Contour Transformer Network for One-Shot Segmentation of Anatomical Structures

Yuhang Lu, Kang Zheng, Weijian Li, Yirui Wang, Adam P. Harrison, Chihung Lin, Song Wang, *Senior Member, IEEE*, Jing Xiao, Le Lu, *Senior Member, IEEE*, Chang-Fu Kuo, and Shun Miao

***Abstract*— Accurate segmentation of anatomical structures is vital for medical image analysis. The state-of-the-art accuracy is typically achieved by supervised learning methods, where gathering the requisite expert-labeled image annotations in a scalable manner remains a main obstacle. Therefore, annotation-efficient methods that permit to produce accurate anatomical structure segmentation are highly desirable. In this work, we present *Contour Transformer Network* (CTN), a one-shot anatomy segmentation method with a naturally built-in human-in-the-loop mechanism. We formulate anatomy segmentation as a contour evolution process and model the evolution behavior by graph convolutional networks (GCNs). Training the CTN model requires only one labeled image exemplar and leverages additional unlabeled data through newly introduced loss functions that measure the global shape and appearance consistency of contours. On segmentation tasks of four different anatomies, we demonstrate that our one-shot learning method significantly outperforms non-learning-based methods and performs competitively to the state-of-the-art fully supervised deep learning methods. With minimal human-in-the-loop editing feedback, the segmentation performance can be further improved to surpass the fully supervised methods.**

***Index Terms*— Image segmentation, one-shot segmentation, graph convolutional network, human-in-the-loop.**

Manuscript received October 16, 2020; revised November 29, 2020; accepted December 5, 2020. Date of publication December 8, 2020; date of current version September 30, 2021. *(Corresponding author: Yuhang Lu.)*

Yuhang Lu was with PAII Inc., Bethesda, MD 20817 USA. He is now with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: yuhang@email.sc.edu).

Kang Zheng, Yirui Wang, Adam P. Harrison, Le Lu, and Shun Miao are with PAII Inc., Bethesda, MD 20817 USA (e-mail: zhengkang86@gmail.com; yiruiwang06@gmail.com; adam.p.harrison@gmail.com; tiger.lelu@gmail.com; miaoshun638@paii-labs.com).

Weijian Li was with PAII Inc., Bethesda, MD 20817 USA. He is now with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: weijianatusa@gmail.com).

Chihung Lin and Chang-Fu Kuo are with Chang Gung Memorial Hospital, Linkou 33305, Taiwan, R.O.C. (e-mail: lin3031@gmail.com; zandis@gmail.com).

Song Wang is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: songwang@cec.sc.edu).

Jing Xiao is with Ping An Technology, Shenzhen 518029, China (e-mail: xiaojing661@pingan.com.cn).

Digital Object Identifier 10.1109/TMI.2020.3043375

I. Introduction

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 [7], [12], [18], [31], [38], [40] cannot achieve their optimal performance [35]. 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 transfer is widely adopted for one-/few-shot segmentation of natural images [13], [29], [33], [52]. These methods leverage on external labeled datasets (e.g., PASCAL VOC [14] and MS-COCO [44]) 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 similarities 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. 1).

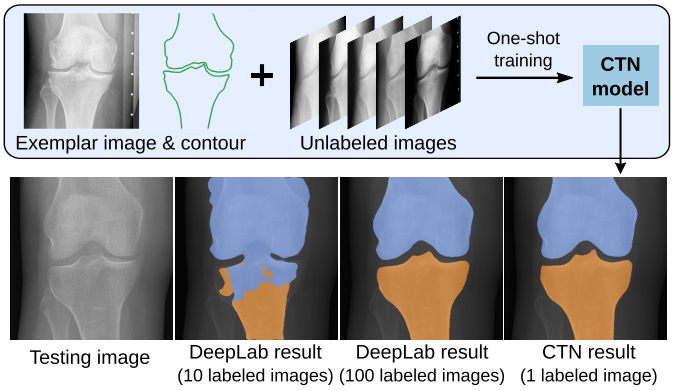

Fig. 1. **An overview of CTN**. CTN could learn to segment the anatomical structure accurately from only one exemplar and a set of unlabeled images. In contrast, fully supervised methods such as DeepLab [7] will fail when training with insufficient labeled images.

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 [2]. 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.

A preliminary version of this work has been published in a conference proceeding [47]. In this paper, we made the following extensions: 1) we add evaluations on a new dataset of hip X-ray images and provide more result analysis and discussion; 2) We conduct new experiments to further analyze the behavior of the proposed method, including evaluations with more unlabeled images, different loss weights and different exemplar images, analysis on failure and corner cases; 3) We add more comprehensive discussion on the relationship/comparison between our work and related work, more detailed technical description of the proposed method and in-depth discussion of the limitations and our future work.

A. Related Work

1) Non-Learning-Based Segmentation: Classic segmentation methods include solutions based on directly optimizing a pre-defined energy function. Well known examples include level-set [10], active contour model (ACM) [23], graph-cut [4], random walker [16] and their variants [5], [6], [28]. Although classic methods have limited performance and are no longer state-of-the-art, their essential concepts and philosophy remain insightful. We adopt the contour evolution scheme from ACM by representing segmentation using contours. Instead of optimizing the gradient-based energy function on individual images to obtain a segmentation, we optimize compound losses concerning shape, appearance, and gradient on the whole training set to learn a contour evolution policy.

Atlas and multi-atlas methods can also perform segmentation task given only one or a few examples [1], [9], [21]. However, the required image registration is a challenging task by itself [30], and inter-subject image appearance variance can lead to inaccurate registration and segmentation [53].

2) Supervised Segmentation: State-of-the-art supervised learning based segmentation methods are predominantly using deep learning, specifically fully convolutional network (FCN) [27] and its variants [7], [12], [31], [40]. These methods follow a per-pixel classification framework, where each pixel is classified individually by the deep neural network. Lacking constraints from a global structure, deep learning segmentation methods typically require a large number of labeled images to be trained effectively. When training data quantities are insufficient, the performance tends to degrade significantly, as shown in Fig. 1.

