Journal of Engineering and Applied Science

REVIEWS Open Access

Exploring topological data analysis for information extraction: application to recognition of Arabic machine-printed numerals

Djamel Bouchaffra[1] and Faycal Ykhlef[1]*

This research was conducted while Professor Djamel Bouchaffra was a faculty at Oakland University, Michigan, USA.

*Correspondence: fykhlef@cdta.dz

[1] Division Architecture des Systèmes et Multimédia, Centre de Développement des Technologies Avancées - CDTA, Cité 20 Août 1956, Baba Hassen, PO. Box 17, Algiers 16081, Algeria

Abstract

This manuscript explores the capability of topological data analysis (TDA) based on homology theory (HT: a subfield of algebraic topology) to extract relevant information for recognition of confusing Arabic machine-printed numerals. In fact, topological properties may significantly reduce the confusion between some numerals such as "1" and "4" in the context of small data sets. These two latter digits differ in the sense that digit 1 has no hole and digit 4 has one hole. Our contribution consists of evaluating the contribution of TDA with its invariant descriptors such as Betti numbers in machine-printed Arabic numerals recognition. Our investigation is driven by the following set of actions: (i) we extract Betti numbers invariant features of each numeral image and partition the ten numerals into three different clusters with respect to these features. (ii) We then perform a classification by assigning a test image to its corresponding cluster, and map this image to a numeral using dynamic-time warping as a metric defined in the Freemans' chaincode space. We compared our proposed approach with major state-of-the-art methods depicting various ways of using TDA in character recognition. The advantages and limitations of TDA (including its pros and cons) are discussed further based on numeral recognition results.

Keywords: Topological data analysis, Information extraction, Barcode filtration, Arabic machine-printed numeral recognition, Chaincode representation, Dynamic-time warping, TDA advantages and limitations

Introduction

Character recognition is divided into two categories: online and offline. Online character recognition attempts to recognize characters while they are being written by an individual. However, offline character recognition pinpoints to characters that have been optically scanned by a machine whether they are handwritten or machine-printed (refer to Fig. 1).

Research in recognition of characters is grounded in the document analysis literature [1]. Intensive research studies on character recognition have been widely reported. Most

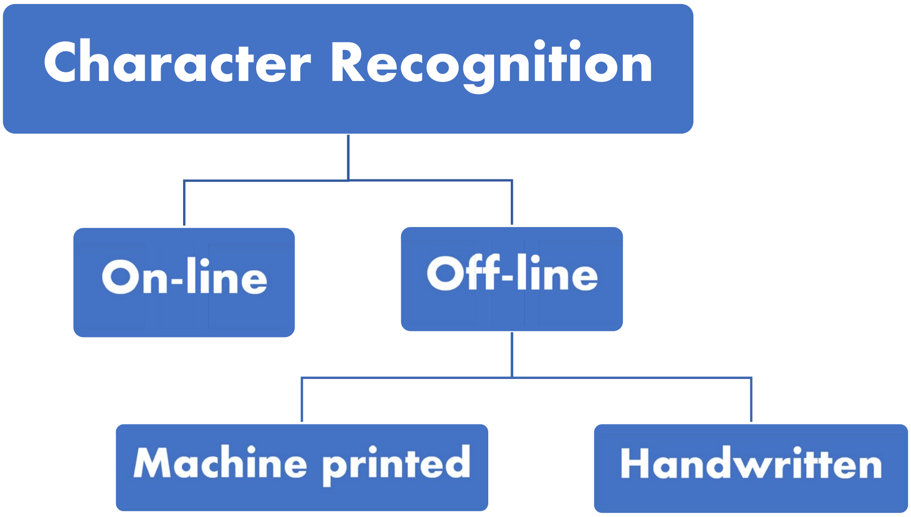


Fig. 1 Tree-based representation depicting online and offline character recognition

of these studies focus mainly on the recognition of handwritten alphanumeric characters. However, not many studies have been devoted to recognition of machine-printed numerals [2]. Khedidja et al. proposed the Hu moment, the number of holes, and surface for feature extraction. However, they implemented seven classifiers for classification of multifont Arabic machine-printed [3]. Moreover, Alqudah et al. implemented a shift and scale invariant procedure for offline machine-printed decimal digit recognition. Their approach is based on a correlation factor [4].

Dhandra et al. introduced a thinning algorithm for multifont numeral recognition. Their feature extraction uses a directional density vector, and their classification relied on a decision tree minimum distance nearest neighbor [5]. Radha et al. proposed a new algorithm for pincode script identification. They developed a new method for chaincode normalization. They conducted recognition of numerals based on neural network and naïve Bayes classifiers [6]. Salameh et al. proposed a method that estimates the number of ends and conjunction nodes of a numeral shape. They conducted a classification using fuzzy logic [7]. Trainable font embeddings [8], horizontal direction features [9], and profile vector [10] are some conventional major features extraction techniques explored in this field. These techniques are very powerful to extract discriminative clues that are relevant for a classification task. Almost all these ideas are based solely on the geometry of the strokes composing the input numeral. In fact, high recognition performance using these traditional techniques has been achieved using large size databases.

More recent approaches that invoke deep graphical networks (deep learning) using private large databases have been investigated to enhance the performance [10–12]. However, large quantities of real data sets are not often publicly available. This data sparseness problem represents one of the challenging problems in statistical learning. To address this issue, researchers rely on three techniques—data augmentation (DA) (including generative adversarial networks)—transfer learning (TL) and collective learning (CL). However, DA based on cropping, padding, and horizontal flipping is not appropriate in several applications (such as license plate recognition, containers code recognition) since augmented data do not occur in real scenarios. Moreover, TL and CL perform optimally only if the original and the destination domains of both models are similar enough. If this is not the case, the trained models might perform worse than what one would expect. Therefore, in this scenario, a template matching technique based on a small data set is far more adequate. Other probabilistic approaches such as hidden Markov models have also been explored; however, they require a large amount of data in order to achieve an accurate recognition [13]. In general, when they are trained on draining large data sets, machine learning (ML) models can exhibit a deep understanding of numeral pattern structures, leading to an extremely high performance. All these research materials have significantly contributed to the success of many applications. The technology of machine-printed numeral recognition is needed in many applications such as license plates recognition (LPR) [12]. Indeed, commercial LPR invokes convolutional neural networks trained with a large quantity of private (not public) datasets of more than thousands of vehicles [12]. In LPR, the background and the number of digits depend on the legislation imposed by the country considered. This modification in legislations explains the creation of many small size public license plate databases around the world [12]. Therefore, in order to rely on a deep learning scheme, one has to gather very large private (and not public) databases; however, these repositories are not made freely available to the scientific community: their purchase incurs a high cost. Furthermore, machine-printed recognition has been also successful in other fields such as automated processing of bank statements [14], barcodes identification [15], documents restoration [16], and zip-code recognition [17]. A recognition system does not tolerate any mistake. For example, an error in a machine-printed US zip-code is significantly expensive since this mail-piece can be incorrectly routed by the US postal service (USPS) to the wrong state (or city) post office. Likewise, an error in a series of digits in bank statement is very detrimental since it induces financial losses. However, there are still some shortcomings that are common to many traditional approaches: (i) geometrical features alone are not sufficient to optimally perceive a machine-printed numeral. It is well known that humans are capable to associate (or perceive) different types of elliptical shapes to the same class even if their roundnesses are different. In other words, topological features of machine-printed numeral may represent valuable clues that can be considered for discriminating between confusing numerals.

To validate this conjecture, we propose to explore persistent homology theory (PHT) that optimally estimate Betti numbers for the purpose of numerals recognition. Our ultimate goal consists of evaluating the significance of topological data analysis (TDA) in the field of Arabic machine-printed numerals recognition in a small data set setting. The advantages and limitations of TDA are underscored. At the end, one should be able to decide whether it is worth to apply TDA or not in this application. It is important to underscore that some authors investigated topological data analysis for the purpose of recognizing handwritten numerals [18, 19]. Our contribution is not directed towards the recognition performance assessment but rather discuss the significance of TDA by highlighting when TDA can or cannot be exploited. Our exploration of TDA relies on topological invariants (known as Betti numbers) during a first stage and subsequently invokes geometrical information during a second stage of classification. The first stage aims at representing the binary images of the ten numerals using their topological signatures (Betti numbers). This representation allows partitioning these images based on their topological signatures into three clusters: $C_1 = \{1,2,3,5,7\}$ contains numerals with no holes, and $C_2 = \{8\}$ includes numerals with two holes and $C_3 = \{0,4,6,9\}$ with one hole. The second stage performs a chaincode representation and a pattern matching distance for classification. In fact, once the numeral input test image is assigned to its corresponding cluster, Free-man's chaincode with eight directions of the binary image is subsequently computed and compared to the set of chaincode templates within the cluster. Finally, dynamic-time warping (DTW) metric is invoked for the overall numeral recognition procedure. It is noteworthy that this similarity measure between temporal sequences does not require any large-scale training and is therefore very useful in the case where the database size is small [20]. Furthermore, we have defined a diversity ratio assigned to the set of fonts used in the numeral databases to assess the robustness of TDA.

