Summer 2022

CNN-Based Semantic Segmentation with Shape Prior Knowledge

Yuhang Lu

CNN-Based Semantic Segmentation with Shape Prior Knowledge

By

Yuhang Lu

Bachelor of Engineering
Chengdu University of Technology, 2013

Master of Engineering
Wuhan University, 2015

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Song Wang, Major Professor

Michael N. Huhns, Committee Member

Yan Tong, Committee Member

Lannan Luo, Committee Member

Karen Y. Smith, Committee Member

Tracey L. Weldon, Vice Provost and Dean of Graduate Studies

Acknowledgments

First and foremost, I would like to sincerely thank my advisor Prof. Song Wang who supervised me under this doctoral thesis and provided persistent help over the past five years. Not only his profound knowledge in computer vision carried me through the difficulties in research, but also his rigorous scholarship and high standard set a great example for me to follow. I am really grateful to have Prof. Wang as my PhD advisor. Being under his guidance encouraged me to be a better researcher.

I would also like to thank my dissertation committee members, Prof. Michael Huhns, Prof. Yan Tong, Prof. Lannan Luo, and Prof. Karen Smith, for their help and constructive suggestions on my work. Their time and efforts are very much appreciated. It is my honor to have them as my committee members.

I would like to extend my sincere thanks to Prof. Karen Smith, Prof. Song Wang, Jun Zhou, Sam McDorman, Deja Scott and the whole Snowvision team for their assistance at every stage of this exciting research project. It was my great pleasure to work with them.

I am truly grateful to my colleagues and friends Dr. Hongkai Yu, Dr. Dazhou Guo, Dr. Kang Zheng, Dr. Yang Mi, Dr. Hao Guo, Zhenyao Wu, Xinyi Wu, Lan Fu, Ahmed Shehab Khan, Jie Cai, Zhiyuan Li, and other labmates. They provided me with a friendly and inspiring environment to work and have fun.

Finally, I want to express my deepest appreciation to my wife and my parents. Their unconditional support and endless love are the best comfort in the journey of pursuing my Ph.D. degree.

Abstract

Semantic segmentation that aims at grouping discrete pixels into connected regions is a fundamental step in many high-level computer vision tasks. In recent years, Convolutional Neural Networks (CNNs) have made breakthrough progresses in public semantic segmentation benchmarks. The ability of learning from large-scale labeled datasets empowers them to generalize to unseen images better than traditional non-learning-based methods. Nevertheless, the heavy dependency on labeled data also limits their applications in tasks where high-quality ground truth segmentation masks are scarce or difficult to acquire. In this dissertation, we study the problem of alleviating the data dependency for CNN-based segmentation with a focus on leveraging the shape prior knowledge of objects.

Shape prior knowledge could provide rich learning-free information of object boundaries if properly utilized. However, this is not trivial for CNN-based segmentation because of its nature of pixel-wise classification. To address this problem, we propose novel methods to integrate three types of shape priors into CNN training, including implicit, explicit and class-agnostic priors. They cover from specific objects with strong prior to general objects with weak prior. To demonstrate the practical value of our methods, we present each of them within a challenging real-world image segmentation task. 1) We propose a weakly supervised segmentation method to extract curve structures stamped on cultural heritage objects, which implicitly takes advantage of the prior knowledge of their thin and elongated shape to relax the training label from pixel-wise curve mask to single-pixel curve skeleton, and outperforms fully supervised alternatives by at least 7.7% in F1 score. 2) We propose a one-shot seg-

mentation method to learn to segment anatomical structure from X-ray images with only one labeled image, which is realized by explicitly modeling the shape and appearance prior knowledge of objects into the objective function of CNNs. It performs competitively compared to state-of-the-art fully supervised methods when using a single label, and could outperform them when a human-in-the-loop mechanism is incorporated. 3) Finally, we attempt to model shape priors in a universal form that is agnostic to object classes, where the knowledge can be distilled from a few labeled samples through a meta-learning strategy. Given a base model pretrained on existing large-scale dataset, our method could adapt it to any unseen domains with the help of a few labeled images and masks. Experimental results show that our method significantly improve the performance of base models in a variety of cross-domain segmentation tasks.

Table of Contents

Acknowledgments iii

Abstract iv

List of Tables viii

List of Figures x

Chapter 1 Introduction 1

1.1 The Goal of Semantic Segmentation 2

1.2 The Problem of Existing Methods 3

1.3 Proposed Approaches 5

Chapter 2 Background 9

2.1 Convolutional Neural Network 10

2.2 Shape Prior Knowledge for Segmentation 14

Chapter 3 Literature review 19

3.1 General Semantic Segmentation 20

3.2 Weakly Supervised Semantic Segmentation 23

3.3 Few-shot Semantic Segmentation 26

Chapter 4 Weakly supervised curve structure segmentation with implicit shape prior 29

4.1 Problem Overview 30

4.2 Method 33

4.3 Experiments 39
4.4 Chapter Summary 46

Chapter 5 One-shot anatomical structure segmentation with explicit shape prior 47
5.1 Problem Overview 48
5.2 Method 51
5.3 Experiments 59
5.4 Chapter Summary 68

Chapter 6 Few-shot natural image segmentation with class-agnostic prior 71
6.1 Problem Overview 72
6.2 Method 75
6.3 Experiments 81
6.4 Chapter Summary 89

Chapter 7 Conclusion 90

Bibliography 94

List of Tables

Table 4.1 The configuration of network for Step II, where n, k, s, p stand for the number of outputs, kernel size, stride and padding size respectively. 37

Table 4.2 Precision, recall and F-measure of the proposed method and six comparison methods, averaged over 280 test images. 42

Table 5.1 Performances of CTN and seven existing methods on four datasets. 61

Table 5.2 Using more unlabeled images in training. We expand the training set of knee and phalanx from 100 to 500 images to examine our method's ability in exploiting unlabeled data. Both cases use only one exemplar. 66

Table 5.3 Ablation study. We remove one of three losses each time and re-train the model. 67

Table 6.1 The IoU (%) results of using our method to fine-tune various prototypical models on four cross-domain FSS tasks. 80

Table 6.2 The IoU (%) results of using three different methods to fine-tune PANet and PPNet on four cross-domain FSS tasks. 83

Table 6.3 The IoU (%) results of using our method to fine-tune PPNet on two in-domain FSS datasets - PASCAL-5^i and COCO-20^i. 84

Table 6.4 The IoU (%) results of using different combinations of loss terms in Eq.(6.10) to fine-tune PPNet for 1-shot segmentation. 85

Table 6.5 The IoU (%) results of our method with and without the uncertainty weight in fine-tuning PPNet for 1-shot segmentation. 87

Table 6.6 The IoU (%) results of using random support images in fine-tuning PPNet for 5-shot segmentation. 88
Table 6.7 The IoU(%) results on seen and unseen images. We fine-tune PPNet for 1-shot segmentation on the “seen“ subset of each dataset, and test on the “unseen” subset. 88

List of Figures

Figure 1.1 An illustration of the difference between semantic, instance and panoptic segmentation [39]. 3

Figure 2.1 The network architecture of AlexNet. 11

Figure 2.2 Objects with implicit and explicit shape priors. 15

Figure 3.1 An illustration of patch-based image segmentation [16]. 20

Figure 3.2 An illustration of the FCN segmentation framework [54]. 21

Figure 3.3 The network architecture of UNet [71]. 22

Figure 3.4 An illustration of Curve-GCN [50]. 23

Figure 4.1 Five unearthed pottery sherds dating to the Woodland period of Southeastern North America. (a) RGB images. (b) Depth images where intensity indicates the depth. 30

Figure 4.2 An illustration of using low-level methods for curve structure segmentation. Serious erosion in the red square leads to very low contrast in the depth image, and low-level method, such as DoG (Difference of Gaussian), may produce very poor segmentation. 31

Figure 4.3 FCN used for skeleton detection. 34

Figure 4.4 Example results after each step of the proposed method. 36

Figure 4.5 An illustration of design matching. (a) Thinned curve structures U segmented on the sherd. (b) Thinned full design V. (c) Partial matching between U to V with minimal Chamfer distance. Original design illustration copyrighted by Frankie Snow. Used with permission. 39

Figure 4.6 Examples of the curve structure segmentation result from the proposed method and six comparison methods. 41

Figure 4.7 Sample segmentation results of the proposed method with modifications to each step. (a) Input depth image. (b) Segmentation result after modifying Step I. (c) Segmentation result after removing Step II. (d) Segmentation result after modifying Step III. (e) Segmentation result of the proposed method without any modification. (f) Ground truth segmentation. 44

Figure 4.8 CMC curves of the proposed method and three comparison methods. 45

Figure 5.1 An overview of CTN. CTN could learn to segment the anatomical structure accurately from only one exemplar and a set of unlabeled images. In contrast, fully supervised methods such as DeepLab [9] will fail when training with insufficient labeled images. 49

Figure 5.2 Contour Transformer Network. CTN is trained to fit a contour to the object boundary by learning from one exemplar. In training, it takes a labeled exemplar and a set of unlabeled images as input. After going through a CNN encoder and five GCN contour evolution blocks, it outputs the predicted contour. We train the network using three one-shot losses (*i.e.*, contour perceptual loss, contour bending loss and edge loss), aiming to let the predicted contour have similar contour features with the exemplar. 52

Figure 5.3 Network Architecture of GCN blocks. CTN uses five cascaded GCN blocks to model the contour evolution behavior. They take image features along the contour and an adjacency matrix that represents vertex connections as input, and predict point-wise offsets to update the contour. Their architectures are identical, but weights are not shared. 54

Figure 5.4 Human-in-the-loop. Given a *red* predicted contour (a), the annotator corrects its wrong parts with *green* curves (b). For each corrected contour segment, we find two points in the predicted contour, closest to its start and end (c), then each predicted point between the two points are assigned to the closest corrected point (d). This prevents the point correspondence to be scattered. 57

Figure 5.5 Segmentation results of four example images. The boundaries of ground truth segmentations (the green lines) are drawn for comparison. 62

Figure 5.6 Visualization of the contour evolution process. The red lines are the contours after each GCN block in CTN. It shows how CTN gradually moves the initial contour to the correct location. . 64

Figure 5.7 Using different number of human corrections to finetune the one-shot model. We test the performance of the human-in-the-loop mechanism with 0, 10%, 25% and 100% corrected training samples, respectively ("0" means no finetuning). Our performance with 25% training samples generally outperforms DeepLab using 100% samples. 66

Figure 5.8 Using different loss weights to train CTN on the hip dataset. Based on the original setting $\lambda_1 = 1$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.1$, we change one of them each time and fix the other two. 68

Figure 6.1 The cross-domain FSS problem. θ is a prototypical FSS model trained on natural images. When testing on medical images, its discriminability will decrease drastically due to the feature distribution discrepancy. We propose a method to fine-tune θ on the given support and query sets to bridge the domain gap. . . 73

Figure 6.2 The training pipeline. We illustrate with an example of 1-way 1-shot segmentation. We first generate the fine-grained support prototypes, and then use them to predict the query mask. With the query mask, we generate the query prototypes and compare against the support ones to minimize the prototype contrastive loss L_{pc}. Besides, the support GT mask is employed to regularize the training through a cross-entropy loss L_{ce}. Finally, a boundary length loss L_{bd} is employed to penalize irregular regions in the query mask. 76

Figure 6.3 Qualitative results. By comparing the last two columns in this figure, we can observe the improvement brings by fine-tuning using our method for PPNet. 82
Figure 6.4 The curves of intra-, inter-class prototype distance, and test IoU v.s. fine-tuning steps. 86

Chapter 1

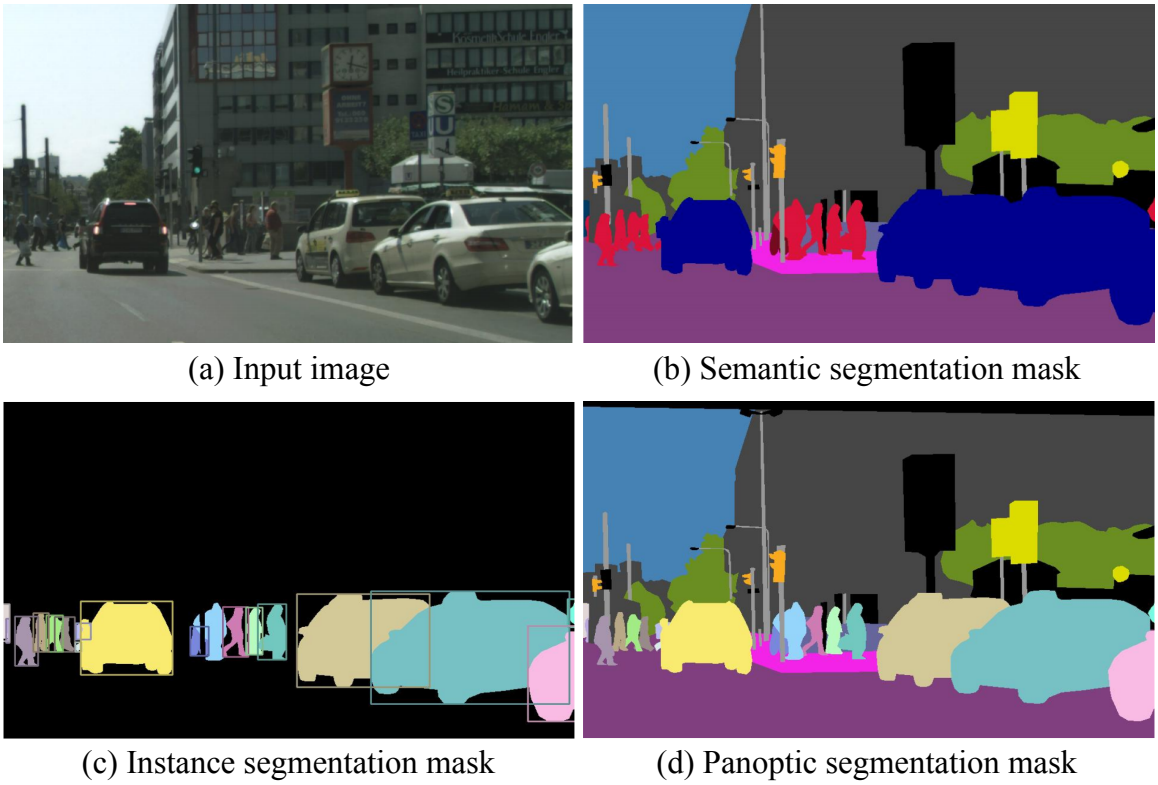
Introduction

1.1 The Goal of Semantic Segmentation

Semantic segmentation is a fundamental problem in image analysis and computer vision. By grouping meaningless pixels into meaningful regions, it serves as the bridge from low-level image processing to high-level image understanding. Without accurate image segmentation results, many downstream computer vision tasks, such as object localization, object measurement and scene understanding, will lose the foundation [85]. Technically, the goal of semantic segmentation is to assign a label to each pixel of an image that corresponds to the object class of the pixel. It can be seen as a task of pixel-wise image classification.

The problem of semantic segmentation is closely related to instance and panoptic segmentation. The difference between semantic and instance segmentation is that, the former treats multiple objects of the same class as a single entity, while the latter treats multiple objects of the same class as distinct individual instances. In another word, semantic segmentation only interests in the class of the pixel, while instance segmentation also interests in which object instance the pixel is on. Panoptic segmentation combines both the information of semantic and instance segmentation, by which not only requiring the label of every pixel, but also distinguishing individual instances. In Figure 1.1, we illustrate the relationship between three tasks with an urban scene image from the CityScape dataset [1]. We can see that the semantic segmentation mask labels all vehicles with the same color; the instance segmentation mask needs to distinguish different vehicles, and interests in only the classes of pedestrian and vehicle, but not every pixel; and the panoptic segmentation mask contains the semantic and instance information of every pixel. In this dissertation, we focus on the problem of semantic segmentation.

[1] https://www.cityscapes-dataset.com



(a) Input image (b) Semantic segmentation mask

(c) Instance segmentation mask (d) Panoptic segmentation mask

Figure 1.1: An illustration of the difference between semantic, instance and panoptic segmentation [39].

1.2 The Problem of Existing Methods

Over the past decades, a broad range of image segmentation methods have been proposed – from early methods, such as thresholding [66], region growing [63], pixel clustering [47], Watersheds [62], to more advanced methods, such as Active Contour Model (ACM) [36], Level Sets [98] and Graph Cuts [4]. These methods mostly use handcrafted features and treat each image as an individual case. It is difficult for handcrafted rules to be generalized to all different scenarios and images. Therefore, these non-learning-based methods often suffer from noises, background clutter, object occlusions, and many other disturbances [20].

More recently, Convolutional Neural Networks (CNNs) have yielded a new generation of image segmentation models with remarkable performance improvements on

popular benchmarks [54, 71, 112, 9, 22, 89]. CNNs are parameter-intensive models composed by multiple specific layers, such as convolutional layers, pooling layers, fully connected layers, etc [27]. The query image goes through every layer in the model and outputs a predicted segmentation mask. The model is typically trained by minimizing a loss function that measures the difference between the predicted segmentation result and the ground truth result. Instead of focusing on individual images, CNN models optimize the loss aggregated over a large-scale training set, which makes them more robust and more accurate than non-learning-based methods [60].

Despite the high performance, training an effective CNN segmentation model usually requires large, representative, and high quality annotated datasets [86]. Without a carefully-labeled and perfect-sized dataset, the training process is prone to overfit or underfit, thus degrading the generalization ability of the model [103]. However, existing large-scale semantic segmentation datasets, including PASCAL VOC [25], MS COCO [92] and CityScape [17], are predominantly focused on the domain of natural image. For less popular segmentation tasks, such as cultural heritage object, remote sensing and medical image segmentation, there usually exists no off-the-shelf public datasets. Meanwhile, manually annotating pixel-level segmentation masks is extremely time-consuming and sometimes requires domain expertise in specific fields. Therefore, obtaining enough ground truth labels for model training has often been an obstacle that prevents powerful CNN models from addressing real-world image segmentation problems.

In contrast to data-hungry CNN models, human annotators can learn to segment novel objects from all kinds of annotations, including scarce annotations where only limited annotated data is available, and weak annotations where the training data has only sparse annotations, noisy annotations, or image-level annotations [86]. A crucial factor that may cause this gap of learning ability is that – human could take advantage of different types of **prior knowledge**, such as shape prior, appearance prior

and spatial location prior, while conventional CNN-based methods could not [65]. Actually, utilizing shape prior knowledge in segmentation models has been proven useful in classic graph-based [15] and contour-based segmentation methods [79]. However, the architecture of CNNs determines that they are good at capturing local features of pixels, but are not effective in discovering the global structure of objects. In typical CNNs, segmentation results are yielded by pixel-wise classification, thus making global prior knowledge such as object shape difficult to be integrated into the framework [65]. Fully-supervised trained CNNs are able to recognize objects with similar shapes because the shape information is implicitly encoded into the model, but this is realized at the cost of large-scale training data and intensive model parameters. Once the shape prior knowledge could be explicitly incorporated into CNN frameworks, we expect that the heavy dependence on training data can be largely relieved.

1.3 Proposed Approaches

In this dissertation, we focus on the problem of utilizing shape prior knowledge in CNNs to achieve annotation-efficient learning, and propose three novel approaches for three different types of shape prior knowledge, respectively. For objects with **implicit shape prior**, the general type of their shape is known, such as curve-like, circle-like or lines. But we are not able to explicitly describe their shapes with a template mask or contour due to the shape variation. In the first work, we show an example of implicitly utilizing mid-level shape prior to simplify the object segmentation [57]. For objects with **explicit shape prior**, the object to be segmented usually have a certain shape that can be represented by a template, which could serve as explicit guidance in CNN training. In the second work, we propose a method to explicitly integrating high-level shape prior into CNN model optimization to enable one-shot learning, namely learning a segmentation model from only one labeled image [93].

Moreover, we try to incorporate a general form of shape prior into CNNs to realize

few-shot segmentation of arbitrary object. The shape and appearance of objects in natural images are usually difficult to be explicitly represented by a single template due to the change of viewing perspective and object pose. Therefore, we simply represent the prior knowledge of these objects in the form of segmentation masks, and expect CNNs could learn to segment them from very few labels. Since this form of shape prior is agnostic to object classes, we name it as "**class-agnostic shape priors**". To avoid the model overfitting on a small number of samples, we resort to meta-learning and existing segmentation datasets to pre-train a base model, and then fine-tune it on the images of the target object class, which is realized by a newly proposed semi-supervised transductive fine-tuning method.

Each type of shape prior knowledge is explored in the context of a representative real-world image segmentation task, and these three tasks‘ are *curve structure segmentation in cultural heritage objects*, *anatomical structure segmentation in medical images*, and *few-shot natural object segmentation*. We next briefly introduce these three tasks and our proposed approaches.

Curve structure segmentation with implicit shape prior

In the first work, we explore the application of CNN segmentation in extracting curve structures stamped on cultural heritage objects. Curve structures on the depth images are usually very weak and with blurry boundaries, which make them very hard to be recognized for both automatic segmentation and manually annotation. Taking advantage of the shape prior of the thin and elongated structures, we proposed to first extract curve skeletons instead of curve regions, because curve skeletons possess more discriminative image features and reveal the structure location. With this motivation, we proposed a CNN architecture to extract curve skeletons from depth images and help restore the full curve structures. We first train a Fully Convolutional Network (FCN) to extract the skeleton of curve structures, and estimate a scale value at each

skeleton pixel. This scale value reflects the curve width at the corresponding skeleton pixel. Next, we propose a dense prediction network to refine the curve skeletons in a pixel-by-pixel manner. Finally, we develop an adaptive thresholding algorithm to achieve the final segmentation of curve structures with width by considering the estimated scale values. In this way, we simplify the annotation target from pixel-wise curves to single-pixel skeletons, and implicitly realize weakly supervised segmentation. Moreover, experiment results show that our method outperforms other CNN-based methods by at least 7.7% in F1 score.

Anatomical structure segmentation with explicit shape prior

In the second work, we attempt to solve the problem of anatomical structure segmentation in medical images by learning from only one labeled image. Owing to the inherent regularized nature of anatomical structures, the same anatomy in different images usually share similar features in shape, appearance and gradients. Human annotators are able to annotate new images by referring to a labeled exemplar. Inspired by the human learning behavior, we propose the Contour Transformer Network (CTN) to explicitly incorporate anatomical priors into network training. CTN formulates the segmentation problem as learning a contour evolution behavior, which is modeled by a cascaded graph convolutional network (GCN). Three differentiable contour-based loss functions namely contour perceptual loss, contour bending loss and edge loss are proposed to describe the common features of appearance, shape and edge response, respectively. For each unlabeled image, CTN takes the exemplar contour as an initialization, then gradually evolves it under the guidance from the three losses. The training of CTN takes only one labeled image and a set of unlabeled images, thus significantly reduce the annotation cost. Experiment results on four different datasets demonstrate that our one-shot learning method performs competitively to the state-of-the-art fully supervised methods.

Few-shot image segmentation with class-agnostic shape prior

In the third work, we try to generalize the utilization of shape prior knowledge in segmentation network to arbitrary objects. Our goal is to develop a method that could learn to segment any unseen objects from very few labeled examples, which is also known as few-shot segmentation. Meta-learning has been a dominant paradigm adopted by most existing few-shot segmentation methods. Instead of directly training class-specific segmentation models, they leverage existing labels of some base classes to train a class-agnostic model which could expediently learn to segment novel classes from a few support images in test. A common problem of these methods is that they consider only in-domain few-shot segmentation, which mean the base classes and novel classes always come from the same domain. When there exists a significant domain shift (*e.g.*, from natural images to medical images), these meta-learning models will fail because of different feature distributions.

