The Classification of Endoscopy Images with Persistent Homology *

Olga Dunaeva[†] [‡], Herbert Edelsbrunner[§] [‡], Anton Lukyanov[†] [‡], Michael Machin[†] [‡], Daria Malkova[†] [‡], Roman O. Kuvaev[¶] and Sergey V. Kashin[¶]

Abstract

Aiming at the automatic diagnosis of tumors using narrow band imaging (NBI) magnifying endoscopy (ME) images of the stomach, we combine methods from image processing, topology, geometry, and machine learning to classify patterns into three classes: *oval*, *tubular* and *irregular*. Training the algorithm on a small number of images of each type, we achieve a high rate of correct classifications. The analysis of the learning algorithm reveals that a handful of geometric and topological features are responsible for the overwhelming majority of decisions.

Keywords. Endoscopy, medical imaging, automated diagnostics, image processing, computational topology, persistent homology, machine learning.

1 Introduction

We study magnifying endoscopy (ME) images applied to the diagnosis of diseases in the gastrointestinal tract. The ME technology permits the detailed visualization of the mucosal micro-surface pattern and the vascular architecture to be obtained. For enhanced contrast between vessels and background mucosal surface, we use the narrow band imaging (NBI) system that became available during the last ten years. In combination, ME and NBI make it possible to find and to differentiate suspicious cancerous lesions in the stomach at an early stage during the procedure.

From patterns to diagnoses. For the diagnosis of tumors with NBI-ME, the VS-classification of K. Yao *et al*. is widely accepted. It is based on the observed microvascular architecture ('V' for vascular) and microsurface structure ('S' for surface). Both patterns are evaluated independently. According to this classification, there are two types of microvascular patterns:

- *regular*, with clearly defined and uniform geometric shape and position of the vessels;
- *irregular*, with non-uniform geometric shape and position of vessels.

The first type is typical for the non-neoplastic surface structure of the stomach without malignant changes. For the early stomach cancer, T. Nakayoshi *et al*. define two basic types of irregular microvascular pattern:

- *fine network pattern*, with a large number of microvessels connected in the form of a fine network;
- *corkscrew pattern*, with vessels of twisted shape that do not join to form a network.

As shown in Table 1, we refine the classification of regular and irregular microvascular patterns by distinguishing between open and closed loops. In the irregular case, the open loops are characteristic of corkscrew patterns, while the closed loops correspond to fine network patterns.

Surface		Vessels			
		Regular		Irregular	
		Closed	Open	Closed	Open
Regular	Oval	LGD - 0% HGD - 0% AC - 0%			
	Tubular		LGD - 8.9% HGD - 9.8% AC - 0%		
	Villous		LGD - 0% HGD - 0% AC - 0%		
Irregular			LGD - 30% HGD - 0% AC - 30%	LGD - 28.8% HGD - 42.9% AC - 28.8%	LGD - 0% HGD - 57.2% AC - 42.9%
Absent				LGD - 0% HGD - 10.3% AC - 89.7%	LGD - 0% HGD - 8.3% AC - 91.7%

Table 1: Classification of the medical cases, each represented by a small set of endoscopy images. We write LGD for low-grade dysplasia, HGD for high-grade dysplasia, and AC for adenocarcinoma, with associated probabilities estimated from a small sample of patients at the Yaroslavl Region Cancer Hospital.

The microstructure of the stomach surface consists of gastic pits and sulci. There are several key components in

*This research is partially supported by the Russian Government under the Mega Project 11.G34.31.0053.

[†]Department of Computer Science, P. G. Demidov Yaroslavl State University, Yaroslavl, Russian Federation.

[‡]Delone Laboratory of Discrete and Computational Geometry, P. G. Demidov Yaroslavl State University, Yaroslavl, Russian Federation.

[§]IST Austria (Institute of Science and Technology Austria), Klosterneuburg, Austria.

[¶]Yaroslavl Region Cancer Hospital, Yaroslavl, Russian Federation.

the microanatomy of the epithelial surface: *crypt opening*, *marginal epithelium*, *intravenous part*. The surface structure of the normal stomach mucosa consists of rounded and oval pits, which histologically correspond to gastric glands. In the NBI vessels become dark. As a result, we can see the specific pattern of the gastric mucosa as bright areas surrounded by a dark rims; see Figure 1. The types of microsurface structure we observe are:

- *regular*, with marginal epithelium that has uniform geometric shape and size;
- *irregular*, with marginal epithelium that has non-uniform geometric shape and size,

but it is also possible that the microsurface pattern is *absent*. The combination of microsurface and microvascular patterns defines the probability of one of three types of cancer risk: *low-grade dysplasia*, *high-grade dysplasia*, and *adenocarcinoma*. In our work, we focus on three combinations of patterns:

OVAL: oval surface structure combined with regular closed microvessels;

TUBULAR: tube-like surface structure combined with regular open microvessels;

IRREGULAR: without any microsurface structure combined with irregular microvessels.

The first type is indicative of a no risk, second type of a low risk and the third of a high risk for cancer. While the three categories cover only four of the shaded fields in Table 1, this limitation is forced on us by a shortage of image data. With time the database will grow to cover the remaining fields. The methods given in this paper can then be used to solve the correspondingly larger classification problem.

Prior work and results. The medical aspects of our work are discussed in [15], where the structure of images from different parts of the stomach obtained by the NBI technique and relevant types of tumors are described, and in [13], where the irregular pattern of colon mucosa for various cancer stages are explained.

On the computational side, the use of image processing and machine learning techniques to analyze images is widespread, but the use of computational topology methods in the analysis is novel. The use of Bayes' Rule to predict the histological structure of mucosa based on endoscopic image analysis predates our paper; see [10]. Even the use of geometric properties, such as the area and perimeter of vessels to diagnose colorectal oncological tumors predates our work; see [18]. Closest to our work is that of Häfner's group in Austria; see [5, 6, 10, 11] and additional publications exploring variations of the same theme. The main differences to our work are (1) that they look at endoscopy images of the colon, which in medical practice are easier to classify than images of the stomach, (2) that they distinguish between two to five classes – compared to three classes in this paper – but in contrast to our data, each of their images represents one uniformly classified surface piece, and (3) that they do not use topological features, with are essential to the success of our classification efforts.

In summary, the main innovative aspect of our work is the use of topological features derived from the persistent homology of the images. These are essential to achieve our best result of about 89% correct classification, which is obtained using a selection of three geometric and two topological features.

Outline. Section 2 explains the *methods* used in our approach, discussing the necessary background from image processing in Section 2.1, from computational topology in Section 2.2, from geometry in Section 2.3, and from machine learning in Section 2.4. Section 3 presents the *results* obtained with our approach, discussing the data and test setting in Section 3.1, the selection of features in Section 3.2, and the success rate of our classification algorithm in Section 3.3. Section 4 concludes this paper.

2 Methods

The input data consists of images represented in one of several well known raster formats, including BMP, Jpeg, and PNG. Each image is a 2-dimensional array of pixels. Every pixel holds the quantified value of color in RGB color space, i.e. a vector of three components $[R, G, B]$ corresponding to the red, green, and blue colors respectively. Each component is an integer in the interval $[0, 255]$. Except for specular reflection recognition, we use gray values of pixels calculated from RGB color using the well known formula [16] and scaled to $[0, 1]$ in all image processing procedures. Assuming that the input image is w pixels wide and h pixels high, we consider the grayscale image as the function $b\colon \mathbb{M} \to [0, 1]$, where $\mathbb{M}$ is the rectangle covered by the pixels, and b maps the pixel with center (x, y) in $\mathbb{M}$ to the grayscale value $b(x, y)$. Bright areas receive large values (close to 1), while dark areas correspond to small values (close to 0).

2.1 Image Processing

We use image processing methods to prepare the image for later analysis. Because of the condition under which endoscopy images are taken, such preparations are indispensable. They include the exclusion of areas with specular reflections, and the equalization of image brightness.

Specular reflection. This phenomenon is caused by local over-exposure to light, which leads to white areas with blurred boundaries. Within these areas, it is not possible to

recognize any pattern information. This is the reason we select areas with specular reflections and remove them from the downstream analysis.

