

# Distinguishing paintings from photographs by complexity estimates

Adrian Carballal<sup>1</sup> · Antonino Santos<sup>1</sup> · Juan Romero<sup>1</sup> · Penousal Machado<sup>2</sup> · João Correia<sup>2</sup> · Luz Castro<sup>1</sup>

Received: 23 March 2015 / Accepted: 18 December 2016  
© The Natural Computing Applications Forum 2016

**Abstract** This study is aimed at exploring the ability of complexity-based metrics to distinguish between paintings and photographs. The proposed features resort to edge detection, compression and entropy estimate methods that are highly correlated with artwork complexity. Artificial neural networks based on these features were trained for this task. The relevance of various combinations of these complexity metrics is also analyzed. The results of the current study indicate that different estimates related to image complexity achieve better results than state-of-the-art feature sets based on color, texture and perceptual edges. The classification success rate achieved is 94.82% on a dataset of 5235 images.

**Keywords** Artificial neural networks · Complexity estimates · Edge detection · Feature extraction · Image retrieval

---

✉ Antonino Santos  
nino@udc.es

Adrian Carballal  
adrian.carballal@udc.es

Juan Romero  
jj@udc.es

Penousal Machado  
machado@dei.uc.pt

João Correia  
jncor@dei.uc.pt

Luz Castro  
maria.luz.castro@udc.es

<sup>1</sup> Artificial Neural Networks and Adaptive Systems LAB, University of A Coruña, 15071 A Coruña, Spain

<sup>2</sup> CISUC, Department of Informatics Engineering, University of Coimbra, 3030 Coimbra, Portugal

## 1 Introduction

The problem of distinguishing between paintings and photographs, a non-trivial task even for human observers, is addressed in this study. Some works [8, 9] suggest the use of metrics related to color, perceptual edges and texture. They argue that photographs and paintings differ substantially in their edge properties; thus, edge properties could be used to automatically differentiate between them. Furthermore, it is shown that the variation of image intensity is substantial and systematic in photographs. Cutzu et al. [8] pointed out that paintings seemed to use color rather than systematic changes of image intensity to represent different objects and object regions.

On a similar subject as the current work, Athitsos et al. [4] tried to distinguish between photographs and computer-generated graphics, with performance levels of over 90% in JPEG and GIF images. Measures related to color and farthest neighbor histograms were used. The authors noted that there was more variability in the color transitions from pixel to pixel in photographs than in graphics, and that graphics contained more distinct colors than photographs.

The current work proposes generic image complexity estimates for image classification related to style and texture [3]. In a simplified way, the complexity of an image is related to entropy and is the opposite of order. It is related to the minimal information, or the minimal program, required to “construct” the image. It can be observed that it depends on the degree of predictability of each pixel of the image [27]. Therefore, a plane image with all the pixels of the same color shows a perfect order, and hence, it is less complex. A pure random image can be seen as extremely complex, and the value of each pixel is impossible to predict even when taking into account the values of neighbor pixels.

It is considered that the visual complexity associated with a painting and a picture may allow distinguishing between them. Taking this supposition as a reference, it is suggested using a series of complexity estimates which have been previously studied to address this problem [3]. Inspired by previous works [18, 29, 39], this paper is aimed at addressing this matter by using a feature extractor to obtain features from images and an artificial neural network (ANN from now on) to classify a set of images and make the automatic discrimination between photographs and paintings.

The remainder of the paper is organized as follows: Sect. 2 presents different state-of-the-art works; Sect. 3 describes the different features which will be used to differentiate between paintings and photographs; the method used for carrying out the classification is shown in Sect. 4; Sect. 5 shows the experimental results; and the conclusions and future work are drawn in Sect. 6.

## 2 State of the art

The most similar to our work is the study carried out by Cutzu et al. [8], which proposed several features derived from the color, edge and gray-scale texture information of the image to differentiate between real-scene photographs and paintings. The authors used a dataset of 12,000 images (6000 paintings and 6000 photographs) gathered from different Web sites. The images had the following restrictions: “(1) no monochromatic images were used; all the images had a color resolution of 8-bits per color channel, (2) frames and borders were removed, (3) no photographs altered by filters or special effects were included, (4) no computer-generated images were used, (5) no images with large areas overlaid with text were used” which can affect the application of their system in some domains. They proposed three different classifiers: classifier operating in the space of scalar-valued features; classifier for RGBXY space; and classifier for Gabor space. The results of the classifiers were 72, 81 and 79%, respectively. Using the three classifiers at the same time (and select the option voted by the majority of the classifiers), Cutzu obtained a result of 93%.

Athitsos et al. [4] proposed a system that discriminated between photographs and computer-generated graphics (desktop icons and web pages), with a success rate of over 90%. They used two color histogram-based features: the prevalent color metric and the color histogram metric. The implemented system used decision trees to classify the images. To create the decision trees, they used 1025 GIF graphics, 362 GIF photographs, 270 JPEG graphics and 643 JPEG photographs as a training set.

While the aforementioned works lie in the use of ad hoc high-level features, it is suggested that different complexity estimates can be used to differentiate between paintings and photographs.

There are some psychological works that establish relationships between image complexity, visual perception and esthetics [5, 11, 12, 26]. A deeper analysis on this topic can be found in the study performed by Nadal [28]. Different measures of image complexity have been employed previously [5, 14, 18, 27, 28], but some of them are not computable or are very difficult to compute. Others were defined for a particular type of image or type of content of the image, e.g., as simple geometric elements.

The proposed metrics can be classified in four different types: (i) based on estimates of the “complexity” of the image, which measure the error of the JPEG and Fractal compression method [18, 21, 30]; (ii) gathered from the application of the Zipf’s Law [43], (iii) calculated according to the Fractal Dimension of the image [41]; and (iv) using the previous methods to the image after applying edge detection filters.

The relevance of the perceived image complexity is a recurring topic in the field of esthetics [2, 5, 27]. Inspired by these theoretical works on esthetics, Machado et al. [18] proposed complexity estimates and achieved human competitive results in the psychological test: “Design Judgment Test” (DJT from now on). Machado et al. [18] used an ANN along with a subset of the features proposed in this paper and obtained an average success rate of 74.49% in DJT. In Machado et al. [20], authors used a similar subset of the features proposed and an ANN classifier for the painter identification task, obtaining identification rates higher than 90% across experiments. Machado et al. [21] explored their use to assign fitness to the images evolved by an evolutionary art tool. Studies such as those conducted by Saunders [36] or Svängård and Nordin [40], among others, follow a similar line of work.

