Topology-Preserving Deep Image Segmentation

*Xiaoling Hu[1], Li Fuxin[2], Dimitris Samaras[1] and Chao Chen[1]

[1]Stony Brook University
[2]Oregon State University

Abstract

Segmentation algorithms are prone to topological errors on fine-scale structures, e.g., broken connections. We propose a novel method that learns to segment with correct topology. In particular, we design a continuous-valued loss function that enforces a segmentation to have the same topology as the ground truth, i.e., having the same Betti number. The proposed topology-preserving loss function is differentiable and we incorporate it into end-to-end training of a deep neural network. Our method achieves much better performance on the Betti number error, which directly accounts for the topological correctness. It also performs superiorly on other topology-relevant metrics, e.g., the Adjusted Rand Index and the Variation of Information. We illustrate the effectiveness of the proposed method on a broad spectrum of natural and biomedical datasets.

1 Introduction

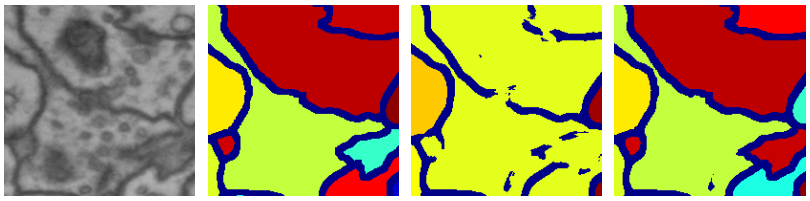
Image segmentation, i.e., assigning labels to all pixels of an input image, is crucial in many computer vision tasks. State-of-the-art deep segmentation methods [25, 20, 8, 9, 10] learn high quality feature representations through an end-to-end trained deep network and achieve satisfactory per-pixel accuracy. However, these segmentation algorithms are still prone to make errors on fine-scale structures, such as small object instances, instances with multiple connected components, and thin connections. These fine-scale structures may be crucial in analyzing the *functionality* of the objects. For example, accurate extraction of thin parts such as ropes and handles is crucial in planning robot actions, e.g., dragging or grasping. In biomedical images, correct delineation of thin objects such as neuron membranes and vessels is crucial in providing accurate morphological and structural quantification of the underlying system. A broken connection or a missing component may only induce marginal per-pixel error, but can cause catastrophic functional mistakes. See Fig. 1 for an example.

We propose a novel deep segmentation method that *learns to segment with correct topology*. In particular, we propose a *topological loss* that enforces the segmentation results to have the same topology as the ground truth, i.e., having the same *Betti number* (number of connected components and handles). A neural network trained with such loss will achieve high topological fidelity without sacrificing per-pixel accuracy. The main challenge in designing such loss is that topological information, namely, Betti numbers, are discrete values. We need a continuous-valued measurement of the topological similarity between a prediction and the ground truth; and such measurement needs to be differentiable in order to backpropagate through the network.

To this end, we propose to use theory from computational topology [13], which summarizes the topological information from a continuous-valued function (in our case, the likelihood function f is predicted by a neural network). Instead of acquiring the segmentation by thresholding f at 0.5 and inspecting its topology, *persistent homology* [13, 14, 44] captures topological information carried by f over all possible thresholds. This provides a unified, differentiable approach of measuring the

*Correspondence to: Xiaoling Hu <xiaolhu@cs.stonybrook.edu>.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



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

Figure 1: Illustration of the importance of topological correctness in a neuron image segmentation task. The goal of this task is to segment membranes which partition the image into regions corresponding to neurons. **(a)** an input neuron image. **(b)** ground truth segmentation of the membranes (dark blue) and the result neuron regions. **(c)** result of a baseline method without topological guarantee [16]. Small pixel-wise errors lead to broken membranes, resulting in merging of many neurons into one. **(d)** Our method produces the correct topology and the correct partitioning of neurons.

topological similarity between f and the ground truth, called the *topological loss*. We derive the gradient of the loss so that the network predicting f can be optimized accordingly. We focus on 0- and 1-dimensional topology (components and connections) on 2-dimensional images.

Our method is the first end-to-end deep segmentation network with guaranteed topological correctness. We show that when the topological loss is decreased to zero, the segmentation is guaranteed to be topologically correct, i.e., have identical topology as the ground truth. Our method is empirically validated by comparing with state-of-the-arts on natural and biomedical datasets with fine-scale structures. It achieves superior performance on metrics that encourage structural accuracy. In particular, our method significantly outperforms others on the Betti number error which exactly measures the topological accuracy. Fig. 1 shows a qualitative result.

Our method shows how topological computation and deep learning can be mutually beneficial. While our method empowers deep nets with advanced topological constraints, it is also a powerful approach on topological analysis; the observed function is now learned with a highly nonlinear deep network. This enables topology to be estimated based on a semantically informed and denoised observation.

Related work. The closest method to ours is by Mosinska *et al.* [27], which also proposes a topology-aware loss. Instead of actually computing and comparing the topology, their approach uses the response of selected filters from a pretrained VGG19 network to construct the loss. These filters prefer elongated shapes and thus alleviate the broken connection issue. But this method is hard to generalize to more complex settings with connections of arbitrary shapes. Furthermore, even if this method achieves zero loss, its segmentation is not guaranteed to be topologically correct.

Different ideas have been proposed to capture fine details of objects, mostly revolving around deconvolution and upsampling [25, 8, 9, 10, 29, 34]. However these methods focus on the prediction accuracy of individual pixels and are intrinsically topology-agnostic. Topological constraints, e.g., connectivity and loop-freeness, have been incorporated into variational [19, 24, 38, 35, 42, 18] and MRF/CRF-based segmentation methods [40, 30, 43, 6, 2, 37, 31, 15]. However, these methods focus on enforcing topological constraints in the inference stage, while the trained model is agnostic of the topological prior. In neuron image segmentation, some methods [17, 39] directly find an optimal partition of the image into neurons, and thus avoid segmenting membranes. These methods cannot be generalized to other structures, e.g., vessels, cracks and roads.

