay by shift of one vertebra of CBCT (right). The shift is shown by the movement of the dark object in CBCT, while the change in spine overlay is barely noticeable. **c** Cardiac CT (left) and CBCT (right) volumes, with weak soft tissue contrasts and severe streaking artifacts in the CBCT

10 times using rigid-body perturbation within the same range as those used for generating the training data, resulting in 5 × 10 × 5 × 2 = 500 test cases.

Three state-of-the-art 3-D image registration methods are applied to the same tasks for comparison. **ITK**: ITK registration [8], where MI computed based on 50 bins for the histogram is used as the matching metric, and optimization is obtained using multi-resolution optimizer based on a variant of gradient descent. **Quasi-global Search (QS)**: [19], where 2-D anatomy targeted projections are generated to surrogate the original 3-D image, allowing for a large number of matching metric evaluations, approximating global search. **Semantic registration (SR)**: [24], where the target organ is segmented from CT and a probability map is calculated from CBCT volumes using probability boosting tree (PBT) [24]. The segmentation and probability map are then used to iteratively register the two images. 600 CT and 393 CBCT volumes are used for training for epicardium segmentation, and 82 CT and 82 CBCT volumes are used for training for spine segmentation.

16.4.2.2 Results

The hierarchical registration introduced in Sect. 16.4.1 is applied for *Spine*. The effectiveness of using saliency map to attend the agent to the object of interest is demonstrated in Fig. 16.4. The median error is reduced from 3.4 mm after applying the first CNN to 2.5 mm after applying the second CNN. For *Cardiac*, the MMD is noticeable even for ground truth transformation due to the large, nonrigid deformation between CT and CBCT. Therefore, the refinement step is not necessary and is not applied/evaluated.

Quantitative results are summarized in Table 16.1. It is clear that the agent-based method is able to perform robust 3-D/3-D registration. Specifically, for *Spine*, the agent could reliably overcome local maxima and is not confused by the highly similar appearance of the neighboring vertebrae. Furthermore, the agent is robust to interfering objects and artifacts, as highlighted by the green arrows in Fig. 16.6 (from left to right: kidney, black background outside the image, and the deployed stent grafts). For *Cardiac*, the agent is able to learn the registration cues from raw high-dimensional training data, despite the low signal-to-noise ratio of the object to be registered. The results demonstrate that while the action of the agent is limited to a set of local movements for each step, thus making the training of the network easier compared to one-shot decision (regression), the contextual understanding and overall strategy of the agent is indeed global, helping the agent avoid local optimum and achieve robust registration.

Contrary to the proposed method, ITK and QS failed frequently in challenging cases, leading to relatively low success rates. While SR is more robust than ITK and QS, it required a significantly larger number of training examples than the agent-based method, and the performance deteriorated significantly when the number of

Table 16.1 Comparison of registration results (#1 and #2 results are marked in red and blue)

Methods	Spine (TRE mm)				Heart (MMD mm)			
	Success	10th	50th	90th	Success	10th	50th	90th
Ground truth	N/A	0.8	0.9	1.2	N/A	2.1	4.0	5.9
Initial position	N/A	35.5	73.9	116.2	N/A	9.2	22.8	30.5
ITK [8]	12%	1.9	77.3	130.4	14%	14.9	34.9	47.6
QS [19]	20%	1.6	60.9	136.2	14%	16.2	35.9	58.7
SR [24]	24%	3.0	34.9	71.0	72%	7.6	15.3	30.6
Agent-based	92%	1.7	2.5	3.8	100%	3.2	4.8	6.9

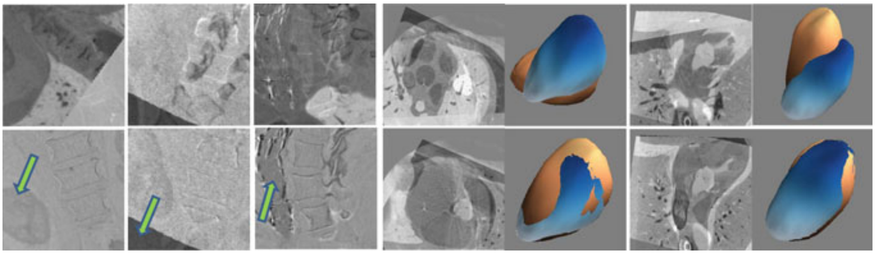


Fig. 16.6 Registration examples shown as the difference between the reference and floating images, before (upper row) and after (lower row) registration. The mesh overlay before and after registration is shown for the epicardium use case *Cardiac*

training samples is limited as in *Spine*. The limitation comes from the fact that SR does not inherently treat image registration as a problem of establishing the correspondence, but rather segments the objects from the two images separately, followed by a standard iterative optimization scheme that is prone to local optimum.

16.5 Agent-Based 2-D/3-D Image Registration

In this section, we introduce an application of the agent-based method on solving 2-D/3-D image registration problems. In Sect. 16.4.1, we give implementation details of the agent-based 2-D/3-D image registration. In Sect. 16.4.2, we present experiments and results on a clinical application of spine 2-D/3-D registration.

16.5.1 Implementation

The selection of the agent coordinate system is critical for agent-based 2-D/3-D registration, as it determines if the optimal policy is learnable. To make the policy learnable, the agent coordinate system needs to account for the X-ray projection geometry such that each action is associated with a specific appearance change of the DRR that is largely independent of the projection geometry. Specifically, the transformation T is described in the image coordinate system, which has its origin at the upper left corner of the image, its (x, y) axes along the image edges, and its z axes perpendicular to the image (illustrated in Fig. 16.2). The agent coordinate system has the same orientation as the image coordinate system, which provides an action space where the translation actions cause 2-D shift and zooming of the DRR, and rotation actions cause rotation of object in the DRR around the agent's origin. Therefore, the image appearance change of the DRR for a given action is largely independent of the underlying X-ray projection geometry, which makes the registration policy learnable without knowing the X-ray projection geometry.

The multi-agent system described in Sect. 16.3.6 is used for 2-D/3-D registration. Densely overlapping ROIs are automatically generated using scanning window. Specifically, the X-ray image is first resampled to a fixed pixel spacing of 1 mm. ROIs of size 64 × 64 scan through the resampled image with 1 pixel step size. The Q-network is modeled by a CNN, and CNNs of the agents observing such densely overlapping ROIs are modeled by a corresponding dilated FCN during both training and deployment of the agents. Table 16.2 shows the network configurations for both the CNN and its corresponding dilated FCN. Figure 16.7 shows examples of confidence maps produced by the dilated FCN. It shows a strong correlation between the confidence score $\hat{R}_i$ and the quality of the corresponding ROI: the confidence score is high on spine (i.e., good visual cue for registration) and low on soft tissue; when severe occlusion is presented due to medical devices, the occluded area has low confidence scores.

A threshold on the confidence score $\hat{R}_i$ is then used to only select agents with confidences above the threshold for further action aggregation. To determine the threshold, the behavior of the confidence score is analyzed on the validation data. Actions are categorized into correct and wrong, based on their impact on the registration (i.e., increase or decrease the distance to the ground truth). We choose a confidence threshold to make the correct rate of selected actions above 95%. To avoid the scenario that too few agents are selected for a given test image, if less than 10% of the agents meet this threshold, the top 10% agents will still be selected regardless of their confidence scores. Actions of the selected agents are then aggregated as described in Sect. 16.3.6 and applied iteratively during registration.

Table 16.2 Layer configurations for encoder/decoder CNNs and their equivalent dilated FCNs. Parameters for convolutional layers are written as $m \times n \times f$, where $n \times m$ is the convolution kernel size, and f is the number of feature maps. sk indicates that the layer has input stride k, and dk indicates that the filter kernel is dilated k times. All convolutional layers have zero padding. SELU activation function is applied after all layers except for the input and output layers. The column "Output size" specifies the output sizes for CNN

	Layer name	Output size	Single-agent CNN	Multi-agent FCN
Encoder	input	61×61	–	–
	conv1	59×59	$3 \times 3 \times 32$	$3 \times 3 \times 32$
	conv2	57×57	$3 \times 3 \times 32$	$3 \times 3 \times 32$
	conv3	27×27	$3 \times 3 \times 64$, s2	$3 \times 3 \times 64$, d2
	conv4	25×25	$3 \times 3 \times 64$	$3 \times 3 \times 64$, d2
	conv5	11×11	$3 \times 3 \times 128$, s2	$3 \times 3 \times 128$, d4
	conv6	9×9	$3 \times 3 \times 128$	$3 \times 3 \times 128$, d4
	conv7	3×3	$3 \times 3 \times 256$, s2	$3 \times 3 \times 256$, d8
	fc1	1×1	1024	$3 \times 3 \times 1024$, d8
	fc2	1×1	1024	$1 \times 1 \times 1024$, d8
	Output	1×1	128	$1 \times 1 \times 128$, d8
Decoder	input	1×1	–	–
	fc1	1×1	1024	$3 \times 3 \times 1024$, d8
	fc2	1×1	1024	$1 \times 1 \times 1024$, d8
	Output	1×1	6	$1 \times 1 \times 6$, d8

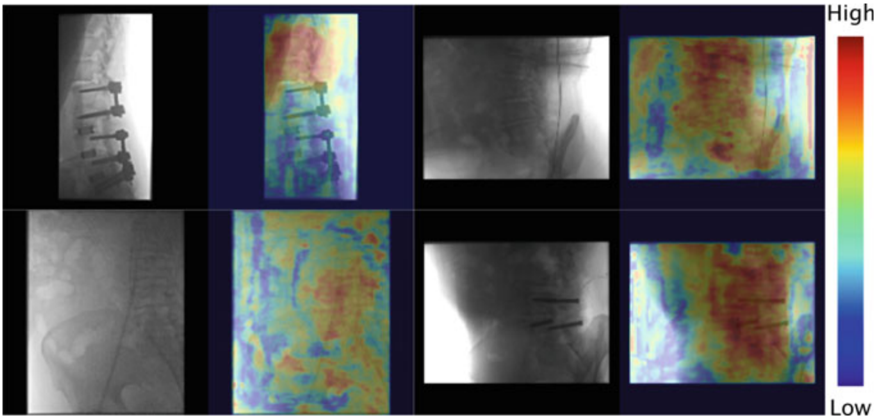


Fig. 16.7 Confidence map of densely overlapping ROIs within the image. Color of each pixel indicates the confidence value from the ROI centered on this pixel

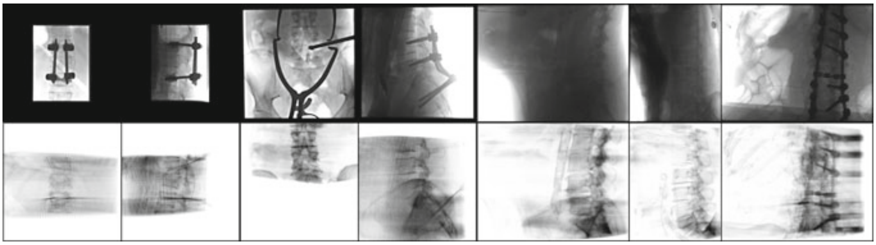


Fig. 16.8 Example X-ray images (top row) and DRRs (bottom row). The first four X-ray images are clinical data from spine surgery, which contain various highly opaque metal objects and have very different FOVs. The last three X-ray images are from CBCT data, which have a relatively low SNR due to a very small dynamic range

16.5.2 Experiments and Results

The agent-based 2-D/3-D registration is evaluated on a clinical application of 2-D/3-D registration during minimally invasive spine surgery, which aims at registering spine in 3-D CBCT and two X-ray images acquired from different angles. This is a challenging problem because surgical objects like screws and guide wires can be presented separately in the 3-D and 2-D images, creating severe image artifacts and occlusion of the target object (examples are shown in Fig. 16.8).

16.5.2.1 Experiment Setup

During minimally invasive spine surgery, the initial pose offset between the CBCT and the X-ray images can be up to 20 mm in translation and 10° in rotation. Therefore, to train the agents to perform registration starting from within this range, the X-ray/DRR pairs used for training have random rotation offset up to 10°, and translation offset up to 20 mm. The training data are generated from 77 CBCT data sets, where each CBCT data set consists of one CBCT and ~350 X-ray images used for reconstructing the CBCT. From each CBCT data set, we extract 350 X-ray/DRR pairs. Since the number of CBCTs is limited, we also generate pairs of synthetic X-ray image and DRR from 160 CTs as additional training data, where 200 pairs are generated from each CT. In total, the training data consist of 58,950 data, i.e., 26,950 CBCT data and 32,000 synthetic data.

Multi-agent training efficiency is analyzed using CNN and dilated FCN. In the training of CNN, random ROIs are extracted from the X-ray and DRR as the CNN input, and ground truth rewards are calculated and used as supervision. Curves for the training loss and correct action rate using CNN-based and dilated FCN-based training are shown in Fig. 16.9. FCN-based training finished in 17 hours, with a testing loss of ~0.13 and a testing correct action rate of ~90%. In comparison, CNN-based training after 17 hours reaches a test loss of ~0.22 and a testing correct action rate of ~80%, which is close to the performance of FCN-based method after 2 hours of training.

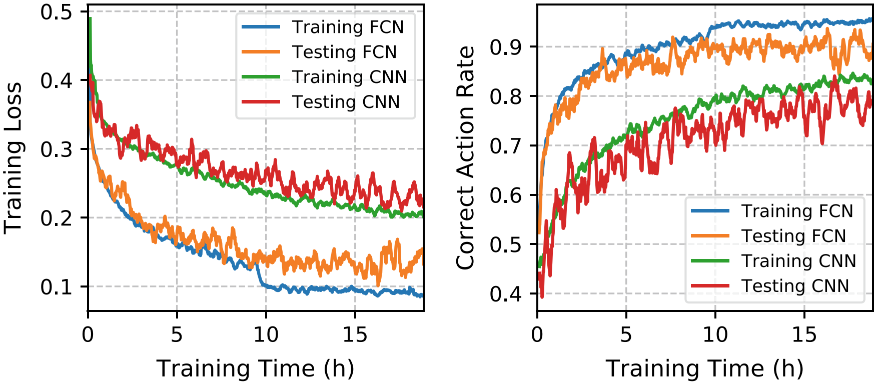


Fig. 16.9 Comparison of training speed using CNN-based training and dilated FCN-based training

Both the single-agent system and multi-agent system are evaluated for comparison, referred to as *agt-s* and *agt-m*, respectively. To apply agent-based 2-D/3-D registration with two X-ray images, in every step, one action is obtained from each X-ray image, and the obtained actions are applied sequentially. A combination of *agt-m* and an optimization-based method is also evaluated, referred to as *agt-m-opt*, where optimization of GC using BOBYQA optimizer is applied starting from the result of *agt-m*. The agent-based method is compared with popular optimization-based methods. Multiple similarity measures were evaluated in [3] using CMA-ES optimizer for spine 2-D/3-D registration, and GC and GO were reported to achieve the best performance. Therefore, CMA-ES optimization of GO and GC are evaluated as the baselines, referred to as *ES-GO* and *ES-GC*, respectively. Registration error is measured by TRE, calculated as the root mean square error (RMSE) of the locations of seven anatomical landmarks located on spine vertebrae.

Testing is first performed on 116 CBCT data sets via threefold cross-validation (77 used for training and 39 used for testing). The typical size of the CBCT data is $512 \times 512 \times 389$ with a pixel spacing of 0.486 mm. On each data set, 10 pairs of X-ray images that are >60° apart (common practice for spine surgery) are randomly selected, and 2-D/3-D registration is performed on each pair, starting from a perturbation of the ground truth transformation within 20 mm translation and 10° rotation, leading to 1,160 test cases.

To evaluate agent-based 2-D/3-D registration in a real clinical setup, one trained model is blindly selected from the threefold cross-validation on CBCT data and tested on 28 clinical data sets collected from minimally invasive spine surgery. Each data set contains a CBCT acquired before the surgery and two X-ray images acquired during the surgery. Ground truth registration is manually annotated by experts. On each data set, 20 perturbations of the ground truth transformation are randomly generated as starting positions for 2-D/3-D registration, leading to 560 test cases.

Table 16.3 Experiment results on bi-plane 2-D/3-D registration on 1,160 test cases from 116 CBCT data sets, and 560 test cases from 28 clinical data sets. Gross failure rate (GFR) accounts for test cases with TRE > 10 mm. Median, 75th percentile and 95th percentile TREs are reported

	Method	GFR (%)	Median (mm)	Percentile (mm)		Run time
				75%	95%	
CBCT data	Start	93.4	19.4	23.2	27.8	–
	ES-GC	32.2	1.67	22.1	44.1	18.7 s
	ES-GO	34.3	1.81	21.0	38.6	33.9 s
	agt-s	17.2	5.30	8.13	23.3	0.5 s
	agt-m	4.1	3.59	5.30	8.98	2.1 s
	agt-m-opt	**2.1**	**1.19**	**1.76**	**4.38**	2.6 s
Clinical data	Start	95.4%	20.4	23.1	26.8	–
	ES-GC	49.3	8.79	26.5	55.6	20.4 s
	ES-GO	42.1	3.18	29.0	84.6	35.8 s
	agt-s	45.7	8.42	14.2	26.4	0.6 s
	agt-m	6.8	4.97	7.36	**10.3**	2.1 s
	agt-m-opt	**6.1**	**1.99**	**2.76**	10.8	2.7 s

16.5.2.2 Results

Experiment results are summarized in Table 16.3. The two optimization-based methods, ES-GO and ES-GC, result in relatively high gross failure rate (account for TRE > 10 mm, which is about 1/2 of vertebrae and considered to be grossly off). This is mainly due to the low image quality (e.g., low SNR, image artifacts in CBCT, etc.), which leads to a highly non-convex optimization problem using low level similarity measures like CC and CC. In comparison, *agt-m* achieve a much lower gross failure rate, demonstrating the robustness advantage of the agent-based method. Comparison of *agt-s* and *agt-m* shows that the multi-agent strategy can noticeably improve robustness by aggregating information from most confident agents. The comparison of median TRE shows that while the agent-based method provides low failure rate, its accuracy is lower than that of optimization-based methods. This is primarily due to the discrete actions of 1 mm and 1°, and location information loss during stride in the Q-network. By applying *opt-local* to refine the result of *agt-m*, *apt-m-opt* achieved both low failure rate and high accuracy.

Experiment results on clinical data are summarized in Table 16.3. Higher TREs are reported for all methods on clinical data than that on the CBCT data, primarily due to three reasons: (1) The ground truth registration for clinical data is manually annotated, which could bear 1∼2 mm error. (2) The complexity of clinical data is much higher than the CBCT data (i.e., artifacts and occlusion caused by surgical devices, varying imaging FOVs, etc.). (3) For agent-based methods, the agent is trained without using any real clinical data from spine surgery. Due to the increased complexity, the heuristically selected ROI used in *agt-s* (i.e., center of the image) become even less reliable. As a result, the robustness of *agt-s* degrades significantly comparing to that on the CBCT data. The multi-agent system, *agt-m*, in contrast, achieves a much higher robustness than *agt-s*, even though the individual agent is trained without using any clinical data from spine surgery, demonstrating the effectiveness of the multi-agent strategy in dealing with complex scenarios.

16.6 Discussion

In this chapter, we have described the agent-based image registration method, and demonstrated its performance on two use cases, a rigid-body 3-D/3-D registration and a rigid-body 2-D/3-D registration. Although both use cases in this chapter are rigid-body registration, the method itself is a generic framework for registration that can be applied to other parametric registration problems as well, given an action space designed according to the parametric transformation space. For example, in [13], agent-based registration is adopted for nonrigid image registration using a statistical deformation model and achieves promising results. However, one challenge associated with more complex transformation model is the higher degree of freedom, which exponentially increases the size of the action space and hence the complexity of training. This can possibly be solved by decoupling the action space to sub-spaces and allowing independent policy learning (e.g., in B-Spline transformation, allow each control point to have its own action space, instead of having one action space that moves all control points.).

Comparing agent-based registration with its conventional optimization-based counterpart, the main advantage of agent-based method is the larger capture range and higher robustness. This is mainly because the policy is learned from observing de-aligned image pairs, allowing the policy to capture more complex rules than the maximization of a heuristically defined similarity metric used in conventional optimization-based methods. The capture range can be controlled by the offset sampling range used in training data generation. Specifically, to have a large capture range, image pairs with large registration offsets need to be used to train the network, and networks with stronger modeling power may be needed in order to capture the registration rule with increased complexity.

One limitation of the agent-based image registration method is the discrete action space, which imposes trade-offs between registration efficiency and accuracy. Using larger step sizes improves registration efficiency, since fewer actions and Q-network inferences are needed in order to complete the registration. However, since minimum movement of the agent is limited by the action step sizes, smaller step sizes are needed in order to achieve a high accuracy. One possible solution to achieve both high efficiency and accuracy is multi-scale action learning, where agents with different step sizes (from coarse to fine) are trained. During registration, they are applied hierarchically to perform coarse to fine registration. Similar method has been adopted in landmark detection to achieve real-time performance [6].

References

1. Besl PJ, McKay ND (1992) Method for registration of 3-d shapes. In: Robotics-DL tentative. International Society for Optics and Photonics, pp 586–606
2. Caicedo JC, Lazebnik S (2015) Active object localization with deep reinforcement learning. In: Proceedings of the IEEE international conference on computer vision, pp 2488–2496
3. De Silva T, Uneri A, Ketcha M, Reaungamornrat S, Kleinszig G, Vogt S, Aygun N, Lo S, Wolinsky J, Siewerdsen J (2016) 3d–2d image registration for target localization in spine surgery: investigation of similarity metrics providing robustness to content mismatch. Phys Med Biol 61(8):3009
4. Fischer P, Dosovitskiy A, Ilg E, Häusser P, Hazırbaş C, Golkov V, van der Smagt P, Cremers D, Brox T (2015) Flownet: learning optical flow with convolutional networks. arXiv:1504.06852
5. Ghesu FC, Georgescu B, Mansi T, Neumann D, Hornegger J, Comaniciu D (2016) An artificial agent for anatomical landmark detection in medical images. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 229–237
6. Ghesu FC, Georgescu B, Zheng Y, Grbic S, Maier A, Hornegger J, Comaniciu D (2019) Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans. IEEE Trans Pattern Anal Mach Intell 41(1):176–189
7. Hartley R, Aftab K, Trumpf J (2011) L1 rotation averaging using the weiszfeld algorithm. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3041–3048
8. Ibanez L, Schroeder W, Ng L, Cates J (2005) The itk software guide. Technical report, U.S. National Library of Medicine
9. James AP, Dasarathy BV (2014) Medical image fusion: a survey of the state of the art. Inf Fusion 19:4–19
10. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
11. Knaan D, Joskowicz L (2003) Effective intensity-based 2d/3d rigid registration between fluoroscopic x-ray and ct. In: Medical image computing and computer-assisted intervention-MICCAI 2003, pp 351–358
12. Kober J, Peters J (2010) Imitation and reinforcement learning. IEEE Robot Autom Magaz 17(2):55–62
13. Krebs J, Mansi T, Delingette H, Zhang L, Ghesu FC, Miao S, Maier AK, Ayache N, Liao R, Kamen A (2017) Robust non-rigid registration through agent-based action learning. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 344–352
14. Levine S, Koltun V (2013) Guided policy search. ICML 3:1–9
15. Liao R, Miao S, de Tournemire P, Grbic S, Kamen A, Mansi T, Comaniciu D (2017) An artificial agent for robust image registration. In: AAAI, pp 4168–4175
16. Liao R, Zhang L, Sun Y, Miao S, Chefd'Hotel C (2013) A review of recent advances in registration techniques applied to minimally invasive therapy. IEEE Trans Multimedia 15(5):983–1000
17. Maes F, Collignon A, Vandermeulen D, Marchal G, Suetens P (1997) Multimodality image registration by maximization of mutual information. IEEE Trans Med Imaging 16(2):187–198
18. Markelj P, Tomaževič D, Likar B, Pernuš F (2012) A review of 3d/2d registration methods for image-guided interventions. Med Image Anal 16(3):642–661
19. Miao S, Liao R, Pfister M, Zhang L, Ordy V (2013) System and method for 3-d/3-d registration between non-contrast-enhanced cbct and contrast-enhanced ct for abdominal aortic aneurysm stenting. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 380–387
20. Miao S, Piat S, Fischer P, Tuysuzoglu A, Mewes P, Mansi T, Liao R (2018) Dilated fcn for multi-agent 2d/3d medical image registration. In: AAAI, pp 4694–4701
21. Miao S, Wang ZJ, Liao R (2016) A cnn regression approach for real-time 2d/3d registration. IEEE Trans Med Imaging 35(5):1352–1363

22. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
23. Murphy K, Van Ginneken B, Reinhardt JM, Kabus S, Ding K, Deng X, Cao K, Du K, Christensen GE, Garcia V et al (2011) Evaluation of registration methods on thoracic ct: the empire10 challenge. IEEE Trans Med Imaging 30(11):1901–1920
24. Neumann D, Grbić S, John M, Navab N, Hornegger J, Ionasec R (2015) Probabilistic sparse matching for robust 3d/3d fusion in minimally invasive surgery. IEEE Trans Med Imaging 34(1):49–60
25. Oliveira FP, Tavares JMR (2014) Medical image registration: a review. Comput Methods Biomech Biomed Eng 17(2):73–93
26. Razlighi QR, Kehtarnavaz N, Yousefi S (2013) Evaluating similarity measures for brain image registration. J Vis Commun Image Represent 24(7):977–987
27. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489
28. Wu G, Kim MJ, Wang Q, Munsell B, Shen D (2015) Scalable high performance image registration framework by unsupervised deep feature representations learning

Chapter 17
Deep Learning for Functional Brain Connectivity: *Are We There Yet?*

Harish RaviPrakash, Arjun Watane, Sachin Jambawalikar and Ulas Bagci

Abstract The detection of behavioral disorders rooted in neurological structure and function is an important research goal for neuroimaging communities. Recently, deep learning has been used successfully in diagnosis and segmentation applications using anatomical magnetic resonance imaging (MRI). One of the reasons for its popularity is that with repeated nonlinear transformations, the algorithm is capable of learning complex patterns in the data. Another advantage is that the feature selection step commonly used with machine learning algorithms in neuroimaging applications is eliminated which could lead to less bias in the result. However, there has been little progress in the application of these black-box approaches to functional MRI (fMRI). In this study, we explore the use of deep learning methods in comparison with conventional machine learning classifiers as well as their ensembles to analyze fMRI scans. We compare the benefits of deep learning against an ensemble of classical machine learning classifiers with a suitable feature selection strategy. Specifically, we focus on a clinically important problem of *Attention Deficit Hyperactivity Disorder (ADHD)*. Functional connectivity information is extracted from fMRI scans of ADHD and control patients (ADHD-200), and analysis is performed by applying a decision fusion of various classifiers—the support vector machine, support vector regression, elastic net, and random forest. We selectively include features by a nonparametric ranking method for feature selection. After initial classification is per-

H. RaviPrakash (✉) · U. Bagci
Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA
e-mail: harishr@knights.ucf.edu

U. Bagci
e-mail: bagci@crcv.ucf.edu

A. Watane
Miller School of Medicine, University of Miami, Miami, FL, USA
e-mail: awatane@med.miami.edu

S. Jambawalikar
Department of Radiology, Columbia University Medical Center, New York, NY, USA
e-mail: sj2532@cumc.columbia.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_17

formed, the decisions are summed in various permutations for an ensemble classifier, and the final results are compared with the deep learning-based results. We achieved a maximum accuracy of 93.93% on the KKI dataset (a subset of the ADHD-200) and also identified significantly different connections in the brain between ADHD and control subjects. In the blind testing with different subsets of the target data (Peking-1), we achieved a maximum accuracy of 72.9%. In contrast, the deep learning-based approaches yielded a maximum accuracy of 70.5% on the Peking-1 dataset and 67.74% on the complete ADHD-200 dataset, significantly inferior to the classifier ensemble approach. With more data being made publicly available, deep learning in fMRI may show a strong potential but as of now deep learning does not provide a magical solution for fMRI-based diagnosis.

17.1 Introduction

Attention Deficit Hyperactivity Disorder (ADHD) is one of the most common neurological brain disorders in children, affecting approximately 5.4 million children in the United States alone [1]. Diagnosed children may suffer from learning difficulties, behavioral abnormalities, disobedience or aggression toward authority. Its effects may be detrimental to their health, education, and social skills [2]. Their cognitive impulsivity and emotional liability may indicate a greater likelihood of the progression to adult antisocial behavior and violent impulse-control disorders [3].

Recently, there has been a lot of effort to discover the underlying cause of this disorder; however, at present there is no standardized biological measure to diagnose ADHD. Instead, physicians and psychologists still rely on behavioral symptoms reported by parents and teachers to aid in identifying the disorder. ADHD screening includes subjective behavioral observations of inattention, impulsiveness, and hyperactivity. When asked to identify these symptoms, however, a person may be subject to confirmation bias, which is the tendency to interpret any evidence as a confirmation of one's belief. For example, a mother or teacher may identify a student with poor grades as "distracted". As a result, the diagnosis may be inaccurate, especially if a parent or teacher believes the child has ADHD prior to examination. This may lead to inaccurate or overdiagnosis of the disorder [4]. To develop an objective and effective diagnostic method for ADHD, scientists collect and analyze genetic, imaging, physiologic, and cognitive data. Among them, radiologic imaging of brain structure and function is promising due to the non-invasive nature of the imaging. Additionally, the availability of automated machine learning methods revealing imaging and brain connectivity-based features that identify ADHD subjects and/or affected brain regions to some extent, complements the radiological imaging approach.

An imaging modality that has potential to identify neural patterns is Functional Magnetic Resonance Imaging (fMRI), which measures signal changes in the brain due to neural activity. During increases of neural activity, there is an increased demand for oxygen in the localized neurological area. The vascular system compensates for this by increasing the amount of oxygenated blood in the area. fMRI measures the ratio of oxygenated hemoglobin versus deoxygenated hemoglobin to create a functional activity measurement over time. This data can then be utilized to compare functional brain activity differences between normal versus neuropathological subjects [5].

Artificial intelligence and deep learning are the buzzwords in the computer vision industry just like cryptocurrency to online transactions. The popularity of these approaches has spread at the rate of knots and they have found application in almost every field, from computer science [6] to chemical engineering [7], civil engineering [8] to medicine [9]. With the high success rate of deep learning, the expectation of its contribution to fMRI data is high. However, when dealing with more challenging tasks, simple or complex deep learning networks are unable to generate satisfactory results on their own. They are generally combined with other classical statistical methods, machine learning paradigms or with other deep neural networks. One such task is the goal of mastering the game of chess. DeepMind recently proposed an algorithm [10] that combines deep learning and Monte Carlo tree search to learn the game of chess. Similarly, in the segmentation of pancreas from MRI images, Cai et al. proposed to use a graph-based decision fusion to combine the results of different deep learning algorithms [11]. fMRI based diagnosis/prediction is another such challenging task. To solve this problem, following along the lines of Silver et al. of combining different machine learning paradigms [10], in this chapter, we present an ensemble classifier approach to boost diagnostic accuracy by identifying specific functional differences in ADHD brains using fMRI connectivity information. We also compare and contrast the proposed approach to standard deep learning models applied on the same imaging data to observe whether standard deep learning approaches are as of yet capable of tackling the ADHD diagnosis problem using fMRI.

17.2 Related Work

Feature selection and dimensionality reduction have been popular in image-based classification. They have been used for both cognitive state and disease classification [12, 13]. Machine learning and network graph analysis approaches have been used extensively in resting-state fMRI-based disease diagnosis problems. In resting-state fMRI analysis of Alzheimer's patients, for instance, network graph theory was shown to be promising [14, 15]. Machine learning classifiers were used effectively in classification of autism [16] and ADHD patients [17], as well as in identifying biomarkers for amyotrophic lateral sclerosis (ALS) [18]. Chu et al. studied the importance of feature selection and sample size on classification using anatomical MR images [19]. Authors found that with increasing sample size, feature selection plays a lesser role and the impact of feature selection is dependent on the classifier chosen.

The classifier ensemble approach has been shown to achieve more accurate results compared to a single classifier. Cabral et al. found that a group of classifiers is able to decode visual stimuli from fMRI data at a higher accuracy than when using an individual classifier alone [20]. Furthermore, Richiardi et al. used decision tree ensembles for brain decoding of fMRI connectivity [21]. In a study by Kuncheva and Rodriguez, various brain patterns associated with different visual stimuli were identified by using a large number of ensembles [22].

Deep Learning Approaches

In recent years, deep learning has gained popularity due to its ability to generate high-level feature representations of the data. In medical imaging, deep learning has been met with great successes in several different tasks. For instance, segmentation of the left atrium in cardiac images [23], lung in CT images [24], anatomical brain regions in MRI [25], mandible in CT skull images [26], deep learning was shown to achieve state-of-the-art results. Deep learning has also received remarkably accurate results for detection and classification tasks such as IPMN classification in pancreatic cancer [27], lung nodule detection [28], disease classification in brain MRI [29]. While deep learning with brain images has received great attention, most works have used MRI [29, 30] and diffusion imaging [31, 32].

The application of fMRI with deep learning approaches has also begun to receive more attention in the past few years. Wen et al. used Convolutional Neural Networks (CNN) to identify the association between visual tasks and neural activity using fMRI recorded data [33]. Horikawa and Kamitani performed object recognition tasks using fMRI recordings and deep neural networks [34]. Toward a better diagnosis model, Li et al. proposed using the fMRI images as input to a 3D convolutional network to classify [35] and identify biomarkers [36] in Autism Spectrum Disorder data. In a study on Alzheimer's data, Yan et al. used dynamic functional connectivity matrices as input to a bidirectional *Long Short-Term Memory network* for mild cognitive impairment diagnosis [37]. A sliding window approach is used to generate the mean fMRI image in a study by Li et al. [35] and dynamic functional connectivity in a study by Yan et al. [37]. However, at present, there is no known optimal method to identify the best window length and stride for the sliding window approach. These studies use empirically determined parameters, mostly based on previous works.

There has been limited work in directly comparing deep learning and classical machine learning algorithms. Sabuncu et al. in their recent work, compared deep learning to kernel regression in the task of fluid intelligence prediction and found that deep learning did not outperform the classical regression approach but had strong future potential [38]. Along similar lines, we propose to compare the deep learning and standard classifiers in the disease diagnosis task. We use a generic feed-forward 1D convolutional network, a standard auto-encoder, and a recently published network designed to work with connectome data.

17.3 Methods

In this study, we test the effect of feature selection on the sample size using ADHD fMRI data and a novel ensemble classifier, thereby negating the impact of classifier-dependent feature selection boost. The ensembles that were used include combinations of Support Vector Machines (SVM), Support Vector Regression (SVR), Elastic Nets (EN), and Random Forest (RF) decision trees. For SVR, the final classification is based on a threshold on the regression output.

The ADHD-200 dataset [39] consists of multi-site data with patients comprising typically developing children and ADHD-diagnosed children. The demographic details of the data used in this study can be found in Table 17.1.

17.3.1 fMRI Preprocessing and Feature Extraction

The recorded fMRI data must be preprocessed in order to minimize inter-scan differences and center the focus of the analysis on only the relevant structures. For all our fMRI experiments, we use the preprocessed fMRI data released by the ADHD-200 competition organizers. The preprocessing is done using the AFNI [40] and FSL [41] tools and computed on the Athena computer clusters. The Athena functional data preprocessing pipeline includes (1) removal of the first four EPI volumes; (2) slice timing correction; (3) deobliquing of the dataset; (4) reorientation; (5) motion correction to the first image of the time series; (6) masking to exclude non-cortical structures (skull and neck stripping); (7) averaging of the volumes to create a mean image; (8) co-registering the fMRI to its corresponding *T1* image; (9) writing fMRI data and mean image into a template space; (10) down-sampling the *WM* and *CSF* masks (from the anatomical preprocessing that occurs in parallel but not used for our experiments); (11) time-course extraction for the *WM* and *CSF*; (12) regressing

Table 17.1 ADHD-200 dataset

Site	ADHD		Control	
	# Subjects	Age (Mean ± Std.)	# Subjects	Age (Mean ± Std.)
KKI	22	10.25 ± 1.27	61	10.21 ± 1.56
NeuroIMAGE	25	17.32 ± 2.57	23	16.68 ± 2.90
NYU	118	12.16 ± 3.14	98	11.26 ± 2.66
OHSU	37	8.88 ± 1.21	42	8.77 ± 1.03
Peking1	24	11.19 ± 1.62	61	11.26 ± 2.31
Peking2	35	11.62 ± 1.79	32	12.58 ± 1.76
Peking3	19	13.20 ± 0.95	23	13.28 ± 1.35
Total	280	11.61 ± 2.88	340	11.64 ± 2.97

out *WM*, *CSF*, and motion time courses; (13) band-pass filtering voxel timecourses to exclude frequencies not implicated in functional connectivity; and (14) blurring of the filtered and unfiltered data using a 6-mm FWHM Gaussian filter.

For feature extraction, a toolbox provided by University College London, known as Statistical Parametric Mapping (SPM), is used in the analysis of the brain fMRI data sequences. Since the data has already been preprocessed to control for any unwanted variability and unnecessary structures within the scans, SPM is used to extract features. The chosen feature extraction method is a region of interest (ROI) correlation matrix consisting of the correlation coefficients of the activity between structures of the brain. The automated anatomical labeling (AAL) atlas is used to parcellate the volumes into 116 substructures, and the correlation coefficients between the structures' level of activity are calculated. From these connectivity matrices, the upper right triangle is extracted, and there are 6670 correlations which serve as the features used in the classification of ADHD versus non-ADHD subjects.

17.3.2 Ensemble Classification Approach

Figure 17.1 represents the proposed pipeline for ADHD classification. The input to this pipeline is the preprocessed fMRI ROI connectivity matrix, i.e., functional correlation between pairs of brain regions. First, the upper right triangle of the fMRI connectivity matrices is extracted. Then, these features, the ROI functional correlation features, are ranked. Next, the top *n* features are selected for classification. A tenfold cross-validation approach is performed on all subjects. Finally, the ensemble classifier outputs the final predicted diagnosis of the test subjects.

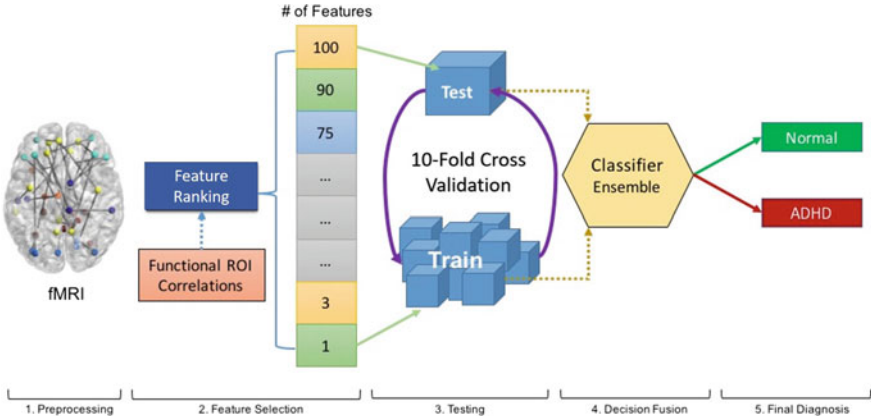


Fig. 17.1 The overall ensemble classification pipeline

Feature Selection

A fundamental problem in machine learning in medical imaging applications is that of redundant information. That is, the dimensionality of the data (the number of pixels and voxels) far outnumbers the amount of data that is helpful in identifying a specific class such as a disorder or disease. Therefore, feature selection algorithms have been optimized in order to improve classification performance by throwing out non-informative features and including only significant features as part of the training and testing data.