To do so, we convert each source image from RGB to the HSB color model, where the three letters stand for *hue*, *saturation*, and *brightness*. Write (x, y) for a point of the image and (h, s, b) for the triplet of hue, saturation, and brightness values at the point. We can recognize specular reflections by checking whether $s < s_0$ and $b > b_0$, for suitable constants s_0 and b_0; see [17]. We use $s_0 = 0.2$ and $b_0 = 0.9$ in this paper. We assume that all three components of the HSB color representation are scaled to the interval $[0, 1]$.

Brightness equalization. Sometimes this operation is referred to as *background modeling*. Its goal is to locally modify the image to achieve a uniform brightness level over the entire area, while maintaining the local patterns in the image. We illustrate the effect of the operation in Figure 1.

To achieve the desired equalization, we apply a median filter within a window around each point several times. Let $W_r(x, y)$ be the square window of $(2r + 1)^2$ pixels centered at (x, y). The result of the *median filter* applied to the image, $b\colon \mathbb{M} \to [0, 1]$, is another image, $\mathcal{M}_r b\colon \mathbb{M} \to [0, 1]$ defined by mapping (x, y) to the median of the values of f at the points inside $W_r(x, y)$. Similarly, we introduce the *iterated* median filter, $\mathcal{M}_r^k$, obtained by k times iterating $\mathcal{M}_r$. To get the final result, we replace the original image by

$$f = b - \mathcal{M}_r^k b + \left\langle \mathcal{M}_r^k b \right\rangle, \tag{1}$$

where r and k are the parameters of the equalization, and $\left\langle \mathcal{M}_r^k b \right\rangle$ is the average brightness of the image $\mathcal{M}_r^k b$. We use $r = 8$ and $k = 5$ in this paper.

Localization. A single image may show different patterns at different places. It is therefore necessary to apply the analysis locally. We achieve this by covering the image with square windows of radius r, noting that the value of r here is significant larger than used to equalize brightness. Specifically, we use $r = 70$ throughout this paper for windows. We choose the centers of the windows randomly, subsampling a random set of points to avoid tight clusters which would lead to redundant windows. To explain the subsampling procedure, let M be a set of points (pixels) chosen uniformly at random from $\mathbb{M}$. We compute a subset $N \subseteq M$ as follows:

1. Pick a random point p_0 from M, set $M = M \setminus \{p_0\}$, $N = \{p_0\}$, and $i = 1$.
2. Repeat the following step until the desired number of windows is reached: let p_i be a point in M that maximizes the distance to the nearest point in N, set $M = M \setminus \{p_i\}$, $N = N \cup \{p_i\}$, and $i = i + 1$.

This heuristic is sometimes referred to as the greedy algorithm for the k-center clustering problem [7, Section 4.2]. It is frequently used in subsampling large datasets.

2.2 Topological Analysis

The core of the analysis is based on geometric and topological concepts. We discuss the topological approach using homology and persistent homology in this subsection. We refer to [12] for general background in algebraic topology including homology, and to [1] for general background in computational topology including persistent homology.

Global binarization. We recall that $f\colon \mathbb{M} \to [0, 1]$ is our image, which is decomposed into small picture elements, called *pixels*, arranged in a square grid. Given a threshold $s \in [0, 1]$, the *sublevel set* consists of all pixels with $f(x, y) \leq s$. We write $\mathbb{M}_s = f^{-1}[0, s]$ for this set. Equivalently, we may think of this operation as a *global binarization* of the image:

$$f_s(x, y) = \begin{cases} 0 & \text{if } f(x, y) \leq s, \\ 1 & \text{if } f(x, y) > s, \end{cases} \tag{2}$$

with $\mathbb{M}_s = f_s^{-1}(0)$. We call $\mathbb{M}_s$ the *background* and its complement, $\mathbb{M} - \mathbb{M}_s$, the *foreground* of the binarized image. The general philosophy of persistent homology is to study the entire family of globally binarized images as a single object. In particular, we analyze how the connectivity evolves as we gradually increase the threshold s. We measure connectivity by counting the components with $\beta_0 = \beta_0(\mathbb{M}_s)$, called the 0-*th Betti number*, and counting the holes with $\beta_1 = \beta_1(\mathbb{M}_s)$, called the 1-*st Betti number* of the sublevel set. Formally, these numbers are the ranks of the 0-th and 1-st homology groups.

A subtle but important aspect in counting components and holes is the notion of local neighborhood. We follow the convention that the background be topologically closed and the foreground be topologically open. The choice of neighborhood is important because four pixels can meet non-generically at a shared corner. If two diagonally opposite pixels belong to the background and the other two belong to the foreground, then this convention implies a locally connected background and a locally disconnected foreground. To implement this convention, we form a 2-dimensional simplicial complex in which the pixels are the vertices. Each (interior) pixel is connected to a number of neighbors that is between 4 and 8. In particular, we draw exactly one of the two possible diagonal edges, namely the one that does not touch the pixel with largest function value among the four. After drawing the edges, we fill in the triangles to complete the simplicial complex, which we refer to as the *adaptive triangulation* of the image. The detailed analysis and topological justification of this construction in two and higher dimensions can be found in [2].

Persistent homology. By increasing the threshold, we generate a time-series of globally binarized images. In this series, we see background components appear, grow, and merge, and we see holes in the background pinch off, shrink,

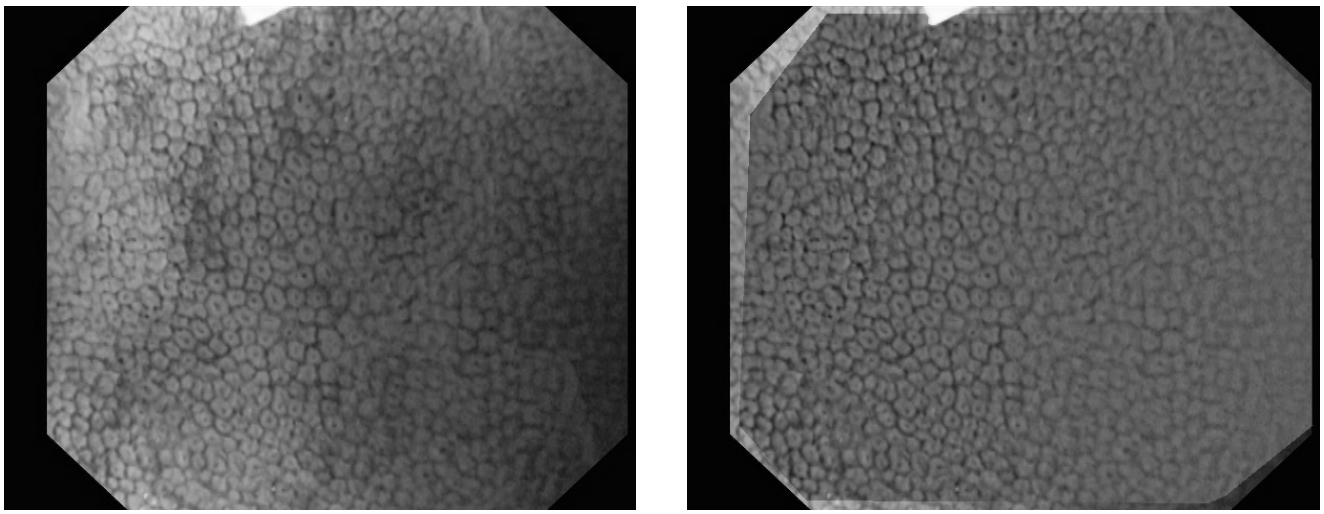

Figure 1: The microsurface pattern is regular and oval, the microvascular pattern is regular with closed loops. The grayscale image before brightness equalization on the *left* and after the brightness equalization on the *right*.