Training a universal model on base classes that could work on all novel classes may not be realistic. Considering the potential huge domain shift, we adapt the base model to the target domain using available labeled and unlabeled images before testing. The main idea is to fine-tune the prototypical model with a few labeled images and a set of unlabeled images of the novel class. Specifically, we first obtain a base model by training the prototypical network on base classes. For each test class, we take prototypes generated from the predicted query mask as anchor features, support prototypes from the same class as positive features, and support prototypes from all other classes as negative features. By minimizing the distance between anchor and positive features and maximizing the distance between anchor and negative features with a triplet loss, we optimize the segmentation of query images and improve the discriminability of the base model on the test class. We demonstrate the effectiveness of our method with extensive experiments on both in-domain and cross-domain few-shot segmentation tasks.

Chapter 2

Background

In this chapter, we briefly introduce the background knowledge that will be mentioned in this dissertation. Since we mainly study the problem of incorporating shape prior knowledge into CNNs, CNNs and shape prior knowledge are two most important foundations of this research, and both of them are extensively used in all three proposed methods. In the following, we first introduce the basic concepts of CNN and its main components, and then discuss different types of shape prior and their roles in semantic segmentation.

2.1 Convolutional Neural Network

According to the definition in [27], Convolutional Neural Networks are a specialized type of artificial neural networks that use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. They are specifically designed for digital images processing and have been used in a wide range of computer vision tasks. The development of Convolutional Neural Networks can date back to as early as 1989. Yann LeCun et al. first used the backpropagation algorithm to train LeNet [44], a pioneer of modern CNNs, for handwritten digit recognition. It did not become the mainstream of image recognition back then, because the limited computational capability and the lack of large-scale datasets make CNNs difficult to train. Thanks to the rapid development of GPUs and public datasets, Krizhevsky et al. came up with a groundbreaking CNN model called AlexNet [42] in 2012, which achieved a top-5 error of 15.3% on the ILSVRC challenge [1], 10% lower than the previous best method. After that, deep learning and CNNs drew significant attention in the computer vision community.

Besides GPUs and the dataset [91], the well-designed network architecture also played an important role in the success of AlexNet, which is illustrated in Fig 2.1 [2].

[1] https://image-net.org/challenges/LSVRC

[2] https://anhreynolds.com/blogs/alexnet.html

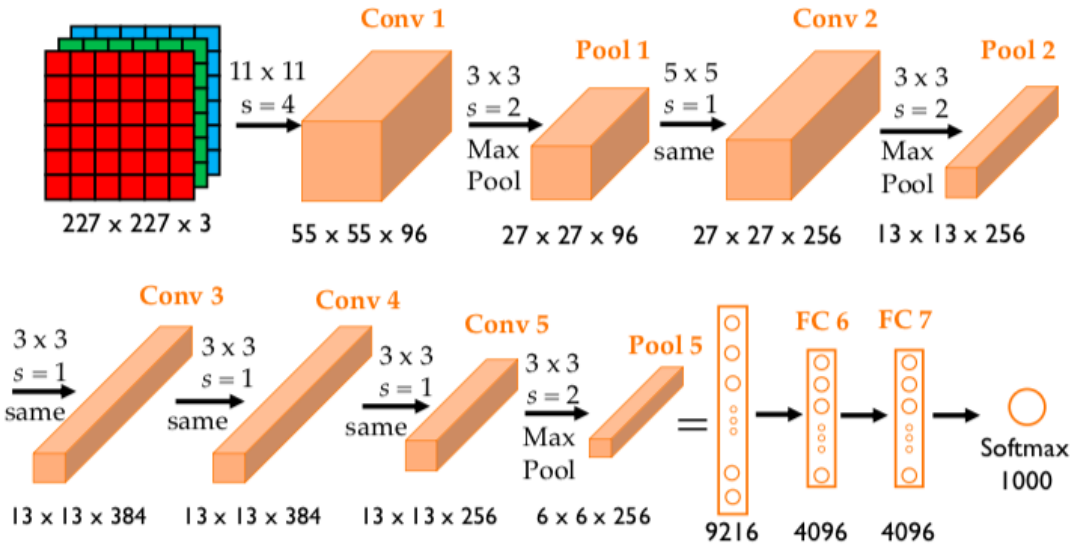


Figure 2.1: The network architecture of AlexNet.

Basic components of image classification CNNs typically include convolutional layer, non-linear activation, pooling layer, normalization layer, fully connected layer, dropout, softmax function, etc. We will elaborate on the most important components of them in the following.

Convolutional Layer

Convolutional layers are the most fundamental component of CNNs. They are responsible for modeling local patterns in the input volume, which can be an input image or the output from a previous layer. The parameters of each convolutional layer include a set of convolution kernels and a bias vector. Each convolution kernel is a small matrix of size *e.g.*, 3×3, 5×5, etc. In analogy with linear transformation, each channel of the input volume is an input variable, the convolution kernels are the weights of each variable, and the bias plays the same role. Different from linear transformations, a convolution layer does not connect each neuron to all neurons in the preceding volume, because it is not practical for high-dimensional volumes and

will ignore the spatial structure of objects in the image. Each kernel convolves along the x and y axes of the input volume, then computes the dot product with a local area of the input each time, and produces a 2-dimensional activation map. In this way, the network learns kernels that activate when certain types of features appear at specified spatial locations in the input.

The size of the output volume is jointly determined by the number of kernels, the convolution stride and the size of padding. The number of kernels determines the number of channels in the output volume, where each kernel usually corresponds to a certain pattern. Taking the first convolutional layer as example, it takes the raw image as input, and different kernels of it along the depth dimension will be active in presence of various color and edges. The convolution stride controls how depth columns are distributed over the spatial dimensions. When the stride is one, a new depth column of kernels is assigned to spatial locations separated by only one spatial unit. This results in a lot of overlapping receptive fields between the columns, as well as a lot of outputs. Higher strides, on the other hand, cause the receptive fields to overlap less, resulting in a smaller output volume with reduced spatial dimensions. Finally, the size of padding decides the convolution behavior near the boundary of volumes. When the convolution kernel approaches the boundary, it will stop to make sure all elements do not exceed the boundary, unless the input volume is padded with zeros or pixel values along the border.

Non-linear activation

In a neural network, activation functions are used to determine if a neuron can be activated. It manipulates the current volume and generates an output for the neural network that includes the data's parameters. It is also called transfer functions in other literature. They can be linear or nonlinear depending on the function it represents and is used to control the output of neural networks in different domains. The

most commonly utilized activation functions are non-linear functions. It is simpler for a neural network model with non-linear functions to adapt to diverse types of data and distinguish between distinct outputs [27].

Logistic, tanh, softsign, rectified linear unit (ReLU), Leaky ReLU, and parametric ReLU are common examples of nonlinear activation functions. It's worth noting that, as pointed out in [27], backpropagated gradients employing the sigmoid function can easily get saturated, resulting in problems updating network weights. While information flowing through numerous layers of feature extractors, ReLU seeks to maintain information about relative intensity. Later, to avoid the vanishing gradient introduced by ReLU, leaky ReLU was developed. To solve the drawbacks of ReLU and leaky ReLU, PReLU revealed a novel method for learning the negative part's slop.

Pooling layer

Pooling, a type of non-linear downsampling, is another key idea in CNNs. Pooling can be implemented using a variety of non-linear functions, the most common of which is max pooling. It divides the input image into a series of non-overlapping grids and outputs the maximum value in each grid. The pooling layer's job is to gradually shrink the spatial size of the representation in order to reduce the number of parameters and computations in the network, and therefore to prevent overfitting. A pooling layer is frequently inserted between cascaded convolutional layers in a CNN architecture.

The pooling layer resizes each depth slice of the input spatially and works independently on each one. A typical pooling layer is with filters of size 2, and is applied with a stride of 2, thus downsampling every channel in the input by a factor of 2, along both the row and column dimensions. It will discard 75% parameters in the volume. In this situation, every **max** operation would take a maximum of four digits.

The depth dimension stays the same. The pooling filters can also be in other forms, such as average pooling, L2-Norm pooling, etc.

Fully connected layer

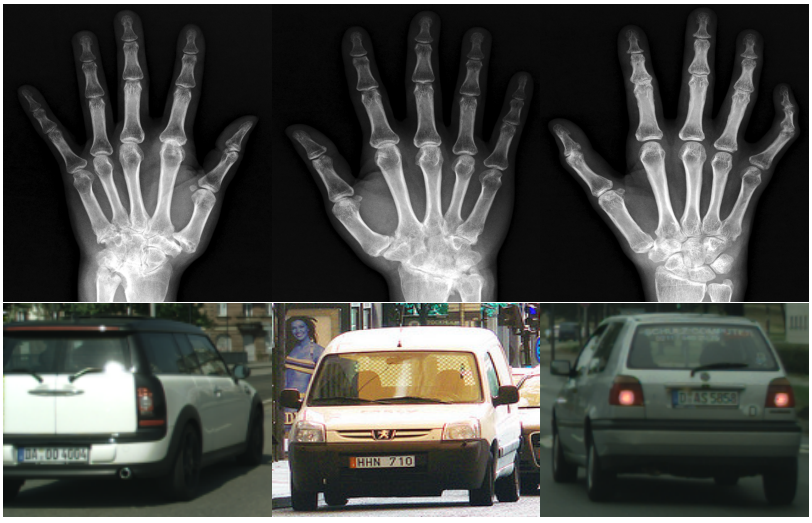
Finally, the classification task of the neural network is achieved by fully connected layers after a series of convolutional and pooling layers. In a fully connected layer, all neurons in the previous layer are completely connected to every neuron in the current layer. A matrix multiplication plus a bias offset can thus be used to determine the activations of a fully connected layer. Fully connected layers in CNNs encode the feature volume generated by convolutional layers to a feature vector specific to the learning task. Fully connected layers typically account for a large portion of the total parameters in CNNs, which can lead to overfitting. For this problem, some studies [54, 11] have removed the entire connections between the final convolutional layer and the fully connected layer that follows, so that the parameter amount can be significantly reduced.

2.2 Shape Prior Knowledge for Segmentation

Since typical handcrafted rules based on intensity, texture homogeneity and edge contrast are insufficient to segment objects in an image, pure low-level segmentation algorithms often fail to produce desired segmentation results. Researchers have advocated incorporating shape prior knowledge into low-level segmentation algorithms to circumvent these limitations. As summarized in [19], shape priors in image segmentation could be categorized into:

- Low-level shape priors which typically favor shorter boundary length, i.e., curves with shorter boundary have lower shape energy, where boundary length can be measured in various ways.

(a) Objects with implicit shape prior



(b) Objects with explicit shape prior

Figure 2.2: Objects with implicit and explicit shape priors.

- Mid-level shape priors which favor, for example, thin and elongated structures, thereby facilitating the segmentation of roads in satellite imagery or of blood vessels in medical imagery.
- High-level shape priors which favor similarity to previously observed shapes, such as hand shapes, silhouettes of humans, or medical organs like the heart, the prostate, the lungs, or the cerebellum.

Before utilizing these shape prior knowledge, how to represent the shape of object is a core problem that needs to be addressed. Existing solutions typically include implicit and explicit representations [20]. Implicit representation describes shape with masks where all points in space are labeled as belonging to the inside or outside of the object. Partial differential equations are used to solve the optimization of

such implicit shape representations in a spatially continuous manner. Level set [8] and convex relaxation techniques are two popular examples. Implicit representations with spatially discrete setting have also been employed by graph cut approaches [5]. For explicit representation, the shape's boundary can be explicitly represented as a polygon or a spline curve in a spatially continuous manner. In the case of spatially discrete setting, it can be a set of edges that formed by regular grids. In Figure 2.2, we illustrate three common objects with two types of shape priors to provide an intuitive understanding.

Implicit shape prior

Objects like roads and buildings have their own characteristics in shape, for example, curve-like or block-like. Except for this implicit constraint, their shapes can be arbitrary. For this kind of objects, we can use implicit representation of masks to describe their shapes. Compared to explicit representations that use polygon or splines, implicit representations can be easily extended to higher dimensions. Level set, graph cuts, and convex relaxation techniques are all examples of algorithms that can be extended to three or more dimensions. In contrast, extending 2D polygons or splines to 3D space is non-trivial, because the concept of arc-length parameterization is only applicable to curves but not surfaces.

Another merit of implicit representations is that they can be applied to any form of shape prior knowledge with ease. The topology of the shape is not restricted when using implicit representation because it only relies on a spatial labeling that indicates if a pixel is within an object. As a result, either level set, graph cut, or modern CNNs can readily handle objects with any topology. But it is not this case for explicit representation, algorithms that model shapes into discrete paths are often limited to open or close curves [74, 18], while algorithms that model shapes into continuous parametric curves suffer from transforming a single curve to complex

object boundaries [19].

Explicit shape prior

Although not as flexible as implicit representations, a considerable advantage of the explicit shape representation is that it allows to build point correspondence. The definition of human on shape similarity is closely related to the correspondence between points on two shapes or semantic elements [20]. However, in high-dimensional data, determining ideal point correspondences is still a problem to be addressed.

Besides, modeling shape similarity is typically more clear and obvious for explicit representations. Taking two spline curves as instance, performing linear interpolation on them produces an intermediate spline curve, which conforms with the human understanding of an average shape. In contrast, the linear interpolation of implicit representations is usually much more difficult, because the convex combination of two binary functions does not necessarily lead to a new binary function.

Shape prior knowledge in CNN

Existing works that incorporate shape prior into CNN segmentation mostly focus on the domain of medical image. Comparing with objects in natural images, anatomical structures in medical images are naturally more constrained in terms of shape. Thus, the inclusion of shape priors in medical imaging could have more impact compared with their usage in natural images [94].

As summarized in [86], proposed methods of utilizing shape priors in CNN segmentation could be divided into two categories: shallow shape regularization and deep shape regularization. The former refers to explicitly imposing certain geometric and structural characteristics on the segmented ROIs at pixel-level. In [61], the authors leveraged the star shape prior of the object in skin lesion segmentation. A new loss function was proposed to regularize segmented ROIs to the star shape. In [24],

the shape prior was used to refine the segmentation result of FCN, by registering a cohort of atlas masks to the target segmentation mask. Zhou et al. [94] utilized the prior knowledge of the average organ size distribution to tackle the problem of multi-organ segmentation when only single organ is annotated.

Deep shape regularization methods usually exploit shape priors in the learned feature space. For example, Ravishankar et al. [70] proposed a convolutional autoencoder that projects segmentation masks to the shape space, and a projection loss that encourages the predicted segmentation to be similar with the ground truth in the shape space. [90] employed a shape regularization autoencoder in a segmentation network to constrain the prediction to follow a learned shape distribution. [45] took a shape template as an additional input channel and deforms it to match the underlying structure through a spatial transformer network. Comparing with shallow shape regularization, deep shape regularization methods are usually more robust to image noise [86].

Chapter 3

Literature review

In this chapter, we will briefly review literature related to our research. We first introduce classic and state-of-the-art CNN-based methods for general-purpose semantic segmentation in Section 3.1, and then discuss weakly supervised semantic segmentation methods that use only weak supervision for segmentation model training in Section 3.2. In Section 3.3, we review another branch of works proposed for label-efficient semantic segmentation, which uses only very few labeled data for training, and is also known as few-shot semantic segmentation.

3.1 General Semantic Segmentation

Early attempts of utilizing CNNs in image segmentation were in a patch-wise manner. They predict the label of a pixel by cropping a neighborhood patch around it and then simply using CNN models as patch classifiers (Figure 3.1). For example, Ciresan et al. [16] trained a network in such a manner for electron microscopy images and achieved better accuracy than previous handcrafted feature based methods. This approach allows to train a CNN model with very few annotations because thousands patches could be cropped from a single image. But the inference speed is a main drawback that limits its use in practical applications, because it has to predict labels pixel by pixel. Also, this approach fixed the size of the patch, which is not the best practice for patch classification.

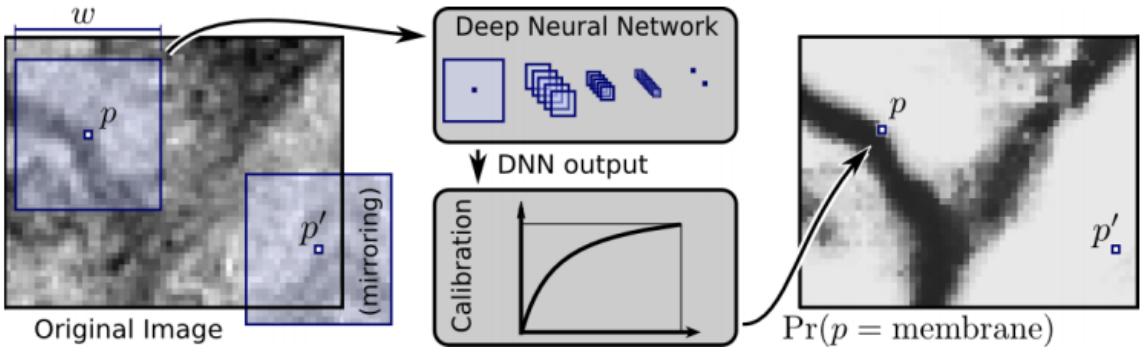


Figure 3.1: An illustration of patch-based image segmentation [16].

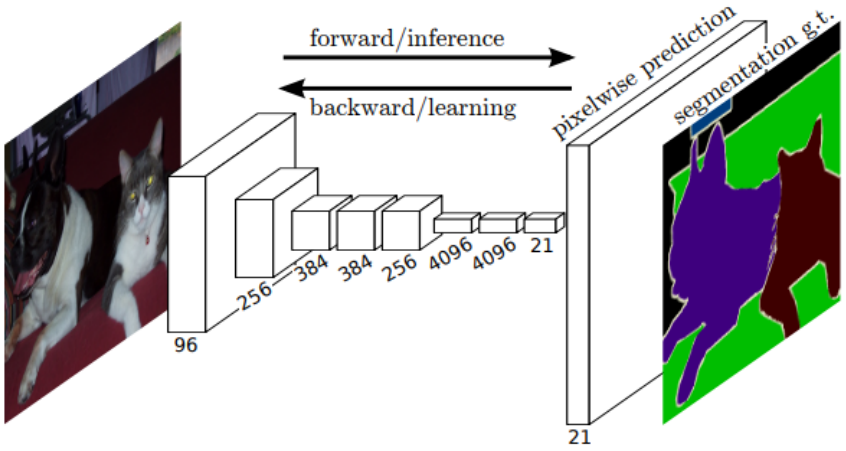


Figure 3.2: An illustration of the FCN segmentation framework [54].

State-of-the-art semantic segmentation methods are derived from a common predecessor – the Fully Convolutional Network (FCN) [54]. FCN is the first end-to-end CNN-based semantic segmentation framework. Comparing with previous patch-wise methods, it takes the input image as a whole and directly outputs the segmentation mask, thus significantly reduces the computational redundancy and inference time. A key idea that realizing the whole-image-at-a-time prediction is to view fully connected layers in CNN classification models as convolutions with kernels that cover their entire input regions. They adapt classification models into feature extractors and upsample the feature map using bilinear interpolation or transposed convolution (Figure 3.2). This method outperforms previous methods by a large margin on multiple segmentation benchmarks including PASCAL VOC, NYUDv2 and SIFT Flow.

Following the framework of FCN, many CNN-based methods have been proposed for semantic segmentation in recent years. Chen et al. [10] proposed to use a fully connected Conditional Random Field (CRF) to refine the upsampled feature map of CNNs, thus improves the accuracy on object boundaries. Noh et al. [64] used unpooling and deconvlutional layers to replace the bilinear interpolation in FCN,

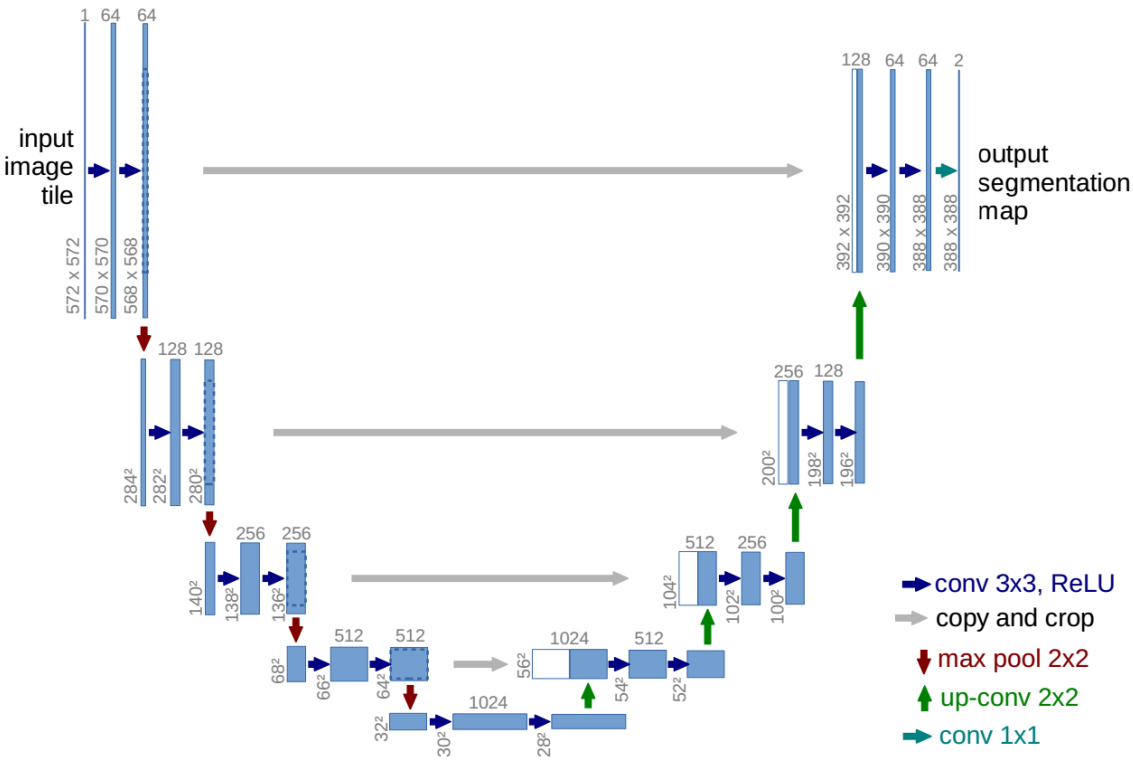


Figure 3.3: The network architecture of UNet [71].

which could further improve the performance. UNet [71] refined the architecture of FCN by adding convolutional layers to the decoder and using skip connection to concatenate encoder outputs and decoder outputs (Figure 3.3). Pyramid feature aggregation techniques have been proven useful in several semantic segmentation architectures, such as Feature Pyramid Network (FPN) [49], Pyramid Scene Parsing Network (PSPNet) [112], and DeepLab v3 [9]. They usually merge feature maps of different scales to enhance the final feature embedding. In DeepLab v3, dilated convolution is proposed to take the place of pooling layers in conventional CNN architectures, thus avoid the loss of resolutions of feature maps.