Recent studies have shown that in a cluster-wise analysis of fMRI, there is almost a 70% false-positive rate [42]. Thus, identifying valuable features are important in order to improve specificity and sensitivity, and ultimately, the accuracy performance. Especially when considering the complexities and commonalities between brains, many redundant and extraneous features are reported. As a result, they usually diminish the performance of a classifier. Thus, an important component of our method is to utilize features that are significantly different between ADHD and Control subjects. We utilize the Wilcoxon rank sum t-test to identify significant features for ranking.

Wilcoxon Rank Sum t-Test

In order to rank the features, the Wilcoxon rank sum t-test is performed on each feature of the ADHD set versus the Control set. The Wilcoxon rank sum t-test is a nonparametric statistical method, and it is ideal in identifying when a feature is significantly different between the populations, because which means that it does not assume that the populations are normally distributed. Based on the z-score that is produced by Eq. 17.1, the p-value is calculated. Normally, any feature with p-value less than 0.05 is considered significant. In this study, numerous tests are conducted by ranking the features by p-value from lowest to highest and then selecting a varied amount of the top-*n* significant features for each experiment:

$$z = \frac{T_1 - \frac{n_1(n_1+n_2+1)}{2}}{\sqrt{\frac{n_1 n_2 (n_1+n_2+1)}{12}}}, \tag{17.1}$$

where T_1 is the test statistic, and n_1, n_2 are the sample sizes.

17.3.3 Deep Learning Models

We adopt *brainNetCNN* [32], which has been previously used with structural connectivity matrices, and train the model with functional connectivity matrices. In *brainNetCNN*, three new layers were introduced:

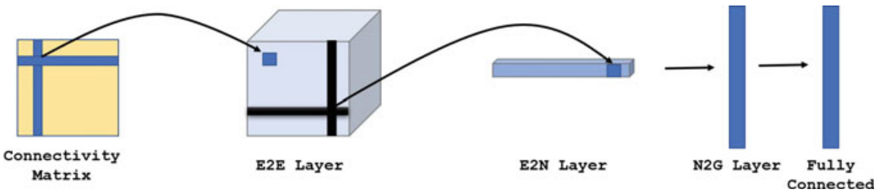


Fig. 17.2 *brainNetCNN* network for ADHD diagnosis. E2E: edge-to-edge, E2N: edge-to-node, and N2G: node-to-graph

Table 17.2 Network design parameters

BrainNetCNN	
Layer	**Filter Size**
2D Conv (E2E)	16
2D Conv (E2N)	64
2D Conv (N2G)	64
Dense	2
1DConvNet	
Layer	**Filter Size**
1D Conv	32, 64, 64, 64
Dense	64, 2
Auto-encoder	
Layer	**Filter Size**
1D Conv	1024, 512, 1024, 6670

Global Hyperparameters	**Value**
BrainNetCNN & 1DConvNet	
Loss	Binary Cross-Entropy
Optimizer	SGD
Learning Rate	0.01
Dropout	0.5
Learning Rate decay	0.0001
Auto-encoder	
Loss	Mean Squared Error
Optimizer	SGD
Learning Rate	0.2
Learning Rate decay	0.0001

i. Edge-to-edge (E2E) layer: Generates a filtered adjacency matrix by combining edges between shared nodes. This works similar to a convolutional layer with a cross-shaped filter.
ii. Edge-to-node (E2N) layer: Unlike the E2E layer, here, 1D spatial convolutions are applied separately row-wise and column-wise, and then combined to generate a vector of node responses.
iii. Node-to-graph (N2G) layer: Generates a weighted combination of nodes by performing 1D spatial convolutions.

We modify this architecture to perform classification instead of regression, which was the task in the original work. The network architecture is shown in Fig. 17.2 and design parameters can be found in Table 17.2.

We also test the performance of a 1D convolutional filters-based architecture, 1DConvNet, for ADHD diagnosis using the 6670 dimension connectivity feature vector as input. The network is designed with four 1D convolutional layers and two fully connected (Dense) layers. The network architecture is shown in Fig. 17.3 and design parameters can be found in Table 17.2.

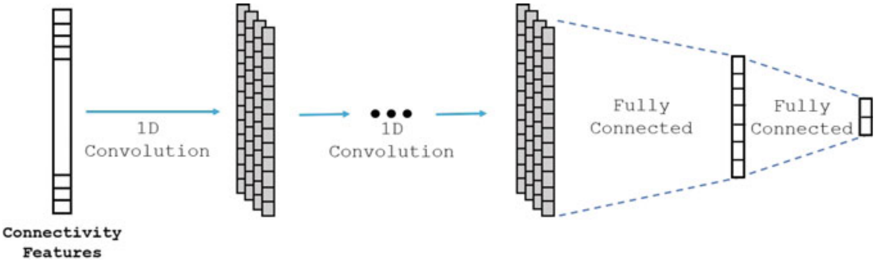


Fig. 17.3 *1DConvNet* architecture for ADHD diagnosis

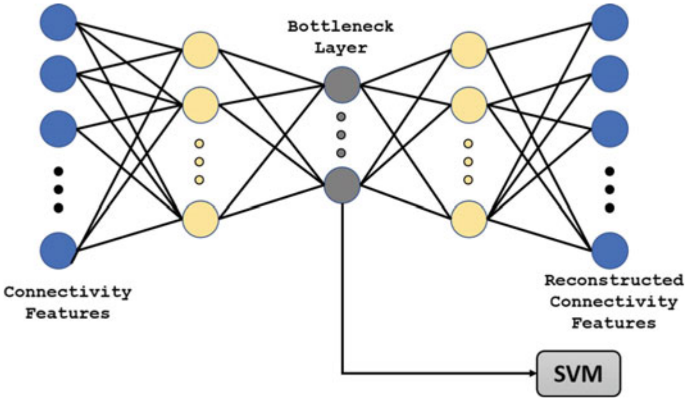


Fig. 17.4 *Auto-encoder* network to reconstruct connectivity features. Features from bottleneck layer fed to SVM for classification

Finally, we also test the popular auto-encoder architecture. This can be described as consecutive Dense layers with the objective being to reconstruct the input. The bottleneck layer is used as the feature representation and SVM is applied to perform classification. The network architecture is shown in Fig. 17.4 and design parameters can be found in Table 17.2. Two different training strategies are employed:

($\mathbf{E}_1$) Models are trained using all available data, i.e., multi-site data, and tenfold cross-validation is performed (no distinction of data from site A, site B, etc.).

($\mathbf{E}_2$) Models are trained using all data sites except one, which is used as a test set. In this approach, a tenfold cross-validation is applied to the training data and the trained model from each fold is saved. The final classification on test data is based on a majority voting of the ensemble of trained models, i.e., majority decision of the 10 trained models.

The abovementioned training strategies are employed to test the power of the model (E_1) and generalization capability of the model (E_2).

17.4 Results

17.4.1 Independent Classifiers

The Wilcoxon rank sum t-test is used to compare each connectivity correlations' difference in distributions between the ADHD and Control subject sets. There are several significantly different (p-value < 0.05) fMRI features amongst the ADHD and Control sets. While there are nearly 400 activity correlation coefficients between structures' that reported a p-value < 0.05, Table 17.3 reports the top 10 significant features from the fMRI data.

The independent classifier results for KKI dataset and NYU datasets are graphed in Figs. 17.5 and 17.6. In KKI, *SVM*/*SVR* does better than *EN* while in NYU, *EN* does better than *SVM*. These observations show that there is no single classifier that is superior in diagnosis of ADHD, and suggest a need for an ensemble of classifiers to collectively output diagnoses.

17.4.2 Ensemble Classifiers

Between a total of four classifiers and keeping the feature selection method results independent of each other, various combinations of majority voting predictions are evaluated. Since there are a large number of possible combinations (24), the combinations are incrementally and intuitively decided to save time and to understand if combining different classifier predictions improves performance. It is observed that the performance of *SVM* and *SVR* are comparable across all datasets. Hence, only

Table 17.3 Top-10 functional connectivity regions contributing to classification/diagnosis. L: left, R: right

fMRI features			
ROI	Hemi	Hemi	ROI
Superior parietal	R.	L.	Orbital middle frontal gyrus
Rolandic operculum	R.	L.	Cerebellum crus II
Rolandic operculum	R.	L.	Cerebellum lobule IV, V
Insula	R.	L.	Lingual gyrus
Cuneus	R.	L.	Fusiform gyrus
Superior occipital	L.	R.	Cerebellum crus II
Inferior parietal	L.	L.	Supramarginal gyrus
Precuneus	L.	L.	Inferior temporal gyrus
Middle temporal gyrus	L.		Vermis lobule 9
Precuneus	L.		Vermis lobule 7

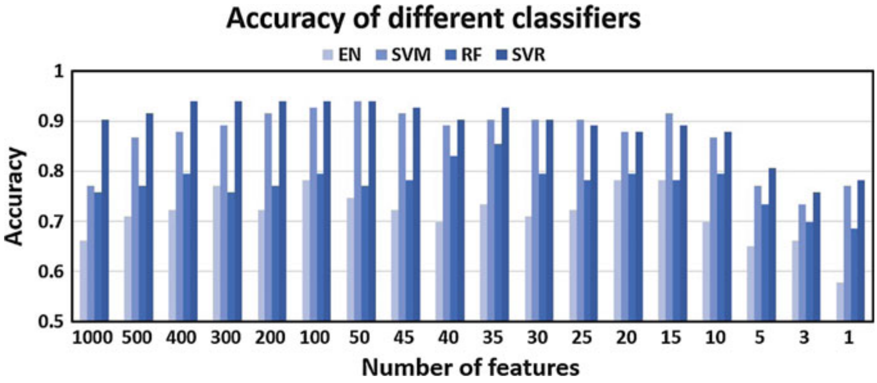


Fig. 17.5 Independent classifiers on fMRI features selected through the Wilcoxon rank sum t-test on KKI dataset. Maximum accuracy is 93.97% from the SVR classifier

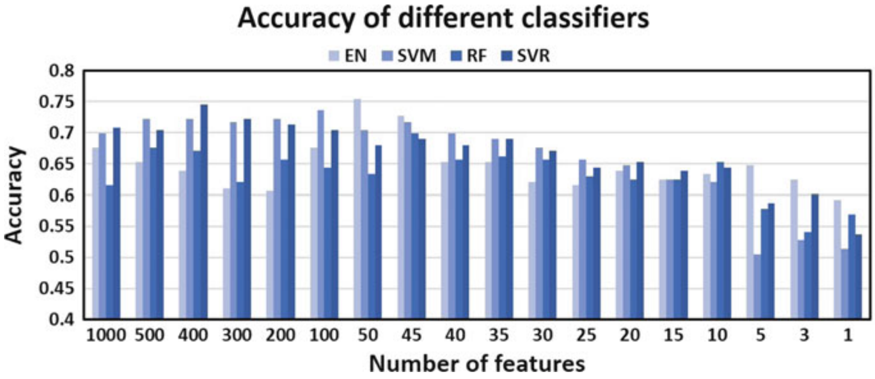


Fig. 17.6 Independent classifiers on fMRI features selected through the Wilcoxon rank sum t-test on NYU dataset. Maximum accuracy is 75.290 from the *EN* and *RF* classifiers

combinations involving *SVM* are tested, and *SVR* is used only for the final ensemble classifier that combined *EN*, *SVM*, *RF*, and *SVR*. In Fig. 17.7, the performance of the different classifier combinations can be seen. The results for these two datasets, where the best performing classifiers are different, show the benefit of combining the different classifiers to create an ensemble classifier.

Figure 17.8 shows the results of the *ensemble classifier* approach using the Wilcoxon rank sum feature ranking approach for the different datasets. For all datasets, with the top-50 ranked features, i.e., network connections, selected, a classification accuracy of 70% or more is achieved.

In Fig. 17.9, the top significant connections identified in Table 17.3 are mapped and visualized using BrainNet Viewer [43]. Few of the identified regions have been identified in previous works. Increased spikes in the rolandic region have been previously identified in ADHD patients [44]. Decreased activations in cerebellar regions [45]

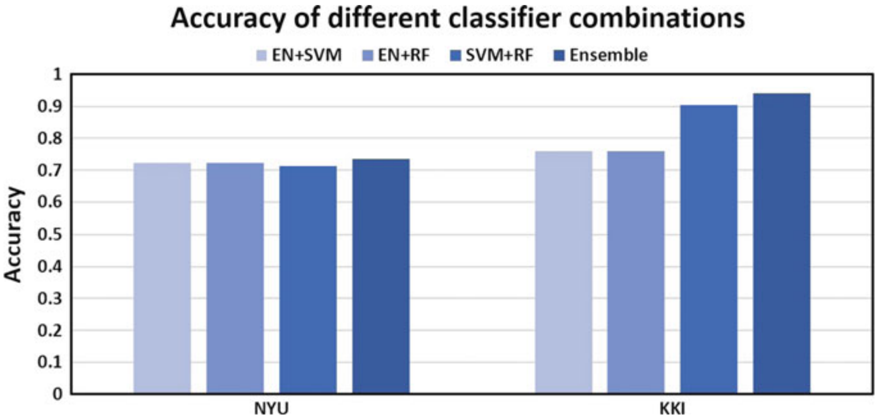


Fig. 17.7 Different combinations of classifiers tested on the NYU and KKI datasets

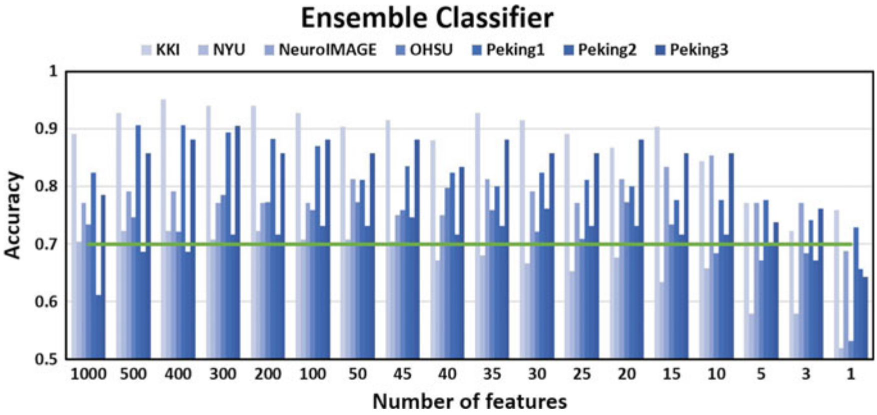


Fig. 17.8 Accuracy of the proposed *ensemble classifier* approach with varying number of ranked features selected, on the individual datasets of the ADHD-200 dataset

as well as structural changes in the vermis region of the cerebellum [46] have been found in ADHD patients. Decreased activation was also found in the bilateral temporal cortex [45, 47]. Structural deficits and decreased functional activations were found in different studies [48, 49].

17.4.3 Deep Learning Classifiers

BrainNetCNN, *1DConvNet*, and *Auto-Encoder* models are trained to convergence for both training strategies E_1 and E_2. For the Auto-encoder architecture, the *SVM* is trained using the Radial basis function. The performance of these different models is graphed in Fig. 17.10a, b. The precision and sensitivity of the models for E_1,

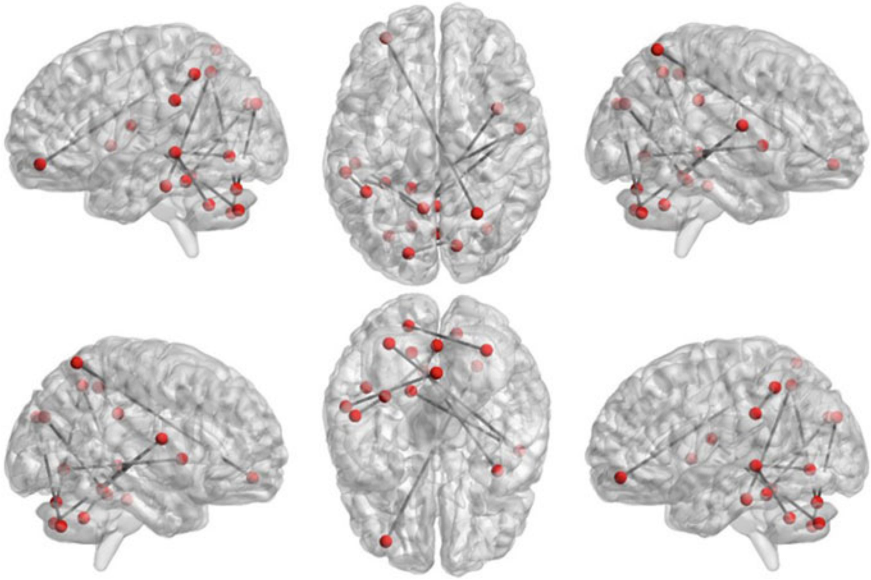


Fig. 17.9 The significant functional connection correlation features listed in Table 17.3 for KKI dataset

represented as “ALL” in Fig. 17.10a, b show that the *1DConvNet* architecture works best. For E_2, *BrainNetCNN* has better precision and sensitivity in comparison to the other deep learning models.

BrainNetCNN is able to better learn the topological structure in the connectivity matrices. *1DConvNet* could perform better with fine-tuning for the network with part of the testing-site dataset. However, in comparison to the proposed ensemble approach, the performance is considerably lower.

To directly compare the generalization capabilities of deep learning approaches against the proposed *ensemble classifier* approach, we test their performance on the multi-site data classification task where the models are trained on data from all but one site and tested on the remaining site. The results are shown in Fig. 17.11. As can be seen from Fig. 17.11a, b, neither the deep learning-based approaches nor the *ensemble classifier* has high sensitivity and specificity. However, the *ensemble classifier* performs better than all the deep learning approaches.

To further complete the analysis, we compare the deep learning algorithms against the *ensemble classifier* when more data is available, i.e., the all data task. As seen in Fig. 17.12, the *ensemble classifier* has a balanced performance with over 60% sensitivity, specificity, and accuracy. However, the *1DConvNet* outperforms the *ensemble classifier* in both sensitivity and accuracy. This shows the promise of deep learning algorithms in fMRI based disease diagnosis with the availability of larger datasets. The limitation of classical classifiers with mixed site data is also observable.

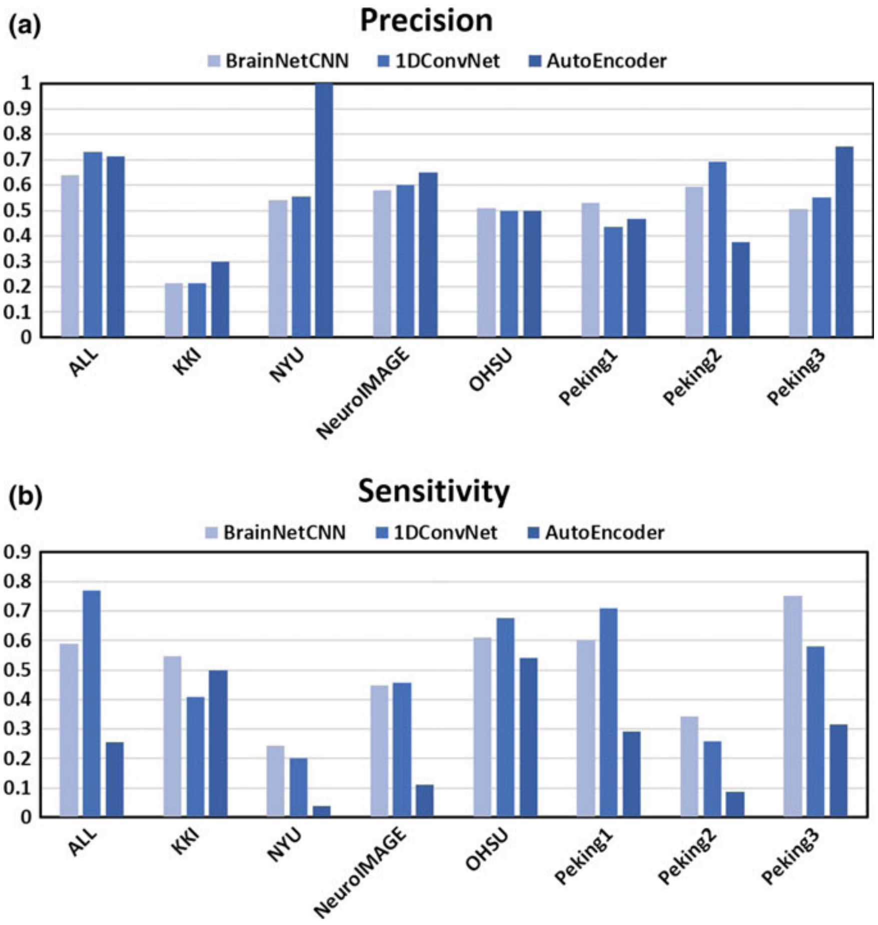


Fig. 17.10 **a** Precision and **b** sensitivity of *BrainNetCNN*, *1DConvNet*, and *auto-encoder* algorithms at all data and multi-site data classification

17.5 Discussion and Conclusions

In this study, we tested different standard deep learning approaches to the fMRI based ADHD diagnosis task and compared the results to our proposed feature selection and *ensemble classifier* approach. The best performing deep learning-based approach was *1DConvNet* with an accuracy of 67.7% and sensitivity of 76.9% on the E_1 training strategy. This compares favorably against the classical classifiers based ensemble approach which had an accuracy of 60.6% and sensitivity of 67.3%. However, when testing the generalizability of the approaches to unseen data from different datasets, the proposed ensemble classifier approach outperformed the deep learning approaches.

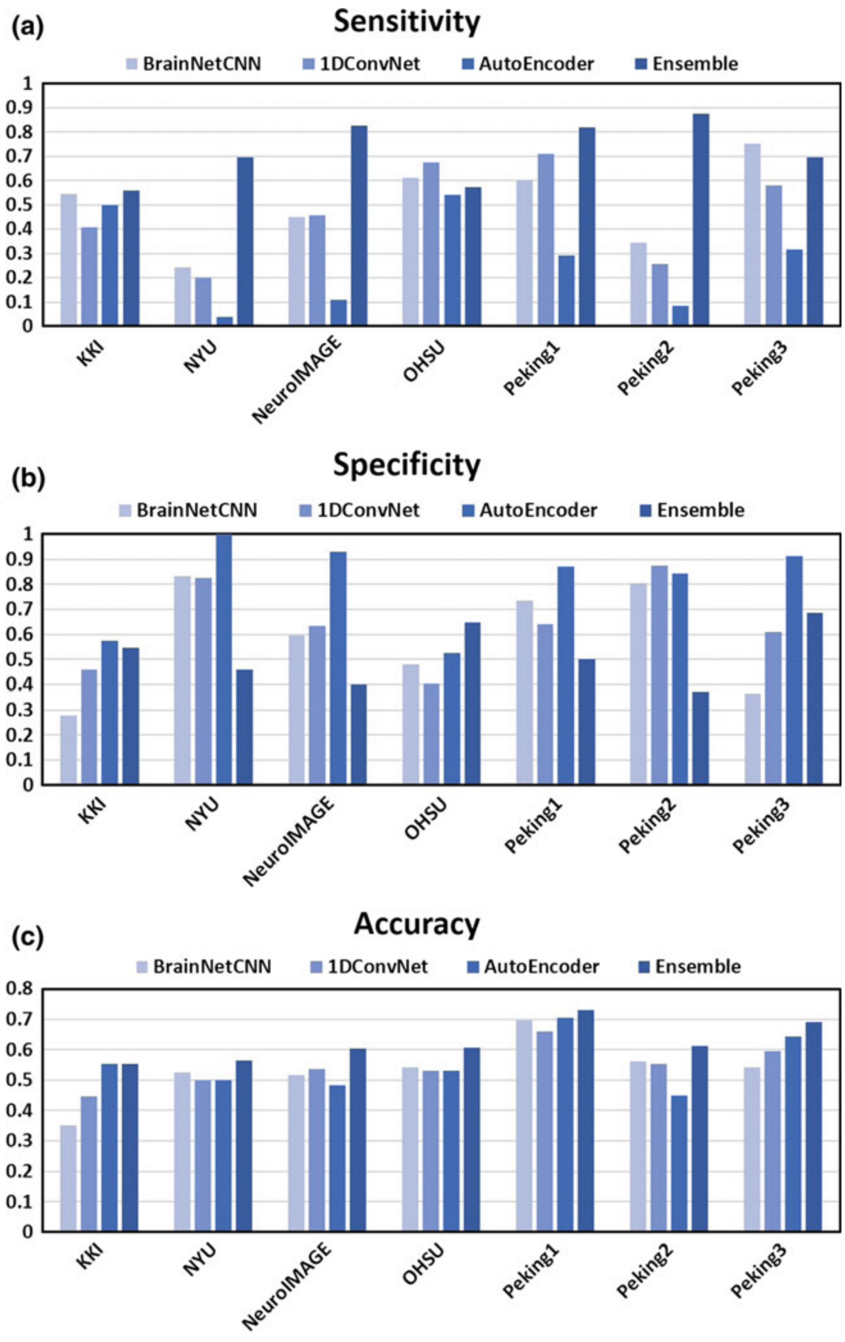


Fig. 17.11 **a** Sensitivity, **b** specificity, and **c** accuracy of *BrainNetCNN*, *1DConvNet*, and *autoencoder* and proposed *ensemble classifier* algorithms for multi-site data classification

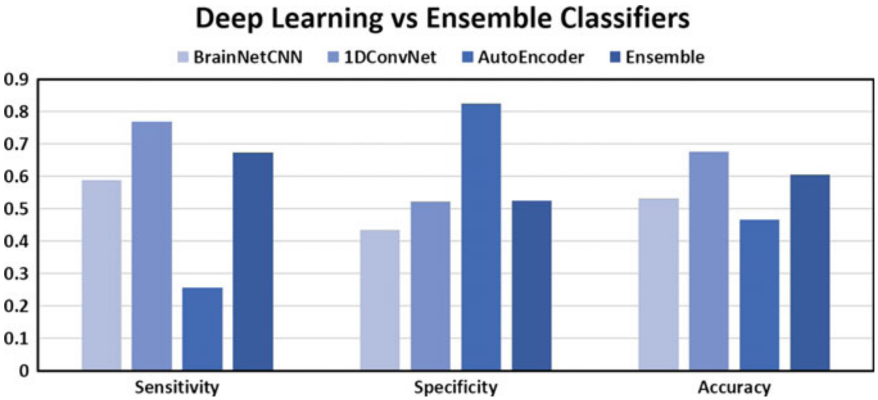


Fig. 17.12 Comparison of *BrainNetCNN*, *1DConvNet*, and *auto-encoder* and *ensemble classifier* algorithms at all data classification task

With the *ensemble classifier* approach, single datasets can be easily analyzed and classified. This is not possible with the deep learning approaches as it needs more than the few hundred input matrices that each dataset has to train to convergence without overfitting. These results indicate that the deep learning algorithms need a lot more data to outperform classical classifiers. Data augmentation in the form of generating more realistic and similar fMRI data using generative adversarial networks [50] is a possible approach to overcome this lack of data problem.

Another challenge is the interpretability task. With the *ensemble classifier* approach, it is a relatively easy task to trace back to the top features used and these can then be used as potential biomarkers. With deep learning approaches, the interpretability becomes more challenging with deeper networks. Standard convolutional layers make use of spatial locality with their filter design, but with a connectivity matrix, this might be hard to interpret. Use of heat maps is a common approach and can be incorporated into these networks to identify the contributing functional connections.

A single fMRI scan comprises spatiotemporal information of millions of voxels. This very large amount of data within a single scan makes analysis very challenging even for deep learning methods, since they run the risk of overfitting to the limited scans available. A solution to this problem is the generation of functional connectivity matrices but this leads to loss of spatiotemporal information which could mean a loss of deterministic complex patterns that deep learning algorithms are very good at identifying.

Additionally, the static functional connectivity may not capture the true signal characteristics which might be needed by the deep networks to converge to an optimal solution. Dynamic functional connectivity-based approach is a possible direction as shown by Yan et al. [37]. However, the lack of a standardized approach toward generation of the optimal windows can affect the performance. Alternately, the time-series data can be used directly with long short-term memory modules to model the signal characteristics from the different brain regions.

Though multi-site data serves to boost the amount of available data, it brings with it another problem which is of different noise levels. At each site, with different scanners and protocols used, the noise in the recording is different. These different noise levels, when mixed (i.e., multi-site data), could add confusion to the interpretation of the connectivity matrices.

With larger publicly available datasets like the *Human Connectome Project* (HCP) [51] and *OpenNeuro* [52], deep networks will become more feasible and with use of additional phenotypic features, the application to disease diagnosis is a viable path. Li et al. showed the potential of using the fMRI images, despite their low resolution, for biomarker identification in autism [36] and this could be a potential direction. We have tested a few of the basic deep learning algorithms, but with different network configurations and inputs (4D image, time-series signals, dynamic functional connectivity matrix), deep learning could potentially improve diagnosis. However, the deep learning methodology is not yet the gold-standard approach toward stand-alone modality fMRI classification.

References

1. Danielson ML, Bitsko RH, Ghandour RM, Holbrook JR, Kogan MD, Blumberg SJ (2018) Prevalence of parent-reported ADHD diagnosis and associated treatment among US children and adolescents, 2016. J Clin Child Adolesc Psychol 47(2):199–212
2. Biederman J (2005) Attention-deficit/hyperactivity disorder: a selective overview. Biol Psychiatry 57(11):1215–1220
3. McKay KE, Halperin JM (2001) ADHD, aggression, and antisocial behavior across the lifespan. Ann N Y Acad Sci 931(1):84–96
4. Merten EC, Cwik JC, Margraf J, Schneider S (2017) Overdiagnosis of mental disorders in children and adolescents (in developed countries). Child Adolesc Psychiatry Ment Health 11(1):5
5. Matthews PM, Honey GD, Bullmore ET (2006) Neuroimaging: applications of fMRI in translational medicine and clinical practice. Nat Rev Neurosci 7(9):732
6. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. Advances in neural information processing systems, pp 3856–3866
7. Spencer M, Eickholt J, Cheng J (2015) A deep learning network approach to ab initio protein secondary structure prediction. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 12(1):103–112
8. Cha YJ, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Infrastruct Eng 32(5):361–378
9. Khosravan N, Celik H, Turkbey B, Jones EC, Wood B, Bagci U (2019) A collaborative computer aided diagnosis (C-CAD) system with eye-tracking, sparse attentional model, and deep learning. Med Image Anal 51:101–115
10. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lillicrap T (2017) Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815
11. Cai J, Lu L, Xie Y, Xing F, Yang L (2017) Pancreas segmentation in MRI using graph-based decision fusion on convolutional neural networks. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 674–682

12. Mitchell TM, Hutchinson R, Just MA, Niculescu RS, Pereira F, Wang X (2003) Classifying instantaneous cognitive states from fMRI data. In: AMIA annual symposium proceedings, vol 2003. American Medical Informatics Association, p 465
13. Du W, Calhoun VD, Li H, Ma S, Eichele T, Kiehl KA, Adali T (2012) High classification accuracy for schizophrenia with rest and task fMRI data. Front Hum Neurosci 6:145
14. Khazaee A, Ebrahimzadeh A, Babajani-Feremi A (2015) Identifying patients with Alzheimer's disease using resting-state fMRI and graph theory. Clin Neurophysiol 126(11):2132–2141
15. Sanz-Arigita EJ, Schoonheim MM, Damoiseaux JS, Rombouts SA, Maris E, Barkhof F, Stam CJ (2010) Loss of 'small-world' networks in Alzheimer's disease: graph analysis of FMRI resting-state functional connectivity. PloS One 5(11):e13788
16. Deshpande G, Libero L, Sreenivasan KR, Deshpande H, Kana RK (2013) Identification of neural connectivity signatures of autism using machine learning. Front Hum Neurosci 7:670
17. Sidhu GS, Asgarian N, Greiner R, Brown MR (2012) Kernel principal component analysis for dimensionality reduction in fMRI-based diagnosis of ADHD. Front Syst Neurosci 6:74
18. Fekete T, Zach N, Mujica-Parodi LR, Turner MR (2013) Multiple kernel learning captures a systems-level functional connectivity biomarker signature in amyotrophic lateral sclerosis. PloS One 8(12):e85190
19. Chu C, Hsu AL, Chou KH, Bandettini P, Lin C (2012) Alzheimer's disease neuroimaging initiative. Does feature selection improve classification accuracy? Impact of sample size and feature selection on classification using anatomical magnetic resonance images. Neuroimage 60(1):59–70
20. Cabral C, Silveira M, Figueiredo P (2012) Decoding visual brain states from fMRI using an ensemble of classifiers. Pattern Recognit 45(6):2064–2074
21. Richiardi J, Eryilmaz H, Schwartz S, Vuilleumier P, Van De Ville D (2011) Decoding brain states from fMRI connectivity graphs. Neuroimage 56(2):616–626
22. Kuncheva LI, Rodríguez JJ (2010) Classifier ensembles for fMRI data analysis: an experiment. Magn Reson Imaging 28(4):583–593
23. Mortazi A, Karim R, Rhode K, Burt J, Bagci U (2017) CardiacNET: segmentation of left atrium and proximal pulmonary veins from MRI using multi-view CNN. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 377–385
24. LaLonde R, Bagci U (2018) Capsules for object segmentation. arXiv:1804.04241
25. de Brebisson A, Montana G (2015) Deep neural networks for anatomical brain segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 20–28
26. Torosdagli N, Liberton DK, Verma P, Sincan M, Lee JS, Bagci U (2018) Deep geodesic learning for segmentation and anatomical landmarking. IEEE Trans Med Imaging
27. Hussein S, Kandel P, Corral JE, Bolan CW, Wallace MB, Bagci U (2018) Deep multi-modal classification of intraductal papillary mucinous neoplasms (IPMN) with canonical correlation analysis. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). IEEE, pp 800–804
28. Khosravan N, Bagci U (2018) S4ND: single-shot single-scale lung nodule detection. arXiv:1805.02279
29. Suk HI, Shen D (2013) Deep learning-based feature representation for AD/MCI classification. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 583–590
30. Liu M, Zhang J, Adeli E, Shen D (2018) Landmark-based deep multi-instance learning for brain disease diagnosis. Med Image Anal 43:157–168
31. Khvostikov A, Aderghal K, Benois-Pineau J, Krylov A, Catheline G (2018) 3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies. arXiv:1801.05968
32. Kawahara J, Brown CJ, Miller SP, Booth BG, Chau V, Grunau RE, Hamarneh G (2017) BrainNetCNN: convolutional neural networks for brain networks; towards predicting neurodevelopment. NeuroImage 146:1038–1049

33. Wen H, Shi J, Zhang Y, Lu KH, Cao J, Liu Z (2017) Neural encoding and decoding with deep learning for dynamic natural vision. Cereb Cortex 1–25
34. Horikawa T, Kamitani Y (2017) Generic decoding of seen and imagined objects using hierarchical visual features. Nat Commun 8:15037
35. Li X, Dvornek NC, Papademetris X, Zhuang J, Staib LH, Ventola P, Duncan JS (2018) 2-channel convolutional 3D deep neural network (2CC3D) for fMRI analysis: ASD classification and feature learning. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). IEEE, pp 1252–1255
36. Li X, Dvornek NC, Zhuang J, Ventola P, Duncan JS (2018) Brain biomarker interpretation in ASD using deep learning and fMRI. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 206–214
37. Yan W, Zhang H, Sui J, Shen D (2018) Deep chronnectome learning via full bidirectional long short-term memory networks for MCI diagnosis. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 249–257
38. He T, Kong R, Holmes AJ, Sabuncu MR, Eickhoff SB, Bzdok D, Yeo BT (2018) Is deep learning better than kernel regression for functional connectivity prediction of fluid intelligence? In: 2018 international workshop on pattern recognition in neuroimaging (PRNI). IEEE, pp 1–4
39. Bellec P, Chu C, Chouinard-Decorte F, Benhajali Y, Margulies DS, Craddock RC (2017) The neuro bureau ADHD-200 preprocessed repository. Neuroimage 144:275–286
40. Cox RW (1996) AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. Comput Biomed Res 29(3):162–173
41. Jenkinson M, Beckmann CF, Behrens TE, Woolrich MW, Smith SM (2012) FSL. Neuroimage 62(2):782–790
42. Eklund A, Nichols TE, Knutsson H (2016) Cluster failure: why fMRI inferences for spatial extent have inflated false-positive rates. Proc Natl Acad Sci 201602413
43. Xia M, Wang J, He Y (2013) BrainNet viewer: a network visualization tool for human brain connectomics. PloS One 8(7):e68910
44. Holtmann M, Becker K, Kentner-Figura B, Schmidt MH (2003) Increased frequency of rolandic spikes in ADHD children. Epilepsia 44(9):1241–1244
45. Schulz KP, Fan J, Tang CY, Newcorn JH, Buchsbaum MS, Cheung AM, Halperin JM (2004) Response inhibition in adolescents diagnosed with attention deficit hyperactivity disorder during childhood: an event-related FMRI study. Am J Psychiatry 161(9):1650–1657
46. Valera EM, Faraone SV, Murray KE, Seidman LJ (2007) Meta-analysis of structural imaging findings in attention-deficit/hyperactivity disorder. Biol Psychiatry 61(12):1361–1369
47. Smith AB, Taylor E, Brammer M, Toone B, Rubia K (2006) Task-specific hypoactivation in prefrontal and temporoparietal brain regions during motor inhibition and task switching in medication-naive children and adolescents with attention deficit hyperactivity disorder. Am J Psychiatry 163(6):1044–1051
48. Norman L, Carlisi CO, Lukito S, Hart H, Mataix-Cols D, Radua J, Rubia K (2016) Comparative meta-analysis of functional and structural deficits in ADHD and OCD. JAMA Psychiatry 73:815–825
49. McCarthy H, Skokauskas N, Frodl T (2014) Identifying a consistent pattern of neural function in attention deficit hyperactivity disorder: a meta-analysis. Psychol Med 44(4):869–880
50. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems, pp 2672–2680
51. Van Essen DC, Smith SM, Barch DM, Behrens TE, Yacoub E, Ugurbil K (2013) Wu-Minn HCP consortium. The WU-Minn human connectome project: an overview. Neuroimage 80:62–79
52. Gorgolewski K, Esteban O, Schaefer G, Wandell B, Poldrack R (2017) OpenNeuro—a free online platform for sharing and analysis of neuroimaging data. In: Organization for human brain mapping. Vancouver, Canada, p 1677

Part IV
Large-Scale Data Mining and Data Synthesis

Chapter 18
ChestX-ray: Hospital-Scale Chest X-ray Database and Benchmarks on Weakly Supervised Classification and Localization of Common Thorax Diseases

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri and Ronald M. Summers

Abstract The chest X-ray is one of the most commonly accessible radiological examinations for screening and diagnosis of many lung diseases. A tremendous number of X-ray imaging studies accompanied by radiological reports are accumulated and stored in many modern hospitals' picture archiving and communication systems (PACS). On the other side, it is still an open question how this type of hospital-size knowledge database containing invaluable imaging informatics (i.e., loosely labeled)

X. Wang—This work was done during his fellowship at National Institutes of Health Clinical Center.
L. Lu—This work was done during his employment at National Institutes of Health Clinical Center.