Main text

The "Problem description" subsection presents the addressed problem, while the "Overview of topological data analysis" subsection provides some background on TDA and HT. The subsection titled "Exploring TDA for recognition of arabic machine-printed numerals" subsection is devoted to the exploration of TDA for the recognition of machine-printed Arabic numerals. It introduces the topological representation and shows how the partitioning of partition of numerals is conducted through the persistent homology (Betti numbers) principles associated with machine-printed numerals. The subsection titled "experiments and results" showcases the data collection strategy, the importance of font diversity and the numerals clustering scheme based on TDA. The "Experiments and results" subsection. The "Performance comparison and assessment of the proposed approach" subsection depicts a comparative task between our approach and some major state of the art techniques. This very subsection also brings forth the advantages and limitations of TDA for the recognition of machine-printed Arabic numerals. Finally, the conclusion and perspectives are laid out in the "Conclusions" subsection.

Problem description

Our investigation consists of exploring topological data analysis (TDA) with its persistent homology concept and analyzing its advantages and limitations when applied to recognition of Arabic machine-printed numerals for small size data sets. This problem consists of determining the correct Betti numbers (denoted βi) that should improve the performance of Arabic machine-printed numeral. It is noteworthy that Betti numbers represent the core of homology theory.

Overview of topological data analysis

We now provide topological definitions deemed necessary to comprehend TDA. For a deeper understanding of this section, please refer to [21–23].

Simplicial complexes

Definition 1: If $E = \{u_0, u_1, \ldots, u_j\}$ represents a set of $(j+1)$ points taken from an affine space, then this set is affinely independent if the *j* vectors, $u_1 - u_0$, $u_2 - u_0, \ldots, u_k - u_0$, are linearly independent.

Definition 2: The convex hull of $(j + 1)$ affinely independent points is called a j-simplex *s*. The points of E are the vertices of the j-simplex s and *j* is its dimension. It is noteworthy that simplices of dimensions 0, 1, 2, and 3 are vertices, edges, triangles, and tetrahedrons, respectively. Vertices, edges, and triangles can be formed in the two-dimensional space as well as in a three-dimensional space. However, tetrahedrons are not observed in a two-dimensional space. Other simplices such as 5-cell (or n-cell in general) can also be formed in higher dimensional spaces [24].

Definition 3: A simplex spanned by a subset of the vertices of *s* is called a face of *s* (refer to the upper triangle of Fig. 2).

Definition 4: A finite set of simplices, *S*, is a simplicial complex if it satisfies the following two conditions:

- $\forall s \in S$, every face of s belongs to *S*
- for every pair of simplices $(s, \phi) \in (SxS)$, the intersection, $s \cap \phi$, is either an empty set or a face common to both simplices.
- The highest dimension of any simplex contained in *S* represents its dimension.

Abstract simplicial complex

Definition 5: An abstract simplicial complex is a sub-collection, B (generic set) of S, such that $s' \in B$ and $\phi' \subseteq s' \Rightarrow \phi' \in B$. The sets s' are defined as abstract simplices.

The abstract simplex dimension is defined as equal to the simplex cardinality which is the number of sets. However, the dimension of B is the maximum dimension exhibited by any of its abstract simplices.

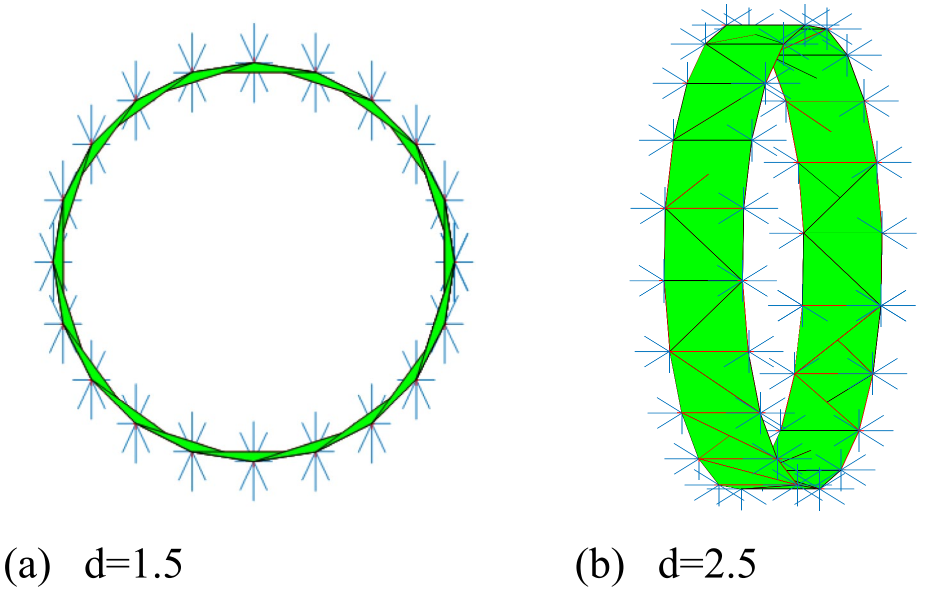


Fig. 2 Simplicial complex VR (E; d). **a** Simplices are formed when the distance d between a a pair of points is 1.5 (2D space), whereas **b** corresponds to $d = 2.5$ (3D space)

Vietoris-Rips simplicial complex

Several geometrical constructors such as Alpha complexes, Čech complexes, and witness complexes are available in the literature [25]. To recover the persistent homology of a space from a finite sample of points, we have invoked the Vietoris-Rips (VR) constructor [23]. We now provide a definition of this constructor.

Definition 6: If E is a finite set of points in $\mathbb{R}^2$, and d is a positive real number, then the Vietoris-Rips complex of E and d, denoted VR (E; d), consists of all abstract simplices in 2E (power set) whose vertices' distances do not exceed d from one another. We namely connect any two vertices whose distance from each other is no more than d by an edge, and we add a triangle or a higher-dimensional simplex to the complex if all its edges are contained in the complex (refer to Fig. 3).

Homology and persistent homology

We now define the notions of filtration, pth homology group, persistent homology, death and birth, and barcodes:

Construction of series of simplicial complexes **Definition 7:** A filtration on a simplicial complex S is a collection of subcomplexes $\{S(\mathrm{d}) \mid \mathrm{d} \in \mathbb{R}\}$ of S such that $S(\mathrm{d}_1) \subset S(\mathrm{d}_2)$ whenever $\mathrm{d}_1 \leq \mathrm{d}_2$. However, the filtration value of a simplex $s \in S$ is the smallest d such that $s \in S(\mathrm{d})$. Larger simplicial complexes are built from basic ones using different values of d to form a filtration.

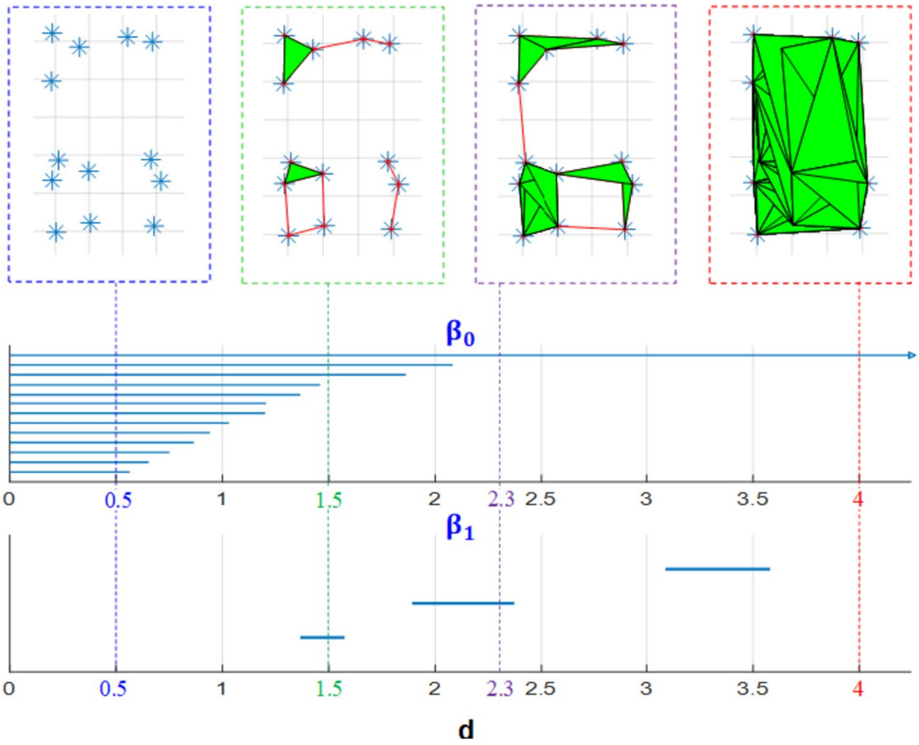


Fig. 3 Formation of the barcode graph associated with a filtration as a function of the threshold value *d*

Computation of Betti numbers A simplicial complex generates a group structure through the addition of p-simplices.

Definition 8: The union of all faces is the boundary of the simplex, and its complement is called the interior, or the open simplex.

Definition 9: The free group obtained through this addition of p-simplices is called the chain group, G_p.

Definition 10: The *p*th homology of a simplicial complex *S* is the quotient vector space:

$H_p(S) = \mathrm{Ker}(l_p)/\mathrm{Im}(l_p + 1) = Z_p/B_p$, in which Z_p is the group structure of all p-chains with empty boundary (which is the sum of boundaries of its simplices); and Bp, which is a subgroup of G_p, is thus a subgroup of Z_p, known as the boundary group. The notion of Betti numbers derives from this quotient group. The function *lp* is a linear map defined as follows:

$$\begin{array}{l} C_p(S)^{lp} \rightarrow C_{p-1}(S) \\ \sigma \rightarrow \sum_{s \subset \sigma, s \in S_{p-1}} s. \end{array} \tag{1}$$

$C_p(S)$ is the $\mathbb{F}_2$ vector space (2-elements field) whose basis is expressed by the p-simplices of *S*.

