International Journal of Combinatorial Optimization Problems and Informatics, 13(3), Sep-Dec 2022, 2-16. ISSN: 2007-1558.

www.editada.org

Machine Learning-based Formulas for Computing the Euler Characteristic of Binary Images

Fernando Arce[1], *Wilfrido Gómez-Flores*[2*], *Humberto Sossa*[3,4] and *Uriel Escalona*[3]

[1] Centro de Investigaciones en Óptica A.C., Loma del Bosque 115, Lomas del Campestre, León 37150, Guanajuato, México.
[2] Centro de Investigación y de Estudios Avanzados del IPN, Unidad Tamaulipas, Parque TECNOTAM, 87130, Ciudad Victoria, Tamaulipas, Mexico.
[3] Instituto Politécnico Nacional - CIC, Av. Juan de Dios Batiz S/N, Gustavo A. Madero, 07738 México City, México.
[4] Tecnológico de Monterrey, Campus Guadalajara. Av. Gral. Ramón Corona 2514 Zapopan, Jalisco. 45138, México.

farce@cio.mx, wilfrido.gomez@cinvestav.mx, humbertosossa@gmail.com, jgonzaleze2017@cic.ipn.mx

Abstract. The computer vision community has proposed numerous formulations for object description based on human perceptivity and vast knowledge of the problem domain. In order to reduce human intervention, deep learning techniques are widely used to learn features automatically. However, they lack the property of explainability; that is, a human being understands the meaning of all the parts that make up the calculation of a feature. In this paper, we empirically show how Artificial Intelligence can automatically discover explainable formulations of a topological feature for binary images called the Euler characteristic. The training images are represented by bit-quad patterns, and a single-layer artificial neural network automatically learns the optimal combination of bit-quads to provide valid formulations to correctly calculate the Euler characteristic. We report the results on binary images of different complexities and sizes and compare them with state-of-the-art machine learning algorithms. Finally, we present 14 new equations to calculate the Euler number never reported in literature.

Keywords: Artificial neural network, bit-quads, combinatorial problem, Euler characteristic

Article Info
Received June 27, 2022
Accepted August 16, 2022

1. Introduction

In computer vision applications, calculating numerical features to describe and recognize objects is a primary step. It is well known that such features should be invariant to image transformations and discriminant among different object classes [1].

Commonly, features for object description are designed by human engineering based on the designer's intuition, perceptivity, and expertise using prior knowledge of the problem domain. These kinds of features are usually named hand-crafted features. Typically, these features have the property of explainability; that is, a human being can understand the meaning of all the parts that make up the calculation of a feature [2]. However, the design of invariant and discriminant hand-crafted features can take a long time to test different operators and image representations to obtain a satisfactory result.

In order to reduce human intervention in the manual formulation of invariant and discriminant features, deep learning-based methods for automatic feature extraction have become popular (e.g., deep neural networks). However, their lack of explainability is emerging as their critical deficiency. Notice that explainability is recognized nowadays as one of the leading research fields in Artificial Intelligence (AI) [3], [4].

In this study, we focus on the Euler characteristic of a binary image, which is a relevant topological feature defined as [1], [5]

$$e = c - h \tag{1}$$

where c is the number of connected components, and h is the number of holes in the image. The Euler characteristic is invariant under linear and nonlinear geometric transformations of the image as long as the objects' topology is preserved.

The Euler characteristic has been widely used for object recognition in many applications. In [6], for example, the Euler characteristic has been utilized to recognize industrial parts. In [7], the same topological feature has been employed for real-time image thresholding. It has also been applied in object number counting in [8], real-time Malayan license plate recognition in [9], digit recognition from pressure sensor data in [10], gender recognition from offline handwritten signatures in [11], image description [12], gender discrimination from offline Hindi signature in [13]. In [14], the Euler characteristic has been used for character recognition.

Several image representations have been used to calculate the Euler characteristic in 2D digital images, including morphological skeletons [15], vertex chain codification [16], contact perimeter [17], and bit-quad patterns [5]. The basic idea in these approaches is to describe the image with primitive features (e.g., edges, nodes, terminal points, etc.) and used them in a hand-crafted formulation to compute the Euler characteristic. Here, we chose the bit-quad patterns because they are easy to compute since it is only required to count the number of coincidences of certain bit-quads in the image. Next, equations like Gray's formulations calculate the Euler characteristic from bit-quads counts [18]. So far, all the formulations to compute the Euler characteristic have been found by human engineering, analyzing distinct solutions manually.

Although the calculation of the Euler characteristic is well-established in the computer vision community (as surveyed in Section 2), we are interested in using AI to automatically discover formulations previously created by human engineering and new ones. The goal is to demonstrate the feasibility of artificially creating new formulations that preserve the property of explainability, which means saying formulas to compute, in this case, the Euler characteristic as proof of concept.

In this paper, we use a single-layer artificial neural network (ANN) to learn automatically new bit-quad-based formulations to compute the Euler characteristic of 2D binary images. We chose this neural structure because the calculation of the Euler characteristic involves a linear combination of certain bit-quads. Hence, the hypothesis is that the learned weights reveal the relevance of some bit-quads patterns to compute this topological feature. The efficiency of artificially discovered formulations is tested on binary images of different sizes and complexities. Furthermore, we demonstrate that these formulations maintain invariance under different image geometric transformations. Besides, we compare the efficacy of our approach with other machine learning algorithms to predict the Euler characteristic, including multilayer perceptron, support vector regression, and convolutional neural network. Finally, it is worth mentioning that the proposed approach can be potentially extended to other image representations to learn explainable features.

The rest of the paper is organized as follows. In Section 2, a review of related works is presented. Section 3 explains how two single-layer ANNs, for 4- and 8-connectivity cases, are learned to find optimal bit-quads sets for computing the Euler characteristic of binary images. Section 4 is oriented to illustrate the experimental setup to evaluate the proposed approach. Section 5 shows the resulting expressions found by ANNs to compute the Euler characteristic after learning. These expressions are also evaluated over distinct sets of binary images, and a comparison with machine learning algorithms to predict the Euler characteristic is shown. Lastly, Section 6 is focused on concluding and enumerating different trends for further research.

2. Related work

In the literature, several methods to compute the Euler characteristic of binary images have been reported based on geometric properties, graph theory, morphological operators, bit-quads, and Morse operators [16], [17], [26]–[35], [18], [36]–[43], [19]–[25]. Also, efficient algorithms for computing the Euler characteristic have been proposed [20], [25], [27], [40], [42], [44].

Notably, the so-called bit-quads have been widely used to compute the Euler characteristic of 2D binary images [8], [18], [46], [47], [19], [23]–[25], [27], [39], [42], [45]. Table 1 depicts the 16 possible bit-quad matrices used to obtain different formulations to compute the Euler characteristic.