X. Wang (✉)
Nvidia Corporation, Bethesda, MD 20814, USA
e-mail: xiaosong.wang@live.com

Y. Peng · Z. Lu
National Center for Biotechnology Information, National Library of Medicine,
National Institutes of Health, Bethesda, MD 20892, USA
e-mail: yifan.peng@nih.gov

Z. Lu
e-mail: luzh@nih.gov

L. Lu
PAII Inc., Bethesda Research Lab, 6720B Rockledge Drive, Ste 410, Bethesda,
MD 20817, USA

Johns Hopkins University, Baltimore, MD 21218, USA
e-mail: le.lu@paii-labs.com; lelu@cs.jhu.edu

M. Bagheri · R. M. Summers
Imaging Biomarkers and Computer-Aided Diagnosis Laboratory, Radiology and Imaging
Sciences Department, Clinical Center, National Institutes of Health,
Bethesda, MD 20892, USA
e-mail: mohammad.bagheri@nih.gov

R. M. Summers
e-mail: rms@nih.gov

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_18

can be used to facilitate the data-hungry deep learning paradigms in building truly large-scale high-precision computer-aided diagnosis (CAD) systems. In this chapter, we present a chest X-ray database, namely, “ChestX-ray”, which comprises 121,120 frontal-view X-ray images of 30,805 unique patients with the text-mined eight disease image labels (where each image can have multi-labels), from the associated radiological reports using natural language processing. Importantly, we demonstrate that these commonly occurring thoracic diseases can be detected and even spatially located via a unified weakly supervised multi-label image classification and disease localization framework, which is validated using our proposed dataset. Although the initial quantitative results are promising as reported, deep convolutional neural network-based “reading chest X-rays” (i.e., recognizing and locating the common disease patterns trained with only image-level labels) remains a strenuous task for fully automated high-precision CAD systems.

18.1 Introduction

Rapid and tremendous progress has occurred in a range of computer vision problems using deep learning and large-scale annotated image datasets [12, 24, 26, 37]. Drastically improved quantitative performances in object recognition, detection, and segmentation are demonstrated in comparison to previous shallow methodologies built upon handcrafted image features. Deep neural network representations further make the joint language and vision learning tasks more feasible to solve, in image captioning [21, 22, 32, 46, 47], visual question answering [1, 45, 49, 53] and knowledge-guided transfer learning [3, 33], and so on. However, the intriguing and strongly observable performance gaps of the current state-of-the-art object detection and segmentation methods, evaluated between using PASCAL VOC [12] and employing Microsoft (MS) COCO [26], demonstrate that there is still significant room for performance improvement when underlying challenges (represented by different datasets) become greater. For example, MS COCO is composed of 80 object categories from 200 k images, with 1.2 M instances (350 k are people) where every instance is segmented and many instances are small objects. Comparing to PASCAL VOC of only 20 classes and 11,530 images containing 27,450 annotated objects with bounding boxes (B-Box), the top competing object detection approaches achieve in 0.413 in MS COCO versus 0.884 in PASCAL VOC under mean average precision (mAP).

Deep learning yields similar rises in performance in the medical image analysis domain for object (often human anatomical or pathological structures in radiology imaging) detection and segmentation tasks. Recent notable work includes (but do not limit to) an overview review on the future promise of deep learning [13] and a collection of important medical applications on lymph node and interstitial lung disease detection and classification [36, 42]; cerebral microbleed detection [10]; pulmonary nodule detection in CT images [39]; automated pancreas segmentation [35]; cell image segmentation and tracking [34]; predicting spinal radiological scores [19];

and extensions of multi-modal imaging segmentation [15, 28]. The main limitation is that all proposed methods are evaluated on some small to middle-scale problems of (at most) several hundred patients. It remains unclear how well the current deep learning techniques will scale up to tens of thousands of patient studies.

In the era of deep learning in computer vision, research efforts on building various annotated image datasets [1, 12, 21, 23, 26, 32, 37, 53] with different characteristics play indispensably important roles on the better definition of the forthcoming problems, challenges, and subsequently possible technological progresses. Particularly, here we focus on the relationship and joint learning of image (chest X-rays) and text (X-ray reports). The previous representative image caption generation work [22, 47] utilize Flickr8K, Flickr30K [51], and MS COCO [26] datasets that hold 8,000, 31,000, and 123,000 images, respectively, and every image is annotated by five sentences via Amazon Mechanical Turk (AMT). The text generally describes annotator's attention of objects and activity occurring on an image in a straightforward manner. Region-level ImageNet pre-trained convolutional neural networks (CNN) based detectors are used to parse an input image and output a list of attributes or "visually grounded high-level concepts" (including objects, actions, scenes, and so on) in [22, 49]. Visual question answering (VQA) requires more detailed parsing and complex reasoning on the image contents to answer the paired natural language questions. A new dataset containing 250k natural images, 760k questions and 10M text answers [1] is provided to address this new challenge. Additionally, databases such as "Flickr30k Entities" [32], "Visual7W" [53], and "Visual Genome" [21, 23] (as detailed as 94,000 images and 4,100,000 region-grounded captions) are introduced to construct and learn the spatially dense and increasingly difficult semantic links between textual descriptions and image regions through the object-level grounding.

Though one could argue that the high-level analogy exists between image caption generation, visual question answering and imaging-based disease diagnosis [40, 41], there are three factors making truly large-scale medical image-based diagnosis (e.g., involving tens of thousands of patients) tremendously more formidable. **1**, Generic, open-ended image-level anatomy and pathology labels cannot be obtained through crowdsourcing, such as AMT, which is prohibitively implausible for non-medically trained annotators. Therefore, we exploit to mine the per-image (possibly multiple) common thoracic pathology labels from the image-attached chest X-ray radiological reports using natural language processing (NLP) techniques. Radiologists tend to write more abstract and complex logical reasoning sentences than the plain describing texts in [26, 51]. **2**, The spatial dimensions of a chest X-ray are usually 2000 × 3000 pixels. Local pathological image regions can show hugely varying sizes or extents but often very small comparing to the full image scale. Figure 18.1 shows eight illustrative examples and the actual pathological findings are often significantly smaller (thus harder to detect). Fully dense annotation of region-level bounding boxes (for grounding the pathological findings) would normally be needed in computer vision datasets [23, 32, 53] but may be completely nonviable for the time being. Consequently, we formulate and verify a weakly supervised multi-label image classification and disease localization framework to address this difficulty. **3**, So far, all image captioning and VQA techniques in computer vision strongly depend on the ImageNet

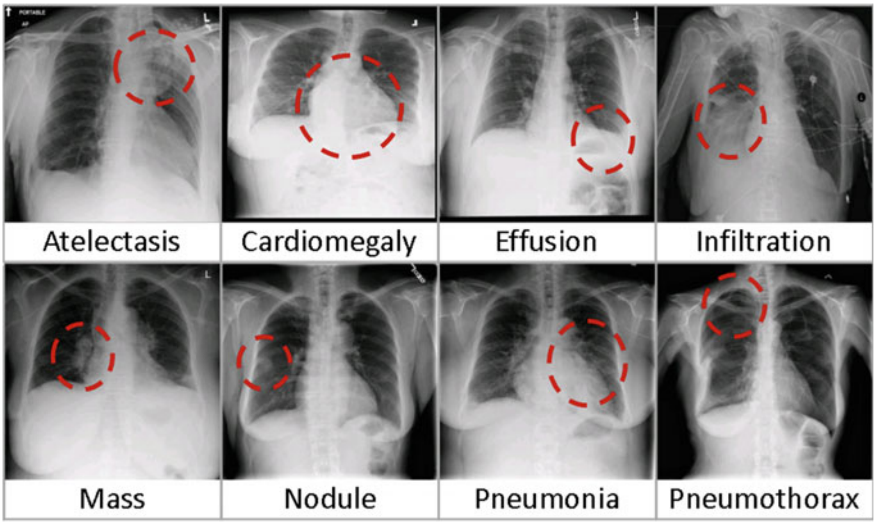


Fig. 18.1 Eight common thoracic diseases observed in chest X-rays that validate a challenging task of fully automated diagnosis

pre-trained deep CNN models which already perform very well in a large number of object classes and serves a good baseline for further model fine-tuning. However, this situation does not apply to the medical image diagnosis domain. Thus, we have to learn the deep image recognition and localization models while constructing the weakly labeled medical image database.

To tackle these issues, we propose a chest X-ray database, which comprises 112,120 frontal-view X-ray images of 30,805 (collected from the year of 1992 to 2015) unique patients with the text-mined eight common disease labels (extended to 14 disease labels later), mined from the text radiological reports via NLP techniques. In particular, we demonstrate that these commonly occurred thoracic diseases can be detected and even spatially located via a unified weakly supervised multi-label image classification and disease localization formulation. Our initial quantitative results are promising. However, developing fully automated deep learning-based "reading chest X-rays" systems is still an arduous journey to be exploited. Details of accessing the ChestX-ray dataset can be found via the website.[1]

18.1.1 Recent Advances

There have been recent efforts on creating openly available annotated medical image databases [35, 36, 48, 50] with the studied patient numbers ranging from a few

[1] https://nihcc.app.box.com/v/ChestXray-NIHCC.

hundreds to two thousands. Particularly for chest X-rays, the largest public dataset is OpenI [29] that contains 3,955 radiology reports from the Indiana Network for Patient Care and 7,470 associated chest x-rays from the hospitals' picture archiving and communication system (PACS). This database is utilized in [41] as a problem of caption generation but no quantitative disease detection results are reported. Our newly proposed chest X-ray database is at least one order of magnitude larger than OpenI [29] (Refer to Table 18.1). To achieve the better clinical relevance, we focus to exploit the quantitative performance on weakly supervised multi-label image classification and disease localization of common thoracic diseases, in analogy to the intermediate step of "detecting attributes" in [49] or "visual grounding" for [21, 32, 53].

18.2 Database Construction

First, we discuss the approach for building a hospital-scale chest X-ray image database, namely, "ChestX-ray8", mined from our institute's PACS system. First, we short-list eight common thoracic pathology keywords that are frequently observed and diagnosed, i.e., atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, and pneumothorax (Fig. 18.1), based on radiologists' feedback. Given those eight text keywords, we search the PACS system to pull out all the related radiological reports (together with images) as our target corpus. A variety of natural language processing (NLP) techniques are adopted for detecting the pathology keywords and removal of negation and uncertainty. Each radiological report will be either linked with one or more keywords or marked with "Normal" as the background category. As a result, the ChestX-ray8 database is composed of 112,120 frontal-view X-ray images (from 30,805 patients) and each image is labeled with one or multiple pathology keywords or "Normal" otherwise. Figure 18.2 illustrates the correlation of the resulted keywords. It reveals some connections between different pathologies, which agree with radiologists' domain knowledge, e.g., infiltration is often associated with atelectasis and effusion. To some extent, this is similar with understanding the interactions and relationships among objects or concepts in natural images [23].

18.2.1 Disease Label Mining

Overall, our approach produces labels using the reports in two passes. In the first iteration, we detected all the disease concepts in the corpus. The main body of each chest X-ray report is generally structured as "Comparison", "Indication", "Findings", and "Impression" sections. Here, we focus on detecting disease concepts in the Findings and Impression sections. If a report contains neither of these two sections, the

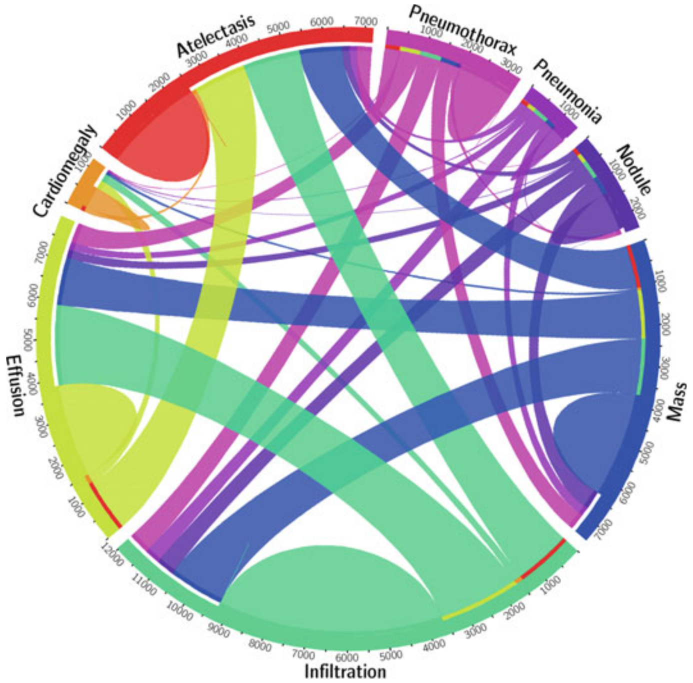


Fig. 18.2 The circular diagram shows the proportions of images with multi-labels in each of eight pathology classes and the labels' co-occurrence statistics

full-length report will then be considered. In the second pass, we code the reports as "Normal" if they do not contain any diseases (not limited to 8 predefined pathologies).

Stage 1: Pathology Entity Extraction

We mine the radiology reports for disease concepts using two tools, DNorm [25] and MetaMap [2]. DNorm is a machine learning method for disease recognition and normalization. It maps every mention of keywords in a report to a unique concept ID in the Systematized Nomenclature of Medicine–Clinical Terms (or SNOMED-CT), which is a standardized vocabulary of clinical terminology for the electronic exchange of clinical health information.

MetaMap is another prominent tool to detect bio-concepts from the biomedical text corpus. Different from DNorm, it is an ontology-based approach for the detection of Unified Medical Language System® (UMLS®) Metathesaurus. In this work, we only consider the semantic types of Diseases or Syndromes and Findings (namely, "dsyn" and "fndg", respectively). To maximize the recall of our automatic disease detection, we merge the results of DNorm and MetaMap. Table 18.1 (in the

Table 18.1 Total number (#) and # of overlap (Ov.) of the corpus in both OpenI and ChestX-ray8 datasets

Item #	OpenI	Ov.	ChestX-ray8	Ov.
Report	2,435	–	108,948	–
Annotations	2,435	–	–	–
Atelectasis	315	122	5,789	3,286
Cardiomegaly	345	100	1,010	475
Effusion	153	94	6,331	4,017
Infiltration	60	45	10,317	4,698
Mass	15	4	6,046	3,432
Nodule	106	18	1,971	1,041
Pneumonia	40	15	1,062	703
Pneumothorax	22	11	2,793	1,403
Normal	1,379	0	84,312	0

supplementary material) shows the corresponding SNOMED-CT concepts that are relevant to the eight target diseases (these mappings are developed by searching the disease names in the UMLS®terminology service,[2] and verified by a board-certified radiologist.

Stage 2: Negation and Uncertainty Detection

The disease detection algorithm locates every keyword mentioned in the radiology report no matter if it is truly present or negated. To eliminate the noisy labeling, we need to rule out those negated pathological statements and, more importantly, uncertain mentions of findings and diseases, e.g., "suggesting obstructive lung disease".

Although many text processing systems (such as [5]) can handle the negation/uncertainty detection problem, most of them exploit regular expressions on the text directly. One of the disadvantages to use regular expressions for negation/uncertainty detection is that they cannot capture various syntactic constructions for multiple subjects. For example, in the phrase of "clear of A and B", the regular expression can capture "A" as a negation but not "B", particularly when both "A" and "B" are long and complex noun phrases ("clear of focal airspace disease, pneumothorax, or pleural effusion" in Fig. 18.3).

To overcome this complication, we handcraft a number of novel rules of negation/uncertainty defined on the syntactic level in this work. More specifically, we utilize the syntactic dependency information because it is close to the semantic relationship between words and thus has become prevalent in biomedical text processing. We defined our rules on the dependency graph, by utilizing the dependency label and direction information between words.

[2] https://uts.nlm.nih.gov/metathesaurus.html.

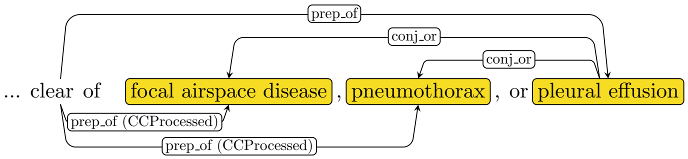


Fig. 18.3 The dependency graph of text: "clear of focal airspace disease, pneumothorax, or pleural effusion"

As the first step of preprocessing, we split and tokenize the reports into sentences using NLTK [4]. Next we parse each sentence by the Bllip parser [6] using David McClosky's biomedical model [27]. The syntactic dependencies are then obtained from "CCProcessed" dependencies output by applying Stanford dependencies converter [7] on the parse tree. The "CCProcessed" representation propagates conjunct dependencies thus simplifies coordination. As a result, we can use fewer rules to match more complex constructions. For an example as shown in Fig. 18.3, we could use "clear → prep_of → DISEASE" to detect three negations from the text ⟨neg, focal airspace disease⟩, ⟨neg, pneumothorax⟩, and ⟨neg, pleural effusion⟩.

Furthermore, we label a radiology report as "normal" if it meets one of the following criteria:

- If there is no disease detected in the report. Note that here we not only consider eight diseases of interest in this paper, but all diseases detected in the reports.
- If the report contains text-mined concepts of "normal" or "normal size" (CUIs C0205307 and C0332506 in the SNOMED-CT concepts, respectively).

18.2.2 Evaluation on Mined Disease Labels

To validate our method, we perform the following experiments. Given the fact that no gold-standard labels exist for our dataset, we resort to some existing annotated corpora as an alternative. Using the OpenI API [29], we retrieve a total of 3,851 unique radiology reports where each OpenI report is assigned with its key findings/disease names by human annotators [8]. Given our focus on the eight diseases, a subset of OpenI reports and their human annotations are used as the gold standard for evaluating our method. Table 18.1 summarizes the statistics of the subset of OpenI [18, 29] reports. Table 18.2 shows the results of our method using OpenI, measured in precision (P), recall (R), and F1-score. Higher precision of 0.90, higher recall of 0.91, and higher F1-score of 0.90 are achieved compared to the existing MetaMap approach (with NegEx enabled). For all diseases, our method obtains higher precisions, particularly in "pneumothorax" (0.90 vs. 0.32) and "infiltration" (0.74 vs. 0.25). This indicates that the usage of negation and uncertainty detection on syntactic level successfully removes false positive cases. More importantly, the higher precisions meet

Table 18.2 Evaluation of image labeling results on OpenI dataset. Performance is reported using P, R, F1-score

Disease	MetaMap			Our method		
	P	R	F	P	R	F
Atelectasis	0.95	0.95	0.95	0.99	0.85	0.91
Cardiomegaly	0.99	0.83	0.90	1.00	0.79	0.88
Effusion	0.74	0.90	0.81	0.93	0.82	0.87
Infiltration	0.25	0.98	0.39	0.74	0.87	0.80
Mass	0.59	0.67	0.62	0.75	0.40	0.52
Nodule	0.95	0.65	0.77	0.96	0.62	0.75
Normal	0.93	0.90	0.91	0.87	0.99	0.93
Pneumonia	0.58	0.93	0.71	0.66	0.93	0.77
Pneumothorax	0.32	0.82	0.46	0.90	0.82	0.86
Total	0.84	0.88	0.86	0.90	0.91	0.90

our expectation to generate a Chest X-ray corpus with accurate semantic labels, to lay a solid foundation for the later processes.

18.2.3 Chest X-ray Image Processing and Hand-Labeled Ground Truth

Comparing to the popular ImageNet classification problem, significantly smaller spatial extents of many diseases inside the typical X-ray image dimensions of 3000 × 2000 pixels impose challenges in both the capacity of computing hardware and the design of deep learning paradigm. In ChestX-ray8, X-rays images are directly extracted from the DICOM file and resized as 1024 × 1024 bitmap images without significantly losing the detail contents, compared with image sizes of 512 × 512 in OpenI dataset. Their intensity ranges are rescaled using the default window settings stored in the DICOM header files.

As part of the ChestX-ray8 database, a small number of images with pathology are provided with hand-labeled bounding boxes (B-Boxes), which can be used as the ground truth to evaluate the disease localization performance. Furthermore, it could also be adopted for one/low-shot learning setup [14], in which only one or several samples are needed to initialize the learning and the system will then evolve by itself with more unlabeled data. We leave this as future work.

In our labeling process, we first select 200 instances for each pathology (1,600 instances total), consisting of 983 images. Given an image and a disease keyword, a board-certified radiologist identified only the corresponding disease instance in the image and labeled it with a B-Box. The B-Box is then outputted as an XML file. If one image contains multiple disease instances, each disease instance is labeled separately and stored into individual XML files. As an application of the proposed ChestX-ray8 database and benchmarking, we will demonstrate the detection and localization of thoracic diseases in the following.

18.3 Applications on Constructed Database

Reading and diagnosing Chest X-ray images may be an entry-level task for radiologists but, in fact, it is a complex reasoning problem which often requires careful observation and good knowledge of anatomical principles, physiology, and pathology. Such factors increase the difficulty of developing a consistent and automated technique for reading chest X-ray images while simultaneously considering all common thoracic diseases.

As the main application of ChestX-ray8 dataset, we present a unified weakly supervised multi-label image classification and pathology localization framework, which can detect the presence of multiple pathologies and subsequently generate bounding boxes around the corresponding pathologies. In detail, we tailor deep convolutional neural network (DCNN) architectures for weakly supervised object localization, by considering large image capacity, various multi-label CNN losses, and different pooling strategies.

18.3.1 Classification and Localization Framework

Our goal is to first detect if one or multiple pathologies are presented in each X-ray image and later we can locate them using the activation and weights extracted from the network. We tackle this problem by training a multi-label DCNN classification model. Figure 18.4 illustrates the DCNN architecture we adapted, with similarity to several previous weakly supervised object localization methods [11, 17, 30, 52]. As shown in Fig. 18.4, we perform the network surgery on the pre-trained models (using ImageNet [9, 38]), e.g., AlexNet [24], GoogLeNet [44], VGGNet-16 [43], and ResNet-50 [16], by leaving out the fully connected layers and the final classification layers. Instead, we insert a transition layer, a global pooling layer, a prediction layer, and a loss layer in the end (after the last convolutional layer). In a similar fashion as described in [52], a combination of deep activations from transition layer (a set of spatial image features) and the weights of prediction inner-product layer (trained feature weighting) can enable us to find the plausible spatial locations of diseases.

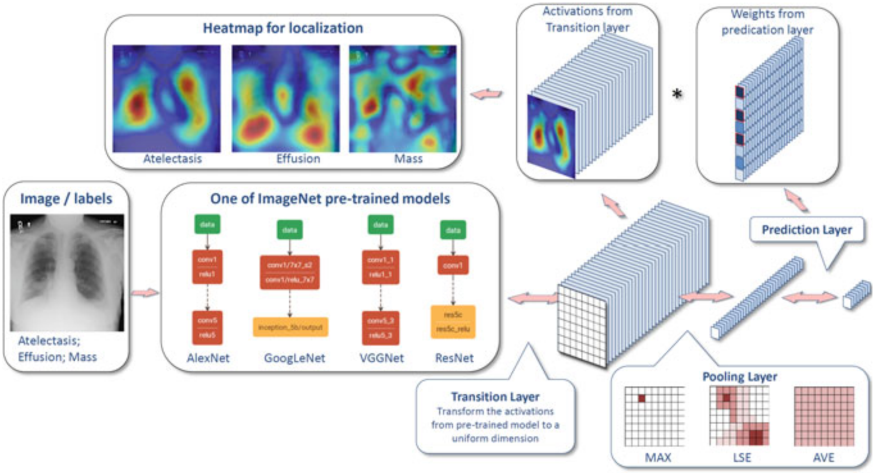


Fig. 18.4 The overall flowchart of our unified DCNN framework and disease localization process

Multi-label Classification

There are several options of image-label representation and the choices of multi-label classification loss functions. Here, we define an eight-dimensional label vector $\mathbf{y} = [y_1, \ldots, y_c, \ldots, y_C]$, $y_c \in \{0, 1\}$, $C = 8$ for each image. y_c indicates the presence with respect to according pathology in the image while an all-zero vector $[0, 0, 0, 0, 0, 0, 0, 0]$ represents the status of "Normal" (no pathology is found in the scope of any of 8 disease categories as listed). This definition transits the multi-label classification problem into a regression-like loss setting.

Transition Layer: Due to the large variety of pre-trained DCNN architectures we adopt, a transition layer is usually required to transform the activations from previous layers into a uniform dimension of output, $S \times S \times D$, $S \in \{8, 16, 32\}$. D represents the dimension of features at spatial location (i, j), $i, j \in \{1, \ldots, S\}$, which can be varied in different model settings, e.g., $D = 1024$ for GoogLeNet and $D = 2048$ for ResNet. The transition layer helps pass down the weights from pre-trained DCNN models in a standard form, which is critical for using this layer's activations to further generate the heatmap in pathology localization step.

Multi-label Loss Layer: We first experiment three standard loss functions for the regression task instead of using the softmax loss for traditional multi-class classification model, i.e., hinge loss (HL), Euclidean loss (EL), and cross-entropy loss (CEL). However, we find that the model has difficulty learning positive instances (images with pathologies) and the image labels are rather sparse, meaning there are extensively more "0's" than "1's". This is due to our one-hot-like image labeling strategy and the unbalanced numbers of pathology and "Normal" classes. Therefore, we introduce the positive/negative balancing factor β_P, β_N to enforce the learning of positive examples. For example, the weighted CEL (W-CEL) is defined as follows,

$$\begin{aligned} &L_{W\text{-}CEL}(f(\mathbf{x}), \mathbf{y}) = \\ &\beta_P \sum_{y_c=1} -\ln(f(x_c)) + \beta_N \sum_{y_c=0} -\ln(1 - f(x_c)), \end{aligned} \tag{18.1}$$

where β_P is set to $\frac{|P|+|N|}{|P|}$ while β_N is set to $\frac{|P|+|N|}{|N|}$. $|P|$ and $|N|$ are the total number of "1's" and "0's" in a batch of image labels.

Disease Localization via Heatmap

In our multi-label image classification network, the global pooling and the predication layer are designed not only to be part of the DCNN for classification but also to generate the likelihood map of pathologies, namely, a heatmap. The location with a peak in the heatmap generally corresponds to the presence of disease pattern with a high probability. The upper part of Fig. 18.4 demonstrates the process of producing this heatmap. By performing a global pooling after the transition layer, the weights learned in the prediction layer can function as the weights of spatial maps from the transition layer. Therefore, we can produce weighted spatial activation maps for each disease class (with a size of $S \times S \times C$) by multiplying the activation from transition layer (with a size of $S \times S \times D$) and the weights of prediction layer (with a size of $D \times C$).

The pooling layer plays an important role that chooses what information to be passed down. Besides the conventional max pooling and average pooling, we also utilize the log-sum-exp (LSE) pooling proposed in [31]. The LSE pooled value x_p is defined as

$$x_p = \frac{1}{r} \cdot \log \left[\frac{1}{S} \cdot \sum_{(i,j)\in\mathbf{S}} exp(r \cdot x_{ij}) \right], \tag{18.2}$$

where x_{ij} is the activation value at (i, j), (i, j) is one location in the pooling region $\mathbf{S}$, and $S = s \times s$ is the total number of locations in $\mathbf{S}$. By controlling the hyperparameter r, the pooled value ranges from the maximum in $\mathbf{S}$ (when $r \to \infty$) to average ($r \to 0$). It serves as an adjustable option between max pooling and average pooling. Since the LSE function suffers from overflow/underflow problems, the following equivalent is used while implementing the LSE pooling layer in our own DCNN architecture,

$$x_p = x^* + \frac{1}{r} \cdot \log \left[\frac{1}{S} \cdot \sum_{(i,j)\in\mathbf{S}} exp(r \cdot (x_{ij} - x^*) \right], \tag{18.3}$$

where $x^* = max\{|x_{ij}|, (i, j) \in \mathbf{S}\}$.

Bounding Box Generation: The heatmap produced from our multi-label classification framework indicates the approximate spatial location of one particular thoracic disease class each time. Due to the simplicity of intensity distributions in these resulting heatmaps, applying an ad hoc thresholding-based B-Box generation method for this task is found to be sufficient. The intensities in heatmaps are first normalized to [0, 255] and then thresholded by {60, 180} individually. Finally, B-Boxes are generated to cover the isolated regions in the resulting binary maps.

18.4 Evaluations

Data: We evaluate and validate the unified disease classification and localization framework using the proposed ChestX-ray database. For the pathology classification and localization task, we randomly shuffled the entire dataset into three subgroups for CNN fine-tuning via stochastic gradient descent (SGD): i.e., training (70%), validation (10%) and testing (20%). We only report the 8 thoracic disease recognition performance on the testing set in our experiments. Furthermore, for the 983 images with 1,600 annotated B-Boxes of pathologies, these boxes are only used as the ground truth to evaluate the disease localization accuracy in testing (not for training purpose).

CNN Setting: Our multi-label CNN architecture is implemented using Caffe framework [20]. The ImageNet pre-trained models, i.e., AlexNet [24], GoogLeNet [44], VGGNet-16 [43], and ResNet-50 [16], are obtained from the Caffe model zoo. Our unified DCNN takes the weights from those models and only the transition layers and prediction layers are trained from scratch.

Due to the large image size and the limit of GPU memory, it is necessary to reduce the image *batch_size* to load the entire model and keep activations in GPU while we increase the *iter_size* to accumulate the gradients for more iterations. The combination of both may vary in different CNN models but we set *batch_size* × *iter_size* = 80 as a constant. Furthermore, the total training iterations are customized for different CNN models to prevent overfitting. More complex models like ResNet-50 actually take less iterations (e.g., 10000 iterations) to reach the convergence. The DCNN models are trained using a Dev-Box Linux server with 4 Titan X GPUs.

Multi-label Disease Classification: Fig. 18.5 demonstrates the multi-label classification ROC curves on eight pathology classes by initializing the DCNN framework with four different pre-trained models of AlexNet, GoogLeNet, VGG, and ResNet-50. The corresponding area under curve (AUC) values are given in Table 18.3. The quantitative performance varies greatly, in which the model based on ResNet-50 achieves the best results. The "Cardiomegaly" (AUC = 0.8141) and "Pneumothorax" (AUC = 0.7891) classes are consistently well recognized compared to other groups while the detection ratios can be relatively lower for pathologies which contain small objects, e.g., "Mass" (AUC = 0.5609) and "Nodule" classes. Mass is difficult to detect due to its huge within-class appearance variation. The lower performance on "Pneumonia" (AUC = 0.6333) is probably because of lack of total instances in

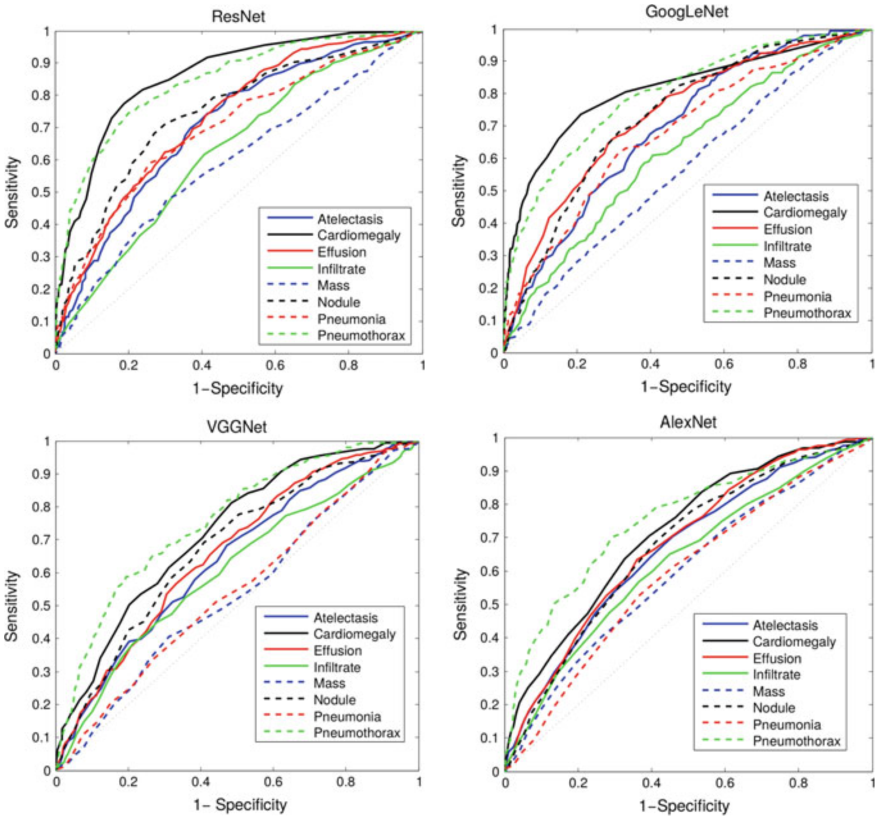


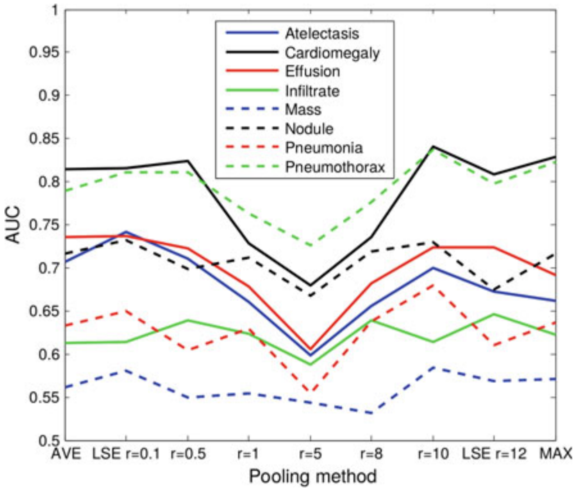
Fig. 18.5 A comparison of multi-label classification performance with different model initializations

our patient population (less than 1% X-rays labeled as Pneumonia). This finding is consistent with the comparison on object detection performance, degrading from PASCAL VOC [12] to MS COCO [26] where many small annotated objects appear. Next, we examine the influence of different pooling strategies when using ResNet-50 to initialize the DCNN framework. As discussed above, three types of pooling schemes are experimented: average looping, LSE pooling, and max pooling. The hyperparameter r in LSE pooling varies in $\{0.1, 0.5, 1, 5, 8, 10, 12\}$. As illustrated in Fig. 18.6, average pooling and max pooling achieve approximately equivalent performance in this classification task. The performance of LSE pooling starts to decline first when r starts to increase and reaches the bottom when $r = 5$. Then it reaches the overall best performance around $r = 10$. LSE pooling behaves like a weighed pooling method or a transition scheme between average and max pooling under different r values. Overall, LSE pooling ($r = 10$) reports the best performance (consistently higher than mean and max pooling).

Table 18.3 AUCs of ROC curves for multi-label classification in different DCNN model setting (**At**: Atelectasis; **Ca**: Cardiomegaly; **Ef**: Effusion; **In**: Infiltration; **Ma**: Mass; **No**: Nodule; **Pa**: Pneumonia; **Px**: Pneumothorax)

Setting	At	Ca	Ef	In	Ma	No	Pa	Px
Initialization with different pre-trained models								
AlexNet	0.6458	0.6925	0.6642	0.6041	**0.5644**	0.6487	0.5493	0.7425
GoogLeNet	0.6307	0.7056	0.6876	0.6088	0.5363	0.5579	0.5990	0.7824
VGGNet-16	0.6281	0.7084	0.6502	0.5896	0.5103	0.6556	0.5100	0.7516
ResNet-50	**0.7069**	**0.8141**	**0.7362**	**0.6128**	0.5609	**0.7164**	**0.6333**	**0.7891**
Different multi-label loss functions								
CEL	0.7064	0.7262	0.7351	0.6084	0.5530	0.6545	0.5164	0.7665
W-CEL	0.7069	0.8141	0.7362	0.6128	0.5609	0.7164	0.6333	0.7891

Fig. 18.6 A comparison of multi-label classification performance with different pooling strategies



Last, we demonstrate the performance improvement by using the positive/negative instances balanced loss functions (Eq. 18.1). As shown in Table 18.3, the weighted loss (W-CEL) provides better overall performance than CEL, especially for those classes with relative fewer positive instances, e.g., AUC for “Cardiomegaly” is increased from 0.7262 to 0.8141 and from 0.5164 to 0.6333 for “Pneumonia”.

Disease Localization: Leveraging the fine-tuned DCNN models for multi-label disease classification, we can calculate the disease heatmaps using the activations of the transition layer and the weights from the prediction layer, and even generate the B-Boxes for each pathology candidate. The computed bounding boxes are evaluated against the hand-annotated ground truth (GT) boxes (included in ChestX-ray8). Although the total number of B-Box annotations (1,600 instances) is relatively small compared to the entire dataset, it may be still sufficient to get a reasonable estimate on how the proposed framework performs on the weakly supervised disease localization

Table 18.4 Pathology localization accuracy and average false positive number for eight disease classes (**At**: Atelectasis; **Ca**: Cardiomegaly; **Ef**: Effusion; **In**: Infiltration; **Ma**: Mass; **No**: Nodule; **Pa**: Pneumonia; **Px**: Pneumothorax)

T(IoBB)	At	Ca	Ef	In	Ma	No	Pa	Px
T(IoBB) = 0.1								
Acc.	0.7277	0.9931	0.7124	0.7886	0.4352	0.1645	0.7500	0.4591
AFP	0.8323	0.3506	0.7998	0.5589	0.6423	0.6047	0.9055	0.4776
T(IoBB) = 0.25 (Two times larger on both x- and y-axes than ground truth B-Boxes)								
Acc.	0.5500	0.9794	0.5424	0.5772	0.2823	0.0506	0.5583	0.3469
AFP	0.9167	0.4553	0.8598	0.6077	0.6707	0.6158	0.9614	0.5000
T(IoBB) = 0.5								
Acc.	0.2833	0.8767	0.3333	0.4227	0.1411	0.0126	0.3833	0.1836
AFP	1.0203	0.5630	0.9268	0.6585	0.6941	0.6189	1.0132	0.5285
T(IoBB) = 0.75								
Acc.	0.1666	0.7260	0.2418	0.3252	0.1176	0.0126	0.2583	0.1020
AFP	1.0619	0.6616	0.9603	0.6921	0.7043	0.6199	1.0569	0.5396
T(IoBB) = 0.9								
Acc.	0.1333	0.6849	0.2091	0.2520	0.0588	0.0126	0.2416	0.0816
AFP	1.0752	0.7226	0.9797	0.7124	0.7144	0.6199	1.0732	0.5437

task. To examine the accuracy of computerized B-Boxes versus the GT B-Boxes, two types of measurement are used, i.e., the standard intersection over union ratio (IoU) or the Intersection over the detected B-Box area ratio (IoBB) (similar to area of precision or purity). Due to the relatively low spatial resolution of heatmaps (32 × 32) in contrast to the original image dimensions (1024 × 1024), the computed B-Boxes are often larger than the according GT B-Boxes. Therefore, we define a correct localization by requiring either $IoU > T(IoU)$ or $IoBB > T(IoBB)$. Refer to the supplementary material for localization performance under varying $T(IoU)$. Table 18.4 illustrates the localization accuracy (Acc.) and average false positive (AFP) number for each disease type, with $T(IoBB) \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$. Please refer to the supplementary material for qualitative exemplary disease localization results for each of 8 pathology classes.

18.5 Extension to 14 Common Thorax Disease Labels

To further complete the list of common thorax diseases in Chest X-ray, we expand the disease categories to include six more common diseases (i.e., consolidation, edema, emphysema, fibrosis, pleural thickening, and hernia) and update the NLP mined labels. The statistics of ChestX-ray14 dataset are illustrated in Table 18.5 and Fig. 18.7. The bounding boxes for Pathologies are unchanged in this extension.