and disappear. Persistent homology quantifies these occurrences by considering the sequence of growing sublevel sets, which we call a *filtration*. Using the algebraic concept of homology, we map each sublevel set to two vector spaces. The first is the 0-*th homology group*, denoted as $H_0(\mathbb{M}_s)$, which is generated by the components, and the second is the 1-*st homology group*, denoted as $H_1(\mathbb{M}_s)$, which is generated by the loops of the sublevel set. Both groups are vector spaces because we compute homology with field coefficients. In addition, we get linear maps between the vector spaces that are induced by the inclusions between the sublevel sets. Simplifying notation by writing $H_{p,i} = H_p(\mathbb{M}_{s_i})$, we get two sequences of vector spaces connected by linear maps:

$$\ldots \to H_{p,i-1} \to H_{p,i} \to \ldots \to H_{p,j-1} \to H_{p,j} \to \ldots,$$

for $p = 0, 1$, where $s_{i-1} < s_i < s_{j-1} < s_j$ are thresholds with different sublevel sets, and s_{i-1}, s_i as well as s_{j-1}, s_j are contiguous thresholds. Now suppose that φ is a component or loop in $\mathbb{M}_{s_i}$ that does not already exist in $\mathbb{M}_{s_{i-1}}$. Formally, $\varphi \in H_{p,i}$ but φ is not in the image of the linear map from $H_{p,i-1}$ to $H_{p,i}$. We say that such a φ is *born* at s_i. After being born, it lives as long as it does not belong to the image of the linear map from $H_{p,i-1}$ to the current homology group. Finally, there is an index j such that φ mapped to $H_{p,j-1}$ does not belong to the image of $H_{p,i-1}$ in $H_{p,j-1}$, but φ mapped to $H_{p,j}$ belongs to the image of $H_{p,i-1}$ in $H_{p,j}$. In this case, we say that φ *dies entering* s_j. Furthermore, we call $s_j - s_i$ the *persistence* of φ.

It may be helpful to rephrase these definitions in more elementary terms separately for components and for holes. For $p = 0$, the homology groups track components. Letting $\varphi \in H_{0,i}$ be a components of the background at s_i, it is born at s_i if all its pixels belong to the foreground at s_{i-1}. It dies entering s_j if this is the first time when φ merges with a component that was born before φ. For $p = 1$, the homology groups track loops or, equivalently, holes in the background. A loop is born when it first closes around a hole, and it dies when this hole fills up. Here is an ambiguity in the language because we do not specify on which side of the loop the hole is located. It is sometimes helpful to consider what this means for the sequence of foregrounds, which forms a filtration running time backward. A hole in the background is a component in the foreground. The hole is born when the component dies and the hole dies when the component is born. This dual view will be useful shortly because it allows us to compute the persistence of components and of holes with the same algorithm applied twice.

Persistence diagrams and moments. The result of the persistence analysis is a collection of birth-death pairs, namely one pair for each component and one pair for each hole. It is common to visualize this information as a set of intervals, the so-called *barcode* of f, or as a set of points in the plane, the *persistence diagram* of f. We prefer the latter, drawing persistence = death - birth vertically, and death + birth horizontally; see Figure 2. We write $\mathrm{Dgm}_p(f)$ for the p-th persistence diagram, which contains all points marking the births and deaths of components, if $p = 0$, and of holes, if $p = 1$. More useful than the individual points or the diagrams are summaries of the information. Writing pers(A) for the persistence of the component or hole represented by a point A, the *q-th norm* of the p-th diagram is

$$N_q = \left[\sum_{A \in \mathrm{Dgm}_p(f)} \mathrm{pers}(A)^q \right]^{\frac{1}{q}}. \tag{3}$$

For example, $N_0(\mathrm{Dgm}_p(f))$ counts the points in the p-th diagram. If we restrict the count to the points whose persistence intervals cover a value, s, then we get the p-th Betti number of $\mathbb{M}_s$. While the zeroth norm is sensitive to small fluctuations, higher order norms are less sensitive and stability can be proved for some class of functions [1, Chapter VIII].

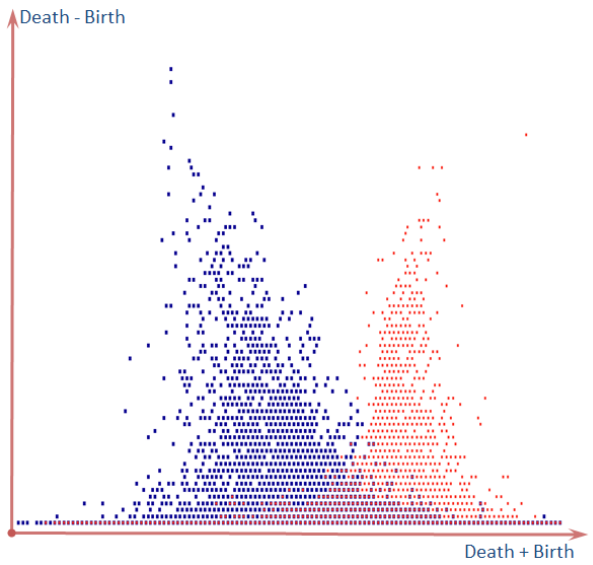

Figure 2: The superposition of two persistence diagrams, one for the components, and the other for the holes.

The norms are sometimes called moments of the persistence diagrams, but we reserve this term for the quantitative description of distributions common in statistics. Consider the histogram over the birth + death axis of the p-th persistence diagram. Similar to a bell curve, the resulting distribution tends to first increase and then decrease, with a single hump in the middle. Letting n be the total number of points and $x_i = b_i + d_i$ the horizontal coordinate of the i-th point, these central moments are

$$m_1 = \frac{1}{n}\sum x_i, \tag{4}$$

$$m_2 = \frac{1}{n}\sum (x_i - \mu)^2, \tag{5}$$

$$m_3 = \frac{1}{n}\sum \left(\frac{x_i - \mu}{\sigma}\right)^3, \tag{6}$$

$$m_4 = \frac{1}{n}\sum \left(\frac{x_i - \mu}{\sigma}\right)^4, \tag{7}$$

where we write $\mu = m_1$ for the *mean* or 1-*st moment*, and $\sigma^2 = m_2$ for the *variance* or 2-*nd central moment*. The *normalized* 3-*rd central moment*, m_3, is also known as the *skewness*, and the *normalized* 4-*th central moment*, m_4, is known as the *kurtosis* of the distribution.

List of topological features. Through numerous experimental tests, we have converged to using the following twenty-two topological features for image classification: maximum Betti numbers over all sublevel sets; three norms of the 0-th and three norms of the 1-st persistence diagram of the images; mean, variance, skewness, and kurtosis of a histogram of the distribution obtained from these two diagrams. In the underlying list of topological features we write *Hgm* for the histogram. We apply the persistence algorithm to the image, but also to derived functions. Given a threshold, s, we consider the *signed distance function*, $d_s \colon \mathbb{M} \to \mathbb{R}$, defined by mapping a point of the foreground to the Euclidean distance from the background, and a point of the background to minus the Euclidean distance from the foreground. Given a sublevel set, the signed distance function can be computed in time $m \log m$ using the fast distance transform algorithm[3]. In our work we use the threshold s_{max} at which the 1-st Betti number attains its maximum. Then we compute signed distance function for the sublevel set defined for s_{max} and obtain three norms of the 0-th and three norms of the 1-st persistence diagram of this sublevel set. To simplify the notation, we write $W = W(x, y)$ for the square window centered at the point $(x, y) \in \mathbb{M}$ and $f|_W$ for the function restricted to that window. The complete list of topological features used in this paper is therefore:

$$t_1(x,y) = \max_s \beta_0(\mathbb{M}_s \cap W); \tag{8}$$

$$t_2(x,y) = N_1(\mathrm{Dgm}_0(f|_W)); \tag{9}$$

$$t_3(x,y) = N_2(\mathrm{Dgm}_0(f|_W)); \tag{10}$$

$$t_4(x,y) = N_3(\mathrm{Dgm}_0(f|_W)); \tag{11}$$

$$t_5(x,y) = \max_s \beta_1(\mathbb{M}_s \cap W); \tag{12}$$

$$t_6(x,y) = N_1(\mathrm{Dgm}_1(f|_W)); \tag{13}$$

$$t_7(x,y) = N_2(\mathrm{Dgm}_1(f|_W)); \tag{14}$$

$$t_8(x,y) = N_3(\mathrm{Dgm}_1(f|_W)); \tag{15}$$

$$t_9(x,y) = m_1(\mathrm{Hgm}(\mathrm{Dgm}_0(f|_W))); \tag{16}$$

$$t_{10}(x,y) = m_2(\mathrm{Hgm}(\mathrm{Dgm}_0(f|_W))); \tag{17}$$

$$t_{11}(x,y) = m_3(\mathrm{Hgm}(\mathrm{Dgm}_0(f|_W))); \tag{18}$$

$$t_{12}(x,y) = m_4(\mathrm{Hgm}(\mathrm{Dgm}_0(f|_W))); \tag{19}$$

$$t_{13}(x,y) = m_1(\mathrm{Hgm}(\mathrm{Dgm}_1(f|_W))); \tag{20}$$

$$t_{14}(x,y) = m_2(\mathrm{Hgm}(\mathrm{Dgm}_1(f|_W))); \tag{21}$$

$$t_{15}(x,y) = m_3(\mathrm{Hgm}(\mathrm{Dgm}_1(f|_W))); \tag{22}$$

$$t_{16}(x,y) = m_4(\mathrm{Hgm}(\mathrm{Dgm}_1(f|_W))); \tag{23}$$

$$t_{17}(x,y) = N_1(\mathrm{Dgm}_0(d_{s_{max}}|_W)); \tag{24}$$

$$t_{18}(x,y) = N_2(\mathrm{Dgm}_0(d_{s_{max}}|_W)); \tag{25}$$

$$t_{19}(x,y) = N_3(\mathrm{Dgm}_0(d_{s_{max}}|_W)); \tag{26}$$

$$t_{20}(x,y) = N_1(\mathrm{Dgm}_1(d_{s_{max}}|_W)); \tag{27}$$

$$t_{21}(x,y) = N_2(\mathrm{Dgm}_1(d_{s_{max}}|_W)); \tag{28}$$

$$t_{22}(x,y) = N_3(\mathrm{Dgm}_1(d_{s_{max}}|_W)). \tag{29}$$

Fast algorithm. The general approach to computing persistence extends the classic matrix reduction algorithm for computing homology; see [1] and [12]. For 2-dimensional images, there are shortcuts we can take that improve the worst-case running time from m^3 to $m \log m$, where m is the number of simplices in the filtration. For a 2-dimensional image, the number of simplices in the adaptive triangulation is a small constant times the number of pixels in the image. To explain how the computations are done, we assume that $\alpha_1, \alpha_2, \ldots, \alpha_m$ is the sequence of simplices sorted by maximum function value of the vertices. In other words, if $i < j$, then the maximum function value of the vertices of α_i is less than or equal to the maximum function value of the vertices

of α_j. In case of a tie, we list simplices with smaller dimension first. Note that this implies that $K_j = \{\alpha_1, \alpha_2, \ldots, \alpha_j\}$ is a complex, for each $1 \leq j \leq m$. Indeed, if α belongs to K_j, then the faces of α are listed before α and therefore belong to K_j as well. To compute the 0-th and 1-st Betti numbers of the K_j, we process the list of simplices from beginning to end:

$\beta_0 = \beta_1 = 0$;
`for` $j = 1$ `to` m `do`
  `if` α_j is a vertex `then` $\beta_0 = \beta_0 + 1$
  `elseif` α_j is an edge `then`
    `if` endpoints are connected `then` $\beta_1 = \beta_1 + 1$
                                `else` $\beta_0 = \beta_0 - 1$
    `endif`
  `elseif` α_j is a triangle `then` $\beta_1 = \beta_1 - 1$
  `endif`
`endfor`.

The values of β_0 and β_1 right after the j-th iteration of the `for`-loop are the Betti numbers of K_j. The only non-trivial step in the algorithm is to decide whether the endpoints of the edge α_j belong to the same component or they belong to different components of K_{j-1}. Maintaining a union-find data structure for the components, this step takes only slightly more than constant time; see e.g. [19, Chapter 2].

It is now straightforward to extend the algorithm so it keeps track of the births and deaths of components. Recall that a component dies whenever it merges with another, older component. The older component lives on. By storing the birth values at the root of the corresponding tree in the union-find data structure, we get the persistence intervals in negligible extra time. Using the same algorithm, we get the persistence intervals for the holes by processing the sequence of simplices backwards. Indeed, we can think of each triangle as a dual vertex, of each edge as a dual edge, and of each vertex as a dual triangle. Writing α_j^* for the dual of α_j, we get a dual filtration consisting of complexes $K_j^* = \{\alpha_{j+1}^*, \ldots, \alpha_{m-1}^*, \alpha_m^*\}$. The components of K_j correspond to the holes of K_j^*, and the holes of K_j correspond to the components of K_j^*. Of course, for this to be strictly correct, we need to compactify $\mathbb{M}$ to a sphere, which we do by adding a 2-dimensional cell that represents the outside.

2.3 Geometric Analysis

To have a comparison, we also implement the analysis for a set of geometric features, as we now explain. It should be mentioned, however, that it is impossible to draw a line that separates geometry from topology. We call the new set of features geometric because the emphasis shifts from connectivity to length and area.

Simple features. First, the image is turned into a binary image through local comparisons. We write $\mathcal{B}_r^k$ for the operator that produces this binary image, where r and k have the same meaning as in the definition of the median filter, $\mathcal{M}_r^k$. Applied to an image, f, it produces a binary image,

$$\mathcal{B}_r^k f(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq \mathcal{M}_r^k f(x,y), \\ 0 & \text{if } f(x,y) < \mathcal{M}_r^k f(x,y). \end{cases} \tag{30}$$

Since there are important dark objects (vessels) as well as important bright objects (elements of surface patterns) in our source images, we use the binarization operator twice, once to the source image and another time to its inverted copy:

$$b_1 = \mathcal{B}_r^k f, \tag{31}$$
$$b_2 = \mathcal{B}_r^k (1 - f), \tag{32}$$

but for different values of the parameters r and k to compensate for the difference in scale of dark and bright objects. Specifically, we use $r = 4$ and $k = 3$ for f and $r = 4$ and $k = 5$ for $-f$. All subsequent processing steps are applied to both binary images independently, and all calculated features are added to the resulting feature vector. We therefore simplify the notation by dropping the index and write $\mathbb{F}_i = b_i^{-1}(1)$ for the *foreground* and $\mathbb{B}_i = b_i^{-1}(0)$ for the *background* of the i-th locally binarized image, where i is either 1 or 2. Note that $\mathbb{F}_1$ and $\mathbb{F}_2$ denote surface pattern and vascular pattern respectively. Besides the foreground and background, we also use the *boundary*, $\partial\mathbb{F}$, which consists of all pixels in the foreground that share at least one edge with a pixel in the background. According to this definition, the boundary has non-zero area, but we will refer to its measure as the *length*. The features are now defined locally, using translates of a small square window, $W = W_r(0,0)$, centered at the origin. As always, we use $r = 70$ for windows in this paper. Writing $W(x,y) = W + (x,y)$ for a translation with center (x,y), we define the *density* of the foreground and of its boundary at a point, and the *average brightness* of the foreground, its boundary, and the background at a point:

$$g_1(x,y) = \text{area}(\mathbb{F} \cap W(x,y))/\text{area}(W), \tag{33}$$
$$g_2(x,y) = \text{length}(\partial\mathbb{F} \cap W(x,y))/\text{area}(W), \tag{34}$$
$$g_3(x,y) = \langle b \rangle_{\mathbb{F} \cap W(x,y)}, \tag{35}$$
$$g_4(x,y) = \langle b \rangle_{\partial\mathbb{F} \cap W(x,y)}, \tag{36}$$
$$g_5(x,y) = \langle b \rangle_{\mathbb{B} \cap W(x,y)}, \tag{37}$$

where $\langle b \rangle_X$ is the average $b(x,y)$ over all points $(x,y) \in X$. We remark that we simplified the description of the above five features by ignoring boundary effects near the sides of the image. These are handled by clipping the window to within the image before we compute the average.

