# Feature Selection and Novelty in Computational Aesthetics

João Correia[1], Penousal Machado[1], Juan Romero[2], and Adrian Carballal[2]

[1] CISUC, Department of Informatics Engineering, University of Coimbra,
3030 Coimbra, Portugal
{jncor,machado}@dei.uc.pt

[2] Faculty of Computer Science, University of A Coruña, Coruña, Spain
{jj@udc.es,adrian.carballal}@udc.es

**Abstract.** An approach for exploring novelty in expression-based evolutionary art systems is presented. The framework is composed of a feature extractor, a classifier, an evolutionary engine and a supervisor. The evolutionary engine exploits shortcomings of the classifier, generating misclassified instances. These instances update the training set and the classifier is re-trained. This iterative process forces the evolutionary algorithm to explore new paths leading to the creation of novel imagery. The experiments presented and analyzed herein explore different feature selection methods and indicate the validity of the approach.

## 1 Introduction

The development of aesthetic judgement systems (AJSs) is one of the major challenges in the field of Evolutionary Art [15] and a necessary step for the creation of an autonomous Artificial Artist (AA) [13]. Over the course of the years, some researchers developed hardwired aesthetic measures while others focused on the application of Machine Learning (ML) techniques to learn aesthetic preferences.

The work presented herein follows this second line of research. We employ a Feature Extractor (FE) to analyze and synthesize the characteristics of images. These are used to train an Artificial Neural Network (ANN), whose output is used to assign fitness to the images produced by an expression-based evolutionary art system. The overall approach is inspired on the work of Machado et al. [13], but there are significant differences at the architecture and implementation level.

The most notable difference pertains to feature selection. Machado et al. perform several offline experiments to determine a subset of the features of the FE. This fixed subset is then used as input to the ANNs throughout all iterations and evolutionary runs. We adopt a different approach: the subset of features is determined dynamically, by automatic means, at the beginning of each iteration. At the implementation level the most significant differences are: (i) the FE was augmented; (ii) new, and significantly larger, initial datasets were constructed; (iii) the strategy used to update the image datasets is different.

Similarly to Machado et al. [13], more than learning aesthetic principles, we are interested in promoting stylistic change from one evolutionary run to another

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 133–144, 2013.

by forcing evolution to explore new paths, with the ultimate goal of promoting novelty and increasing the potential for creativity.

## 2 Related Work

There are several notable examples of AJSs relying on hardwired aesthetic measures (e.g. [21,12,6]) and also recent works comparing the merits of such aesthetic measures [5,8,1,18]. Usually the AJSs that employ ML, extract information from the images and employ ML to perform aesthetic-based classification or evaluation, which is a common approach in the field of content based image retrieval (e.g., [4,9,17]).

The combination of this type of AJS with an Evolutionary Art tool has also been explored. In their seminal work Baluja et al. [2] used an ANN trained with a set of images generated by user-guided evolution to assign fitness. [19] used Self Organizing Maps to evolve novel images. Machado et al. [13,14] study the development of AAs able to perform style changes over the course of several runs. To attain this goal they employ a FE, ANNs trained to distinguish between internal and external imagery, and an expression-based EC engine, promoting an arms-race between the EC engine and the ANNs. In a related work, Li et al. [11] investigate aesthetic features to model human preferences. The aesthetic model is built by learning both phenotype and genotype features, which are extracted from internal evolutionary images and external real world paintings.

Kowaliw et al. [10] compared biomorphs generated randomly, through interactive evolution, and through automatic evolution using a classifier system inspired by content based image retrieval metrics. The experimental results indicate that the results of the automatic system were comparable to those obtained by interactive evolution.

## 3 The Framework

The framework comprises three main modules: Evolutionary Computation (EC) engine, Classifier and Supervisor. Figures 1 and 2 present an overview of the framework and the interaction between the EC engine and Classifier, respectively. The application of this framework involves the following steps:

1. Selection of an Internal and an External image dataset;
2. A Classifier System (CS) is trained to distinguish between the Internal and the External instances;
3. N independent EC runs are started; The output of the CS is used to assign fitness to the generated individuals;
4. The EC runs stop when a termination criterion is met (e.g., a defined number of generations, attaining a fitness value);
5. A Supervisor selects and filters instances gathered from the EC runs;
6. The process is repeated from step 2 until a termination criterion is met.