Table 18.5 Total number (#) and # of overlap of the corpus in ChestX-ray8 and ChestX-ray14 datasets. PT: Pleural Thickening

Item #	ChestX-ray8	Overlap	ChestX-ray14	Overlap
Report	108,948	–	112,120	–
Atelectasis	5,789	3,286	11,535	7,323
Cardiomegaly	1,010	475	2,772	1,678
Effusion	6,331	4,017	13,307	9,348
Infiltration	10,317	4,698	19,871	10,319
Mass	6,046	3,432	5,746	2,138
Nodule	1,971	1,041	6,323	3,617
Pneumonia	1,062	703	1,353	1,046
Pneumothorax	2,793	1,403	5,298	3,099
Consolidation	–	–	4,667	3,353
Edema	–	–	2,303	1,669
Emphysema	–	–	2,516	1,621
Fibrosis	–	–	1,686	959
PT	–	–	3,385	2,258
Hernia	–	–	227	117
No findings	84,312	0	60,412	0

18.5.1 Evaluation of NLP Mined Labels

To validate our method, we perform the following experiments. First, we resort to some existing annotated corpora as an alternative, i.e., OpenI dataset. Furthermore, we annotated clinical reports suitable for evaluating finding recognition systems. We randomly selected 900 reports and asked two annotators to mark the above 14 types of findings. Each report was annotated by two annotators independently and then agreements are reached for conflicts.

Table 18.6 shows the results of our method using OpenI and our proposed dataset, measured in precision (P), recall (R), and F1-score. Much higher precision, recall and F1-scores are achieved compared to the existing MetaMap approach (with NegEx enabled). This indicates that the usage of negation and uncertainty detection on syntactic level successfully removes false positive cases.

18.5.2 Benchmark Results

In a similar fashion to the experiment on ChestX-ray8, we evaluate and validate the unified disease classification and localization framework on ChestX-ray14 database. In total, 112,120 frontal-view X-ray images are used, of which 51,708 images con-

Table 18.6 Evaluation of image labeling results on OpenI and ChestX-ray14 dataset. Performance is reported using P, R, F1-score. PT: Pleural Thickening

Disease	MetaMap			Our method		
	Precision	Recall	F1-score	Precision	Recall	F1-score
OpenI						
Atelectasis	87.3	96.5	91.7	88.7	96.5	92.4
Cardiomegaly	100.0	85.5	92.2	100.0	85.5	92.2
Effusion	90.3	87.5	88.9	96.6	87.5	91.8
Infiltration	68.0	100.0	81.0	81.0	100.0	89.5
Mass	100.0	66.7	80.0	100.0	66.7	80.0
Nodule	86.7	65.0	74.3	82.4	70.0	75.7
Pneumonia	40.0	80.0	53.3	44.4	80.0	57.1
Pneumothorax	80.0	57.1	66.7	80.0	57.1	66.7
Consolidation	16.3	87.5	27.5	77.8	87.5	82.4
Edema	66.7	90.9	76.9	76.9	90.9	83.3
Emphysema	94.1	64.0	76.2	94.1	64.0	76.2
Fibrosis	100.0	100.0	100.0	100.0	100.0	100.0
PT	100.0	75.0	85.7	100.0	75.0	85.7
Hernia	100.0	100.0	100.0	100.0	100.0	100.0
Total	77.2	84.6	80.7	89.8	85.0	87.3
ChestX-ray14						
Atelectasis	88.6	98.1	93.1	96.6	97.3	96.9
Cardiomegaly	94.1	95.7	94.9	96.7	95.7	96.2
Effusion	87.7	99.6	93.3	94.8	99.2	97.0
Infiltration	69.7	90.0	78.6	95.9	85.6	90.4
Mass	85.1	92.5	88.7	92.5	92.5	92.5
Nodule	78.4	92.3	84.8	84.5	92.3	88.2
Pneumonia	73.8	87.3	80.0	88.9	87.3	88.1
Pneumothorax	87.4	100.0	93.3	94.3	98.8	96.5
Consolidation	72.8	98.3	83.7	95.2	98.3	96.7
Edema	72.1	93.9	81.6	96.9	93.9	95.43
Emphysema	97.6	93.2	95.3	100.0	90.9	95.2
Fibrosis	84.6	100.0	91.7	91.7	100.0	95.7
PT	85.1	97.6	90.9	97.6	97.6	97.6
Hernia	66.7	100.0	80.0	100.0	100.0	100.0
Total	82.8	95.5	88.7	94.4	94.4	94.4

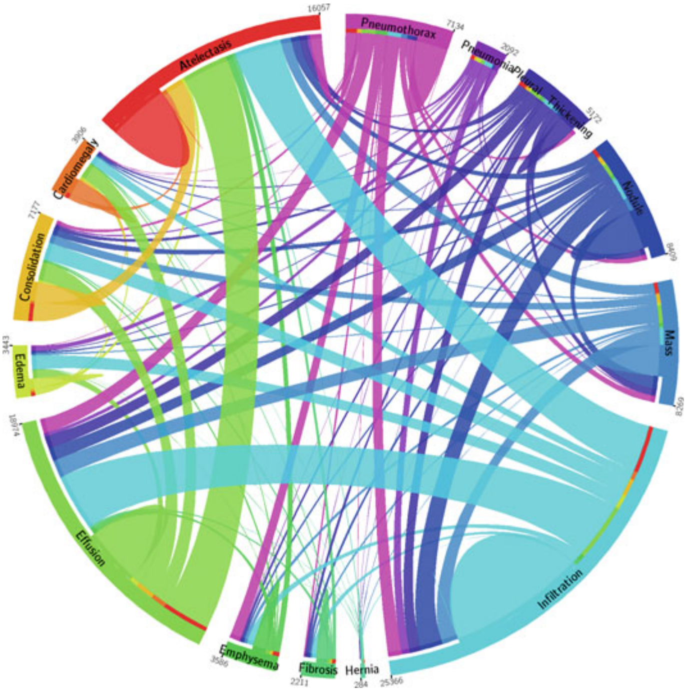


Fig. 18.7 The circular diagram shows the proportions of images with multi-labels in each of 14 pathology classes and the labels' co-occurrence statistics

Table 18.7 AUCs of ROC curves for multi-label classification for ChestX-ray14 using published data split. PT: Pleural Thickening

ResNet-50	ChestX-ray8	ChestX-ray14
Atelectasis	0.7069	0.7003
Cardiomegaly	0.8141	0.8100
Effusion	0.7362	0.7585
Infiltration	0.6128	0.6614
Mass	0.5609	0.6933
Nodule	0.7164	0.6687
Pneumonia	0.6333	0.6580
Pneumothorax	0.7891	0.7993
Consolidation	–	0.7032
Edema	–	0.8052
Emphysema	–	0.8330
Fibrosis	–	0.7859
PT	–	0.6835
Hernia	–	0.8717

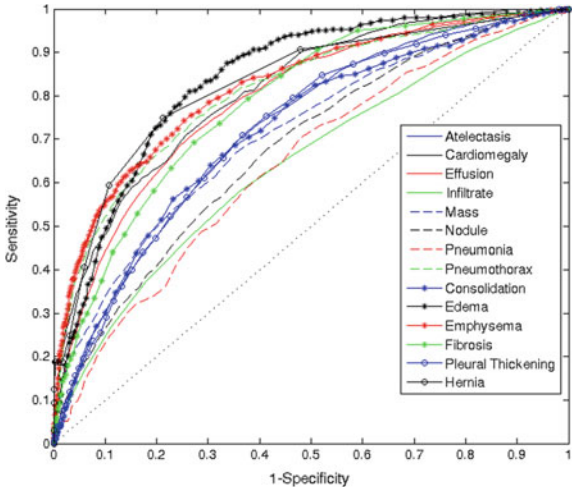


Fig. 18.8 Multi-label classification performance on ChestX-ray14 with ImageNet pre-trained ResNet

tain one or more pathologies. The remaining 60,412 images do not contain the listed 14 disease findings. For the pathology classification and localization task, we randomly shuffled the entire dataset into three subgroups on the patient level for CNN fine-tuning via stochastic gradient descent (SGD): i.e., training ($\sim$70%), validation ($\sim$10%), and testing ($\sim$20%). All images from the same patient will only appear in one of the three sets.[3] We report the 14 thoracic disease recognition performance on the published testing set in comparison with the counterpart based on ChestX-ray8, shown in Table 18.7 and Fig. 18.8.

Since the annotated B-Boxes of pathologies are unchanged, we only test the localization performance on the original eight categories. Results measured by the Intersection over the detected B-Box area ratio (IoBB) (similar to area of precision or purity) are demonstrated in Table 18.8.

18.6 Summary

Constructing hospital-scale radiology image databases with computerized diagnostic performance benchmarks has not been addressed until this work. We attempt to build a "machine–human annotated" comprehensive chest X-ray database that presents the realistic clinical and methodological challenges of handling at least tens of thousands of patients (somewhat similar to "ImageNet" in natural images). We also conduct extensive quantitative performance benchmarking on eight common thoracic pathology classification and weakly supervised localization using ChestX-ray database. The main goal is to initiate future efforts by promoting public datasets

[3] Data split files could be downloaded via https://nihcc.app.box.com/v/ChestXray-NIHCC.

Table 18.8 Pathology localization accuracy and average false positive number for ChestX-ray14 (**At**: Atelectasis; **Ca**: Cardiomegaly; **Ef**: Effusion; **In**: Infiltration; **Ma**: Mass; **No**: Nodule; **Pa**: Pneumonia; **Px**: Pneumothorax)

T(IoBB)	At	Ca	Ef	In	Ma	No	Pa	Px
T(IoBB) = 0.1								
Acc.	0.6222	1	0.7974	0.9106	0.5882	0.1519	0.8583	0.5204
AFP	0.8293	0.1768	0.6148	0.4919	0.3933	0.4685	0.4360	0.4543
T(IoBB) = 0.25 (Two times larger on both x- and y-axes than ground truth B-Boxes)								
Acc.	0.3944	0.9863	0.6339	0.7967	0.4588	0.0506	0.7083	0.3367
AFP	0.9319	0.2042	0.6880	0.5447	0.4288	0.4786	0.4959	0.4857
T(IoBB) = 0.5								
Acc.	0.1944	0.9452	0.4183	0.6504	0.3058	0	0.4833	0.2653
AFP	0.9979	0.2785	0.7652	0.6006	0.4604	0.4827	0.5630	0.5030
T(IoBB) = 0.75								
Acc.	0.0889	0.8151	0.2287	0.4390	0.1647	0	0.2917	0.1735
AFP	1.0285	0.4045	0.8222	0.6697	0.4827	0.4827	0.6169	0.5243
T(IoBB) = 0.9								
Acc.	0.0722	0.6507	0.1373	0.3577	0.0941	0	0.2333	0.1224
AFP	1.0356	0.4837	0.8445	0.7043	0.4939	0.4827	0.6331	0.5346

in this important domain. Building truly large-scale, fully automated high-precision medical diagnosis systems remains a strenuous task. ChestX-ray can enable the data-hungry deep neural network paradigms to create clinically meaningful applications, including common disease pattern mining, disease correlation analysis, automated radiological report generation, etc. For future work, ChestX-ray will be extended to cover more disease classes and integrated with other clinical information, e.g., follow-up studies across time and patient history.

Acknowledgements This work was supported by the Intramural Research Programs of the NIH Clinical Center and National Library of Medicine. We thank NVIDIA Corporation for the GPU donation.

References

1. Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Zitnick L (2015) Vqa: visual question answering. In: ICCV
2. Aronson AR, Lang FM (2010) An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc 17(3):229–236. https://doi.org/10.1136/jamia.2009.002733
3. Ba J, Swersky K, Fidler S, Salakhutdinov R (2015) Predicting deep zero-shot convolutional neural networks using textual descriptions. In: ICCV
4. Bird S, Klein E, Loper E (2009) Natural language processing with Python. O'Reilly Media, Inc

5. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG (2001) A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 34(5):301–310. https://doi.org/10.1006/jbin.2001.1029, http://www.sciencedirect.com/science/article/pii/S1532046401910299
6. Charniak E, Johnson M (2005) Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In: Proceedings of the 43rd annual meeting on association for computational linguistics (ACL), pp 173–180
7. De Marneffe MC, Manning CD (2015) Stanford typed dependencies manual. Stanford University (2015)
8. Demner-Fushman D, Kohli MD, Rosenman MB, Shooshan SE, Rodriguez L, Antani S, Thoma GR, McDonald CJ (2015) Preparing a collection of radiology examinations for distribution and retrieval. J Am Med Inform Assoc 23(2):304–310. https://doi.org/10.1093/jamia/ocv080, http://jamia.oxfordjournals.org/content/jaminfo/early/2015/07/01/jamia.ocv080.1.full.pdf
9. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Computer vision and pattern recognition. IEEE, pp 248–255
10. Dou Q, Chen H, Yu L, Zhao L, Qin J, Wang D, Mok V, Shi L, Heng P (2016) Automatic detection of cerebral microbleeds from mr images via 3D convolutional neural networks. IEEE Trans Med Imaging 35(5):1182–1195
11. Durand T, Thome N, Cord M (2016) Weldon: weakly supervised learning of deep convolutional neural networks. IEEE CVPR
12. Everingham M, Eslami SMA, Van Gool LJ, Williams C, Winn J, Zisserman A (2015) The pascal visual object classes challenge: a retrospective. Int J Comput Vis 111(1):98–136
13. Greenspan H, van Ginneken B, Summers RM (2016) Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans Med Imaging 35(5):1153–1159
14. Hariharan B, Girshick R (2016) Low-shot visual object recognition. arXiv:1606.02819
15. Havaei M, Guizard N, Chapados N, Bengio Y (2016) Hemis: hetero-modal image segmentation. In: MICCAI, (2). Springer, Berlin, pp 469–477
16. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385
17. Hwang S, Kim HE (2015) Self-transfer learning for weakly supervised lesion localization. In: MICCAI, (2). pp 239–246
18. Jaeger S, Candemir S, Antani S, Wáng YXJ, Lu PX, Thoma G (2014) Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. Quant Imaging Med Surg 4(6). http://qims.amegroups.com/article/view/5132
19. Jamaludin A, Kadir T, Zisserman A (2016) Spinenet: automatically pinpointing classification evidence in spinal MRIs. In: MICCAI. Springer, Berlin
20. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. arXiv:1408.5093
21. Johnson J, Karpathy A, Fei-Fei L (2016) Densecap: fully convolutional localization networks for dense captioning. In: CVPR
22. Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: CVPR
23. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA, Bernstein M, Fei-Fei L (2016) Visual genome: connecting language and vision using crowdsourced dense image annotations. https://arxiv.org/abs/1602.07332
24. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
25. Leaman R, Khare R, Lu Z (2015) Challenges in clinical natural language processing for automated disorder normalization. J Biomed Inform 57:28–37. https://doi.org/10.1016/j.jbi.2015.07.010
26. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick L (2014) Microsoft coco: common objects in context. In: ECCV, (5). pp 740–755

27. McClosky D (2009) Any domain parsing: automatic domain adaptation for natural language parsing. Department of Computer Science, Brown University, Thesis
28. Moeskops P, Wolterink J, van der Velden B, Gilhuijs K, Leiner T, Viergever M, Isgum I (2016) Deep learning for multi-task medical image segmentation in multiple modalities. In: MICCAI. Springer, Berlin
29. Open-i: an open access biomedical search engine. https://openi.nlm.nih.gov
30. Oquab M, Bottou L, Laptev I, Sivic J (2015) Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: IEEE CVPR, pp 685–694
31. Pinheiro PO, Collobert R (2015) From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1713–1721
32. Plummer B, Wang L, Cervantes C, Caicedo J, Hockenmaier J, Lazebnik S (2015) Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV
33. Qiao R, Liu L, Shen C, van den Hengel A (2016) Less is more: zero-shot learning from online textual documents with noise suppression. In: CVPR
34. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: MICCAI. Springer, Berlin, pp 234–241
35. Roth H, 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. Springer, Berlin, pp 556–564
36. Roth HR, Lu L, Seff A, Cherry KM, Hoffman J, Wang S, Liu J, Turkbey E, Summers RM (2014) A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations. In: MICCAI. Springer, Berlin, pp 520–527
37. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
38. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
39. Setio A, Ciompi F, Litjens G, Gerke P, Jacobs C, van Riel S, Wille M, Naqibullah M, Sánchez C, van Ginneken B (2016) Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks. IEEE Trans Med Imaging 35(5):1160–1169
40. Shin H, Lu L, Kim L, Seff A, Yao J, Summers R (2016) Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation. J Mach Learn Res 17:1–31
41. Shin H, Roberts K, Lu L, Demner-Fushman D, Yao J, Summers R (2016) Learning to read chest x-rays: recurrent neural cascade model for automated image annotation. In: CVPR
42. Shin H, Roth H, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers R (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learnings. IEEE Trans Med Imaging 35(5):1285–1298
43. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
44. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
45. Tapaswi M, Zhu Y, Stiefelhagen R, Torralba A, Urtasun R, Fidler S (2015) Movieqa: understanding stories in movies through question-answering. In: ICCV
46. Vendrov I, Kiros R, Fidler S, Urtasun R (2016) Order-embeddings of images and language. In: ICLR
47. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: CVPR, pp 3156–3164
48. Wilke HJ, Kümin M, Urban J (2016) Genodisc dataset: the benefits of multi-disciplinary research on intervertebral disc degeneration. Eur Spine J. http://www.physiol.ox.ac.uk/genodisc/

49. Wu Q, Wang P, Shen C, Dick A, van den Hengel A (2016) Ask me anything: free-form visual question answering based on knowledge from external sources. In: CVPR
50. Yao J, et al (2016) A multi-center milestone study of clinical vertebral ct segmentation. Comput Med Imaging Graph 49(4):16–28
51. Young P, Lai A, Hodosh M, Hockenmaier J (2014) From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. In: TACL
52. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2015) Learning deep features for discriminative localization. arXiv:1512.04150
53. Zhu Y, Groth O, Bernstein M, Fei-Fei L (2016) Visual7w: grounded question answering in images. In: CVPR

Chapter 19
Automatic Classification and Reporting of Multiple Common Thorax Diseases Using Chest Radiographs

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu and Ronald M. Summers

Abstract Chest X-rays are one of the most common radiological examinations in daily clinical routines. Reporting thorax diseases using chest X-rays is often an entry-level task for radiologist trainees. Yet, reading a chest X-ray image remains a challenging job for learning-oriented machine intelligence, due to (1) shortage of large-scale machine-learnable medical image datasets, and (2) lack of techniques that can mimic the high-level reasoning of human radiologists that requires years of knowledge accumulation and professional training. In this paper, we show the clinical free-text radiological reports that accompany X-ray images in hospital picture and archiving communication systems can be utilized as a priori knowledge for tackling these two key problems. We propose a novel text-image embedding network (TieNet) for extracting the distinctive image and text representations. Multi-level

Wang—This work was done during his fellowship at National Institutes of Health Clinical Center.
Lu—This work was done during his employment at National Institutes of Health Clinical Center.

X. Wang (✉)
Nvidia Corporation, Bethesda, MD 20814, USA
e-mail: xiaosong.wang@live.com

Y. Peng · Z. Lu
National Center for Biotechnology Information,
National Library of Medicine, National Institutes of Health, Bethesda, MD 20892, USA
e-mail: yifan.peng@nih.gov

Z. Lu
e-mail: luzh@nih.gov

L. Lu
PAII Inc., Bethesda Research Lab, 6720B Rockledge Drive, Ste 410, Bethesda,
MD 20817, USA
e-mail: le.lu@paii-labs.com; lelu@cs.jhu.edu

Johns Hopkins University, Baltimore, MD, USA

R. M. Summers
Imaging Biomarkers and Computer-Aided Diagnosis Laboratory, Radiology and Imaging
Sciences Department, Clinical Center, National Institutes of Health, Bethesda,
MD 20892, USA
e-mail: rms@nih.gov

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_19

attention models are integrated into an end-to-end trainable CNN-RNN architecture for highlighting the meaningful text words and image regions. We first apply TieNet to classify the chest X-rays by using both image features and text embeddings extracted from associated reports. The proposed auto-annotation framework achieves high accuracy (over 0.9 on average in AUCs) in assigning disease labels for our hand-label evaluation dataset. Furthermore, we transform the TieNet into a chest X-ray reporting system. It simulates the reporting process and can output disease classification and a preliminary report together, with X-ray images being the only input. The classification results are significantly improved (6% increase on average in AUCs) compared to the state-of-the-art baseline on an unseen and hand-labeled dataset (OpenI).

19.1 Introduction

In the last decade, challenging tasks in computer vision have gone through different stages, from sole image classification to multi-category multi-instance classification/detection/segmentation to more complex cognitive tasks that involve understanding and describing the relationships of object instances inside the images or videos. The rapid and significant performance improvement is partly driven by publicly accessible of the large-scale image and video datasets with quality annotations, e.g., ImageNet [8], PASCAL VOC [10], MS COCO [22], and Visual Genome [18] datasets. In particular, ImageNet pre-trained deep Convolutional Neural Network (CNN) models [15, 19, 21] has become an essential basis (indeed an advantage) for many higher level tasks, e.g., recurrent neural network (RNN) based image captioning [11, 17, 30, 34], Visual Question Answering [27, 36, 38, 42], and instance relationship extraction [6, 14, 16].

On the contrary, there are few publicly available large-scale image datasets in the medical image domain. Conventional means of annotating natural images, e.g., crowdsourcing, cannot be applied to medical images due to the fact that these tasks often require years of professional training and domain knowledge. On the other hand, radiological raw data (e.g., images, clinical annotations, and radiological reports) have been accumulated in many hospitals' picture archiving and communication systems (PACS) for decades. The main challenge is how to transform those retrospective radiological data into a machine-learnable format. Accomplishing this with chest X-rays represents a major milestone in the medical imaging community [35].

Different from current deep learning models, radiologists routinely observe multiple findings when they read medical images and compile radiological reports. One main reason is that these findings are often correlated. For instance, liver metastases can spread to regional lymph nodes or other body parts. By obtaining and maintaining a holistic picture of relevant clinical findings, a radiologist will be able to make a more accurate diagnosis. To our best knowledge, developing a universal or multi-purpose CAD framework, which is capable of detecting multiple disease types

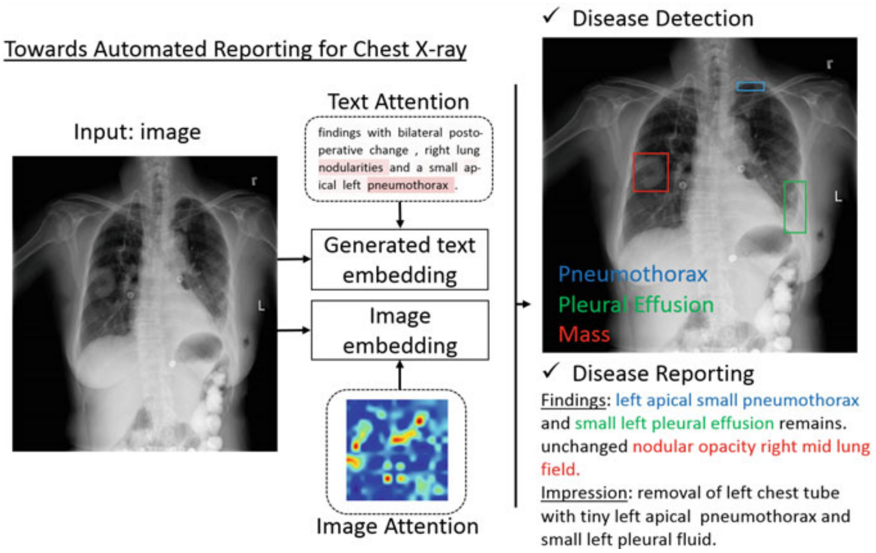

Towards Automated Reporting for Chest X-ray
Input: image
Text Attention
findings with bilateral postoperative change , right lung nodularities and a small apical left pneumothorax .
Generated text embedding
Image embedding
Image Attention
✓ Disease Detection
Pneumothorax
Pleural Effusion
Mass
✓ Disease Reporting
Findings: left apical small pneumothorax and small left pleural effusion remains. unchanged nodular opacity right mid lung field.
Impression: removal of left chest tube with tiny left apical pneumothorax and small left pleural fluid.

Fig. 19.1 Overview of the automated chest X-ray reporting framework. Multi-level attention are introduced

in a seamless fashion, is still a challenging task. However, such a framework is a crucial part to build an automatic radiological diagnosis and reporting system.

Toward this end, we investigate how free-text radiological reports can be exploited as a priori knowledge using an innovative text-image embedding network. We apply this novel system in two different scenarios. We first introduce a new framework for auto-annotation of the chest X-rays by using both images features and text embeddings extracted from associated reports. Multi-level attention models are integrated into an end-to-end trainable CNN-RNN architecture for highlighting the meaningful text words and image regions. In addition, we convert the proposed annotation framework into a chest X-ray reporting system (as shown in Fig. 19.1). The system stimulates the real-world reporting process by outputting disease classification and generating a preliminary report spontaneously. The text embedding learned from the retrospective reports is integrated into the model as a priori knowledge and the joint learning framework boosts the performance in both tasks in comparison to previous state of the art.

In this chapter, we discussed four key aspects of automated reporting: (1) We introduced the text-image embedding network, which is a multi-purpose end-to-end trainable multi-task CNN-RNN framework. (2) We show how raw report data, together with paired image, can be utilized to produce meaningful attention-based image and text representations using the proposed TieNet. Raw reports are not that convenient for publicly sharing due to the difficulty of PHI anonymization while attention-encode sentence embedding is a feasible solution for sharing associated diagnosis info for each image. (3) We outline how the developed text and image embeddings are able to boost the auto-annotation framework and achieve extremely high accuracy for chest X-ray labeling. (4) Finally, we present a novel image classification framework which takes images as the sole input, but uses the paired text-image representations from training as a prior knowledge injection, in order to produce improved classification scores and preliminary report generations.

Importantly, we validate our approach on three different datasets and the TieNet improves the image classification result (6% increase on average in area under the curve (AUC) for all disease categories) in comparison to the state of the art on an unseen and hand-labeled dataset (OpenI [7]) from other institutes. Our multi-task training scheme can help not only the image classification but also the report generation by producing reports with higher BLEU scores than the baseline method.

19.2 Previous Works in CAD

Computer-aided detection (CADe) and diagnosis (CADx) have long been a major research focus in medical image processing [5]. In recent years, deep learning models start to outperform conventional statistical learning approaches in various tasks, such as automated classification of skin lesions [9], detection of liver lesions [4], and detection of pathological-image findings [40]. However, current CADe methods typically target one particular type of disease or lesion, such as lung nodules, colon polyps or lymph nodes [24].

Wang et al. [35] provide a recent and prominent exception, where they introduced a large-scale chest X-ray dataset by processing images and their paired radiological reports (extracted from their institutional PACS database) with natural language processing (NLP) techniques. The publicly available dataset contains 112, 120 front-view chest X-ray images of 30, 805 unique patients. However, radiological reports contain richer information than simple disease binary labels, e.g., disease location and severity, which should be exploited in order to fully leverage existing PACS datasets. Thus, we differ from Wang et al.'s approach by leveraging this rich text information in order to produce an enhanced system for chest X-ray CADx.

In vision of visual captioning, our work is closest to [27, 29, 33, 37, 38]. Xu et al. [37] first introduced the sequence-to-sequence model and spatial attention model into the image captioning task. They conditioned the long short-term memory (LSTM) decoder on different parts of the input image during each decoding step, and the attention signal was determined by the previous hidden state and CNN features. Vinyals et al. [33] cast the syntactical parsing problem as a sequence-to-sequence learning task by linearizing the parsing tree. Pederoli et al. [29] allowed a direct association between caption words and image regions. More recently, multi-attention models [27, 38] extract salient regions and words from both image and text and then combine them together for better representations of the pair. In medical imaging domain, Shin et al. [32] proposed to correlate the entire image or saliency regions with MeSH terms. Promising results [41] are also reported in summarizing the findings in pathology images using task-oriented reports in the training. The difference between our model and theirs lies in that we employ multi-attention models with a mixture of image and text features in order to provide more salient and meaningful embeddings for the image classification and report generation task. Apart from visual attention, text-based attention has also been increasingly applied in deep learning for NLP [2, 26, 31]. It attempts to relieve one potential problem that the traditional encoder–decoder framework faces, which is that the input is long or very information-rich and selective encoding is not possible. The attention mechanism attempts to ease the above problems by allowing the decoder to refer back to the input sequence [23, 25, 39]. To this end, our work closely follows the one used in [23] where they extracted an interpretable sentence embedding by introducing self-attention. Our model paired both the attention-based image and text representation from training as a prior knowledge injection to produce improved classification scores.

19.3 Multi-level Attention in a Unified Framework

The radiological report is a summary of all the clinical findings and impressions determined during examination of a radiography study. A sample report is shown in Fig. 19.1. It usually contains richer information than just disease keywords, but also may consist of negation and uncertainty statements. In the “findings” section, a list of normal and abnormal observations will be listed for each part of the body examined in the image. Attributes of the disease patterns, e.g., specific location and severity, will also be noted. Furthermore, critical diagnosis information is often presented in the “impression” section by considering all findings, patient history, and previous studies. Suspicious findings may cause recommendations for additional or follow-up imaging studies. As such, reports consist of a challenging mixture of information and a key for machine learning is extracting useful parts for particular applications. In addition to mining the disease keywords [35] as a summarization of the radiological reports, we want to learn a text embedding to capture the richer information contained in raw reports. Multiple levels of attention mechanisms are introduced here; Fig. 19.2 illustrates the proposed text-image embedding network. First, we discuss two kinds of attention we develop and integrate, i.e., attention-encoded text embedding (AETE, attention on textual information) and saliency weighted global average pooling (SW-GAP, attention on image spatial information). Then, we presented a basic end-to-end trainable CNN-RNN architecture and an enhanced version with afore-discussed attention mechanisms integrated. Finally, we outline the joint learning loss function used to optimize the framework.

19.3.1 AETE: Attention on Text

To compute a global text representation, we use an approach that closely follows the one used in [23]. More specifically, we use attention to combine the most salient

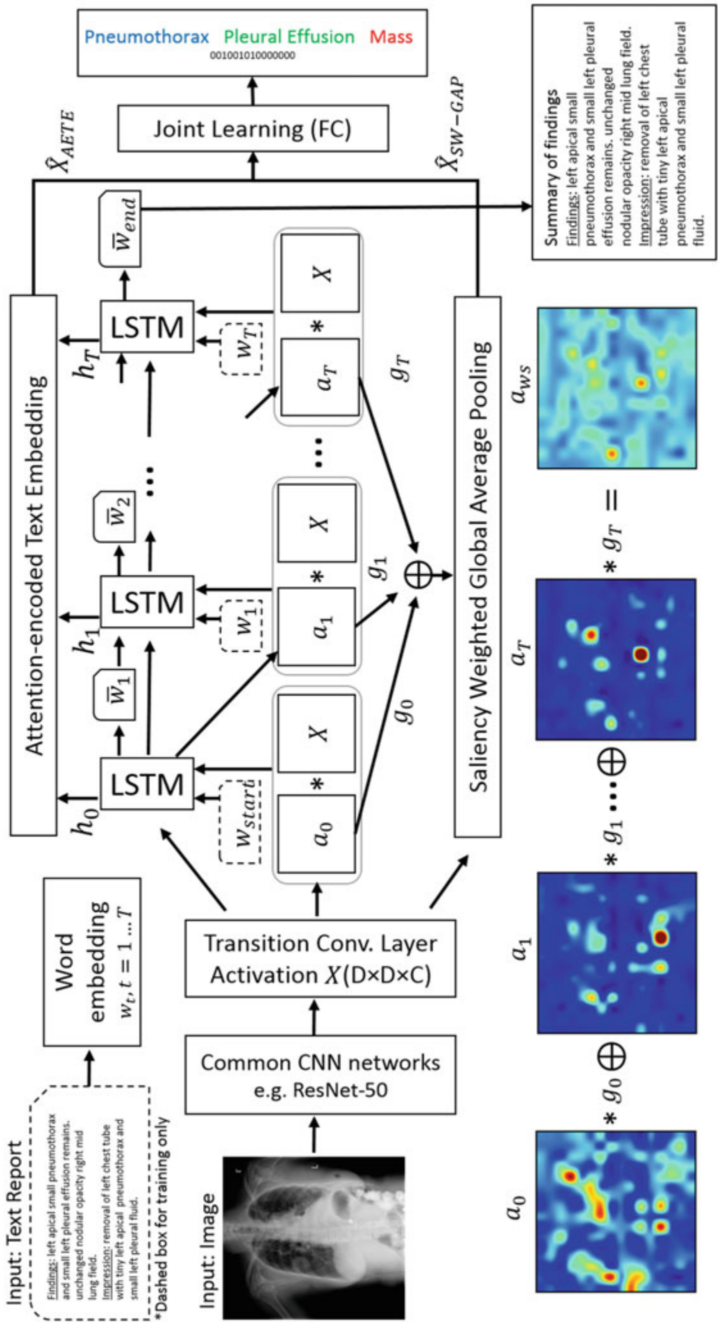

Input: Text Report
Findings: left apical small pneumothorax and small left pleural effusion remains. unchanged nodular opacity right mid lung field.
Impression: removal of left chest tube with tiny left apical pneumothorax and small left pleural fluid.
*Dashed box for training only
Input: Image
Word embedding $w_t, t = 1 \dots T$
Common CNN networks e.g. ResNet-50
Transition Conv. Layer Activation X(D×D×C)
Attention-encoded Text Embedding
LSTM
LSTM
LSTM
h_0
h_1
h_T
$\bar{w}_1$
$\bar{w}_2$
$\bar{w}_{end}$
w_{start}
w_1
w_T
a_0 * X
a_1 * X
a_T * X
g_0
g_1
g_T
⊕
Saliency Weighted Global Average Pooling
$\hat{X}_{AETE}$
$\hat{X}_{SW-GAP}$
Joint Learning (FC)
Pneumothorax Pleural Effusion Mass
001001010000000
Summary of findings
Findings: left apical small pneumothorax and small left pleural effusion remains. unchanged nodular opacity right mid lung field.
Impression: removal of left chest tube with tiny left apical pneumothorax and small left pleural fluid.
a_0
a_1
a_T
a_{ws}
$* g_0 \oplus$
$* g_1 \cdots \oplus$
$* g_T =$

Fig. 19.2 Framework of the proposed chest X-ray auto-annotation and reporting framework. Multi-level attention are introduced to produce saliency-encoded text and image embeddings

portions of the RNN hidden states. Let $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_T)$ be the $d_h \times T$ matrix of all the hidden states. The attention mechanism outputs a $r \times T$ matrix of weights $\mathbf{G}$ as

$$\mathbf{G} = softmax(\mathbf{W}_{s2}\, tanh(\mathbf{W}_{s1}\, \mathbf{H})), \tag{19.1}$$

where r is the number of global attention we want to extract from the sentence, and $\mathbf{W}_{s1}$ and $\mathbf{W}_{s2}$ are s-by-d_h and r-by-s matrices, respectively. s is a hyperparameter governing the dimensionality, and therefore maximum rank, of the attention-producing process.

With the attention calculated, we compute an $r \times d_h$ embedding matrix, $\mathbf{M} = \mathbf{GH}$, which in essence executes r weighted sums across the T hidden states, aggregating them together into r representations. Each row of $\mathbf{G}$, denoted $\mathbf{g}^i$ $(i \in \{1 \ldots r\})$, indicates how much each hidden state contributes to the final embedded representation of $\mathbf{M}$. We can thus draw a heat map for each row of the embedding matrix M (see Fig. 19.5 for examples). This way of visualization gives hints on what is encoded in each part of the embedding, adding an extra layer of interpretation.

To provide a final global text embedding of the sentences in the report, the AETE executes max-over-r pooling across $\mathbf{M}$, producing an embedding vector $\hat{\mathbf{X}}_{AETE}$ with size d_h.

19.3.2 SW-GAP: Attention on Image

In addition to using attention to provide a more meaningful text embedding, our goal is also to produce improved visual embeddings for classification. For this purpose, we re-use the attention mechanism, $\mathbf{G}$, except that we perform a max-over-r operation, producing a sequence of saliency values, $g_t(t = 1, \ldots, T)$, for each word, $\mathbf{w}_t$. These saliency values are used to weight and select the spatial attention maps, $\mathbf{a}_t$, generated at each time point:

$$\mathbf{a}_{ws}(x, y) = \sum_t \mathbf{a}_t(x, y) * g_t. \tag{19.2}$$

This map is encoded with all spatial saliency regions guided by the text attention. We use this map to highlight the spatial regions of $\mathbf{X}$ with more meaningful information:

$$\hat{\mathbf{X}}_{SW-GAP}(c) = \sum_{(x,y)} \mathbf{a}_{ws}(x, y) * \mathbf{X}(x, y, c), \tag{19.3}$$

where $x, y \in \{1 \ldots D\}$ and $\hat{\mathbf{X}}_{SW-GAP}$ is a 1-by-C vector representing the global visual information, guided by both text- and visual-based attention. The lower part of Fig. 19.2 illustrates an example of such pooling strategy.

19.3.3 Overall CNN-RNN Model

As shown in Fig. 19.2, our end-to-end trainable CNN-RNN model takes an image I and a sequence of 1-of-V encoded words:

$$\mathbf{S} = \{\mathbf{w}_1, \ldots, \mathbf{w}_T\}, \mathbf{w}_t \in \mathbb{R}^V, \tag{19.4}$$

where $\mathbf{w}_t$ is a vector standing for a d_w-dimensional word embedding for the t-th word in the report, V is the size of the vocabulary, and T is the length of the report. The initial CNN component uses layers borrowed from ImageNet pre-trained models for image classification, e.g., ResNet-50 (from Conv1 to Res5c).

The CNN component additionally includes a convolutional layer (transition layer) to manipulate the spatial grid size and feature dimension.

Our RNN is based off of Xu et al.'s visual image spatial attention model [37] for image captioning. The convolutional activations from the transition layer, denoted as $\mathbf{X}$, initialize the RNN's hidden state, $\mathbf{h}_t$, where a fully connected embedding, $\phi(\mathbf{X})$, maps the size d_X transition layer activations to the LSTM state space of dimension d_h. In addition, X is also used as one of the RNN's input. However, following Xu et al. [37], our sequence-to-sequence model includes a deterministic and soft visual spatial attention, $\mathbf{a}_t$, that is multiplied element-wise to $\mathbf{X}$ before the latter is inputted to the RNN. At each time step, the RNN also outputs the subsequent attention map, $\mathbf{a}_{t+1}$.

In addition to the soft-weighted visual features, the RNN also accepts the current word at each time step as input. We adopt standard LSTM units [13] for the RNN. The transition to the next hidden state can then be denoted as

$$\mathbf{h}_t = LSTM\left([\mathbf{w}_t, \mathbf{a}_t, \mathbf{X}], \mathbf{h}_{t-1}\right). \tag{19.5}$$

The LSTM produces the report by generating one word at each time step conditioned on a context vector, i.e., the previous hidden state $\mathbf{h}_t$, the previously generated words $\mathbf{w}_t$, and the convolutional features of $\mathbf{X}$ whose dimension is $D \times D \times C$. Here $D = 16$ and $C = 1024$ denote the spatial and channel dimensions, respectively. Once the model is trained, reports for a new image can be generated by sequentially sampling $\mathbf{w}_t \sim p(\mathbf{w}_t|\mathbf{h}_t)$ and updating the state using Eq. 19.5.