The perceptual relevance of Fractal Dimension is explored in the works of Aks and Sprott [1] to quantify esthetic preferences and in several papers by Taylor et al. (e.g., Spehar et al. [39] and Taylor et al. [41]) to analyze the evolution of Pollock’s dripping technique and the esthetic qualities of landscape silhouettes.

The relevance of Zipf’s Law-based features in the musical field was established in a series of studies (e.g., Voss et al. [42] and Manaris et al. [23–25]), which developed musical classifiers based on Zipf’s metrics and ANNs. These systems were thoroughly tested in author identification tasks, in pleasantness prediction and in predicting the number of downloads of MIDI pieces from ClassicalMidiArchive.com within a month by using two classes, musical pieces with high and low number of downloads in a similar way to Datta et al. [10]. All these experiments

obtained success rates above 87% (e.g., Manaris et al. [24, 25]). Other works that used Zipf's Law-based features in the visual field have been reviewed by Machado et al. [20, 24].

### 3 Feature set

This section describes the proposed features related to complexity. The feature extraction process can be summarized to the following steps:

- *Preprocessing*, including all the transformation and normalization operations applied to a given input image.
- *Filter application*, in particular edge detection, to identify points in images at which the image brightness has discontinuities.
- *Metrics application*, application of certain operations based on image complexity estimates, and statistical measurements such as average and standard deviation.

#### 3.1 Image transformation

Every image is individually submitted to a series of transformations before they are analyzed. A given input image is loaded and resized to a standard width and height of  $256 \times 256$  pixels, transformed into a three-channel image in the RGB (red, green and blue) color space, with a depth of 8-bit per channel, and all pixel values scaled to the  $[0; 255]$  interval. This step ensures that all input images share the same format and dimensions.

Afterward, the image is converted into the HSV (Hue, Saturation and Value) color. Each color channel is stored as a 1-channel gray-scale image. These images will be referred to as H, S and V.

A new gray-scale image is also created by performing a pixel-by-pixel multiplication of S and V channels (referred to as CS for Colorfulness) and scaling the result to  $[0; 255]$ . From here on, this new method of calculation of H will be referred to as HCS. This picture is obtained by using jointly the information of the H channel angle with that of CS, obtaining a new value associated with the H channel. This is a possible solution for the existing problem as regards the HSV color model for extreme values of the S and H channels.

Some steps of this transformation, such as the change of the aspect ratio to 1:1 or the resize, involve a loss of information and a distortion of the picture, but in previous experiments in other fields it has been proved that such a transformation did not affect the ability of the system to carry out the classification [31, 33].

Even Cutzu et al. [9] argued that such transformations had to be taken into account, since they should have reduced the real ability of the resulting system: "It is easy to convince oneself that reducing image size (by smoothing and subsampling) renders the perceptual painting/photograph discrimination more difficult if the paintings have "realistic" content. Thus, it is reasonable to expect that the discrimination performance of our classifier will also improve with increasing image resolution hypothesis that we are planning to verify in future work." Even though it is not proved, we are fully aware that by scaling the image information may be lost and it could hinder the performance of the classifiers.

#### 3.2 Edge detection filters

The resulting images are subject of new transformation operations related to the application of classic filters of edge detection. Edges in an image usually indicate changes in depth, orientation, illumination, material, object boundaries, and so on. In this case, Sobel [38] and Canny [7] filters have been used, since they are filters with a wide scientific spreading.

The Canny edge detection algorithm [7] includes four stages, and these are: image smoothing, gradient calculation, non-maximum suppression and thresholding to detect "true" edges, while suppressing "false" non-edge filter responses.

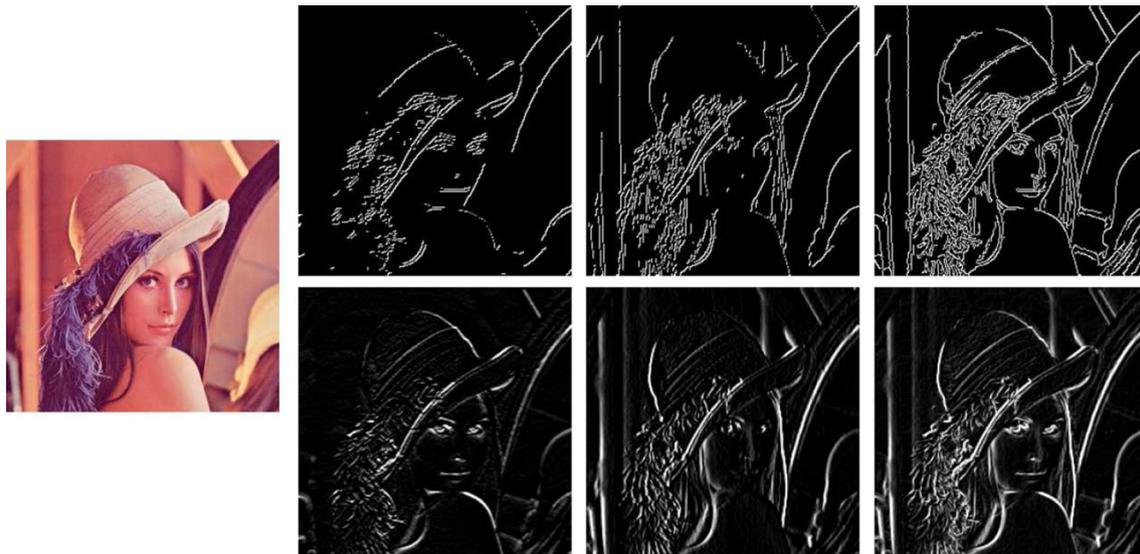
The Sobel edge detection method [38] consists of a discrete differentiation operator, which computes an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable and integer-valued filter in horizontal and vertical directions.

These edge detection operators return a value for the first derivative in the horizontal direction ( $G_x$ ) and vertical direction ( $G_y$ ). The edge gradient (Eq. (1)) and the direction (Eq. (2)) can be determined from this value.

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (2)$$

The feature extractor uses both filters. Both in the case of Canny and Sobel, three transformations are carried out, the first of them applying the filter in its horizontal direction, the second one in the vertical direction and the third one applying both of them. Figure 1 shows an example of this type of transformation. From the given picture "Lenna.gif" (the color picture placed on the left), seven



**Fig. 1** Examples of edge detection filters application. The first one is the same picture without applying any filters. The other six are the aforementioned pictures. The first column matches the pictures of the

Sobel and Canny filters applied horizontally, the second column of the filters applied vertically and, the last one, to the filters applied vertically and horizontally

different pictures will be obtained. The first one is the same picture without applying any filters. The other six are the aforementioned pictures. The first column matches the pictures of the Sobel and Canny filters applied horizontally, the second column of the filters applied vertically and, the last one, to the filters applied vertically and horizontally.

