Article

Cubical Homology-Based Machine Learning: An Application in Image Classification

Seungho Choe [†] and Sheela Ramanna ^{*}

Department of Applied Computer Science, University of Winnipeg, Winnipeg, MB R3B 2E9, Canada; choe-s@webmail.uwinnipeg.ca

^{*} Correspondence: s.ramanna@uwinnipeg.ca

[†] This study is part of Seungho Choe's MSc Thesis of Cubical homology-based Image Classification—A Comparative Study, defended at the University of Winnipeg in 2021.

Abstract: Persistent homology is a powerful tool in topological data analysis (TDA) to compute, study, and encode efficiently multi-scale topological features and is being increasingly used in digital image classification. The topological features represent a number of connected components, cycles, and voids that describe the shape of data. Persistent homology extracts the birth and death of these topological features through a filtration process. The lifespan of these features can be represented using persistent diagrams (topological signatures). Cubical homology is a more efficient method for extracting topological features from a 2D image and uses a collection of cubes to compute the homology, which fits the digital image structure of grids. In this research, we propose a cubical homology-based algorithm for extracting topological features from 2D images to generate their topological signatures. Additionally, we propose a novel score measure, which measures the significance of each of the sub-simplices in terms of persistence. In addition, gray-level co-occurrence matrix (GLCM) and contrast limited adapting histogram equalization (CLAHE) are used as supplementary methods for extracting features. Supervised machine learning models are trained on selected image datasets to study the efficacy of the extracted topological features. Among the eight tested models with six published image datasets of varying pixel sizes, classes, and distributions, our experiments demonstrate that cubical homology-based machine learning with the deep residual network (ResNet 1D) and Light Gradient Boosting Machine (lightGBM) shows promise with the extracted topological features.

Keywords: cubical complex; cubical homology; image classification; deep learning; persistent homology

Citation: Choe, S.; Ramanna, S. Cubical Homology-Based Machine Learning: An Application in Image Classification. *Axioms* **2022**, *11*, 112. <https://doi.org/10.3390/axioms11030112>

Academic Editor: Oscar Humberto Montiel Ross

Received: 10 January 2022

Accepted: 24 February 2022

Published: 3 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The origin of topological data analysis (TDA) and persistent homology can be traced back to H. Edelsbrunner, D. Letscher, and A. Zomorodian [1]. More recently, TDA has emerged as a growing field in applied algebraic topology to infer relevant features for complex data [2]. One of the fundamental methods in computational topology is persistent homology [3,4], which is a powerful tool to compute, study, and encode efficiently multi-scale topological features of nested families of simplicial complexes and topological spaces [5]. Simplices are building blocks used to study the shape of data and a simplicial complex is its higher-level counterpart. The process of shape construction is commonly referred to as filtration [6]. There are many forms of filtrations and a good survey is presented in [7]. Persistent homology extracts the birth and death of topological features throughout a filtration built from a dataset [8]. In other words, persistent homology is a concise summary representation of topological features in data and is represented in a persistent diagram or barcode. This is important since it tracks changes and makes it possible to analyze data at multiple scales since the data structure associated with topological features is a multi-set, which makes learning harder. Persistent diagrams are then mapped into metric spaces with additional structures useful for machine learning tasks [9]. The application of TDA in

machine learning (also known as TDA pipeline) in several fields is well documented [2]. The TDA pipeline consists of using data (e.g., images, signals) as input and then filtration operations are applied to obtain persistence diagrams. Subsequently, ML classifiers such as support vector machines, tree classifiers, and neural networks are applied to the persistent diagrams.

The TDA pipeline is an emerging research area to discover descriptors useful for image and graph classification learning and in science—for example, quantifying differences between force networks [10] and analyzing polyatomic structures [11]. In [12], microvascular patterns in endoscopy images can be categorized as *regular* and *irregular*. Furthermore, there are three types of regular surface of the microvasculature: *oval*, *tubular*, and *villous*. In this paper, topological features were derived with persistence diagrams and the q -th norm of the p -th diagram is computed as

$$N_q = \left[\sum_{A \in \text{Dgm}_p(f)} \text{pers}(A)^q \right]^{\frac{1}{q}},$$

where $\text{Dgm}_p(f)$ denotes the p -th diagram of f and $\text{pers}(A)$ is the persistence of a point A in $\text{Dgm}_p(f)$. Since N_q is a norm of p -th Betti number with restriction (or threshold) s , it will obtain the p -th Betti number of $\mathbb{M}_s$, where $\mathbb{M}$ is the rectangle covered by pixels. Then, $\mathbb{M}$ is mapped to $\mathbb{R}$ by a signed distance function. A naive Bayesian learning method that combines the results of several Adaboost classifiers is then used to classify the images. The authors in [13] introduce a multi-scale kernel for persistence diagrams that is based on scale space theory [14]. The focus is on the stability of persistent homology since any occurrence of small changes in the input affects both the 1-Wasserstein distance and persistent diagrams. Experiments on two benchmark datasets for 3D shape classification/retrieval and texture recognition are discussed.

Vector summaries of persistence diagrams is a technique that transforms a persistence diagram into vectors and summarizes a function by its minimum through a pooling technique. The authors in [15] present a novel pooling within the bag-of-words approach that shows a significant improvement in shape classification and recognition problems with the Non-Rigid 3D Human Models SHREC 2014 dataset.

The topological and geometric structures underlying data are often represented as point clouds. In [16], the RGB intensity values of each pixel of an image are mapped to the point cloud $P \in \mathbb{R}^5$ and then a feature vector is derived. Computing and arranging the persistence of point cloud data by descending order makes it possible to understand the persistence of features. The extracted topological features and the traditional image processing features are used in both vector-based supervised classification and deep network-based classification experiments on the CIFAR-10 image dataset. More recently, multi-class classification of point cloud datasets was discussed in [17]. In [8], a random forest classifier was used to classify the well-known MNIST image dataset using the voxel structure to obtain topological features.

In [18], persistent diagrams were used with neural network classifiers in graph classification problems. In TDA, Betti numbers represent counts of the number of homology groups, such as points, cycles, and so on. In [19], the similarity of the brain networks of twins is measured using Betti numbers. In [20,21], persistent barcodes were used to visualize brain activation patterns in resting-state functional magnetic resonance imaging (rs-fMRI) video frames. The authors used a geometric Betti number that counts the total number of connected cycles forming a vortex (nested, usually non-concentric, connected cycles) derived from the triangulation of brain activation regions.

