International Journal of Combinatorial Optimization Problems and Informatics, 13(1), Jan-April 2022, 4-17. ISSN: 2007-1558.

www.editada.org

Training an Artificial Neural Network to Compute the Euler Number of a 2-D Binary Image based on Vertex Chain Codification

Fernando Arce[1], Humberto Sossa[2,3], Wilfrido Gómez-Flores[4] and Laura Lira [2]

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

fernando.arce.vega@gmail.com, humbertosossa@gmail.com, wilfrido.gomez@cinvestav.mx, laugi2808@hotmail.com

Abstract. So-called Vertex Chain Codes have been widely used to describe the shape of the objects. From these codes, several describing features can be obtained, e.g., the Euler characteristic. In this research, we show how Vertex Chain Codes can be used to train an Artificial Neural Network to compute the Euler characteristic of a 2-D binary image. We experimentally demonstrate how a simple linear neuron is enough to attain the goal. We present results with sets of 2-D binary images and objects of different complexity and size.

Keywords: Artificial neural network, Euler number, Vertex chain codes, Binary image

Article Info
Received Aug 21, 2021
Accepted Sep 10,2021

1 Introduction

Object classification is one of the main problems in computer vision. Many methods and techniques have been proposed by the scientific community. The interested reader may refer, for example, to [1].

A digital binary image is a 2-D array that has been obtained from a gray-level image which has been discretized at two levels, 0 and 1. An image like this is composed of all connected regions representing projections of perceived objects onto the discrete plan. Pixels or cells that compose regions are labelled with level 1, whereas the background is labelled with level 0. As an example, Figure 1 shows an 8×8 image with three objects in gray: one of them with a hole and two with no holes.

The Euler number E of a binary image $I(x, y)$ can be globally computed as follows:

$$E = O - H \tag{1}$$

with O, the number of 4(8)-connected components or objects and H, the number of holes or background connected regions inside the O objects contained in $I(x, y)$.

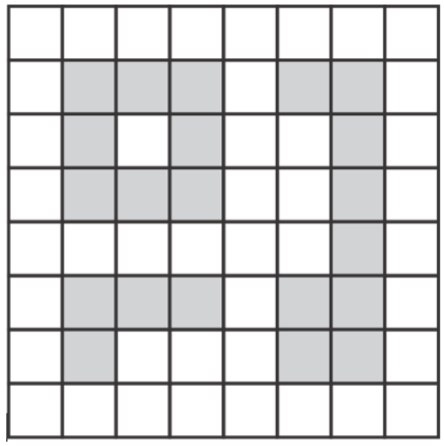


Figure 1. 8×8 image with three objects, one with one hole and two with no holes.

1.1 Related Work

Multiple methods to compute the Euler number of a binary image $I(x, y)$ have been reported in the literature. Probably the first proposal was introduced by Gray in 1971 [2]. Here, the Euler number is computed in terms of so-called bit-quads. In [3], the Euler characteristic is obtained through a quad-tree representation of the image under scrutiny. Linear quad-trees are used to perform the same task in [4], while in [5] employed a bin-tree representation. On the other hand, the Euler number is considered as the value of a certain additive function that belongs to the so-called Quermass-integrals family in [6]. The Euler number can be also defined in terms of vertices, basic square faces and edges from a binary image represented as a graph [7]. In [8] the Euler number is computed using the connectivity of the image graph. An integral geometric approach for the Euler feature is computed upon spatial images in [9] while a full proof about the Euler number equation can be found in [10]. In [11], authors use the notion of algebraic topology to compute the Euler number of a given object and the mathematical morphology operations and the additive property of this feature are adopted to calculate the Euler number of binary objects in [12].

The Euler characteristic of a discrete object and a discrete quasi-object is computed in [13] in terms of so-called vertex angles of the discrete surfaces.

In [14] the number of connected components, first planar Betti number, and the number of holes, second planar Betti number, are estimated by approximating the digital image by polygonal sets derived from its digitalization. Contrarily, in [15], the authors describe a method that combines image processing techniques and graph theory to compute the Euler number with connected components and holes in the binary image. In [16] the Euler characteristics of a digital image composed of k-connected shapes are computed in terms of so-called Morse operators.

Authors define the Euler number of a bipartite graph composed of n-vertices as the number of labellings of the vertices with $1, \dots, n$ in [17]. However, in [18], the authors compute the Euler number of a binary image in terms of the terminal and three-edged points.

In [19], the authors propose computing the Euler number in terms of a run-based algorithm. For this, they calculate 8-neighbor runs, unlike the conventional run-based algorithms, which need to record the start and endpoints of all runs. Nonetheless, in [20], the authors describe a different run-based algorithm to do the same task. They do in two phases. In the first phase, they process odd rows alternately to find runs and only record its end location. In the second phase, they process each of the remaining even rows to find runs and calculate neighbouring runs between the current row and the rows immediately above and below using the recorded run data.