Each of these transformations is applied on any image obtained from the picture preprocessing stage, i.e., on the original picture and on every picture that has been obtained by extracting the different color channels.

### 3.3 Average and standard deviation-based estimates

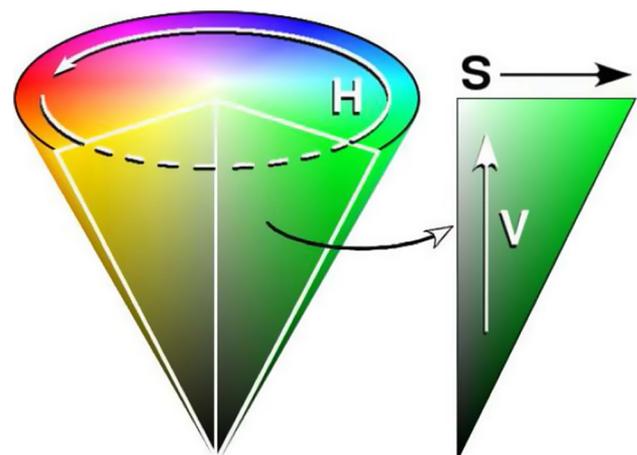
The average and the standard deviation are simply calculated using the pixel intensity value of each image, except for the H (Hue) channel image. Since the Hue channel is circular, the average and the standard deviation are calculated based on the norm and angle of Hue values (see Fig. 2). In addition, a multiplication of the Hue angle value by CS value is made, and consequentially, a norm is calculated using Hue and CS values.

Thus, both Hue and CS pixel values are scaled to [0,1] values and both average and standard deviation Hue are calculated as follows:

$$hx_i = \cos(H[x_i, y_i] \times 2\pi) \tag{3}$$

$$hy_i = \sin(H[x_i, y_i] \times 2\pi) \tag{4}$$

The above equations represent the Hue components for the *i*th pixel. Therefore, it is possible to compute:



**Fig. 2** Demonstration of the circular channel Hue, and the relation between V and S in the HSV cone

$$\text{avg}_{\text{angle}} = \begin{cases} \arccos(\overline{hx}), & \arcsin(\overline{hy}), \\ 2\pi - \arccos(\overline{hx}), & \arcsin(\overline{hy}), \end{cases} \tag{5}$$

$$\text{avg}_{\text{norm}} = \sqrt{\overline{hx}^2 + \overline{hy}^2} \tag{6}$$

$$\text{std} = \frac{\sqrt{\sum_i^{s(I)} ((hx_i - \overline{hx})^2 + (hy_i - \overline{hy})^2)}}{s(I)} \tag{7}$$

where  $\overline{hx}$  and  $\overline{hy}$  stands for the average values of the respective components, *I* the image and *s* is the file size function.

The following modifications in Eqs. (8) and (9) allow the computation of these metrics considering CS:

$$hx_i = \cos(H[x_i, y_i] \times 2\pi) \times CS[x_i, y_i] \quad (8)$$

$$hy_i = \sin(H[x_i, y_i] \times 2\pi) \times CS[x_i, y_i] \quad (9)$$

The average and standard deviation for each channel image (H, S and V) consist of two values, with the exception of the two representations for the Hue channel (H and HCS) that holds four values for the average (norm and angle) and two values for the standard deviation. A total of ten features correspond to average and standard deviation.

### 3.4 Complexity estimates proposed

The features based on complexity estimates can be classified into three groups:

1. Based on Image Compression Error, similar to the ones used in previous works [18, 19], these metrics estimate image complexity by considering the compression rate and error associated with the Fractal and JPEG compression of the images.
2. Based on Zipf's Law, inspired by previous work in the musical field [23], these measurements consider the slope of the trend line of the Zipf distribution of the pixel intensities and the linear correlation with the trend line [43].
3. Based on Fractal Dimension, inspired by the work performed by Taylor et al. [41], the Fractal Dimension of the image and of the image edges was calculated through the box-counting method.

The selection of these metrics was inspired by several studies which associate esthetics with complexity [3], Zipf's Law [23] and Fractal Dimension [41].

### 3.5 Features based on image compression

Some image compression schemes are lossy; therefore, they yield a compression error, i.e., the compressed image will not exactly match the original. All other factors being equal, it is assumed that complex images will tend toward higher compression errors and simple images will tend toward lower compression errors. Additionally, complex images will tend to generate larger files than simple ones. Thus, the compression error and file size are positively correlated with image complexity [15].

Three levels of detail are considered for the JPEG and Fractal compression metrics: low, medium and high. For each compression level, the process is the same, and the image in analysis at a specific time is encoded in a JPEG or Fractal format. It is estimated that image complexity of image  $I$  uses the following equation:

$$\text{Complexity}(I) = \text{RMSE}(I, CT(I)) \times \frac{s(CT(i))}{s(i)} \quad (10)$$

where RMSE stand for the root mean square error,  $CT$  is the JPEG or Fractal compression transformation and  $s$  is the file size function.

A quad-tree fractal image compression scheme [13] is used with the set of parameters given in Table 1. Note that letting the minimum partition level be 3 implies that the selected region is always first partitioned into 64 blocks. Subsequently, at each step, for each block, if one finds a transformation that gives a good enough pixel-by-pixel match, then that transformation is stored and the image block is not further partitioned. Herein, the pixel-by-pixel match is performed with respect to the usual 0 to 255 gray-scale interval encoding. If the pixel-by-pixel match error is more than 8 for at least one of the pixels of the block in the partition, that image block is further partitioned into 4 subblocks, the level increases, and the process is repeated. When the maximum partition level is reached, the best transformation found is stored, even if the pixel-by-pixel match error for the block exceeds 8. The quality settings of the JPEG encoding for low, medium and high level of detail were 20, 40 and 60, respectively.

Romero et al. [31, 33] showed that such features had been previously used to tackle an image classification problem according to esthetic criteria. The only images taken into account in that paper were those related to the V color channel, while for the current study, all the available channels have been used. Cutzu et al. and Athistos et al. noted that color was useful for distinguishing photographs from paintings and computer-generated graphics. For that reason, it was decided to extract information from all color channels.

