An algebraic topology approach to the 2018 ISIC Skin lesion challenge

Yu-Min Chung*, Austin Lawson*, Chuan-Shen Hu†, Clifford Smyth*
*Department of Mathematics and Statistics, University of North Carolina at Greensboro, USA
† Department of Computer Science and Information Engineering, National Normal Taiwan University, Taiwan

***Abstract*—This report describes the methods our team used for our entries in the 2018 ISIC Challenge Task 1 and 3. The main tool we used is from the field of algebraic topology, specifically, persistent homology. Topological data analysis (TDA) is a rising field combining algebraic topology and machine learning. This topological tool allows one to extract intrinsic features from an object. To demonstrate this tool and idea, we extract topological features from different types of diseases, and utilize support vector machine to classify them.**

I. Introduction

The ISIC challenge present a unique opportunity for researchers to test novel computer vision ideas for the betterment of detection, hence early treatment of skin cancer. The challenge made use of ISIC's archive of over 13000 dermoscopic images collected from a variety of sources in the hopes of improving Melanoma detection. Task 1 challenged participants to segment the images to filter our healthy skin leaving only the diseased skin. Task 3 challenged participants to diagnose the disease shown in an image from seven different types of skin lesions including Melanoma, Melanocytic nevus, Basal cell carcinoma, Actinic keratosis / Bowen's disease (intraepithelial carcinoma), Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis), Dermatofibroma, and Vascular lesions. In this challenge, we seek to demonstrate some of the capabilities of Topological Data Analysis (TDA) through the use of its powerful tool persistent homology and two novel ideas in this field, namely the persistence curve and persistence statistics.

The outline of this reports is as follows. In Section II, we briefly introduce persistent homology. In Section III, we describe our segmentation algorithm. We describe our classification algorithm in section IV and end with concluding remarks in section V.

II. Persistent Homology

Algebraic topology is a classical subject and has a long history within mathematics. *Persistent homology*, formally introduced in [13], brings the power of algebraic topology to bear on real world data. The field has proven useful in many applications, such as neuroscience [4], medical biology [15], sensor networks [10], social networks [6], physics [12], computation [16], nanotechnology [17] and more. We'll give a brief informal overview of persistent homology for images and refer the reader to [11] for a more detailed explanation.

Informally, homology counts topological features such as connected components (0 dimensional homology), holes (1 dimensional homology), voids (2 dimensional homology), and so on. We can translate 0- and 1-dimensional homology to binary images by counting connected clusters of white pixels as components and connected clusters of black pixels (surrounded by white pixels) as holes. However, when dealing with images, we most often aren't dealing with binary images. Consider a gray-scale image I where each pixel value $I(x, y)$ is between 0 and 255. To obtain a binary image, one might threshold I by some value t to obtain a binary image $T(I, t)$. The pixel function of $T(I, t)$ is $T(x, y, t)$ where $T(x, y, t) = 1$ if $I(x, y) \leq t$ and 0 otherwise. However, this constitutes a choice of threshold that a user would have to make. Persistent homology offers a methodology to consider all possible threshold values. Notice $T(I, t) \subset T(I, s)$ if $s \leq t$. (Here we are alternately viewing $T(I, t)$ as the set of pixels (x, y) for which $T(x, y, t) = 1$.) The **filtration** of the image I is the sequence $\{T(I, t)\}_{t=0}^{255}$. We can calculate the homologies of each of the threshold images in the filtration. By keeping track of when holes and components appear (are born) and disappear (die) throughout the filtration, we can create a topological summary of the image I. The collection of these birth-death pairs give rise to a **persistence diagram**. It has been proven that the persistence diagram is a stable summary of the image I in the sense that small changes in the original image correspond to small changes in the corresponding diagram [9] . These diagrams are an integral part of our algorithm as described in section IV. We can extend this notion to color images by considering each channel in the color space individually.