Incorporating anatomical priors into neural network training has been proven useful in recent studies. [41] employs a shape regularization autoencoder in a segmentation network to

constrain the prediction to follow a learned shape distribution. [25] takes a shape template as an additional input channel and deforms it to match the underlying structure through a spatial transformer network. While these methods exploit shape prior to improve segmentation robustness, they still require a large number of labeled images. In contrast, CTN exploits the shape and appearance commonality between labeled and unlabeled images to achieve one-shot segmentation.

Learning-based ACMs have also been studied to segment a variety of objects including heart [32], [46], blood vessel [17] and building [17], [37]. These methods incorporate deep learning and ACM by learning the ACM energy terms [37] or evolution directions [32], constructing a ACM-inspired network architecture [17] and loss [46]. While CTN also learns the contour evolution policy, it differs from [32] by introducing novel losses to support one-shot learning and employs GCN to allow effective information exchange along the contour.

3) One-/Few-Shot Segmentation: One-/few-shot segmentation methods aim to segment objects of a new category learned from a small support set of labeled examples [50]. Existing works on natural images [13], [29], [33], [52] mostly leverage the pre-training on a large and comprehensive annotated dataset like MS-COCO [44]. This condition renders the above approach inapplicable to the medical image domain, where such equivalent large labeled datasets simply do not exist, especially when considering a collection of specialized anatomical structure and imaging modality to be addressed.

Data augmentation is another common approach to solve this problem in medical imaging. Various generative models have been used in recent works to generate synthetic training data, such as variational autoencoders [11], generative adversarial networks [36], [43] and transformation networks [53]. A comprehensive survey of using imperfect datasets in medical image segmentation can be found in [35]. Zhao *et al.* [53] propose an approach to model both spatial and appearance transformations between images in the entire dataset, and synthesize images with the learned transformations for training segmentation models. Our work is partially inspired by the same motivation, but instead of learning a data augmentation model, we exploit the inherent regularized nature of anatomical structures, using one exemplar to guide the segmentation.

II. Method

A. 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 [8], [40], [45], [51], 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. 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:

$$F_{\boldsymbol{\theta}}(I_E, C_E, I, C) = \Delta C \tag{1}$$

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 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 II-C. CTN is trained by minimizing weighted combination of the three losses:

$$\min_{\boldsymbol{\theta}} \sum_{\{I\}} \lambda_1 \cdot L_{perc} + \lambda_2 \cdot L_{bend} + \lambda_3 \cdot L_{edge} \tag{2}$$

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. 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.

B. Network Architecture

Following [26], we use a CNN-GCN architecture to model contour evolution. As shown in Fig. 2, CTN consists of two parts: an image encoding CNN block and subsequent cascaded contour evolution GCN blocks. ResNet-50 [19] 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:

$$f = F_{cnn}(I). \tag{3}$$

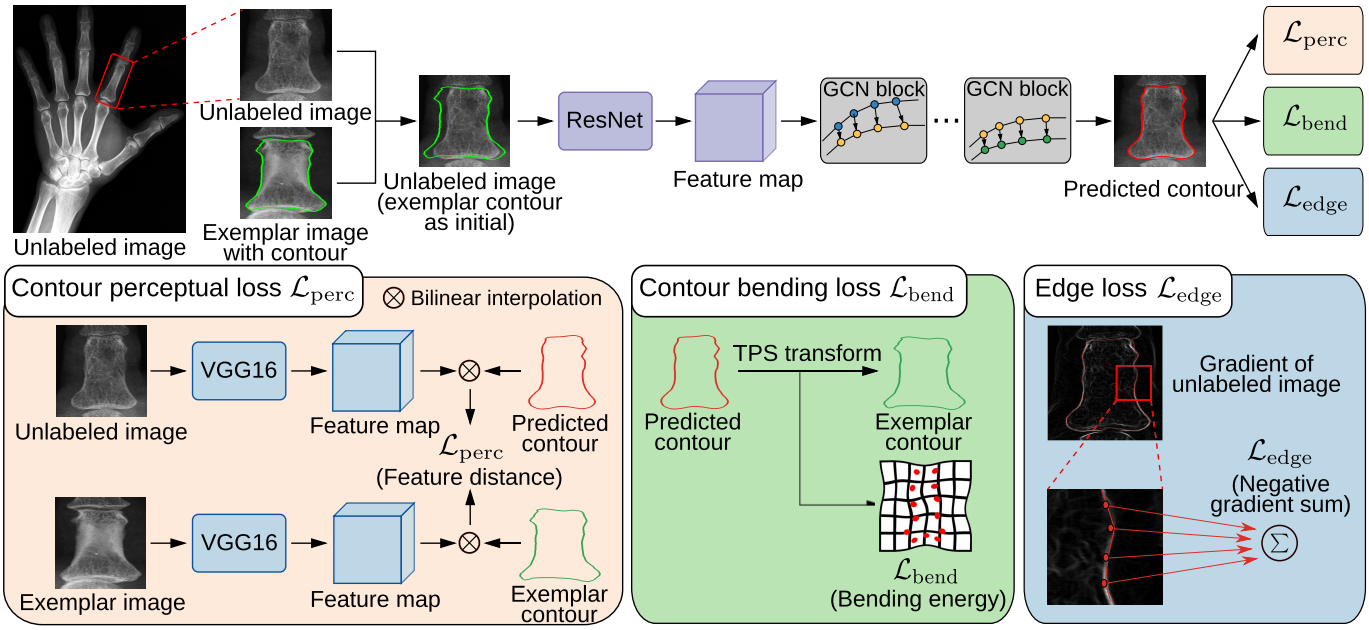

Fig. 2. **Contour Transformer Network.** CTN is trained to fit a contour to the object boundary by learning from one exemplar. In training, it takes a labeled exemplar and a set of unlabeled images as input. After going through a CNN encoder and five GCN contour evolution blocks, it outputs the predicted 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.

