Image Segmentation with Topological Priors

Shakir Showkat Sofi
CDISE,
Skolkovo Institute of Science and Technology
Moscow, Russia
Shakir.Sofi@skoltech.ru

Nadezhda Alsahanova
CDISE,
Skolkovo Institute of Science and Technology
Moscow, Russia
Nadezhda.Alsahanova@skoltech.ru

***Abstract*—Solving segmentation tasks with topological priors proved to make fewer errors in fine-scale structures. In this work, we use topological priors before and during the deep neural network training procedure. We compared the results of the two approaches on a simple segmentation task using various accuracy metrics and the Betti number error metric, which is directly related to topological correctness. It was found that incorporating topological information into the classical U-Net model performed significantly better. We conducted experiments on the ISBI EM segmentation dataset to confirm the effectiveness of the proposed approaches.**

***Index Terms*—Segmentation, Topological loss, Persistent homology, U-Net.**

I. Introduction

The important task in computer vision is to know the location and shape of objects in the image or which pixel belongs to which object. This task is accomplished through image segmentation, which involves assigning labels to all input image pixels. The use of an end-to-end trained deep network to segment images aids in achieving acceptable per-pixel accuracy. However, when dealing with intricate structures, such as thin connections in neurons and vessels, high accuracy is vital to avoid catastrophic mistakes. For example, a segmentation error in the thin cell membrane could result in the union of two distinct cells. Such errors prevent segmentation algorithms from making decisions on fine-scale structures. To address this problem, we incorporate topological prior knowledge into the segmentation model.

In this work, we investigate two approaches to introducing topological priors. The first approach is adding topological loss to cross-entropy loss, commonly used in segmentation tasks. Topological loss measures the difference between persistence diagrams for true and predicted masks. The second method is to use topological image processing prior to training the neural network. Both strategies improve segmentation performance without sacrificing pixel-wise accuracy.

In Section II we present a short review of existing solutions for the implementation of topological priors in segmentation tasks. Then in section III we present a short theoretical basis of persistence homology III-A, topological loss and its differentiability III-B, and topological input image processing III-C. Finally, we present details of the training process and the results of the experiments in section IV.

II. Related Work

Topologically aware networks have already shown a significant improvement in results, especially for segmentation and classification problems. One such earlier attempt uses topological awareness in the loss function based on the response of selected filters from a pre-trained VGG19 network [1]. It was successful in capturing some topological features, such as the connectedness of small components. It constructs the topologically aware losses, but this approach is difficult to generalize for complex settings. The interpretation and relevance of these captured features were even more difficult. The other researchers proposed a similar scheme, where the output of the second network was used to define a loss function to identify global structural features to enforce anatomical constraints [2]. Topological regularizers were used for classification problems with considerations of the stability of connected components within imposed topological constraints on the shape of the classification boundary [3]. Different ideas have been proposed so far for capturing fine details, some based on deconvolution and upsampling, some using Persistent Homology (PH), and some have used topological processing of inputs for simple geometries for unsupervised tasks [4] before applying the *Chan-Vese, ISODATA, Edge-detections, etc* for segmenting the processed image. For further information, see [5], [6], [7], [8], and the references therein. Most of the methods are problem-specific. Our method is closest to the one proposed in [9], which also poses topological priors in the training phase. Apart from that, our scheme looks at different stages of feeding topological priors with their effectiveness and efficiency for supervised learning tasks.

III. Theory

We have witnessed in recent years that the volume of data is increasing exponentially. However, it is also complex, noisy, multidimensional, and sometimes incomplete, necessitating the development of efficient and robust data analysis methods. The existing methods, which are based on statistics, machine learning, and uncertainty quantification, work well, but when it comes to making sense of vast, multidimensional, and noisy data, the analysis process becomes more challenging. However, advances in computational mathematics have yielded a wealth of insights into the study and application of data in an entirely new set of directions. Topological Data Analysis (TDA) is one such technique that combines computational

geometry, algebraic topology, data analysis, and statistics to uncover intricate patterns and structures within complex datasets. TDA is particularly effective at analyzing the geometric properties of data, such as topological features, and revealing the underlying relationships between data points. By harnessing the power of TDA, researchers and data analysts can gain a more profound comprehension of complex systems and make better-informed decisions based on their findings. Furthermore, the prime focus is on Persistent Homology (PH). We provide a brief overview of PH here, but the reader is directed to [10], [11] for a more in-depth discussion.