All above CNN image segmentation methods are heatmap-based, namely obtaining the segmentation mask by thresholding a probability heatmap. There is another choice that directly predicts the contours of segmentation objects. Given an initial contour that is close to the object, traditional ACM-based methods could deform

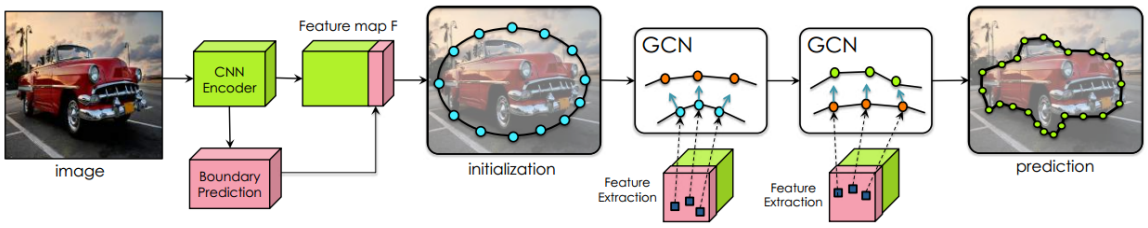


Figure 3.4: An illustration of Curve-GCN [50].

the contour to the object boundary by minimizing an energy function. Due to the limitation of handcrafted energy functions, ACMs are not robust to noise, occlusion, and other complexities. Recently, Curve-GCN [50] was proposed to learn the contour evolution behavior using Graph Convolutional Networks (GCNs). It assumes the bounding box of object is known and initialize the contour with an ellipse. A set of control points are uniformly sampled from the contour, and the point features are sampled from the image feature map, then cascaded GCNs are employed to predict the contour offset by inferring the contour features (Figure 3.4). This method achieved the state-of-the-art performance in interactive segmentation annotation. Comparing with heatmap-based segmentation, contour-based segmentation naturally keeps the integrity of objects, but cannot handle objects with holes, such as chairs and ladders.

3.2 Weakly Supervised Semantic Segmentation

Fully supervised CNNs have achieved tremendous performance in natural image segmentation, but they require a large-scale fully annotated training set. For tasks that have limited fully annotated data, powerful fully supervised approaches are not always applicable. To address these tasks, weakly supervised methods that aim at training CNN segmentation model with less annotation cost are proposed. Instead of using pixel-level segmentation masks as supervision, weakly supervised methods take advantage of various forms of labels that are easier to annotate to train the segmentation model, including image-level tags, image scribbles, bounding boxes,

etc.

Weak supervision of image-level tag

When given the weak supervision of image-level tag, the multiclass classification information is provided by image-level class labels, but there are no object localisation cues. Graph-based models were explored to addressing this task in early research [110]. They consider superpixels similarity of images to perform segmentation tasks. In the age of deep learning, the difficulty of semantic segmentation was transferred to the process of giving class labels for each pixel in images, allowing neural network frameworks to be used more easily. For example, multiple instance learning (MIL) is used to train the segmentation model in [69], while the Expectation Maximization (EM) algorithm is employed in [67]. As early attempts, these methods are relatively time-consuming and inaccurate compared to more recent ones.

Some novel methodologies have recently been developed that make significant progress in public datasets. In [40], Kolesnikov et al. presented a hybrid loss function that combines seeding, expansion and constrain-to-boundary to train segmentation networks. Wei et al. [104] proposed an iterative learning strategy, in which the network is first trained using basic images and matching saliency maps as labels, and the network's segmentation capability is gradually improved as the train data becomes more complicated. Coarse-to-fine training/prediction is a commonly used strategy in recent works. They first roughly localize objects in the image and then refine their locations using features with finer scale [12].

Weak supervision of bounding box

When training segmentation networks with bounding box labels, there are usually two steps involved, where the first step is to generate pseudo ground truth mask from bounding boxes, and the second step employs the pseudo ground truth to train a

semantic segmentation model [33]. In the first step, pixels that in the outside of all bounding box can be labeled as the background class without question. For pixels in the bounding boxes, there are various methods to process them. In the easiest case, we can take every pixel in the bounding box as a positive sample of the corresponding object class [67]. If a pixel belong to two bounding boxes, the conflict can be addressed by assigning it to the smaller one instead of the bigger one. If taking all pixels in boxes as positive is inappropriate, we can also take only a small fraction of pixels in the box's central region as positive [37]. But these simple methods are not accurate enough, because they rely on a poor approximation of an item using a bounding box. More advanced foreground segmentation algorithms can be used to acquire a more precise approximation of objects. Prior research has employed DenseCRF, GrabCut, etc [41, 72].

Weak supervision of scribble

Scribble annotations are only a few strokes created to label a small portion of an object or the background. Because scribbles only carry incomplete semantic information and no fine-grained boundary is provided to guide the model to precisely segment each object, the model trained naively by scribbles only gives crude segmentation results. Several weakly supervised segmentation algorithms using scribbles have been proposed to circumvent this problem [107]. Scribble-supervised segmentation was commonly approached in an interactive manner in the early stages, with feedback scribbles being continuously produced to refine the segmentation findings [28]. In this case, most methods transformed an image into a weighted undirected graph. Many studies have sought to use deep neural networks to handle scribble-supervised segmentation since the rise of deep learning. ScribbleSup [48] was the first to use deep learning in scribble-supervised segmentation. Using the weakly annotated scribbles and a CRF model, a comprehensive annotation map was initially created [41]. Sub-

sequently, the segmentation findings were then refined using a combination of neural network optimization and the CRF energy function. To progressively update the segmentation network and the propagated dense annotations, RAWKS [97] included a deep segmentation network and a learnable label-propagator. A perception refinement network is proposed in [99] to make better use of the encoder's data, particularly the higher resolution feature maps.

3.3 Few-shot Semantic Segmentation

Few-shot semantic segmentation (FSS) aims to segment unseen object classes in query images by referring to only few labeled (support) images [17]. As a potential solution to annotation-efficient segmentation, it has received increasing attention in recent years [1, 6, 11, 17, 22, 24–26]. Existing FSS methods typically solve this problem in a metalearning framework. They leverage existing large-scale datasets as base classes, and organize the training data into episodes of query and support images to train a two-branch model that is agnostic to object classes [1]. These methods show good generalization ability in popular FSS benchmarks (e.g., PASCAL-5i and COCO-20i) where the base and novel classes are from natural images. Most existing few-shot semantic segmentation (FSS) methods follow the meta-learning paradigm to train a class-agnostic two-branch model that could leverage the knowledge in support images to perform segmentation. Based on how the support knowledge is incorporated, these methods can be categorized into relation-based, prototype-based and transductive inference.

Relation-based FSS

Relation-based methods use the two-branch network to learn to compare query images against support images by fusing their features [84]. For example, Zhang *et al.* [109] adopt an attention-based fusion scheme to fuse query and support features and pre-

dict the query mask using an iterative optimization module. Liu *et al.* [51] propose a cross-reference network where two branches concurrently make predictions for both query and support images. Tian *et al.* [95] propose to adaptively refine the concatenated query and support features with intra-source enrichment and inter-source interaction. Graph CNNs have also been employed to establish more robust correspondences between the support and query images, enhancing the learned prototypes [36]. Alternative solutions to learn better class representations include: imprinting the weights for novel classes [30], decomposing the holistic class representation into a set of part-aware prototypes [21] or mixing several prototypes, each corresponding to diverse image regions [38].

Prototype-based FSS

Instead of fusing query and support features, prototype-based methods predict masks by computing distance from query features to support prototypes [81]. Particularly, the support images are employed to generate class prototypes, which are later used to segment the query images via a prototype-query comparison module. Dong *et al.* [23] made the first attempt to build a prototypical framework for few-shot semantic segmentation. Wang *et al.* [100] propose PANet which uses a prototype alignment regularization to maintain a cycle constraint between support and query. PPNet proposed by Liu *et al.* [53] introduces clustering into FSS, of which the core idea is to decompose the holistic prototype into a set of part-aware prototypes. These approaches mainly aim at exploiting better guidance for the segmentation of query images [42, 23, 36, 40], by learning better class-specific representations [37, 19, 21, 38, 30] or iteratively refining these [41].

FSS with transductive inference

A recent method ReRPI [3] conducts FSS without meta-learning. They propose a transductive inference method to fine-tune a linear layer of the base model on each test episode, which also follows the pre-training and fine-tuning approach. Unlike inductive inference, their transductive setting also exploits the unlabeled pixels from the query image, which are naturally accessible, when building the classifier for a task. Therefore, their inference leverages the structure and global statistics of both the unlabeled and labeled pixels of a given few-shot segmentation task by optimizing an original task-specific loss function. Without episodic pre-trained base model, they achieved superior performance than competitors that use meta-learning.

Chapter 4

Weakly supervised curve structure segmentation with implicit shape prior

4.1 Problem Overview

Embellished designs on the surface of cultural heritage objects, such as pottery, shell, stone and wood contain important information for archaeologists [113]. These designs, if successfully identified and correlated, can be used to build chronologies and track trade networks of a region thousands of years ago. In archeology, most of these designs are found to be curve patterns stamped or carved by their makers. Therefore, it is of great interest to archaeologists to accurately segment the curve structures on the surface of unearthed fragments of cultural heritage objects and identify their underlying designs [35, 29]. Figure 4.1 shows several unearthed pottery sherds dating to the Woodland period of Southeastern North America. The curve structures on their surfaces reflect a portion of the curve pattern carved into wooden paddles and applied onto hand-built clay vessels designed by southeastern Native Americans around 2000 years ago. There are hundreds of thousands of such fragmented culture heritage objects stored in museums, which calls for more intelligent and automatic tools to explore them.

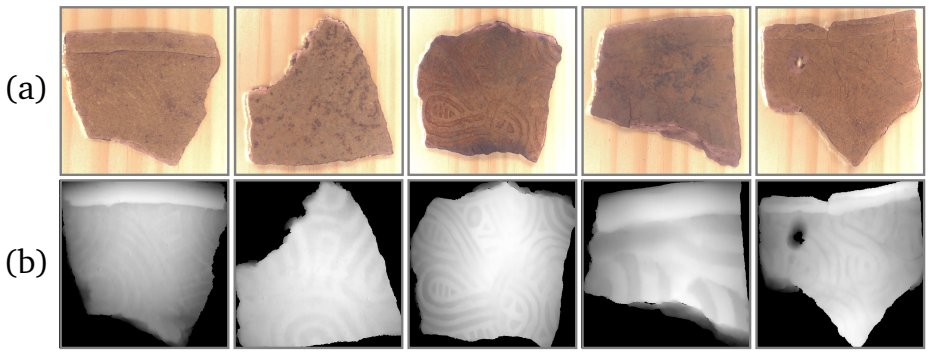


Figure 4.1: Five unearthed pottery sherds dating to the Woodland period of Southeastern North America. (a) RGB images. (b) Depth images where intensity indicates the depth.

Clearly, accurately segmenting the curve structures stamped on the surface is the first step to explore these cultural heritage objects. In most cases, these curve structures do not bear distinctive colors and it is very difficult, if not impossible,

to segment them from an RGB image of the sherd, e.g., Figure 4.1(a), taken by traditional cameras. In archeology, 3D scanners are usually utilized to produce a depth image of the object surface – with paddle stamping, the locations of curves exhibit a larger depth than the non-curve portion of surface, as shown in Figure 4.1(b).

However, three complexities may lead to very weak curve structures on the obtained depth map and make the curve structure segmentation a very challenging problem. First, the carved paddles used for stamping are usually flat while the object surfaces are usually not. As a result, the paddle typically does not well fit the object surface, which leads to shallow curves at many locations. Second, purposeful smoothing of the stamped surface during vessel manufacture or weathering and erosion after vessel discard can lead to subtle depth differences between the curve and the non-curve portions of the surface. Third, erosion and weathering make the object surface highly rough, which is equivalent to adding random noise to the depth map of the initial object surface. With these three complexities, it is difficult to use a low-level image segmentation algorithm to accurately segment these depth images for curve structures, as shown by an example in Figure 4.2.

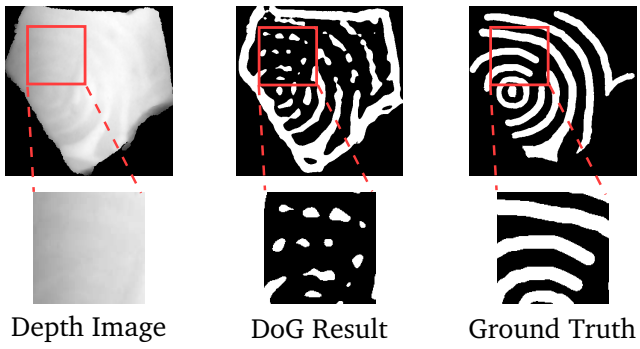


Figure 4.2: An illustration of using low-level methods for curve structure segmentation. Serious erosion in the red square leads to very low contrast in the depth image, and low-level method, such as DoG (Difference of Gaussian), may produce very poor segmentation.

In this work, we propose a new supervised learning approach to segment such curve structures that were weakly stamped on object surface. The basic idea is that,

in most applications, such as exploring cultural heritage objects in archeology, the underlying designs of the curve structures bear certain geometries and patterns. For example, most of the curve structures consist of smooth curve segments. Furthermore, many curves in the structures show good parallelism against each other. These characteristics give the material a visually distinctive style [80]. Consideration of these high-level geometry and pattern information may help improve the accuracy and reliability of curve structure segmentation. While it is difficult to handcraft the features of all relevant curve geometry and pattern in an application, we expect the proposed approach can automatically learn these features from a set of training data with labeled ground truth.

In practice, the curve structures of interest have width, which may vary along the curve and need to be inferred in segmentation. However, it is well known that the curve geometry and pattern are independent of the curve width. Mixing all of them may substantially increase the difficulty of feature learning for segmentation. In this work, we handle them separately by developing a three-step curve structure segmentation algorithm. In the first step, a Fully Convolutional Network (FCN) is employed to extract the skeleton of curve structures, and estimate a scale value at each skeleton pixel. This scale value reflects the curve width at the corresponding skeleton pixel. In the second step, we propose a dense prediction network to refine the curve skeletons. In the third step, we develop an adaptive thresholding algorithm to achieve the final segmentation of curve structures with width by considering the estimated scale values.

For the experiments, we collected the depth image of a set of pottery sherds excavated from archaeological sites associated with the Swift Creek paddle-stamped tradition of southeastern North America. Ground truth curve structure segmentation are manually constructed. We evaluate the proposed method on the collected depth images and compare its performance against several other existing algorithms. We

also evaluate the segmentation results in the task of design matching in archeology.

4.2 Method

The proposed method consists of three steps. First, we train an FCN to detect the skeletons of the curve structures in the depth image. This FCN network also estimates a scale value at each detected skeleton pixel to reflect the curve width at this skeleton pixel. Second, we train a dense prediction convolutional network to identify and prune false positive skeleton pixels. Finally, we develop a scale-adaptive thresholding algorithm to recover the curve width and achieve the final segmentation of curve structures.

Step I: Detecting Curve Skeletons using FCN

In this work, skeletons are the center lines of the curve structures and they are of one-pixel width. By ignoring the curve width, the skeletons reflect the geometry and pattern of the curve structures. Therefore, in the first step, we train a FCN to detect the skeletons of the curve structures from an input depth image. Just like image segmentation, skeleton detection can be formulated as a pixel-labeling problem: skeleton pixel has a label 1 and non-skeleton pixel has a label 0.

We design an FCN, as illustrated in Figure 4.3, to label skeleton pixels. It follows the encoder-decoder architecture developed in [55]. Encoders 1 and 2 are small convnets made up of two 3×3 convolutional layers, two ReLu layers and one 2×2 max-pooling layer. Encoder 3 is a small convnet made up of three 3×3 convolutional layers, three ReLu layers and one 2×2 max-pooling layer. After an encoder, the image size will be reduced to 1/4. Therefore, the receptive field sizes of feature maps generated by the three encoders are 2×2, 4×4, and 8×8, respectively. After each encoder, a fully connected layer is employed to match the number of feature maps with the number of labels. In order to generate pixelwise prediction result, the fully

connected layers are implemented by 1×1 convolutional layers. These results are denoted as S_1, S_2 and S_3, respectively, as shown in Figure 4.3. Note that the size of S_1, S_2 and S_3 are successively downsampled by factors of 2, 4, and 8 from the original image size. The decoders are three deconvolution layers with a kernel size of 4×4 and a stride of 2. The kernels are fixed to perform bilinear interpolation [106].

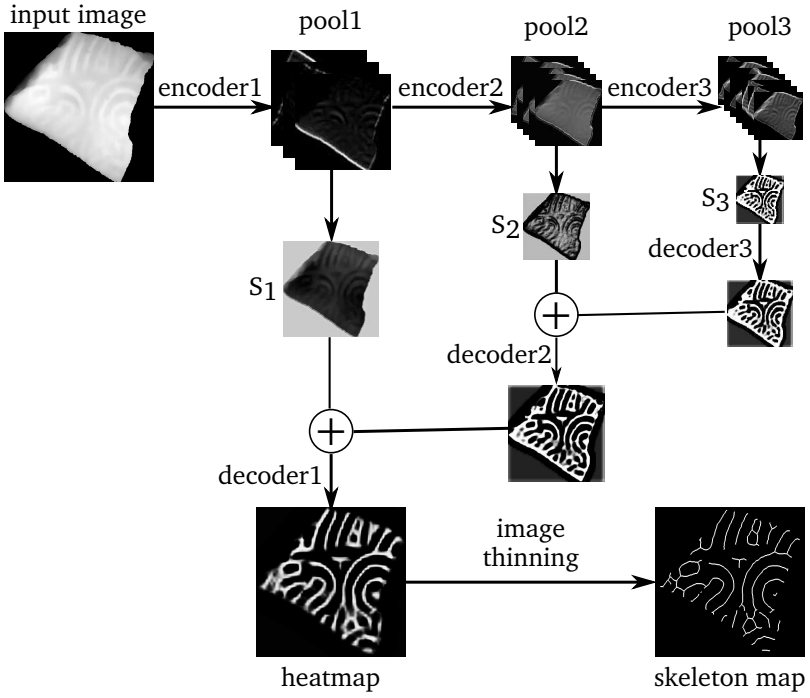


Figure 4.3: FCN used for skeleton detection.

The use of multiple encoders/decoders can extract image features in different levels of details. To make full use of all the extracted features, the decoders are organized in a way of stepwise accumulation when fusing them together.The output skeleton heat map S can be computed by

$$S = softmax(\Psi^{(2)}(S_1 + \Psi^{(2)}(S_2 + \Psi^{(2)}(S_3)))) \tag{4.1}$$

where Ψ indicates the upsampling operation performed by the decoders and its associated superscript is the upsampling factor, e.g., $\Psi^{(2)}$ indicates an upsampling of map

by a factor of 2. With the skeleton heat map S, we apply a common image thinning algorithm [43] to generate the single-pixel width skeleton map.

Inspired by [76], we can compare the three score maps S_1, S_2 and S_3 to estimate the scale at each detected skeleton pixel. The scale value at a skeleton pixel reflects the local curve width at this pixel. More specifically, since different encoders correspond to different receptive field sizes, at each pixel the receptive field size of the encoder with the largest score reflects the scale at this pixel. Before we compare the score of different maps, we need to first upsample them to the original image size. This way, the scale $s(x, y)$ at the skeleton pixel (x, y) can be computed by

$$s(x, y) = \arg \max_{k \in \{1,2,3\}} \hat{S}_k(x, y) \tag{4.2}$$

where $\hat{S}_k = \Psi^{(2^k)}(S_k)$ is the upsampled score map of S_k. Later we will use the estimated scale values to help recover the curve width.

Step II: Refining Skeletons using Dense Prediction Convnet

While we expect the FCN trained in Step I can learn curve geometry and pattern features in detecting skeletons, we find that it still detects many false positive skeletons, as shown in Figure 4.4. In this step, we further train a supervised classifier to identify and prune such false positives by learning more curve features. Specifically, for each skeleton pixel (x, y) detected in Step I, we take a neighboring 45×45 window in the original depth image around the pixel (x, y) as the input and train a dense prediction convnet to determine whether (x, y) is a true skeleton pixel or a false positive.

On real images, detecting a skeleton with small dislocation to its real position is totally fine and unavoidable – even a manually labeled skeleton may not be perfectly aligned with the real center line of the curve structures. Therefore, our aim is not to directly train a hard classifier to distinguish skeleton pixels and non-skeleton pixels. Instead, we hope to train a soft classifier where a skeleton probability is outputted at

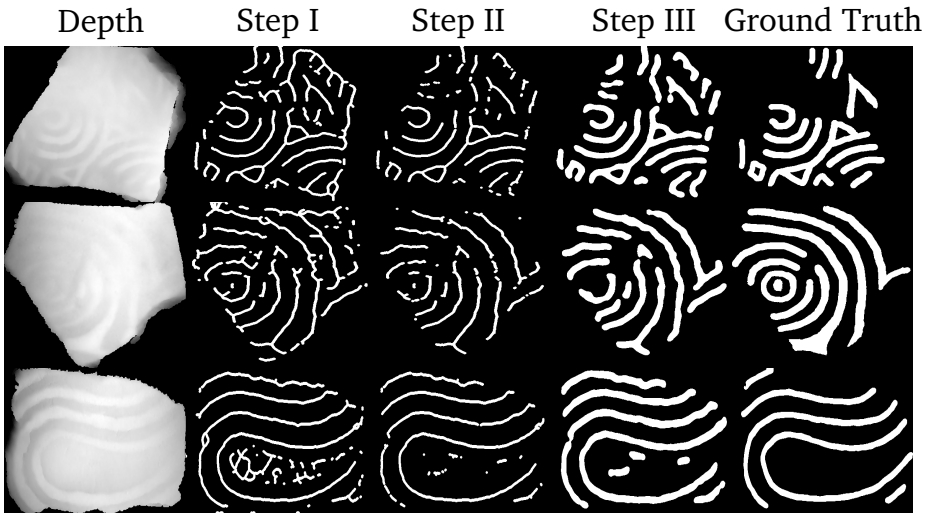


Figure 4.4: Example results after each step of the proposed method.

each pixel. To achieve this goal, in the training we transform a binary skeleton map to a skeleton probability map by

$$D(x,y) = \frac{1}{1 + \min\limits_{(x',y')\in P} \sqrt{(x-x')^2 + (y-y')^2}} \tag{4.3}$$

where P is the set of skeleton pixels in the binary skeleton map. Using D as output of the network, the binary classification problem is converted to a regression problem. Accordingly, we need to use a sigmoid function instead of softmax in the last layer of the proposed dense prediction convnet.

In this work, we propose to use a convnet consisting of three convolutional layers, three max-pooling layers and two fully connected layers. Its specific configuration is summarized in Table 4.1. For a testing image, let the set of the skeleton pixels detected in Step I be $\hat{P}$ and the skeleton probability map generated by the prediction convnet in this step be D, we prune the low-probability (< 0.5) skeleton pixels in $\hat{P}$ to achieve a refined set of skeleton pixels as

$$P = \{(x,y) | (x,y) \in \hat{P}; \quad D(x,y) \geq 0.5\} \tag{4.4}$$

Sample results of skeleton map after this step of refinement can be found in Figure 4.4.

Table 4.1: The configuration of network for Step II, where n, k, s, p stand for the number of outputs, kernel size, stride and padding size respectively.

Type	**Configuration**
Sigmoid	-
Fully Connected	n:2
Dropout	ratio:0.5
Fully Connected	n:512
MaxPooling	k:2 $\times$ 2, s:2
Convolution	n:128, k:3 $\times$ 3, s:1, p:1
Batch Normalization	-
MaxPooling	k:2 $\times$ 2, s:2
Convolution	n:64, k:3 $\times$ 3, s:1, p:1
Batch Normalization	-
MaxPooling	k:2 $\times$ 2, s:2
Convolution	n:32, k:3 $\times$ 3, s:1, p:1
Input	45 $\times$ 45 gray-scale image

Step III: curve structure Segmentation by Recovering Curve Width

In this step, we recover the width of curve structures from the skeleton map derived in Step II, with the help of the scale values derived in Step I. Note that the width of the curve structures is not a constant and it may vary along the skeleton. Denote the original depth image by I and let P be the set of refined skeleton pixels detected on I after Step II. For each skeleton pixel $(x, y) \in P$, we have a scale value $s(x, y) \in \{1, 2, 3\}$ derived in Step I. We construct the curve structure segmentation, in the form of a binary map C of the same size as I, using the following algorithm 1.

