DATA-EFFICIENT, MEMORY-EFFECTIVE, AND SHAPE PRIORS-CONSTRAINED LEARNING FOR SEGMENTING MEDICAL IMAGES

YOUYI SONG

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University

School of Nursing

Data-efficient, Memory-effective, and Shape Priors-constrained Learning for Segmenting Medical Images

Youyi Song

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Sept 2021

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signature)

Youyi Song (Name of student)

Abstract

This thesis aims at advancing deep learning models on medical image segmentation tasks by investigating three key problems: alleviating the burden on training data collection, reducing GPU memory consumption, and leveraging shape priors to boost the performance. The main results are five generic and effective approaches to these three problems, called selective learning, adversarial redrawing, surface projection, shape constructing, and shape mask generator, respectively.

Selective learning is a simple training framework that alleviates the burden on training data collection by using external data. The key idea is to learn a weight for each external data such that informative external data can have large weights and thus contribute more to the training loss, thereby implicitly encouraging the network to mine more valuable knowledge from them while suppressing to memorize irrelevant patterns from ‘useless’ or even ‘harmful’ data.

Adversarial redrawing is an unsupervised segmentation method for alleviating the burden of collecting training annotation. It is developed under the assumption that the imaging process can be modeled by a latent variable with two steps: objects’ binary mask generating (equivalent to segmentation) and objects’ intensity value drawing. It then uses the adversarial learning paradigm to train two deep networks to model the mask generating and

intensity drawing steps, by altering their parameters' value until they can generate images that cannot be distinguished by the discriminator.

Surface projection is a GPU memory-efficient learning technique that enables 2D networks to learn 3D features. We observe that boundary pixels of a 3D object form a surface that can be described by a 2D variable, and so 2 networks should be able to recognize these boundary pixels. We hence learn 3D features by using a 2D network to learn the projection distance mapping between the object's surface and a set of sampled spherical surfaces.

Shape constructing is a productive approach to modeling shape priors. The key idea is to leverage contour fragments rather than pixels to model shape priors, as fragments provide far more informative geometric information and shape cues. It is developed as an iterative algorithm of three key processes: fragments grouping, shape templates estimation, and fragments connecting, for progressively refining the modeled shape priors.

Shape mask generator is an effective method that models shape priors by learning how to refine the modeled ones. It first models shape priors from shape templates and then produces objects' shape masks according to the modeled shape priors. It next refines the modeled shape priors by minimizing a quantity, the generating residual, whose value is smaller when the produced shape masks are more accurate.

All five methods are assessed on publicly available datasets, with positive results obtained on extensive experiments, showing performance gains of

them against existing methods. These methods hence have great potential to advance deep learning models on a wide range of medical image segmentation tasks.

Keywords: Medical Image Segmentation · Deep Learning Models · Training Data Alleviation · GPU Memory Reduction · Shape Priors Exploitation.

List of Publications

1: **Youyi Song**, Zhen Yu, Teng Zhou, Jeremy Yuen-Chun Teoh, Baiying Lei, Kup-Sze Choi, and Jing Qin, "Learning 3D Features with 2D CNNs via Surface Projection for CT Volume Segmentation", *International Conference on Medical Image Computing and Computer-Assisted Interventions* (MICCAI), pp. 176-186 (2020)

2: **Youyi Song**, Lei Zhu, Baiying Lei, Bin Sheng, Qi Dou, Jing Qin, and Kup-Sze Choi, "Shape Mask Generator: Learning to Refine Shape Priors for Segmenting Overlapping Cervical Cytoplasms", *International Conference on Medical Image Computing and Computer-Assisted Interventions* (MICCAI), pp. 639-649 (2020)

3: **Youyi Song**, Teng Zhou, Jeremy Yuen-Chun Teoh, Jing Zhang, and Jing Qin, "Unsupervised Learning for CT Image Segmentation via Adversarial Redrawing", *International Conference on Medical Image Computing and Computer-Assisted Interventions* (MICCAI), pp. 309-320 (2020)

4: **Youyi Song**, Zhen Yu, Teng Zhoum Jeremy Yuen-Chun Teoh, Baiying Lei, Kup-Sze Choi, and Jing Qin, "CNN in CT Image Segmentation: Beyond Loss Function for Exploiting Ground Truth Images", *IEEE International Symposium on Biomedical Imaging* (ISBI), pp. 325-328 (2020)

5: **Youyi Song**, Lei Zhu, Jing Qin, Baiying Lei, Bin Sheng, and Kup-Sze Choi, "Segmentation of Overlapping Cytoplasm in Cervical Smear Images via Adaptive Shape Priors Extracted from Contour Fragments", *IEEE Transactions on Medical Imaging*, 38(12), 2849-2862 (2019)

6: **Youyi Song**, Jing Qin, Baiying Lei, Shengfeng He, and Kup-Sze Choi, "Joint Shape Matching for Overlapping Cytoplasm Segmentation in Cervical Smear Images", *IEEE International Symposium on Biomedical Imaging* (ISBI), pp. 191-194 (2019)

7: **Youyi Song**, Jing Qin, Baiying Lei, and Kup-Sze Choi, "Automated Segmentation of Overlapping Cytoplasm in Cervical Smear Images via Contour Fragments", *AAAI Conference on Artificial Intelligence* (AAAI), pp. 168-175 (2018)

8: **Youyi Song**, Lequan Yu, Baiying Lei, Kup-Sze Choi, and Jing Qin, "Selective Learning from External Data for CT Image Segmentation", *International Conference on Medical Image Computing and Computer-Assisted Interventions* (MICCAI 2021), Accepted.

9: Shuangyi Zhang, **Youyi Song**, Dazhi Jiang, Teng Zhou, and Jing Qin, "Noise-identified Kalman Filter for Short-term Traffic Flow Forecasting", *International Conference on Mobile Ad-Hoc and Sensor Networks* (MSN), pp. 462-466 (2019) (Co-first author)

10: Zhen Yu, **Youyi Song**, and Jing Qin, “Dense Pyramid Context Encoder-decoder Network for Kidney Lesion Segmentation”, *University of Minnesota Libraries Publishing* (2019)

11: Huakang Lu, Zuhao Ge, **Youyi Song**, Dazhi Jiang, Teng Zhou, and Jing Qin, “A Temporal-aware LSTM Enhanced by Loss-switch Mechanism for Traffic Flow Forecasting”, *Neurocomputing*, 427, 169-178 (2021)

Acknowledgements

First and foremost, I would like to thank my chief supervisor, Dr. Jing Qin, for his valuable guidance, support, and encouragement during the Ph.D. period. I got so many skills from him about research problem identification, flexible solution recognization, and data analysis.

I also want to express my sincere gratitude to my co-supervisor, Prof. Kup-Sze Choi, for his excellent supervision and valuable suggestions. His detailed comments and suggestions on my manuscripts and presentations helped me a lot.

I was fortunate to work with all great researchers in the Centre for Smart Health in the Hong Kong Polytechnic University. It is so helpful of their valuable and critical comments on my presentation skills.

Additionally, I have to thank all the administrative staff in the School of Nursing in the Hong Kong Polytechnic University. It is all of their contributions that give me a friendly and creative studying environment.

Lastly, I dedicate this tiny success to my family for their unconditional love, understanding, patience, encouragement, and support. Without their substantial help, this study period would be far more difficult.

Table of Contents

List of Figures

List of Tables

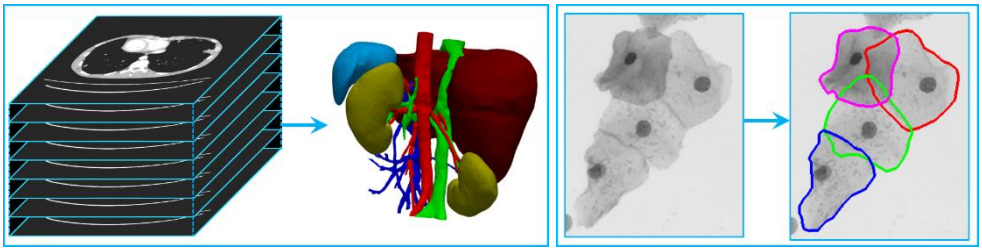
Chapter 1

Introduction

This thesis investigates medical image segmentation by developing deep learning models, which is a fundamental task making it possible to measure the quantitative information of objects. We begin with a brief account of medical image segmentation and its importance, following with the identified research problems and our contributions to solving these problems. We close this chapter by presenting the significance of our contributions and the structure of this thesis.

1.1 Research Task

Medical image segmentation is a task that aims to find the binary mask of objects of interest in the given medical images. In the deep learning context, this task is formulated as a learning problem, for learning a label assignment function that assigns categorical labels to pixels [1-2]. The space of assignment functions is specified by the network's architecture. Learning is to search an assignment function from the specified space that performs well enough in the training dataset, by altering the value of network's parameters through the lens of minimizing the loss function, a measure that is designed to have a small value when the network performing well.



(a) (b)

Fig. 1.1 Illustration of two types of segmentation tasks we addressed in this thesis: (a) point-to-point mapping, the most common case, and (b) point-to-multi-points mapping, caused by objects' overlapping, arising often in microscope images.

In this thesis, we study two types of segmentation tasks: (1) point-to-point mapping and (2) point-to-multi-points mapping, as shown in Fig. 1.1. In the first type, a pixel can be segmented into just one object. This is the most common case. In the second type, some pixels may belong to several objects. This is encountered often in the microscope images, caused by objects' overlapping. It is worth to note here that these two types cover almost all medical image segmentation scenarios.

1.2 Task Importance

Medical image segmentation is a fundamental task in the medical image analysis field, having been studied for ages and still being a hot topic in both industry and academia [1-6]. It is the function for this task to simplify the image representation that makes this task so important. The simplified binary representation allows us to measure object-level information, which underpins a huge range of medical research and clinical applications [1-6].

We below present three examples to detail the importance. The first one is disease diagnosis in which objects' shape or size information is often required. For example, for screening cervical cancer, the size ratio of the nucleus against cytoplasm is significant to decide whether that cell has been becoming abnormal [7-8]. The second example is injury prediction, like the kidney, in which we have to analyze the kidney's shape and position changes varying the time [133-134]. The third one is the postoperative follow-up in which we have to track objects' shape-changing rate [135-136]. Besides them, there are many other applications depending on medical image segmentation [137-150].

1.3 Research Problems

Medical image segmentation has been studied for ages, as mentioned above, and so there are many algorithms [1-6]. In this thesis, we investigate deep learning models. Deep learning models, though have achieved remarkable success, still face many problems to be addressed for applying them to practical applications. To this end, we here study three key problems for further advancing deep learning models in medical image segmentation.

Problem 1: Expensive Burden on Training Data Collection. The success of deep networks heavily relies on a large amount of training data, which is not always available. It is time-consuming or even prohibitively expensive for medical image segmentation tasks to collect an appropriate amount of

training data for deep learning models. Besides the image itself, collecting pixel-wise annotations requires tedious efforts from domain experts, with multiple rounds to correct annotation errors for reaching a consensus among experts. This problem makes deep networks in fact are very expensive to employ, substantially restricting their usability.

Problem 2: Unaffordable GPU Memory for Learning 3D Features. In a wide range of medical image segmentation tasks, 3D features in nature are desired, for example for segmenting CT (computed tomography) and MRI (magnetic resonance imaging) images. It is computationally expensive for 3D deep networks to learn 3D features, however. 3D networks consume GPU memory cubically with the increasing of pixel's resolution, making them memory-prohibitive to learn from high-resolution 3D medical images. Learning from low-resolution data, however, results in information loss, and some pixels are then becoming indistinguishable, greatly degrading the learning performance, considerably restricting their scalability and usability.

Problem 3: Unguaranteed Modeling of Shape Priors. Shape priors in medical image segmentation can be understood as a form of anatomical constraints to some extent. These clinical constraints should be satisfied for practical applications. However, deep learning networks learn shape priors in an implicit manner, enforcing their outputs as similar to the annotations as possible, with no guarantee on eliminating violations; it is not rare in practice

to see visually implausible segmentation results of deep networks. This problem can seriously harm the clinical significance of deep learning models.

These three problems are significant to advance deep learning models in medical image segmentation tasks. Without proper treatment on them, using deep models needs an expensive budget on collecting training data and unaffordable GPU memory consumption for training, but what is even worse is probably yielding paradoxical results, all substantially hindering the progress of deep models in the real-world medical image segmentation tasks.

1.4 Contributions

The main contributions are five effective and generic methods to address these three problems. They show positive results on extensive experiments, outperforming existing methods, and thus have great potential to advance deep learning models in medical image segmentation. Two methods, called selective learning and adversarial redrawing, are designed for alleviating the burden on training data collection by resorting to external data and training deep networks without annotations, respectively. Surface projection is to reduce GPU memory consumption by enabling 2D networks to learn 3D features. Another two methods, called shape constructing and shape mask generator, are to explicitly model shape priors for boosting the segmentation performance, being able to substantially reduce visually implausible segmentation results.

Selective Learning. It is a simple, effective, and generic training framework to alleviate the burden on training data collection by using external data. The key idea is to learn a weight for each external data such that informative external data can have large weights and thus contribute more to the training loss, for implicitly encouraging the network to mine more valuable knowledge from informative external data while suppressing to memorize irrelevant patterns from 'useless' data. It is formulated as a constrained non-linear programming problem, solved by an iterative solution that alternatively implements weights estimating and network updating. It is not limited to particular learning models and loss functions, does not require to compute second-order gradients, and extensive experiments on multi-organ CT segmentation datasets show its efficacy and performance gains against existing methods, capable of substantially alleviating the burden on training data collection.

Adversarial Redrawing. It is a generic unsupervised segmentation method, for alleviating the burden on training annotations collection which requires tedious efforts from domain experts and becomes infeasible or prohibitively expensive with the increasing of data's scale. The underlying assumption of this method is that the imaging process can be modeled by a latent variable with two steps: objects' binary mask generating and objects' intensity value drawing. It hence uses two convolutional neural networks (CNN) to model the mask generating and intensity drawing steps, and trains these two CNNs

by using adversarial learning paradigm, i.e. sampling random vectors from a presumed latent space, passing the sampled vectors through these two CNNs to generate images, and finally using a discriminator to distinguish the generated images from the real images. Once the adversarial training has been done, to produce segmentation results for given images, it first trains an encoder that maps the given images into the presumed latent space and then passes the mapped vectors through the trained mask generating CNN. The mapping encoder is trained in an unsupervised manner by minimizing the discrepancy between the input images and the reconstructed images generated by passing the encoder's output through the trained mask generating and intensity drawing CNNs. We conducted extensive experiments on a publicly available CT dataset, with positive results showing the effectiveness of this method, performing favorably against existing methods for different objects.

Surface Projection. It is a GPU memory-efficient learning technique that enables 2D networks to learn 3D features. We observed that boundary pixels of a 3D object can be represented by a surface parameterized by a 2D variable. It hence learns 3D features by using a 2D network to learn the projection distance between the object's surface and a set of spherical surfaces, for recognizing these boundary pixels. Unlike existing methods, this method is without any information loss, by sampling sufficiently dense spherical surfaces, being able to consider all information in the volumetric data. We used a publicly available dataset to assess this method, and the extensive

experimental results show its effectiveness, considerably outperforming existing methods.

Shape Constructing. It is an efficient method to model shape priors. It leverages shape priors from contour fragments rather than pixels, as fragments provide far more informative geometric information. It shows excellent performance on the overlapping objects segmentation task, being able to segment occluded boundary parts even for those nearly visually indistinguishable. This method starts by cutting the clump's contour, generated by deep models, into fragments. It then groups fragments for locating each object's fragments. For each object, it next estimates object's shape template and then connects grouped fragments to segment the object based on the grouped fragments and the collected shape templates. It is developed as an iterative scheme that alternatively implements fragments grouping, shape template estimation, and fragments connecting, for continually enhancing the representation ability of the modeled shape priors. It is assessed on two datasets, with positive results demonstrating its effectiveness.

Shape Mask Generator. It is a genric and effective method of modeling shape priors. We progressively refine the shape priors by learning them until they can describe most objects' shapes. Unlike existing methods, it is able to model shape priors with strong representation ability, and therefore can substantially reduce visually implausible segmentation results. This method

first uses shape templates to model shape priors and then uses the modeled shape priors to generate shape masks of objects for the segmentation. It next refines the modeled shape priors in the training dataset by minimizing the generating residual, which is smaller when the segmentation results are more accurate. We assess this method on two cervical smear datasets for segmenting overlapping cytoplasms. The empirical evidence from extensive experiments shows that this method works better against existing methods.

1.5 Research Significance

Deep learning networks have been known to be able to model complex patterns, given a sufficient amount of training data and abundant computing capability. However, in medical image segmentation scenarios, collecting training data can be prohibitively expensive, modeling volumetric images may consume unaffordable GPU memory, and clinical knowledge should not be violated, all of which pose different challenges in the usage of deep networks in medical image segmentation tasks, compared to that in other learning tasks. The proposed methods, as shown later, have great potential to alleviate the burden on training data collection, reduce GPU memory consumption, and consider clinical knowledge by leveraging shape priors. They, therefore, are significant to advance deep networks in a wide range of medical image segmentation tasks.

By alleviating the burden on training data collection, it becomes possible to apply deep learning networks to any segmentation tasks of interest, largely reducing the cost of data collection and giving a quick chance to implement the segmentation task at hand. By reducing GPU memory consumption, it becomes possible to process high-resolution volumetric medical data without any information loss, solving the dilemma between computational efficiency and segmentation performance. Finally, by leveraging shape priors, visually implausible segmentation results are considerably reduced, increasing the clinical significance.

1.6 Thesis Structure

This thesis is organized as follows. We first present selective learning in Chapter 2, a training framework that alleviates the burden of collecting training data by learning selectively from external data. In Chapter 3 we describe adversarial redrawing, an unsupervised segmentation method that alleviates the burden of collecting training annotations by developing adversarial learning to medical image segmentation. In Chapter 4 we introduce surface projection, while in Chapters 5 and 6 we present shape constructing and shape mask generator, for reducing GPU memory consumption of learning 3D features and leveraging shape priors to reduce visually implausible segmentation results, respectively. We finally conclude this thesis and discuss future works in Chapter 7. An illustration of the relation

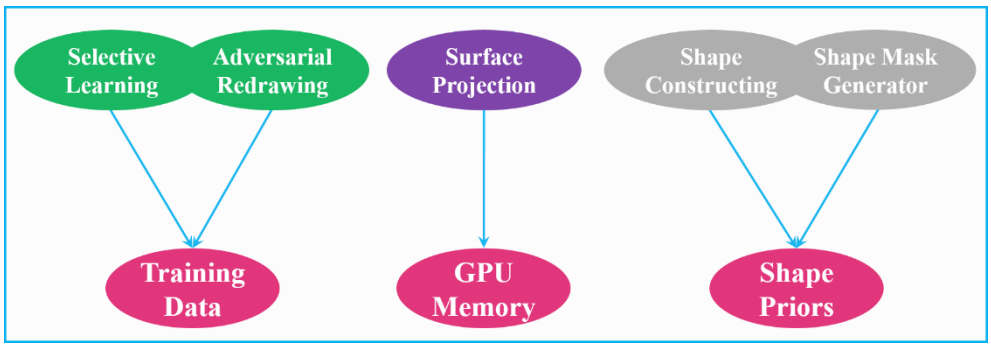


Fig. 1.2 Illustration of the relations among the identified problems and the methods.

among the identified problems and the methods is shown in Fig. 1.2.

Chapter 2

Selective Learning from External Data for Medical Image Segmentation

This chapter presents selective learning, an effective training framework that alleviates the burden on training data collection by resorting to external data. It is especially helpful for applying deep networks to medical image segmentation tasks where collecting a large amount of pixel-wise annotations is expensive or even impractical. The key idea is to learn a weight for each external data, such that good ones can have large weights and then contribute more to the training loss, for implicitly encouraging the network to mine more valuable knowledge from informative external data while suppressing to memorize irrelevant patterns from 'useless' data. It is not limited to particular learning models and loss functions, does not require to compute second-order gradients, and extensive experiments on multi-organ CT segmentation datasets show the efficacy and performance gains against existing methods.

2.1 Problem Background

Deep learning networks have achieved remarkable success on a wide range of medical image segmentation tasks [1-6]. Their success, however, heavily

relies on a very large training dataset, which is not always available. In many tasks, it is time-consuming, costly, and even prohibitively expensive to collect training data, especially in the volumetric medical image (e.g., CT or MRI) segmentation tasks in which besides the image itself, collecting pixel-wise annotations requires tedious efforts from domain experts, with multiple rounds to correct the annotation errors for reaching a consensus [151-160].

To circumvent this difficulty, a promising way is to use external data to train deep networks [9-11]. External data, however, may have a different distribution from the internal data; they may vary hugely in quality, reliability, and relevance, perhaps not being a reliable reflection of the task to be learned [12]. Therefore, it is worthy to study the training mechanism of deep models on external data, for promoting the practical development of deep networks to real-world medical image segmentation tasks, especially to those where external data are much cheaper to acquire

Existing methods include mainly data selection [13-20] and data weighting [21-30]. The key idea of data selection is to select informative data and use them only to train the network. A common way is to select small-loss-data to update the network during training. Methods in this type hence are in favor of learning easy patterns and ignoring numerous informative data with hard patterns that have been known to make deep networks more accurate and robust [31, 32]. Also, how to judge the loss value to be small enough for selecting data remains elusive and is often done heuristically.

Instead of completely ignoring some ‘bad’ data, data weighting-based methods attempt to assign a small weight for them, just making their contributions less to the network updating. Earlier works include mainly importance sampling [21, 22], boosting [23, 24], and hard example mining [25]. These works, however, cannot distinguish informative data with hard patterns from outliers. Recently, Ren *et al*. [29] learn data’ weights by using gradient descent direction on the mini-batch data’ weights. This method, however, requires to compute second-order gradients of deep networks, being computationally expensive.

The proposed selective learning belongs to data weighting-based methods and is formulated as a constrained non-linear optimization problem to allow deep networks to learn selectively from external data. It aims at jointly learning external data’ weights and the network for maximally leveraging their complementary benefits. It also puts a hard constraint, requiring the network to learn better than without using external data. The formulated problem is solved by an iterative solution that alternatively implements weights estimating and network updating. It estimates data weights based on the similarity of loss values, a large weight to be assigned for data with high similarity. It updates the deep network with constraints by using the Lagrange multipliers technique, the Lagrangian variable to be increased when the constraint is violated. It obtains positive results, demonstrating its efficacy, being able to learn selectively from external data.

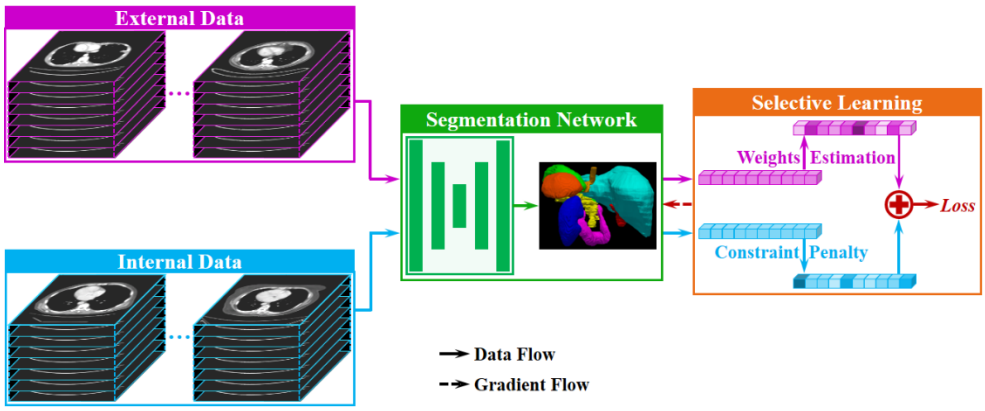


Fig. 2.1 The illustrative pipeline of how we selectively learning from external data: for each external data, we learn a weight to adjust its importance in the training loss, while for internal data, we put a hard constraint to enforce the network learning better than without using external data.

2.2 Methodology

Fig. 2.1 shows an illustrative pipeline of how our selective learning works. For learning selectively, it learns a weight for each external data for adjusting its importance in the training loss while putting a hard constraint on internal data for enforcing the network learning better than without using external data. Details are presented below.

2.2.1 Problem Setup

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input space and output space, respectively. Given a segmentation network with the function space of $\mathcal{F}$, training this network is

to find a function $f \in \mathcal{F}$ that best approximates the unknown target mapping, and is done by solving the following problem

$$\underset{f \in \mathcal{F}}{\operatorname{argmin}} \, \mathbb{E}_{p(x,y)}[\ell(f(x), y)], \quad (x, y) \in \mathcal{X} \times \mathcal{Y}, \tag{2.1}$$

where $\mathbb{E}_{p(x,y)}$ denotes the expectation over the underlying joint density $p(x, y)$ and $\ell: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ denotes the loss function; we here fold in any regularization terms into ℓ for simplicity.