The end-to-end trainable CNN-RNN model provides a powerful means to process both text and images. However, our goal is also to obtain an interpretable global text and visual embedding for the purposes of classification. For this reason, we introduce two key enhancements in the form of the AETE and SW-GAP.

19.3.4 Joint Learning

With global representations computed for both the image and report, these must be combined together to produce the final classification. To accomplish this, we concatenate the two forms of representations $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_{AETE}; \hat{\mathbf{X}}_{SW-GAP}]$ and use a final fully connected layer to produce the output for multi-label classification. The intuition behind our model is that the connection between the CNN and RNN network will benefit the training of both because the image activations can be adjusted for the text embedding task and salient image features could be extracted by pooling based on high text saliency.

In a similar fashion as Wang et al. [35], we define an M-dimensional disease label vector $\mathbf{y} = [y_1, \ldots, y_m, \ldots, y_M], y_m \in \{0, 1\}$ for each case and $M = 15$ indicates the number of classes. y_m indicates the presence with respect to a pathology or "no finding" (of listed disease categories) in the image. Here, we adopt the NLP-mined labels provided by [35] as the "ground truth" during the training.

The instance numbers for different disease categories are highly unbalanced, from hundreds to tens of thousands. In addition to the positive/negative balancing introduced in [35], we add weights to instances associated with different categories,

$$\begin{aligned} L_m(f(I, \mathbf{S}), \mathbf{y}) = \beta_P \sum_{y_m=1} - \ln(f(I, \mathbf{S})) \cdot \lambda_m \\ + \beta_N \sum_{y_m=0} - \ln(1 - f(I, \mathbf{S})) \cdot \lambda_m, \end{aligned} \tag{19.6}$$

where $\beta_P = \frac{|N|}{|P|+|N|}$ and $\beta_N = \frac{|P|}{|P|+|N|}$. $|P|$ and $|N|$ are the total number of images with at least one disease and with no diseases, respectively. $\lambda_m = (Q - Q_m)/Q$ is a set of precomputed class-wised weights, where Q and Q_m are the total number of images and the number of images that have disease label m. λ_m will be larger if the number of instances from class m is small.

Because the TieNet can also generate text reports, we also optimize the RNN generative model loss [37], L_R. Thus, the overall loss is composed of two parts, the sigmoid cross-entropy loss L_M for the multi-label classification and the loss L_R from the RNN generative model [37],

$$L_{overall} = \alpha L_M + (1 - \alpha) L_R, \tag{19.7}$$

where α is added to balance the large difference between the two loss types.

19.4 Applications

19.4.1 Annotation of Chest X-Ray Images

One straightforward application of the TieNet is the auto-annotation task to mine image classification labels. By omitting the generation of sequential words, we accumulate and back-propagate only the classification loss for better text-image embeddings in image classification. Here, we use the NLP-mined disease labels as "ground truth" in the training. Indeed, we want to learn a mapping between the input image–report pairs and the image labels. The report texts often contain more easy-to-learn features than the image side. The contribution of both sources to the final classification prediction should be balanced via either controlling the feature dimensions or drop-off partial of the "easy-to-learn" data during training.

19.4.2 Automatic Reporting of Thorax Diseases

For a more difficult but real-world scenario, we transform the image-text embedding network to serve as a unified system of image classification and report generation when only the unseen image is available. During the training, both image and report are fed and two separate losses are computed as stated above, i.e., the loss for image classification and the loss for sequence-to-sequence modeling. While testing, only the image is required as the input. The generated text contained the learned text embedding recorded in the LSTM units and later used in the final image classification task. The generative model we integrated into the text-image embedding network is the key to associate an image with its attention encoded text embedding.

19.5 Experiments

19.5.1 Datasets for Evaluation

I. ChestX-ray14 [35]
ChestX-ray14 is a recently released benchmark dataset for common thorax disease classification and localization. It consists of 14 disease labels that can be observed in chest X-ray, i.e., atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, pneumothorax, consolidation, edema, emphysema, fibrosis, pleural thickening, and hernia. The NLP-mined labels are used as "ground truth" for model training and testing throughout the experiments. We adopt the patient-level data splits published with the data.[1]

[1] https://nihcc.app.box.com/v/ChestXray-NIHCC.

II. Hand-labeled
In addition to NLP-mined labels, we randomly select 900 reports from the testing set and have two radiologists to annotate the 14 categories of findings for the evaluation purpose. A trial set of 30 reports was first used to synchronize the criterion of annotation between two annotators. Then, each report was independently annotated by two annotators. In this paper, we used the inter-rater agreement (IRA) to measure the consistency between two observers. The resulting Cohen's kappa is 84.3%. Afterward, the final decision was adjudicated between two observers on the inconsistent cases.

III. OpenI [7]
is a publicly available radiography dataset collected from multiple institutes by Indiana University. Using the OpenI API, we retrieved 3,851 unique radiology reports and 7,784 associated frontal/lateral images where each OpenI report was annotated with key concepts (MeSH words) including body parts, findings, and diagnoses. For consistency, we use the same 14 categories of findings as above in the experiments. In our experiments, only 3,643 unique front-view images and corresponding reports are selected and evaluated.

19.5.2 Report Vocabulary

We use all 15,472 unique words in the training set that appear at least twice. Words that appear less frequently are replaced by a special out-of-vocabulary token, and the start and the end of the reports are marked with a special ⟨START⟩ and ⟨END⟩ token. The pre-trained word embedding vectors was learned on PubMed articles using the gensim word2vec implementation with the dimensionality set to 200.[2] The word embedding vectors will be evolved along with other LSTM parameters.

19.5.3 Evaluation Metrics

To compare previous state-of-the-art works, we choose different evaluation metrics for different tasks so as to maintain consistency with data as reported in the previous works. Receiver operating curves (ROC) are plotted for each disease category to measure the image classification performance and afterward, Areas Under Curve (AUC) are computed, which reflect the overall performance as a summary of different operating points. To assess the quality of generated text report, BLEU scores [28], METEOR [3], and ROUGE-L [20] are computed between the original reports and the generated ones. Those measures reflect the word overlapping statistics between

[2] https://radimrehurek.com/gensim/models/word2vec.html.

two text corpora. However, we believe their capabilities are limited for showing the actual accuracy of disease words (together with their attributes) overlapping between two text corpora.

19.5.4 Details on Training

The LSTM model contains a 256-dimensional cell and $s = 2000$ in $\mathbf{W}_{s1}$ and $\mathbf{W}_{s2}$ for generating the attention weights $\mathbf{G}$. During training, we use 0.5 dropout on the MLP and 0.0001 for L2 regularization. We use Adam optimizer with a mini-batch size of 32 and a constant learning rate of 0.001.

In addition, our self-attention LSTM has a hidden layer with 350 units. We choose the matrix embedding to have five rows (the r), and a coefficient of 1 for the penalization term. All the models are trained until convergence is achieved and the hyperparameters for testing is selected according to the corresponding best validation set performance.

Our text-image embedding network is implemented based on TensorFlow [1] and Tensorpack.[3] The ImageNet pre-trained model, i.e., ResNet-50 [12] is obtained from the Caffe model zoo and converted into the TensorFlow compatible format. The proposed network takes the weights from the pre-trained model and fixes them during the training. Other layers in the network are trained from scratch. In a similar fashion as introduced in [35], we reduce the size of mini-batch to fit the entire model in each GPU while we accumulate the gradients for a number of iterations and also across a number of GPUs for better training performance. The DCNN models are trained using a Dev-Box Linux server with 4 Titan X GPUs.

19.5.5 Evaluation on Image Annotation

Figure 19.3 illustrates the ROC curves for the image classification performance with three different inputs evaluated on three different testing sets, i.e., ChestX-ray14 testing set (ChestX-ray14), the hand-labeled set (Hand-labeled) and the OpenI set (OpenI). Separate curves are plotted for each disease categories and “No finding”. Here, two different auto-annotation frameworks are trained by using different inputs, i.e., taking reports only (R) and taking image–report pairs (I+R) as inputs. When only the reports are used, the framework will not have the saliency weighted global average pooling path. In such way, we can get a sense of how the features from text path and image path individually contribute to the final classification prediction.

We train the proposed auto-annotation framework using the training and validation sets from the ChestX-ray14 dataset and test it on all three testing sets, i.e., ChestX-ray14, hand-labeled and OpenI. Table 19.1 shows the AUC values for each class

[3] https://github.com/ppwwyyxx/tensorpack/.

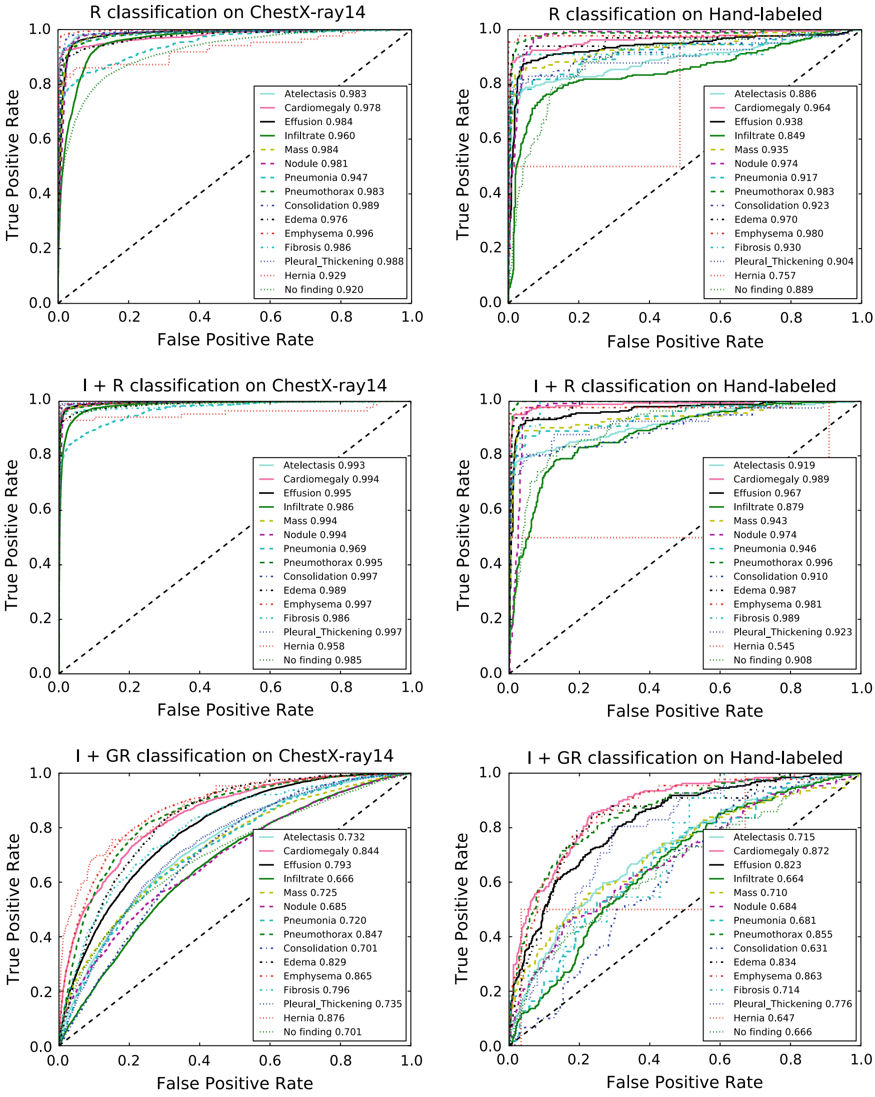


Fig. 19.3 A comparison of classification performance using ROC curves with different testing inputs, i.e., report (R), image+report (I+R), and image+generative report (I+GR)

Table 19.1 Evaluation of image classification results (AUCs) on ChestX-ray14, hand-labeled and OpenI dataset. Performances are reported on four methods, i.e., multi-label classification based on report (R), image + report (I+R), image [35], and image + generative report (I+GR)

Disease	ChestX-ray14					Hand-labeled					OpenI				
	R	I+R	I [35]	I+GR	#	R	I+R	I [35]	I+GR	#	R	I+R	I [35]	I+GR	#
Atelectasis	0.983	0.993	0.700	0.732	3255	0.886	0.919	0.680	0.715	261	0.981	0.976	0.702	0.774	293
Cardiomegaly	0.978	0.994	0.810	0.844	1065	0.964	0.989	0.820	0.872	185	0.944	0.962	0.803	0.847	315
Effusion	0.984	0.995	0.759	0.793	4648	0.938	0.967	0.780	0.823	257	0.968	0.977	0.890	0.899	140
Infiltration	0.960	0.986	0.661	0.666	6088	0.849	0.879	0.648	0.664	271	0.981	0.984	0.585	0.718	57
Mass	0.984	0.994	0.693	0.725	1712	0.935	0.943	0.696	0.710	93	0.959	0.903	0.756	0.723	14
Nodule	0.981	0.994	0.668	0.685	1615	0.974	0.974	0.662	0.684	130	0.967	0.960	0.647	0.658	102
Pneumonia	0.947	0.969	0.658	0.720	477	0.917	0.946	0.724	0.681	55	0.983	0.994	0.642	0.731	36
Pneumothorax	0.983	0.995	0.799	0.847	2661	0.983	0.996	0.784	0.855	166	0.960	0.960	0.631	0.709	22
Consolidation	0.989	0.997	0.703	0.701	1815	0.923	0.910	0.609	0.631	60	0.969	0.989	0.790	0.855	28
Edema	0.976	0.989	0.805	0.829	925	0.970	0.987	0.815	0.834	33	0.984	0.995	0.799	0.879	40
Emphysema	0.996	0.997	0.833	0.865	1093	0.980	0.981	0.835	0.863	44	0.849	0.868	0.675	0.792	94
Fibrosis	0.986	0.986	0.786	0.796	435	0.930	0.989	0.688	0.714	11	0.985	0.960	0.744	0.791	18
PT	0.988	0.997	0.684	0.0.735	1143	0.904	0.923	0.679	0.776	41	0.948	0.953	0.691	0.749	52
Hernia	0.929	0.958	0.871	0.876	86	0.757	0.545	0.864	0.647	2	–	–	–	–	0
NoFinding	0.920	0.985	–	0.701	9912	0.889	0.908	–	0.666	85	0.933	0.936	–	0.747	2789
AVG	0.976	0.989	0.745	0.772	–	0.922	0.925	0.735	0.748	–	0.960	0.965	0.719	0.779	–
#wAVG	0.978	0.992	**0.722**	**0.748**	–	0.878	0.900	**0.687**	**0.719**	–	0.957	0.966	**0.741**	**0.798**	–

computed from the ROC curves shown in Fig. 19.3. The auto-annotation framework achieves high performance on both ChestX-ray14 and Hand-labeled, i.e., over 0.87 in AUC with reports alone as the input and over 0.90 in AUC with image–report pairs on sample number weighted average ($\#wAVG$). The combination of image and report demonstrates the supreme advantage in this task. In addition, the auto-annotation framework trained on ChestX-ray14 performed equivalently on OpenI. It indicates that the model trained on a large-scale image dataset could easily be generalized to the unseen data from other institutes. The model trained solely based on images could also be generalized well to the datasets from other sources. In this case, both the proposed method and the one in [35] are able to perform equally well on all three testing sets.

19.5.6 Evaluation on Classification and Automated Reporting

When the TieNet is switched to an automatic disease classification and reporting system, it takes a single image as the input and is capable of outputting a multi-label prediction and corresponding radiological report together. The ROC curves on the right in Fig. 19.3 and Table 19.1 show the image classification performance produced by the multi-purpose reporting system. The AUCs from our TieNet (I+GR) demonstrate the consistent improvement in AUCs (2.3–5.7% on $\#wAVG$ for all the disease categories) across all three datasets. The multi-label classification framework [35] serves as a baseline model that also takes solely the images. Furthermore, the performance improvement achieved on the Hand-labeled and OpenI datasets (with ground truth image labels) is even larger than the performance gain on ChestX-ray14 (with NLP-mined labels). It indicates that the TieNet is able to learn more meaningful and richer text embeddings directly from the raw reports and correct the inconsistency between embedded features and erroneous mined labels.

Table 19.2 shows that the generated reports from our proposed system obtain higher scores in all evaluation metrics in comparison to the baseline image captioning model [37]. It may be because the gradients from RNN are back-propagated to the

Table 19.2 Evaluation of generated reports in ChestX-ray14 testing set using BLEU, METEOR, and ROUGE-L

	Captioning [37]	TieNet I+GR
BLEU-1	0.2391	0.2860
BLEU-2	0.1248	0.1597
BLEU-3	0.0861	0.1038
BLEU-4	0.0658	0.0736
METEOR	0.1024	0.1076
ROUGE-L	0.1988	0.2263

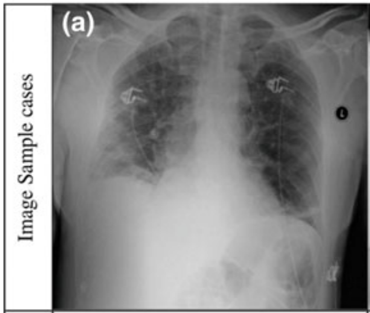
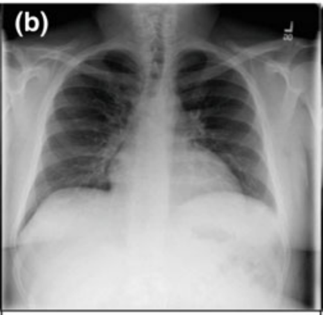
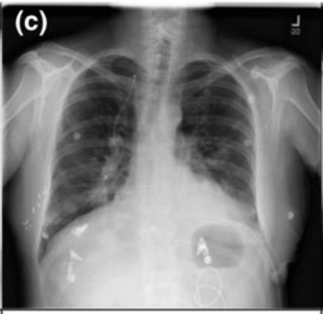
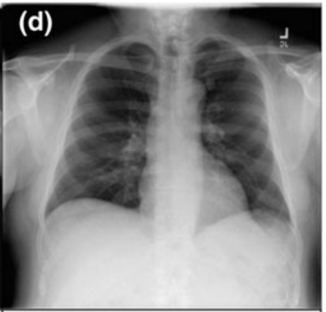
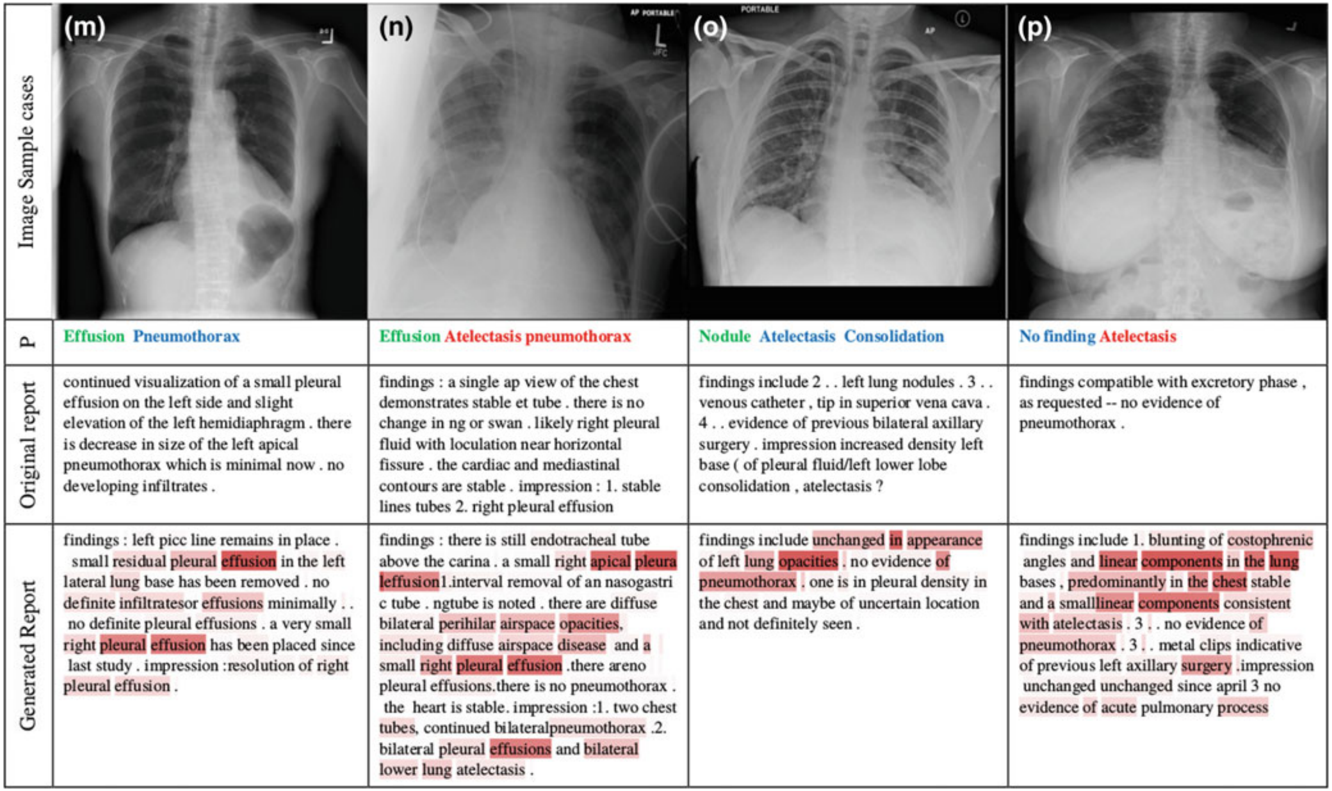
Image Sample cases				
P	Atelectasis Effusion	No finding	Nodule Pneumothorax Mass Consolidation	Mass
Original report	findings : a single ap view of the chest demonstrates increasing bibasilar interstitial opacities with decreased overall aeration . increasing blunting of right costophrenic angle. … impression : increasing bibasilar atelectasis with possible development of right pleural effusion .	Normal no evidence of lung infiltrate .	findings : heart and mediastinum unchanged . multiple lung nodules . evidence of recent left chest surgery with left chest tube in place . very small left apical pneumothorax . lungs unchanged , no evidence of acute infiltrates . impression : stable chest .	findings : large left suprahilar and infrahilar masses as well as the well circumscribed nodule the level of the aortic knob . the right infrahilar mass as well . no effusion . impression : metastatic lung disease .
Generated Report	findings : a single ap view of the chest demonstrates unchanged bilateral reticular opacities , consider atelectasis . continued left basilar atelectasis . no evidence of developing infiltrate . the cardiac and mediastinal contours are stable . impression : no evidence of developing infiltrate .	findings : pa and lateral views of the chest demonstrate lungs that are clear without focal mass , infiltrate or effusion . cardiomediastinal silhouette is normal size and contour . pulmonary vascularity is normal in caliber and distribution . impression : no evidence of acute pulmonary pathology	findings : pa and lateral views of the chest demonstrate unchanged bilateral chest tubes . again pulmonary nodules are seen on the right and cardiac silhouette unchanged . the cardiac and mediastinal contours are stable . impression : 1. bilateral masses and left lower lung field consolidation . 2.new bilateral lung masses .	comparison is to previous upright study of no significant interval change is seen in the appearance of the chest . the mediastinal soft tissue and pulmonary vascularity are stable . there are blastic bone lesions in the chest . bones , soft tissues are normal . the lung fields are clear . there are calcified lymph nodes in the left lower lung . impression : . sclerotic lesions in the left humeral , consistent with metastasis.

Fig. 19.4 Four sample image classification predictions (P) along with original and generated reports. Text attention are highlighted over the generated text. Correct predication is marked in green, false prediction in red, and missing prediction in blue



P	Effusion Pneumothorax	Effusion Atelectasis pneumothorax	Nodule Atelectasis Consolidation	No finding Atelectasis
Original report	continued visualization of a small pleural effusion on the left side and slight elevation of the left hemidiaphragm . there is decrease in size of the left apical pneumothorax which is minimal now . no developing infiltrates .	findings : a single ap view of the chest demonstrates stable et tube . there is no change in ng or swan . likely right pleural fluid with loculation near horizontal fissure . the cardiac and mediastinal contours are stable . impression : 1. stable lines tubes 2. right pleural effusion	findings include 2 . . left lung nodules . 3 . . venous catheter , tip in superior vena cava . 4 . . evidence of previous bilateral axillary surgery . impression increased density left base (of pleural fluid/left lower lobe consolidation , atelectasis ?	findings compatible with excretory phase , as requested -- no evidence of pneumothorax .
Generated Report	findings : left picc line remains in place . small residual pleural effusion in the left lateral lung base has been removed . no definite infiltratesor effusions minimally . . no definite pleural effusions . a very small right pleural effusion has been placed since last study . impression :resolution of right pleural effusion .	findings : there is still endotracheal tube above the carina . a small right apical pleura leffusion1.interval removal of an nasogastri c tube . ngtube is noted . there are diffuse bilateral perihilar airspace opacities, including diffuse airspace disease and a small right pleural effusion .there areno pleural effusions.there is no pneumothorax . the heart is stable. impression :1. two chest tubes, continued bilateralpneumothorax .2. bilateral pleural effusions and bilateral lower lung atelectasis .	findings include unchanged in appearance of left lung opacities . no evidence of pneumothorax . one is in pleural density in the chest and maybe of uncertain location and not definitely seen .	findings include 1. blunting of costophrenic angles and linear components in the lung bases , predominantly in the chest stable and a smalllinear components consistent with atelectasis . 3 . . no evidence of pneumothorax . 3 . . metal clips indicative of previous left axillary surgery .impression unchanged unchanged since april 3 no evidence of acute pulmonary process

Fig. 19.5 Another four sample image classification predictions (P) along with original and generated reports. Text attentions are highlighted over the generated text. Correct predication is marked in green, false prediction in red, and missing prediction in blue

CNN part and the adjustment of image features from Transition layer will benefit the report generation task.

Figures 19.4 and 19.5 illustrate 8 sample results from the proposed automatic classification and reporting system. Original images are shown along with the classification predications, original reports, and generated reports. Text-attended words are also highlighted over the generated reports. If looking at generated reports alone, we find that they all read well. However, the described diseases may not truly appear in the images. For example, "Atelectasis" is correctly recognized in sample A but "Effusion" is missed. "Effusion" (not too far from the negation word "without") is erroneously highlighted in sample B but the system is still able to correctly classify the image as "No finding". In sample D, the generated report misses "Mass" while it states right about the metastasis in the lung. One promising finding is that the false predictions ("Mass" and "Consolidation") in sample C can actually be observed in the image (verified by a radiologist) but somehow were not noted in the original report, which indicates our proposed network can to some extent associate the image appearance with the text description.

19.6 Summary

Automatically extracting the machine-learnable annotation from the retrospective data remains a challenging task, among which images and reports are two main useful sources. Here, we proposed a novel text-image embedding network integrated with multi-level attention models. TieNet is implemented in an end-to-end CNN-RNN architecture for learning a blend of distinctive image and text representations. Then, we demonstrate and discuss the pros and cons of including radiological reports in both auto-annotation and reporting tasks. While significant improvements have been achieved in multi-label disease classification, there is still much space to improve the quality of generated reports. For future work, we will extend TieNet to include multiple RNNs for learning not only disease words but also their attributes and further correlate them and image findings with the description in the generated reports.

Acknowledgements This work was supported by the Intramural Research Programs of the NIH Clinical Center and National Library of Medicine. Thanks to Adam Harrison and Shazia Dharssi for proofreading the manuscript. We are also grateful to NVIDIA Corporation for the GPU donation.

References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mane D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viegas F, Vinyals O,

Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2016) TensorFlow: large-scale machine learning on heterogeneous distributed systems
2. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: International conference on learning representations (ICLR), pp 1–15
3. Banerjee S, Lavie A (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp 65–72
4. Ben-Cohen A, Diamant I, Klang E, Amitai M, Greenspan H (2016) Fully convolutional network for liver segmentation and lesions detection. In: International workshop on large-scale annotation of biomedical data and expert label synthesis, pp 77–85
5. Chartrand G, Cheng PM, Vorontsov E, Drozdzal M, Turcotte S, Pal CJ, Kadoury S, Tang A (2017) Deep learning: a primer for radiologists. Radiogr Rev 37(7):2113–2131. Radiological Society of North America, Inc
6. Dai B, Zhang Y, Lin D (2017) Detecting visual relationships with deep relational networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 3076–3086
7. Demner-Fushman D, Kohli MD, Rosenman MB, Shooshan SE, Rodriguez L, Antani S, Thoma GR, McDonald CJ (2015) Preparing a collection of radiology examinations for distribution and retrieval. J Am Med Inf Assoc 23(2):304–310
8. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 248–255
9. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118
10. Everingham M, Eslami SMA, Gool LV, Williams CKI, Winn J, Zisserman A (2015) The PASCAL visual object classes challenge: a retrospective. Int J Comput Vis 111(1):98–136
11. Gan Z, Gan C, He X, Pu Y, Tran K, Gao J, Carin L, Deng L (2017) Semantic compositional networks for visual captioning. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 1–13
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
13. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
14. Hu R, Rohrbach M, Andreas J, Darrell T, Saenko K (2017) Modeling relationships in referential expressions with compositional modular networks. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 1115–1124
15. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia, pp 675–678
16. Johnson J, Karpathy A, Fei-Fei L (2016) DenseCap: fully convolutional localization networks for dense captioning. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 4565–4574
17. Karpathy A, Fei-Fei L (2017) Deep visual-semantic alignments for generating image descriptions. IEEE Trans Pattern Anal Mach Intell 39(4):664–676
18. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA, Bernstein MS, Li FF (2016) Visual genome: connecting language and vision using crowdsourced dense image annotations
19. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
20. Lin CY (2004) ROUGE: a package for automatic evaluation of summaries. In: Text summarization branches out: proceedings of the ACL-04 workshop, Barcelona, Spain, vol 8, pp 1–8 (2004)
21. Lin M, Chen Q, Yan S (2014) Network in network. In: International conference on learning representations (ICLR), pp 1–10
22. Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2014) Microsoft COCO: common objects in context. In: European conference on computer vision (ECCV), pp 740–755

23. Lin Z, Feng M, dos Santos CN, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. In: 5th international conference on learning representations (ICLR), pp 1–15
24. Liu J, Wang D, Lu L, Wei Z, Kim L, Turkbey EB, Sahiner B, Petrick N, Summers RM (2017) Detection and diagnosis of colitis on computed tomography using deep convolutional neural networks. Med Phys 44(9):4630–4642
25. Liu Y, Sun C, Lin L, Wang X (2016) Learning natural language inference using bidirectional LSTM model and inner-attention
26. Meng F, Lu Z, Wang M, Li H, Jiang W, Liu Q (2015) Encoding source language with convolutional neural network for machine translation. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (ACL-CoNLL), pp 20–30
27. Nam H, Ha JW, Kim J (2017) Dual attention networks for multimodal reasoning and matching. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 299–307
28. Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics (ACL), pp 311–318
29. Pedersoli M, Lucas T, Schmid C, Verbeek J (2017) Areas of attention for image captioning. In: International conference on computer vision (ICCV), pp 1–22
30. Plummer B, Wang L, Cervantes C, Caicedo J, Hockenmaier J, Lazebnik S (2015) Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: International conference on computer vision (ICCV)
31. Rush AM, Chopra S, Weston J (2015) A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP), pp 379–389
32. Shin HC, Roberts K, Lu L, Demner-Fushman D, Yao J, Summers RM (2016) Learning to read chest X-rays: recurrent neural cascade model for automated image annotation. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 2497–2506
33. Vinyals O, Fortunato M, Jaitly N (2015) Pointer networks. In: Advances in neural information processing systems, pp 2692–2700
34. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 3156–3164
35. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) ChestX-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 2097–2106
36. Wu Q, Wang P, Shen C, Dick A, van den Hengel A (2016) Ask me anything: free-form visual question answering based on knowledge from external sources. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 1–5
37. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning (ICML), pp 2048–2057
38. Yu D, Fu J, Mei T, Rui Y (2017) Multi-level attention networks for visual question answering. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9
39. Yulia WLLCC, Amir TS, Alan RFACD, Trancoso WBI (2015) Not all contexts are created equal: better word representations with variable attention. In: Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP), pp 1367–1372
40. Zhang Z, Chen P, Sapkota M, Yang L (2017) TandemNet: distilling knowledge from medical images using diagnostic reports as optional semantic references. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 320–328
41. Zhang Z, Xie Y, Xing F, McGough M, Yang L (2017) MDNet: a semantically and visually interpretable medical image diagnosis network. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 6428–6436
42. Zhu Y, Groth O, Bernstein M, Fei-Fei L (2016) Visual7W: grounded question answering in images. In: The IEEE conference on computer vision and pattern recognition (CVPR)

Chapter 20
Deep Lesion Graph in the Wild: Relationship Learning and Organization of Significant Radiology Image Findings in a Diverse Large-Scale Lesion Database

Ke Yan, Xiaosong Wang, Le Lu, Ling Zhang, Adam P. Harrison, Mohammadhadi Bagheri and Ronald M. Summers

Abstract Radiologists in their daily work routinely find and annotate significant abnormalities on a large number of radiology images. Such abnormalities, or lesions, have collected over years and stored in hospitals' picture archiving and communication systems. However, they are basically unsorted and lack semantic annotations like type and location. In this paper, we aim to organize and explore them by learning a deep feature representation for each lesion. A large-scale and comprehensive dataset, DeepLesion, is introduced for this task. DeepLesion contains bounding boxes and size measurements of over 32K lesions. To model their similarity relationship, we leverage multiple supervision information including types, self-supervised location

K. Yan (✉) · M. Bagheri · R. M. Summers
Imaging Biomarkers and Computer-Aided Diagnosis Laboratory, National Institutes of Health Clinical Center, 10 Center Drive, Bethesda, MD 20892, USA
e-mail: ke.yan@nih.gov

M. Bagheri
e-mail: mohammad.bagheri@nih.gov

R. M. Summers
e-mail: rms@nih.gov

X. Wang · L. Zhang
Nvidia Corporation, Bethesda, MD 20814, USA
e-mail: xiaosongw@nvidia.com

L. Zhang
e-mail: zhangling0722@163.com

L. Lu · A. P. Harrison
PAII Inc., Bethesda Research Lab, 6720B Rockledge Drive, Ste 410, Bethesda, MD 20817, USA
e-mail: le.lu@paii-labs.com

A. P. Harrison
e-mail: adam.p.harrison@gmail.com

L. Lu
Johns Hopkins University, Baltimore, MD, USA

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_20

coordinates, and sizes. They require little manual annotation effort but describe useful attributes of the lesions. Then, a triplet network is utilized to learn lesion embeddings with a sequential sampling strategy to depict their hierarchical similarity structure. Experiments show promising qualitative and quantitative results on lesion retrieval, clustering, and classification. The learned embeddings can be further employed to build a lesion graph for various clinically useful applications. An algorithm for intra-patient lesion matching is proposed and validated with experiments.

20.1 Introduction

Large-scale datasets with diverse images and dense annotations [10, 13, 23] play an important role in computer vision and image understanding, but often come at the cost of vast amounts of labeling. In computer vision, this cost has spurred efforts to exploit weak labels [6, 20, 53], e.g., the enormous amount of weak labels generated every day on the web. A similar situation exists in the medical imaging domain, except that annotations are even more time-consuming and require extensive clinical training, which precludes approaches like crowdsourcing. Fortunately, like web data in computer vision, a vast, loosely labeled, and largely untapped data source does exist in the form of hospital picture archiving and communication systems (PACS). These archives house patient images and accompanying radiological reports, markings, and measurements performed during clinical duties. However, data is typically unsorted, unorganized, and unusable in standard supervised machine learning approaches. Developing means to fully exploit PACS radiology database becomes a major goal within the field of medical imaging.

This work contributes to this goal of developing an approach to usefully mine, organize, and learn the relationships between lesions found within computed tomography (CT) images in PACS. Lesion detection, characterization, and retrieval are an important task in radiology [12, 22, 24, 48]. The latest methods based on deep learning and convolutional neural networks (CNNs) have achieved significantly better results than conventional handcrafted image features [16, 24]. However, large amounts of training data with high-quality labels are often needed. To address this challenge, we develop a system designed to exploit the routine markings and measurements of significant findings that radiologists frequently perform [11]. These archived measurements are potentially highly useful sources of data for computer-aided medical image analysis systems. However, they are basically unsorted and lack semantic labels, e.g., lung nodule, mediastinal lymph node. As such, they are a challenging source of data to use, requiring sophisticated approaches to be able to leverage them.

We take a feature embedding and similarity graph approach to address this problem [52]. First, we present a new dataset: DeepLesion,[1] which was collected from the PACS of the National Institute of Health Clinical Center. It contains 32,120 axial CT

[1] Available at https://nihcc.box.com/v/DeepLesion.

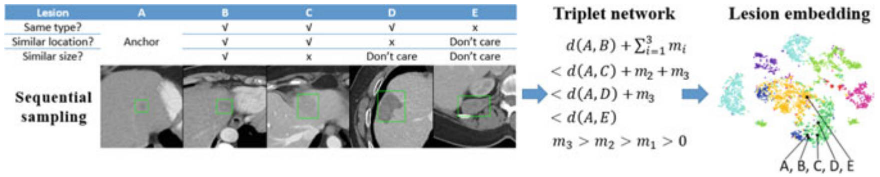


Fig. 20.1 The proposed framework. Using a triplet network, we learn a feature embedding for each lesion in our comprehensive DeepLesion dataset. Training samples A–E are selected with a sequential sampling strategy so as to make the embeddings respects similarity in type, location, and size

slices from 10,594 CT imaging studies of 4,427 unique patients. There are one to three lesions in each image with accompanying bounding boxes and size measurements. The lesions are diverse but unorganized. Our goal is to understand them and discover their relationships. In other words, can we organize them so that we are able to (1) know their type and location; (2) find similar lesions in different patients, i.e., content-based lesion retrieval; and (3) find similar lesions in the same patient, i.e., lesion instance matching for disease tracking?

As Fig. 20.1 illustrates, the above problems can be addressed by learning feature representations for each lesion that keeps a proper similarity relationship, i.e., lesions with similar attributes should have similar embeddings. To reduce annotation workload and leverage the intrinsic structure within CT volumes, we use three weak cues to describe each lesion: type, location, and size. Lesion types are obtained by propagating the labels of a small amount of seed samples to the entire dataset, producing pseudo-labels. The 3D relative body location is obtained from a self-supervised body part regression algorithm. Size is directly obtained by the radiological marking. We then define the similarity relationship between lesions based on a hierarchical combination of the cues. A triplet network with a sequential sampling strategy is utilized to learn the embeddings. We also apply a multi-scale multi-crop architecture to exploit both context and detail of the lesions, as well as an iterative refinement strategy to refine the noisy lesion-type pseudo-labels.

Qualitative and quantitative experimental results demonstrate the efficacy of our framework for several highly important applications. (1), We show excellent performance on content-based lesion retrieval [22, 30, 44, 49]. Effective solutions to this problem can help identify similar case histories, better understand rare disorders, and ultimately improve patient care [24]. We show that our embeddings can be used to find lesions similar in type, location, and size. Most importantly, the embeddings can match lesions with semantically similar body structures that are not specified in the training labels. (2), The embeddings are also successfully applied in intra-patient lesion matching. Patients under therapy typically undergo CT examinations (studies) at intervals to assess the effect of treatments. Comparing lesions in follow-up studies with their corresponding ones in previous studies constitutes a major part of a radiologist's workload [27]. We provide an automated tool for lesion matching which can significantly save time, especially for patients with multiple follow-up studies [33].

