

# Learning an Artificial Neural Network for Discovering Combinations of Bit-Quads to Compute the Euler Characteristic of a 2-D Binary Image

Fernando Arce<sup>1</sup>, Humberto Sossa<sup>2,3</sup>, Wilfrido Gómez-Flores<sup>4</sup>, Laura Lira<sup>2</sup>

<sup>1</sup> Centro de Investigaciones en Óptica A. C.,  
Mexico

<sup>2</sup> Instituto Politécnico Nacional,  
Centro de Investigación en Computación,  
Mexico

<sup>3</sup> Tecnológico de Monterrey,  
Mexico

<sup>4</sup> Centro de Investigación y de Estudios Avanzados del IPN,  
Mexico

{fernando.arce.vega, humbertosossa}@gmail.com, wgomez@cinvestav.mx, laugi2808@hotmail.com

**Abstract.** The Image Analysis community has widely used so-called bit-quads to propose formulations for computing the Euler characteristic of a 2-D binary image. Reported works have manually proposed different combinations of bit-quads to provide one or more formulations to calculate this important topological feature. This paper empirically shows how an Artificial Neural Network can be trained to find an optimal combination of bit-quads to compute the Euler characteristic of any binary image. We present results with binary images of different complexities and sizes and compare them with state-of-the-art machine learning algorithms.

**Keywords.** Euler characteristic, bit-quads, holes, objects, artificial neural network.

## 1 Introduction

The Euler characteristic is a topological feature commonly used to describe the connectivity properties of an isolated object or a binary image.

Let  $S \in \mathbb{R}^2$  be a digital 0-1-image, where  $S$  is related to the foreground pixels, and its

complement  $S^c$  is associated with the background pixels. In the plane, the Euler characteristic can be written as an alternating sum of Betti numbers as [20]:

$$E(S) = N_c(S) - N_h(S), \quad (1)$$

where  $N_c(S)$  is the number of connected components of  $S$  and  $N_h(S)$  is the number of holes; that is, bounded connected components of  $S^c$ .

The Euler characteristic of an image  $I(x, y)$  has been widely used since the eighties in many applications. In [42], for example, the Euler number has been used to recognize industrial parts. In [33], the same topological feature has been used for real-time image thresholding. It has also been applied in object number counting in [21], real-time Malayan license plate recognition in [10], digit recognition from pressure sensor data in [28], gender recognition from offline handwritten signature in [23], image description in [49], gender discrimination from offline Hindi signature in [27]. In short, in [24], the Euler characteristic has been used for character recognition.

**Table 1.** Possible bit-quads that can be used to compute the Euler characteristic of a 2-D binary image

$Q_1: Q_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$				
$Q_2: Q_{21} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$Q_{22} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$Q_{23} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$Q_{24} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$	
$Q_3: Q_{31} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$	$Q_{32} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$Q_{33} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$	$Q_{34} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	
$Q_4: Q_{41} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$Q_{42} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$Q_{43} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$Q_{44} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	
$Q_5: Q_{51} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$				
$Q_6: Q_{61} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$Q_{62} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$			

Several methods to compute the Euler characteristic of a binary image  $I(x, y)$  have been reported in the literature. Refer, for example, to [2, 1, 4, 6, 5, 8, 9, 11, 13, 14, 12, 15, 17, 16, 18, 20, 21, 25, 26, 30, 39, 38, 41, 40, 36, 35, 37, 34, 19, 22, 47, 48, 46, 3, 43, 45, 44, 50, 51].

So-called bit-quads have been widely used since they were proposed in the image processing area to count the Euler characteristic of a 2-D binary image. Table 1 depicts the 16 possible bit-quad matrices that people have proposed to obtain different formulations to compute the Euler characteristic of a 2-D binary image.

The set of papers that we have found in the literature that uses a combination of bit-quads to compute Euler characteristic of a binary image are the following: [11, 13, 14, 12, 17, 16, 21, 36, 35, 37, 34, 47, 48, 46, 3, 43, 45, 44].

In [11, 13, 14, 12, 17, 16, 21, 47, 46, 3, 45], authors used ten bit-quads to compute the Euler characteristic of a binary image  $I(x, y)$  as follows:

$$4\text{-connected case: } E = \frac{\#Q_2 - \#Q_4 + 2\#Q_6}{4}, \quad (2)$$

and

$$8\text{-connected case: } E = \frac{\#Q_2 - \#Q_4 - 2\#Q_6}{4}, \quad (3)$$

where:

- $\#Q_2$  is the occurrences of bit-quads  $Q_{21}, Q_{22}, Q_{23}$ , and  $Q_{24}$ .

- $\#Q_4$  is the occurrences of bit-quads  $Q_{41}, Q_{42}, Q_{43}$ , and  $Q_{44}$ .

- $\#Q_6$  is the occurrences of bit-quads  $Q_{61}$ , and  $Q_{62}$ .

Equations 2 and 3 are also known as Gray's formulas.

As we have seen before, Equation (2) needs to perform ten comparisons on each image pixel. The time complexity for this method is  $O(N)^2$  for a  $N \times N$  image.