Since the expectation $\mathbb{E}_{p(x,y)}[\ell(f(x), y)]$ is computationally intractable, it is approximated by its empirical counterpart, $\frac{1}{K}\sum_{k \in [K]} \ell(f(x_k), y_k)$, where $[K] \coloneqq \{1, \cdots, K\}$ stands for the index set, with the strong assumption that the training data $\{x_k, y_k\}_{k=1}^{K}$ are independent and identically distributed from $p(x, y)$. Under this assumption, this empirical approximation is unbiased, and so the learned function through the empirical approximation is also unbiased as $K \to \infty$. This strong assumption, however, is very likely to be violated when learning from external data, as external data may have a different distribution from the internal data. In this case, the empirical approximation becomes biased and so does the learned function.

We hence aim at learning a weight w_k for each external training data (x_k, y_k), for correcting the distribution discrepancy that is able to help the network to mine more knowledge from informative external data while suppressing to memorize irrelevant patterns from ‘useless’ or even ‘harmful’ data.

2.2.2 Problem Formulation

We formulate weights learning as a constrained non-linear programming problem. More specifically, let $\{x_n, y_n\}_{n=1}^N$ and $\{x_m, y_m\}_{m=1}^M$be the external and internal data, respectively. We learn weights, along with training the network, by solving the problem below

$$\underset{\substack{f\in\mathcal{F},\\ \mathbf{w}\in[0,1]^N}}{\operatorname{argmin}} \left(\frac{1}{N}\sum_{n=1}^N w_n \ell_n(f) + \frac{1}{M}\sum_{m=1}^M \ell_m(f)\right), \tag{2.2 a}$$

$$\text{s.t. } \sum_{m=1}^M \ell_m(f) \leq \sum_{m=1}^M \ell_m(\hat{f}_M^*), \tag{2.2 b}$$

$$\mathbf{1}^T\mathbf{w} > 0, \tag{2.2 c}$$

where $\mathbf{w} \coloneqq (w_1, \cdots, w_N)^T$, $\ell_i(f) \coloneqq \ell(f(x_i), y_i)$, and $\hat{f}_M^*$ is the optima without using external data; $\hat{f}_M^* \in \underset{f\in\mathcal{F}}{\operatorname{argmin}} \frac{1}{M}\sum_{m\in[M]} \ell_m(f)$. The constraint (2b) is to first prevent overfitting the external data and second guarantee the training performance on internal data; it is intuitively reasonable that learning should be better when using external data. The constraint (2c) is to prevent degenerate solutions of $\mathbf{w}$, e.g., all $w_n = 0$ in which case not all external data are correctly used.

Feasibility Analysis. Our problem, certainly, has a feasible solution. To see this, assuming that we have known the optima of weights, $\mathbf{w}^*$, our problem in that case degenerates into a standard learning problem with the weighted loss for external data; the constraints (2.2 b) and (2.2 c) here are satisfied almost surely. The main concern is whether our problem is well formulated such that its optimum corresponds to the best function that we can learn. We

check this by using reductio ad absurdum. Assume that $\{\hat{f}^*, \widehat{\mathbf{w}}^*\}$ is the optimum and f^* is the best learnable function. If $\hat{f}^*$ behaves worse than f^*, then by altering $\widehat{\mathbf{w}}^*$ our objective function (2.2 a) can have a smaller value by replacing $\hat{f}^*$ with f^*, which means $\{\hat{f}^*, \widehat{\mathbf{w}}^*\}$ is not the optima, contradicted. This result shows that our problem has been well formulated; solving it results in getting the best learnable function.

2.2.3 Optimization

Fig. 2.1 shows an illustrative pipeline of how we solve our problem. We use block coordinate descent, splitting the problem into two subproblems: (1) weights estimating and (2) constrained network updating. We alternatively implement weights estimating where the network is fixed and network updating where the weights are fixed.

Weights Estimating. It is done by solving the problem below

$$\mathbf{w}^{(t+1)} \in \underset{\mathbf{w}\in[0,1]^N}{\operatorname{argmin}} \left(\frac{1}{N}\sum_{n=1}^{N} w_n d(\mathcal{D}_n, \mathcal{D}|f^{(t)}) + \lambda \mathbf{w}^T\mathbf{w}\right),$$

$$\text{s.t. } \sum_{n=1}^{N} w_n = N, \tag{2.3}$$

where t and $d(\mathcal{D}_n, \mathcal{D}|f^{(t)})$ denote the updating step and the discrepancy between the distribution $\mathcal{D}_n$ and $\mathcal{D}$ measured with the network at step t. Here $\mathcal{D}_n$ and $\mathcal{D}$ stand for the distribution of the external data (x_n, y_n) and internal data $\{x_m, y_m\}_{m=1}^{M}$. In addition, $\lambda > 0$ is a hyperparameter to balance two terms.

The first term in (2.3) is small when external data with a small discrepancy value having a large weight. This is expected because external data whose distribution is similar to the internal data intuitively should contribute more to the network updating. The second term has a small value when all external data having similar weights. It hence encourages to learn from as many external data as possible. Furthermore, when the constraint is satisfied, the constraint (2.2 c) is also satisfied. These observations suggest that this subproblem is well formulated.

Finally, we define $d(\mathcal{D}_n, \mathcal{D}|f^{(t)})$ as $\left|\ell_n(f) - \bar{\ell}_{[M]}(f)\right|$, where $\bar{\ell}_{[M]}(f)$ denotes the average of $\ell_m(f)$ on the internal data $\{x_m, y_m\}_{m=1}^{M}$. As expected, we can see that it has a large value when the network performing differently on external data (x_n, y_n) and internal data $\{x_m, y_m\}_{m=1}^{M}$, while a small value when performing similarly.

Network Updating. Given the weights $\mathbf{w}^{(t+1)}$, we update the network by

$$f^{(t+1)} = f^{(t)} - \gamma^{(t)}\nabla\left(\sum_{i=1}^{b} w_i^{(t+1)}\ell_i(f^{(t)}) + \xi^{(t)}\mathcal{C}(f^{(t)})\right), \qquad (2.4)$$

where $\gamma^{(t)}$ and b stand for the learning rate and batch size, respectively. The only difference from the canonical network updating with the weighted loss is that we convert the constraint (2.2 b) to the term $\mathcal{C}(f^{(t)})$ with the penalty coefficient$\xi^{(t)}$. We define$\mathcal{C}(f^{(t)}) = \sum_{i\in[M]} \max(\ell_i(f^{(t)}) - \ell_i(\hat{f}_M^*), 0)$. We can see that if (2.2 b) is satisfied, $\mathcal{C}(f^{(t)}) = 0$, then it has no effect on the

network updating. Otherwise, i.e. violated, $\mathcal{C}(f^{(t)}) > 0$, then by using a large $\xi^{(t)}$ the updating step will be enforced to primarily eliminate the violation.

We set ξ to a very small value at the early stage of training, because at that stage it is normal that the network behaves worse than $\hat{f}_M^*$. Its value is increased when meeting violations. We update it as $\xi^{(t+1)} = \xi^{(t)} + \gamma_\xi^{(t)}\mathcal{C}(f^{(t)})$, where $\gamma_\xi^{(t)}$ stands for the increasing rate. We get this updating rule by using sub-gradients [33], as $\mathcal{C}(f^{(t)})$ is non-differentiable.

2.3 Experimental Evaluation

2.3.1 Experimental Setup

Dataset. The proposed method was assessed on the abdominal multi-organ segmentation task from two CT datasets [34-36]. We term them as BTCV and TCIA for simplicity. BTCV has 47 CT volumes with resolutions of 0.6∼0.9 *mm* (in-plane) and 0.5∼5.0 *mm* (inter-slice), while TCIA has 43 CT volumes with resolutions of 0.6∼0.9 *mm* (in-plane) and 1.5∼2.5 *mm* (inter-slice). Each CT volume has 122.85 slices on average in the BTCV dataset while 238.02 in the TCIA dataset. They provide pixel-wise annotations for 8 abdominal organs: (1) Duodenum, (2) Esophagus,(3) Gallbladder, (4) Liver, (5) Left Kidney, (6) Pancreas, (7) Spleen, and (8)Stomach.

Network Architecture. We used the same architecture as the original 2D U-Net [37], except for five modifications: (1) we replaced batch normalization with instance normalization [38], (2) we replaced ReLU with leaky ReLU [39]

with the slope of 0.01, (3) we implemented down-sampling and up-sampling by the strided and transposed convolutions, (4) we add dropout [40] with the probability of 0.1after each convolutional layer, and (5) we used softmax function to normalize network's output.

Implementation. We clipped the intensity value to [−200, 250], and then normalized it to [0, 255]. We resized the image size of 512×512 to 256×256. We used Dice loss to train the network, employed Adam [41] with the learning rate of 0.0003, and set the batch size to 16. We set λ to 5, initialized ξ as 0.05, and fixed γ_ξ of 0.0001. We ran 40 epochs; in the first 10 epochs, we adopted the standard training for the warm-up purpose, and enforced the hard constraint after 20 epochs. We produced segmentation results by assigning the categorical label with the highest prediction value to pixels.

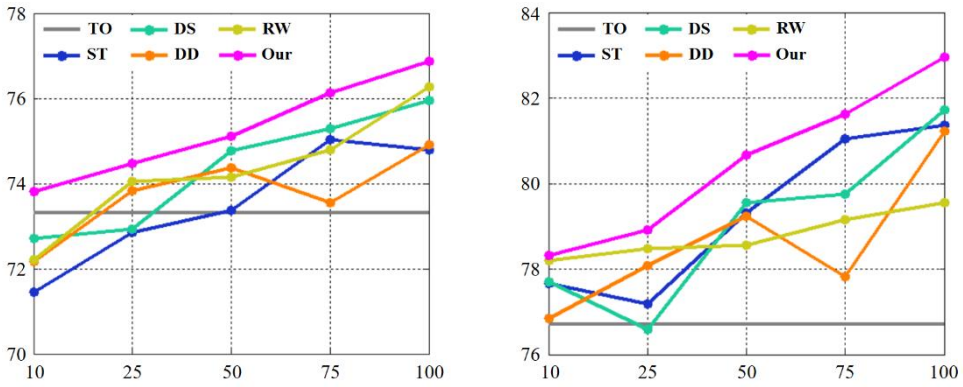
Competitors. We compared the proposed method with three methods, denoted by DS [17], RW [29], and DD [30]. DS belongs to data selection-based methods, while RW and DD belong to data weighting-based methods. DS selects k data with the smallest loss value from the batch at each iteration to update the network; here we set k to 10. RW assigns data weights by minimizing the weighted loss on the validation set; we used 20% of the given internal training data for the validation. DD assigns data weights based on the distribution discrepancy between external data and internal data; we here assumed that all external data are sampled from the same distribution. For fair

comparisons, we used the same experimental setting as ours while their hyperparameters are those recommended by the authors.

2.3.2 Experimental Results

Performance Improvement. We first compared performance improvement by using external data. The performance metric throughout this work is Dice Similarity Coefficient (*DSC*). The result, the mean of organs on the test data, is presented in Fig. 2.2, (a) for BTCV where the external dataset is the TCIA dataset and (b) for TCIA where the external data is the BTCV dataset. We randomly selected 50% for training and the remaining 50% for testing, and experimented with 5 different amounts of external data: 10%, 25%, 50%, 75%, and 100% of the external dataset.

We first look at results on the BTCV dataset, shown in Fig. 2.2 (a). We can see that our method consistently works better than all other methods in all 5 amounts, which demonstrates the efficacy of our method. In addition, using external data without careful treatment cannot ensure performance boosting. We can see that when using 10% of the TCIA dataset only our method boosts, and all methods boost when the external data amount is over 50%; TO and ST in the figure stand for target only, without using external data, and standard training, no treatments on external data, respectively. This evidence suggests that our method can learn selectively, leveraging informative data while suppressing non-informative or ‘harmful’ data.



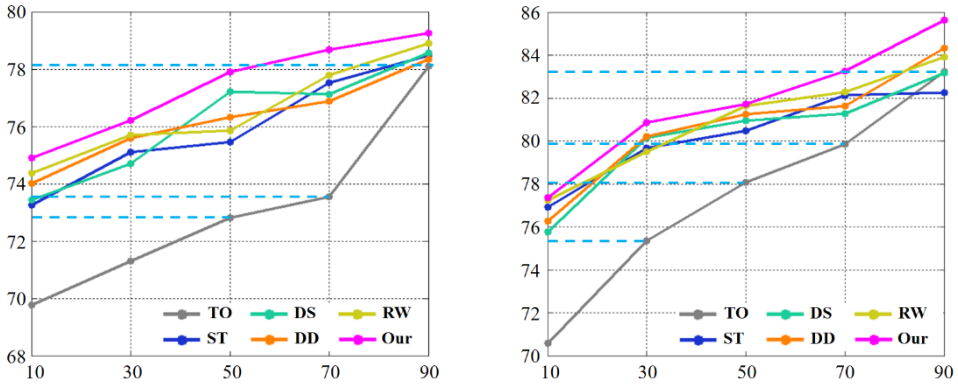
(a) (b)

Fig. 2.2. The test accuracy, the mean *DSC*, with 5 varying amounts (%) of external data used, (a) for BTCV and (b) for TCIA.

Furthermore, using more external data does not necessarily improve the performance for all methods (see ST and DD), which means when learning from external data some data do not help and thus the learning is encouraged to be done selectively.

Similar trends are also found in the TCIA dataset, shown in Fig. 2.2 (b), though the extent is slightly different. The new main finding is that DS, RW, and DD behave differently compared to ST while our method is consistent; they generally work better than ST in BTCV dataset while not here. This may indicate that these methods are dataset-sensitive, using data properties not effective as ours.

Training Data Reduction. We also assessed the ability of these methods to alleviate the amount of training data by using external data. To simulate the performance varying with the increasing of training data, we randomly selected 10% for testing and 5 amounts: 10%, 30%, 50%, 70%, and 90%, for

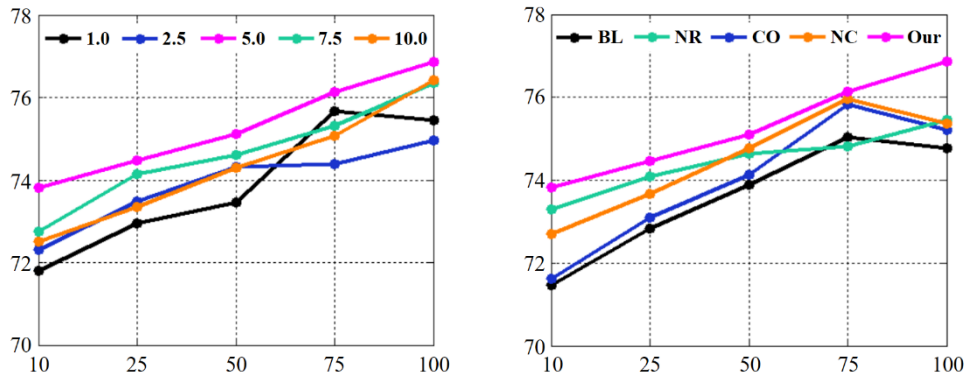


(a) (b)

Fig. 2.3. The test mean *DSC* with 5 varying amounts (%) of training data when using all external data, (a) for BTCV and (b) for TCIA.

training. The external data here were all used for training. The test mean *DSC* is presented in Fig. 2.3, (a) for BTCV and (b) for TCIA.

We can see that our method yields the highest accuracy in both datasets in all 5 cases, which means that our method alleviates the training data most, demonstrating the efficacy of our method again. We also can see that all methods outperform TO, except for ST in the TCIA dataset when using 90% data for training, and have a narrowing performance gap when more training data are given, while our method outperforms most and has the biggest gap. This finding implies that our method is more effective to leverage informative external data and suppress non-informative data. In addition, among all methods, only RW and Our consistently work better than ST (in fact RW slightly works worse than ST in the TCIA dataset when using 30% data for training), while other methods are with mixing results. This evidence suggests that RW and our method are more robust while our method is more accurate.



(a) (b)

Fig. 2.4. Effect of different values of the hyperparameter (a) and different components of our method (b) on the test accuracy (mean *DSC* of organs).

2.3.3 Component Analysis

Hyperparameter Selection. The proposed method receives the hyper-parameter: λ, in Eq. (2.3), a larger value of it resulting in external data to be assigned weights that are more similar. To decide the best value of it, we experimented with 5 values: 1.0, 2.5, 5.0, 7.5, and 10.0, on the BTCV dataset with 50% for training and the remaining 50% for testing (randomly selected). Fig. 2.4 (a) shows the test accuracy when using 5 amounts of external data (10%, 25%, 50%, 75%, and 100% of the TCIA dataset). We can see that when $\lambda = 5$ the accuracy is the highest in all amounts, and we hence set λ to 5 in all other experiments. We also can see that a large value (7.5 or 10.0) generally works better than a small value (1.0 or 2.5).

Ablation study. For investigating the effect of three main components of our method: (1) data weighting, (2) weight regularization, and (3) constrained network updating, we compared our method to four methods, denoted by BL,

NR, CO, NC, standing for baseline, no weight regularization, constraint only, and no constraint, respectively. BL uses the canonical training, all external data having a weight of 1 and no constraint enforced. NR deletes the regularization term, $\mathbf{w}^T\mathbf{w}$, in Eq. (2.3), conducted by setting λ to 0. CO uses the constrained network updating only, done by setting all data weights to 1 in the network updating. NC does not use the constraint in the network updating, implemented by setting ξ^t in Eq. (2.4) to 0.

Experiments were conducted on the BTCV dataset (50% for training and 50% for testing) with five amounts of the TCIA dataset as the external data (10%, 25%, 50%, 75%, and 100%). The test accuracy is presented in Fig. 2.4 (b). We can see that our method works better than all other methods, which suggests all three components are necessary and mutually reinforcing. In addition, CO works better than BL while worse than NC, implying that constrained network updating indeed helps while the key is data weighting.

2.4 Closing Remarks

There is tremendous motivation for training deep networks with as few data as possible. Here we have investigated a simple, generic, and effective method that allows deep networks to selectively learn from external data. It is simple, with the key idea of assigning a weight to external data for learning selectively. It is generic, not limited to particular learning networks and loss functions. It is also effective, yielding positive results on various experimental

scenarios with consistent performance gains over existing methods. This method hence has great potential to alleviate the burden on training data collection.

Chapter 3

Adversarial Redrawing for Unsupervised Medical Image Segmentation

This chapter presents adversarial redrawing, a generic unsupervised segmentation method, for alleviating the burden on training annotations collection which requires tedious efforts from domain experts. The underlying assumption of this method is that the imaging process can be modeled by a latent variable with two steps: (1) objects' binary mask generating and (2) objects' intensity value drawing. Under this assumption, two CNNs of modeling the mask generating and intensity drawing steps can be trained without using any annotations by using the adversarial learning paradigm. We use a publicly available CT dataset to assess this method, and obtains positive results on extensive experiments, showing the effectiveness.

3.1 Problem Background

Medical image segmentation, though is rather challenging, researchers recently have achieved remarkable progress on it by developing deep convolutional neural networks (CNNs) [37, 42-44]. CNNs extract hierarchical and multi-resolution features on their own for most accurately

and effectively evaluating their input, having shown powerful capabilities on modeling complex patterns of medical images [161-170].

CNN's achievement, however, relies on the vast amount of pixel-wise human annotations about the desired segmentation. But collecting such annotations for medical images is not trivial. Instead, it is laborious, time-consuming, and costly, requiring tedious efforts from domain experts. It even becomes infeasible or prohibitively expensive with the increasing of data's scale. CNNs' scalability and usability, therefore, are dramatically restricted for many practical medical image segmentation tasks where available annotations are insufficient [45]. A natural question then arises here: is it possible to train segmentation CNNs using less or even no human annotations?

In the literature, the most frequently used technique is transfer learning [46, 47] that trains CNNs by using other available datasets, but it requires the used datasets having a similar data distribution to the target dataset that is not always satisfied. Domain adaption [48-51] then relaxes this requirement, allowing to train segmentation CNNs by using other domain images, by exploiting features' similarity in different domains. The relaxed requirement, however, is still not ready to be satisfied; a serious mismatch of datasets' distribution often incurs a substantial segmentation performance drop.

Another parallel line of techniques includes mainly weakly supervised [52, 53] and semi-supervised learning [54, 55]. Weakly supervised learning attempts at training CNNs by using object-level annotations (e.g. object's

center), while in the semi-supervised learning setting, only a small fraction of training data has pixel-wise human annotations. Whilst the workload on collecting human annotations is alleviated, they still require domain experts' effort, which motivates the development of unsupervised learning techniques [56-58]. The key idea of most existing unsupervised learning techniques is to design the loss function such that it can express the desired properties of objects to be segmented. These techniques, therefore, face great challenges in modeling objects' properties.

Instead of handcrafted modeling of objects' properties, the proposed adversarial redrawing gradually grasps objects' properties on their own during training. The key idea is to assume that the imaging process can be modeled by a latent variable with two steps: (1) objects' binary mask generating and (2) objects' intensity value drawing, as shown in Fig. 3.1 (a). By doing so, the CNN that models the objects' mask generating can be trained in an unsupervised manner by exploiting the adversarial learning paradigm [59]; see Fig. 3.1 (b). Segmentation then can be implemented by first finding the corresponding latent vector of the given image and then passing the found vector through the trained binary mask generating CNN; see Fig. 3.1 (c).

We concrete our idea by first employing two CNNs to model the mask generating and intensity drawing steps, and train them by first sampling random vectors from a presumed latent space, then passing the sampled vectors through the two CNNs to generate images, and finally using a

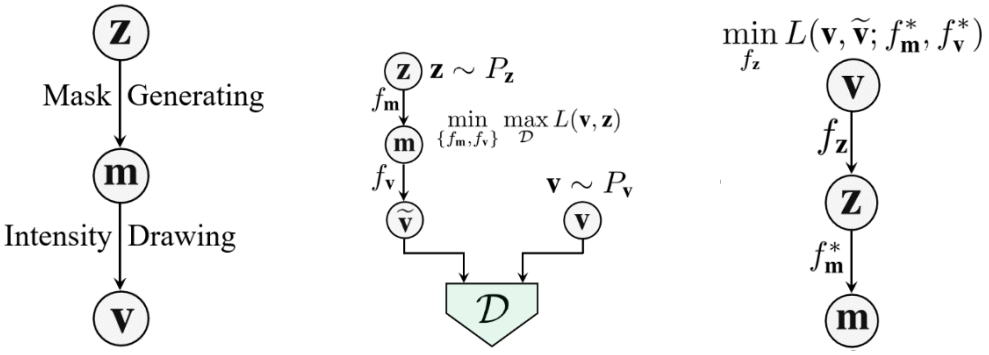


Fig. 3.1 Schema of the idea behind this work. (a) The assumption of the imaging process; images (denoted by **v**) can be generated by a latent variable (**z**) with two steps: objects' binary mask generating and objects' intensity value drawing. (b) The unsupervised training mechanism of CNNs to model the mask generating and intensity drawing steps (f_{m} and f_{v}, respectively). (c) The segmentation result producing process; the given image is first mapped into the latent space by a CNN (f_{z}) which is also trained in an unsupervised manner.

discriminator to distinguish the generated images from the real images. We then employ another CNN to model the mapping function of images into the presumed latent space, and train it by minimizing the discrepancy between the input images and the reconstructed images generated by passing its output through the trained mask generating and intensity drawing CNNs.

3.2 Literature Review

Unsupervised deep learning receives more and more researchers' attention on medical image segmentation tasks and also other learning tasks, especially classification tasks [60, 61]. It indeed deserves, because its supervised counterparts have already shown a powerful capacity to handle these tasks.

Recent advances of unsupervised deep learning techniques include mainly clustering [45, 58, 61], sample specificity analysis [60, 62], and self-supervised learning [63–65]. The key of clustering is to identify clusters such that each of them can represent an underlying concept. The main focus currently is on jointly optimizing clustering and representation learning for maximally leveraging their complementary benefits. However, it is rather challenging to identify representative clusters due to the enormous combinatorial space. Sample specificity analysis therefore avoids identifying clusters by simply treating each sampled data as an independent cluster. It assumes that the underlying visual correlations among classes can be automatically revealed by deep networks via an end-to-end optimization. It is, however, likely to produce more ambiguous class structures than clustering. Self-supervised learning, its key is to add an extra auxiliary network to provide supervision. Existing methods are varying hugely in the sense of how to design the auxiliary network. An important principle in such a design is to consider intrinsical information that is also available in the unlabelled data, e.g., spatial context and spatio-temporal continuity. The auxiliary network design currently is handcrafted and remains an open problem, however.

These advanced techniques, however, are not suitable for medical image segmentation tasks, because most of them are tailored for classification tasks. Although classification techniques technically can be directly applied to segmentation tasks by classifying every pixel one by one, they cannot

consider correlations among pixels which are essential for medical image segmentation tasks. In other words, classification techniques classify each pixel independently, but in the medical image segmentation scenario pixels are required to be jointly classified.

We hence do not pay our effort to extend these advanced techniques. Instead, we develop adversarial redrawing. Our idea is in spirit similar to layered scene decomposition [66, 67] which assumes that an image can be represented by a segmented depth-map with piecewise planar or b-spline surfaces. However, layered scene decomposition aims at inferring occluded geometry of the scene while our adversarial redrawing is for unsupervised medical image segmentation. Two works that are most closely related to ours are [68, 69], both exploiting adversarial learning paradigm to provide supervision for unsupervised natural image segmentation. However, [68] faces model collapse issue, especially at the early stage, while [69] does not consider anatomical constraint; to alleviate model collapse, it shifts segmentation masks, which is not allowed in medical image segmentation tasks (for example, kidney cannot be shifted above on liver).