20.2 Related Work

Deep Metric Learning: Metric learning can be beneficial whenever we want to keep certain similarity relationship between samples [2]. The Siamese network [3] is a seminal work in deep metric learning, which minimizes the distance between a pair of samples with the same label and pushes samples with different labels apart. It was improved by the triplet network [31], which considers relative distances. The triplet network requires three samples to compute a loss: an anchor A, a positive sample P with the same label as A, and a negative sample N with a different label. The network learns embeddings that respect the following distance relationship:

$$\|f(A) - f(P)\|_2^2 + m < \|f(A) - f(N)\|_2^2, \tag{20.1}$$

where f is the embedding function to be learned and m is a predefined margin. Various improvements to the standard triplet network have been proposed [5, 38–40, 54]. Three key aspects in these methods are how to define similarity between images, how to sample images for comparison, and how to compute the loss function. Zhang et al. [54] generalized the sampling strategy and triplet loss for multiple labels with hierarchical structures or shared attributes. Son et al. [39] employed label hierarchy to learn object embeddings for tracking, where object class is a high-level label and detection timestamp is low level. Our sequential sampling strategy shares the similar spirit with them, but we lack well-defined supervision cues in the dataset, so we proposed strategies to leverage weak cues, e.g., self-supervised body part regressor and iterative refinement.

Lesion Management: Great efforts have been devoted to lesion detection [43, 48, 50], classification [7, 12], segmentation [4, 42], and retrieval [22, 30, 44, 49]. Recently, CNNs have become the method of choice over handcrafted features due to the former's superior performance [16, 24, 35, 41]. Our work is in line with content-based medical image retrieval, which has been surveyed in detail by [22]. Existing methods generally focus on one type of lesion (e.g., lung lesion or mammographic mass) and learn the similarity relationship based on manually annotated labels [44, 49] or radiology reports [30]. To the best of our knowledge, no work has been done on learning deep lesion embeddings on a large comprehensive dataset with weak cues. Taking a different approach, [17, 46] cluster images or lesions to discover concepts in unlabeled large-scale datasets. However, they did not leverage multiple cues to explicitly model the semantic relationship between lesions. Several existing works on intra-patient lesion matching focus on detecting follow-up lesions and matching them pixel by pixel [18, 28, 36, 45], which generally requires organ segmentation or time-consuming nonrigid volumetric registration. Besides, they are designed for certain types of lesions, whereas our lesion embedding can be used to match all kinds of lesions.

20.3 Dataset

The DeepLesion dataset[2] consists of over 32K clinically significant findings mined from a major institute's PACS. To the best of our knowledge, this dataset is the first to automatically extract lesions from challenging PACS sources. Importantly, the workflow described here can be readily scaled up and applied to multiple institutional PACS, providing a means for truly massive scales of data.

Radiologists routinely annotate clinically meaningful findings in medical images using arrows, lines, diameters or segmentations. These images, called "bookmarked images", have been collected over close to two decades in our institute's PACS. Without loss of generality, we study one type of bookmark in CT images: lesion diameters. As part of the RECIST guidelines [11], which is the standard in tracking lesion progression in the clinic, lesion diameters consist of two lines: one measuring the longest diameter and the second measuring its longest perpendicular diameter in the plane of measurement. We extract the lesion diameter coordinates from the PACS server and convert them into corresponding positions on the image plane. After removing some erroneous annotations, we obtain 32,120 axial CT slices (mostly 512 × 512) from 10,594 studies of 4,427 unique patients. There are one to three lesions in each image, adding up to 32,735 lesions altogether. We generate a box tightly around the two diameters and add a 5-pixel padding in each direction to capture the lesion's full spatial extent. Samples of the lesions and bounding boxes are in Fig. 20.2. More introduction of the dataset can be found in [51].

To provide an overview of the DeepLesion dataset, we draw a scatter map to show the distribution of the types and relative body locations of the lesions in Fig. 20.2. From the lesion types and sample images, one can see that the relative body locations of the lesions are consistent with their actual physical positions, proving the validity of the location information used in the paper, particularly the self-supervised body part regressor. Some lesion types like bone and soft tissue have widespread locations. Neighboring types such as lung/mediastinum and abdomen/liver/kidney have large overlap in location due to inter-subject variabilities. Besides, we can clearly see the considerable diversity of DeepLesion.

Figure 20.3 illustrates the approach to obtain the location and size of a lesion. In order to locate a lesion in the body, we first obtain the mask of the body in the axial slice, and then compute the relative position (0–1) of the lesion center to get the x- and y-coordinates. As for z, the self-supervised body part regressor (SSBR) is used.

The 12-bit CT intensity range is rescaled to floating-point numbers in $[0, 255]$ using a single windowing covering the intensity ranges in lungs, soft tissues, and bones. Each image is resized so that the spacing is 1 mm/pixel. For each lesion, we crop a patch with 50 mm padding around its bounding box. To encode 3D information, we use three neighboring slices (interpolated at 2 mm slice intervals) to compose a three-channel image. No data augmentation was used.

[2] https://nihcc.box.com/v/DeepLesion.

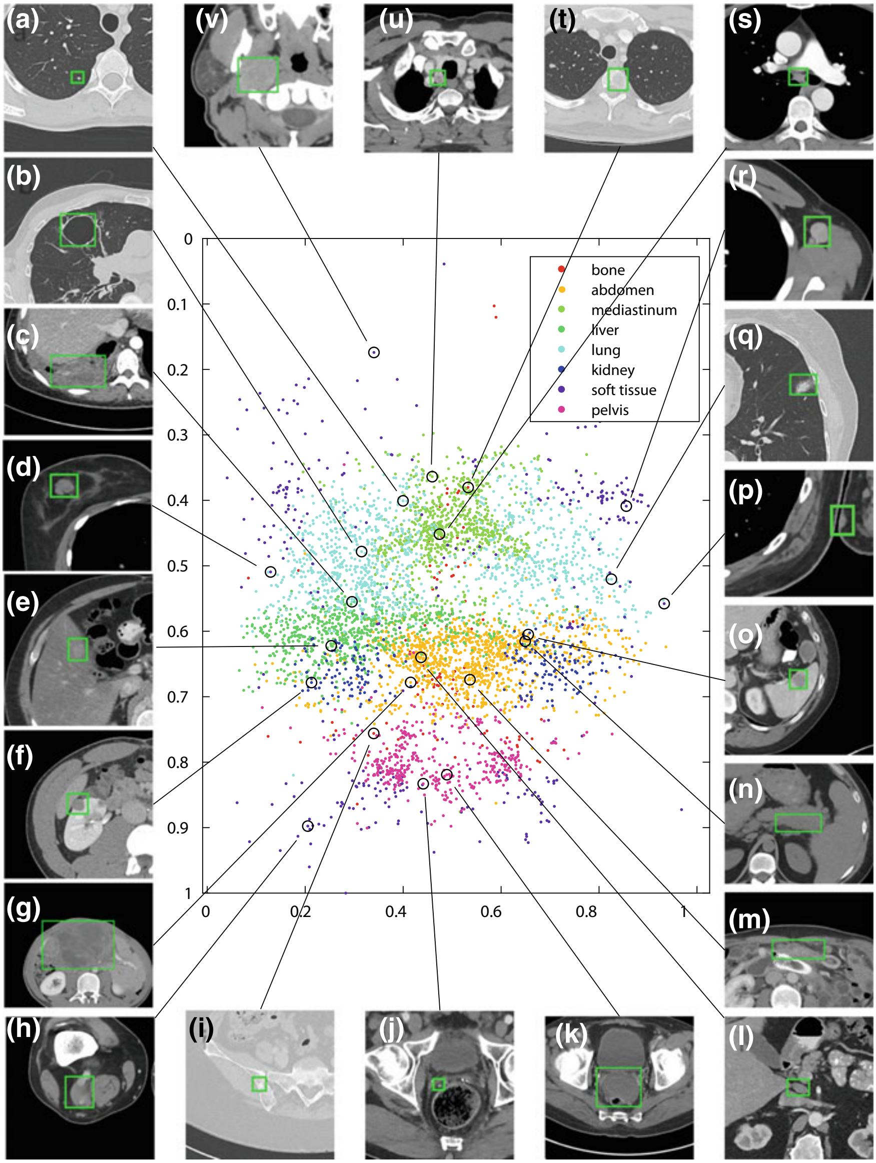
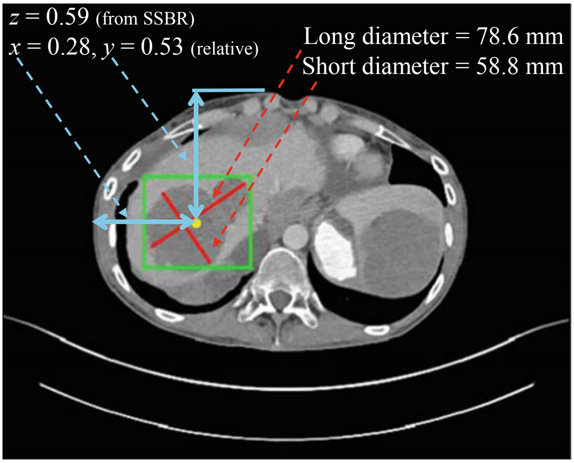


Fig. 20.2 Visualization of the DeepLesion dataset (test set). The x- and y-axes of the scatter map correspond to the x- and z-coordinates of the relative body location of each lesion, respectively. Therefore, this map is similar to a frontal view of the human body. Colors indicate the manually labeled lesion types. Sample lesions are exhibited to show the great diversity of DeepLesion, including **a** lung nodule; **b** lung cyst; **c** costophrenic sulcus (lung) mass/fluid; **d** breast mass; **e** liver lesion; **f** renal mass; **g** large abdominal mass; **h** posterior thigh mass; **i** iliac sclerotic lesion; **j** perirectal lymph node (LN); **k** pelvic mass; **l** periportal LN; **m** omental mass; **n** peripancreatic lesion; **o** splenic lesion; **p** subcutaneous/skin nodule; **q** ground-glass opacity; **r** axillary LN; **s** subcarinal LN; **t** vertebral body metastasis; **u** thyroid nodule; **v** neck mass

Fig. 20.3 Location and size of a sample lesion. The red lines are the long and short diameters annotated by radiologists during their daily work. The green box is the bounding box calculated from the diameters. The yellow dot is the center of the bounding box. The blue lines indicate the relative x- and y-coordinates of the lesion. The z-coordinate is predicted by SSBR. Best viewed in color



20.4 Method

In this section, we describe our framework that extracts supervision cues from DeepLesion, learns lesion embeddings, and finally does lesion retrieval and matching.

20.4.1 Supervision Cues

Supervision information, or cues, are key in defining the similarity relationship between lesions. Because it is prohibitively time-consuming to manually annotate all lesions in a PACS-based dataset like DeepLesion, a different approach must be employed. Here we use the cues of lesion type, relative body location, and size.

Size information (lengths of long and short lesion diameters) has been annotated by radiologists and ranges from 0.2 to 343 mm with a median of 15.6 mm. They are significant indicators of patients' conditions according to the RECIST guideline [11]. For example, larger lymph nodes are considered lesions while those with short diameters $<$ 10 mm are treated as normal [11]. While size can be obtained directly from radiologists' markings, type and relative body location require more complex approaches.

Lesion Type: Among all 32,735 lesions, we randomly select 30% and manually label them into eight types: lung, abdomen, mediastinum, liver, pelvis, soft tissue, kidney, and bone. These are coarse-scale attributes of the lesions. An experienced radiologist verified the labels. The mediastinum class mainly consists of lymph nodes in the chest. Abdomen lesions are miscellaneous ones that are not in liver or kidney. The soft tissue class contains lesions in the muscle, skin, fat, etc. Among the labeled samples, we randomly select 25% as training seeds to predict pseudo-labels, 25% as the validation set, and the other 50% as the test set. There is no patient-level overlap between all subsets.

The type of lesion is related to its location, but the latter information cannot replace the former because some lesion types like bone and soft tissue have widespread locations. Neighboring types such as lung/mediastinum and abdomen/liver/kidney are hard to classify solely by location. The challenge with using PACS data is that there are no annotated class labels for each lesion in DeepLesion. Therefore, we use labeled seed samples to train a classifier and apply it to all unlabeled samples to get their pseudo-labels [21]. Details on the classifier are provided in Sect. 20.4.2.2.

Relative Body Location: Relative body location is an important and clinically relevant cue in lesion characterization. While the x and y coordinates of a lesion are easy to acquire in axial CT slices, the z coordinate (e.g., 0–1 from head to toe) is not as straightforward to find. The slice indices in the volume cannot be used to compute z because CT volumes often have different scan ranges (start, end), not to mention variabilities in body lengths and organ layouts. For this reason, we use the self-supervised body part regressor (SSBR), which provides a relative z coordinate based on context appearance.

SSBR operates on the intuition that volumetric medical images are intrinsically structured, where the position and appearance of organs are relatively aligned. The superior–inferior slice order can be leveraged to learn an appearance-based z. SSBR randomly picks m equidistant slices from a volume, denoted $j, j+k, \dots, j+k(m-1)$, where j and k are randomly determined. They are passed through a CNN to get a score s for each slice, which is optimized using the following loss function:

$$\begin{aligned} L_{\text{SSBR}} &= L_{\text{order}} + L_{\text{dist}}; \\ L_{\text{order}} &= -\sum\nolimits_{i=0}^{m-2} \log h\left(s_{j+k(i+1)} - s_{j+ki}\right); \\ L_{\text{dist}} &= \sum\nolimits_{i=0}^{m-3} g(\Delta_{i+1} - \Delta_i), \\ \Delta_i &= s_{j+k(i+1)} - s_{j+ki}, \end{aligned} \tag{20.2}$$

where h is the sigmoid function, and g is the smooth L1 loss [15]. L_{order} requires slices with larger indices to have larger scores. L_{dist} makes the difference between two slice scores proportional to their physical distance. The order loss and distance loss terms collaborate to push each slice score toward the correct direction relative to other slices. After convergence, slices scores are normalized to [0, 1] to obtain the z coordinates without having to know which score corresponds to which body part. The framework of SSBR is shown in Fig. 20.4.

In DeepLesion, some CT volumes are zoomed in on a portion of the body, e.g., only the left half is shown. To handle them, we train SSBR on random crops of the axial slices. Besides, SSBR does not perform well on body parts that are rare in the training set, e.g., head and legs. Therefore, we train SSBR on all data first to detect hard volumes by examining the correlation coefficient (r) between slice indices and slice scores, where lower r often indicates rare body parts in the volume. Then, SSBR is trained again on a resampled training set with hard volumes oversampled.

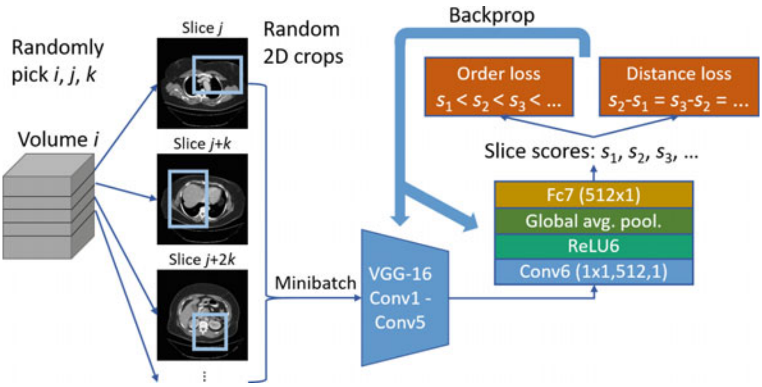


Fig. 20.4 Framework of the self-supervised body part regressor (SSBR)

20.4.2 *Learning Lesion Embeddings*

To learn lesion embeddings, we employ a triplet network with sequential sampling, as illustrated in Fig. 20.1.

20.4.2.1 Sequential Sampling

Similar to [39, 54], we leverage multiple cues to describe the relationship between samples. A naïve strategy would be to treat all cues equally, where similarity can be calculated by, for instance, averaging the similarity of each cue. Another strategy assumes a hierarchical structure exists in the cues. Some high-level cues should be given higher priority. This strategy applies to our task, because intuitively lesions of the same type should be clustered together first. Within each type, we hope lesions that are closer in location to be closer in the feature space. If two lesions are similar in both type and location, they can be further ranked by size. This is a conditional ranking scheme.

To this end, we adopt a sequential sampling strategy to select a sequence of lesions following the hierarchical relationship above. As depicted in Fig. 20.1, an anchor lesion A is randomly chosen first. Then, we look for lesions with similar type, location, and size with A and randomly pick B from the candidates. Likewise, C is a lesion with similar type and location but dissimilar size; D is similar in type but dissimilar in location (its size is not considered); E has a different type (its location and size are not considered). Here, two lesions are similar in type if they have the same pseudo-label; they are similar in location (size) if the Euclidean distance between their location (size) vectors is smaller than a threshold T_{low}, whereas they are dissimilar if the distance is larger than T_{high}. We do not use hard triplet mining as in [29, 31] because of the noisy cues. Figure 20.5 presents some examples of lesion

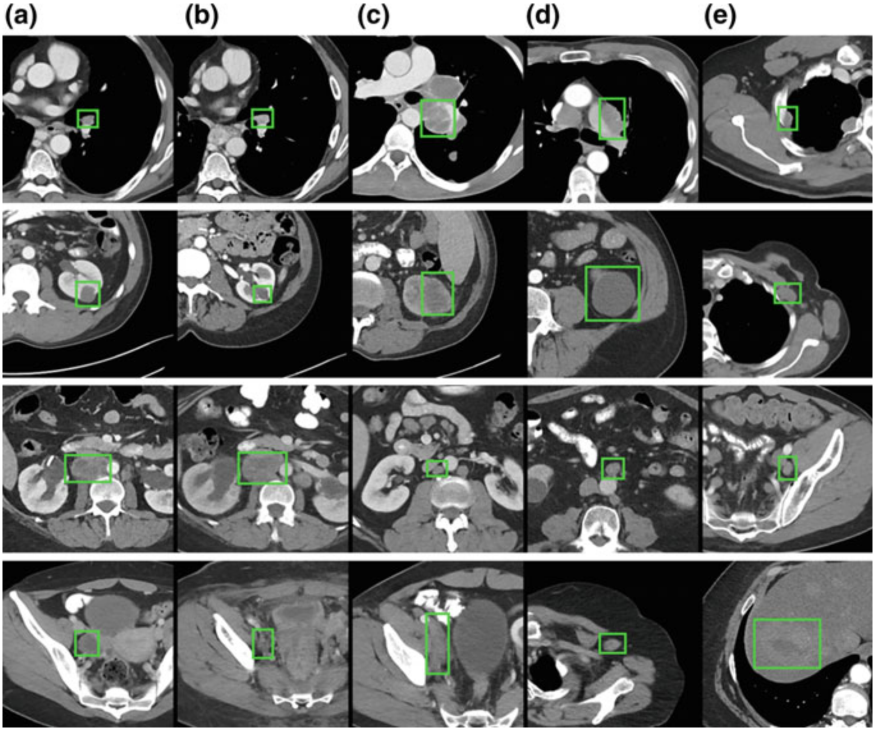


Fig. 20.5 Sample training sequences. Each row is a sequence. Columns 1–5 are examples of lesions *A*–*E* in Fig. 20.1, respectively

sequences. Note that there is label noise in the fourth row, where lesion D does not have the same type with A–C (soft tissue versus pelvis).

A selected sequence can be decomposed into three triplets: ABC, ACD, and ADE. However, they are not equal, because we hope two lesions with dissimilar types to be farther apart than two with dissimilar locations, followed by size. Hence, we apply larger margins to higher level triplets [5, 54]. Our loss function is defined as

$$
\begin{aligned}
L = \frac{1}{2S}\sum_{i=1}^{S}\big[& \max(0, d_{AB}^2 - d_{AC}^2 + m_1) \\
& + \max(0, d_{AC}^2 - d_{AD}^2 + m_2) \\
& + \max(0, d_{AD}^2 - d_{AE}^2 + m_3)\big].
\end{aligned}
\tag{20.3}
$$

$m_3 > m_2 > m_1 > 0$ are the hierarchical margins; S is the number of sequences in each mini-batch; d_{ij} is the Euclidean distance between two samples in the embedding

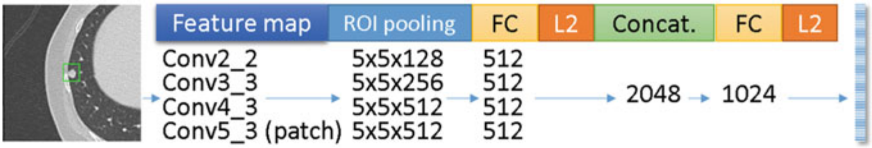


Fig. 20.6 Network architecture of the proposed triplet network

space. The idea in sequential sampling resembles that of SSBR (Eq. 20.2): ranking a series of samples to make them self-organize and move to the right place in the feature space.

20.4.2.2 Network Architecture and Training Strategy

VGG-16 [37] is adopted as the backbone of the triplet network. As illustrated in Fig. 20.6, we input the 50 mm-padded lesion patch, then combine feature maps from four stages of VGG-16 to get a multi-scale feature representation with different padding sizes [14, 19]. Because of the variable sizes of the lesions, region of interest (ROI) pooling layers [15] are used to pool the feature maps to $5 \times 5 \times num\_channel$ separately. For conv2_2, conv3_3, and conv4_3, the ROI is the bounding box of the lesion in the patch to focus on its details. For conv5_3, the ROI is the entire patch to capture the context of the lesion [14, 19]. Each pooled feature map is then passed through a fully connected layer (FC), an L2 normalization layer (L2), and concatenated together. The final embedding is obtained after another round of FC and L2 normalization layers.

To get the initial embedding of each lesion, we use ImageNet [10] pretrained weights to initialize the convolutional layers, modify the output size of the ROI pooling layers to $1 \times 1 \times num\_channel$, and remove the FC layers in Fig. 20.6 to get a 1408D feature vector. We use the labeled seed samples to train an 8-class RBF-kernel support vector machine (SVM) classifier and apply it to the unlabeled training samples to get their pseudo-labels. We also tried semi-supervised classification methods [1, 56] and achieved comparable accuracy. Seed samples were not used to train the triplet network. We then sample sequences according to Sect. 20.4.2.1 and train the triplet network until convergence. With the learned embeddings, we are able to retrain the initial classifier to get cleaner pseudo-labels, and then fine-tune the triplet network with a lower learning rate [46]. In our experiments, this iterative refinement improves performance.

20.4.3 Lesion Retrieval and Matching

The lesion graph can be constructed after the embeddings are learned. It can be used to directly achieve content-based lesion retrieval by finding nearest neighbors of query lesions. However, intra-patient lesion matching requires additional techniques to accomplish.

We assume that lesions in all studies have been detected by other lesion detection algorithms [43] or marked by radiologists, which is the case in DeepLesion. In this section, our goal is to match the same lesion instances and group them for each patient. Potential challenges include appearance changes between studies due to lesion growth/shrinkage, movement of organs or measurement positions, and different contrast phases. Note that for one patient not all lesions occur in each study because the scan ranges vary and radiologists only mark a few target lesions. In addition, one CT study often contains multiple series (volumes) that are scanned at the same time point but differ in image filters, contrast phases, etc. To address these challenges, we design the lesion matching algorithm described in Algorithm 1.

The basic idea is to build an intra-patient lesion graph and remove the edges connecting different lesion instances. The Euclidean distance of lesion embeddings is adopted as the similarity measurement. First, lesion instances from different series within the same study are merged if their distance is smaller than T_1. They are then treated as one lesion with embeddings averaged. Second, we consider lesions in all studies of the same patient. If the distance between two lesions is larger than T_2 ($T_2 > T_1$), they are not similar and their edge is removed. After this step, one lesion in study 1 may still connect to multiple lesions in study 2 if they look similar, so we only keep the edge with the minimum distance and exclude the others. Finally, the remaining edges are used to extract the matched lesion groups.

Algorithm 1 Intra-patient lesion matching

```
Input: Lesions of the same patient represented by their embeddings; the study index $s$ of each
    lesion; intra-study threshold $T_1$; inter-study threshold $T_2$.
Output: Matched lesion groups.
Compute an intra-patient lesion graph $G = \langle V, \mathcal{E} \rangle$, where $V$ are nodes (lesions) and $\mathcal{E}$ are edges.
    Denote $d_{ij}$ as the Euclidean distance between nodes $i, j$.
Merge nodes $i$ and $j$ if $s_i = s_j$ and $d_{ij} < T_1$.
Threshold: $\mathcal{E} \leftarrow \mathcal{E} - \mathcal{D}, \mathcal{D} = \{\langle i, j \rangle \in \mathcal{E} | d_{ij} > T_2\}$.
Exclusion: $\mathcal{E} \leftarrow \mathcal{E} - \mathcal{C}, \mathcal{C} = \{\langle i, j \rangle \mid \langle i, j \rangle \in \mathcal{E}, \langle i, k \rangle \in \mathcal{E}, s_j = s_k$, and $d_{ij} \geq d_{ik}\}$.
Extraction: Each node group with edge connections is considered as a matched lesion group.
```

20.5 Experiments

Our experiments aim to show that the learned lesion embeddings can be used to produce a semantically meaningful similarity graph for content-based lesion retrieval and intra-patient lesion matching.

20.5.1 Implementation Details

We empirically choose the hierarchical margins in Eq. 20.3 to be $m_1 = 0.1, m_2 = 0.2, m_3 = 0.4$. The maximum value of each dimension of the locations and sizes is normalized to 1. When selecting sequences for training, the similarity thresholds for location and size are $T_{\text{low}} = 0.02$, $T_{\text{high}} = 0.1$. We use $S = 24$ sequences per mini-batch. The network is trained using stochastic gradient descent (SGD) with a learning rate of 0.002, which is reduced to 0.0002 in iteration 2K. After convergence (generally in 3K iterations), we do iterative refinement by updating the pseudo-labels and fine-tuning the network with a learning rate of 0.0002. This refinement is performed only once because we find that more iterations only add marginal accuracy improvements. For lesion matching, the intra-study threshold T_1 is 0.1, and we vary the inter-study threshold T_2 to compute the precision–recall curve.

To train SSBR, we randomly pick 800 unlabeled CT volumes of 420 subjects from DeepLesion. Each axial slice in the volumes is resized to 128×128 pixels. No further preprocessing or data augmentation was performed. In each mini-batch, we randomly select 256 slices from 32 volumes (8 equidistant slices in each volume, see Eq. 20.2) for training. The network is trained using stochastic gradient descent with a learning rate of 0.002. It generally converges in 1.5K iterations.

The sample lesions in Fig. 20.2 can be used to qualitatively evaluate the learned slice scores, or z-coordinates. We also conducted a preliminary experiment to quantitatively assess SSBR. A test set including 18,195 slices subsampled from 260 volumes of 140 new subjects is collected. They are manually labeled as one of the three classes: chest (5903 slices), abdomen (6744), or pelvis (5548). The abdomen class starts from the upper border of the liver and ends at the upper border of the ilium. Then, we set two thresholds on the slice scores to classify them to the three classes. The classification accuracy is 95.99%, with all classification errors appearing at transition regions (chest–abdomen, abdomen–pelvis) partially because of their ambiguity. The result proves the effectiveness of SSBR. More importantly, SSBR is trained on unlabeled volumes that are abundant in every hospital's database, thus zero annotation effort is needed.

20.5.2 Content-Based Lesion Retrieval

First, we qualitatively investigate the learned lesion embeddings in Fig. 20.7, which shows the Barnes–Hut t-SNE visualization [25] of the 1024D embedding and some sample lesions. The visualization is applied to our manually labeled test set, where we have lesion-type ground truth. As we can see, there is a clear correlation between data clusters and lesion types. It is interesting to find that some types are split into several clusters. For example, lung lesions are separated to left lung and right lung, and so are kidney lesions. Bone lesions are split into three small clusters, which are found to be mainly chest, abdomen, and pelvis ones, respectively. Abdomen, liver,

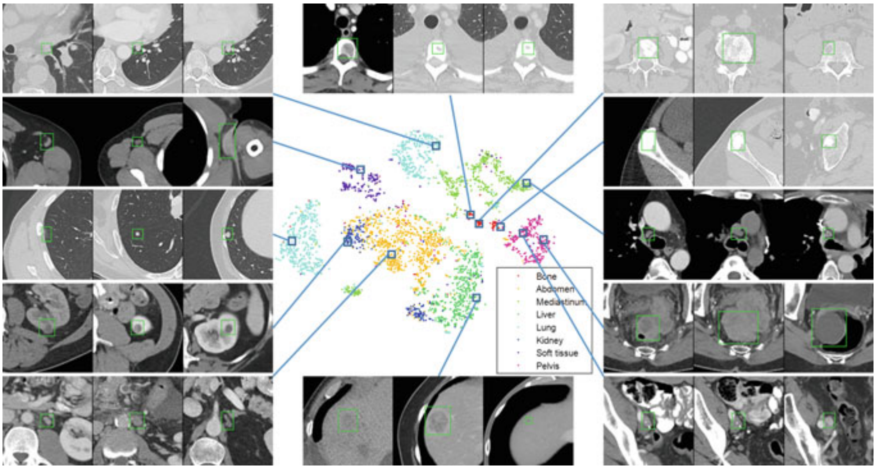


Fig. 20.7 t-SNE visualization of the lesion embeddings on the test set (4,927 samples) of DeepLesion. Colors indicate the manually labeled lesion types. We also split the samples to 128 clusters using K-means and show three random lesions in 12 representative clusters. We did not choose to show closest samples because they are very similar. Best viewed in color

and kidney lesions are close both in real-world location and in the feature space. These observations demonstrate the embeddings are organized by both type and location. The sample lesions in Fig. 20.7 are roughly similar in type, location, and size. Two exceptions are the location variability of the example lesions in the pelvis and soft tissue, which is not surprising considering the large intra-class variance of these two groups and that these examples are randomly chosen.

Figure 20.8 displays several retrieval results using the lesion embeddings. They are ranked by their Euclidean distance with the query one. We find that the top results are mostly the same lesion instances of the same patient, as shown in the first row of Fig. 20.8. It suggests the potential of the proposed embedding on lesion matching, which will be further evaluated in the following section. To better exhibit the ability of the embedding in finding semantically similar lesions, rows 2–4 of Fig. 20.8 depict retrieved lesions from different patients. Spiculated nodules in the right lung and left para-aortic lymph nodes are retrieved in rows 2 and 3, respectively. Row 4 depicts lesions located on the tail of the pancreas, and also some failure cases marked in red. Note that our type labels used in supervision are too coarse to describe either abdomen lymph nodes or pancreas lesions (both are covered in the abdomen class). However, the framework naturally clusters lesions from the same body structures together due to similarity in type, location, size, and appearance, thus discovering these subtypes. Although appearance is not used as supervision information, it is intrinsically considered by the CNN-based feature extraction architecture and strengthened by the multi-scale strategy. To explicitly distinguish sub-types and enhance the semantic information in the embeddings, we can either enrich the type labels by mining knowledge from radiology reports [9, 34, 47, 55], or integrate training samples from

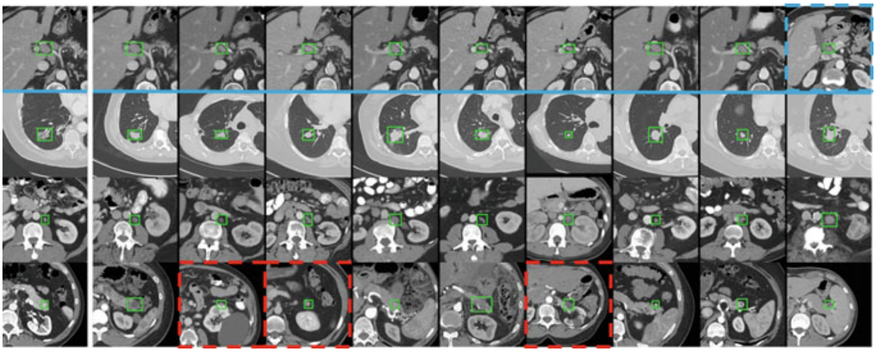


Fig. 20.8 Examples of query lesions (first column) and the top-9 retrieved lesions on the test set of DeepLesion. In the first row, the blue dashed box marks the lesion from a different patient than the query one, whereas the other nine are all from the same patient. In rows 2–4, we constrain that the query and all retrieved lesions must come from different patients. Red dashed boxes indicate incorrect results, see text

other medical image datasets with more specialized annotations [8, 32]. These new labels may be incomplete or noisy, which fits the setting of our system.

Quantitative experimental results on lesion retrieval, clustering, and classification are listed in Table 20.1. For retrieval, the three supervision cues are thoroughly inspected. Because location and size (all normalized to 0–1) are continuous labels, we define an evaluation criterion called average retrieval error (ARE):

$$\mathrm{ARE} = \frac{1}{K}\sum_{i=1}^{K}\|\mathbf{t}^Q - \mathbf{t}_i^R\|_2, \tag{20.4}$$

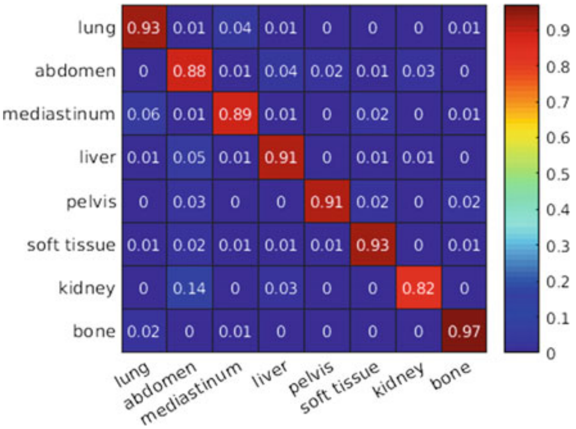
where $\mathbf{t}^Q$ is the location or size of the query lesion and $\mathbf{t}_i^R$ is that of the ith retrieved lesion among the top-K. On the other hand, the ARE of lesion type is simply $1 - precision$. Clustering and classification accuracy are evaluated only on lesion type. Purity and normalized mutual information (NMI) of clustering are defined in [26]. The multi-scale ImageNet feature is computed by replacing the 5 × 5 ROI pooling to 1 × 1 and removing the FC layers.

In Table 20.1, the middle part compares the results of applying different supervision information to train the triplet network. Importantly, when location and size are added as supervision cues, our network performs best on lesion-type retrieval—even better than when only lesion type is used as the cue. This indicates that location and size provide important supplementary information in learning similarity embeddings, possibly making the embeddings more organized and acting as regularizers. The bottom part of the table shows the results of ablation studies, which demonstrate the effectiveness of multi-scale features and iterative refinement, highlighting the importance of combining visual features from different context levels. When only coarse-scale features (conv5, conv4) are used, location ARE is slightly better

Table 20.1 Evaluation results on the test set (4,927 samples) of DeepLesion. For retrieval, we compute the average retrieval error (%) in type, location, and size of the top-5 retrieved lesions compared to the query one. For clustering, lesions are clustered to eight groups using K-means to calculate the purity and NMI (%). For classification, we train an eight-way softmax classifier on the seed labeled samples and apply it on the test set. The CNN in each method was trained five times using different random seeds. Mean results and standard deviations are reported

Feature representation	Average retrieval error			Clustering		Classification
	Type	Location	Size	Purity	NMI	Accuracy
Baseline: Multi-scale ImageNet feature	15.2	9.6	6.9	58.7	35.8	86.2
Baseline: Location feature	22.4	**2.5**	8.8	51.6	32.6	59.7
Triplet with type	8.8 ± 0.2	10.8 ± 0.2	5.7 ± 0.1	84.7 ± 2.8	71.5 ± 1.7	89.5 ± 0.3
Triplet with location	13.0 ± 0.1	6.5 ± 0.1	6.2 ± 0.1	61.1 ± 4.4	39.5 ± 4.3	87.8 ± 0.5
Triplet with type + location	8.7 ± 0.2	7.2 ± 0.1	6.0 ± 0.1	81.3 ± 4.7	68.0 ± 2.4	89.9 ± 0.3
Triplet with type + location + size	**8.5 ± 0.1**	7.2 ± 0.0	**5.1 ± 0.0**	**86.0 ± 3.9**	**72.4 ± 4.6**	**90.5 ± 0.2**
w/o Multi-scale feature: conv5	11.5 ± 0.2	7.1 ± 0.1	6.3 ± 0.0	79.8 ± 0.6	64.8 ± 1.2	86.6 ± 0.4
w/Multi-scale feature: conv5 + conv4	9.7 ± 0.2	7.0 ± 0.0	5.4 ± 0.1	82.4 ± 3.3	67.9 ± 2.2	89.0 ± 0.6
w/o Iterative refinement	8.7 ± 0.2	7.3 ± 0.0	5.2 ± 0.1	85.4 ± 2.8	69.8 ± 2.0	90.2 ± 0.2

because location mainly relies on high-level context information. However, fusing fine-level features (conv3, conv2) significantly improves type and size prediction, which indicates that details are important in these aspects. We also inspected the confusion matrix for lesion classification (Fig. 20.9). The most confusing types are mediastinum/lung lesions, and abdomen/liver/kidney lesions, since some of them are similar in both appearance and location.



	lung	abdomen	mediastinum	liver	pelvis	soft tissue	kidney	bone
lung	0.93	0.01	0.04	0.01	0	0	0	0.01
abdomen	0	0.88	0.01	0.04	0.02	0.01	0.03	0
mediastinum	0.06	0.01	0.89	0.01	0	0.02	0	0.01
liver	0.01	0.05	0.01	0.91	0	0.01	0.01	0
pelvis	0	0.03	0	0	0.91	0.02	0	0.02
soft tissue	0.01	0.02	0.01	0.01	0.01	0.93	0	0.01
kidney	0	0.14	0	0.03	0	0	0.82	0
bone	0.02	0	0.01	0	0	0	0	0.97

Fig. 20.9 The confusion matrix of lesion classification

20.5.3 *Intra-patient Lesion Matching*

We manually grouped 1313 lesions from 103 patients in DeepLesion to 593 groups for evaluation. Each group contains instances of the same lesion across time. There are 1–11 lesions per group. Precision and recall are defined according to [26], where a true positive decision assigns two lesions of the same instance to the same group, and a false positive decision assigns two lesions of different instances to the same group, etc. As presented in Fig. 20.10, our proposed embedding achieves the highest area under the curve (AUC). The location feature does not perform well because different lesion instances may be close to each other in location. This problem can be mitigated by combining location with appearance and using multi-scale features (accomplished in our triplet network). Our algorithm does not require any annotation

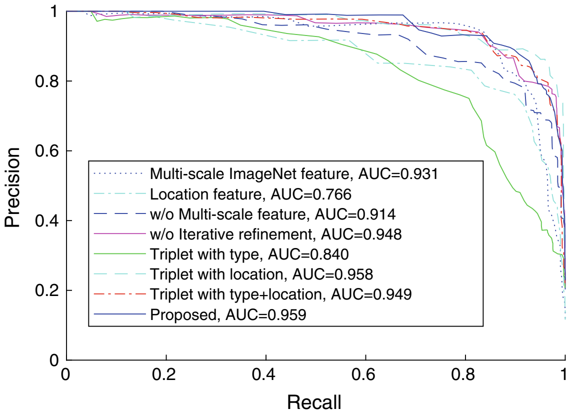


Fig. 20.10 Precision–recall curves of various methods on the intra-patient lesion matching task using DeepLesion. The area under curve (AUC) values are shown in the legends

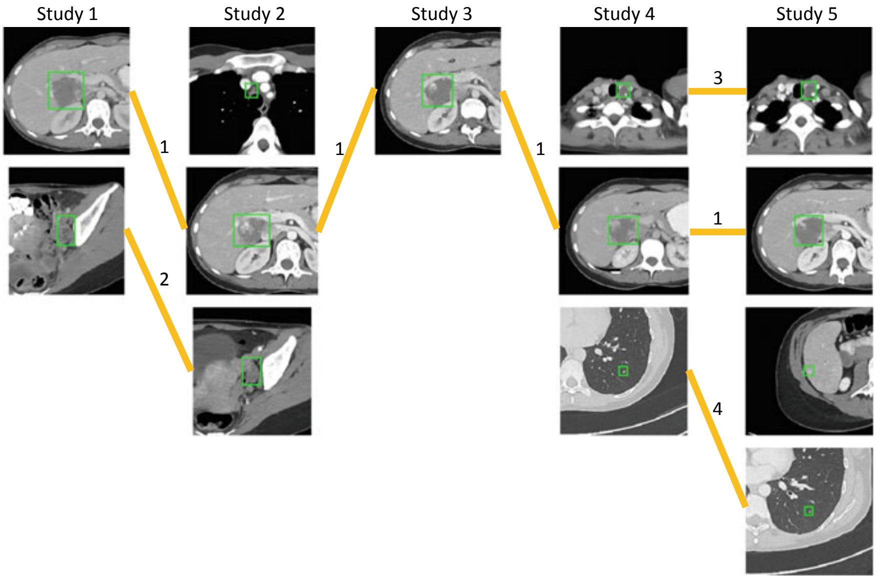


Fig. 20.11 All lesions of a sample patient in DeepLesion. Lesions in each study (CT examination) are listed in a column. Not all lesions occur in each study, because the scan ranges of each study vary and radiologists only mark a few target lesions. We group the same lesion instances to sequences. Four sequences are found and marked in the figure, where the numbers on the connections represent the lesion IDs

of matched lesions for training. It is appearance-based and needs no registration or organ mask, thus is fast.

