Statistical Image Analysis for Image Evolution

Elham Salimi

Submitted in partial fulfilment

of the requirements for the degree of

Master of Science

Department of Computer Science

Brock University

St. Catharines, Ontario

Abstract

This thesis is focused on using genetic programming to evolve images based on lightweight features extracted from a given target image. The main motivation of this thesis is research by Lombardi *et al.* in which an image retrieval system is developed based on lightweight statistical features of images for comparing and classifying them in painting style categories; primarily based on color matching. In this thesis, automatic fitness scoring of variations of up to 17 lightweight image features using many-objective fitness evaluation was used to evolve textures. Evolved results were shown to have similar color characteristics to target images. Although a human survey was conducted to confirm those results, it was inconclusive.

Acknowledgements

I would like to thank Dr. Brian J. Ross for guiding me in this path patiently and then I would like to thank my parents and my sister, my friends, and faculty of department of computer science for their great support**.**

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

Evolutionary art uses evolutionary algorithms to create artistic images [1]. A big advantage of using evolutionary algorithms in art is their ability to generate creative and innovative images. Moreover, multiple solutions can be evolved for a given problem using evolutionary algorithms [1].

Researchers have developed intelligent evolutionary systems to generate solutions to problems that are human-competitive. In other words, recently evolutionary systems are producing solutions that are comparable to human-produced results. However, human-competitive evolutionary art is not as advanced. Original systems used user-interactive image evaluation during the evolution. Automated evaluation criterion is the biggest difference between earlier evolutionary systems and contemporary ones [2]. New evolutionary art systems are using aesthetic models and computer vision algorithms to automate image evaluation [3].

Genetic programming is used in this research to evolve 2D images based on lightweight image features. Our research uses a paper by Lombardi *et al.* [4], which uses lightweight image features for image retrieval. They use to them compare, classify, and retrieve images based on the style of different painters. They have tested their system on several famous paintings and reported promising results. We wish to use the same technology for automatic image evolution. To our knowledge, there is no previous attempt to evolve textures based on lightweight features of images.

In this research, we propose an evolutionary system to evolve procedural textures using lightweight image analysis. Our system uses a set of extracted features from images as fitness measurements on evolved textures. These features are those used in Lombardi *et al*'s. system [4]. The goal is to evolve images whose features are similar to a target image as measured by the lightweight image characteristics.

We are interested to see if these feature sets can be effective in automatically evolving images. These lightweight features might also be beneficial for genetic programming system because they are by definition fast to compute.

In our evolutionary system, we have used many-objectivity in our fitness analysis to handle the high number of objectives. Experience shows that Pareto ranking is not an effective technique when the number of objectives goes above 4. In this research there are up to 18 simultaneous objectives to be tested, and therefore many-objective strategies seem necessary.

1.1 Thesis Structure

This thesis is structured as follows. Chapter 2 contains some background information about genetic programming and many-objectivity. Chapter 3 presents a literature review in evolutionary art and image retrieval techniques. Chapter 4 discusses Lombardi *et al's*. lightweight image features. In chapter 5, the system architecture including algorithms, parameters, and evaluation functions are explained. Chapter 6 describes the results of experiments and performance analysis of the system. Chapter 7 presents human survey results. Chapter 8 concludes the thesis and discusses future work.

Chapter 2

Background

2.1 Genetic Programming

Evolutionary algorithms (EA) use the idea of Darwinian natural selection to solve a problem [5]. Genetic programming (GP) is a technique in evolutionary computation (EC) which breeds working program code for problem solving [6].

2.1.1 GP Representation

The evolved computer programs are denoted by tree structures. In a GP tree, the internal nodes of the tree are function nodes and the leaf nodes are the terminals. The set of terminals and functions is called the GP language. Functions can be arithmetic functions, mathematical functions, logical, or conditional statements like if-then-else arguments. Terminals can be variables or constant values such as image pixel coordinates (x, y). As an example, in the statement X+Y, X and Y are two terminals that are added together using the arithmetic “add” function. Figure 1 shows the tree for this statement.

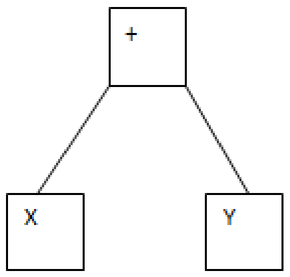


Figure 1: GP expression tree.

2.2 Genetic Programming Algorithm

In a GP system, the first step would be to initialize the first generation with a set of randomly generated individuals using a composition of functions and terminals. Then each program code is assigned a fitness score. After creation of a new population, in order to find a good solution for a problem GP iterates over the previous steps by applying crossover and mutation operations on the computer program. The algorithm continues until the termination criterion is met. The pseudo code in Table 1 shows the functionality of a GP algorithm [7].

Table 1: Pseudo code for genetic programming.

Genetic Programming Algorithm

```
P -> Population size
R -> Maximum number of runs
Run 0;
    for (i:1 to R) {
        Randomly  initialized population ;
        Generation = 0;
        While (! Termination Condition ) {
            Evaluate fitness of each individual;
            for (j:1 to P) {
                select an operator:  op;
                if (op == unary) {
                    select an individual based on its fitness;
                    perform reproduction with probability pr and copy
                    offsprings into a new population;
                    }
                else if (op == binary) {
                    Select two individuals based on their fitness scores;
                    Perform cross over with probability pc;
                    }
                J++;
                Insert two offsprings into a new population;
                }
    }
```

2.3 Fitness Evaluation

Since the initial randomly generated individuals are probably inadequate to solve a problem, there should be an evaluation strategy to evaluate the individuals of each generation in the problem space. The fitness function scores individuals of a problem space considering how good they can perform in solving the problem. Individuals with higher scores will have higher chance to be selected for the reproduction in the next step [6].

2.4 Crossover and Mutation Operators

In a crossover operation, two randomly selected parents, that are usually of different shapes and sizes, exchange their subtrees from the crossover point. These points are chosen randomly in the first parent and the second parent. During the crossover operation, the subtree of the first parent rooted at the crossover point will be replaced by the subtree of the second parent in the crossover rooted point. Thus, the children will have a combination of the replaced subtrees of their parents. This operation is predominant and is used over 90 percent of the time in the parameter set of GP. Figure 2 shows how cross over works on parents and Figure 3, shows the resulted children using crossover operator [7].

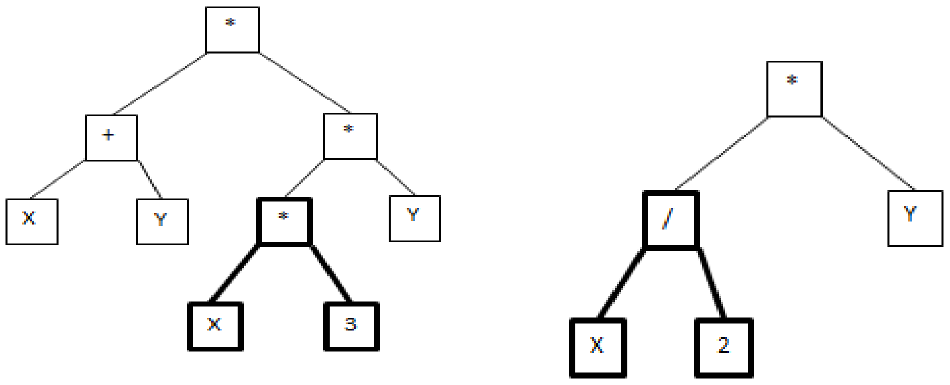


Figure 2: Crossover operation on parents.

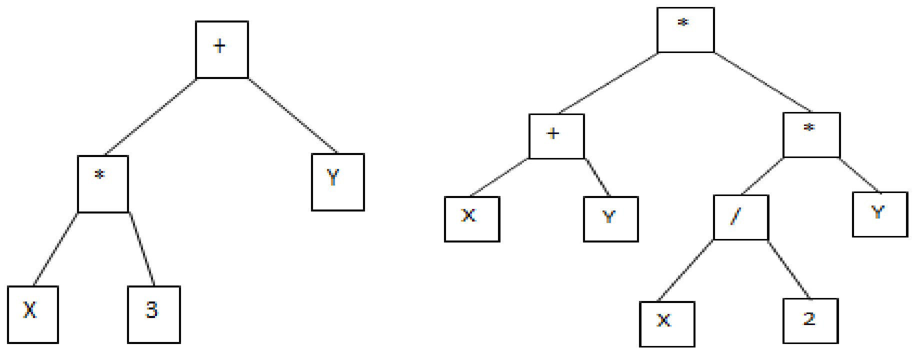


Figure 3: Resulted children after the crossover operation.

While the crossover operation works on two selected parents, the mutation operator functions on a single parent. In this process, the mutation point is randomly selected on a parent and the subtree rooted at this point will be replaced with random subtree, which normally follows the rules of the initially grown individuals. Since this operator is not the major operator of the evolutionary process, it has a lower percentage of use compared to crossover (about 10 percent during each generation) [7]. Figure 4 shows how the mutation operation effects on individuals.

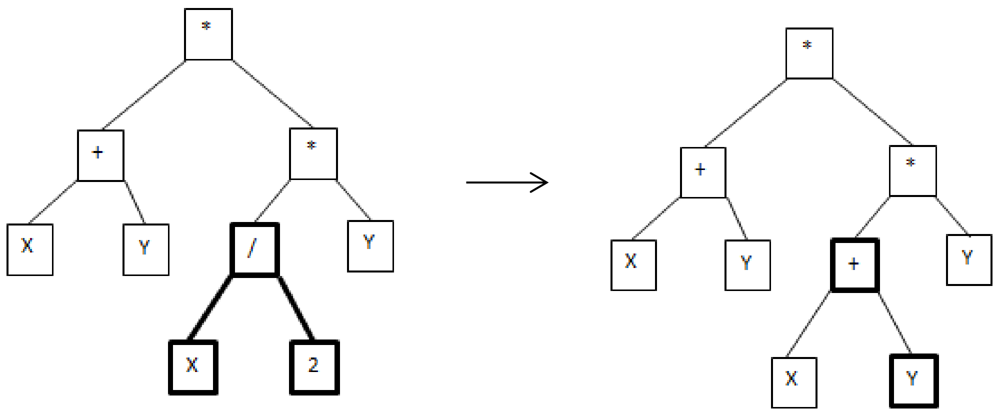


Figure 4: The original individual before mutation (right) and the mutated individual (left).

2.5 Selection Strategy

Among the existing fitness-based selection strategies, which GP systems use to select good individuals for reproduction, tournament selection is the most commonly used. In the tournament selection strategy, K individuals are randomly selected from the population. The higher the k value, the higher the selection pressure used to select individuals. Individuals with the highest fitness in the selected set are the winners of the tournament and will be selected as parents for reproduction. Therefore, for the crossover operation, tournament needs to be applied twice to pick two parents to breed the next generation [8].

2.6 Multi-objectivity and Many-objectivity

Depending on the problem space, the number of objectives may vary in different problems. In problems where the number of objectives goes beyond only one single objective, the problem is called multi-objective. In some cases where the number of objectives goes beyond 5, the problem is called many-objective [9] [10].

The goal of multi-objectivity is to consider all the objectives of the problem space to find an optimal solution to solve the problem. There are ways to handle a multiobjective evaluation, which are explained in this section.

2.6.1 Weighted Sum

A common method to evaluate a number of objectives in single-objective fitness criteria is the weighted sum. In this method, each objective value will be given a weight, which points to impact of that objective on the problem space. Formula 2.1 is an equation showing how it works.

$$FitnessScore = f1\_W1 + f2\_W2 + ::: \qquad (2.1)$$

where f_n are fitness scores and W_n are weights (n ≥ 1). The score calculated by this formula will be assigned to the individual as a fitness score. This interprets the problem as a single objective problem.

2.6.2 Pareto Ranking

Pareto Ranking ranks each solution in the population individually considering the notion of dominance, and keeps the scores independent from other individuals [11]. If individual X is as good as individual Y in different dimensions and X has been scored higher in at least one dimension, then X dominates Y. This concept can be formulated as follows:

X dominates Y:

$$\forall i : Xi >= Yi \wedge \exists i : Xi > Yi \qquad (2.2)$$

where there are i objectives (i ≥ 1).

Those individuals which have not been dominated by other individuals in the population will be scored as rank 1. Then the not dominated individuals in the rest of the population are given a rank of 2. The process continues until all individuals are scored. At the end of a run, the solutions are the ones that are considered as the potential solutions for the problem. Experience shows that Pareto is ineffective in high dimensions problems where the number of objectives goes beyond 4.

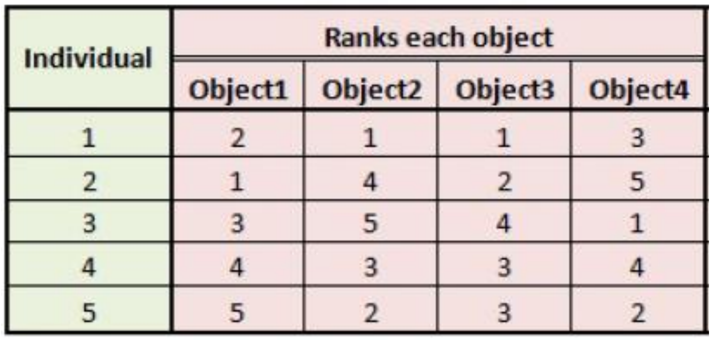
2.6.3 Sum of Ranks

Another method, which has been used in previous research in evaluation criteria in many-objective problems, is sum of ranks. In this method, the objective scores for all dimensions of a problem are calculated per individual. These scores are converted to ranks where rank 1 is the best score in that objective in the population. Formula 2.3 shows how the fitness score is calculated [2] [12].

$$fitnessscore = \sum_{i=1}^{k} ri * wi \qquad (2.3)$$

where k is the number of individuals, r is the rank given to that individual and w is the weight of that objective for the ranked individual.

Similar to sum of ranks, normalized sum of ranks takes the ranks of each individual and divides all the ranks by the maximum of obtained ranks among all individuals for each objective. At the end, the normalized ranks will be added up together for each objective per individual. Figure 5 [13] shows the scoring for 4 objectives and 5 individuals. Here, sum of ranks is calculated by adding objective ranks per individual, and by dividing each rank by the maximum rank of each objective for that individual normalized sum of ranks is calculated as shown in this figure. For example, individual 1 has got the following ranks for all of the problem objectives: 2,1,1,3. Assuming that all of these objectives have equal weights, sum of ranks is calculated as 2+1+1+3 = 7. To find the normalized sum of ranks, by looking at ranks of all individuals and for all objectives in the table, the maximum obtained ranks by individuals is 5, 5, 4, 5. Therefore, normalized sum of rank as explained earlier will be 1.4



Individual	Ranks each object			
	Object1	Object2	Object3	Object4
1	2	1	1	3
2	1	4	2	5
3	3	5	4	1
4	4	3	3	4
5	5	2	3	2

Figure 5 : Ranks and scores for individuals per objective.

Chapter 3

Literature Review

Recently, research in computer vision, pattern recognition, and image processing has developed numerous computer methods for addressing problems in art. In some cases, computers can analyze textures very accuratly, and researchers are developing different methods to take advantage of computers where more precision is required. Therefore, different techniques are being developed to analyze digital images specifically paintings and drawings [14].

On the other hand, there is a rapid increase in the size of digital image databases, and there needs to be ways to make accessing, organizing, searching, and retrieving the stored images easier. Several image classification and retrieval systems have been proposed since the early 1970s for this purpose. In this chapter, we briefly introduce some research on image classification and image retrieval using different techniques and then we explain different evolutionary art approaches in texture generation.