3.3 Methodology

Given an image with the size of $w \times l \times h$, we denote K as the number of objects to be segmented. The segmentation is, in principle, to find a function $f: \mathbb{R}^{w \times l \times h} \rightarrow \{0,1,2,\cdots,K\}$, where the categorical label 0 is for assigning

pixels to the background, such that pixels assigned to the same label represent the corresponding object. In the CNN context, we approximate such a function by first specifying a function space via designing the CNN's architecture and then selecting a function from the space as the approximation by optimizing the value of the CNN's parameters. The optimization criterion is to make the selected function produce segmentation results that match human annotations best in the training set.

The best segmentation function f can be theoretically found by CNNs if (1) the architecture is designed appropriately such that f is in the specified function space, (2) the employed optimization algorithm is able to find the global optimum f, and (3) available human annotations are sufficient such that the 'matching extent' measured on them is able to reflect the real matching extent. However, in the unsupervised medical image segmentation scenarios, we have not any human annotations. The matching extent then cannot be evaluated by using the canonical CNN training workflow. We hence have no guidance to find f even though a good architecture and optimization algorithm have been already available.

The key in developing unsupervised medical image segmentation algorithms, therefore, is to accurately evaluate the matching extent, i.e., the segmentation quality, with no human annotations involved. To fill this knowledge gap, we develop the adversarial redrawing that bridges image segmentation and adversarial learning so that we can use the adversarial loss

to evaluate the segmentation quality. The key idea is to model the image generation process as two steps: (1) objects' binary mask generating and (2) objects' intensity value drawing. By doing so, in order to generate images that cannot be distinguished by the discriminator from the real images, the adversarial training has to drive both the mask generating network and the intensity value drawing network towards more accurate segmentation.

3.3.1 Problem Formulation

Fig. 3.2 shows an illustrative pipeline of the training process of our adversarial redrawing that employs three CNNs. Two CNNs, denoted by 'Mask Generating' and 'Intensity Drawing', are used to model the image generation process, and trained in the first phase by using the adversarial training workflow. It then trains the 'Image Mapping' CNN that aims at mapping images into the latent space, from which random vectors are sampled for generating images in the phase 1, in an unsupervised manner like auto-encoder, while the decoder, the trained 'Mask Generating' and 'Intensity Drawing' CNNs, is frozen to update during the training. Once the training is done, segmentation is implemented by first passing the given image through the trained 'Image Mapping' CNN and then passing the mapped vector through the trained 'Mask Generating' CNN. For simplicity, in the text below, we shall use f_{m}, f_{v}, and f_{z} to denote the 'Mask Generating', 'Intensity Drawing', and 'Image Mapping' CNNs, respectively.

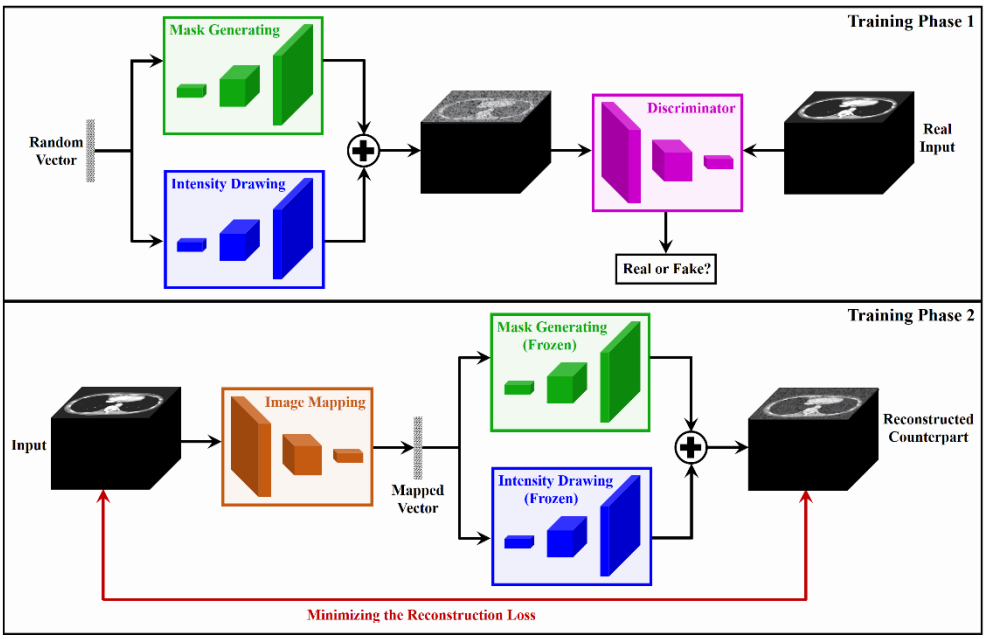


Fig. 3.2 The illustrative pipeline of how our adversarial redrawing to train a segmentation CNN in an unsupervised manner. The training includes two phases; the phase 1 is to train the 'Mask Generating' and 'Intensity Drawing' CNNs by using the adversarial training paradigm while the phase 2 is to train the 'Image Mapping' CNN that maps images into the latent space by minimizing the reconstruction loss. Note that once the training is done segmentation results are produced by first passing the given image through the trained 'Image Mapping' CNN and then passing the mapped latent vector through the trained 'Mask Generating' CNN.

Unsupervised Training of f_{m} and f_{v}. 'Mask Generating' CNN (f_{m}) and 'Intensity Drawing' CNN (f_{v}) have the same architecture. They both take the random vector $\mathbf{z}\sim p(\mathbf{z})$ as the input, and output $K+1$ maps, each of which has the same size as the real images; recall that K is the number of objects to be segmented. To generate images, we first implement an element-wise multiplication on these two CNNs' outputs, and then aggregate all maps' multiplication results by implementing an element-wise summation, i.e., $\tilde{\mathrm{v}} =$

$\sum_{k=0}^{K} f_{\mathrm{m}}^{k} \otimes f_{\mathrm{v}}^{k}$, where f_{m}^{k} and f_{v}^{k} denote the *k*-th map of the 'Mask Generating' and 'Intensity Drawing' CNNs' output, respectively.

For training f_{m} and f_{v} in an unsupervised manner, we employ a discriminator for distinguishing the generated images from the real training images, and finally implement the training by

$$\min_{\{f_{\mathrm{m}}, f_{\mathrm{v}}\}} \max_{\mathbf{D}} (\mathbb{E}_{\mathrm{v}}[\log \mathbf{D}(\mathrm{v})] + \mathbb{E}_{\mathrm{z}}[\log(1 - \mathbf{D}(\tilde{\mathrm{v}}))]), \tag{3.1}$$

where **D** stands for the discriminator, and the symbol $\mathbb{E}_{\mathrm{x}}$ represents the expectation on the variable x. Note that in Eq. 3.1 we directly use the symbol $\tilde{\mathrm{v}}$ to denote the generated images from z to avoid the equation being too cumbersome; formally $\tilde{\mathrm{v}} = \sum_{k=0}^{K} f_{\mathrm{m}}^{k}(\mathrm{z}) \otimes f_{\mathrm{v}}^{k}(\mathrm{z})$.

The essence of the training, i.e., solving Eq. 3.1, can be simply understood as a process that attempts to find a better parameters' value of f_{m} and f_{v} such that the generated images cannot be distinguished by the discriminator **D** from the real images while at the same time to find a better parameters' value of **D** in order to distinguish all generated images. The training stops when the two sides get a balance, i.e., when Nash Equilibrium is reached.

Unsupervised Training of f_{z}. Since the 'Mask Generating' CNN takes the latent vector as the input, for producing segmentation results, we have to map the given image into the latent space for getting the corresponding latent vector. As mentioned above, we employ a CNN, f_{z}, to model this mapping function. The question now is how to train f_{z} in an unsupervised manner.

Intuitively, if the trained 'Mask Generating' CNN can produce the true objects' binary mask via the mapped latent vectors, then by these vectors the trained 'Mask Generating' and 'Intensity Drawing' CNNs should be able to generate the same images as the given images. It implies that f_{z} can be trained by reducing the discrepancy between the input images and the images generated by passing f_{z}'s output through the trained 'Mask Generating' and 'Intensity Drawing' CNNs. This purpose then can be fulfilled by solving the following learning problem

$$\min_{f_{\mathrm{z}}} \mathbb{E}_{\mathrm{v}} \left[L(\mathrm{v}, \tilde{\mathrm{v}} | f_{\mathrm{m}}^*, f_{\mathrm{v}}^*) \right], \tag{3.2}$$

where L is a metric function to measure the discrepancy between the input images (v) and the generated images ($\tilde{\mathrm{v}}$) by the trained 'Mask Generating' and 'Intensity Drawing' CNNs (f_{m}^* and f_{v}^*, respectively); here we employ the Euclidean norm as the metric function L.

Image Segmentation. Once the training is done, we produce segmentation results for the given images as follows. We first pass the given image through the trained 'Image Mapping' CNN (f_{z}^*) to get the latent vector, i.e., $\mathrm{z}_i = f_{\mathrm{z}}^*(\mathrm{v}_i)$. We then pass the mapped latent vector (z_i) through the trained 'Mask Generating' CNN to produce objects' binary mask; $\mathrm{m}_i = f_{\mathrm{m}}^*(\mathrm{z}_i)$.

Remarks. For unsupervised segmentation methods, unlike supervised methods, we do not know which maps correspond to which objects. They essentially just split the given images into $K + 1$ regions. In such a case, one may wonder why the 'Mask Generating' CNN can produce a meaningful

segmentation? In fact, the segmentation performance cannot be strictly guaranteed, as analyzed later. By Eq. 3.1, in theory, we can just ensure the generated images to be indistinguishable by the discriminator from the real images. However, in practice, since objects segmented accurately are much easier to draw than those with inaccurate segmentations, Eq. 3.1 can help to enforce the 'Mask Generating' CNN to segment objects as accurately as possible, for easing difficulties in the training of the 'Intensity Drawing' CNN; this is mainly because the intensity patterns of accurately segmented objects are much easier to model.

3.3.2 Regularized Adversarial Redrawing

The plain adversarial redrawing, as mentioned above, cannot strictly guarantee the segmentation performance. There exist three cases of the segmentation results: (1) a meaningful and reasonable segmentation, (2) an arbitrary segmentation, i.e. randomly splitting the given image into $K + 1$ segments, and (3) an empty segmentation, i.e. splitting the given images into no more than K segments. We hence further improve our plain adversarial redrawing by considering two segmentation constraints: non-arbitrary and non-empty, in a form of regularization terms, for excluding the two types of segmentation errors. The improved regularized adversarial redrawing therefore will drive the model towards more meaningful and reasonable segmentation results.

Non-arbitrary Constraint. We revise the 'Intensity Drawing' CNN such that each object (including the background) has an independent intensity drawing CNN and just one of them is updated in each iteration during training. It is done by first revising the output layer of the original 'Intensity Drawing' CNN (it now just outputs one map, no longer $K + 1$ maps), and then using $K + 1$ such revised 'Intensity Drawing' CNNs to draw $K + 1$ objects. By doing so, in order to continually optimize the adversarial loss, there are just two possible cases: (1) improving the corresponding 'Intensity Drawing' CNN and (2) improving the 'Mask Generating' CNN. Therefore, during training, when updating the corresponding 'Intensity Drawing' CNN no longer helps, the only way is to improve the 'Mask Generating' CNN until optimal segmentation results are produced.

Non-empty Constraint. We design a regularization term $\|\mathbf{R}(\tilde{\mathrm{v}}) - \mathrm{z}_k\|_2^2$ into the adversarial loss. The regressor **R** aims at regressing the randomly sampled latent vector z_k from the generated images $\tilde{\mathrm{v}}$. By doing so, if the object drawn at the current iteration is segmented to be empty, then the generated image $\tilde{\mathrm{v}}$ cannot reflect any information about the current latent vector z_k, and so does $\mathbf{R}(\tilde{\mathrm{v}})$, hence with a very high probability to yield a large value of $\|\mathbf{R}(\tilde{\mathrm{v}}) - \mathrm{z}_k\|_2^2$. Therefore, minimizing such a regularization term during training will be helpful to hinder the 'Mask Generating' CNN from producing empty segmentation results.

The Regularized Adversarial Loss Function. Our adversarial learning problem now can be defined as:

$$\min_{\{f_{\mathrm{m}}, f_{\mathrm{v}}, \mathbf{R}\}} \mathbb{E}_{k,\mathrm{z}_k}[\log \mathbf{D}(\tilde{\mathrm{v}}|k,\mathrm{z}_k) - \alpha \|\mathbf{R}(\tilde{\mathrm{v}}|k,\mathrm{z}_k) - \mathrm{z}_k\|_2^2],$$

$$\max_{\mathbf{D}} \big(\mathbb{E}_{\mathrm{v}}[\log \mathbf{D}(\mathrm{v})] + \mathbb{E}_{k,\mathrm{z}_k}[\log 1 - \mathbf{D}(\tilde{\mathrm{v}}|k,\mathrm{z}_k)]\big), \tag{3.3}$$

where all $K+1$ 'Intensity Drawing' CNNs are collectively denoted by f_{v}, and k is the index of them (randomly sampled in each iteration). In addition, α here is a hyperparameter to balance the importance of the two regularization terms.

3.4 Experimental Evaluation

3.4.1 Experimental Setup

Dataset. We conduct experiments on a famous publicly available CT dataset[1] [34-36]. This dataset has 90 CT volumes; in particular, 43 volumes are collected from the TCIA Pancreas-CT dataset while 47 are collected from the BTCV Abdomen dataset. This dataset provides pixel-wise human annotations for 8 organs: (1) duodenum, (2) esophagus, (3) gallbladder, (4) liver, (5) left kidney, (6) pancreas, (7) spleen, and (8) stomach.

Evaluation Metric. We employ the most widely used segmentation performance metric: Dice similarity coefficient (*DSC*). *DSC* evaluates segmentation performance by measuring the matching extent between human

[1] Available on https://zenodo.org/record/1169361\#.XSFOm-gzYuU

annotations and segmentation results, using the ratio of pixels' number in their intersection against the average in them. Its value hence ranges in [0, 1], with a larger value representing a better segmentation. We here compute *DSC* to four decimal places and report its percentage counterpart.

Network Architecture. Our 'Intensity Drawing' CNN and discriminator follow DCGAN's generator and discriminator [70]. Our 'Mask Generating' CNN has the same architecture as our 'Intensity Drawing' CNN, plus a 'softmax' layer. Our 'Image Mapping' CNN and the regressor have the same architecture as our discriminator, except for the last layer (Sigmod activation is replaced by Tanh activation, and the output dimension is set to the latent vector's size).

Network Initialization and Optimization. The network is initialized with values sampled from a normal distribution $\mathcal{N}(0, 0.02)$ and we chose Adam [41] as the optimizer to train the network. The learning rate for the 'Intensity Drawing' CNN, the discriminator, and the 'Image Mapping' CNN is set to 0.0003 while for the 'Mask Generating' CNN and the regressor to 0.0001, all with the beats value (0.5, 0.999). We update the discriminator once at an iteration while three times for the 'Intensity Drawing' CNN, the 'Mask Generating' CNN, and the regressor. Optimization stops after 25000 iterations for adversarial training (phase 1) while 10000 iterations for image mapping training (phase 2), with the batch size as 64.

Parameters Setting. The latent space is set to Gaussian space with the mean as 0 and standard deviation as 1, and the vector's size is set to 128. The output of the 'Image Mapping' CNN and the regressor is hence multiplied by a factor of 5, as it originally ranges in (-1, 1) and does not match with our latent vector's value. The balance parameter α in Eq. 3.3 is set to 3/128. In the inference stage, to produce the binary mask, we set the threshold value of the 'Mask Generating' CNN's output as 0.6.

3.4.2 Comparison with Existing Unsupervised Methods

Competitors. Four competitive unsupervised methods are compared, denoted by: (1) W-Net [56], (2) BP [57], (3) IIC [58], and (4) AR [71]. W-Net employs two CNNs, one for segmentation and the other for reconstruction from segmentation masks, and the two CNNs are jointly trained by making reconstructed images similar to input images. BP first computes superpixels, and then trains the CNN by making pixels in the same superpixel have the same prediction. IIC is, in principle, a clustering technique, and hence it implements segmentation in a patch-based manner; in order to segment the pixel at the center of the patch, IIC passes the patch and its randomly perturbed version through a CNN, which is trained by maximizing the mutual information between two patches' outputs. AR is the method presented in our conference version.

Implementation. For a fair comparison, in all four methods, the CNN has the architecture as our 'Image Mapping' CNN plus 'Mask Generating' CNN; one for encoder and another for decoder. The inputs are obtained as follows. We first find the smallest box containing the object to be segmented according to the given annotation information. Then, for W-Net, BP, AR, and our method, we sample the boxed region and resize it into 64×64, while for IIC we first randomly sample the centered pixel in the boxed region and then cut the patch with the size of 64×64, as the input. All methods are with the same initialization and optimization setting and use the same threshold (0.6) in the inference stage.

Results. Table 3.1 compares the segmentation results on the BTCV dataset while Table 3.2 on the TCIA dataset. It is observed that our method works best for all objects, except for the standard deviation results of duodenum on the BTCV dataset and pancreas on both datasets. The performance improvement is often by a large margin over all competitors. These experimental results indicate that our adversarial redrawing works better than all competitors and works for different objects.

3.4.3 Statistical Bias Analysis

The above experiments take the boxed region as the input, so there exists statistical bias because the object occupies a large region in the input. We here define the bias as the area ratio of the object against the input, and investigate

Table 3.1 Comparison results of our adversarial redrawing with unsupervised methods on the BTCV dataset.

	W-Net	BP	IIC	AR	Ours
Duodenum	70.64±10.78	72.51±**9.43**	74.91±10.90	76.79±9.71	**79.59**±10.01
Esophagus	73.82±9.86	77.91±8.92	81.07±9.07	84.42±8.74	**86.45**±**8.51**
Gallbladder	73.47±9.71	76.29±9.11	78.69±9.57	78.91±9.26	**84.67**±**8.51**
Liver	70.83±9.62	72.15±9.84	75.17±9.21	78.75±8.47	**81.07**±**7.82**
L-Kidney	74.27±7.43	77.94±8.01	76.43±7.14	79.31±7.63	**87.59**±**6.11**
Pancreas	64.39±10.88	67.23±11.74	68.94±11.69	71.80±**10.82**	**75.60**±11.87
Spleen	69.82±8.26	72.11±7.44	75.07±8.01	77.70±7.89	**81.96**±**6.22**
Stomach	69.34±7.44	73.92±7.69	75.83±6.92	78.23±7.13	**83.45**±**6.64**

Table 3.2 Comparison results of our adversarial redrawing with unsupervised methods on the TCIA dataset.

	W-Net	BP	IIC	AR	Ours
Duodenum	69.29±10.62	72.83±9.98	75.03±11.14	76.17±10.07	**80.62**±**9.46**
Esophagus	71.34±10.12	74.91±9.28	80.12±9.42	84.30±9.17	**85.94**±**8.32**
Gallbladder	71.49±9.67	75.84±8.92	79.71±9.31	79.71±9.07	**83.95**±**8.63**
Liver	68.34±9.23	71.27±10.17	77.60±9.77	77.60±9.12	**78.95**±**8.93**
L-Kidney	71.25±6.03	74.62±6.96	79.72±5.73	79.72±6.84	**84.63**±**4.90**
Pancreas	66.09±11.79	67.43±12.13	72.18±11.28	72.18±**10.76**	**76.60**±11.70
Spleen	68.99±9.76	71.47±8.24	76.38±8.51	76.38±8.39	**79.83**±**7.83**
Stomach	69.89±8.26	71.26±8.46	78.25±7.93	78.25±8.01	**81.20**±**7.43**

its effect on our method's performance by conducting three sets of experiments with different box sizes, denoted by Boxed0, Boxed5, and Boxed10, respectively. Boxed0 is the same as above, while Boxed5 and Boxed10 set the box size as $\left(l + 2\frac{5b}{100}\right) \times \left(w + 2\frac{5b}{100}\right)$ and $\left(l + 2\frac{10b}{100}\right) \times$

Table 3.3 Segmentation results of our method under different box sizes on the BTCV dataset.

	Boxed0		Boxed5		Boxed10	
	Bias	Ours	Bias	Ours	Bias	Ours
Duodenum	61.29	79.59	54.95	75.09	47.20	74.23
Esophagus	70.48	86.45	66.97	84.34	55.54	77.45
Gallbladder	66.83	84.67	60.76	80.74	51.45	76.67
Liver	58.72	81.07	51.67	75.59	44.36	73.35
L-Kidney	70.04	87.59	61.01	78.53	51.65	84.18
Pancreas	55.13	75.60	49.76	71.06	42.48	66.24
Spleen	58.34	81.96	51.41	79.81	44.03	73.43
Stomach	68.10	83.45	59.60	78.76	80.94	73.10

Table 3.4 Segmentation results of our method under different box sizes on the TCIA dataset.

	Boxed0		Boxed5		Boxed10	
	Bias	Ours	Bias	Ours	Bias	Ours
Duodenum	61.36	80.62	55.03	76.99	47.32	71.16
Esophagus	68.38	85.94	65.08	83.86	54.13	76.33
Gallbladder	66.46	83.95	60.15	76.60	50.73	71.89
Liver	59.68	78.95	52.56	74.55	45.11	73.97
L-Kidney	70.39	84.63	61.22	78.28	51.66	72.07
Pancreas	54.34	76.60	49.03	74.02	41.95	64.21
Spleen	57.30	79.83	50.46	74.50	43.33	74.83
Stomach	67.04	81.20	58.56	77.88	49.98	72.19

$(w + 2\frac{10b}{100})$, where l and w stand for the length and the width of the smallest box and $b = \min(l, w)$.

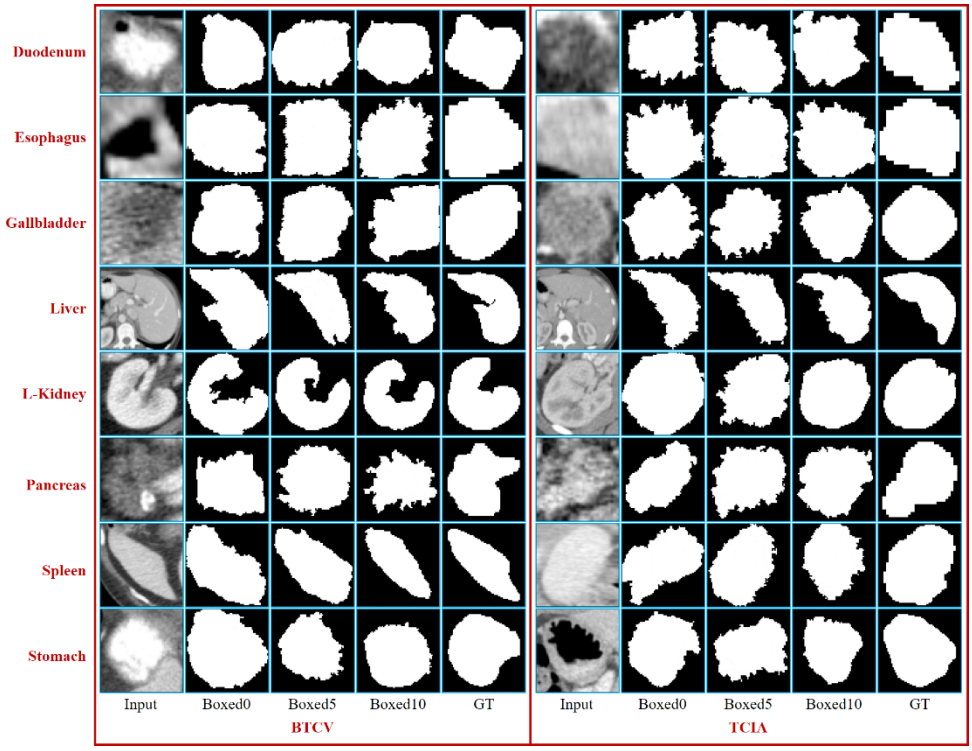


Fig. 3.3 Visual examples of our method on different box sizes; the input and GT come from the Boxed10.

Quantitative Results. Table 3.3 presents the results for the BTCV dataset while Table 3.4 for the TCIA dataset, respectively. We can see that with the increasing of the box size, the bias decreases and also our performance drops. However, the bias decreases far sharply than the performance, and the gap between the bias and our performance remains similar in general, which implies that our method is independent of the box size.

Qualitative Results. Fig. 3.3 shows visual examples of our method on different box sizes. We can see from it that our method does not produce the same results when the box size is different, and does not favor which box size. This experimental observation suggests that box size can affect the

segmentation result but will not affect the segmentation performance significantly.

3.4.4 Model Bias Analysis

Since we implement segmentation in the adversarial learning paradigm, most advanced segmentation architectures are not ready to be used due to the optimization difficulty in the adversarial training, which brings model bias, i.e., some segmentation errors are caused by the model's capacity rather than the method itself. We here investigate this bias by studying our model's performance in four different learning settings.