Definition 11: The rank of H_p (the smallest cardinality of a generating set for H_p) is known as the *p*th Betti number given by β_p. In fact, the *p*th Betti number records the number of p-dimensional holes in *S*. When $p = 0$, β_0 computes the number of path-connected components of *S*. If $E = \mathbb{R}^3$, β_1 accounts for the number of independent tunnels (holes), and β_2 tallies the number of cavities.

For example, the shape in Fig. 2a depicts a circle whose Betti numbers are (1,1). However, Fig. 2b depicts an empty cylinder whose Betti numbers (β_0, β_1, β_2), respectively are (1,1,0).

Persistent homology (PH) PH aims to reveal the different scales (values of *d*) from which a set of points can be observed within a single formalism. Therefore, several simplicial complexes are exhibited in order to approximate the true shape of the cloud of points (dataset). PH seeks to determine the homology that best represents the true shape disclosed by the set points. In fact, this optimal homology attempt to generate the qualitative noise-free features.

Definition 12: Let $S_1 \subset S_2 \subset \ldots \subset S_n = S$, be a filtered simplicial complex; therefore, we define the *p*th persistent homology of *S* as the pair:

$$\left(\left\{H_p(S_i)\right\}, 1 \leq i \leq n, \left\{f_{ij}\right\}, 1 \leq i,j \leq n\right) \tag{2}$$

where $\forall(i,j) \in \{1,\ldots,n\}$, $i \leq j$, the linear maps: $H_p(S_i) \xrightarrow{fij} H_p(S_j)$ represent the maps induced by the inclusion maps from S_i to S_j.

Birth and death The concept of birth and death is essential in the computation of Betti numbers that represent the qualitative features.

Definition 13: We assert that $x \in H_p(S_i)$ ($x \neq 0$) is born in $H_p(S_i)$, if it is not present in the image of $f_{i-j,i}$. Likewise, x is said to die in $Hp(S_j)$ if $j > i$ is the smallest index such that $f_{ij}(x) = 0$.

Barcodes Through the variation of d, Betti numbers are computed for each simplicial complex. From one step to the next one, matching up the births and deaths (as described in the previous section), we obtain a set of bars, known as the barcode of the filtration. A bar corresponds to a class in one of the homology groups. Figure 3 shows barcodes obtained via a filtration on a simplicial complex based on 13 points randomly spaced. For example, for $d = 1.5$, the persistent homology (barcodes) reveals, three connected components ($\beta_0 = 3$), and one hole ($\beta_1 = 1$). However, for $d = 2.3$, it discloses one connected component ($\beta_0 = 1$), and one hole ($\beta_1 = 1$). Finally, for $d = 4$, ($\beta_0 = 1$) and ($\beta_1 = 0$).

Statistical interpretation of topological information Once barcodes have been produced through PH, one needs to provide an interpretation of the computed results. The question that is addressed is how to assert that topological information extracted from a certain sample of points is compatible with the topological information derived from a null model. However, it is important to distinguish between two scenarios: (i) the set of points depicts an object with known topological properties, and (ii) the set of points describes an object with unknown topological properties.

In the first scenario, a barcode is a theoretical parameter and needs to be compared to the observed one (computed via PH). A goodness of fit statistical test such as the chi-square indicates how well does the observed barcode match the theoretical one. For example, if the object is the numeral “0” (one class among 10), its barcode exhibits one connected component ($\beta_0 = 1$) and one hole ($\beta_1 = 1$) and represents the null model (expected value in a null hypothesis of a statistical test of significance). However, the observed barcode computed through PH using different random fonts of the same numeral “0” can be viewed as the empirical mean.

The chi-square can therefore be applied to assert the significance of the observed barcode. In the second scenario, topological properties of an object represented via a set of points are unknown. In this case, the null hypothesis of a statistical test can be provided by a generative model (such as many realizations from a probability distribution of barcodes). For example, a large number of barcodes (representing a population) can be generated using many fonts with different resolution levels. All these barcodes are assigned to the numeral class “0”. Topological properties (such as the Betti numbers) of the observed barcode can subsequently be compared to those present in the population using a statistical test of significance. Finally, one can underscore that the computation of PH through data can only be performed through the sequence of tasks (pipeline) depicted by Fig. 4.

Exploring TDA for recognition of Arabic machine printed numerals

We show in this section how TDA can be exploited to improve recognition of confusing Arabic machine-printed numerals.

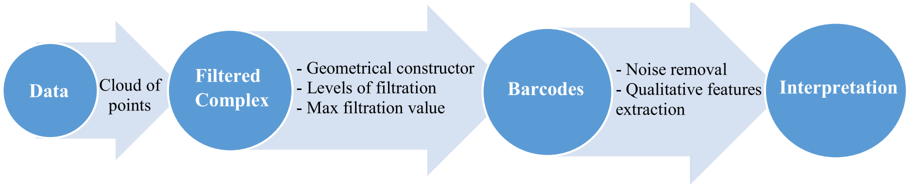


Fig. 4 Persistent homology successive steps (pipeline)

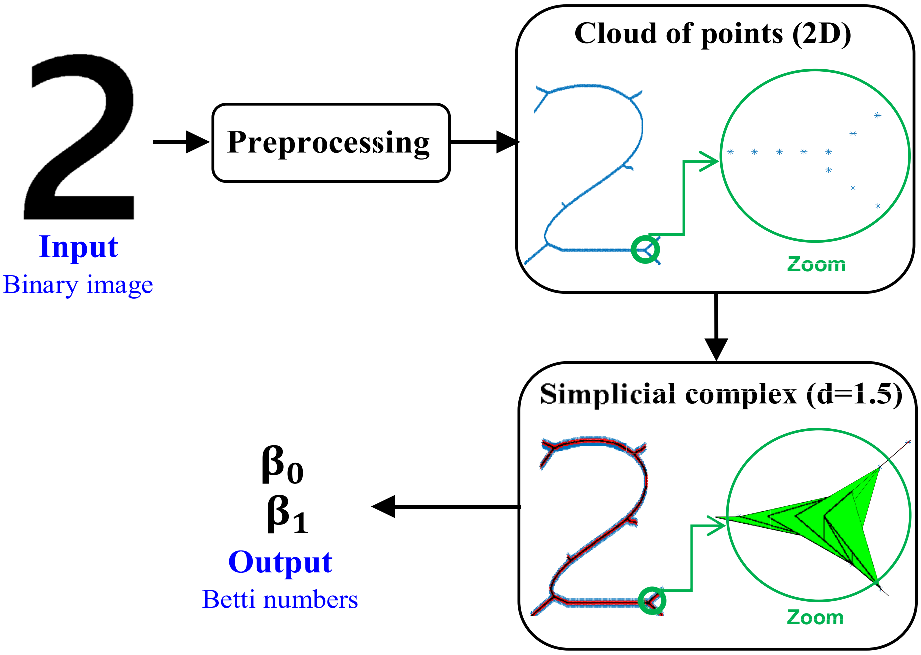


Fig. 5 Simplicial complex assigned to the numeral image "2" is built from the set of pixels issued from the skeletonization operation. A simplicial complex assigned to the zoomed region is shown

Numeral image characterization via simplicial complexes

We apply homology theory on a cloud of points E forming the skeleton of a printed numeral image within a set of various fonts. This set of points and its neighborhood system satisfying a set of axioms represents a topological space. The skeleton formed is obtained during a preprocessing phase including image complement, bounding box computation, scaling (or resizing) for normalization, and then skeletonization of the binary numeral image I [26]. The cloud of points is defined in a two-dimensional metric space (E; d) whereby a neighborhood system is formed using VR constructor (refer to Fig. 5).

Noise generated from the thinning operation

Once the thinning operation is applied on the bitmap image, this latter object becomes noisy. This noise prevents any classification from being accurate. One can easily notice the branches on the numeral contour of digit 2 depicted by Fig. 5. A simpler approach for extracting topological features would be computing a graph out of the skeleton.