In [21], authors also propose using bit-quads  $Q_{23}, Q_{42}, Q_{61}$ , and  $Q_{62}$  to compute the Euler characteristic of image an  $I(x, y)$  as follows:

$$4\text{-connected case: } E = \#Q_{23} - \#Q_{42} + \#Q_{61}, \quad (4)$$

$$8\text{-connected case: } E = \#Q_{23} - \#Q_{42} - \#Q_{62}, \quad (5)$$

where  $\#Q_{23}, \#Q_{42}, \#Q_{61}$ , and  $\#Q_{62}$  are the occurrences of bit-quads  $Q_{23}, Q_{42}, Q_{61}$  and  $Q_{62}$ , respectively.

In [34], authors propose combining bit-quads  $Q_{21}, Q_{44}, Q_{61}$ , and  $Q_{62}$  to compute the Euler characteristic of a 2-D binary image as follows:

$$4\text{-connected case: } E = \#Q_{21} - \#Q_{44} + \#Q_{61}, \quad (6)$$

$$8\text{-connected case: } E = \#Q_{21} - \#Q_{44} - \#Q_{62}, \quad (7)$$

where  $\#Q_{21}, \#Q_{44}, \#Q_{61}$ , and  $\#Q_{62}$  are the occurrences of bit-quads  $Q_{21}, Q_{44}, Q_{61}$  and  $Q_{62}$ , respectively.

In [48, 43, 44], authors describe three different improvements for computing the Euler characteristic based on counting previous bit-quads. This procedure allows reducing the time and the number of required pixels for processing a bit-quad.

In [32], the authors discuss how to use bit-quads in the so-called tree of shapes by using the depth of the node as the value of pixels in a larger and interpolated image representation. As a result, they propose an algorithm that uses larger image representations. In [31], the authors present a review of previously published works to compute attributes such as area, perimeter, and the Euler characteristic by incrementally counting patterns while traversing nodes of a component tree.

In short, in [36, 35, 37], the authors propose using the 16 bit-quads shown in Table 1 to train distinct kinds of learning machines for estimating the Euler characteristic of a 2-D binary image: a multilayered perceptron, a morphological neural network, and a support vector machine.

Inspired in the previous discussion, in this paper we propose solving the following problem: automatically derive equations for computing the Euler characteristic of a binary image. For this, we suggest training an Artificial Neural Network (ANN) to automatically find optimal combinations of bit-quads to compute the Euler characteristic of a binary image.

The rest of the paper is organized as follows. In the Section 2, we explain the methodology followed. We show how two ANNs, one for the 4-connected and the other for the 8-connected case, can be trained to find optimal combinations of bit-quads to calculate the Euler characteristic of a binary image. Section 3 is oriented to present the results after training both ANNs. Besides, a set of experimental results are presented, where the performance of the ANNs is tested when computing the Euler characteristic over different images with distinct resolutions and complexities and compared them with state-of-the-art machine learning algorithms. Section 4 is finally focused on concluding remarks and future research.

## 2 Methodology

From the material presented in Section 1, for about 47 years, only three combinations of bit-quads and three improvements have been proposed. Probably, some more can be found. As we have seen, the most popular uses ten bit-quads, the one initially proposed in [11]. The other two, as explained in [21, 34], combine three bit-quads. Apparently, three is the minimum number of bit-quads that can be used to compute the Euler characteristic of a 2-D binary image in 4(8) connectivity.

Henceforward, the 16 bit-quads shown in Table 1 will be ordered and represented as depicted in Table 2.

**Table 2.** New representation of the 16 bit-quads from Table 1

Bit-quad number	Bit-quad (matrix form)	Bit-quad (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

### 2.1 Finding an Optimal set of Bit-Quads to Compute the Euler Characteristic of a 2-D Binary Image

Finding an appropriate combination  $C$  of bit-quads to compute the Euler characteristic of a 2-D binary image can be manually obtained by testing the performance of such combinations from a set of images  $SI = \{I_1, I_2, \dots, I_p\}$  and their corresponding Euler characteristics  $SEN = \{E_1, E_2, \dots, E_p\}$ . It is clear that such combination  $C$  is one of those that can be obtained from the set:

$$C_1^{16} + C_2^{16}, \dots, + C_{16}^{16}. \quad (8)$$

**Table 3.** Binary encoding example

4 × 4 binary image	Vector $v$
$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$v = [0 \ 1 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$

One could test with all those combinations to find a convenient formulation. Doing this certainly could take a lot of time. Using common sense, we can appreciate that some bit-quads do not contribute to solving the problem. This is the case of bit-quads:  $Q_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$  and  $Q_{51} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$  that describe the background of the image and the interior of the objects, respectively.