Taking into account that there are 2 compression methods and 3 levels of detail per method, a total of 6 features are related to each image channel. As for the method based on JPEG, 3 features are also obtained using the different levels of detail on the HSV image.

A total of 21 features correspond to JPEG and Fractal compressions methods.

**Table 1** Image compression parameters

Settings	Low	Medium	High
Image size	256 × 256 pixels		
Minimum partition level	2	2	3
Maximum partition level	4	5	6
Maximum error per pixel	8	8	8

### 3.6 Features based on Zipf's Law

Zipf's Law is the observation of phenomena generated by self-adaptive organisms, such as humans, also known as the principle of least effort. Once the phenomenon has been selected for study, the contribution of each case to the whole is examined and ranked according to its importance or predominance [43].

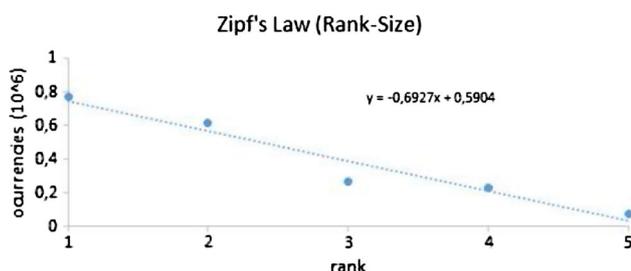
Bill Manaris used the Zipf's Law as a criterion of beauty in music classification [23]. Since 2003, he has been collaborating with our research group in musical classification, mainly using metrics based on Zipf's Law [21, 24]. Taking into account the knowledge and experiences of the group, several metrics based on this principle were added, applying them to the pixel-to-pixel value in monochromatic channels.

In many cases, the statistical rank of an event is inversely proportional to the event size. Informally, little events tend to occur more frequently, while bigger events tend to occur less frequently [23]. The use of size instead of rank creates a size frequency distribution. It is an alternative formulation of Zipf's Law which applies to architecture and city planning [35]. This formulation is also used in the box-counting technique for calculating the Fractal Dimension of phenomena [16].

#### 3.6.1 Zipf rank frequency

The calculation of the Zipf rank frequency metric implies the following: counting the number of occurrences of each pixel intensity value in the image; ordering according to the number of occurrences; tracing a rank vs. number of occurrences plot using a logarithmic scale on both axes; calculating the slope of the trend line and the linear correlation with the trend line [29].

Figure 3 shows a graph including the frequency of English vowels. Once the number of instances of every vowel is counted, it is ordered by rank and the slope of the line linking them is calculated, together with the error of the same line with regard to the dots.



**Fig. 3** Zipf's Law calculation with the distribution of the five English vowels

In a totally random figure, all the pixel intensities would have approximately the same number of occurrences, and therefore, the slope of the line is zero. In an image with a great area of one single color, the slope will be much greater than one.

For the H channel, this metric is calculated in two ways: i) as described above; ii) instead of counting the number of occurrences of each Hue value, the CS channel values of the corresponding pixels are added (and divided by 255 for normalization purposes). The rationale is the following, the perceived H depends on the saturation and value of the corresponding pixel.

#### 3.6.2 Zipf size frequency

The Zipf size frequency metric is calculated exactly like the Zipf rank frequency, but instead of using the pixel value, we will use the difference between that value and its neighbors. For each pixel, we calculate the difference between its value and each of its neighbor pixels [22, 32]. We count the number of occurrences of differences in size 1, size 2 ...size 255. A difference value versus number of occurrences plot is traced using a logarithmic scale in both axes and calculate slope and linear correlation of the trend line [29].

For the H channel, a circular distance is considered. The H channel is scaled to [0; 1], and the following formulas are used:

$$dx_i = \cos(H[x_i, y_i] \times 2\pi) \quad (11)$$

$$dy_i = \sin(H[x_i, y_i] \times 2\pi) \quad (12)$$

$$\text{dist}x = dx_i - dx_{i+1} \quad (13)$$

$$\text{dist}y = dy_i - dy_{i+1} \quad (14)$$

$$\text{distance} = \sqrt{\text{dist}x^2 + \text{dist}y^2} \quad (15)$$

$$\text{circular}_{\text{distance}} = \min(\text{distance}, |\text{distance} - \sqrt{8}|) \quad (16)$$

Since the  $H$  values are between 0 and 1, the max distance is reached when  $(\text{dist}x = 2) \wedge (\text{dist}y = 2)$  that results from  $\text{distance} = \sqrt{2^2 + 2^2} = \sqrt{8}$ . Equation (16) ensures that the minimum distance in angle (e.g., considering the clockwise and counterclockwise distance) is considered.

The H size frequency is also calculated using the CS channel, which is obtained by making the following variations in Eqs. (11) and (12), respectively:

$$dx_i = \cos(H[x_i, y_i] \times 2\pi) \times CS[x_i, y_i] \quad (17)$$

$$dy_i = \sin(H[x_i, y_i] \times 2\pi) \times CS[x_i, y_i] \quad (18)$$

After calculating the distance values, they are always scaled back to [0; 255] before counting the number of occurrences of each distance value. For this metric, a size vs. number of occurrences plot is also traced, using a

logarithmic scale on both axes, and the slope and linear correlation of the trend line are calculated.

As for rank frequency and size frequency, the linear trend line slope (M) and coefficient of correlation (R2) of all histograms are extracted to form 2 features per each channel image. For the H channel, there are 2 more values that correspond to the HCS image. A total of 16 features correspond to both Zipf's Law methods.

### 3.7 Features based on Fractal Dimension

The last metric to be retrieved is the Fractal Dimension based on the box-counting method, chosen for its conceptual simplicity and ease of implementation.

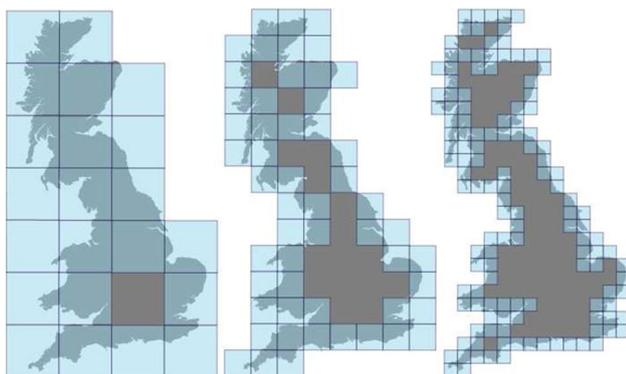
The method starts by binarizing one of the channels in the image (each channel may have a value of 1 or 0 depending on the threshold). The number of black pixels in the image is calculated (dimension 1). The number of squares of size 2 ( $2 \times 2$ ) filled with black pixels is calculated in the image. Later, the number of  $4 \times 4$ , and thus, with every dimension ( $8 \times 8$ ,  $16 \times 16$ ) until we reach the image size. Zipf's Law is calculated with the number of occurrences of each cube size and the quotients are generated.