III. Segmentation

We view RGB images I as tuples $I = [R, G, B]$ where for each pixel (x, y) the red, blue, and green channels $R(x, y)$, $G(x, y)$, $B(x, Y)$ are integer values between 0 and 255. The segmentation algorithm follows:

(Step 1) For each RGB image $I = [R, G, B]$, we transform it into a gray image I^* by setting scalars of channels equally i.e.

$$I^* = \frac{1}{3}R + \frac{1}{3}G + \frac{1}{3}B. \tag{1}$$

(Step 2) Because the region of skin in the lesion looks different than the healthy region in its pixel value, we first compute the average value A of I^*. If the pixel value in I^* is less than a we take this to mean that it is more likely to be a part of illness region. Therefore, the second step is to observe

the life interval of each pixel. Like persistent homology, for each step t, we define I_t^* to be the binary image

$$I_t^*(i,j) = \begin{cases} 0 & \text{if } I^*(i,j) > a \cdot \frac{T-t}{T}, \\ 255 & \text{if } I^*(i,j) \le a \cdot \frac{T-t}{T}, \end{cases} \tag{2}$$

where T is the number of total step we performed and we set $T = 50$ in the application. We also note that in our experiment, pixels in I_t^* usually become 0 when $t \ge 30$. Up to this setting, if we define S_t to be the set of all white pixels in I_t^* in step t, then we get the following filtration:

$$S_1 \supseteq S_2 \supseteq S_3 \supseteq \cdots \supseteq S_T, \tag{3}$$

and this filtration could be illustrated in Figure 1. This computation is similar to the computation for persistent homology. However, instead of computing homologies in each S_t, we measure the life interval of each white pixel in S_1, white pixels in S_1 with long life intervals have a more robust property in the whole image, and are more likely to be a part of the lesion.

(Step 3) In our current segmentation algorithm, for each image, we choose a $1 < T' < T$ as a threshold, and the output segmentation would be mainly determined by connected components in $S_{T'}$. We note that if T' is too large, then there would be too many white connected components, while the original illness region might be broken. So we tend to choose small T'. In proposed method (Version 3), we choose

$$T' = 1 + \left\lfloor \frac{\max_{p \in I^*} L(p)}{30} \right\rfloor, \tag{4}$$

where $L(p)$ is the length life interval of pixel $p = (i,j) \in I^*$, which is defined by the number of $t \in \{1, 2, \ldots, T\}$ such that p has pixel value 255 in S_t i.e. p is a white pixel in S_t. For example, if p is a white pixel in S_1 until S_{t_0} $(1 < t_0 < T)$ i.e. p is white in S_t if and only if $t \le t_0$, then $L(p) = t_0$. Because of the relation mentioned in (3), pixel who has value 0 (i.e. a black pixel) in S_1 would have zero length of life interval.

(Step 4) However, a binary image with small T' usually contains tiny or noisy white components, so we remove those tiny objects and define life scores LS of connected components of $S_{T'}$. More precisely, if C is a (white) connected component of $S_{T'}$, then its life score is defined by

$$\mathrm{LS}(C) = \frac{\sum_{p \in C} L(p)}{|C| \cdot (1 + d(C,o))^2}, \tag{5}$$

where p is a pixel in C and $d(C,o)$ is the minimal distance beween c and middle point o of the image. The reason of setting $d(C,o)$ in (5) as an punishment since we assumed that illness region usually close to the center of the image. A connected component C with higher score means that this region might be more significant in the whole image, so finally, (in Version 3) we choose the (white) connected component with maximal life score as the segmentation of the image.

In validation, two evaluation metrics were considered in proposed method, those are standard Jaccard score (i.e. IOU, intersection over union) and constrained Jaccard score, which is defined by

$$S(A,B) = \begin{cases} \mathrm{IOU}(A,B) & \text{if } \mathrm{IOU}(A,B) \ge 0.65, \\ 0 & \text{otherwise,} \end{cases}$$

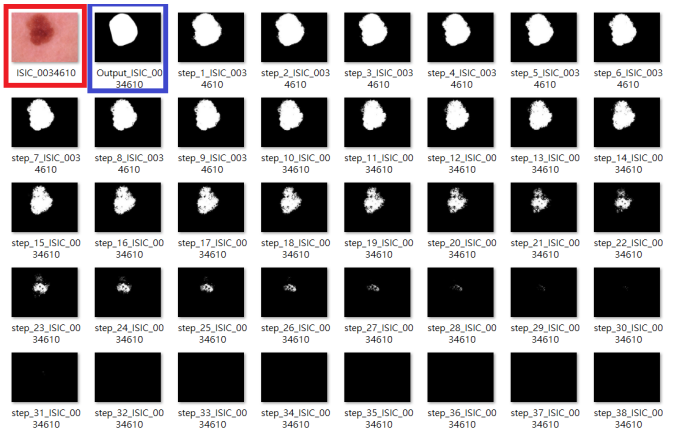

Fig. 1. An example of our main idea in proposed algorithm. The images who are bounded by red and blue bounding boxes are original skin image and output segmentation respectively. The other images are $S_1 \sim S_{38}$ in (3). In this class, our segmentation algorithm selects $t = 2$ by equation (4), and the output segmentation would be the convex hull of the main connected component in S_2, which is derived by (5).

where A and B are two masks of a skin image. Because the segmentation algorithm proposed here doesn't need any training process, entire images (2595 images) in the training set of Task 1 were used for validation. The average Jaccard score is 0.667 and the average constrained Jaccard score is 0.515.