In what follows, we demonstrate how an ANN can be used to find an optimal combination of  $C$  bit-quads from the set of 16 bit-quads depicted in Table 2 to efficiently compute the Euler characteristic of a 2-D binary image  $I(x, y)$ . For this, we propose using the following two-step procedure. Given a set of 2-D binary images  $SI = \{I_1, I_2, \dots, I_p\}$  and their corresponding Euler characteristics:  $SEN = \{E_1, E_2, \dots, E_p\}$ :

1. Map the set of images  $SI = \{I_1, I_2, \dots, I_p\}$  to a matrix  $V = \{v_1, v_2, \dots, v_p\}$ , with  $v_i = (\#Q_1, \#Q_2, \dots, \#Q_{16})^T$ , where  $\#Q_1, \#Q_2, \dots, \#Q_{16}$  represent the occurrences of bit-quads  $Q_1, Q_2, \dots, Q_{16}$ , respectively. As an example, refer to Table 3, where nine bit-quads can be found.
2. Present each pair  $(v_i, E_i)$  to an untrained neural network and iteratively adjust its weights until convergence. Algorithm 1 shows the pseudo-code of the training process.

We used 400  $4 \times 4$  2-D images for the 4-connected case and 400  $4 \times 4$  2-D images for the 8-connected case. All random images were generated using a discrete and uniform distribution and zero-padded. Furthermore, both linear perceptrons were trained with a learning rate of 0.001, using the RMSprop optimizer [29] and 250 epochs [7]. Besides, the weights were initialized using Xavier normal initializer.

---

**Algorithm 1** Training process
 

---

```

Begin
Generate the set training images  $SI = \{I_1, I_2, \dots, I_{400}\}$ 
Calculate the Euler characteristic  $SEN = \{E_1, E_2, \dots, E_{400}\}$ , using Gray's formulas
Encode the  $SI$  images to matrix  $V = \{v_1, v_2, \dots, v_{400}\}$ 
for  $i$  in range of 250 epochs:
  for each  $(v_i, E_i)$ :
    Perform feed-forward
    Calculate the weight and bias gradients
    Update weights and bias
End
  
```

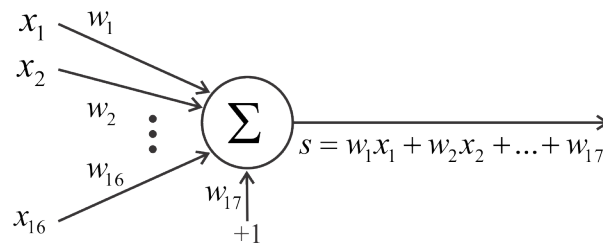
---

## 2.2 Performance of Four Different Machine Learning (ML) Algorithms to Forecast the Euler Characteristic

We also test the performance of four different machine learning (ML) algorithms to accurately forecast the Euler characteristic: the proposed single-layer ANN, a multi-layer perceptron (MLP) with one hidden layer, a support vector regression (SVR) with Gaussian kernel, and one dimensional convolutional neural network (CNN-1D). A benchmark with 400 binary images of  $10 \times 10$ ,  $15 \times 15$ ,  $20 \times 20$  and  $25 \times 25$  pixels are used for training the models, considering 4- and 8-connectivity cases. With  $k=10$ , the  $k$ -fold cross-validation technique is used to generate disjoint training and test sets.

For all the algorithms, the input vectors correspond to the 16 bit-quad patterns calculated from binary images. Furthermore, a tuning process is performed to find the hyperparameters that reduce the mean squared error.

For MLP, the number of hidden units is searched in the range  $[2, 16, 32]$  with ReLU activation functions and with a linear neuron in the output. For SVR, the penalization cost  $C$  and the kernel bandwidth  $\gamma$  are in the ranges  $[2^5, 2^6, \dots, 2^9]$  and  $[2^{-4}, 2^{-3}, \dots, 2^4]$ , respectively. For CNN-1D, the number filters are examined in the range  $[8, 16, 32]$  with hyperbolic tangent activation functions and with a kernel size in range  $[1, 2, 3, 4]$ , with a hidden layer with  $[8, 16, 32]$  neurons and a linear neuron in the output.



**Fig. 1.** Proposed artificial neural network to find a good combination of bit-quads to estimate the Euler characteristic  $E$  of a 2-D binary image

The performance is tested 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.

### 3 Results

In this section, we present the results at which we arrived after training an artificial neural network to estimate the Euler characteristic in 2-D binary images. We experimentally began testing with the simplest artificial neural network. We used a linear perceptron with 17 inputs (i.e., 16 bit-quads plus the bias value) with a linear activation function, as illustrated in Fig. 1, trained with Algorithm 1. As we will see, we found that this very simple learning machine was enough to find an optimal combination of bit-quads.

#### 3.1 Learned ANN Weight Values

For the 4-connected case, we end up with a vector of weights, as shown in Table 4. For the 8-connected case, on the other hand, we finalized with a vector of weights depicted in Table 5.