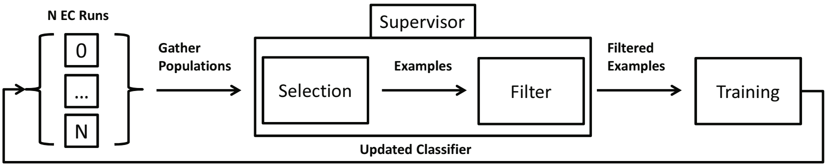

**Fig. 1.** Overview of the system

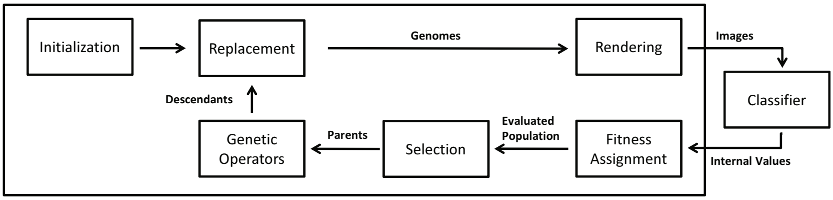

**Fig. 2.** Evolutionary model and its interaction with the classifier

For the purpose of this paper the framework was instantiated as described next. The EC engine is an expression-based evolutionary art tool (see section 4.2). The Internal set is composed, exclusively, of images generated by the EC engine in runs where fitness was assigned randomly. As such, it represents the type of imagery that the system tends to create when no aesthetic preference is imposed. The External set is composed of famous paintings and artworks. It serves as an aesthetic reference for the system. The CS comprises feature extraction, feature selection and an ANN, which performs the classification. The output of the ANN is used to assign fitness. Images that are classified as External, i.e. misclassified, have higher fitness than those classified as Internal. The Supervisor manages the evolved instances. In this instantiation, it picks individuals from the EC run that are not present in the *internal* set and sequentially substitutes the existing ones. Thus, the Supervisor modifies the internal set, by iteratively replacing the old examples with unique individuals generated during the EC runs. Although this replacement strategy may eliminate relevant instances, it ensures balance between the cardinality of the classes.

The entire approach relies on promoting a competition between the EC engine and the CS. In each iteration the EC engine must evolve individuals that are misclassified by the CS. To do so, it must evolve images that are atypical to the EC engine. Since the images evolved in each iteration are added to the internal dataset, the engine is forced to reinvent itself, exploring new regions of the search space, in a permanent quest for novelty.

## 4 Experimental Setup

In this section we describe the settings employed in the experiments presented in this paper, explaining the details of the image classifier (4.1); of the EC engine (4.2); initialization and updating methodology (4.3).

### 4.1 Image Classification

The CS is composed of an image classification module that uses, a FE, an ANN, and a Feature Selection methodology.

The FE is responsible for extracting characteristics from the images. This process implies: (i) Pre-processing, which includes all the transformation and normalization operations applied to a given input image; (ii) Application of metrics, the application of certain methods based on statistical measurements and image complexity estimates; (iii) Feature building, the extraction of results from the metrics in order to build the image feature set.

The FE converts all images to a 128 × 128 pixels and 8 bits per channel format, to ensure that all input images are evaluated under the same conditions. The images resulting from these operations are subject to transformation operations. These transformations include: no filter, which means no transformation applied; Sobel and Canny based edge detection; an image transform operation, the distance transform; a quantization algorithm; a salience algorithm, the subject salience.

Afterwards the FE calculates the following metrics: average (i) and standard deviation; (ii) of the image pixel values; complexity estimates based on JPEG; (iii) and fractal compression [18]; (iv) Zipf Rank-Frequency (v) and Size-Frequency (vi), which result from the application of the Zipf's law [23]; (vii) Fractal dimension estimates using the box-counting method [22].

Splitting the image in color channels, applying the previously mentioned transformations to each channels, and applying the metrics to each of the resulting images, yields a total of 804 features per image. More information on the feature extractor can be found in [3].