Figure 4 shows the generation of squares of different sizes in a basic map of the UK.

The features related to the Fractal Dimension are given by the slope of the line obtained by linear regression of points  $\log(2^n) \log(N_n(I))$  [16]. Thus, the linear trend line slope (M) and coefficient of correlation (R2) create 2 features per each channel image. A total of 6 features correspond to the Fractal Dimension.

## 4 Artificial neural network

The classifiers were implemented as a feed forward back-propagation fully connected ANN [34]. Sections 3.3–3.7 show the features related to the average and standard



**Fig. 4** Box-counting measure of Great Britain's coast by grids of decreasing sizes (artwork by Prokofiev)

deviation, together with the features obtained by applying different complexity metrics. On the whole, a group made up of 53 metrics was considered:

- 21 features determined from the compression error using JPEG and Fractal methods;
- 16 features obtained applying the rank frequency and size frequency methods related to the Zipf's Law;
- 6 features related to the Fractal Dimension, calculated through the box-counting method;
- 10 features related to average and standard deviation.

Table 2 shows the metrics used.

As seen in Sect. 3.2, from each image 6 new ones were obtained (applying Canny and Sobel filters in a horizontal and vertical way). For this reason, the resulting group of features, which will be taken into consideration, is made up of 371 metrics (53 complexity features over 7 filters).

Using this set of metrics, 12 ANNs are trained. Each of them will employ a different set of metrics. Table 3 shows the ANN architectures considered. As it can be observed, the main difference lies in the considered input metrics. There are two groups, those considering different sets of metrics but not using edge filters and, on the other hand, those incorporating these filters.

In all cases, the metric values are normalized between 0 and 1 as input to the ANN processing elements. A single hidden layer is used with a series of processing elements, which is half of the total number of input plus output processing elements. Blum [6] proposed a number of hidden units that should be "somewhere between the input layer size ...and the output layer size".

Previous experiments have been carried out with a bigger set of hidden neurons, achieving longer training times and results with no significant differences.

In the output layer, two processing elements are employed in all ANNs. The most active processing element will determine the classification, photography or painting. An identity output function with an output range [0,1] is used.

Having two output neurons allows each neuron to employ different strategies in order to generate each output, since they have connections with different weights to the hidden layer. Thus, two contrary strategies in the same network (one searching for 0 while the other searches for 1) allow a better differentiation between two sets. If we use an output neuron (which, for instance, indicates a painting with  $a < 0.5$  value, and a photograph with  $a > 0.5$  value), we can only use one strategy.

The training files and scripts of the corresponding ANNs, conducted in SNNs, can be downloaded at <https://figshare.com/s/75d8142b683f0482f7af>.

**Table 2** List of metrics employed in the experiments

Number	Metrics	Channel	Number	Metrics	Channel		
1	Jpeg complexity	Low	All	28	Standard deviation	Angle	H
2	Jpeg complexity	Medium	All	29	Standard deviation	Angle	HCS
3	Jpeg complexity	High	All	30	Standard deviation	Value	S
4	Jpeg complexity	Low	H	31	Standard deviation	Value	V
5	Jpeg complexity	Medium	H	32	Zipf rank frequency	M	H
6	Jpeg complexity	High	H	33	Zipf rank frequency	R2	H
7	Jpeg complexity	Low	S	34	Zipf rank frequency	M	HCS
8	Jpeg complexity	Medium	S	35	Zipf rank frequency	R2	HCS
9	Jpeg complexity	High	S	36	Zipf rank frequency	M	S
10	Jpeg complexity	Low	V	37	Zipf rank frequency	R2	S
11	Jpeg complexity	Medium	V	38	Zipf rank frequency	M	V
12	Jpeg complexity	High	V	39	Zipf rank frequency	R2	V
13	Fractal complexity	Low	H	40	Zipf size frequency	M	H
14	Fractal complexity	Medium	H	41	Zipf size frequency	R2	H
15	Fractal complexity	High	H	42	Zipf size frequency	M	HCS
16	Fractal complexity	Low	S	43	Zipf size frequency	R2	HCS
17	Fractal complexity	Medium	S	44	Zipf size frequency	M	S
18	Fractal complexity	High	S	45	Zipf size frequency	R2	S
19	Fractal complexity	Low	V	46	Zipf size frequency	M	V
20	Fractal complexity	Medium	V	47	Zipf size frequency	R2	V
21	Fractal complexity	High	V	48	Box-counting	M	H
22	Average	Angle	H	49	Box-counting	R2	H
23	Average	Norm	H	50	Box-counting	M	S
24	Average	Angle	HCS	51	Box-counting	R2	S
25	Average	Norm	HCS	52	Box-counting	M	V
26	Average	Value	S	53	Box-counting	R2	V
27	Average	Value	V				

**Table 3** Considered ANNs architectures

ANN	PE input	PE hidden	PE output
Base_image (Compr + Zipf + FDim+AVG+STD)	53	27	2
Base_image (Compr + Zipf + FDim)	43	22	2
Base_image (Compr)	21	11	2
Base_image (Zipf)	16	9	2
Base_image (FDim)	6	4	2
Base_image (AVG + STD)	10	6	2
Image and edges (Compr + Zipf + FDim + AVG + STD)	371	186	2
Image and edges (Compr + Zipf+FDim)	301	151	2
Image and edges (Compr)	147	74	2
Image and edges (Zipf)	112	57	2
Image and edges (FDim)	42	22	2
Image and edges (AVG + STD)	70	36	2

#### 4.1 Datasets, training and test

Attempts were made to access data sets used by previous researchers, but these were not available. Consequently, it was decided to create a new one. Two different types of images were used in the present work:

- National Geographic photographs-2625 images with a  $1024 \times 768$  resolution, which represent: nature, animals, landscapes, some journalism and abstract photographs. These images were gathered from the National Geographic Web site. Frames were eliminated manually.

- Paintings—2610 images of several known authors, including Caravaggio, Kandinsky, Picasso, Van Gogh and Dali. All images were extracted from different sources and with a great variation of style.