The most common way to calculate the Euler characteristic of a binary image $I(x, y) \in \{0,1\}$ in terms of bit-quads is [5], [18]

$$4 - \text{connected case:}\, e_4 = \frac{1}{4}(\#Q_2 + \#Q_3 + \#Q_5 + \#Q_9 - \#Q_8 - \#Q_{12} - \#Q_{14} - \#Q_{15} + 2\#Q_7 + 2\#Q_{10}) \tag{2}$$

$$8 - \text{connected case:}\, e_8 = \frac{1}{4}(\#Q_2 + \#Q_3 + \#Q_5 + \#Q_9 - \#Q_8 - \#Q_{12} - \#Q_{14} - \#Q_{15} - 2\#Q_7 - 2\#Q_{10}) \tag{3}$$

where the operator # counts the number of times that the bit-quad $Q\{\bullet\}$ appears in the image $I(x, y)$. Notice that equations 2 and 3 use ten bit-quad patterns, requiring ten comparisons on each image pixel.

For reducing the number of bit-quads counts, [8] proposed two formulations for 4- and 8-connectivity that use three bit-quads as

$$e_4 = \#Q_2 - \#Q_8 + \#Q_{10} \tag{4}$$

$$e_8 = \#Q_2 - \#Q_7 - \#Q_8 \tag{5}$$

Likewise, [39] introduced another pair of expressions to compute the Euler characteristic of a 2D binary image as

$$e_4 = \#Q_9 - \#Q_{15} + \#Q_{10} \tag{6}$$

$$e_8 = \#Q_9 - \#Q_7 - \#Q_{15} \tag{7}$$

Some works proposed algorithmic improvements for computing the Euler characteristic based on counting previous bit-quads [42], [48], [49]. These methods allow reducing the run-time for counting bit-quads, but without proposing new equations.

On the other hand, the estimation of the Euler characteristic can be addressed using machine learning algorithms. The goal is to learn a known equation to predict the Euler characteristic (given by equations 6 and 4), where the inputs are the 16 bit-quad patterns shown in Table 1. Several classifiers have been trained for this task, including a Multilayer Perceptron (MLP) [50] that uses the 16 bit-quads and their corresponding -1, 0, and 1 labels to train it. To obtain the Euler number of a binary image, the trained MLP was moved along the image outputing each time a -1, a 0 or a 1. The final Euler number was obtained by summing up these partial quantities. In~\cite [50] and [50] we used a morphological neural network and a support vector machine, respectively, for the same propuse.

Bit-quad number	Matrix form	Binary form
Q_1	$\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$	0000
Q_2	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	0001
Q_3	$\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}$	0010
Q_4	$\begin{bmatrix}0 & 0\\1 & 1\end{bmatrix}$	0011
Q_5	$\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}$	0100
Q_6	$\begin{bmatrix}0 & 1\\0 & 1\end{bmatrix}$	0101
Q_7	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	0110
Q_8	$\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}$	0111
Q_9	$\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$	1000
Q_{10}	$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$	1001
Q_{11}	$\begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}$	1010
Q_{12}	$\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}$	1011
Q_{13}	$\begin{bmatrix}1 & 1\\0 & 0\end{bmatrix}$	1100
Q_{14}	$\begin{bmatrix}1 & 1\\0 & 1\end{bmatrix}$	1101
Q_{15}	$\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}$	1110
Q_{16}	$\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$	1111

Table 1. The 16 possible bit-quad patterns used to compute the Euler characteristic of a 2D image.

Almost all formulations so far described have been manually derived. To the best of our knowledge, only a set of publications by the same authors describe a method to automatically generate image feature formulations in terms of suitable combinations of bit-quads. In [51], the authors propose a collection of pattern sets to compute bit-quad matchings in component tree structures of grayscale images. The patterns are designed to run in linear time by counting each pattern matching per pixel incrementally. This algorithm computes any attribute obtained from a combination of bit-quads such as area, perimeter, and Euler characteristic. Based on this research, in [52], the same group of authors discusses how to count these patterns from the shape tree. First, they show how counting quads in component trees can be used to count them in a tree of shapes by using the node's depth as the value of pixels in a larger and interpolated image representation. Next, they describe an algorithm that uses this image representation. This approach gives exactly the counts for the original image. In short, in [53], the authors review their original method. A novel theoretical background and algorithm correctness intuition are provided. Besides, a novel version of the algorithm shows improvements for run-time execution and precision analysis.

On the other hand, recently, the simulated annealing algorithm to automatically create new bit-quad-based expressions to calculate the Euler characteristic has been proposed [54]. This approach explores combinations of bit-quads to minimize the error between the true Euler characteristic and the artificially calculated. Hence, an optimal expression must produce zero error in validation images. In the end, we found four new expressions that use three bit-quads. A drawback of this approach is its slow convergence to reach optimal solutions. Furthermore, we published a different approach to compute the Euler's number using vertex chain codification [55], which does not use bit-quads or remove features neither remove features.

3. Proposed approach

From the review presented in Section 2, for about 50 years, only three hand-crafted formulations of bit-quads to compute the Euler characteristic have been proposed. The most popular is Gray's formulation that uses ten bit-quads [18]. The other two formulations reduced the number of bit-quad patterns from ten to three [8], [39].

In general, the bit-quad-based Euler characteristic can be expressed by the linear combination

$$e = w_1 \cdot \#Q_1 + w_2 \cdot \#Q_2 + \cdots + w_{16} \cdot \#Q_{16} \tag{8}$$

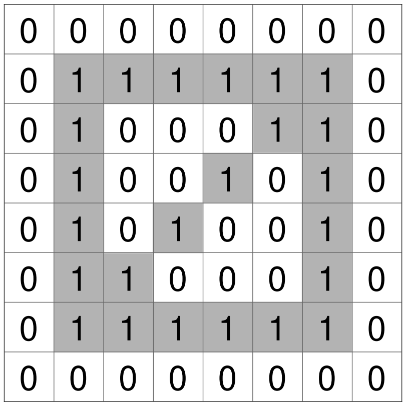
where $w = [w_1, w_2, \cdots, w_{16}]^T$ is an optimum combination of bit-quads in Table 1, where for the jth bit-quad, the coefficient $w_j \in \{-1,0,+1\}$. For instance, Equations 6 and 7 can be expressed in terms of equation 8 using the following optimal solutions:

$e_4 :\ w = [0,0,0,0,0,0,0,0,+1,+1,0,0,0,0,-1,0]^T$,
$e_8 :\ w = [0,0,0,0,0,0,-1,0,+1,0,0,0,0,0,-1,0]^T$.

Finding an optimal combination of bit-quad patterns can be obtained by testing the performance of a plethora of combinations on a set of images. The total number of combinations is $3^q - 1$, where q is the total number of bit-quad patterns. For 16 bit-quads, the total number of combinations is about 43×10^6, which is computationally expensive to evaluate exhaustively.

Since the Euler characteristic in Equation 8 represents the linear combination of specific bit-quad patterns, we propose using a single-layer ANN to automatically learn bit-quad patterns weights through gradient descent to calculate the Euler characteristic. The basic idea is that the higher the magnitude of the weight, the more the relevance of its respective bit-quad pattern. Hence, the highest weights implicitly reveal an expression in terms of equation 8 to compute the Euler characteristic using bit-quads.