IV. Classification

In this section, we describe our classification approach. The main idea is to extract topological information from persistence diagrams, and use them as features to build a support vector machine model. In this work, we refer to such information as *persistence curves* (PC) and *persistence statistics* (PS). Persistence curves were introduced in [8], and are proven successful to a certain texture dataset; persistence statistics were studied in [7] to describe different types of human red blood cells. Since persistence diagrams contain fruitful topological information, summarizing them is one of the major directions (see e.g. [5], [1], [14] in the field of topological data analysis (TDA). Both PC and PS are considered as summaries of persistence diagrams.

Persistence statistics, as suggested by its name, are statistical measurements of the birth and death coordinates. Given a persistence diagram P, denote b and d by birth and death, respectively, coordinate. Note that (b,d) is a point in P. Let $M = (b+d)/2$, $L = \sum_{(b,d) \in P} (d-b)$, and $p = (d-b)/L$. The PS we used in this challenge were

1) mean of M_0, M_1, p_0, and p_1;
2) standard deviation of M_0, M_1, p_0, and p_1;
3) skewness of M_0, M_1, p_0, and p_1;
4) kurtosis of M_0, M_1, p_0, and p_1;
5) median of M_0, M_1, p_0, and p_1;
6) 25-th and 75-th percentiles of M_0, M_1, p_0, and p_1;
7) interquartile range of M_0, M_1, p_0, and p_1;
8) entropy of p_0 and p_1.

where the entropy is defined as $-\sum_{(b,d) \in P} p \log(p)$.

	Features	Validation Score
Model 1	XYZ curves, XYZ stats	65.6%
Model 2	XYZ curves, XYZ stats, RGB stats	67.2%
Model 3	XYZ curves, XYZ stats, RGB stats, HSV stats	66.0%

TABLE I
BALANCED ACCURACY ON VALIDATION SET.

Persistence curves offer a method to create a vectorized summary of a persistence diagram, thus allowing input to the plethora of available machine learning algorithms. Given a persistence diagram P, we place a function, f, on the off-diagonal points. Next we consider the set $\xi_t = \{(b,d) \in P \mid b \leq t, d > t\}$ where $t = 0, 1, \ldots 255$. Then we vectorize the diagram by calculating

$$F(t) = \int_{\xi_t} f \mathrm{d}\# = \sum_{(b,d)\in\xi_t} f(b,d).$$

For each diagram, this gives us a vector in $\mathbb{R}^{256}$ called a **persistence curve**. The two particular functions that were of greatest use were the$B(b,d) = 1$ giving rise to the Betti curve $\beta(t)$ and $e(b,d) = -\frac{d-b}{L}\log\frac{d-b}{L}$ giving rise to the entropy summary (curve) $E(t)$. The entropy summary and its stability are discussed in [2], [3]. We calculate the curves for the 0 and 1 dimensional persistence diagrams for each channel in our color space. Finally, we fed these features into a linear support vector machine algorithm. The persistence curves we used in our final model are

1) $\beta_0(t)$ and $\beta_1(t)$.
2) $E_0(t)$ and $E_1(t)$.

We summarize our approach in the following. First, we apply the segmentation algorithm discussed in Section III to obtain the image mask. Second, we apply the mask to the original image. Third, we transform the RGB color space into RGB, HSV, or XYZ color space, and extract each channel. Fourth, we use persistent homology software, specially, `Perseus` [18] and `CubicalRipser` [19], to compute persistence diagrams for each channel. Fifth, from each persistence diagram, we calculate persistence curves, and persistence statistics as features. Finally, we use multiclass SVM with "one-against-one" strategy. The general pipeline of our classification is depicted in Figure IV.