Component features. To prepare the next set of features, we compute the connected components of $\mathbb{F}$, which we denote as c_1, c_2, etc. For every c_i, we compute its *area*, the *length* of its boundary, the *radius* (defined as the maximum distance of a point from the center of mass), the *compact-*

ness, and the *elongation*:

$$\gamma_6(c_i) = \text{area}(c_i), \tag{38}$$

$$\gamma_7(c_i) = \text{length}(\partial c_i), \tag{39}$$

$$\gamma_8(c_i) = \max_{(x,y)\in c_i} \|(x,y) - (x_i, y_i)\|, \tag{40}$$

$$\gamma_9(c_i) = \text{length}(\partial c_i)^2/\text{area}(c_i), \tag{41}$$

$$\gamma_{10}(c_i) = (A + B^{1/2})/(A - B^{1/2}), \tag{42}$$

where (x_i, y_i) is the center of mass of c_i. The last measure – the elongation of a component – computes the aspect ratio of the ellipse that has the same moments. In particular, we set

$$A = \left\langle (x - x_i)^2 \right\rangle + \left\langle (y - y_i)^2 \right\rangle,$$

$$B = \left(\left\langle (x - x_i)^2 \right\rangle - \left\langle (y - y_i)^2 \right\rangle \right)^2 - 4 \left\langle (x - x_i)(y - y_i) \right\rangle^2$$

and compute $\gamma_{10}(c_i)$ using (42); see [14, Chapter 3]. For every measure of the components, we introduce a related feature by averaging over all components that have a non-empty intersection with the window:

$$g_k(x, y) = \langle \gamma_k(c_i) \rangle_{c_i \cap W(x,y) \neq \emptyset}, \tag{43}$$

for $k = 6, 7, 8, 9, 10$. In summary, we thus have twenty features for each sampled point, namely ten each for the two locally binarized images. All features are real numbers.

2.4 Machine Learning

The main method we use to learn is the boosting algorithm, which constructs a strong classifier as a weighted combination of weak classifiers.

Boosting. The general problem of *learning* a concept considers a collection of examples, z_i, each labeled ± 1, where $+1$ means that z_i is an example of the concept, and -1 means that z_i is not an example of the concept. We call this collection the *training set*, which is used to fine-tune parameters of the classifier to be used to predict the classification of examples from a *test set*. A *weak classifier* is an algorithm that is only slightly positively correlated with the correct classification, while a *strong classifier* is one that is arbitrarily well correlated with the correct classification. The idea of *boosting* goes back to a question raised by Michael Kearns, namely whether it is possible to create a strong classifier from a set of weak classifiers, to which Robert Schapire gave an affirmative answer [4].

In many applications of boosting, including the one in this paper, we start with a collection of simple weak classifiers. Writing n for the number of collected features, we note that each object may be interpreted as a point in $\mathbb{R}^n$. A *stump weak classifier* discriminates along a single coordinate simply by deciding whether or not that coordinate of the object is smaller than or larger than some threshold:

$$stump_k(z) = \begin{cases} -1 & \text{if } a_k < s, \\ +1 & \text{if } a_k \geq s, \end{cases} \tag{44}$$

where a_k is either $+1$ or -1 times the k-th coordinate of z, and s is the threshold selected for the stump weak classifier.

Adaboost. Adaboost is an early, and perhaps the most important version of boosting [4]. To explain how it works, we assume that the training set consists of m objects, each a pair (z_i, v_i), with $z_i \in \mathbb{R}^n$ and $v_i \in \{-1, +1\}$. We assign to each object the same initial weight, $w_1(z_i) = \frac{1}{m}$, for $1 \leq i \leq m$. To learn the training set, we fix a number of iterations, T, and we perform the following two steps for $t = 1, 2, \ldots, T$:

1. We train the weak classifier, $h_t \colon \mathbb{R}^n \to \{-1, +1\}$. More precisely, we determine the coordinate, k, the sign $+$ or -1, and the threshold, s, such that the corresponding stump weak classifier given in (44) minimizes the *error*, which we define as

$$\epsilon_t = \tfrac{1}{2} \left(1 - \sum_{i=1}^{m} w_t(z_i) \cdot v_i \cdot h_t(z_i) \right). \tag{45}$$

   Assuming the weights are non-negative and add up to 1, the error is a real number in $[0, 1]$, namely the probability that h_t gives the wrong classification.

2. Setting $\delta_t = \frac{\epsilon_t}{1-\epsilon_t}$, we compute the weight of the weak classifier and we update the weights of the objects in the training set:

$$\omega_t = \tfrac{1}{2} \log \tfrac{1}{\delta_t}, \tag{46}$$

$$w_{t+1}(z_i) = w_t(z_i) \cdot \delta_t^{v_i \cdot h_t(z_i)/2} / Z_t, \tag{47}$$

   where Z_t is chosen to normalize the weights such that $\sum_i w_{t+1}(z_i) = 1$.

The constructed weak classifier is at least as good as its negation, which implies $\epsilon_t \in [0, \frac{1}{2}]$. If $\epsilon_t = 0$, then we can stop because we have found a perfect classifier. If $\epsilon_t = \frac{1}{2}$, then we can also stop because there is no hope for improving the result. To see this, we note that in the first step, we optimize over both signs and all values of k and s, which implies that *all* stump classifiers have a success rate of only 50%. This can happen for very special configurations of points, such as checker-board arrangements with alternating signs.

Since $\epsilon_t \in [0, \frac{1}{2}]$, we have $\delta_t \in [0, 1]$. It follows that in each iteration, the weights of incorrectly classified objects increase and the weights of correctly classified objects decrease. The change is more pronounced for small error and negligible for error close to one half. Finally, we construct the strong classifier as a weighted combination of the weak classifiers:

$$H_T(z) = \text{sign} \left(\sum_{t=1}^{T} \omega_t h_t(z) \right). \tag{48}$$

In practice, we construct strong classifiers also for integers smaller than T, and we choose a number of iterations such that increasing this number does not lead to a significant improvement of the strong classifier.

Importance of features. Since we use heterogeneous geometric and topological features, it is useful to assess the contribution of every feature to the final classification. For the k-th feature, this is the sum, over the weak classifiers that discriminate along the k-th coordinate, of the average difference this weak classifier makes to the updated strong classifier. Writing k_t for the coordinate chosen during the t-th iteration of the algorithm, we can formally define the *importance* of the k-th feature as

$$\mathsf{Imp}(k) = \tfrac{1}{2} \sum_{k_t=k} \left[1 - \sum_{i=1}^{m} w_t(z_i) \cdot H_{t-1}(z_i) \cdot H_t(z_i) \right]. \qquad (49)$$

Here we assume that $H_0(z_i) = 0$. We use the insight into the classification offered by the importance measure to lower the dimension. Observe that the subset of $\ell \geq 2$ most important features is not necessarily the subset of size ℓ that performs best. Indeed, features may be redundant with overlapping expertise.

To select the best subset of ℓ features is a hard problem, which motivates us to use a heuristic knows as the *greedy forward selection algorithm*. Starting with all our features in the *candidate set* and an initially empty *selected set*, the algorithm works in rounds, each time moving a feature from the candidate to the selected set. In each round, the chosen feature complements already selected features optimally in the sense that it improves the classification the most. We stop iterating when there are no features in the candidate set that improve the classification.