Figures 5 and 6 show some example images from both sets.

The dataset provides a very wide set of images of both categories. In both cases, the sets have works of great quality and diversity. The painting set comprises works by well-known painters, such as Picasso, Kandinsky or Van Gogh. The set of pictures from National Geographic has a wide diversity (landscapes textures, animals, and abstract photos), while having a very high quality level, including several works which received awards. In order to create a dataset similar to the one used by Cutzu, the frames of the images and gray-scale images were excluded from the dataset.

## 5 Results

As for the training and validation used in this experiment, a method known as tenfold cross-validation [17] have been used, consisting of dividing the pattern set into 10 subsets of the same size. The ANN validation and training has been performed 10 times. In each case, one of the 10 subsets has been used as a validation set and the other 9 as training patterns. Thus, all the patterns have been used once for validation and 9 times for training. The results are the average validation results obtained over the 10 times.

Table 4 shows the most important parameters used in all trained ANNs.

The learning process is finished when a maximum of 1500 cycles or a root mean square error (RMSE) lower than 0.07 is reached. All training patterns are presented once during each cycle. The fact that an error tolerance threshold ( $d_{max}$ ) [37] of 0.3 has been determined means

that when an output of PE deviates less than 0.3 regarding the desired output, no weight update occurs. In other words, during the training process, the interval [0–0.3] is equivalent to 0 and the interval [0.7–1] is equivalent to 1. This has been done in order to avoid overfitting in a binary problem where the ANN provides a real value. Without error tolerance threshold, the ANN must provide a value of exactly 1 or 0 in every painting or photograph, and it has no flexibility to find solutions providing values with minor differences between images of the same category.

Previous experiments have been made with different values of error tolerance threshold (including 0), with results that were worse than those corresponding to the present experiment.

Using tenfold cross-validation, Tables 5 and 6 present the results of the different ANNs. Both tables include the global rate as well as the painting and photograph success rates.

Table 5 presents all ANNs which do not employ any edge detection filters. Table 6 shows the same combination of metrics as shown in Table 5, but applying edge detection filters. Both cases show the error in the test set.

If filters are not applied, as shown in Table 5, the metric which individually provides the best results is the compression one, with 83.25%. In cases where the metrics are combined, the use of all of them provides the best result, 88.92%.

In each of the ANNs in Table 6, the number of PEs of the input layer is seven times that of the corresponding ANNs in Table 5. In these cases, both the original images and the 6 images generated through the application of edge detection filters (Sobel and Canny) are used as ANN input. For example, on the first row in Table 6, for each image, 6 new images (corresponding to the image filtered by canny, vertical canny, horizontal canny, sobel, horizontal sobel, vertical sobel) are generated. Each of these seven images,



**Fig. 5** Examples of photographs



**Fig. 6** Examples of paintings

**Table 4** Parameters relative to the ANNs

Parameter	Setting
Epochs	20
Update function	Topological_order
Input function	Identity
Transfer function	Sigmoid logistic
Output function	Identity
Initial of weights	Random, $[-0.1; 0.1]$
Training algorithm	Backpropagation
Learning rate	0.01
Max. tolerated error	0.3
Convergence criterium	0.07
Runs	10
Cycles	1500

including the original one, is submitted to the metrics of compression, Fractal Dimension, Zipf's Law, average and standard deviation. For row 1, there would be 53 metrics (see Table 5), each of which would generate seven values,

resulting in 371 values corresponding to the PEs of the input layer.

The ANN which provides the best results is the one that uses filters and the full set of metrics, with an overall outcome of 94.82%. It is important to highlight that such a success rate is well balanced between both sets, meaning that our ANN is able to equally differentiate between both sets, 94.67% in the painting set and 94.97% in the photograph one. Thus, when eliminating 70 features corresponding with average and standard deviation (row 2, Table 6), the results are marginally lower, with 94.67% of matches. Such results are also well balanced. The same as the results in Table 6, when using metrics individually, compression is again the one with the best result (92.72%). In all cases, the results obtained by ANNs which employed filters (Table 6) are higher than those which did not (Table 5).

The computational cost of using base images plus edges is bigger than using only the base image; therefore, based on the applications to be employed, it may be worth it to use only the base image, even though the results are worse.

**Table 5** Classification rate achieved according to different features set and not applying edge detection filters

Feature set	Number of features	Global rate (%)	Paintings rate (%)	Photographs rate (%)	Time ANN	Time FE
Base_Image (Compr + Zipf + FDim + AVG + STD)	53	88.92	91.03	86.82	0:18:11	1:14:25
Base_Image (Compr + Zipf + FDim)	43	87.26	86.90	87.26	0:15:51	0:45:50
Base_Image (Compr)	21	83.25	84.64	81.78	0:06:38	0:26:54
Base_Image (Zipf)	16	75.63	76.51	74.74	0:05:24	0:18:48
Base_Image (FDim)	6	69.05	64.71	73.37	0:03:36	0:26:39
Base_Image (AVG + STD)	10	75.09	75.36	74.82	0:04:32	0:09:21

Time ANN and Time FE are the computational time of training the ANN and of extracting the features of all the images in the experiment in HH:MM:SS format. The work environment was integrated by an Intel 4790K with 16Gb of RAM, SSD Kingston V300 and Nvidia GeForce GT730

**Table 6** Classification rate achieved according to different features set and applying edge detection filters

Feature Set	Number of features	Global rate (%)	Paintings rate (%)	Photographs rate (%)	Time ANN	Time FE
Image and edges (Compr + Zipf + FDim + AVG + STD)	371	94.82	94.67	94.97	8:01:09	7:31:45
Image and edges (Compr + Zipf + FDim)	301	94.67	94.98	94.36	6:39:06	7:30:13
Image and edges (Compr)	147	92.72	91.80	93.64	2:39:04	4:41:43
Image and edges (Zipf)	112	88.98	89.95	88.11	1:38:18	1:32:35
Image and edges (FDim)	42	75.57	76.36	74.78	0:15:34	1:43:17
Image and edges (AVG + STD)	70	85.96	86.74	85.18	0:34:55	0:40:14

Time ANN and Time FE are the computational time of training the ANN and of extracting the features of all the images in the experiment in HH:MM:SS format. The work environment was integrated by an Intel 4790K with 16Gb of RAM, SSD Kingston V300 and Nvidia GeForce GT730

The average time for extracting a complete set of one images characteristics (a total of 371) is 00:05.2 seconds.