For completeness, we also refer to other existing works on topological features and their kernels [1, 33, 23, 5]. In graphics, topological similarity was used to simplify and align shapes [32]. Chen et al. [7] proposed a topological regularizer to simplify the decision boundary of a classifier. As for deep neural networks, Hofer *et al.* [21] proposed a CNN-based topological classifier. This method directly extracts topological information from an input image/shape/graph as input for CNN, hence cannot generate segmentations that preserve topological priors learned from the training set. To the best of our knowledge, no existing work uses topological information as a loss for training a deep neural network in an end-to-end manner.

2 Method

Our method achieves both per-pixel accuracy and topological correctness by training a deep neural network with a new topological loss, $L_{topo}(f, g)$. Here f is the likelihood map predicted by the

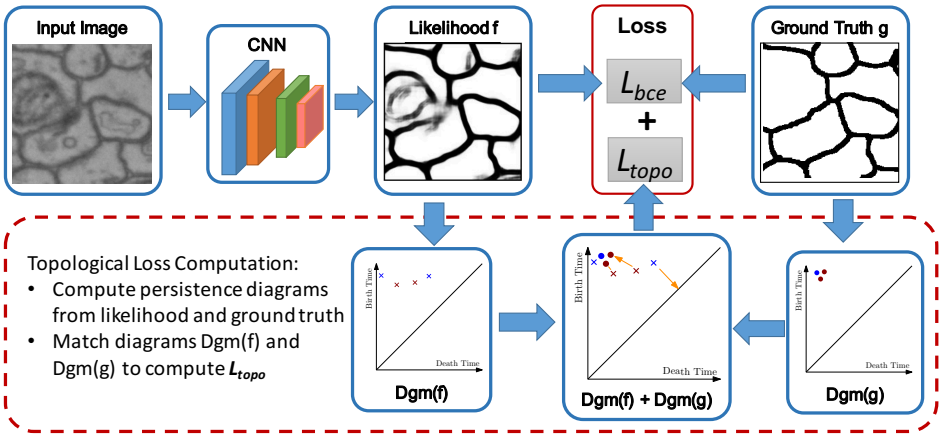


Figure 2: An overview of our method.

network and g is the ground truth. The loss function on each training image is a weighted sum of the per-pixel cross-entropy loss, L_{bce}, and the topological loss:

$$L(f, g) = L_{bce}(f, g) + \lambda L_{topo}(f, g), \tag{2.1}$$

in which λ controls the weight of the topological loss. We assume a binary segmentation task. Thus, there is one single likelihood function f, whose value ranges between 0 and 1.

In Sec. 2.1, we introduce the mathematical foundation of topology and how to measure topology of a likelihood map robustly using persistent homology. In Sec. 2.2, we formalize the topological loss as the difference between persistent homology of f and g. We derive the gradient of the loss and prove its correctness. In Sec. 2.3 we explain how to incorporate the loss into the training of a neural network. Although we fix one architecture in experiments, our method is general and can use any neural network that provides pixel-wise prediction. Fig. 2 illustrates the overview of our method.

2.1 Topology and Persistent Homology

Given a continuous image domain, $\Omega \subseteq \mathbb{R}^2$ (e.g., a 2D rectangle), we study a likelihood map $f(x) : \Omega \to \mathbb{R}$, which is predicted by a deep neural network (Fig. 3(c)).[2] Note that in practice, we only have samples of f at all pixels. In such case, we extend f to the whole image domain Ω by linear interpolation. Therefore, f is piecewise-linear and is controlled by values at all pixels. A segmentation, $X \subseteq \Omega$ (Fig. 3(a)), is calculated by thresholding f at a given value α (often set to 0.5).

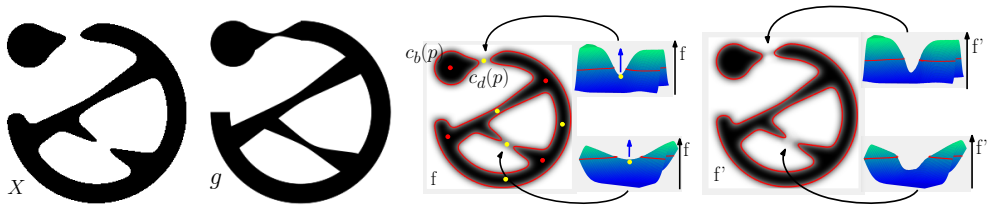
Given X, its d-dimension topological structure, called a *homology class* [13, 28], is an equivalence class of d-manifolds which can be deformed into each other within X.[3] In particular, 0-dim and 1-dim structures are connected components and handles, respectively. For example, in Fig. 3(a), the segmentation X has two connected components and one handle. Meanwhile, the ground truth (Fig. 3(b)) has one connected component and two handles. Given X, we can compute the number of topological structures, called the *Betti number*, and compare it with the topology of the ground truth.

However, simply comparing Betti numbers of X and g will result in a discrete-valued topological error function. To incorporate topological prior into deep neural networks, we need a continuous-valued function that can reveal subtle difference between similar structures. Fig. 3(c) and 3(d) show two likelihood maps f and f' with identical segmentations, both with incorrect topology comparing with the ground truth g (Fig. 3(b)). However, f is more preferable as we need much less effort to change it so that the thresholded segmentation X has a correct topology. In particular, look closely to Fig. 3(c) and 3(d) near the broken handles and view the landscape of the function. To restore the broken handle in Fig. 3(d), we need to spend more effort to fill a much deeper gap than Fig. 3(c). The same situation happens near the missing bridge between the two connected components.

To capture such subtle structural difference between different likelihood maps, we need a holistic view. In particular, we use the theory of *persistent homology* [14, 13]. Instead of choosing a fixed

[2] f depends on the network parameter ω, which will be optimized during training. For convenience, we only use x as the argument of f.

