Aesthetic Learning for Image Synthesis

Michael Gircys

***Abstract*—Building on a previously constructed evolutionary art system, the possibility of using machine learning techniques to enhance system autonomy is explored. Pairing Genetic Programming with aesthetic fitness, images are synthesized to allow for feature extraction which permits a pair of classifiers to learn aesthetic judgements. The experiments performed for this report attempt to clarify configurations from Learning Aesthetic Judgements in Evolutionary Art Systems by Li *et al.* [9], from which tuning options and the effectiveness of such a system is explored.**

I. Introduction

A SIGNIFICANT problem for interactive evolutionary art systems is user fatigue, where continued evaluation of many evolved individuals becomes exhausting for the user [7] [8] [10]. Common resolutions are to reduce population sizes and the number of generations that a system evolves, however this introduces new issues of genetic diversity. The ability to capture and learn aesthetic preferences would be a great boon for the resolution of user fatigue issues, and would provide excellent resources for the development of aesthetic models. It is for these reasons that Li *et al.* [9] proposed their system of aesthetic learning within the context of texture synthesis.

In this system, a pair of classifiers can be trained using the previous generation of evolved images. In Learning Aesthetic Judgements in Evolutionary Art Systems, Li *et al.* [9] used a collaboration of C4.5 decision trees and feed-forward neural networks. Using PCA techniques to reduce a large array of image attributes, both algorithms should be able to efficiently classify an image as holding a certain discrete rating. The report by Li *et al.* outlines much of the needed information to duplicate their system's configuration, however a number of key implementation details have not been completely covered. It the purpose of this report to both explore some of the missing details, and to see if performance can't be further tuned.

This report will continue with a brief examination of the former work by Li *et al.*, and the ambiguities that we will attempt to clarify. The previously developed interactive art system will be introduced to provide a note on the capabilities and limitations of the system's image synthesis abilities. The extracted features will be briefly noted, along with any divergences from Li's system. We can then discuss the learning system proposed by Li, and any immediate limitations and questions. With a number of concerns about the system collected, we will define a number of experiments that may increase the effectiveness of such a learning system, and then verify if any performance increases were found. Finally, we will summarize and conclude any found results.

II. Problem Definition & Background

Previous work was completed at Brock University for the COSC 5P71 (Genetic Programming) course where rudimentary, interactive evolutionary art system (EAS) was developed. Genetic Programming (GP) methods were used to evolve images based on feedback produced in the form of user ratings. Evolved images were rated on a discrete [1,5] scale, where a higher rating suggests a proportionately better fitness score.

We hope to improve the user experience through the addition of a supervised learning system capable of making aesthetic judgements. The ratings will constitute the possible classifications that a supervised aesthetic learning system will produce and learn from. This learning system should be able to receive sets of evolved individuals - their ratings and extracted features - and train classifiers to emulate the learned aesthetic judgements. It is hoped that such a system will permit a user to occasionally skip evaluating each individual themselves, enabling them to produce a larger amount of candidate images for their consideration. One such system was proposed in a paper by Li *et al.* [9], and we intend to implement and further configure such a system.

One critical motivation among many for the development of an aesthetic learning system is to combat the issue of user fatigue [7] [8] [10]. If a user is required to assess and rank every individual, the work can become monotonous, and otherwise unenjoyable and tiring. Consequently, both population size and the number of generations used in a model may need to be reduced to maintain user engagement. However, this can greatly reduce genetic diversity, and require systems with low convergence.

A. Procedural Texture Synthesis

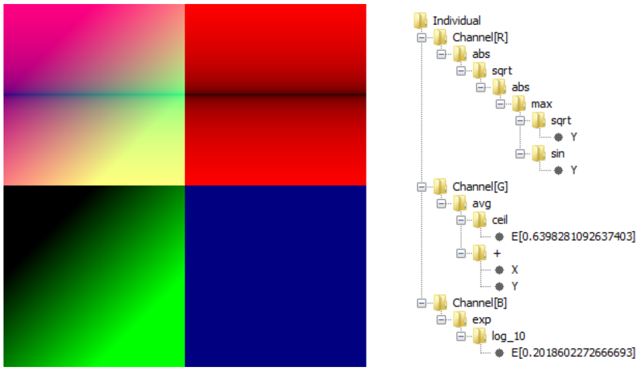
A procedural texture uses a function to specify the colour value of a pixel at each position. This approach may permit composition, and produces an abstraction away from specific details. Texture synthesis and evolutionary art has been explored since 1991 with the work of Karl Sims [1] [5]. Genetic programming representations were used to symbolically encode and evolve 2D pixel colour evaluating functions (similar to Figure 1).

B. GP Language and Settings

The parameters used for the GP system are based on the defaults proposed by Koza [12], and further empirically tuned from previous academic course work.

A few noted variances between our developed system and Li's includes the population size, initial maximum tree depth, and breeder pipeline settings. As Li's system displays smaller

Fig. 1: Texture Synthesis with GP



images, an increased number of them may be shown on screen; the population in our system was chosen to match the maximum that could be displayed on a 1920x1080 resolution screen. A population of 56 vs 67 may reduce the amount of samples provided to the learning system after each generation, and should be considered. The use of an initial maximum tree depth of 5 is likely not substantially different from 6. However, Li does not specify a initial minimum tree depth. As our experiments aim to produce simple image characteristics, a low initial tree depth should not be problematic. Li reports finding high fitness and classification ability after 20 evolutionary generations. We will examine 30 generations to further confirm any convergence or divergence on aesthetic values.