The success of deep learning [22] in computer vision problems has led to its use in deep networks that can handle barcodes [23]. Hofer et al. used a persistence diagram as a topological signature and computed a parametrized projection from the persistence diagram, and then leveraged it during the training of the network. The output of this process is stable when using the 1-Wasserstein distance. Classification of 2D object shapes and

social network graphs was successfully demonstrated by the authors. In [24], the authors apply topological data analysis to the classification of time series data. A 1D convolutional neural network is used, where the input data are a *Betti sequence*. Since machine learning models rely on accurate feature representations, multi-scale representations of features are becoming increasingly important in applications involving computer vision and image analysis. Persistence homology is able to bridge the gap between geometry and topology, and persistent homology-based machine learning models have been used in various areas, including image classification and analysis [25].

However, it has been shown that the implementations of persistent homology (of simplicial complexes) are inefficient for computer vision since it requires excessive computational resources [26] due to the formulations based on triangulations. To mitigate the problem of complexity, cubical homology was introduced, which allows the direct application of its structure [27,28]. Simply, cubical homology uses a collection of cubes to compute the homology, which fits the digital image structure of grids. Since there is neither skeletonization nor triangulation in the computation of cubical homology, it has advantages in the fast segmentation of images for extracting features. This feature of cubical homology is the motivation for this work in exploring the extraction of topological features from 2D images using this method.

The focus in this work is twofold: (i) on the extraction of topological features from 2D images with varying pixel sizes, classes, and distributions using cubical homology; (ii) to study the effect of extracted 1D topological features in a supervised learning context using well-known machine learning models trained on selected image datasets. The work presented in this paper is based on the thesis by Choe [29]. Figure 1 illustrates our proposed approach. Steps 2.1 and 2.2 form the core of this paper, namely the generation of 1D topological signatures using a novel *score* that is proposed by this study. This score allows us to filter out low persistence features (or noise). Our contribution is as follows: (i) we propose a cubical homology-based algorithm for extracting topological features from 2D images to generate their topological signatures; (ii) we propose a score, which is used as a measure of the significance of the subcomplex calculated from the persistence diagram. Additionally, we use gray-level co-occurrence matrix (GLCM) and contrast limited adapting histogram equalization (CLAHE) for obtaining additional image features, in an effort to improve the classification performance, and (iii) we discuss the results of our supervised learning experiments of eight well-known machine learning models trained on six different published image datasets using the extracted topological features.

Figure 1. Classification pipeline.

The paper is organized as follows. Section 2 gives basic definitions for simplicial, cubical, and persistent homology used in this work. Section 3.2 illustrates the feature engineering process, including the extraction of topological and other subsidiary features. Section 3.1 introduces the benchmark image datasets used in this work. Section 4 gives the results and analysis of using eight machine learning models trained on each dataset. Finally, Section 5 gives the conclusions of this study.

2. Basic Definitions

We recall some basic definitions of the concepts used in this work. A simplicial complex is a space or an object that is built from a union of points, edges, triangles, tetrahedra, and higher-dimensional polytopes. Homology theory is in the domain of algebraic topology related to the connectivity in multi-dimensional shapes [26].

2.1. Simplicial Homology

Graphs are mathematical structures used to study pairwise relationships between objects and entities.

Definition 1. A graph is a pair of sets, $G = (V, E)$, where V is the set of vertices (or nodes) and E is a set of edges.

Let S be a subset of a group G . Then, the subgroup generated by S , denoted $\langle S \rangle$, is the subgroup of all elements of G that can be expressed as the finite operation of elements in S and their inverses. For example, the set of all integers, $\mathbb{Z}$, can be expressed by the operation of elements $\{1\}$ so $\mathbb{Z}$ is the subgroup generated by $\{1\}$.

Definition 2. A rank of a group G is the size of the smallest subset that generates G .

For instance, since $\mathbb{Z}$ is the subgroup generated by $\{1\}$, $\text{rank}(\mathbb{Z})=1$.

Definition 3. A simplex complex on a set V is a family of arbitrary cardinality subsets of V closed under the subset operation, which means that, if a set S is in the family, all subsets of S are also in the family. An element of the family is called a simplex or face.

Definition 4. Moreover, p -simplex can be defined to the convex hull of $p + 1$ affinely independent points $x_0, x_1, \dots, x_p \in \mathbb{R}^d$.

For example, in a graph, 0-simplex is a point, 1-simplex is an edge, 2-simplex is a triangle, 3-simplex is a tetrahedron, and so on (see Figure 2).

Figure 2. Examples of p -simplex for $p = 0, 1, 2, 3$ in tetrahedron. A 0-simplex is a point, a 1-simplex is an edge with a convex hull of two points, a 2-simplex is a triangle with a convex hull of three distinct points, and a 3-simplex is a tetrahedron with a convex hull of four points [30].

Chain, Boundary, and Cycle

To extend simplicial homology to persistent homology, the notion of *chain*, *boundary*, and *cycle* is necessary [31].

Definition 5. A p -chain is a subset of p -simplices in a simplicial complex K . Assume that K is a triangle. Then, a 1-chain is a subset of 1-simplices—in other words, a subset of the three edges.

Definition 6. A boundary, generally denoted ∂ , of a p -simplex is the set of $(p - 1)$ -simplices' faces.

For example, a triangle is a 2-simplex, so the boundary of a triangle is a set of 1-simplices which are the edges. Therefore, the boundary of the triangle is the three edges.

Definition 7. A cycle can be defined using the definitions of chain and boundary. A p -cycle c is a p -chain with an empty boundary. More simply, it is a path where the starting point and destination point are the same.

2.2. Cubical Homology

Cubical homology [27] is efficient since it allows the direct use of the cubical structure of the image, whereas simplicial theory requires increasing the complexity of data. While the simplicial homology is built with the triangle and its higher-dimensional structure, such as a tetrahedron, cubical homology consists of *cubes*. In cubical homology, each cube has a unit size and the n -cube represents its dimension. For example, 0-cubes are points, 1-cubes are lines with unit length, 2-cubes are unit squares, and so on [27,32,33].

Definition 8. Here, 0-cubes can be defined as an interval,

$$[m] = [m, m], m \in \mathbb{Z}, \quad (1)$$

which generate subsets $I \in \mathbb{R}$, such that

$$I = [m, m + 1], m \in \mathbb{Z}. \quad (2)$$

Therefore, I is called a 1-cube, or *elementary interval*.

Definition 9. An n -cube can be expressed as a product of elementary intervals as

$$Q = I_1 \times I_2 \times \cdots \times I_n \subseteq \mathbb{R}^n, \quad (3)$$

where Q indicates that n -cube $I_i (i = 1, 2, \dots, n)$ is an elementary interval.

A d -dimensional image is a map $\mathcal{I} : I \subseteq \mathbb{Z}^d \rightarrow \mathbb{R}$.

Definition 10. A pixel can be defined as an element $v \in I$, where $d = 2$. If $d > 2$, v is called a voxel.