3.1 Image Classification and Image Retrieval

Text-based image retrieval systems were popular in early research in the 1970s. The way text-based systems work is by annotating images by text and then using a text-based database management system (DBMS) to access and retrieve images. Growth in the size of storage space, nonlinear distribution of image files, and high dimensional feature space created difficulties for text-based image retrieval systems. The amount of work to be handled for annotating images was significant, and differences between image contents and human perception could cause wrong annotations and, as a result, classification errors [15] [16].

To tackle problems caused by this issue, computer vision techniques for classifying images were introduced. Image classification is the task of categorizing images based on extracted visual contents of those images. Classification techniques can be supervised, in which the classification process uses training sets provided for the system. It can also be unsupervised, where no training set is provided at the beginning, and the system examines large data sets to learn about different groups and classes that can be created for the existing data [17]. Recently most of the image classification and retrieval systems are content-based. All content-based systems have three basic parts: (i) preprocessing, which involves feature extraction; (ii) database multidimensional indexing; and (iii) designing the retrieval system. In our research, we are not focusing on designing an image retrieval system. Instead, we will examine different feature extraction techniques that these systems use.

Two types of features can be extracted from images in image retrieval systems. Text-based features that could be considered as part of the image contents. Visual features are also important, and features such as color, texture, shape, color moment, color sets, and color histogram are widely used in image retrieval systems [15].

Color moment was introduced by Stricker and Orengo [18], and it calculates color similarities between images based on probability of color distribution by characterizing

HSV color model by three moments (mean, standard deviation, and skewness). Where mean is the average color value, standard deviation is the square root of variance of color distribution, and skewness is a measure of asymmetry in color distribution in the image. Color set, was proposed by Smith and Chang [19]. They found color bins used in an image based on HSV color model could be as effective as color histogram. The color histogram represents color distribution in HSV or RGB color models.

The second group of features is texture, which deals with visual patterns in all surfaces, including trees, bricks, fabrics etc. It provides important information about the structure of each surface other than colors and color intensities. One famous technique in texture representation is wavelet transform. Much research since the early 1990s has been conducted based on variations of wavelet transforms [20] [21].

The third group of features is shape related features. Shape representations deal with the outer boundary of the shape or the entire shape region in the image. Two famous examples are Fourier descriptor, which uses Fourier transformed boundary as the shape feature, and moment invariants, which use region-based moments as shape features [22] [15].

3.1.1 Image processing for artist identification

A practical example of image processing techniques is artist identification based on classification of paintings of different painters. Richard Johnson *et al.* [23] have done research on classification of paintings using analysis of brush stork characteristics in each painting. They have done their experiment on a dataset of high-resolution grayscale scans of paintings provided by Van Gogh and Kroller Muller museum. Using several wavelet transforms, they could classify images based on captured local features of images in a wider range of scale and orientations. They also compared textures based on features extracted from patches of the textures (computing feature distances in patches). Wavelet based features and geometric characteristics of strokes are the two types of features they have studied. Wavelet features are based on orthonormal transform of pixel intensities (texture-based) while strokes are higher-level features that require edge detection techniques to trace contours of strokes and the computation of geometrical features of strokes such as length, orientation, and average curvature of each stroke line. Using a stochastic statistical model, they could compare two feature sets. Results of their experiment showed that wavelet features were able to compare any paintings against any subset of paintings with any sort of features extracted.

3.1.2 VisualSEEK

In research by John smith and Shih-Fu Chang [24], a content-based image retrieval system has been introduced that looks for feature properties such as color, and spatial layout of the image such as size, location, and relationships to other regions. In content-based image retrieval systems, user's queries are sent to a database of stored images and the system should find the best matches to the requested image among the stored images. They have specifically focused on an important similarity criterion, which content-based techniques do not consider: spatial information and spatial relationship. Using this technique, in addition to comparing images based on their regions, the system gives the control to the user to select regions.

3.1.3 QBIC (Quary Based Image Content)

This image retrieval system was developed by IBM Almadan Research Center and is considered as the first commercial content-based image retrieval system [15] [21]. The technical framework of QBIC has influenced many retrieval systems. Supporting high dimensional indexing and receiving queries based on image samples, user-created sketches, and given color or patterns are the most important technical features in this system.

3.1.4 Other commercial systems

Research groups have developed many other systems in this area such as Virage [15] [21]. This system is a commercial content-based image retrieval system, which receives weighted visual features as queries sent by users. Retrievalware, is another system which takes advantage of neural networks as the classification method [15] [21]. It accepts query features such as color, shape, texture, color brightness or a combination of these features. Netra, a system developed at UCSB that takes colors, textures, shapes, and spatial location information to look for similar images of the chosen segment of an image in the database [15] [21]. MARS (multimedia analysis and retrieval system), is different from other image retrieval systems [15] [21]. Because it mostly cares about finding a meaningful and adaptable architecture of storing images so they can be retrieved based on applications. This system is designed by university of Illinois. Other valuable research in this area have been conducted that worth mentioning for example a research by Justing Johnson *et al.* [25]. In their proposed method, scene graphs take semantic details of textures by encoding objects, their attributes, and the relationship between them and represent images with those graphs [25].

3.1.5 Lombardi *et al.*'s Image Retrieval

Lombardi *et al*. [4] designed an image retrieval system for classifying images stored in a small dataset, in which a classification model is responsible for classifying images relevant to the artist's style as well as determining which historical period the artwork belongs. This framework calculates the local features of the images after the image has been saved to the database. There are two preliminary feature styles defined for the model. The first feature characteristics is based on the image colors and is called palette feature set. It measures the total number of unique RGB triples found in each image. The second set is based on spatial statistical attributes of the image like the max, min, standard deviation etc. It is called the canvas features set. In our research these features are categorized in 2 sets. The total number of features calculated for each image in the first feature set is 16 palette (unique triples) and canvas features (RGB). In spite of the previous feature set, the second preliminary feature set employs HSV model to bring the focus on defining color features of each image. These features including intensity (level of brightness normally measured on a gray scale image), the color entropy (the degree of color frequency changes in an image) and the edge characteristics of the image (to identify the lines in a painting), totaling 18 number of features in this set per image.

A sophisticated graphical user interface lets the user interact with two different windows: comparison and classification. In the comparison window, the user is able to modify values for any of the mentioned features, and compare two images with each other. In the classification window, as well as modifying the features, the user can compare all the existing images with a targeted image. At the end of each test, the user is prompted for giving a descriptive feedback to the system about his or her decision. The results for different tests have shown an accuracy of a range between 71 to 94%, and an overall accuracy of 56% that shows promise in this problem domain.

3.2 Evolutionary Art

Besides computer vision techniques in arts and graphics, research has used evolutionary computation to address problems related to art. Karl Sims introduced the concept of using evolutionary computation to create textures or 3D structures [26]. He came up with the idea of how to use Lisp symbolic expressions to create 3D structures and to generate textures. In his paper, he defines all the basic concepts related to genetic algorithm and proves this methodology can be used to generate interesting art.

3.2.1 Gentropy

The second inspirational research [27], named Gentropy, aims at generating 2D procedural textures using a combination of mathematical functions and texture features. Gentropy evolves the textures through an unsupervised approach and without any further involvement of the user. The evolutionary process starts by feeding Gentropy with one or more target images as well as indicating fitness goals for each texture, such as color and shape. Gentropy evolves a texture at the end of each run. This solution texture shows features of the target textures, but is not identical.

Gentropy’s texture language includes noise functions (noise, turb, turbflow, cloud, and marble), warps and tiling functions, and automatically defined iterations. The fitness evaluation consists of a package including different image analysis tools for comparing the generated texture with a target image. This package consists of image comparison with feature tests (color direct, color histogram, color histogram quadratic, wavelet, and smoothness histogram). Image comparisons use color quantization for shape and color comparison in images based on the image features. The feature test set is listed in Table 2.

Table 2: Color matching tests.

Test	Description
Color Direct (CDIR)	Matches color similarities pixel by pixel
Color Histogram (CHIST)	Matches colors, position irrelevant
Color Histogram Quadratic (CHISTQ)	Matches similar color, position irrelevant
Wavelet (WAV)	Matches shape using wavelet theory
Smoothness Histogram (SHIST)	Matches color smoothness (contrast), position irrelevant

To finalize the experimental process, Weins and Ross [27] have used island model evolution in order to prevent premature convergence during the evolutionary steps. The results mentioned for the experiments show between 60 to 78% of best fitness score, and evolved images had characteristics of target images.

3.2.2 GenShade

Like Gentropy, GenShade [28], performs automatic texture evolution. The generated textures are compared with one or more target textures through the evaluation process. GenShade evolves RenderManShaders, in which chromosomes take the form of a hierarchical directed acyclic graph; the nodes of these graphs refer to the texture functions that are shader's primitives. Furthermore, Genshade employs a multiple-generation of shaders, which are inserted into the population in sorted order and with respect to their gender. Thus, all the shaders regardless of having a high or poor score have the chance to participate in reproduction. The shaders with higher scores on illumination are the male shaders and those with higher score on chromaticity are considered as female shaders.

Regarding to gender consideration, the aging option has been used in the experiments, in which more successful shaders are attained several opportunities to take part in the selection. The selection strategy uses male and female shader’s ranks based on probability in Gaussian distribution, where high rank shaders having small standard deviation are selected in odd generations and the ones with high standard deviation are selected in even generation numbers. These set of experiments, as mentioned above, have been conducted on genome textures: standard genetic algorithm, multiple-generation population, gender-based selection, multiple processes method, and aging. In the evaluation part, the system uses comparing the shader score of the produced texture in the current population with the shader score of the target images. These set of experiments with their own strategies, improved the results of the system comparing to the canonical genetic algorithm.

The texture language for the experiments consists of float X and Y, representing the current coordinates, an ephemeral constant, lum (luminosity), avg (mean of two arguments), tilrerad (repeating tile patterns), texture effects such as (noise, turb, turbflow, and cloud), if function (conditional processing), and forv, chn, and ichn (perform iterative processing on vectors).

3.2.3 Procedural Texture Evolution Using MOP

In another paper [29], automatic texture generation using multiobjective fitness evaluation has been studied. This research, applies Pareto ranking to the fitness objectives. The reason for using this strategy is its success in multiobjective optimization problems as well as maintaining objective's independence from one another.

The system takes image features mentioned in the Gentropy system (color, shape, and smoothness) with the difference that the feature tests hired in this research fall into a broader range comparing to the tests in Gentropy. In order to have a diverse population and preventing premature convergence, the diversity heuristic strategy was added to fitness. As a result, experiments used pure Pareto ranking without the diversity was still facing premature convergence, while performing Pareto ranking with diversity solved the problem with premature convergence.

Chapter 4

Lightweight Features of Images

In this chapter, the features used from Lombardi *et al.* [4] in this research are described. The motivation for using these image features relying on previous research in image retrieval and evolutionary design will be explained in detail. At the end of this chapter, results of distance tests conducted on different images to verify how the feature tests apply to real art world are shown.

4.1 Features Definition

As mentioned in Chapter 3, Lombardi *et al.* [4] defined lightweight image features to compare and classify images of different painters. Two types of features have been considered in their system: palette and canvas features. Palette features relate to the color space and are taken from the color map of the image. Canvas features are related to the frequency level of those colors in the image. Features used in this research have been divided into the two sets used in Lombardi *et al.* [4]. The first set includes 16 image features. Table 3 show palette scope features, which is the number of unique RGBs and 15 canvas scope features, such as min, max, mean, median, and standard deviation for red, green, and blue color channels of each pixel.

The second set of features, shown in Table 4, use the HSV color model that is more understandable to human perception than RGB color model. The second set of features comprises min, max, mean, median, and standard deviation for hue, saturation, and value of each pixel. Furthermore, other features including mean image intensity, color entropy, and line count are considered in this set of features that are basic shape descriptors. Mean image intensity is used to define the average brightness of a grayscale image [4]

Table 3: First feature set.

Feature Name	Type	Description
Palette Scope	Palette	The total number of unique RGB triples in an image.
Red Max	Canvas	The maximum value in the Red channel.
Red Min	Canvas	The minimum value in the Red channel.
Red Mean	Canvas	The arithmetic mean of the values in the Red channel.
Red Median	Canvas	The median of the values in the Red channel.
Red Standard Deviation	Canvas	The standard deviation of the values in the Red channel.
Green Max	Canvas	The maximum value in the Green channel.
Green Min	Canvas	The minimum value in the Green channel.
Green Mean	Canvas	The arithmetic mean of the values in the Green channel.
Green Median	Canvas	The median of the values in the Green channel.
Green Standard Deviation	Canvas	The standard deviation of the values in the Green channel.
Blue Max	Canvas	The maximum value in the Blue channel.
Blue Min	Canvas	The minimum value in the Blue channel.
Blue Mean	Canvas	The arithmetic mean of the values in the Blue channel.
Blue Median	Canvas	The median of the values in the Blue channel.
Blue Standard Deviation	Canvas	The standard deviation of the values in the Blue channel.

Table 4: Second feature set.

Feature Name	Type	Description
Hue Max	Canvas	The maximum value in the Hue channel.
Hue Min	Canvas	The minimum value in the Hue channel.
Hue Mean	Canvas	The arithmetic mean of the values in the Hue channel.
Hue Median	Canvas	The median of the values in the Hue channel.
Hue Standard Deviation	Canvas	The standard deviation of the values in the Hue channel.
Saturation Max	Canvas	The maximum value in the Saturation channel.
Saturation Min	Canvas	The minimum value in the Saturation channel.
Saturation Mean	Canvas	The arithmetic mean of the values in the Saturation channel.
Saturation Median	Canvas	The median of the values in the Saturation channel.
Saturation Standard Deviation	Canvas	The standard deviation of the values in the Saturation channel.
Value Max	Canvas	The maximum value in the Value channel.
Value Min	Canvas	The minimum value in the Value channel.
Value Mean	Canvas	The arithmetic mean of the values in the Value channel.
Value Median	Canvas	The median of the values in the Value channel.
Value Standard Deviation	Canvas	The standard deviation of the values in the Value channel.
Intensity Mean	Canvas	The global brightness of an image.
Color Entropy	Canvas	The degree of disorder in the frequency distribution of colors.
Line Count	Canvas	The number of lines detected by the Sobel edge detector.

Min and max RGB are obviously minimum and maximum pixel value for each color channel in the whole image. Mean and standard deviation RGB are the average and standard deviation of pixel values for each color channel and median RGB is the middle element of a sorted array of pixel values using quick sort algorithm. The same explanation applies for HSV channel too. Other features included in feature sets are number of edge pixels counted by Sobel edge detection algorithm, entropy, average intensity, and unique number of RGB pixels, are explained in detail in the following sections.

4.2 Sobel Edge Detection

In higher-level computer vision algorithms, edge detection techniques are considered as a valuable source of information about images [22].They provide important features extracted from edges (corners, lines, and curves) of an image. The reason edges can be a very helpful source of information is for the sudden change in intensity of one pixel to another. Generally, edges are the areas in images with strong intensity contrasts. In other words edge pixels have higher intensities comparing to other pixels of the image and this can be explained by the gradient values of the pixels. There are several edge detection techniques that are highly used in computer vision area including Sobel, Prewitt, Roberts, and Canny [30].

Lombardi *et al.* [4] use line count in the second feature set, which uses the Sobel edge detection algorithm. We have interpreted the line count to be the number of edge pixels in a Sobel filtered image that meet a luminosity threshold determined by the user.

The Sobel edge algorithm computes an approximation of image intensity function for edges by performing 2D gradient measurements on an image. It convolves the input image with two NXN (here, N=3) vertical and horizontal kernels to find the gradient magnitude and direction of each pixel of the image (Gx and Gy) in its neighborhood [31] [32]. The following formulas show oriented gradient magnitudes calculated from convolution of a 3X3 filter with the input image [30].