Multi-class classification. We use a naive Bayesian approach to combine results of several binary Adaboost classifiers to select among a set $K = \{1, 2 \ldots, L\}$ of more than two classes. Assume now that the objects in the training set are pairs (z_i, u_i), where $z_i \in \mathbb{R}^n$ and $u_i \in K$. Fixing two disjoint but non-empty subsets $P, R \subseteq K$, we obtain a binary classification problem for the pairs (z_i, v_i), where

$$v_i = \begin{cases} -1 & \text{if } u_i \in P, \\ +1 & \text{if } u_i \in R. \end{cases} \qquad (50)$$

Consider a suite of subset pairs (P^j, R^j), for $j = 1, 2, \ldots, J$. The suite (P^j, R^j) is calculated using the following schemes:

ONE-VS-ONE:

$$\forall u_i, u_j \in K,\ u_i \neq u_j : P = \{u_i\},\ R = \{u_j\},\ \text{and } J = C_L^2;$$

ONE-VS-ALL:

$$\forall u \in K : P = \{u\},\ R = K \setminus P,\ \text{and } J = L.$$

We compute binary classifiers, H^j, and use them to produce a multi-class classifier as follows. For an object, z_i, we use superscripts to write $H(z_i) = \left(H^1(z_i), H^2(z_i), \ldots, H^J(z_i)\right)$ for the vector of answers. The *confidence* that z_i belongs to class $C_\ell \in K$ is then defined as the conditional probability that an object with the same vector of classifications belongs to C_ℓ:

$$\mathsf{Conf}(z_i, C_\ell) = \mathsf{Prob}[z \in C_\ell \mid H(z) = H(z_i)]. \qquad (51)$$

Finally, we report the class with maximum confidence as the result of the classification. Recall the definition of conditional probability: $\mathsf{Prob}[A \mid B] = \mathsf{Prob}[A \cap B]/\mathsf{Prob}[B]$, and the related Bayes' Rule:

$$\mathsf{Prob}[A \mid B] = \frac{\mathsf{Prob}[A] \cdot \mathsf{Prob}[B \mid A]}{\mathsf{Prob}[B]}. \qquad (52)$$

Applying this to (51), we get $\mathsf{Conf}(z_i, C_\ell) = \mathsf{Prob}[z \in C_\ell] \cdot X(z_i)$, with

$$X(z_i) = \frac{\mathsf{Prob}[H(z) = H(z_i) \mid z \in C_\ell]}{\mathsf{Prob}[H(z) = H(z_i)]} \qquad (53)$$

$$= \prod_{j=1}^{J} \frac{\mathsf{Prob}[H^j(z) = H^j(z_i) \mid z \in C_\ell]}{\mathsf{Prob}[H^j(z) = H^j(z_i)]}, \qquad (54)$$

where the second line is obtained under the assumption that the binary classifiers are mutually independent. In this case, the probability of agreeing vectors is the product of probabilities of agreeing coordinates. All parts of this formula can be estimated using the statistics of the training set.

2.5 Segmentation

Several types of tissue may be present in one endoscopic image and it is necessary to select and classify all such areas into classes defined by specialists. There are many approaches widely used to solve segmentation problem. In this work segmentation based on classification results is used, when every pixel in the image is classified into one of the classes. In order to speed up computations only specifically selected points (keypoints) are classified, since considering all points in the image is unpractical and would lead to time-consuming computations.

To classify a keypoint we compute features in square window centered at this point. As a result, each keypoint is given a class label. Thus the problem of segmentation consists in expansion of classification results onto all pixels in the image. This step is realized with the use of Voronoi diagram built on top of the used collection of keypoints: all points inside each Voronoi cell are given the class label which was given to the keypoint that generated this cell.

To summarize, we consider the following steps of solving the problem:

1. Selection of points (see Localization paragraph in section 2.1);
2. Computation of features (section 2.2 and section 2.3);
3. Multi-class classification (section 2.4);
4. Segmentation.

3 Results

After introducing our data and the protocol for processing, we discuss the selection of a small subset of most important features, and we present the results obtained by running our algorithm with this subset of selected features.

3.1 Data and Test Setting

Our database contains 90 endoscopy images divided into 25 images with round pits (*oval mucosa*), 31 images with oval surface pattern and vessels inside pits (*tubular surface patterns*), and 34 images without visible surface pattern (*irregular patterns*). As learning algorithms go, this is a very small collection, and we have to make decisions accordingly.

Test protocol. To process the images, we first split images of each class into two: *training* and *test* images. In situations in which the data set is small, it is undesirable to separate training data from test data. In such cases, we evaluate the training quality using cross-validation, which is one of several available approaches to estimate how well the model built on training data performs on unseen data. The *leave-one-out cross-validation* (LOO-CV) method considers all partitions of our 90 images into a training set of size 89 and a test set of size 1. There are 90 such partitions, and we repeat the computations for each. The percentage of correct classifications is of course the ratio of correct over the total number of classifications. We also follow a second, *bootstrap* test protocol. Here, we split the 90 images ten times into a training set of 75 images and a test set of 15 images. Each of these partitions is chosen to reflect a subjectively defined similarity between the two sets, or the opposite namely not to reflect that same similarity. The results give us information on performance differences with and without this similarity. As before, we report the ratio of correct classifications over the total number of classifications.

Over-fitting. A common danger in construction of classifiers is the over-fitting to the available training data. To avoid it, we split windows in images of a training set into 90% of *training windows* and 10% of *control windows*. All windows within images of the test set are *test windows*. The classifier is constructed using the training windows, the construction is monitored using the control windows, and the final classifier is evaluated with the test windows.

To monitor the construction, we keep evaluating the evolving classifier on the control windows, while the test windows have no part in the training. Specifically, we calculate two types of errors of the classifier after each iteration: the percentage of misclassified training windows, and the percentage of misclassified control windows. Figure 3 shows the typical case in which the error for the training windows goes to zero, while the error on the control windows flattens out at a positive value. It is quite possible that the second error not only does not go to zero but even increases after some number of iterations – a sign of over-fitting. In any case, we stop the iteration when this error flattens out or starts to increase.

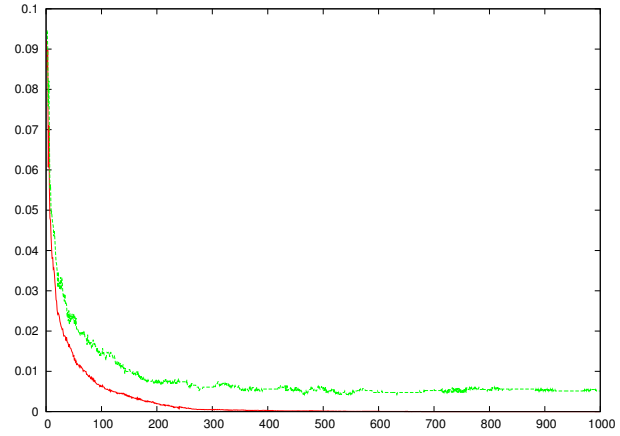

Figure 3: Graphs of the error for the training set and the control set.

3.2 Feature Selection

We have a rather small database but many features, each contributing a coordinate to the high-dimensional vector of features. We therefore aim at reducing the dimension of these vectors, which means we look for the features that contribute significantly to the classification and focus only on those.

Importance. Recall the definition of the importance of a feature given in Section 2.4. For multi-class classification, we use the adaboost algorithm three times: to separate oval from tubular patterns, oval from irregular patterns, and tubular from irregular patterns. We therefore measure the importance three times, displaying the results in the first three histograms of Figures 5 and 4, with the feature-wise maximum displayed in the respective fourth histograms.

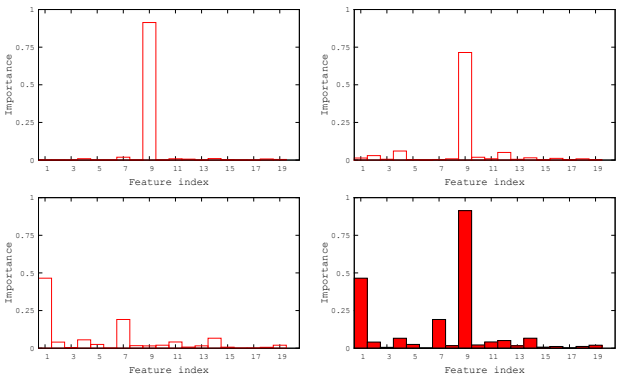

Figure 4: The importance of the geometric features in distinguishing classes. From *left* to *right* and *top* to *bottom*: oval versus tubular, oval versus irregular, tubular versus irregular, the maximum of the three.

On the endoscopy image, bright areas correspond to the elements of the surface pattern and dark areas correspond to the microvascular pattern. The vector of geometric features consists of ten features of the surface pattern structure (calculated using image b_1) and ten features of vascular pattern (calculated using negative image b_2). Note, that the elongation (42) of surface pattern components is the most useful feature when we separate oval and tubular classes or oval and irregular classes. But when we separate tubular and irregular classes, the area (38) of surface pattern components is used in half of the cases.