Fully and Weakly Supervised Settings. Our segmentation model is comprised of the 'Image Mapping' and 'Mask Generating' CNNs; other parts essentially are just to provide supervision. In the weakly supervised setting, we train it using object's center [52]. In both settings, we employ Dice loss and use the same initialization and optimization setup as default. Results are presented in Fig. 3.4 from which we can see that the performance in the weakly supervised setting is slightly better than that in the unsupervised setting and the performance gap between fully supervised and unsupervised settings is no more than 10%. This observation suggests that once advanced segmentation architectures can be used our method is very likely to yield a segmentation performance close to that of its weakly supervised counterpart and about 90% of its fully supervised counterpart.

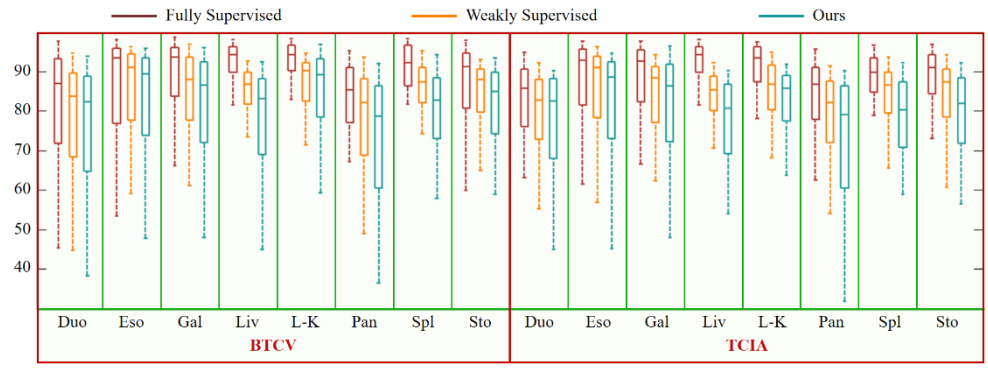


Fig. 3.4 Segmentation performance comparison of our segmentation model on fully supervised, weakly supervised, and unsupervised settings.

Semi-supervised and Transfer Learning Settings. We selected (randomly) some amount of data with annotations from the dataset (representing 10%, 30%, 50%, 70%, and 90%). In the semi-supervised setting, we first train a teacher model using the selected data in a fully supervised manner, and next train the student model in the dataset with the supervision as teacher model's prediction [55]. In the transfer learning setting, the model is first trained in another dataset, and then fine-tune it using the selected data. We compare them with two variations of our method, denoted by 'Our+Fully' and 'Our+Transfer'. 'Our+Fully' first uses the selected data to train the segmentation model in a fully supervised manner, and then uses the trained 'Mask Generating' CNN to initialize our 'Mask Generating' CNN. Likewise, 'Our+Transfer' uses the 'Mask Generating' CNN trained by transfer learning to initialize our 'Mask Generating' CNN. Results are presented in Fig. 3.5 from which we can see that under the same amount of supervision our method works better than its transfer learning counterpart when the supervision's

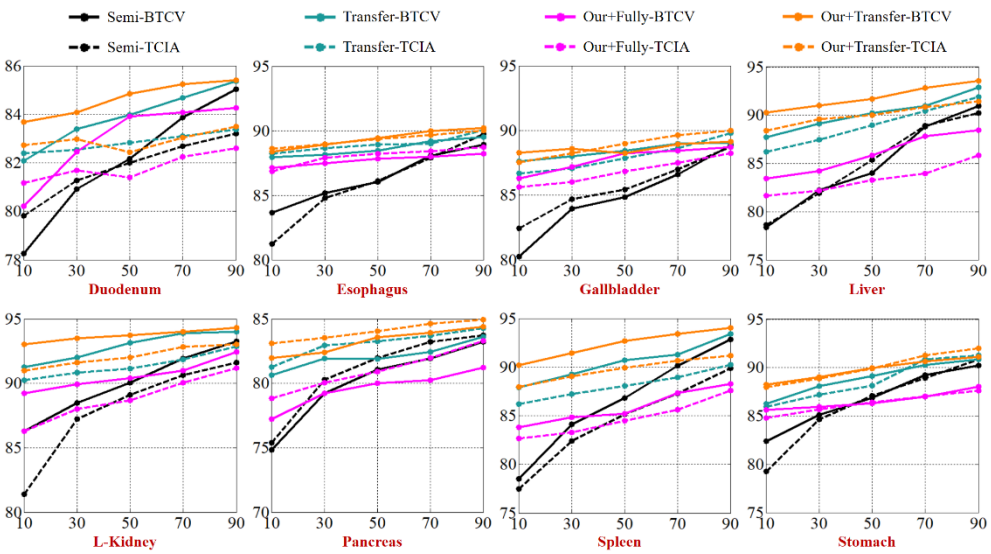


Fig. 3.5 Segmentation performance comparison of our segmentation model on semi-supervised and transfer learning settings.

proportion is less than 70% and its semi-supervised counterpart when the supervision's proportion is less than 30%. This finding indicates that our method has great potential to outperform its semi-supervised and transfer learning counterparts in practical medical image segmentation tasks where available annotations just account for a very small proportion, far less than 30%.

3.4.5 Component Analysis

We improve the plain adversarial redrawing by developing two constraints to avoid unreasonable results. We here evaluate each constraint's effectiveness in performance improvement. Results are presented in Table 3.5 and Table 3.6, where 'Plain', 'Non-arbitrary', 'Non-empty' denote the plain adversarial

Table 3.5 Component analysis results on the BTCV dataset.

	Plain	Non-arbitrary	Non-empty	Ours
Duodenum	74.58	75.22	75.27	**79.59**
Esophagus	80.64	82.43	81.72	**86.45**
Gallbladder	78.15	80.43	79.36	**84.67**
Liver	74.25	76.75	76.08	**81.07**
L-Kidney	80.68	82.72	81.94	**87.59**
Pancreas	68.27	71.75	70.90	**75.60**
Spleen	74.37	76.22	76.37	**81.96**
Stomach	75.24	77.89	78.28	**83.45**

Table 3.6 Component analysis results on the TCIA dataset.

	Plain	Non-arbitrary	Non-empty	Ours
Duodenum	74.90	76.92	75.60	**80.62**
Esophagus	77.64	80.30	79.21	**85.94**
Gallbladder	78.16	79.43	79.17	**83.95**
Liver	73.08	74.64	74.28	**78.95**
L-Kidney	77.79	81.22	80.18	**84.63**
Pancreas	69.56	72.64	71.02	**76.60**
Spleen	73.67	76.51	75.21	**79.83**
Stomach	74.36	77.83	76.22	**81.20**

redrawing model, the model after adding the non-arbitrary constraint, and the model after adding the non-empty constraint, respectively. We can see that the ‘Non-arbitrary’ and ‘Non-empty’ consistently work better than the ‘Plain’, which indicates that both constraints are able to improve the performance. It is also observed that ‘Ours’ works best for all objects, suggesting that the two constraints are necessary and mutually reinforcing.

3.5 Closing Remarks

In this chapter, we presented a generic unsupervised image segmentation method. This method is developed by associating image segmentation with adversarial learning such that we can use the adversarial loss as the supervision for training the segmentation CNN. We assessed its performance on a publicly available CT dataset, and the rich experimental results show that this method produced more accurate results compared to existing unsupervised image segmentation methods for different objects. The proposed method is hence possible to handle other segmentation tasks where human annotations are expensive to collect, and has impacts on advancing unsupervised image segmentation tasks and revealing new knowledge about designing unsupervised training frameworks.

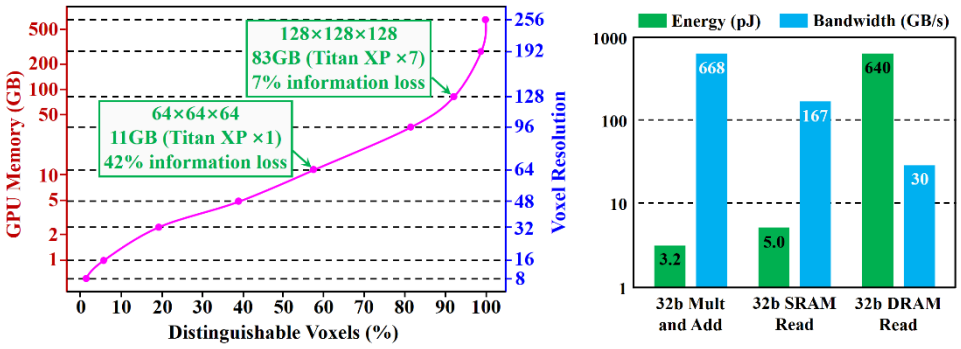
Chapter 4

Surface Projection: Learning 3D Features with 2D CNNs for Segmenting Volumetric Medical Images

This chapter is about surface projection, a GPU-memory efficient learning method that enables 2D networks to learn 3D features, and it hence can substantially reduce the GPU memory consumption in volumetric medical image segmentation tasks where 3D features usually are desired. Unlike existing methods, this method can consider all information in data, without any information loss. We observed that boundary pixels of a 3D object make up a surface which is possible to be expressed by a 2D variable. It, therefore, learns 3D features by organizing the object's surface into a 2D plane and then using a 2D network to predict the distance from the object's surface to the sampled spherical surfaces. Its performance is evaluated in a publicly available CT dataset, with extensive experiments showing its effectiveness.

4.1 Problem Background

Volumetric images, say computed tomography (CT) and magnetic resonance imaging (MRI), are one of the main data forms [171-180]. For segmenting such types of images, 3D features are desired to ensure the performance.



(a) (b)

Fig. 4.1 Illustration of expensive GPU consumption of 3D CNNs: (a) memory and (b) footprint; figures are about U-Net with the batch size of 16.

However, it consumes expensive GPU memory for 3D deep networks to learn 3D features [72, 73]. For example, 3D networks consume GPU memory cubically with the increasing of pixel's resolution, as shown in Fig. 4.1 (a). This leads 3D networks to be memory-prohibitive for learning from high-resolution volumetric data. However, learning from low-resolution data results in information loss, thus unable to distinguish some pixels, which can substantially degrade the segmentation performance. Moreover, 3D networks consume large memory footprints; see Fig. 4.1 (b). This leads to ineffective learning, as memory operations are more costly than arithmetic operations that are for forming representative features.

In order to meet the GPU memory limitation, advanced techniques attempt at learning 3D features by using 2D networks and at the same time try to consider more information. They include mainly three categories: 2D slice distillation [74-77], 2.5D [78-81], and 2D multiple views [82-84]. 2D slice distillation-based methods combine 2D networks and often a conditional

random field algorithm (CRF) [85, 86] or recurrent neural network (RNN) [87] to learn 3D features. Methods in the second category feed several neighboring 2D slices into a 2D network to learn 3D features. 2D multiple views-based methods use several 2D networks that learn from different views, i.e. axial, coronal, and sagittal, to learn 3D features. These existing methods, yet work better than plain 2D networks as they consider more information, remain unable to learn from the whole data, resulting in performance degradation.

We hence designed a generic method for learning 3D features that uses 2D networks to segment volumetric medical images. We observed that boundary pixels of a 3D object make up exactly a 2D surface, and hence they are possible to be perfectly recognized by a 2D CNN in theory. We hence employ a 2D network to learn the distance from the object's surface to some sampled spherical surfaces. We then fuse all predicted surface distance results to recognize boundary pixels for the segmentation. Compared to existing methods, the proposed method is able to consider the whole volumetric data, without information loss, by sampling spherical surfaces sufficiently dense.

4.2 Methodology

Fig. 4.2 shows an illustrative pipeline. We first sample spherical surfaces; objects are coarsely segmented for doing so. We then organize the surface into 2D a plane. Next, we use a 2D U-Net [37] to learn the projection distance.

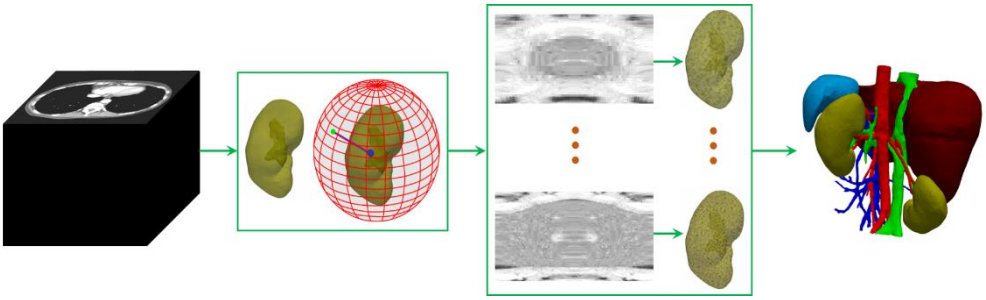


Fig. 4.2 The illustrative pipeline of the proposed surface projection. Given a volumetric data, it samples spherical surface (the red mesh) which is then organized into a 2D plane for using a 2D CNN to learn the projection distance mapping, and it finally fuses prediction results to decide the object's boundary pixels.

The last step is to fuse all recognization results to finally decide object's boundary pixels.

4.2.1 Spherical Surfaces Sampling

We start by presenting how to sample spherical surfaces. We denote the surface by $\mathbf{S}_n$. For the sampling of it, it is required to determine the origin of coordinates ($\mathbf{p}_n$) and the radius r_n. To determine them, we use a 2D U-Net to coarsely segment objects, and then place $\mathbf{p}_n$ at the centerline, with the distance interval of d, of the coarse segmentation results. As shown later, this method is not sensitive to the accuracy of the coarse segmentation result, so 2D CNNs are suitable.

We now just focus on the sampling of $\mathbf{S}_n$ and the organization of it into a 2D plane when the $\mathbf{p}_n$ and r_n are both given; details of how to get them are presented later. For this purpose, we use the angular coordinate system to

present data. In particular, we use two symbols: θ and φ, to denote the angular indexes of $\mathbf{S}_n$; here note that $\theta \in [0, 360)$ and $\varphi \in [0, 180)$. We then organize $\mathbf{S}_n$ as

$$\mathbf{S}_n(\theta, \varphi) = \mathbf{V}(\mathbf{p}_n + r_n \mathbf{d}(\theta, \varphi)), \tag{4.1}$$

where $\mathbf{V}(x, y, z)$ stands for the value of the intensity information at the pixel, indexed b (x, y, z); here we use the symbol $\mathbf{d}$ to indicate the 3D unit vector in the Cartesian coordinate system. Here note that the relationship is $\mathbf{d}(\theta, \varphi) = (\cos\theta \cos\varphi, \cos\theta \sin\varphi, \sin\theta)$.

4.2.2 Surface's Projection Distance Predicting

We below present details of how er learn 3D features by using a 2D network. To do so, we employ a 2D network to predict the projection distance mapping from the object's surface to sampled spherical surfaces. We define the projection distance as the distance of pixels in $\mathbf{S}_n$ to the object's surface, measured along the line from the pixel to the object's center, and thus a pixel has a projection distance. We set its value to negative if the pixel is inside the surface of the object and otherwise a positive value.

This 2D CNN is trained to predict the mapping of the projection distance,

$$\underset{\omega}{\operatorname{argmin}} \frac{1}{M} \sum_{n=1}^{M} \| f(\mathbf{S}_n; \omega) - \mathbf{G}_n \|_2^2, \tag{4.2}$$

where ω stands for network's parameters, f denotes the network, M is the number of training samples, and $\mathbf{G}_n$ is the ground truth of $\mathbf{S}_n$, representing the projection distance information. The symbol $\|A - B\|_2^2$ is an operation to

count the average square distance of pixels between A and B; here we use the Euclidean distance of pixel's value as the distance measure. It is worth to note here that more advanced or tailored functions can be used as the loss function; we here do not consider this problem.

We now move on to boundary pixels identification from the predicted projection distance result of $\mathbf{S}_n$. To do so, a pixel indexed by (θ, φ) in $\mathbf{S}_n$, if its predicted projection distance is ℓ, then we set the pixel at the position $\mathbf{p}_n + (r_n - \ell)\mathbf{d}(\theta, \varphi)$ in the input volumetric image as the boundary pixel.

Such a learning way is able to learn 3D features. We below present an intuitive explanation. 3D features, in principle, are captured by the geometrical and intensity information. So, if one just uses a plain 2D network, we miss the geometrical information that is among the 2D slices of the 3D medical data. However, by using our method, both types of intensity and geometrical information are considered; the ground truth provides geometrical information and the sampled surfaces provide intensity information. Moreover, existing 2D networks are able to model the mapping from the intensity to geometrical information [88-90], and hence the geometrical information is guaranteed to be well exploited.

4.2.3 Surfaces' Projection Distance Fusing

For each object, we sampled several surfaces to leverage comprehensive intensity and geometric information. For producing the segmentation result,

their results have to be fused. Below are details of how we fuse these complementary pieces of information of surfaces. In particular, suppose that there are N surfaces sampled for the object, and there are in total K pixels in the data. We use $\mathbf{D} \in \mathbb{B}^{K\times N}$ to denote the joint segmentation result matrix of surfaces. For this matrix, if the pixel k is recognized as a foreground pixel based on $\mathbf{S}_n$, we then set $\mathbf{D}(k,n) = 1$. Our goal is to estimate the unknown truth segmentation $\mathbf{G} \in \mathbb{B}^K$. To do so, we first estimate the segmentation accuracy $\mathbf{A}_n = (\delta_n, \xi_n)$ of $\mathbf{S}_n$. Here δ_n and ξ_n are the true positive and negative rates. We next estimate $\mathbf{G}$ according to $\boldsymbol{\mathcal{A}} = [\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_n]$ and $\mathbf{D}$ by solving the following problem

$$\boldsymbol{\mathcal{A}}^* = \underset{\boldsymbol{\mathcal{A}}}{\operatorname{argmax}}\, p(\mathbf{D}, \mathbf{G}|\boldsymbol{\mathcal{A}}) \text{ and } \mathbf{G}^* = \underset{\mathbf{G}}{\operatorname{argmax}}\, p(\mathbf{G}|\mathbf{D}, \boldsymbol{\mathcal{A}}^*), \tag{4.3}$$

here the symbol $p(\mathbf{D}, \mathbf{G}|\boldsymbol{\mathcal{A}})$ denotes the mass probability function of the joint data $(\mathbf{D}, \mathbf{G})$.

We then develop an iterative solution to solve Eq. 4.3. It initializes $\boldsymbol{\mathcal{A}}$ using the values $\{\delta_n, \xi_n\}_{n=1}^N$ of the training set, and updates $\mathbf{G}$ (at the step k) by

$$\mathbf{G}^t(k) = p(\mathbf{G}(k) = 1|\boldsymbol{\mathcal{A}}^t, \textstyle\sum_{n=1}^N \mathbf{D}(k,n)) = \frac{\mathbf{F}^t(k)}{\mathbf{F}^t(k)+\mathbf{B}^t(k)}, \tag{4.4}$$

where

$$\mathbf{F}^t(k) = p(\mathbf{G}(k) = 1) \textstyle\prod_n^{\mathbf{D}(k,n)=1} \delta_n^t \prod_n^{\mathbf{D}(k,n)=0} (1 - \delta_n^t),$$

$$\mathbf{B}^t(k) = p(\mathbf{G}(k) = 0) \textstyle\prod_n^{\mathbf{D}(k,n)=0} \xi_n^t \prod_n^{\mathbf{D}(k,n)=1} (1 - \xi_n^t). \tag{4.5}$$

$p(\mathbf{G}(k) = 1)$ denotes the prior probability. We then update $\boldsymbol{\mathcal{A}}$ by

$$\delta_n^{t+1} = \sum_k^{\mathbf{D}(k,n)=1} \mathbf{G}^t(k) / \sum_k \mathbf{G}^t(k),$$

$$\xi_n^{t+1} = \sum_k^{\mathbf{D}(k,n)=0} (1 - \mathbf{G}^t(k)) / \sum_k (1 - \mathbf{G}^t(k)). \quad (4.6)$$

We terminate the updating when $\mathbf{G}$ and $\boldsymbol{\mathcal{A}}$ are stable.

4.3 Experimental Evaluation

4.3.1 Experimental Setup

Dataset. This method is evaluated on a CT dataset[2] that is publicly available. It provides 90 CT volumes, and provides pixel-wise annotations for 8 organs. Their resolution is 0.6~0.9 mm (in-plane) and 0.5~5.0 mm (inter-slice).

Evaluation Metric. We used two commonly used performance metrics: Dice similarity coefficient and average surface distance, denoted by *DSC* and *ASD*, respectively. These two metrics are most commonly used in the application of 3D medical objects segmentation. In particular, *DSC* is to evaluate the matching extent of the result and ground truth, by computing pixels' number in their intersection region over the average of pixels' number in them. *ASD* is to evaluate the average distance of boundary pixels in the segmentation result and the ground truth. They take values in [0, 1] and [0, ∞]. A better segmentation method yields a larger *DSC* and a smaller *ASD*.

Implementation Details. We interpolate volumes to make them their resolution are same at three directions, by using linear interpolation. We use

[2] Available on https://zenodo.org/record/1169361\#.XSFOm-gzYuU

nn-UNet to search network's architecture[6]. The U-Net for sampling surfaces is trained by using Dice loss. We used Adam [41] to train two U-Nets. We set the initial learning rate to 0.0003. We terminated the training when the loss value is not decreasing. As for surfaces sampling, the parameter d, is set to 3. In addition, we set the parameter r_n to 5~$1.5r$ (the interval is set to 5); r here stands for the minimum ball's radius that covers the object in the coarse segmentation results. Note that these values are determined by cross-validation.

4.3.2 Experimental Results

Performance Improvement. We first evaluated the ability of our method to improve the segmentation performance. To do so, we compared the proposed method to the state-of-the-art methods: 2D-CRF, 2D-RNN, 2.5D, and 2D-MV, respectively. Among them, 2D-CRF [75] and 2D-RNN [76] are based on 2D slice distillation, learning 3D features by adding CRF and RNN on the learned 2D features. 2.5D [80] feeds neighboring 2D slices into a 2D network; we here set to 6 that works best in the validation dataset. 2D-MV [81] belongs to 2D multiple views-based methods, learning 3D features by fusing 2D features learned by different 2D networks at different views of the medical image; we here sampled three views (the common views: axial, coronal, and sagittal).

Table 4.1 Segmentation accuracy results: *DSC* (%) and *ASD* (mm).

	Raw		Coarse	
	DSC	*ASD*	*DSC*	*ASD*
2D-CRF	84.2±6.5	1.69±1.37	85.1±6.3	1.65±1.35
2D-RNN	84.8±6.3	1.65±1.33	85.4±6.2	1.61±1.32
2.5D	84.7±6.4	1.68±1.32	85.2±6.1	1.62±1.34
2D-MV	84.5±6.3	1.64±1.41	85.0±6.2	1.60±1.37
Ours	—	—	**87.5±5.6**	**1.57±1.24**

For a fair comparison, the network in all methods has the same architecture. We also train them using the same setup; all use Dice loss, Adam, and the same learning rate. Furthermore, we tested their performance by using the coarse segmentation results for coarsely locating objects. We obtained results by using a 5-fold cross-validation, and reported these results in Table 4.1 in which 'Raw' and 'Coarse' are results of them by taking the raw images and the coarse result as the input. It is observed from Table 4.1 that our method produces the highest segmentation performance, which shows the effectiveness of our method.

Comparison to 3D CNNs. We also compared our method to 3D networks. We compared them under the same amount of GPU memory used; we used 40%, 60%, 80%, and 100% to search the architecture of 3D networks and that of our network. We trained 3D networks in patch-based manner and low resolution-based manner; taking 3D patches of the raw image and low-resolution ones.

Table 4.2 Performance comparison results (*DSC*) of our method to 3D networks; they are compared by using the same GPU memory of 40%, 60%, 80%, and 100%; data in the brackets are the paired *t*-test results.

	40%	60%	80%	100%
3D-P-R	78.4±7.4 (0.001)	82.6±6.9 (0.004)	84.3±6.5 (0.003)	85.1±6.3 (0.003)
3D-P-C	79.7±7.1 (0.002)	83.1±6.7 (0.001)	85.7±6.1 (0.002)	86.2±6.0 (0.002)
3D-R-R	76.9±7.6 (0.007)	81.9±7.1 (0.005)	85.2±6.8 (0.004)	84.4±6.5 (0.004)
3D-R-R	78.1±7.4 (0.003)	82.7±6.8 (0.003)	83.2±6.6 (0.002)	85.3±6.1 (0.002)
Ours	**86.4±5.9**	**87.1±5.7**	**84.4±5.6**	**87.5±5.6**

Table 4.3 Performance comparison results (*ASD*) of our method to 3D networks; they are compared by using the same GPU memory of 40%, 60%, 80%, and 100%; data in the brackets are the paired *t*-test results.

	40%	60%	80%	100%
3D-P-R	1.81±1.52 (0.002)	1.74±1.43 (0.003)	1.67±1.39 (0.002)	1.65±1.35 (0.004)
3D-P-C	1.77±1.47 (0.004)	1.72±1.40 (0.001)	1.64±1.32 (0.001)	1.62±1.31 (0.003)
3D-R-R	1.84±1.54 (0.001)	1.76±1.47 (0.004)	1.72±1.41 (0.002)	1.68±1.39 (0.004)
3D-R-R	1.79±1.50 (0.002)	1.73±1.42 (0.002)	1.68±1.37 (0.002)	1.63±1.34 (0.006)
Ours	**1.62±1.29**	**1.59±1.27**	**1.57±1.26**	**1.57±1.24**

Also the learning setup is the same, both Dice loss and Adam have the same setup, for the training. We obtained results by using a 5-fold cross-validation. Results are reported in Tables 4.2 and 4.3. In the table, the symbols: '-P-R', '-P-C', 'R-R', and 'R-C', denote 3D **p**atch from **r**aw images, 3D **p**atch from **c**oarse segmentation, low **r**esolution from **r**aw images, and low **r**esolution from **c**oarse segmentation, respectively. These two tables show that our method works better than 3D networks when they consume the same amount of GPU memory. This means that our method is able to learn 3D features effectively.