The contact perimeter for “unit-with” shapes is used in Bribiesca [21] to compute the Euler number. Two variants of such proposal for the case of region-based shapes are described in [22], [23] for the cases of shapes composed of square and hexagonal cells, respectively. In [24], the authors propose two equations based on the pixel geometry and connectivity properties to compute the Euler number of a binary digital image with either thick or thin boundaries. Both equations are specialized only for 4-connectivity cases. In short, in [25], the authors introduce a method to compute the Euler number of a binary image based on a codification of contour pixels of the objects in the image.

An improvement on the Euler number computing algorithm used in MATLAB is well described in [26]. A graph-theory-based Euler number computing algorithm is introduced in [19], [27]. A novel bit-quad-based Euler number computing algorithm is also presented in [28]. In [29], authors also use bit-quads to compute the Euler number of a 2-D binary image. They present two variations, one useful for the case of images containing only 4-connected objects and one useful in the case of 8-connected objects.

Lastly, in [30], [31], [32], authors propose using so-called 16 bit-quads to train a kind of learning machine for estimating the Euler number of a 2-D binary image, such as a multilayered perceptron, and a morphological neural network, and a support vector machine, respectively.

1.2 Applications

The Euler number has been successfully applied in many applications. It has been employed, for example, in industrial part recognition as reported in [33]. In [34], the same topological feature has been used in real-time image thresholding. In [35], the Euler number has been utilized to analyse textural and topological features of benchmark images. It has been also harnessed to describe structural defects upon binary images that have been affected by noise in [36], and to extract lung regions from grey-level chest x-ray images in [37]. In [38], it has been also applied in object number counting. In [39], on the other hand, it has been used in real-time Malayan license plate recognition. It has been also employed in digit recognition from pressure sensor data as described in Paul et al. [40]. In [41], it has been utilized in a gender recognition system from offline handwritten signature, and in image description as explained in [42]. Another interesting application of the Euler number for gender discrimination from offline Hindi signature can be found in [43]. In [44], the Euler number, has been harnessed in character recognition. In short, in [45], authors incorporate several algebraic-geometric tools, namely α-Shapes, Betti numbers, and the Euler characteristic, into the topological analyses of cellular networks.

1.3 Hardware Implementations and Patents

To begin this section, it is convenient to say that a fast algorithm for computing the Euler number of an image and its Very Large-Scale Integration (VLSI) implementation is introduced in [46]. On the other hand, an on-chip computation of a binary image Euler number with applications to efficient database searching is well described in [47]. A novel pipeline architecture to compute this important topological feature is comprehensively described in [48]. A modification of the algorithm introduced in [39] allocated in a Field Programmable Gate Array (FPGA) with a pipelined architecture applied also to image binarization can be found in [49]. To end up this section, it is worth mentioning that one of the first patents about the Euler number computation for binary images is described by Acharya et al. [50].

1.4 Contributions

The rest of the paper is organized as follows. In the second section, we provide the background and definitions necessary to follow the lecture of the rest of the document. In the third section, we explain how an ANN can be trained to calculate the Euler number of a 4-connected binary image, based on its VCC representation. Section fourth and fifth test the performance of the trained neuron to estimate the Euler number of 2-D images with images of different sizes and complexities. Section sixth, finally concludes and states the directions for future work.

1.5 Organization of the Paper

The rest of the paper is organized as follows. In the second section, we provide the background and definitions necessary to follow the lecture of the rest of the document. In the third section, we explain how an ANN can be trained to calculate the Euler number of a 4-connected binary image, based on its VCC representation. Section fourth and fifth test the performance of the trained neuron to estimate the Euler number of 2-D images with images of different sizes and complexities. Section sixth, finally concludes and states the directions for future work.

2 Background and definitions

In this section, we present a set of definitions necessary to understand the rest of the paper. In the content of this work, we assume objects are composed of only square pixels.

Definition 1 [25]. A binary shape S_n is a k-connected region composed of n square cells. In the case of square cells, *Sn* can be 4-connected or 8-connected.

As an example, the image shown in Figure 1 has three 4-connected objects, one with a hole and composed of eight pixels, one formed by nine pixels, and one integrated of four pixels.

There are two ways of connecting pixels: 4-connected, Figure 2a, and 8-connected, Figure 2b. In the content of this paper, four connectivity is considered. In other words, if p and q are any two pixels belonging to the shape S_n, then p and q will appear connected by one of their sides.

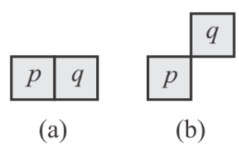


Figure 2. (a) 4-connectivity between pixels p and q. (b) 8-connectivity between pixels p and q.

In this paper, shapes with and without holes will be considered, refer to Figure 1. The contour of a shape is built by the exterior contour plus the interior contours derived from holes, if present. According to [51], each exterior corner of a contour cell (when it is in direct contact with the background of the shape) can be coded by the number of cells it touches at that position. As an example, let us consider Figure 3(a) that depicts a shape composed of 10 pixels. Figure 3(b) shows the labelled corners of the shape of Figure 3(a) with the number of faces they touch. As shown in Figure 3(b), in the case of contour vertices VC, only three different corner codes can be found: 1, 2 and 3. In this paper, such corners are coded as $VC1$, $VC2$ and $VC3$, respectively.