From the steps 3 and 5 of this algorithm, we can see that the curve width at each skeleton pixel is determined by both the scale value s at this pixel and the depth values I at and around this pixel. This algorithm does not require the detected skeleton to be exactly aligned with the center line of the curves – a small dislocation of the skeletons may not change the final segmentation if the dislocated skeletons are still located inside the underlying curves. Sample results after Step III are shown in

Algorithm 1 curve structure Segmentation by Recovering Curve Width

Input: Depth image I, Refined skeleton P, Scale values s
Output: Binary segmentation map C

Initialize all the elements in C to zero.
for each skeleton pixel $(x, y) \in P$ **do**
  Compute neighborhood:
  $\mathbb{N} = \left\{(x', y') | \sqrt{(x - x')^2 + (y - y')^2} \leq 2^{s(x,y)}\right\}$.
  for each pixel $(x', y') \in \mathbb{N}$ **do**
    if $I(x', y') \geq \frac{I(x,y) + \min_{(x'', y'') \in \mathbb{N}} I(x'', y'')}{2}$ **then**
      $C(x', y') = 1$
    end if
  end for
end for

Figure 4.4.

Design Matching

One important application of the segmented curve structures in archeology is the task of design matching. In the later experiments, we will use this task to evaluate the performance of curve structure segmentation. As shown in Figure 4.5(c), a design is a full curve pattern of the paddle that are used for stamping the object surface. In the past decades, archaeologists have restored a small number of full designs by manually examining thousands of sherds [6, 82]. The goal of design matching is to identify whether the segmented curve structures are originated from a known design. This is a classical partial matching problem and the key component is the definition of a matching score or distance.

In this work, we use the classical Chamfer matching [1, 113] for this purpose. As shown in Figure 4.5, we first thin both the segmented curve structures and the considered design into one-pixel wide skeletons and denote them as U and V, respectively. We then transform U to match the design V and compute the Chamfer distance

$$d'_{CM}(U_T, V) = \frac{1}{|U|} \sum_{u \in U_T} \min_{v \in V} \|u - v\|_2 \tag{4.5}$$

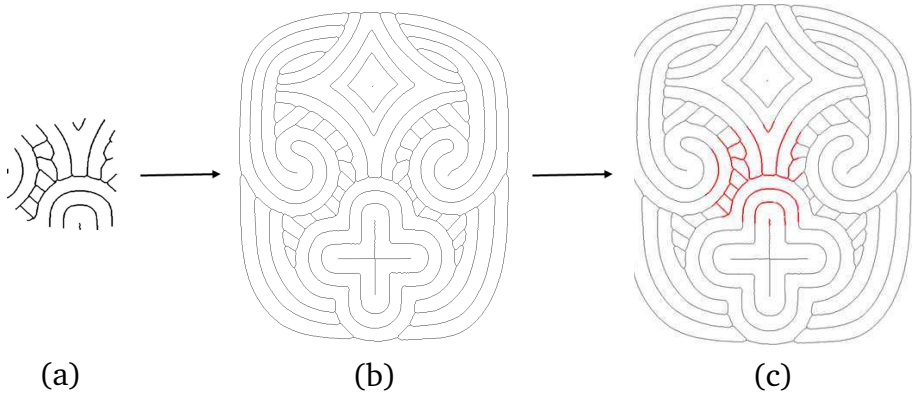


Figure 4.5: An illustration of design matching. (a) Thinned curve structures U segmented on the sherd. (b) Thinned full design V. (c) Partial matching between U to V with minimal Chamfer distance. Original design illustration copyrighted by Frankie Snow. Used with permission.

where U_T is the curve pattern U after the transform T, $u \in U_T$ indicates all the skeleton-pixel coordinates u in the transformed partial pattern U_T, and $v \in V$ indicates all the skeleton-pixel coordinates v in the curve pattern V. $|U|$ is the total number of skeleton pixels in the partial pattern U. Eq. (4.5) actually finds the nearest skeleton-pixel coordinate in V for each skeleton-pixel coordinate in U_T, records its Euclidean distance $\|u - v\|_2$ and finally averages over all the skeleton-pixel coordinates in U_T. The matching distance between U and V is then defined by

$$d(U, V) = \min_T d'_{CM}(U_T, V) \tag{4.6}$$

with T covers all possible translations and rotations. The scaling transforms is not considered here because both U and V have known actual sizes.

4.3 Experiments

In this section, we validate the effectiveness of the proposed method from three perspectives. First, we evaluate the proposed method in terms of the classical metrics of precision, recall and F-measure and compare it against other six comparison methods. Second, we conduct experiment to justify the usefulness of each step in our

method. Third, we evaluate the curve structure segmentation results in the task of design matching.

Dataset

For this study, we collected the depth images of 1,174 pieces of pottery sherds that are excavated in various archaeological sites located in southeastern North America. We used a linear array 3D laser scanner, NextEngine, to get the point cloud of sherd surfaces with the resolution of 100 points per mm^2. Then their depth images are sampled with the same resolution, i.e., each pixel in depth image covers $0.01mm^2$. The average size of the collected depth images is 446×421. We have 530 of these depth images with manually labeled Ground truth curve structure segmentations. Among all 530 images, we randomly pick 250 for training and the remaining 280 for testing.

To train the FCN in Step I, we thin all the Ground truth curve structures to one-pixel width skeletons, using a standard image thinning algorithm [43]. Data augmentation is employed here to generate sufficient training data. Specifically, we first split the whole image into small blocks with a size of 100×100. Then these blocks are rotated, scaled and flipped with the same scheme as in [76]. Finally, 141,696 blocks are used in FCN training in Step I. As for the network training in Step II, we randomly take 44,906 window images with a size of 45×45 around the skeleton pixels identified in Step I for training.

Implementation Details

For the purpose of better training, the parameters of encoders in the skeleton extraction network are initialized with the pre-trained FCN-8s model [55]. The parameters of decoder are fixed to perform bilinear interpolation [106]. The maximum number of training iterations is set as 20,000, with a mini-batch size of 10. The base learning

rate is 1×10^{-7} and decays to 1×10^{-8} after 10,000 iterations. Momentum and weight decay are set to 0.9 and 5×10^{-4} respectively.

Because the dense prediction convnet in Step II is relatively lightweight, we choose to train it from scratch. The maximum number of training iterations is set to 100,000, with a mini-batch size of 10. The base learning rate is 1×10^{-3}, and it decays in an inverse way with the parameter $\gamma = 10^{-3}$ and $power = 0.75$. Momentum and weight decay are set to be the same as the FCN in Step I.

F-measure based Segmentation Performance

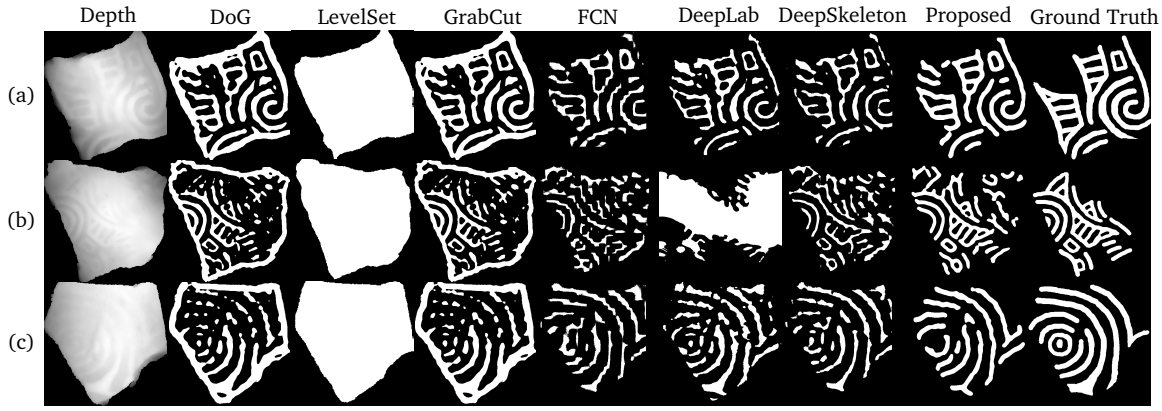


Figure 4.6: Examples of the curve structure segmentation result from the proposed method and six comparison methods.

To evaluate the effectiveness of our method of curve structure segmentation, we select six widely-used segmentation methods for comparison – Difference of Gaussian (DoG), Level Set [98], GrabCut [72], Fully Convolutional Network (FCN) [55], DeepSkeleton [76] and DeepLab [10]. The experiment is conducted on the 280 testing images as described above, and the evaluation criteria is the traditional F-measure of $\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$.

For most of these comparison methods, we keep the default settings in their source codes. But there are several exceptions need to be clarified. Since there is no default setting in DoG, we determine its parameters by trial-and-error. The best performing

Table 4.2: Precision, recall and F-measure of the proposed method and six comparison methods, averaged over 280 test images.

Methods	**Precision**	**Recall**	**F-measure**
DoG	0.366	0.774	0.490
LevelSet	0.262	0.938	0.399
GrabCut	0.357	0.671	0.448
FCN	0.589	0.472	0.514
DeepLab	0.585	0.670	0.583
DeepSkeleton	0.634	0.690	0.654
Proposed	0.660	0.827	**0.731**

setting we found is: $k_1 = k_2 = 45$, $\sigma_1 = 11$, $\sigma_2 = 5$, where k and σ are the kernel size and standard deviation of Gaussian filters. The filtered images are transformed to curve maps with the threshold of 1. In GrabCut, an initialization of the foreground object is required, for which we simply use the DoG result. In DeepSkeleton, we calculated the Ground truth scale maps by applying distance transform on Ground truth segmentation maps. Performance of all methods, averaged over all 280 testing images, are summarized in Table 4.2.

We can see that the proposed method achieves the best F-measure, and outperforms the second best (DeepSkeleton) by 7.7%. Figure 4.6 shows the segmentation results on three sample images, using all seven methods. In these images, we can observe that DoG actually enhances the difference between adjacent pixels. As a purely low-level method, it may not capture deep and shallow curves simultaneously. GrabCut was initialized by DoG, but its performance becomes even worse. One major reason might be that the data and smoothness energy defined in GrabCut are not sufficiently discriminative to segment the curve structures and non-curve object surface in such a low-contrast image. This is probably the same reason that makes Level Set fail. As expected, the three CNN-based comparison methods, i.e., FCN, DeepSkeleton and DeepLab, normally achieve better performances than the low-level methods. However, their segmentation results usually contain many false positives and the boundaries of the segmented curve structures are quite rough. While the pro-

posed method does not achieve the first place in either precision or recall, it achieves the best performance in final F-measure.

Usefulness of Each Step

Intuitively, the three steps of our method can be replaced by other alternatives or simply ignored. To justify the usefulness of each step, we design three additional experiments, in each of which, we modify or remove one step of the proposed method, and then check its influence to the final segmentation performance.

Modifying Step I: Step I of the proposed method is skeleton extraction. Actually, the FCN we used in this step can be trained to produce curve structure segmentation directly. However, we choose to extract skeletons first, and then take additional steps to recover the curve width. In this experiment, we make several adjustments in the FCN in Step I to let it output curve structures with width directly. For this purpose, we just use the Ground truth segmentation as the output for training and remove extra upsampling layers in FCN. All the implementation parameters keep unchanged. Sample results of this modified method are shown in Figure 4.7(b) . We can see that these results contain more false positives and rougher segmentation boundaries. Quantitatively, F-measure of the proposed method decreases from 0.731 to 0.665 if we make this modification to Step I.

Removing Step II: Step II of the proposed method employs a dense prediction convnet as a pixel-wise classifier to refine skeletons extracted by FCN in Step I. To justify its usefulness, we remove this step and recover curve width directly from the skeletons generated in Step I. Sample results are shown in Figure 4.7(c). We can see that the removal of Step II leads to more false positives. Quantitatively, F-measure of the proposed method decreases from 0.731 to 0.662 if we remove Step II.

Modifying Step III: Simple morphological dilation seems to be a very intuitive approach to recover curve width in Step III. In this experiment, we modify Step III

by replacing it with a dilation operation with a radius of 15 pixels, which is the best parameter after we try and test all different values. Sample results are shown in Figure 4.7(d). While the dilation produces very smooth curve structures, they do not align well with the ground truth. Quantitatively, F-measure of the proposed method decreases by 3.5% if we make this modification to Step III.

Figure 4.7: Sample segmentation results of the proposed method with modifications to each step. (a) Input depth image. (b) Segmentation result after modifying Step I. (c) Segmentation result after removing Step II. (d) Segmentation result after modifying Step III. (e) Segmentation result of the proposed method without any modification. (f) Ground truth segmentation.

Design Matching Performance

In this experiment, we evaluate curve segmentation results in the task of design matching. We take the depth images of 292 sherds with known full designs and in total they come from 29 different designs. The matching distance is the minimal Chamfer distance as defined above.

We use the Cumulative Matching Characteristics (CMC) ranking metric to evaluate the design matching performance. For each sherd curve pattern U, we match it

against all 29 designs by Chamfer matching. We then sort these 29 designs in terms of the matching distance and pick the top L matching designs with the smallest matching distances. If the Ground truth design of this sherd is among the identified top L designs, we count it as a correct matching under rank L. We repeat this for all 292 sherds and calculate the accuracy, i.e., the percentage of the correctly matched sherds, under each rank $L = 1, 2, \cdots, 29$. This way, we can draw a CMC curve in terms of rank L as shown in Figure 4.8, which reflects the performance of curve structure segmentation – the higher the CMC curve, the better the segmentation performance.

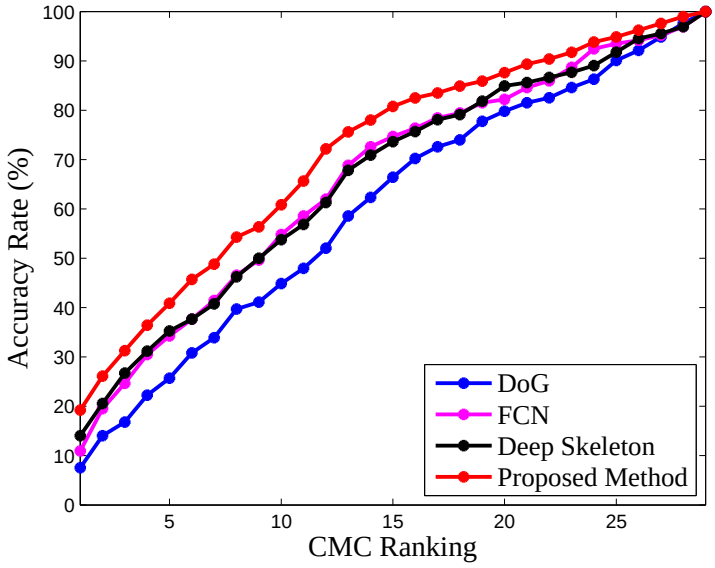


Figure 4.8: CMC curves of the proposed method and three comparison methods.

Besides the proposed method, we select three other representative comparison segmentation methods for performance evaluation in this experiment. These three comparison methods are DoG, FCN and DeepSkeleton. Figure 4.8 shows the CMC curves of the proposed method and these three comparison methods in the task of design matching. The proposed method achieves a CMC rank-1 rate of 20% and a CMC rank-15 rate of 78%, which are much better than the other three comparison

methods.

4.4 Chapter Summary

In this chapter, we put forward a novel and challenging image segmentation problem: weak curve structure segmentation from noisy depth images, which has important applications in archaeology for exploring large collections of fragmented cultural heritage objects. We developed a new three-step supervised-learning based method to address this problem, by first extracting and refining the skeletons of underlying curve structures and then producing the final segmentation by recovering the curve width at each skeleton pixel. In the experiment, we tested the proposed method on a dataset of depth images scanned from unearthed pottery sherds from southeastern North America. We found that the proposed method performs better than several widely used low-level and deep-learning based image segmentation methods in terms of F-measure.

Chapter 5

One-shot anatomical structure segmentation with explicit shape prior

5.1 Problem Overview

Segmentation of anatomical structures serves as a core element in a wide spectrum of medical image analysis applications. Recent advances in deep learning research have significantly boosted the accuracy of medical image segmentation. However, without abundant pixel-level labels, the state-of-the-art segmentation methods [71, 22, 87, 89, 9, 30] cannot achieve their optimal performance [86]. Annotating segmentation masks for medical images is extremely time-consuming and requires specialized expertise on human anatomy and its variations. As a result, prompt solutions are demanded to train an accurate segmentation model with limited labeled data.

One-/few-shot image segmentation methods have been studied in recent years aiming to reduce the dependency on large labeled data. Knowledge transferring is widely adopted for one-/few-shot segmentation of natural images [75, 59, 23, 109]. These methods leverage on external labeled datasets (e.g., PASCAL VOC [25] and MS-COCO [92]) to learn general knowledge of segmentation and is able to transfer the knowledge to object categories given a small labeled support set. Although the object category to be segmented is not seen during training, a large labeled dataset of diversified objects is still required. In the medical image domain, especially plain X-ray, such a labeled dataset is still not available yet. More important, there is still a significant accuracy gap between existing one-/few-shot methods and fully supervised ones.

In this work, we propose an annotation-efficient anatomical structure segmentation method, termed *Contour Transformer Network* (CTN). Our work is inspired by the human annotator's capability of learning segmentation of anatomical structure from one or very few exemplars. This is achieved by understanding the shape and appearance traits of the target object from the exemplars and actively looking for objects with similar traits in new images. To mimic this behavior, we propose a semi-supervised learning approach that exploits the shape and appearance similari-

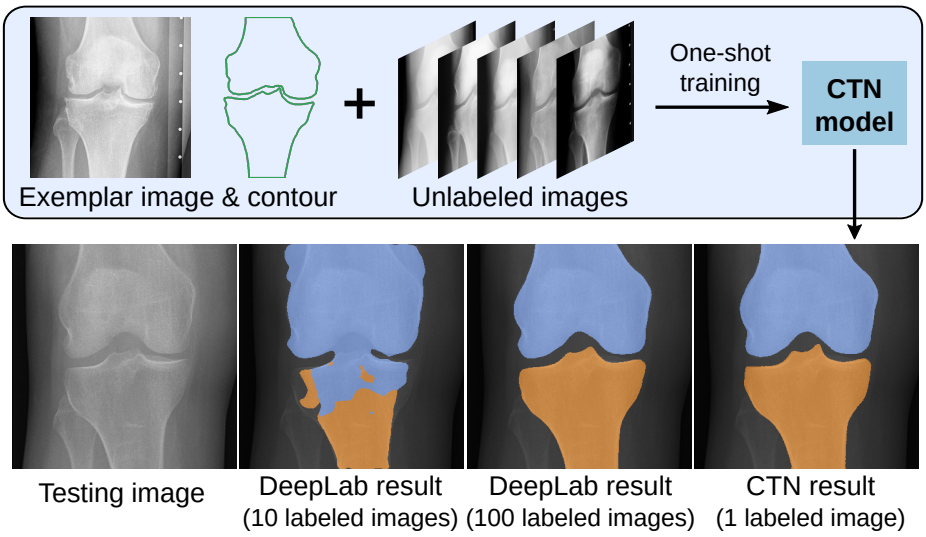


Figure 5.1: An overview of CTN. CTN could learn to segment the anatomical structure accurately from only one exemplar and a set of unlabeled images. In contrast, fully supervised methods such as DeepLab [9] will fail when training with insufficient labeled images.

ties of the target object between labeled and unlabeled images to train a segmentation model. As a result, CTN is able to learn segmentation from one labeled exemplar and a set of unlabeled images without dependency on external labeled datasets (Fig. 5.1).

Owing to the inherent regularized nature of anatomical structures, the same anatomy in different (X-ray) images may share common features or properties, such as the anatomical structure's *shape*, *appearance* and *gradients* along the structural object boundary. Although different images are not directly comparable, we can compare their common features only and use the exemplar segmentation to guide other unlabeled images partially, thus making CTN trainable in a one-shot setting. Specifically, we formulate the segmentation problem as learning a contour evolution behavior modeled by a cascaded graph convolutional network (GCN). Three differentiable contour-based loss functions namely *contour perceptual loss*, *contour bending loss* and *edge loss* are proposed to describe the common features of appearance, shape and edge response, respectively. For each unlabeled image, CTN takes the exemplar

contour as an initialization, then gradually evolves it under the guidance from the three losses. We evaluated CTN on four X-ray image segmentation tasks and demonstrated that it significantly outperforms previous one-shot segmentation methods and performs competitively when compared to fully supervised methods.

An efficient *human-in-the-loop* mechanism is a compelling feature for one-/few-shot segmentation in applications demanding extreme precision, e.g., measuring the joint space in X-rays. However, existing one-/few-shot methods often lack such a mechanism, leaving an accuracy gap that renders them unfit for many accuracy-critical applications. In contrast, CTN has a native human-in-the-loop mechanism that allows its performance to be improved by learning from annotation-efficient corrections. Namely, we format manual corrections as partial contours where users need to only redraw incorrectly segmented parts and leave correct parts untouched. These partial contour annotations can be naturally incorporated back into the training via an additional Chamfer loss [1]. We demonstrate that with minimum human-in-the-loop feedback, CTN can outperform fully supervised methods on all four X-ray datasets evaluated.

In summary, our contributions are four-fold: 1) We propose CTN, a one-shot anatomical structure segmentation method that can be trained using one exemplar and a set of unlabeled images, without depending on external labeled data. 2) We propose two new differentiable loss functions *contour perceptual loss* and *contour bending loss*, plus the existing *edge loss*, to enable GCNs to integrate anatomical priors of appearance, shape and gradient, respectively. 3) We design a human-in-the-loop mechanism to allow CTN to utilize additional manual labels with low annotation cost. 4) We demonstrate on four datasets that CTN achieves the state-of-the-art one-shot segmentation results, i.e., it performs competitively when compared to fully supervised alternatives and outperforms them with minimal human-in-the-loop feedback.

Method overview

The problem of anatomical structure segmentation can be decomposed into two steps: ROI (Region of Interest) detection; and ROI segmentation. ROI detection can be achieved via landmark detection and has been well-studied in past literature [105, 13, 89, 46], so we focus on achieving very high segmentation accuracy by taking the detected ROI (with noise and errors) as input images.

The training pipeline of CTN is illustrated in Fig. 5.2. Our task is to learn an segmentation model of an anatomical structure from a set of unlabeled images $\{I\}$ and an exemplar image I_E with its segmentation C_E of the target structure. We model each segmentation as a contour, represented by a fixed number of evenly spaced vertices, $C = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$. For each unlabeled image I, its contour C is initialized by placing the exemplar contour C_E at the center of the image. CTN models the contour evolution policy that displaces the initial contour C to the boundary of the target structure in I. It can be written as:

$$F_{\boldsymbol{\theta}}(I_E, C_E, I, C) = \Delta C \tag{5.1}$$

where $F_{\boldsymbol{\theta}}$ denotes the CTN with weights $\boldsymbol{\theta}$. It takes the exemplar and the target image as input, and outputs estimated offsets of contour vertices.