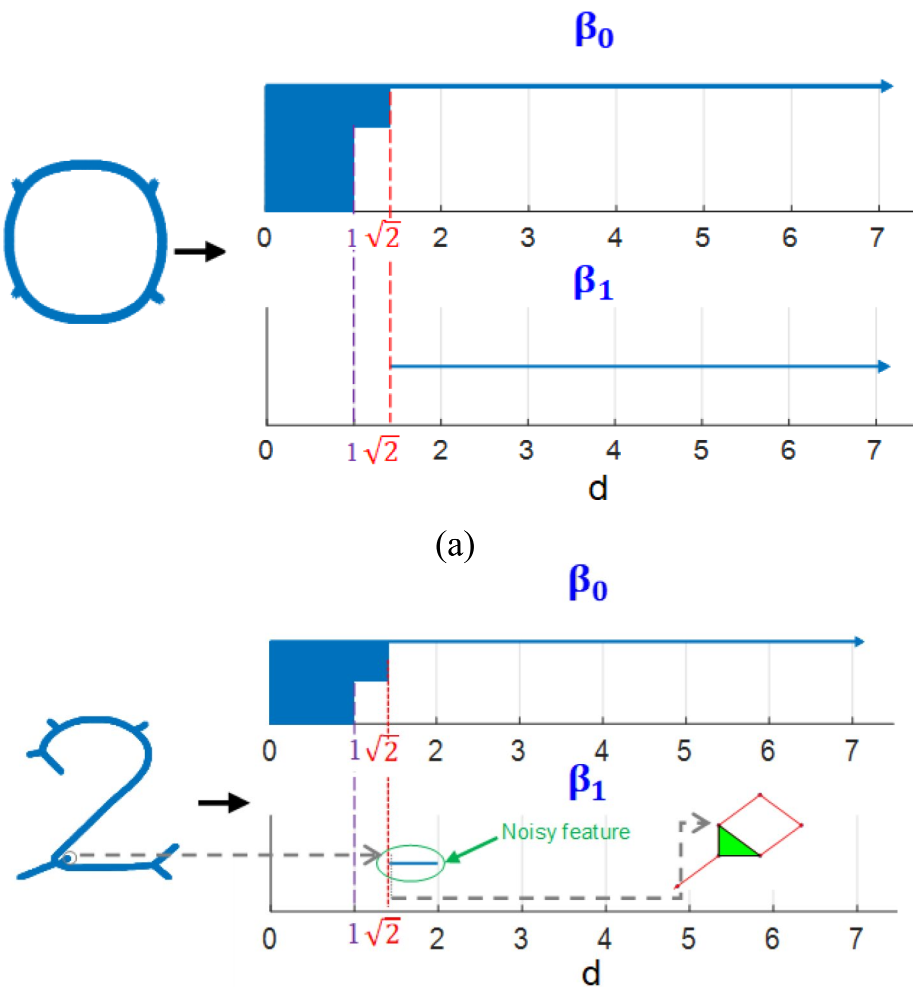
Once this is done, the count of cycles in this graph would have been an option for the estimation of topological clues. However, such an approach would require a table that depicts the connectivity between vertices (pixels). Unfortunately, this information is not available since only a noisy bitmap image obtained from the skeletonization procedure is in our possession (refer to Fig. 5). Therefore, TDA that does not require the vertices connectivity is the best alternative. Another variant to compare is using an alternative method without computing the skeleton and computing PH on the VR-filtration associated to the raw image after a binarization process. However, although this option is possible, its complexity computation is intractable.

Extraction of Betti numbers as topological features

The goal of this phase is to compute the optimal value d^* from which topological features of numerals can be extracted. These numerals are represented within various set of fonts. Once simplicial complexes assigned to a numeral are computed (as part of a filtration), topological features are extracted. However, it is noteworthy that the feature extraction task is supervised. In other words, the class assigned to a numeral image is known a priori. The cavities do not appear since only two-dimensional simplicial complexes are considered. Figure 3 shows a set of points whose distance between adjacent points is random. In this case, the extraction of Betti numbers assigned to the digit “6” is conditioned on an unknown value d. One can notice from this figure that the value $d=2.3$ depicts the numeral “6” with its Betti numbers β_0, and β_2, equal both to 1. However, when the value of d is above 3.6, the hole contained in the numeral “6” disappears. Fortunately, in the case of machine-printed numeral recognition, where the input is a bitmap image, the distance between all adjacent pixels composing this image is either $d=1$ (vertical or horizontal positions) or $d=\sqrt{2}$ (diagonal positions). Therefore, these two values represent the two possible scales from which any numeral is perceived. Ideally, the PH graph assigned to numerals relies only on two d values (refer to Fig. 6a). However, due to some artifacts emanated from the numeral skeletonization procedure, noisy features could be generated for some particular fonts when d exceeds the value of $\sqrt{2}$. For instance, the horizontal bar in the β_1 graph of the numeral “2” represents a noisy feature and therefore should be disregarded. When the threshold distance d between adjacent pixels is chosen between $\sqrt{2}$ and 2, a fake hole is formed (red quadrilateral in Fig. 6b). To remove this quirk, one has to select a threshold value greater than 2. Thus, non-adjacent pixels will be connected by creating additional triangles and edges which suppress this fake hole. Since our task is supervised, we could determine the optimal value d^* that precisely represents the topological features assigned to numerals. The optimal value sought should not generate fake holes nor eliminate real holes that topologically characterize numerals. This optimal value d can be selected from the interval $[\sqrt{2}, \sqrt{\mathrm{M}^2+\mathrm{N}^2}]$, where M and N are the number of rows and columns, respectively, of the resized image I.

Partitioning of numeral images based on their Betti numbers

This task is achieved by clustering the ten numerals with respect to their topological signatures (i.e., after the computation of their Betti numbers). Three clusters have been identified using this procedure: cluster $\mathrm{C}_1=\{1,2,3,5,7\}$ contains numerals with no holes,



(a)

(b)

Fig. 6 Barcode graphs depicting the qualitative topological features for the numerals "0" and "2". **a** Betti numbers for numeral "0" are ($\beta_0 = 1, \beta_1 = 1$) and **b** Betti numbers for numeral "2" are ($\beta_0 = 1, \beta_1 = 0$)

cluster $C_2 = \{8\}$ includes numerals with two holes, and cluster $C_3 = \{0,4,6,9\}$ includes those numerals with one hole. In this application, all numerals have one connected component.

Recognition of numerals

We cover in this section the generation of templates using Freeman image representation and the classification task of numerals based dynamic time warping.

Freeman chaincode computation **Definition 14:** Freeman chaincode is a compact method for representing the contours of an object. This representation was first proposed by Herbert Freeman [27].

This method represents a boundary of an object by a connected sequence of straight line segments of specified length and direction. More precisely, this depiction is based on 4 (or) 8 connectivity of the segments. The direction of each segment is coded through a numbering procedure. A boundary code which is a sequence of these directional numbers is called a Freeman chaincode. The chaincode of a boundary depends on the initial point considered. Code numbers exhibits one possibility to characterize the shape of the boundary. A chaincode is extracted by delimiting the contour in a counter clockwise manner by recording the directions as we move from one contour pixel to the next (refer to Fig. 7).

Dynamic time warping distance **Definition 15:** DTW is an algorithm that computes the similarity between two temporal sequences, which may have different speeds.

This methodology has been proposed independently in the literature by Vintsyuk [28] and Sakoe [29] for speech applications. DTW is a way to compare two temporal sequences that do not coincide perfectly. It is a method that computes the optimal matching between two sequences. It aims at determining the temporal alignment that minimizes Euclidean distance between two aligned series, (refer to Fig. 8). DTW is useful in many areas such as speech recognition, data mining, financial markets, and others.

- DTW distance and warping path

Let's consider two temporal sequences: $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots., b_n\}$, the variable DTW(i, j) denotes the DTW distance between $A_{1\ldots i}$ et $B_{1\ldots j}$. This distance is expressed through the following recursive equation:

$$DTW\left(i,j\right) = \begin{cases} 0ifi = 0andj = 0 \\ Inftyifi = 0orb = 0, andi \neq j \\ dist\left(a_i, b_j\right) + min\begin{cases} DTW\left(i-1,j\right) \\ DTW\left(i,j-1\right) \\ DTW\left(i-1,j-1\right) \end{cases} if 1 \leq i \leq mand 1 \leq j \leq n, \end{cases}$$

where $dist\left(a_i, b_j\right)$ represents the distance between the two elements a_i and b_j.

For example, if the sequences $A = \{5, 5, 6, 6, 6, 7, 7, 7, 7\}$ and $B = \{1, 1, 1, 2, 2, 2, 3, 3\}$, then the computation of the dynamic time warping distance is depicted by Table 1.

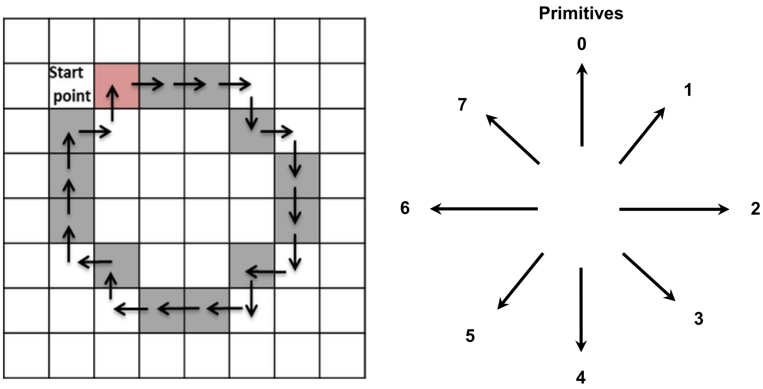


Fig. 7 Shape boundary containing only vertical (code 0 and 4) and horizontal (code 2 and 6) directions in the eight-connectivity of Freeman chaincode representation system

