# 18

# Experiments in Computational Aesthetics

## An Iterative Approach to Stylistic Change in Evolutionary Art

Penousal Machado,[1] Juan Romero,[2] and Bill Manaris[3]

[1] CISUC, Department of Informatics Engineering, University of Coimbra, 3030 Coimbra, Portugal machado@dei.uc.pt
[2] Faculty of Computer Science, University of Coruña, Coruña, Spain jj@udc.es
[3] Computer Science Department, College of Charleston, Charleston, SC 29424, USA manaris@cs.cofc.edu

**Summary.** A novel approach to the production of evolutionary art is presented. This approach is based on the promotion of an arms race between an adaptive classifier and an evolutionary computation system. An artificial neural network is trained to discriminate among images previously created by the evolutionary engine and famous paintings. Once training is over, evolutionary computation is used to generate images that the neural network classifies as paintings. The images created throughout the evolutionary run are added to the training set and the process is repeated. This iterative process leads to the refinement of the classifier and forces the evolutionary algorithm to explore new paths. The experimental results attained across iterations are presented and analyzed. Validation tests were conducted in order to assess the changes induced by the refinement of the classifier and to identify the types of images that are difficult to classify. Taken as a whole, the experimental results show the soundness and potential of the proposed approach.

## 18.1 Introduction

We are interested in the development of Artificial Artists (AAs), i.e., artificial systems with artistic capabilities similar to their human counterparts. In our view, an AA should be able to perform aesthetic and/or artistic judgments – i.e., be able to assess the merits of the artworks it creates, as well as the works of other, artificial or human, artists [1] – and to adapt to the requirements of a dynamic hybrid society [2], populated by artificial and human agents. Taking this into consideration, our architecture for the development of AAs comprises two modules: a Creator and an Artificial Art Critic (AAC) [3].

Although, the results presented in this paper concern the visual domain, we are also interested in the music domain, and on the cross-transfer of concepts between both. Therefore, our approach to aesthetics in the visual domain is

informed by our work in aesthetic modeling and classification in the music domain [4, 5].

We consider that the ability to learn how to perform aesthetic judgments is vital. It allows the system to be fully autonomous, and to use the works of other artists as source of inspiration [1]; also it creates some of the preconditions for stylistic change in the system's artistic performance, enabling it to explore, or set, new trends [3, 2].

The results presented in this chapter are a step in that direction. In particular, the main research goal is to develop a system that: (i) builds its own aesthetic model from a set of examples (thus allowing it to be influenced by other artists); and (ii) autonomously modifies its artistic style.

To achieve this goal, we explore an approach where the role of the creator is played by an Evolutionary Computation (EC) engine, and the role of the AAC by a classifier that uses a Feature Extractor (FE) and an Artificial Neural Network (ANN)-based Evaluator. Our proposal has two distinctive characteristics:

1. The use of an ANN to distinguish between images generated by the EC engine and a selected set of external images (e.g., famous paintings, artworks of a given style, landscape photographs, portraits, etc.).
2. The iterative execution of the following steps:
   - The EC engine tries to find images that are classified as external ones; the fitness is a function of the output of the ANN.
   - Once the EC run is over, the created images are added to the training set of the ANN as instances of internal images.
   - The ANN is trained to distinguish between the two sets.

Conceptually, this approach can be seen as a compromise between approaches with a static/global fitness function and those with dynamic/contextual fitness assignment such as co-evolutionary ones.

The external set of target images constitutes an "inspiring set", a stable attractor that is meant to ensure that the evolved imagery tends to incorporate aesthetic qualities recognized by humans. On the other hand, the systematic addition of the evolved images to the training set as "counter-examples" and the subsequent training of the ANN causes a competition between evolver and classifier, allowing us to attain a dynamic behavior. The fitness changes from iteration to iteration, hence promoting stylistic change.

The proposed approach has been tested with an external set of 3322 images of renowned painters and an internal set of EC-generated images. The employed AAC architecture [3] has been used in the music domain in author identification tasks, and in classification experiments related to aesthetic judgment [5, 6]. In the field of visual art, this architecture has been used in author identification tasks [7, 8, 9]. The employed FE incorporates ideas inspired by the use of complexity [10], fractal dimension [11, 12], and the Zipf–Mandelbrot law [13, 14, 7, 15] in both the visual and music domains.

Our experimental results show that the AACs used are able to discriminate between the two sets and to guide the EC algorithm. Conversely, in each iteration, the EC algorithm is able to find images that are classified as external. Therefore, an autonomous neuro-evolutionary framework able to perform stylistic variation is attained. In Manaris et al. [6] we explore a similar approach in the music domain.

The chapter is structured as follows: we begin with an overview of previous work on automatic fitness assignment and computational aesthetics (Sect. 18.2), and with a short overview of our previous work on fitness automation in the field of visual art (Sect. 18.2); in Sect. 18.4 we describe our approach; this is followed by a global overview of the system and an in-depth look at its main modules (Sect. 18.5); the experimental results attained in the visual domain and their analysis are presented in Sect. 18.6; next, we present the results attained in several validation tests; finally, in Sect. 18.8, we draw overall conclusions, and indicate other research directions and application areas.

We include all images generated throughout the evolutionary runs in the accompanying DVD, as well as the corresponding fitness values. Although the number of images, over 30000, is probably too large for a close inspection, browsing over them will, hopefully, allow the reader to get a better grasp of the types of imagery produced in each iteration.

## 18.2 State of the Art

One of the main difficulties in the application of EC to artistic tasks is the development of appropriate fitness assignment schemes.

Fitness assignment plays an important role in any EC system; artistic tasks are not an exception. Focusing on the domain of visual art, there are essentially five approaches to fitness assignment: interactive evolution – an approach that has been popularized by Karl Sims [16] (see Chap. 1 for a wider list of references); similarity based – i.e., evolving towards a specific image or images (see Chap. 1); hardwired fitness functions [17, 18, 19, 20]; machine-learning approaches [21]; and co-evolutionary approaches [12, 22] (see Chap. 17). The combination of several of the above methods has also been explored, for instance Saunders and Gero [12] use ANNs in the context of a co-evolutionary approach and Machado et al. [8, 9] combine interactive evolution with a hardwired fitness function.

Taking into account that we are interested in systems able to perform aesthetic judgments, interactive evolution and similarity-based approaches are not of particular relevance for the present chapter. The remaining approaches pose several complex problems.

Even if we consider that there is such a thing as a global aesthetic value that can be objectively measured, or that we are only interested in mimicking the aesthetic preferences of a particular user, building a hardwired function capable of assessing it is definitely a difficult task. If we take into consideration

that this function will be used in conjunction with EC algorithms, which are prone by nature to explore the shortcomings of the fitness function to maximize fitness, the scenario becomes even worse.

Using a machine-learning approach can alleviate some of the burden of coding the fitness function. Nevertheless, learning to evaluate aesthetic judgments has proven to be a complex task. The work of Baluja et al. [21] is one of the few published attempts at doing so, and it is also the closest one to the approach presented in this chapter. Baluja et al. [21] use interactive evolution to build a set of evaluated images, in a later stage these images are used to train an ANN that receives as input an image and produces an evaluation. Several ANN architectures have been tested.

Although the approach is elegant, the results are far from being a success [21]. The ANN is able to train, yet it fails to generalize properly. In the best configuration found, the error in the test set is 0.2, which is only marginally superior to the results attainable by using a random function, biased to match the probabilistic distribution of the training set, to assign fitness (0.24) [21].

Independently of the fitness function being built or learned, static approaches to fitness assignment share a common problem: the EC algorithm tends to converge to a specific type of imagery that depends on the particularities of the fitness function and of the EC algorithm.

Co-evolutionary approaches overcome, to some extent, this limitation. However, it is difficult to incorporate aesthetic criteria in the evaluation scheme.[4] Even if this difficulty is overcome, ensuring that the evolved imagery relates to human aesthetics is not a trivial task. As Todd and Werner state:

> "One of the biggest problems with our coevolutionary approach is that, by removing the human influence from the critics ... the system can rapidly evolve its own unconstrained aesthetics." [23]

Taking into consideration the limited number of EC systems, in the field of visual art, where an attempt to incorporate aesthetic criteria in fitness assignment is made, it becomes relevant to consider approaches that, while not related to EC, are of pertinence to the field of computational aesthetics.

The work of Birkhoff [24] is probably the first attempt to present a formal measurement of aesthetics. Birkhoff suggests that aesthetic value results from the ratio between order and complexity, applying this principle to measure the aesthetic value of several 2D contours. The works of Moles [25], Arnheim [26, 27, 28] and Bense [29], draw upon the ideas of Birkhoff, bringing into play Shannon's [30] Information Theory. More recently, Staudek [31, 32] also presents an aesthetic theory where notions such as chaos and complexity play an important role.

Machado and Cardoso [10] use complexity estimates, based on JPEG and fractal image compression, to estimate the aesthetic value of grayscale images,

[4] A thorough description and analysis of co-evolutionary approaches in the field of visual art can be found in Chap. 17.

attaining results higher than the human average in Maitland Graves' "Design Judgment Test" [33]. This work was the basis for the automated fitness assignment scheme [17, 9] which we briefly describe in Sect. 18.3.

Svangärd and Nordin [20] also resort to complexity estimates based on compression schemes to evaluate images. Ritendra et al. [34] resort to a set of features – which includes texture, colorfulness, shape convexity and familiarity measures – and an support vector machine-based classifier to discriminate between photographs with high and low ratings, using as data source an online photo sharing Web site (Photo.net).

The concept of fractal dimension (FD) has also been considered a relevant aesthetic feature [35]. Taylor et al. [11] show the evolution of the FD of Jackson Pollock's paintings, later exploring the use of this technique to authenticate them. Finally, they study the relations between FD and aesthetics [36]. Interestingly, FD has also been used in Evolutionary Art (EA) to automate fitness assignment [37, 12].

## 18.3 Background Work on Automated Fitness Assignment in Visual Art

Inspired by the works of Moles [25] and Arnheim [26, 27, 28] and by studies that indicate a preference for simplified representations of the world, and a tendency to perceive it in terms of regular, symmetric and constant shapes [38, 27, 39, 40], Machado and Cardoso [10, 17, 9] have explored the following hypothesis: Aesthetic value is related to the sensorial and intellectual pleasure resulting from finding a compact percept (internal representation) of a complex visual stimulus.

This approach rewards images that are simultaneously visually complex and easy to perceive, employing estimates for the Complexity of the Percept (CP) and for the Complexity of the Visual Stimulus (CV). CP and CV are estimated through the division of the root mean square error (RMSE) by the compression ratio resulting, respectively, from the fractal (quadratic tree based) and JPEG encoding of the image.