Definition 11. Let $\mathcal{I}(v)$ be the intensity or grayscale value. Moreover, in the case of binary images, we consider a map $\mathcal{B} : I \subseteq \mathbb{Z}^d \rightarrow \{0, 1\}$.

A voxel is represented by a d -cube and, with all of its faces added, we have

$$\mathcal{I}'(\sigma) := \min_{\sigma \text{ face of } \tau} \mathcal{I}(\tau). \quad (4)$$

Let K be the cubical complex built from the image I , and let

$$K_i := \{\sigma \in K | \mathcal{I}'(\sigma) \leq i\}, \quad (5)$$

be the i -th sublevel set of K . Then, the set $\{K_i\}_{i \in \text{Im}(I)}$ defines a filtration of the cubical complexes. Thus, the pipeline to filtration from an image with a cubical complex is as follows:

Image $\rightarrow$ Cubical complex $\rightarrow$ Sublevel sets $\rightarrow$ Filtration

Moreover, *chain*, *boundary*, and *cycle* in cubical homology can be defined in the same manner as in Section 2.1.

2.3. Persistent Homology

In topology, there are subcomplexes of complex K and *cubes* are created (*birth*) and destroyed (*death*) by filtration. Assume that K^i ($0 \leq i \leq n$, $i \in \mathbb{Z}$) is a subcomplex of filtered complex K such that

$$\emptyset \subseteq K^0 \subseteq K^1 \subseteq \cdots \subseteq K^n = K,$$

and $\mathcal{Z}_k^i, \mathcal{B}_k^i$ are its corresponding cycle group and boundary group.

Definition 12. The k th homology group [1] can be defined as

$$\mathcal{H}_k = \mathcal{Z}_k / \mathcal{B}_k. \quad (6)$$

Definition 13. The p -persistent k th homology group of K^i [1] can be defined as

$$\mathcal{H}_k^{i,p} = \mathcal{Z}_k^i / \mathcal{B}_k^{i+p} \cap \mathcal{Z}_k^i. \quad (7)$$

Definition 14. A persistence is a lifetime of these attributes based on the filtration method used [1].

One can plot the birth and death times of the topological features as a barcode, also known as a *persistence barcode*, shown in Figure 3. This diagram graphically represents the topological signature of the data. Illustration of persistence is useful when detecting a change in terms of topology and geometry, which plays a crucial role in supervised machine learning [34].

Figure 3. An example of persistent homology for grayscale image. (a) A given image, (b) a matrix of gray level of given image, (c) the filtered cubical complex of the image, (d) the persistence barcode according to (c).

3. Materials and Methods

3.1. Image Datasets

In this section, we give a brief description of the six published image datasets used in this work. Datasets used for benchmarking were collected from various sources that include *Mendeley Data* (<https://data.mendeley.com/> accessed on 7 January 2022), *Tensorflow dataset* (<https://www.tensorflow.org/datasets> accessed on 7 January 2022), and *Kaggle competition* (<https://www.kaggle.com/competitions> accessed on 7 January 2022). The *concrete crack images* dataset [35] contains a total of 40,000 images, where each image consists of 227×227 pixels. These images were collected from the METU campus building and consist of two classes: 20,000 images where there are no cracks in the concrete (positive) and 20,000 images of concrete that is cracked (negative). A crack on an outer wall occurs with the passage of time or due to natural aging. It is important to detect these cracks in terms of evaluating and predicting the structural deterioration and reliability of buildings. Samples of the two types of images are shown in Figure 4.

The *APTOS blindness detection* dataset [36] is a set of retina images taken by fundus photography for detecting and preventing diabetic retinopathy from causing blindness

(<https://www.kaggle.com/c/aptos2019-blindness-detection/overview> accessed on 7 January 2022). This dataset has 3662 images and consists of 1805 images diagnosed as non-diabetic (labeled as 0) retinopathy and 1857 images diagnosed as diabetic retinopathy, as shown in Figure 5. Figure 6 shows the distribution of examples in the four classes using a severity range from 1 to 4 with the following interpretation: 1: Mild, 2: Moderate, 3: Severe, 4: Proliferative DR.

Figure 4. Sample images of the concrete crack dataset.

Figure 5. Sample images of APTOS dataset. (a) is a picture of Non-diabetic retionopathy which is ordinary case and (b) is a picture of diabetic retinopathy which cas cause blindness.

Figure 6. Data distribution for APTOS 2019 blindness detection dataset. This dataset can be classified as non-diabetic and diabetic. Around 50% of images are categorized as non-diabetic retinopathy (label 0) and diabetic retinopathy is subdivided according to the severity range from 1 to 4.

The *pest classification in mango farms* dataset [37] is a collection of 46,500 images of mango leaves affected by 15 different types of pests and one normal (unaffected) mango leaf, as shown in Figure 7. Some of these pests can be detected visually. Figure 8 shows the data distribution of examples in the 15 classes of pests and one normal class.

Figure 7. Sample images of pest classification in mango farms.

Figure 8. Data distribution for pest classification in mango farms dataset. Labels are assigned in alphabetical order.

The *Indian fruits* dataset [38] contains 23,848 images that cover five popular fruits in India: apple, orange, mango, pomegranate, and tomato. This dataset includes variations of each fruit, resulting in 40 classes. This dataset was already separated into training and testing sets by the original publishers of the dataset, as shown in Figure 9. Note that this dataset has an imbalanced class distribution.

Figure 9. Data distribution for the Indian fruits dataset. These data are already split by the original publishers of the dataset by the ratio of 9 to 1. Labels are assigned in alphabetical order.

The *colorectal histology* dataset [39] contains 5000 histological images of different tissue types of colorectal cancer. It consists of 8 classes of tissue types with 625 images for each class, as shown in Figure 10.

Figure 10. Example of colorectal cancer histology. (a) Tumor epithelium, (b) simple stroma, (c) complex stroma, (d) immune cell conglomerates, (e) debris and mucus, (f) mucosal glands, (g) adipose tissue, (h) background.

The *Fashion MNIST* dataset [40] is a collection of 60,000 training images of fashion products, as shown in Figure 11. It consists of 28×28 grayscale images of products from 10 classes. Since the dataset contains an equal number of images for each class, there are 6000 test images in each class, resulting in a balanced dataset.

Figure 11. Example of the Fashion MNIST dataset.

Table 1 gives the dataset characteristics in terms of the various image datasets used in this work. Moreover, we provide the preprocessing time per image. For example, the feature extraction time for the concrete dataset was 5 h 12 min.

Table 1. Dataset details with preprocessing times.