The best 3 scores we had on the validation set was 65.6%, 66%, and 67.2% as shown in Table I. This was done by using a subset of the full training set (5000 images chosen randomly from the training set by MATLAB's random seed 1). We then applied the segmentation algorithm as given in section 2. Then after transforming to the XYZ color space, we calculated the $\beta_0(t)$, $\beta_1(t)$, $E_0(t)$, $E_1(t)$ persistence curves as well as all features above for each color channel.

V. CONCLUSION AND FUTURE WORK

The appeal of the persistence curves and persistence statistics lie in their simplicity. The features themselves do not require user defined parameters, thus one only needs to tune the attached machine learning algorithm. In addition, these features give intuitive shape summaries of the original space.

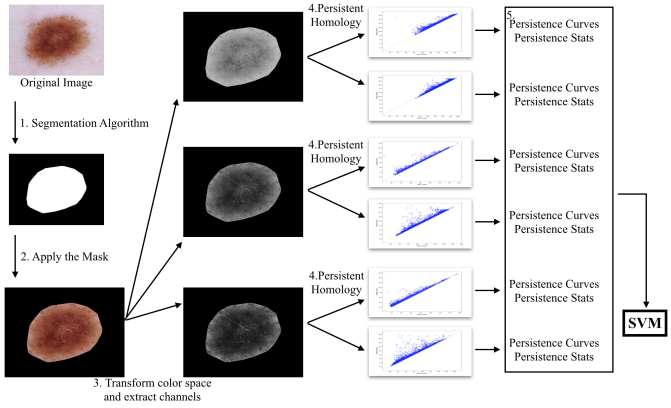

Fig. 2. ISIC Algorithm Pipeline.

The generalized nature of the persistence curve definition allow for a rich library of usable curves. In this paper, we've chosen to combine the persistence stats with the Betti and entropy curves then feed into the linear support vector machine algorithm.

We have also created neural nets that take as input the $\beta_0(t)$ and $\beta_1(t)$ curves of the images. So far the neural net topologies and training we have tried have yielded results that are close but not superior to the support vector approach. We are pursuing this direction vigorously.

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] N. Atienza, R. Gonzalez-Diaz, and M. Soriano-Trigueros. A new entropy based summary function for topological data analysis. *Electronic Notes in Discrete Mathematics*, 68:113 – 118, 2018. Discrete Mathematics Days 2018.

[3] Nieves Atienza, Rocío González-Díaz, and M. Soriano-Trigueros. On the stability of persistent entropy and new summary functions for TDA. *CoRR*, abs/1803.08304, 2018.

[4] Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *The annals of applied statistics*, 10(1):198, 2016.

[5] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

[6] Corrie J Carstens and Kathy J Horadam. Persistent homology of collaboration networks. *Mathematical problems in engineering*, 2013, 2013.

[7] Yu-Min Chung, Madalena Costa, and Sarah Day. Topological data analysis, roughness, and human red blood cells. in preparation, 2018.

[8] Yu-Min Chung and Austin Lawson. Image classification by persistence curves. in preparation, 2018.

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

[10] Vin De Silva, Robert Ghrist, et al. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

[11] Paweł Dłotko and Thomas Wanner. Topological microstructure analysis using persistence landscapes. *Physica D: Nonlinear Phenomena*, 334:60–81, 2016.

[12] Irene Donato, Matteo Gori, Marco Pettini, Giovanni Petri, Sarah De Nigris, Roberto Franzosi, and Francesco Vaccarino. Persistent homology analysis of phase transitions. *Physical Review E*, 93(5):052138, 2016.

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

[14] 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.

[15] Li Li, Wei-Yi Cheng, Benjamin S Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P Bottinger, and Joel T Dudley. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science translational medicine*, 7(311):311ra174–311ra174, 2015.

[16] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7:10138, 2016.

[17] Takenobu Nakamura, Yasuaki Hiraoka, Akihiko Hirata, Emerson G Escolar, and Yasumasa Nishiura. Persistent homology and many-body atomic structure for medium-range order in the glass. *Nanotechnology*, 26(30):304001, 2015.

[18] Vidit Nanda. Perseus, the persistent homology software. http://www.sas.upenn.edu/∼vnanda/perseus, 2013.

[19] Takeki Sudo and Kazushi Ahara. Cubicalripser: calculator of persistence pair for 2 dimensional pixel data. https://github.com/CubicalRipser/CubicalRipser_2dim, 2018.