Let $X = \{x_1, x_2, \cdots, x_n\}$ be a training dataset, with n observations, described by bit-quads, where the ith sample is a 16-dimensional vector defined by $x_i = [\#Q_{1,i}, \#Q_{2,i}, \cdots, \#Q_{16,i}]^T$, which is associated with an actual Euler characteristic $e_i \in$ Z, such that the couple (x_i, e_i) is formed. Figure 1 shows an example of an 8×8 binary image encoded to a vector with 16 bit-quads, whose Euler characteristic equals -1 since the binary shape is a single connected component with two holes. Notice that this type of image encoding enables to describe images of any size.



0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

$$\mathbf{x} = [2, 3, 1, 7, 1, 7, 3, 1, 3, 0, 7, 2, 7, 2, 1, 2]^T$$

Figure 1. Binary image encoded by 16 bit-quads to form the vector x.

Since the Euler characteristic in Equation 8 can be computed by the inner product of weights and bit-quad counts, we propose the single-layer ANN with a linear activation function, shown in Figure 2, to find relevant weights using the learning process in Algorithm 1. Notice that each bit-quad pattern is associated with a learnable weight.

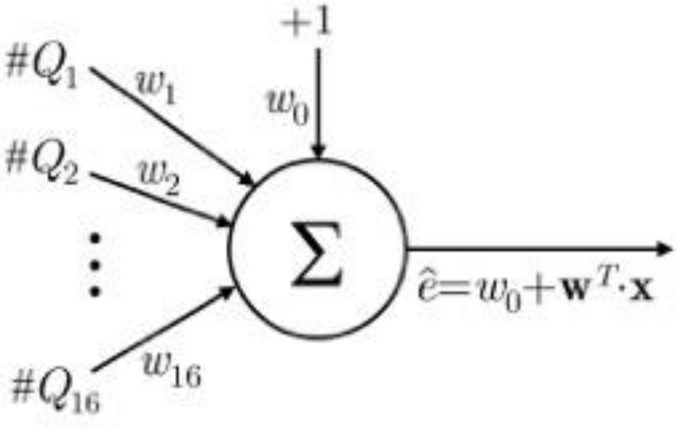


Figure 2. The proposed single-layer ANN with a linear activation function to find the optimal bit-quads to compute the Euler characteristic of 2D binary images.

Algorithm 1. Training process by gradient descent.

```
Input: Training set $(x_i, e_i)$, $i = 1, \dots, n$; learning rate $\lambda$
$\widetilde{x_\iota} \leftarrow [1, x_i^T]^T, \forall_i$ //Augment for bias
$\boldsymbol{w} \leftarrow w_j, j = 0,1, \dots, 16$ // Intialize weights
Wile stop criterion not reached Do
    $\Delta w_j \leftarrow 0, \forall_j$
    For each training sample $(\widetilde{x_\iota, e_i})$ Do
        $\widehat{e_\iota} \leftarrow w^T \cdot \widetilde{x_i}$ // Predict Euler characteristic
        $\frac{dJ(w)}{dw_j} \leftarrow -(e_i - \widehat{e_\iota})\#Q_{j,i}$ // Compute gradients
        $\Delta w_j \leftarrow \Delta w_i + \lambda \frac{dJ(w)}{dw_i}$ // Accumulate
    $w_j \leftarrow w_j - \Delta w_j$ // Update weights
Output: Weight vector $\boldsymbol{w}$
```

We used gradient descend to learn the proposed single-layer ANN because it is numerically stable and avoids working with large matrices used in other methods like least squares based on the Moore-Penrose pseudoinverse matrix.

The loss function is the mean squared error (MSE) defined as

$$J(w) = \frac{1}{2}\sum\nolimits_{i=1}^{n}(e_i - \widehat{e_\iota})^2 \tag{9}$$

where e_i and $\widehat{e_\iota}$ are the actual and estimated Euler characteristics, respectively, of the ith training sample.

4. Experimental setup

Three experiments evaluate the proposed approach. In the first one, two single-layer ANNs based on the architecture in Figure 2 are learned for the 4-connected and 8-connected cases, respectively. The training algorithm uses a learning rate of~0.001 and~2500 epochs. Both ANNs are trained using 60 synthetic images manually generated of 10×10 pixels, as shown in Figure 3. The Euler characteristic of training images is in the range $[-4,8]$. From the learned single-layer ANN weights, expressions to calculate the Euler characteristic are inferred by rounding their values. Hence, zero weights indicate that the related bit-quad pattern is irrelevant and is removed from the calculation; otherwise, the bit-quad pattern is part of the final expression.

The second experiment evaluates the expressions found by ANNs to correctly calculate the Euler characteristic on 2D binary images of different sizes and complexities. A dataset with 11,500 binary images is used for this evaluation, considering the following sizes for each image: 24×24, 48×48, 96×96, and 192×192 pixels, as shown in Figure 4. The actual Euler characteristic of each image is calculated with Gray's equations for 4- and 8-connectivity cases [5]. The Euler characteristics of images are in the range $[-35,23]$. The mean discrepancy between the actual and the estimated Euler characteristic is measured as

$$D = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(e_i \neq \hat{e_i}) \tag{10}$$

where $\mathbf{1}(\cdot)$ is an indicator function. Additionally, for evaluating the robustness to geometric transformations, images of 320×240 pixels with a single object subjected to different geometric transformations are used.

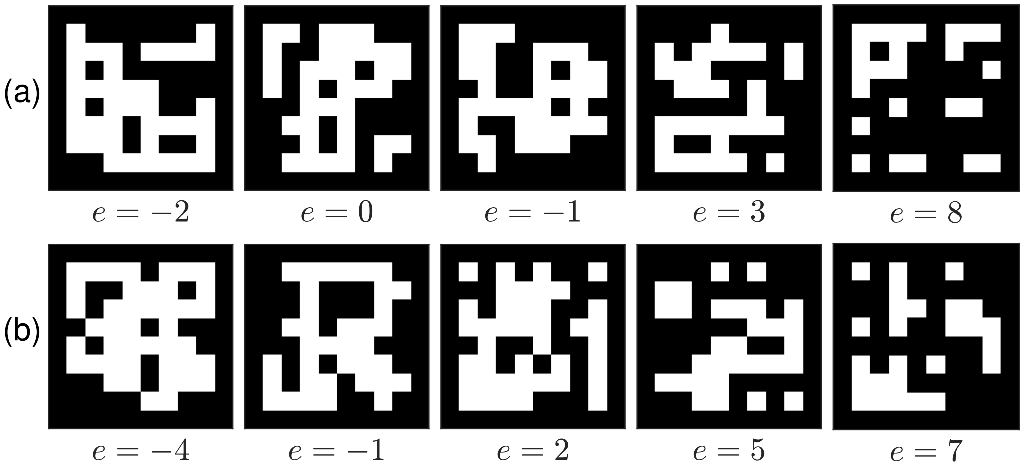


Figure 3. Examples of synthetic images of size 10×10 pixels used to train the proposed single-layer ANNs. Below each image, the actual Euler characteristic is shown for (a) 4-connectivity and (b) 8-connectivity cases.