In both cases, note that most of the weight values are almost zero. This finding indicates that the ANN found that the corresponding bit-quads are not representative for estimating the Euler characteristic of a 2-D image. On the other hand, note also that in both 4(8)-connected cases, three weights approach +1 and -1 values. To establish the validity of the obtained results, we re-trained the linear perceptron 100 times over sets of 400  $4 \times 4$  2-D binary images and their corresponding

**Table 4.** Weights obtained when training a linear perceptron using  $4 \times 4$  2-D binary images for the 4-connected case

Weight number	Weight value	Rounded values	Bit-quad
1	-0.016	0	0000
2	-0.022	0	0001
3	-0.011	0	0010
4	-0.017	0	0011
5	-0.011	0	0100
6	-0.017	0	0101
7	-0.007	0	0110
8	-0.013	0	0111
<b>9</b>	<b>0.978</b>	<b>1</b>	1000
<b>10</b>	<b>0.976</b>	<b>1</b>	1001
11	-0.015	0	1010
12	-0.021	0	1011
13	-0.015	0	1100
14	-0.020	0	1101
<b>15</b>	<b>-1.000</b>	<b>-1</b>	1110
16	-0.017	0	1111
bias	0.145		

Bold values indicate the most representative weights and their corresponding bit-quads

Euler characteristics for both the 4(8) connected cases. Tables 4 and 5 show the average of the perceptron weights after training it 100 times:

1. For the 4-connected case, note that three weights approach to +1 and -1 values, corresponding to bit-quads 9, 10, and 15, respectively. Both weights 9 and 10 approach to +1, while weight 15 approaches to -1. Notice that the highest weights exactly correspond to the bit-quads used in Equation (5) to calculate the Euler characteristic of a 4-connected 2-D binary image.
2. For the 8-connected case, note that three weights approach to +1 and -1 values,

**Table 5.** Weights obtained when training a linear perceptron using  $4 \times 4$  2-D binary images for the 8-connected case

Weight number	Weight value	Rounded values	Bit-quad
1	0.000	0	0000
2	-0.003	0	0001
3	0.004	0	0010
4	0.001	0	0011
5	0.005	0	0100
6	0.000	0	0101
7	<b>-0.991</b>	<b>-1</b>	0110
8	0.004	0	0111
9	<b>0.997</b>	<b>1</b>	1000
10	-0.006	0	1001
11	0.001	0	1010
12	-0.002	0	1011
13	0.001	0	1100
14	-0.003	0	1101
15	<b>-0.994</b>	<b>-1</b>	1110
16	0.001	0	1111

bias 0.108

Bold values indicate the most representative weights and their corresponding bit-quads

corresponding to bit-quads 9, 7, and 15, respectively. Weight 9 approaches to +1, while both weights 7 and 15 approach to -1. Notice that the highest weights exactly correspond to the bit-quads used in Equation (6) to calculate the Euler characteristic of a 8-connected 2-D binary image.

Furthermore, we changed the weight initialization method and retraining 100 times the linear perceptron. We averaged and rounded all the learned weights by the ANNs to the nearest integer to reduce the influence of undesired bit-quads. As a result, we obtained the following new equations never reported in the literature for the 4-connected case:

$$E = \#Q_5 + \#Q_7 - \#Q_{14}, \quad (9)$$

and for the 8-connected case:

$$E = \#Q_5 - \#Q_{10} - \#Q_{14}. \quad (10)$$

Based on the Stochastic Gradient Descent (SGD) training algorithm [29], the linear perceptron has learned to select an optimal combination of bit-quads  $C$  to estimate the Euler number of a 2-D binary image. The perceptron arrived precisely at the same results that a human being has

empirically done before, in this case, Equations (5) and (6). Moreover, when changing the weight initialization method, the linear perceptron discovered new formulations for computing the Euler characteristic. This relevant finding suggests the ability of the ANN to automatically provide new expressions that would take a human being a long time to discover.

### 3.2 Assessment of Discovered Expressions

Next, we tested the new expressions of the Euler characteristics shown in Equations 9 and 10 in images with distinct sizes and complexities. Three experiments were performed. In the first case, we used the small resolution images of 10 logos shown in Fig. 2. In the second case, we used medium resolution images with different objects with and without holes, as shown in Fig. 3. Lastly, in the third experiment, we used medium resolution images with one object subjected to different image transformations, Fig. 4. The idea is to assess invariance to different variations of the same object.

Tables 6, 7 and 8 summarize the results. As the reader can rapidly appreciate, the desired Euler characteristic for all images has been obtained with perfect precision in all cases.

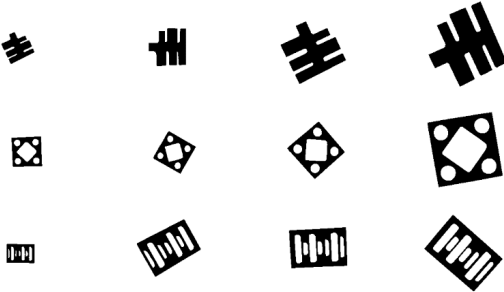


**Fig. 2.** Binary images of  $128 \times 128$  pixels of ten different logos used to test the performance of the trained perceptron to estimate the Euler characteristic of a 2-D binary image

The trained neuron is applied to each of the three sequences of images shown in Fig. 4, and it obtains the same output. This results could be explained as follows. Suppose we have a binary image  $I_1$  with  $O$  objects and  $H$  holes. Then, this image is transformed into another image by applying to  $I_1$  an image transformation



**Fig. 3.** Binary images of  $320 \times 240$  pixels with a different number of objects of diverse complexity used to test the performance of the trained perceptron to estimate the Euler characteristic of a 2-D binary image



**Fig. 4.** Binary images of  $320 \times 240$  pixels with one object subjected to different image transformation to test the performance of the trained perceptron to estimate the Euler characteristic of a 2-D binary image

$T$  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 bit-quad representations of these two images are  $v_1$  and  $v_2$ , respectively. Of course, due  $T$ ,  $v_1 \neq v_2$ . However, as depicted in Table 6, the neuron's output in the three examples is the same.