Definition 2 [25]. Let VC depict the contour of cells of an object and this is equal to $vc = \{N1, N2, N3\}^T$; where $N1$, $N2$ and $N3$ represent the number of times that $VC1$, $VC2$ and $VC3$ appear in an object, respectively.

As an example, for the shape shown in Figure 3(b): $N1 = 5, N2 = 8$ and $N3 = 1$, severally.

It is worth mentioning that if more pixels are appended or deleted from a given shape, numbers $N1, N2$ and $N3$ will change. To appreciate this, let's consider the three examples illustrated in Figures 3(c), 3(d) and 3(e). As shown in Figure 3(c) a pixel is deleted. In this case $N1 = 6, N2 = 6$ and $N3 = 3$. Now, suppose that the interior pixel is deleted as shown in Figure 3(d). If this is the case, as depicted a hole emerges, with $N1 = 5, N2 = 8$ and $N3 = 5$. In short, a pixel is added to the same shape as shown by Figure 3(e), $N1 = 6, N2 = 8$ and $N3 = 2$.

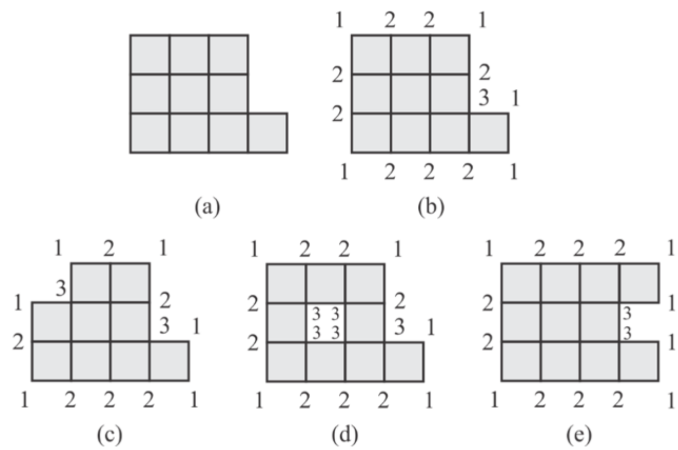


Figure 3. (a) A shape composed of 10 Pixels. (b) Vertex codes of the contour corners of the shape. (c) Vertex codes of the contour vertices when a pixel is removed from the contour of the shape. (d) Vertex codes when a hole is generated by removing an interior pixel from the shape. (e) Vertex codes of the contour vertices when a pixel is 4-connected to the exterior contour of the shape.

To end up this section, the equation that allows computing the Euler number of a 2-D binary image in terms of variables $N1$ and $N3$, reported in [25], is stated as follows:

Theorem 1 [25]. The Euler number E of a 2-D a binary image composed of O shapes and H holes is always given as follows:

$$E = \frac{N1 - N3}{4} \tag{2}$$

3 The Proposal

In this section, we describe the proposed methodology for training an ANN to accurately estimate the Euler number of a 2-D binary image. The methodology comprises three phases as follows:

1. **Image encoding**. Given a set of 2-D binary images $I = \{I^1, I^2, \dots, I^p\}$ and their corresponding Euler numbers $E = \{E^1, E^2, \dots, E^p\}$, map the set of images I into their corresponding Vertex Chain Codes $V = \{vc^1, vc^2, \dots, vc^p\}$, where $vc^i = \{N1, N2, N3\}^T$.
2. **ANN training.** Train a linear neural using k pairs: $\{vc^i, E^i\}$ with $i = 1,2, \dots, k$.
3. **Euler number computation**. Take any binary image I^i of size $m \times n$, transform it into its vector $vc^i = \{N1, N2, N3\}^T$ and obtain its corresponding Euler number E through the trained ANN.

Next, each of these steps are explained in detail.

3.1 Image encoding

Taking into account the material provided in Section 2, and as explained before, any 2-D binary image $I(x, y)$ of size $m \times n$ and composed of O objects and H holes, can be represented as a vector vc integrated by three components. As an example, let's consider the 8×8 image with three objects shown in Figure 4. For this image, $vc = \{14,21,6\}^T$. According to Equation (2), for this image, $E = \frac{14-6}{4} = 2$, as desired.

To map a binary image $I^i(x, y)$ into its corresponding code $vc^i = \{N1, N2, N3\}^T$, we apply the set of rules shown in Table 1. As illustrated in Figure 5(a), pixel $p(x, y)$ is taken as reference, then pixels $p(x-1, y), p(x-1, y-1)$ and $p(x, y-1)$ are considered to form the 16 rules to map an image into its corresponding codes. As an example, consider the configurations shown in Figure 5(b) and Figure 5(c), respectively. For the example shown in Figure 5(b), $p(x,y)=1$, $p(x-1,y)=0$, $p(x-1,y-1)=0$ and $p(x,y-1)=0$, thus, according to Table 1.

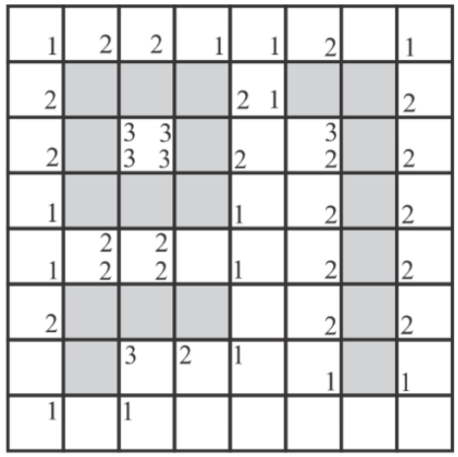


Figure 4. An 8×8 binary image with three 4-connected shapes, one of them with a hole.