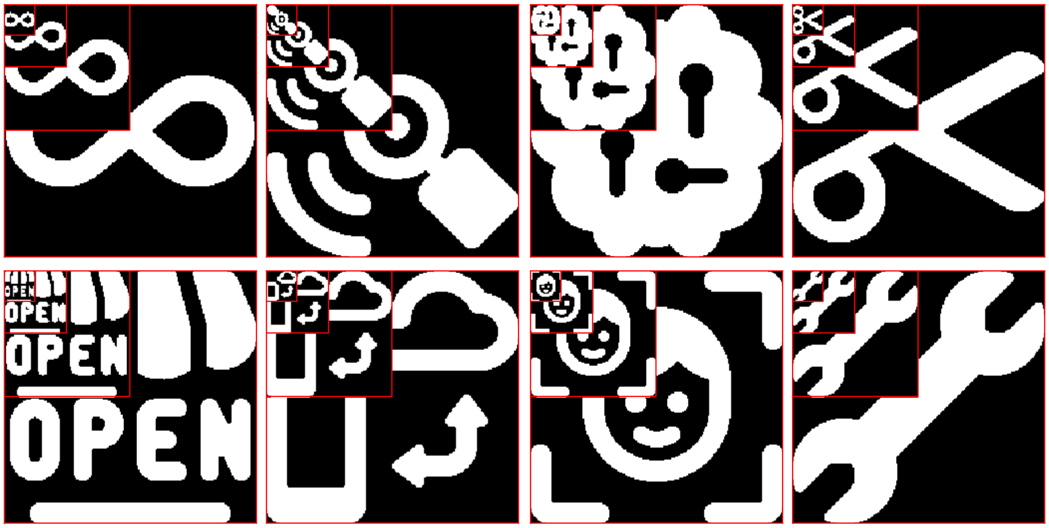


Figure 4. Examples of binary images with four different sizes: 24×24, 48×48, 96×96, and 192×192 pixels. This example overlaps images from the smallest to the largest size to visualize the proportions.

The third experiment evaluates the capability of four distinct machine learning (ML) algorithms to predict the Euler characteristic correctly: proposed single-layer ANN, multilayer perceptron (MLP) with one hidden layer, support vector regression (SVR) with Gaussian kernel, and convolutional neural network (CNN). A dataset with 11,500 binary images of 96×96 pixels is used for training the models, considering 4- and 8-connectivity cases. The k-fold cross-validation method (with $k = 10$) is used to create disjoint training and test sets.

For the single-layer ANN, MLP, and SVR, the input vector corresponds to the 16 bit-quad patterns calculated from a binary image. In MLP and SVR, a tuning process is performed to find the hyperparameters that minimize the mean squared error. For MLP, the number of hidden units is searched in the range $[2{,}16]$, and for SVR, the penalization cost C and the kernel bandwidth γ are in the ranges $[2^5, 2^6, \cdots, 2^9]$ and $[2^{-4}, 2^{-3}, \cdots, 2^4]$, respectively.

For the CNN, the input is the raw binary image. Besides, five CNN configurations are used by varying the number of convolution layers from one to five. The filter sizes are 3×3, and at each convolutional layer, the number of filters is twice of the previous layer, starting with 32 filters. After each convolutional layer, the ReLU activation function followed by max-pooling with a stride of two is used. The final layer is fully-connected with one unit. The CNNs are trained with stochastic gradient descent, with a learning rate of 0.001, a maximum number of epochs of 500, with validation patience of 10 epochs. The loss function is the mean squared error.

The prediction performance is measured by the mean squared error (MSE), mean absolute error (MAE), Pearson correlation coefficient (PCC), and discrepancy (D). The mean value over the 10-folds of cross-validation is calculated. Besides, the wall-clock training time of the algorithms is measured.

All the ML algorithms were implemented in Matlab 2020b (The MathWorks, Natick, MA, USA) and run on a computer platform with an Intel i9-9900K at 3.6GHz, Nvidia GeForce RTX 2070 graphic card, and 64 GB of RAM.

5. Results

The experimental results of the proposed approach are presented in this section. For convenience, discussions are held in this section to interpret the obtained results.

5.1 Single-layer ANN models

The learned weights of the single-layer ANN models for 4- and 8-connectivity cases are shown in Table 2. For both ANNs, the weights were initialized using Xavier normal initializer [56], and Algorithm 1 performed the learning.

In both cases, note that most of the weights are small values close to zero. This behavior suggests that the ANNs found irrelevant bit-quads for estimating the Euler characteristic of 2D images. Contrarily, note that in both 4- and 8-connected cases, three weights approach to $+1$ and -1 values, indicating that the corresponding bit-quads are relevant to compute the Euler characteristic.

For the 4-connected case, the rounded weights reveal that the bit-quads Q_9, Q_{10}, and Q_{15} create the formulation

$$e_4 = \#Q_9 + \#Q_{10} - \#Q_{15} \tag{11}$$

Similarly, for the 8-connected case, the optimum set of bit-quads includes Q_7, Q_9, and Q_{15} to build the expression

$$e_8 = -\#Q_7 + \#Q_9 - \#Q_{15} \tag{12}$$

Hence, according to equations 11 and 12, the single-layer ANN automatically discovered the same formulations in equations 6 and 7 that human beings have intuitively and empirically determined.

It is worth mentioning that by using other weights initialization methods, Lecun [57] and He [58] , and by retraining ten times these architectures, the proposed single-layer ANNs figured out the following new equations never reported in the literature for the 4-connected case:

Weight	**4-connectivity**		**8-connectivity**		**Bit-quad**
	Value	**Rounded**	**Value**	**Rounded**	
W_1	-0.001	0	0.000	0	$\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$
W_2	-0.002	0	-0.003	0	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$
W_3	-0.001	0	0.004	0	$\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}$
W_4	-0.007	0	0.001	0	$\begin{bmatrix}0 & 0\\1 & 1\end{bmatrix}$
W_5	-0.001	0	0.005	0	$\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}$
W_6	-0.003	0	0.000	0	$\begin{bmatrix}0 & 1\\0 & 1\end{bmatrix}$
$\boldsymbol{W_7}$	-0.004	0	**-0.991**	**-1**	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
W_8	-0.001	0	0.004	0	$\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}$
$\boldsymbol{W_9}$	**0.997**	**1**	**0.997**	**1**	$\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$
$\boldsymbol{W_{10}}$	**0.996**	**1**	-0.006	0	$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
W_{11}	-0.002	0	0.001	0	$\begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}$
W_{12}	-0.006	0	-0.002	0	$\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}$
W_{13}	-0.001	0	0.001	0	$\begin{bmatrix}1 & 1\\0 & 0\end{bmatrix}$
W_{14}	-0.003	0	-0.003	0	$\begin{bmatrix}1 & 1\\0 & 1\end{bmatrix}$
$\boldsymbol{W_{15}}$	**-1.000**	**-1**	**-0.994**	**-1**	$\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}$
W_{16}	-0.002	0	0.001	0	$\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$

Table 2. Weights obtained after training the single-layer ANNs using 10×10 2D binary images for the 4- and 8-connected cases. In bold, the most relevant weights and their corresponding bit-quads.