[3] To be exact, a homology class is an equivalent class of cycles whose difference is the boundary of a $(d+1)$-dimensional patch.



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

Figure 3: Illustration of topology and topology of a likelihood. For visualization purposes, the higher the function values are, the darker the area is. **(a)** an example segmentation X with two connected components and one handle. **(b)** The ground truth with one connected component and two handles. It can also be viewed as a binary valued function g. **(c)** a likelihood map f whose segmentation (bounded by the red curve) is X. The landscape views near the broken bridge and handle are drawn. Critical points are highlighted in the segmentation. **(d)** another likelihood map f' with the same segmentation as f. But the landscape views reveal that f' is worse than f due to deeper gaps.

threshold, persistent homology theory captures all possible topological structures from all thresholds, and summarize all these information in a concise format, called a *persistence diagram*.

Fig. 3 shows that only considering one threshold $\alpha = 0.5$ is insufficient. We consider thresholding the likelihood function with all possible thresholds. The thresholded results, $f^\alpha := \{x \in \Omega | f(x) \geq \alpha\}$ at different α's, constitute a filtration, i.e., a monotonically growing sequence induced by decreasing the threshold $\alpha : \varnothing \subseteq f^{\alpha_1} \subseteq f^{\alpha_2} \subseteq ... \subseteq f^{\alpha_n} = \Omega$, where $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n$. As α decreases, the topology of f^α changes. Some new topological structures are born while existing ones are killed. When $\alpha < \alpha_n$, only one connected component survives and never gets killed. See Fig. 4(a) and 4(d) for filtrations induced by the ground truth g (as a binary-valued function) and the likelihood f.

For a continuous-valued function f, its *persistence diagram*, $\mathrm{Dgm}(f)$, contains a finite number of dots in 2-dimensional plane, called *persistent dots*. Each persistent dot $p \in \mathrm{Dgm}(f)$ corresponds to a topological structure born and dies in the filtration. Denote by $\mathrm{birth}(p)$ and $\mathrm{death}(p)$ the birth and death time/threshold of the structure. For the connected component born at global minimum and never dies, we say it dies at $\max_x f(x) = 1$. The coordinates of the dot p in the diagram are $(1 - \mathrm{birth}(p), 1 - \mathrm{death}(p))$.[4] Fig. 4(b) and 4(e) show the diagrams of g and f, respectively. Instead of comparing discrete Betti numbers, we can use the information from persistence diagrams to compare a likelihood f with the ground truth g in terms of topology.

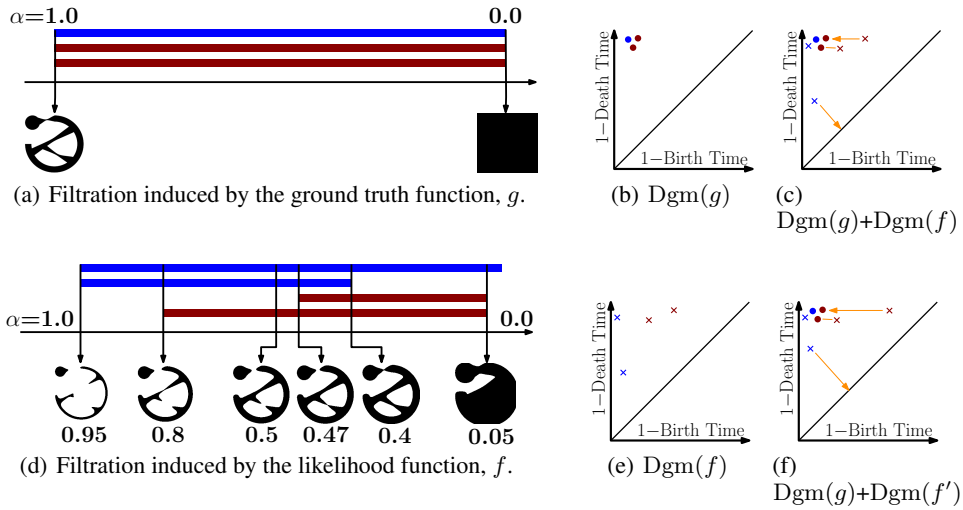
To compute the persistence diagram $\mathrm{Dgm}(f)$, we use the classic algorithm [13, 14]: we first discretize an image patch into vertices (pixels), edges and squares. Note we adopt a cubical complex discretization, which is more suitable for image analysis [41]. The adjacency relationship between these discretized elements and their likelihood function values are encoded in a boundary matrix, whose rows and columns correspond to vertices/edges/squares. The matrix is reduced using a modified Gaussian elimination algorithm. The pivoting entries of the reduced matrix correspond to all the dots in $\mathrm{Dgm}(f)$. This algorithm is cubic to the matrix dimension, which is linear to the image size.

2.2 Topological Loss and its Gradient

We are now ready to formalize the topological loss, which measures the topological similarity between the likelihood f and the ground truth g. We abuse the notation and also view g as a binary valued function. We use the dots in the persistence diagram of f as they capture all possible topological structures f potentially has. We slightly modify the *Wasserstein distance* for persistence diagrams [12]. For persistence diagrams $\mathrm{Dgm}(f)$ and $\mathrm{Dgm}(g)$, we find a best one-to-one correspondence between the two sets of dots, and measure the total squared distance between them.[5] An unmatched dot will be matched to the diagonal line. Fig. 4(c) shows the optimal matching of the diagrams of g and f. Fig. 4(f) shows the optimal matching of $\mathrm{Dgm}(g)$ and $\mathrm{Dgm}(f')$. The latter is clearly more expensive.

[4] Unlike traditional setting, we use $1 - \mathrm{birth}$ and $1 - \mathrm{death}$ as the x and y axes, because we are using an upperstar filtration, i.e., using the superlevel set, and decreasing α value.