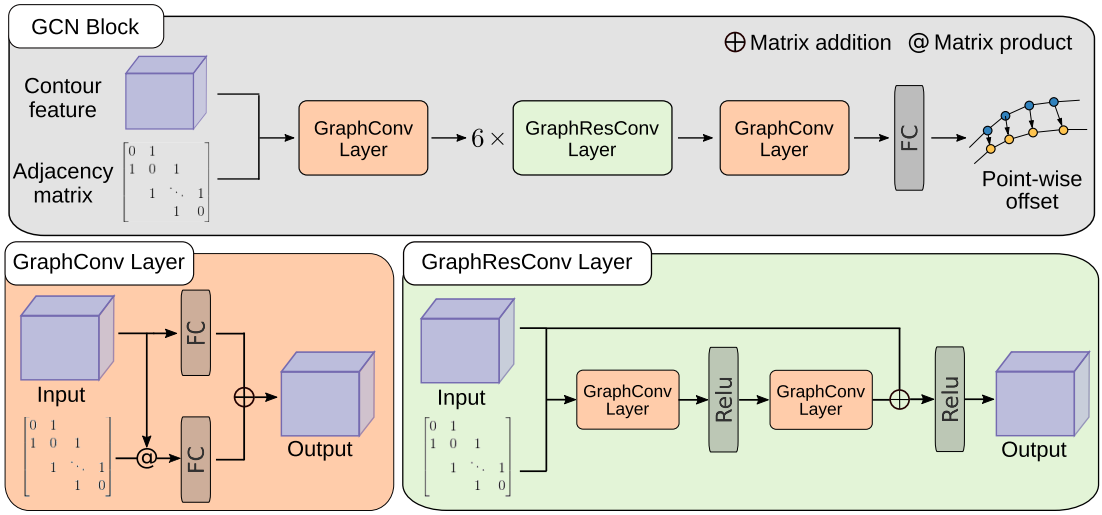

Fig. 3. **Network Architecture of GCN blocks.** CTN uses five cascaded GCN blocks to model the contour evolution behavior. They take image features along the contour and an adjacency matrix that represents vertex connections as input, and predict point-wise offsets to update the contour. Their architectures are identical, but weights are not shared.

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:

$$Q = \{f(\mathbf{p})\}_{\mathbf{p}\in C} \tag{4}$$

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:

$$C_{k+1} = C_k + F^k_{gcn}(C_k, E, Q_k). \tag{5}$$

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 3. Each GCN block consists of 2 graph convolutional (GraphConv) layers [24], 6 graph residual convolutional (GraphResConv) layers [48] 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.

C. One-Shot Training Losses

1) 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 [22] developed for image super-resolution, which measures image perceptual similarities in the feature space of VGG-Net [34], 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. 4), and their L1 distance is calculated as the contour perceptual loss:

$$L_{perc} = \sum_{i=1,\dots,N} \|P(\mathbf{p}_i) - P_E(\mathbf{p}'_i)\|_1 \tag{6}$$

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. Following [22], we use features at the layers of `relu1_2`, `relu2_2`, `relu3_3`, and `relu4_3` in VGG-16. We first downscale the contour vertices using the downsample factor of each feature map, and then sample contour features from feature maps using bilinear interpolation, and last the contour features of four layers are concatenated for comparison. The VGG-16 weights are pretrained on ImageNet [42].

Instead of using L2 distance found in the original perceptual loss formulation [22], 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 [42] 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.

2) 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 [3] 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:

$$\mathbf{K} = \left(\left\| \mathbf{p}'_i - \mathbf{p}'_j \right\|_2^2 \cdot log \left\| \mathbf{p}'_i - \mathbf{p}'_j \right\|_2 \right) \tag{7}$$

$$\mathbf{P} = (\mathbf{1}, \mathbf{x}', \mathbf{y}') \tag{8}$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \tag{9}$$

where $\mathbf{p}_i = (x_i, y_i)$, $\mathbf{p}'_i = (x'_i, y'_i)$ are points of C and C_E, respectively. $\mathbf{x}' = \{x'_1, x'_2, \dots, x'_N\}^T$, $\mathbf{y}' = \{y'_1, y'_2, \dots, y'_N\}^T$. $\mathbf{K}, \mathbf{P}, \mathbf{L}$ 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

$$\mathcal{L}_{bend} = \max \left[\frac{1}{8\pi} (\mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{y}^T \mathbf{H} \mathbf{y}), 0 \right] \tag{10}$$

where $\mathbf{x} = \{x_1, x_2, \dots, x_N\}^T$, $\mathbf{y} = \{y_1, y_2, \dots, y_N\}^T$, and $\mathbf{H}$ is the $N \times N$ upper left submatrix of $\mathbf{L}^{-1}$ [49].

3) Edge Loss: Although the contour perceptual and bending losses can achieve robust segmentation, they are inherently insensitive to (very) small segmentation fluctuations, such 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 [20], 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:

$$\mathcal{L}_{edge} = -\frac{1}{N} \sum_{\mathbf{p} \in C} \|\nabla I(\mathbf{p})\|_2 \tag{11}$$

where ∇ is the gradient operator.

D. 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. 4(b). The point-wise training of CTN makes it possible

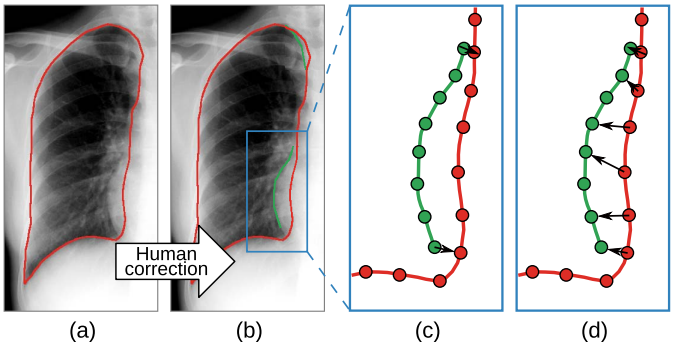

Fig. 4. **Human-in-the-loop**. Given a **red** predicted contour (a), the annotator corrects its wrong parts with **green** curves (b). For each corrected contour segment, we find two points in the predicted contour, closest to its start and end (c), then each predicted point between the two points are assigned to the closest corrected point (d). This prevents the point correspondence to be scattered.