- $\#Q_5 + \#Q_7 - \#Q_{14}$,
- $\#Q_3 + \#Q_7 - \#Q_{12}$,
- $\#Q_2 - \#Q_8 + \#Q_{10}$,
- $\#Q_3 - \#Q_6 + \#Q_7 - \#Q_8 + \#Q_{11} - \#Q_{14} + \#Q_{15}$,
- $\#Q_3 - \#Q_4 + \#Q_7 - \#Q_8 - 2\#Q_{11} + \#Q_{13} + \#Q_{14} + \#Q_{15}$,

and for the 8-connected case:

- $\#Q_5 - \#Q_{10} - \#Q_{14}$,
- $\#Q_3 - \#Q_{10} - \#Q_{12}$,
- $\#Q_5 + \#Q_6 + \#Q_8 - \#Q_{10} - \#Q_{11} - \#Q_{12} - \#Q_{15}$,
- $-\#Q_2 + \#Q_3 + \#Q_8 + \#Q_9 - \#Q_{10} - \#Q_{12} - \#Q_{15}$,
- $\#Q_2 + \#Q_5 - \#Q_8 - \#Q_9 - \#Q_{10} - \#Q_{14} + \#Q_{15}$,
- $\#Q_2 + \#Q_5 + \#Q_6 - \#Q_9 - \#Q_{10} - \#Q_{11} - \#Q_{12}$,
- $\#Q_3 - \#Q_5 - \#Q_6 - \#Q_7 - \#Q_8 + \#Q_9 + \#Q_{11}$,
- $\#Q_3 - \#Q_6 - \#Q_8 - \#Q_{10} + \#Q_{11} - \#Q_{14} + \#Q_{15}$,
- $-\#Q_2 + 2\#Q_3 + \#Q_4 - \#Q_5 - \#Q_6 + \#Q_8 + \#Q_9 - \#Q_{10} + \#Q_{11} - \#Q_{13} - \#Q_{14} - \#Q_{15}$.

5.2 Euler characteristic estimation with artificial expressions

The expressions found by the trained single-layer ANNs were tested on 11,500 binary images of different sizes and objects. The discrepancy index (D) was measured between the estimated and actual Euler characteristics. For all artificially created expressions, the Euler characteristic was correctly estimated for the entire dataset; that is, the mean discrepancy was zero. Figure 5 shows the results of the estimated Euler characteristic on some test images.

Moreover, Figure 6 shows the robustness of artificially generated expressions to geometric transformations, an essential requirement when calculating the Euler characteristic. Note that the trained ANN is applied to each of the three sequences of images and obtained the same output. This observation can be explained as follows. Suppose we have a binary image I_1 with c connected components (or objects) and h holes. Then, I_1 is transformed into another image by applying an image transformation as $I_2 = \Gamma(I_1)$. In this case, Γ can be any geometric transformation: translation, rotation, scale change, affine, projection, and even a combination of several of them. The corresponding bit-quad representations of these two images are x_1 and x_2, respectively. Of course, due to Γ , $x_1 \neq x_1$; notwithstanding, as shown in Figure 6, the Euler characteristic is always the same.

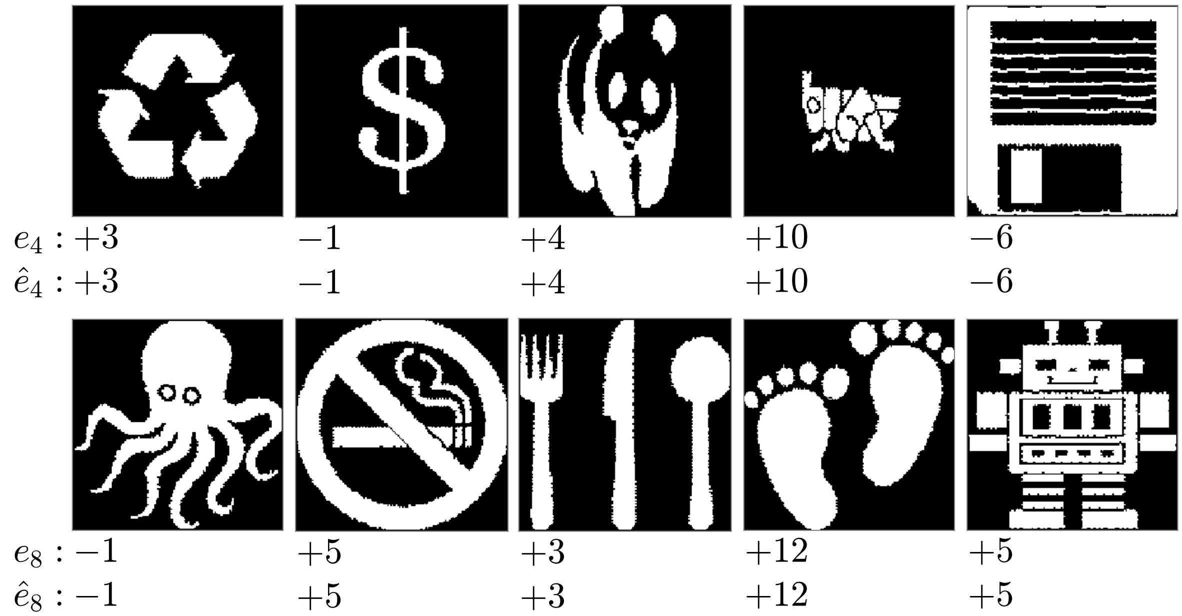


Figure 5. Estimated Euler characteristics for 2D binary images obtained by the expressions in equations 11 and 12.

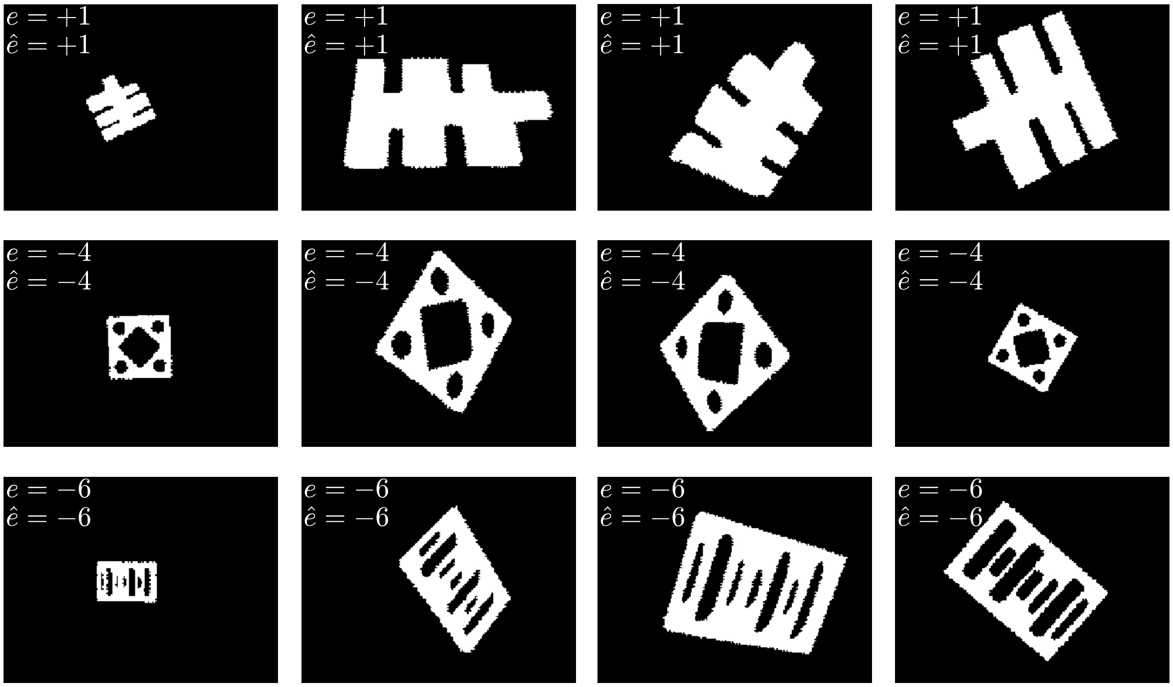


Figure 6. Binary images of 320 × 240 pixels with a single object subjected to different geometric transformations. Both the estimated and the actual Euler characteristics are shown in the upper-left corner.