[5] To be exact, the matching needs to be done on separate dimensions. Dots of 0-dim structures (blue markers in Fig. 4(b) and 4(e)) should be matched to the diagram of 0-dim structures. Dots of 1-dim structures (red markers in Fig. 4(b) and 4(e)) should be matched to the diagram of 1-dim structures.



(a) Filtration induced by the ground truth function, g. (b) $\mathrm{Dgm}(g)$ (c) $\mathrm{Dgm}(g)$+$\mathrm{Dgm}(f)$

(d) Filtration induced by the likelihood function, f. (e) $\mathrm{Dgm}(f)$ (f) $\mathrm{Dgm}(g)$+$\mathrm{Dgm}(f')$

Figure 4: An illustration of persistent homology. **Left** the filtrations on the ground truth function g and the likelihood function f. The bars of blue and burgundy colors are connected components and handles respectively. **(a)** For g, all structures are born at $\alpha = 1.0$ and die at $\alpha = 0$. **(d)** For f, from left to right, birth of two components, birth of the longer handle, segmentation at $\alpha = 0.5$, birth of the shorter handle, death of the extra component, death of both handles. **(b)** and **(e)** the persistence diagrams of g and f. **(c)** the overlay of the two diagrams. Orange arrows denote the matching between the persistent dots. The extra component (a blue cross) from the likelihood is matched to the diagonal line and will be removed if we move $\mathrm{Dgm}(f)$ to $\mathrm{Dgm}(g)$. **(f)** the overlay of the diagrams of g and the worse likelihood $\mathrm{Dgm}(f')$. The matching is obviously more expensive.

The matching algorithm is as follows. A total of k (=Betti number) dots from ground truth ($\mathrm{Dgm}(g)$) are at the upper-left corner $p_{ul} = (0, 1)$, with $\mathrm{birth}(p_{ul}) = 1$ and $\mathrm{death}(p_{ul}) = 0$ (Fig. 4(b)). In $\mathrm{Dgm}(f)$, we find the k dots closest to the corner p_{ul} and match them to the ground truth dots. The remaining dots in $\mathrm{Dgm}(f)$ are matched to the diagonal line. The algorithm computes and sorts the squared distances from all dots in $\mathrm{Dgm}(f)$ to p_{ul}. The complexity is $O(n \log n)$, n = the number of dots in $\mathrm{Dgm}(f)$. In general, the state-of-the-art matches two arbitrary diagrams in $O(n^{3/2})$ time [22].

Let Γ be the set of all possible bijections between $\mathrm{Dgm}(f)$ and $\mathrm{Dgm}(g)$. The loss $L_{topo}(f, g)$ is:

$$\min_{\gamma \in \Gamma} \sum_{p \in \mathrm{Dgm}(f)} ||p - \gamma(p)||^2 = \sum_{p \in \mathrm{Dgm}(f)} [\mathrm{birth}(p) - \mathrm{birth}(\gamma^*(p))]^2 + [\mathrm{death}(p) - \mathrm{death}(\gamma^*(p))]^2 \tag{2.2}$$

where γ^* is the optimal matching between two different point sets.

Intuitively, this loss measures the minimal amount of necessary effort to modify the diagram of $\mathrm{Dgm}(f)$ to $\mathrm{Dgm}(g)$ by moving all dots toward their matches. Note there are more dots in $\mathrm{Dgm}(f)$ (Fig. 4(c)) than in $\mathrm{Dgm}(g)$ (Fig. 4(b)); there will usually be some noise in predicted likelihood map. If a dot p cannot be matched, we match it to its projection on the diagonal line, $\{(1-b, 1-d) | b = d\}$. This means we consider it as noise that should be removed. The dots matched to the diagonal line correspond to small noisy components or noisy loops. These dots will be pushed to the diagonal. And their corresponding components/loops will be removed or merged with others.

In this example, the extra connected component (a blue cross) in $\mathrm{Dgm}(f)$ will be removed. For comparison, we also show in Fig. 4(f) the matching between diagrams of the worse likelihood f' and g. The cost of the matching is obviously higher, i.e., $L_{topo}(f', g) > L_{topo}(f, g)$. As a theoretical reassurance, it has been proven that this metric for diagrams is stable, and the loss function $L_{topo}(f, g)$ is Lipschitz with regard to the likelihood function f [11].

The following theorem guarantees that the topological loss, when minimized to zero, enforces the constraint that the segmentation has the same topology and the ground truth.

Theorem 1 (Topological Correctness)**.** *When the loss function $L_{topo}(f, g)$ is zero, the segmentation by thresholding f at 0.5 has the same Betti number as g.*

Proof. Assume $L_{topo}(f, g)$ is zero. By Eq. (2.2), $\mathrm{Dgm}(f)$ and $\mathrm{Dgm}(g)$ are matched perfectly, i.e., $p = \gamma^*(p), \forall p \in \mathrm{Dgm}(f)$. The two diagrams are identical and have the same number of dots.

Since g is a binary-valued function, as we decrease the threshold α continuously, all topological structures are created at $\alpha = 1$. The number of topological structures (Betti number) of g^α for any $0 < \alpha < 1$ is the same as the number of dots in $\mathrm{Dgm}(g)$. Note that for any $\alpha \in (0, 1)$, g^α is the ground truth segmentation. Therefore, the Betti number of the ground truth is the number of dots in $\mathrm{Dgm}(g)$. Similarly, for any $\alpha \in (0, 1)$, the Betti number of f^α equals to the number of dots in $\mathrm{Dgm}(f)$. Since the two diagrams $\mathrm{Dgm}(f)$ and $\mathrm{Dgm}(g)$ are identical, the Betti number of the segmentation $f^{0.5}$ is the same as the ground truth segmentation.[6] □

Topological gradient. The loss function (Eq. (2.2)) depends on crucial thresholds at which topological changes happen, e.g., birth and death times of different dots in the diagram. These crucial thresholds are uniquely determined by the locations at which the topological changes happen. When the underlying function f is differentiable, these crucial locations are exactly *critical points*, i.e., points with zero gradients. In the training context, our likelihood function f is a piecewise-linear function controlled by the neural network predictions at pixels. For such f, a critical point is always a pixel, since topological changes always happen at pixels. Denote by ω the neural network parameters. For each dot $p \in \mathrm{Dgm}(f)$, we denote by $c_b(p)$ and $c_d(p)$ the birth and death critical points of the corresponding topological structure (See Fig. 3(c) for examples).