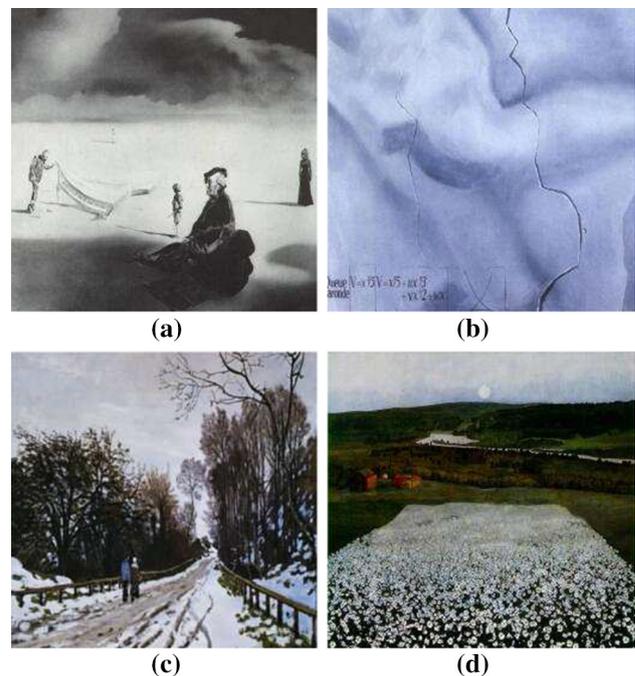
A priori it may seem that the task carried out is trivial in many cases. In order to prove the reader the difficulty of such a classification, Figs. 7 and 8 show some randomly chosen examples of images that our best set was not able to correctly classify.

As for Fig. 7c, d, it may be observed that some paintings related to landscapes with such a reduced image detail ( $256 \times 256$  pixels) may lead to a misinterpretation by the observer. In Fig. 8b, the error is perfectly understandable since it is a photograph taken of a cave painting, and therefore, this image could have perfectly belonged to the set of paintings. On the other hand, the effect of movement in Fig. 8c could be mistaken for the strokes of a painting. In any case, it can be observed that it is not a trivial classification task and that the errors made in the classification could also have been made by a human observer.

## 6 Conclusions and future work

The complexity features and edge detection filter sets have been successfully used in order to distinguish between photographs and paintings. They have also been compared with another feature set obtained by applying the same statistical formulae both on the original image and on the six auxiliary images obtained by applying the Canny and Sobel edge detection filters. In both cases, the exclusive use of features related to the compression error using the JPEG and Fractal methods has turned out to be more suitable.

The experiments carried out have also shown how an ANN with different complexity features with the Canny and Sobel filters allows us to distinguish between both sets with a match rate of 94.82%. It should be borne in mind that the best result obtained so far in this kind of tasks is



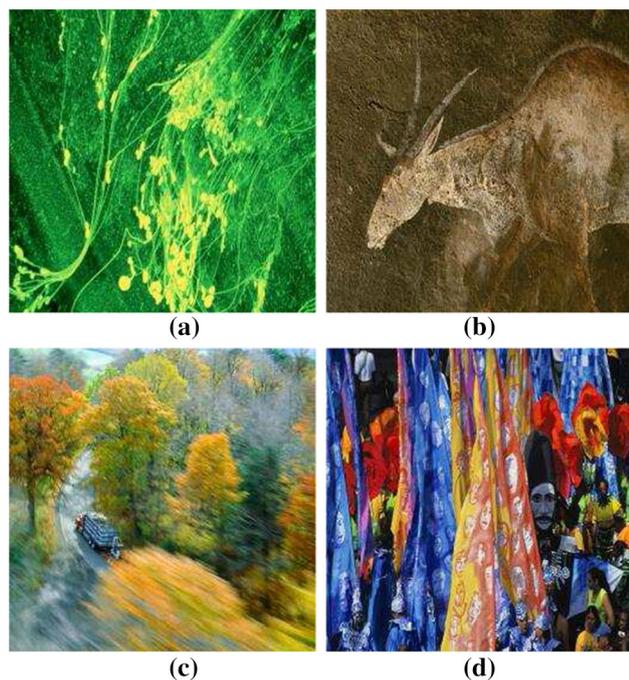
**Fig. 7** Example of paintings which have been wrongly classified by filters (Compr+Zipf+ FDim)

93% using three classifiers in parallel, although with a different set of images.

Some of the best feature set errors match pictures that even the human eye has difficulty evaluating properly.

It is our plan to continue searching and studying new features, as well as improving the current ones, for a better solution to this problem. Other classifier architectures will be studied, implemented and tested. Refinement of the current dataset and building distinct ones will also be explored in further research studies.

We will plan to explore other styles of photographs and paintings, for instance to distinguish between human face



**Fig. 8** Example of photographs which have been wrongly classified by filters (Compr+Zipf+FDim)

photographs (e.g., from passports) and paintings of faces (e.g., self-portraits).

**Acknowledgements** We are very grateful for the reviewers suggestions. This work was supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia (Ref. GRC2014/049) and the European Fund for Regional Development (FEDER) in the European Union, the Portuguese Foundation for Science and Technology in the scope of project SBIRC (Ref. PTDC/EIA/EIA/115667/2009), Xunta de Galicia (Ref. XUGA - PGIDIT - 10TIC105008-PR) and the Spanish Ministry for Science and Technology (Ref. TIN2008-06562/TIN).