When we feed x_1 and x_2 to the trained single-layer ANN, we obtain the outcomes $y_1 = w_0 + w^T x_1$ and $y_2 = w_0 + w^T x_2$. If we remove the elements that do not contribute to the computation of the Euler characteristic (i.e., zero weights after rounding as shown in Table 2, for the 4-connected case, we have $y_1 = \#Q_{9,1} + \#Q_{10,1} - \#Q_{15,1}$ and $y_2 = \#Q_{9,2} + \#Q_{10,2} - \#Q_{15,2}$. It is clear that for obtaining the same output of the trained single-layer ANN, it is necessary that $y_1 = y_2$.

For example, let us take the sequence of four images shown in the first row of Figure 6, with the three most representative bit-quads (i.e., $\#Q_9$, $\#Q_{10}$, and $\#Q_{15}$). We have that $x_1 = [69, 0, 68]^T$, $x_2 = [8,0,7]^T$, $x_3 = [140,0,139]^T$, and $x_4 = [66,0,65]^T$ for these four images, where for the i-th image, $x_i = [\#Q_{9,i}, \#Q_{10,i}, \#Q_{15,i}]^T$. This discussion can be formally stated as follows:

Proposition 1. The Euler characteristic e of any binary image is the same after applying any geometric transformation by counting the most relevant bit-quads found by the single-layer ANN.

Proof. **Basis**: Let us consider first and fourth images in the first row of Figure 6, with a single connected object $c = 1$ and no holes $h = 0$, the most representative bit-quads from each of these two images are $x_1 = [69, 0, 68]^T$ and $x_4 = [66,0,65]^T$, respectively. Using the expression found by the single-layer ANN to calculate the Euler characteristic, we obtain that $y_1 = 69 + 0 - 68 = 1$ and $y_2 = 66 + 0 - 65 = 1$, which is true because both images have the same Euler characteristic $e = c - h = 1 - 0 = 1$.

5.3 Euler characteristic prediction with machine learning algorithms

Table 3 shows the results of the proposed single-layer ANNs and three ML algorithms to predict the Euler characteristic for the 4- and 8-connectivity cases. Notably, the single-layer ANN, MLP, and SVR methods outperformed the CNNs. This finding points out that encoding the input binary image to a bit-quad representation allows describing the image's topology such that a single-layer ANN, MLP, and SVR methods can build a mapping function to predict the Euler characteristic accurately.

Besides, considering the discrepancy (D) index, it is notable that both the proposed single-layer ANN and SVR attained precisely zero, which means that the Euler characteristic was correctly predicted for all test images. However, training a single-layer ANN requires 99.999% less computing time than SVR, which is estimated from the training times in the last column of Table 3.

It is worth mentioning that ML algorithms address the Euler characteristic prediction as a regression problem; hence, the outcomes were rounded to obtain an integer value that can be compared with the actual Euler characteristic using the discrepancy index. From a practical viewpoint, obtaining the exact Euler characteristic is crucial since an erroneous prediction implies that the input image has different topology. For instance, if the image has a single object with two holes, the actual Euler characteristic is -1, but if the ML algorithm predicts an Euler characteristic of zero, it means that the object may have a single hole. Hence, the discrepancy is a hard index that measures the ML algorithm's capability to predict the correct Euler characteristic, whereas the MSE, MAE, and CPP measure the approximation degree of the ML algorithm output to the target.

Methods	4-connectivity				8-connectivity				Time(m)
	MSE	MAE	PCC	C	MSE	MAE	PCC	C	
ANN	**0.000**	**0.000**	**1.000**	**0.000**	**0.000**	**0.000**	**1.000**	**0.000**	**0.004**
MLP	0.002	0.030	**1.000**	0.002	0.003	0.033	**1.000**	0.002	2.104
SVR	0.001	0.012	**1.000**	**0.000**	**0.000**	0.012	**1.000**	0.000	76.156
CNN$_1$	3.471	1.850	0.543	0.773	3.415	1.848	0.544	0.771	0.383
CNN$_2$	0.190	0.403	0.980	0.275	0.201	0.409	0.979	0.278	8.500
CNN$_3$	0.072	0.240	0.992	0.109	0.085	0.251	0.991	0.118	9.298
CNN$_4$	0.065	0.244	0.993	0.115	0.086	0.247	0.991	0.111	5.127
CNN$_5$	**0.000**	0.230	0.993	0.097	0.081	0.245	0.992	0.109	5.522

Table 3. Comparison of ML algorithms to predict the Euler characteristic. The mean value of 10-fold cross validation experiments is shown. The last column is the wall-clock time in minutes to train the models. In bold are highlighted the best results.

Concerning CNNs, as expected, the deeper CNN, the higher prediction performance. However, they cannot predict the Euler characteristic correctly in any case. This result is because the convolution layers and max-pooling damage the image topology such that object holes may disappear, and different objects may become merged after these operations. Hence, the learned features cannot describe the actual image topology. Besides, the image representation based on bit-quads allows estimating the Euler characteristic of images of any size, whereas, for CNNs, the input image size should be adjusted to the input layer's size.

6. Conclusions

One of the current research directions in computer vision is automatic feature learning through AI techniques, such as convolutional neural networks. This learning paradigm has reduced human intervention, impacting the time required to design invariant and discriminant features manually. However, learned features lack explainability, which is the property of explaining what is happening such that a human can understand it. Therefore, the automatic design of explainable features is an open research field. In this paper, we demonstrated for the first time that an AI, defined by a single-layer ANN, can be used to automatically learn optimal combinations of bit-quads to estimate the Euler characteristic of 2D binary images.

The proposed solution is explainable, a desirable property of any artificial intelligence-based solution. We found 14 new equations, never reported in the literature, to compute the Euler characteristic by training a single-layer ANN. The learned weights implicitly define the relevance of the certain bit-quad patterns, and we used this information to infer expressions to calculate the Euler characteristic in both 4- and 8-connectivity cases. The experimental results have shown that the estimated Euler characteristics obtained by the trained machines are accurate, regardless of the number of connected components and holes in the image.

Moreover, bit-quads are adequate image features capable of describing the objects' topology such that the Euler characteristic can be calculated exactly, whereas learned features by CNN are inadequate to describe the image topology. Indeed, notably, the single-ANN just required a small dataset with 60 synthetic images of 10×10 pixels to find general formulations to compute the Euler characteristic for images of any size. Contrarily, although the CNNs were trained with thousands of images, they could not learn an adequate representation to compute the Euler characteristic correctly. Besides, the CNNs always require input images of the same size, unlike bit-quad-based representation that can be used for images of arbitrary sizes.