Coming back to our example of transforming  $I_1$  to  $I_2$  by  $T$ , when we feed  $v_1$  and  $v_2$  to the trained neuron, we obtain  $s_1 = w_1x_{1,1} + w_2x_{2,1} + \dots + w_{17}$  and  $s_2 = w_1x_{1,2} + w_2x_{2,2} + \dots + w_{17}$ . If we take out the elements that do not contribute to the computation, as illustrated in Table 4, for the 4-connected case, we have  $s_1 = x_{9,1} + x_{10,1} - x_{15,1}$  and  $s_2 = x_{9,2} + x_{10,2} - x_{15,2}$ . It is clear that, in order to the trained neuron output the same result, it is necessary that  $x_{9,1} + x_{10,1} - x_{15,1} = x_{9,2} + x_{10,2} - x_{15,2}$ .

As an example, let us take the first sequence of four images shown in Fig. 4, with the three most representative bit-quads. For these four images we have that:  $v_1 = [69, 0, 68]^T$ ,  $v_2 = [8, 0, 7]^T$ ,  $v_3 = [140, 0, 139]^T$  and  $v_4 = [66, 0, 65]^T$ , where

$v_i = [\#Q_9, \#Q_{10}, \#Q_{15}]^T$ . This discussion can be formally stated as follows:

**Proposition 1.** The Euler characteristic  $E$  of any binary image is the same after applying any type of image transformation by counting the most representative bit-quads found by the linear perceptron.

**Proof. Basis:** For the first and fourth images of the third row of Table 8, the most representative bit-quads from each of these two images are  $v_1 = [328, 0, 334]^T$  and  $v_2 = [6, 0, 12]^T$ , respectively. By using the trained neuron, we obtain that  $328 + 0 - 334 = -6$  and  $6 + 0 - 12 = -6$ , which is true because both images have the same Euler characteristic  $E$  of  $-6$ .

**Induction step:** Let  $I_1$  be a binary image with  $O$  objects and  $H$  holes,  $I_2 \rightarrow T(I_1)$  its transformed version through image transformation  $T$ , and  $v_1$  and  $v_2$  their two corresponding bit-quad representations. So  $s_1 = x_{9,1} + x_{10,1} - x_{15,1}$  and  $s_2 = x_{9,2} + x_{10,2} - x_{15,2}$ , the two outputs of the trained neuron, respectively. If  $s_1 = s_2$ , then  $x_{9,1} + x_{10,1} - x_{15,1} = x_{9,2} + x_{10,2} - x_{15,2}$ .










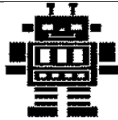
Notice that this proposition was focused on the 4-connected case shown in Table 4, but it also applies, with minimal changes, for the 8-connected case shown in Table 5, as well as for Equations 9 and 10.

### 3.3 Euler Characteristic Prediction





Table 9 displays the results of the proposed single-layer ANN and three ML algorithms to predict the Euler characteristic for the 4- and 8-connectivity cases. Remarkably, the single-layer ANN method shows an outstanding capacity to predict the Euler characteristic with fewer computational resources. Although the other three algorithms also achieve significant results.

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, SVR and CNN-1D methods can build a mapping function to predict the Euler characteristic accurately. Nevertheless, though the MLP, SVR and CNN-1D predict the Euler characteristic very efficiently, they lack the property of explainability; that is, a human being













**Table 6.** Estimated Euler characteristics by the trained perceptron over images of Fig. 2

Image					
Computed Euler characteristic	+3	-1	+4	+10	-6
Desired Euler characteristic	+3	-1	+4	+10	-6
Image					
Computed Euler characteristic	+1	+5	+3	+12	+5
Desired Euler characteristic	+1	+5	+3	+12	+5

**Table 7.** Estimated Euler characteristics by the trained perceptron over images of Fig. 3

Image				
Computed Euler characteristic	-6	+1	-9	-18
Desired Euler characteristic	-6	+1	-9	-18

**Table 8.** Estimated Euler characteristics by the trained perceptron over images of Fig. 4

Image				
Computed Euler characteristic	+1	+1	+1	+1
Desired Euler characteristic	+1	+1	+1	+1
Image				
Computed Euler characteristic	-4	-4	-4	-4
Desired Euler characteristic	-4	-4	-4	-4
Image				
Computed Euler characteristic	-6	-6	-6	-6
Desired Euler characteristic	-6	-6	-6	-6

**Table 9.** Comparison of ML algorithms to predict the Euler characteristic. The mean value of 10-fold cross validation experiments is shown

Methods	4-connectivity				8-connectivity			
	MSE	MAE	PCC	D	MSE	MAE	PCC	D
ANN	<b>0.000</b>	<b>0.000</b>	<b>1.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>1.000</b>	<b>0.000</b>
MLP	<b>0.000</b>	0.002	0.998	0.010	<b>0.000</b>	0.004	0.996	0.050
CNN-1D	<b>0.000</b>	0.003	0.999	0.037	<b>0.000</b>	0.003	0.999	0.037
SVR	<b>0.000</b>	0.002	0.999	0.007	<b>0.000</b>	0.001	0.999	0.001

In bold are highlighted the best results



understands the meaning of all the parts that make up the calculation of the Euler characteristic.