A key notable difference can be found among the breeder pipelines of this system and Li's. Li *et al.* permit 4 unique mutation operators which can affect how coarse or fine the resultant mutation will appear. On inspection of their colour mutation, it would appear that each individual also tracks which colour scheme in which the individual's phenotype should be rendered. In an attempt to provide for a finer mutation operation, our system has included an Ephemeral constant adjustment operation as a possible breeder option. The report by Li *et al.* does not appear to mention the breeder operation frequencies, outside of the fact that it can be adjusted by the user in a [0%,100%] range, which our system also supports. For the purpose of the automated testing, breeder pipeline probabilities are outlined in Table I.

The GP system developed for the report shares a set of common mathematical function nodes between it and the system by Li. In comparison to Li's EAS (evolutionary art system) [9], we provide a number of additional mathematical operators, but omit a few of the noise functions, and the pair of positionally dependant geometric operators (spiral, circle). A full list of functions and terminals is provided in Table II.

Experimentation will be done to ensure sufficient expressibility within the system.

TABLE I: GP Default Parameters

Parameter	Value
Generations	30
Population Size	56
Generation 0	
Builder	Ramped Half & Half
New Node Depth	[2,6]
Grow Probability	50%
Parent Selection	
Elitism	1
Selection	Tournament
Selection Size	3
Node Selection	
Terminals	10%
Non-Terminals	90%
Breeder Pipeline	
Reproduction	0%
Crossover	80%
Mutation	10%
Ephemeral Mutation	10%
Crossover Settings	
Max Depth	17
Attempts	1
Mutation Settings	
Max Depth	17
Attempts	1
Builder	Grow
New Node Depth	[5,5]
Ephemeral Mutation Settings	
Change Factor	[0%, 1%] Max

TABLE II: GP Default Language

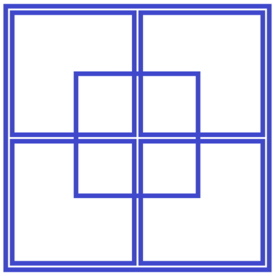
Sign	Description
Variables	
X	Current horizontal position
Y	Current vertical position
Ephemerals	
E[#]	Random constant in [0,1]
E[#]	Random constant in [0,10]
E[#]	Random constant in [0,100]
Math, Unary	
-	Sign change
sin	Trigonometric cosine (taking radians)
cos	Trigonometric sine (taking radians)
tan	Safe trigonometric tangent. Defaults to 0
exp	e (Euler's number) raised to p_1
abs	Absolute / unsigned value
avg	Average of both parameters
floor	Round up and truncate
ceil	Round down and truncate
log_10	log_{10} of p_1
log_e	log_e of p_1
sqrt	Square root
pow2	p_1 to the power 2
pow3	p_1 to the power 3
Math, Binary	
+	Arithmetic addition
-	Arithmetic subtraction
*	Arithmetic multiplication
/	Safe division (0 divisor $\Leftarrow$ 0)
max	Greater of the two parameters
min	Lesser of the two parameters
pow	p_1 to the power p_2
Math, Ternary	
lerp	Linear interpolation of p_3 $\in$[0,1] within [p_1,p_2]
Conditionals	
IfGT	if($p_1 > p_2$) then p_3 else p_4

III. Learning Algorithm & Methodology

A. Aesthetic Features

Before considering the specifics of any classifier models, one should give considerable consideration to the features which will be used to analyse the classified object. Li proposes 25 individual features. Each feature is extracted from each defined window of the image: the full image, top-left, top-right, bottom-left, bottom-right, centre (see Figure 2. This gives 150 individual measures per image.

Fig. 2: Feature Windows



Many other features are available, particularly those for colour frequency and edge points. We will be focusing on recreating the features as described by Li, with exceptions noted as required.

Features f_1 through f_9 include the first 3 movements of the image window colour. Within the HSV colour model, f_1 describes the mean hue values, f_2 describes the standard deviation of the hue values, and f_3 describes the skewness (or, asymmetry) of the hue values [13]. Similarly, f_4-f_6 is found on the saturation channel, and f_7-f_9 on the value / brightness channel.

Benford's Law (or, first-digit law)suggests that sets of naturally collected data should have the digits in their decimal representation follow a known distribution [6]. We will consider, for a feature, the distribution of the Y_{709} lightness channel as it compares with a natural distribution. With the expected frequency of each digit d having (according to Benfor'd Law) $P_{BL}(d)$, and actual frequency $P(d)$, we calculate

$$f_{10} = \sum_{d=1}^{9} |P(d) - P_{BL}(d)|$$

It was noticed that the formula as described in [9] had an oversight in permitting signed increments. As no increment was squared or cubed, the feature should have always been 0. This was likely a typographical error, and has been corrected in our implementation.

Local binary patterns are common texture features which are enjoyed due to their relatively low processing cost. Each pixel compares itself to each of its neighbouring pixels. Based on the relative position between pixel and neighbour, each neighbour is responsible for activating a bit in an 8 bit result. If the difference between the centre and neighbouring pixel exceeds a threshold, the corresponding result bit is active. Our implementation uses a threshold value of $1/256$. The set of local binary patterns for all image pixels is used to create a histogram with 4 bins ([0,63],[64,127],[128,191],[192,255]),. Each bin can be used to find a mean and standard deviation, give the values for features f_{11} to f_{14}, and f_{15} to f_{18} respectively. Our implementation uses a left to right, top down scan for ordering of the bit positions amongst pixel neighbours, with the top left pixel as most significant.

Complexity has been found to frequently relate with many models of aesthetics [4]. As such, we can consider using the entropy of discretized channel values as measures of image complexity. For features f_{19} to f_{21}, we can use the complexity of the hue, saturation, and brightness channels across 360, 100, and 100 bins respectively. The RGB complexity could also be considered if we first quantize the distribution. By reducing each RGB channel to 3 bits, we can produce a complexity meaasure f_{22} with 512 bins. Additionally, we can consider the complexity of Y_{709} lightness in 256 bins for feature f_{23}.