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 $\hat{C}_i$:

$$D(\hat{C}_i, C) = \sum_{\mathbf{p} \in C_i} \min_{\hat{\mathbf{p}} \in \hat{C}_i} \left\| \mathbf{p} - \hat{\mathbf{p}} \right\|_2 \tag{12}$$

and the partial matching loss of C is defined as:

$$L_{pcm} = \frac{1}{N} \sum_{\hat{C}_i \in \hat{\mathbf{C}}} D(\hat{C}_i, C). \tag{13}$$

In the human-in-the-loop scenario, we combine all losses to train the CTN, and rewrite the Eq. 2 as:

$$\min_{\boldsymbol{\theta}} \sum_{\{I\}} \lambda_1 \cdot L_{perc} + \lambda_2 \cdot L_{bend} + \lambda_3 \cdot L_{edge} + \lambda_4 \cdot L_{pcm} \tag{14}$$

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. (14) to finetune the existing CTN model.

III. Experiments

A. Datasets and Experimental Settings

1) 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 [39] with 247 posterior-anterior chest radiographs, where lung segmentation labels originate from the SCR dataset [15][2]. Left lung and right lung ROIs are extracted from the image and resized to 512×256 pixels. Following [15], 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 [20] 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.

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 the knee and lung segmentation tasks, where there are multiple objects to be segmented, we train separate CTNs to segment the objects.

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.

2) 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.

3) 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.

B. 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

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

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

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

TABLE I
PERFORMANCES OF CTN AND EIGHT 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 [28]	65.89±6.07	54.07±3.77	76.09±6.39	55.35±17.82	74.33±6.49	69.13±10.66	48.05±4.70	94.11±9.90	66.09	68.17
	MorphGAC [28]	87.42±1.87	15.78±3.02	70.79±4.16	45.67±6.92	82.15±5.15	24.73±7.21	83.42±4.43	32.20±10.44	80.95	29.60
One-shot	CANet [52]	29.22±3.63	175.86±9.74	56.90±7.09	73.46±12.03	60.90±7.02	67.13±7.09	48.89±16.26	88.39±23.35	48.98	101.21
	Brainstorm [53]	90.17±1.72	29.07±5.32	77.13±4.71	43.28±8.38	80.05±5.17	30.30±6.90	82.48±3.18	44.17±9.29	82.46	36.71
	CTN (Ours)	**97.32±0.67**	**6.01±1.42**	**94.75±1.97**	**12.16±5.87**	**96.96±1.29**	**8.19±4.49**	**97.29±0.72**	**8.27±3.06**	**96.58**	**8.66**
Fully supervised	UNet [31]	96.60±1.61	7.14±4.24	95.38±1.87	12.48±6.40	96.76±1.76	10.10±6.84	96.51±4.22	13.28±14.55	96.31	10.75
	DeepLab [7]	97.18±0.67	5.41±2.27	96.18±1.40	10.81±6.26	97.63±0.93	6.52±3.32	97.64±0.72	6.24±2.63	97.16	7.25
	HRNet [40]	96.99±0.65	5.18±2.52	95.99±1.39	10.44±6.03	97.47±1.31	7.03±4.43	97.66±2.38	7.57±6.71	97.03	7.56

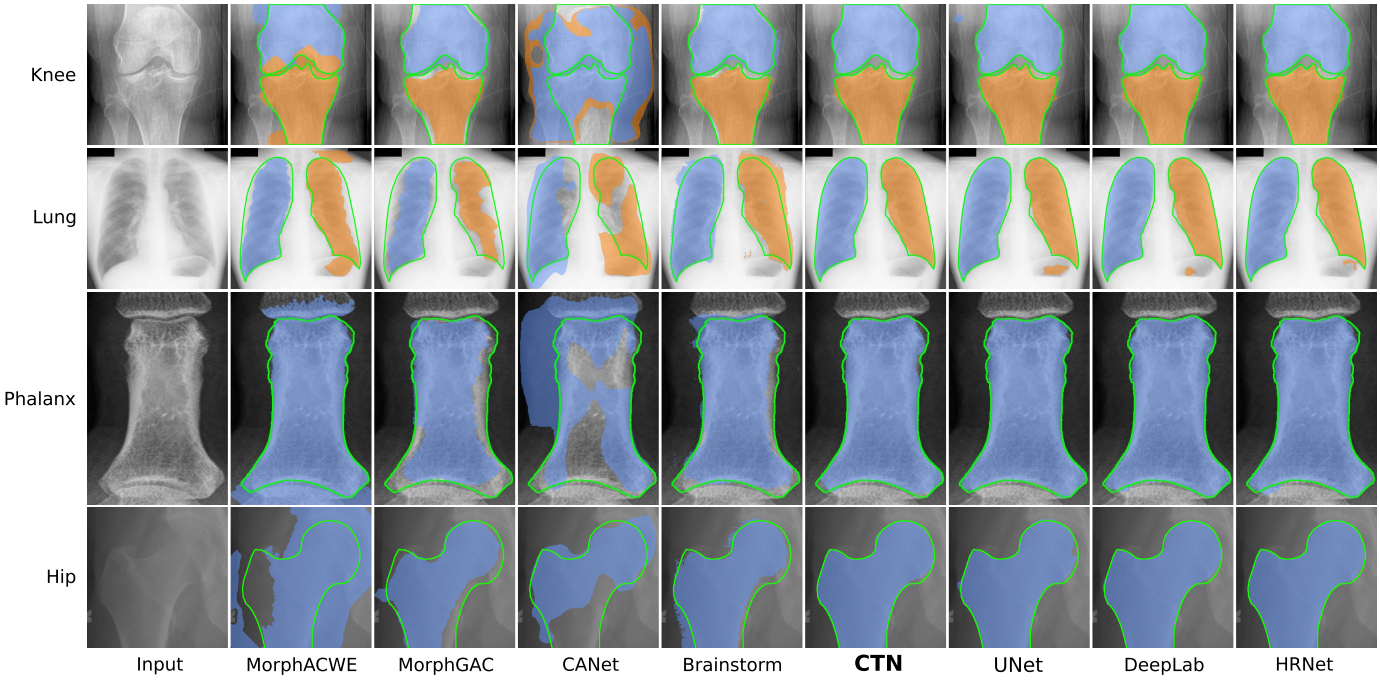

Fig. 5. **Segmentation results of four example images.** The boundaries of ground truth segmentations (the green lines) are drawn for comparison.

reported in Table I and visualizations of segmentation results are shown in Fig. 5.