## References

1. Aks DJ, Sprott JC (1996) Quantifying aesthetic preference for chaotic patterns. *Empir Stud Arts* 14(1):1–16
2. Arnheim R (1954) *Art and visual perception: a psychology of the creative eye*. Univ of California Press, California
3. Arnheim R (1966) *Towards a psychology of art/entropy and art? An essay on disorder and order*. The Regents of the University of California, California
4. Athitsos V, Swain MJ, Frankel C (1997) Distinguishing photographs and graphics on the world wide web. *Content-Based Access of Image and Video Libraries, 1997. Proceedings. IEEE Workshop on, IEEE*, pp 10–17
5. Birkhoff GD (1933) *Aesthetic measure*. Mass, Cambridge
6. Blum A (1992) *Neural networks in C++: an object-oriented framework for building connectionist systems*. Wiley, New York
7. Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 8(6):679–698
8. Cutzu F, Hammoud R, Leykin A (2003) Estimating the photorealism of images: distinguishing paintings from photographs. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, IEEE*, 2: II–305
9. Cutzu F, Hammoud R, Leykin A (2005) Distinguishing paintings from photographs. *Comput Vision Image Underst* 100(3):249–273
10. Datta R, Joshi D, Li J, Wang JZ (2006) Studying aesthetics in photographic images using a computational approach. In: Leonardis A, Bischof H, Pinz A (eds) *Computer Vision – ECCV 2006. Lecture notes in computer science*, vol 3953. Springer, Berlin, Heidelberg, pp 288–301
11. Eysenck HJ (1941) The empirical determination of an aesthetic formula. *Psychol Rev* 48(1):83
12. Eysenck HJ (1942) The experimental study of the ‘good gestalt’? A new approach. *Psychol Rev* 49(4):344
13. Fisher Y (1994) Fractal image compression. *Fractals* 2(03):347–361
14. Forsythe A, Nadal M, Sheehy N, Cela-Conde CJ, Sawey M (2011) Predicting beauty: fractal dimension and visual complexity in art. *Br J Psychol* 102(1):49–70
15. Greenfield G, Machado P (2009) Simulating artist and critic dynamics—an agent-based application of an evolutionary art system. In: Dourado A, Rosa AC, Madani K (eds) *IJCCI 2009—Proceedings of the international joint conference on computational intelligence*. INSTICC Press, Funchal, Madeira, pp 190–197
16. Huttenlocher DP, Klanderman GA, Rucklidge WJ (1993) Comparing images using the hausdorff distance. *IEEE Trans Pattern Anal Mach Intell* 15(9):850–863
17. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI Morgan Kaufmann, Burlington*, pp 1137–1145
18. Machado P, Cardoso A (1998) Computing aesthetics. In: de Oliveira FM (ed) *Advances in artificial intelligence. SBIA 1998. Lecture notes in computer science*, vol 1515. Springer, Berlin, Heidelberg, pp 219–228
19. Machado P, Cardoso A (2002) All the truth about nevar. *Appl Intell* 16(2):101–118
20. Machado P, Romero J, Santos ML, Cardoso A, Manaris B (2004) Adaptive critics for evolutionary artists. In: Raidl GR et al (eds) *Applications of evolutionary computing. EvoWorkshops 2004. Lecture notes in computer science*, vol 3005. Springer, Berlin, Heidelberg, pp 437–446
21. Machado P, Romero J, Manaris B (2008) Experiments in computational aesthetics. In: *The art of artificial evolution*, Springer, pp 381–415
22. Machado P, Romero J, Nadal M, Santos A, Correia J, Carballal A (2015) Computerized measures of visual complexity. *Acta Psychol* 160:43–57. doi:10.1016/j.actpsy.2015.06.005, <http://www.sciencedirect.com/science/article/pii/S0001691815300160>
23. Manaris B, Purewal T, McCormick C (2002) Progress towards recognizing and classifying beautiful music with computers—midi-encoded music and the Zipf-Mandelbrot law. *SoutheastCon, 2002. Proceedings IEEE, IEEE*, pp 52–57
24. Manaris B, Romero J, Machado P, Krehbiel D, Hirzel T, Pharr W, Davis RB (2005) Zipf’s law, music classification, and aesthetics. *Comput Music J* 29(1):55–69
25. Manaris B, Roos P, Machado P, Krehbiel D, Pellicoro L, Romero J (2007) A corpus-based hybrid approach to music analysis and composition. In: *Proceedings of the 22nd National Conference on Artificial Intelligence, Vol 1, AAAI Press, AAAI’07*, pp 839–845
26. Meier NC (1942) *Art in human affairs; an introduction to the psychology of art*. McGraw-Hill, New York
27. Moles AA (1957) *Théorie de l’information et perception esthétique*. *Rev Philos de la France et de l’Étranger* 147:233–242
28. Nadal Roberts M (2007) *Complexity and aesthetic preference for diverse visual stimuli*. Ph.D. thesis, Universitat de les Illes Balears

29. Powers DM (1998) Applications and explanations of Zipf's law. In: Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, pp 151–160
30. Rigau J, Feixas M, Sbert M (2005) An information-theoretic framework for image complexity. In: Proceedings of the First Eurographics conference on Computational Aesthetics in Graphics. Visualization and Imaging, Eurographics Association, pp 177–184
31. Romero J, Machado P, Carballal A, Osorio O (2011) Aesthetic classification and sorting based on image compression. In: Di Chio C et al (eds) Applications of evolutionary computation. EvoApplications 2011. Lecture notes in computer science, vol 6625. Springer, Berlin, Heidelberg, pp 394–403
32. Romero J, Machado P, Carballal A, Correia J (2012a) Computing aesthetics with image judgement systems, Springer, Berlin, pp 295–322. doi:10.1007/978-3-642-31727-9\_11.
33. Romero J, Machado P, Carballal A, Santos A (2012b) Using complexity estimates in aesthetic image classification. *J Math Arts* 6(2–3):125–136
34. Rumelhart DE, Hinton GE, Williams RJ (1988) Neurocomputing: foundations of research. MIT Press, Cambridge. chap Learning representations by back-propagating errors, pp 696–699
35. Salingaros NA, West BJ (1999) A universal rule for the distribution of sizes. *Environ Plan B* 26:909–924
36. Saunders R, Gero JS (2001) Artificial creativity: a synthetic approach to the study of creative behaviour. Computational and cognitive models of creative design V, Key Centre of Design Computing and Cognition. University of Sydney, Sydney, pp 113–139
37. Schiffmann W, Joost M, Werner R (1994) Optimization of the backpropagation algorithm for training multilayer perceptrons. Technical report, University of Koblenz, Institute of Physics, Rheinland
38. Sobel I (1990) An isotropic  $3 \times 3$  image gradient operator. *Mach Vision Three Dimens Sci* 3:376–379
39. Spehar B, Clifford CW, Newell BR, Taylor RP (2003) Universal aesthetic of fractals. *Comput Gr* 27(5):813–820
40. Svängård N, Nordin P (2004) Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In: Raidl GR et al (eds) Applications of evolutionary computing. Lecture notes in computer science, vol 3005. Springer, Berlin, Heidelberg, pp 447–456
41. Taylor RP, Micolich AP, Jonas D (1999) Fractal analysis of pollock's drip paintings. *Nature* 399(6735):422–422
42. Voss RF, Clarke J (1978) "1/f noise" in music: music from 1/f noise. *J Acoust Soc Am* 63:258
43. Zipf GK (1949) Human behavior and the principle of least effort. Addison-Wesley press, Boston