A. Persistent homology and Persistent diagrams

Persistent homology is an important algebraic topological tool used in TDA to analyze qualitative data features on multiple scales. PH is robust, dimension- and coordinate-independent, and provides compact summaries of the underlying data. To compute PH, we need two things: a simplicial complex K and the filtration $\mathcal{F}$ defined on K. A simplicial complex is a generalization of a graph that includes 0-simplices (nodes), 1-simplices (edges), 2-simplices (triangles), and so on up to k-simplices, where $k \in \mathbb{N}$ denotes the dimensionality of the complex. It is formally a space that is built from a union of these nodes, edges, triangles, and/or higher-dimensional polytopes [11]. A simplicial complex K is closed under inclusion, which means that if $\sigma' \subseteq \sigma \in K$ then $\sigma' \in K$. A nested sequence of $K_0 \subseteq K_1 \subseteq \ldots \subseteq K_N = K$ of subcomplexes of K is then defined as the filtration $\mathcal{F}$ on K, please refer to [4] for details.

Due to the grid-like structure of images, it became easy to use TDA techniques. Considering an image I of size $M \times N$, the simplicial complex can be obtained from pixel configurations of this image. Technically, the simplicial complex (1-dim simplicial) is defined by connecting each pixel to 8 neighboring pixels. After that, by filling the triangular shapes of this complex, a 2-dimensional simplicial complex can be obtained. Assuming that images have a uniform grid structure, we may prefer the *cubical complex*[1]

PH measures lifetimes of topological features through the birth and death of holes across the filtration [4]. For example, if any hole persists for long consecutive values of the varying parameter(e.g., diameter τ of data point), then that depicts an important object in an image, and small persistent components are noise. For each complex K_i, we compute its topology using Homology group $\mathcal{H}_n$[2], the rank of which is k^{th} Betti number (β_k), represents a k-dimensional hole. β_0 is the number of connected components, β_1 is the number of loops, β_2 is the number of hallow-cavities, and so on. [6]. In the case of image data sets, however, we go up to 1-dimensional holes. There is a pretty good method for visualising and interpreting births and deaths of these holes using *Persistent Diagram (PD).* PD is simply a multiset that contains a point (b, d) for every hole that was born at data point diameter $\tau = b$ and died at data point diameter $\tau = d$. It is a plot between the birth and death of these holes for different values of the diameter of the data points τ.

[1]Space formed by the union of vertex, edges, squares, cubes, and so on.

[2]We don't need all details of computations of Homology groups, as it is already implemented in `Gudhi Library`

B. Topological Loss and Differentiability

It has been demonstrated that training a deep neural network, typically a TopoNet, according to the methodology proposed in [9] can achieve both per-pixel accuracy and topological correctness. Likewise, let's define $\mathbf{f}$, the likelihood map predicted by the network, and $\mathbf{g}$ the ground truth. The overall loss function can be expressed in mathematical terms as the weighted sum of the cross-entropy loss and the topological loss:

$$\mathbf{L}(\mathbf{f}, \mathbf{g}) = \mathbf{L_{BCE}}(\mathbf{f}, \mathbf{g}) + \lambda \cdot \mathbf{L_{topo}}(\mathbf{f}, \mathbf{g}) \tag{1}$$

A binary segmentation task is assumed. Consequently, there is a single likelihood function $\mathbf{f}$, whose value ranges from 0 to 1.