The last two features Li *et al.* propose have been omitted due to implementation complexity and time constraints. The Machado and Cardoso aesthetic model has proven to yield interesting results as a fitness measure. It relies on a ratio of image complexity to processing complexity as determined by JPEG compression and fractal image compression. While an interesting model, it has a complex implementation, and is absent from our system. In a similar vein, another omitted feature measure was the "order" of the image, as determined by fractal compression ratios and times. It is believed that the simplicity of the fixed blue hue fitness scheme should remove the immediate need for these features.

B. Adaptive Learning Model

The adaptive learning model employed for the experiments mimics the one brought forward by Li *et al.* in their work Learning Aesthetic Judgements in Evolutionary Art Systems (Figure 3) [9]. Li was not the first to experiment with learning systems for aesthetic preferences; Much work was done by Machado and Penousal [7], as well as many others [2] [11]. However, the details provided by Li *et al.* allowed for a seemingly easy replication of the system.

The learning system relies on two classifiers: a C4.5 decision tree, and an multilayer perceptron. Both are efficient classifiers capable of handling multiple exclusive classes. Following Li, the developed system makes use of the Waikato Environment for Knowledge Analysis (WEKA) library. The J48 approximation of C4.5 decision trees are employed, as is their implementation of multilayer perceptrons.

The system is initialized after the first time the user has provided an evaluation to an evolved population. Once the user feedback is received, a new decision tree and a new feed-forward neural network are produced and trained. Each rated image provides an expected class (the rating) and a number of extracted features which were computed shortly after phenotype conversion. At each generation (after the first), the user may choose to defer the image ratings to the learning system. The stronger classifier, as determined by classification accuracy, will be used to classify and provide ratings for the current generation of individuals.

One ambiguity in the paper by Li is the lack of details regarding the persistence of the classifiers across generations. As it was stated that the classifiers were produced using WEKA defaults, and insofar that neither the decision tree nor the multilayer perceptron are update-able models within the WEKA library, it would follow that the classifier models must be regenerated each time new data is received. However, Li *et al.* also mention that by the 10th generation of user-provided fitness ratings, 670 images are used for training. One considered possibility is that the previous training examples are stored and used for and later model generation. To asses whether or not the previously provided user classifications will aid in generation of the next classifier instances, a number of experiments will be performed.

We produce classifications in the form of a [1,5] discrete rating, where more appealing images should receive higher values. The learning system could have produced classifications in a continuous range, as the underlying rating and fitness conversion could easily be interpreted. However, the decision to remain with nominal classifications was favoured based on the input method provided to the user. It is quicker for users to rate images along a nominal set than it would be for them to specify larger precision. Both methods could lead to the introduction of noise. The trade-off was considered, and a compromise was reached by expanding the possible classifications to 5 from Li's 3.

Li *et al.* were able to seed the learning system with external reference images, at which point the learning system classified each generation without further adaptation. However, this falls out of our current experimental scope.

C. Feature Reduction and Standardization

Further ambiguity of Li's model is found in the details of the feature reduction methods. The difficulty with searches in high-dimensional spaces is reinforced, and with the 150 proposed extracted features, a strong argument is made for dimensionality reduction within the system.

Li had stated that he had employed the WEKA *Ranker* search method, using the entropy-based *InfoGainAttributeEval* function to guide the search. However, at no point does Li mention the final count of attributes used. Further, Li mentions that the perceptron classifier includes a hidden layer, but we can gain no insight into the feature count, as the hidden layer sizes are also absent. Variable numbers of reduced attribute counts will be explored.

The system produced by Li *et al.* does not specify any means of normalization or standardization, though the benefits may be found with them. Some difficulty could arise in finding the proper ranges of each feature, particularly as we will be training on small sets of data. Some exploration into the effectiveness of normalization in this application will be considered.

D. Analysis Methodology

While the intended purpose of the system is to learn the highly subjective aesthetic preferences of the user, a number of concessions will need to be made to provide a more consistent testing environment. In an attempt to remove subjective variance from the users rating, and the consequential noise, a fixed fitness formula (Equation 4) will be used in place of the user's aesthetic fitness. A rating of 1 to 5 would be given based on the evolved image's mean hue, and its distance from a known measure of blue. It was expected that such a simple fitness measure would be easily learned by the adaptive model, and would provide a basis from which additional, more complex fitness models could be evaluated.

Key performance indicators of evolutionary algorithms relate to fitness scores among the evolved individuals. For comparing final performance of the evolutionary art system, we will be comparing the mean fitness of individuals in the last generation which received its rating through a fixed fitness scheme.

Fig. 4: Fixed Fitness Scheme: Blue Mean Hue

$$\begin{aligned} hue_{current} &= f_1 \\ hue_{desired} &= (240/360) \\ dist &\cong (Hue_{current} - Hue_{desired}) \mod 1 \\ rating &= [(dist * 8) + 1] \end{aligned}$$

While the mean fitness provides a good measure of performance pertaining to the users final goal, the specific experiments performed also aim to improve the classification abilities of the adaptive learning system employed to simulate a user's aesthetic preferences. While the two measures are expected to be related, the periodic use of the fixed fitness scheme may provide too much direct guidance to the evolutionary model, instead of an indirect guidance via the classifier. Further, as the adaptive classification system uses both decision trees and artificial neural networks, the fitness of each individual classifier can not be displayed through the single fitness measure. To these ends, an additional metric will need to be explored in conjunction with fitness.

To gauge performance of individual classifiers, a number of other measures are tracked per classifier for each experiment. We have available the amount of correct and incorrect classifications, as well as a full confusion matrix between the 5 classes. It should be noted that reliance on the correct amount of classifications may not be an adequate measure for all cases. A classifier which seems to perform well but only ever assigns one classification may not be showing a true reflection of the classifier's performance; the samples could be evolved to have a higher distribution of that class among them through external influence. A number of the other performance measures provided through confusion matrix aggregates may need to be examined.