Due to the lack of labels on I, fully supervised losses cannot be used to train CTN. Here, we exploit the advantage of modeling segmentation as contour, *i.e.*, it provides natural representations of the segmentation's boundary and shape. In particular, instead of comparing model predictions with ground truth as in a fully supervised setting, we compare C with the exemplar contour C_E, by measuring the dissimilarities between their shapes and the local image patterns along with them. This is motivated by the insight that the correct segmentation in the target image should be similar to the exemplar contour in its overall shape, as well as local image appearance patterns

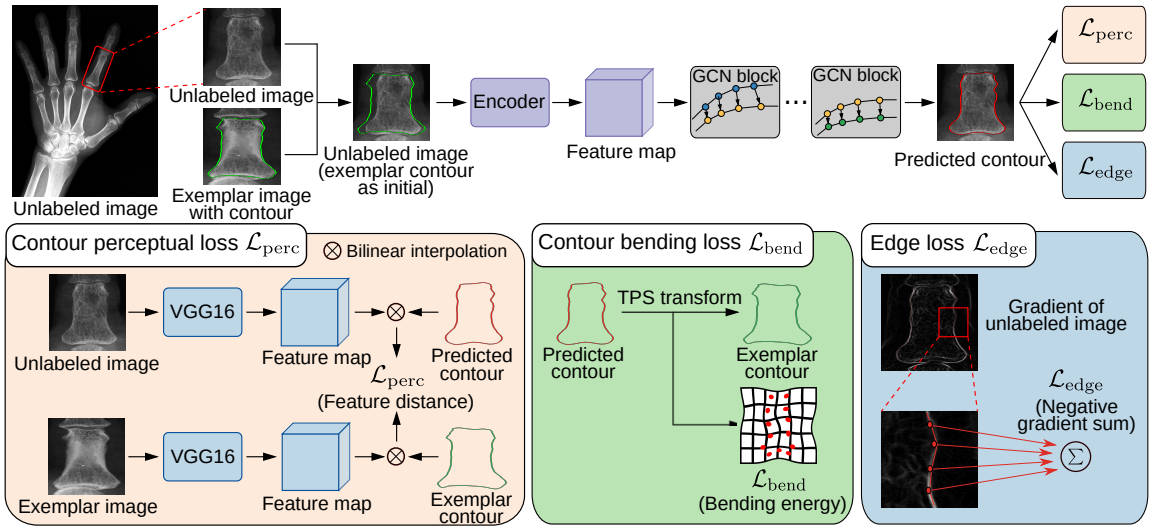


Figure 5.2: Contour Transformer Network. CTN is trained to fit a contour to the object boundary by learning from one exemplar. In training, it takes a labeled exemplar and a set of unlabeled images as input. After going through a CNN encoder and five GCN contour evolution blocks, it outputs the predicted contour. We train the network using three one-shot losses (*i.e.*, contour perceptual loss, contour bending loss and edge loss), aiming to let the predicted contour have similar contour features with the exemplar.

of corresponding vertices. As a side benefit, the predicted contours of CTN are naturally corresponded to the exemplar contour.

We propose two new losses to measure the shape and appearance dissimilarities: namely contour perceptual loss, denoted as L_{perc}, and contour bending loss, denoted as L_{bend}. In addition, we employ the classic gradient-based loss, denoted as L_{edge}, to further drive the contour to edges. Details of these losses will be described in Section 5.2. CTN is trained by minimizing weighted combination of the three losses:

$$\min_{\boldsymbol{\theta}} \sum_{\{I\}} \lambda_1 \cdot L_{perc} + \lambda_2 \cdot L_{bend} + \lambda_3 \cdot L_{edge} \tag{5.2}$$

where $\lambda_1, \lambda_2, \lambda_3$ are weighting factors of the three losses. An illustration of the training process of CTN is shown in Fig. 5.2.

These losses imitate the human's behavior in learning contouring from one exemplar, *i.e.*, drawing new contours by referring to the exemplar to compare shapes and

local appearances. Another key insight is that although these losses can be used in an ACM setting (where the contour vertices are directly optimized to minimize the energy), training CTN on aggregating over the entire unlabeled dataset is robust, stable and can inhibit the boundary leaking issue on individual cases often encountered by ACM.

Network architecture

Following [50], we use a CNN-GCN architecture to model contour evolution. As shown in Fig. 5.2, CTN consists of two parts: an image encoding CNN block and subsequent cascaded contour evolution GCN blocks. ResNet-50 [31] is employed as the backbone of the image encoding block. It takes the target image as input and outputs a feature map encoding local image appearances, denoted as:

$$f = F_{cnn}(I). \tag{5.3}$$

All contour evolution blocks have the same multi-layer GCN structure, although weights are not shared. The GCN takes the contour graph with vertex features as input, denoted as $G = (C, E, Q)$, where C denotes the contour vertices, E denotes the connectivity, and Q denotes the vertex features. Each vertex in the contour is connected to four neighboring vertices, two on each side. The vertex features are extracted from the feature map f at vertex locations via bilinear interpolation, which can be written as:

$$Q = \{f(\mathbf{p})\}_{\mathbf{p}\in C} \tag{5.4}$$

where $f(\mathbf{p})$ denotes the result of bilinear interpolation of f at location $\mathbf{p}$.

Five GCN blocks are cascaded to evolve the contour. The k-th block takes the graph $G_k = (C_k, E, Q_k)$ as input, and outputs offsets of the contour vertices:

$$C_{k+1} = C_k + F^k_{gcn}(C_k, E, Q_k). \tag{5.5}$$

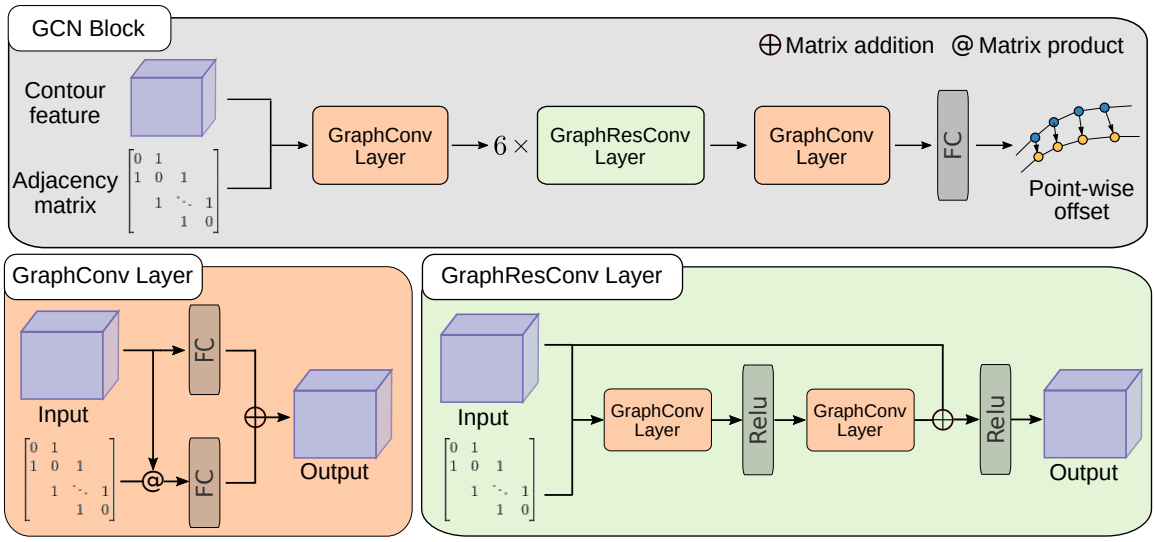


Figure 5.3: Network Architecture of GCN blocks. CTN uses five cascaded GCN blocks to model the contour evolution behavior. They take image features along the contour and an adjacency matrix that represents vertex connections as input, and predict point-wise offsets to update the contour. Their architectures are identical, but weights are not shared.

The contour is initialized using the exemplar contour, $C_0 = C_E$, and the output of the last contour evolution block is the final output.

The architecture of GCN blocks is shown in Fig 5.3. Each GCN block consists of 2 graph convolutional (GraphConv) layers [38], 6 graph residual convolutional (GraphResConv) layers [101] and 1 fully connected (FC) layer. The first GraphConv layer and all GraphResConv layers have 256 channels. The last GraphConv layer has 32 channels. The FC layer has 2 channels outputting the offsets on x and y axis, respectively.

One-shot training losses

Contour perceptual loss

We propose a contour perceptual loss to measure the dissimilarity between the visual patterns of the exemplar contour C_E on the exemplar image I_E and the predicted

contour C on the target image I. Partially enlightened by the perceptual loss [34] developed for image super-resolution, which measures image perceptual similarities in the feature space of VGG-Net [78], we measure contour perceptual similarities in the graph feature space. In particular, graph features are extracted from the VGG-16 feature maps of the two images along the two contours (similar to Eq. 5.4), and their L1 distance is calculated as the contour perceptual loss:

$$L_{perc} = \sum_{i=1,...,N} \|P(\mathbf{p}_i) - P_E(\mathbf{p}'_i)\|_1 \tag{5.6}$$

where $\mathbf{p}_i \in C$, $\mathbf{p}'_i \in C_E$, and P and P_E denote the VGG-16 features of I and I_E, respectively. The VGG-16 weights are trained on ImageNet [91].

Instead of using L2 distance found in the original perceptual loss formulation [34], we employ L1 distance since it empirically performed better in our experiments. Because of the inevitable appearance variations across images, we hypothesize that the similarity representation between pairs of local image patterns is often limited according to certain aspects, *e.g.*, specific texture, context, or shape features. Given that different channels of VGG-16 features capture different characteristics of local image patterns, a distance metric learning with modeling flexibility to select which salient features to match is more appropriate. The sparsity-inducing nature of L1 distance definition provides additional "selection" mechanism over L2, which may explain the improved performance observed.

Using the contour perceptual loss to measure appearance similarity between contours has a few advantages: 1) Since VGG-16 network features can capture the image pattern of a neighboring area with spatial contexts (*i.e.*, network receptive field), the contour perceptual loss enjoys a relatively large capturing range (*i.e.*, the convex region around the minimum), making the CTN training optimization easier; 2) The backbone VGG-16 model is trained on ImageNet [91] for classification tasks, so that its learned features are more sensitive to underlying structure and less sensitive to noises and illumination variations, which improves the robustness of CTN training.

Contour bending loss

If we operate under the assumption that an exemplar contour is broadly informative to other data samples, then it should be beneficial to use the exemplar shape to ground any predictions on such other samples. To this end, we propose a contour bending loss to measure the shape dissimilarity between contours. The loss is calculated as the bending energy of the TPS warping [2] that maps C_E to C. It is worth noting that TPS warping achieves the minimum bending energy among all warpings that map C_E to C. Since bending energy measures the magnitude of the 2nd order derivatives of the warping, the contour bending loss penalizes more on local and acute shape changes, which are often associated with mis-segmentation.

Given a predicted contour C, the TPS bending energy can be calculated as follows:

$$\mathbf{K} = \left(\left\| \mathbf{p}'_i - \mathbf{p}'_j \right\|_2^2 \cdot log \left\| \mathbf{p}'_i - \mathbf{p}'_j \right\|_2 \right) \tag{5.7}$$

$$\mathbf{P} = (\mathbf{1}, \mathbf{x}', \mathbf{y}') \tag{5.8}$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \tag{5.9}$$

where $\mathbf{p}_i = (x_i, y_i)$, $\mathbf{p}'_i = (x'_i, y'_i)$ are points of C and C_E, respectively. $\mathbf{x}' = \{x'_1, x'_2, \ldots, x'_N\}^T$, $\mathbf{y}' = \{y'_1, y'_2, \ldots, y'_N\}^T$. $\mathbf{K}$, $\mathbf{P}$, $\mathbf{L}$ are matrices of size $N \times N$, $N \times 3$ and $(N+3) \times (N+3)$, respectively. Finally, the TPS bending energy is written as

$$\mathcal{L}_{bend} = \max \left[\frac{1}{8\pi} (\mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{y}^T \mathbf{H} \mathbf{y}), 0 \right] \tag{5.10}$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}^T$, $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}^T$, and $\mathbf{H}$ is the $N \times N$ upper left submatrix of $\mathbf{L}^{-1}$ [102].

Edge loss

Although the contour perceptual and bending losses can achieve robust segmentation, they are inherently insensitive to (very) small segmentation fluctuations, such

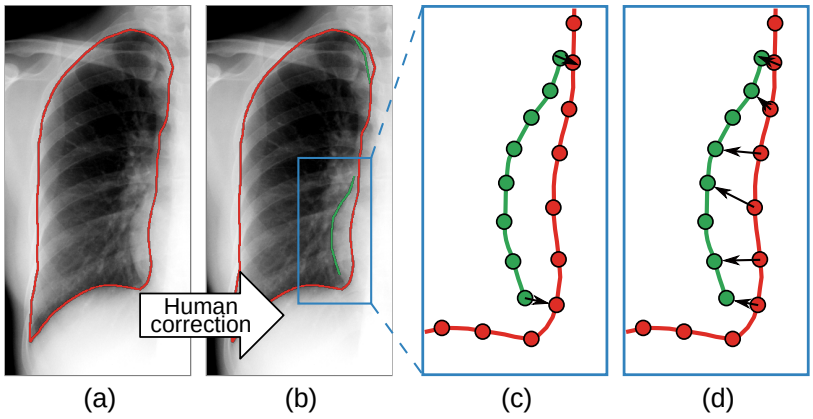


Figure 5.4: Human-in-the-loop. Given a *red* predicted contour (a), the annotator corrects its wrong parts with *green* curves (b). For each corrected contour segment, we find two points in the predicted contour, closest to its start and end (c), then each predicted point between the two points are assigned to the closest corrected point (d). This prevents the point correspondence to be scattered.

as deviations from the correct boundary by a few pixels. Therefore, in order to obtain desirably high segmentation accuracies to adequately facilitate the downstream workflows like rheumatoid arthritis quantification [32], we also employ an edge loss measuring the image gradient magnitude along the contour, which attracts the contour toward edges in the image. The edge loss is written as:

$$\mathcal{L}_{edge} = -\frac{1}{N}\sum_{\mathbf{p}\in C} \|\nabla I(\mathbf{p})\|_2 \tag{5.11}$$

where ∇ is the gradient operator.

Human-in-the-loop

Learning from one exemplar is based on the assumption that the anatomical structure has similar boundary features in all images. It works in most cases, but outliers are inevitable. To achieve even higher accuracy in testing, sometimes we need to consider more possibilities in training. To this end, the proposed CTN offers a natural way to incorporate additional labeled images with a human-in-the-loop mechanism.

Assuming a CTN model is trained with one exemplar, we want to finetune it with more segmentation annotations. We first run this model on a set of unlabeled images and select a number of images with wrong predictions as new samples. Instead of drawing the whole contour from scratch on these new images, the annotator only needs to draw some partial contours, in order to correct the wrong prediction (as shown in Fig. 5.4(b). The point-wise training of CTN makes it possible to learn from these partial corrections. This way, we reduce the labor cost to the minimum.

A *partial contour matching loss* is proposed to utilize the partial ground truth contours during the CTN training. Denote $\hat{\mathbf{C}}$ as a set of partial contours in image I, each element of which is an individual contour segment. For each contour segment $\hat{C}_i \in \hat{\mathbf{C}}$, we build the point correspondence between $\hat{C}_i$ and C. For each $\hat{C}_i$, we find two points in the predicted contour C that are closest to the start and end points of $\hat{C}_i$, then each predicted point between the two points are assigned to the closest corrected point. Denote the corresponding predicted contour segment by C_i ($C_i \in C$). We define the distance between C and $\hat{C}_i$ as the Chamfer distance from C_i to $\hat{C}_i$:

$$D(\hat{C}_i, C) = \sum_{\mathbf{p} \in C_i} \min_{\hat{\mathbf{p}} \in \hat{C}_i} \|\mathbf{p} - \hat{\mathbf{p}}\|_2 \tag{5.12}$$

and the partial matching loss of C is defined as:

$$L_{pcm} = \frac{1}{N} \sum_{\hat{C}_i \in \hat{\mathbf{C}}} D(\hat{C}_i, C). \tag{5.13}$$

In the human-in-the-loop scenario, we combine all losses to train the CTN, and rewrite the Eq. 5.2 as:

$$\min_{\boldsymbol{\theta}} \sum_{\{I\}} \lambda_1 \cdot L_{perc} + \lambda_2 \cdot L_{bend} + \lambda_3 \cdot L_{edge} + \lambda_4 \cdot L_{pcm} \tag{5.14}$$

which allows CTN to be trained with fully labeled, partially labeled and unlabeled images simultaneously and seamlessly. Whenever new labeled image are available, we can use Eq. (5.14) to finetune the existing CTN model.

Datasets and experimental settings

Datasets

We evaluate our method on four X-ray image datasets focusing on different anatomical structures of knee, lung, phalanx and hip, respectively.

- **Knee:** We randomly selected 212 knee X-ray images from the Osteoarthritis Initiative (OAI) database [1]. Each knee image is cropped from the original scan with automatic knee joint detection, and resized to 360×360 pixels. The dataset is randomly split into 100 training and 112 testing images.

- **Lung:** We use the public JSRT dataset [88] with 247 posterior-anterior chest radiographs, where lung segmentation labels originate from the SCR dataset [2][26]. Left lung and right lung ROIs are extracted from the image and resized to 512×256 pixels. Following [26], the 124 images with odd indices are used for training, and the 123 images with even indices for testing.

- **Phalanx:** We collected an in-house dataset of hand X-ray images from patients with rheumatoid arthritis. 202 ROIs of proximal phalanx are extracted from images automatically based on hand joint detection [32] and resized to 512×256 pixels. We randomly split the dataset into 100 training and 102 testing images.

- **Hip:** We randomly selected 300 pelvic X-ray images from the OAI database, 100 for training and 200 for testing. Each hip image is cropped from the original scan with automatic landmark detection, and resized to 360×360 pixels.

[1] https://nda.nih.gov/oai/

[2] https://www.isi.uu.nl/Research/Databases/SCR/

On the knee, phalanx and hip datasets, we manually annotated the target objects, namely tibia, femur, phalanx and hip bones, under the guidance of a senior rheumatologist. The image lists and annotations of the knee and hip datasets are publicly available [3].

For every dataset, we selected the most representative image in the training set as the exemplar image based on the distance to other images. Specifically, for every image in the training set, we calculate its distance to all other images in the ImageNet-trained VGG feature space, which represents the semantic similarity between the two images. The image with minimum average distance to other images is selected as the exemplar.

Evaluation metrics

For each segmentation result, we evaluate segmentation accuracy by IoU and for the corresponding object contour by the Hausdorff distance (HD). For methods that do not explicitly output object contours, we extract the external contour of the largest region of each class from the segmentation mask. On the knee dataset, we report the average HD of femur and tibia segmentation.

Implementation details

The hyper-parameter settings are $N = 1000$, $\lambda_1 = 1$, $\lambda_2 = 0.25$, $\lambda_3 = 0.1$, $\lambda_4 = 1$. The network is trained using the Adam optimizer with a learning rate of 1×10^{-4}, a weight decay of 1×10^{-4} and a batch size of 12 for 500 epochs. We use the same hyper-parameter setting in both one-shot training and human-in-the-loop finetuning.

[3] https://github.com/rudylyh/CTN_data

Table 5.1: Performances of CTN and seven existing methods on four datasets.

Methods		Knee		Lung		Phalanx		Hip		**Mean**	
		IoU(%)	HD(px)	IoU(%)	HD(px)	IoU(%)	HD(px)	IoU(%)	HD(px)	IoU(%)	HD(px)
Non-learning-based	MorphACWE [7, 58]	65.89	54.07	76.09	55.35	74.33	69.13	48.05	94.11	66.09	68.17
	MorphGAC [8, 58]	87.42	15.78	70.79	45.67	82.15	24.73	83.42	32.20	80.95	29.60
One-shot	CANet [109]	29.22	175.86	56.90	73.46	60.90	67.13	48.89	88.39	48.98	101.21
	Brainstorm [111]	90.17	29.07	77.13	43.28	80.05	30.30	82.48	44.17	82.46	36.71
	CTN (Ours)	**97.32**	**6.01**	**94.75**	**12.16**	**96.96**	**8.19**	**97.29**	**8.27**	**96.58**	**8.66**
Fully supervised	UNet [71]	96.60	7.14	95.38	12.48	95.76	10.10	96.51	13.28	96.06	10.75
	DeepLab [9]	97.18	5.41	96.18	10.81	97.63	6.52	97.64	6.24	97.16	7.25
	HRNet [89]	96.99	5.18	95.99	10.44	97.47	7.03	97.66	7.57	97.03	7.56

Comparison with existing methods

We compare CTN against seven representative methods from three categories: non-learning-based, one-shot, and fully supervised segmentation methods. The quantitative results are reported in Table 5.1 and visualizations of segmentation results are shown in Fig. 5.5.

Comparison with non-learning-based methods

We first compare with two non-learning-based methods: MorphACWE [7, 58] and MorphGAC [8, 58]. Both of them are based on ACM, which evolves an initial contour to the object by minimizing an energy function. We use the exemplar contour of our method as their initial contours.

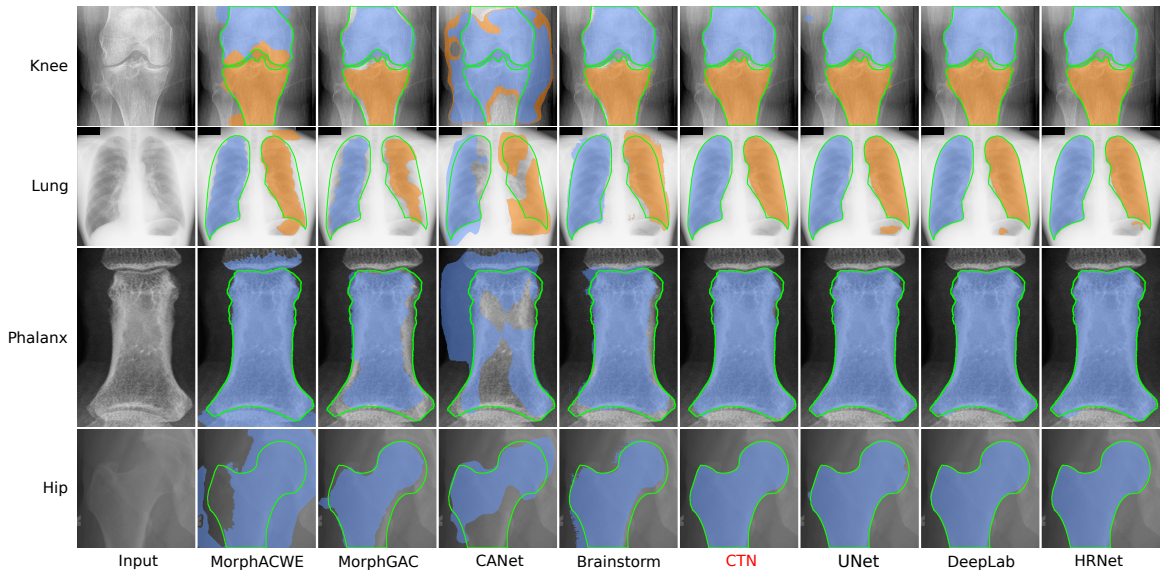


Figure 5.5: Segmentation results of four example images. The boundaries of ground truth segmentations (the green lines) are drawn for comparison.

Results in Table 5.1 show that our method significantly outperforms both MorphACWE and MorphGAC. Specifically, on average we achieve 15.63% higher IoU and 20.94 pixels less HD than MorphGAC, the better of the two. The visualizations of segmentation results in Fig. 5.5 confirm that these two approaches cannot provide satisfactory segmentation accuracy, especially when the boundary of such structures is not clear, e.g., lung segmentation. We posit that the inferior performance of ACM-based methods is owing to two factors: 1) the gradient-based energy function is not suitable for objects without clear boundary, 2) optimizing the energy function on single image often encounters local minima (i.e., causing segmentation leakage). In contrast, CTN optimizes shape and appearance-based loss functions on an aggregated of the unlabeled dataset to achieve high robustness. Fig. 5.6 shows the evolution process of the CTN contour on a phalanx image.