1) Comparison With Non-Learning-Based Methods: We first compare with two non-learning-based methods: MorphACWE [5], [28] and MorphGAC [6], [28][4]. 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.

The results in Table I 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 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. 6 shows the evolution process of the CTN contour on a phalanx image.

2) Comparison With One-Shot Methods: We also compare with two representative one-shot segmentation methods: CANet [52][5] and Brainstorm [53][6]. 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 process images in our datasets. For all one-shot methods, including ours, we use the same exemplar as the one-shot data.

[4]https://github.com/pmneila/morphsnakes

[5]https://github.com/icoz69/CaNet

[6]https://github.com/xamyzhao/brainstorm

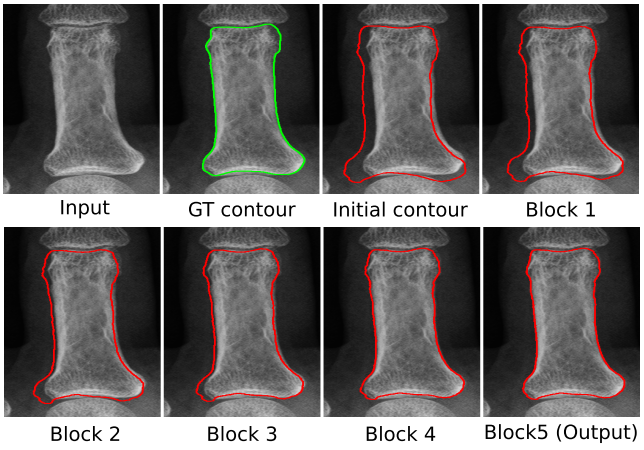

Fig. 6. **Visualization of the contour evolution process**. The red lines are the contours after each GCN block in CTN. It shows how CTN gradually moves the initial contour to the correct location.

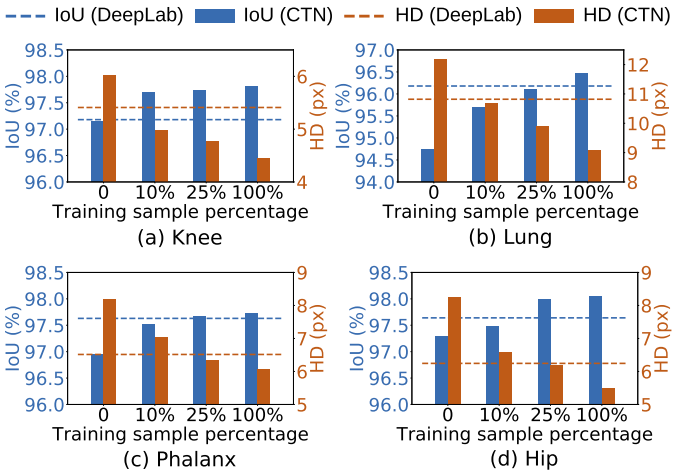

Fig. 7. **Using different number of human corrections to finetune the one-shot model**. We test the performance of the human-in-the-loop mechanism with 0, 10%, 25% and 100% corrected training samples, respectively ("0" means no finetuning). Our performance with 25% training samples generally outperforms DeepLab using 100% samples.

As shown in Table I, 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 shows that while Brainstorm is able to segment the object's overall structure, it has low accuracy on the segmentation boundaries.

3) Comparison With Fully Supervised Methods: We also evaluate the performance of three fully supervised methods on our datasets: UNet [31][7], DeepLab-v3+ [7][8] and HRNet-W18 [40][9]. We train each of them with all available training data, *i.e.*, 100 knee images, 124 lung images, 100 phalanx images, and 100 hip images, respectively. 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 baseline methods, 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 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. In contrast, CTN is able to retain the topology. Moreover, we will demonstrate in Section III-C that with minimal human feedback, CTN can even outperform fully supervised models.

C. 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. 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.

Knowing that human-in-the-loop fine-tuning improves the overall segmentation performance, we further investigate if the fine-tuned CTN may produce degraded performance on individual cases compared to the one-shot CTN. On the hip dataset, we found that on 178 out of 200 testing images, the IoU improved after fine-tuning using 25% corrections (average IoU from 97.27% to 98.12%). On the other 22 testing images, the IoU degraded (average IoU from 97.48% to 97.06%). Overall, the average IoU of all 200 images increased

[7]https://github.com/milesial/Pytorch-UNet
[8]https://github.com/jfzhang95/pytorch-deeplab-xception
[9]https://github.com/HRNet/HRNet-Semantic-Segmentation

TABLE II

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

from 97.3% to 98.0%. The results show that in the majority of the cases (89%), fine-tuned CTN improves the segmentation performance by a noticeable IoU gap (0.85%). While in some cases (11%), the performance degrades, the degradation is on average smaller (IoU gap 0.42%) than the improvement.

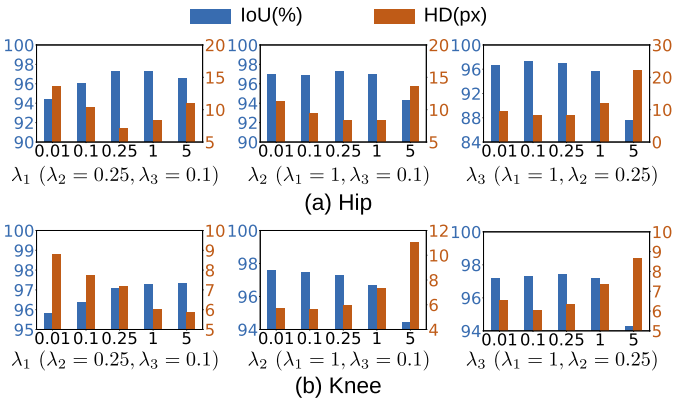

Fig. 8. **Using different loss weights to train CTN on the hip and the knee datasets.** Based on the original setting $\lambda_1 = 1$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.1$, we change one of them each time and fix the other two.