## 4 Conclusions and Future Work

In this section, we present the conclusions we have arrived at in this research. We also talk about future trends that emanate from this investigation.

We have shown that a very simple machine can find optimal combinations of bit-quads to estimate the Euler characteristic of a 2-D binary image.

This result is relevant because the described procedure, i.e., training a linear perceptron to find the most relevant bit-quads, could be applied to solve other combinatorial problems with possibly hundreds of variables.

Moreover, the neuron arrives at the same equations found manually by humans without a priori knowledge by using a limited set of training images. Also, the linear perceptron was capable of discovering new expressions never reported previously in the literature.

Additionally, the estimated Euler numbers obtained by the discovered expressions, in both cases (4-connected and 8-connected), are accurate, regardless of the number of objects and holes in the image.

These promising results encourage us to train an artificial neural network to estimate the Euler number of binary 3-D images. In this case, instead of bit-quads, we should use bit-octos represented by  $2 \times 2 \times 2$  matrices.

## Acknowledgments

Authors would like to thank the Instituto Politécnico Nacional and the Centro de Investigaciones en Óptica A.C., for the economic support under project 20200630 to undertake this research. Laura Lira thanks CONACYT for the scholarship granted to carry out her doctoral studies.

## References

1. **Bieri, H. (1987)**. Computing the Euler characteristic and related additive functionals of digital objects from their bintree representation. *Computer Vision, Graphics, and Image Processing*, Vol. 40, No. 1, pp. 115–126. DOI: [https://doi.org/10.1016/0734-189X\(87\)90059-4](https://doi.org/10.1016/0734-189X(87)90059-4).
2. **Bieri, H., Nef, W. (1984)**. Algorithms for the Euler characteristic and related additive functionals of digital objects. *Computer Vision, Graphics, and Image Processing*, Vol. 28, No. 2, pp. 166–175. DOI: [https://doi.org/10.1016/S0734-189X\(84\)80019-5](https://doi.org/10.1016/S0734-189X(84)80019-5).
3. **Bin, Y., Lifeng, H., Shiyang, K., Yuyan, C., Xiao, Z. (2015)**. A novel bit-quad-based Euler number computing algorithm. *SpringerPlus*, Vol. 4. DOI: 10.1186/s40064-015-1511-8.
4. **Bribiesca, E. (2010)**. Computation of the Euler number using the contact perimeter. *Computers & Mathematics with Applications*, Vol. 60, No. 5, pp. 1364–1373. DOI: <https://doi.org/10.1016/j.camwa.2010.06.018>.
5. **Chao, Y., Kang, S., Yao, B., Zhao, X., He, L. (2015)**. An efficient Euler number computing algorithm. *2015 IEEE International Conference on Information and Automation*, pp. 1167–1171.
6. **Chen, M.-H., Yan, P.-F. (1988)**. A fast algorithm to calculate the Euler number for binary images. *Pattern Recognition Letters*, Vol. 8, No. 5, pp. 295–297. DOI: [https://doi.org/10.1016/0167-8655\(88\)90078-5](https://doi.org/10.1016/0167-8655(88)90078-5).
7. **Dauphin, Y., de Vries, H., Bengio, Y. (2015)**. Equilibrated adaptive learning rates for non-convex optimization. In **Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R.**, editors, *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 1504–1512.
8. **Díaz-De-León, J. L., Sossa, H. (1996)**. On the computation of the Euler number of a binary object. *Pattern Recognition*, Vol. 29, No. 3, pp. 471–476. DOI: [https://doi.org/10.1016/0031-3203\(95\)00098-4](https://doi.org/10.1016/0031-3203(95)00098-4).
9. **Dyer, C. R. (1980)**. Computing the Euler number of an image from its quadtree. *Computer Graphics and Image Processing*, Vol. 13, No. 3, pp. 270–276. DOI: [https://doi.org/10.1016/0146-664X\(80\)90050-7](https://doi.org/10.1016/0146-664X(80)90050-7).
10. **Faqheri, W. A., Mashohor, S., Ehsan, S. D. (2009)**. A real-time malaysian automatic license plate recognition (m-alpr) using hybrid fuzzy.