We use information from persistent diagrams of $\mathbf{f}$ and $\mathbf{g}$, i.e., $\mathbf{D}(\mathbf{f}), \mathbf{D}(\mathbf{g})$, to formalize the topological loss, which measures the topological similarity between the likelihood $\mathbf{f}$ and the ground truth $\mathbf{g}$. Mathematically we can write topological loss as:

$$\mathbf{L_{topo}}(\mathbf{f}, \mathbf{g}) = \sum_{(p,p') \in \mathbf{D}(\mathbf{f}), \mathbf{D}(\mathbf{g})} \Big[(\texttt{birth}(p) - \texttt{birth}(p'))^2 + \\ + (\texttt{death}(p) - \texttt{death}(p'))^2 \Big] \tag{2}$$

Where p and p' are the points from persistence diagrams $\mathbf{D}(\mathbf{f}), \mathbf{D}(\mathbf{g})$ sorted by their 'lifetimes':

$$\texttt{death}(p) - \texttt{birth}(p)$$

So, the loss in equation (2) measures the difference between persistent diagrams. It is dependent on the critical thresholds at which topological changes take place, such as the birth and death times of the various dots depicted in the diagram $\mathbf{D}(\mathbf{f})$ [9]. And if $\mathbf{f}$ is differentiable, these vital thresholds are critical points where the derivative of $\mathbf{f}$ equals zero. As a consequence of this, it is possible to represent $\mathbf{f}$ as a piecewise-linear function that has a gradient equal to zero on critical points. Therefore, $\mathbf{f}$ is differentiable. The differentiability of the loss function in equation (2) follows from the differentiability of $\mathbf{f}$. So, gradient of loss function $\nabla_{\mathbf{w}} \mathbf{L_{topo}}(\mathbf{f}, \mathbf{g})$ can be written as follows:

$$\sum_{(p,p') \in \mathbf{D}(\mathbf{f}), \mathbf{D}(\mathbf{g})} \Big[2(\texttt{birth}(p) - \texttt{birth}(p')) \frac{\partial \mathbf{f}(c_b(p))}{\partial w} + \\ + 2(\texttt{death}(p) - \texttt{death}(p')) \frac{\partial \mathbf{f}(c_d(p))}{\partial w} \Big] \tag{3}$$

Where, for each dot $p \in \mathbf{D}(\mathbf{f})$, we designate the birth and death critical points of the related topological structure as $c_b(p)$ and $c_d(p)$, respectively [9]. Using the chain rule, we can easily compute this gradient.

C. Topological input image processing

As previously stated, topological priors can be fed into the segmentation process at many stages, such as the input stage, the training stage, or others. In the above section, we have seen that when we are posing topological priors during the training phase, the process becomes sluggish and computationally expensive. To address this issue, we use a new technique (in U-Net architecture) that processes the image using topological information before applying the actual segmentation algorithm, removing irrelevant objects and noise, and ensuring that the modified image has well-defined topological characteristics. The primary benefit of utilizing this method is that it enables us to perform training with a simple cross-entropy loss function, and it also enables us to perform preprocessing on images whenever we like, prior to solving the actual segmentation problem. Therefore, it should come as no surprise that the method is effective from a computational standpoint. In addition to this, it is simple to apply to unsupervised segmentation problems such as those found in [4], which would not have been achievable (at least easily) if we had used priors during the training phase. The purpose of this work is to incorporate topological image processing into supervised learning.

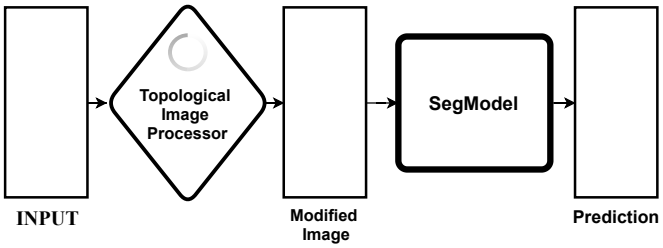

Fig. 1: Topological input image processing for segmentation

The first type of processing of the input image is *image smoothening* since this method has previously shown the advantages with persistent homology [12]. Smoothening can be used to reduce noise, remove pixel-wise focus, to improve quality. It can be performed by spatial or frequency filters, here in our case, we use a `uniform filter` from the standard `scipy` library, which moves the filter mask of size, say $k \times k$, and replaces each pixel value by the average of its neighbors, including itself. We know the real-world images; some images may or may not have borders or other irrelevant objects. So, it becomes difficult to guarantee that most persistent components are important features in the image. The idea of *border modification* was proposed in [4], in which we suppose that the objects of interest in many real-world images do not link with borders, or, in a weaker sense, that most images have objects of interest near the center. The border processing, in technical terms, constructs an image I_b from an original image by ensuring that each pixel within a distance d of the I_b border reaches the minimum value, while the rest values remain fixed. From elder rule[3], this assures that every object that connects to this border will be born through it. In the PD of I_b, all irrelevant portions correspond to a single point with infinite persistence, while features have finite persistence.