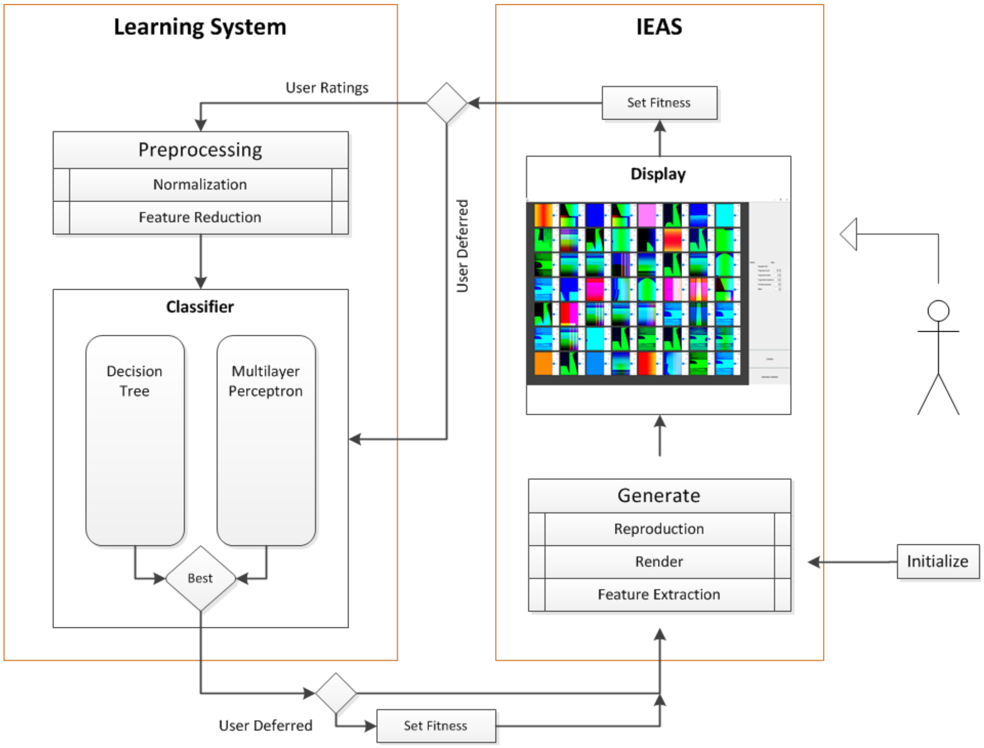
One concern is the lack of separate training and testing data sets. As the samples are expected to be provided through user feedback, the GP population size has been kept low to maintain a consistent environment where user fatigue would be a significant factor. However, with the low amount of samples, it is not feasible to partition the training data in that circumstance, as there is a high likelihood of each sample to carry a meaningful decision.

IV. Experiments

The paper by Li *et al.* alleges that the default parameters provided within the WEKA library were maintained for their tests. As some changes may have occurred between versions of the library, the key settings for each classifier are outlined in Tables III and IV.

For all experiments, a reduced sample size of 5 runs per experimentation configuration is used. While a larger size would be ideal, the time needed to run 30 generations, coupled with minimal but required interaction for log archival

Fig. 3: The Learning System Adaptive Classifier Model



places limits on the feasibility of larger run sizes.

TABLE III: Learning System C4.5 Tree Configuration

Parameter	Value
confidenceFactor	0.25
minNumObj	2
numFolds	3
reducedErrorPruning	False
saveInstanceData	False
seed	1
subtreeRaising	True

TABLE IV: Learning System ANN Configuration

Parameter	Value
Hidden Layer Topology	30,15
Activation Function	Sigmoid
Learning Method	Backpropagation
Learning Rate	0.3
Momentum Rate	0.2
Epochs	500

TABLE V: Learning System Common Baseline Configuration

Parameter	Value
Generations	30
Population Size	56
Normalize Inputs	True
Final Feature Count	15
Reuse Old Data	False
Fixed Fitness Scheme	Full-Image Blue Mean Hue
Fixed to Classifier Fitness Pattern	1 - 3

As we are using a predefined fitness scheme to remove subjectivity from the classification performance measures, we will also need to set intervals which determines whether each generation makes use of the learning system or the fixed fitness scheme. This should simulate a user providing intermittent feedback alongside their deferral to the classifiers. The first set of experiments will determine the effects of extending the number of generations since simulated user ratings are provided. We will begin with a baseline, where every generation receives user feedback, and then examine final generation fitness and classifier accuracy when user feedback is provided every other round, and every third round.

Further experiments will evaluate changes to the feed-forward neural network classifier used by the system. While adjustments may be made to the C4.5 decision tree classifier, it is outside the scope of this report.

A second group of experiments will examine the effects of adjusting the number of extracted features and the hidden layer topologies. From the baseline, which used a 15-30-10-5 topology, we will also examine 15-15-5, and 8-8-5 topologies. While Li did not specify the size of the hidden layers used in their report, they had mentioned that a single layer was used. Given that the default hidden layer size is in terms of the number of attributes and classes, and that the reduced number of attributes was not provided, an estimate of 25 ($(25 + 3)/2 = 14$ hidden nodes) and 15 ($(15 + 3)/2 = 9$ hidden nodes) was speculated when considered topology options. A 25-15-5 topology was omitted, as the simplistic blue hue fitness scheme can be produced from a single measure, and reinforced by up to 5 additional measures.

A third experiment will evaluate the effectiveness of retaining training samples from previous generations. The parameters for the second experiment will be reused with single exception of a training sample retention flag.