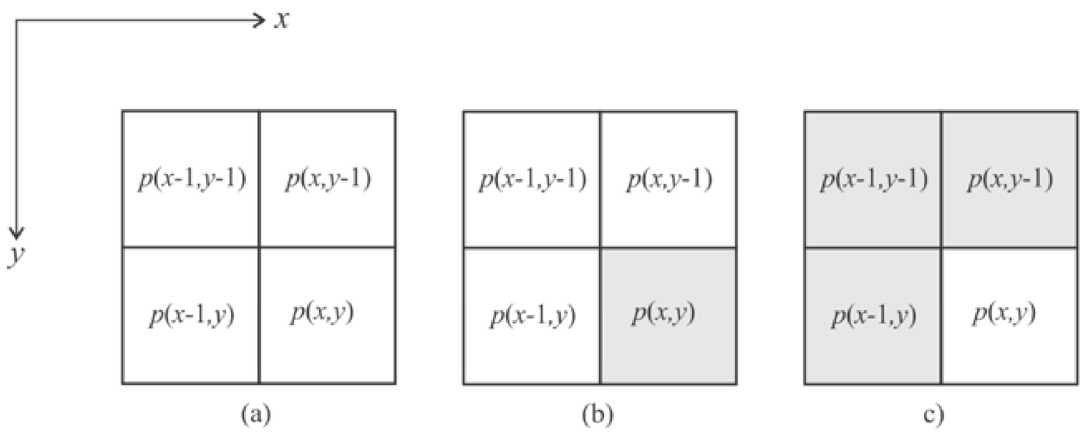


Figure 5. (a) Pixel values $p(x,y)$, $p(x-1,y)$, $p(x-1,y-1)$ and $p(x,y-1)$ to establish the set of rules to map image $I^i(x,y)$ into its code $vc^i=\{N1,N2,N3\}^T$. (b) Example where rule 9 is applied for incrementing variable $N1$. (c) An example where rule 8 is applied for incrementing variable $N3$.

Rule number	Rule
1	$If\ p(x,y)=0\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=0$ then do nothing
2	$If\ p(x,y)=0\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=1$ then $N_1=N_1+1$
3	$If\ p(x,y)=0\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=0$ then $N_1=N_1+1$
4	$If\ p(x,y)=0\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=1$ then $N_2=N_2+1$
5	$If\ p(x,y)=0\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=0$ then $N_1=N_1+1$
6	$If\ p(x,y)=0\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=1$ then do nothing
7	$If\ p(x,y)=0\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=0$ then $N_2=N_2+1$
8	$If\ p(x,y)=0\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=1$ then $N_3=N_3+1$
9	$If\ p(x,y)=1\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=0$ then $N_1=N_1+1$
10	$If\ p(x,y)=1\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=1$ then $N_2=N_2+1$
11	$If\ p(x,y)=1\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=0$ then do nothing
12	$If\ p(x,y)=1\ and\ p(x-1,y)=0\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=1$ then $N_3=N_3+1$
13	$If\ p(x,y)=1\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=0$ then $N_2=N_2+1$
14	$If\ p(x,y)=1\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=0\ and\ p(x,y-1)=1$ then $N_3=N_3+1$
15	$If\ p(x,y)=1\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=0$ then $N_3=N_3+1$
16	$If\ p(x,y)=1\ and\ p(x-1,y)=1\ and\ p(x-1,y-1)=1\ and\ p(x,y-1)=1$ then do nothing

Table 1. Set of rules to map binary image $I^i(x,y)$ into its corresponding code $vc^i=\{N1,N2,N3\}^T$.

Number of images	Number of epochs	w_1	w_2	w_3	Error
20	697	0.2517	0.0014	-0.2441	0.0171
40	156	0.2417	-0.0107	-0.2618	0.0613
60	122	0.2537	0.0040	-0.2463	0.0315
80	92	0.2554	0.0042	-0.2472	0.0257
100	76	0.2551	0.0039	-0.2468	0.0230
300	48	0.2495	1.0E-06	-0.2499	0.0065

Table 2. Estimated weight values by the linear neuron in estimating the Euler of a 2-D binary image with 20, 40, 60, 80, 100 and 300 images.

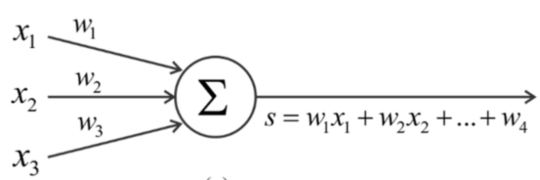
3.2 ANN training

We automatically generated 300 7×7 4-connected images with a different number of objects with and without holes. These 300 images were next transformed into their corresponding codes $V = \{vc^1, vc^2, \ldots, vc^p\}$, where $vc^i = \{N1, N2, N3\}^T$ and $i = 1,2,\ldots,300$ with their corresponding Euler number. After this, we trained a linear perceptron with three inputs as depicted in Figure 6(a). Surprisingly, this very simple ANN sufficed to solve the problem. We first, trained the linear perceptron with 20 images selected at random. We repeated the training process with 40 images, 60, 80, 100 and 300. In all cases, an Adam optimizer was used. Table 2 depicts the results.