Formally, we can show that the gradient of the topological loss $\nabla_\omega L_{topo}(f, g)$ is:

$$\sum_{p \in \mathrm{Dgm}(f)} 2[f(c_b(p)) - \mathrm{birth}(\gamma^*(p))] \frac{\partial f(c_b(p))}{\partial \omega} + 2[f(c_d(p)) - \mathrm{death}(\gamma^*(p))] \frac{\partial f(c_d(p))}{\partial \omega} \quad (2.3)$$

To see this, within a sufficiently small neighborhood of f, any other piecewise linear function will have the same super level set filtration as f. The critical points of each persistent dot in $\mathrm{Dgm}(f)$ remains constant within such small neighborhood. So does the optimal mapping γ^*. Therefore, the gradient can be straightforwardly computed based on the chain rule, as Eq. (2.3). When function values at different vertices are the same, or when the matching is ambiguous, the gradient does not exist. However, these cases constitute a measure zero subspace in the space of likelihood functions. In summary, $L_{topo}(f, g)$ is a piecewise differentiable loss function over the space of all possible likelihood functions f.

Intuition. During training, we take the negative gradient direction, i.e.,$-\nabla_\omega L_{topo}(f, g)$. For each topological structure the gradient descent step is pushing the corresponding dot $p \in \mathrm{Dgm}(f)$ toward its match $\gamma^*(p) \in \mathrm{Dgm}(g)$. These coordinates are the function values of the critical points $c_b(p)$ and $c_d(p)$. They are both moved closer to the matched persistent dot in $\mathrm{Dgm}(g)$. We also show the negative gradient force in the landscape view of function f (blue arrow in Fig. 3(c)). Intuitively, force from the topological gradient will push the saddle points up so that the broken bridge gets connected.

2.3 Training a Neural Network

We present some crucial details of our training algorithm. Although our method is architecture-agnostic, we select one architecture inspired by DIVE [16], which was designed for neuron image segmentation tasks. Our network contains six trainable weight layers, four convolutional layers and two fully connected layers. The first, second and fourth convolutional layers are followed by a single max pooling layer of size 2×2 and stride 2 by the end of the layer. Particularly, because of the computational complexity, we use a patch size of 65×65 during all the training process.

We use small patches (65×65) instead of big patches/whole image. The reason is twofold. First, the computation of topological information is relatively expensive. Second, the matching process between the persistence diagrams of predicted likelihood map and ground truth can be quite difficult. For example, if the patch size is too big, there will be many persistent dots in $\mathrm{Dgm}(g)$ and even more dots in $\mathrm{Dgm}(g)$. The matching process is too complex and prone to errors. *By focusing on smaller patches, we localize topological structures and fix them one by one.*

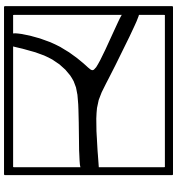
Topology of small patches and relative homology. The small patches (65×65) often only contain partial branching structures rather than closed loops. To have meaningful topological measure on these small patches, we apply *relative persistent homology* as a more localized approach for the computation of topological structures. Particularly, for each patch, we consider the topological structures relative to the boundary. It is equivalent to padding a black frame to the boundary and compute the topology to avoid trivial topological structures. As

[6] Note that a more careful proof should be done for diagrams of 0- and 1-dimension separately.

Table 1: Quantitative results for different models on several medical datasets.

Dataset	Method	Accuracy	ARI	VOI	Betti Error
ISBI12	**DIVE**	0.9640 ± 0.0042	0.9434 ± 0.0087	1.235 ± 0.025	3.187 ± 0.307
	U-Net	**0.9678 ± 0.0021**	0.9338 ± 0.0072	1.367 ± 0.031	2.785 ± 0.269
	Mosin.	0.9532 ± 0.0063	0.9312 ± 0.0052	0.983 ± 0.035	1.238 ± 0.251
	TopoLoss	0.9626 ± 0.0038	**0.9444 ± 0.0076**	**0.782 ± 0.019**	**0.429 ± 0.104**
ISBI13	**DIVE**	**0.9642 ± 0.0018**	0.6923 ± 0.0134	2.790 ± 0.025	3.875 ± 0.326
	U-Net	0.9631 ± 0.0024	0.7031 ± 0.0256	2.583 ± 0.078	3.463 ± 0.435
	Mosin.	0.9578 ± 0.0029	0.7483 ± 0.0367	1.534 ± 0.063	2.952 ± 0.379
	TopoLoss	0.9569 ± 0.0031	**0.8064 ± 0.0112**	**1.436 ± 0.008**	**1.253 ± 0.172**
CREMI	**DIVE**	**0.9498 ± 0.0029**	0.6532 ± 0.0247	2.513 ± 0.047	4.378 ± 0.152
	U-Net	0.9468 ± 0.0048	0.6723 ± 0.0312	2.346 ± 0.105	3.016 ± 0.253
	Mosin.	0.9467 ± 0.0058	0.7853 ± 0.0281	1.623 ± 0.083	1.973 ± 0.310
	TopoLoss	0.9456 ± 0.0053	**0.8083 ± 0.0104**	**1.462 ± 0.028**	**1.113 ± 0.224**

shown in the figure on the right, with the additional frame, a Y-shaped branching structure cropped within the patch will create two handles and be captured by persistent homology.