The choice of an ANN based classifier is justified by its success in [13]. The ANN receives as input the feature vector. It has one hidden layer of 15 neurons, and two output neurons, one per each class, and is trained by backpropagation. This architecture was established in preliminary experiments.

To avoid a "binary" output, i.e. both neurons returning either 0 or 1, which would result in an unsuitable fitness landscape, we employ a tolerance threshold during the training stage. This means that during the backpropagation of the error, if the difference between the output of the network and the desired output is below the maximum tolerated threshold, then the error is propagated back as zero (no error). The classifier was built using WEKA's[1] FastNeuralNetwork.

Feature selection methods are typically composed by an *evaluation* criteria and a *search* method. The necessity of adopting a cost effective approach made us to adopt a *filter* evaluation based on the feature statistics and *sub-optimal* search. The choice of deterministic search approach makes the analysis of the results less complex. Considering these constraints, we employed CfsSubsetEval [7] for evaluation combined with a best first search algorithm. The CfsSubsetEval evaluates the worth of a subset of features, by statistically processing each feature, in terms of information redundancy and correlation with the class. Resulting subsets of features tend to be highly correlated with the class and with

[1] WEKA 3: Data Mining Software in Java - `http://www.cs.waikato.ac.nz/ml/weka/`

low intercorrelation. The best-first algorithm, defines the CsfSubsetEval as its heuristic function and scores the set of features near its starting point, then it expands to the available node of features with highest score. The search stops when a pre-determined number of non-improving sets of features is encountered. Both algorithms are provided by WEKA and were integrated in the system.

To assess the validity of the proposed approach we conducted two independent experiments with different feature selection methods:

1. FW – which uses forward feature selection;
2. BW – which uses backward feature selection;

When forward selection is used, the selection algorithm starts with an empty set of features and it incrementally adds features until a termination criterion is met. Backward selection starts with the full set and removes features until the criterion is reached. From here on, we will use the terms FW and BW to refer to the each of the experiments.

### 4.2 Genetic Programming Engine

The EC engine is inspired by the works of Sims [20]. It is a general purpose, expression-based, GP image generation engine that allows the evolution of populations of images. The genotypes are trees composed from a lexicon of functions and terminals. The function set is composed of simple functions such as arithmetic, trigonometric and logic operations. The terminal set is composed of two variables, $x$ and $y$, and random constant values. The phenotypes are images that are rendered by evaluating the expression-trees for different values of $x$ and $y$, which serve both as terminal values and image coordinates. In other words, to determine the value of the pixel in the $(0, 0)$ coordinates, one assigns zero to $x$ and $y$ and evaluates the expression-tree. A thorough description of the GP engine can be found in [16]. The following settings where used: pop. size = 100; generations = 50; crossover probability = 0.8; initialization method = ramped half-and-half; initial maximum depth = 5; mutation max tree depth = 3; Function set = { +, -, *, /, min, max, abs, sin, cos, if, pow, mdist, warp, sqrt, sign, neg}; Terminal set = {X, Y, scalar and vector random constants}.

To assign fitness: the individuals are rendered; the FE is applied to extract features; the feature vector composed of the features determined by the feature selection method is the input of the ANN; the output of the ANN, i.e. the classification, determines fitness. In this case, and since we intend to promote the evolution of novel imagery, fitness of an individual is equal to the activation value of the output neuron indicating external class membership. In other words, images that cause classification errors are preferred.

### 4.3 Initialization and Updating

The *External* dataset holds 25000 paintings from different authors, styles and periods (Fig. 3). Among others, it includes paintings of: Michelangelo, Picasso,