Dataset	Size	Num of Classes	Pixel Dimension	Balanced	Time in Sec/Image
Concrete ¹	40,000	2	227 × 227	Yes	0.4713
Mangopest ²	46,000	16	from 500 × 333 to 1280 × 853	No	0.5394
Indian fruits ³	23,848	40	100 × 100	No	0.4422
Fashion MNIST ⁴	60,000	10	28 × 28	Yes	0.4297
APTOS ⁵	3662	5	227 × 227	No	0.5393
Colorectal histology ⁶	5000	8	150 × 150	Yes	0.3218

¹ Çağlar Fırat Özgenel, 23 July 2019, <https://data.mendeley.com/datasets/5y9wdsg2zt/2/>; ² Kusrini Kusrini et al., accessed on 27 February 2020, <https://data.mendeley.com/datasets/94jf97jzc8/1/>; ³ prabira Kumar sethy, accessed on 12 June 2020, <https://data.mendeley.com/datasets/bg3js4z2xt/1/>; ⁴ Han Xiao et al., accessed on 28 August 2017, <https://github.com/zalandoresearch/fashion-mnist>; ⁵ Asia Pacific Tele-Ophthalmology Society, accessed on 27 June 2019, <https://www.kaggle.com/c/aptos2019-blindness-detection/overview>; ⁶ Kather, Jakob Nikolas et al., 26 May 2016, <https://zenodo.org/record/53169#.XGZemKwzbmG>.

3.2. Methods—Feature Engineering

In this section, we describe the feature engineering process. The main purpose of this process is to obtain a 1-dimensional array from each image in the dataset. Each point from the persistence diagram plays a significant role in the extraction of the topological features. Moreover, the *gray-level co-occurrence matrix* (GLCM) supports these topological features as additional signatures. Because every image dataset is not identical in size and some images have very high resolution, resizing every image to 200 × 200 and converting them to grayscale guarantees a relatively constant duration of extraction (approximately 4 s) regardless of its original size.

Algorithm 1 gives the method for extracting topological features from a dataset. In this algorithm, β_0 and β_1 are Betti numbers derived from Equation (6), where the dimension of i th homology is called the i th Betti number of K . β_0 gives the number of connected components and β_1 gives the number of holes. Betti numbers represent the count of the number of topological features.

Algorithm 1 Extraction of Topological Features.

Input: $N \leftarrow$ number of dataset

for $i = 1, 2, \dots, N$ **do**

$img \leftarrow$ load i th image from dataset

$img \leftarrow$ resize img to (200, 200) and convert to grayscale

$PD_0 \leftarrow$ set of points of β_0 in persistence diagram of img with cubical complex

$PD_1 \leftarrow$ set of points of β_1 in persistence diagram of img with cubical complex

$PD_0 \leftarrow$ sort PD_0 in descending order of *persistence*

$PD_1 \leftarrow$ sort PD_1 in descending order of *persistence*

$d_i \leftarrow$ project each point in PD_0 to $[0, 1]$

$d_i \leftarrow d_i +$ project each point in PD_1 to $[1, 2]$

$fimg \leftarrow$ adapt CLAHE filter to img

$fPD_0 \leftarrow$ set of points of β_0 in persistence diagram of $fimg$ with cubical complex

$fPD_1 \leftarrow$ set of points of β_1 in persistence diagram of $fimg$ with cubical complex

$fPD_0 \leftarrow$ sort fPD_0 in descending order of *persistence*

$fPD_1 \leftarrow$ sort fPD_1 in descending order of *persistence*

$d_i \leftarrow d_i +$ project each point in fPD_0 to $[0, 1]$

$d_i \leftarrow d_i +$ project each point in fPD_1 to $[1, 2]$

$d_i \leftarrow d_i +$ convert img to GLCM with distances (1, 2, 3), directions ($0^\circ, 45^\circ, 90^\circ, 135^\circ$), and properties (*energy, homogeneity*)

Output: $D(d_1, d_2, \dots, d_N)$

3.3. Projection of Persistence Diagrams

The construction of a persistence diagram is possible once the filtration (using cubical complex) is completed. The d th persistence diagram, $\mathcal{D}_d$, contains all of the d -dimensional topological information. These are a series of points with a pair of (*birth*, *death*), where *birth* indicates the time at which the topological features were created and the *death* gives the time at which these features are destroyed. From here, *persistence* is defined using the definition of *birth* and *death* as

$$pers(birth, death) := death - birth, \text{ where } (birth, death) \in \mathcal{D}_d. \quad (8)$$

Low-persistence features are treated as having low importance, or ‘noise’, whereas high-persistence features are regarded as *true* features [1]. However, using *persistence* as a result of a projection of a topological feature to a 1-dimensional value is not helpful, because it is impossible to distinguish the features which have the same *persistence* but different values for *birth*. Therefore, we propose a measure (*score*) to compensate for this limitation of *persistence*, shown in Equation (9).

$$score_d(birth, death) := \begin{cases} 0 & \text{if } persistence < threshold \\ d + \left(\frac{e^{\sin(\frac{death}{255} \cdot \frac{\pi}{2})} - 1}{e - 1} \right)^3 - \left(\frac{e^{\sin(\frac{birth}{255} \cdot \frac{\pi}{2})} - 1}{e - 1} \right)^3 & \text{if } persistence \geq threshold \end{cases} \quad (9)$$

Since the sinusoidal term is increasing and has the value $[0, 1]$ when the input is $[0, \frac{\pi}{2}]$, the $score_d$ has a value range from d to $d + 1$. Hence, it is easy to distinguish the dimension and persistence of each feature. Moreover, a higher exponent emphasizes a feature that has longer persistence and, significantly, ignores a feature that has shorter persistence. This in is keeping with ideas underlying homology groups, where longer the persistence, the

higher the significance of the homology. Conversely, the homology group that has short persistence is considered noise, which degrades the quality of the digital image and, as such, is less significant as a topological feature [10]. By ignoring such noise, using a threshold (as a parameter) allows us to separate useful features from noise. The optimal threshold (value = 10) was determined experimentally by comparing the performance of machine learning models. In summary, the *score* takes into account not only the *persistence*, but also other aspects such as the *dimension*, *birth*, and *death* of topological features.

3.4. Contrast Limited Adapting Histogram Equalization (CLAHE)

When pixel values are concentrated in a narrow range, it is hard to perceive features visually. Histogram equalization makes the distribution of pixel values in the image balanced, thereby enhancing the image. However, this method often results in degrading the content of the image and also amplifying the noise. Therefore, it produces undesirable results. Contrast limited adapting histogram equalization (CLAHE) is a well-known method for compensating for the weakness of histogram equalization by dividing an image into small-sized blocks and performing histogram equalization for each block [41]. After completing histogram equalization in all blocks, bilinear interpolation makes the boundary of the tiles (blocks) smooth. In this paper, we used the following hyperparameters: `clipLimit=7` and `tileGridSize=((8, 8))`. An illustration of the CLAHE method on the APTOS data is given in Figure 12.