D. 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 II, 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 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.

E. Ablation Study on the Proposed 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 III. 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 leads to slightly lower performance in this particular scenario, where the IoU decreases from 97.49% to 97.32% and the HD increases from 5.87 to 6.01. We note that the changes (0.17% for IoU and 0.14 for HD) are below the standard deviations of CTN on knee (0.67% for IoU and 1.42 for HD). Despite the exception on the knee dataset, 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.

To further understand the impact of the losses, we analyze CTN's sensitivity to the three loss weights, λ_1, λ_2 and λ_3. Specifically, on both the knee and hip datasets, three experiments are conducted to evaluate the impact of varying the three loss weights individually while fixing the other two. 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 varying loss weights are reported in Fig. 8. On the hip dataset, very small and large loss weights in general lead to degraded performance. On the knee dataset, larger λ_1 and smaller λ_2 achieve better performances. We posit that due to the distinct appearance, knees can be reliably segmented using the visual patterns (measured by L_{perc}) only, and strong shape regularization (measured by L_{bend}) degrades the performance by imposing unnecessary shape constraints.

We also compare the performances of CTN using L1 and L2 distances in the contour perceptual loss (Section II-C.1) on the hip dataset. The results show that using L2 distance results in degraded performance compared to using L1 distance, reporting an IoU of 96.82% (compared to 97.29%) and a HD of 12.41 (compared to 8.27). We note that the degradation in HD is more obvious than IoU, hypothetically owing to the forgiving nature of L2 distance to small errors.

F. Analysis on Failure Cases

In Fig. 9, we show and examine a few typical failure cases to analyze the performance characteristics of CTN. Fig. 9(a) is a knee with severe osteoporosis, which significantly reduces the joint space and makes the tibia and femur bones overlap. CTN fails to segment the overlapped region properly. However, DeepLab also produces wrong segmentation on this challenging case. In Fig 9(b), the acute change of lung shape differs from the mean shape, and CTN mis-segments this part and produces a result closer to mean shape. Although DeepLab also mis-segments the same part, its result is closer to the ground truth than CTN. Fig. 9(c) is a hip with severe osteoporosis, similar to Fig. 9(a), where the joint space is reduced, making the bone boundary less recognizable. On this case, CTN produces wrong segmentation on the bone boundary affected by the osteoporosis, while DeepLab produces

TABLE III
ABLATION STUDY. THE THREE LOSSES OF CTN ARE REMOVED INDIVIDUALLY TO EVALUATED THEIR IMPACTS ON THE SEGMENTATION PERFORMANCE

L_{perc}	L_{bend}	L_{edge}	Knee		Lung		Phalanx		Hip		**Mean**	
			IoU (%)	HD (px)	IoU (%)	HD (px)	IoU (%)	HD (px)	IoU (%)	HD (px)	IoU (%)	HD (px)
	✓	✓	94.62	8.28	87.45	26.51	94.01	15.80	92.90	16.58	92.24	16.79
✓		✓	**97.49**	**5.87**	84.93	36.74	94.24	26.13	94.53	13.91	92.80	20.66
✓	✓		94.43	11.90	93.00	16.22	96.45	9.84	96.61	9.92	95.12	11.97
✓	✓	✓	97.32	6.01	**94.74**	**12.17**	**96.96**	**8.19**	**97.29**	**8.27**	**96.58**	**8.66**

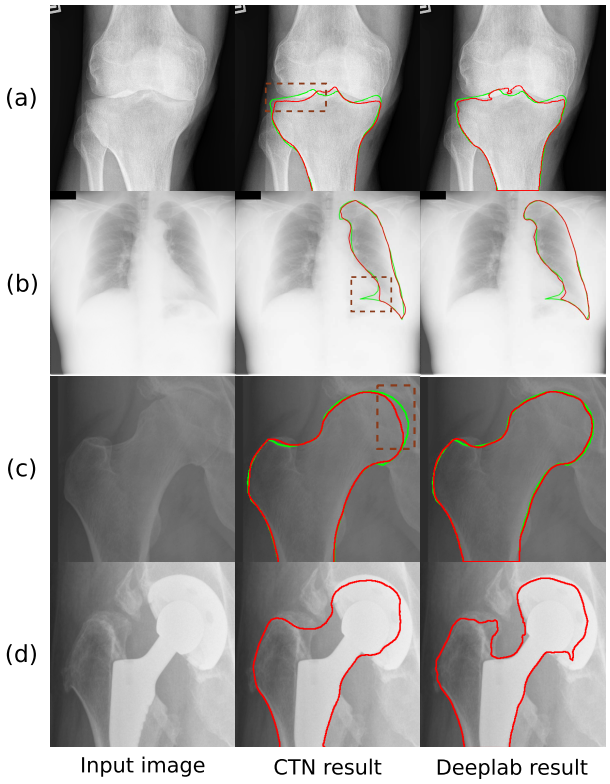

Fig. 9. **Typical failure cases**. (a) Intersected boundaries. (b) Acute shape change. (c) Blurry boundary. (d) Metal implant (this case is not in the test set). The green curves are the ground truth contours, the red curves are the predicted contours, and the dashed boxes show the wrong part of CTN predictions.

satisfactory results. Fig. 9(d) shows an extreme case of hip X-ray with total hip replacement (this case is not in our test since there is no ground truth segmentation). While there is no standard for correct segmentation on this particular case, we observe that the segmentation produced by CTN tends to follow the mean shape of a normal hip. In comparison, DeepLab tends to produce segmentation results following the edges in the image.