Additionally, a temporal component is also considered [10, 17, 9]. As time passes the level of detail in the perception of the image varies. It is therefore necessary to estimate CP for different moments in time, in this case $t_0$ and $t_1$, which is attained by performing fractal image compression with increasing levels of detail. The proposed approach values images where CP is stable for different levels of detail. To capture the previously described notions the following formula was proposed [10]:

$$\text{value} = \frac{CV^a}{\left(CP(t_1) \times CP(t_0)\right)^b} \times \frac{1}{\left(\frac{CP(t_1)-CP(t_0)}{CP(t_1)}\right)^c} \tag{18.1}$$

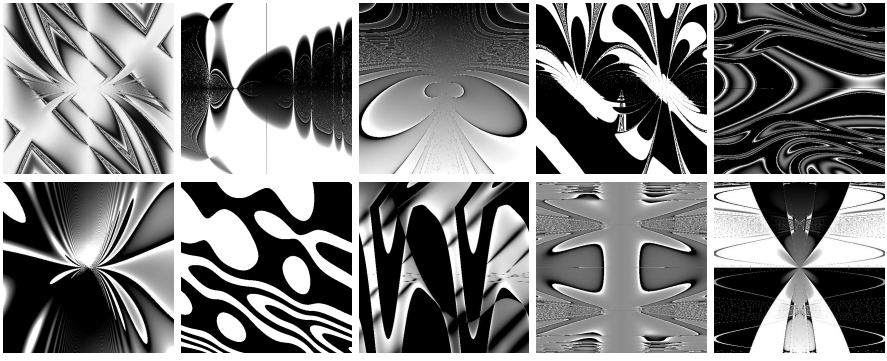


**Fig. 18.1.** Best individuals from 10 independent runs

where $a$, $b$ and $c$ are parameters used to adjust the importance given to each of the components. The left side of the formula rewards images that have, simultaneously, high CV and low CP estimates. The right side rewards images whose CP is stable across time. The division by $CP(t_1)$ is a normalization operation.

To apply this set of ideas in an evolutionary context, we limit the different components of the formula, as follows:

$$\text{value} = \frac{\min(\alpha, CV)^a}{\max(\beta, CP(t_1) \times CP(t_0))^b} \times \frac{1}{\max\left(\gamma, \frac{CP(t_1)-CP(t_0)}{CP(t_1)}\right)^c} \tag{18.2}$$

where $\alpha$, $\beta$ and $\gamma$ are constants defined by the user.

Machado and Cardoso [17, 9] conducted several experiments, using the Genetic Programming (GP) engine of NEvAr[5] and (18.2) as fitness function.

The results attained with this autonomous EA system are quite surprising [17, 9]. Although the approach has several shortcomings – e.g., it only deals with grayscale images – it allows the evolution of a wide set of imagery with arguable aesthetic merit. In Fig. 18.1, we present the fittest images from several independent runs.

In a subsequent study [8, 9], a variation of this approach was used in the context of a partially interactive system. In this variation, the user was allowed to specify optimum values for $CV$, $CP(t_1) \times CP(t_0)$, and $(CP(t_1) - CP(t_0))/CP(t_1)$. The user was also able to intervene at any stage of the evolutionary run, supplying fitness values to the current population. The evaluations performed by the user took precedence over the ones made by the system. Using this variation it became possible to overcome some of the shortcomings of the previous approach, including the limitation to grayscale images.

[5] NEvAr stands for "Neuro Evolutionary Art". In Sect. 18.5.1 a brief overview of the system is provided.

## 18.4 The Approach

Succinctly, the proposed approach can be described as follows:

1. A set of *external* images is selected.
2. Using the EC system, a set of randomly generated *internal* images is created.
3. The ANN is trained to distinguish among *internal* and *external* imagery.
4. A new EC run is started. The output of the ANN is used to assign fitness. Images classified as *external* have higher fitness than those classified as *internal*.
5. The EC run stops when a termination criterion is met (e.g., a pre-established number of generations, attaining a given fitness).
6. The images generated by the EC algorithm are added to the set of *internal* images.
7. The process is repeated from step 3.

One of the key aspects of this approach is the definition of two classes of images. The first class contains *external imagery*. Images that were not created by the GP system and that are usually considered "interesting" or of "high aesthetic value". This class represents an "inspiring set". In the experiments presented in this chapter we employ a set of paintings made by famous artists. The second class contains *internal imagery*, i.e., images previously generated by the GP engine. Although part of these images may be considered interesting by some, for the purposes of the present chapter this class represents undesirable imagery – we are interested in stylistic change, therefore, in this context, images that were already created by the system are not desirable. Nevertheless, and for different purposes, the inclusion of remarkable images generated by the GP engine in the inspiring set may be appropriate.

Like Baluja et al. [21], we use ANNs to assign fitness to the evolved images. In their work, the ANN is trained to mimic the evaluations performed by users in previous interactive runs.

This poses problems, such as: (i) it is difficult to create a set of consistently evaluated images; and (ii) it is difficult to ensure that such a set is representative of the range of imagery that the system may produce. In our proposal the ANN is trained to discriminate between *internal* and *external* images. These sets can be objectively defined, which solves the first problem. Regarding the second problem, although the set of *external* images should be representative of a given type of imagery, this is arguably easier than creating a set that is representative of all the images that an EA tool can create (note that, for instance, NEvAr can, in theory, generate any image [17, 9]). Additionally, the internal set is iteratively expanded, which also contributes to overcome the second problem.

The approach of Baluja et al. [21] has another type of limitation: since the training images are generated by the GP system, even if the ANN is able to reproduce the evaluations of the user(s) (note that this was not the case

[21]), its use to assign fitness will, most likely, lead to the evolution of images that were already generated in interactive runs, or which are similar to them. Although this can be valuable, the generation of novel imagery would be far more interesting.

In our approach, the task of the evolutionary module is to evolve images that the ANN classifies as external imagery. This may be accomplished by evolving images that:

1. are similar (from the perspective of the ANN) to those belonging to the *external* set;
2. are different from the set of GP-generated images used to train the ANN (e.g., images that are entirely novel, hence dissimilar from both sets).

Once such images are found, they become part of the *internal imagery* of the system and are used to train the ANN that will be used in the next iteration, which forces the GP to explore new paths in subsequent iterations.

In the long run, there are two possible final scenarios: (i) the EC system becomes unable to find images that are classified as external; (ii) the ANN becomes unable to discriminate between internal and external imagery.

The first outcome reveals a weakness of the EC engine. This can be caused by a wide variety of factors, for instance: the set of EC parameters may be inadequate; the fitness landscape may be deceptive; etc.

In the second outcome, there are two possible sub-scenarios: (ii.a) the images created by the EC system are similar to some of the external images, which implies that the EC and the classifier are performing flawlessly; (ii.b) the images created by the EC system are stylistically different from the external imagery provided.

The second sub-scenario indicates a weakness of the classifier system. In theory, this can indicate: the existence of stylistic differences that are not captured by the set of features; that the employed ANN and training technique is not able to take full advantage of the provided features; that the settings used in the training of the ANN were not appropriate; etc.

Distinguishing between situations (ii.a) and (ii.b) may encompass some degree of subjectivity. Nevertheless, if that were the case, this difficulty alone would imply, that a considerable success was attained, i.e., an autonomous EC system capable of producing artworks that are arguably similar to those made by humans.

## 18.5 Description of the System

The system employed, schematically represented in Fig. 18.2, uses a GP engine to generate images and an AAC to classify them.

The GP engine, described briefly in the next section, is the same engine employed in NEvAr (a detailed description can be found in Machado and Cardoso [17, 9]). The AAC is presented in Sect. 18.5.2. The integration of both systems is discussed in Sect. 18.5.3.

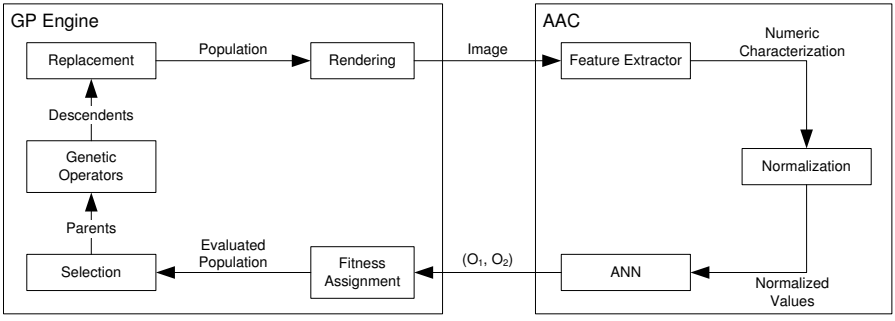



**Fig. 18.2.** Overview of the system

### 18.5.1 Genetic Programming Engine

NEvAr is an expression-based EA system (inspired by the work of Karl Sims [16]) that allows the evolution of populations of images.

NEvAr employs GP. As such, the genotypes are trees constructed from a lexicon of functions and terminals. The function set is composed mainly of simple functions such as arithmetic, trigonometric and logic operations. The terminal set is composed of two variables, $x$ and $y$, and random constants. The phenotype (image) is generated by evaluating the genotype for each $(x, y)$ pair belonging to the image. Thus, the images generated by NEvAr are graphical portrayals of mathematical expressions. As usual, the genetic operations (recombination and mutation) are performed at the genotype level. In order to produce color images, NEvAr resorts to a special kind of terminal that returns a different value depending on the color channel being processed.

The initial versions of NEvAr allowed only user-guided evolution. Later, the system was expanded by the integration of modules that allow fully or partial automation of the fitness assignment (see Sect. 18.3).

Figure 18.3 displays typical imagery produced via interactive evolution by several users.[6]

Obviously, different users may have different preferences. The tastes of a given user may also change from time to time, leading to the exploration of distinct evolutionary paths. User fatigue also tends to decrease the consistency of the evaluations. Additionally, the user may get tired of a particular type of imagery; therefore, novelty may become more important to the user than the aesthetic qualities of the image.

Although user-guided evolution is characterized by subjectivity, by inconsistency, and by the search for novelty, the interaction between system and user typically results in a type of image, an identifiable *signature* (in the sense defined by Cope [41]). This signature depends on the particularities of NEvAr

[6] Further samples can be found on NEvAr's Web site: `http://eden.dei.uc.pt/~machado/NEvAr/index.html`.

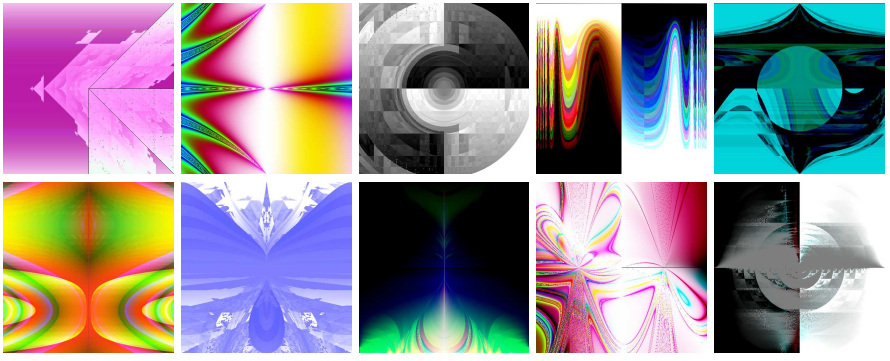


**Fig. 18.3.** Examples of images generated with NEvAr by different users

(primitives, genetic operators, genotype–phenotype mapping, etc.) and on the evaluations performed by the user that directly affect the fitness landscape. Nevertheless, in theory, any image can be generated [17, 9] and, therefore, stylistic change is possible.

### 18.5.2 Artificial Art Critic

Our AAC architecture [3] has been used in the field of visual art for author identification tasks [7, 8, 9]. In the musical domain it was tested in experiments on author identification, and in pleasantness and popularity prediction [5, 6].

It is composed of two modules, an FE and an Evaluator. The FE makes an analysis of the image, measuring several characteristics which are thought to be aesthetically relevant. Based on the collected measurements, the evaluator, implemented by means of an ANN, performs the classification task.