Training using these localized topological loss can be very efficient via random patch sampling. Specifically, we do not partition the image into patches. Instead, we randomly and densely sample patches which can overlap. As Theorem 1 guarantees, Our loss enforces correct topology within each sampled patch. These overlaps between patches propagate correct topology everywhere. On the other hand, correct topology within a patch means the segmentation can be a deformation of the ground truth. But the deformation is constrained within the patch. The patch size controls the tolerable geometric deformation.

Note that during training, even for a same patch, the diagram $\mathrm{Dgm}(f)$, the critical pixels, and the gradients change over every epoch. At each epoch, we resample patches, reevaluate their persistence diagrams, and the loss gradients. After computing topological gradients of all sampled patches from a mini-batch, we aggregate them for backpropagation.

3 Experiments

We evaluate our method on six natural and biomedical datasets: **CREMI**[7], **ISBI12** [4], **ISBI13** [3], **CrackTree** [45], **Road** [26] and **DRIVE** [36]. The first three are neuron image segmentation datasets. **CREMI** contains 125 images of size 1250x1250. **ISBI12** [4] contains 30 images of size 512x512. **ISBI13** [3] contains 100 images of size 1024x1024. These three datasets are neuron images (Electron Microscopy images). The task is to segment membranes and eventually partition the image into neuron regions. **CrackTree** [45] contains 206 images of cracks in road (resolution 600x800). **Road** [26] has 1108 images from the Massachusetts Roads Dataset. The resolution is 1500x1500. **DRIVE** [36] is a retinal vessel segmentation dataset with 20 images. The resolution is 584x565. For all datasets, we use a three-fold cross-validation and report the mean performance over the validation set.

Evaluation metrics. We use four different evaluation metrics. **Pixel-wise accuracy** is the percentage of correctly classified pixels. The remaining three metrics are more topology-relevant. The most important one is **Betti number error**, which directly compares the topology (number of handles) between the segmentation and the ground truth[8]. We randomly sample patches over the segmentation and report the average absolute difference between their Betti numbers and the corresponding ground truth patches. Two more metrics are used to indirectly evaluate the topological correctness: **Adapted Rand Index (ARI)** and **Variation of Information (VOI)**. They are used in neuron reconstruction to compare the partitioning of the image induced by the segmentation. ARI is the maximal F-score of the foreground-restricted Rand index, a measure of similarity between two clusters. On this version of the Rand index we exclude the zero component of the original labels (background pixels of the ground truth). VOI is a measure of the distance between two clusterings. It is closely related to mutual information; indeed, it is a simple linear expression involving the mutual information.

[7] https://cremi.org/

[8] Note we focus on 1-dimensional topology in evaluation and training as they are more crucial in practice.

Table 2: Quantitative results for different models on retinal, crack, and aerial datasets.

Dataset	Method	Accuracy	ARI	VOI	Betti Error
DRIVE	**DIVE**	**0.9549 ± 0.0023**	0.8407 ± 0.0257	1.936 ± 0.127	3.276 ± 0.642
	U-Net	0.9452 ± 0.0058	0.8343 ± 0.0413	1.975 ± 0.046	3.643 ± 0.536
	Mosin.	0.9543 ± 0.0047	0.8870 ± 0.0386	1.167 ± 0.026	2.784 ± 0.293
	TopoLoss	0.9521 ± 0.0042	**0.9024 ± 0.0113**	**1.083 ± 0.006**	**1.076 ± 0.265**
CrackTree	**DIVE**	**0.9854 ± 0.0052**	0.8634 ± 0.0376	1.570 ± 0.078	1.576 ± 0.287
	U-Net	0.9821 ± 0.0097	0.8749 ± 0.0421	1.625 ± 0.104	1.785 ± 0.303
	Mosin.	0.9833 ± 0.0067	0.8897 ± 0.0201	1.113 ± 0.057	1.045 ± 0.214
	TopoLoss	0.9826 ± 0.0084	**0.9291 ± 0.0123**	**0.997 ± 0.011**	**0.672 ± 0.176**
Road	**DIVE**	0.9734 ± 0.0077	0.8201 ± 0.0128	2.368 ± 0.203	3.598 ± 0.783
	U-Net	**0.9786 ± 0.0052**	0.8189 ± 0.0097	2.249 ± 0.175	3.439 ± 0.621
	Mosin.	0.9754 ± 0.0043	0.8456 ± 0.0174	1.457 ± 0.096	2.781 ± 0.237
	TopoLoss	0.9728 ± 0.0063	**0.8671 ± 0.0068**	**1.234 ± 0.037**	**1.275 ± 0.192**