As can be appreciated in all 6 cases, the linear neuron tends to the same three values: 0.25 for weight w_1, 0.0 for weight w_2 and -0.25 for weight w_3. Also note that as the number of samples increases, the number of iterations diminishes. In all five cases, the processing time maintains almost the same. Row 7 demonstrates that if the number of samples is increased, the approximation to the desired weights is closer.

Now take Equation (2) and express it as follows:

$$E = 0.25N1 - 0.25N3 \tag{3}$$



(a)

Figure 6. (a) Artificial Neural Network trained to estimate the Euler number of a 2-D binary image.

3.3 Euler number estimation

Given an image $I^i(x, y)$, transform it to its corresponding code $vc^i = \{N1, N2, N3\}^T$. Take this code and present it to the trained neuron to estimate the Euler number.

As a numerical example, let us take the image shown in Figure 4. For this image, we already know that $= \{14,21,6\}^T$. From Table 2, row six, $E = 0.2551 \times 14 + 0.0039 \times 21 - 0.2468 \times 6 = 2.1725$, which is close to desired value of 2.

Note how expected, the trained neuron arrived to accurately approximate the desired function to compute the Euler number of a 2-D binary image. Furthermore, the ANN weighted and selected the most important weights.

4 Results with Images of Different Sizes and Complexities

In this section, we test the performance of the trained neuron to estimate the Euler number of 2-D images. For this, we first approximated the three obtained weights to $0.25, 0.0$ and -0.25, respectively. We did this because the obtained weights tend to these values.

We describe two experiments. In the first experiment, we used the four images shown in Figure 7 with 3, 4, 5 and 6 shapes, respectively. In the second case, we used three sets of images of the same object under different image transformations, Figure 8. The idea, in this case, was to test the invariance of the estimated value of the Euler number of an image when applying the neuron under image transformations on the same object.

Tables 3 and 4 summarize the results. As the reader can rapidly appreciate, in all cases the desired Euler number for all images have been obtained as desired.

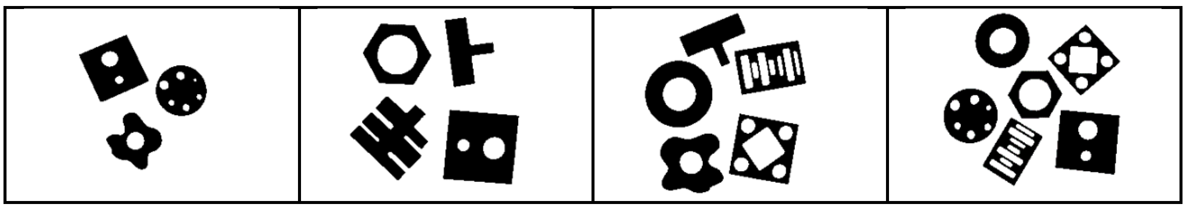


Figure 7. Binary images of 320×240 pixels with a different number of objects were used to test the performance of the trained neuron.

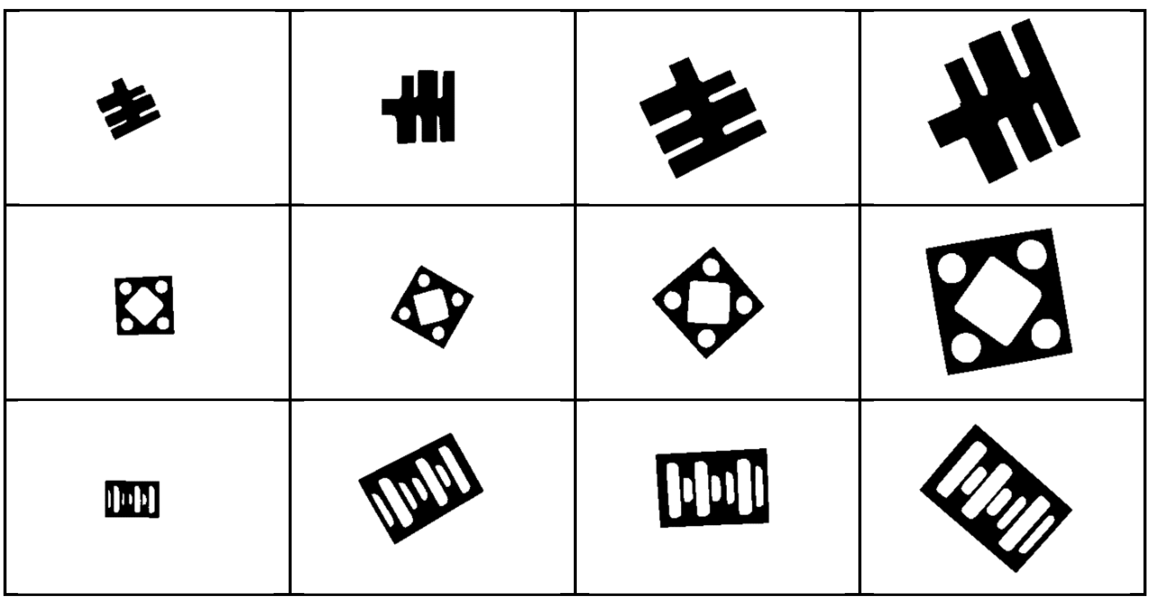


Figure 8. Binary images of 320×240 pixels with one object subjected to different image transformations to test the performance of the trained neuron to estimate the Euler number of a 2-D binary image.