While previous experiments aimed to keep the same learning and momentum rates as used by Li *et al.*, the lack of progress leads us to consider alternate configurations of these rates. Using the best found configuration from the previous experiments, we will evaluate the effects of adjusted learning and momentum rates. Training samples from previous generations will be used on a 15-30-10-5 network. Based on previous work with artificial neural networks, intermediate values of learning rate and momentum rate appeared to perform well. We will consider learning rates of 0.5 and 0.8, and momentum rates of 0.2 and 0.4, in comparison to the baseline measure.

V. Analysis

We started with a baseline that used a fixed fitness measure to ensure that the GP system had the expressiveness to generate the type of images required for the experiment. For the first set of tests, the fixed fitness measure was a simple mean colour distance. We ranked the images based on the full image mean hue's distance from RGB absolute blue.

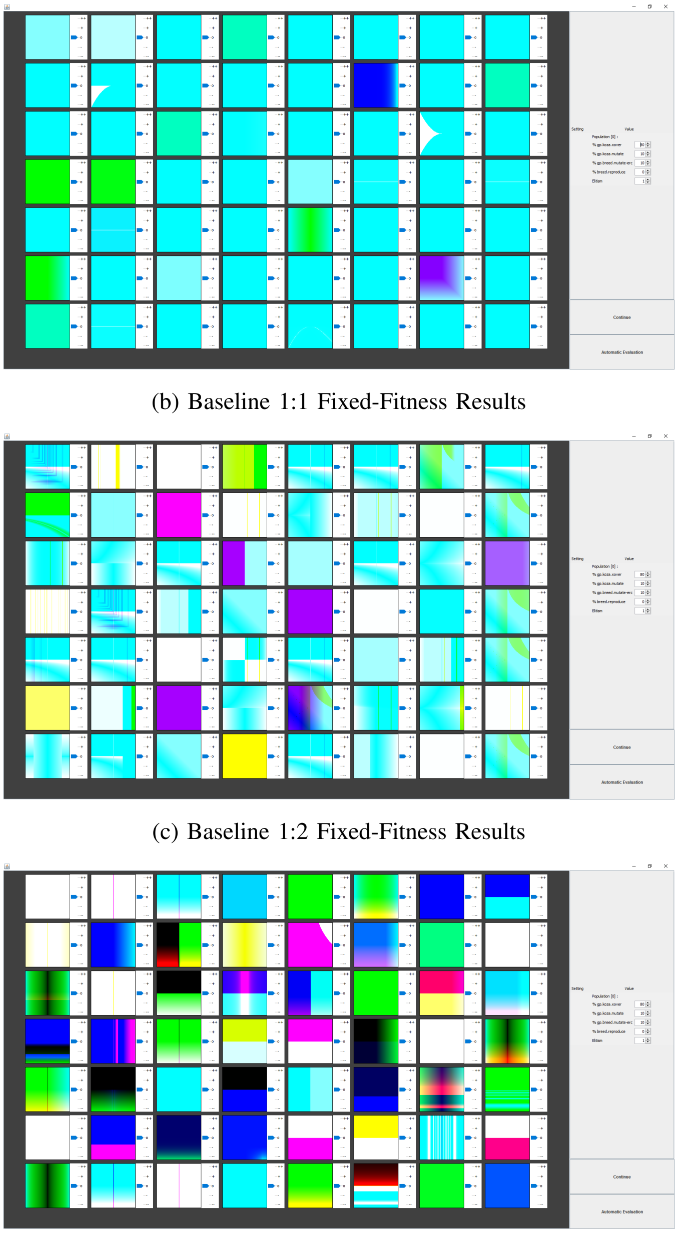
In the performance graphs include (such as Figures 6, 7, and 8), the bold smoothed lines represent a default local polynomial regression fitting from R among the data points. Where the bold red line shows mean population fitness, green shows mean decision tree classification accuracy, and blue shows mean perceptron classification accuracy.

We can clearly see in Figure 6 that as we increase the frequency of using the learning system for fitness assignment, we have a final population which is less fit in terms of our experimental fixed fitness metric. This would seem to indicate that the classifications from the learning system are rather poor, and that we are not favouring those individuals with the ideal hues as strongly as we could be. The more preferable individuals found in the final generation may just have been carried over from the last generation with fixed fitness through coincidence.

This is further indicated by suspicious mean fitness values. On generations rated by the classifier, mean population fitness seems to consistently return to values produced from a single rating value. That is, it seems likely that the classifier is only returning a single classification, regardless of features.

With further experiments, the alternation between 1 generation rated through a fixed fitness scheme, and two generations rated through the adaptive classifier is maintained. This appeared to provide both a minimal amount of correct selective pressure from the fixed fitness scheme, and sufficient examples on which the classifier could be trained. A smaller ratio would not permit stronger conclusions about the classifiers, as it would be more likely for performance gains to come from the accuracy of the fixed fitness scheme.

Fig. 5: Experimenting with Fixed-Fitness Frequency



(a) Baseline 1:0 Fixed-Fitness Results

Setting Value
Population [0] :
% gp.koza.xover 80
% gp.koza.mutate 10
% gp.breed.mutate-erc 10
% breed.reproduce 0
Elitism 1

(b) Baseline 1:1 Fixed-Fitness Results

Setting Value
Population [0] :
% gp.koza.xover 80
% gp.koza.mutate 10
% gp.breed.mutate-erc 10
% breed.reproduce 0
Elitism 1

(c) Baseline 1:2 Fixed-Fitness Results

Setting Value
Population [0] :
% gp.koza.xover 80
% gp.koza.mutate 10
% gp.breed.mutate-erc 10
% breed.reproduce 0
Elitism 1

In varying the network topologies, from hidden layer changes and dimensionality adjustments, no immediate improvements were noted. By observing Figure 7, it might appear that the 15-15-5 topology performed worst, with the baseline 15-30-10-5 doing well, and similarly for 8-8-5. Seeing that the 8-8-5 topology didn't do drastically worse than the baseline, and perhaps even showing slight improvement, one might suggest overfitting with larger topologies. However, we might not allot serious merit to this idea, as the significantly larger 15-30-10-5 topology showed comparable performance. Ultimately, it is disheartening that any variation seen in performance does not carry heavy statistical significance.