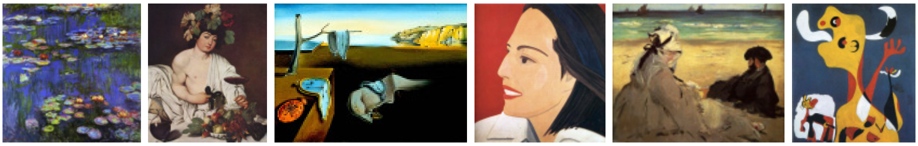

**Fig. 3.** Examples of images of the external dataset

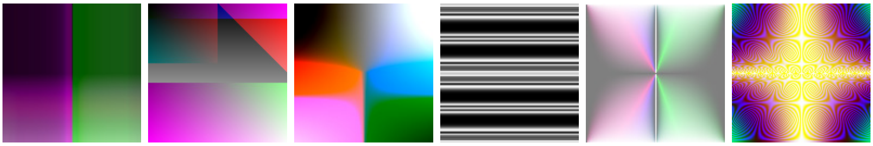

**Fig. 4.** Examples of images of the internal dataset

Monet, Gaugin, Dalí, Cézanne, Da Vinci, Matisse, and Chagall. The *Internal* dataset is a randomly generated population of 25000 individuals (Fig. 4).

The ANN training parameters imply that 45000 images will be used for training and 5000 for testing purposes, 22500 *internal* and 22500 *external* images for training and 2250 of each for testing.

The supervisor updates the internal dataset by iteratively replacing the initial images with evolved images. Repeated instances are not added to the set. The boosting process stops when the Supervisor replaces 22500 of the internal images by novel evolved images.

## 5 Experimental Results

As previously mentioned, the most important difference between our work and the work of Machado et al. [13] is the adoption of dynamic feature selection mechanisms. We test two feature selection mechanisms, FW and BW, by conducting independent experiments with each of them.

Each iteration starts with the selection of the features that will be used by the ANN. Once the feature set is determined a classifier is trained. Table 1 shows the performance of the classifiers across iterations in both training and test sets.

In the first iteration, the performance in training is almost perfect for both feature selection methodologies: FW misclassifies one and BW zero out of 45000 training instances. In the test instances both classifiers performed flawlessly. In the first iteration the BW classifier uses 45 features while the FW classifier uses 30 (table 2). This indicates that a small number of features is sufficient to correctly identify all training and test instances in the initial conditions.

These classifiers are used to guide the evolutionary runs of each configuration during the first iteration of the boosting algorithm. As it can be observed in table 3, the EC engine was able to generate a significant number of individuals that were classified as external in both cases.

The individuals evolved during the first iteration are added to the internal set images, replacing the randomly generated ones, and a new iteration is started. This implies a new feature selection step and the training of new classifiers.

**Table 1.** Performance of FW and BW during training and test, in terms of percentage of correctly classified examples (%C), False Externals (Ext), i.e. an internal image that is classified as external, and False Internals (Int)

| | Train | | | | | | Test | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FW | | | BW | | | FW | | | BW | | |
| Iteration | %C | Int | Ext | %C | Int | Ext | %C | Int | Ext | %C | Int | Ext |
| 1 | 99.99 | 1 | 0 | 100.00 | 0 | 0 | 100.00 | 0 | 0 | 100.00 | 0 | 0 |
| 2 | 99.89 | 31 | 18 | 100.00 | 0 | 0 | 99.90 | 2 | 3 | 100.00 | 0 | 0 |
| 3 | 99.70 | 58 | 75 | 99.98 | 11 | 0 | 99.70 | 4 | 11 | 100.00 | 0 | 0 |
| 4 | 99.95 | 3 | 20 | 99.90 | 39 | 5 | 99.94 | 0 | 3 | 99.94 | 1 | 2 |
| 5 | 99.00 | 254 | 196 | 99.62 | 53 | 117 | 99.04 | 24 | 24 | 99.60 | 5 | 15 |
| 6 | 98.69 | 412 | 179 | 99.97 | 3 | 12 | 98.38 | 52 | 29 | 99.96 | 0 | 2 |
| 7 | 99.57 | 131 | 64 | 99.96 | 14 | 6 | 99.62 | 13 | 6 | 99.96 | 1 | 1 |
| 8 | 98.90 | 349 | 147 | 99.95 | 13 | 10 | 99.02 | 35 | 14 | 99.90 | 1 | 4 |
| 9 | 99.56 | 183 | 14 | 99.96 | 6 | 11 | 99.58 | 19 | 2 | 99.96 | 0 | 2 |
| 10 | 99.85 | 48 | 19 | 99.98 | 6 | 3 | 99.82 | 7 | 2 | 99.98 | 0 | 1 |
| 11 | 99.96 | 19 | 0 | 99.99 | 3 | 0 | 100.00 | 0 | 0 | 99.98 | 1 | 0 |