Figure 12. Comparison of the original image and the CLAHE-filtered image. (a) Original image, (b) persistence diagram of the original image (a), (c) CLAHE-filtered image, (d) persistence diagram of the filtered image (c).

The texture of an image can be described by its statistical properties and this information is useful to classify images [42]. For extracting texture features, we used the well-known gray-level co-occurrence matrix (GLCM) [43]. GLCM extracts texture infor-

mation regarding the structural arrangement of surfaces by using a displacement vector defined by its radius and orientation. We used three distances (1, 2, 3) and four directions (0° , 45° , 90° , 135°) to obtain the GLCM features. From each of the co-occurrence matrices, two global statistics were extracted, energy and homogeneity, resulting in $3 \times 4 \times 2 = 24$ texture features for each image.

Table 2 gives a list of extracted features from the APTOS dataset during the filtration process. In total, 144 features were extracted for each dimension ($fdim_0$ and $fdim_1$) from the CLAHE-filtered image in descending order of persistence. Similarly, 100 (dim_0 and dim_1) topological features for each dimension were extracted. Note that dim_0 represents β_0 and dim_1 represents β_1 Betti numbers, respectively. A total of 24 GLCM features were extracted from the original gray-level image.

In this paper, feature engineering and learning algorithms were implemented with the following Python libraries: Gudhi [44,45] for calculating persistent homology, PyTorch [46] for modeling and execution of ResNet 1D, and scikit-learn [47] for implementation of other machine learning algorithms. Moreover, libraries such as NumPy [48] and pandas [49] were used for computing matrices and analyzing the data structure. All tests were conducted using a desktop workstation with Intel i7-9700K at 3.6 GHz, 8 CPU cores, 16 GB RAM, and Gigabyte GeForce RTX 2080 GPU. The following algorithms were used.

Deep Residual Network, suggested by [50], is an ensemble of VGG-19 [51], plain network, and residual network as a solution to the network depth-accuracy degradation problem. This is done by a residual learning framework, which is a feedforward network with a shortcut. Multi-scale 1D ResNet is used in this work, where multi-scale refers to flexible convolutional kernels rather than flexible strides [52]. The authors use different sizes of kernels so that the network can learn features from original signals with different views with multiple scales. The structure of the model is described in Figure 13. The 1D ResNet model consists of a number of subblocks of the basic CNN blocks. A *basic CNN block* computes batch normalization after convolution as follows: $y = W \otimes x + b$, $s = \text{BN}(y)$, and $h = \text{ReLU}(s)$, where $\otimes$ denotes the convolution operator and BN is a batch normalization operator. Moreover, stacking two basic CNN blocks forms a *subblock* of the basic CNN blocks as follows: $h_1 = \text{Basic}(x)$, $h_2 = \text{Basic}(h_1)$, $y = h_2 + x$, and $\hat{h} = \text{ReLU}(y)$, where the *Basic* operator denotes the basic block described above. Using the above method, it is possible to construct multiple sub-blocks of the CNN with different kernel sizes. For training the network, we used an early stopping option if there was no improvement in the validation loss after 20 epochs. Using the early stopping option and a learning rate of 0.01, the network was trained over 100 epochs, since the average training time was around 50 epochs.

For the other machine learning models, the *random forest* algorithm with 200 trees, *gini* as a criterion, and unlimited depth was used. For the K – *nearest neighbors* (*kNN*), the following parameters were used: $k = 5$ and *Minkowski* as the distance metric. While the random forest algorithm is an ensemble method based on the concept of bagging, *GBM* [53] uses the concept of boosting, iteratively training the model by adding new weak models consecutively with the negative gradient from the loss function. Both extreme gradient boosting (*XGBoost*) [54,55] and *lightGBM* [56] are advanced models of the gradient boosting machines. *LightGBM* combines two techniques: *Gradient-based One-Side Sampling* and *Exclusive Feature Bundling* [57]. Since our dataset is tabular, *XGBoost*, which is a more efficient implementation of *GBM*, was used. For the *XGBoost* implementation, the following training parameters were used: 1000 $n_estimators$ for creating weak learners, $learning_rate = 0.3$ (*Eta*), and $max_depth = 6$. For the *LightGBM* implementation, the following training parameters were used: 1000 $n_estimators$ for creating weak learners, $learning_rate = 0.1$ (*Eta*), and $num_leaves = 31$.

Table 2. Results of feature engineering process applied to the APTOS dataset.

img	label	glcm1	glcm2	...	glcm24	dim0_ 0	dim0_ 1	...	dim0_ 99	dim1_ 0	dim1_ 1	...	dim1_ 99	fdim0_ 0	fdim0_ 1	...	fdim0_ 143	fdim1_ 0	fdim1_ 1	...	fdim1_ 143
0	2	0.1603	0.1571	...	0.4639	1	0.0366	...	0	1.2054	1.1815	...	0	0.9999	0.2060	...	0.0319	1.7698	1.6339	...	1.1067
⋮																					
3661	2	0.1196	0.1160	...	0.5387	0.9999	0.0020	...	0	1.4787	1.0636	...	0	0.9999	0.1295	...	0.0042	1.9493	1.3658	...	1.10478

For all the datasets (except the Indian fruits dataset), 80% of the dataset was used for training and 20% was used for testing. The Indian fruits dataset was already separated into 90% for training and 10% for testing. This ratio was used in our experiments.

Figure 13. Structure of the multi-scale 1D ResNet.

4. Results and Discussion

Table 3 gives the accuracy, weighted F1 score, and run time information for each of the datasets. The accuracy score reported with the benchmark datasets is given in the last column. The best result is indicated in blue. Overall, ResNet 1D outperforms other ML models, while different types of gradient boosting machines show fairly good accuracy and weighted F1 scores. In terms of binary classification image datasets, as in the concrete dataset, most of the algorithms achieve 0.99 accuracy and F1 score. However, for the multi-class image datasets, the SVM and kNN perform poorly, mainly due to the inherent difficulty of finding the best parameters. All machine learning models perform significantly worse than the benchmark results with the Fashion MNIST and APTOS datasets with the extracted topological features. This is because it is hard to obtain good trainable topological signatures from images that have low resolution, even though Fashion MNIST was resized (please see Table 1).

In the case of the APTOS dataset, imbalanced training data were the main cause of the poor results. For example, Label 0 indicates the absence of diabetic retinopathy and has the highest number of images (See Figure 6). However, the presence of diabetic retinopathy can be found in four classes, of which Label 2 (severity level 2.0) has the largest number of cases. As a result, more than half of the examples were classified as Label 2 (see Figure 14c).