In the second experiment, one may wonder why when the trained neuron is applied to each of the three sequences of images shown in Figure 8 outputs the same result. This could be explained as follows. Suppose we have a binary image I^1 with O objects and H holes. Then, I^1 is transformed into image I^2 by applying an image transformation T as follows $I^2 = T(I^1)$. T could be any image transformation: translation, rotation, scale change, affine or projection, even a combination of several of them. The corresponding representations of these two images are vc^1 and vc^2, respectively. Of course, due T, $vc^1 \neq vc^2$. However, as depicted in Table 3, the output of the neuron in the three examples shown, is the same.

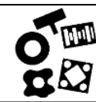
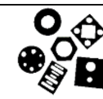
Image				
Estimated Euler number	-6	+1	-9	-18
Desired Euler number	-6	+1	-9	-18

Table 3. Estimated Euler numbers by the trained neuron over images of Figure 7.

When we feed vc^1 and vc^2 to the trained neuron, we obtain $E^1 = w_1x_{11} + w_2x_{12} + w_3x_{13} + w_4$ and $E^2 = w_1x_{21} + w_2x_{22} + w_3x_{23} + w_4$, respectively. If we take out the elements that do not contribute the computations as suggested in Table 2, we have that $E^1 = 0.25x_{11} - 0.25x_{13}$ and $E^2 = 0.25x_{21} - 0.25x_{23}$, respectively. So, it is necessary that $0.25x_{11} - 0.25x_{13} = 0.25x_{21} - 0.25x_{23}$, that coincides with Equation (3).

As an example, let us take the third sequence of four images shown in Figure 3, with the two more representatives *VC*s. For these four images we have that:

$$vc^1 = (564{,}486{,}588)^T, vc^2 = (127{,}1132{,}151)^T, vc^3 = (940{,}250{,}964)^T, vc^4 = (48{,}516{,}72)^T$$

With $vc^i = \{N1, N2, N3\}^T$. This discussion can be formally stated as follows:

Proposition 1. The Euler number calculated by the trained neuron in terms of the VC of any transformed binary image $I^2 = T(I^1)$ is the same, this is $E^2 = E^1$.

Proof:

Basis: For the first and fourth image of the third row of Table 4, $E = -6$ in both cases. The corresponding VCs from each of these two images are $vc^1 = (564{,}486{,}588)^T$ and $vc^2 = (127{,}1132{,}151)^T$, respectively. By using the trained neuron, we obtain that $0.25 \times 564 + 0 - 0.25 \times 588 = 0.25 \times 127 + 0 - 0.25 \times 151$, which is true because both images have the same Euler number of -6.

Induction step: Let I^1 be a binary 4-connected object with O objects and H holes and $I^2 = T(I^1)$ its transformed version trough image transformation T, $VC1$ and $VC2$ their two corresponding VC representations. So $E^1 = w_1x_{11} + w_2x_{12} + w_3x_{13} + w_4$ and $E^2 = w_1x_{21} + w_2x_{22} + w_3x_{23} + w_4$ the output of the trained neuron. If $E^1 = E^2$, then $x_{11} - x_{13} = x_{21} - x_{23}$.

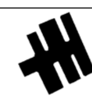
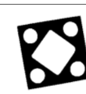
Image				
Estimated Euler number	+1	+1	+1	+1
Desired Euler number	+1	+1	+1	+1
Image				
Estimated Euler number	-4	-4	-4	-4
Desired Euler number	-4	-4	-4	-4
Image				
Estimated Euler number	-6	-6	-6	-6
Desired Euler number	-6	-6	-6	-6

Table 4. Estimated Euler numbers by the trained neuron over images of Figure 8.

5 Results with Real Objects

In this section, we present results with real objects. For this, we use the six images depicted in Table 5. We applied the trained neuron shown in Figure 6(a) with the weights obtained when using 300 images $\{0.2495, 0.0, -0.2499\}$ to these six images and obtained the Euler numbers shown in rows 4 and 8 of Table 5. Note that estimated values approach the theoretical values obtained using Equations (2) or (3).