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \text{(Input Image)} \qquad (4.1)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \text{(Input Image)} \qquad (4.2)$$

Considering A to be the input image to the algorithm, the following computations help to understand the concept of image convolution where a set of pixels A are convoluted by the vertical convolution matrix (4.1)

$$A = \begin{bmatrix} P_0 & P_1 & P_2 \\ P_3 & P_4 & P_5 \\ P_6 & P_7 & P_8 \end{bmatrix}$$

$G_x = (P_8 * (-1)) + (P_7 * (0)) + (P_6 * (1)) + (P_5 * (-2)) + (P_4 * (0)) + (P_3 * (2)) + (P_2 * (-1)) + (P_1 * (0)) + (P_0 * (1))$ (4.3)

where the P_i values represent the pixel values of the image. The result G_x is the result for pixel P4. After this is applied to the entire image, the vertical edges are computed. A similar computation should be conducted at y direction with the horizontal matrix (4.2) to compute the horizontal edges to find G_y.

For each pixel, the vertical and horizontal gradient approximations can be combined to find the absolute gradient magnitude of a pixel as follows. |G| is the new edge luminosity for the pixel [33]:

$$|G| = \sqrt{Gx^2 + Gy^2} \qquad (4.4)$$

The result of Sobel edge detection implemented in our system is shown in Figure 6. In this figure, the left side picture is the original picture given to the algorithm and picture at the right side is the result of applying Sobel algorithm on that picture.

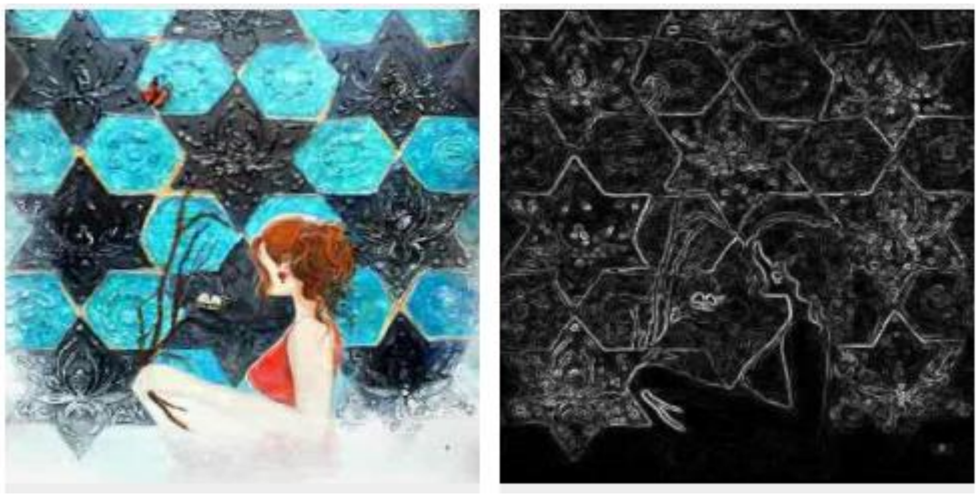


Figure 6: Sobel edge detection: original image and Sobel filtered image.

In the edge image (right side), many pixels have a non-zero luminosity, and might be counted as edge pixels, even though they are not clearly on edges. We assumed that defining a threshold for the algorithm would be helpful in counting the number of real edge pixels. We decided to allow the user to set the sensitivity of Sobel edge detection, to give the system flexibility for different experiments and target images. Therefore, we defined a user-defined threshold range of between 0 and 1. A threshold of 0.5 is reasonable in many cases. A Sobel image pixel that exceeds this threshold luminosity will be considered an edge pixel, while a pixel that is less than the threshold is a non-edge. For example, the number of edge pixels for this image considering the threshold of 0.5 is shown in Figure 7.

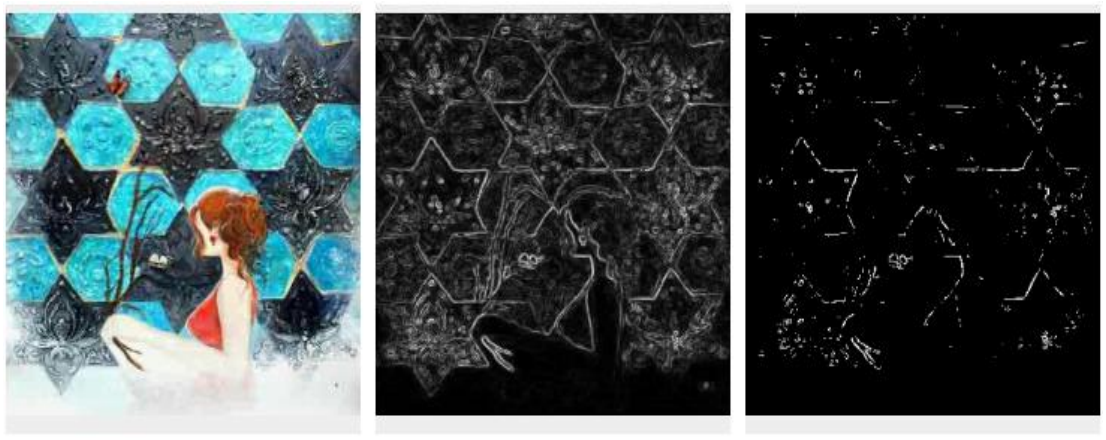


Figure 7: Sobel edge detection: original image, Sobel filtered (no threshold), and Sobel filtered (threshold of 0.5).

Obviously, fewer pixels are selected as edge pixels that are actually located on the edges. Higher threshold values means that fewer pixels are included as edge pixels.

4.3 Color / Grayscale Entropy

Color entropy is a scalar value, formally known as color distribution entropy, which is a statistical measure of randomness of an image [22] [34]. Shannon entropy, used in this research, was named after Claude Shannon who proposed the idea in 1948 as part of the field of information theory [35], which is defined as follows [36] [37]:

$$H = -\sum_{k=0}^{m-1} P(x) log_b P(x) \qquad (4.5)$$

where H is entropy of set X, $0 \leq H \leq log_b P(x)$, and $P(x)$ is the probability of the occurrence of the same values for x, i.e. $P_i = P_r (X = x_i)$. m is the number of grayscale levels. Based on the formula, entropy is a measure of uncertainty (probability distribution of each event) associated with random variables or measure of randomness of the information (amount of information for every event). Depending on the base of the logarithm, different units of entropy can be defined. Shannon entropy uses b = 2, which means that entropy measures the number of bits necessary to represent the signals denoted by the measured signal in the question. If the events have unequal probability distribution, a less amount of information can be received from the source of those events. This is because if one event is less probable to occur, the average amount of information received from that event will be less than $log_2 P(x)$. This means there is lower entropy. Accordingly, if an event is certain to be the outcome, the entropy is zero [36].

In image processing, the entropy of an image is a color descriptor that uses image histograms to define the probability of spatial distribution of each pixel color in the whole image [37] that indicates the degree of color disorder in that image. It can be inferred that entropy corresponds to the state of gray levels an individual pixel can obtain. Therefore, considering $T = M \times N$ be the total number of pixels in image X and $l \in \{0, 1, \ldots, L - 1\}$, be the intensity scale, where $L = 2^T$, and the number of pixels in the image at the intensity level l to be n_l , Shannon entropy will be computed as follows [38]:

$$P_l = P_r (X = l) = \frac{nl}{T} \tag{4.6}$$

$$H = -\sum_{l=0}^{L-1} \frac{nl}{T} log_2 \frac{nl}{T}$$

Using the above formula entropy of images can be computed easily. In this research, entropy of the grayscale image is been computed based on counting the number of grayscale values for the image pixels stored in a vector and then finding the entropy value using formula (4.6). Our system does this, using a modification of code available at [39].

It has been shown that in images with lower entropy, such as a solid color appear to have little contrast and a huge number of pixels with the same color densities. A solid color image, like a perfectly flat black, has entropy of zero. On the other hand, higher entropy images have a great contrast from one pixel to the next and a higher distribution of colors can be seen in those images [34]. Figure 8 shows two images of entropy zero and higher entropy value.

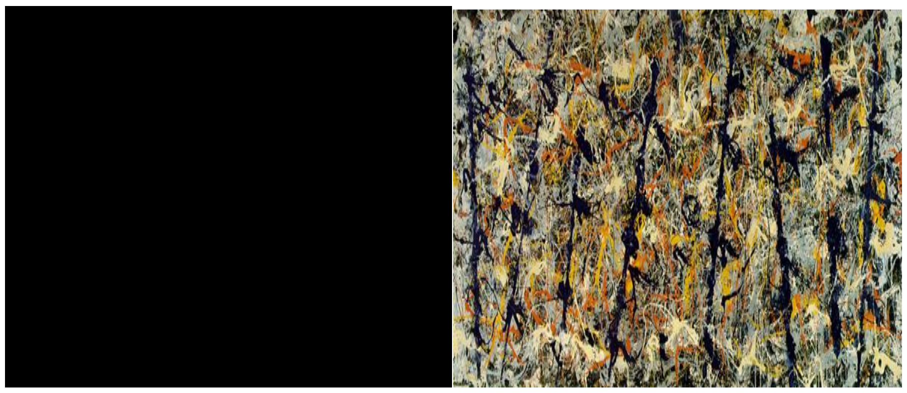


Figure 8: Low and high entropy.

4.4 Unique Triples

Unique triples refer to the number of unique pixel values in the image. By reading and storing pixel values into a data structure such as an array list, and deducting repeated values from the set, the number of unique pixel values is the remaining size of the array list.

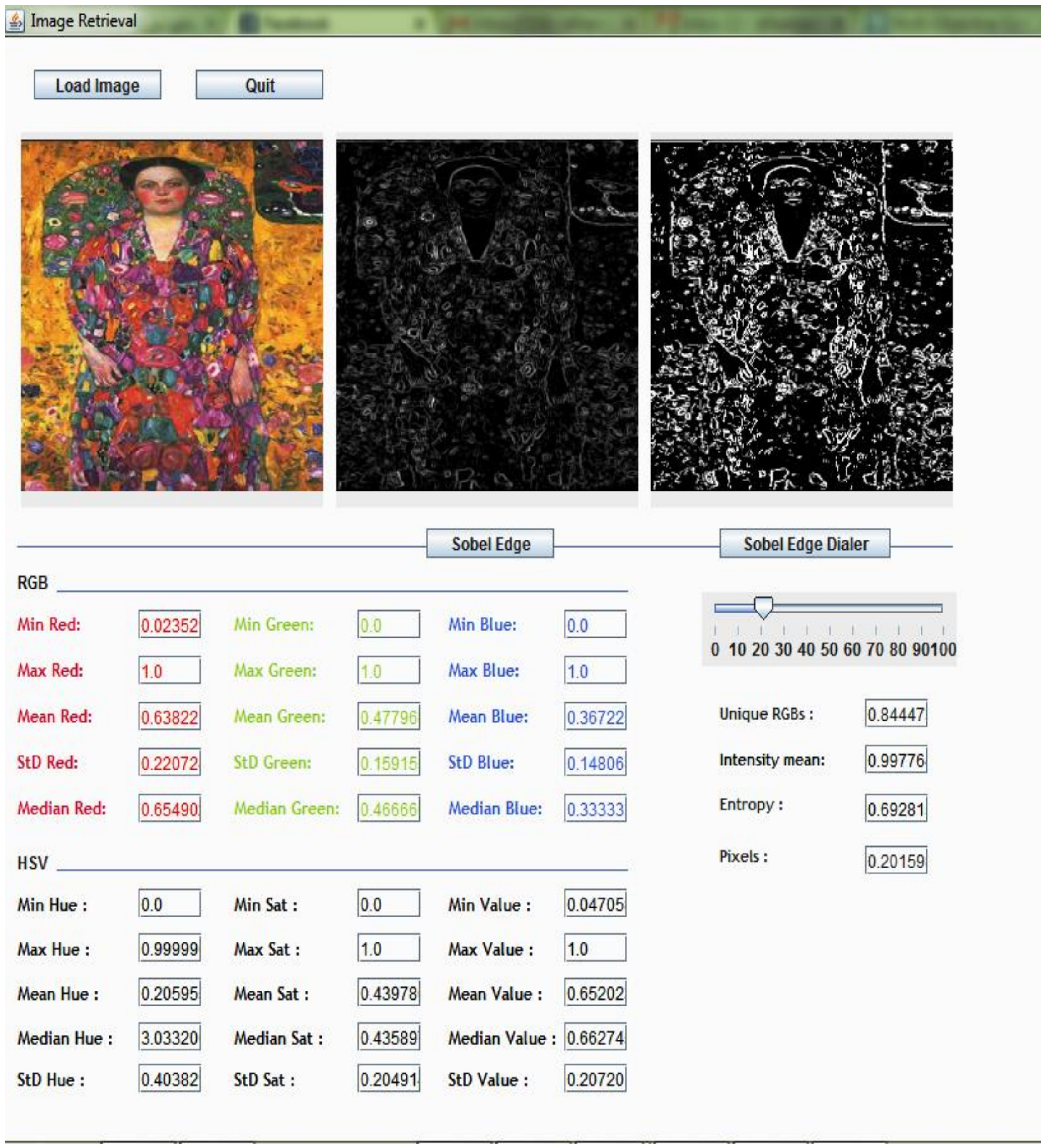
4.5 Average Intensity

Average intensity is average brightness of pixel colors in a gray scale image. To compute the average intensity of pixels in an image, we used the NTSC intensity formula in which weighted sum of pixel values in RGB color model represents the intensity value [40]. Storing intensity values in an array and dividing it by the size of image gives us the average intensity of the whole image.

$$Pixel\ Intensity = \ 0.299 * Red + 0.587 * Green + 0.114 * Blue \qquad (4.7)$$

4.6 Implemented GUI

We implemented a GUI with all feature sets from Lombardi's *el al* and redesigned Sobel edge detection. Figure 9 shows our implementation of Lombardi's feature sets.



RGB

Min Red:	0.02352	Min Green:	0.0	Min Blue:	0.0
Max Red:	1.0	Max Green:	1.0	Max Blue:	1.0
Mean Red:	0.63822	Mean Green:	0.47796	Mean Blue:	0.36722
StD Red:	0.22072	StD Green:	0.15915	StD Blue:	0.14806
Median Red:	0.65490	Median Green:	0.46666	Median Blue:	0.33333

HSV

Min Hue :	0.0	Min Sat :	0.0	Min Value :	0.04705
Max Hue :	0.99999	Max Sat :	1.0	Max Value :	1.0
Mean Hue :	0.20595	Mean Sat :	0.43978	Mean Value :	0.65202
Median Hue :	3.03320	Median Sat :	0.43589	Median Value :	0.66274
StD Hue :	0.40382	StD Sat :	0.20491	StD Value :	0.20720

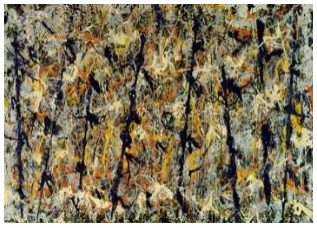
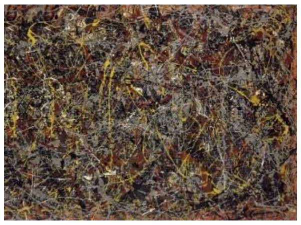
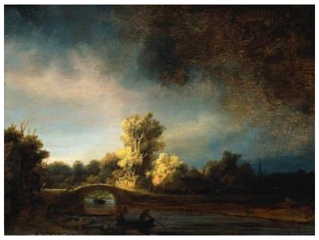
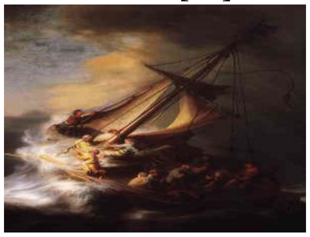
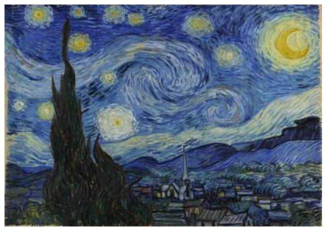
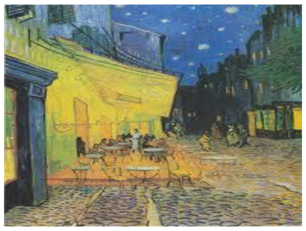
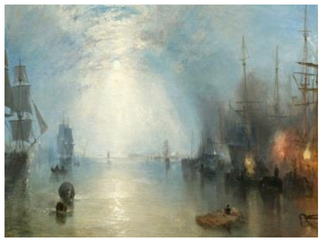
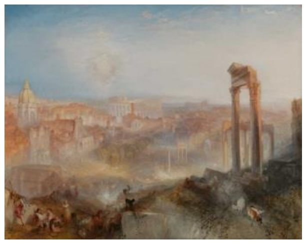
Unique RGBs :	0.84447
Intensity mean:	0.99776
Entropy :	0.69281
Pixels :	0.20159

Figure 9: GUI of distance measures.