Topological input image processing is simply processing the images using the geometric information we have. Here, we will use these as processing steps, but there can be plenty of such modifiers.

After the first step, which is to determine the number of components shown by the input image, using lifetime distribution. Border modification may restrict to finite lifetimes. We may also use any outlier detector for threshold selections. According to the method described in [13], pertinent peaks can be retrieved from the persistence diagram if the diagram includes a band of a certain width that does not include any points. In this work, we apply this method. In a more technical sense, it is the largest empty region parallel to the diagonal. We are able to draw it into the persistence diagram by simply iterating through lifetimes in decreasing order in order to track the difference between consecutive lifetimes [4]. Finally, that threshold is selected, which gives the largest difference between two lifetimes. After selecting the threshold, we increase the contrast between the objects in the image with a lifetime above that threshold and background. For, marking objects in image I, we use the below algorithm, for details please see [13], [4].

Algorithm 1 Object marking in an image based on persistence diagram

Input: Image I, Persistent diagram Dgm, and threshold α
Output: Binary Image J marking objects in I
J, ds = zeros-like(I), list()
Lifetimes = Dgm.death - Dgm.birth
Obj-idxs = where($\alpha < Lifetimes < \infty$)
Obj-idxs = Obj-idxs$\big[$argsort(Dgm.death[Obj-idxs], 'desc')$\big]$

for idx in Obj-idxs **do**
  b = birth-pixel(idx)
  ds.append(death-pixel(idx))
  C = component$\big(I[I(b) \leq Dgm.death[idx]], b\big)$
  new-dval = min$\big($I[Intersects(ds, C), Dgm.death[idx]]$\big)$
  C = component$\big(I[I(b) \leq new - dval], b\big)$
  J[C] = 1
end for

Conceptually, after obtaining the lifetime of components directly from the persistent diagram, we identifying the diagram point having a significant finite lifetime using threshold, then sorting these identified points by decreasing death times. For all sorted points, we first find the image pixel that corresponds

[3]The elder rule asserts that when constructing a persistence diagram, if two components or holes are merged, the youngest component or hole dies. This rule is also known as the *old survive rule* [4]

to the sorted point and save the image pixel for the death time of that point. Then, we add all pixels connected to 'b' before its death to 'C' without overlapping with previous components. Finally, we mark the component for that diagram point in the output and repeat for all points.

In the end, we use multi-variate interpolation to fill in the background pixels in order to get an image with a smooth transition between the background and the objects in the image, as proposed in [4], but here we have a supervised mechanism for training.

IV. Experiments

A. Dataset

The first challenge on the 2D segmentation of neuronal processes in EM images began in May 2012. They have provided training data consisting of 30 image stacks. These images represent a set of consecutive slices within one 3D volume, basically which contains a set of 30 sequential sections from a *serial section Transmission Electron Microscopy (ssTEM)* data set of the Drosophila first instar larva ventral nerve cord (VNC). They have also provided ground-truth images. Below, there is the sample of the image and corresponding mask. We can easily observe the complexity of the geometry of this dataset. Each training image is of size 512×512 in grayscale, with corresponding ground truth. The first pre-processing was to divide the dataset into training and evaluation sets in the 80:20 ratio. Then, we saved the final dataset for which we did all our analyses. Choosing this dataset was a great idea because of its complex geometry, so it was a challenge to use the topological priors in this data.