In short, the proposed single-layer ANN is the best model to predict the Euler characteristic correctly in 4- and 8-connectivity cases since gradient descent efficiently finds the optimal solution to the linear combination of bit-quads in a reduced computing time. Additionally, it could be extended to find linear combinations of other image representations to create new features automatically.

Since the obtained results are promising, future work considers training an artificial neural network to estimate the Euler characteristic of binary 3D images. In this case, instead of bit-quads, we should use bit-octos, which are matrices of size $2 \times 2 \times 2$.

Acknowledgments

Authors would like to thank the Instituto Politécnico Nacional and the Centro de Investigaciones en Óptica A.C., for the economic support under projects SIP 20220226 and CONACYT FORDECYT PRONACES 6005 to undertake this research.

Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.

[2] F. Doshi-Velez and B. Kim, "Considerations for evaluation and generalization in interpretable machine

learning," in *Explainable and Interpretable Models in Computer Vision and Machine Learning*, 2018, pp. 3–17.

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, pp. 206–2015, 2019.

[4] G. Raayoni *et al.*, "Generating conjectures on fundamental constants with the Ramanujan Machine," *Nature*, vol. 590, no. 7844, pp. 67–73, Feb. 2021, doi: 10.1038/s41586-021-03229-4.

[5] W. K. Pratt, *Digital Image Processing*. John Wiley {\&} Sons, Inc., 2007.

[6] H. S. Yang and S. Sengupta, "Intelligent shape recognition for complex industrial tasks," *IEEE Control Syst. Mag.*, vol. 8, no. 3, pp. 23–30, 1988.

[7] S. L. and F. G. L., "Real-Time Thresholding with Euler Numbers," *Pattern Recogn. Lett.*, vol. 24, no. 9–10, pp. 1533–1544, Jun. 2003, doi: 10.1016/S0167-8655(02)00392-6.

[8] X. Lin, J. Ji, and Y. Gu, "The Euler Number Study of Image and Its Application," in *2007 2nd IEEE Conference on Industrial Electronics and Applications*, 2007, pp. 910–912.

[9] W. Al Faqheri, S. Mashohor, and S. D. Ehsan, "A Real-Time Malaysian Automatic License Plate Recognition (M-ALPR) using Hybrid Fuzzy," 2009.

[10] A. Paul, N. Bhattacharya, and A. S. Chowdhury, "Digit recognition from pressure sensor data using Euler number and central moments," in *2012 International Conference on Communications, Devices and Intelligent Systems (CODIS)*, 2012, pp. 93–96.

[11] P. Maji, S. Chatterjee, S. Chakraborty, N. Kausar, S. Samanta, and N. Dey, "Effect of Euler number as a feature in gender recognition system from offline handwritten signature using neural networks," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015, pp. 1869–1873.

[12] Q. Zhang, L. Wang, J.-H. Yu, and M. Zhang, "Segmentation-based Euler number with multi-levels for image feature description," *Procedia Comput. Sci.*, vol. 111, pp. 245–251, 2017, doi: https://doi.org/10.1016/j.procs.2017.06.060.

[13] M. Pal, S. Bhattacharyya, and T. Sarkar, "Euler number based feature extraction technique for gender discrimination from offline Hindi signature using SVM BPNN classifier," in *2018 Emerging Trends in Electronic Devices and Computational Techniques (EDCT)*, 2018, p. 16.

[14] Y. R. Matsuoka, G. A. R. Sandoval, L. P. Q. Say, J. S. Y. Teng, and D. D. Acula, "Enhanced Intelligent Character Recognition (ICR) Approach Using Diagonal Feature Extraction and Euler Number as Classifier with Modified One-Pixel Width Character Segmentation Algorithm," in *2018 International Conference on Platform Technology and Service (PlatCon)*, 2018, pp. 1–6.

[15] D.-D.-L. J. L. and S. Humberto, "On the computation of the Euler number of a binary object," *Pattern Recognit.*, vol. 29, no. 3, pp. 471–476, 1996, doi: https://doi.org/10.1016/0031-3203(95)00098-4.

[16] Ernesto Bribiesca, Ulf-Dietrich Braumann, Angel Carrillo-Bermejo, and H. Sossa-Azuela, "An Approach to the Computation of the {{Euler}} Number by means of the Vertex Chain Code," *Comput. Math. Methods Med.*, vol. 2020, no. 5632159, 2020.

[17] E. Bribiesca, "Computation of the Euler number using the contact perimeter," *Comput. \& Math. with Appl.*, vol. 60, no. 5, pp. 1364–1373, 2010, doi: https://doi.org/10.1016/j.camwa.2010.06.018.

[18] S. B. Gray, "Local Properties of Binary Images in Two Dimensions," *IEEE Trans. Comput.*, vol. C–20, no. 5, pp. 551–561, 1971.

[19] L. He, B. Yao, X. Zhao, Y. Yang, Y. Chao, and A. Ohta, "A Graph-Theory-Based Algorithm for Euler Number Computing," *IEICE Trans. Inf. Syst.*, vol. E98.D, no. 2, pp. 457–461, 2015, doi: 10.1587/transinf.2014EDL8155.

[20] M.-H. Chen and P.-F. Yan, "A fast algorithm to calculate the Euler number for binary images," *Pattern Recognit. Lett.*, vol. 8, no. 5, pp. 295–297, 1988, doi: https://doi.org/10.1016/0167-8655(88)90078-5.

[21] J. L. Díaz-De-León and H. Sossa, "On the computation of the Euler number of a binary object," *Pattern Recognit.*, vol. 29, no. 3, pp. 471–476, 1996, doi: https://doi.org/10.1016/0031-3203(95)00098-4.

[22] C. R. Dyer, "Computing the Euler number of an image from its quadtree," *Comput. Graph. Image Process.*, vol. 13, no. 3, pp. 270–276, 1980, doi: https://doi.org/10.1016/0146-664X(80)90050-7.

[23] L. He, Y. Chao, and K. Suzuki, "A new algorithm for labeling connected-components and calculating the Euler number, connected-component number, and hole number," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 3099–3102.

[24] B. Yao, H. Wu, Y. Yang, Y. Chao, and L. He, "An improvement on the Euler number computing algorithm used in MATLAB," in *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, 2013, pp. 1–4.

[25] Y. Chao, S. Kang, B. Yao, X. Zhao, and L. He, "An efficient euler number computing algorithm," in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 1167–1171.

[26] Y. Bin, H. Lifeng, K. Shiying, Z. Xiao, and C. Yuyan, "A New Run-Based Algorithm for Euler Number Computing," *Pattern Anal. Appl.*, vol. 20, no. 1, pp. 49–58, Feb. 2017, doi: 10.1007/s10044-015-0464-4.

[27] H. Lifeng *et al.*, "A Fast Algorithm for Integrating Connected-Component Labeling and Euler Number Computation," *J. Real-Time Image Process.*, vol. 15, no. 4, pp. 709–723, Dec. 2018, doi: 10.1007/s11554-015-0499-1.