4.7 Testing the Features

Lombardi's *et al.*, system was tested with two different sets of experiments. The tests were to compare paintings of different painters to see how distinct two paintings of a painter are and to categorize paintings of different style. In order to check the validity of the method, the distance tests on extracted features were conducted to identify if the difference between feature values of two paintings of different painters are distinguishable. Two paintings of 4 famous painters were chosen for this test as shown in Table 5. All painters were evaluated with all features in sets 1 and 2.

Table 5: Paintings used for distance measure test.

Painter	Painting 1	Painting 2
Jackson Pollock	Blue Poles [41] 	Number 5 [42] 
Rembrandt	The Stone Bridge [43] 	The Storm on the Sea of Galilee [44] 
Van Gogh	The Starry Night [45] 	Café Terrace at Night [46] 
Turner	Keelmen Heaving in Coals by Moonlight [47] 	Modern Rome – Campo Vaccino [48] 

In order to compute the distances between features of each set for each two paintings, X and Y, we used the following mathematical distance formula for the first and second set respectively.

$$Feature\ Distance = \sqrt{\sum_{i=0}^{n}\left(feature\ i\ (X) - feature\ i\ (Y)\right)^2} \qquad (4.8)$$

where n is the number of features in each set.

The test results for the first set are mentioned in Table 6, and for the second set of features in Table 8.

Table 6: Distance measures for the first feature set.

	café terrace	starry night	Keelmen	modern rome	number5	Galilee lake	stone bridge
Blue poles	0.2498	0.6410	0.7848	0.7770	0.5898	0.7490	0.6308
café terrace		0.6955	0.8236	0.8066	0.7101	0.8497	0.7307
starry night			0.6021	0.6119	0.3848	0.4940	0.3845
Keelmen				0.4136	0.8052	0.9748	0.8567
modern rome					0.6685	0.8080	1.E-05
number5						0.2040	0.1150
Galilee lake							0.1502

In Table 6, the distance between every two paintings has been computed using formula 4.8. The goal is to find the distances between two paintings of the same painter, to see if paintings by the same painter are close to each other. Although in some cases paintings of different painters might have closer feature values and be less distinct, distance values can be explained by replacing distances with ranks, where rank 1 is the closest distance painting, rank 2 the next, and so on.

Table 6 shows average ranks for each painter, which is the average rank of that painter's 2 paintings with one another.

Table 7: Average set 1 ranks for each painter.

Painter’s Name	Average Rank
Jackson Pollock	3
Van Gogh	4.5
Turner	1
Rembrandt	1.5
Set 1 for all paintings	4

The average ranking for set 1 is 2.5, which is less than 4 (half of comparison cases). The same analysis is applicable to the set 2 features in Table 8, and the ranks are shown in Table 9.

Table 8: Distance measures for the second feature set.

	café terrace	starry night	Keelmen	modern rome	number 5	Galilee lake	stone bridge
Blue poles	0.5899	0.4916	0.4811	0.3470	0.3829	0.6186	0.4422
café terrace		0.2486	0.4787	0.5016	0.8326	1.005E	0.8035
starry night			0.4820	0.4710	0.6781	0.8618	0.6596
Keelmen				0.2390	0.7567	0.9114	0.6231
modern rome					0.6327	0.8079	0.5332
number5						0.2858	0.3632
Galilee lake							0.4519

Table 9: Average set 2 ranks for each painter.

Painter’s Name	Average Rank
Jackson Pollock	2.5
Van Gogh	1
Turner	1
Rembrandt	2.5
Set 2 for all paintings	4

The average of average ranks for set 2 for all painters is 1.75, which is less than 4 (half of comparison cases) and less the average rank calculated for set 1. This shows that the lightweight features are generally successful in matching the examples of painter’s art works with each other.

Chapter 5

System Implementation

In this chapter, the implementation details of our system for this research are explained. These details include the GP implementation, GP parameters, texture language, and fitness criteria.

5.1 GP System

An evolutionary arts system called JNetic Textures designed by Steve Bergen [49], has been used in this research. The core of the system is the ECJ [50] , which is a Java-based GP system. JNetic Textures takes the advantage of a GUI that gives the users the ability to set and modify the parameters and select each objective easily. Therefore, it can be integrated with the designed GUI interface for this research. Moreover, the texture evolution of his designed system is similar to that required in this project, with differences in evaluation criteria and multiobjective methodology.

At the beginning of the run, a target image is selected. The image features will be extracted and a subset will be used as targets for GP evolution. It worthwhile to mention that all the extracted feature values for the experiments are float numbers between 0 and 1. Then the GP system initializes the first generation of the population. During the evolutionary process, a texture formula will be generated. The goal is to evolve textures that have similar feature values as the chosen target image.

The user can choose any of the 34 objectives as the evaluation criteria. Then the fitness function evaluates each of the generated individuals and a fitness score will be assigned to the individual. At the end of the generation all evaluated individuals will be ranked using "normalized sum of ranks" mentioned in Section 2.1.4. The assigned ranks will be used in tournament selection for selecting parent individuals for the next generation. Figure 10 shows the approach.

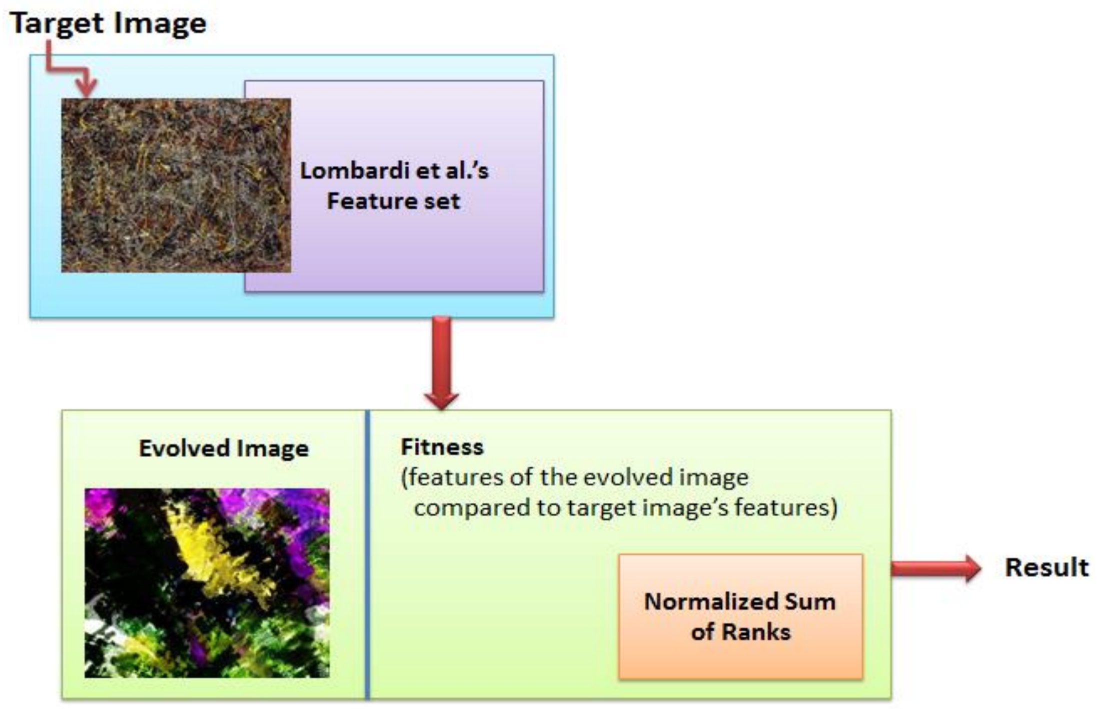

Target Image
Lombardi et al.'s Feature set
Evolved Image
Fitness
(features of the evolved image compared to target image's features)
Normalized Sum of Ranks
Result

Figure 10 : The architecture of our proposed system.

As Figure 10 shows, the purpose of this system is not to regenerate the original input image. The evolved image can be very different from the input image.

5.2 GP System Parameters

The provided GUI enables the user with the option to choose and modify parameters for each run. Below is the list of parameters in the GUI [49]. See [6] for further explanation on these parameters.

- Number of Generations : The total number of generations in each run.
- Population Size: The number of individuals in each run.
- Tree Initializer: Is ramp of half-and-half. This method creates a variety of tree shapes and sizes.
- Initialize tree max and min depth: These parameters restrict the size of initialized trees.
- Crossover and Mutation Percentage: These parameters show probability of using crossover or mutation for reproduction.
- Max Tree Depth: Maximum size of GP tree.
- Probability of Selection Terminals/Non-terminals: These parameters define the probability the crossover and mutation select terminal or nonterminal nodes to do the genetic operation.
- Size of tournament: This parameter defines how many individuals are randomly selected by tournament selection method.

5.3 GP Language

In this research, the GP language used for the experiments is similar to the language set used in JNetic Textures for the experiments. The 2D coordinates of a pixel (X, Y), are terminals in a range of between 0 and 1. The GP tree (texture formula), uses different primitives including mathematical functions (arithmetic operators, trigonometric functions, log, and other common mathematical functions). In addition to the mentioned function sets, other functions such as Perlin Noise can be used for the experiments. Perlin Noise is a gradient noise that is widely used in computer graphics area, that adds pseudo-random noise effects to mathematical expressions [51]. Mathematical functions used in our function set are listed below:

Add = Arithmetic sum function on two operands

Cos = Cosine function on single operand

Div = Arithmetic devision function on two operands

ERC = Ephemeral Random Constant

Log = Logarithm function on single operand

Mod = Remainder calculation function on two operands

Mul = Arithmetic multiplication function on two operands

Pow = Power function on two operands (base , and power)

Sin = Sinus function on single operand

Sub = Arithmetic subtraction function on two operands

5.4 Fitness Function

As mentioned earlier, the goal is to compare statistical features of a given target image with features of an evolved texture. Therefore, an automatic fitness evaluation is designed to score measures of objective distances, which is based on the texture features the user has chosen to be included in the evaluation. Objectives are ranked using normalized sum of ranks method. In each run, mean fitness values for the best individual of each generation will be computed for fitness plotting.

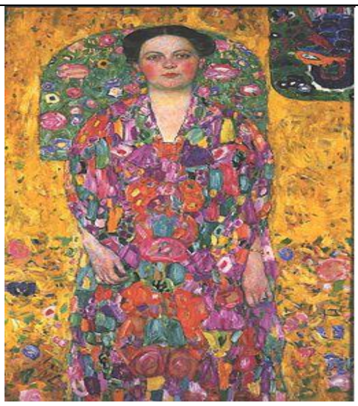
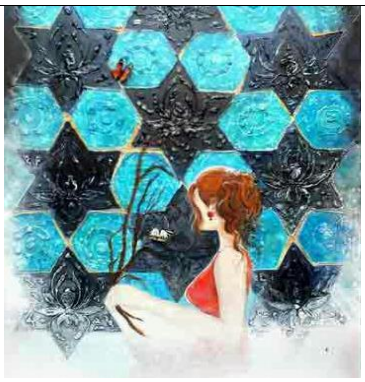
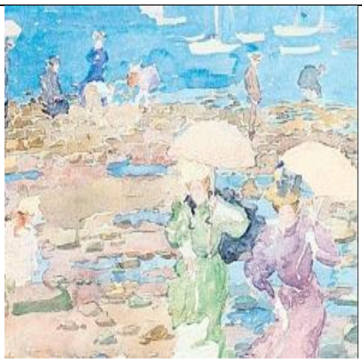
Chapter 6

Experiments and Results

Several sets of experiments were conducted on three chosen images to analyze the performance of the lightweight features mentioned in chapter 4. In choosing images for experiments, we tried to consider distinct image characteristics, to see how they work with the feature measures. Table 10 shows the target images chosen for the experiments.

Target image 1, "Portrait of Eugenia Primavesi," is a painting by Gustav Klimt [52]. This painting features various colors with different hues and saturations, high entropy, and high average intensity. "The Look" is a painting by Fatemeh Rashidi, and the focus of this painting is mostly on different saturations of blue color, fewer numbers of colors, and higher number of edges. The last target image is "Maurice Summer Visitors" by Prendergast [53], and clearly features low saturation, brighter colors, lower entropy, edges, and higher intensity.

Table 10: Target images used in experiments.

Image name	Image
Target image 1: “ Portrait of Eugenia Primavesi ”	
Target Image 2: “The Look”	
Target Image 3: “Maurice Summer Visitors”	

All the experiments have been setup to use fixed GP parameter values. Images should not be too large to reduce run time during experiments. Therefore, after testing different image sizes the following setup, shown in Table 11, was decided to be the standard for all experiments.

Table 11: GP setup parameters.

Parameter	Value
Tournament Size	3
Number of Generations	30
Population Size	200
Maximum Tree Size (depth)	10
Crossover (%)	90
Mutation (%)	10
Image Size	400X300
Tree Initializer	Ramped half and half
Number of Runs/ Experiment	10

The following sections discuss each experiment thoroughly.

6.1 Full Sets Experiments

This experiment is designed to test all the lightweight features in each feature set. Experiments are done separately on sets 1 and 2 for each image. The following subsections describe the experiments in detail.

6.1.1 Full Set 1

The first set includes 16 features including min, max, mean, median, and standard deviation of RGB values in addition to the number of unique RGBs. These features are listed in Table 2 in Section 4.1.

- **Target Image 1**

Figure 11 show the average fitness over 30 generations for this target image. The fitness plots for this image show that error values are converging. This means that the distance between feature values for the evolved image and the target are reducing for the whole population through the generations.