To provide an intuitive illustration of the lesion matching task, we show lesions of a sample patient in Fig. 20.11, with their lesion graph in Fig. 20.12 and the final extracted lesion sequences in Fig. 20.13. We show that the lesion graph and Algorithm 1 in the paper can be used to accurately match lesions in multiple studies.

20.6 Conclusion and Future Work

In this paper, we present a large-scale and comprehensive dataset, DeepLesion, which contains significant radiology image findings mined from PACS. Lesion embeddings are learned with a triplet network to model their similarity relationship in type, location, and size. The only manual efforts needed are the class labels of some seed images. Promising results are obtained in content-based lesion retrieval and intra-patient lesion matching. The framework can be used as a generic lesion search engine, classifier, and matching tool. After being classified or retrieved by our system, lesions can be further processed by other specialist systems trained on data of a certain type.

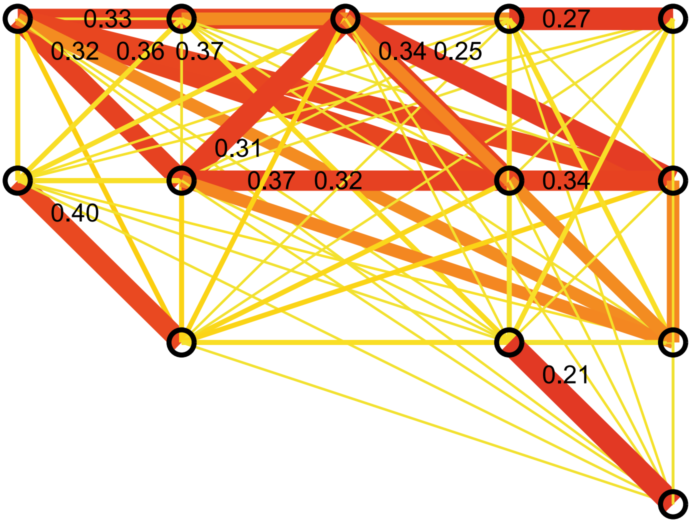


Fig. 20.12 The intra-patient lesion graph of the patient in Fig. 20.11. For clarity, the lesions in Fig. 20.11 are replaced by nodes in this figure. The numbers on the edges are the Euclidean distances between nodes. We only show small distances in the figure. Red, thick edges indicate smaller distances. Note that some edges may overlap with other edges or nodes

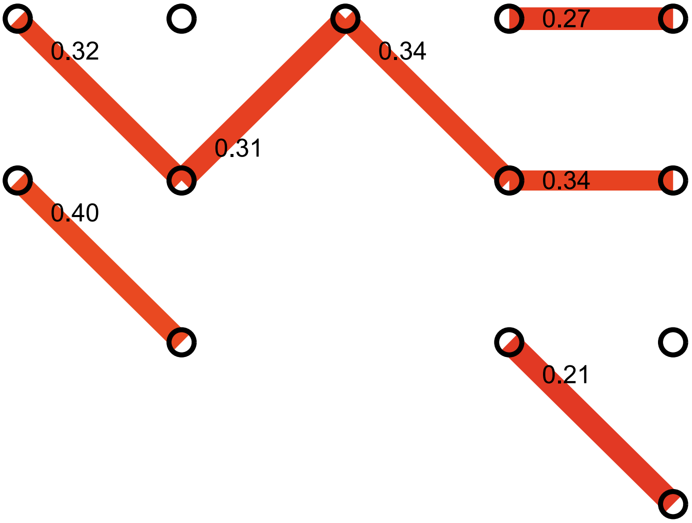


Fig. 20.13 The final lesion sequences found by processing the lesion graph in Fig. 20.12 using Algorithm 1. They are the same as the ground truth in Fig. 20.11

In the future, we plan to incorporate more fine-grained semantic information (e.g., from radiology reports, other specialized datasets, or active learning). For example, radiology reports contain rich semantic information about the lesions, such as their type, location, and other attributes. By mining these information, we can train a more fine-grained, accurate, and clinically meaningful lesion retrieval and classification algorithm, meanwhile using minimal manual annotation effort.

Acknowledgements This research was supported by the Intramural Research Program of the NIH Clinical Center. We thank NVIDIA for the donation of GPU cards.

References

1. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
2. Bellet A, Habrard A, Sebban M (2013) A survey on metric learning for feature vectors and structured data. arXiv:1306.6709
3. Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R (1994) Signature verification using a "Siamese" time delay neural network. In: Advances in neural information processing systems, pp 737–744
4. Cai J, Tang Y, Lu L, Harrison AP, Yan K, Xiao J, Yang L, Summers RM (2018) Accurate weakly-supervised deep lesion segmentation using large-scale clinical annotations: slice-propagated 3D mask generation from 2D RECIST. In: MICCAI. http://arxiv.org/abs/1801.08614
5. Chen W, Chen X, Zhang J, Huang K (2017) Beyond triplet loss: a deep quadruplet network for person re-identification. In: CVPR. https://arxiv.org/abs/1704.01719
6. Chen X, Gupta A (2015) Webly supervised learning of convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 1431–1439
7. Cheng JZ, Ni D, Chou YH, Qin J, Tiu CM, Chang YC, Huang CS, Shen D, Chen CM (2016) Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans. Sci Rep 6(1):24454. https://doi.org/10.1038/srep24454, http://www.nature.com/articles/srep24454
8. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F (2013) The cancer imaging archive (TCIA): maintaining and operating a public information repository. J Digit Imaging 26(6):1045–1057. https://doi.org/10.1007/s10278-013-9622-7
9. Cornegruta S, Bakewell R, Withey S, Montana G (2016) Modelling radiological language with bidirectional long short-term memory networks. arXiv:1609.08409
10. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
11. Eisenhauer E, Therasse P, Bogaerts J, Schwartz LH, Sargent D, Ford R, Dancey J, Arbuck S, Gwyther S, Mooney M, Others (2009) New response evaluation criteria in solid tumours: revised RECIST guideline (version 1.1). Eur J Cancer 45(2):228–247
12. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118
13. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
14. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware CNN model. In: Proceedings of the IEEE international conference on computer vision, pp 1134–1142

15. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
16. Greenspan H, van Ginneken B, Summers RM (2016) Guest Editorial deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans Med Imaging 35(5):1153–1159. https://doi.org/10.1109/TMI.2016.2553401, http://ieeexplore.ieee.org/document/7463094/
17. Hofmanninger J, Krenn M, Holzer M, Schlegl T, Prosch H, Langs G (2016) Unsupervised identification of clinically relevant clusters in routine imaging data. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 192–200
18. Hong H, Lee J, Yim Y (2008) Automatic lung nodule matching on sequential CT images. Comput Biol Med 38(5):623–634. https://doi.org/10.1016/j.compbiomed.2008.02.010, http://www.sciencedirect.com/science/article/pii/S0010482508000449
19. Hu P, Ramanan D (2017) Finding tiny faces. In: CVPR. https://doi.org/10.1109/CVPR.2017.166, http://arxiv.org/abs/1612.04402
20. Krause J, Sapp B, Howard A, Zhou H, Toshev A, Duerig T, Philbin J, Fei-Fei L (2016) The unreasonable effectiveness of noisy data for fine-grained recognition. In: European conference on computer vision. Springer, Berlin, pp 301–320
21. Lee DH (2013) Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML, vol 3, p 2
22. Li Z, Zhang X, Müller H, Zhang S (2018) Large-scale retrieval for medical image analytics: a comprehensive review. Med Image Anal 43:66–84. https://doi.org/10.1016/j.media.2017.09.007, http://www.sciencedirect.com/science/article/pii/S136184151730138X
23. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: European conference on computer vision. Springer, Berlin, pp 740–755
24. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JA, van Ginneken B, Sánchez CI (2017) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88. https://doi.org/10.1016/j.media.2017.07.005, http://www.sciencedirect.com/science/article/pii/S1361841517301135
25. van der Maaten L (2014) Accelerating t-SNE using tree-based algorithms. J Mach Learn Res 15:3221–3245. http://jmlr.org/papers/v15/vandermaaten14a.html
26. Manning CD, Raghavan P, Schutze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge. https://nlp.stanford.edu/IR-book/
27. Moltz JH, D'Anastasi M, Kießling A, Dos Santos DP, Schülke C, Peitgen HO (2012) Workflow-centred evaluation of an automatic lesion tracking software for chemotherapy monitoring by CT. Eur Radiol 22(12):2759–2767
28. Moltz JH, Schwier M, Peitgen HO (2009) A general framework for automatic detection of matching lesions in follow-up CT. In: IEEE international symposium on biomedical imaging: from nano to macro, 2009, ISBI'09. IEEE, pp 843–846
29. Oh Song H, Xiang Y, Jegelka S, Savarese S (2016) Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4004–4012
30. Ramos J, Kockelkorn TTJP, Ramos I, Ramos R, Grutters J, Viergever MA, van Ginneken B, Campilho A (2016) Content-based image retrieval by metric learning from radiology reports: application to interstitial lung diseases. IEEE J Biomed Health Inform 20(1):281–292. https://doi.org/10.1109/JBHI.2014.2375491, http://www.ncbi.nlm.nih.gov/pubmed/25438332, http://ieeexplore.ieee.org/document/6966720/
31. Schroff F, Kalenichenko D, Philbin J (2015) FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
32. Setio AAA, Traverso A, De Bel T, Berens MS, van den Bogaard C, Cerello P, Chen H, Dou Q, Fantacci ME, Geurts B et al (2017) Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. Med Image Anal 42:1–13

33. Sevenster M, Travis AR, Ganesh RK, Liu P, Kose U, Peters J, Chang PJ (2015) Improved efficiency in clinical workflow of reporting measured oncology lesions via pacs-integrated lesion tracking tool. Am J Roentgenol 204(3):576–583
34. Shin HC, Lu L, Kim L, Seff A, Yao J, Summers R (2016) Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation. J Mach Learn Res 17(1–31):2
35. Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers RM (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 35(5):1285–1298. https://doi.org/10.1109/TMI.2016.2528162, http://www.ncbi.nlm.nih.gov/pubmed/26886976, http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4890616, http://ieeexplore.ieee.org/document/7404017/
36. Silva JS, Cancela J, Teixeira L (2011) Fast volumetric registration method for tumor follow-up in pulmonary CT exams. J Appl Clin Med Phys 12(2):362–375
37. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: ICLR 2015
38. Sohn K (2016) Improved deep metric learning with multi-class N-pair loss objective. In: Neural information processing systems, pp 1–9
39. Son J, Baek M, Cho M, Han B (2017) Multi-object tracking with quadruplet convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5620–5629
40. Song HO, Jegelka S, Rathod V, Murphy K (2017) Deep metric learning via facility location. In: IEEE CVPR
41. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang J (2016) Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans Med Imaging 35(5):1299–1312. https://doi.org/10.1109/TMI.2016.2535302, http://ieeexplore.ieee.org/document/7426826/
42. Tang Y, Harrison AP, Bagheri M, Xiao J, Summers RM (2018) Semi-automatic RECIST labeling on CT scans with cascaded convolutional neural networks. In: MICCAI. http://arxiv.org/abs/1806.09507
43. Teramoto A, Fujita H, Yamamuro O, Tamaki T (2016) Automated detection of pulmonary nodules in PET/CT images: ensemble false-positive reduction using a convolutional neural network technique. Med Phys 43(6):2821–2827
44. Tsochatzidis L, Zagoris K, Arikidis N, Karahaliou A, Costaridou L, Pratikakis I (2017) Computer-aided diagnosis of mammographic masses based on a supervised content-based image retrieval approach. Pattern Recognit
45. Vivanti R (2015) Automatic liver tumor segmentation in follow-up ct studies using convolutional neural networks. In: Proceedings of the patch-based methods in medical image processing workshop
46. Wang X, Lu L, Shin HC, Kim L, Bagheri M, Nogues I, Yao J, Summers RM (2017) Unsupervised joint mining of deep features and image labels for large-scale radiology image categorization and scene recognition. In: 2017 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 998–1007
47. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) ChestX-ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: CVPR. https://doi.org/10.1109/CVPR.2017.369, http://arxiv.org/abs/1705.02315
48. Wang Z, Yin Y, Shi J, Fang W, Li H, Wang X (2017) Zoom-in-Net: deep mining lesions for diabetic retinopathy detection. Springer International Publishing, Berlin, pp 267–275
49. Wei G, Ma H, Qian W, Qiu M (2016) Similarity measurement of lung masses for medical image retrieval using kernel based semisupervised distance metric. Med Phys 43(12):6259–6269. https://doi.org/10.1118/1.4966030, http://www.ncbi.nlm.nih.gov/pubmed/27908158
50. Yan K, Bagheri M, Summers RM (2018) 3D context enhanced region-based convolutional neural network for end-to-end lesion detection. In: MICCAI, pp 511–519

51. Yan K, Wang X, Lu L, Summers RM (2018) DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. J Med Imaging 5. https://doi.org/10.1117/1.JMI.5.3.036501, https://www.spiedigitallibrary.org/journals/journal-of-medical-imaging/volume-5/issue-03/036501/DeepLesion-automated-mining-of-large-scale-lesion-annotations-and/10.1117/1.JMI.5.3.036501.full
52. Yan K, Wang X, Lu L, Zhang L, Harrison A, Bagheri M, Summers R (2018) Deep lesion graphs in the wild: relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In: CVPR. http://arxiv.org/abs/1711.10535
53. Zhang H, Shang X, Yang W, Xu H, Luan H, Chua TS (2016) Online collaborative learning for open-vocabulary visual classifiers. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2809–2817
54. Zhang X, Zhou F, Lin Y, Zhang S (2016) Embedding label structures for fine-grained feature representation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1114–1123
55. Zhang Z, Xie Y, Xing F, McGough M, Yang L (2017) MDNet: a semantically and visually interpretable medical image diagnosis network. In: CVPR. https://doi.org/10.1109/CVPR.2017.378, http://arxiv.org/abs/1707.02485
56. Zhu X, Ghahramani Z (2002) Learning from labeled and unlabeled data with label propagation. Technical report CMU-CALD-02-107, Carnegie Mellon University

Chapter 21
Simultaneous Super-Resolution and Cross-Modality Synthesis in Magnetic Resonance Imaging

Yawen Huang, Ling Shao and Alejandro F. Frangi

Abstract Multi-modality magnetic resonance imaging (MRI) has enabled significant progress to both clinical diagnosis and medical research. Applications range from differential diagnosis to novel insights into disease mechanisms and phenotypes. However, there exist many practical scenarios where acquiring high-quality multi-modality MRI is restricted, for instance, owing to limited scanning time. This imposes constraints on multi-modality MRI processing tools, e.g., segmentation and registration. Such limitations are not only recurrent in prospective data acquisition but also when dealing with existing databases with either missing or low-quality imaging data. In this work, we explore the problem of synthesizing high-resolution images corresponding to one MRI modality from a low-resolution image of another MRI modality of the same subject. This is achieved by introducing the cross-modality dictionary learning scheme and a patch-based globally redundant model based on sparse representations. We use high-frequency multi-modality image features to train dictionary pairs, which are robust, compact, and correlated in this multimodal feature space. A feature clustering step is integrated into the reconstruction framework speeding up the search involved in the reconstruction process. Images are partitioned into a set of overlapping patches to maintain the consistency between neighboring pixels and increase speed further. Extensive experimental validations on two multi-modality databases of real brain MR images show that the proposed method outperforms state-of-the-art algorithms in two challenging tasks: image super-resolution and simultaneous SR and cross-modality synthesis. Our method was assessed on both healthy subjects and patients suffering from schizophrenia with excellent results.

Y. Huang (✉)
Malong Technologies, Shenzhen, China
e-mail: yawen.huang@malong.com

Shenzhen Malong Artificial Intelligence Research Center, Shenzhen, China

L. Shao
Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates
e-mail: ling.shao@ieee.org

A. F. Frangi
University of Leeds, Leeds, UK
e-mail: A.Frangi@leeds.ac.uk

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_21

21.1 Introduction

Magnetic Resonance Imaging (MRI) has advanced both clinical diagnosis and biomedical research in the neurosciences. MRI has widely been used given its non-invasiveness and the versatility associated with multi-modality imaging protocols that unravel both brain structure and function. Each MRI sequence (hereafter called an MRI modality) is based upon different image contrast mechanisms that relate to complementary properties of brain tissue structure and function and help to unravel anatomical differences and physiologic alterations of brain tissue in health and disease [1]. For instance, T1-weighted (T1-w) images highlight the differences of longitudinal relaxation time in brain tissue, while T2-weighted (T2-w) images reflect transverse relaxation time, and proton density-weighted (PD-w) images depend on the levels of hydrogen protons within the volume of interest. Tissues with high fat content (e.g., white matter) appear bright and compartments filled with water (e.g., cerebral spinal fluid or CSF) appear dark in T1-w MRI. This is particularly good to depict anatomy. Conversely, in T2-w MRI, compartments filled with water (e.g., CSF) appear bright and those with high fat tissue content (e.g., white matter) appear dark. This is useful for depicting pathology as several lesions (e.g., edema, tumor, infarction, inflammation, and infections) are associated with increased water content. Several other contrast mechanisms associated to water diffusion, tissue perfusion, etc.

Despite these benefits, acquiring a full battery of MRI modalities faces constraints associated with increased scanning costs, limited availability of scanning time, and patient comfort, among others. Also, as MRI technologies improve, enhanced resolution or new contrast mechanisms can be utilized. However, in longitudinal imaging cohorts, its benefits will not be available retrospectively for earlier time points in the study, imposing a natural limitation on the dataset. This brings an additional complexity to image analysis and interpretation as the imaging protocol can change. Finally, many reasons can lead to incomplete records for a subject who took part in a large imaging study owing to imaging artifacts, acquisition errors, and lost or corrupted datasets. In all such scenarios, it would be desirable to have a mechanism to synthesize the high-resolution missing data in a different modality with the available MRI modality. However, most of the existing methods tackling this problem either focuses on image super-resolution (SR) or cross-modality synthesis, but not on solving both problems jointly.

Image SR aims to reconstruct a high-resolution (HR) image from a low-resolution (LR) counterpart. It is an underdetermined inverse problem since a multiplicity of solutions exist for the LR input. To solve such a problem, solution space is often constrained by involving strong prior information. In the early years of studies, some simple interpolation-based smooth methods [2–5] were proposed to zoom up LR images. However, Van Ouwerkerk [6] pointed out that such interpolation methods cannot recover detailed information lost in the down-sampling procedure, and even may blur sharp edges. SR techniques were then proposed [7–13], which take the degradation model (e.g., blurring, noise, and down-sampling effects) into account, to reconstruct the image with much higher accuracy. Such methods estimate the HR image by learning co-occurrence priors between the LR and HR image pairs [10]. For instance, Freeman et al. [14] presented a learning-based approach to estimate an HR image from an LR input via Markov network and Bayesian belief propagation. Although the resolution can generally be improved effectively, corners, edges, and ridges are still blurred. Based on such a strategy, Sun et al. [15] addressed the above problem by a computationally intensive process of analysis of millions of LR-HR patch pairs. Neighbor embedding was then proposed for single-image SR [12]. This consists of projecting the local geometry from the LR feature space onto the HR feature space to estimate the HR embedding. Although a small dataset was used in the training process (partly to solve the massive computational load), results were confined to the small number of neighbors. To adequately recover the general image structure and its details, nonlocal means (NL means) [16, 17] was presented to reconstruct the HR image with noise suppression exploiting image self-similarities. However, for strong denoising levels, images are visually over-smooth. Recently, sparse representations were exploited for solving the SR problem. For example, Yang et al. [10] adopted a joint dictionary learning framework for mapping LR and HR image pairs into a common representation space. Rueda et al. [13] took advantage of this model and applied it to address the SR problem in brain MRI. A common drawback shared by both methods is that they only consider local image information in the image synthesis leading to suboptimal reconstructions.

In parallel to the SR technique, researchers have been developing methods to solve the problem of cross-modality image synthesis [18, 18–20]. This problem can be tackled either by transforming MRI intensities across modalities or by synthesizing tissue contrast in the target domain based on patches of the source domain. Histogram matching is a simple way of transforming image intensities from one modality into another or to normalize histogram ranges across subjects [21–25]. Applications such as segmentation and registration can benefit from histogram normalization and/or transformation to reduce the dependency of the results to intensity variations across individuals or imaging protocols. Although this method is widely used in neuroimaging (e.g., [21–25]), it has demonstrated its weakness for converting data with inconsistent intensities and apparent errors [26]. An alternative approach to reconstruct a target MRI modality from a source MRI modality (or more generally, from any other imaging modality) is the example-based image synthesis [26]. In this approach, two dictionaries are independently trained on corresponding patches from registered image pairs of the source and target modalities, respectively. Then the target image is synthesized from the source data based on a reconstruction algorithm that links the patches to reconstruct the source image to the corresponding patches in the target dictionary. Such approaches have also been applied with very promising results to the related problems of label fusion [27] and image hallucination [11]. The procedure to reconstruct the target image imposes that the same code that optimally reconstructs the source patches from the source dictionary must be applied directly to reconstruct the target patches from the target dictionary based on a mapping learned from a set of image pairs. To do so, the most common procedure is to train two dictionaries via random sampling of the registered image patches from two domains and build the correspondence between patches of two modalities. Such methods concatenate both domains according to the intensities of the paired patches, leading to two separate dictionary learning processes in their respective modalities. In this context, the joint representation of the two domains (juxtaposing the two independently computed codes) is suboptimal regarding a jointly learned code that exploits the cross-modality correlations. In addition, example-based methods rely on the given cross-modality exemplar pairs and do not capture the rich variability in image texture across a population of subjects. According to the similarity measurement between training and test data of the same modality, Ye et al. [28] proposed a patch-based modality propagation method. Through global search, the input patch was compared against the training patches in the dataset. Several nearest neighbors with similar properties were picked from the source domain and corresponding patches in the target modality used for image synthesis. In [29], a pseudo-CT synthesis algorithm was proposed, which aims at generating CT-like images from T1-w/T2-w inputs, using multi-atlas registration and tissue contrast fusion. Nguyen et al. [30] proposed a location-sensitive deep network method to integrate image intensities and spatial information into a deep network for cross-modality image synthesis. To verify the effectiveness of synthesized data, Tulder et al. used restricted Boltzmann machines to learn abstract representations from training data for synthesizing the missing image sequences. More recently, a nonlinear regression-based image synthesis approach [31] was proposed to predict the intensities in the target modality. While training, this method used registered image pairs from source and target modalities to learn a random forest regressor for regressing the target modality data. Besides these methods, Vemulapalli et al. proposed an unsupervised approach which relaxed needing registered image pairs during training, to deal with the synthesis problem.

In this work, we present a novel MRI simultaneous super-resolution and cross-modality synthesis (SiSCS) method for reconstructing the HR version of the target modality based on an LR image of the source modality while treating each 3D volume as a stack of 2D images. We simultaneously train a cross-modality dictionary pair based on registered patches of the LR source modality and the HR target modality. For an accurate image synthesis, the sparse codes of the LR source modality should be the same as those of the HR ground truth on the premise of high correlation between the paired LR source data and HR target data. We map high-frequency (HF) features of the registered image pairs between source and target modalities into a common feature space to fit the style-specific local structures and resolutions. We introduce patch-based global redundancy, consisting of cross-modal matching and self-similarity, to enhance the quality of image reconstruction based on sparse representations. Prior papers such as [32, 33] and follow-up studies [16, 17, 34, 35] have shown that self-similar image properties were used for enabling exact local image reconstruction. However, classical NL means [36] are computationally expensive. To overcome such problem, we present an integrated clustering algorithm into the original redundancy framework for making the data of the same class correlated and speeding up the similarity measure from each subclass. In addition, we set patches as the unit to preserve the intrinsic neighbor information of pixels and reduce the computational cost.

In summary, our method offers these four contributions: (1) We normalize the vectors of dictionary pairs in an HF feature space (rather than in the original image space) to a unified range to achieve intensity consistent learning. (2) A novel cross-modality dictionary learning based on a compact representation of HF features in both domains is proposed to derive co-occurrence prior. (3) Simultaneous estimation of the dictionaries corresponding to both modalities leads to matched representations for a given sparse code. (4) Sparse code pre-clustering provides a globally redundant reconstruction scheme incorporated into the local reconstruction model, enhances the robustness of the synthesis, and speeds up code search. Extensive experiments on a public dataset of brain MR images show that the proposed method achieves a competitive performance compared to other state-of-the-art algorithms. To the best of our knowledge, this work is the first to undertake SR reconstruction of an arbitrary target MRI modality from an available source MRI LR modality.

The rest of this work is organized as follows. The second part reviews the basis of super-resolution and dictionary learning techniques. The novel SiSCS method and experimental setup are presented in the third and fourth parts, respectively. The fourth part presents a comprehensive experimental evaluation of our technique in brain MRI. Finally, conclusions are provided in the fifth part.

21.2 Background

21.2.1 Image Degradation Model

SR image reconstruction, understood as an inverse problem, attempts to recover an HR image in matrix form $\mathbf{X}^H$ from an LR input $\mathbf{X}^L$. A degradation model (Fig. 21.1) is assumed as prior information to solving this inverse problem. In its simplest form, the source LR image $\mathbf{X}^L$ is modeled as a blurred and down-sampled counterpart of its HR image $\mathbf{X}^H$ by

$$\mathbf{X}^L = \mathcal{B}\mathcal{S}(\mathbf{X}^H), \qquad (21.1)$$

where $\mathcal{B}$ and $\mathcal{S}$ represent the blurring and down-sampling operators, respectively [6].

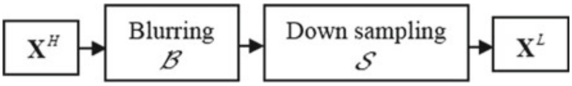


Fig. 21.1 The degradation model

21.2.2 Dictionary Learning

Dictionary learning has been successfully applied to a number of problems in image processing [20, 37], such as image restoration [38–40], denoising [32, 34, 36], and enhancement [10, 39, 41]. In image reconstruction based on dictionary learning, an image is normally treated as the combination of many patches [10, 32, 38–40] and denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{k\times N}$. An image is approximated as $\mathbf{X} \approx \boldsymbol{\Phi}\mathbf{A}$, where $\mathbf{X}$ is the target matrix being approximated, $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_K\right] \in \mathbb{R}^{k\times K}$ denotes a projection dictionary with K atoms, and $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_N] \in \mathbb{R}^{K\times N}$ is a set of N K-dimensional sparse codes of $\mathbf{X}$ with $\|\mathbf{A}\|_0 \ll K$. Representing Eq. (21.1) for sparse reconstruction of $\mathbf{X}^L$ regarding $\boldsymbol{\Phi}^L$ can be achieved by

$$\mathbf{X}^L \approx \boldsymbol{\Phi}^L\mathbf{A} = \mathcal{BS}(\boldsymbol{\Phi}^H\mathbf{A}), \tag{21.2}$$

where $\boldsymbol{\Phi}^L$ and Φ^H denotes an LR dictionary and an HR dictionary, respectively. For each image, the sparse decomposition is obtained by solving

$$\min_{\mathbf{A}} \|\mathbf{A}\|_0 \;\; s.t.\; \mathbf{X} = \boldsymbol{\Phi}\mathbf{A} \; (or \; \|\mathbf{X} - \boldsymbol{\Phi}\mathbf{A}\|_p \le \epsilon), \tag{21.3}$$

where $\|\cdot\|_0$ controls the number of nonzero elements in $\mathbf{A}$, and ϵ is used for managing the reconstruction errors. As shown in [42], the minimization problem as stated in Eq. (21.3) is an NP-hard problem under the l_0-norm with the l_1-norm to obtain a near-optimal solution [43]. The estimation is then accomplished by minimizing a least squares problem with a quadratic constraint, whose Lagrange multiplier formulation is

$$< \boldsymbol{\Phi}, \mathbf{A} >= \arg\min_{\boldsymbol{\Phi},\mathbf{A}} \|\mathbf{X} - \boldsymbol{\Phi}\mathbf{A}\|_2^2 + \lambda \,\|\mathbf{A}\|_1 \,, \tag{21.4}$$

where λ is a regularization factor trading-off the parametric sparsity and the reconstruction error of the solution.

21.3 Method

The proposed SiSCS method computes an estimation of an HR version of a target MRI modality based on an LR version of a source MRI modality using jointly learned dictionary. SR reconstruction in this work is inspired in earlier work on brain hallucination [11], with an assumption that an HR image can be reconstructed from the LR input with helped by another HR image using dictionaries of paired data in a sparse representation framework [26, 44]. In this work, we partition the images in the training database into a set of overlapping image patches. These image patches are built simultaneously on the source and target spaces by registered source–target image pairs. We propose a cross-modality dictionary learning that enforces the computation of joint sparse codes. Instead of working with the original data of the

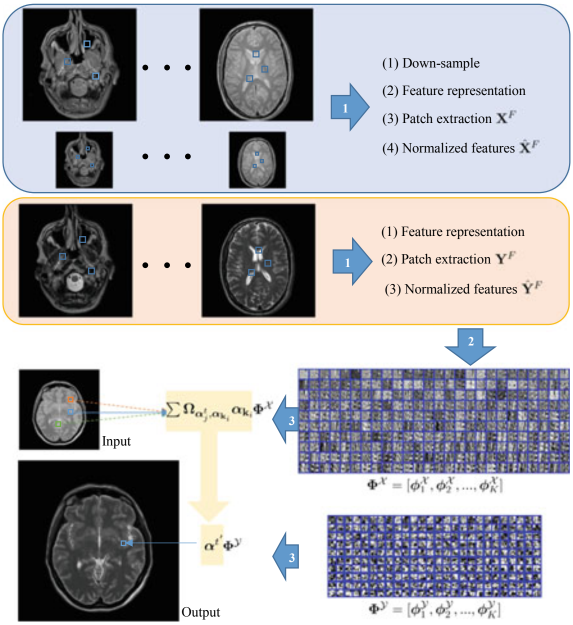

(1) Down-sample
(2) Feature representation
(3) Patch extraction $\mathbf{X}^F$
(4) Normalized features $\hat{\mathbf{X}}^F$
1
(1) Feature representation
(2) Patch extraction $\mathbf{Y}^F$
(3) Normalized features $\hat{\mathbf{Y}}^F$
1
2
3
3
Input
$\sum \Omega_{\alpha_j^t,\alpha_{k_i}} \alpha_{k_i} \Phi^{\mathcal{X}}$
$\Phi^{\mathcal{X}} = [\phi_1^{\mathcal{X}}, \phi_2^{\mathcal{X}}, ..., \phi_K^{\mathcal{X}}]$
$\alpha^{t'} \Phi^{\mathcal{Y}}$
Output
$\Phi^{\mathcal{Y}} = [\phi_1^{\mathcal{Y}}, \phi_2^{\mathcal{Y}}, ..., \phi_K^{\mathcal{Y}}]$

Fig. 21.2 Illustration of the SiSCS model. Step 1: Feature collection. Step 2: Cross-modality dictionary learning. Step 3: Globally redundant synthesis

paired patches, we choose an HF representation of the data in the gradient domain, so the sparse codes promote a high correlation between the two modalities regarding the LR and HR, respectively. In brief, given the test image in matrix form $\mathbf{X}^t$ (with modality $\mathcal{M}_1$), the proposed method will synthesize an SR image $\mathbf{Y}^t$ with modality $\mathcal{M}_2$ from $\mathbf{X}^t$ through a patch-based global redundant reconstruction model regarding the learned cross-modality dictionary pair. The entire framework of SiSCS model is summarized in Fig. 21.2.

21.3.1 Data Description

Let $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m\}$ be m training images of modality $\mathcal{M}_1$ in the source domain, and $\mathcal{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m\}$ be m training images of modality $\mathcal{M}_2$ in the target domain. We denote cross-modality image pairs as $\{\mathbf{X}_i, \mathbf{Y}_i\}$, while $\mathbf{X}_i$ and $\mathbf{Y}_i$ are registered. In this work, we consider the LR input and HR output and define the observed LR counterparts based on the HR images in $\mathcal{X}$ as Eq. (21.1). $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m\}$ is then updated as $\mathcal{X}^L = \left\{\mathbf{X}_1^L, \mathbf{X}_2^L, \ldots, \mathbf{X}_m^L\right\}$, and cross-modality image pairs can be rewritten as $\{\mathbf{X}_i^L, \mathbf{Y}_i\}$. After that, we build our algorithm based on these data.

21.3.2 Gradient Feature Representation

A mapping is constructed between each pair of LR and HR images based on HF edges and texture features. This feature scheme is reasonable from the perceptual viewpoint since humans are more sensitive to changes in HF content [12]. It has been shown that HF feature representations of LR images are arguably the most important for forecasting the missing HF components of HR images [10]. Such kind of feature representation also makes the sparse codes of paired data that possess the same information close to each other [12, 14, 26]. To achieve this, researchers have suggested using a high-pass filter [14]. In this context, we define a feature operator $\mathcal{F}$ to perform feature extraction for the LR image set. For this purpose, we follow [12] and adopt the first-order and second-order gradients to represent features of each LR image $\mathbf{X}_i^L$. The derivatives are then defined as [10, 12]

$$\begin{aligned} f_1^1 &= [-1, 0, 1] \quad , \quad & f_1^2 &= [-1, 0, 1]^T \\ f_2^1 &= [-2, -1, 0, 1, 2] \, , & f_2^2 &= [-2, -1, 0, 1, 2]^T , \end{aligned}$$

where each LR image results in four filtered images involving horizontal and vertical gradients for both orders by $\mathcal{F} * \mathbf{X}^L$, with $*$ respecting the convolution operator and $\mathcal{F}$ takes the form of one of the following operators: f_1^1, f_1^2, f_2^1, f_2^2. We then denote the features of LR images as $\mathbf{X}_i^F = \mathcal{F} * \mathbf{X}_i^L$. On the other hand, for the HR image set, we capture their HF features through directly removing the corresponding low-frequency information, which can be done by subtracting the mean value of HR data for $\mathbf{Y}_i$ [12], i.e., $\mathbf{Y}_i^F = \mathbf{Y}_i - mean(\mathbf{Y}_i)$. Further, images are treated as the collection of n patches and denoted as the matrices $\mathbf{X} = \left[\mathbf{x}_1^L, \mathbf{x}_2^L, \ldots, \mathbf{x}_n^L\right]$ and $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\right]$, and the corresponding HF features can be represented as $\mathbf{X}^F = \left[\mathbf{x}_1^F, \mathbf{x}_2^F, \ldots, \mathbf{x}_n^F\right]$ and $\mathbf{Y}^F = \left[\mathbf{y}_1^F, \mathbf{y}_2^F, \ldots, \mathbf{y}_n^F\right]$ in the source and target domains, respectively.

21.3.3 Cross-Modality Dictionary Learning

Following the dictionary learning procedure described in the second part, instead of considering the relationship between two sets of training data, we can learn two independent dictionaries [11] regarding the source domain and the target domain:

$$\begin{aligned} \mathbf{\Phi}^{\mathcal{X}} &= \arg \min_{\mathbf{\Phi}^{\mathcal{X}}, \mathbf{A}^{\mathcal{X}}} \left\| \mathbf{X}^F - \mathbf{\Phi}^{\mathcal{X}} \mathbf{A}^{\mathcal{X}} \right\|_2^2 + \lambda \left\| \mathbf{A}^{\mathcal{X}} \right\|_1 , \\ \mathbf{\Phi}^{\mathcal{Y}} &= \arg \min_{\mathbf{\Phi}^{\mathcal{Y}}, \mathbf{A}^{\mathcal{Y}}} \left\| \mathbf{Y}^F - \mathbf{\Phi}^{\mathcal{Y}} \mathbf{A}^{\mathcal{Y}} \right\|_2^2 + \lambda \left\| \mathbf{A}^{\mathcal{Y}} \right\|_1 . \end{aligned} \tag{21.5}$$

However, such a strategy is time-consuming and results in two sets of independent sparse codes. To solve a similar problem, Yang et al. [10] explored an image SR method that uses joint dictionary learning to correlate the sparse codes of LR data with those corresponding HR data. This is done by mapping LR and HR patch pairs into a common space to enforce the sparse codes of paired data possess the same value.