Comparison with one-shot methods

We also compare with two representative one-shot segmentation methods: CANet [109] and Brainstorm [111]. CANet is trained on the PASCAL VOC 2012 dataset and can segment unseen objects by referring to the support set (the exemplar). Brainstorm tackles the one-shot segmentation problem by learning both spatial and appearance transformations between images in a dataset and further synthesizes image-label pairs to train the segmentation model. We follow their procedures to train the model on our datasets. For all one-shot methods, including ours, we use the same exemplar as the one-shot data.

As shown in Table 5.1, CANet achieves only 48.98% IoU on average. We speculate that the poor performance is caused by the domain gap between natural images and medical images. Brainstorm achieves better performances with an average IoU and HD of 82.46 % and 36.71, respectively. This is still significantly lower than CTN, of which the average IoU and HD are 96.58 % and 8.66, respectively. Fig. 5.5 shows that while Brainstorm is able to segment the object's overall structure, it has low accuracy on the segmentation boundaries.

Comparison with fully supervised methods

We also evaluate the performance of three fully supervised methods on our datasets: UNet [71], DeepLab-v3+ [9] and HRNet-W18 [89]. We train each of them for 500 epochs with all available training data, *i.e.*, 100 knee images, 124 lung images, 100 phalanx images and 100 hip images. Post-processing procedures are excluded for fair comparison.

CTN trained with only one exemplar performs comparably with the fully supervised UNet, and slightly falls behind DeepLab, the best of the three, by 0.58% in IoU and 1.41 pixel in HD, respectively. These results suggest that with only one exemplar, CTN can compete head-to-head with very strong fully supervised baselines. We note

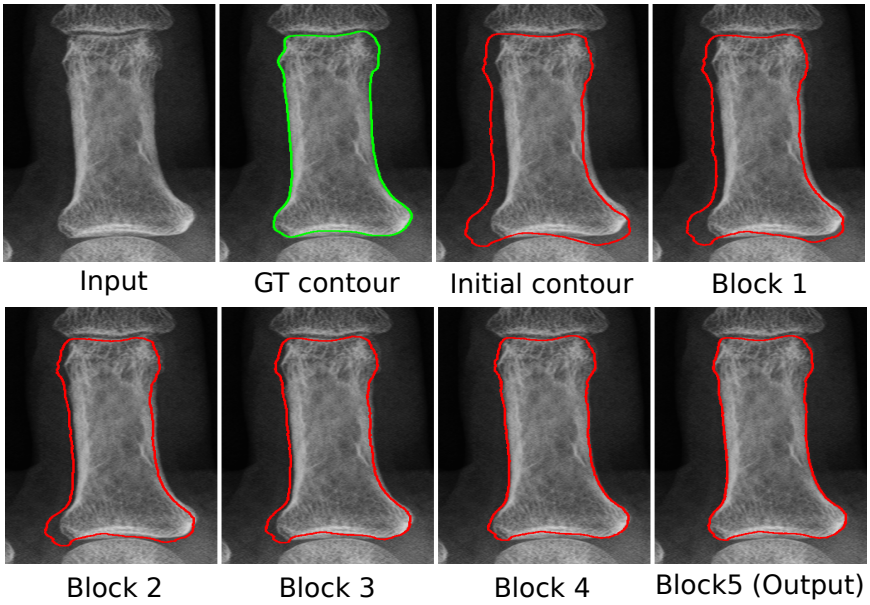


Figure 5.6: Visualization of the contour evolution process. The red lines are the contours after each GCN block in CTN. It shows how CTN gradually moves the initial contour to the correct location.

that since these fully supervised methods predict segmentation labels at pixel-level, the topology of the segmentation is not guaranteed, *e.g.*, small isolated lung masks in Fig. 5.5. In contrast, CTN is able to retain the topology. Moreover, we will demonstrate in Section 5.3 that with minimal human feedback, CTN can even outperform fully supervised models.

Incorporating human corrections

In this section, we validate the effectiveness of the proposed human-in-the-loop mechanism by simulating manual corrections of wrong segmentation by an annotator. Specifically, we assume that the annotator tends to correct more severe errors with higher priority. To simulate this behavior, we first segment the unlabeled training images using the one-shot trained model and calculate their HD to the ground-truth segmentation (which is not used in training). Then, we select the worst $n\%$ images as candidates for correction. For each predicted contour in these images, we calculate its

point-wise L2 distances to the ground-truth and mark vertices with distances larger than 3 pixels as errors. We group consecutive error vertices into segments and use the corresponding ground-truth vertices as corrections. Under this setting, we conduct human-in-the-loop training using corrections of 10%, 25% and 100% training images, respectively.

Fig. 5.7 shows the performances of the original one-shot model and three human-in-the-loop finetuned models. We observe that our model consistently improves with more corrections. Specifically, using 10% corrections, the mean IoU is improved from 96.58% to 97.10% and the mean HD is reduced from 8.66 to 7.32, respectively. When using 25% corrections, CTN can outperform DeepLab, (IoUs of 97.38% vs. 97.16%, and HDs of 6.81 vs. 7.25). With corrections on all training samples, CTN further reaches an IoU of 97.52% and a HD of 6.27. We also stress that the effort of our human-in-the-loop correction of unlabeled training samples is significantly lower than annotating them from scratch (as required by fully supervised methods), as only partial corrections are needed. Thus, these results indicate that on all 4 evaluated tasks, CTN with the human-in-the-loop mechanism can achieve superior performance than fully supervised methods and require considerably less annotation effort.

Training with more unlabeled data

Another advantage of CTN is that it can utilize more unlabeled data (which are often easy to obtain) in training to improve its performance. To evaluate the impact of more unlabeled data by expanding the unlabeled training sets of knee, hip and phalanx from 100 images to 500 images, with the exemplar unchanged. We do not conduct this experiment on the lung dataset, because there is no additional images available in the JSRT dataset.

As shown in Table 5.2, by increasing the number of unlabeled images from 100 to 500, the performance improves on average by 0.22% in IoU and 0.6 in HD. Among

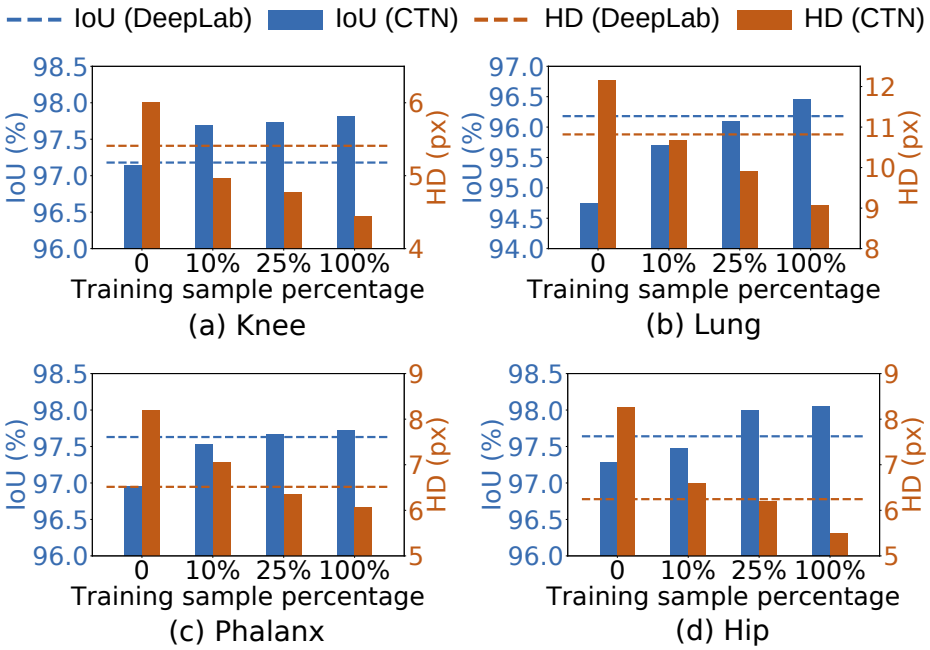


Figure 5.7: Using different number of human corrections to finetune the one-shot model. We test the performance of the human-in-the-loop mechanism with 0, 10%, 25% and 100% corrected training samples, respectively ("0" means no finetuning). Our performance with 25% training samples generally outperforms DeepLab using 100% samples.

Table 5.2: Using more unlabeled images in training. We expand the training set of knee and phalanx from 100 to 500 images to examine our method's ability in exploiting unlabeled data. Both cases use only one exemplar.

Unlabeled images	Knee		Phalanx		Hip	
	IoU(%)	HD(px)	IoU(%)	HD(px)	IoU(%)	HD(px)
100	97.32	6.01	96.96	8.19	97.29	8.27
500	97.53	5.73	97.33	6.96	97.37	7.97

the three datasets, the improvement on the phalanx dataset is the largest. Phalanx dataset has larger appearance and shape variations than hip and knee, since it contains bones from 5 fingers. We hypothesize that CTN needs more training samples to fully capture the large appearance and shape variations.

Table 5.3: Ablation study. We remove one of three losses each time and re-train the model.

L_{perc}	L_{bend}	L_{edge}	Knee		Lung	
			IoU(%)	HD(px)	IoU(%)	HD(px)
	✓	✓	94.62	8.28	87.45	26.51
✓		✓	**97.49**	**5.87**	84.93	36.74
✓	✓		94.43	11.90	93.00	16.22
✓	✓	✓	97.32	6.01	**94.74**	**12.17**

Phalanx		Hip		**Mean**	
IoU(%)	HD(px)	IoU(%)	HD(px)	IoU(%)	HD(px)
94.01	15.80	92.90	16.58	92.24	16.79
94.24	26.13	94.53	13.91	92.80	20.66
96.45	9.84	96.61	9.92	95.12	11.97
96.96	**8.19**	**97.29**	**8.27**	**96.58**	**8.66**

Ablation study on three losses

We conduct an ablation experiment to evaluate the effectiveness of the three employed losses, namely the contour perceptual loss L_{perc}, the contour bending loss L_{bend}, and the edge loss L_{edge}. The results are summarized in Table 5.3. The performance of CTN degrades if any loss is removed, with an average IoU decrease of 4.34%, 3.78%, and 1.46% for L_{perc}, L_{bend}, and L_{edge}, respectively. This demonstrates the contributions of all three losses. An exception is the knee dataset when L_{bend} is removed. Knee X-ray images share similar appearance features along the contour so that they can be segmented robustly with just the contour perceptual loss and edge loss. Thus, adding contour bending loss lead to statistically insignificant decreases (*i.e.*, IoUs of 97.32% vs 97.50%, HDs of 6.01 vs 5.87) in this particular scenario. However, such a regularization effect by the contour bending loss is generally desired to alleviate the worst-case scenarios and is proved useful in the other three datasets.

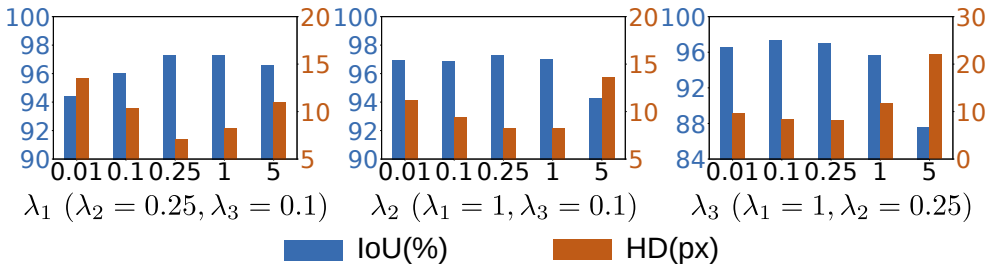


Figure 5.8: Using different loss weights to train CTN on the hip dataset. Based on the original setting $\lambda_1 = 1$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.1$, we change one of them each time and fix the other two.

Effect of loss weights

We've shown that the same loss weights $\lambda_1 = 1$, $\lambda_2 = 0.25$, $\lambda_3 = 0.1$ work well on 4 different datasets. In this section, we further analyze CTN's sensitivity to the loss weights using the hip dataset. Three experiments are conducted to evaluated the impact of λ_1, λ_2 and λ_3 individually while fixing the other two weights. The CTN is trained and tested using 5 different values [0.01, 0.1, 0.25, 1, 5] for λ_1, λ_2 and λ_3. The IoUs and HDs obtained using different weights are reported in Fig. 5.8. We observe that CTN is largely stable to weights ranging from 0.1 to 1 for all three losses, with the IoU between 95.68% to 97.29% and the HD between 7.15 to 11.82.

5.4 Chapter Summary

In this work, we presented CTN, a one-shot segmentation method that can be trained using one labeled exemplar and a set of unlabeled images. We demonstrated that by properly exploiting the regularized nature of anatomical structures, CTN trained with one labeled data (exemplar) can compete head-to-head with fully supervised methods trained with abundant labeled data. A key assumption of our work is that the same anatomy have similar shape and visual patterns in different images. Based on this assumption, CTN employs a semi-supervised training strategy with losses that measures the similarity between the segmentation from unlabeled images and the

exemplar. A key difference between CTN and most existing segmentation methods (one-shot and supervised) is that CTN models segmentation as contour and learns the contour evolution behavior. Using contour representation makes it possible to directly compare the shapes of segmentation results, as well as measure the similarity of visual appearance along the segmentation boundary. We have shown that shape similarities can be measured using TPS bending energy of the two contours and used as training loss, which is sensitive to acute shape changes and is suitable for imposing shape regularization to prevent irregular segmentation. Visual pattern similarities of two contours can be evaluated by comparing the features of corresponding vertices in the ImageNet trained VGG feature space. Since the VGG is trained on ImageNet, its feature is salient to the structure and insensitive to low level image variations, which is ideal for comparing the visual similarity of two segmentation contours.

Section 5.3 and 5.3 demonstrate that the performance of CTN can be further improved in two ways, training with more unlabeled data and incorporating human-in-the-loop corrections, respectively. By using more unlabeled training data, without addition annotation effort, CTN can reach the performance of the state-of-the-art supervised segmentation methods (e.g., DeepLab). The human-in-the-loop correction is high labor cost-effective, i.e., the annotator only needs to draw the mis-segmented partial contour. As shown in Fig.5.4, with human-in-the-loop, CTN can outperform supervised methods by a large margin, especially on HD. For one-shot learning methods to be useful in clinical applications, especially the accuracy demanding ones, the capability to effectively incorporate human-in-the-loop corrections to boost performance is a critical feature. However, most existing one-shot methods fail to provide such mechanism.

We recognize that CTN also has its limitations. The success of CTN is achieved by heavily exploiting the assumption that the target anatomical structure has similar shape and appearance in different images. If the anatomical structure has significant

difference from the exemplar in shape and/or appearance (e.g., caused by pathology), the contour bending loss and contour perceptual loss may provide misinformed guidance to CTN and we expect the performance of CTN to degrade. This limitation can be partially addressed by the human-in-the-loop mechanism with certain manual correction efforts. Another limitation of CTN is that it can only utilize one exemplar and does not support few-shot learning scenarios. This is mainly because the contour bending loss and contour perceptual loss are calculated pair-wise between the exemplar and the unlabeled images. We will investigate the extension of CTN to few-shot learning scenario via group-wise loss calculation in our future work.

Chapter 6

Few-shot natural image segmentation with class-agnostic prior

6.1 Problem Overview

Few-shot semantic segmentation (FSS) aims to segment unseen object classes in query images by referring to only few labeled (support) images [75]. As a potential solution to annotation-efficient segmentation, it has received increasing attention in recent years [75, 23, 100, 109, 53, 95, 3, 108]. Existing FSS methods typically solve this problem in a meta-learning framework. They leverage existing large-scale datasets as base classes, and organize the training data into episodes of query and support images to train a two-branch model that is agnostic to object classes [3]. These methods show good generalization ability in popular FSS benchmarks (*e.g.*, PASCAL-5^i and COCO-20^i) where the base and novel classes are from natural images. However, when the novel class comes from different domains, the model trained on base classes often fail due to the large discrepancy of feature distributions [96], as shown in Fig. 6.1. For example, in a real application that motivated this study, we have a set of unlabeled chest X-ray images and need to extract the lungs with one support image. Although the object to be segmented has relatively low variation across these images, we cannot find any existing FSS model that could extract it very well. Similar cases can be found in many other fields where the annotations are scarce and require domain expertise to label, such as material images, archaeological images and satellite images. Therefore, the domain gap greatly limits the application of FSS models.

A straightforward way to address the domain gap is to fine-tune the base model on support images of the novel class. However, due to the small size of support set in FSS, supervised fine-tuning may lead to overfitting [52]. In the field of few-shot classification, transductive fine-tuning has been proven effective to improve the base model's performance on query images of unseen domains [21]. They incorporate unlabeled query images into fine-tuning to avoid overfitting. By minimizing a supervised loss on labeled images and an unsupervised loss on unlabeled image, the base model is optimized on the entire query set. But transductive fine-tuning for

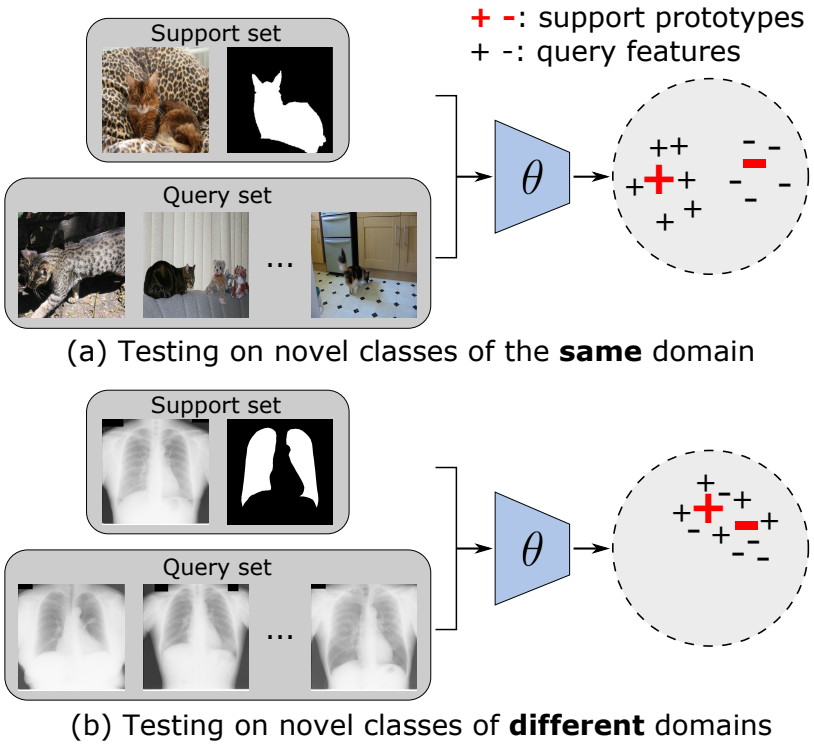


Figure 6.1: The cross-domain FSS problem. θ is a prototypical FSS model trained on natural images. When testing on medical images, its discriminability will decrease drastically due to the feature distribution discrepancy. We propose a method to fine-tune θ on the given support and query sets to bridge the domain gap.

FSS is still under explored. In the experiment, we show that applying these methods to the cross-domain FSS problem could bring considerable improvement. However, when proper designs for the segmentation problem are adopted, the cross-domain FSS performance can be further improved.

In this work, we present a transductive fine-tuning method to address the domain gap for prototypical FSS models. Instead of learning from labeled and unlabeled images using separate losses, we try to better utilize the connection between query and support images, and the core idea is to *use support labels to implicitly supervise query segmentation*. Prototypical network is a commonly-used FSS method that generates class-wise prototypes from support images, and makes prediction on query images by finding the closest prototype of each pixel [81]. No matter in the query or support images, pixels from the same class are desired to be closer in the feature

space than pixels from other classes. Therefore, in absence of query labels, we can use the labels of support images to implicitly guide the query segmentation by aligning their class-wise prototypes, thus making full use of known query and support images.

Given a set of query and support images, our goal is to fine-tune the base FSS model to generalize it to the target domain. We first extract image features using the base model, and then generate fine-grained prototypes by clustering in the feature space. A prototype contrastive loss is proposed to provide implicit supervision to query images by contrasting its prototypes against support prototypes. To avoid inaccurate query prototypes, an uncertainty factor is introduced to adjust the loss weight dynamically. Moreover, our optimization objectives also include a supervised cross-entropy loss that take advantage of support labels, and an unsupervised boundary loss that penalizes scattered query masks. With these regularizations, our semi-supervised fine-tuning strategy is more stable and more effective.

In summary, the contributions of this work include:

1. We raise a novel problem of fine-tuning meta-learned models for cross-domain FSS.

2. We propose a method to address the domain gap for prototypical FSS models by naturally incorporating unlabeled images into fine-tuning, thus extending their application range from in-domain to cross-domain.

3. We investigate the performance of representative FSS methods under various cross-domain settings; compare the performance of different fine-tuning methods in cross-domain FSS; and validate the effectiveness of our method in fine-tuning various prototypical models.

Problem setup

The goal of FSS is to learn to segment any novel classes from only a few support examples. In the term n-way k-shot segmentation, n refers to the number of novel classes (expect for the background), and k refers to the number of support examples. We denote the support set as $\mathbf{S} = \{S_1, S_2, ..., S_k\}$, where each sample S consists of a support image I_S and its ground-truth segmentation mask M_S. Similarly, the query set is denoted as $\mathbf{Q} = \{Q_1, Q_2, ...\}$, but the query mask M_Q is only available in training. The base classes for training and the novel classes for testing are denoted as C_{base} and C_{novel} respectively, where $C_{base} \cap C_{novel} = O$. Meta-learning methods usually adopt episodic training to train class-agnostic FSS models. They structure the data of C_{base} into episodes, where each episode contains a query image and a support set, thus simulating the testing environment.

When C_{base} and C_{novel} are from different domains, the class-agnostic base model may lose generalization ability due to the feature distribution discrepancy. In this work, we address this problem for prototypical FSS models. Given a prototypical model θ trained on C_{base}, our goal is to generalize θ to the domain of C_{novel} without additional labels. We assume the support set $\mathbf{S}$ and the whole query set $\mathbf{Q}$ are available in testing, and fine-tune θ on $\mathbf{S}$ and $\mathbf{Q}$, which is also referred as transductive fine-tuning.

6.2 Method

We propose Cross-image Prototype Contrast (CPC) to fine-tune prototypical models on the query and support images of unseen domains. To avoid overfitting on support images, our core idea is to incorporate unlabeled query images into training by implicitly guiding their segmentation with support prototypes. The fine-tuning also follows the episodic training scheme. As illustrated in Fig. 6.2, we use two branches to process support and query images respectively. The support branch extracts the pro-

totypes from support images, and outputs a supervised cross-entropy loss. Without any labels, the query branch leverages support prototypes to predict a query mask, and minimizes an unsupervised boundary loss. More importantly, a semi-supervised prototype contrast loss is proposed to build connection between two branches by contrasting query prototypes with support prototypes. Note that our method does not modify the internal structure of θ, so it can be applied to any prototypical encoders. The following describes each component of CPC in details.