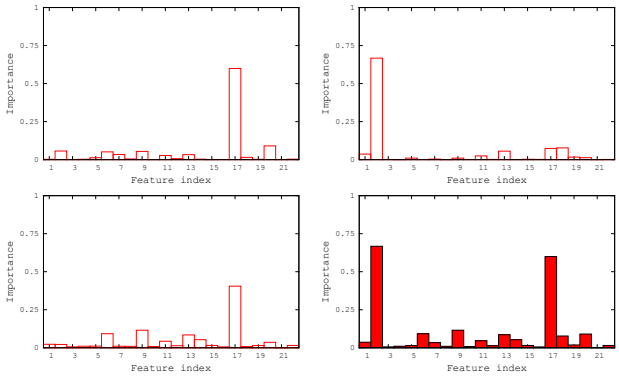

Figure 5: The importance of the topological features in distinguishing classes. From *left* to *right* and *top* to *bottom*: oval versus tubular, oval versus irregular, tubular versus irregular, the maximum of the three.

Winning features. Using the greedy forward selection algorithm explained in Section 2.4, we obtain a select set of five most significant features: two topological and three geometric. To explain the topological features, we mention that we take two types of persistence diagrams: of the image as a map from the pixel array to the range of gray values, and of the signed distance function of the binary image obtained by thresholding the image. The third feature in our selection uses the distribution obtained by integrating the persistence diagram of the image along the persistence axis, while the fifth feature uses the 0-dimensional persistence diagram of the signed distance function obtained for the threshold at which the 1-st Betti number of the binary image is a maximum. The five selected features are:

1. the compactness of the surface pattern, g_9; see (38) and (43) applied to $\mathbb{F}_1$,
2. the perimeter of the surface pattern, g_2; see (34) and (43) applied to $\mathbb{F}_1$,
3. the kurtosis of the histogram of the 0-th persistence diagram of the image, t_{12}; see (19),
4. the average of area of the components of the vascular pattern, g_6; see (38) and (43) but applied to $\mathbb{F}_2$,
5. the first norm of the 0-th persistence diagram of the signed distance function, t_{17}; see (24).

In short, $g_9, g_7, t_{12}, g_{16}, t_{17}$ have been selected in this sequence. By comparison, the five most important features are $g_9, t_2, t_{17}, g_1, g_7$; see Figures 4 and 5. The two lists share only three features, including the most important, which is selected in the first round. The two least important selected features, t_{12} and g_{16}, have surprising low measure, which suggests that they can distinguish some rare but difficult cases. Note also that the most important topological feature, t_2, has not been selected. Its importance is based on distinguishing oval from irregular patterns, where it is dominated by g_9. This confirms that geometric and topological features differ in emphasis while they retain similarities.

3.3 Final Classification

In Figure 6, we show two examples of images and the results of our classification algorithm. The result are obtained using the five selected features as explained above. The first image has two different types of surface patterns, while the second image has a oval pattern with physiological artefacts around. These artefacts are the cause of the partial misclassification we see in this example.

We present the percentages of correct classifications in two histograms, the left one in Figure 7 for the LOO-CV protocol and right one for the bootstrap protocol. Each vertical bar in the histograms represents an interval of percentages (marked on the horizontal axis) and shows the fraction of images with this percentage of correctly recognized neighborhoods. As we can see, most images have between 90 and

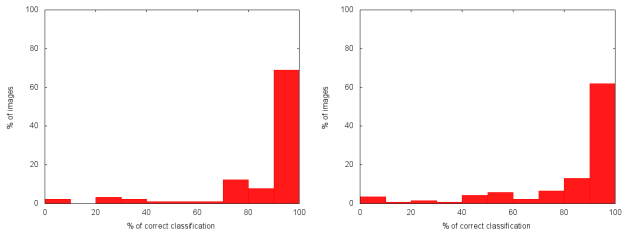

Figure 7: From *left* to *right*: the histograms of correctly classified neighborhoods for the LOO-CV and the bootstrap protocols.

100 percent correctly recognized neighborhoods. The average of correctly recognized neighborhoods is 89 percent, with a variance of 4.4 percent. The results obtained with the bootstrap method are very similar; see the right histogram in Figure 7. Moreover, the results with and without similarity between the training and test sets are almost the same, and we show the average rates from all tests. This implies that our features give stable results on unseen data.

4 Discussion

In this paper, we present our classification results of NBI-ME images of the stomach. Apart from using novel topological features derived from the persistence diagrams of the images, we take a conventional approach using image process-

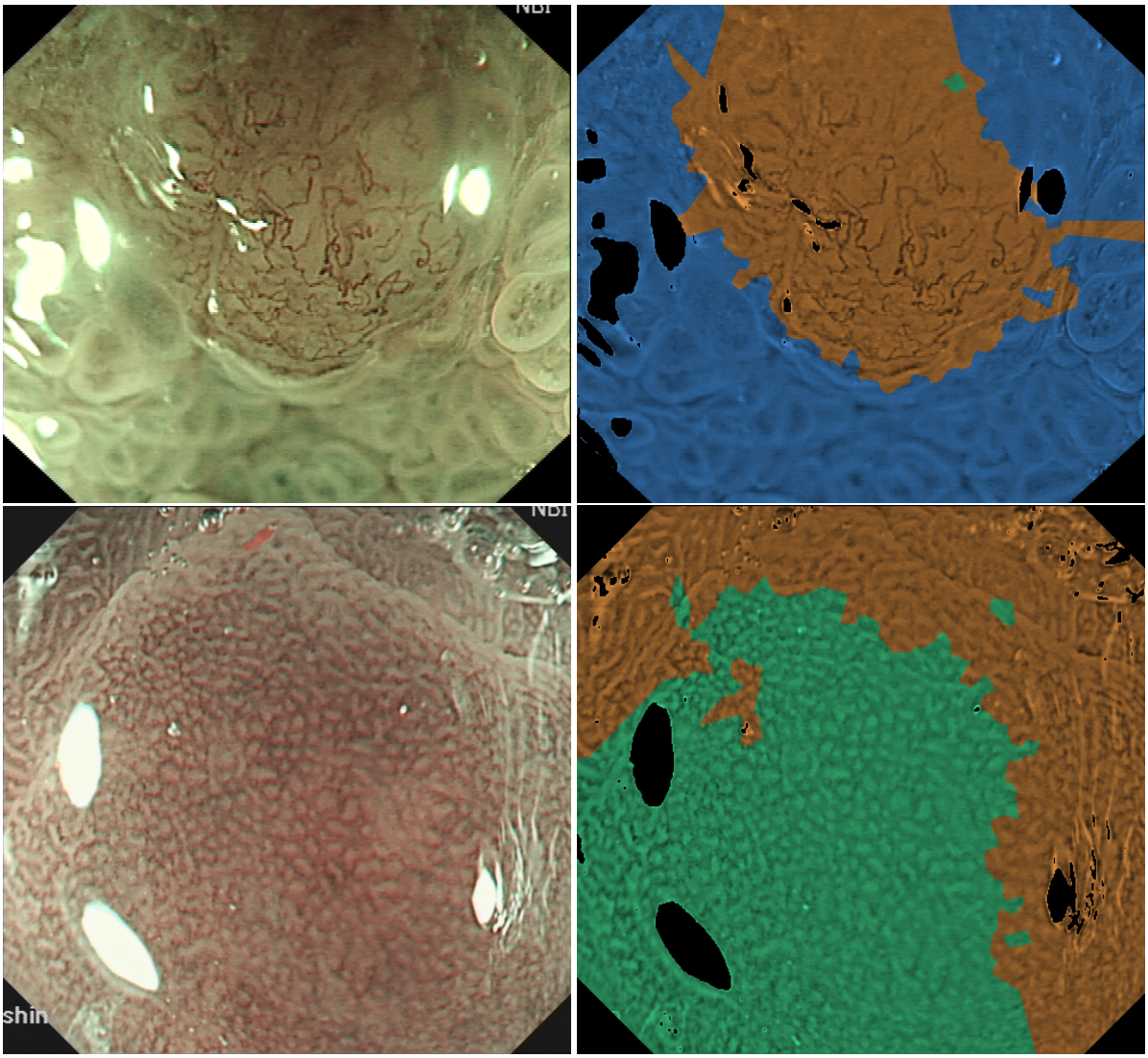

Figure 6: The colors *blue*, *brown*, *green* corresponds to the tubular pattern, irregular pattern, oval pattern, in this sequence. *Top-right:* the correctly colored tubular and irregular patterns on the image to its *left*. *Bottom-right:* the correctly colored oval pattern and incorrectly colored irregular pattern on the image to its *left*.

ing techniques to extract features and learning algorithms to distill this information to an automatic classification. With only a small database of images available, our algorithm is able to make local classifications of imaged tissue with a high success rate near 90%. To the best of our knowledge, there are no similar results available elsewhere.