Table 4.4 Performance varying by using 4 different networks for the coarse segmentation.

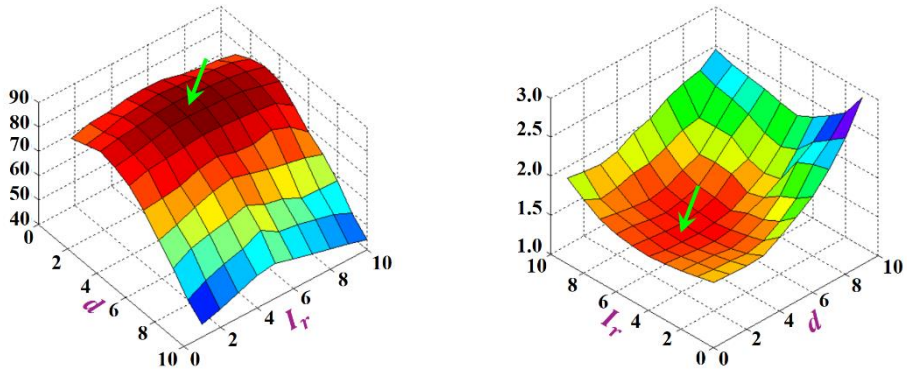
	40%	60%	80%	100%
DSC	87.2±5.7	87.4±5.6	87.5±5.5	87.5±5.6
ASD	1.60±1.26	1.58±1.25	1.58±1.24	1.57±1.24

Sensitivity to Coarse Segmentation Results. Our method samples spherical surfaces according to coarse segmentation results. It hence is necessary to evaluate the sensitivity of our method to coarse segmentation results. To this end, we tested our method on coarse segmentation results produced by four different networks whose architectures are searched by allocating 40%, 60%, 80%, and 100% of GPU memory. They were also trained in the same learning setup. We obtained the results using a 5-fold cross-validation. The results are reported in Table 4.4. We can see that the proposed method varies slightly, which suggests that our method produces segmentation results that are not affected by the coarse network's architecture.

Ablation Study. We also evaluated the key components of the proposed method. We compared the difference between our fusing procedure and majority voting (denoted by 'Voting' in Table 4.5). We can see that our procedure works better. We next evaluate the role of learning the projection distance. We compared it to the way that predicts pixels' category, denoted by 'Pro-S'. This way has the same idea as [91]. However, Table 4.5 shows that this way works worse than ours. We finally evaluate the role of sampling

Table 4.5 Ablation study results.

	Voting	Pro-S	Sur-B	Sur-S	Ours
DSC	86.2±6.0	84.1±6.8	86.3±6.0	86.9±5.8	**87.5±5.6**
ASD	1.60±1.30	1.69±1.37	1.63±1.31	1.58±1.29	**1.57±1.24**



(a) (b)

Fig. 4.3. Segmentation performance varying by setting different values to the parameters; results for *DSC* (a) and *ASD* (b).

surfaces that are spherical. To do so, we tested two types of surfaces: box and mean shape (Sur-B, Sur-S). It is observed from Table 4.5 that these two types of surfaces work worse than ours; Note that there are several works on organizing 3D object into 2D surfaces [181, 182], but spherical surface is the easiest way to implement.

Hyperparameters Sensitivity. Our method receives two hyperparameters: d and I_r. The first one is the distance interval for placing $\mathbf{p}_n$ while the second one is the radius interval for sampling surfaces. We decided their values by grid search from 1 to 10. Fig. 4.3 shows the results with different values of these two hyperparameters. We can see that there are different performances

with different values of them and when d and I_r equal to 3 and 5 our method produces the best results.

4.4 Closing Remarks

In this chapter, we proposed surface projection, a GPU-memory efficient method to learn 3D features. This method has great importance in real-world medical image segmentation tasks, because it allows 2D networks to learn 3D features, substantially reducing GPU memory requirement. We extensively evaluated this method on a large CT dataset that is publicly available, and obtained positive results that show the effectiveness of this method.

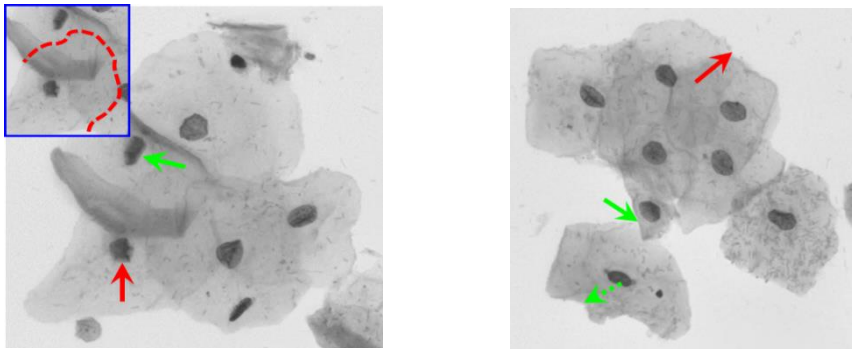
Chapter 5

Shape Constructing: Adaptive Shape Priors Modeling from Fragments for Segmenting Overlapping Objects

This chapter is about shape constructing, an effective method of modeling shape priors. We apply this algorithm to overlapping objects segmentation tasks in which some occluded boundary parts are visually indistinguishable, and so shape priors are strongly desired for guaranteeing the performance. This method models shape priors adaptively from contour fragments. To do so, it first uses a CNN to segment the clump and then cuts the contour into fragments. Next, it groups each object's fragments, estimates the object's shape template, and finally connects grouped fragments to produce the object's segmentation result. It iteratively conducts fragments grouping, shape template estimation, and fragments connecting, to continually refine shape priors for improving segmentation performance. It is assessed on two datasets, with clear experimental evidence showing its effectiveness.

5.1 Problem Background

Shape priors are strongly desired in overlapping objects segmentation tasks in which visual information often is deficient to segment occluded boundary



(a) (b)

Fig. 5.1 Illustration of the difficulties in overlapping cervical cytoplasms segmentation. (a) Some occluded boundary parts are visually indistinguishable; see the cytoplasm, highlighted by the red arrow the ground truth is depicted by the red dashed line). (b) It is difficult to locate intersection points; see two that points, highlighted by the green and red arrows (the point highlighted by the green dashed arrow is not such a point).

parts, as shown in Fig. 5.1 (a) for example. This example is about overlapping cervical cytoplasms segmentation for screening cervical cancer that is 2-nd cancer of women globally [7, 8].

Existing approaches include mainly two categories: (1) intensity-based and (2) shape priors-based. The former approaches [92-97] attempt to leverage intensity information for segmenting occluded boundaries. These methods hence cannot handle overlapping cases in which the occluded boundaries are visually indistinguishable. The second type of methods [98-102], differently, aim at leveraging shape priors of cytoplasm for the segmentation. These methods model shape priors by either using mathematical analysis or matching a shape template. They hence are limited to handle cytoplasm with the simple shape and heavily rely on the representation ability of the collected shape templates.

The proposed shape constructing models shape priors by leveraging shape information from both contour fragments and shape templates of objects. In particular, given an overlapping clump of cytoplasms, it first uses a CNN to segment the clump, and then cuts the clump's boundary into fragments. It next groups fragments of each cytoplasm, estimates the shape template, and connects the grouped fragments to produce the segmentation results. It iteratively conducts fragments grouping, shape template estimation, and fragments connecting, to continually refine shape priors and boosting the performance.

Unlike existing methods, this method holds three advantages. it first can handle overlapping cases that are complex and with irregular shapes. This is because our shape space of modeling shape priors is rather large than those by using stringent mathematical formulations or shape templates. Second, it is not restricted to the collected set of shape templates, as we use shape templates as just guidance to model shape priors. It third substantially reduces visual implausible results, because we put shape constraints on the fragments grouping. It is assessed on two cervical datasets, with positive results showing its effectiveness.

5.2 Literature Review

Intensity-based Methods. This type of methods uses intensity information for the segmentation by assigning pixels to its cytoplasm. The simplest

method perhaps is adaptive threshold [92, 103]. It often uses a circular region surrounding the nucleus to estimate the threshold for segmenting that cytoplasm; pixels with intensity information greater than the estimated threshold are segmented into that cytoplasm. It also can be done by using the curve that connects local peaks (or valleys) of intensity information surrounding the nucleus as cytoplasm's boundary [96]. Another way is to use region growing techniques that take the nucleus as the seed [94, 97] or as the marker of watershed techniques [93, 95]. The seed or marker is growing by merging neighboring pixels with similar intensity values. This type of methods is computationally effective, however suffers from the imaging quality. They usually produce unsatisfactory results when the intensity information is unclear, a common case that can be often encountered in practice.

Shape Priors-based Methods. This type of methods segments overlapping objects by leveraging shape priors. These methods often have two ways of modeling shape priors: shape assumption and shape template matching. The former way [98, 99, 104] attempts at designing a mathematical formulation to describe cytoplasm's shape. The most commonly used technique in this line is to assume cytoplasms to be elliptical. Therefore, they fail to capture shape details, especially for cytoplasms that have irregular or complex shapes. This is mainly because of the difficulty of formulating such highly summarized but also computationally feasible formulations. The later way

[100, 104, 105]) is to match a shape template from the collected template set that best describes the cytoplasm's shape and then use the matched shape template as shape priors for the segmentation. For capturing sufficient shape details, it is required for them to collect shape templates carefully. Such a process is often time-consuming and requires expensive domain expertise. Compared to existing methods, the proposed method can capture sufficient shape details, as it models shape priors from contour fragments and shape statistics of cytoplasms.

5.3 Methodology

Fig. 5.2 shows how our method to model shape priors for segmenting overlapping cervical cytoplasms. We first employ a CNN to segment the clump, and then cut the clump boundary into fragments. We next group fragments into its cytoplasm. Then we estimate shape templates for cytoplasms based on the grouping results. Next, we connect each cytoplasm's fragments to form its boundary based on the estimated shape template that is used as a shape constraint. We developed fragments grouping, shape template estimation, and fragments connecting as an iterative solution, to continually refine the modeled shape priors for boosting the performance. This iterative process is terminated when either reaching the maximal iteration number or a stable result of fragments grouping.

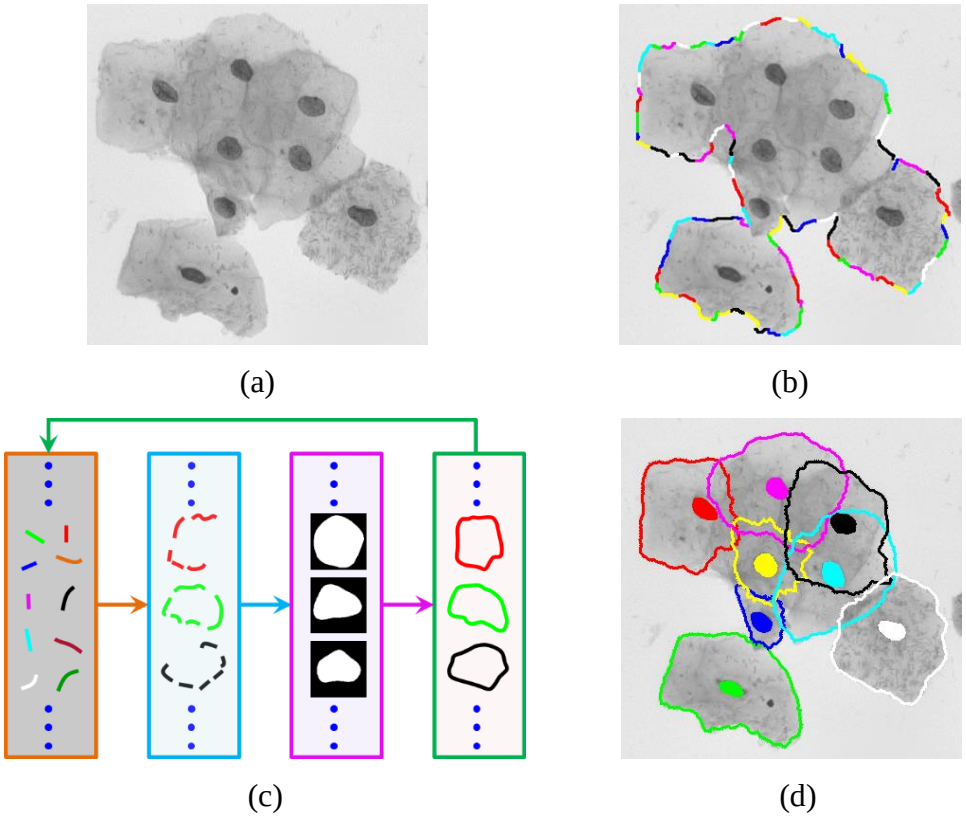


Fig. 5.2 Illustration of the proposed method: (a) the input image; (b) generating fragments; (c) the iterative process of three key steps; (d) the segmentation result.

5.3.1 Fragments Generating

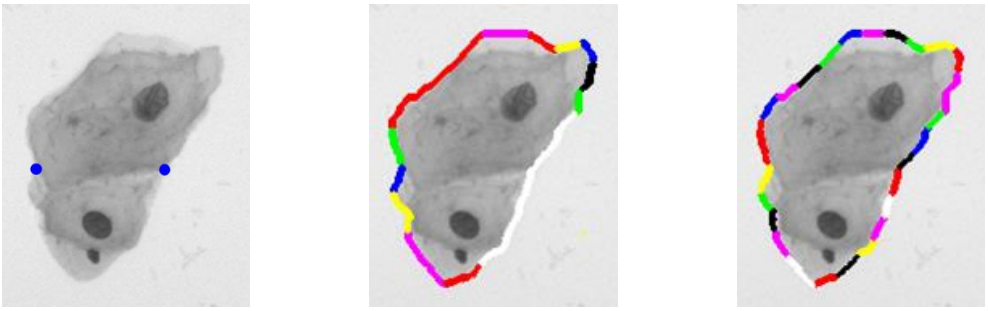
We first use a CNN [106] to segment the cytoplasms clump by classifying each pixel into either nuclei, cytoplasm, or background. The used CNN has three scales in the sense of patch sizes. Such a network architecture has been proven to be able to learn contextual information at different information scales of the training image. The extracted features then are fused for classifying pixels. We finally use a graph partitioning technique to refine CNN's result; more technical details can be found in [106].

We next estimate each boundary point's curvature information, and cut the clump's boundary into contour fragments at the points that have local extremum of curvature for generating fragments. Here note that for cytoplasms with convex shape, boundary points that are of the local curvature extremum are intersection points, but for other complex shapes, not all these points are intersection points; Fig. 5.1 (b) shows such an example. For cervical cytoplasms, most of them have a non-convex shape, and so only the above procedure cannot accurately locate intersection points.

To handle complex shapes, our idea is to limit fragments' length such that the endpoints of fragments can be as near to intersection points as possible. In particular, when fragments are longer than the prescribed value, we cut them at their midpoints to make them shorter. We iteratively do this process until there are no fragments longer than the prescribed value. An illustration of why this idea works is shown in Fig. 5.3. We can see from it that by doing so we can find more intersection points.

5.3.2 Fragments Grouping

We next group fragments of each cytoplasm by developing a Markov random fields (MRF) model. The purpose of this step is to cluster each cytoplasm's fragments such that we can use them to estimate the shape template for cytoplasms. Its performance is significant in shape template estimation. Without a proper grouping, shape templates are estimated from incorrect



(a) (b) (c)

Fig. 5.3 Illustration of why length limit helps to find intersection points: (a) the input image has two intersection points, the blue points, (b) fragments generating without using length limit, and (c) fragments generating with using length limit.

shape cues. Details are presented below.

Let $\mathcal{F} = \{1, \cdots, M\}$ be the M fragments of the clump that has N nuclei denoted by $\mathcal{N} = \{1, \cdots, N\}$. For the grouping, we design the energy function of the MRF model as below

$$E(\ell) = \sum_{i \in \mathcal{F}} (\omega_1 D_i(\ell_i) + \omega_2 \sum_{i \in \varepsilon_i} V_{ii'}(\ell_i, \ell_{i'})), \tag{5.1}$$

where ℓ represents a grouping function in the admissible space. We use ℓ_i to denote the fragment i that is grouped into the nucleus ℓ_i; here for simplicity we interchangeably use ℓ_i and n to denote the nuclei. In addition, $D_i(\ell_i)$ is the cost of such a grouping. We set its value to the probability of i belonging to the boundary parts of the cell with the nucleus of ℓ_i (after the negative logarithm operation). Similarly, $V_{ii'}(\ell_i, \ell_{i'})$ is the cost of grouping two neighboring fragments i and i' to ℓ_i and $\ell_{i'}$ at the same time, with the value of the probability of i and i' belonging to ℓ_i and $\ell_{i'}$. We here use the symbol ε_i to denote the neighboring fragments of i. Finally, we use two parameters:

ω_1 and ω_2, to balance the importance of these two terms in the energy function.

The key of the grouping quality is to enable the energy function to consider shape priors. To this end, we here model two types of shape priors. The first one is the spatial relationships between nuclei and fragments, and another one is the shape relationships between fragments. Below are the details of how we define $D_i(\ell_i)$ and $V_{ii'}(\ell_i, \ell_{i'})$ in a manner that can model the above-mentioned shape priors.

For introducing $D_i(\ell_i)$, we below present our definition of the spatial relationship. We use $e_i = \{p_1, \cdots, p_m\}$ to denote boundary pixels of the fragment i that has m boundary points. We define the spatial distance between i to the nucleus n as

$$g(i,n) = \frac{1}{m}\sum_{p_k \in e_i} g_k(p_k, n), \tag{5.2}$$

where $g_k(p_k, n)$ is a distance of the nucleus' centroid to the boundary point p_k by Euclidean measure, if pixels in the line segment from the centroid to p_k are all in the clump and no other nucleus' pixels in that line segment, otherwise, we set it to ∞.

Next, we define the shape distance between fragments. To do so, we match a shape template from the collected set, and then use it to measure the distance. More specifically, it is defined as:

$$h(i,n) = \underset{\phi \in \Phi}{\operatorname{argmin}} \frac{1}{m}\sum_{p_k \in e_i} \| g_k(p_k, n) - g_k(p_{k'}, n|\phi) \|, \tag{5.3}$$

where ϕ is the matched shape template from the collected shape templates set Φ, and $p_{k'}$ stands for the location of p_k in ϕ after shape alignment by using [107].

The shape templates set plays a significant role in measuring the distance. It should have the capability of representing most cytoplasms' shape, otherwise, there will be a considerable biased error. We hence collect shape templates as below. We first cluster the manually collected templates into K clusters by using the k-means algorithm, and then select all the center templates that have the minimal inner class distance as the shape templates into the set.

Based on these two types of distance, we formulate $D_i(\ell_i)$ as below

$$D_i(\ell_i) = \omega_3 g(i, \ell_i) - (1 - \omega_3) h(i, \ell_i), \tag{5.4}$$

where ω_3 is a parameter to control two terms' importance. Here note that these two types of distance g and h have been normalized into [0, 1] (a larger value of them is normalized into a smaller value).

We next formulate $V_{ii'}(\ell_i, \ell_{i'})$ as:

$$V_{ii'}(\ell_i, \ell_{i'}) = \min(D_{ii'}(\ell_i), D_{ii'}(\ell_{i'})), \tag{5.5}$$

if $\ell_i \neq \ell_{i'}$, otherwise 0. $D_{ii'}(\ell_i)$ here stands for the cost of a combined fragment consisting of two neighboring fragments i and i' being assigned to the same label ℓ_i, measured by Eq. 5.4. Here note that more shape cues become available by considering these combined fragments along with the original fragments.

We finally solve Eq. 5.1 by employing the optimization method [108]. This method first evaluates the value of the objective function for a possible grouping ℓ. It then attempts to reduce the energy function by finding a new grouping function. It terminates when getting the grouping function ℓ^* that yields the minimal value of the objective function.

5.3.3 Shape Template Estimation

Once fragments of cytoplasms have been grouped, we start to estimate the cytoplasm's shape template based on the grouping result. For the estimation, we describe a shape by

$$\mathcal{C} = \mu_s + \lambda b, \tag{5.6}$$

where $\mathcal{C}$ stands for the estimated shape, μ_s and λ are shape statistics extracted from the collected shape templates set; μ_s: mean shape, λ: eigenvectors of shape templates set' covariance matrix, and b: the vector that controls the shape.

Our goal here is to estimate a good value of b. Below are the details. Let $\mathcal{O}_n$ be the grouped fragments of the cytoplasm n. Since there may not exist a deterministic mapping from $\mathcal{O}_n$ to $\mathcal{C}_n$, we instead try to maximize $p(\mathcal{C}_n|\mathcal{O}_n; \mu_s, \lambda)$, the probability of $\mathcal{C}_n$ controlled by b_n is the optimal shape template of the cytoplasm n given $\mathcal{O}_n$, μ_s and λ. In other words, we estimate the shape template by solving the problem

$$\mathcal{C}_n^* = \underset{\mathcal{C}_n \in \Omega_{\mathcal{C}}}{\operatorname{argmax}}\, p(\mathcal{C}_n|\mathcal{O}_n; \mu_s, \lambda), \tag{5.7}$$

where $\Omega_{\mathcal{C}}$ stands for the shape space.

We solve the above problem by using the Bayes rule, according to which we get that it is equivalent to solve

$$\underset{\mathcal{C}_n \in \Omega_{\mathcal{C}}}{\operatorname{argmax}}(p(\mathcal{C}_n; \mu_s, \lambda)\, p(\mathcal{O}_n | \mathcal{C}_n; \mu_s, \lambda)). \tag{5.8}$$

We next bridge the gap between the probability $p(\mathcal{C}_n | \mathcal{O}_n; \mu_s, \lambda)$ and b_n by using [109, 110]. According to them, we can solve Eq. 5.8 by solving the below problem

$$b_n^* = \underset{b_n \in \Omega_b}{\operatorname{argmin}}(b_n^T \sum_s^{-1} b_n + \omega_4 \|\mathcal{O}_n^* - \mathcal{O}_n\|), \tag{5.9}$$

where b_n^T and $\sum_s^{-1}$ denote the transpose of b_n and the inverse of the matrix $\sum_s$ and $\mathcal{O}_n^*$ stands for the corresponding location of the grouped $\mathcal{O}_n$ on the shape template $(\mu_s + \lambda b_n)$. In addition, the operation ||x|| is the L_2 norm of the vector x. It finally uses the hyperparameter parameter ω_4 to control two terms' contribution.

For the implementation, we represent $\mathcal{O}_n$ by a vector in the polar coordinate. In particular, we place the nucleus' centroid as the coordinate's origin, and organize boundary pixels of the cytoplasm as the vector to represent that cytoplasm's shape. This way is effective in the sense of computation and implementation.

Why Eq. 5.9 bridges the gap is below. $b_n^T \sum_s^{-1} b_n$ is the appropriate value of the negative logarithm of $p(\mathcal{C}_n; \mu_s, \lambda)$ that models global shape priors.

Similarly, $\|\mathcal{O}_n^* - \mathcal{O}_n\|$ is the negative logarithm of $p(\mathcal{O}_n|\mathcal{C}_n; \mu_s, \lambda)$ that model local shape priors.

We finally optimize Eq. 5.9 by using [111], and produce the shape template of the cytoplasm n by $\mathcal{C}_n^* = \mu_s + \lambda b_n^*$.

5.3.4 Fragments Connecting

Given the estimated shape template, our goal is to connect the grouped fragments to produce the segmentation result. Wo connect two fragments by finding a curve γ^* by solving the below problem

$$\gamma^* = \underset{\gamma \in p_s \to p_e}{\operatorname{argmin}} \int_\gamma f(\gamma(s))\, ds, \tag{5.10}$$

where p_s and p_e denote the starting point and ending point of the curve. s is the variable that controls γ. $f(\gamma(s))$ is the energy function that evaluates the quality of the curve $\gamma(s)$.

We here design $f(\gamma(s))$ by considering shape priors, intensity information, and curvature information, because these types of information are necessary to segment occluded boundary parts. This function hence takes the following form

$$f\big(\gamma(s)\big) = -\omega_5 \|\nabla I(s)\| + \omega_6 \|\mathcal{C}_d(s)\| + \|\gamma\| k^2(s), \tag{5.11}$$

where $\|\nabla I(s)\|$, $\|\mathcal{C}_d(s)\|$, and $k(s)$ stand for the intensity gradient magnitude, shape distance to the estimated shape template, and the curvature, at s; $\|\gamma\|$ is the length of the curve.

For computational efficiency, we consider curvature information in the form of $\|\gamma\|k^2(s)$ rather than $k(s)$, because it can be directly approximated with sufficient accuracy [112], that is,

$$\int_\gamma \|\gamma\|\, k^2(s)\, ds \approx \frac{4}{d}(4(\theta_s - \theta_e)^2 + \theta_s\theta_e + {\theta_p}^2), \tag{5.12}$$

where d is the Euclidean distance between p_s and p_e, and θ_s, θ_e, and θ_p are the orientation at p_s and p_e, and of the vector $\overrightarrow{p_s p_e}$, respectively.