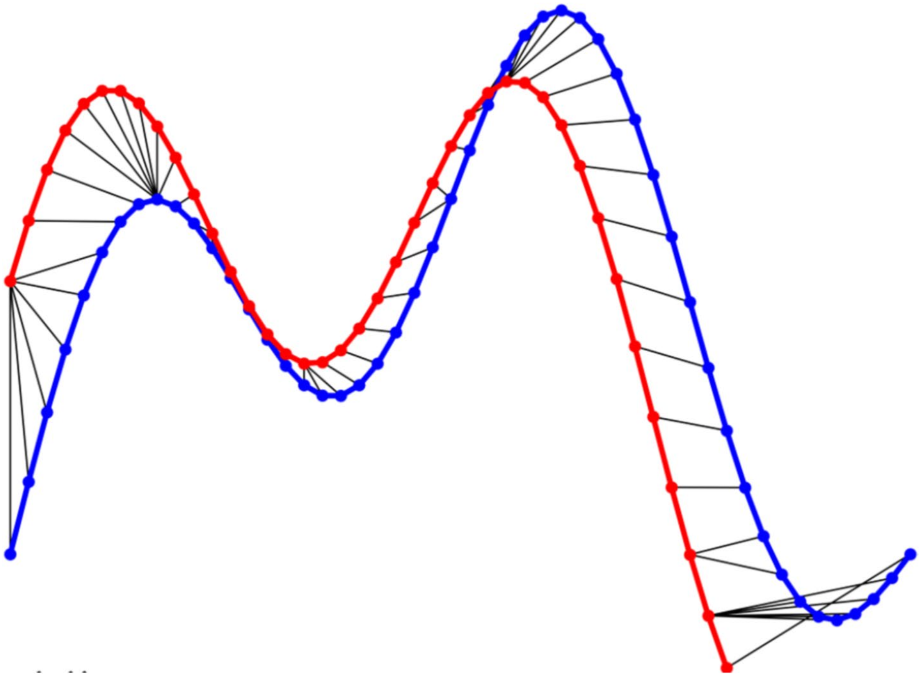


Fig. 8 Minimization of the Euclidean distance between two time series sequences

The warping path is the one that connects the bold numbers in each column of Table 1. The total DTW distance corresponds to the last bold element, equal in this example to 1125. The term "Infty" corresponds to the mathematical sign "∞".

- Restrictions on the warping function

The warping path is determined based on a dynamic programming approach that aligns two sequences. Computing all possible paths is "combinatorically intractable". Therefore, there is a need to limit the number of possible warping paths. The following constraints are required to reduce the search space.

❖*Boundary condition*: This constraint ensures that the warping path begins with the starting points of both signals and ends with their endpoints.
❖*Monotonicity condition*: This constraint preserves the time-order of points (there is no return in time)
❖*Continuity (step size) condition*: This constraint restricts the path moves to adjacent points in time (not jumps in time).
❖*Warping window condition*: Permissible points can be limited to fall within a given warping window of width r (Sakoe-Chiba band). An acceptable warping path uses chess king moves that are as follows:
❖Horizontal moves
❖Vertical moves
❖Diagonal moves

Table 1 Computation of the optimal warping path whose cells are in bold

B / A		**1**	**1**	**1**	**2**	**2**	**2**	**3**	**3**
	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
5	Inf	**4**	8	12	15	18	21	23	25
5	Inf	**8**	8	12	15	18	21	23	25
6	Inf	**13**	13	13	16	19	22	24	26
6	Inf	18	**18**	18	17	20	23	25	27
6	Inf	23	**23**	23	21	21	24	26	28
7	Inf	29	29	**29**	**26**	**26**	26	28	30
7	Inf	35	35	35	31	31	**31**	30	32
7	Inf	41	41	41	36	36	36	**34**	34
7	Inf	47	47	47	41	41	41	38	**38**

Template generation Templates are generated by computing the Freeman's chaincode representation of each numeral within this set of three clusters to respectively form Codebook C_1, Codebook C_2, and Codebook C_3. The Codebook generation is achieved through the sequence of tasks explained through the diagram depicted by Fig. 9 (proc function). It is worth noticing that only a few fonts are considered during template generation.

Classification of numerals This classification phase is conducted as follows: (i) initially, we assign the numeral input test to its corresponding cluster (the winner cluster). (ii) We then apply all tasks described by Fig. 9 to compute its Freeman's chaincode. We finally invoke dynamic-time warping distance to classify all numerals. Since we are conducting a multi-font numeral recognition task, the optimal class associated to the input test is the most frequent class of numerals across all fonts considered in the Codebooks. This optimal class is determined via the statistical mode measure. Therefore, the decision criterion is expressed mathematically as follows: Determine the class ω^* such that:

$$\boldsymbol{\omega^* = \mathbf{Mode}\left\{\mathbf{argmin}_{\omega_i}\left(\mathbf{DTW}\left(\mathbf{S}, \mathbf{S}^{f_j}_{\omega_i}\right)\right), \forall \mathbf{j}\right\}} \tag{3}$$

where S is the input Freeman's chaincode and $S^{f_j}_{\omega_i}$ is the chaincode assigned to class ω_i for the font f_j. One can notice from Fig. 10 that cluster 2 has only one numeral and therefore does not require any further processing. It is crucial to outline that pattern-matching technique does not need training and is very efficient within the context of small-size databases. Since TDA was applied a priori, only a few numerals are in competition and pattern matching as a subsequent task removes the confusions of the remaining numerals (refer to Fig. 10).

Experiments and results

This section is devoted to the collection of data, the computation of the font diversity ratio, and the numerals clustering based on TDA.

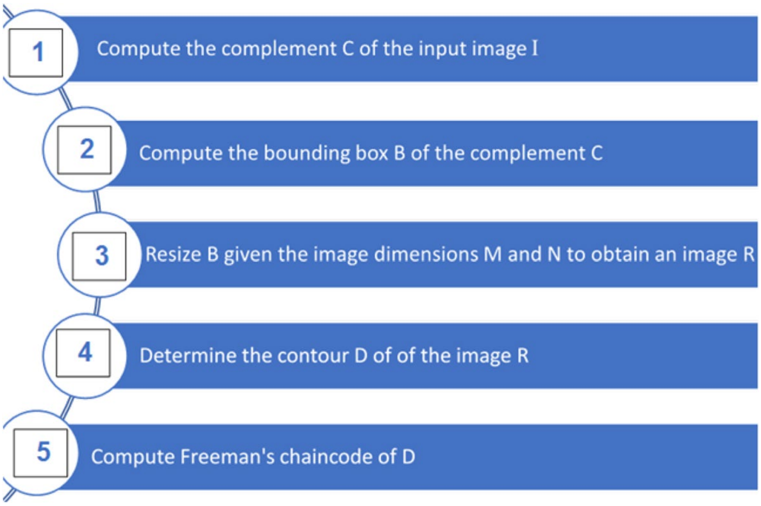

1
Compute the complement C of the input image I
2
Compute the bounding box B of the complement C
3
Resize B given the image dimensions M and N to obtain an image R
4
Determine the contour D of of the image R
5
Compute Freeman's chaincode of D

Fig. 9 Pipeline for computing the Freeman's chainecode of a numeral image (proc function)

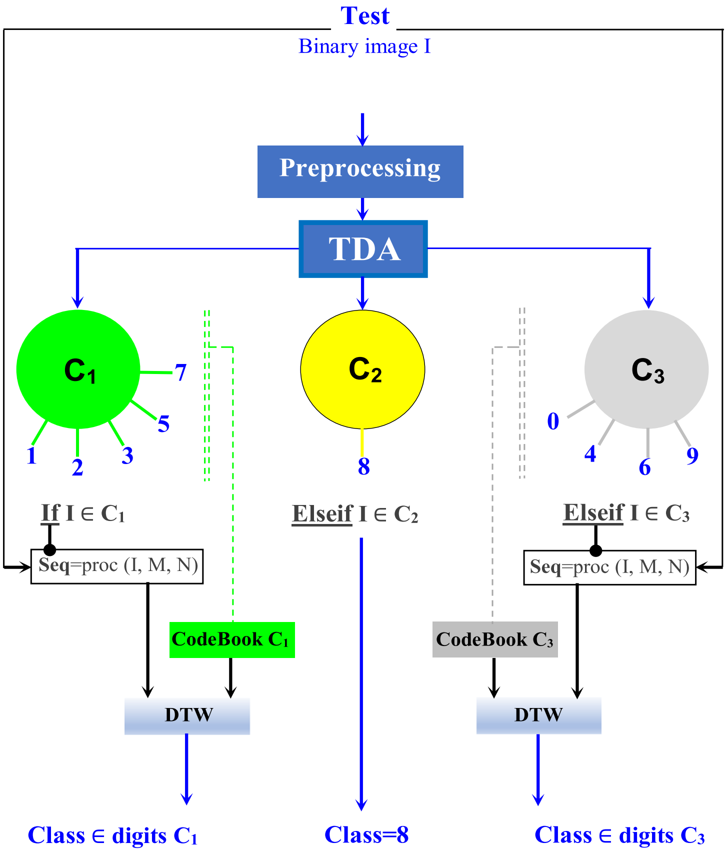


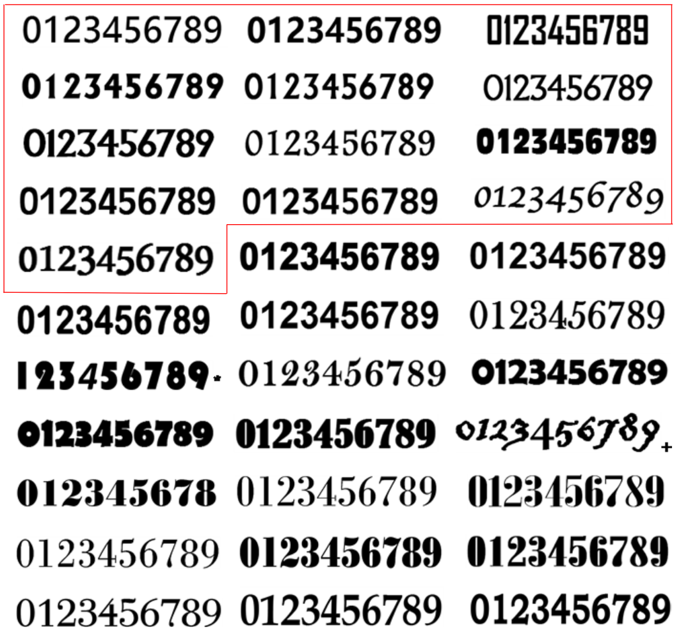
Fig. 10 A two-stage recognition approach of numerals in which the "proc" function computes the Freeman's chaincode illustrated by Fig. 9

Data collection

We have collected a database of numeral images using 33 fonts. Therefore, the total number of numeral images is 330 (33 × 10). This set is heterogeneous in the sense that it contains similar and dissimilar font shape types (refer to Fig. 11). It is worth underscoring that since digit 4 of both Blackadder ITC and Bauhaus 93 fonts do not include hole; therefore, they were respectively replaced by digit 4 of Calibri font (sign + in Fig. 11a) and digit 4 of Bauhaus 93 font was replaced by digit 4 of Constantia font (sign * in Fig. 11a).