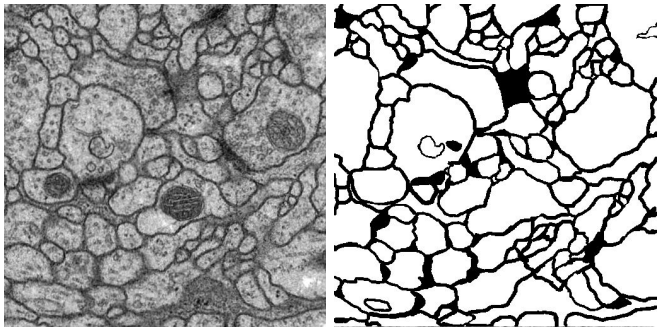

Fig. 2: Image and corresponding Mask

B. Evaluation metrics

We have a lot of leeway to use previously provided measures because the segmentation challenge has already been extensively studied in deep learning research; in our research expository, we employ a range of assessment criteria; some of them are taken from [14], where they were applied to a similar task. In addition, average accuracy, correctness, quality, dice, and many more, we have used the topological relevant metric, i.e., **Betti number error**, which is directly comparing the topology of prediction and ground truth. Below we can see the basic definitions,

- Accuracy: This is the most used metric for binary segmentation problems, which simply tells us the percentage of correctly classified pixels.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \in [0; 1]$$

- Sørensen-Dice Coefficient: This metric is the statistical assessment for the similarity of two given samples.

$$Dice = \frac{2TP}{2TP + FP + FN}, \in [0; 1]$$

- Completeness/inclusion score/Recall: Assessing how well the predicted encompasses the ground-truth.

$$Completeness = \frac{TP}{TP + FN}, \in [0; 1]$$

- Correctness/Precision: The correctness represents the percentage of correctly extracted pixels

$$Correctness = \frac{TP}{TP + FP}, \in [0; 1]$$

- Quality : Quality combines completeness and correctness into a single measure, so it is more general and is defined as:

$$Quality = \frac{TP}{TP + FP + FN}, \in [0; 1]$$

All the above metrics have optimal value 1

- Betti number-Error: This metric is more important in our case as our main focus is on topological segmentation, the betti-number error is more topology-relevant. In this, we randomly sample the small patches of ground truth and prediction, then compute the average absolute difference of their betti-numbers β_k (give us a measure of the number of handles, closed loops, etc in that patch), this reveals how much the geometry of prediction is different from ground-truth. The smaller the betti-number error, the better is segmentation, the ideal value is zero, which means the prediction map has exactly the same topology as that of ground truth, sometimes called *topologically correct.*

C. Details of training procedure

Many deep neural networks are used for the segmentation problem. However, for the task of biomedical segmentation, the best results are shown by U-Net (Fig.3).

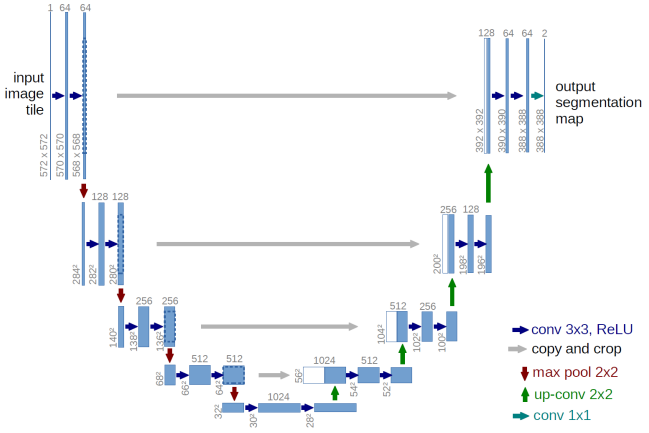

Fig. 3: Architecture of U-Net

We trained U-Net on augmented small patches from images (64×64) as computing topological priors on small patches is faster. Augmentation was used because we have a small dataset, and random choice of patches and their random flipping helps to enlarge the dataset for training. Because of computational complexity, we also take batch size equal to 1. On such a small batch size U-Net with and without topological priors gave better results.

For the realization of topological loss, we used LevelSetLayer2D function from TopologyLayer [4], which calculates persistence diagrams. We added topological loss to CrossEntropy loss (1) with $\lambda = \frac{1}{12000}$ to make these losses have the same order. Also, for visualization of diagrams, we used the Ripser library.

We trained U-Net with and without topological priors on 100 epochs using the Adam optimizer. Examples of changing persistence diagrams for prediction during training with topological loss are on Fig. 4. We can see that the number of points decreased.