The current version of the feature extractor includes two types of complexity estimates based on JPEG and fractal compression. As in previous work, (Sect. 18.3) the ratio between the RMSE and the compression rate is used to estimate complexity. The fractal and JPEG estimates are calculated at three levels of detail, which is accomplished by establishing different upper bounds for the error per pixel.

The image is split into its Hue, Saturation and Value channels. For each channel the FE calculates the above-mentioned features. Inspired by previous results attained with similar metrics in music classification [4], the FE also comprises Zipf-based metrics. Namely, the rank-frequency (Zipf [42]) distribution (slope and linear correlation, R2, of the trendline) of the Hue, Saturation and Value.

Additionally, for each channel, the FE also determines:

1. The average value of each channel;[7]

[7] Since the Hue channel is circular, we compute its average angle.

2. The standard deviation of the value (STD);

For the Value channel, the FE also estimates the FD of the image, edges, and horizontal and vertical edges. Each of these measurements results in two values, the FD and the linear correlation with the FD.

The FD is measured using an approach similar to the one employed by Taylor et al. [11]: the image is converted to black and white and the FD estimated using the box-counting technique. To calculate the FD of the edges, a Sobel filter is employed to detect them (see e.g., [43]) and the FD of the resulting image is calculated.

In Table 18.1 we present the different groups of features and the components of the image to which they are applied.

**Table 18.1.** Characteristics considered by the FE and components of the image to which they are applied

| Feature | Image | Hue | Saturation | Value |
|---|---|---|---|---|
| JPEG complexity | X | X | X | X |
| Fractal complexity | | X | X | X |
| Average and STD | | X | X | X |
| Zipf distribution and corresponding R2 | | X | X | X |
| FD and corresponding R2 | | | | X |

To determine the variation of the considered characteristics, we partition the image into five regions of the same size – the four quadrants, and an overlapping central rectangle – and compute the previously described measurements for each partition. This process yields a total of 246 measurements.

This FE is an improvement over the one used in previous experiments that did not include an evolutionary component [7, 8, 9]. Taking into account the results attained in those tasks, we decided to use a similar one to discriminate between internal and external imagery. The main difference between the FE used in previous experiments and the current one is the measurement of the FD of several characteristics of the image.

The evaluator is composed of a feed-forward ANN with one hidden layer. We resorted to Stuttgart Neural Network Simulator (SNNS) [44] for training purposes. Standard backpropagation was employed. The values resulting from the feature extractor are normalized between 1 and −1. We adopt an architecture similar to the one used in author identification experiments. We use ANNs with one input unit per feature, 12 units in the hidden layer and 2 units in the output layer (one for each category).

### 18.5.3 Integration of the AAC and GP Engine

The fitness of the images is determined by the output of the AAC and requires five steps: (i) rendering the image; (ii) extracting the features; (iii) normalizing

the feature values and feeding them to the ANN; (iv) determining the ANN output; (v) calculating the fitness value.

In essence, the ANN is trained to perform a binary classification task. This creates a problem: a fitness landscape composed exclusively by values close to zero or close to one will definitely cause problems for the GP system. In other words, a binary classification is not adequate to guide evolution, intermediate values are necessary.

To overcome this difficulty, while training the ANN, we allow differences between desired and obtained output values. If the difference is below the maximum tolerated error threshold, then it is propagated back as being zero (i.e., non-existent). In the experiments presented in the following sections the maximum tolerated error is set to 0.3. This allows us to get outputs that are not in the limits of the $[0, 1]$ interval, creating a smoother fitness landscape.

We use ANNs with two output neurons. The activation value of the first neuron, $O_1$, determines the degree of belonging to the class of selected external images. The activation of the second output neuron, $O_2$, determines the degree of belonging to the class of internal images.

Considering this architecture – and a maximum tolerated error of 0.3 – it is possible to devise fitness functions to evolve images that:

1. are classified as belonging to one category and not to the other (one neuron with an activation value above 0.7 and the other with an activation below 0.3);
2. are simultaneously classified as belonging to both categories (both neurons with an activation above 0.7);
3. are not classified as belonging to any of the categories (both neurons with an activation below 0.3);
4. the ANN is unable to classify with certainty (both neurons with an activation value in the $[0.3, 0.7]$ interval).

We are interested in evolving images that are classified as belonging to the set of paintings considered, $O_1 \geq 0.7$, and as not belonging to the set of NEvAr images, $O_2 \leq 0.3$. A suitable fitness function for this goal is as follows:[8]

$$\text{fitness} = \left[\frac{1 + (O_1 - O_2)}{2}\right]^2 \times 10 \tag{18.3}$$

Accordingly, an image that results in an ANN output of $(0.3, 0.7)$, thus being marginally considered as a NEvAr image, has a fitness of 0.9. Conversely, an image that is in the threshold of being classified as a *painting*, i.e., that results in an ANN output of $(0.7, 0.3)$, has a fitness of 4.9. In other words, a fitness value in the $[0, 0.9[$ interval corresponds to images that are classified as NEvAr's; in the $[0.9, 4.9]$ interval to images that the ANN is unable to classify; and in the $]4.9, 10]$ interval to images classified as paintings by the ANN.

[8] The multiplication by 10 is a scaling operation performed to allow an easier integration with the user interface.

## 18.6 Experimental Results

In this section we present some of the experimental results attained with our approach.

As previously stated, one of the key issues of our approach is the iterative methodology used, i.e., once an evolutionary run ends, the images generated by NEvAr are added to the set of internal images and the ANN is retrained.

For the purpose of the current paper, we are mainly interested in analyzing the differences, in terms of produced imagery, that occur from iteration to iteration, which implies presenting the images obtained in each of them. This necessity, coupled with space constraints, makes it infeasible to present results from several independent experiments. Therefore, we focus on a single experiment, paying particular attention to the first and last iterations.

### 18.6.1 Experimental Setup

The settings of the GP engine are presented in Table 18.2. The settings are similar to those used by default when NEvAr is run in interactive mode [17, 9]. The images are rendered in full color, at a resolution of $128 \times 128$ pixels. External images of higher dimension are resized to $128 \times 128$ for feature extraction.

To calculate the complexity estimates used in the FE, the images are compressed at three different levels of detail by setting the maximum error per pixel to 8, 14 and 20. These values were determined empirically in previous tests.

One of the sub-goals of our experiments is the assessment of the relevance of some aspects of the FE, namely the relevance of the features concerning color information and the importance of the features gathered from the partitions of the images.

**Table 18.2.** Parameters of the GP engine. See Machado and Cardoso [17, 9] for a detailed description

| Parameter | Setting |
|---|---|
| Population size | 50 |
| Number of generations | 50 |
| Crossover probability | 0.8 (per individual) |
| Mutation probability | 0.05 (per node) |
| Mutation operators | sub-tree swap, sub-tree replacement, node insertion, node deletion, node mutation |
| Initialization method | ramped half-and-half |
| Initial maximum depth | 5 |
| Mutation max tree depth | 3 |
| Function set | $+$, $-$, $\times$ , $/$, min, max, abs, neg, warp, sign, sqrt, pow, mdist, sin, cos, if |
| Terminal set | $X$, $Y$, scalar and vector random constants |

To serve this goal, we employ different combinations of features, which result in ANNs with different input layers. The first group of ANNs includes features pertaining to the three color channels of the image. The second group of ANNs uses features related to the Saturation and Value channel. The information contained in the Hue channel is only relevant when the saturation and value are taken into consideration. In addition, Hue is circular by nature, which creates difficulties in the measurement of several features. For these reasons, the information extracted from the Hue channel is likely to be less reliable. The third group only considers the Value channel, thus analyzing the grayscale version of the image.

We gather the same set of features for the entire image and for its partitions, which results in some degree of redundancy. It is, therefore, important to assess the relevance of the information gathered from the partitions. To do so, we further divide the three groups above into ANNs which use features relative to the partitions and those which do not.

Overall, we employ six ANN architectures. The particularities of each are summarized in Table 18.3. In Table 18.4, we present other relevant parameters relative to the ANNs.

For each considered architecture we perform 30 independent repetitions of the training stage, in order to get statistically significant results. For each of these repetitions we randomly create training, test and validation sets with, respectively, 70%, 10%, and 20% of the patterns. The same randomly created sets are used for the different architectures. The training of the ANNs is halted when one of the following criteria are met: 1000 training cycles, or an RMSE in both the training and test sets lower than 0.005. These parameters were empirically established in previous experiments.

**Table 18.3.** Features considered in each ANN

| Artificial Neural Network | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Color channels | 3 | 3 | 2 | 2 | 1 | 1 |
| Partitions | yes | no | yes | no | yes | no |
| Number of features | 246 | 41 | 186 | 31 | 108 | 18 |

The number of internal images increases from iteration to iteration, while the number of external images remains constant. This creates a disproportion between the two classes, which could jeopardize training. To avoid this problem we use a one-to-one class distribution scheme [44]. During training, the ANN is exposed to the same number of patterns from each class, by including repetitions of patterns from the class of lower cardinality in the training set (this does not affect the test or validation sets).

We also wish to detect at what stage the different ANNs become unable to fully distinguish between the two classes. For this purpose we carry out an additional test in which all the patterns are included in the training set. In

**Table 18.4.** Parameters relative to the ANNs and their training

| Parameter | Setting |
|---|---|
| Number of architectures | 6 |
| Activation function | logistic |
| Output function | identity |
| Initialization of weights | random, $[-0.1, 0.1]$ interval |
| Learning function | backpropagation |
| Learning rate | 0.3 |
| Momentum | 0 |
| Update function | topological order |
| Maximum n. of training cycles | 1000 |
| RMSE limit (train and test sets) | 0.005 |
| Shuffle weights | yes |
| Class distribution (training set) | one-to-one |

this case, the ANN training is halted after 5000 training cycles or when an RMSE of zero is reached. From here on we will call this specific test "*Entire Corpus*".

The generation of the 2500 images of each iteration takes, approximately, 4.5 hours. The creation and rendering of one population takes an average of 15 seconds, 0.3 seconds per image. Feature extraction is a time-consuming process, taking, on average, 6 seconds per image. The ANN training is even more time-consuming, depending on the number of patterns, the 30 training runs for a single architecture can take up to 12 hours. All time estimates where performed using a Pentium Mobile at 1.8 GHz.

### Initial Sets

The initial sets of external and internal images play an important role in the performance of our system. We use an external set containing 3322 paintings of the following artists: Cézanne, de Chirico, Dalí, Gaugin, Kandinsky, Klee, Klimt, Matisse, Miró, Modigliani, Monet, Picasso, Renoir, van Gogh. The images where gathered from different online sources. The rationale was to collect a wide and varied set of artworks.

The set of internal images is created using NEvAr to generate 7 initial random populations of size 500. In order to obtain a more representative set of internal images, these generations were created using different upper bounds for the tree depths. Additionally, in order to avoid a bias towards simplicity in the internal set of images, a primitive that generates random noise in two of the populations (2 and 7) is used.

Although the images were created randomly, some of the phenotypes may appear more than once. The same can happen throughout iterations.

We performed tests in which these repetitions were removed, arriving to the conclusion that it was not advantageous to do so. Considering our goals,

images that occur frequently in one iteration should be avoided at all costs in subsequent ones. Having repeated instances of "popular" images ensures that these have more influence on the training of the ANN than others. Therefore, classification errors are unlikely to happen in images that become popular, avoiding a future convergence of the EC to these images.

The existence of repetitions implies a partial overlap between the training, test and validation sets. Consequently, there is a bias in the ANN results. To overcome it, we performed a set of independent validation experiments, presented in Sect. 18.7.