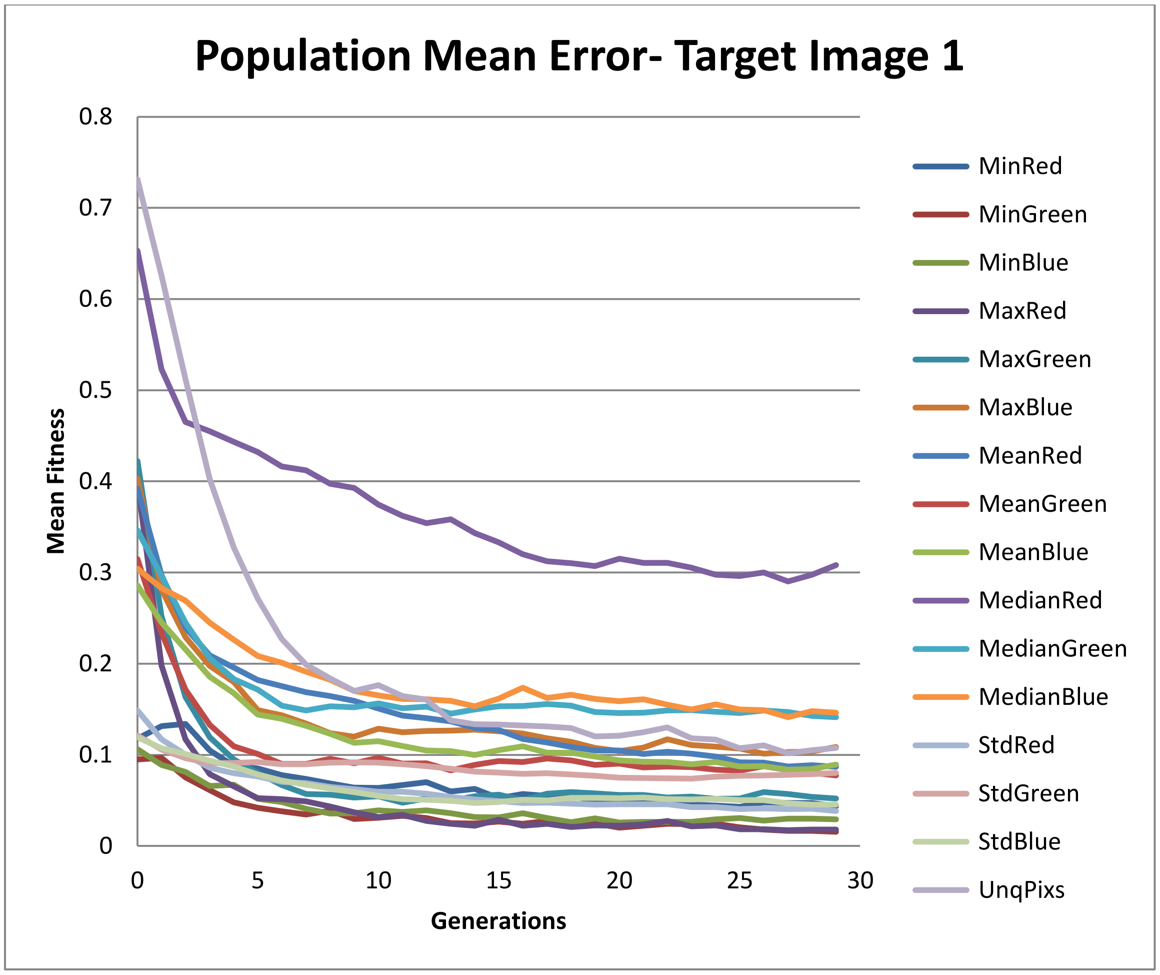


Figure 11: Population mean error for target image 1for 10 runs - full set1.

As the plot shows, among all the features, unique pixel value has significantly dropped since the early generations. Other features such as Median Green, Median Red, Mean Green, Mean Red, Max Red, Max Blue, Mean Blue, and Max Green have shown considerable improvement through the first 10 generations. Min Red has an increase in the distance values in the early generations but as the plot shows, it has started converging after the first couple of generations. This explains the influence of sum of ranks that some objectives sacrifice to benefit the majority.

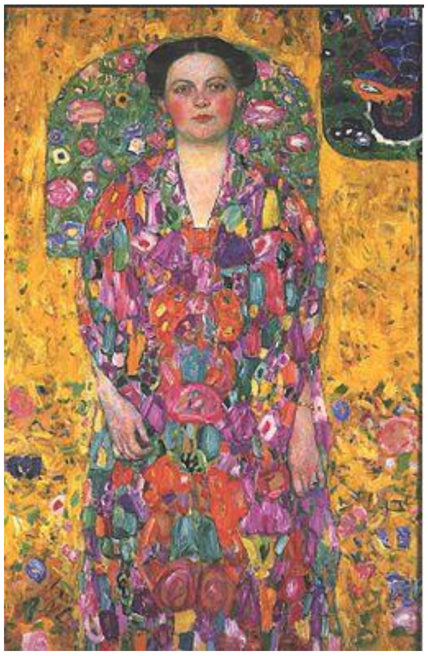
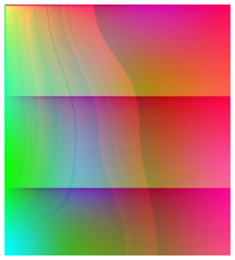
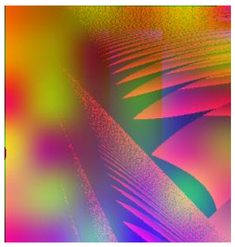
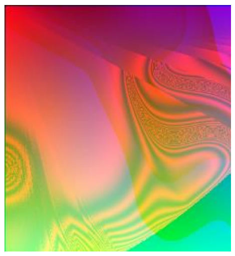
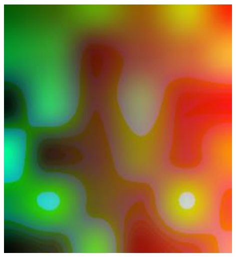
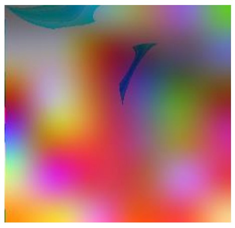
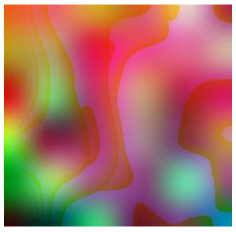
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 12: List of best solutions for target image 1 for 10 runs - full set 1.

Figure 12 shows top ranked solutions in the population for 10 experiments for this target image. This target image demonstrates a variety of colors from golden yellow, red, purple, and green to darker colors such as dark blue and black. The best solutions presented in the figure are expressing a considerable effort to maintain their closeness to the target image. In all of the solutions, presence of the mentioned colors exists in the target image is noticeable.

- **Target Image 2:**

The experiments for this target image show that all objectives are converging too. Min Blue and standard deviation Red have risen at the first generation but since the second generation until the last one, they have started to converge. Median green and median blue have started their convergence from a higher point in the graph but have noticeably dropped at the early stages. They have continued their converging trend from that point but at a higher level than the other objectives.

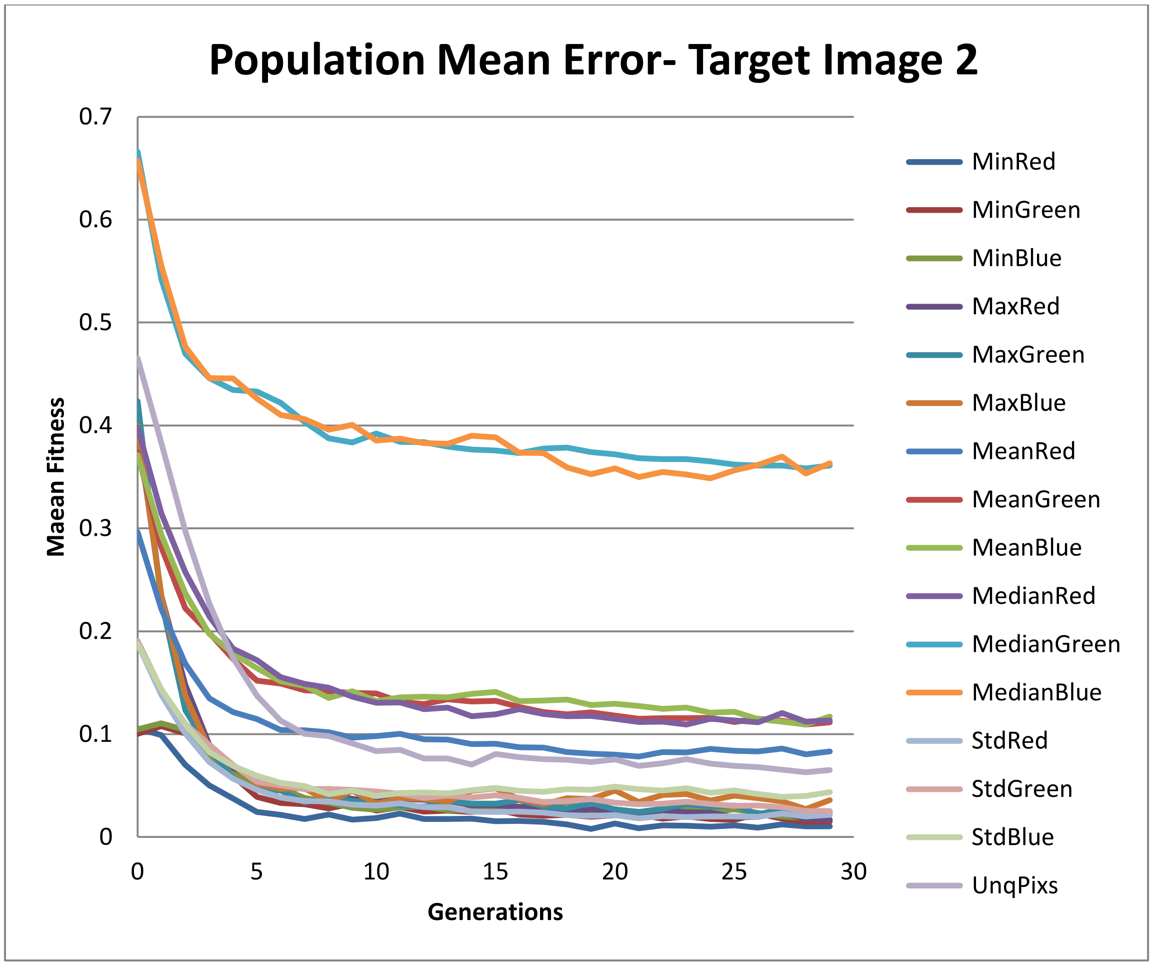


Figure 13: Population mean error for target image 2 for 10 runs - full set1.

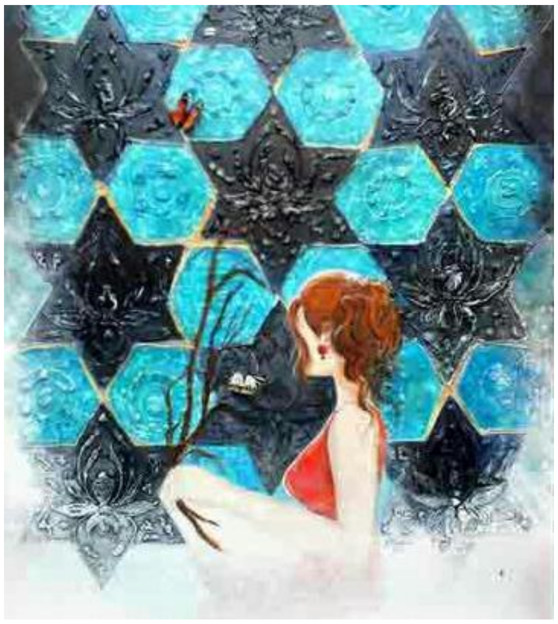
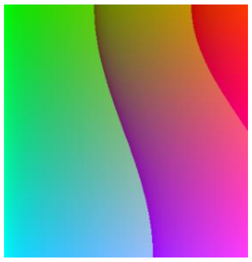
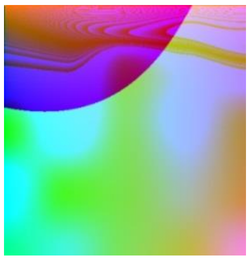
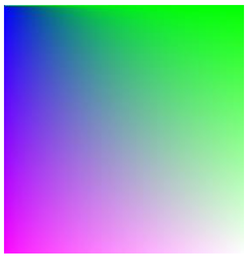
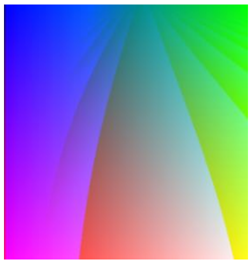
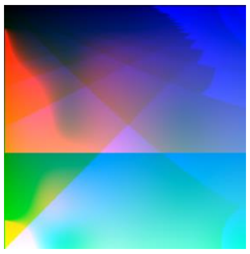
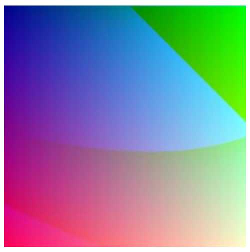
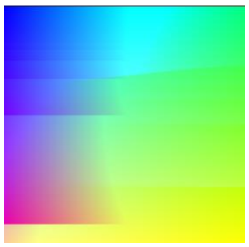
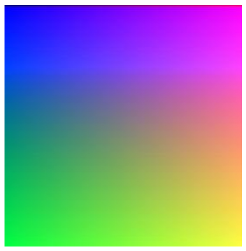
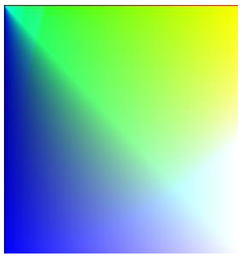
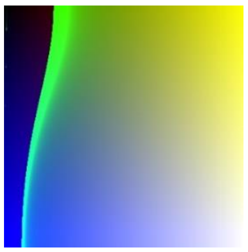
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 14: List of best solutions for target image 2 for 10 runs - full set 1.

Figure 14 shows best solutions for this target image over 10 experiments. The color palette used in this target image, shows shades of blue, white, red, and brown. In generated images, plenty of blue and white pixels show that textures are getting close to the target. However, in all the images other colors rather than red are observable such as pink, yellow, and green. According to the color model below [54], by mixing the major colors having these colors in the generated textures is not far from expectations. In fact, this shows an effective evolution is happening in experiments.

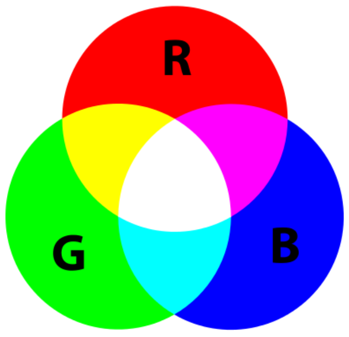


Figure 15: RGB color model

Among all the solutions listed in the figure, solutions 3 and 6 have shown a better closeness in terms of colors to the target image, as almost all the different colors used in the target image are used in these solutions too.

- **Target image 3:**

This target image demonstrates a combination of bright colors. The plots in Figure 16 for this image show that except for three of the features including min blue, min green, and min red that have an increase at the beginning, the rest have perfectly converged from the first generation.