11. **Gray, S. B. (1971).** Local properties of binary images in two dimensions. *IEEE Transactions on Computers*, Vol. C-20, No. 5, pp. 551–561.
12. **He, L., Chao, Y. (2015).** A very fast algorithm for simultaneously performing connected-component labeling and Euler number computing. *IEEE Transactions on Image Processing*, Vol. 24, No. 9, pp. 2725–2735.
13. **He, L., Chao, Y., Suzuki, K. (2012).** A new algorithm for labeling connected-components and calculating the Euler number, connected-component number, and hole number. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3099–3102.
14. **He, L., Chao, Y., Suzuki, K. (2013).** An algorithm for connected-component labeling, hole labeling and Euler number computing. *Journal of Computer Science and Technology*, Vol. 28, pp. 468–478.
15. **He, L., Yao, B., Zhao, X., Yang, Y., Chao, Y., Ohta, A. (2015).** A graph-theory-based algorithm for Euler number computing. *IEICE Transactions on Information and Systems*, Vol. E98.D, No. 2, pp. 457–461. DOI: 10.1587/transinf.2014EDL8155.
16. **He, L., Yao, B., Zhao, X., Yang, Y., Shi, Z., Kasuya, H., Chao, Y. (2018).** A fast algorithm for integrating connected-component labeling and Euler number computation. *J. Real-Time Image Process.*, Vol. 15, No. 4, pp. 709–723. DOI: 10.1007/s11554-015-0499-1.
17. **He, L., Zhao, X., Yao, B., Yang, Y., Chao, Y., Shi, Z., Suzuki, K. (2017).** A combinational algorithm for connected-component labeling and Euler number computing. *J. Real-Time Image Process.*, Vol. 13, No. 4, pp. 703–712. DOI: 10.1007/s11554-014-0433-y.
18. **Imiya, A., Eckhardt, U. (1999).** The Euler characteristics of discrete objects and discrete quasi-objects. *Computer Vision and Image Understanding*, Vol. 75, No. 3, pp. 307–318. DOI: <https://doi.org/10.1006/cviu.1999.0791>.
19. **Joshi, O. S. (2018).** Binary image processing for computation of connected components, image holes and Euler number using graph theory. *International Conference On Advances in Communication and Computing Technology (ICACCT)*, pp. 230–234.
20. **Kiderlen, M. (2006).** Estimating the Euler characteristic of a planar set from a digital image. *Journal of Visual Communication and Image Representation*, Vol. 17, No. 6, pp. 1237–1255. DOI: <https://doi.org/10.1016/j.jvcir.2006.05.001>.
21. **Lin, X., Ji, J., Gu, Y. (2007).** The Euler number study of image and its application. *2007 2nd IEEE Conference on Industrial Electronics and Applications*, pp. 910–912.
22. **Lin, X., Sha, Y., Ji, J., Wang, Y. (2006).** A proof of image Euler number formula. *Science in China Series F-Information Sciences*, Vol. 49, No. 3, pp. 364–371. DOI: <https://doi.org/10.1007/s11432-006-0364-8>.
23. **Maji, P., Chatterjee, S., Chakraborty, S., Kausar, N., Samanta, S., Dey, N. (2015).** 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.
24. **Matsuoka, Y. R., R. Sandoval, G. A., Q. Say, L. P., Teng, J. S. Y., Acula, D. D. (2018).** 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.
25. **Nagel, Ohser, Pischang (2000).** An integral-geometric approach for the Euler-poincaré characteristic of spatial images. *Journal of Microscopy*, Vol. 198, No. 1, pp. 54–62.
26. **Nonato, L. G., Filho, A. C., Minghim, R., Batista, J. (2004).** Morse operators for digital planar surfaces and their application to image segmentation. *IEEE Transactions on Image Processing*, Vol. 13, No. 2, pp. 216–227.
27. **Pal, M., Bhattacharyya, S., Sarkar, T. (2018).** 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. 16.
28. **Paul, A., Bhattacharya, N., Chowdhury, A. S. (2012).** 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.
29. **Ruder, S. (2016).** An overview of gradient descent optimization algorithms.
30. **Samet, H., Tamminen, M. (1985).** Computing geometric properties of images represented by linear quadtrees. *IEEE Transactions on Pattern*