In what concerns the external set, repetitions were avoided. Nevertheless, it is relatively common for an artist to paint several versions of the same motif. In these cases, and in order to avoid the subjectivity of deciding what was sufficiently different, we decided to include the different variations.

### 18.6.2 First Iteration

In this section we present the experimental results attained in the first iteration. An in-depth analysis of the results concerning the training of the ANN and of the relative importance of the different groups of features, of this and further iterations, will be published in the future.

In Table 18.5 we provide an overview of the results attained in training, for the different ANN architectures, presenting the average number of training cycles, the average RMSE and its STD for the training, test and validation sets. The results are calculated from the 30 independent training repetitions made for each of the 6 architectures.

**Table 18.5.** Overview of the ANNs' training results in iteration 1

| | | | Training | | Test | | Validation | |
|---|---|---|---|---|---|---|---|---|
| Network | Features | Cycles | avg | std | avg | std | avg | std |
| 1 | 246 | 733.3 | .0001 | .0001 | .0065 | .0027 | .0069 | .0017 |
| 2 | 41 | 863.3 | .0010 | .0016 | .0079 | .0037 | .0086 | .0025 |
| 3 | 186 | 940.0 | .0001 | .0007 | .0108 | .0035 | .0102 | .0022 |
| 4 | 31 | 926.6 | .0021 | .0011 | .0088 | .0043 | .0098 | .0025 |
| 5 | 108 | 1000.0 | .0005 | .0005 | .0212 | .0060 | .0232 | .0041 |
| 6 | 18 | 1000.0 | .0125 | .0028 | .0214 | .0057 | .0225 | .0049 |
| Average | | 910.5 | .0027 | .0011 | .0128 | .0043 | .0135 | .0029 |

We are also interested in the number of images that are incorrectly identified. In Table 18.6 we provide an overview of these results considering a "winner-takes-all" strategy, i.e., the output neuron with the highest activation value determines the category in which the corresponding image is classified. We present the average number of misclassified images and its STD for the test, training and validation sets. The last two columns of the Table concern

**Table 18.6.** Average number of misclassified patters in iteration 1. The training, test and validation set have, respectively, 4775, 682 and 1364 patterns

| | Training | | | Test | | | Validation | | | Entire Corpus | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Network | avg | std | % | avg | std | % | avg | std | % | Ext. | Int. |
| 1 | .1 | .2 | .001 | 2.9 | 1.9 | .430 | 6.5 | 2.4 | .475 | - | - |
| 2 | 1.9 | 3.6 | .040 | 3.7 | 2.2 | .545 | 8.7 | 2.9 | .635 | - | - |
| 3 | .4 | 1.9 | .008 | 6.2 | 2.5 | .911 | 11.3 | 2.9 | .828 | - | - |
| 4 | 2.6 | 2.6 | .054 | 3.9 | 2.7 | .574 | 9.2 | 3.6 | .672 | - | - |
| 5 | 1.0 | 1.5 | .020 | 10.2 | 3.4 | 1.491 | 22.6 | 4.5 | 1.660 | - | - |
| 6 | 36.1 | 7.0 | .756 | 10.8 | 3.4 | 1.581 | 21.9 | 5.0 | 1.604 | 6 | 5 |
| Average | 7.0 | 2.8 | .147 | 6.3 | 2.7 | .922 | 13.4 | 3.6 | .976 | 1.00 | .83 |

the *Entire Corpus* test. They depict the number of external images classified as internal (Ext.) and the number of internal images classified as external (Int.).

A brief perusal of Tables 18.5 and 18.6 shows that most of the ANNs successfully discriminate between internal and external images. The average RMSE in test and validation is lower than 0.0232 for all ANNs. Additionally, the average percentage of correctly classified images always exceeds 98.43%.

There are no statistically significant differences between the RMSEs attained in the test and validation sets,[9] which tends to indicate that the ANNs are generalizing properly.

The comparison between the RMSEs attained by the ANNs that use information gathered from the three color channels and those that only use information from the Saturation and Value reveals statistically significant differences for test, train and validation sets. Comparing the ANNs that resort to the Saturation and Value channels with those that only use Value also shows statistically significant differences for the three sets.

The analysis of the results of the ANNs that use features gathered from the images' partitions (ANNs 1, 3 and 5) with those that do not points out the following: although significant differences exist when we consider the RMSEs achieved in training, there are not statistically significant differences in the results attained in the test and validation sets. This indicates that, in the considered experimental conditions, the information gathered from the images' partitions is not relevant for generalization purposes.

To confirm the experimental results described above, we performed several control experiments. In these tests we randomly assigned a category to all the patterns used in the different sets. The experimental results indicate misclassification percentages of, roughly, 50% for the test and validation sets, regardless of the architecture, thus confirming that the previously described results do not arise from some implicit bias in the methodology.

[9] The statistical significance of these results, and subsequent ones, was determined through the Wilcoxon–Mann–Whitney test. Confidence level of 0.99.

Considering the above experimental findings – which suggest that the information associated with the different color channels is relevant for generalization purposes, while the information gathered from the images' partitions is not – we chose the second architecture to guide the evolutionary algorithm. This architecture employs a relatively low number of features (41), which was also relevant for our choice. To assign fitness, we select the ANN with the lowest average RMSE across training, test and validation, among the 30 trained ANNs corresponding to the second architecture.

**Analysis of the EC Results**

Figure 18.4 depicts the best individual of each population and the corresponding fitness values. In order to provide a better overview of the full range of imagery produced in the run we selected some examples, which are presented in Fig. 18.5. These images have a fitness higher than 4.9, which means that the ANN is classifying them as external (see Sect. 18.5.2).

The comparison between the images produced and the ones from the internal set reveals that, while the images of the internal set tend to have very low or very high complexity, the fittest images found during the run tend to have intermediate levels of complexity.

This result was expected. Randomly generated images of small depth tend to be simple. That is why we used a noise generation primitive in two of the seven random populations of the initial internal set. However, the inclusion of the noise primitive resulted frequently in mostly random images, which by definition have high complexity. Since images of intermediate complexity are frequent in the external set and rare in the internal one, it is only natural that the ANN has chosen this path to discriminate between both.

The EC algorithm found images that the ANN classifies as external without difficulties. From the fourth population onwards the fitness of the best individual is above 4.9, from the tenth population onwards the best individual has a fitness above 9.

### 18.6.3 Intermediate Iterations

Once an iteration is over, the 2500 images produced by the GP engine are added to the internal set and the ANNs retrained. As previously stated, we use a class distribution of one-to-one. The number of external images does not grow, but the set of internal images keeps expanding. Therefore, in each training cycle the ANNs are only exposed once to each internal pattern; the number of times they are exposed to each external pattern steadily increases from iteration to iteration.

To ensure that the initial conditions are the same for all iterations, we use a fixed random seed. Therefore, the initial population, albeit randomly generated, is the same for all iterations. This ensures that the variations in

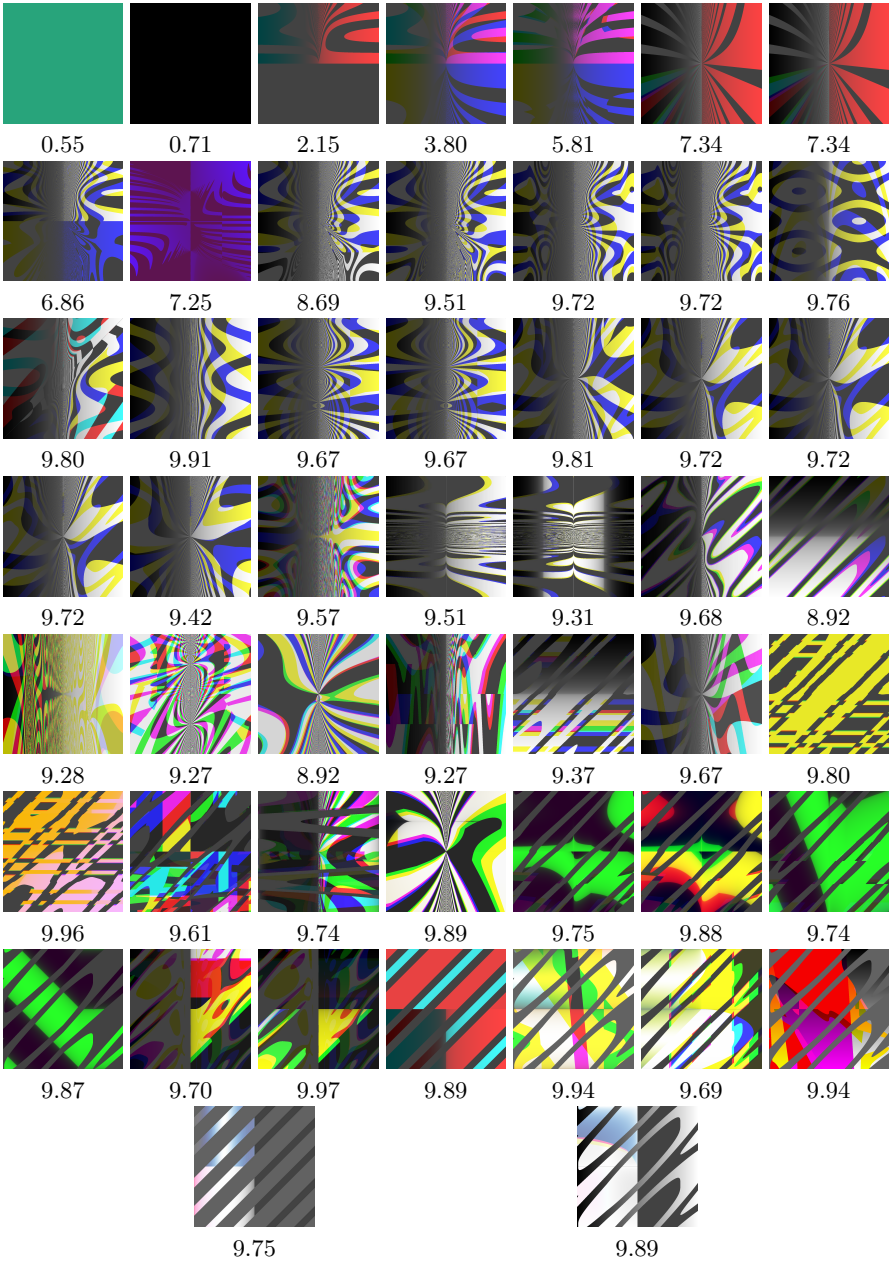
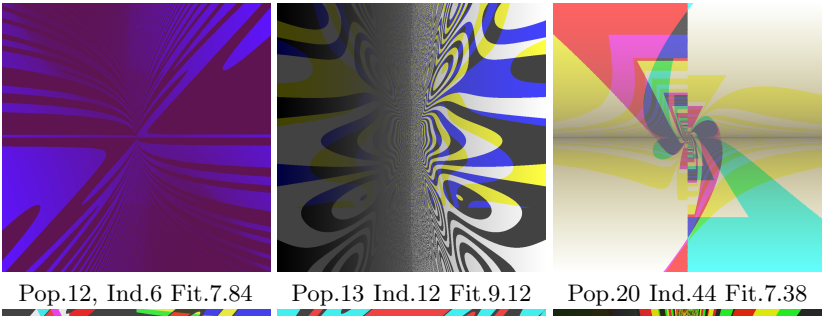



**Fig. 18.4.** Fittest individual from each population of the first iteration. The image in the upper-left corner corresponds to population 0; remaining images in standard reading order. The numbers indicate the fitness values



Pop.12, Ind.6 Fit.7.84 Pop.13 Ind.12 Fit.9.12 Pop.20 Ind.44 Fit.7.38

Pop.39 Ind.49 Fit.9.64 Pop.41 Ind.24 Fit.9.6 Pop.49 Ind.42 Fit.9.65

**Fig. 18.5.** Selected images from the first iteration (Pop=population, Ind=individual, Fit=fitness attributed by the ANN)

the type of imagery produced by the GP engine do not result from different initial conditions.