Table 3. Accuracy and weighted F1 score for each dataset.

		ResNet 1D	CART	GBM	LightGBM	Random Forest	SVM	XGBoost	kNN	Related Works—Benchmark
Concrete	Accuracy	0.994	0.989	0.991	0.9945	0.993	0.956	0.9935	0.890	0.999 with CNN [35]
	Weighted F1	0.994	0.988	0.989	0.994	0.992	0.955	0.993	0.884	
	Run time	465.87	9.08	252.15	11.25	7.63	214.05	59.15	1.93	
Mangopest	Accuracy	0.931	0.764	0.681	0.898	0.869	0.474	0.889	0.666	0.76 with CNN [37]
	Weighted F1	0.931	0.764	0.676	0.898	0.869	0.439	0.889	0.663	
	Run time	760.94	17.17	5562.09	260.62	13.94	662.45	2041.22	2.33	
Indian fruits	Accuracy	1.000	0.9608	0.9608	0.9608	0.9608	0.7313	0.9608	0.676	0.999 SVM with deep features [38]
	Weighted F1	1.000	0.9608	0.9608	0.9608	0.9608	0.7236	0.9608	0.656	
	Run time	271.21	4.44	4265.09	82.55	4.13	72.73	451.65	1.18	
Fashion MNIST	Accuracy	0.7427	0.567	0.696	0.749	0.693	0.535	0.746	0.397	0.99 with CNN [58]
	Weighted F1	0.7414	0.569	0.694	0.749	0.692	0.529	0.746	0.390	
	Run time	467.12	8.36	1808.07	89.37	7.66	935.21	1108.20	3.38	
APTOS	Accuracy	0.7326	0.698	0.760	0.787	0.782	0.674	0.775	0.655	0.971 with CNN [59]
	Weighted F1	0.667	0.695	0.737	0.771	0.757	0.591	0.764	0.637	
	Run time	61.81	0.63	86.02	13.16	0.70	3.49	42.34	0.08	
Colorectal histology	Accuracy	0.892	0.75	0.842	0.869	0.855	0.679	0.874	0.759	0.874 with SVM [39]
	Weighted F1	0.89	0.727	0.832	0.850	0.834	0.686	0.843	0.743	
	Run time	86.23	1.18	255.08	12.52	1.10	4.06	44.63	0.14	
	Accuracy	0.882 ± 0.109	0.789 ± 0.147	0.822 ± 0.121	0.876 ± 0.087	0.856 ± 0.102	0.675 ± 0.154	0.873 ± 0.90	0.674 ± 0.148	
	Weighted F1	0.871 ± 0.125	0.784 ± 0.148	0.815 ± 0.124	0.870 ± 0.091	0.851 ± 0.105	0.654 ± 0.164	0.866 ± 0.092	0.662 ± 0.147

Figure 14. Confusion matrices of implementation with ResNet 1D. (a) Concrete, (b) Fashion MNIST, (c) APTOS, (d) Colorectal Histology.

Imbalanced data such as Mangopest and Indian fruits were classified well because there were sufficient training examples. In summary, the best classification performance using cubical homology with the ResNet 1D classifier was obtained for 2 out of 6 datasets using our proposed feature extraction method and score measure. However, these topological signatures were not helpful in the classification of the Fashion MNIST and APTOS images. For the Indian fruits dataset, the model classifies with an accuracy of 1.00, which is

comparable since there is just 0.001 improvement. Similarly, with the concrete dataset, the result is comparable, with only a slight difference (≤ 0.005) with the benchmark result.

In Table 4, we give an illustration of the performance of classifiers using features derived from cubical homology (TDA), GLCM, and combined TDA-GLCM for two datasets. It is clear from the results that the combined feature set using topological features and GLCM features results in better classifier accuracy.

Confusion matrices for experiments with the 1D ResNet model are given in Figures 14–16. It is noteworthy that, for these datasets, the application of cubical homology has led to meaningful results in 4 out of 6 datasets.

Figure 15. Confusion matrix for the Indian fruits dataset with ResNet 1D implementation.

True label	0	195 80%	1 0%	3 1%	0 0%	3 1%	12 5%	10 4%	0 0%	3 1%	0 0%	4 2%	0 0%	4 2%	10 4%	0 0%	0 0%
	1	2 0%	598 91%	3 0%	9 1%	0 0%	1 0%	16 2%	1 0%	2 0%	4 1%	15 2%	0 0%	1 0%	5 1%	1 0%	0 0%
	2	5 2%	1 0%	281 85%	3 1%	0 0%	7 2%	15 5%	0 0%	3 1%	1 0%	12 4%	1 0%	2 1%	1 0%	0 0%	0 0%
	3	0 0%	4 0%	0 0%	968 96%	0 0%	0 0%	6 1%	14 1%	3 0%	4 0%	5 0%	0 0%	4 0%	3 0%	0 0%	0 0%
	4	1 0%	0 0%	1 0%	0 0%	210 93%	3 1%	6 3%	0 0%	2 1%	0 0%	0 0%	0 0%	0 0%	2 1%	0 0%	2 1%
	5	6 1%	2 0%	4 1%	3 1%	2 0%	471 88%	11 2%	0 0%	4 1%	5 1%	13 2%	0 0%	4 1%	7 1%	1 0%	0 0%
	6	2 0%	4 0%	13 1%	2 0%	3 0%	12 1%	1028 93%	3 0%	4 0%	6 1%	20 2%	0 0%	7 1%	5 0%	0 0%	1 0%
	7	1 0%	1 0%	1 0%	12 1%	0 0%	1 0%	11 1%	1506 98%	2 0%	0 0%	2 0%	0 0%	0 0%	2 0%	0 0%	0 0%
	8	2 0%	0 0%	2 0%	1 0%	1 0%	13 2%	3 1%	3 1%	557 95%	0 0%	0 0%	1 0%	2 0%	1 0%	0 0%	1 0%
	9	0 0%	6 2%	4 1%	3 1%	0 0%	0 0%	8 2%	7 2%	0 0%	357 90%	4 1%	0 0%	3 1%	2 1%	2 1%	0 0%
	10	0 0%	18 3%	8 1%	8 1%	0 0%	2 0%	23 3%	1 0%	1 0%	7 1%	583 88%	1 0%	6 1%	4 1%	1 0%	0 0%
	11	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	4 6%	0 0%	0 0%	66 94%	0 0%	0 0%	0 0%	0 0%
	12	5 1%	2 0%	5 1%	5 1%	1 0%	3 0%	10 1%	2 0%	1 0%	4 1%	6 1%	0 0%	635 93%	0 0%	1 0%	0 0%
	13	2 0%	2 0%	1 0%	6 1%	0 0%	12 2%	2 0%	1 0%	6 1%	1 0%	11 2%	0 0%	6 1%	435 89%	3 1%	1 0%
	14	0 0%	1 0%	0 0%	0 0%	0 0%	0 0%	1 0%	1 0%	3 1%	0 0%	0 0%	0 0%	2 1%	0 0%	228 97%	0 0%
	15	0 0%	0 0%	0 0%	0 0%	3 1%	1 0%	0 0%	0 0%	1 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	519 99%
		Predicted label															

Figure 16. Confusion matrix for the Mangopest dataset with ResNet 1D implementation.