Font diversity ratio

The diversity ratio of fonts represents the level of similarity of fonts contained in the entire database that is used for numeral recognition. This criterion requires a classification scheme of all types of fonts. Table 2 illustrates this taxonomy extracted from reference (https://www.freecodecamp.org/news/typography-type-families-classifications-and-combining-typefaces). The proposed methodology is based on the fonts list depicted by Fig. 11. Table 3 shows the mapping of each font of Fig. 11 to its type numbered from 1 to 8 in Table 2.



(a)

Fonts

@Microsoft JhengHei UI	@Microsoft YaHei UI	Agency FB
Albertus Extra Bold	Albertus Medium	Albertus MT Lt
Albertus MT	Andalus	Antique Olive Compact
Antique Olive Roman	Antique Olive	Apple Chancery
Arabic Typesetting	Arial Black	Arial Narrow
Arial Rounded MT Bold	Arial	Baskerville Old Face
Bauhaus 93	Bell MT	Berlin Sans FB Demi
Berlin Sans FB	Bernard MT Condensed	Blackadder ITC
Bodoni MT Black	Bodoni MT Condensed	Bodoni MT Poster Compressed
Bodoni MT	Bodoni Poster	Bodoni Poster Compressed
Bodoni	Book Antiqua	Bookman Old Style

(b)

Fig. 11 **a** The set of all font types used in the experimental phase. **b** List of font names

From this table, one can compute a diversity ratio δ (representing a dissimilarity level across fonts) which is expressed as follows:

$$\delta = \frac{DT}{AT} \times 100\% \tag{4}$$

where DT denotes the number of different types of fonts used, and AT is the total number of types. One can easily compute the diversity ratio assigned to Table 2. This ratio is in this case equal to 7/8 = 87.5%. This value corresponds to a low similarity of fonts.

Table 2 Font type classification

Number	Type	Font
1	Serifs/slab	{Garamond, Baskerville, Bodoni, Rockwell,…}
2	Sans Serif	{Gill, Helvetica,…}
3	Script	{Bistro Script, Shelly, Minstral,…}
4	Blackletter (Gothic)	{Bold Script, Fraktur,…}
5	Decorative	{Headlines, Posters,…}
6	Humanist	{Garamond, Sabon, Centaur,…}
7	Transitional	{Baskerville, Times Roman, Helvetica-Transitional Era}
8	Modern	{Bodoni, Futura,…}

Table 3 Mapping of fonts used in the proposed approach to their types

Font	Type
@Microsoft JhengHei UI	8
Albertus extra bold	1
Albertus MT	1
Antique olive roman	2
Arabic typesetting	5
Arial rounded MT bold	2
Bauhaus 93	8
Berlin Sans FB	2
Bodoni MT black	8
Bodoni MT	8
Bodoni	8
@Microsoft YaHei UI	8
Albertus medium	1
Andalus	5
Antique olive	2
Arial black	2
Arial	2
Bell MT	1
Bernard MT condensed	5
Bodoni MT condensed	8
Bodoni poster	5
Book antiqua	6
Agency FB	2
Albertus MT Lt	1
Antique olive compact	2
Apple chancery	4
Arial narrow	2
Baskerville old face	7
Berlin Sans FB Demi	2
Blackadder ITC	1
Bodoni MT poster compressed	8
Bodoni poster compressed	5
Bookman old style	1

Two-stage numeral recognition

To validate our methodology during this experimental phase, we have performed an automatic clustering of the 330 numeral images into 3 clusters based on topological features. Furthermore, we have considered only 13 fonts out of the 33. These 13 fonts are selected arbitrarily. Hence, 130 (13 × 10) images are collected. We extracted topological features on these 130 images and performed a clustering based on these features with a VR simplicial complex threshold set to $d = 3\sqrt{2}$. This optimal value was determined through a finetuning procedure (minimum recognition error rate achieved) based on the rescaling parameters M and N ($M = N = 128$). We relied on the JAVAPLEX platform [23] to perform this operation. We finally conducted a classification based on DTW procedure. The results of the first-stage recognition are illustrated by Table 4. The accuracy reported is 100%. This high performance shows that TDA is very accurate during the initial clustering of numeral images. We subsequently partitioned the database in order to create two cases: (i) the fonts of the test samples are contained in the codebook (set of reference templates), and (ii) the fonts of the test samples are not part of the codebook. We now provide some details about these two settings.

Case where testing fonts are contained in the reference codebook In case (1), we have designed two different scenarios: In the first scenario, only 13 fonts out of 33 have been considered (fonts delimited by the red region in Fig. 11a and the shaded region in Fig. 11b). Therefore, the total amount of images is equal to 130 = 13 × 10 numeral images with a fixed numeral font size set to 350. This set of images was designed as reference templates to form the codebook. To form a test set with 200 images; we have considered only the first ten fonts of Fig. 9a counted line by line from left to right. Only two font sizes were considered, the first one is 250 and the second one is 450. Therefore, the tally is 10 × 10 from the 250-font size and 10 × 10 from the 450-font size, which is in total equal to 200 images. Furthermore, one can underscore that in this scenario, the fonts adopted in the codebook are similar in shape to the fonts used in the test set. The only difference between the reference set and the testing is the size. Tables 5 and 6 depict the confusion matrices obtained in this scenario without the incorporation of TDA and with its incorporation, respectively.

In the second scenario, to build the test set, we considered all 13 fonts from which we generated 130 images with font size set to 450 and 130 images with font size equal 250, which produces a total of 260 images. The codebook reference templates set is left unchanged. The confusion matrices obtained in this scenario without the incorporation of TDA and with its incorporation are depicted by both Tables 7 and 8, respectively.

Table 4 Numeral clustering using only TDA, accuracy = 100%

		Predicted labels		
		C_1	C_2	C_3
True labels	C_1	165	0	0
	C_2	0	33	0
	C_3	0	0	132

Table 5 Confusion matrix using DTW without TDA clustering, accuracy = 93.5% (case 1, scenario 1)

		Predicted labels									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True labels	**0**	**20**	0	0	0	0	0	0	0	0	0
	1	0	**18**	0	0	2	0	0	0	0	0
	2	0	0	**16**	0	4	0	0	0	0	0
	3	0	0	0	**16**	0	0	0	0	0	4
	4	0	0	0	0	**20**	0	0	0	0	0
	5	0	0	0	0	2	**18**	0	0	0	0
	6	0	0	0	0	0	0	**20**	0	0	0
	7	0	0	0	0	0	0	0	**20**	0	0
	8	0	0	0	0	0	0	1	0	**19**	0
	9	0	0	0	0	0	0	0	0	0	**20**

Table 6 Confusion matrix using DTW with TDA clustering, accuracy = 95% (case 1, scenario 1)

		Predicted labels									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True labels	**0**	**20**	0	0	0	0	0	0	0	0	0
	1	0	**20**	0	0	0	0	0	0	0	0
	2	0	4	**16**	0	0	0	0	0	0	0
	3	0	0	0	**16**	0	0	0	4	0	0
	4	0	0	0	0	**20**	0	0	0	0	0
	5	0	0	2	0	0	**18**	0	0	0	0
	6	0	0	0	0	0	0	**20**	0	0	0
	7	0	0	0	0	0	0	0	**20**	0	0
	8	0	0	0	0	0	0	0	0	**20**	0
	9	0	0	0	0	0	0	0	0	0	**20**

Case where testing fonts are not part of the reference codebook In case (2), like in the previous scenario, we have chosen the same 13 fonts out of the 33 as a template set. The test set contains 10 numerals from the 20 remaining fonts. Consequently, the size of the test set adds up to 200 images. In order to assess the contribution of TDA alone, we have classified the numerals captured by their chaincodes using only DTW pattern matching technique (no prior clustering was performed). The class decision adopted in this case is illustrated via Eq. (3), with the size of Codebook equals to 130 images. Table 9 depicts the confusion matrix as well as the accuracy obtained which is equal to 85%. Finally, Table 10 depicts the confusion matrix using both TDA and DTW. The accuracy in this latter experiment has reached 92%. This improvement in accuracy is explained mainly by the contribution of TDA.