In Tables 18.7 and 18.8 we provide a synthesis of the results attained in the training of the ANNs with the second architecture for iterations 1 to 12.

In the first iteration, most training runs end because the test RMSE becomes lower than the specified threshold. In the second iteration, most training runs end because the maximum number of cycles is met. Therefore, in the first iteration, the training RMSE is higher and test and validation lower than in the second one. This may be explained by a higher correlation among the test, training and validation sets in the first iteration – where all internal images are randomly generated – than in the second one – where the internal set contains 3500 random images and 2500 evolved ones.

The increase of test and validation RMSE in the second iteration is temporary. The addition of new images resulting from evolutionary runs leads to better generalization since the set becomes more representative.

As can be observed, from the third iteration onwards the RMSE and the percentage of misclassified images remain relatively stable for the training, test and validation sets. The main exception to this trend is the sudden, and statistically significant, increase in the training RMSE and the misclassification percentage from iteration 11 to 12. Due to this difference in performance,

**Table 18.7.** Overview of the training results in iterations 1 to 12 of the ANNs with the second architecture

| | | Training | | Test | | Validation | |
|---|---|---|---|---|---|---|---|
| Iteration | Cycles | avg | std | avg | std | avg | std |
| 1 | 863.3 | .0010 | .0016 | .0079 | .0037 | .0086 | .0025 |
| 2 | 953.3 | .0003 | .0009 | .0113 | .0039 | .0111 | .0024 |
| 3 | 913.3 | .0003 | .0007 | .0082 | .0025 | .0082 | .0018 |
| 4 | 943.3 | .0003 | .0009 | .0090 | .0028 | .0095 | .0020 |
| 5 | 823.3 | .0005 | .0009 | .0074 | .0029 | .0083 | .0021 |
| 6 | 920.0 | .0003 | .0006 | .0081 | .0024 | .0079 | .0018 |
| 7 | 930.0 | .0005 | .0005 | .0088 | .0027 | .0085 | .0022 |
| 8 | 886.7 | .0004 | .0005 | .0079 | .0026 | .0079 | .0014 |
| 9 | 880.0 | .0007 | .0008 | .0076 | .0024 | .0077 | .0017 |
| 10 | 940.0 | .0006 | .0009 | .0083 | .0025 | .0084 | .0011 |
| 11 | 893.3 | .0008 | .0006 | .0079 | .0026 | .0079 | .0019 |
| 12 | 916.7 | .0016 | .0017 | .0085 | .0026 | .0085 | .0025 |

**Table 18.8.** Average number and percentage of misclassified patterns attained by the ANNs with the second architecture in iterations 1 to 12. The column "Patterns" shows the total number of patterns in each iteration

| | | Training | | | Test | | | Validation | | | Entire Corpus | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Iteration | Patterns | avg | std | % | avg | std | % | avg | std | % | Ext. | Int. |
| 1 | 6822 | 1.9 | 3.6 | .040 | 3.7 | 2.2 | .545 | 8.7 | 2.9 | .635 | - | - |
| 2 | 9322 | .9 | 3.4 | .013 | 7.4 | 2.8 | .798 | 14.2 | 4.2 | .759 | - | - |
| 3 | 11822 | .9 | 2.4 | .011 | 6.5 | 2.7 | .552 | 13.3 | 3.8 | .564 | - | - |
| 4 | 14322 | .8 | 2.8 | .008 | 8.9 | 3.4 | .624 | 19.0 | 4.6 | .665 | - | - |
| 5 | 16822 | 2.1 | 4.3 | .018 | 8.8 | 4.0 | .524 | 19.7 | 4.8 | .586 | - | - |
| 6 | 19322 | .8 | 1.4 | .006 | 11.1 | 3.7 | .572 | 20.9 | 5.5 | .542 | - | 1 |
| 7 | 21822 | 2.5 | 3.6 | .017 | 12.8 | 4.5 | .586 | 25.6 | 6.5 | .586 | - | 1 |
| 8 | 24322 | 1.6 | 3.0 | .009 | 13.2 | 4.5 | .545 | 26.9 | 4.8 | .552 | - | - |
| 9 | 26822 | 2.6 | 4.2 | .014 | 14.4 | 5.1 | .537 | 27.4 | 6.3 | .511 | - | 1 |
| 10 | 29322 | 4.2 | 5.8 | .021 | 16.8 | 5.4 | .574 | 34.5 | 5.3 | .589 | - | 3 |
| 11 | 31822 | 5.7 | 4.3 | .026 | 17.1 | 6.2 | .537 | 35.5 | 9.9 | .557 | - | 1 |
| 12 | 34322 | 16.6 | 25.1 | .069 | 19.8 | 5.7 | .578 | 39.6 | 11.8 | .577 | - | 6 |

we decided to stop our iterative approach in order to perform a more detailed analysis.

In the next section we present the results attained by the EC algorithm in iterations 2 to 11. The ANNs of iteration 12 are analyzed in Sect. 18.6.4.

### Analysis of the EC Results

In Fig. 18.6 we present the fittest individual from populations 0, 10, 20, 30, 40 and 50, for iterations 2 to 10. In Fig. 18.7 we present selected examples from these iterations.