G. Analysis on the Behaviors of CTN

1) Robustness to Detected ROIs: As a prerequisite of CTN, ROI detection is an important step to help reduce the contour searching space. Therefore, the performance of CTN also depends on the accuracy of ROI detection. To evaluate the influence of ROI detection, we conduct an experiment to compare the performances of CTN when using automatic ROIs and manually perturbed ROIs. Specifically, three offsets are imposed on the bounding boxes of hip ROIs to perturb their locations, Δx, Δy and $\Delta\theta$, denoting the translation on x-axis and y-axis, and the rotation around the ROI center, respectively. We randomly generate $-5px \leq \Delta x, \Delta y \leq 5px$ and $-5° \leq \Delta\theta \leq 5°$ to simulate ROIs produced with certain landmark detection errors. Note that in our experiment, this perturbation is added on the automatically detected ROIs, which already contains errors from the ROI detector. We test the model trained without ROI perturbation on perturbed ROIs, to examine CTN's robustness to ROI localization errors unseen in the training data. Table IV summarizes the testing results. With all three perturbations, the IoU dropped by 0.78% and the HD increased by 1.88 px, indicating that the performance of CTN can be affected by ROI localization errors. However, the performance degradation is relatively small, *i.e.*, comparable to the standard deviations (IoU 0.72% and HD 3.06 px), indicating that CTN holds a good robustness against the perturbations. We evaluated DeepLab under the same ROI perturbation settings, and observed similar performance degradation, *i.e.* IoU by 0.71% and HD by 1.2 px.

TABLE IV
TESTING CTN ON PERTURBED HIP ROIS. WE MANUALLY MODIFY THE LOCATION OF ROIS WITH OFFSETS ON X-AXIS, Y-AXIS, AND ROTATION

Δx(px)	Δy(px)	$\Delta\theta$(°)	IoU (%)	HD (px)
0	0	0	97.29	8.27
5	0	0	96.90	8.76
0	5	0	96.91	8.88
0	0	5	96.62	9.79
5	5	5	96.51	10.15

2) Impact of the Selection of the Exemplar Image: Since the exemplar image is the main source of supervision signal, the selection of the exemplar image may be critical to the generalizability of CTN. In Section III-A.1, we propose to select the image with the minimum average VGG distance to other training images as the exemplar. In this section, we conduct experiments on the hip dataset to evaluate CTN's performance using four randomly selected exemplars and compare them with the automatically selected exemplar. The five exemplar images are shown in Fig. 10, and their resulting performances are reported in Table V. We can observe that the automatically selected exemplar Fig. 10(a) based on VGG distance results in better performance than the randomly selected ones.

Fig. 10. **Five exemplar images from the hip training set.** (a) The exemplar image automatically selected using the proposed method. (b)-(e) Four randomly selected exemplars.

TABLE V
USING FIVE DIFFERENT EXEMPLARS TO TRAIN CTN ON THE HIP DATASET. THE AVERAGE VGG DISTANCE IS CALCULATED BY AVERAGING THE L2 DISTANCES FROM THE EXEMPLAR IMAGE TO ALL OTHER TRAINING IMAGES IN THE VGG FEATURE SPACE

Exemplar image	Avg. VGG distance	IoU(%)	HD(px)
Fig. 10(a)	2.98×10^5	**97.29**	**8.27**
Fig. 10(b)	3.51×10^5	96.88	9.39
Fig. 10(c)	3.32×10^5	96.08	11.92
Fig. 10(d)	4.83×10^5	96.77	9.36
Fig. 10(e)	3.20×10^5	94.57	15.37

TABLE VI
USING DIFFERENT NUMBERS OF GCN BLOCKS TO TRAIN CTN ON THE HIP DATASET

Num. of GCN blocks	1	3	5	7	9
IoU (%)	97.10	97.11	97.29	96.91	96.67
HD (px)	8.51	8.44	8.27	8.88	9.29

TABLE VII
MODEL EFFICIENCY OF LEARNING-BASED METHODS. WE COMPARE THE NUMBER OF PARAMETERS, THE NUMBER OF FLOAT-POINT OPERATIONS (FLOPS), AND THE INFERENCE FPS OF ALL LEARNING-BASED METHODS

Methods	# of Params	FLOPs	FPS
CANet	19.01M	27.42G	20.77
Brainstorm	1.78M	7.55G	15.44
UNet	13.40M	30.65G	28.57
DeepLab	59.34M	22.55G	18.18
HRNet	9.64M	4.67G	9.86
CTN	42.26M	32.99G	15.39

We also observe that the performance of CTN is not always correlated with the VGG distance. For example, the exemplar Fig. 10(d) with larger distance produces a better CTN model than Fig. 10(e) with smaller distance. We note that even with randomly selected exemplars, CTN consistently outperforms previous one-shot segmentation methods, *e.g.* Brainstorm.

3) Impact of the Number of GCN Block Iterations: In this section, we evaluate the impact of the number of GCN block iterations by training and testing the CTN with 1, 3, 5, 7 and 9 GCN block iterations on the hip dataset. The results of this analysis are summarized in Table VI. It shows that as the number of GCN blocks increases from 1 to 5, the performance improves from IoU 97.10% to 97.29% and HD 8.51 px to 8.27 px, respectively. It demonstrates that by stacking multiple GCN blocks, the later GCN block can further correct the segmentation errors produced by the earlier GCN blocks, which is beneficial to the final performance. However, the performance starts to slightly degrade when the number of GCN blocks increases over 5. We posit that the increased number of layers in the CTN caused by the additional GCN blocks make the network more difficult to train, which contributes to the performance degradation.

4) Computational Efficiency: We analyze the computational efficiency of CTN and compare it with other learning-based segmentation methods. Table VII summarizes the number of parameters, the number of float-point operations (FLOPs) and frames per second (FPS). All evaluations are conducted on the hip dataset with a Nvidia GTX 1080Ti GPU. While the computational efficiency varies significantly among the evaluated methods (*e.g.* number of parameters from 1.78M to 59.34M, FLOPs from 4.67G to 32.99G, FPS from 9.86 to 28.57), all methods report sufficient speed (above 9 FPS) for off-line image analysis tasks. A few methods, including CTN, measure above 15 FPS, which is the common fluoroscopic imaging frame rate, showing potential applicability on real-time image analysis tasks.