**Table 2.** Number of features selected per iteration

| Iteration | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FW | 30 | 22 | 19 | 37 | 11 | 11 | 11 | 7 | 15 | 8 | 25 | 19.25 |
| BW | 45 | 138 | 98 | 97 | 105 | 48 | 55 | 53 | 53 | 58 | 43 | 72.09 |

As it can be observed in table1 the BW feature selection approach obtains better overall performance in classification than FW. The results attained in training are similar to those attained in the test sets, both in terms of overall trend and in terms of percentage of errors, indicating that the classifiers generalize well. Several fluctuations in performance exist, which may be explained by the following factors: (i) The adopted feature selection methodologies do not ensure optimality; (ii) The replacement of individuals of the internal set may have eliminated instances that are hard to classify.

Table 3 displays, for each experiment and across iterations, the number of images generated by the EC that are classified as external. As it can be observed, in both experiments, the evolutionary engine was able to find images classified as externals in all iterations. The number of populations necessary to find a false external tends to increase as the number of iterations increases. This indicates that the classifiers are becoming more selective. Further testing is necessary to draw conclusive statements regarding this issue.

Table 2 shows the number of features used by the classifiers in each iteration. As expected, updating the internal set causes changes in the feature selection process both in terms of number and type of the selected features. Although variations occur, the number of features used is significantly lower than the total number of features available (804) indicating that good performance is attainable

**Table 3.** Number of false externals generated in each iteration

| Iteration | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FW | 1165 | 1081 | 1575 | 840 | 832 | 1755 | 1110 | 651 | 271 | 70 | 1184 | 10534 |
| BW | 1451 | 1091 | 1074 | 1261 | 409 | 441 | 613 | 499 | 292 | 484 | 1458 | 9073 |

with a relatively low number of features. FW tends to use a smaller number of features than BW selection, which is a direct consequence of the algorithm.

It is interesting to notice that the number of features used by BW increases abruptly from the 1st to the 2nd iteration (45 to 138 features), remains relatively high during several iterations (3rd to 5th), but then drops to values comparable to the initial ones. Relating these results with the high number of false externals found in the 11th iteration, where the EC run found such images in few generations, leads us to to believe that the replacement of the initial randomly generated images from the internal set may be eliminating relevant instances. That is, the correlation among the evolved images may be higher than the correlation among randomly generated ones and when the number of randomly generated images drops significantly the classification task becomes easier.

Figures 5 and 6 present the fittest individuals from populations 1, 10, 20, 30 and 50, for all iterations. Looking at the images produced in the first iteration, the only where all both classifiers share the same internal and external set, one can observe that different feature selection methods converged to different types of imagery. The FW experiment converged to highly complex, "noisy" colored images. Since the initial internal set is composed, due to the function set, of simplistic images, the evolution of complex images is an obvious way to misguide the CS. The BW experiment appears to have converged to images that are characterized simultaneously by (i) a complex and noisy "background" and by (ii) low contrast and high luminosity. The convergence to this type of imagery can also be explained by the unlikelihood of randomly creating images with these properties, and hence their absence from the initial internal set.

From the first iteration onwards the classifiers no longer share a similar set of external and internal images. As expected, the images evolved in a given iteration of a particular experiment tend to be stylistically different from the ones evolved in the corresponding iteration of a different experiment. Nevertheless, and although it is subjective to say it, comparing the results of the FW and BW experiments appears to indicate that they may be exploring similar styles in different iterations (for instance, the false externals evolved in the 7th iteration of the FW experiment are somewhat similar to the false externals evolved in the 4th iteration of BW).

It seems that the images evolved within each iteration of the FW experiment tend to be less diverse than the ones evolved using BW. In the 2nd, 3rd, 6th, 9th and 10th iterations of FW the algorithm quickly converged and there is little variety. In contrast, with the exception of the 7th and 8th iterations, the BW experiments produced diverse imagery within each iteration.