In fact, TDA has removed the confusion between several pairs of numerals such as 1 and 4 that are non-equivalent topologically. One can notice for example that the true positive rate of class 1 was 40% without TDA and has been elevated to 70% with the contribution of TDA. Likewise, the true positive rate of class 8 was 85% without TDA, and has reached 100% with the contribution of TDA. Furthermore, the same improvement

Table 7 Confusion matrix using only DTW, accuracy = 91.92% (case 1, scenario 1)

		Predicted labels									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True labels	**0**	**26**	0	0	0	0	0	0	0	0	0
	1	0	**20**	2	0	4	0	0	0	0	0
	2	0	2	**20**	0	4	0	0	0	0	0
	3	0	0	0	**22**	0	0	0	0	0	4
	4	0	0	0	0	**26**	0	0	0	0	0
	5	0	0	2	0	0	**24**	0	0	0	0
	6	0	0	0	0	0	0	**26**	0	0	0
	7	0	0	0	0	0	0	0	**26**	0	0
	8	0	0	0	0	0	0	1	0	**25**	0
	9	0	0	0	0	0	0	0	0	2	**24**

Table 8 Confusion matrix using DTW with TDA clustering, accuracy = 93.85% (case 1, scenario 2)

		Predicted labels									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True labels	**0**	**26**	0	0	0	0	0	0	0	0	0
	1	0	**24**	2	0	0	0	0	0	0	0
	2	0	6	**20**	0	0	0	0	0	0	0
	3	0	0	0	**22**	0	0	0	4	0	0
	4	0	0	0	0	**26**	0	0	0	0	0
	5	0	0	2	0	0	**24**	0	0	0	0
	6	0	0	0	0	0	0	**26**	0	0	0
	7	0	0	0	0	0	0	0	**26**	0	0
	8	0	0	0	0	0	0	0	0	**26**	0
	9	0	0	0	0	0	0	2	0	2	**24**

Table 9 Confusion matrix using DTW without TDA clustering, accuracy = 85% (case 2)

		Predicted labels									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True labels	**0**	**20**	0	0	0	0	0	0	0	0	0
	1	0	**8**	1	0	10	0	0	0	1	0
	2	0	0	**16**	0	1	0	0	0	1	2
	3	0	0	0	**18**	0	0	0	0	2	0
	4	0	0	0	0	**20**	0	0	0	0	0
	5	0	0	0	0	0	**18**	2	0	0	0
	6	0	0	0	0	0	0	**20**	0	0	0
	7	0	0	0	0	2	0	1	**16**	0	1
	8	3	0	0	0	0	0	0	0	**17**	0
	9	2	0	0	0	0	0	0	0	1	**17**

happened between the following pairs of numerals (7,4) and (7,9). TDA has removed these confusions. Overall, the relative accuracy improvement using TDA is 8.23%. Since the topological properties (Betti numbers) of all numerals are known beforehand, and

Table 10 Confusion matrix using DTW with TDA clustering, accuracy = 92% (case 2)

		Predicted labels									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True labels	**0**	**20**	0	0	0	0	0	0	0	0	0
	1	0	**14**	5	0	0	1	0	0	0	0
	2	0	2	**17**	0	0	1	0	0	0	0
	3	0	0	1	**19**	0	0	0	0	0	0
	4	0	0	0	0	**20**	0	0	0	0	0
	5	0	0	2	0	0	**18**	0	0	0	0
	6	0	0	0	0	0	0	**20**	0	0	0
	7	0	0	1	0	0	0	0	**19**	0	0
	8	0	0	0	0	0	0	0	0	**20**	0
	9	2	0	0	0	0	0	1	0	0	**17**

the accuracy from Table 4 is 100%, one can state that the topological information from a sample of points is compatible with the one conveyed by the null model.

Performance comparison and assessment of the proposed approach

We cover a comparative task between our approach and some major state of the arts techniques.

Comparison of the proposed approach with different scenarios

The task of improving recognition of confusing Arabic machine-printed numerals depends on the contribution of TDA. Table 4 has shown that TDA is capable to categorize perfectly (with 100% accuracy) numerals into three clusters. This contribution allows to subsequently classifying numerals from these three clusters only. However, one has to distinguish between the two cases: (i) testing fonts are part of the reference Codebook and (ii) testing fonts are not part of the reference Codebook. In the first case, the use of TDA has improved the accuracy from 93.5 to 95% (refer to the confusion matrices of Tables 5 and 6). Likewise, the accuracy went up from 91.92 to 93.85% when the 13 fonts are used (refer to the confusion matrices of Tables 7 and 8). However, in the second case (testing fonts are not part of the reference Codebook), the accuracy increased drastically from 85 to 92% when TDA is applied (refer to the confusion matrices of Tables 9 and 10). This accuracy rate achieved represents an achievement since (i) the classifier was tested on numerals with unknown fonts, and (ii) the database used is relatively small and the classifier does not involve any large-scale training.

Comparison with some major states of the art approaches

We compare our methodology with some papers from the literature that use TDA and others that are more general:

TDA approaches In this section, we highlight some applications of TDA and show how they differ from our approach (refer to Table 11).

General approaches We have compared our approach with major methodologies proposed in the literature. Table 12 shows the size of training and testing sets, the number of fonts involved, the visual similarity between fonts, and the average accuracy for each technique deployed for the recognition task. This table also indicates whether all testing fonts have been included in the training set. Table 13 depicts the feature extraction techniques and the classifiers used by the same contributors. The comparison task is undertaken by taking into account three criteria, which are the following:

a. The size of the training set (or Codebook for template matching) and testing sets used.

Table 11 Comparison with some major TDA approaches

Reference number	Contribution	How they differ with our approach
[18]	Presentation of algebraic functions as a ring on persistence bar codes: An illustration on how to recognize handwritten digit using conventional classifiers based on the extraction of topological features from barcodes was provided	The technique proposed relies on four selected features (polynomial coordinates) that are computed from the two dimensions of the barcode associated to a handwritten numeral. Once a feature vector that gather all this information is extracted, the authors invoked traditional shallow classifiers such as SVM for handwritten numeral recognition. However, our work exploits TDA to first cluster numerals based on their topological features and extract chjaincodes from the bit-mapped versions of these numerals. Since a chain code is a time-series sequence, and our context is small-size dataset, we have reasonably applied dynamic time warping for numerals recognition
[19]	Introduction of a methodology that extracts tropical coordinates from a barcode. This work has been compared to the one of reference [18]. Instead of using polynomial coordinates as features, the author applied max-plus type coordinates extracted from barcodes. A support vector classifier is subsequently invoked for recognition of handwritten numerals	The same remark as in [18] applies to this reference
[30]	Introduction of a Python library that integrates high-performance TDA with machine learning	This library allows extraction of persistent diagram of objects. An example of the extraction of persistent diagrams from a bitmap handwritten numeral is provided. It does not propose a recognition scheme for Arabic machine-printed numerals
[31]	Use of TDA persistent diagram to extract relevant features. The authors subsequently invokes a machine learning paradigm based on random forest for handwritten digit recognition	Our approach is different since it is based on a prior TDA clustering technique of machine-printed numerals. Subsequently, a chaincode representation is applied to numerals pertaining to each cluster formed. Furthermore, a dynamic time warping procedure is invoked for the purpose of numerals recognition. Our proposed work does not use any learning scheme
[32]	The author's objective was to study the filtration sensitivity with respect to noise added to the handwritten digit image. This sensitivity measure is depicted by the performance of SVM classifier. The kernel of this classifier is based on the Wasserstein metric between two persistent diagrams	Our objective is far different since we are not analyzing the sensitivity of TDA filtrations to noise in the input images. It is worth underscoring that the noise we are addressing emanates from the thinning procedure

Table 12 Average accuracies of some state of the art methodologies

Ref.	**Database size (# of images)**		**# of fonts**	**Font diversity ratio**	**All fonts are included during training**	**Avg. acc. (%)**
	Training size	**Testing size**				
[2]	6240		**-**	Low	Yes	99.8
	Resubstitution (all sample are used for training and testing)					
[3]	2000		-	Low	**-**	99
	500	1500				
[4]	3200		16	Low	Yes	99
	160	3040				
[5]	531		-	-	-	95
Exp 1	400	131				
[5]	531					93.1
Exp 2	400	131				
[6]	70		7	Medium	Yes	74
	-	70				
Our study	330		13	Medium	Yes	**95**
Exp 1	130	200				
Our study	390		13	Medium	Yes	**93.8**
Exp 2	130	260				
Our study	330		33	High	No	**92**
Exp 3	130	200				

Table 13 Feature extraction and classification technique depicted by Table 10

Ref.	**Features**	**Classification schemes**
[2]	Hue moment and east cavity	K-nearest neighbors *(*kNN*)*
[3]	Correlation factor	-
[4]	Directional density vector	Decision tree
[5] Exp 1	Chaincode	Neural network
[5] Exp 2	Chaincode	Naïve Bayes
[6]	Statistical analysis of nodes	Fuzzy logic
Our study	Chaincode	Two stage: TDA + DTW

b. The size of training samples (or samples from the Codebook) contained in the testing sets.
c. The diversity ratio assigned to the set of fonts used to build the database.

- The authors in reference [2] indicated that all the training was used as testing set under the "resubstitution" operation. Their fonts diversity ratio is low (i.e., they have used similar fonts), and the size of their database is very large, which is far from the small database size setting conveyed by our approach. Furthermore, it is important to outline that our approach does not require training.
- The authors in reference [3] did not provide much information about the number of fonts used and whether training samples are contained in the testing set. How-

ever, they claimed that the font diversity ratio is low, which may explain the high accuracy obtained.