[28] A. Imiya and U. Eckhardt, "The Euler Characteristics of Discrete Objects and Discrete Quasi-Objects," *Comput. Vis. Image Underst.*, vol. 75, no. 3, pp. 307–318, 1999, doi: https://doi.org/10.1006/cviu.1999.0791.

[29] O. S. Joshi, "Binary Image Processing for Computation of Connected Components, Image Holes and Euler Number Using Graph Theory," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 2018, pp. 230–234.

[30] M. Kiderlen, "Estimating the Euler Characteristic of a planar set from a digital image," *J. Vis. Commun. Image Represent.*, vol. 17, no. 6, pp. 1237–1255, 2006, doi: https://doi.org/10.1016/j.jvcir.2006.05.001.

[31] X. Lin, Y. Sha, J. Ji, and Y. Wang, "A proof of image Euler Number formula," *Sci. China Ser. F-Information Sci.*, vol. 49, no. 3, pp. 364–371, 2006, doi: https://doi.org/10.1007/s11432-006-0364-8.

[32] Nagel, Ohser, and Pischang, "An integral-geometric approach for the Euler-Poincaré characteristic of spatial images," *J. Microsc.*, vol. 198, no. 1, pp. 54–62, 2000.

[33] L. G. Nonato, A. C. Filho, R. Minghim, and J. Batista, "Morse operators for digital planar surfaces and their application to image segmentation," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 216–227, 2004.

[34] H. Samet and M. Tamminen, "Computing Geometric Properties of Images Represented by Linear Quadtrees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-7, no. 2, pp. 229–240, 1985.

[35] H. Sossa, C.-J. E. V., and D. Zaldivar-Navarro, "Computation of the {{Euler}} Number of a Binary Image Composed of Hexagonal Cells," *J. Appl. Res. Technol.*, vol. 8, pp. 340–350, 2010.

[36] S. H., C.-J. E. B., and Z.-N. D., "Alternative Way to Compute the Euler Number of a Binary Image," *J. Appl. Res. Technol.*, vol. 9, pp. 335–341, 2011, [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-64232011000300006&nrm=iso.

[37] H. Sossa, R. Santiago-Montero, M. Perez-Cisneros, and E. Rubio-Espino, "Computing the {{Euler}} Number of a Binary Image Based on a Vertex Codification," *J. Appl. Res. Technol.*, vol. 11, pp. 360–369, 2013.

[38] H. Sossa, E. Rubio-Espino, R. Santiago, A. López, A. Pena Ayala, and E. V Cuevas Jimenez, "Alternative formulations to compute the binary shape {{Euler}} number," *IET Comput. Vis.*, vol. 8, no. 3, pp. 171–181, 2014.

[39] H. Sossa, Á. Carreón, R. Santiago-Montero, E. Bribiesca-Correa, and A. Petrilli-Barceló, "Efficient computation of the {{{Euler}}} number of a {{2-D}} binary image," in *Advances in Soft Computing*, 2017, vol. 10061, pp. 401–413.

[40] J. Weinstein, "An Improved Algorithm for the Computation of the {{Euler}} Number of a Binary Image Using the Perimeter," *Int. J. Inf. Comput. Sci.*, vol. 7, no. 8, pp. 1–6, 2020.

[41] J. Weinstein, "Computation of the {{Euler}} Number of a Hexagonal Binary Image Using the Perimeter," *Int. J. Inf. Comput. Sci.*, vol. 7, no. 9, pp. 20–28, 2020.

[42] B. Yao, L. He, S. Kang, X. Zhao, and Y. Chao, "A Further Improvement on Bit-Quad-Based Euler Number Computing Algorithm," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 2, pp. 545–549, 2016, doi: 10.1587/transinf.2015EDL8159.

[43] Zhao Zhang, R. H. Moss, and W. V Stoecker, "A novel morphological operator to calculate Euler number," in *Proceedings International Workshop on Medical Imaging and Augmented Reality*, 2001, pp. 226–228.

[44] S. Yang, D. Chen, X. Han, and Y. Chen, "A novel and fast connected component count algorithm based on graph theory," in *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2016, pp. 643–647.

[45] L. He, Y. Chao, and K. Suzuki, "An Algorithm for Connected-Component Labeling, Hole Labeling and Euler Number Computing," *J. Comput. Sci. Technol.*, vol. 28, pp. 468–478, 2013.

[46] B. Yao, L. He, S. Kang, X. Zhao, and Y. Chao, "A New Run-Based Algorithm for {{Euler}} Number Computing," *Pattern Anal. Appl.*, vol. 20, no. 1, pp. 49–58, 2017.

[47] S. Humberto, C. Ángel, and S. Raúl, “Training a multilayered perceptron to compute the Euler number of a 2-D binary image,” in *Training a multilayered perceptron to compute the Euler number of a 2-D binary image*, Jan. 2016, pp. 44–53, doi: 10.1007/978-3-319-39393-3_5.

[48] B. Yao *et al.*, “An Efficient Strategy for Bit-Quad-Based Euler Number Computing Algorithm,” *IEICE Trans. Inf. Syst.*, vol. E97.D, no. 5, pp. 1374–1378, 2014, doi: 10.1587/transinf.E97.D.1374.

[49] B. Yao, L. He, S. Kang, X. Zhao, and Y. Chao, “Bit-Quad-Based Euler Number Computing,” *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 9, pp. 2197–2204, 2017, doi: 10.1587/transinf.2017EDP7012.

[50] H. Sossa, Á. Carreón, R. Santiago, and A. Petrilli-Barcelo, “Support vector machines applied to {2-D} binary image {{Euler}} number computation,” Dec. 2016, pp. 3–8.

[51] D. J. Silva, W. A. L. Alves, A. Morimitsu, and R. F. Hashimoto, “Efficient incremental computation of attributes based on locally countable patterns in component trees,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3738–3742.

[52] D. J. da Silva, W. A. L. Alves, A. Morimitsu, C. F. Gobber, and R. F. Hashimoto, “Incremental bit-quads count in tree of shapes,” 2019.

[53] D. J. Silva, W. A. L. Alves, and R. F. Hashimoto, “Incremental bit-quads count in component trees: Theory, algorithms, and optimization,” *Pattern Recognit. Lett.*, vol. 129, pp. 33–40, 2020, doi: https://doi.org/10.1016/j.patrec.2019.10.036.

[54] W. Gómez-Flores, H. Sossa, and F. Arce, “Finding the Optimal Bit-Quad Patterns for Computing the {Euler} Number of {2D} Binary Images Using Simulated Annealing,” in *Pattern Recognition*, 2021, pp. 240–250.

[55] A. Fernando, S. Humberto, G.-F. Wilfrido, and L. Laura, “Training an Artificial Neural Network to Compute the Euler Number of a 2-D Binary Image based on Vertex Chain Codification,” *Int. J. Comb. Optim. Probl. Informatics*, vol. 13, pp. 4–17, 2022.

[56] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, vol. 9, pp. 249–256, [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html.

[57] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient BackProp,” in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48.

[58] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *CoRR*, vol. abs/1512.0, 2015, [Online]. Available: http://arxiv.org/abs/1512.03385.