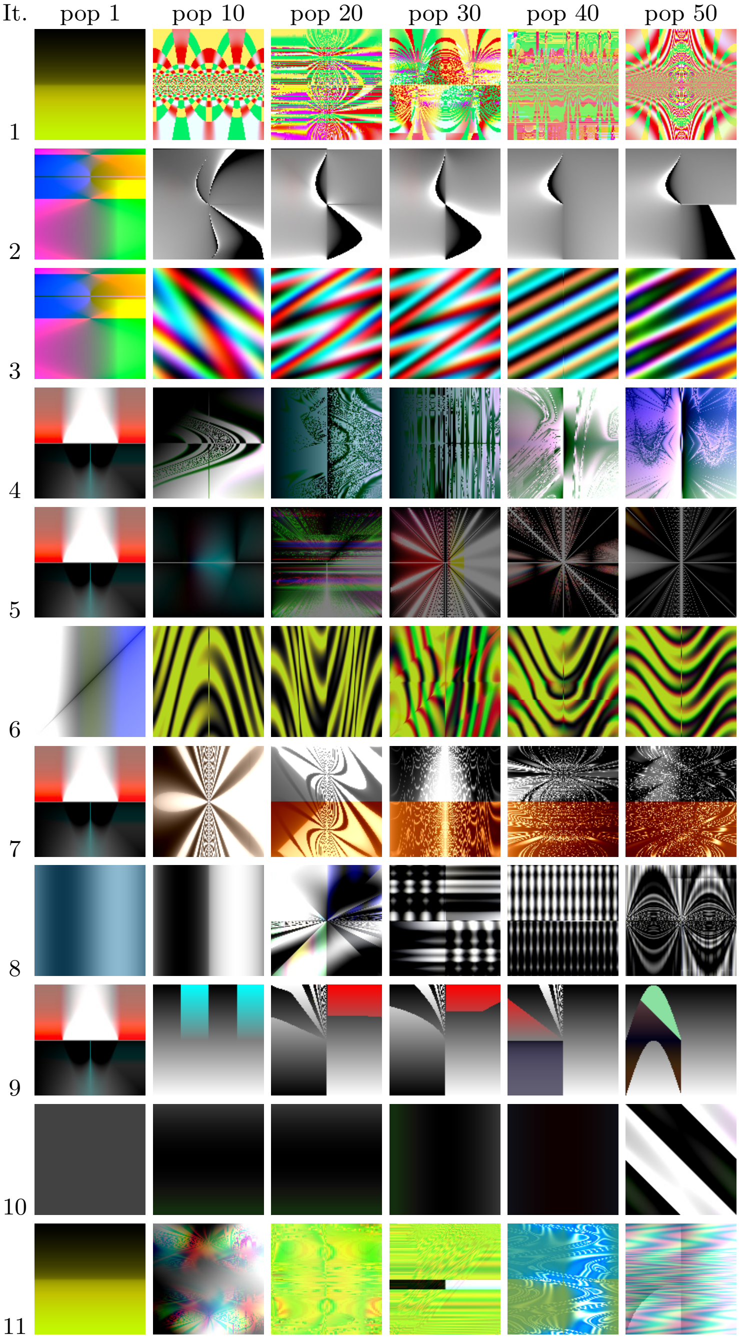

**Fig. 5.** Fittest images from populations 1, 10, 20, 30, 40, 50 evolved using FW

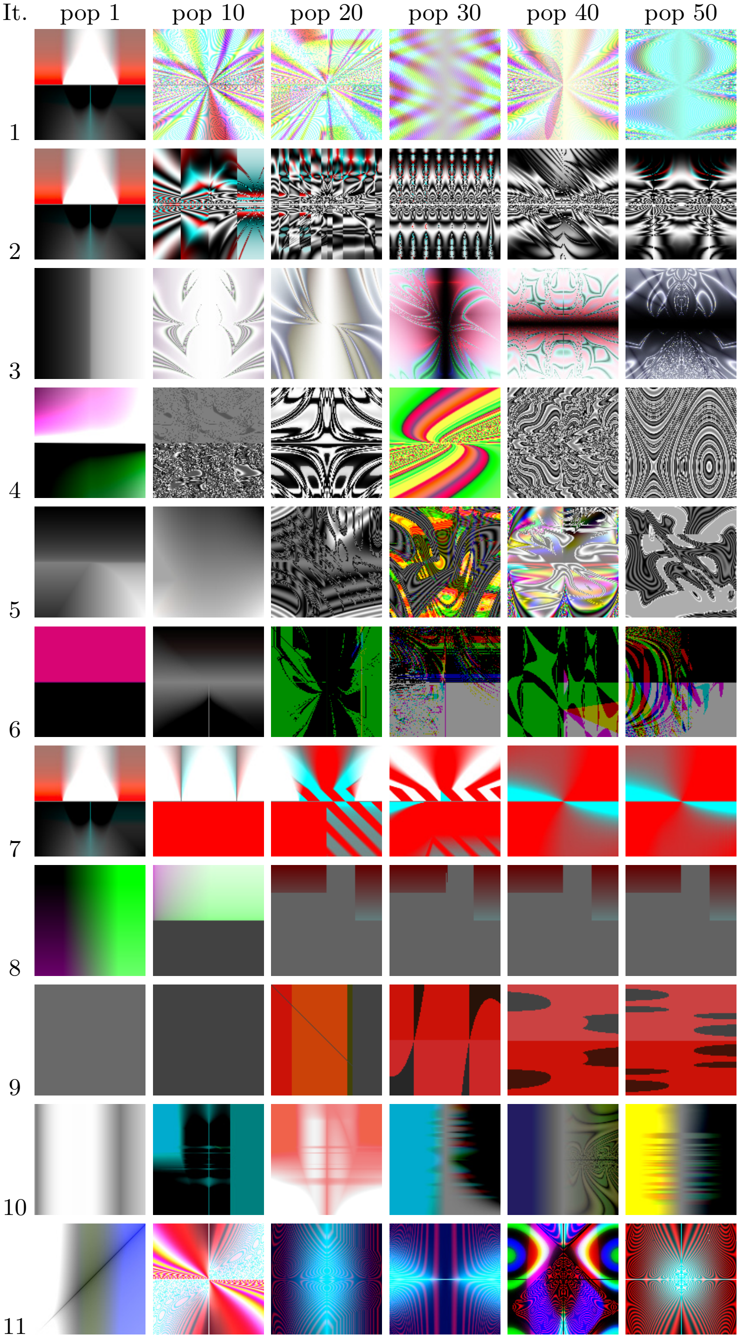

**Fig. 6.** Fittest images from populations 1, 10, 20, 30, 40, 50 evolved using BW

When we compare the imagery produced in each iteration, one can observe (for BW and FW) "stylistic" changes across iterations which was the desired result. It is also obvious that these images do not match the ones belonging to the external set composed of paintings. Like in Machado et al. [13] more than evolving images that resemble paintings the system is, apparently, evolving images that are atypical of the EC engine and hence cause classification errors.

## 6 Conclusions and Future Work

We have presented an approach to explore novelty and stylistic change in expression based evolutionary art systems. The system relies in a framework composed of a classifier system, evolutionary engine and a supervisor. The most prominent feature of our approach is the adoption of dynamic feature selection methods.

We focused our experiments in the comparison of the results obtained when using different feature selection approaches. The results show that the evolutionary runs explore different paths throughout the iterations. The changes produced by the replacement of images of the internal set lead to the selection of new sub-sets of features, which, in turn, contribute for the development of novel imagery. The experimental results also indicate that, in the considered settings, it is possible to achieve good performance using a relatively small subset of features.

Although the results are not entirely conclusive, they can be considering promising and demonstrate the potential of the proposed framework. Further experimentation is already taking place. We are particularly interested in: (i) Confirming the experimental findings via further testing and analysis (ii) testing alternative image replacement schemes (e.g. only adding images that were misclassified and avoiding replacement) (iii) Producing experiments with an higher number of iterations (iv) Testing alternative feature selection schemes.