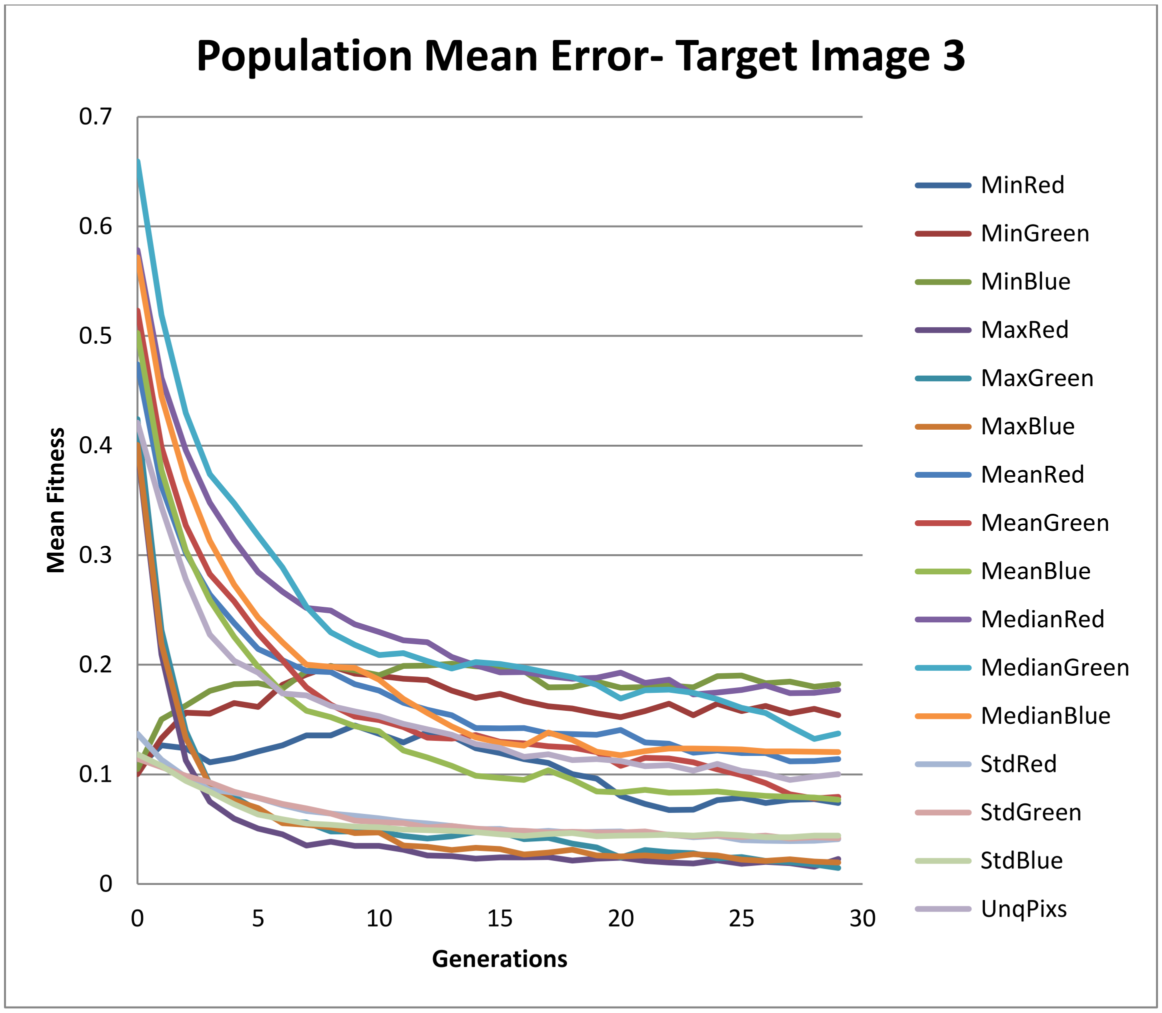


Figure 16: Population mean error for target image 3 for 10 runs - full set1.

Figure 17 shows best solutions for this target image over 10 experiments. Generated textures listed in the following figure, clearly show that the color palette used in the target image can visibly be found in the solutions.

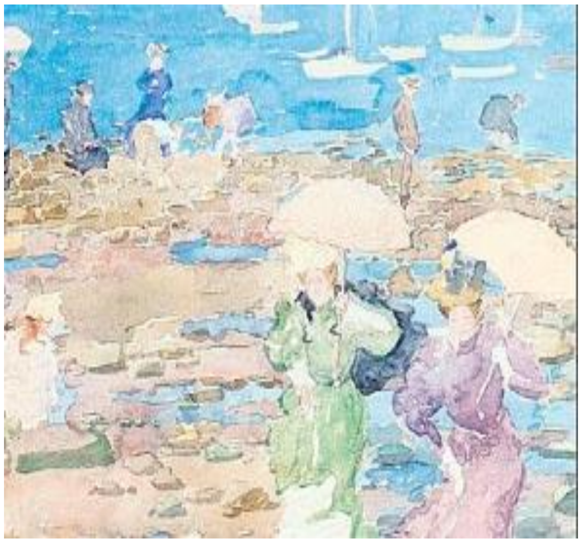
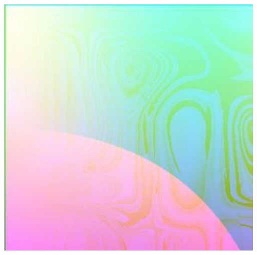
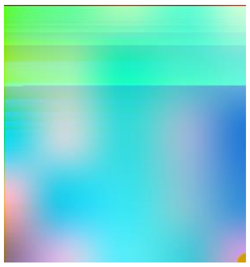
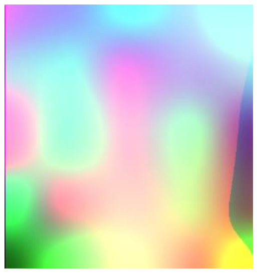
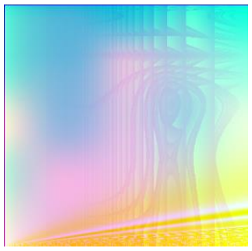
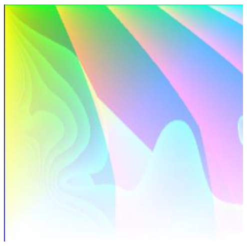
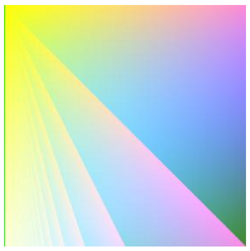
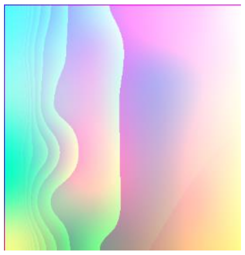
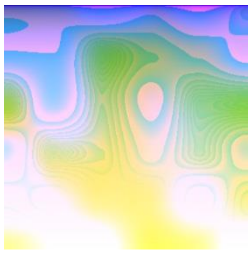
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 17: List of best solutions for target image 3 for 10 runs - full set 1.

6.1.2 Full Set 2

The second set contains 18 features including min, max, mean, median, and standard deviation of HSV values in addition to mean intensity, color entropy, and number of edge pixels in the image using Sobel edge detection. These features are listed in Table 3 in Section 4.1. Similar to the set1 experiments, the same three images have been tested with features in this set.

- **Target Image 1:**

Fitness plots for this image show that fitness values are converging. Figure 18 shows the average fitness over 30 generations for this target image.

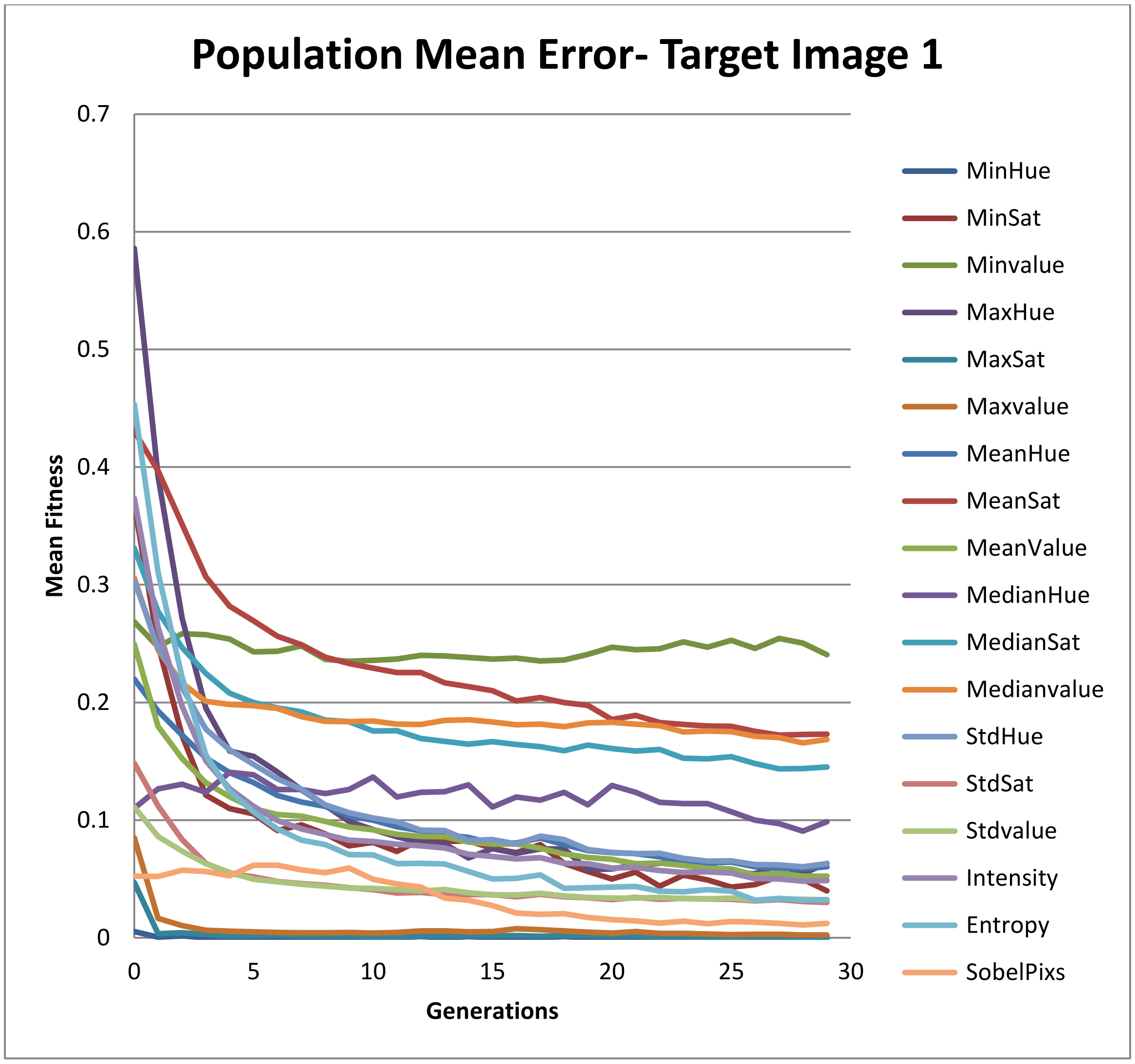


Figure 18: Population mean error for target image 1 - full set 2.

According to the fitness plots above, min hue, max sat, and max value had the lowest distance values so their plots were very close to the base. Max hue, mean sat, entropy, intensity, and min sat have shown a descent convergence since the first generation.

Median hue and Sobel pixels have slightly risen in the first 10 generations but have converged after that. Min value showed a different behavior than other objectives. Although it has dropped in the first couple of generations but it has remained steady till generation 20 and has risen a bit after that. However, it has smoothly fallen in later generations.

Figure 19 Shows best solutions during 10 experiments. Solutions listed in the figure, clearly show that all the colors used in the target image’s palette are used through the evolutionary process. The difference between solutions generated using this set of objectives and set 1 is that set 2 has been paid attention to different shades of red existing in the target image rather than the absolute red value.

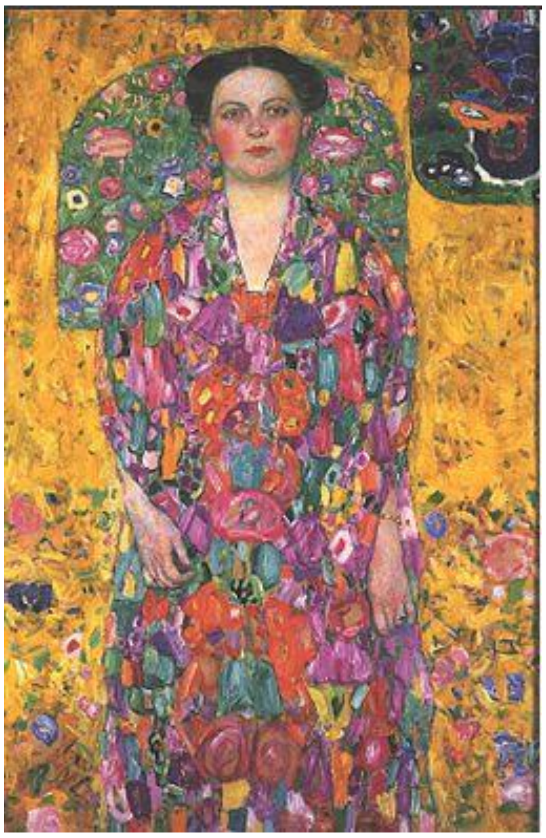
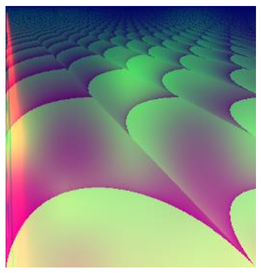
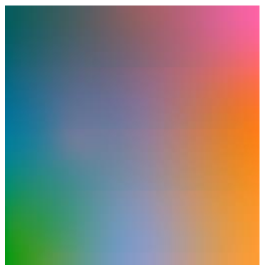
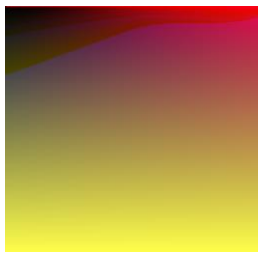
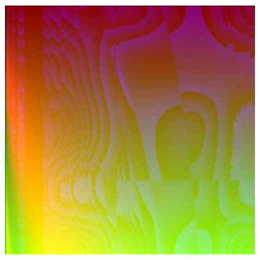
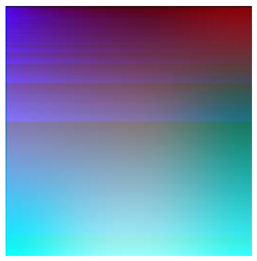
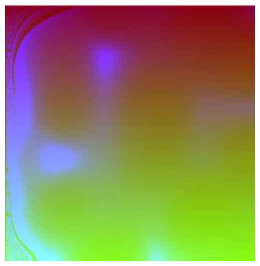
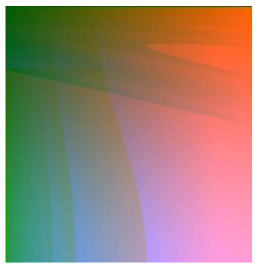
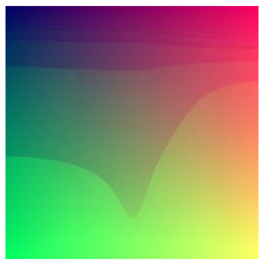
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 19: List of best solutions for target image 1for 10 runs - full set 2.

- **Target Image 2:**

Similar to the previous experiments on target image 1, in this set of experiments too max sat, min hue, and max value are converging to 0 but very close to the base line. All the other objectives are perfectly converging except for max hue. It shows a great convergence at the first 5 generations but suddenly rises above 0.6 within 10 generations. After generation 15, it improves by reducing distances to near 0.5 at generation 30.