We finally use two parameters ω_5 and ω_6 to balance the importance of three types of information. Below is the solution of Eq. 5.10. We first evaluate the minimal action map by a forward propagation that gives the cost of pixels being the curve from p_s to p_e. We next find the optimal γ^* by connecting pixels from p_e to p_s with minimal cost.

The minimal action map is evaluated as follows. From p_s, we assign its neighboring pixels with the action value $\int_\gamma f(\gamma(s))\, ds$. We then select the neighboring pixel that has the minimal action value, and start to evaluate the action value of its neighboring points. The above process is repeated until the action value of p_e has been evaluated.

5.3.5 Iterative Refinement

By fragments connecting, we have got the segmentation result. However, it can be refined by iteratively conducting fragments grouping, shape template estimation, and fragments connecting. The reason why it works is below. The segmentation results can be used to help the fragments grouping. They can

replace shape templates and provide more accurate shape information, and thus improve the segmentation results. They also can be used to exploit clump-level shape information, not only the cytoplasm-level shape information. Intuitively, more information available will help to refine the result.

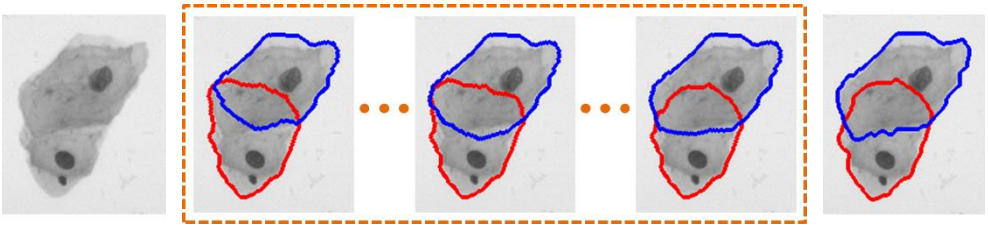
We now present the iterative procedure. Our idea is to promote the similarity between the clump and the combined clump of segmented cytoplasms. We hence refine the segmentation results by improving the similarity.

Before presenting the similarity measure, we present two symbols: $\mathbb{R}$ and $\mathbb{C}$, denoting by the combined clump and input clump, respectively. The similarity then is measured by the operation $\mathrm{Card}(\mathbb{R}\backslash\mathbb{C} + \mathbb{C}\backslash\mathbb{R})$; the operation Card is to count the elements' number of a set. Based on this measure, the more similar between $\mathbb{R}$ and $\mathbb{C}$ yields a larger similarity value.

We finally refine the segmentation results by minimizing the following equation

$$\mathbb{E}(\ell) = E(\ell) + \mathrm{Card}(\mathbb{R}\backslash\mathbb{C} + \mathbb{C}\backslash\mathbb{R}). \tag{5.13}$$

It starts to update the shape distance and then produces a better grouping of fragments that reduces the value of $\mathbb{E}(\ell)$. It next estimates the shape templates based on the updated fragments grouping results, then updates the fragments connecting results, and finally repeats this iterative process. This process



(a) (b) (c)

Fig. 5.4 Illustration of why our iterative procedure works: (a) the input image of an example, (b) the intermediate results, especially, of 1st-3rd iteration, and (c) the final result produced by the proposed method.

is terminated either reaching the given iteration number or getting the stable fragments grouping results.

The iteration number, in this application, we set to 10 that produces the best results; it can be determined by cross-validation in other applications. In our application, it is also observed that most cervical clumps just need 3~4 iterations. Fig. 5.4 shows an example of how the segmentation result is continually improved by the processed iterative process, showing the effectiveness.

5.4 Experimental Evaluation

5.4.1 Experimental Setup

Datasets. This method is assessed on two cervical smear datasets. The first dataset is collected from the ISBI 2015 Overlapping Cervical Cytology Image Segmentation Challenge. We called it as Pap stain dataset, because it is prepared by Papanicolaou (Pap) stain. It consists of 8 public images, each of

them having 20~60 cells. There are 11 clumps on average on them, and 3.3 cells in a clump.

Another one is called H&E stain dataset, because it is prepared by Hematoxylin and Eosin (H&E) stain. It is collected from Shenzhen Sixth People's Hospital in China. It consists of 21 cervical images, each of them having 30~80 cells. There are 7 clumps on average on them, and 6.1 cells in a clump.

Performance Metrics. We first employ the Dice similarity coefficient, defined as $DSC = 2|R_s \cap R_g|/(|R_s| + |R_g|)$, to measure the performance, where R_s and R_g stand for the segmentation result and the ground truth. We also introduce two other metrics that are based on shape smoothness $\mathcal{S}(R) = \oint_{p\in\partial R} |\, d\theta(p)\,|$ and shape roundness $\mathcal{R}(R) = \int_{p\in R} dp\,/(\oint_{p\in\partial R} dp)^2$, where ∂R indicates the boundary of region R, $\theta(p)$ indicates the change of the tangent angle at the boundary point p, and $|.|$ indicates the absolute value. We define the smoothness similarity coefficient as $SSC = 1 - |\mathcal{S}(R_s) - \mathcal{S}(R_g)|/\mathcal{S}(R_g)$ and the roundness similarity coefficient as $RSC = 1 - |\mathcal{R}(R_s) - \mathcal{R}(R_g)|/\mathcal{R}(R_g)$. Note that all *DSC*, *SSC*, and *RSC* have a value in [0, 1], and a better segmentation algorithm produces a higher value for these three metrics.

Implementation. Training images and shape templates are randomly selected. We used 5-fold cross-validation to report the result. In each fold, there are usually about 250 cells. We cluster them into 40 clusters by using *k*-means.

We therefore have 40 shape templates that are used to compute shape distances in fragments grouping at the first iteration. In the shape templates estimation step, however, all cytoplasms are used as shape templates, for providing more shape information.

Parameters Selection. Our shape constructing has 7 parameters: length threshold of fragments, and ω_1 to ω_6. Their values are determined by cross-validation. Fragments' length threshold has some relationship to cytoplasm's perimeter. Our experimental evidence shows that a good value of it should range in 5~10% of the average perimeter of cytoplasms. We here empirically set it to 30 pixels.

As for ω_1 and ω_2, a small value of them will ignore the importance of shape priors, but a large value of them can bring optimization difficulties, reducing the convergence speed. They are set to 10 and 4 by cross-validation results.

The parameter ω_3 is to balance two types of shape distance. There is no formal analysis of which of them should be set larger, mainly because of the huge range of cervical cytoplasms' size. We hence empirically set its value to 0.5.

The parameter ω_4 is decided by the overlapping degree. In principle, when cytoplasms are with a higher overlapping degree, a larger value should be set to ω_4. We finally set ω_4 to 7 and 10 in the Pap and H&E stain datasets, respectively.

Finally, the parameters ω_5 and ω_6, their value is related to the imaging quality. It is observed that a large value of them is more likely to produce more accurate results if the given images have clear intensity information. We empirically set ω_5 to 1 and ω_6 to 0.2 in the Pap dataset, and 0.7 and 0.4 in another dataset.

Competitors. We compared the proposed method to four existing methods, denoted by LSF [98], MCL [100], DSM [101], and GSD [102]. They are all developed by using shape priors; LSF using shape assumption while MCL, DSM, and GSD using shape matching. We get the segmentation results of LSF and MCL by re-running the codes provided by the authors, and of DSM and GSD by reproducing them with the recommend implementations by the authors.

5.4.2 Experimental Results

Quantitative Results. Tables 5.1 to 5.3 reported the results of the proposed method on the Pap stain dataset while 5.4 to 5.6 report the results on the H&E dataset, under different overlapping degrees. We define it as the length ratio of the occluded boundary to the whole cytoplasm's boundary. The results are organized into three groups according to the overlapping degree: $(0, 0.3]$, $(0.3, 0.6]$, and $(0.6, 1)$, with the number of cytoplasms of 621, 407, and 203, respectively.

We can see that all methods' performance measured by all metrics is decreasing with the increasing of overlapping degree. It is also observed that

Table 5.1 Performance comparison results on the Pap stain dataset under the overlapping degree of $(0, 0.3]$.

	DSC	*SSC*	*RSC*
LSF [98]	0.79±0.06	0.75±0.08	0.88±0.10
MCL [100]	0.79±**0.04**	0.80±0.07	**0.92**±0.11
DSM [101]	0.81±0.08	0.81±0.10	0.84±0.08
GSD [102]	0.80±0.10	0.78±0.09	0.85±0.10
Ours	**0.84**±0.06	**0.86**±**0.05**	0.90±**0.06**

Table 5.2 Performance comparison results on the Pap stain dataset under the overlapping degree of $(0.3, 0.6]$.

	DSC	*SSC*	*RSC*
LSF [98]	0.76±0.11	0.80±0.10	0.85±0.09
MCL [100]	0.77±0.08	0.83±0.09	**0.88**±0.10
DSM [101]	0.78±0.07	0.82±0.08	0.85±0.08
GSD [102]	0.77±0.08	0.82±**0.06**	0.86±0.09
Ours	**0.81**±**0.05**	**0.85**±0.08	0.87±**0.06**

Table 5.3 Performance comparison results on the Pap stain dataset under the overlapping degree of $(0.6, 1)$.

	DSC	*SSC*	*RSC*
LSF [98]	0.73±0.07	0.75±0.08	0.81±0.09
MCL [100]	0.75±0.06	0.76±0.10	0.82±**0.07**
DSM [101]	0.74±**0.05**	0.76±0.07	0.82±0.08
GSD [102]	0.75±0.07	**0.77**±**0.06**	0.82±0.10
Ours	**0.79**±0.06	**0.77**±0.08	**0.85**±**0.07**

the proposed method works better than all compared methods by almost all three metrics. Specifically, when the overlapping degree is no greater than

Table 5.4 Performance comparison results on the H&E stain dataset under the overlapping degree of $(0, 0.3]$.

	DSC	*SSC*	*RSC*
LSF [98]	0.79±0.06	0.77±0.10	0.89±0.07
MCL [100]	0.80±0.07	0.82±0.11	**0.90±0.06**
DSM [101]	0.80±**0.04**	0.79±**0.07**	0.87±0.07
GSD [102]	0.81±0.08	0.80±0.08	0.86±0.08
Ours	**0.83**±0.06	**0.85**±0.08	0.89±0.08

Table 5.5 Performance comparison results on the H&E stain dataset under the overlapping degree of $(0.3, 0.6]$.

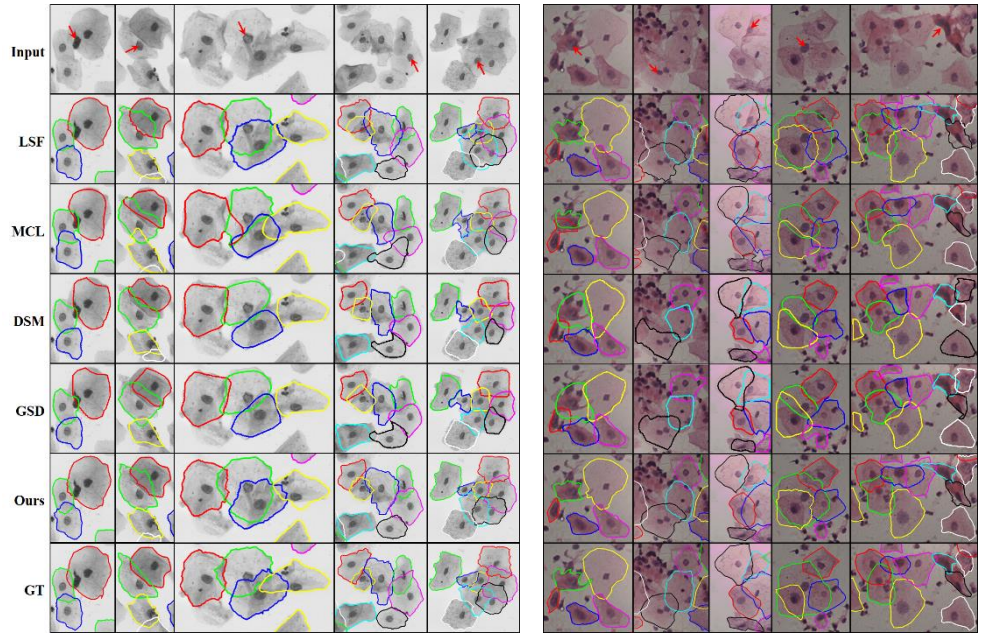
	DSC	*SSC*	*RSC*
LSF [98]	0.74±0.10	0.77±**0.07**	0.82±0.09
MCL [100]	0.74±0.11	0.82±0.08	0.84±0.10
DSM [101]	0.75±0.09	**0.83**±0.09	0.83±**0.07**
GSD [102]	0.76±**0.08**	0.82±0.09	0.84±0.08
Ours	**0.80**±0.09	**0.83±0.07**	**0.86**±0.08

Table 5.6 Performance comparison results on the H&E stain dataset under the overlapping degree of $(0.6, 1)$.

	DSC	*SSC*	*RSC*
LSF [98]	0.69±**0.06**	0.71±0.08	0.79±0.09
MCL [100]	0.69±**0.06**	0.71±0.08	0.82±0.11
DSM [101]	0.72±**0.06**	0.74±0.09	0.80±**0.07**
GSD [102]	0.74±0.09	**0.75±0.06**	0.81±0.08
Ours	**0.78**±0.07	**0.75**±0.07	**0.83**±0.09

0.3, the proposed method, on average, has a 3% performance improvement.

With the increasing of the overlapping degree, the improvement is becoming



(a) (b)

Fig. 5.5 Visual comparison results: (a) examples (Pap stain dataset) and (b) examples (H&E stain dataset); samples' size is scaled for better viewing.

more noticeable; the performance gain is about 7% when the overlapping degree is no less than 0.6. These empirical results suggest that, compared to other methods, our shape constructing is a more effective method to segment complicated and highly overlapping cervical cytoplasms.

Qualitative Results. In Fig. 5.5, we provide several visual examples of the segmentation results of these methods, for the qualitative comparison about them. Note that we arrange the sampled examples in Fig. 5, according to the overlapping degree with the increasing order. We can see from them that our method works better than all compared methods in these cases, and visually implausible results are significantly reduced.

Table 5.7 Ablation study results on the Pap stain dataset under the overlapping degree of $(0, 0.3]$.

	DSC	*SSC*	*RSC*
CS-A	0.80±0.10	0.73±0.06	0.83±0.08
CS-S	0.79±0.07	0.76±**0.05**	0.87±0.09
CS-K	0.81±0.06	0.82±0.07	0.89±0.10
CS-1	0.80±**0.05**	0.83±0.09	0.83±0.07
Ours	**0.84**±0.06	**0.86±0.05**	**0.90±0.06**

Ablation Study. We also implemented four variations of our method, denoted respectively by CS-A, CS-S, CS-K, and CS-1. CSA, CS-S, CS-K are variations considering only $\|\nabla I(s)\|$, $\|\mathcal{C}_d(s)\|$, and $\|\gamma\|k^2(s)$ in Eq. 5.11, respectively, for evaluating the effects of the intensity cue, shape priors, and curvature cue on the segmentation accuracy, while CS-1 removes the iterative scheme. Experimental results are presented in Table 5.7 from which it is observed that our method works better than all its variations, suggesting that all the components of the proposed method are necessary and mutually reinforcing.

5.5 Closing Remarks

Shape constructing is an effective method of modeling shape priors, and extensive experimental results show that it works well in overlapping cervical cytoplasms segmentation. Compared to existing methods, most of which exploit intensity information or shape formulations, we model shape priors

from clump’s contour fragments to integrate more shape details for the segmentation. We evaluate this method by comparing the segmentation accuracy to several existing methods, and our experimental results obtained on two datasets indicate that our shape constructing has great potential to segment overlapping cervical cytoplasms, outperforming existing methods.

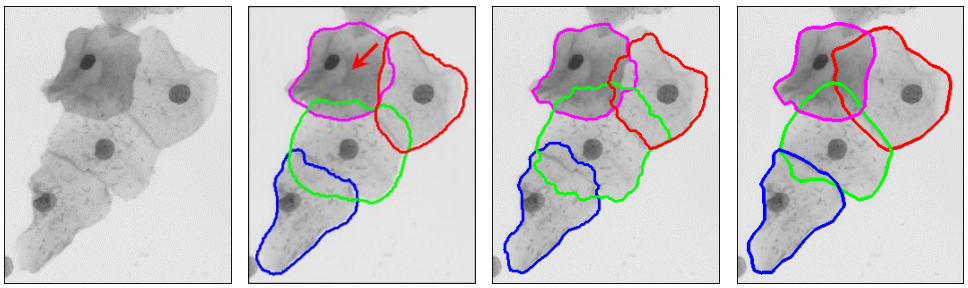
Chapter 6

Shape Mask Generator: Learning to Refine Shape Priors for Segmenting Overlapping Objects

This chapter presents shape mask generator that models shape priors by learning how to refine them. We use learning method to refine shape priors until they can describe cytoplasms' most shape. In particular, we first model shape priors from a collected shape template set and then use the modeled shape priors to estimate each cytoplasm's shape mask. We next refine the modeled shape priors by reducing the generating residual which is designed to be smaller when the resulting shape masks are more accurate. We assess the proposed method on two cervical smear datasets, with the extensive results showing the effectiveness of this method in overlapping cervical cytoplasms segmentation.

6.1 Problem Background

Overlapping cervical cytoplasms segmentation makes it possible to measure cell-level information that is required to screen cervical cancer [113-115], both 4-th of the morbidity and mortality in women in global [117]. Currently, an effective way of combating this cancer is to screen cervical cancer [8, 116].



(a) (b) (c) (d)

Fig. 6.1 Illustration of the difficulty in this task: (a) deficient image information, (b) intensity-based methods to be sensitive to imaging quality, (c) shape priors-based methods producing visually implausible results, and (d) the ground truth.

It is, however, rather challenging, as often the intensity information is deficient, as shown in Fig. 6.1 (a).

Existing approaches of this task include mainly two types: (1) intensity-[92, 93, 96, 97, 118-122] and (2) shape priors-based [98-105, 123-126]. The key idea of the first type of methods is based on the usage of intensity information, often by developing classic segmentation algorithms, e.g. thresholding [92, 97, 118], watershed [93, 119, 120], graph-cut [121], and morphological filtering [96, 122]. This type of methods hence relies on the imaging quality, and often does not work when the intensity information is deficient, as shown in Fig. 6.1 (b).

The second type of methods assume that intensity deficiency can be compensated by using shape priors if they are modeled appropriately. They often integrate the modeled shape priors into shape-based algorithms, e.g. level set model [127-129], to produce the segmentation results. Existing

methods model shape priors by either limiting cytoplasms' shape (star shape [19] or elliptical shape [98, 123, 124] for example) or matching a shape template [100-102, 105, 125, 126]. The modeled shape priors by these methods therefore are often lack of representation capability, and so these methods may produce visually implausible results, as shown in Fig. 6.1 (c) for example.

The proposed shape mask generator is a simple method to model shape priors for segmenting overlapping cervical cytoplasms. We refine shape priors by learning to refine them. This method directly produces the shape mask of cytoplasms for segmentation, according to the modeled shape priors. In detail, we refine shape priors by reducing the discrepancy between segmentation results and the ground truth. This method was assessed on two cervical datasets, and we obtained positive results that show the effectiveness of this method.

6.2 Methodology

Fig. 6.2 shows an illustrative pipeline of our shape mask generator. We first collect shape templates, and use them to model, $\boldsymbol{\mu}$ and $\mathbf{M}$, the mean shape of the templates and eigenvectors of the templates' covariance matrix. They are what we called shape priors and want to refine for improving the segmentation accuracy. For refining them, a key is to design the objective function that can measure the quality of the modeled shape priors. Here, we

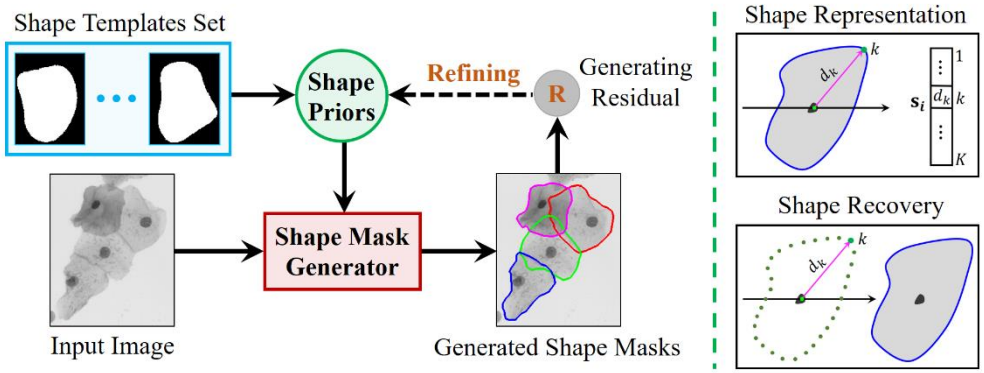


Fig. 6.2 The illustrative pipeline of our method to refine shape priors.

refine them by

$$\{\boldsymbol{\mu}^*, \mathbf{M}^*\} = \underset{\boldsymbol{\mu},\mathbf{M}}{\operatorname{argmin}}\, \mathbb{E}_{\mathcal{D}}[\mathrm{R}(\mathbf{I}, \mathrm{G}(\mathbf{I}; \boldsymbol{\mu}, \mathbf{M}))], \tag{6.1}$$

where $\mathbb{E}_{\mathcal{D}}$ denotes the expectation of generating residual R in the training dataset $\mathcal{D}$; $\mathbf{I}$ and $\mathrm{G}(\mathbf{I}; \boldsymbol{\mu}, \mathbf{M})$ stand for the input image and generated shape masks of cytoplasms, respectively.

6.2.1 Shape Mask Estimation

We now move on to shape mask estimation. In particular, we estimate a shape mask $\mathbf{s}_i$ of cytoplasm i by $\mathbf{s}_i = \boldsymbol{\mu} + \mathbf{M}\mathbf{x}_i$. To do so, we first represent a shape by using it's boundary information. We organize each shape as a vector with size K, the k-th entry of it storing the distance d_k between the boundary point k to the nucleus' centroid; K boundary points are sampled with the same angle interval. The estimation is to find an appropriate $\mathbf{x}_i$, and when we got it, we recover $\mathbf{s}_i$ to the image by first locating the corresponding K boundary points and next filling the region.

Below are details of how to find $\mathbf{x}_i$. At first, we set all $\mathbf{x}_i$ as $\mathbf{0}$, and then align the corresponding shape mask to the image by rotating and scaling, that is, solving the below problem

$$\text{argmax}\, B_i \cap B_c, \ \text{s.t.}\ B_i \in B_c, \tag{6.2}$$

where B_i is the aligned shape mask, and B_c is the clump area segmented by a CNN [106]. Note that we illustrate how get shape mask from shape vector in Fig. 6.2 (the Shape Recovery plot).

Once we have got the aligned shape masks, we compute that discrepancy that is defined as

$$\mathbb{E}(B_c, \mathbf{x}) = \sum_{(x,y)\in\Omega_B}(B_u(x,y) - B_c(x,y))^2, \tag{6.3}$$

where $\mathbf{x}$ is the symbol of all $\mathbf{x}_i$ for the cytoplasms in the clump, and $B_u = \bigcup B_i$ is the generated clump area by aligned shape masks. As expected, it has a value of 0 when the generated clump B_u is the same as B_c. Also, it has a convex function with good property, a monotone increasing function.

Our solution is to iteratively find a $\mathbf{x}_i$ that is better than the current one. This is can be done by reducing the value of $\mathbb{E}(B_c, \mathbf{x})$. We will cycle the above process until that we cannot find a better $\mathbf{x}_i$. In other words, reducing $\mathbb{E}(B_c, \mathbf{x})$ is no longer tenable. The terminated $\mathbf{x}_i$ then is aligned to the image by the above-mentioned shape recovery procedure, and set the generating residual $\mathrm{R}(\mathbf{I}, \mathrm{G}(\mathbf{I}; \boldsymbol{\mu}, \mathbf{M}))$ to $\mathbb{E}(B_c, \mathbf{x}^*)$ that, as mentioned above, will guide how to refine the modeled shape priors.

The key of our solution is to decrease $\mathbb{E}(B_c, \mathbf{x})$. We do it by finding a $\mathbf{p}$ such that $\mathbb{E}(B_c, \mathbf{x}+\mathbf{p}) < \mathbb{E}(B_c, \mathbf{x})$. To find such a $\mathbf{p}$, we approximate $\mathbb{E}(B_c, \mathbf{x}+\mathbf{p})$ as

$$\mathbb{E}(B_c, \mathbf{x}+\mathbf{p}) \approx \mathbb{E}(B_c, \mathbf{x}) + \nabla\mathbb{E}(B_c, \mathbf{x})^T\mathbf{p} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{p}^{\boldsymbol{T}}\nabla^2\mathbb{E}(B_c, \mathbf{x})\mathbf{p}, \qquad (6.4)$$

where ∇ and ∇^2 stand for the gradient and the Hessian. It is worth to note here that this way is accurate for the approximation, especially when $\|\mathbf{p}\|_2$ is small, with the approximation error of $O(\|\mathbf{p}\|_2^3)$. We then search the optimal $\mathbf{p}^*$ by

$$\mathbf{p}^* = \underset{\mathbf{p}\in\Omega_{\mathbf{p}}}{\operatorname{argmin}}\, \mathbb{E}(B_c, \mathbf{x}+\mathbf{p}), \quad \text{s.t. } \|\mathbf{p}\|_2 \le \Delta, \qquad (6.5)$$

where $\Delta > 0$ is the radius of the searching region of $\mathbf{p}^*$. We finally use the trust-region algorithm [130] to solve Eq. 6.5, because it can automatically determine Δ.