- Analysis and Machine Intelligence, Vol. PAMI-7, No. 2, pp. 229–240.
31. **Silva, D. J., Alves, W. A., Hashimoto, R. F. (2020).** Incremental bit-quads count in component trees: Theory, algorithms, and optimization. *Pattern Recognition Letters*, Vol. 129, pp. 33–40. DOI: <https://doi.org/10.1016/j.patrec.2019.10.036>.
  32. **Silva, D. J. d., Alves, W. A. L., Morimitsu, A., Gobber, C. F., Hashimoto, R. F. (2019).** Incremental bit-quads count in tree of shapes. *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing -ISMM*, Springer. DOI: 10.1007/978-3-030-20867-7\_13.
  33. **Snidaro, L., Foresti, G. L. (2003).** Real-time thresholding with Euler numbers. *Pattern Recogn. Lett.*, Vol. 24, No. 9–10, pp. 1533–1544. DOI: 10.1016/S0167-8655(02)00392-6.
  34. **Sossa, H., Carreón, Á., Santiago-Montero, R., Bribiesca-Correa, E., Petrilli-Barceló, A. (2017).** Efficient computation of the Euler number of a 2-D binary image. *Advances in Soft Computing - 15th Mexican International Conference on Artificial Intelligence, MICAI 2016, Proceedings, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, Alemania, pp. 401–413. DOI: 10.1007/978-3-319-62434-1\_33.
  35. **Sossa, H., Carreón, Á., Guevara, E., Santiago, R. (2016).** Computing the 2-D image Euler number by an artificial neural network. *Proceedings of the International Joint Conference on Neural Networks*, pp. 1609–1616. DOI: 10.1109/IJCNN.2016.7727390.
  36. **Sossa, H., Carreón, Á., Santiago, R. (2016).** Training a multilayered perceptron to compute the Euler number of a 2-D binary image. *Lecture Notes in Computer Science*, pp. 44–53. DOI: 10.1007/978-3-319-39393-3\_5.
  37. **Sossa, H., Carreón, Á., Santiago, R., Petrilli-Barcelo, A. (2016).** Support vector machines applied to 2-D binary image Euler number computation. *International Conference on Mechatronics, Electronics, and Automotive Engineering, ICMEAE 2016*, pp. 3–8. DOI: 10.1109/ICMEAE.2016.010.
  38. **Sossa, H., Cuevas-Jimenez, E. B., Zaldivar-Navarro, D. (2011).** Alternative way to compute the Euler number of a binary image. *Journal of applied research and technology*, Vol. 9, pp. 335–341.
  39. **Sossa, H., Cuevas-Jimenez, E. V., Zaldivar-Navarro, D. (2010).** Computation of the Euler number of a binary image composed of hexagonal cells. *Journal of applied research and technology*, Vol. 8, pp. 340–350.
  40. **Sossa, H., Rubio-Espino, E., Santiago, R., López, A., Pena Ayala, A., Cuevas Jimenez, E. (2014).** Alternative formulations to compute the binary shape Euler number. *IET Computer Vision*, Vol. 8, No. 3, pp. 171–181.
  41. **Sossa, H., Santiago-Montero, R., Perez-Cisneros, M., Rubio-Espino, E. (2013).** Computing the Euler number of a binary image based on a vertex codification. *Journal of applied research and technology*, Vol. 11, pp. 360–369.
  42. **Yang, H. S., Sengupta, S. (1988).** Intelligent shape recognition for complex industrial tasks. *IEEE Control Systems Magazine*, Vol. 8, No. 3, pp. 23–30.
  43. **Yao, B., He, L., Kang, S., Zhao, X., Chao, Y. (2016).** A further improvement on bit-quad-based Euler number computing algorithm. *IEICE Transactions on Information and Systems*, Vol. E99.D, No. 2, pp. 545–549. DOI: 10.1587/transinf.2015EDL8159.
  44. **Yao, B., He, L., Kang, S., Zhao, X., Chao, Y. (2017).** Bit-quad-based Euler number computing. *IEICE Transactions on Information and Systems*, Vol. E100.D, No. 9, pp. 2197–2204. DOI: 10.1587/transinf.2017EDP7012.
  45. **Yao, B., He, L., Kang, S., Zhao, X., Chao, Y. (2017).** A new run-based algorithm for Euler number computing. *Pattern Anal. Appl.*, Vol. 20, No. 1, pp. 49–58. DOI: 10.1007/s10044-015-0464-4.
  46. **Yao, B., Kang, S., Zhao, X., Chao, Y., He, L. (2015).** A graph-theory-based Euler number computing algorithm. *2015 IEEE International Conference on Information and Automation*, pp. 1206–1209.
  47. **Yao, B., Wu, H., Yang, Y., Chao, Y., He, L. (2013).** An improvement on the Euler number computing algorithm used in matlab. *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, pp. 1–4.
  48. **Yao, B., Wu, H., Yang, Y., Chao, Y., Ohta, A., Kawanaka, H., He, L. (2014).** An efficient strategy for bit-quad-based Euler number computing algorithm. *IEICE Transactions on Information and Systems*, Vol. E97.D, No. 5, pp. 1374–1378. DOI: 10.1587/transinf.E97.D.1374.

49. **Zhang, Q., Wang, L., Yu, J.-H., Zhang, M. (2017).** Segmentation-based Euler number with multi-levels for image feature description. *Procedia Computer Science*, Vol. 111, pp. 245–251. DOI: <https://doi.org/10.1016/j.procs.2017.06.060>.
50. **Zhao Zhang, Moss, R. H., Stoecker, W. V. (2001).** A novel morphological operator to calculate Euler number. *Proceedings International Workshop on Medical Imaging and Augmented Reality*, pp. 226–228.
51. **Ziou, D., Allili, M. (2002).** Generating cubical complexes from image data and computation of the Euler number. *Pattern Recognition*, Vol. 35, No. 12, pp. 2833–2839. DOI: [https://doi.org/10.1016/S0031-3203\(01\)00238-2](https://doi.org/10.1016/S0031-3203(01)00238-2).

*Article received on 02/08/2021; accepted on 17/09/2021.  
Corresponding author is Humberto Sossa.*