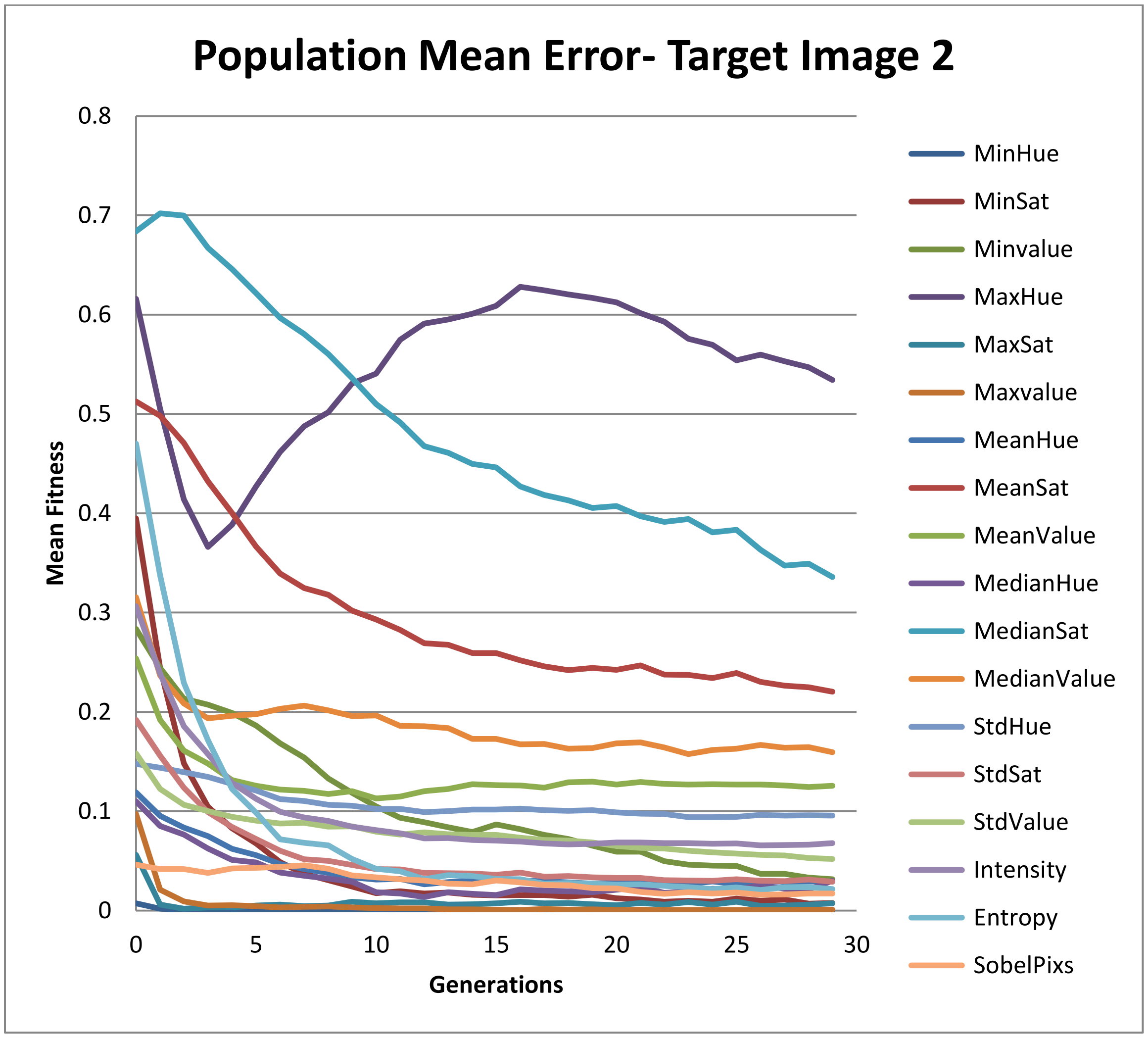


Figure 20: Population mean error for target image 2 - full set 2.

As explained earlier for target image 1, set 2 features are concerned with shades and brightness of the colors in addition to entropy, edge pixels, and mean intensity of the image. It is expected to see different shades of blue, red, and white. Nevertheless, as discussed in the previous section for the RGB color space model, other colors such as pink and green may appear in the generated textures. Figure 21 shows best solutions for 10 experiments on this target image.

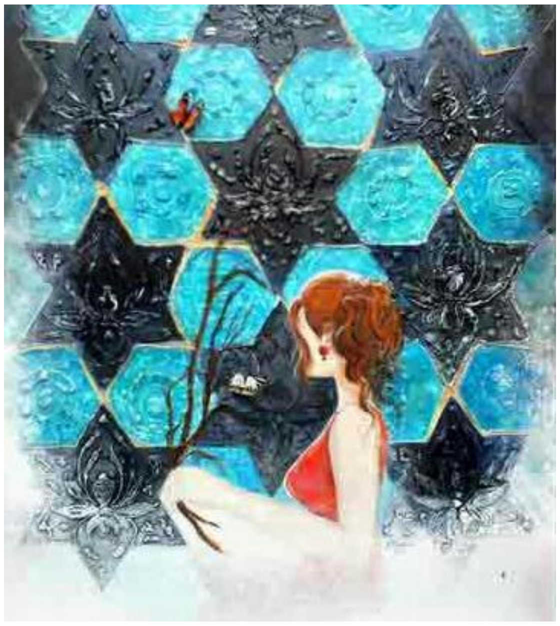
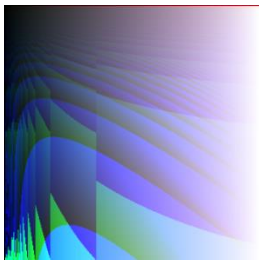
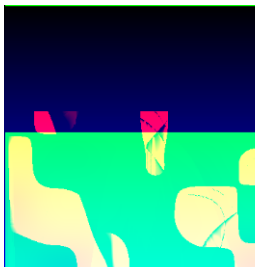
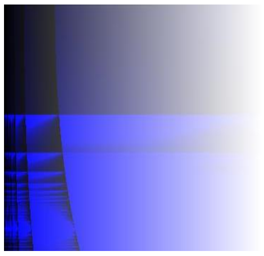
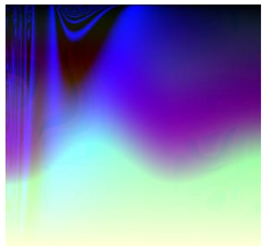
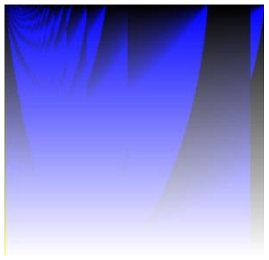
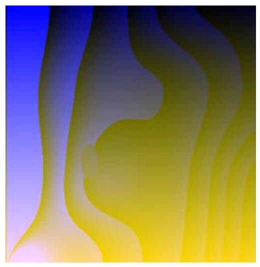
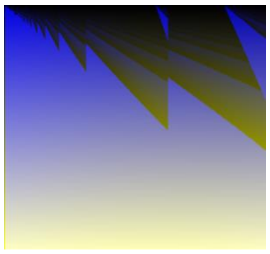
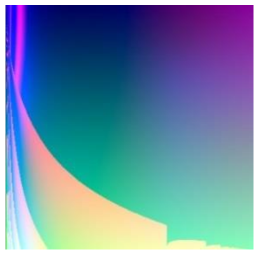
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		

Figure 21: List of best solutions for target image 2 for 10 runs - full set 2.

All the textures are successfully showing that shades of blue, which take over the most area of the target image, have been used in all the textures. In addition, as mentioned in Section 6.1.1, presence of other colors could be explained based on color mixes according to the RGB color space model. Not all the runs had good results, for example run 6 shows unconvincing results.

- **Target Image 3:**

All the plots in Figure 22 in this set of experiments are converging except for median sat, min value, and max sat that have risen after a few generations. Median sat has dropped until generation 5 but has started to rise until the last generation. Min value has risen in the first 5 generations from 0.3 to 0.5 and has remained steady after the 5th generation until the end. Max sat has dropped to reach the base line in the first 2 generations but has risen after that and remained steady.

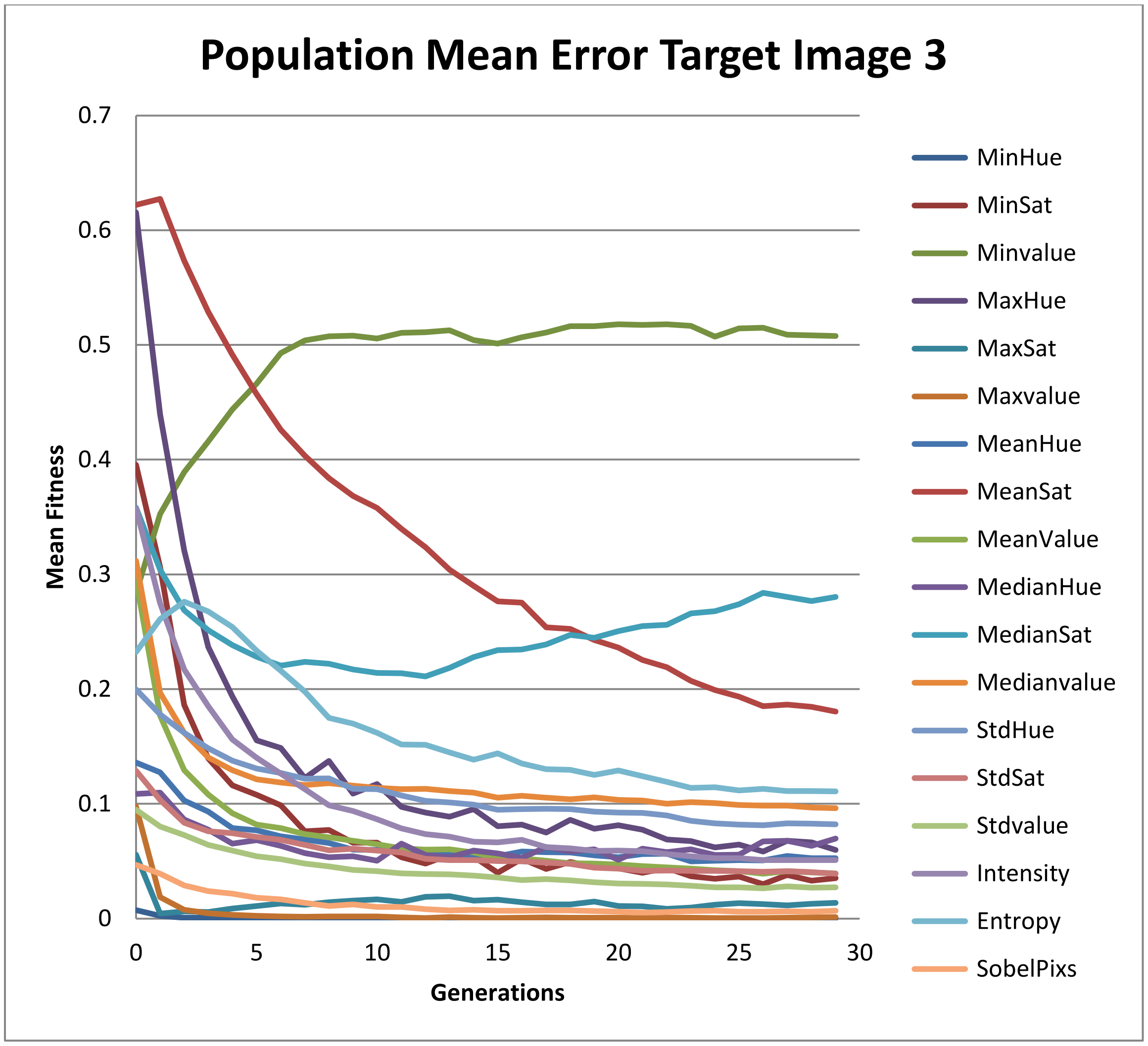


Figure 22: Population mean error for target image 3 for 10 runs - full set 2.

One possibility for these misbehaviors could be the brightness of the colors in the target image. According to the HSV color model shown in Figure 23 [55], all the colors reach their highest level of brightness when the value gets bigger. The same explanation works for saturation as well. Figure 24 shows best solutions of these experiments.

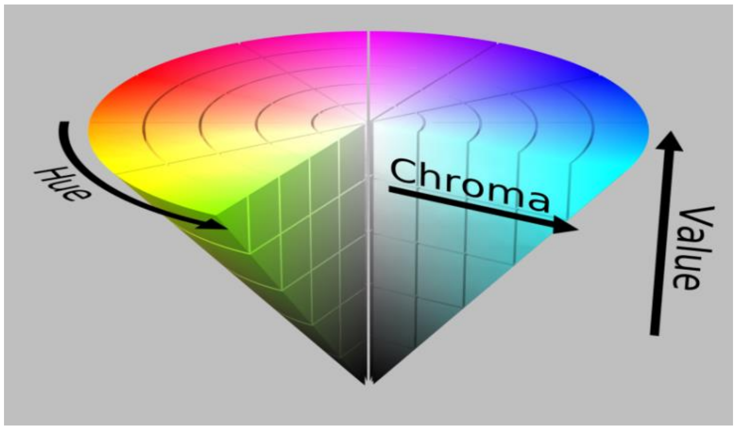


Figure 23: HSV color model.

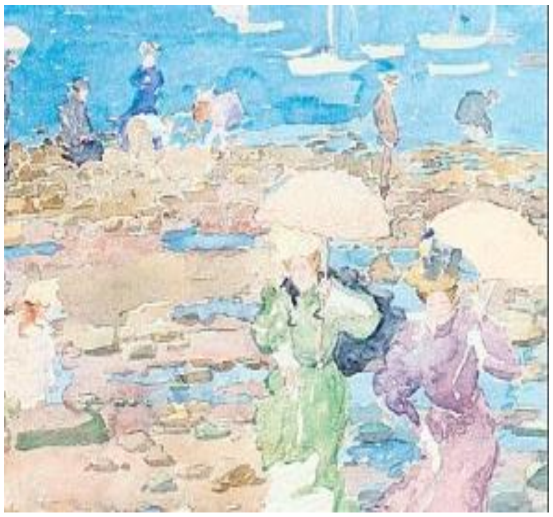
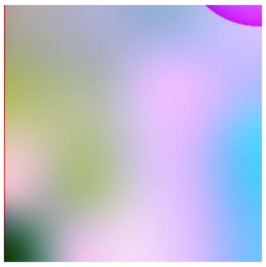
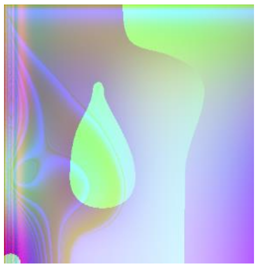
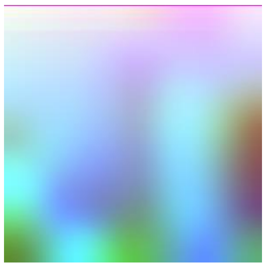
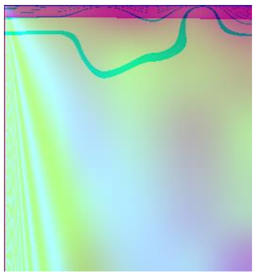
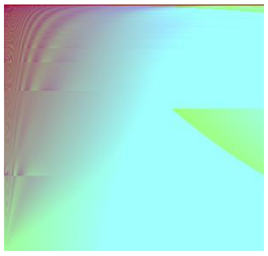
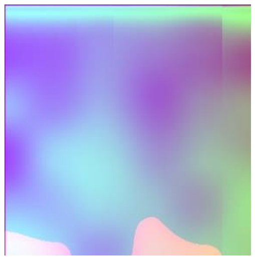
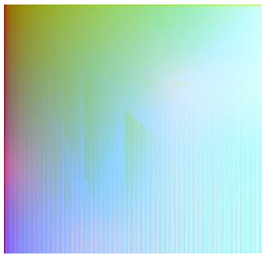
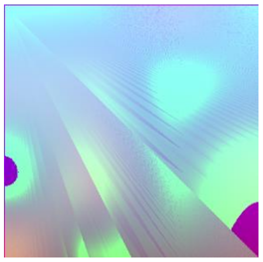
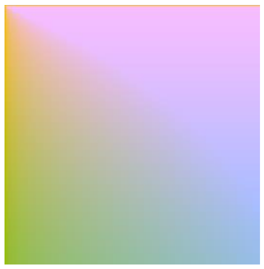
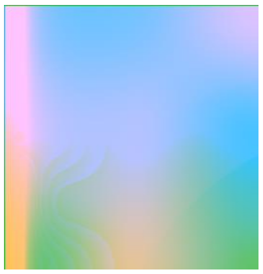
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 24: List of best solutions for target image 3 for 10 runs - full set 2.