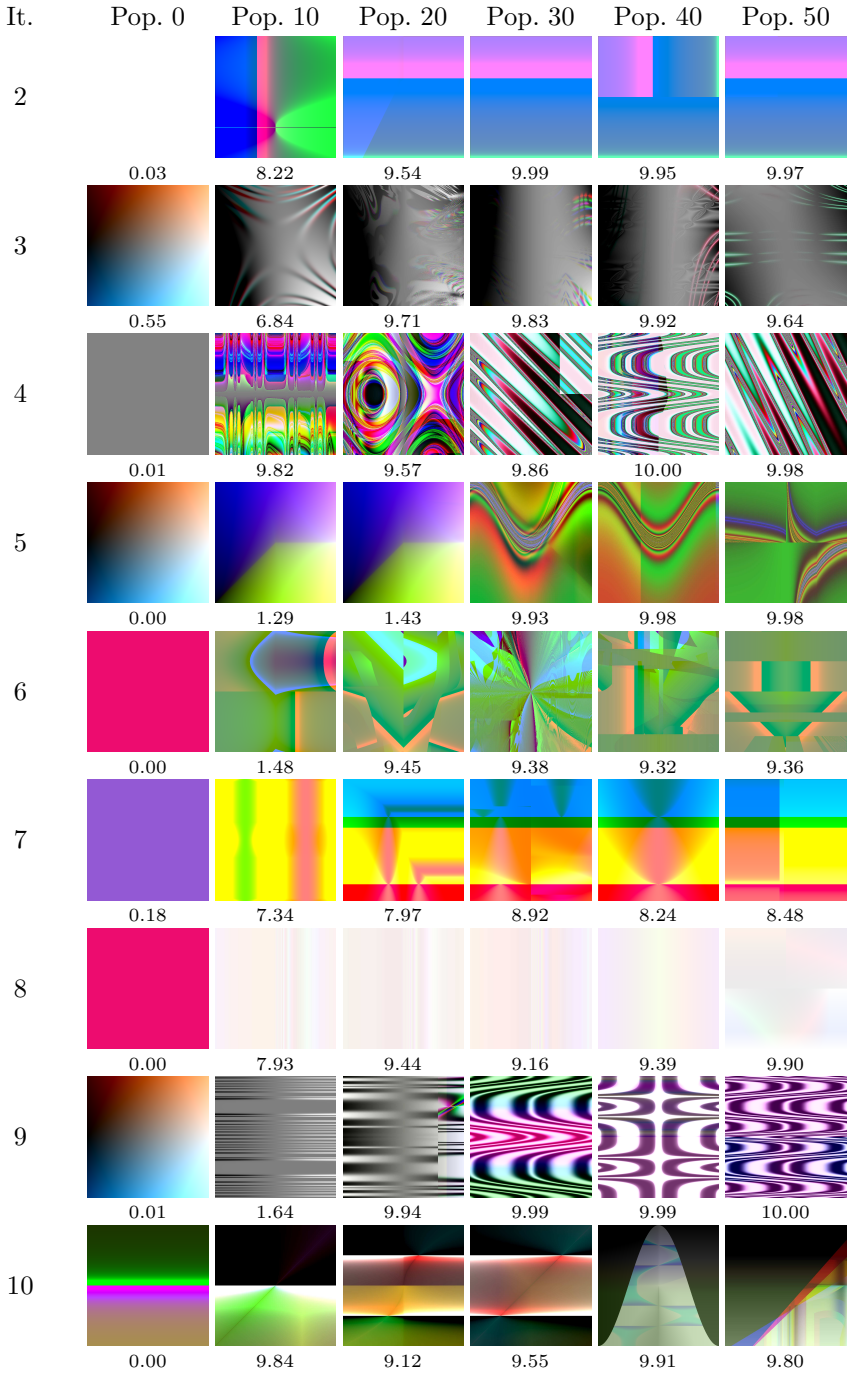



**Fig. 18.6.** Fittest images from populations 0, 10, 20, 30, 40, 50 of Iterations 2–10

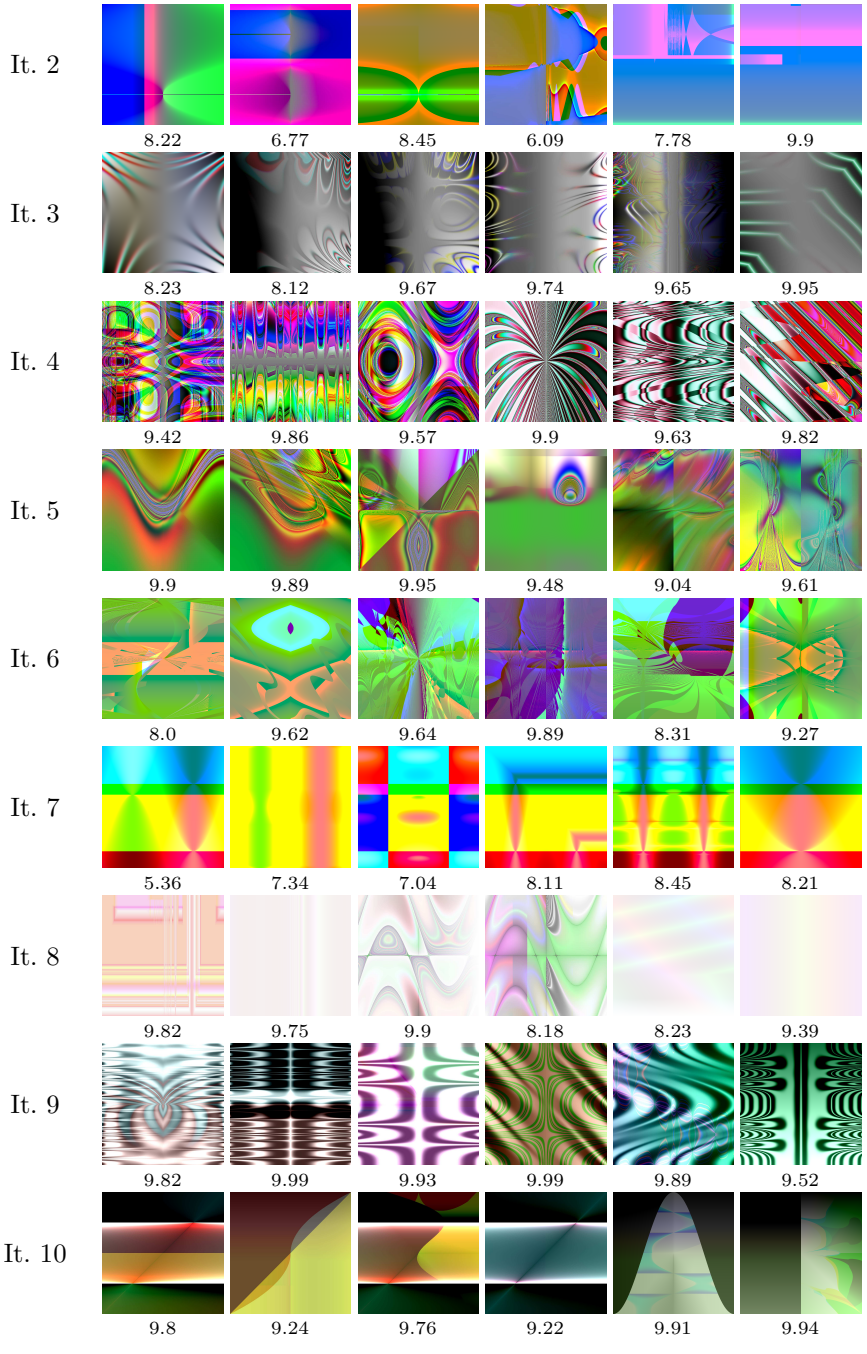



**Fig. 18.7.** Selected images from Iterations 2 to 10

Since the addition of the images of iteration 11 to the internal set led to the increase of the training RMSE and misclassification percentages on the twelfth iteration, this iteration is presented with greater detail. In Fig. 18.8 we present the best individual of each population and the corresponding fitness value, while in Fig. 18.9 we present selected images from the eleventh iteration.

By nature, the analysis of visual results entails some degree of subjectivity. We believe, however, that it is safe to say that there are significant differences in the type of imagery produced from iteration to iteration. For instance, in the eleventh iteration the EC algorithm converged to a style, characterized by the use of specific hues and by the low saturation values, that diverges from those explored in previous iterations. Considering that one of our main goals was to attain stylistic variation in an autonomous EC framework, the unlikeness of iterations is a key result.

We are not however, just interested in change. The inclusion of a fixed aesthetic reference frame is also a key aspect of our approach. It is therefore important that the generated imagery relates to human aesthetics.

One would expect to observe imagery that gets increasingly closer to the set of external images as the number of iterations increases. Although this may be the case in the long run, it is not reasonable to expect this approach to the aesthetic reference to be steady. To understand why this is the case, it is important to ponder about the reasons that may lead an ANN to classify an image as an external one.

The ANNs are trained to distinguish among two sets. However, conceptually, three sets can be considered, images that: (i) appear to be paintings; (ii) appear to be EC-generated; (iii) do not resemble paintings or EC created ones. Even if an ANN that can fully discriminate between the first two sets exists, occasionally this ANN will classify images of the third set as paintings.

When this situation occurs, the EC algorithm will, most likely, explore that path, leading to the generation of images that do not resemble those belonging to any of the sets. This does not constitute a flaw, in the next iteration these images will be added to the set of internal images and the EC algorithm will no longer be able to explore that path. It does mean, however, that the "approach" to the aesthetic reference frame is not steady.

A simplified example may help in the clarification of the previous statements. Let us consider that only two features exist $(x, y)$, and that external images are closely scattered around the point $(4, 3)$ while the considered internal images are scattered around $(6, 4)$. A possible classifier for this system is: $x \leq 5.5 \rightarrow external$, $x > 5.5 \rightarrow internal$. Using this classifier can lead to the evolution of images with $x \cong 4$, the EC algorithm can also overcompensate, e.g., generating images with $x \cong 1$. More importantly, $y$ is a free variable, anything can happen in that dimension, e.g., an image with $(0, 10)$ would be classified as external. As such, considering a Euclidian space, the generated images are not necessarily closer to the set of external images than those belonging to the internal set.

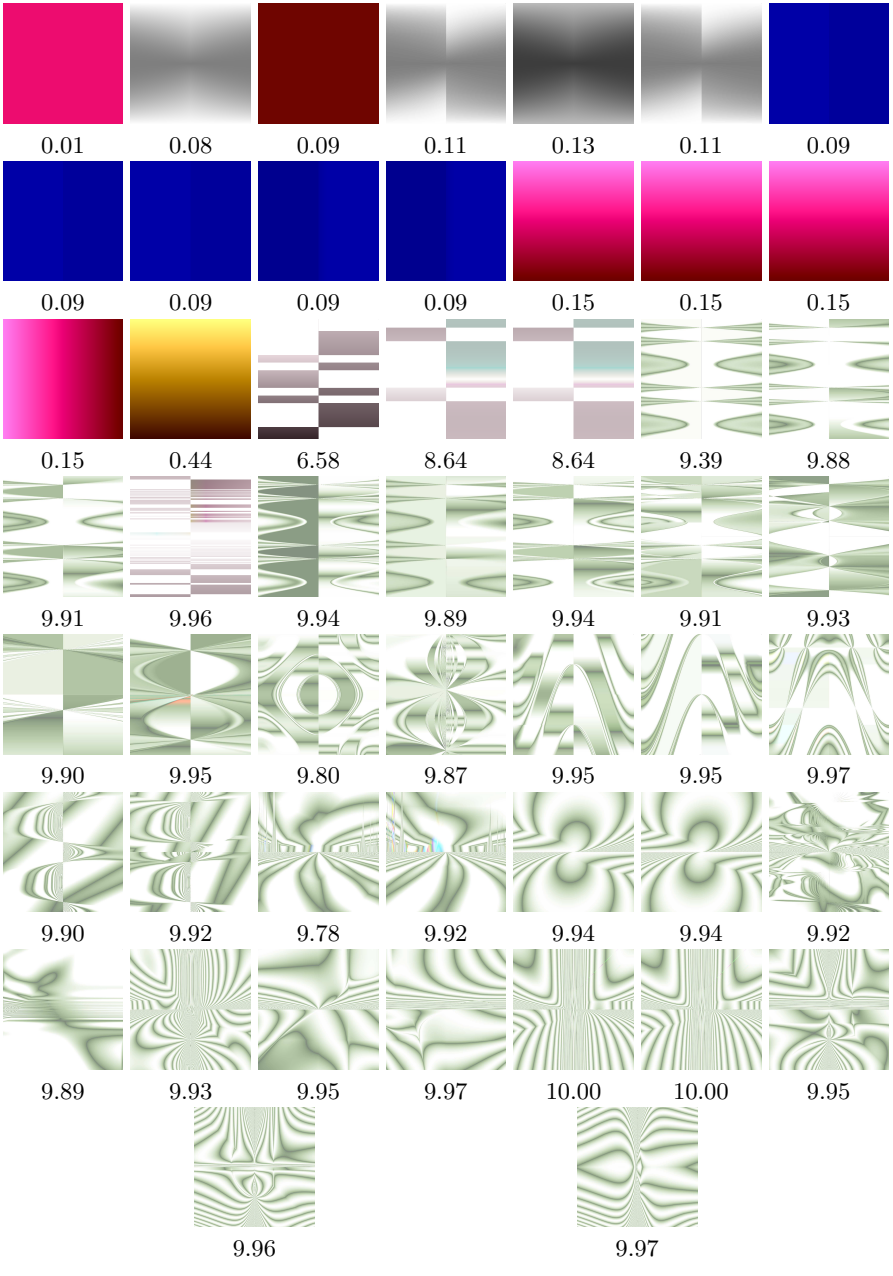



**Fig. 18.8.** Fittest individual from each population of the eleventh iteration

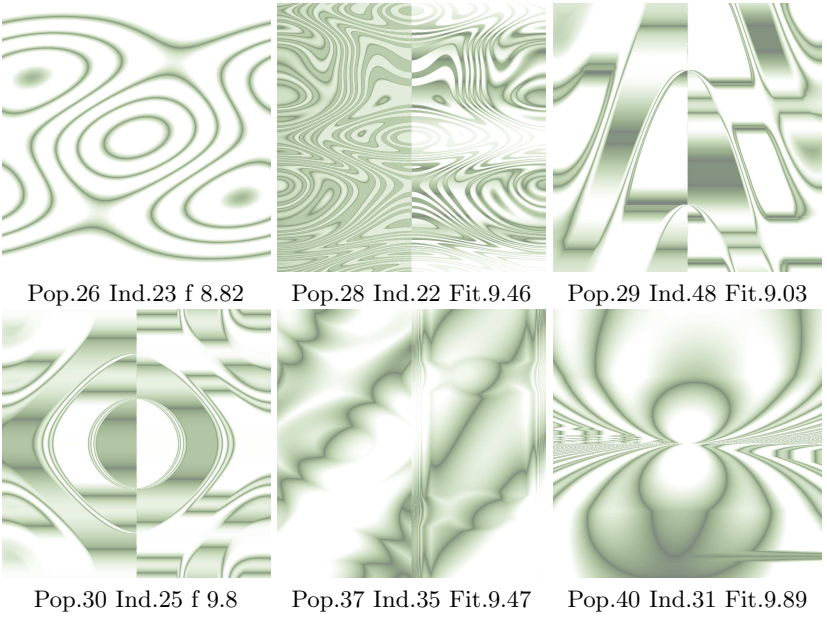



**Fig. 18.9.** Selected images from the eleventh iteration

The successive addition of new images leads to the refinement of the classifier. For instance, assuming that images with $x \cong 1$ were generated, in the next iteration we could have a classifier such as "$2.5 \leq x \leq 5.5 \rightarrow external$". Alternatively, we could have a classifier that took into account feature $y$. What is important is that the space of images classified as internal is expanding.

Thus, from a theoretical standpoint – assuming that the EC engine and the AAC are adequate and always able to cope – the combination of a fixed aesthetic reference frame with the ANN training, and the iterative expansion of the internal set leads necessarily to change (since the EC algorithm is forced to explore new paths) and to the erratic, but certain convergence to the aesthetic reference frame.

### 18.6.4 Iteration 12 – Training Stage

In Tables 18.9 and 18.10 we provide a synthesis of the experimental results attained in the training stage of iteration 12 for the 6 architectures considered.

The results presented in these tables show that most ANNs were able to achieve high discrimination rates across training, test and validation sets. All ANN architectures attain average classification percentages higher than 97.62 in training, test and validation.