Table 4. Comparison of performance of GLCM+TDA, TDA feature-only, and GLCM-only implemented by 1D ResNet model on two datasets.

	Colorectal Histology Dataset	APTOS Dataset
GLCM+TDA	0.892	0.7326
TDA	0.7697	0.6739
GLCM	0.694	0.7252

5. Conclusions

The focus of this paper was on feature extraction from 2D image datasets using a specific topological method (cubical homology) and a novel score measure. These features were then used as input to well-known classification algorithms to study the efficacy of the proposed feature extraction process. We proposed a novel scoring method to transform the 2D input images into a one-dimensional array to vectorize the topological features. In this study, six published datasets were used as benchmarks. ResNet 1D, LightGBM, XGBoost, and five well-known machine learning models were trained on these datasets. Our experiments demonstrated that, in three out of six datasets, our proposed topological feature method is comparable to (or better than) the benchmark results in terms of accuracy. However, with two datasets, the performance of our proposed topological feature method is poor, due to either low resolution or an imbalanced dataset. We also demonstrate that topological features combined with GLCM features result in better classification accuracy

in two of the datasets. This study reveals that the application of cubical homology to image classification shows promise. Since the conversion of input images to 2D data is very time-consuming, future work will involve (i) seeking more efficient ways to reduce the time for pre-processing and (ii) experimentation with more varied datasets. The problem of poor accuracy with imbalanced datasets needs further exploration.

Author Contributions: S.C.: Conceptualization, Investigation, Formal Analysis, Supervision, Resources, Original Draft. S.R.: Methodology, Validation, Software, Editing. All authors have read and agreed to the publisher version of the manuscript.

Funding: This research was funded by NSERC Discovery Grant#194376 and University of Winnipeg Major Research Grant#14977.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Edelsbrunner, H.; Letscher, D.; Zomorodian, A. Topological persistence and simplification. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 12–14 November 2000; IEEE Computer Society Press: Los Alamitos, CA, USA, 2000; pp. 454–463.
- Chazal, F.; Michel, B. An Introduction to Topological Data Analysis: Fundamental and Practical aspects for Data Scientists. *arXiv* **2017**, arXiv:1710.04019.
- Zomorodian, A.; Carlsson, G. Computing persistent homology. *Discr. Comput. Geom.* **2005**, *33*, 249–274. [[CrossRef](#)]
- Carlsson, G. Topology and data. *Bull. Am. Math. Soc.* **2009**, *46*, 255–308. [[CrossRef](#)]
- Edelsbrunner, H.; Harer, J. Persistent homology. A survey. *Contemp. Math.* **2008**, *453*, 257–282.
- Zomorodian, A.F. Computing and Comprehending Topology: Persistence and Hierarchical Morse Complexes. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, 2001.
- Aktas, M.E.; Akbas, E.; Fatmaoui, A.E. Persistence Homology of Networks: Methods and Applications. *arXiv* **2019**, arXiv:math.AT/1907.08708.
- Garin, A.; Tauzin, G. A topological “reading” lesson: Classification of MNIST using TDA. In Proceedings of the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1551–1556.
- Adams, H.; Chepushtanova, S.; Emerson, T.; Hanson, E.; Kirby, M.; Motta, F.; Neville, R.; Peterson, C.; Shipman, P.; Ziegelmeier, L. Persistence Images: A Stable Vector Representation of Persistent Homology. *arXiv* **2016**, arXiv:cs.CG/1507.06217.
- Kramár, M.; Goulet, A.; Kondic, L.; Mischaikow, K. Persistence of force networks in compressed granular media. *Phys. Rev. E* **2013**, *87*, 042207. [[CrossRef](#)]
- Nakamura, T.; Hiraoka, Y.; Hirata, A.; Escobar, E.G.; Nishiura, Y. Persistent homology and many-body atomic structure for medium-range order in the glass. *Nanotechnology* **2015**, *26*, 304001. [[CrossRef](#)]
- Dunaeva, O.; Edelsbrunner, H.; Lukyanov, A.; Machin, M.; Malkova, D.; Kuvaev, R.; Kashin, S. The classification of endoscopy images with persistent homology. *Pattern Recognit. Lett.* **2016**, *83*, 13–22. [[CrossRef](#)]
- Reininghaus, J.; Huber, S.; Bauer, U.; Kwitt, R. A stable multi-scale kernel for topological machine learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4741–4748.
- Iijima, T. Basic theory on the normalization of pattern (in case of typical one-dimensional pattern). *Bull. Electro-Tech. Lab.* **1962**, *26*, 368–388.
- Bonis, T.; Ovsjanikov, M.; Oudot, S.; Chazal, F. Persistence-based pooling for shape pose recognition. In Proceedings of the International Workshop on Computational Topology in Image Context, Marseille, France, 15–17 June 2016; pp. 19–29.
- Dey, T.; Mandal, S.; Varcho, W. Improved image classification using topological persistence. In Proceedings of the Conference on Vision, Modeling and Visualization, Bonn, Germany, 25–27 September 2017; pp. 161–168.
- Kindelan, R.; Frías, J.; Cerda, M.; Hitschfeld, N. Classification based on Topological Data Analysis. *arXiv* **2021**, arXiv:cs.LG/2102.03709.
- Carrière, M.; Chazal, F.; Ike, Y.; Lacombe, T.; Royer, M.; Umeda, Y. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In Proceedings of the International Conference on Artificial Intelligence and Statistics (PMLR), Online, 26–28 August 2020; pp. 2786–2796.
- Chung, M.K.; Lee, H.; DiChristofano, A.; Ombao, H.; Solo, V. Exact topological inference of the resting-state brain networks in twins. *Netw. Neurosci.* **2019**, *3*, 674–694. [[CrossRef](#)]