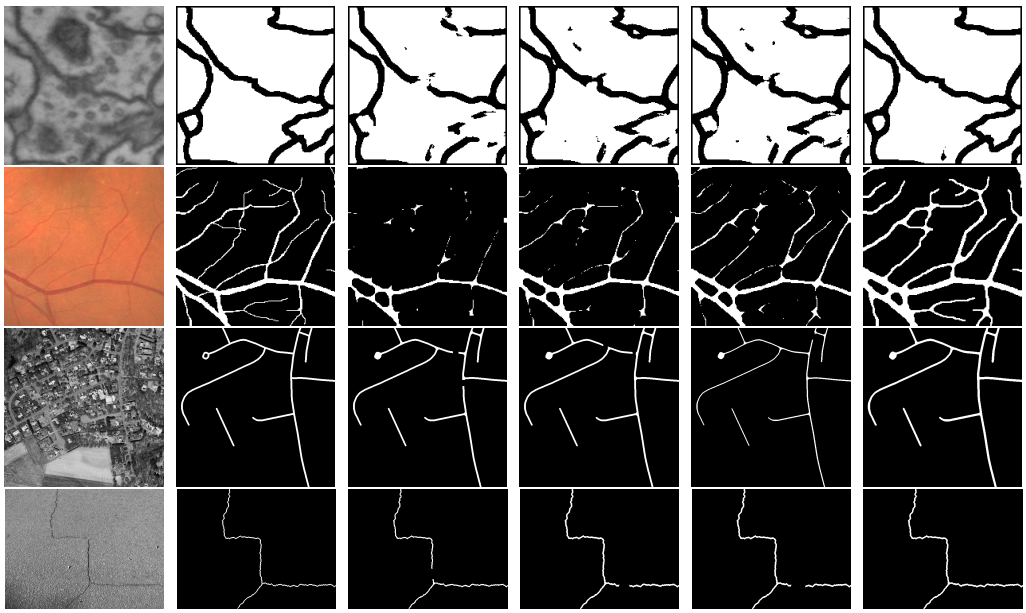
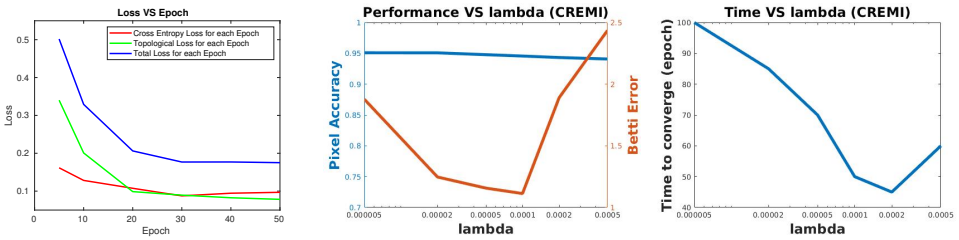


Figure 5: Qualitative results of the proposed method compared to other models. From left to right, sample images, ground truth, results for **DIVE**, **U-Net**, **Mosin.** and our proposed **TopoLoss**.

Baselines. **DIVE** [16] is a state-of-the-art neural network that predicts the probability of every individual pixel in a given image being a membrane (border) pixel or not. **U-Net** [34] is a popular image segmentation method trained with cross-entropy loss. **Mosin.** [27] uses the response of selected filters from a pretrained CNN to construct the topology aware loss. For all methods, we generate segmentations by thresholding the predicted likelihood maps at 0.5.

Quantitative and qualitative results. Table 1 shows the quantitative results for three different neuron image datasets, ISBI12, ISBI13 and CREMI. Table 2 shows the quantitative results for DRIVE, CrackTree and Road. Our method significantly outperforms existing methods in topological accuracy (in all three topology-aware metrics), without sacrificing pixel accuracy. Fig. 5 shows qualitative results. Our method demonstrates more consistency in terms of structures and topology. It correctly segments fine structures such as membranes, roads and vessels, while all other methods fail to do so. Note that the topological error cannot be solved by training with dilated ground truth masks. We run additional experiments on CREMI dataset by training a topology-agnostic model with dilated ground truth masks. For 1 and 2 pixel dilation, We have Betti Error 4.126 and 4.431, respectively. They are still significantly worse than TopoLoss (Betti Error = 1.113).

Ablation study: loss weights. Our loss (Eq. (2.1)) is a weighted combination of cross entropy loss and topological loss. For convenience, we drop the weight of cross entropy loss and weight the topological loss with λ. Fig. 6(b) and 6(c) show ablation studies of λ on CREMI w.r.t. accuracy,



(a) (b) (c)

Figure 6: **(a)** Cross Entropy loss, Topological loss and total loss in terms of training epochs. **(b)** Ablation studies of lambda on CREMI w.r.t. accuracy, Betti error. **(c)** Ablation study of lambda on CREMI w.r.t. convergence rate.

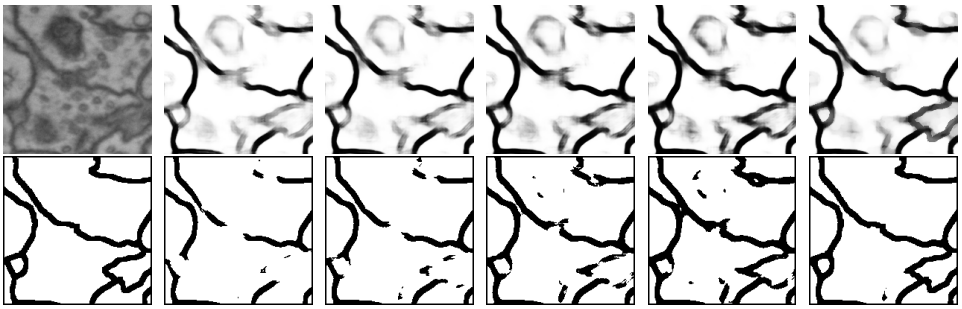


Figure 7: For a sample patch from CREMI, we show the likelihood map and segmentation at different training epochs. The first row correspond to likelihood maps and the second row are thresholded results. From left to right, original patch/ground truth, results after 10, 20, 30, 40 and 50 epochs.

Betti error and convergence rate. As we increase lambda, per-pixel accuracy is slightly compromised. The Betti error decreases first but increases later. One important observation is that a certain amount of topological loss improves the convergence rate significantly. Empirically, we choose λ via cross-validation. Different datasets have different λ's. In general, λ is at the magnitude of 1/10000. This is understandable; while cross entropy loss gradient is applied to all pixels, topological gradient is only applied to a sparse set of critical pixels. Therefore, the weight needs to be much smaller to avoid overfitting with these critical pixels.

Fig. 6(a) shows the weighted topological loss (λL_{topo}), cross entropy loss (L_{bce}) and total loss (L) at different training epochs. After 30 epochs, the total loss becomes stable. Meanwhile, while L_{bce} increases slightly, L_{topo} decreases. This is reasonable; incorporating of topological loss may force the network to overtrain on certain locations (near critical pixels), and thus may hurt the overall pixel accuracy slightly. This is confirmed by the pixel accuracy of TopoLoss in Tables 1 and 2.