The comparison of the training results attained in the first and twelfth iteration is interesting. There are statistically significant differences in training

**Table 18.9.** Overview of the ANNs' training results in iteration 12. The entries in bold indicate statistically significant differences between the results attained in this iteration and the corresponding ones of the first iteration

| | | | Training | | Test | | Validation | |
|---|---|---|---|---|---|---|---|---|
| Network | Features | Cycles | avg | std | avg | std | avg | std |
| 1 | 246 | 580.0 | .0001 | .0001 | **.0052** | .0017 | **.0058** | .0015 |
| 2 | 41 | 916.7 | **.0016** | .0017 | .0085 | .0026 | .0085 | .0025 |
| 3 | 186 | 866.7 | 0 | .0001 | **.0071** | .0018 | **.0077** | .0020 |
| 4 | 31 | 1000.0 | **.0035** | .0014 | **.0117** | .0028 | **.0107** | .0022 |
| 5 | 108 | 1000.0 | .0008 | .0010 | .0222 | .0038 | .0224 | .0048 |
| 6 | 18 | 1000.0 | **.0268** | .0027 | **.0330** | .0083 | **.0332** | .0071 |
| Average | | 893.9 | .0055 | .0011 | .0146 | .0035 | .0147 | .0033 |

**Table 18.10.** Average number and percentage of misclassified patterns in twelfth iteration. The training, test and validation set have, respectively, 24025, 3432 and 6864 patterns

| | Training | | | Test | | | Validation | | | Entire Corpus | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Network | avg | std | % | avg | std | % | avg | std | % | Ext. | Int. |
| 1 | 0 | 0 | 0 | 12.9 | 5.2 | .376 | 28.4 | 7.9 | .414 | - | - |
| 2 | 16.6 | 25.1 | .069 | 19.8 | 5.7 | .578 | 39.6 | 11.8 | .577 | - | 6 |
| 3 | .08 | .43 | .000 | 17.1 | 4.2 | .499 | 36.2 | 9.4 | .528 | - | - |
| 4 | 44.2 | 23.7 | .184 | 28.7 | 7.2 | .836 | 50.8 | 11.0 | .740 | 2 | 55 |
| 5 | 9.8 | 11.2 | .041 | 51.0 | 9.1 | 1.485 | 101.3 | 21.9 | 1.476 | - | 8 |
| 6 | 416.5 | 56.0 | 1.734 | 81.3 | 21.2 | 2.370 | 163.3 | 37.9 | 2.380 | 23 | 461 |
| Average | 81.2 | 19.4 | .338 | 35.1 | 8.7 | 1.024 | 70.0 | 16.7 | 1.019 | 4.17 | 88.33 |

RMSE for all architectures (2, 4, and 6) that do not use information gathered from the images' partitions. Taking into consideration that the ANNs used to guide evolution have the second architecture, these results reveal that the GP engine is able to find images which are found difficult to classify by the ANN guiding evolution. The fourth and sixth architectures use a subset of the features used in the second, which explains the decrease of performance observed.

For the architectures using partition information architectures, the differences in training RMSE are not significant. Most features considered in these architectures are not present in the second one. As such, there was not a specific evolutionary pressure on these features, and consequently their performance in training was relatively unaffected.

The increase of cardinality of the training sets allowed the ANNs with the first and third architectures (the ones that have a higher number of features) to achieve better generalization. For the ANNs with the second architecture, in spite of the increased difficulty in training, the performance in the test and the validation sets is similar to that attained in the first iteration. The same does not apply to the ANNs with the fourth and sixth architecture, whose

performance is significantly worse than that attained in the first iteration. The performance of the ANNs with the fifth architecture appears to be unaffected. This is probably due to the low degree of overlap between the features considered in this architecture and in the second.

Nevertheless, it is important to remember that the ANNs of the first and twelfth iterations are not being tested on the same sets. The ANNs of the first iteration would attain poor results if tested on the validation sets of the twelfth.

## 18.7 Independent Validation Experiments

The existence of repeated patterns induces a bias in the experimental results presented in the previous section. Therefore, we conducted a series of control experiments in order to understand better the changes induced by the iterative refinement of the internal set of images. We are mainly interested in comparing the performances of the ANNs used to guide the evolution in iteration 1 and 11.

To achieve this goal we employ three sets of images. The first comprises 2000 images, made by artists that were not on the training set, from a collection of painting masterpieces [45]. The second consists of images retrieved with Google image search using the keyword "painting", containing the first 947 hits that do indeed correspond to paintings. In the context of an "Artificial Art" event, several students used NEvAr in interactive mode to evolve a large number of images, submitting their favorite ones to the online gallery associated with the event.[10] The third set comprises the 278 images submitted (a sample of these set can be found in Fig. 18.3).

In Table 18.11 we provide a synthesis of the results attained in these experiments, presenting the percentage of images classified as external, following a winner-takes-all strategy, attained by the ANNs used in the first and eleventh iteration.

These results suggest that the fixed aesthetic reference frame provided by the external set is achieving its task, allowing both ANNs to discriminate between images that may be classified as paintings and images that were created with NEvAr.

**Table 18.11.** Percentage of images classified as external by the ANNs used to guide evolution in iterations 1 and 11, and difference among them

| Set | Iteration 1 | Iteration 11 | Difference |
|---|---|---|---|
| Painting masterpieces | 99.68% | 96.88% | −2.80% |
| Images retrieved with Google | 96.41% | 90.92% | −5.49% |
| User-guided evolution | 17.99% | 10.07% | −7.91% |

[10] `http://sion.tic.udc.es/jornadas/`

When these results are compared with those attained in the validation sets of iterations 1 and 11, where these ANNs correctly classify roughly 99.5% of the images, a difference in performance can be observed. Several factors may contribute to it:

1. The sets used in the interactions have artworks by the same painters and images from the same evolutionary runs. Therefore, the correlation between the training and validation sets of each iteration is stronger than the correlation between the training set and the images used in these experiments.
2. The images of the external set used to train the ANNs are artworks of renowned artists. The images retrieved from Google originate from a wide variety of sources (e.g., amateur works, child art, etc.), which also explains why the results attained with these images are worse than those attained with the collection of masterpieces.
3. The images resulting from interactive evolution are those selected by the users, i.e., images that were considered remarkable by them. As such, the percentage of atypical images in this set is likely to be higher than the percentage of atypical images in the entire evolutionary run.

Comparing the results of the ANN of the first iteration with those of the eleventh reveals a decrease of the percentage of the images classified as external (2.80% and 5.49%, respectively). On the other hand, the increase in performance in the set resulting from user-guided evolution is 7.91%, a value that surpasses the differences observed in the other sets.

When combined, these results indicate that the ANN from iteration 11 is able to refine its capacity to recognize internal imagery without seriously hindering its performance in a set of external images that was not used in training, i.e., the ANN appears to be able to keep the provided aesthetic reference frame and to generalize properly, which confirms the experimental findings of the previous sections.

### 18.7.1 Borderline Images

We are also interested in determining which images are the most difficult to classify. To achieve this goal we conducted two different experiments. In both cases, the training set is composed of 100% of the images of the external set, of the initial random internal set, and of the images generated in generations 1 to 11 (inclusive).

In the first test, we do not employ a class distribution. Due to the different cardinalities of the sets, the ANN is exposed to, approximately, 10 internal images for each external one. In the second test we employed a class distribution of 10 to 1, which means that the ANN is exposed to, roughly, 100 external images for each internal image. The rationale is the following: in the first test the ANN is exposed to more internal images, as such it will tend to

misclassify external ones. Conversely, in the second test, the ANN will tend to misclassify internal images.

We employed the second architecture. The parameters used in training are similar to those used in the "Entire Corpus" tests, the exception being the use of a lower learning rate (0.1) and a higher number of training cycles (10000). In this case, for each experiment, 30 repetitions were performed.

In the first experiment, on average, 2.60 internal images were classified as external, and 4.56 external images were classified as internal. In the second experiment, an average of 2.93 internal images are misclassified, while no external images are misclassified.

The external images that are misclassified most frequently are: Salvador Dalí, "Fried Egg on the Plate Without the Plate" (1932), "Battle in the Clouds" (1979); Pablo Picasso, "Paul as a Pierrot" (1925); Amedeo Modigliani "Nude — Anna Akhmatova" (1911) and "Stone Head" (1910); Henri Matisse, "La Musique" (1910).[11]

These results were, to some extent, unexpected. One could assume that the ANN would tend to misclassify external images that resemble those created by the EC algorithm. Instead, the misclassification errors occur in images that are atypical in the scope of the considered external set.

Dalí's artwork, "Fried Egg on the Plate Without the Plate" and Picasso's "Paul as a Pierrot" are, in the employed version, images of little detail and texture. "Battle in the Clouds" is a stereoscopic work, which is odd in the present context. The artwork "Nude — Anna Akhmatova", by Modigliani is a pencil on paper drawing, while "Stone Head" is a sculpture. Both stand out for obvious reasons in a set composed mainly of paintings. Matisse's work "La Musique" can also be considered atypical in regard to the remaining images that compose the set. In addition, the image was saved at a low resolution, which may cause perturbations of the features values.

The internal images that are misclassified most frequently are presented in Fig. 18.10. Although one can argue that some of these images are more similar to paintings than the images typically created with NEvAr, the comparison between these images and those generated throughout the runs shows that they are, above all, uncommon.

It is interesting to notice that three of these images were previously selected to illustrate the types of imagery produced during the iterative runs (see Figs. 18.5 and 18.7). By browsing the book's DVD the reader can verify that these images stand out from the images of their iteration, capturing the attention of the viewer, which alone grants them more chances of being selected.

Overall, the results presented in this section confirm that the ANNs are able to generalize properly and to identify correctly images that are stylistically consistent with the set of internal or external images.

[11] For copyright reasons we are unable to reproduce these images, nevertheless the interested reader will easily find them on the Web.

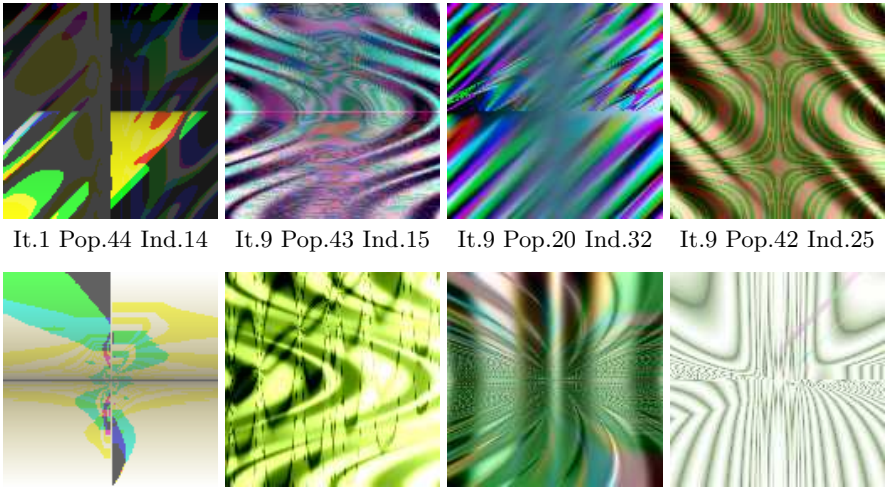



**Fig. 18.10.** Internal images that are most frequently misclassified

## 18.8 Conclusions

We have presented a novel approach that relies on the competition between a classifier and an EC algorithm to promote the iterative refinement of the classifier and to change the fitness landscape of the EC.

The use of a training set that contains human-made artworks and evolutionary ones is one of the key ingredients of our approach. The inclusion of a static aesthetical reference frame composed of human-made artworks provides a stable attractor across evolutionary runs, fostering the production of imagery that relates to human aesthetics. The systematic addition of evolutionary artworks to the training set fosters the refinement of the classifier, promoting stylistic change from one evolutionary run to the other.

The experimental results attained point towards the following results: stylistic change was achieved; the classifiers are able to discriminate between internal and external imagery, attaining success rates above 97.5% in the validation sets; the iterative refinement of the training set, by the addition of the images created by the EC system, gave rise to more discerning classifiers.