20. Don, A.P.H.; Peters, J.F.; Ramanna, S.; Tozzi, A. Topological View of Flows Inside the BOLD Spontaneous Activity of the Human Brain. *Front. Comput. Neurosci.* **2020**, *14*, 34. [\[CrossRef\]](#)
21. Don, A.P.; Peters, J.F.; Ramanna, S.; Tozzi, A. Quaternionic views of rs-fMRI hierarchical brain activation regions. Discovery of multilevel brain activation region intensities in rs-fMRI video frames. *Chaos Solitons Fractals* **2021**, *152*, 111351. [\[CrossRef\]](#)
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [\[CrossRef\]](#)
23. Hofer, C.D.; Kwitt, R.; Niethammer, M. Learning Representations of Persistence Barcodes. *J. Mach. Learn. Res.* **2019**, *20*, 1–45.
24. Umeda, Y. Time series classification via topological data analysis. *Inf. Media Technol.* **2017**, *12*, 228–239. [\[CrossRef\]](#)
25. Pun, C.S.; Xia, K.; Lee, S.X. Persistent-Homology-Based Machine Learning and its Applications—A Survey. *arXiv* **2018**, arXiv:math.AT/1811.00252.
26. Allili, M.; Mischaikow, K.; Tannenbaum, A. Cubical homology and the topological classification of 2D and 3D imagery. In Proceedings of the 2001 International Conference on Image Processing (Cat. No. 01CH37205), Thessaloniki, Greece, 7–10 October 2001; Volume 2, pp. 173–176.
27. Kot, P. Homology calculation of cubical complexes in Rn. *Comput. Methods Sci. Technol.* **2006**, *12*, 115–121. [\[CrossRef\]](#)
28. Strömbom, D. Persistent Homology in the Cubical Setting: Theory, Implementations and Applications. Master's Thesis, Lulea University of Technology, Lulea, Sweden, 2007.
29. Choe, S. Cubical homology-based Image Classification-A Comparative Study. Master's Thesis, University of Winnipeg, Winnipeg, MB, Canada, 2021.
30. Fisher, M.; Springborn, B.; Schröder, P.; Bobenko, A.I. An algorithm for the construction of intrinsic Delaunay triangulations with applications to digital geometry processing. *Computing* **2007**, *81*, 199–213. [\[CrossRef\]](#)
31. Otter, N.; Porter, M.A.; Tillmann, U.; Grindrod, P.; Harrington, H.A. A roadmap for the computation of persistent homology. *EPJ Data Sci.* **2017**, *6*, 1–38. [\[CrossRef\]](#)
32. Kaczynski, T.; Mischaikow, K.M.; Mrozek, M. *Computational Homology*; Springer: Berlin/Heidelberg, Germany, 2004.
33. Kalies, W.D.; Mischaikow, K.; Watson, G. Cubical approximation and computation of homology. *Banach Cent. Publ.* **1999**, *47*, 115–131. [\[CrossRef\]](#)
34. Marchese, A. Data Analysis Methods Using Persistence Diagrams. Ph.D. Thesis, University of Tennessee, Knoxville, TN, USA, 2017.
35. Özgenel, Ç.F.; Sorguç, A.G. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), Berlin, Germany, 20–25 July 2018; Volume 35, pp. 1–8.
36. Avilés-Rodríguez, G.J.; Nieto-Hipólito, J.I.; Cosío-León, M.d.l.Á.; Romo-Cárdenas, G.S.; Sánchez-López, J.d.D.; Radilla-Chávez, P.; Vázquez-Briseño, M. Topological Data Analysis for Eye Fundus Image Quality Assessment. *Diagnostics* **2021**, *11*, 1322. [\[CrossRef\]](#)
37. Kusurini, K.; Suputa, S.; Setyanto, A.; Agastya, I.M.A.; Priantoro, H.; Chandramouli, K.; Izquierdo, E. Data augmentation for automated pest classification in Mango farms. *Comput. Electron. Agric.* **2020**, *179*, 105842. [\[CrossRef\]](#)
38. Behera, S.K.; Rath, A.K.; Sethy, P.K. Fruit Recognition using Support Vector Machine based on Deep Features. *Karbala Int. J. Mod. Sci.* **2020**, *6*, 16. [\[CrossRef\]](#)
39. Kather, J.N.; Weis, C.A.; Bianconi, F.; Melchers, S.M.; Schad, L.R.; Gaiser, T.; Marx, A.; Zöllner, F.G. Multi-class texture analysis in colorectal cancer histology. *Sci. Rep.* **2016**, *6*, 27988. [\[CrossRef\]](#)
40. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
41. Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; ter Haar Romeny, B.; Zimmerman, J.B.; Zuiderveld, K. Adaptive histogram equalization and its variations. *Comput. Vision Graph. Image Process.* **1987**, *39*, 355–368. [\[CrossRef\]](#)
42. Gadkari, D. Image Quality Analysis Using GLCM. Master's Thesis, University of Central Florida, Orlando, FL, USA, 2004.
43. Mohanaiah, P.; Sathyanarayana, P.; GuruKumar, L. Image texture feature extraction using GLCM approach. *Int. J. Sci. Res. Publ.* **2013**, *3*, 1–5.
44. The GUDHI Project. *GUDHI User and Reference Manual*, 3.4.1 ed.; GUDHI Editorial Board; GUDHI: Nice, France, 2021.
45. Dlotko, P. Cubical complex. In *GUDHI User and Reference Manual*, 3.4.1 ed.; GUDHI Editorial Board; GUDHI: Nice, France, 2021.
46. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: North Adams, MA, USA, 2019; pp. 8024–8035.
47. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
48. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [\[CrossRef\]](#) [\[PubMed\]](#)
49. McKinney, W. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; Volume 445, pp. 51–56.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

51. Mateen, M.; Wen, J.; Song, S.; Huang, Z. Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry* **2019**, *11*, 1. [[CrossRef](#)]
52. Liu, R.; Wang, F.; Yang, B.; Qin, S.J. Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3797–3806. [[CrossRef](#)]
53. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
54. Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H. Xgboost: Extreme gradient boosting. *R Package Version 0.4-2* **2015**, *1*, 1–4.
55. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
56. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.
57. Patel, V.; Choe, S.; Halabi, T. Predicting Future Malware Attacks on Cloud Systems using Machine Learning. In Proceedings of the 2020 IEEE 6th International Conference on Big Data Security on Cloud (BigDataSecurity), High Performance and Smart Computing, (HPSC) and Intelligent Data and Security (IDS), Baltimore, MD, USA, 25–27 May 2020; pp. 151–156.
58. Kayed, M.; Anter, A.; Mohamed, H. Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture. In Proceedings of the 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), Aswan, Egypt, 8–9 February 2020; pp. 238–243.
59. Tymchenko, B.; Marchenko, P.; Spodarets, D. Deep learning approach to diabetic retinopathy detection. *arXiv* **2020**, arXiv:2003.02261.