It was hoped that by permitting the reuse of older generation's training cases, we could increase the total sample size

Fig. 6: Performance of Fixed-Fitness Frequency Experiments

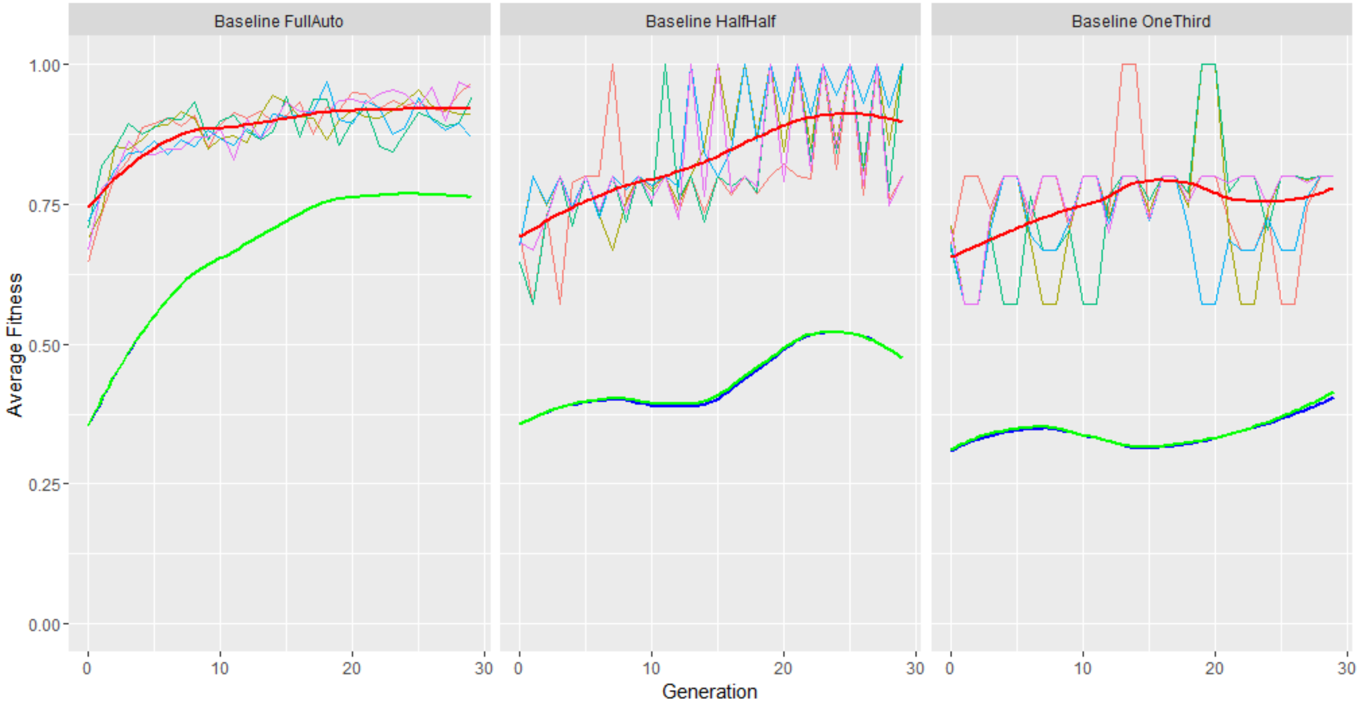


Fig. 7: Performance of Topology Variations

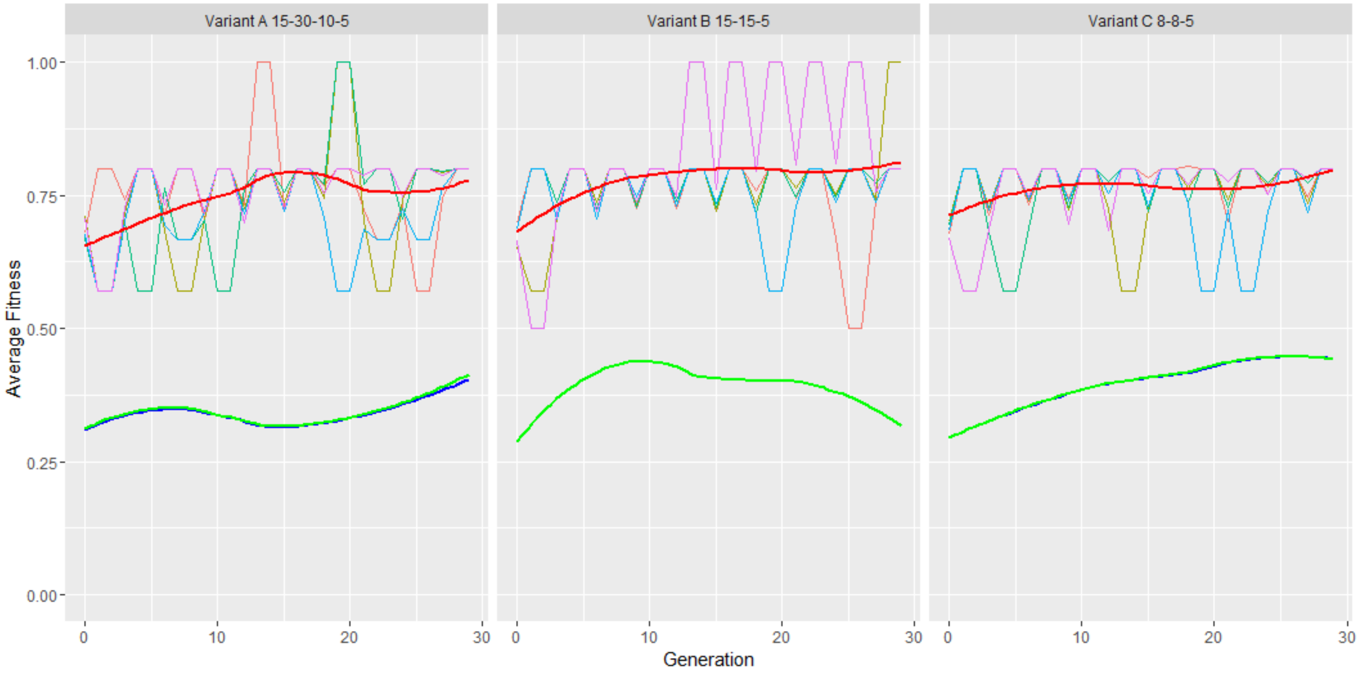
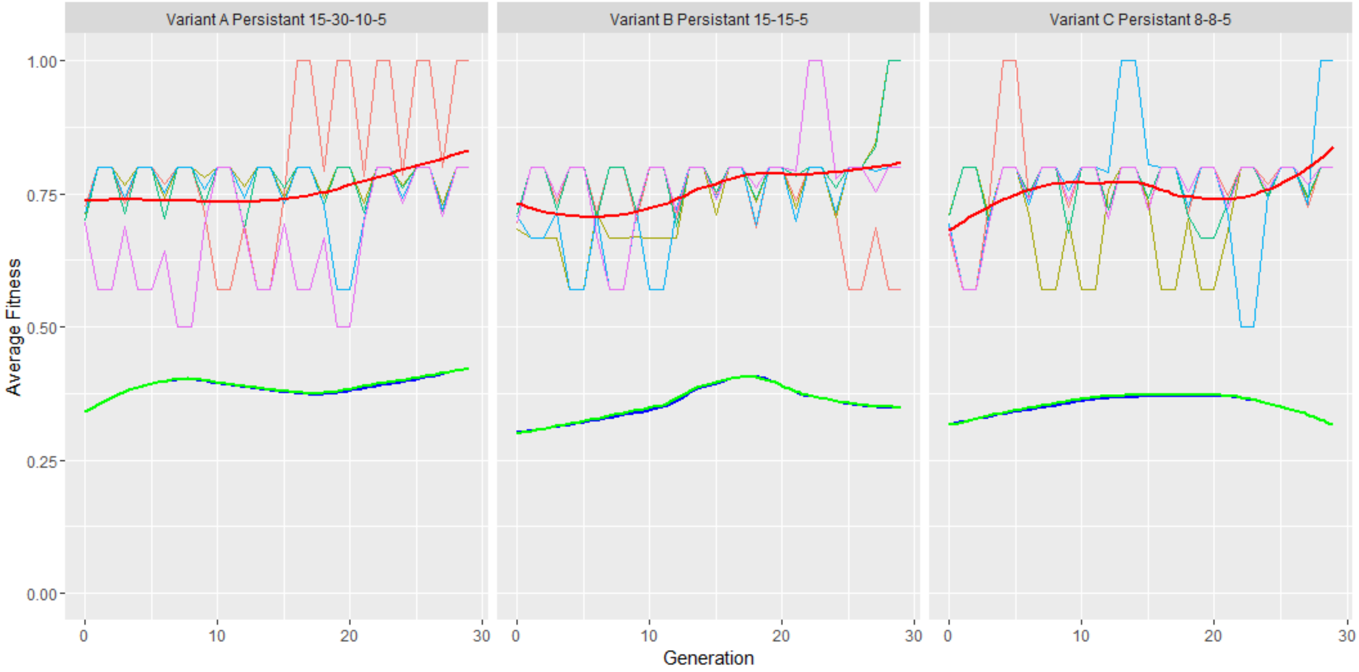


Fig. 8: Performance of Topology Variations, Training Samples Reused



for each newly created classifier. With more training samples, the ideal case would also permit for greater classification accuracy. Some concerns held beforehand were that a user would rate images relative among those presented in a single generation. With the fixed fitness scheme providing an absolute rating regardless of the best and worst individuals seen in a given population, it was expected that sample reuse would be strictly beneficial.

The largest effects of this adjusted sample reuse scheme should be apparent when ranking the later evolved generations. Unfortunately, we once again see little change from the previous experiment set. There is still extremely heavy tenancies to rank all images in a generation uniformly. No statistically significant performance changes could be found.

One mildly interesting note, is that the mean classification accuracy appeared to show similar curvatures for a given configuration both with and without sample reuse. The actual significance of this is likely low, as decision tree classification accuracy shows similar effects despite not being directly adjusted (though, the resultant effect on mean fitness would less directly influence the decision tree performance).

As the configuration stated in the report by Li *et al.* was not as effective as hoped, a final attempt at improving classification ability through learning and momentum rate adjustments was attempted and outlined in Figure 9. While a few of the runs did appear to show some convergence, and completed with an array of individuals which had a generally blue hue, no conclusive improvements were found. The frequency and spread of the improved individuals did not appear to have any substantial significance, and was likely produced purely through fortunate coincidence.

Seeing that we were unable to get any reasonable classification of straight blue images, it seemed unlikely that more complex fixed fitness schemes would prove more successful.