Based on this method, we develop a cross-modality dictionary learning algorithm using the features of registered patch pairs to build the mapping relationship for highly dissimilar training data. To proceed with the synthesis, a dictionary pair $\mathbf{\Phi}^{\mathcal{X}}$ and $\mathbf{\Phi}^{\mathcal{Y}}$ should be simultaneously trained from data relating both modalities and resolutions. For doing this, we first capture the HF features of both sets and then project them into a common space to achieve an effective correlation. Once the LR and HR patch pairs are incorporated in the feature space, we proceed with the joint dictionary learning. However, such a strategy fails to consider different modalities involving inconsistent intensity scales in the feature space. To solve this problem, we introduce a normalization function so we can handle dissimilar features within the same range. The maximum l_2-norm values are then computed for both feature sets:

$$\sigma^{\mathcal{X}} = \max\left\{\left\|\mathbf{x}_i^F\right\|_2\right\}, \ \sigma^{\mathcal{Y}} = \max\left\{\left\|\mathbf{y}_i^F\right\|_2\right\}. \tag{21.6}$$

Once $\sigma^{\mathcal{X}}$ and $\sigma^{\mathcal{Y}}$ are obtained, we use them for intensity normalization of all patch features, i.e.,

$$\hat{\mathbf{x}}_i^F = \frac{\mathbf{x}_i^F}{\sigma^{\mathcal{X}}}, \ \hat{\mathbf{y}}_i^F = \frac{\mathbf{y}_i^F}{\sigma^{\mathcal{Y}}}. \tag{21.7}$$

To maximize the correlation between normalized feature pairs in both modalities, we map them into a common high-dimensional space and propose a cross-modality dictionary learning method to simultaneously train two dictionaries for both datasets, leading to

$$\begin{aligned} &\arg\min_{\mathbf{\Phi}^{\mathcal{X}},\mathbf{\Phi}^{\mathcal{Y}},\mathbf{A}} \frac{1}{P}\left\|\hat{\mathbf{X}}^F - \mathbf{\Phi}^{\mathcal{X}}\mathbf{A}\right\|_2^2 + \frac{1}{Q}\left\|\hat{\mathbf{Y}}^F - \mathbf{\Phi}^{\mathcal{Y}}\mathbf{A}\right\|_2^2 \\ &+\lambda(\frac{1}{P}+\frac{1}{Q})\left\|\mathbf{A}\right\|_1 \quad s.t. \ \left\|\mathbf{\Phi}_i^{\mathcal{X}}\right\|_2^2 \le 1, \left\|\mathbf{\Phi}_i^{\mathcal{Y}}\right\|_2^2 \le 1, \end{aligned} \tag{21.8}$$

where $\frac{1}{P}$ and $\frac{1}{Q}$ are the regularization parameters for balancing two error terms, and P and Q represent the sizes of LR and HR patches, respectively. The above formulation is convex regarding each dictionary (or sparse codes) assuming the other one fixed. Constructing $\mathbf{\Phi}^{\mathcal{X}}$ and $\mathbf{\Phi}^{\mathcal{Y}}$ is achieved by alternating the computation of the sparse codes and the update of the dictionary pairs. We summarize the training part of our SiSCS method in Algorithm 1.

21.3.4 Clustering-Based Globally Redundant Codes

Once $\mathbf{\Phi}^{\mathcal{X}}$ and $\mathbf{\Phi}^{\mathcal{Y}}$ have been obtained from Eq. (21.8), we seek to reconstruct a test image $\mathbf{X}^t$ by sparsely representing the normalized features of $\mathbf{X}^t$ and $\mathbf{\Phi}^{\mathcal{X}}$. This is done by solving Eq. (21.4) as

$$\arg\min_{\mathbf{A}^t} \left\|\hat{\mathbf{X}}^t - \mathbf{\Phi}^{\mathcal{X}}\mathbf{A}^t\right\|_2^2 + \lambda\left\|\mathbf{A}^t\right\|_1, \tag{21.9}$$

```
Algorithm 1: SiSCS Training
Input: Training data $\mathbf{X}$ and $\mathbf{Y}$, parameters $\lambda$, $\mu$, $h$, $\gamma$.
Down sample and blur $\mathbf{X}$ by Eq. (21.1) to obtain $\mathbf{X}^L$.
Extract HF features and treat images as patches: $\mathbf{X}^F$, $\mathbf{Y}^F$.
Normalize patches by Eq. (21.7).
Initialize $\boldsymbol{\Phi}_0^{\mathcal{X}}$, $\boldsymbol{\Phi}_0^{\mathcal{Y}}$, $\mathbf{A}_0$.
while not converged do
    Update $\mathbf{A}_{i+1}$ by sparse coding in Eq. (21.8) with $\boldsymbol{\Phi}_i^{\mathcal{X}}$ and $\boldsymbol{\Phi}_i^{\mathcal{Y}}$ fixed.
    Update $\boldsymbol{\Phi}_{i+1}^{\mathcal{X}}$ and $\boldsymbol{\Phi}_{i+1}^{\mathcal{Y}}$ by dictionary learning in Eq. (21.8) with $\mathbf{A}_{i+1}$.
end
Output: $\boldsymbol{\Phi}^{\mathcal{X}}$, $\boldsymbol{\Phi}^{\mathcal{Y}}$.
```

where each patch of $\mathbf{X}^t$ is treated as its feature representation and normalized following Eq. (21.7) regarding LR and $\mathcal{M}_1$ modality by $\hat{\mathbf{X}}^t = \frac{\mathcal{F} * \mathbf{X}^t}{\sigma^{\mathcal{X}}}$, and $\mathbf{A}^t$ indicates the sparse coefficients of $\hat{\mathbf{X}}^t$. The estimated sparse codes can be directly used to synthesize the image $\mathbf{Y}^t$ of our desired modality $\mathcal{M}_2$ and HR by a linear combination of elements in the dictionary $\boldsymbol{\Phi}^{\mathcal{Y}}$, namely, $\mathbf{Y}^t = \boldsymbol{\Phi}^{\mathcal{Y}} \mathbf{A}^t$.

Integrating nonlocal reconstruction was successfully explored in [16, 33, 34]. Nonlocal reconstruction method recognizes that images often display repetitive patterns across the image field, and that at each location the local texture resembles a weighted combination of the local textures at other locations [33]. We then assume there exist patches in $\mathbf{X}^t$ and $\mathbf{X}$ that resemble the jth patch $\mathbf{X}_j^t$ of the test image. Groups of similar patches based on self- and cross-modal are identified. Then nonlocal means (NL means) [17, 33] is applied to synthesize each target patch, which is reconstructed as a weighted average of all similar patches. Each neighboring patch is weighed inversely proportionally to its distance to the reference patch in the source image [17]. The patch similarity calculations and global search involved in establishing the set of similar patches is computationally intensive. To speed up computing the distance between the reference patch and each patch in the training database, we perform a two-stage search that eliminates grossly dissimilar patches first, and then refines with a local search. This multi-level search is more robust to noise and also addresses the problem of searches leading to very few retrievals due to less repetitive patterns. The first level search is carried out using K-means clustering using as input the sparse codes of the source patch and based on the Euclidean distance (cf. Fig. 21.3).

Let Ψ be the collection of the normalized HF features collected from $\hat{\mathbf{X}}^t$ and $\hat{\mathbf{X}}^F$. Then, we assume that they provide with s observations $\{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_s\}$ leading to s coefficients $\left\{\boldsymbol{\alpha}_{\mathbf{k}_1}, \boldsymbol{\alpha}_{\mathbf{k}_2}, \ldots, \boldsymbol{\alpha}_{\mathbf{k}_s}\right\}$ from which we wish to generate K clusters $\left\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_K\right\}$. The clusters are computed as

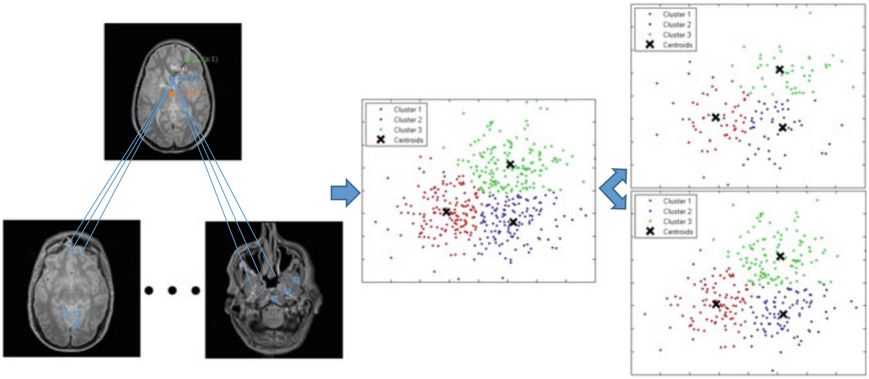


Fig. 21.3 Example of the clustering-based global redundancy model including cross-modal matching and self-similarity. For each reference patch in the test image, groups of similar patches including cross-modal matching and self-similarity can be found based on K-means clustering

$$
\begin{aligned}
&\arg\min_{\{\boldsymbol{\psi}_c\}_{c=1}^{K}} \sum_{c=1}^{K} \sum_{\mathbf{k}_i \in \boldsymbol{\psi}_c} \left\| \boldsymbol{\alpha}_{\mathbf{k}_i} - \boldsymbol{\delta}_c \right\|^2, \\
&\boldsymbol{\delta}_c = \frac{1}{\boldsymbol{\psi}_c} \sum_{\boldsymbol{\alpha}_{\mathbf{k}_i} \in \boldsymbol{\psi}_c} \boldsymbol{\alpha}_{\mathbf{k}_i},
\end{aligned}
\tag{21.10}
$$

where $\boldsymbol{\delta}_c$ is the mean vector for cth cluster $\{\boldsymbol{\psi}_c\}_{c=1}^{K}$. We pool the patches (other than the one to be synthesized) from the reference image with those in the training set as this has particular advantages when the reference image is abnormal or when the database is small. With the experiments reported with the IXI database, we found this is used in less than 2.76% of the subjects. In those cases, the nonlocal self-similarity has a stronger influence that the cross-subject self-similarity. This clustering yields a representative NL mean patch. We estimate the sparse codes for the jth patch of the test image as the weighted average of the sparse codes $\boldsymbol{\alpha}_{\mathbf{x}}$ associated with the corresponding cluster $\boldsymbol{\psi}_c$ via

$$
\hat{\boldsymbol{\alpha}}_j^t = \sum_{\boldsymbol{\alpha}_{\mathbf{k}_i} \in \boldsymbol{\psi}_c} \Omega_{\boldsymbol{\alpha}_j^t, \boldsymbol{\alpha}_{\mathbf{k}_i}} \boldsymbol{\alpha}_{\mathbf{k}_i},
\tag{21.11}
$$

where $\hat{\boldsymbol{\alpha}}_j^t$ is the optimized sparse codes, $\boldsymbol{\alpha}_{\mathbf{k}_i}$ denotes the sparse codes of $\mathbf{k}_i$ within the corresponding cluster, and $\Omega_{\boldsymbol{\alpha}_j^t, \boldsymbol{\alpha}_{\mathbf{k}_i}}$ is the weight for computing the level of similarity to be inversely proportional to the Euclidean distance between $\boldsymbol{\alpha}_j^t$ and $\boldsymbol{\alpha}_{\mathbf{k}_i}$, where

$$
\Omega_{\boldsymbol{\alpha}_j^t, \boldsymbol{\alpha}_{\mathbf{k}_i}} = \frac{1}{\mu} \exp\left\{ -\frac{\left\| \boldsymbol{\alpha}_j^t - \boldsymbol{\alpha}_{\mathbf{k}_i} \right\|_2^2}{h^2} \right\},
\tag{21.12}
$$

with $\Omega_{\boldsymbol{\alpha}_j^t,\boldsymbol{\alpha}_{\mathbf{k}_i}}$ satisfying $0 \le \Omega_{\boldsymbol{\alpha}_j^t,\boldsymbol{\alpha}_{\mathbf{k}_i}} \le 1$ and $\sum \Omega_{\boldsymbol{\alpha}_j^t,\boldsymbol{\alpha}_{\mathbf{k}_i}} = 1$, μ being a normalization constant, and h being a scalar. Similar to the NL means method, the coefficient with higher similarity to $\boldsymbol{\alpha}_j^t$ will have a larger weight on average (an example is shown in Fig. 21.3). Vectors within the cluster $\boldsymbol{\psi}_c$ contains not only test items but also training components. The conventional nonlocal method (e.g., NL means) cannot express the complex structures in MR images. In this scenario, our global redundancy approach can efficiently adapt to different structures. Therefore, the local sparse representation model in Eq. (21.9) that meets the complementary function to those of Eq. (21.11) will be modified as

$$\arg\min_{\mathbf{A}^{t'}} \left\|\hat{\mathbf{X}}^t - \boldsymbol{\Phi}^{\mathcal{X}}\mathbf{A}^t\right\|_2^2 + \lambda \left\|\mathbf{A}^t\right\|_1 + \gamma \left\|\mathbf{A}^t - \hat{\mathbf{A}}^t\right\|_2, \tag{21.13}$$

where γ is a tunable regularization parameter. Finally, we update the synthesized image via $\mathbf{Y}^t = \boldsymbol{\Phi}^{\mathcal{Y}}\mathbf{A}^{t'}$. As with most of the super-resolution methods [6, 10], the contents of an LR test image beside the HF components are first preserved by extracting the features of each patch and then added the subtracted mean values back into the reconstructed SR version. Considering the domain-specific information, we use the proposed globally redundant information to replace the original mean values from each patch of the test image. Then, we generate the target image by adding the pseudo-mean values into the obtained HF components. The pseudocode for cross-modality synthesis is shown in Algorithm 2.

Algorithm 2: SiSCS Synthesis

```
Input: Test image $\mathbf{X}^t$, dictionary pairs $\boldsymbol{\Phi}^{\mathcal{X}}$, $\boldsymbol{\Phi}^{\mathcal{Y}}$.
Extract features, divide patches and normalize: $\hat{\mathbf{X}}^t$.
Initialize $\mathbf{A}_0^t$.
while not converged do
    Update $\mathbf{A}_{i+1}^t$ by Eq. (21.9).
    Perform clustering by Eq. (21.10).
    Update $\mathbf{A}_{i+1}^{t'}$ using Eq. (21.13).
end
Compute $\mathbf{Y}_{i+1}^t \leftarrow \boldsymbol{\Phi}^{\mathcal{Y}}\mathbf{A}_{i+1}^{t'}$.
Output: Synthesized image $\mathbf{Y}^t$.
```

21.4 Experiments

To evaluate the performance of the proposed method, two scenarios were considered: (1) MR image super-resolution; (2) simultaneous SR and cross-modality synthesis. We evaluated our model on two datasets: IXI[1] (containing 578 $256\times256\times p$ MR

[1] http://brain-development.org/ixi-dataset/.

healthy subjects) and NAMIC[2] (including 19 $128\times128\times q$ subjects, ten are normal controls and nine are schizophrenic). In our experiment, we applied leave-one-out cross-validation where removing the testing image from the entire dataset and learn on the remaining ones. For the experimental settings, we first sliced each 3D volume as the 2D stacks and then treated the 2D slices as many patches of 5×5 pixels size. We randomly sampled 100,000 patch pairs for training the cross-modality dictionary pair. The regularization parameters h, γ were set to be 16 and 0.1, respectively. We took the factor of dictionary size and sparsity regularization parameter into consideration and fixed the dictionary size to 1024 and $\lambda = 0.15$ based on the quantitative analysis in Sects. 21.4.1 and 21.4.2. The parameter K of the K-means was fixed to 10 to guarantee each cluster had enough candidates. Finally, we adopt the widely used peak signal-to-noise ratio (PSNR) in decibels (dB) and structural similarity index (SSIM) [45] for illustrating the effectiveness of different methods. PSNR is employed to measure the quantitative evaluation of reconstruction of lossy compression codecs, which is defined as

$$\mathrm{PSNR}(s,t) = 10\,\log_{10}\left(\frac{\mathrm{MAX}^2 mn}{\sum_i^m \sum_j^n [s(i,j) - t(i,j)]^2}\right), \tag{21.14}$$

where MAX denotes the maximum pixel value, m and n are the sizes of the synthesized image s and its ground truth t, and $s(i, j)$ and $t(i, j)$ represent the pixels of s and t at positions i and j, respectively. SSIM is a metric for measuring the perceived visual image quality [45]. SSIM is calculated:

$$\mathrm{SSIM}(s,t) = \frac{(2\mu_s\mu_t + c_1)(2\sigma_{st} + c_2)}{(\mu_s^2 + \mu_t^2 + c_1)(\sigma_s^2 + \sigma_t^2 + c_2)}, \tag{21.15}$$

where μ_s and μ_t are the mean values in s and t; σ_s and σ_t are their standard deviations; σ_{st} is the covariance of s and t; and $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ with L, the image intensity dynamic range, and $k_1 = 0.01$, $k_2 = 0.03$ [45].

21.4.1 Dictionary Size

Larger dictionaries can yield higher accuracy at the price of more calculations. In this experiment, we randomly picked 10 PD-w subjects to test the influence of four dictionary sizes (*viz.* 256, 512, 1024, and 2048) on both SR and simultaneous SR and cross-modality synthesis (PD-w→T2-w). Table 21.1 specifies relevant times for training dictionaries of different sizes, and the averaged PSNRs and SSIMs for image reconstructions using different dictionaries. From Table 21.1, we can see that a larger dictionary contributes a better estimation with larger PSNRs and SSIMs at a higher

[2] http://hdl.handle.net/1926/1687.

Table 21.1 Effects of dictionary size on sr reconstruction and simultaneous super-resolution and cross-modality synthesis

	Dictionary size	256	512	1024	2048
Cost	(min)	**8.51**	12.96	18.68	28.77
SR	PSNR (dB)	45.29	45.35	45.43	**45.62**
	SSIM	0.9865	0.9867	0.9867	**0.9872**
SiSCS	PSNR (dB)	39.54	39.55	39.57	**40.00**
	SSIM	0.8995	0.8995	0.8996	**0.8997**

computation cost. We selected the size 1024 to yield a good image quality within acceptable computational time.

21.4.2 Sparsity

In Eq. (21.4), λ plays an important role in the sparse representation as it is used for controlling the sparsity of the results. Empirically, λ is suitable from 0 to 0.85 [5, 26] for maintaining the model stability. In this section, we assess how λ influences the results through quantifiably measuring the PSNRs and SSIMs of the reconstructed image for different λs. To evaluate this, we utilized the same test data reported in Sect. 21.4.1 and fixed the dictionary size to 1024. The experimental results are listed in Table 21.2. As shown, $\lambda \in [0.15, 0.45]$ yielded better performance, especially when $\lambda = 0.15$, the results on both scenarios achieve the highest PSNRs and SSIMs among all reconstructions. To comprehensively analyze the most suitable sparsity value for our algorithm, we computed the elapsed time for $\lambda \in [0.10, 0.85]$ and show the results in Table 21.2. As λ increased, the computational cost decreased, and the quality of reconstruction declined. Therefore, we chose no larger value; rather, a smaller λ was selected for achieving better results. We finally chose a sparsity parameter of 0.15.

Table 21.2 Error measures of SR resolution and simultaneous super-resolution and cross-modality synthesis for different sparsity values

	λ	0.10	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85
Cost	(min)	12.55	11.06	10.97	10.34	9.17	8.13	8.10	**8.02**	8.25
SR	PSNR(dB)	47.41	**49.85**	49.80	46.47	40.82	36.93	36.92	36.90	36.90
	SSIM	0.9935	**0.9962**	0.9960	0.9932	0.9831	0.9429	0.9429	0.9428	0.9428
SiSCS	PSNR(dB)	39.36	**39.39**	39.32	37.62	35.46	34.95	34.95	34.94	34.93
	SSIM	0.9066	**0.9077**	0.9076	0.9055	0.8667	0.8500	0.8500	0.8499	0.8498

21.4.3 MRI Super-Resolution

First, we evaluated the performance of our clustering-based global redundancy framework for MR image SR on all PD-w subjects of IXI dataset. Generally, LR images can be generated by first blurring the HR images with a 2D Gaussian kernel with standard deviation 1.6 and then down-sampling the blurred images by scaling factor 2 in both horizontal and vertical directions. To ensure the one-to-one correspondence for each extracted LR-HR patch pair, we further up-sampled the LR images by factor of 2 using bi-cubic interpolation (making the SR problem more challenging) and then extracted patches from them. This ensures that samplings from the same locations of both domains indicate the same tissue information. Based on the extracted patch pairs, we can train the corresponding dictionaries. Finally, we inputted an LR counterpart of the test image for reconstructing its HR image via the proposed model with a scaling factor of 2. To show the performance of our approach, we compared our results with these methods: nearest-neighbor interpolation (nearest), bi-cubic interpolation (Bi-cubic), nonlocal MRI up-sampling (NLM) [17], and single-image SR (SSR) of brain MRI [13]. For all experiments, we tuned parameters in the characteristics of each method and demonstrated their best results among overall records by PSNRs and SSIMs.

Figure 21.4 presents a comparison between the SR reconstructed PD-w slices based on different methods. Top row shows the axial views of the SR results for a subject. In the second and third rows, we zoom in two specific regions for better visualization. The last row provides PSNRs and SSIMs for each listed method. The proposed method outperforms all other methods displaying the highest PSNR and SSIM. Although NLM yields a cleaner image with noise lower than bi-cubic interpolation, its effectiveness is nearly the same as bi-cubic. Figure 21.5 provides quantitative results on all PD-w subjects of IXI dataset. Our method achieved the highest PSNR and SSIM compared to other methods.

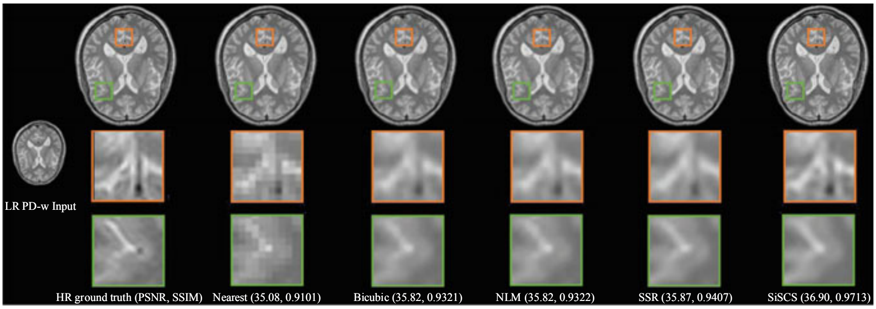


Fig. 21.4 Comparison of the SR results with ground truth

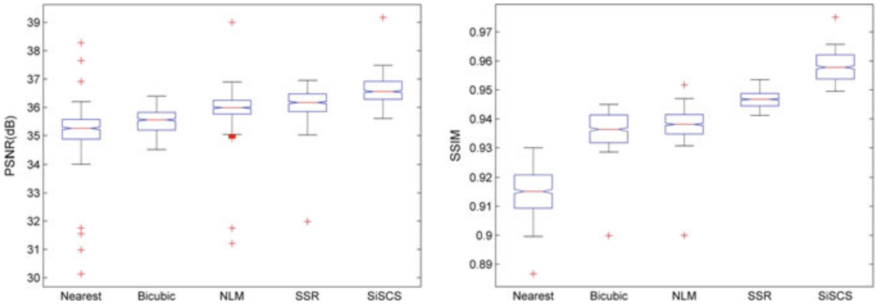


Fig. 21.5 Boxplots of error measures (PSNRs, SSIMs) for SR reconstructions with different methods

21.4.4 Simultaneous Super-Resolution and Cross-Modality Synthesis

We then addressed the problem of simultaneous super-resolution and cross-modality synthesis by evaluating our SiSCS model on both IXI and NAMIC datasets. First, we used PD-w and T2-w subjects from IXI for synthesizing HR T2-w image considering LR PD-w acquisition from the same subject and *vice versa*. Second, generating HR T2-w image from LR PD-w input based on the preprocessed data (i.e., performing skull strapping and bias corrections[3]) and *vice versa*. Third, we considered the generation of T1-w image based on T2-w input and *vice versa*. We conducted the first two sets of experiments on the IXI dataset, while the third one was explored on the NAMIC dataset. The representative and state-of-the-art synthesis methods including MR image example-based contrast synthesis (MIMECS) [26] approach, Vemulapalli's supervised method (V-s) [46] and Vemulapalli's unsupervised method (V-us) [46] were employed to compare with our SiSCS. However, Vemulapalli's methods are limited by single cross-modality synthesis used in the NAMIC dataset. Therefore, original data (without degradation processing) were used in all Vemulapalli's methods. All existing synthesis approaches must preprocess the data first. In our algorithm, such preprocessing is unnecessary and can be exchangeable which can be validated in the first set of experiments. The benefits of performing preprocessing are reflected in the reduction of the interference by non-desired tissue compartments, as the skull. However, such processes also bring problems, for instance, the lack of tissue contrast due to the imprecise skull stripping.

For the first two sets of experiments, we evaluated our algorithm and compared with MIMECS while displaying results in Fig. 21.6 for visual inspection. For each scenario, we applied the proposed method on IXI from the PD-w and T2-w subjects and reported the quantitative results in Fig. 21.8. Our algorithm performs consistently across the whole dataset, reaching the best performance for almost all subjects. We

[3] Following [28, 46], all the experiments data were skull stripped, linear registered and/or inhomogeneity corrected.

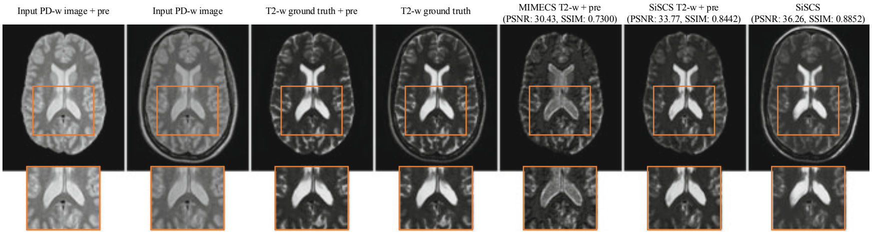


Fig. 21.6 Axial views of synthesized HR T2-w examples based on the LR PD-w inputs using different methods (zoom in for details)

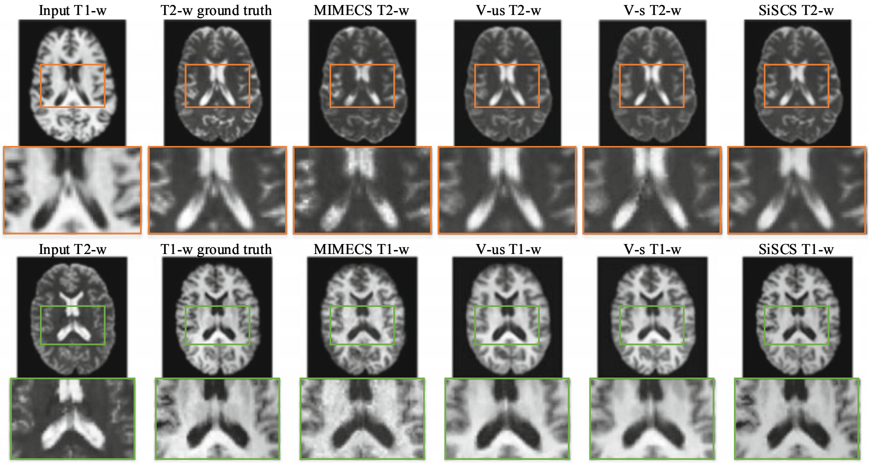


Fig. 21.7 Visual comparison of synthesized results using different methods on the NAMIC dataset (zoom in for details)

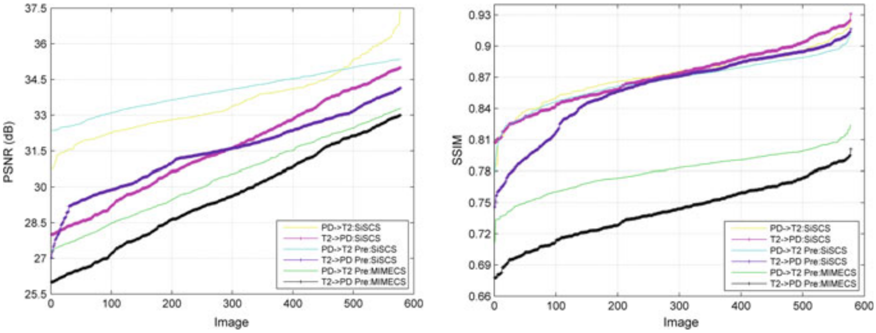


Fig. 21.8 Overall performance comparison between the proposed method and MIMECS on the IXI dataset

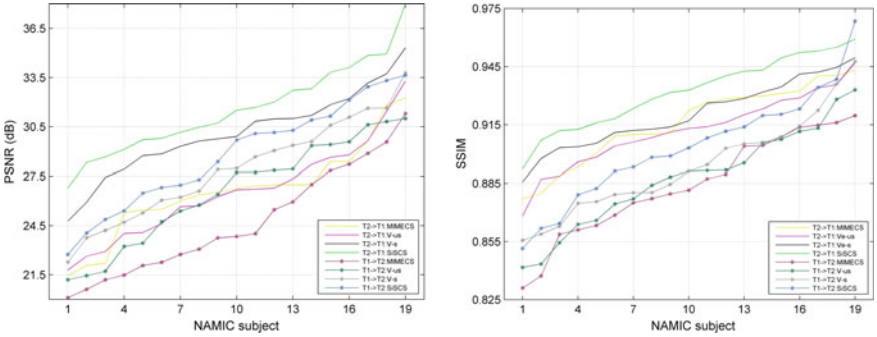


Fig. 21.9 Synthesis performance comparison between SiSCS and other state-of-the-art methods on the NAMIC dataset

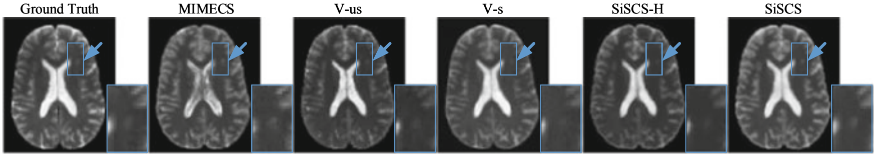


Fig. 21.10 Synthesis result of a pathological case comparison between SiSCS and other state-of-the-art methods

evaluated SiSCS and relevant methods in the third scenario, allowing us to comprehensively compare the performance of the proposed method in both healthy and pathological cases with the recently published algorithms. The advantage of SiSCS over other methods is shown in Fig. 21.7, and the close-up views of the selected parts are also provided for better visualization. The overall performance comparison is given in Fig. 21.9. From Fig. 21.9, we can see that SiSCS is always better than MIMECS and Vemulapalli's approaches. This demonstrates the effectiveness of our simultaneous SR and cross-modality technique.

The following experiments show synthesized images of schizophrenic patients [47]. We carry out simultaneous SR and synthesis in two different learning scenarios: a) dictionary learning based on healthy subjects (denoted by SiSCS-H), and b) dictionary learning based on both healthy and schizophrenic (denoted by SiSCS) cases. In both scenarios, we report synthesis results corresponding to schizophrenia cases only. Figure 21.10 provides visual results of various synthesis methods of an illustrative patient. Table 21.3 summarizes key quantitative performance metrics over the total set of nine schizophrenic subjects. Both visual and quantitative results show that, compared to earlier methods, our approach provides the best results. Our experiments also show that SiSCS-H is outperformed by SiSCS trained on both control and pathologic cases and tested using cross-validation.

Table 21.3 Average assessment measures for image synthesis of nine pathological cases

Case		MIMECS	V-us	V-s	SiSCS-H	SiSCS
T1->T2	PSNR (dB)	22.95	23.87	26.69	24.86	**27.03**
	SSIM	0.8698	0.8701	0.8895	0.8712	**0.8912**
T2->T1	PSNR (dB)	27.38	27.47	29.12	27.91	**30.01**
	SSIM	0.9000	0.9002	0.9087	0.9003	**0.9177**

21.5 Conclusion

We present a novel approach to simultaneous super-resolution and cross-modality synthesis (SiSCS) in brain MRI. SiSCS first learns a cross-modality dictionary in a high-frequency space. Then, SiSCS reconstructs the target image using a patch-based cross-modal estimation model with a nonlocal sparse image representation. We conducted extensive experimental assessment of our technique in both health and schizophrenic subjects. Across experiments, both on PSNR and SSIM metrics, SiSCS outperformed three major contending techniques. Experiments showed consistent outperformance across super-resolution and joint super-resolution and cross-modality synthesis, respectively. In our experiments, we showed that dictionary learning for synthesis of schizophrenic images requires that pathological sets are included.

References

1. Wang, Z. Lemmon, M (2015) Stability analysis of weak rural electrification microgrids with droop-controlled rotational and electronic distributed generators. In: Power and energy society general meeting. IEEE, pp 1-5
2. Lemann TM, Gonner C, Spitzer K (1999) Survey: interpolation methods in medical image processing [J]. IEEE Trans on Med Imaging 18(11)
3. Grevera GJ, Udupa JK (1998) An objective comparison of 3-D image interpolation methods. IEEE Trans Med Imaging 17(4):642–652
4. Herman GT, Rowland SW, Yau MM (1979) A comparative study of the use of linear and modified cubic spline interpolation for image reconstruction. IEEE Trans Nuclear Sci 26(2):2879–2894
5. Stytz MR, Parrott RW (1993) Using kriging for 3D medical imaging. Comput Med Imaging Graph 17(6):421–442
6. Van Ouwerkerk JD (2006) Image super-resolution survey. Image Vis Comput 24(10):1039–1052
7. Ongie G, Jacob M (2015) Recovery of discontinuous signals using group sparse higher degree total variation. IEEE Signal Process Lett 22(9):1414–1418
8. Greenspan H, Oz G, Kiryati N, Peled S (2002) MRI inter-slice reconstruction using super-resolution. Magn Reson Imaging 20(5):437–446
9. Shi F, Cheng J, Wang L, Yap PT, Shen D (2015) LRTV: MR image super-resolution with low-rank and total variation regularizations. IEEE Trans Med Imaging 34(12):2459–2466

10. Yang J, Wright J, Huang TS, Ma Y (2010) Image super-resolution via sparse representation. IEEE Trans Image Process 19(11):2861–2873
11. Rousseau F (2008) Brain hallucination. In: European conference on computer vision. Springer, Berlin, pp 497–508
12. Chang H, Yeung DY, Xiong Y (2004) Super-resolution through neighbor embedding. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition. CVPR 2004, vol 1. IEEE, pp I–I
13. Rueda A, Malpica N, Romero E (2013) Single-image super-resolution of brain MR images using overcomplete dictionaries. Med Image Anal 17(1):113–132
14. Freeman WT, Pasztor EC, Carmichael OT (2000) Learning low-level vision. Int J Comput Vis 40(1):25–47
15. Sun J, Zheng NN, Tao H, Shum HY (2003). Image hallucination with primal sketch priors. In: Proceedings 2003 IEEE Computer Society Conference on computer vision and pattern recognition, vol 2. IEEE, pp II–729
16. Manjn JV, Carbonell-Caballero J, Lull JJ, Garca-Mart G, Mart-Bonmat L, Robles M (2008) MRI denoising using non-local means. Med Image Anal 12(4):514–523
17. Manjn JV, Coup P, Buades A, Fonov V, Collins DL, Robles M (2010) Non-local MRI upsampling. Med Image Anal 14(6):784–792
18. Huang Y, Shao L, Frangi AF (2017) Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding. arXiv:1705.02596
19. Huang Y, Beltrachini L, Shao L, Frangi AF (2016) Geometry regularized joint dictionary learning for cross-modality image synthesis in magnetic resonance imaging. In: International workshop on simulation and synthesis in medical imaging. Springer, Cham, pp 118–126
20. Huang Y, Shao L, Frangi AF (2018) Cross-modality image synthesis via weakly coupled and geometry co-regularized joint dictionary learning. IEEE Trans Med Imaging 37(3):815–827
21. Bogunovic H, Pozo JM, Villa-Uriol MC, Majoie CB, van den Berg R, Gratama van Andel HA, Macho JM, Blasco J, San Romn L, Frangi AF (2011) Automated segmentation of cerebral vasculature with aneurysms in 3DRA and TOF?MRA using geodesic active regions: an evaluation study. Med Phys 38(1):210–222
22. Nyl LG, Udupa JK, Zhang X (2000) New variants of a method of MRI scale standardization. IEEE Trans Med Imaging 19(2):143–150
23. Christensen JD (2003) Normalization of brain magnetic resonance images using histogram even-order derivative analysis. Magn Reson Imaging 21(7):817–820
24. Robitaille N, Mouiha A, Crpeault B, Valdivia F, Duchesne S (2012) Tissue-based MRI intensity standardization: application to multicentric datasets. J Biomed Imaging 2012:4
25. Shinohara RT, Sweeney EM, Goldsmith J., Shiee, N., Mateen, FJ, Calabresi, PA, Jarso S, Pham DL, Reich DS, Crainiceanu CM, Alzheimer's Disease Neuroimaging Initiative (2014) Statistical normalization techniques for magnetic resonance imaging. NeuroImage: Clinical 6:9–19
26. Roy S, Carass A, Prince JL (2013) Magnetic resonance image example-based contrast synthesis. IEEE Trans Med Imaging 32(12):2348–2363
27. Thaipanich T, Oh BT, Wu PH, Xu D, Kuo CCJ (2010) Improved image denoising with adaptive nonlocal means (ANL-means) algorithm. IEEE Trans Consum Electron 56(4)
28. Ye DH, Zikic D, Glocker B, Criminisi A, Konukoglu E (2013) Modality propagation: coherent synthesis of subject-specific scans with data-driven regularization. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 606–613
29. Burgos N, Cardoso MJ, Thielemans K, Modat M, Pedemonte S, Dickson J, Barnes A, Ahmed R, Mahoney CJ, Schott JM, Duncan JS (2014) Attenuation correction synthesis for hybrid PET-MR scanners: application to brain studies. IEEE Trans Med Imaging 33(12):2332–2341
30. Van Nguyen H, Zhou K, Vemulapalli R (2015) Cross-domain synthesis of medical images using efficient location-sensitive deep network. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 677–684

31. Jog A, Carass A, Roy S, Pham DL, Prince JL (2017) Random forest regression for magnetic resonance image synthesis. Med Image Anal 35:475–488
32. Elad M, Aharon M (2006) Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans Image Process 15(12):3736–3745
33. Buades A, Coll B, Morel JM (2010) Image denoising methods: a new nonlocal principle. SIAM Rev 52(1):113–147
34. Dong W, Zhang L, Shi G, Li X (2013) Nonlocally centralized sparse representation for image restoration. IEEE Trans Image Process 22(4):1620–1630
35. Zhu F, Shao L, Lin M (2013) Multi-view action recognition using local similarity random forests and sensor fusion. Pattern Recognit Lett 34(1):20–24
36. Yan R, Shao L, Cvetkovic SD, Klijn J (2012) Improved nonlocal means based on pre-classification and invariant block matching. J Display Technol 8(4):212–218
37. Huang Y, Zhu F, Shao L, Frangi AF (2016) Color object recognition via cross-domain learning on RGB-D images. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1672–1677
38. Mairal J, Elad M, Sapiro G (2008) Sparse representation for color image restoration. IEEE Trans Image Process 17(1):53–69
39. Shao L, Gao X, Li H (2014) Image restoration and enhancement: recent advances and applications. Signal Process 103:1–5
40. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311
41. Frangi AF, Niessen WJ, Vincken KL, Viergever MA (1998) Multiscale vessel enhancement filtering. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 130-137
42. Davis G, Mallat S, Avellaneda M (1997) Adaptive greedy approximations. Constr Approx 13(1):57–98
43. Chen SS, Donoho DL, Saunders MA (2001) Atomic decomposition by basis pursuit. SIAM Rev 43(1):129–159
44. Kainz B, Steinberger M, Wein W, Kuklisova-Murgasova M, Malamateniou C, Keraudren K, Torsney-Weir T, Rutherford M, Aljabar P, Hajnal JV, Rueckert D (2015) Fast volume reconstruction from motion corrupted stacks of 2D slices. IEEE Trans Med imaging 34(9):1901–1913
45. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
46. Vemulapalli R, Van Nguyen H, Zhou kS (2015) Unsupervised cross-modal synthesis of subject-specific scans. In: Proceedings of the IEEE international conference on computer vision, pp 6309–638
47. Shenton ME, Dickey CC, Frumin M, McCarley RW (2001) A review of MRI findings in schizophrenia. Schizophr Res 49(1–2):1–52

Index

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