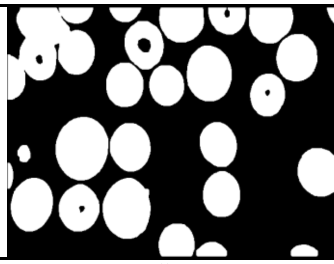
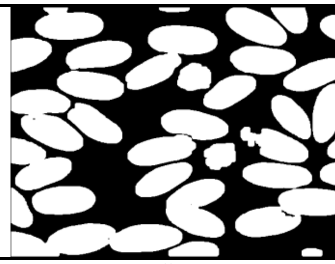
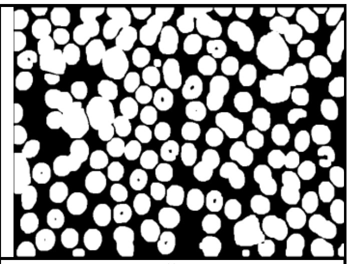
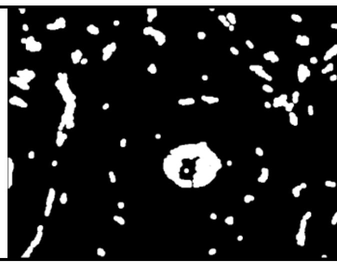
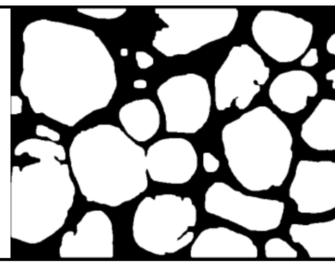
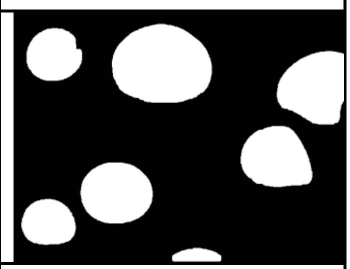
			
Name	Cell 01	Cell 02	Cell 03
N_1, N_2, N_3	2168, 4002, 2084	2964, 6448, 2852	5828, 10436, 5408
Calculated E by the linear perceptron	21	28	105
			
Name	Bone 01	Osteoporosis 01	Cheese 01
N_1, N_2, N_3	1963, 2970, 1691	2585, 5290, 2473	847, 1628, 819
Calculated E by the linear perceptron	68	28	7

Table 5. Real images of 640×480 pixels were used to test the performance of the trained neuron.

6 Conclusions and Future Work

In this section, we present the conclusions to which we have arrived at the end of this research. We also talk about future trends that emanate from this investigation:

1. We have shown that a very simple machine, a linear perceptron, can be used to approximate with precision the function to estimate the Euler number of a 2-D image and the number of holes of an object. In both cases, the VC of the 2-D image, object, is used. This is so important because the described procedure, training a linear perceptron to find the most relevant features, could be used to solve other combinatorial problems with possibly hundreds of variables.
2. The estimated value obtained by the trained machines is very precise, regardless of the number of objects and holes in the image.
3. As future work, we propose training an artificial neural network to estimate the Euler number of a binary image but in the 3-D case. In this case, we will use VC for voxels.

6 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 20200630 and SIP 20210788 CONACYT 65 (Fronteras de la Ciencia) and CONACYT FORDECYT PRONACES 6005 to undertake this research. Laura Lira thanks CONACYT for the scholarship granted to carry out her doctoral studies.

References

[1] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognit.*, vol. 37, no. 1, pp. 1–19, 2004, doi: https://doi.org/10.1016/j.patcog.2003.07.008.

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

[3] 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.

[4] 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.

[5] H. Bieri, "Computing the Euler characteristic and related additive functionals of digital objects from their bintree representation," *Comput. Vision, Graph. Image Process.*, vol. 40, no. 1, pp. 115–126, 1987, doi: https://doi.org/10.1016/0734-189X(87)90059-4.

[6] H. Bieri and W. Nef, "Algorithms for the euler characteristic and related additive functionals of digital objects," *Comput. Vision, Graph. Image Process.*, vol. 28, no. 2, pp. 166–175, 1984, doi: https://doi.org/10.1016/S0734-189X(84)80019-5.

[7] 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.

[8] F. Chiavetta and V. Di Gesù, "Parallel computation of the Euler number via Connectivity Graph," *Pattern Recognit. Lett.*, vol. 14, no. 11, pp. 849–859, 1993, doi: https://doi.org/10.1016/0167-8655(93)90148-7.

[9] 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.

[10] 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.

[11] D. Ziou and M. Allili, "Generating cubical complexes from image data and computation of the Euler number," *Pattern Recognit.*, vol. 35, no. 12, pp. 2833–2839, 2002, doi: https://doi.org/10.1016/S0031-3203(01)00238-2.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] R. Ehrenborg and Y. Farjoun, “Asymptotics of the Euler number of bipartite graphs,” *Adv. Appl. Math.*, vol. 44, no. 2, pp. 155–167, 2010, doi: https://doi.org/10.1016/j.aam.2009.05.002.

[18] 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.

[19] B. Yao, S. Kang, X. Zhao, Y. Chao, and L. He, “A graph-theory-based Euler number computing algorithm,” in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 1206–1209.

[20] 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.

[21] 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.

[22] S. H., C.-J. E. V., and Z.-N. D., “Computation of the Euler Number of a Binary Image Composed of Hexagonal Cells,” *J. Appl. Res. Technol.*, vol. 8, pp. 340–350, 2010, [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-64232010000300004&nrm=iso.