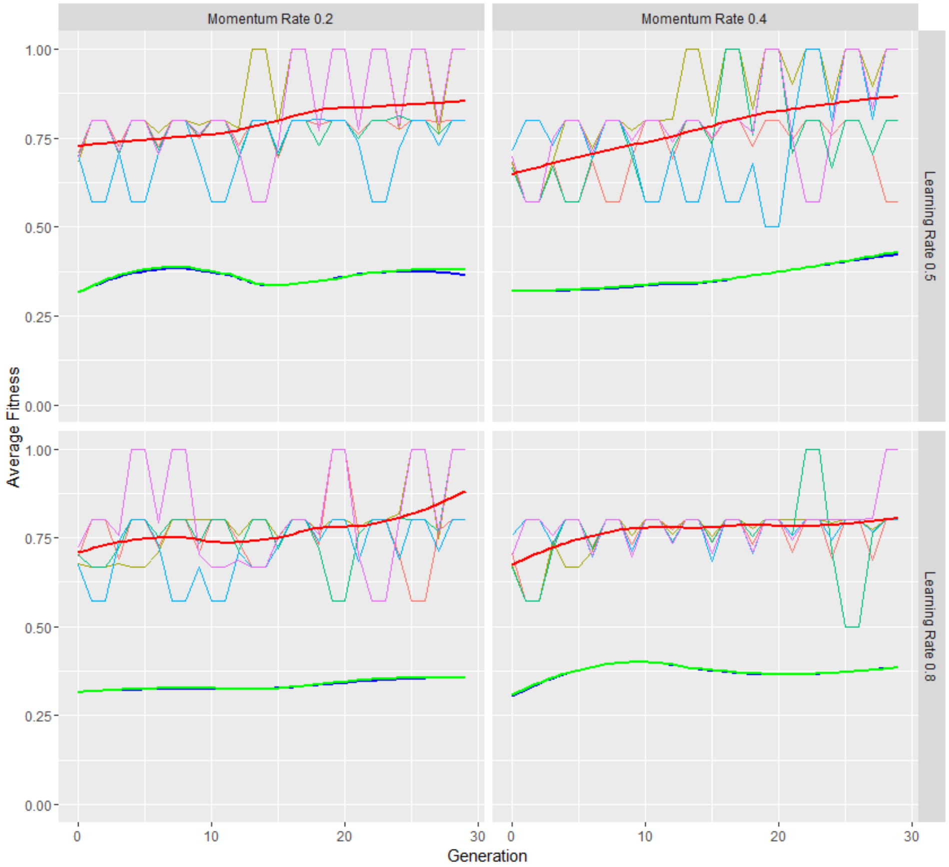
VI. Conclusion and Future Work

Despite any measured performance seen in models which reuse previous generations' training data, such an approach may require additional consideration. While such a method worked will with our fixed fitness measure, its use in conjunction with a user may lead to early convergence. A user may very well change what attributes they seek mid process. A model which retains too much early generation data may not be able to adapt. Further, as ratings are typically relative among only the individuals presented at any given time, old training data will be inherently noisy at best, as what was an ideal candidate in one generation may be completely inadequate in a later generation. One potential workaround to the issue of old data may be to implement an ageing system, where data is only retained for a number of generations.

Neither adjustments to topologies nor learning rates appeared to positively affect classification performance in any significant manner.

In consideration of the difficulty found in classifying images based on extremely simple (single-feature) fitness criteria, a full multi-user experiment like the one discussed by Li may need to be held until the configurations for the learning system accuracy is matured. Much further experimentation in more controlled fixtures and review of implementation details may be required to achieve the consistent 80%+ classification ability reported by Li.

Fig. 9: Performance of Rate Variations



References

[1] Karl Sims. Artificial evolution for computer graphics. *ACM Computer Graphics*, 25(4):319–328, July 1991. SIGGRAPH '91 Proceedings.

[2] Shumeet Baluja, Dean Pomerleau, and Todd Jochem. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6(2 and 3):325–354, 1994.

[3] Tom M Mitchell et al. Machine learning, 1997.

[4] Penousal Machado and Amílcar Cardoso. Computing aesthetics. In *Advances in artificial intelligence*, pages 219–228. Springer, 1998.

[5] Andrea L. Wiens and Brian J. Ross. Gentropy: Evolutionary 2D texture generation. In Darrell Whitley, editor, *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*, pages 418–424, Las Vegas, Nevada, USA, 8 July 2000.

[6] E Acebo and Mateu Sbert. Benford's law for natural and synthetic images. In *Proceedings of the First Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 169–176. Eurographics Association, 2005.

[7] Penousal Machado, Juan Romero, Amílcar Cardoso, and Antonino Santos. Partially interactive evolutionary artists. *New Generation Computing*, 23(2):143–155, 2005.

[8] Simon Colton, Michael Cook, and Azalea Raad. Ludic considerations of tablet-based evo-art. In *Applications of Evolutionary Computation*, pages 223–233. Springer, 2011.

[9] Yang Li, Changjun Hu, Leandro L. Minku, and Haolei Zuo. Learning aesthetic judgements in evolutionary art systems. *Genetic Programming and Evolvable Machines*, 14(3):315–337, September 2013. Special issue on biologically inspired music, sound, art and design.

[10] Jinhong Zhang, Rasmus Taarnby, Antonios Liapis, and Sebastian Risi. Drawcompileevolve: Sparking interactive evolutionary art with human creations. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 261–273. Springer, 2015.

[11] João Correia, Penousal Machado, Juan Romero, and Adrián Carballal. *Feature selection and novelty in computational aesthetics*. Springer, 2013.

[12] John R Koza. *Genetic programming II, automatic discovery of reusable subprograms*. MIT Press, Cambridge, MA, 1992.

[13] National Institute of Standards, Technology (US), Carrol Croarkin, Paul Tobias, and Chelli Zey. *Engineering statistics handbook*. The Institute, 2001.

[14] Juan J Romero and Penousal Machado. *The art of artificial evolution: a handbook on evolutionary art and music*. Springer Science & Business Media, 2007.