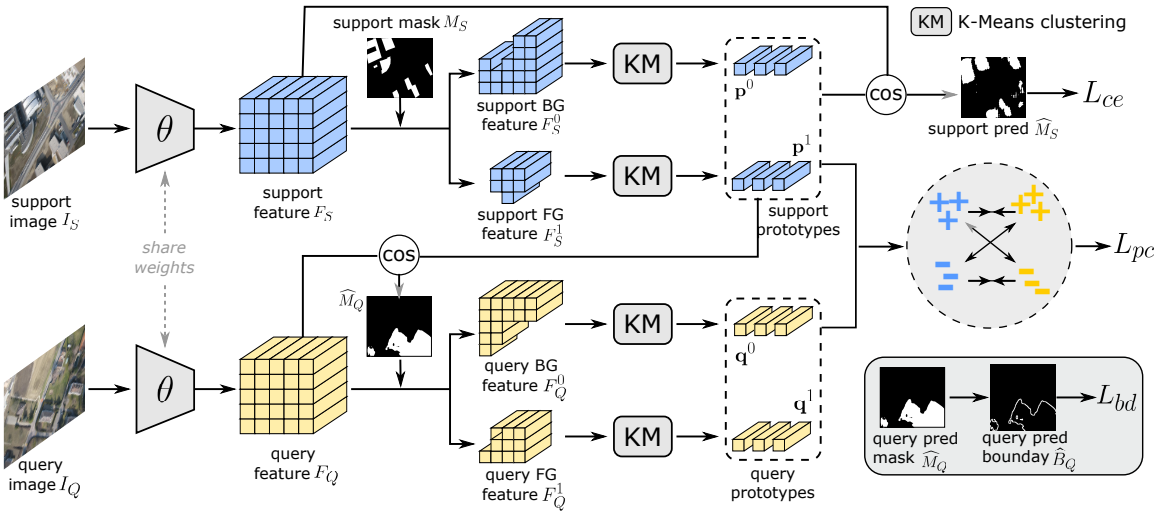


Figure 6.2: The training pipeline. We illustrate with an example of 1-way 1-shot segmentation. We first generate the fine-grained support prototypes, and then use them to predict the query mask. With the query mask, we generate the query prototypes and compare against the support ones to minimize the prototype contrastive loss L_{pc}. Besides, the support GT mask is employed to regularize the training through a cross-entropy loss L_{ce}. Finally, a boundary length loss L_{bd} is employed to penalize irregular regions in the query mask.

Prototype generation

In each episodic fine-tuning step, the encoder θ takes a random query image I_Q and the support set S as input. We first extract fine-grained prototype features on the query and support images.

Given a query image I_Q and a support image I_S, we use θ to obtain their feature

maps F_Q and F_S, where $F_Q = \theta(I_Q)$, $F_S = \theta(I_S)$. Subsequently, the support feature F_S can be split into several class-wise feature sets $\{F_S^0, F_S^1, ..., F_S^n\}$ according to the support mask M_S, where

$$F_S^i = \{F_S(x,y) | M_S(x,y) = i\}\,. \tag{6.1}$$

For k-shot segmentation, we just put all features of the same class in the same feature set to compute prototypes.

For each class, we can obtain the holistic prototype by averaging all features in the corresponding feature set. But considering the intra-class diversity of objects (*e.g.*,different types of background), the holistic prototype may be too coarse to represent the whole class. We follow [53] to utilize K-Means clustering to obtain fine-grained prototypes. For each class i, the feature set F_S^i is split into c clusters, and the cluster centers are denoted by $\{G_1^i, G_2^i, ..., G_c^i\}$. Finally, we obtain c prototypes for each class, and each prototype is the weighted sum of a cluster center and the holistic prototype, namely

$$\mathbf{p}_j^i = G_j^i + \frac{\lambda}{|F_S^i|} \sum F_S^i, \tag{6.2}$$

where $i \in [0, n]$, $j \in [1, c]$ and λ is a weight set as 0.5.

With the set of support prototypes, we are able to predict the segmentation mask of the query image I_Q. For each pixel (x, y) in I_Q, its class is predicted as the class of the support prototype that has the highest cosine similarity with its feature. The class-wise probability map of I_Q is:

$$\widehat{P}_Q^i(x,y) = softmax(\max_{j\in[1,c]} \left[\cos(\mathbf{p}_j^i, F_Q(x,y))\right]), \tag{6.3}$$

and the predicted query mask is $\widehat{M}_Q$ can be obtained by: $\widehat{M}_Q(x,y) = \arg\max_{i\in[0,n]} \widehat{P}_Q^i(x,y)$.

With the predicted mask $\widehat{M}_Q$ and the query feature map F_Q, we follow Eq.(6.1) and Eq.(6.2) to generate the fine-grained prototypes $\mathbf{q}$ of query images, which are pseudo prototypes that are supposed to be aligned with $\mathbf{p}$.

Next, we introduce the optimization objectives of CPC.

Prototype contrastive loss

The role of the prototype contrastive loss is to contrast the support prototypes $\mathbf{p}$ with the query prototypes $\mathbf{q}$. For prototypical inference, an ideal feature encoder should map pixels to a feature space where intra-class distances are smaller than inter-class distances. When testing on unseen domains, the feature encoder may not be able to extract meaningful features, thus making different classes inseparable in the feature space. Therefore, the prototype contrastive loss is designed to improve the discriminability of θ by reducing the intra-class distance and enlarging the inter-class distance from $\mathbf{p}$ to $\mathbf{q}$.

Given the support prototype $\mathbf{p}$ and the query prototype $\mathbf{q}$, the distance between the i-th class of $\mathbf{p}$ and the j-th class of $\mathbf{q}$ is computed by:

$$d(\mathbf{p}^i, \mathbf{q}^j) = 1 - \frac{1}{c} \sum_{g=1}^{c} \max_{h \in [1,c]} \cos(\mathbf{p}_h^i, \mathbf{q}_g^j). \tag{6.4}$$

Eq. (6.4) means that, for each prototype in $\mathbf{p}^i$, we compare it with only the most similar prototype in $\mathbf{q}^j$.

For the class j, the intra-class distance between the query and support prototypes is the distance between $\mathbf{p}^j$ and $\mathbf{q}^j$, and the inter-class distance is averaged over all classes in $\mathbf{p}$ except for j:

$$\begin{cases} d_{intra}^j = d(\mathbf{p}^j, \mathbf{q}^j) \\ d_{inter}^j = \dfrac{1}{n} \displaystyle\sum_{i=0 \wedge i \neq j}^{n} d(\mathbf{p}^i, \mathbf{q}^j). \end{cases} \tag{6.5}$$

Solely minimizing d_{intra} or maximizing d_{inter} will lead to collapse solutions. We employ the triplet loss to optimize them simultaneously. To avoid overly maximizing the inter-class distance to make $\mathbf{p}$ and $\mathbf{q}$ negatively correlated, we use a margin value m to balance the intra- and inter-class distances, $m = 0.2$. The prototype contrast loss is:

$$L_{pc} = \sum_{j=0}^{n} \max(d_{intra}^j - d_{inter}^j + m, 0). \tag{6.6}$$

As a semi-supervised loss, L_{pc} takes advantage of unlabeled images to enrich training data. But it is not as reliable as supervised losses. When aligning query and support prototypes, we are actually assuming the query mask that produces the query prototype is accurate. However, it is not this case in practice, especially in the beginning of fine-tuning. If the query mask is wrong, pushing the query features to the wrong support prototype may affect the fine-tuning adversely. To alleviate this problem, we propose to weight L_{pc} by a **dynamic uncertainty** during fine-tuning.

Given the probability map $\widehat{P}_Q$ of a query image, we compute its global uncertainty w_{un} as the average ratio of the second largest probability to the largest probability:

$$w_{un} = \frac{1}{|\Phi|} \sum_{(x,y)\in\Phi} \frac{\max_{i\in[0,n]\wedge i\neq j} \widehat{P}_Q^i(x,y)}{\max_{i\in[0,n]} \widehat{P}_Q^i(x,y)}, \tag{6.7}$$

where j is the class index with the largest probability, and Φ is the set of all pixels in $\widehat{P}_Q$. It indicates how confident the encoder is on the predicted query mask. We use w_{un} to dynamically adjust the weight of the semi-supervised loss.

Supervised cross-entropy loss

To make full use of support labels, we employ a cross-entropy loss to supervise the prediction on support images. Similar to Eq.(6.3), we use the support prototypes $\mathbf{p}$ to predict the softmax probability map $\widehat{P}_S$ on F_S. The cross-entropy loss of the support image is calculated as:

$$L_{ce} = -\frac{1}{(n+1)\,|\Psi|} \sum_{i=0}^{n} \sum_{(x,y)\in\Psi} M_S^i(x,y) \cdot log(\widehat{P}_S^i(x,y)), \tag{6.8}$$

where Ψ is the set of valid pixels in M_S.

L_{ce} forces θ to extract similar features for pixels of the same class in I_S. Using L_{ce} alone will lead to overfitting, especially for 1-shot segmentation, but when using with L_{pc} together, it provides strong regularization to prevent the optimization from deviating from the right direction.

Table 6.1: The IoU (%) results of using our method to fine-tune various prototypical models on four cross-domain FSS tasks.

Method	1-shot				
	Rooftop	Road	Lung	Knee	Mean
ImageNet weights	19.5	25.6	59.9	72.5	44.4
ImageNet weights + Ours	**32.8**	**30.3**	**73.7**	**73.7**	**52.6**
PANet [100]	27.8	31.4	65.1	72.0	49.1
PANet + Ours	**36.0**	**33.1**	**81.7**	**84.9**	**58.9**
PPNet [53]	22.3	31.2	61.6	73.8	47.2
PPNet + Ours	**34.5**	**39.7**	**81.6**	**83.9**	**59.9**
PFENet [95]	16.8	1.3	41.6	55.6	28.8
ReRPI [3]	5.6	8.7	65.6	76.8	39.2
Method	5-shot				
	Rooftop	Road	Lung	Knee	Mean
ImageNet weights	34.1	32.5	65.9	75.4	52.0
ImageNet weights + Ours	**40.4**	**37.1**	**78.8**	**86.1**	**60.6**
PANet [100]	37.5	34.7	65.1	73.2	52.6
PANet + Ours	**53.5**	**42.2**	**85.8**	**88.3**	**67.5**
PPNet [53]	39.4	34.9	67.8	75.3	54.4
PPNet + Ours	**55.9**	**46.7**	**88.9**	**94.8**	**71.6**
PFENet [95]	20.5	1.9	39.7	56.5	29.7
ReRPI [3]	27.4	18.6	67.8	79.6	48.4

Unsupervised boundary loss

Because of the pixel-wise classification strategy, prototypical models may produce "small islands" in the prediction mask at some ambiguous pixels, which will downgrade the visual quality of results. Therefore, we employ an unsupervised boundary loss [14] to penalize the boundary length in the query prediction $\widehat{M}_Q$, which is formulated as:

$$L_{bd} = \sum_{(x,y)\in M_Q} \sqrt{\left|(\nabla^{\mathbf{x}}_{M_Q}(x,y))^2 + (\nabla^{\mathbf{y}}_{M_Q}(x,y))^2\right|}, \tag{6.9}$$

where $\nabla^{\mathbf{x}}$ and $\nabla^{\mathbf{y}}$ are the gradient of M_Q in the $\mathbf{x}$ and $\mathbf{y}$ directions, respectively.

Finally, the optimization objective of CPC is written as:

$$L = L_{ce} + (1 - w_{un})L_{pc} + L_{bd}. \tag{6.10}$$

6.3 Experiments

Experimental setup

Datasets To evaluate the performance of our method, we resort to four public datasets from different domains, including two remote sensing and two medical image datasets: 1) **Rooftop**: Aerial images from the EPFL Rooftop dataset [83]; 2) **Road**: Satellite images for the EPFL Road Segmentation Challenge [1]; 3) **Lung**: Chest X-ray images from the SCR dataset [77] for lung segmentation; 4) **Knee**: X-ray images of knees sampled from the OAI dataset [2] [56]. Each of them has only one foreground class. Models trained on PASCAL-5^i [75] are tested on these four datasets to simulate realistic cross-domain FSS tasks.

Implementation details Our method is implemented in PyTorch [68]. We fine-tune models using the SGD optimizer with a fixed learning rate of 1×10^{-5} and a momentum of 0.9 for 1000 iterations. All images are resized to 417×417 in both training and testing. As a transductive method, we need to fix support images before fine-tuning. We select the most representative images in each dataset as the support images, and the rest are query images. We use a ResNet-50 [31] encoder pretrained on ImageNet [73] to extract all image features, and split them into k clusters by K-Means, then the cluster centers are selected as support images. For fair comparison, all methods use the same support images.

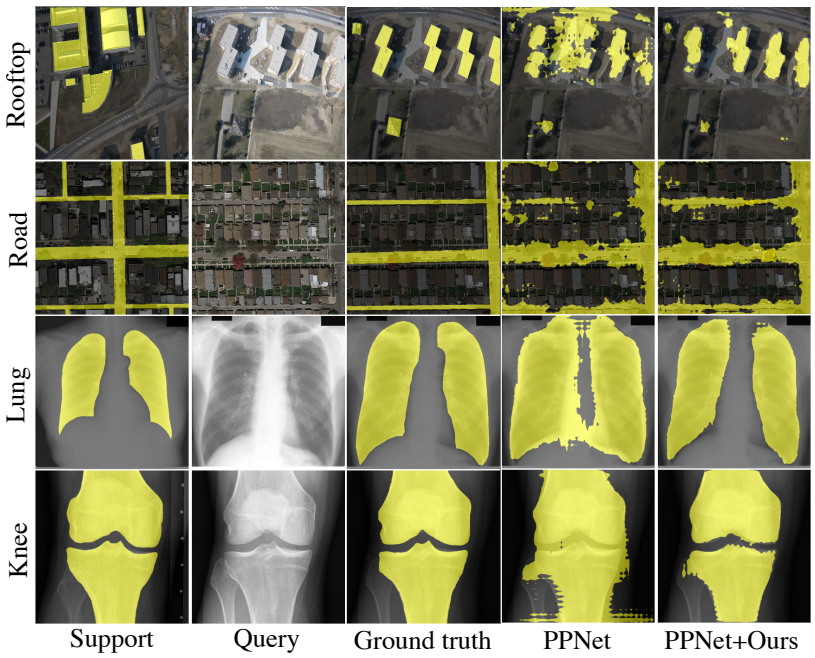


Figure 6.3: Qualitative results. By comparing the last two columns in this figure, we can observe the improvement brings by fine-tuning using our method for PPNet.

Performance evaluation

Evaluation on cross-domain FSS tasks

We first evaluate the performance of the proposed fine-tuning method on cross-domain FSS tasks. We fine-tune the models of two representative prototypical methods, PANet [100] and PPNet [53]. We also directly fine-tune the ResNet-50 pretrained on ImageNet, which is the initial weight of many FSS models. It tries to skip meta-learning to directly deploy ImageNet weights to downstream FSS tasks. Without any fine-tuning, we also evaluate a relation-based method PFENet [95] and a transductive inference method ReRPI [3] on cross-domain tasks. Except for the ImageNet weights, all models are pretrained on the fold-0 of PASCAL-5^i with a ResNet-50 backbone.

The evaluation results are reported in Table 6.1. We can see that our method sig-

[1] https://www.crowdai.org/challenges/epfl-ml-road-segmentation

[2] https://nda.nih.gov/oai

Table 6.2: The IoU (%) results of using three different methods to fine-tune PANet and PPNet on four cross-domain FSS tasks.

Method	1-shot				
	Rooftop	Road	Lung	Knee	Mean
PANet [100]	27.8	31.4	65.1	72.0	49.1
PANet + Sup-FT [51]	32.0	32.9	79.5	82.6	56.8
PANet + Trans-FT [21]	30.2	28.1	78.9	81.7	54.7
PANet + Ours	**36.0**	**33.1**	**81.7**	**84.9**	**58.9**
PPNet [53]	22.3	31.2	61.6	73.8	47.2
PPNet + Sup-FT [51]	28.9	33.2	73.1	81.5	54.2
PPNet + Trans-FT [21]	17.5	26.2	74.2	83.6	50.4
PPNet + Ours	**34.5**	**39.7**	**81.6**	**83.9**	**59.9**
Method	5-shot				
	Rooftop	Road	Lung	Knee	Mean
PANet [100]	37.5	34.7	65.1	73.2	52.6
PANet + Sup-FT [51]	47.3	39.6	85.6	89.3	65.5
PANet + Trans-FT [21]	48.4	39.5	85.2	88.2	65.3
PANet + Ours	**53.5**	**42.2**	**85.8**	**91.1**	**68.2**
PPNet [53]	39.4	34.9	67.8	75.3	54.4
PPNet + Sup-FT [51]	49.5	40.4	87.7	88.1	66.4
PPNet + Trans-FT [21]	48.5	35.9	86.1	86.2	64.2
PPNet + Ours	**55.9**	**46.7**	**88.9**	**94.8**	**71.6**

nificantly improve the performances of all three prototypical models on cross-domain FSS. Among them, we achieve the greatest improvement on PPNet with 12.7% and 17.2% higher IoU, averaged on four tasks, for 1-shot and 5-shot segmentation, respectively. The gain on medical images is more significant than it on remote sensing images, because objects with less appearance variation are more suitable for few-shot learning once the feature distribution shift is removed. Directly fine-tuning on ImageNet weights has inferior performance to fine-tuning on meta-learned models. It indicates that pretraining on large-scale base classes is beneficial. Besides, we notice that prototype-based methods have better generalization ability to large domain gap than the representative relation-based and transductive inference methods. We show the results of four example images in Fig. 6.3.

Table 6.3: The IoU (%) results of using our method to fine-tune PPNet on two in-domain FSS datasets - PASCAL-5^i and COCO-20^i.

Dataset	Method	1-Shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean
PASCAL-5^i	PPNet [53]	59.8	65.9	64.4	58.5	62.1
	PPNet + Ours	**67.3**	**67.3**	**68.0**	**58.6**	**65.3**
COCO-20^i	PPNet [53]	39.9	**41.5**	45.1	38.1	41.1
	PPNet + Ours	**41.8**	41.2	**45.7**	**39.4**	**42.0**
Dataset	Method	5-Shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean
PASCAL-5^i	PPNet [53]	70.2	71.1	71.0	60.7	68.2
	PPNet + Ours	**72.3**	**73.1**	**73.3**	**64.4**	**70.8**
COCO-20^i	PPNet [53]	49.2	52.6	48.7	**46.3**	49.2
	PPNet + Ours	**50.3**	**53.9**	**49.5**	45.7	**49.8**

Comparison with other fine-tuning methods

Furthermore, we compare with two existing fine-tuning strategies to demonstrate the superiority of our method. The first approach is fine-tuning with only support images in a supervised manner. We follow [51] to structure k support images into k^2 pairs of support-query episode, and then perform fully-supervised episodic training. The second approach is a transductive fine-tuning method proposed for few-shot classification [21]. Besides the cross-entropy loss on support images, they use an unsupervised loss to minimize the entropy of predictions on query images. We refer to these two approaches as "Sup-FT" and "Trans-FT", respectively.

The results of using all methods to fine-tune PANet and PPNet are reported in Table 6.2. From Table 6.2, we can see that: 1) Our method has better performance than the other two fine-tuning methods. We averagely outperform Sup-FT, the better of the two, by 5.7% of IoU on 1-shot and 5.2% on 5-shot, respectively; 2) When using our improvement as reference, Sup-FT and Trans-FT have inferior performance on rooftop and road than on lung and knee. In 1-shot segmentation, Trans-FT results in lower IoU on rooftop (22.3% to 17.5%) and road (31.2% to 26.2%) after fine-tuning.

Table 6.4: The IoU (%) results of using different combinations of loss terms in Eq.(6.10) to fine-tune PPNet for 1-shot segmentation.

	Rooftop	Road	Lung	Knee	Mean
w/o L_{ce}	11.8	27.7	56.7	58.9	38.8
w/o L_{pc}	28.9	33.7	74.5	82.0	54.8
w/o L_{bd}	**35.2**	38.8	80.7	79.7	58.6
all	34.5	**39.7**	**81.6**	**83.9**	**59.9**

It indicates that the information of unlabeled images is not well leveraged by directly minimizing the entropy. 3) When using 5 support images, all fine-tuning methods yield better performance, while our method is still the best of three.

Evaluation on in-domain FSS tasks

When there is no significant domain discrepancy between the base and novel classes, it is still beneficial to use our method to fine-tune the base models. To validate this point, we conduct two experiments to fine-tune the PPNet [53] model on two in-domain FSS datasets PASCAL-5^i and COCO-20^i, respectively. The detailed results are reported in Table 6.3. In Table 6.3, we can observe that our method increases the mean IoU of PPNet by 3.2% and 2.6% on 1-shot and 5-shot tasks of PASCAL-5^i, respectively. The images in COCO-20^i are more difficult, but our methods still make improvements of 0.9% and 0.6% on 1-shot and 5-shot tasks, respectively. Compared to our performance on cross-domain tasks, the improvements of our method on in-domain tasks are relatively minor, but the results still prove that our method works no matter if there is a domain gap.

Ablation studies

Impact of each loss term

In Eq.(6.10), we use three losses to optimize the fine-tuned model, respectively are the supervised cross-entropy loss L_{ce}, the prototype contrastive loss L_{pc}, and the un-

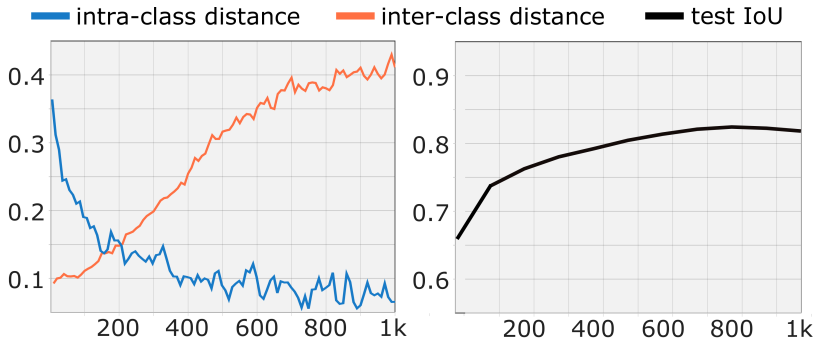


Figure 6.4: The curves of intra-, inter-class prototype distance, and test IoU v.s. fine-tuning steps.

supervised boundary loss L_{bd}. In this experiment, we remove one of three terms each time and observe how the performance changes. The IoU results of using modified losses to fine-tune PPNet for 1-shot segmentation on four tasks are reported in Table 6.4. The performance drops bring by removing L_{ce}, L_{pc} and L_{bd} are 21.1%, 5.1% and 1.3%, respectively. It indicates the supervised loss L_{ce} is the most essential term in Eq.(6.10). Without L_{ce} as regularization, the semi-supervised and unsupervised losses may not be able to find the right direction for optimization. But removing L_{pc} also results in substantial decrease, it shows the effectiveness of support prototypes in guiding query segmentation.

To better understand the behavior of L_{pc}, we visualize how the intra- and inter-class prototype distances change during fine-tuning in Fig. 6.4. We can see that, with the push and pull forces provided by L_{pc}, the intra-class distance keeps decreasing while the inter-class distance keeps increasing. It means the discriminability of the base model is improving, thus resulting higher test IoU.

Impact of the uncertainty weight

In Eq.(6.7), we introduced the dynamic uncertainty to adjust the weight of L_{pc}. The purpose was to prevent from aligning wrong query prototypes when the query prediction is not accurate. In this experiment, we remove this term from the objective

Table 6.5: The IoU (%) results of our method with and without the uncertainty weight in fine-tuning PPNet for 1-shot segmentation.

	Rooftop	Road	Lung	Knee	Mean
w/o w_{un}	31.7	**39.9**	62.1	66.5	50.1
w/ w_{un}	**34.5**	39.7	**81.6**	**83.9**	**59.9**

function Eq.(6.7) to validate its usefulness. The results of our method with and without the uncertainty weight w_{un} are reported in Table 6.5. We can see that removing w_{un} leads to a 9.8% IoU decrease on average. The decreases on medical images are more significant than them on remote sensing images. This is because the increased weight of L_{pc} diminished the influence of L_{ce} in fine-tuning, which could provide very strong supervision for medical images, even if there is only one support image. In contrast, the fine-tuning of remote sensing images relies more on unlabeled images and L_{pc}, because one support image cannot cover all types of rooftops and roads.