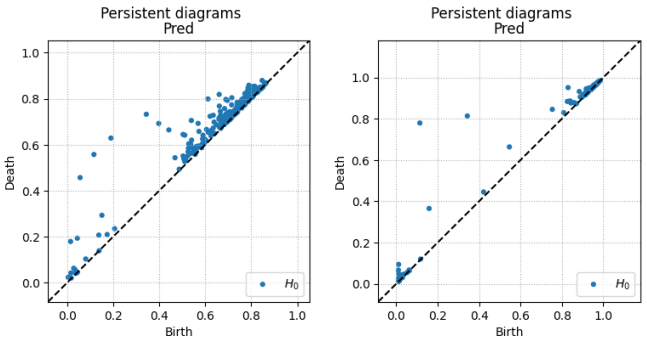

Fig. 4: Persistence diagrams for 5 (left) and 95 (right) epochs

D. Results

After training on small patches, we plotted predictions on all images. A sample for the comparison of predictions of U-Net trained without any topological priors with those trained with a topological loss is shown in Fig. 5.

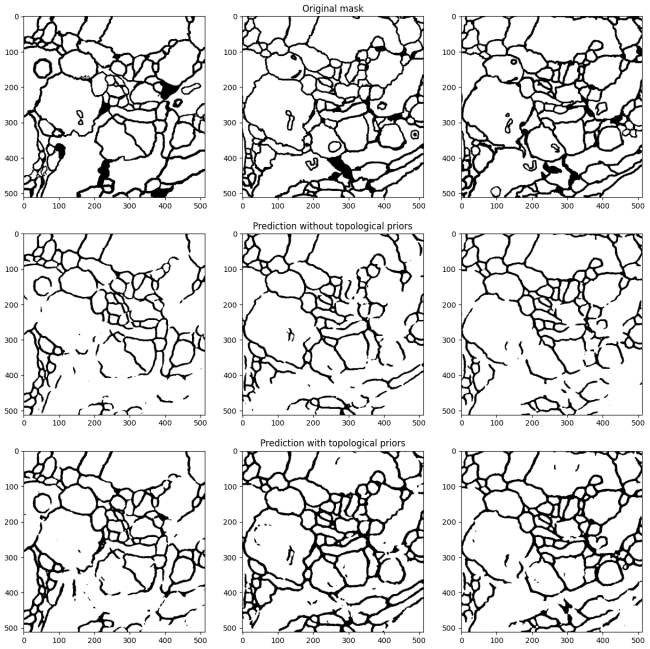

Fig. 5: Predictions without topological priors (middle row) and with topoloss (bottom row)

Also, the comparison of predictions of U-Net trained without any topological priors with trained with a topological input image processing on Fig. 6. We can see that using topological priors makes a prediction with a smaller amount of holes in cells.

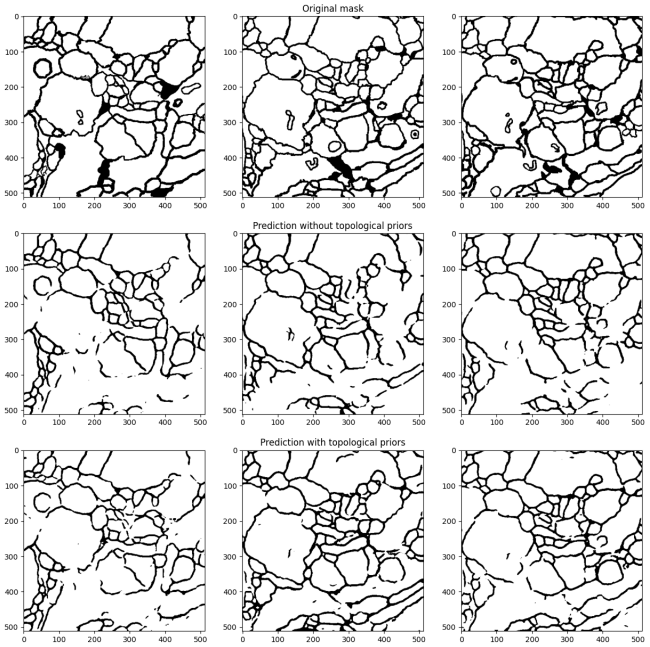

Fig. 6: Predictions without topological priors (middle row) and with topological image processing (bottom row)

[4]TopologyLayer

TABLE I: Results for different strategies.