There are many directions for further research. On the processing front, it is interesting to compare the effectivity of our geometric-topological features with others, for example the local binary patterns as used in [6]. On the technology front, it is useful to study the applicability of our approach to other input modalities, for example low-magnification video. Finally, it is important to extend our results to other organs of medical interest, such as the colon or the esophagus.

References

[1] H. Edelsbrunner and J. L. Harer. *Computational Topology. An Introduction.* Amer. Math. Soc., Providence, Rhode Island, 2010.

[2] H. Edelsbrunner and O. Symonova. The adaptive topology of a digital image. *In* "Proc. 9th Internat. Sympos. Voronoi Diagrams Sci. Engin., 2012", 41–48.

[3] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory Comput.* **8** (2012), 415–428.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application of boosting. *J. Comput. Sys. Sci.* **55** (1997), 119–139.

[5] M. Häfner, A. Gangl, M. Liedlgruber, A. Uhl, A. Vécsei and F. Wrba. Endoscopic image classification using edge-based features. *In* "Proc. 20th Internat. Conf. Pattern Recognition, 2010", 2724–2727.

[6] M. Häfner, A. Gangl, M. Liedlgruber, A. Uhl, A. Vécsei and F. Wrba. Pit pattern classification using extended local binary patterns.

In "Proc. 9th Internat. Conf. Inform. Techn. Appl. Biomed., 2009", 1–4.

[7] S. Har-Peled. *Geometric Approximation Algorithms.* Amer. Math. Soc., Providence, Rhode Island, 2011.

[8] T. Ishitani and M. Sato. Contrast-to-gradient method for the evaluation of image resolution in scanning electron microscopy. *J. Electron Microscopy* **51** (2002), 369–382.

[9] R. O. Kuvaev, S. V. Kashin, V. A. Kapranov, H. Edelsbrunner, M. L. Myachin, O. A. Dunaeva and A. I. Rusakov. Novel computer technologies for the prediction of histological structure in the stomach. *Evidence-based Gastroenterology* **1** (2013) 3–13.

[10] R. Kwitt, M. Häfner, A. Gangl, F. Wrba and A. Vécsei. Predicting the histology of colorectal lesions in a probabilistic framework. *In* "Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition, 2010", 103–110.

[11] R. Kwitt, N. Vasconcelos, N. Rasiwasia, A. Uhl, B. Davis, M. Häfner and F. Wrba. Endoscopic image analysis in semantic space. *Medical Image Analysis* **16** (2012), 1415–1422.

[12] J. R. Munkres. *Elements of Algebraic Topology.* Perseus, Cambridge, Massachusetts, 1984.

[13] S. Saito, H. Tajiri, T. Ohya, T. Nikami, H. Aihara and M. Ikegami. Imaging by magnifying endoscopy with NBI implicates the remnant capillary network as an indication for endoscopic resection in early colon cancer. *Internat. J. Surg. Oncol.* **2011** (2011), Article ID 242608, doi: 10.1155/2011/242608.

[14] L. G. Shapiro and G. Stockman *Computer Vision.* Prentice Hall, Upper Saddle River, New Jersey, 2001.

[15] L. Song, D. Adler, J. Conway, D. Diehl, F. Farraye, S. Kantsevoy, R. Kwon, P. Mamula, B. Rodriguez, R. Shah and W. Tierney. Narrow band imaging and multiband imaging. *Gastrointest. Endosc.* **67** (2008), 581–589.

[16] M. Sonka, V. Hlavac and R. Boyle. *Image Processing, Analysis and Machine Vision.* Second edition, PWS Publishing, Pacific Grove, California, 1999.

[17] T. Stehle, R. Auer, S. Gross, A. Behrens, J. Wulff, T. Aach, R. Winograd, C. Trautwein and J. Tischendorf. Classification of colon polyps in NBI endoscopy using vascularization features. In *Medical Imaging 2009: Computer-Aided Diagnosis*, eds. N. Karssemeijer and M. L. Giger, SPIE **7260**, 2009.

[18] Y. Takemura, S. Yoshida, S. Tanaka, K. Onji, S. Oka, T. Tamaki, K. Kaneda, M. Yoshihara and K. Chayama. Quantitative analysis and development of a computer-aided system for identification of regular pit patterns of colorectal lesions. *Gastrointest. Endosc.* **72** (2010), 1047–1097.

[19] R. E. Tarjan. *Data Structures and Network Algorithms.* SIAM, Philadelphia, Pennsylvania, 1983.

A Notation

$b; f : \mathbb{M} \to [0,1]$	original; brightness equalized image
$W_r(x,y)$	square window with radius r
$\mathcal{M}_r^k b : \mathbb{M} \to [0,1]$	median filter operator
$\mathcal{B}_r^k f : \mathbb{M} \to [0,1]$	local binarization operator
$b_1 = \mathcal{B}_n^k f$	binary image for light features
$b_2 = \mathcal{B}_n^k (1-f)$	binary image for dark features
$p \in N \subseteq M$	points selected from $\mathbb{M}$
$\mathbb{M}_s = f^{-1}[0,s]$	sublevel set
$\beta_p(\mathbb{M}_s)$	p-th Betti number
φ	homology class (component or hole)
$H_{p,i} = H_p(\mathbb{M}_{s_i})$	p-th homology group of sublevel set
$\mathrm{Dgm}_p(f)$	p-th persistence diagram
N_q	q-th norm of persistence diagram
m_1, m_2, m_3, m_4	moments of distribution
$\mu = m_1; \sigma^2 = m_2$	mean; variance
$K_j = \{\alpha_1, \ldots, \alpha_j\}$	j-th complex in filtration
$K_j^* = \{\alpha_{j+1}^*, \ldots, \alpha_m^*\}$	j-th dual complex in filtration
$d_s : \mathbb{M} \to \mathbb{R}$	signed distance function
$W(x,y)$	square window with center (x,y)
$\mathbb{F}, \mathbb{B}$	foreground, background
c_i	connected components
$t_k(x,y)$	k-th topological feature
$g_k(x,y)$	k-th geometric feature
$\gamma_k(\mathbb{F}_j)$	k-th measure of j-th component
$(z_i, v_i), (z_i, u_i)$	object/example in training set
$stump_k, h_t; H_T$	stump weak classifier; strong classifier
ϵ_t	error
$\omega_t; w_t(z_i)$	weight of weak classifier; of object
$C_\ell \in K; P, R \subseteq K$	family of classes, disjoint subsets
$\mathsf{Imp}(k)$	importance of k-th feature
$\mathsf{Conf}(z_i, C_\ell)$	confidence that z_i is in C_ℓ
$1 \le i \le m$	index over objects
$1 \le k \le n$	index over features/dimensions
$1 \le j \le J$	index over partitions of K
$1 \le \ell 1 \le L$	index over classes in K
$1 \le t \le T$	steps in adaboost

Table 2: Notation used in this paper.

B To do or think about

- To make the two histograms in Figure 7 easier to compare, it would be good to change the vertical scale to percentage.
- The behavior of the greedy algorithm is interesting. We may try variants (such as taking the maximum instead of the average of the importance of a feature during different runs) to see whether some such variants give better overall performance.