**Acknowledgments.** This research is partially funded by: the Portuguese Foundation for Science and Technology, project PTDC/EIA–EIA/115667/2009; Xunta de Galicia, project XUGA–PGIDIT10TIC105008PR.

## References

1. Atkins, D.L., Klapaukh, R., Browne, W.N., Zhang, M.: Evolution of aesthetically pleasing images without human-in-the-loop. In: IEEE Congress on Evolutionary Computation, pp. 1–8. IEEE (2010)
2. Baluja, S., Pomerlau, D., Todd, J.: Towards automated artificial evolution for computer-generated images. Connection Science 6(2), 325–354 (1994)
3. Correia, J.: Evolutionary Computation for Assessing and Improving Classifier Performance. Master's thesis, University of Coimbra (2009)
4. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. 40, 5:1–5:60 (2008)
5. Ekárt, A., Joá, A., Sharma, D., Chalakov, S.: Modelling the underlying principles of human aesthetic preference in evolutionary art. Journal of Mathematics and the Arts 6(2-3), 107–124 (2012)

6. Greenfield, G., Machado, P.: Simulating Artist and Critic Dynamics. In: Proceedings of the International Joint Conference on Computational Intelligence, Funchal, Madeira, Portugal, October 5-7, pp. 190–197 (2009)
7. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Langley, P. (ed.) Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29-July 2, pp. 359–366. Morgan Kaufmann (2000)
8. den Heijer, E., Eiben, A.E.: Comparing Aesthetic Measures for Evolutionary Art. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) EvoApplications 2010, Part II. LNCS, vol. 6025, pp. 311–320. Springer, Heidelberg (2010)
9. Ke, Y., Tang, X., Jing, F.: The Design of High-Level Features for Photo Quality Assessment. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 419–426 (2006)
10. Kowaliw, T., Dorin, A., McCormack, J.: An Empirical Exploration of a Definition of Creative Novelty for Generative Art. In: Korb, K., Randall, M., Hendtlass, T. (eds.) ACAL 2009. LNCS, vol. 5865, pp. 1–10. Springer, Heidelberg (2009)
11. Li, Y., Hu, C., Chen, M., Hu, J.: Investigating Aesthetic Features to Model Human Preference in Evolutionary Art. In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 153–164. Springer, Heidelberg (2012)
12. Machado, P., Romero, J., Cardoso, A., Santos, A.: Partially interactive evolutionary artists. New Generation Computing 23(42), 143–155 (2005)
13. Machado, P., Romero, J., Manaris, B.: Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 381–415. Springer, Heidelberg (2007)
14. Machado, P., Romero, J., Santos, A., Cardoso, A., Pazos, A.: On the development of evolutionary artificial artists. Computers & Graphics 31(6), 818–826 (2007)
15. McCormack, J.: Facing the future: Evolutionary possibilities forhuman-machine creativity. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution. Natural Computing Series, pp. 417–451. Springer (2008)
16. Romero, J., Machado, P.: The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Natural Computing Series. Springer, Heidelberg (2007)
17. Romero, J., Machado, P., Carballal, A., Correia, J.: Computing aesthetics with image judgement systems. In: McCormack, J., D'Inverno, M. (eds.) Computers and Creativity. Springer (2012)
18. Romero, J., Machado, P., Carballal, A., Santos, A.: Using complexity estimates in aesthetic image classification. Journal of Mathematics and the Arts 6(2-3), 125–136 (2012)
19. Saunders, R.: Curious Design Agents and Artificial Creativity. Ph.D. thesis, University of Sydney, Sydney, Australia (2001)
20. Sims, K.: Artificial Evolution for Computer Graphics. ACM Computer Graphics 25, 319–328 (1991)
21. Staudek, T.: Computer-aided aesthetic evaluation of visual patterns. In: ISAMA-BRIDGES Conference Proceedings, Granada, Spain, pp. 143–149 (July 2003)
22. Taylor, R.P., Micolich, A.P., Jonas, D.: Fractal analysis of Pollock's drip paintings. Nature 399, 422 (1999)
23. Zipf, G.K.: Human Behavior and the Principle of Least-Effort. Addison-Wesley, Cambridge (1949)