Rationale. To further explain the rationale of topological loss, we first study an example training patch. In Fig. 7, we plot the likelihood map and the segmentation at different epochs. Within a short period, the likelihood map and the segmentation are stabilized globally, mostly thanks to the cross-entropy loss. After epoch 20, topological errors are gradually fixed by the topological loss. Notice the change of the likelihood map is only at specific topology-relevant locations.

Our topological loss compliments cross-entropy loss by combating sampling bias. In Fig. 7, for most membrane pixels, the network learns to make correct prediction quickly. However, for a small amount of difficult locations (blurred regions), it is much harder to learn to predict correctly. The issue is these locations only take a small portion of training pixel samples. Such disproportion cannot be changed even with more annotated training images. Topological loss essentially identifies these difficult locations during training (as critical pixels). It then forces the network to learn patterns near these locations, at the expense of overfitting and consequently slightly compromised per-pixel accuracy. On the other hand, we stress that topological loss cannot succeed alone. Without cross-entropy loss, inferring topology from a completely random likelihood map is meaningless. Cross-entropy loss finds a reasonable likelihood map so that the topological loss can improve its topology.

Acknowledgement. The research of Xiaoling Hu and Chao Chen is partially supported by NSF IIS-1909038. The research of Li Fuxin is partially supported by NSF IIS-1911232.

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[2] Bjoern Andres, Jörg H Kappes, Thorsten Beier, Ullrich Köthe, and Fred A Hamprecht. Probabilistic image segmentation with closedness constraints. In *2011 International Conference on Computer Vision*, pages 2611–2618. IEEE, 2011.

[3] I Arganda-Carreras, HS Seung, A Vishwanathan, and D Berger. 3d segmentation of neurites in em images challenge-isbi 2013, 2013.

[4] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.

[5] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 664–673. JMLR. org, 2017.

[6] Chao Chen, Daniel Freedman, and Christoph H Lampert. Enforcing topological constraints in random field image segmentation. In *CVPR 2011*, pages 2089–2096. IEEE, 2011.

[7] Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2573–2582, 2019.

[8] 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.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[11] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[12] David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have l p-stable persistence. *Foundations of computational mathematics*, 10(2):127–139, 2010.

[13] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[14] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE, 2000.

[15] Rolando Estrada, Carlo Tomasi, Scott C Schmidler, and Sina Farsiu. Tree topology estimation. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1688–1701, 2014.

[16] Ahmed Fakhry, Hanchuan Peng, and Shuiwang Ji. Deep models for brain em image segmentation: novel insights and improved performance. *Bioinformatics*, 32(15):2352–2358, 2016.

[17] Jan Funke, Fabian David Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C Turaga. A deep structured learning approach towards automating connectome reconstruction from 3d electron micrographs. *arXiv preprint arXiv:1709.02974*, 2017.

[18] Mingchen Gao, Chao Chen, Shaoting Zhang, Zhen Qian, Dimitris Metaxas, and Leon Axel. Segmenting the papillary muscles and the trabeculae from high resolution cardiac ct through restoration of topological handles. In *International Conference on Information Processing in Medical Imaging*, pages 184–195. Springer, 2013.

[19] Xiao Han, Chenyang Xu, and Jerry L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, pages 1634–1644, 2017.

[22] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics (JEA)*, 22:1–4, 2017.

[23] Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, pages 2004–2013, 2016.

[24] Carole Le Guyader and Luminita A Vese. Self-repelling snakes for topology-preserving segmentation models. *IEEE Transactions on Image Processing*, 17(5):767–779, 2008.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[26] Volodymyr Mnih. *Machine learning for aerial image labeling*. University of Toronto (Canada), 2013.

[27] Agata Mosinska, Pablo Marquez-Neila, Mateusz Koziński, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3136–3145, 2018.

[28] James R Munkres. *Elements of algebraic topology*. CRC Press, 2018.

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

[30] Sebastian Nowozin and Christoph H Lampert. Global connectivity potentials for random field models. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 818–825. IEEE, 2009.

[31] Martin Ralf Oswald, Jan Stühmer, and Daniel Cremers. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *European Conference on Computer Vision*, pages 32–46. Springer, 2014.

[32] Adrien Poulenard, Primoz Skraba, and Maks Ovsjanikov. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, volume 37, pages 13–25. Wiley Online Library, 2018.

[33] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4741–4748, 2015.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[35] Florent Ségonne. Active contours under topology control—genus preserving level sets. *International Journal of Computer Vision*, 79(2):107–117, 2008.

[36] Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4):501–509, 2004.

[37] Jan Stuhmer, Peter Schroder, and Daniel Cremers. Tree shape priors with connectivity constraints using convex relaxation on general graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2336–2343, 2013.

[38] Ganesh Sundaramoorthi and Anthony Yezzi. Global regularizing flows with topology preservation for active contours and polygons. *IEEE Transactions on Image Processing*, 16(3):803–812, 2007.

[39] Srinivas C Turaga, Kevin L Briggman, Moritz Helmstaedter, Winfried Denk, and H Sebastian Seung. Maximin affinity learning of image segmentation. *arXiv preprint arXiv:0911.5372*, 2009.

[40] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[41] Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient computation of persistent homology for cubical data. In *Topological methods in data analysis and visualization II*, pages 91–106. Springer, 2012.

[42] Pengxiang Wu, Chao Chen, Yusu Wang, Shaoting Zhang, Changhe Yuan, Zhen Qian, Dimitris Metaxas, and Leon Axel. Optimal topological cycles and their application in cardiac trabeculae restoration. In *International Conference on Information Processing in Medical Imaging*, pages 80–92. Springer, 2017.

[43] Yun Zeng, Dimitris Samaras, Wei Chen, and Qunsheng Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n–d images. *Computer vision and image understanding*, 112(1):81–90, 2008.

[44] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[45] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.