The experimental results attained in the independent validation tests confirm these findings, showing the following: the classifiers are able to classify properly images, human-made or artificial, that are not related to the employed training sets, attaining success rates above 89%; that the classifiers can also be used to identify images that stand out from the remaining images of the set.

Although the approach was primarily designed for stylistic change, it can be used for different goals. One of the most obvious applications is its use

to identify shortcomings of classifier systems (e.g., face recognition ones), through the evolution of patterns that are misclassified.

It can also be used to evolve images that match the aesthetic preferences of a given user or set of users. This goal might be attained by replacing the external set of artworks by a set of artworks that match these preferences. In the present paper, the search was oriented to images outside the normal range of NEvAr, which is inherently a difficult task. Using the same approach, one could search images that, although characteristic of NEvAr, are highly valued by the user(s).

It could also be worth exploring the use of the proposed approach, in the context of a partially automated EA tool [8], to assign fitness or to eliminate undesirable imagery. Another possible application is the creation of images that mimic a specific style, including the style of other artificial art tools. A further possibility is the combination of the evaluation made by the classifier with those made by hand-coded fitness functions.

Although the experimental results attained so far are promising, there is still room for improvement. The feature extractor is probably the module that will undergo more changes in the near future.

As previously mentioned, the FE used has limitations in the handling of color information, in particular Hue. The transition to a perceptually uniform color space may mitigate this problem. Following the same set of ideas explored in our research in the musical domain [5, 6], the inclusion of metrics specifically designed to characterize relevant aspects of the images' coloring, such as color consonance or color neighborhood, may also play an important role. The inclusion of features that deal with aspects such as the distribution of the points of interest, texture and contour analysis, can also be a significant improvement.

Due to the considerable computational effort associated with feature extraction and image rendering, we were forced to use a working resolution of $128 \times 128$, which may be too small to allow a good characterization of the images, in particular of the external ones, and the evolution of more refined artworks.

Exploring different ways to map the output of the classifier to fitness values may also prove relevant. In particular, since the analysis of the connection weights of the different ANNs suggests that different strategies are being employed to classify images, using a set of ANNs to guide evolution, instead of using just one, may contribute to a smoother fitness landscape. The replacement of the ANN by an evolutionary classifier is also an interesting possibility.

Although our system has now, in some sense, the ability to "see", it lacks the ability to wander. More precisely, the external images are supplied by us. It would be both interesting and conceptually relevant to let our system navigate through the Internet, collect images, build its own aesthetic references, etc. Alternatively, connecting the system to a camera or TV in order to retrieve images from the "real world" is also an intriguing possibility.

## Acknowledgements

The authors would like to acknowledge Santiago Gonzalez for the integration of the AAC and NEvAr, and for his help on the collection of data and on the setting of the experimental environment. Antonio Seoane provided the scripts used in the analysis of the experimental results. Patrick Roos wrote the code upon which the calculation of the FD is based. Jorge Tavares revised the original version of this paper and provided insightful comments, suggestions and ideas. Alejandro Pazos, Amílcar Cardoso, Antonino Santos, Dwight Krehbiel, Dallas Vaughn, Luca Pellicoro and Marisa Santos made relevant contributions to previous projects upon which the presented work is based, and to related ones. This work was partially supported by research project XUGA-PGIDIT04TIC105012PR.

## References

1. Machado, P., Cardoso, A. (1997). Model proposal for a constructed artist. In Callaos, N., Khoong, C., Cohen, E., eds.: *First World Multiconference on Systemics, Cybernetics and Informatics, SCI97/ISAS97*. Caracas, Venezuela, 521–528
2. Romero, J. (2002). *Metodología Evolutiva para la Construcción de Modelos Cognitivos Complejos. Exploración de la Creatividad Artificial en Composición Musical*. PhD thesis. University of Corunha. Corunha, Spain (in Spanish)
3. Romero, J., Machado, P., Santos, A., Cardoso, A. (2003). On the development of critics in evolutionary computation artists. In Günther, R., et al., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2003: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. Vol. 2611 of LNCS. Essex, UK. Springer
4. Manaris, B., Purewal, T., McCormick, C. (2002). Progress towards recognizing and classifying beautiful music with computers — MIDI-encoded music and the Zipf–Mandelbrot law. In: *Proceedings of the IEEE SoutheastCon*. Columbia, SC
5. Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., Davis, R. (2005). Zipf's law, music classification and aesthetics. *Computer Music Journal*, **29**(1): 55–69
6. 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 Conference on Artificial Intelligence (AAAI 07)*. Vancouver, BC
7. Machado, P., Romero, J., Santos, A., Cardoso, A., Manaris, B. (2004). Adaptive critics for evolutionary artists. In Günther, R., et al., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. Vol. 3005 of LNCS. Coimbra, Portugal. Springer, 435–444
8. Machado, P., Romero, J., Cardoso, A., Santos, A. (2005). Partially interactive evolutionary artists. *New Generation Computing – Special Issue on Interactive Evolutionary Computation*, **23**(42): 143–155

9. Machado, P. (2007). *Inteligência Artificial e Arte*. PhD thesis. University of Coimbra. Coimbra, Portugal (in Portuguese)
10. Machado, P., Cardoso, A. (1998). Computing aesthetics. In Oliveira, F., ed.: *Proceedings of the XIVth Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*. Vol. 1515 of LNCS. Porto Alegre, Brazil. Springer, 219–229
11. Taylor, R.P., Micolich, A.P., Jonas, D. (1999). Fractal analysis of Pollock's drip paintings. *Nature*, **399**: 422
12. Saunders, R. (2001). *Curious Design Agents and Artificial Creativity — A Synthetic Approach to the Study of Creative Behaviour*. PhD thesis. University of Sydney, Department of Architectural and Design Science Faculty of Architecture. Sydney, Australia
13. Manaris, B.Z., Vaughan, D., Wagner, C., Romero, J., Davis, R.B. (2003). Evolutionary music and the Zipf–Mandelbrot law: Developing fitness functions for pleasant music. In Günther, R., et al., eds.: *Applications of Evolutionary Computing, EvoWorkshop 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM*. Vol. 2611 of LNCS. Springer, 522–534
14. Machado, P., Romero, J., Manaris, B., Santos, A., Cardoso, A. (2003). Power to the critics — A framework for the development of artificial art critics. In: *IJCAI 2003 Workshop on Creative Systems*. Acapulco, Mexico
15. Manaris, B., Machado, P., McCauley, C., Romero, J., Krehbiel, D. (2005). Developing fitness functions for pleasant music: Zipf's law and interactive evolution systems. In Rothlauf, F., et al., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. Vol. 3449 of LNCS. Lausanne, Switzerland. Springer, 498–507
16. Sims, K. (1991). Artificial evolution for computer graphics. *ACM Computer Graphics*, **25**: 319–328
17. Machado, P., Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems*, **16**(2): 101–119
18. Greenfield, G. (2002). Color dependent computational aesthetics for evolving expressions. In Sarhangi, R., ed.: *Bridges: Mathematical Connections in Art, Music, and Science; Conference Proceedings 2002*. Winfield, Kansas. Central Plains Book Manufacturing, 9–16
19. Greenfield, G. (2003). Evolving aesthetic images using multiobjective optimization. In McKay, B., et al., eds.: *Congress on Evolutionary Computation, CEC 2003*. Vol. 3. Canberra, Australia. IEEE Press, 1903–1909
20. Svangård, N., Nordin, P. (2004). Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In Günther, R., et al., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. Vol. 3005 of LNCS. Coimbra, Portugal. Springer, 445–454
21. Baluja, S., Pomerlau, D., Todd, J. (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, **6**(2): 325–354
22. Greenfield, G.R. (2002). On the co-evolution of evolving expressions. *International Journal of Computational Intelligence and Applications*, **2**(1): 17–31
23. Todd, P.M., Werner, G.M. (1999). Frankensteinian approaches to evolutionary music composition. In Griffith, N., Todd, P.M., eds.: *Musical Networks: Parallel Distributed Perception and Performance*. MIT Press, 313–340

24. Birkhoff, G. (1933). *Aesthetic Measure*. Harvard University Press
25. Moles, A. (1958). *Théorie de L'Information et Perception Esthétique*. Denoel
26. Arnheim, R. (1956). *Art and Visual Perception, a Psychology of the Creative Eye*. Faber and Faber. London
27. Arnheim, R. (1966). *Towards a Psychology of Art/Entropy and Art — An Essay on Disorder and Order*. The Regents of the University of California
28. Arnheim, R. (1969). *Visual Thinking*. University of California Press. Berkeley, CA
29. Bense, M. (1965). *Aesthetica. Einführung in die neue Aesthetik*. Agis (reprinted by Baden-Baden, 1982)
30. Shannon, C.E. (1951). Prediction and entropy of printed english. *Bell System Technical Journal*, (30): 50–64
31. Staudek, T. (2002). *Exact Aesthetics. Object and Scene to Message*. PhD thesis. Faculty of Informatics, Masaryk University of Brno
32. Staudek, T., Linkov, V. (2004). Personality characteristics and aesthetic preference for chaotic curves. *Journal of Mathematics and Design*, **4**(1): 297–304
33. Graves, M. (1948). *Design Judgement Test*. The Psychological Corporation. New York
34. Datta, R., Joshi, D., Li, J., Wang, J.Z. (2006). Studying aesthetics in photographic images using a computational approach. In: *Computer Vision – ECCV 2006, 9th European Conference on Computer Vision, Part III*. LNCS. Graz, Austria. Springer, 288–301
35. Aks, D., Sprott, J.C. (1996). Quantifying aesthetic preference for chaotic patterns. *Empirical Studies of the Arts*, **14**: 1–16
36. Spehar, B., Clifford, C.W.G., Newell, N., Taylor, R.P. (2003). Universal aesthetic of fractals. *Computers and Graphics*, **27**(5): 813–820
37. Wannarumon, S., Bohez, E.L.J. (2006). A new aesthetic evolutionary approach for jewelry design. *Computer-Aided Design & Applications*, **3**(1-4): 385–394
38. Wertheimer, M. (1939). Laws of organization in perceptual forms. In Ellis, W.D., ed.: *A Source Book of Gestalt Psychology*. Harcourt Brace. New York, 71–88
39. Tyler, C.W., ed. (2002). *Human Symmetry Perception and Its Computational Analysis*. Lawrence Erlbaum Associates
40. Field, D.J., Hayes, A., Hess, R.F. (2000). The roles of polarity and symmetry in the perceptual grouping of contour fragments. *Spatial Vision*, **13**(1): 51–66
41. Cope, D. (1992). On the algorithmic representation of musical style. In Laske, O., ed.: *Understanding Music with AI: Perspectives on Music Cognition*. MIT Press. Cambridge, Massachusetts, 354–363
42. Zipf, G.K. (1949). *Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley
43. Kittler, J. (1983). On the accuracy of the Sobel edge detector. *Image Vision Computing*, **1**(1): 37–42
44. Zell, A., Mamier, G., Vogt, M., Mache, N., Hübner, R., Döring, S., Herrmann, K.U., Soyez, T., Schmalzl, M., Sommer, T., et al. (2003). *SNNS: Stuttgart Neural Network Simulator User Manual, Version 4.2*. Technical Report 3/92. University of Stuttgart. Stuttgart
45. Directmedia (2002). 10000 Meisterwerke der Malerei — von der Antike bis zum Beginn der Moderne. DVD-ROM