- The authors in reference [4] reported that their training set was reused during testing, and their font diversity ratio is low.
- The authors in [5] did not disclose any detailed information about the context of their experiment. However, the size of their database is close to our database size, which makes these two approaches comparable.
- The authors in reference [6] relied on a small database size, and their font diversity ratio is medium but their performance remains low.
- Finally, we obtained a series of performance, which are 95%, 93.85%, and 92%, depending on the database size, the font diversity ratios, and whether training is part of testing.

Pros and cons of TDA

The following table provides an insight into whether TDA is worth to be applied to the task of recognition of machine-printed Arabic numerals. Table 14 depicts the advantages and limitations of TDA in this specific application:

Conclusions

We have explored TDA in order to investigate whether this mathematical paradigm is capable to extract relevant information that improves recognition of Arabic machine-printed numerals. This exploration is performed through clustering of numeral images based on their topological invariants. Hence, we have reduced the number of classes through clustering and consequently lowered the confusion between numerals. We

Table 14 Advantages and limitations of TDA

Pros	Cons
• TDA provides a general framework for analyzing data, with the advantages of being capable of extracting information from large volumes of high-dimensional data, while being independent of the choice of metrics. Furthermore, TDA provides stability against noise generated from the thinning operation • Persistent homology is quite efficient at comparing images with a much smaller sample size than deep learning/CNNs would need. A pattern matching distance such as DTW is adequate in this small-size data set setting • Persistent homology can differentiate features like loops or voids, which are important in numeral recognition, and a TDA-based approach tends to perform well, even with very small small-size datasets • One of the advantage of topological analysis is low dimensional representation of higher dimensional connectivity • TDA performs effectively in clustering Arabic machine-printed numerals even when numeral fonts are dissimilar in shapes	• High computational cost of persistent homology in the case of large-size dataset and TDA requires a profound understanding of algebraic topology in order to be able to correctly exploit all algorithms generated from it • TDA relies only on closed shapes that are not degraded during the determination of Betti numbers: Any numeral whose contour is not perfectly closed due to some preprocessing will not be accounted for • Persistent homology covers a reduced set of topological invariants expressed via Betti numbers. Six of numerals out of 10 are not fully characterized by Betti numbers. However, some languages such as Arabic convey more topological information: in this languages TDA is more efficient • TDA as presented in this manuscript does not discriminate between directions of loops (difference between ellipses and circles for example). This discrimination is vital in several pattern recognition problems. For example the digit 0 has a round hole than the digit 4 whose hole is triangulized

have demonstrated that TDA performs precisely in clustering Arabic machine-printed numerals even when numeral fonts are dissimilar in shape. This dissimilarity is expressed via a font diversity ratio that we have developed as a criterion for performance comparison purposes. We have observed that the recognition accuracy using TDA and DTW successively is significantly higher than the one using only DTW. We have assessed the limitations and advantages of TDA in this numerals recognition task. Our future work will be focused on:

- Improving the second-stage classification by modifying DTW distance through the incorporation of a matrix that contains cost values proportional to the difference of two angular directions of symbols depicting a chaincode.
- Extending this work exploration to font-independent alphanumeric characters recognition.

Abbreviations

CL	Collective learning
DA	Data augmentation
DTW	Dynamic-time warping
HT	Homology theory
kNN	K-nearest neighbors
LPR	License plates recognition
ML	Machine learning
PHT	Persistent homology theory
TDA	Topological data analysis
TL	Transfer learning
USPS	US postal service
VR	Vietoris-Rips

Acknowledgements
We are grateful to the General Direction of Scientific Research (DGRSDT) for their continuous financial support in this research.

Authors' contributions
D. Bouchaffra designed the study and wrote the manuscript. F. Ykhlef performed the experiments, collected the data, provided critical feedback, and contributed to the writing of the manuscript. Both authors confirmed the results, and read and approved the final version of the manuscript.

Funding
This research work is supported by a grant from the General Direction of Scientific Research & Development (DGRSDT), under the number (DGRSDT-13), Algeria.

Availability of data and materials
The corresponding author can provide the datasets used during the current study upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 17 May 2023 Accepted: 18 December 2023
Published online: 13 January 2024

References

1. Kumar M, Jindal MK, Sharma RK, Jindal SR (2019) Character and numeral recognition for non-Indic and Indic scripts: a survey. Artif Intell Rev 52(4):2235–2261
2. Hassanpour H, Samadiani N, Akbarzadeh F (2017) A modified self-organizing map neural network to recognize multi-font printed persian numerals. Int J Eng IJE 30(11):1700–1706

3. Khedidja D, Hayet M (2019) Multiple classifiers and invariant features extraction for digit recognition. IJECE 11(1):41–52
4. Alqudah AT, Al-Zoubi HR, Al-Khassaweneh M (2012) Shift and scale invariant recognition of printed numerals. Abhath Al-Yarmouk Basic Sci Eng 21(1):41–49
5. Dhandra BV, Malemath VS, Mallikarjun H, Hegadi R (2006) Multi-font Numeral recognition without Thinning based on Directional Density of pixels. In: 2006 1st International Conference on Digital Information Management. IEEE, Bangalore, p 157–160
6. Radha R, Aparna RR (2014) Automatic extraction, segmentation and recognition of multi-font Indian Pincode. IJCVR 4(3):247–258
7. Salameh M, Salem AA (2016) hyper recognition techniques for English digits using statistical analysis of nodes and Fuzzy Logic for pattern recognition. Int J Multi Sci Eng 7(8):1–7
8. Wang Y, Lian Z (2020) Exploring font-independent features for scene text recognition. Proceedings of the 28th ACM International Conference on Multimedia. pp 1900–1920
9. Kundaikar T, Pawar JD (2020) Multi-font Devanagari Text Recognition Using LSTM Neural Networks. First International Conference on Sustainable Technologies for Computational Intelligence. Springer, Singapore, pp 495–506
10. Sharma R, Kaushik B, Gondhi N (2020) Character recognition using machine learning and deep learning-a survey. In: 2020 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, Pune, p 341–345
11. Stricker D (2019) Multi-font Printed Amharic Character Image Recognition: Deep Learning Techniques. Advances of Science and Technology: 6th EAI International Conference, ICAST 2018, Bahir Dar, Ethiopia, October 5–7, 2018, Proceedings, vol 274. Springer, Bahir Dar, p 322
12. Silva SM, Jung CR (2020) Real-time license plate detection and recognition using deep convolutional neural networks. J Vis Commun Image Represent 71:102773
13. Bouchaffra D, Tan J (2006) Structural hidden Markov models: An application to handwritten numeral recognition. Intelligent Data Analysis 10(1):67–79
14. Jha M, Kabra M, Jobanputra S, Sawant R (2019) Automation of cheque transaction using deep learning and optical character recognition. In: 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, Tirunelveli, p 309–312
15. Chowdhury AI, Rahman MS, Sakib N (2019) A study of multiple barcode detection from an image in business system. Int J Comput Appl 181(37):30–37
16. Savino P, Tonazzini A (2016) Digital restoration of ancient color manuscripts from geometrically misaligned recto-verso pairs. J Cult Herit 19:511–521
17. Bouchaffra D, Govindaraju V, Srihari SN (1999) Postprocessing of recognized strings using nonstationary Markovian models. IEEE Trans Pattern Anal Mach Intell 21(10):990–999
18. Adcock A, Carlsson E, Carlsson G (2016) The ring of algebraic functions on persistence bar codes. Homol Homotopy Appl 16(1):381–402
19. Kališnik S (2019) Tropical coordinates on the space of persistence barcodes. Found Comput Math 19(1):101–129
20. Choi HR, Kim T (2018) modified dynamic time warping based on direction similarity for fast gesture recognition. Math Probl Eng 2018:1–9
21. Edelsbrunner H, Harer JL (2010) Computational topology: An introduction. American Mathematical Society, Providence, Rhode Island
22. Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA (2017) A roadmap for the computation of persistent homology. EPJ Data Science 6(1):17
23. Adams H, Tausz A, Vejdemo-Johansson M (2014) JavaPlex: a research software package for persistent (co) homology. In: International Congress on Mathematical Software. Springer, Seoul, p 129–136
24. Pola FPB, Pola IRV (2019) Optimizing computational high-order schemes in finite volume simulations using unstructured mesh and topological data structures. Appl Math Comput 342:1–17
25. De Silva V, Gunnar EC Topological estimation using witness complexes. In: Symposium on Point Based Graphics. IEEE, Goslar, Germany, p 157–166
26. Lee T-C, Kashyap RL, Chu C-N (1994) Building skeleton models via 3-D medial surface/axis thinning algorithms. Comp Vision Graph Image Proc 56(6):462–478
27. Freeman H (1961) On the encoding of arbitrary geometric configurations. IRE Trans Elec Comput EC-10(2):260–268
28. Vintsyuk TK (1968) "Speech discrimination by dynamic programming", Cybernetics
29. Sakoe H, Chiba S (1978) "Dynamic programming algorithm optimization for spoken word recognition," IEEE Transactions on Acoustics, Speech and Signal Processing
30. Tauzin G, Lupo U, Pérez TL, Caorsi JB, Medina-Mardones M, Hess K (2021) giotto-tda: A topological data analysis toolkit for machine learning and data exploration. J Mach Learn Res 22(39):1–6
31. A Garin, G Tauzin (2019) A topological "reading" lesson: Classification of MNIST using TDA. In 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA). pp 1551–1556
32. Turkeš N, Nys R, Verdonck J, Latré S (2021) Noise robustness of persistent homology on greyscale images across filtrations and signatures. PloS One 16(9):e0257215

Publisher's Note