Metric	*Simple U-Net*	*U-Net with topological priors in training*	*U-Net with topological image processing*
Accuracy	0.899	**0.910**	0.909
Completeness	**0.975**	0.951	0.971
Correctness	0.907	**0.938**	0.92
Quality	0.886	0.894	**0.895**
Dice	0.939	0.944	**0.945**
Betti error	1.16	**1.02**	1.1

Finally, we compute all metrics for various techniques and summarise the findings in table I. From metrics, it can be seen that topological priors give better results than classical segmentation (Simple U-Net) in terms of accuracy, quality, and correctness, as well as topological metric (Betti-error).

V. Conclusion

We have shown that incorporating the topological priors into deep neural networks (U-Net) is a worthwhile idea, especially for problems whose solutions are primarily dependent upon the geometry of the underlying data. We have also shown the usefulness of posing topological information at different stages in the architecture for supervised learning problems. The results of the experiments demonstrate the superiority of the concept that was presented and should be able to discover a variety of applications in data analysis and other prospective domains that are linked. Despite the fact that our findings were encouraging, the training process is somewhat slow, which limits its applicability to datasets that contain multiple variables. So, the plan for the future is to improve the computational efficiency of this method and apply it to multi-dimensional datasets like 3D segmentation tasks. This project's source code is also accessible [5]

References

[1] A. Mosinska, P. Marquez-Neila, M. Kozinski, and P. Fua, "Beyond the pixel-wise loss for topology-aware delineation," 2017.

[2] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. de Marvao, T. Dawes, D. P. O'Regan, and et al., "Anatomically constrained neural networks (acnns): Application to cardiac image enhancement and segmentation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, p. 384–395, Feb 2018. [Online]. Available: http://dx.doi.org/10.1109/TMI.2017.2743464

[3] C. Chen, X. Ni, Q. Bai, and Y. Wang, "A topological regularizer for classifiers via persistent homology," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 2573–2582. [Online]. Available: https://proceedings.mlr.press/v89/chen19g.html

[4] N. G. Robin Vandaele and O. Gevaert, "Topological image modification for object detection and topological image processing of skin lesions," in *Scientific Reports, nature 10:21061*, 2020. [Online]. Available: https://doi.org/10.1038/S41598-020-77933-Y

[5] R. Assaf, A. Goupil, V. Vrabie, and M. Kacim, "Homology Functionality for Grayscale Image Segmentation," in *CRIMSTIC*, ser. Current Research in Information Technology, Mathematical Sciences, vol. 8, Melaka, Malaysia, 2016, pp. 281–286. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02108426

[6] J. R. Clough, I. Oksuz, N. Byrne, J. A. Schnabel, and A. P. King, "Explicit topological priors for deep-learning based image segmentation using persistent homology," 2019.

[7] J. Clough, N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. King, "A topological loss function for deep-learning based image segmentation using persistent homology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[8] Y.-J. Xin and Y.-H. Zhou, "Topology on image processing," in *Proceedings of ICSIPNN '94. International Conference on Speech, Image Processing and Neural Networks*, 1994, pp. 764–767 vol.2.

[9] X. Hu, L. Fuxin, D. Samaras, and C. Chen, "Topology-preserving deep image segmentation," 2019.

[10] H. Edelsbrunner and J. Harer, "Persistent homology — a survey."

[11] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology," *EPJ Data Science*, vol. 6, no. 1, Aug 2017. [Online]. Available: http://dx.doi.org/10.1140/epjds/s13688-017-0109-5

[12] R. Assaf, A. Goupil, V. Vrabie, T. Boudier, and M. Kacim, "Persistent homology for object segmentation in multidimensional grayscale images," *Pattern Recognition Letters*, vol. 112, pp. 277–284, 09 2018.

[13] B. Rieck and H. Leitte, "Agreement analysis of quality measures for dimensionality reduction," in *Topological Methods in Data Analysis and Visualization IV: Theory, Algorithms, and Applications*. Springer International Publishing, 2017, pp. 103–117.

[14] C. Heipke, H. Mayer, C. Wiedemann, and O. Jamet, "Empirical evaluation of automatically extracted road axes," 1998.

[5] https://github.com/ShakirSofi/TopoSeg.git