Changing support images

Our support images are selected as the most representative images in the deep feature space, *i.e.*, the cluster centers. In this experiment, we investigate the performance of our method in fine-tuning PPNet with different support images for 5-shot segmentation. We set five consecutive random seeds to generate the support lists. The experiment results are shown in Table 6.6. We can see that, with five different sets of random support images, the mean IoU ranges from 67.7% to 70.1%, which is relatively stable. But the selected cluster centers still achieve better average performance than random support images.

Table 6.6: The IoU (%) results of using random support images in fine-tuning PPNet for 5-shot segmentation.

Random seed	Rooftop	Road	Lung	Knee	Mean
1000	55.2	45.2	87.3	91.2	69.7
1001	56.7	41.6	86.4	91.8	69.1
1002	**57.1**	38.9	85.5	89.6	67.8
1003	51.7	43.3	86.2	89.5	67.7
1004	55.8	46.0	**89.7**	88.5	70.1
cluster centers	55.9	**46.7**	88.9	**94.8**	**71.6**

Table 6.7: The IoU(%) results on seen and unseen images. We fine-tune PPNet for 1-shot segmentation on the "seen" subset of each dataset, and test on the "unseen" subset.

Subset	Rooftop	Road	Lung	Knee	Mean
Seen	**34.3**	**37.8**	**82.3**	**83.7**	**59.5**
Unseen	30.5	36.6	80.2	81.9	57.3

Testing on unseen images

Since CPC is a transductive fine-tuning method that directly optimizes on the query set, we want to know that if the fine-tuned model can generalize to images unseen in fine-tuning. For this purpose, we evenly split each dataset to two subsets, one for fine-tuning and one for testing. Images in the test subset are from the same class but unseen in fine-tuning. For fair comparison, we perform two-fold cross validation in this experiment. The results are aggregated over two folds. We report results of this experiment in Table 6.7. As we can see, the performances of our method on unseen images is 2.2% lower than it on seen images on average, but still 10.1% higher than the base model. It indicates that our model is not overfitted on the fine-tuning sets and can used on other images of the novel class.

Discussion

With extensive experiments and ablation studies, we validated the effectiveness of our method in addressing domain gap for prototypical models. But it still has several limitations. As a transductive method, we expect to fine-tune on a bunch of query images as a whole. When we have very few query images, it may not be efficient to use our method to fine-tune models. On a single Nvidia 1080Ti GPU, our method takes approximately 11 minutes to fine-tune a ResNet-50 encoder for 1-shot segmentation and 25 minutes for 5-shot segmentation. Moreover, as a common problem of FSS methods, the selection of support images is critical to our performance. When unlabeled images are scarce, finding a representative image as support may be difficult. However, considering the application scenario of FSS, we usually have much more query images than support images in practice. In this case, fine-tuning the base model using our method is usually beneficial, especially for cross-domain tasks.

6.4 Chapter Summary

The domain gap issue seriously limited the application of existing FSS models in domains other than natural images, but was ignored by past literature. In this paper, we proposed the first transductive fine-tuning method to address this problem for prototypical FSS models. Without ground truth query masks, we employed support labels as implicit supervision to incorporate unlabeled query images into fine-tuning, which is realized by a novel uncertainty-aware semi-supervised loss. Our method could simultaneously generalize the base model to the target domain and optimize the segmentation results of the given query set. Extensive experiments on remote sensing and medical images validated the effectiveness of our methods. We hope this work could facilitate the application of FSS models in practice, and motivate more works in cross-domain FSS.

Chapter 7

Conclusion

Object shape is an important cue for human vision mechanism in visual tasks such as object classification, localization, and segmentation. In this research, we mainly study the problem of incorporating shape prior knowledge into state-of-the-art CNN image segmentation frameworks to relieve their reliance on fully annotated training data, which is difficult to acquire. Based on three representative applications, we propose different approaches to utilize three types of shape priors, respectively are implicit shape prior, explicit shape prior, and class-agnostic shape prior.

In curve structure segmentation of cultural heritage objects, we utilize the implicit shape prior of curve structures to design a skeleton extraction network. It first extracts the skeletons of curve structures, and then recover full curves through a semi-heuristic semi-automatic curve dilating algorithm, thus implicitly taking advantage of the thin and long shape prior of object of interest. It is trained with skeleton-level annotations but predicts full curves, which significantly reduces the annotation cost.

In the task of anatomical structure segmentation of medical images, a one-shot image segmentation framework is proposed to learn to segment accurate anatomical structures from only one exemplar. Anatomical structures in X-ray possess stable shape and appearance across different patients and different images, so we model their segmentation as a contour evolution problem, and start from a common initial contour. A graph convolution network is designed to evolve contours in a learnable fashion, and three one-shot trainable losses are proposed to optimize the network by explicitly constraining the shape and appearance of contours.

As last, we extend the utilization of shape priors from specific objects to general natural object segmentation. By leveraging existing large-scale fully annotated datasets, we first train a meta-learned base segmentation model. To enable the base model to generalize to any unseen object, we propose a semi-supervised transductive fine-tuning method to adapt the model to unseen domains with the help of a few labeled images, of which the basic idea is to align the class-wise prototypes of labeled

and unlabeled images.

In all three tasks, experiment results show that the proposed methods achieved promising performance with economic annotation costs, outperforming either fully supervised alternatives or state-of-the-art weakly supervised competitors. We believe this research demonstrates the practical value and potential of annotation-efficient semantic segmentation.

Future work

In this dissertation, we explained how to utilize the shape prior knowledge in CNNs to segment specific and general objects. For specific objects, we can achieve comparable results with fully supervised methods, but need to design customized method to utilize the shape of object of interest. The proposed few-shot segmentation method could work for general objects by utilizing class-agnostic shape prior, but there is still a performance gap when compared to fully supervised methods. Therefore, it is desirable if there exists an approach that works for arbitrary objects with decent performance and requires low annotation cost.

For the next step, we plan to improve the proposed few-shot segmentation method to be more accurate without loss of generality. Designing a new network for every object class is not scalable, while using a general network for all classes is not accurate enough, so we expect to achieve a balance between two options. Our idea is to add a set of adjustable parameters to allow users to customize the fine-tuning step in the current meta-learning framework. Each parameter corresponds to a characteristic of the object of interest, and controls the weight of a loss term in fine-tuning. The main parameters will include the shape consistency, the appearance consistency, the location consistency and the number of objects. The shape consistency decides how strict the shape of predicted segmentation needs to be consistent with the shape in support labels. For objects with stable shapes, such as anatomical structures,

the parameter should be high; The appearance consistency decides if the class-wise features of predicted segmentation needs to be consistent with the support features. For objects with similar appearance but various shapes, the parameter should to be high while the shape consistency should be low. Similarly, the location consistency is for those objects that have a specific location in the image, while the number of objects controls how many instances we want in the segmentation result. The above information are prior knowledges that are easy for human to perceive but difficult to learn in a data-driven manner. In this way, these knowledges can be easily incorporated into network training through modifying only a few parameters. We expect this framework to improve the performance of the current few-shot segmentation method on specific objects by a large margin.

Bibliography

[1] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf, *Parametric correspondence and chamfer matching: Two new techniques for image matching*, Tech. report, SRI International Menlo Park Ca Artificial Intelligence Center, 1977.

[2] F. L. Bookstein, *Principal warps: Thin-plate splines and the decomposition of deformations*, IEEE Transactions on Pattern Analysis and Machine Intelligence **11** (1989), no. 6, 567–585.

[3] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz, *Few-shot segmentation without meta-learning: A good transductive inference is all you need?*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13979–13988.

[4] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001), no. 11, 1222–1239.

[5] Yuri Boykov and Vladimir Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, IEEE transactions on pattern analysis and machine intelligence **26** (2004), no. 9, 1124–1137.

[6] Bettye J Broyles, *Reconstructed designs from swift creek complicated stamped sherds*, Southeastern Archaeological Conference Bulletin, 1968.

[7] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, International Journal of Computer Vision **22** (1997), no. 1, 61–79.

[8] T. F. Chan and L. A. Vese, *Active contours without edges*, IEEE Transactions on Image Processing **10** (2001), no. 2, 266–277.

[9] L.-C. Chen, K. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, Proceedings of the European Conference on Computer Vision, 2018, pp. 801–818.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, *Semantic image segmentation with deep convolutional nets and fully connected crfs*, arXiv preprint arXiv:1412.7062 (2014).

[11] ———, *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, **40** (2017), no. 4, 834–848.

[12] Liyi Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang, *Weakly supervised semantic segmentation with boundary exploration*, European Conference on Computer Vision, Springer, 2020, pp. 347–362.

[13] R. Chen, Y. Ma, N. Chen, D. Lee, and W. Wang, *Cephalometric landmark detection by attentive feature pyramid fusion and regression-voting*, International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019, pp. 873–881.

[14] Xu Chen, Bryan M Williams, Srinivasa R Vallabhaneni, Gabriela Czanner, Rachel Williams, and Yalin Zheng, *Learning active contour models for medical image segmentation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11632–11640.

[15] Yunmei Chen, Hemant D Tagare, Sheshadri Thiruvenkadam, Feng Huang, David Wilson, Kaundinya S Gopinath, Richard W Briggs, and Edward A Geiser, *Using prior shapes in geometric active contours in a variational framework*, International Journal of Computer Vision **50** (2002), no. 3, 315–328.

[16] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber, *Deep neural networks segment neuronal membranes in electron microscopy images*, Advances in neural information processing systems, 2012, pp. 2843–2851.

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, *The cityscapes dataset for semantic urban scene understanding*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[18] James Coughlan, Alan Yuille, Camper English, and Dan Snow, *Efficient deformable template detection and localization without user initialization*, Computer Vision and Image Understanding **78** (2000), no. 3, 303–319.

[19] D. Cremers, M. Rousson, and R. Deriche, *A review of statistical approaches to*

level set segmentation: integrating color, texture, motion and shape, International Journal of Computer Vision **72** (2007), no. 2, 195–215.

[20] Daniel Cremers, *Image segmentation with shape priors: Explicit versus implicit representations.*, Handbook of Mathematical Methods in Imaging **2** (2015), 1909–1944.

[21] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto, *A baseline for few-shot image classification*, International Conference on Learning Representations, 2020.

[22] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. B. Ayed, *Hyperdense-net: A hyper-densely connected cnn for multi-modal image segmentation*, IEEE Transactions on Medical Imaging **38** (2018), no. 5, 1116–1126.

[23] N. Dong and E. Xing, *Few-shot semantic segmentation with prototype learning.*, Proceedings of the British Machine Vision Conference, vol. 1, 2018, p. 6.

[24] Jinming Duan, Ghalib Bello, Jo Schlemper, Wenjia Bai, Timothy JW Dawes, Carlo Biffi, Antonio de Marvao, Georgia Doumoud, Declan P O'Regan, and Daniel Rueckert, *Automatic 3d bi-ventricular segmentation of cardiac images by a shape-refined multi-task deep learning approach*, IEEE transactions on medical imaging **38** (2019), no. 9, 2151–2164.

[25] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The pascal visual object classes (voc) challenge*, International Journal of Computer Vision **88** (2010), no. 2, 303–338.

[26] B. Van Ginneken, M. B. Stegmann, and M. Loog, *Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database*, Medical Image Analysis **10** (2006), no. 1, 19–40.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[28] Leo Grady, *Random walks for image segmentation*, IEEE transactions on pattern analysis and machine intelligence **28** (2006), no. 11, 1768–1783.

[29] Radim Halir, *An automatic estimation of the axis of rotation of fragments of archaeological pottery: A multi-step model-based approach*, Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media, 1999.

[30] A. P. Harrison, Z. Xu, K. George, L. Lu, R. M. Summers, and D. J. Mollura, *Progressive and multi-path holistically nested neural networks for pathological lung segmentation from ct images*, International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp. 621–629.

[31] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[32] Y. Huo, K. L. Vincken, D. van der Heijde, M. J. H. De Hair, F. P. Lafeber, and M. A. Viergever, *Automatic quantification of radiographic finger joint space width of patients with early rheumatoid arthritis*, IEEE Transactions on Biomedical Engineering **63** (2015), no. 10, 2177–2186.

[33] Zongliang Ji and Olga Veksler, *Weakly supervised semantic segmentation: From box to tag and back*, (2021).

[34] J. Johnson, A. Alahi, and Li F.-F., *Perceptual losses for real-time style transfer and super-resolution*, Proceedings of the European Conference on Computer Vision, 2016, pp. 694–711.

[35] Martin Kampel and Robert Sablatnig, *Rule based system for archaeological pottery classification*, Pattern Recognition Letters **28** (2007), no. 6, 740–747.

[36] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision **1** (1988), no. 4, 321–331.

[37] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele, *Simple does it: Weakly supervised instance and semantic segmentation*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 876–885.

[38] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, arXiv preprint arXiv:1609.02907 (2016).

[39] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár, *Panoptic segmentation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9404–9413.

[40] Alexander Kolesnikov and Christoph H Lampert, *Seed, expand and constrain: Three principles for weakly-supervised image segmentation*, European conference on computer vision, Springer, 2016, pp. 695–711.

[41] Philipp Krähenbühl and Vladlen Koltun, *Efficient inference in fully connected crfs with gaussian edge potentials*, Advances in neural information processing systems **24** (2011).

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in neural information processing systems **25** (2012).

[43] Louisa Lam, Seong-Whan Lee, and Ching Y Suen, *Thinning methodologies-a comprehensive survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 9, 869–885.

[44] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, *Backpropagation applied to handwritten zip code recognition*, Neural computation **1** (1989), no. 4, 541–551.

[45] M. C. H. Lee, K. Petersen, N. Pawlowski, B. Glocker, and M. Schaap, *Tetris: Template transformer networks for image segmentation with shape priors*, IEEE Transactions on Medical Imaging **38** (2019), no. 11, 2596–2606.

[46] Weijian Li, Yuhang Lu, Kang Zheng, Haofu Liao, Chihung Lin, Jiebo Luo, Chi-Tung Cheng, Jing Xiao, Le Lu, Chang-Fu Kuo, and Shun Miao, *Structured landmark detection via topology-adapting deep graph learning*, arXiv preprint arXiv:2004.08190 (2020).

[47] Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang, *Segmentation using superpixels: A bipartite graph partitioning approach*, 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 789–796.

[48] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun, *Scribblesup: Scribble-supervised convolutional networks for semantic segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3159–3167.

[49] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, *Feature pyramid networks for object detection*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[50] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, *Fast interactive object annotation with curve-gcn*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5257–5266.

[51] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu, *Crnet: Cross-reference networks for few-shot segmentation*, 2020, pp. 4165–4173.

[52] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang, *Learning to propagate labels: Transductive propagation network for few-shot learning*, International Conference on Learning Representations, 2019.

[53] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He, *Part-aware prototype network for few-shot semantic segmentation*, Springer, 2020, pp. 142–158.

[54] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[55] Jonathan Long, Evan Shelhamer, and Trevor Darrell, *Fully convolutional networks for semantic segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[56] Yuhang Lu et al., *Contour transformer network for one-shot segmentation of anatomical structures*, IEEE Transactions on Medical Imaging (2020).

[57] Yuhang Lu, Jun Zhou, Jing Wang, Jun Chen, Karen Smith, Colin Wilder, and Song Wang, *Curve-structure segmentation from depth maps: A cnn-based approach and its application to exploring cultural heritage objects*, Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[58] P. Marquez-Neila, L. Baumela, and L. Alvarez, *A morphological approach to curvature-based evolution of curves and surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence **36** (2013), no. 1, 2–17.

[59] C. Michaelis, I. Ustyuzhaninov, M. Bethge, and A. S. Ecker, *One-shot instance segmentation*, arXiv preprint arXiv:1811.11507 (2018).

[60] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos, *Image segmentation using deep learning: A survey*, arXiv preprint arXiv:2001.05566 (2020).

[61] Zahra Mirikharaji and Ghassan Hamarneh, *Star shape prior in fully convolutional networks for skin lesion segmentation*, International Conference on Med-

ical Image Computing and Computer-Assisted Intervention, Springer, 2018, pp. 737–745.

[62] Laurent Najman and Michel Schmitt, *Watershed of a continuous function*, Signal Processing **38** (1994), no. 1, 99–112.

[63] Richard Nock and Frank Nielsen, *Statistical region merging*, IEEE Transactions on pattern analysis and machine intelligence **26** (2004), no. 11, 1452–1458.

[64] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han, *Learning deconvolution network for semantic segmentation*, Proceedings of the IEEE international conference on computer vision, 2015, pp. 1520–1528.

[65] Masoud S Nosrati and Ghassan Hamarneh, *Incorporating prior knowledge in medical image segmentation: a survey*, arXiv preprint arXiv:1607.01092 (2016).

[66] Nobuyuki Otsu, *A threshold selection method from gray-level histograms*, IEEE transactions on systems, man, and cybernetics **9** (1979), no. 1, 62–66.

[67] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille, *Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation*, Proceedings of the IEEE international conference on computer vision, 2015, pp. 1742–1750.

[68] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., *Pytorch: An imperative style, high-performance deep learning library*, Advances in neural information processing systems **32** (2019), 8026–8037.

[69] Pedro O Pinheiro and Ronan Collobert, *From image-level to pixel-level labeling with convolutional networks*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1713–1721.

[70] Hariharan Ravishankar, Rahul Venkataramani, Sheshadri Thiruvenkadam, Prasad Sudhakar, and Vivek Vaidya, *Learning and incorporating shape models for semantic segmentation*, International conference on medical image computing and computer-assisted intervention, Springer, 2017, pp. 203–211.

[71] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.

[72] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, *Grabcut: Interactive foreground extraction using iterated graph cuts*, ACM Transactions on Graphics **23** (2004), no. 3, 309–314.

[73] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., *Imagenet large scale visual recognition challenge*, International journal of computer vision **115** (2015), no. 3, 211–252.

[74] Thomas Schoenemann and Daniel Cremers, *Globally optimal image segmentation with an elastic shape prior*, 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–6.

[75] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, *One-shot learning for semantic segmentation*, arXiv preprint arXiv:1709.03410 (2017).

[76] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Zhijiang Zhang, and Xiang Bai, *Object skeleton extraction in natural images by fusing scale-associated deep side outputs*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 222–230.

[77] Junji Shiraishi, Shigehiko Katsuragawa, Junpei Ikezoe, Tsuneo Matsumoto, Takeshi Kobayashi, Ken-ichi Komatsu, Mitate Matsui, Hiroshi Fujita, Yoshie Kodera, and Kunio Doi, *Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules*, American Journal of Roentgenology **174** (2000), no. 1, 71–74.

[78] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556 (2014).

[79] Greg Slabaugh and Gozde Unal, *Graph cuts segmentation using an elliptical shape prior*, IEEE International Conference on Image Processing 2005, vol. 2, IEEE, 2005, pp. II–1222.

[80] Karen Y Smith and Vernon James Knight, *Style in swift creek paddle art*, Southeastern Archaeology **31** (2012), no. 2, 143–156.

[81] Jake Snell, Kevin Swersky, and Richard S Zemel, *Prototypical networks for few-shot learning*, arXiv preprint arXiv:1703.05175 (2017).

[82] Francis H Snow, *Swift creek designs and distributions: A south georgia study*, Early Georgia **3** (1975), no. 2, 38–59.

[83] Xiaolu Sun, C Mario Christoudias, and Pascal Fua, *Free-shape polygonal object localization*, European Conference on Computer Vision, Springer, 2014, pp. 317–332.

[84] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, *Learning to compare: Relation network for few-shot learning*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1199–1208.

[85] Richard Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.

[86] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, *Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation*, Medical Image Analysis (2020), 101693.

[87] H. R. Roth *et al*, *Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation*, Medical Image Analysis **45** (2018), 94–107.

[88] J. Shiraishi *et al*, *Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules*, American Journal of Roentgenology **174** (2000), no. 1, 71–74.

[89] J. Wang *et al*, *Deep high-resolution representation learning for visual recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).

[90] O. Oktay *et al*, *Anatomically constrained neural networks (acnns): application to cardiac image enhancement and segmentation*, IEEE Transactions on Medical Imaging **37** (2017), no. 2, 384–395.

[91] O. Russakovsky *et al*, *Imagenet large scale visual recognition challenge*, International Journal of Computer Vision **115** (2015), no. 3, 211–252.

[92] T.-Y. Lin *et al*, *Microsoft coco: Common objects in context*, European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[93] Y. Lu *et al*, *Learning to segment anatomical structures accurately from one exemplar*, arXiv preprint arXiv:2007.03052 (2020).

[94] Y. Zhou *et al*, *Prior-aware neural network for partially-supervised multi-organ segmentation*, Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 10672–10681.

[95] Z Tian, H Zhao, M Shu, Z Yang, R Li, and J Jia, *Prior guided feature enrichment network for few-shot segmentation.*, (2020).

[96] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang, *Cross-domain few-shot classification via learned feature-wise transformation*, 2020.

[97] Paul Vernaza and Manmohan Chandraker, *Learning random-walk label propagation for weakly-supervised semantic segmentation*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7158–7166.

[98] Luminita A Vese and Tony F Chan, *A multiphase level set framework for image segmentation using the mumford and shah model*, International Journal of Computer Vision **50** (2002), no. 3, 271–293.

[99] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang, *Boundary perception guidance: A scribble-supervised semantic segmentation approach*, IJCAI International joint conference on artificial intelligence, 2019.

[100] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng, *Panet: Few-shot image semantic segmentation with prototype alignment*, Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9197–9206.

[101] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, *Pixel2mesh: Generating 3d mesh models from single rgb images*, Proceedings of the European Conference on Computer Vision, 2018, pp. 52–67.

[102] S. Wang, B. Munsell, and T. Richardson, *Correspondence establishment in statistical shape modeling: Optimization and evaluation*, Statistical Shape and Deformation Analysis, Elsevier, 2017, pp. 67–87.

[103] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, *Generalizing from a few examples: A survey on few-shot learning*, arXiv preprint arXiv: 1904.05046 (2019).

[104] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan, *Stc: A simple to complex framework for weakly-supervised semantic segmentation*, IEEE transactions on pattern analysis and machine intelligence **39** (2016), no. 11, 2314–2320.

[105] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, *Look at boundary: A boundary-aware face alignment algorithm*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2129–2138.

[106] Saining Xie and Zhuowen Tu, *Holistically-nested edge detection*, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.

[107] Jingshan Xu, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang, *Scribble-supervised semantic segmentation inference*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15354–15363.

[108] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao, *Mining latent classes for few-shot segmentation*, arXiv preprint arXiv:2103.15402 (2021).

[109] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, *Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5217–5226.

[110] Luming Zhang, Yue Gao, Yingjie Xia, Ke Lu, Jialie Shen, and Rongrong Ji, *Representative discovery of structure cues for weakly-supervised image segmentation*, IEEE transactions on multimedia **16** (2013), no. 2, 470–479.

[111] A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca, *Data augmentation using learned transformations for one-shot medical image segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8543–8553.

[112] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, *Pyramid scene parsing network*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.

[113] Jun Zhou, Haozhou Yu, Karen Smith, Colin Wilder, Hongkai Yu, and Song Wang, *Identifying designs from incomplete, fragmented cultural heritage objects by curve-pattern matching*, Journal of Electronic Imaging **26** (2017), no. 1, 011022–011022.