IV. DISCUSSION AND CONCLUSION

In this paper, 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 III-D and III-C 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.7, 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. Future research could investigate the extension of CTN to few-shot learning scenario via group-wise loss calculation. In addition, the extension of CTN to 3D segmentation might prove an important area for future research. Unlike FCN-based segmentation methods, which can be directly applied on 3D tasks by using 3D convolutions, extending the 2D contour-based formulation of CTN to a 3D surface-based formulation requires is non-trivial and warrants further investigation.

REFERENCES

[1] C. Baillard, P. Hellier, and C. Barillot, "Segmentation of brain 3D MR images using level sets and dense registration," *Med. Image Anal.*, vol. 5, no. 3, pp. 185–194, Sep. 2001.

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *Proc. Image Understand. Workshop*, Apr. 1977, pp. 21–27.

[3] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[4] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[5] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.

[6] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[7] L.-C. Chen, K. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 801–818.

[8] R. Chen, Y. Ma, N. Chen, D. Lee, and W. Wang, "Cephalometric landmark detection by attentive feature pyramid fusion and regression-voting," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 873–881.

[9] C. Ciofolo and C. Barillot, "Atlas-based segmentation of 3D cerebral structures with competitive level sets and fuzzy control," *Med. Image Anal.*, vol. 13, no. 3, pp. 456–470, Jun. 2009.

[10] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 195–215, Apr. 2007.

[11] A. V. Dalca, J. Guttag, and M. R. Sabuncu, "Anatomical priors in convolutional networks for unsupervised biomedical segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9290–9299.

[12] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. Ben Ayed, "HyperDense-Net: A hyper-densely connected CNN for multi-modal image segmentation," *IEEE Trans. Med. Imag.*, vol. 38, no. 5, pp. 1116–1126, May 2019.

[13] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, 2018, p. 6.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[15] 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," *Med. Image Anal.*, vol. 10, no. 1, pp. 19–40, Feb. 2006.

[16] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[17] S. Gur, L. Wolf, L. Golgher, and P. Blinder, "Unsupervised microvascular image segmentation using an active contours mimicking neural network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10722–10731.

[18] 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," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*, 2017, pp. 621–629.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[20] 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 Trans. Biomed. Eng.*, vol. 63, no. 10, pp. 2177–2186, Oct. 2016.

[21] J. E. Iglesias and M. R. Sabuncu, "Multi-atlas segmentation of biomedical images: A survey," *Med. Image Anal.*, vol. 24, no. 1, pp. 205–219, Aug. 2015.

[22] J. Johnson, A. Alahi, and F.-F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[25] M. C. H. Lee, K. Petersen, N. Pawlowski, B. Glocker, and M. Schaap, "TeTrIS: Template transformer networks for image segmentation with shape priors," *IEEE Trans. Med. Imag.*, vol. 38, no. 11, pp. 2596–2606, Nov. 2019.

[26] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-GCN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5257–5266.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[28] P. Marquez-Neila, L. Baumela, and L. Alvarez, "A morphological approach to curvature-based evolution of curves and surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 2–17, Jan. 2014.

[29] C. Michaelis, I. Ustyuzhaninov, M. Bethge, and A. S. Ecker, "One-shot instance segmentation," 2018, *arXiv:1811.11507*. [Online]. Available: http://arxiv.org/abs/1811.11507

[30] F. P. M. Oliveira and J. M. R. S. Tavares, "Medical image registration: A review," *Comput. Methods Biomech. Biomed. Eng.*, vol. 17, no. 2, pp. 73–93, 2014.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*, 2015, pp. 234–241.

[32] C. Rupprecht, E. Huaroc, M. Baust, and N. Navab, "Deep active contours," 2016, *arXiv:1607.05074*. [Online]. Available: http://arxiv.org/abs/1607.05074

[33] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," 2017, *arXiv:1709.03410*. [Online]. Available: http://arxiv.org/abs/1709.03410

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[35] 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," *Med. Image Anal.*, vol. 63, Jul. 2020, Art. no. 101693.

[36] Y.-B. Tang, S. Oh, Y.-X. Tang, J. Xiao, and R. M. Summers, "Ct-realistic data augmentation using generative adversarial network for robust lymph node segmentation," *Proc. SPIE*, vol. 10950, Mar. 2019, Art. no. 109503V.

[37] L. Zhang *et al.*, "Learning deep structured active contours end-to-end," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8877–8885.

[38] H. R. Roth *et al.*, "Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation," *Med. Image Anal.*, vol. 45, pp. 94–107, Apr. 2018.

[39] 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," *Amer. J. Roentgenology*, vol. 174, no. 1, pp. 71–74, Jan. 2000.

[40] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 1, 2020, doi: 10.1109/TPAMI.2020.2983686.

[41] O. Oktay *et al.*, "Anatomically constrained neural networks (ACNNs): Application to cardiac image enhancement and segmentation," *IEEE Trans. Med. Imag.*, vol. 37, no. 2, pp. 384–395, Feb. 2018.

[42] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[43] S. Wang *et al.*, "LT-Net: Label transfer by learning reversible voxel-wise correspondence for one-shot medical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9162–9171.

[44] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[45] W. Li *et al.*, "Structured landmark detection via topology-adapting deep graph learning," 2020, *arXiv:2004.08190*. [Online]. Available: http://arxiv.org/abs/2004.08190

[46] X. Chen, B. M. Williams, S. R. Vallabhaneni, G. Czanner, R. Williams, and Y. Zheng, "Learning active contour models for medical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11632–11640.

[47] Y. Lu *et al.*, "Learning to segment anatomical structures accurately from one exemplar," 2020, *arXiv:2007.03052*. [Online]. Available: http://arxiv.org/abs/2007.03052

[48] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 52–67.

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

[50] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2019, *arXiv:1904.05046*. [Online]. Available: http://arxiv.org/abs/1904.05046

[51] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2129–2138.

[52] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5217–5226, 2019.

[53] A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca, "Data augmentation using learned transformations for one-shot medical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8543–8553.