[23] 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.

[24] S. H., R.-E. Elsa, S. Raúl, L. Alejandro, P. A. Alejandro, and C. J. E.V., “Alternative formulations to compute the binary shape Euler number,” *IET Comput. Vis.*, vol. 8, no. 3, pp. 171–181, 2014.

[25] S. H., S.-M. R., P.-C. M., and R.-E. E., “Computing the Euler Number of a Binary Image Based on a Vertex Codification,” *J. Appl. Res. Technol.*, vol. 11, pp. 360–369, 2013, [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-64232013000300006&nrm=iso.

[26] 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.

[27] L. He and Y. Chao, “A Very Fast Algorithm for Simultaneously Performing Connected-Component Labeling and Euler Number Computing,” *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2725–2735, 2015.

[28] Y. Bin, H. Lifeng, K. Shiying, C. Yuyan, and Z. Xiao, “A novel bit-quad-based Euler number computing algorithm,” vol. 4, 2015, doi: 10.1186/s40064-015-1511-8.

[29] S. Humberto, C. Ángel, S.-M. Raúl, B.-C. Ernesto, and P.-B. Alberto, “Efficient computation of the euler number of a 2-D binary image,” in *Advances in Soft Computing - 15th Mexican International Conference on Artificial Intelligence, MICAI 2016, Proceedings*, Jan. 2017, pp. 401–413, doi: 10.1007/978-3-319-62434-1_33.

[30] S. Humberto, C. Ángel, S. Raúl, and P.-B. Alberto, “Support vector machines applied to 2-D binary image Euler number computation,” Dec. 2016, pp. 3–8, doi: 10.1109/ICMEAE.2016.010.

[31] S. Humberto, C. Ángel, G. Elizabeth, and S. Raúl, “Computing the 2-D image euler number by an Artificial Neural Network,” Oct. 2016, pp. 1609–1616, doi: 10.1109/IJCNN.2016.7727390.

[32] 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.

[33] 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.

[34] 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.

[35] V. Mayank, S. Richa, M. Pabitra, and N. Afzel, "Signature Verification Using Static and Dynamic Features," in *Neural Information Processing*, 2004, pp. 350–355.

[36] Chune Zhang, Zhengding Qiu, Dongmei Sun, and Jie Wu, "Euclidean Quality Assessment for Binary Images," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, vol. 2, pp. 300–303.

[37] W. L. P. and E. H. T., "A Study of Nodule Detection Using Opaque Object Filter," in *3rd Kuala Lumpur International Conference on Biomedical Engineering 2006*, 2007, pp. 236–240.

[38] 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.

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

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

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

[42] Q. Zhang, L. Wang, J.-H. Yu, M. Zhang. "Segmentation-based Euler number with multi-levels for image feature description". *Procedia Computer Science Volume* 111:245-251, 2017.

[43] M. Pal, S. Bhattacharyya, T. Sarkar. "Euler number based feature extraction technique for gender discrimination from offline Hindi signature using SVM & BPNN classifier". *2018 Emerging Trends in Electronic Devices and Computational Techniques (EDCT)*, pp. 1-6.

[44] Y. R. Matsuoka, G. Angelo R. Sandoval, L. Paolo Q. Say, J. Skyler Y. Teng, 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". *2018 International Conference on Platform Technology and Service (PlatCon)*, pp. 1-6

[45] Y. Chen, R. Li, Zh. Zhao, H. Zhang. "Study on Base Station Topology in National Cellular Networks: Take Advantage of Alpha Shapes, Betti Numbers, and Euler Characteristics". *IEEE Systems Journal* 2019

[46] S. Dey, B. B. Bhattacharya, M. K. Kundu, and T. Acharya, "A fast algorithm for computing the Euler number of an image and its VLSI implementation," in *VLSI Design 2000. Wireless and Digital Imaging in the Millennium. Proceedings of 13th International Conference on VLSI Design*, 2000, pp. 330–335.

[47] A. Bishnu, B. B. Bhattacharya, M. K. Kundu, C. A. Murthy, and T. Acharya, "On-chip computation of Euler number of a binary image for efficient database search," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, 2001, vol. 3, pp. 310–313.

[48] A. Bishnu, B. B. Bhattacharya, M. K. Kundu, C. A. Murthy, and T. Acharya, "A pipeline architecture for computing the Euler number of a binary image," *J. Syst. Archit.*, vol. 51, no. 8, pp. 470–487, 2005, doi: https://doi.org/10.1016/j.sysarc.2004.12.001.

[49] N. Abbasi, J. Athow, and A. Amer, "Real-time FPGA architecture of modified Stable Euler-Number algorithm for image binarization," in *2009 16th IEEE International Conference on Image Processing*

(ICIP), 2009, pp. 3253–3256.

[50] T. Acharya, B. B. Bhattacharya, A. Bishnu, M. K. Kundu, and C. A. Murthy, “Computing the Euler Number of a Binary Image,” no. 7027649. 2006.

[51] E. Bribiesca, “A new chain code,” *Pattern Recognit.*, vol. 32, no. 2, pp. 235–251, 1999, doi: https://doi.org/10.1016/S0031-3203(98)00132-0.