6.2.2 Refining Shape Priors

We below present the details of solving Eq. 6.1 for refining the shape priors: $\boldsymbol{\mu}$ and $\mathbf{M}$. In particular, let $\mathcal{D} = \{\mathbf{I}^j, B_c^j, \{\mathbf{s}_i^j\}_{i=1}^{N_j}\}_{j=1}^{N}$ be the training dataset, where B_c^j stands for the clump area of the example $\mathbf{I}^j$, while $\mathbf{s}_i^j$ is the ground truth of the cytoplasm i in $\mathbf{I}^j$; we use the symbol N_j to denote cytoplasms' number in this clump. For training, we first use all $\{\mathbf{s}_i^j\}$ in the training dataset to computer the mean shape and the covariance matrix, that is,

$$\boldsymbol{\mu} = \frac{1}{W}\sum_{j=1}^{N}\sum_{i=1}^{N_j} w_i^j \mathbf{s}_i^j,$$

$$\mathbf{M}_c = \frac{1}{N_c}\sum_{j=1}^{N}\sum_{i=1}^{N_j}(\mathbf{s}_i^j - \boldsymbol{\mu})(\mathbf{s}_i^j - \boldsymbol{\mu})^T, \tag{6.6}$$

where W is the sum of each template's importance $\{w_i^j\}$ that are set to 1 initially. N_c is cytoplasms' number of in $\mathcal{D}$. $\mathbf{M}$ is a matrix comprised of the first t eigenvectors of $\mathbf{M}_c$.

We next use a learning method to refine $\boldsymbol{\mu}$ and $\mathbf{M}$ that automatically adjusts the value of $\{w_i^j\}$ by reducing the average generating residual, as shown in Eq. 1. This method starts by randomly taking a training example $(\mathbf{I}^j,\ B_c^j,\ \{\mathbf{s}_i^j\}_{i=1}^{N_j})$, and then update $\{w_i^j\}_{i=1}^{N_j}$; we increase w_i^j by ℓ as long as the resulting $\mathrm{R}(\mathbf{I}^j, \mathrm{G}(\mathbf{I}^j; \boldsymbol{\mu}, \mathbf{M}))$ is decreased by the update. We update the importance one by one at each step, and once all $\{w_i^j\}_{i=1}^{N_j}$ have been updated, we take another example for the updating. The learning procedure is terminated when all data's weights have been properly assigned.

This learning algorithm has good theoretical properties: the guaranteed performance of learning and not sensitive to the updating order. To see this, we can simply understand the updating rule as a random walk [131], that is, the learned $\mathbf{w}$ moves towards the right importance $\mathbf{w}^*$ at some steps and is away from $\mathbf{w}^*$ at other steps.

It is clear that $\mathbf{w}^T(t)\mathbf{w}^*$ is growing linearly with the increasing of t while $\|\mathbf{w}(t)\|$, the learned importance at step t, is growing at most $\sqrt{t}$, meaning that this random walk is biased. Therefore, if t can be infinitely large, then $\frac{\mathbf{w}^T(t)\mathbf{w}^*}{\|\mathbf{w}(t)\|\|\mathbf{w}^*\|} \propto \frac{t}{\sqrt{t}} = \sqrt{t} = +\infty$. But $\frac{\mathbf{w}^T(t)\mathbf{w}^*}{\|\mathbf{w}(t)\|\|\mathbf{w}^*\|}$ is no more than 1, contradicted,

suggesting that t has to be finite. In other words, we will get an accurate approximation of $\mathbf{w}^*$ after finite t steps.

6.3 Experimental Evaluation

6.3.1 Experimental Setup

Datasets. This method is assessed on two cervical smear datasets, called Pap stain and H&E stain, according to their staining manner (Papanicolaou and Hematoxylin and Eosin). The Pap stain dataset [132] has 60 clumps with 316 cytoplasms, while the H&E stain dataset [100] has 160 clumps with 962 cytoplasms.

Evaluation Metrics. We use Dice Similarity Coefficient (*DSC*) and Shape Similarity Coefficient (*SSC*) to measure the segmentation performance. *DSC* is a measure of the matching extent between the ground truth and the segmentation result, the ratio of pixels' number in their intersection against the averaged pixels' number in them. *SSC* is a measure of visually implausible extent between the ground truth and the segmentation result [125, 126]. These two metrics have values in [0, 1], and have a larger value for a better segmentation result.

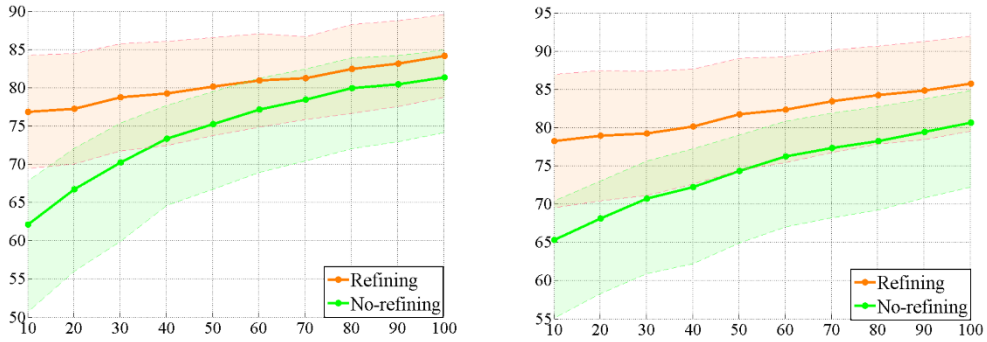
Parameters Selection. This method receives two parameters: (1) K, boundary points' number for representing shape, and (2) eigenvectors' number for computing $\mathbf{M}$. K is used as 360 here; we found that there is no considerable difference in the performance when it is set to 180 or 720. As

how to decide eigenvectors' number, it is found that more detailed shape information is modeled by using a large value of it that will enhance the representation ability [109]. This way, on the other hand, needs more resources, so we assign this parameter as 20.

6.3.2 Experimental Results

Representation Ability Improvement. We here assess the ability of our method to increase shape priors' representative ability. We hence compared our method to its variant that removes the refinement procedure. In addition, we assess the effect of shape templates' number on the segmentation accuracy. In principle, more shape templates provide more shape information, and hence the modeled shape priors are likely to have a better representative ability. To do so, we look at the varying of the segmentation performance under the setting of using shape templates from 10~100% (the interval is 10%).

Note that shape templates are randomly sampled from the dataset, and for reducing the sampling bias, we produced the segmentation results of all images in the dataset rather than the remaining images. We sampled 5 times. We here reported the mean result with the standard deviation in Fig. 6.3 in a percentage manner. It is observed that the refinement procedure works better than the variant in all cases, though the improvement gap is narrowing with the increasing of shape templates' number. This evidence suggests that the



(a) (b)

Fig. 6.3 The test segmentation performance comparison between our method against its variant that removes the shape priors refinement procedure: (a) *DSC*, (b) *SSC*, with the increasing of shape templates' number.

refinement procedure is effective to boost the segmentation performance, and it is especially effective when the shape templates are limited, demonstrating that shape priors' representative ability is improved by the refining. It is also observed that using more shape templates can boost the segmentation performance. However, the performance gains are decreasing with the increasing of the shape templates' number. This is not unusual, because the using more shape templates will narrow the accuracy boosting bound in a specified dataset. For example, it is more difficult to enhance shape priors' representative ability when we have used 100% shape templates than in other cases.

Segmentation Accuracy Improvement. We here assess the segmentation accuracy improvement of our shape mask generator. To do so, we compared our shape mask generator to LSF [98], MCL [100], MPW [120], and CF [126], four very competitive methods in overlapping cervical cytoplasms

Table 6.1 Performance comparison results, *DSC* (%), on the Pap stain dataset.

	$(0, \frac{1}{4})$	$[\frac{1}{4}, \frac{2}{4})$	$[\frac{2}{4}, \frac{3}{4})$	$[\frac{3}{4}, 1)$
LSF [98]	81.1±7.4	77.4±6.1	74.3±7.1	71.9±10.7
MCL [100]	80.1±5.2	79.7±6.4	77.3±7.4	72.7±8.7
MPW [120]	82.2±6.4	80.4±7.6	77.6±7.0	73.2±9.4
CF [126]	84.1±8.2	82.1±5.1	79.6±6.7	77.2±7.6
Ours	**85.4±4.9**	**83.9±4.7**	**82.3±6.4**	**81.0±6.3**

Table 6.2 Performance comparison results, *DSC* (%), on the H&E stain dataset.

	$(0, \frac{1}{4})$	$[\frac{1}{4}, \frac{2}{4})$	$[\frac{2}{4}, \frac{3}{4})$	$[\frac{3}{4}, 1)$
LSF [98]	80.7±8.0	75.2±7.6	72.8±8.4	69.4±11.2
MCL [100]	81.3±6.4	76.7±8.2	72.2±8.2	70.7±9.3
MPW [120]	81.3±7.2	79.1±9.2	74.1±8.3	71.6±10.4
CF [126]	83.2±7.1	81.3±8.3	79.7±8.4	75.2±9.4
Ours	**84.6±5.4**	**83.4±5.2**	**82.1±6.4**	**80.7±7.2**

MPW belongs to intensity-based methods; it is developed by the watershed algorithm. And LSF, MCL, and CF are all belong to shape priors-based. Particularly, LSF assumes cytoplasms' shape to be elliptical for modeling shape priors, but MCL and CF match shape templates for modeling shape priors. Their results are produced by either re-running their codes obtained from the authors or reproducing their recommend implementations.

Table 6.1 and 6.2 present the *DSC* results of Pap stain and H&E stain datasets, and Table 6.3 and 6.4 present the *SSC* results, both in a percentage

Table 6.3 Performance comparison results, *SSC* (%), on the Pap stain dataset.

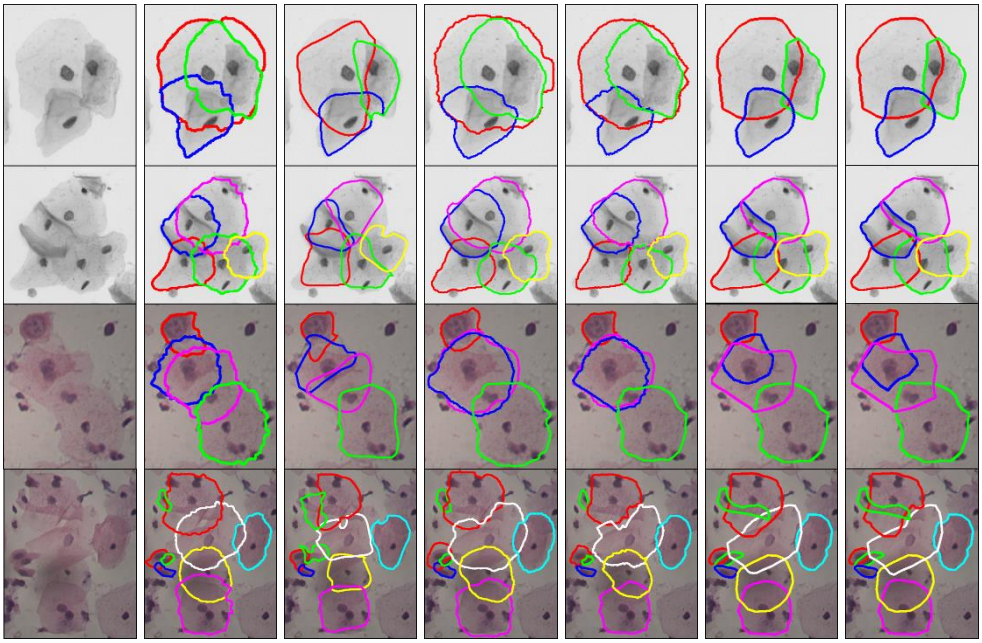
	$(0, \frac{1}{4})$	$[\frac{1}{4}, \frac{2}{4})$	$[\frac{2}{4}, \frac{3}{4})$	$[\frac{3}{4}, 1)$
LSF [98]	77.8±8.2	77.1±8.4	75.6±8.1	73.2±8.4
MCL [100]	81.1±7.5	79.6±7.9	75.2±8.9	74.1±10.3
MPW [120]	82.4±7.1	80.1±7.3	77.2±8.5	74.7±9.2
CF [126]	86.3±5.9	84.1±6.7	79.7±7.4	75.2±8.7
Ours	**87.8±5.4**	**86.1±5.7**	**83.8±6.3**	**81.7±6.8**

Table 6.4 Performance comparison results, *SSC* (%), on the H&E stain dataset.

	$(0, \frac{1}{4})$	$[\frac{1}{4}, \frac{2}{4})$	$[\frac{2}{4}, \frac{3}{4})$	$[\frac{3}{4}, 1)$
LSF [98]	78.1±8.2	80.0±8.7	75.6±9.8	71.4±10.1
MCL [100]	83.4±8.7	81.2±9.0	76.1±10.7	72.1±11.2
MPW [120]	84.3±9.7	82.1±10.4	77.4±10.7	74.8±10.4
CF [126]	86.7±7.1	84.2±7.7	80.1±8.4	76.7±8.2
Ours	**88.4±7.2**	**86.4±7.4**	**83.2±7.6**	**81.3±7.4**

Manner (5-fold cross-validation). We reported them according to the overlapping degree (4 degrees), to evaluate methods' effectiveness in different overlapping cases. We define overlapping degree by using the length ratio of the occluded boundary against the whole boundary. It is observed from these tables that the proposed method produced more accurate results than all compared methods.

Qualitative Results. Fig. 6.4 shows four visual results for the qualitative comparison. These examples all face the intensity deficiency issue. It is



(a) (b) (c) (d) (e) (f) (g)

Fig. 6.4 Visual results for the qualitative comparison: (a) input images, (b) LSF [98], (c) MCL [100], (d) MPW [120], (e) CF [126], (f) ours, and (g) the ground truth, sampled from the Pap stain dataset (the top two) and the H&E stain dataset (the bottom two), respectively; images' size is scaled for better viewing.

observed from Fig. 6.4 that the proposed shape mask generator yields better results than other methods; visually implausible results are substantially reduced. Our method in some cases can even produce results that are comparable with human annotations.

6.4 Closing Remarks

The idea of shape mask generator is to improve the shape pirors' representation ability by refining them, for segmenting overlapping cytoplasms of cervical cells. We hence first model shape priors by using

shape templates, and then use the modeled shape priors to estimate cytoplasms' shape mask. Next, we refine the modeled shape priors by reducing the discrepancy between the segmentation results and the ground truth. The proposed refinement procedure has good theoretical properties, being able to guarantee the representation ability of shape priors. We assessed the proposed shape mask generator on two datasets, with positive results showing the effectiveness, outperforming existing methods and substantially reducing visually implausible segmentation results.

Chapter 7

Conclusion and Future Works

This chapter aims at ending this thesis by presenting the conclusion and discussing future works. The main conclusion is that the proposed five methods are effective to address the identified three key problems according to the extensive empirical evidence, consistently outperforming existing methods, and thus have great potential to advance deep networks in medical image segmentation. Future works will focus mainly on further improving these methods and investigating other relevant issues of solving these three key problems.

7.1 Conclusion

Medical image segmentation, making it possible to measure object-level information, underpins a huge range of medical applications, playing a significant role in smart health. This task, however, is not trivial. Recent advances are achieved by deep networks that have shown remarkable success, but there are still several key problems to apply them to practical medical image segmentation tasks.

We hence identify three key problems to advance deep networks in medical image segmentation tasks. The first problem is to alleviate the burden

on training data collection which is time-consuming, or even prohibitively expensive, and thus hinders the practical usage of deep networks, posing challenges to the training. The second problem is to reduce the unaffordable GPU memory consumption of learning 3D features, which limits the applicability of deep networks. The last problem is to leverage shape priors for further improving the results of deep networks.

To alleviate the burden on collecting training data, we proposed two simple and effective techniques: selection learning and adversarial redrawing, by selectively learning from external data and developing unsupervised training procedures, respectively. These two techniques make it very cheap and more ready to use deep networks, increasing their usability to real-world medical applications.

To reduce the GPU memory consumption of learning 3D features, we proposed surface projection that allows 2D networks to learn 3D features. This method lowers the GPU requirement of using deep networks, and many practical applications thus can be solved by deep networks, prompting their real usage.

To leverage shape priors, we proposed shape constructing and shape mask generator, by modeling shape priors from contour fragments and learning to refine them, respectively. We applied them to overlapping cervical cytoplasms segmentation tasks motivated by screening cervical cancer, and showed that they can compensate intensity deficiency, being able to segment

visually indistinguishable occluded boundary parts and substantially reducing visually implausible results.

We assessed these five methods on publicly available datasets with extensive and comprehensive experiments. The obtained experimental results show that these methods are effective, consistently outperforming existing methods. We also evaluated the key components of these methods by conducting ablation studies, identifying what makes these methods work and demonstrating our interpretation of these methods is cautious. It hence may be safe to conclude that these methods have great potential to advance deep networks in medical image segmentation.

7.2 Future Works

There are still several works to be studied in the future, including three main issues: (1) applying them to other tasks, (2) integrating them into a unified framework, and (3) further improvement of them. These three key issues play an important role to push the proposed methods forward, being able to have clinical significance. We also hope that by solving them we can fill some knowledge gap in this field.

In future work, we first will focus on applying them to other medical segmentation tasks. As shown in the text, the segmentation tasks we studied in this thesis are general, and the proposed methods work better than existing methods. These methods hence might be able to be applied to other medical

image segmentation applications with similar problems, with perhaps appropriate revision or adjustment if necessary.

Then we hope to integrate these methods into a unified framework. These methods in this thesis are proposed for addressing only one of these three problems, but these problems in practice are arising altogether and also maybe with other problems. Therefore, for real usage, we have to combine these methods into a new technique that can address all these three problems, for maximally boosting the performance.

We finally will devote ourselves to further improvement on these five methods. For selective learning, we look forward to seeing further theoretical analysis for selectively learning from external data, and further experimentation comparing methods with heterogeneous extent analysis to those with smarter learning mechanisms. In order to improve adversarial redrawing, we will study to assess network's output without human annotations, and our idea for this is to exploit domain knowledge as segmentation constraints into the training process.

For surface projection, we hope to integrate the surface fusing procedure into the learning mechanism such that more decision processes can be directly learned. Finally, to improve shape constructing and shape mask generator for better modeling shape priors, we plan to study the relationship between objects, the relationship between cytoplasms and nuclei for example for segmenting overlapping cervical cytoplasms. It is possible to generate line

segments by using the gradients of intensity information and then use them as internal fragments. This would be more effective to leverage shape priors.

References

1. Litjens, G., Kooi, T., Bejnordi, B., *et al.*: A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88 (2017)
2. Shen, D., Wu, G. and Suk, H.: Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19, 221–248 (2017)
3. Kakeya, H., Okada, T. and Oshiro, Y.: 3D U-JAPA-Net: Mixture of convolutional networks for abdominal multi-organ CT segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 426–433 (2018)
4. Song, Y., Yu, Z., Zhou, T., *et al.*: Learning 3D features with 2D CNNs via surface projection for CT volume segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 176–186 (2020)
5. Huang, R., Zheng, Y., Hu, Z., *et al.*: Multi-organ segmentation via co-training weight-averaged models from few-organ datasets. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 146–155 (2020)
6. Isensee, F., Jaeger, P., Kohl, S., *et al.*: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 1–9 (2020)

7. WHO/ICO: Information centre on hpv and cervical cancer (hpv information centre), human papillomavirus and related diseases report in china. Available at: *www.who.int/hpvcentre* (2013)
8. Saslow, D., Solomon, D., Lawson, H., *et al.*: American cancer society, American society for colposcopy and cervical pathology, and American society for clinical pathology screening guidelines for the prevention and early detection of cervical cancer. *CA: A Cancer Journal for Clinicians*, 62, 147–172 (2012)
9. Mansour, Y., Mohri, M., Suresh, A., *et al.*: A theory of multiple-source adaptation with limited target labeled data. *arXiv preprint*, arXiv:2007.09762 (2020)
10. Cortes, C., Mohri, M., Suresh, A., *et al.*: Multiple-source adaptation with domain classifiers. *arXiv preprint*, arXiv:2008.11036 (2020)
11. Konstantinov, N., Frantar, E., Alistarh, D., *et al.*: On the sample complexity of adversarial multi-source PAC learning. In: *International Conference on MachineLearning*, pp. 5416–5425 (2020)
12. Charikar, M., Steinhardt, J. and Valiant, G.: Learning from untrusted data. In: *ACM SIGACT Symposium on Theory of Computing*, pp. 47–60 (2017)
13. Qiao, M. and Valiant, G.: Learning discrete distributions from untrusted batches. *arXiv preprint*, arXiv:1711.08113 (2017)

14. Awasthi, P., Blum, A., Haghtalab, N., *et al.*: Efficient PAC learning from the crowd. In: *International Conference on Learning Theory*, pp. 127–150 (2017)
15. Hendrycks, D., Mazeika, M., Wilson, D., *et al.*: Using trusted data to train deep networks on labels corrupted by severe noise. *arXiv preprint*, arXiv:1802.05300 (2018)
16. Han, B., Yao, Q., Yu, X., *et al.*: Co-teaching: Robust training of deep neural net-works with extremely noisy labels. In: *Advances in Neural Information Processing Systems*, pp. 8536–8546 (2018)
17. Ghadikolaei, H., Ghauch, H., Fischione, C., *et al.*: Learning and data selection in big datasets. In: *International Conference on Machine Learning*, pp. 2191–2200 (2019)
18. Jain, A. and Orlitsky, A.: A general method for robust learning from batches. *arXivpreprint*, arXiv:2002.11099 (2020)
19. Zhang, C., Yao, Y., Liu, H., *et al.*: Web-supervised network with softly update-drop training for fine-grained visual classification. In: *AAAI Conference on Artificial Intelligence*, pp. 12781–12788 (2020)
20. Zhang, C., Yao, Y., Shu, X., *et al.*: Data-driven meta-set based fine-grained visual classification. *arXiv preprint*, arXiv:2008.02438 (2020)
21. Bugallo, M., Elvira, V., Martino, L., *et al.*: Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34 (4), 60–79 (2017)

22. Katharopoulos, A. and Fleuret, F.: Not all samples are created equal: Deep learning with importance sampling. In: *International Conference on Machine Learning*, pp. 2525–2534 (2018)
23. Sun, Y., Kamel, M., Wong, A., *et al.*: Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40 (12), 3358–3378 (2007)
24. Liu, Q., Ihler, A. and Fisher, J.: Boosting crowdsourcing with expert labels: Local vs. global effect. In: *International Conference on Information Fusion*, pp. 9–14 (2015)
25. Malisiewicz, T., Gupta, A. and Efros, A.: Ensemble of exemplar-SVMs for object detection and beyond. In: *International Conference on Computer Vision*, pp. 89–96 (2011)
26. Dumitrache, A., Aroyo, L. and Welty, C.: Crowdsourcing ground truth for medical relation extraction. *arXiv preprint*, arXiv:1701.02185 (2017)
27. Yan, K., Cai, J., Zheng, Y., *et al.*: Learning from multiple datasets with heterogeneous and partial labels for universal lesion detection in CT. *IEEE Transactions on Medical Imaging*, 43 (2), 3792–3804 (2020)
28. Luo, L., Yu, L., Chen, H., *et al.*: Deep mining external imperfect data for chest X-ray disease screening. *IEEE Transactions on Medical Imaging*, 39 (6), 3583–3594 (2020)

29. Ren, M., Zeng, W., Yang, B., *et al.*: Learning to reweight examples for robust deep learning. In: *International Conference on Machine Learning*, pp. 4334–4343 (2018)
30. Konstantinov, N. and Lampert, C.: Robust learning from untrusted sources. In: *International Conference on Machine Learning*, pp. 3488–3498 (2019)
31. Lin, T., Goyal, P., Girshick, R., *et al.*: Focal loss for dense object detection. In: *IEEE International Conference on Computer Vision*, pp. 2980–2988 (2017)
32. Song, H., Kim, M. and Lee, J.: Selfie: Refurbishing unclean samples for robust deep learning. In: *International Conference on Machine Learning*, pp. 5907–5915 (2019)
33. Nandwani, Y., Pathak, A. and Singla, P.: A primal-dual formulation for deep learning with constraints. In: *Advances in Neural Information Processing Systems*, pp. 171–180 (2019)
34. Gibson, E., Giganti, F., Hu, Y., *et al.*: Automatic multi-organ segmentation on abdominal CT with dense v-networks. *IEEE Transactions on Medical Imaging*, 37 (8), 1822–1834 (2018)
35. Landman, B., Xu, Z., Eugenio, I., *et al.*: MICCAI multi-atlas labeling beyond the cranial vault–workshop and challenge (2015)
36. Roth, H., Farag, A., Turkbey, E., *et al.*: Data from pancreas-CT. *The cancer imaging archive* (2015)

37. Ronneberger, O., Fischer, P. and Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241 (2015)
38. Ulyanov, D., Vedaldi, A. and Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. *arXiv preprint*, arXiv:1607.08022 (2016)
39. Xu, B., Wang, N., Chen, T., *et al.*: Empirical evaluation of rectified activations in convolutional network. *arXiv preprint*, arXiv:1505.00853 (2015)
40. Srivastava, N., Hinton, G., Krizhevsky, A., *et al.*: Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15 (1),1929–1958 (2014)
41. Kingma, D. and Ba, J.: Adam: A method for stochastic optimization. *arXivpreprint*, arXiv:1412.6980 (2014)
42. Long, J., Shelhamer, E. and Darrell, T.: Fully convolutional networks for semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015)
43. He, K., Zhang, X., Ren, S., *et al.*: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)

44. Cicek, O., Abdulkadir, A., Lienkamp, S., *et al.*: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432 (2016)
45. Huang, J., Dong, Q., Gong, S., *et al.*: Unsupervised deep learning by neighbourhood discovery. *arXiv preprint*, arXiv:1904.11567 (2019)
46. Van, A., Achterberg, H., Vernooij, M., *et al.*: Transfer learning for image segmentation by combining image weighting and kernel learning. *IEEE Transactions on Medical Imaging*, 38 (1), 213–224 (2018)
47. Sun, R., Zhu, X., Wu, C., *et al.*: Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In: *IEEE International Conference on Computer Vision*, pp. 4360–4369 (2019)
48. Wilson, G. and Cook, D.: A survey of unsupervised deep domain adaptation. *arXiv preprint*, arXiv:1812.02849 (2018)
49. Chen, C., Dou, Q., Chen H., *et al.*: Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation. In: *AAAI Conference on Artificial Intelligence*, pp. 865–872 (2019)
50. Tajbakhsh, N., Jeyaseelan, L., Li, Q., *et al.*: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *arXiv preprint*, arXiv:1908.10454 (2019)

51. Ouyang, C., Kamnitsas, H., Biffi, C., *et al.*: Data efficient unsupervised domain adaptation for cross-modality image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 669–677 (2019)
52. Kervadec, H., Dolz, J., Tang, M., *et al.*: Constrained-CNN losses for weakly supervised segmentation. *Medical Image Analysis*, 54, 88–99 (2019)
53. Cai, J., Tang, Y., Lu, L., *et al.*: Accurate weakly-supervised deep lesion segmentation using large-scale clinical annotations: Slice-propagated 3D mask generation from 2D recist. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 396–404 (2018)
54. Lee, H., Tang, Y., Tang, O., *et al.*: Semi-supervised multi-organ segmentation through quality assurance supervision. *arXiv preprint*, arXiv:1911.05113 (2019)
55. Zhou, Y., Wang, Y., Tang, P., *et al.*: Semi-supervised 3D abdominal multi-organ segmentation via deep multi-planar co-training. In: *IEEE Winter Conference on Applications of Computer Vision*, pp. 121–140 (2019)
56. Xia, X. and Kulis, B.: W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint*, arXiv:1711.08506 (2017)

57. Kanezaki, A.: Unsupervised image segmentation by backpropagation. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1543–1547 (2018)
58. Ji, X., Henriques, J. and Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: *IEEE International Conference on Computer Vision*, pp. 9865–9874 (2019)
59. Goodfellow, I., Pouget-Abadie, J., Mirza, M., *et al.*: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014)
60. Wu, Z., Xiong, Y., Yu, S., *et al.*: Unsupervised feature learning via non-parametric instance discrimination. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742 (2018)
61. Caron, M., Bojanowski, P., Joulin, A., *et al.*: Deep clustering for unsupervised learning of visual features. In: *European Conference on Computer Vision*, pp. 132–149 (2018)
62. Bojanowski, P. and Joulin, A.: Unsupervised learning by predicting noise. *arXiv preprint*, arXiv:1704.05310 (2017)
63. Noroozi, M., Pirsiavash, H. and Favaro, P.: Representation learning by learning to count. In: *IEEE International Conference on Computer Vision*, pp. 5898–5906 (2017)

64. Zhang, R., Isola, P. and Efros, A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067 (2017)
65. Wang, X., He, K. and Gupta, A.: Transitive invariance for self-supervised visual representation learning. In: *IEEE International Conference on Computer Vision*, pp. 1329–1338 (2017)
66. Bleyer, M., Rother, C. and Kohli, P.: Surface stereo with soft segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1570–1577 (2010)
67. Liu, C., Kohli, P. and Furukawa, Y.: Layered scene decomposition via the occlusion-CRF. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 165–173 (2016)
68. Chen, M., Artieres, T. and Denoyer, L.: Unsupervised object segmentation by redrawing. In: *Advances in Neural Information Processing Systems*, pp. 12726–12737 (2019)
69. Bielski, A. and Favaro, P.: Emergence of object segmentation in perturbed generative models. In: *Advances in Neural Information Processing Systems*, pp. 7256–7266 (2019)
70. Radford, A., Luke, M. and Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint*, arXiv:1511.06434 (2015)

71. Song, Y., Zhou, T., Teoh, J., *et al.*: Unsupervised learning for CT image segmentation via adversarial redrawing. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 309–320 (2020)

72. Roth, H., Shen, C., Oda, H., *et al.*: Deep learning and its application to medical image segmentation. *Medical Imaging Technology*, 36 (2), 63–71 (2018)

73. Roth, H., Shen, C., Oda, H., *et al.*: A multi-scale pyramid of 3D fully convolutional networks for abdominal multi-organ segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 471–425 (2018)

74. Chen, J., Yang, L., Zhang, Y., *et al.*: Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In: *Advances in Neural Information Processing Systems*, pp. 3036–3044 (2016)

75. Christ, P., Elshaer, M., Ettlinger, F., *et al.*: Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 415–423 (2016)

76. Cai, J., Lu, L., Xie, Y., *et al.*: Improving deep pancreas segmentation in CT and MRI images via recurrent neural contextual learning and direct loss function. *arXivpreprint*, arXiv:1707.04912 (2017)

77. Novikov, A., Major, D., Wimmer, M., *et al.*: Deep sequential segmentation of organs in volumetric medical scans. *IEEE Transactions on Medical Imaging*, 38 (5), 1207–1215 (2018)

78. Li, X., Chen, H., Qi, X., *et al.*: H-DenseUNet: Hybrid densely connected U-Net for liver and tumor segmentation from CT volumes. *IEEE Transactions on Medical Imaging*, 37 (12), 2663–2674 (2018)

79. Xia, Y., Xie, L., Liu, F., *et al.*: Bridging the gap between 2d and 3d organ segmentation with volumetric fusion net. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 445–453 (2018)

80. Yu, Q., Xia, Y., Xie, L., *et al.*: Thickened 2D networks for 3D medical image segmentation. *arXiv preprint*, arXiv:1904.01150 (2019)

81. Ambellan, F., Tack, A., Ehlke, M., *et al.*: Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: data from the osteoarthritis initiative. *Medical Image Analysis*, 52, 109–118 (2019)

82. Wang, Y., Zhou, Y., Shen, W., *et al.*: Abdominal multi-organ segmentation with organ-attention networks and statistical fusion. *Medical Image Analysis*, 55, 88–102 (2019)

83. Wang, Z. and Wang, G.: Tri-planar convolutional neural network for automatic liver and tumor image segmentation. *International Journal of Performability Engineering*, 14 (12), 3151–3158 (2019)
84. Li, Y., Zhu, Z., Zhou, Y., *et al.*: Volumetric medical image segmentation: A 3D deep coarse-to-fine framework and its adversarial examples. In: *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, pp. 69–91 (2019)
85. Quattoni, A., Collins, M. and Darrell, T.: Conditional random fields for object recognition. In: *Advances in Neural Information Processing Systems*, pp. 1097–1104 (2005)
86. Zheng, S., Jayasumana, S., Romera, B., *et al.*: Conditional random fields as recurrent neural networks. In: *IEEE International Conference on Computer Vision*, pp. 1529–1537 (2015)
87. Mandic, D. and Chambers, J.: Recurrent neural networks for prediction: learning algorithms, architectures and stability (2001)
88. Zhang, C., Bengio, S., Hardt, M., *et al.*: Understanding deep learning requires rethinking generalization. In: *International Conference on Learning Representations*, pp. 1–15 (2017)
89. Arpit, D., Jastrzebski, S., Ballas, N., *et al.*: A closer look at memorization in deep networks. In: *International Conference on Machine Learning*, pp. 233–242 (2017)

90. Ma, X., Wang, Y., Houle, M., *et al.*: Dimensionality-driven learning with noisy labels. In: *International Conference on Machine Learning*, pp. 3361–3370 (2018)

91. Tianwei, N., Lingxi, X., Huangjie, Z., *et al.*: Elastic boundary projection for 3D medical image segmentation. In: *IEEE International Conference on Computer Vision*, pp. 2109–2118 (2019)

92. Harandi, N., Sadri, S., Moghaddam, N., *et al.*: An automated method for segmentation of epithelial cervical cells in images of ThinPrep. *Journal of Medical Systems*, 34 (6), 1043–1058 (2010)

93. Sulaiman, S., Isa, N., Yuso, I., *et al.*: Overlapping cells separation method for cervical cell images. In: *IEEE International Conference on Intelligent Systems Design and Applications*, pp. 1218–1222 (2010)

94. Arslan, S., Ersahin, T., Cetin, R., *et al.*: Attributed relational graphs for cell nucleus segmentation in fluorescence microscopy images. *IEEE Transactions on Medical Imaging*, 32 (6), 1121–1131 (2013)

95. Yang, H. and Ahuja, N.: Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed. *Pattern Recognition*, 47 (6) 2266–2279 (2014)

96. Guan, T., Zhou, D. and Liu, Y.: Accurate segmentation of partially overlapping cervical cells based on dynamic sparse contour searching and GVF snake model. *IEEE Journal of Biomedical and Health Informatics*, 19 (4), 1494–1504 (2014)

97. Kumar, P., Happy, S., Chatterjee, S., *et al.*: An unsupervised approach for overlapping cervical cell cytoplasm segmentation. In: *IEEE International Conference on Biomedical Engineering and Sciences*, pp. 106–109 (2016)
98. Lu, Z., Carneiro, G. and Bradley, A.: An improved joint optimization of multiple level set functions for the segmentation of overlapping cervical cells. *IEEE Transactions on Image Processing*, 24 (4), 1261–1272 (2015)
99. Nosrati, M. and Hamarneh, G.: Segmentation of overlapping cervical cells: a variational method with star-shape prior. In: *IEEE International Symposium on Biomedical Imaging*, pp. 186–189 (2015)
100. Song, Y., Tan, E., Jiang, X., *et al.*: Accurate cervical cell segmentation from overlapping clumps in Pap smear images. *IEEE Transactions on Medical Imaging*, 36 (1), 288–300 (2017)
101. Tareef, A., Song, Y., Huang, H., *et al.*: Optimizing the cervix cytological examination based on deep learning and dynamic shape modeling. *Neurocomputing*, 248, 28–40 (2017)
102. Tareef, A., Song, Y., Cai, W., *et al.*: Automatic segmentation of overlapping cervical smear cells based on local distinctive features and guided shape deformation. *Neurocomputing*, 221, 94–107 (2017)
103. Song, Y., He, L., Zhou, F., *et al.*: Segmentation, splitting, and classification of overlapping bacteria in microscope images for automatic bacterial vaginosis diagnosis. *IEEE Journal of Biomedical and Health Informatics*, 21 (4) 1095–1104 (2017)

104. Park, C., Huang, J., Ji, X., *et al.*: Segmentation, inference and classification of partially overlapping nanoparticles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (3), 669–681 (2013)

105. Song, Y., Cheng, J., Ni, D., *et al.*: Segmenting overlapping cervical cell in Pap smear images. In: *IEEE International Symposium on Biomedical Imaging*, pp. 1159–1162 (2016)

106. Song, Y., Zhang, L., Chen, S., *et al.*: Accurate segmentation of cervical cytoplasm and nuclei based on multi-scale convolutional network and graph partitioning. *IEEE Transactions on Biomedical Engineering*, 62 (10), 2421–2433 (2015)

107. McNeill, G. and Vijayakumar, S.: Hierarchical procrustes matching for shape retrieval. In: *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 885–894 (2006)

108. Delong, A., Osokin, A., Isack, H., *et al.*: Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96 (1), 1–27 (2012)

109. Cootes, T., Taylor, C., Cooper, D., *et al.*: Active shape models: their training and application. *Computer Vision and Image Understanding*, 61 (1), 38–59 (1995)

110. Li, S.: Markov random field modeling in image analysis. *Springer Science & Business Media* (2009)

111. Duchi, J., Shalev, S., Singer, Y., *et al.:* Efficient projections onto the 1-ball for learning in high dimensions. In: *International Conference on Machine Learning*. pp. 272–279 (2008)

112. Sharon, E., Brandt, A. and Basri, R.: Completion energies and scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (10), 1117–1131 (2000)

113. Davey, E., Barratt, A., Irwig, L., *et al.*: Effect of study design and quality on unsatisfactory rates, cytology classifications, and accuracy in liquid-based versus conventional cervical cytology: a systematic review. *The Lancet*, 367 (9505), 122–132 (2006)

114. Kitchener, H., Blanks, R., Dunn, G., *et al.*: Automation-assisted versus manual reading of cervical cytology (MAVARIC): a randomised controlled trial. *The Lancet Oncology*, 12 (1), 56–64 (2011)

115. Guven, M. and Cengizler, C.: Data cluster analysis-based classification of overlapping nuclei in Pap smear samples. *Biomedical Engineering Online*, 13 (1), 159 (2014)

116. Schifiman, M., Castle, P., Jeronimo, J., *et al.*: Human papillomavirus and cervical cancer. *The Lancet*, 370 (9590), 890–907 (2007)

117. WHO: World cancer report, chapter 5.12, *ISBN 9283204298* (2014)

118. Plissiti, M., Vrigkas, M. and Nikou, C.: Segmentation of cell clusters in Pap smear images using intensity variation between superpixels. In: *IEEE*

International Conference on Systems, Signals and Image Processing, pp. 184–187 (2015)

119. Bfieliz, N., Crespo, J., Garcfia, M., *et al.*: Cytology imaging segmentation using the locally constrained watershed transform. In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pp. 429–438 (2011)
120. Tareef, A., Song, Y., Huang, H., *et al.*: Multi-pass fast watershed for accurate segmentation of overlapping cervical cells. *IEEE Transactions on Medical Imaging*, 37 (9), 2044–2059 (2018)
121. Lee, H. and Kim, J.: Segmentation of overlapping cervical cells in microscopic images with superpixel partitioning and cell-wise contour refinement. In: *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, pp. 63–69 (2016)
122. Kaur, S. and Sahambi, J.: Curvelet initialized level set cell segmentation for touching cells in low contrast images. *Computerized Medical Imaging and Graphics*, 49, 46–57 (2016)
123. Nosrati, M. and Hamarneh, G.: A variational approach for overlapping cell segmentation. In: *IEEE International Symposium on Biomedical Imaging Overlapping Cervical Cytology Image Segmentation Challenge*, pp. 1–2 (2014)

124. Islam, Z. and Haque, M.: Multi-step level set method for segmentation of overlapping cervical cells. In: *IEEE International Conference on Telecommunications and Photonics*, pp. 1–5 (2015)

125. Song, Y., Qin, J., Lei, L., *et al.*: Automated segmentation of overlapping cytoplasm in cervical smear images via contour fragments. In: *AAAI Conference on Artificial Intelligence*, pp. 168–175 (2018)

126. Song, Y., Zhu, L., Qin, J., *et al.*: Segmentation of overlapping cytoplasm in cervical smear images via adaptive shape priors extracted from contour fragments. *IEEE Transactions on Medical Imaging*, 38 (12), 2849–2862 (2019)

127. Kass, M., Witkin, A. and Terzopoulos, D.: Snakes: active contour models. *International Journal of Computer Vision*, 1 (4), 321–331 (1988)

128. Chan, T. and Vese, L.: Active contours without edges. *IEEE Transactions on Image Processing*, 10 (2), 266–277 (2001)

129. Li, C., Xu, C., Gui, C., *et al.*: Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 19 (12), 32–43 (2010)

130. Nocedal, J. and Wright, S.: Numerical optimization. *Springer* (2016)

131. Spitzer, F.: Principles of random walk. *Springer Science & Business Media* (2013)

132. Lu, Z., Carneiro, G., Bradley, A., *et al.*: Evaluation of three algorithms for the segmentation of overlapping cervical cells. *IEEE Journal of Biomedical and Health Informatics*, 21 (2), 441–450 (2017)

133. Koyner, J., Carey, K., Edelson, D., *et al.*: The development of a machine learning in patient acute kidney injury prediction model. *Critical Care Medicine*, 46 (7), 1070–1077 (2018)

134. Tixier, A., Hallowell, M., Rajagopalan, B., *et al.*: Application of machine learning to construction injury prediction. *Automation in Construction*, 69, 102–114 (2016)

135. López, K., Aranjuelo, N., Kabongo, L., *et al.*: Fully automatic detection and segmentation of abdominal aortic thrombus in post-operative CTA images using deep convolutional neural networks. *Medical Image Analysis*, 46, 202–214 (2018)

136. Retson, T., Besser, A., Sall, S., *et al.*: Machine learning and deep neural networks in thoracic and cardiovascular imaging. *Journal of Thoracic Imaging*, 34 (3), 192–201 (2019)

137. Tajbakhsh, N., Jeyaseelan, L., Li, Q., *et al.*: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63, 672–683 (2020)

138. Sinha, A. and Dolz, J.: Multi-scale self-guided attention for medical image segmentation. *IEEE Journal of Biomedical and Health Informatics*, 78 (12), 296–311 (2020)

139. Zhang, L., Wang, X., Yang, D., *et al.*: Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE Transactions on Medical Imaging*, 39 (7), 2531–2540 (2020)

140. Mehrtash, A., Wells, W., Tempany, C., *et al.*: Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE Transactions on Medical Imaging*, 39 (12), 3868–3878 (2020)

141. Xie, Y., Zhang, J., Lu, H., *et al.*: SESV: Accurate medical image segmentation by predicting and correcting errors. *IEEE Transactions on Medical Imaging*, 40 (1), 286–296 (2020)

142. Calisto, M. and Lai, S.: AdaEn-Net: An ensemble of adaptive 2D-3D fully convolutional networks for medical image segmentation. *Neural Networks*, 126, 76–94 (2020)

143. Yu, Q., Yang, D., Roth, H., *et al.*: C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4126–4135 (2020)

144. Sun, J., Darbehani, F., Zaidi, M., *et al.*: SAUNet: shape attentive U-Net for interpretable medical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 797–806 (2020)

145. Xia, Y., Yang, D., Yu, Z., *et al.*: Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation. *Medical Image Analysis*, 65, 101–126 (2020)

146. Valanarasu, J., Oza, P., Hacihaliloglu, I., *et al.*: Medical transformer: Gated axial-attention for medical image segmentation. *arXiv preprint*, arXiv:2102.10662 (2021)

147. Wang, S., Cao, S., Wei, D., *et al.*: LT-Net: Label transfer by learning reversible voxel-wise correspondence for one-shot medical image segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9162–9171 (2020)

148. Li, X., Yu, L., Chen, H., *et al.*: Transformation-consistent self-ensembling model for semi-supervised medical image segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 78 (4), 889–907 (2020)

149. Huang, H., Lin, L., Tong, R., *et al.*: Unet 3+: A full-scale connected U-Net for medical image segmentation. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1055–1059 (2020)

150. Xu, J., Li, M. and Zhu, Z.: Automatic data augmentation for 3D medical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 378–387 (2020)

151. Zhu, Z., Xia, Y., Shen, W., *et al.*: A 3D coarse-to-fine framework for volumetric medical image segmentation. In: *International Conference on 3D Vision*, pp. 682–690 (2018)

152. Li, Y., Zhu, Z., Zhou, Y., *et al.*: Volumetric medical image segmentation: a 3D deep coarse-to-fine framework and its adversarial examples. *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, 42 (4), 69–91 (2019)

153. Tang, M., Zhang, Z., Cobzas, D., *et al.*: Segmentation-by-detection: a cascade network for volumetric medical image segmentation. In: *IEEE International Symposium on Biomedical Imaging*, pp. 1356–1359 (2018)

154. Rickmann, A., Roy, A., Sarasua, I., *et al.*: 'project & excite' modules for segmentation of volumetric medical scans. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 39–47 (2019)

155. Novikov, A., Major, D., Wimmer, M., *et al.*: Deep sequential segmentation of organs in volumetric medical scans. *IEEE Transactions on Medical Imaging*, 38 (5), 1207–1215 (2018)

156. Milletari, F., Frei, J. and Ahmadi, S.: TOMAAT: volumetric medical image analysis as a cloud service. *arXiv preprint*, arXiv:1803.06784 (2018)

157. Roy, A., Siddiqui, S., Pölsterl, S., *et al.*: Squeeze & excite guided few-shot segmentation of volumetric images. *Medical Image Analysis*, 59, 11–28 (2020)

158. Wang, X., Han, S., Chen, Y., *et al.*: Volumetric attention for 3D medical image segmentation and detection. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 175–184 (2019)

159. Brügger, R., Baumgartner, C. and Konukoglu, E.: A partially reversible U-Net for memory-efficient volumetric image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 429–437 (2019)

160. Hesamian, M., Jia, W., He, X., *et al.*: Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of Digital Imaging*, 32 (4), 582–596 (2019)

161. Kayalibay, B., Jensen, G. and Vander, P.: CNN-based segmentation of medical imaging data. *arXiv preprint*, arXiv:1701.03056 (2017)

162. Zhou, X., Li, Y. and Liang, W.: CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 24 (2), 661–679 (2020)

163. Mortazi, A. and Bagci, U.: Automatically designing CNN architectures for medical image segmentation. In: *International Workshop on Machine Learning in Medical Imaging*, pp. 98–106 (2018)

164. Bullock, J., Cuesta, C. and Quera, A.: XNet: A convolutional neural network (CNN) implementation for medical X-ray image segmentation suitable for small datasets, *Medical Imaging*, 10953, 109531Z (2019)

165. Qiu, T., Wen, C., Xie, K., *et al.*: Efficient medical image enhancement based on CNN-FBB model. *IET Image Processing*, 13 (10), 1736–1744 (2019)

166. Liu, Y., Ma, Z., Liu, X., *et al.*: Privacy-preserving object detection for medical images with faster R-CNN. *IEEE Transactions on Information Forensics and Security*, 91 (2), 1102–1120 (2019)

167. Sundararajan, S., Sankaragomathi, B. and Priya, D.: Deep belief CNN feature representation based content based image retrieval for medical images. *Journal of Medical Systems*, 43 (6), 101–109 (2019)

168. Papanastasopoulos, Z., Samala, R., Chan, H., *et al.*: Explainable AI for medical imaging: deep-learning CNN ensemble for classification of estrogen receptor status from breast MRI. *Medical Imaging*, 11314, 113140Z (2020)

169. Xie, Y., Zhang, J., Shen, C., *et al.*: Efficiently bridging CNN and transformer for 3D medical image segmentation. *arXiv preprint*, arXiv:2103.03024 (2021)

170. Lou, A., Guan, S. and Loew, M.: DC-UNet: rethinking the U-Net architecture with dual channel efficient CNN for medical image segmentation. *Medical Imaging*, 11596 (2021)
171. Buzug, T.: Computed tomography. *Springer Handbook of Medical Technology*, 311–342 (2011)
172. Smith, R.: Is computed tomography safe. *New England Journal of Medicine*, 363 (1), 1–4 (2010)
173. Kalender, W.: X-ray computed tomography. *Physics in Medicine & Biology*, 51 (13), R29 (2006)
174. Brenner, D. and Hall, E.: Computed tomography-an increasing source of radiation exposure. *New England Journal of Medicine*, 357 (22), 2277–2284 (2007)
175. Hathcock, J. and Stickle, R.: Principles and concepts of computed tomography. *Veterinary Clinics of North America: Small Animal Practice*, 23 (2), 399–415 (1993)
176. Morris, S. and Slesnick, T.: Magnetic resonance imaging. *Visual Guide to Neonatal Cardiology*, 104–108 (2018)
177. Vlaardingerbroek, M. and Boer, J.: Magnetic resonance imaging: theory and practice. *Springer Science & Business Media* (2013)
178. Young, S.: Magnetic resonance imaging: basic principles (1987)
179. Dill, T.: Contraindications to magnetic resonance imaging. *Heart*, 94 (7), 943–948 (2008)

180. Huettel, S., Song, A. and McCarthy, G.: Functional magnetic resonance imaging (2004)

181. Gu, X., Gortler, S., and Hoppe, H.: Geometry images. In: *Annual Conference on Computer Graphics and Interactive Techniques*, pp. 355–361 (2002)

182. Gu, X., and Yau, S. Gortler, S.: Global conformal surface parameterization. In: *ACM SIGGRAPH Symposium on Geometry Processing*, pp. 127–137 (2003)