6.2 Reduced Sets Experiments

Based on results from previous experiments, we decided to reduce some of the features in both sets to see if we are going to get any improvement in results compared to the full set experiments. Therefore, some features such as min, max, median from both sets were removed. Below, results for these experiments with reduced sets of features are explained.

6.2.1 Reduced Set 1

Features chosen for these experiments are mean (R, G, B), standard deviation (R, G, B), and number of unique RGB pixels.

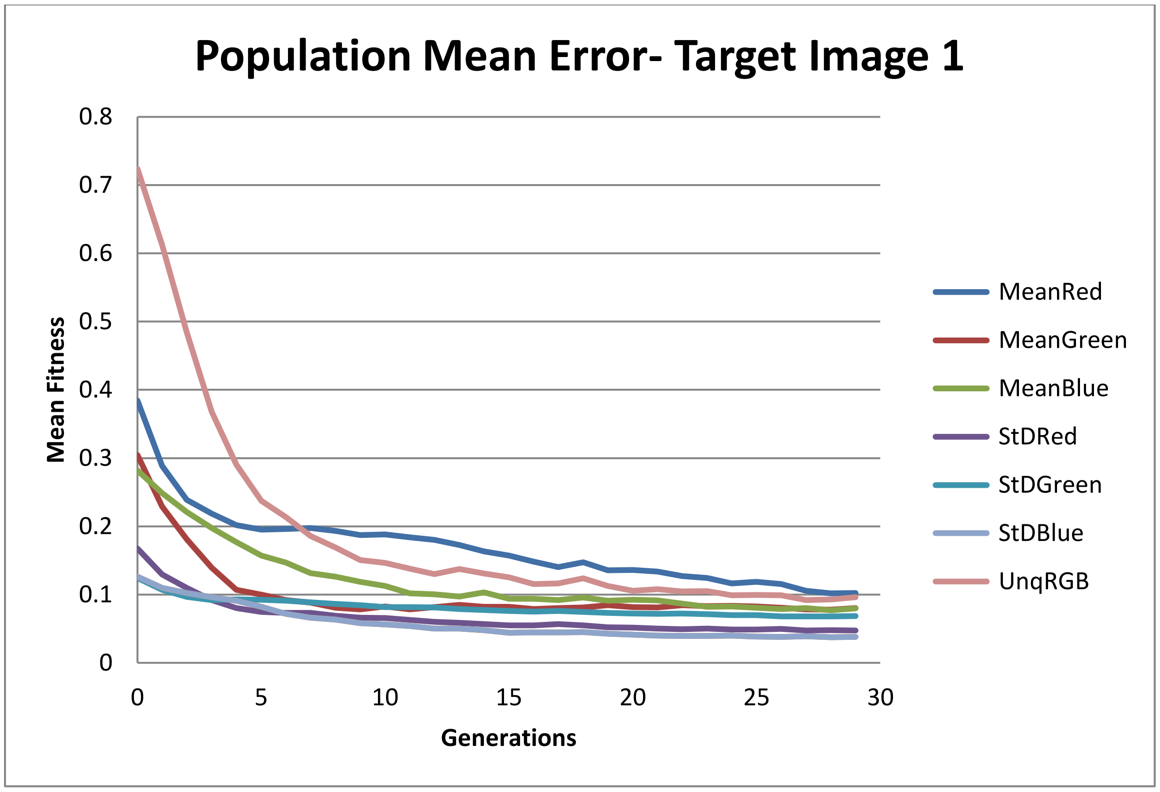


Figure 25: Population mean error for target image 1 for 10 runs - reduced set 1.

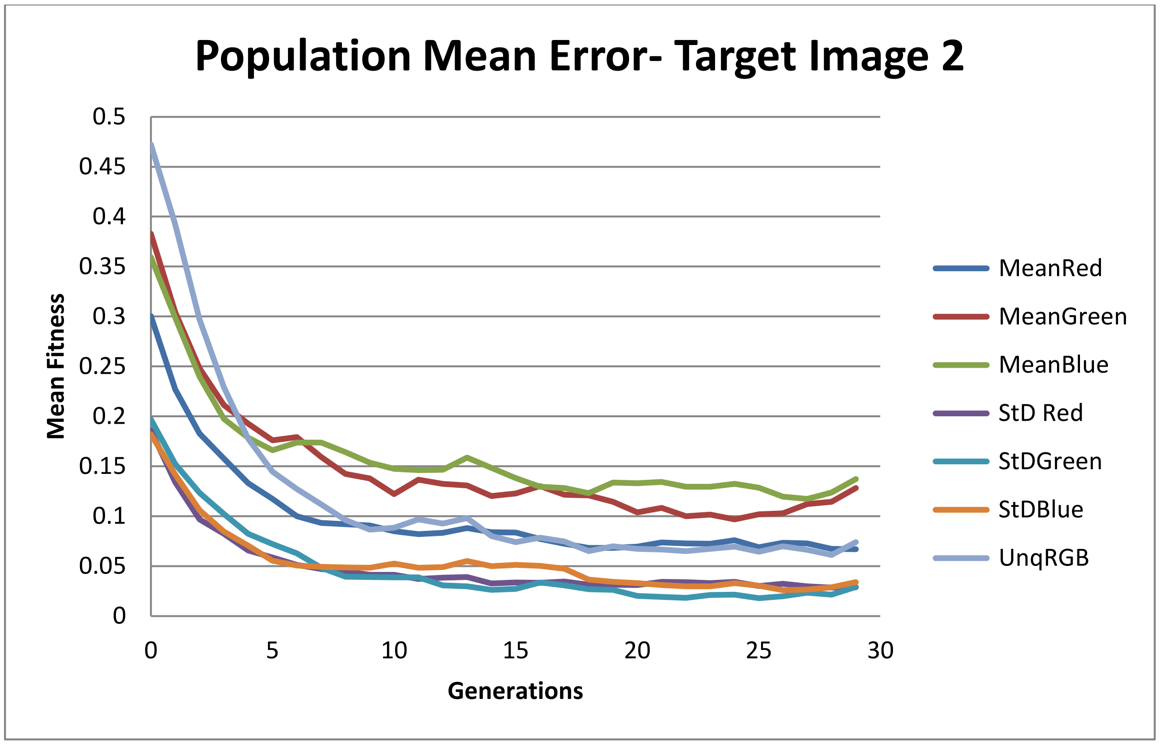


Figure 26: Population mean error for target image 2 for 10 runs - reduced set 1.

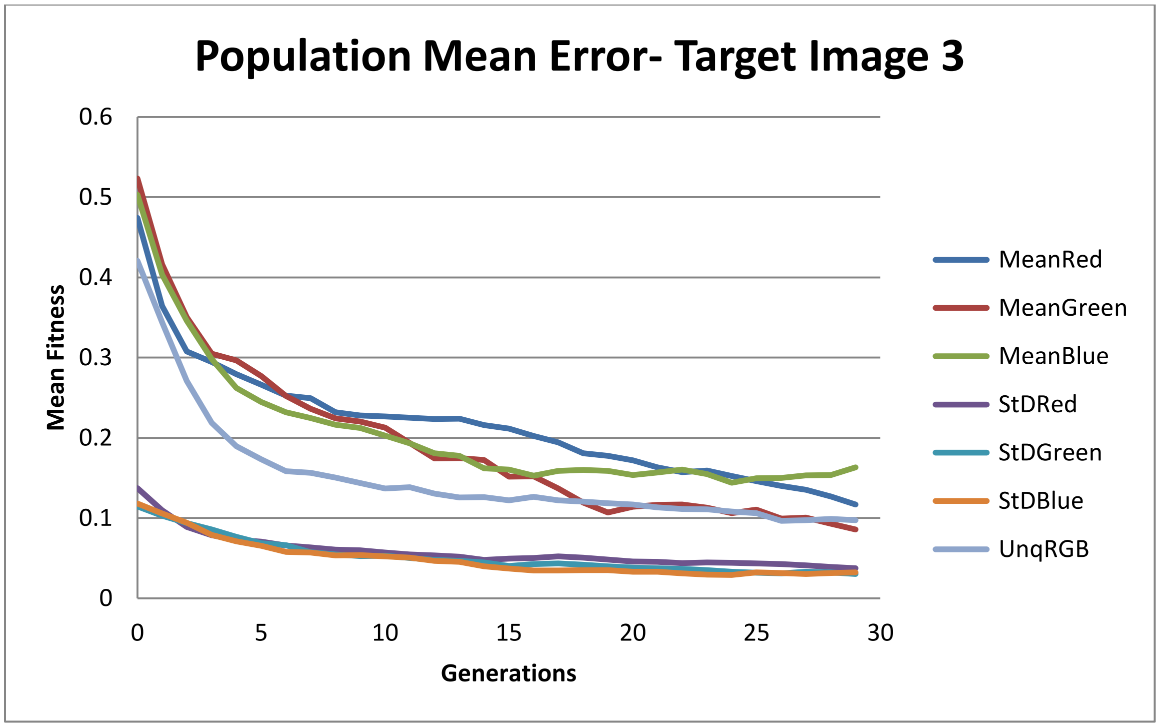


Figure 27: Population mean error for target image 3 for 10 runs - reduced set 1.

According to the plots obtained for these experiments, all the objectives are converging for all target images. Specifically, the number of unique pixels has converged in all images. This helps with distinguishing the number of different colors existing in the target image. Figures 28 to 30 show the list of best solutions of these experiments for all target images.

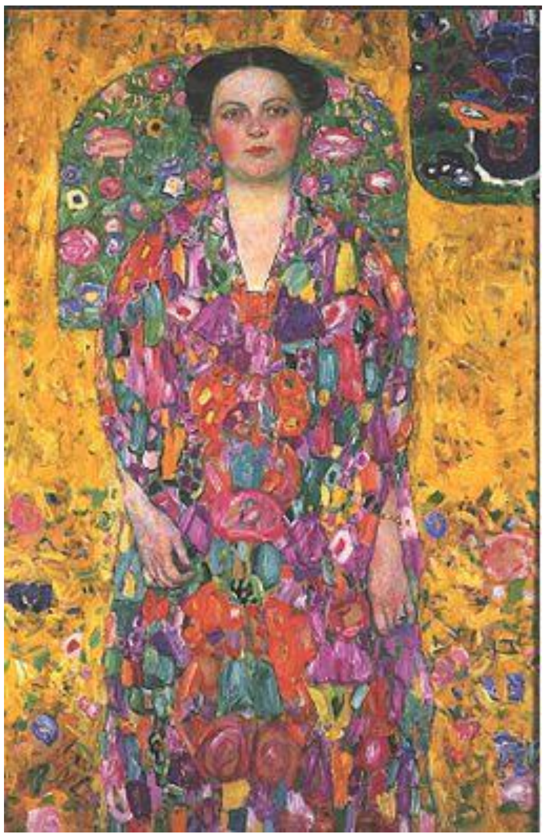
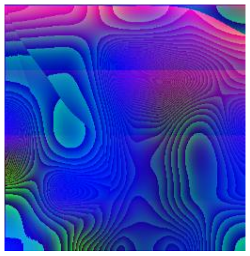
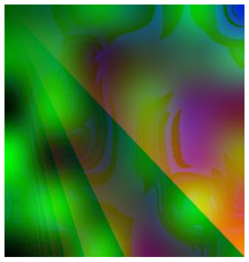
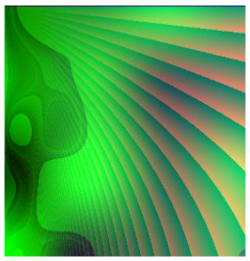
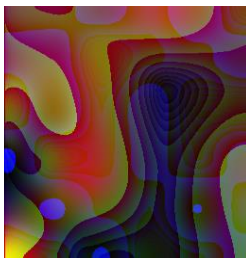
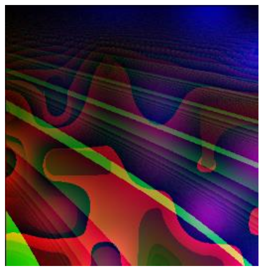
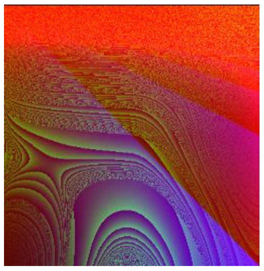
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		

Figure 28: List of best solutions for target image 1for 10 runs - reduced set 1.

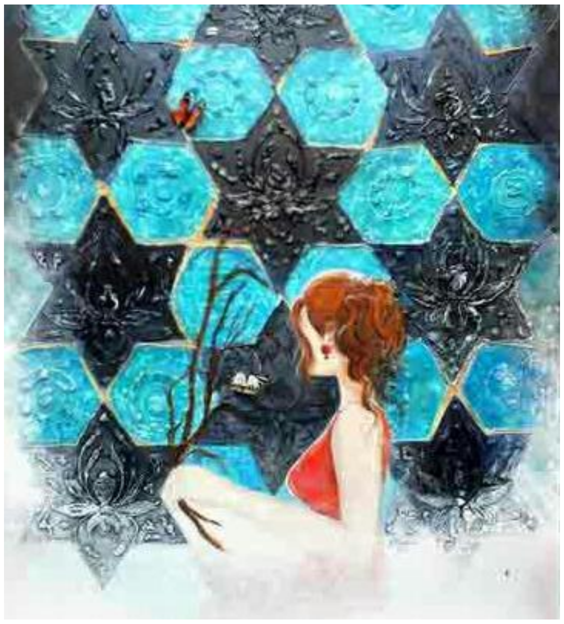
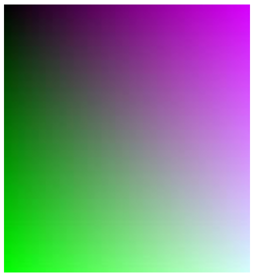
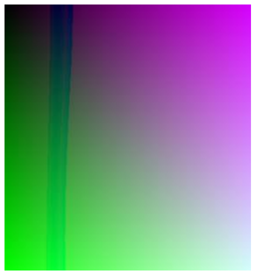
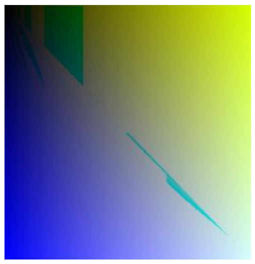
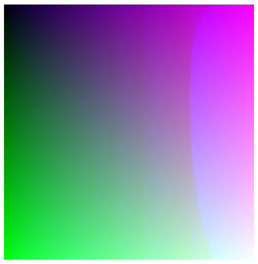
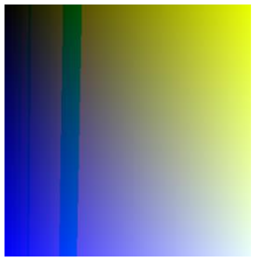
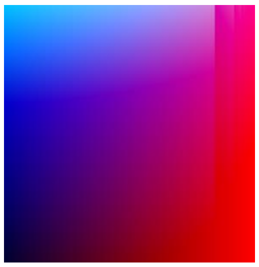
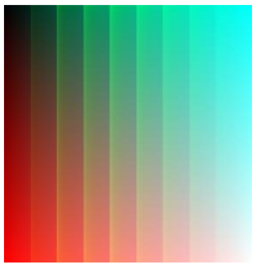
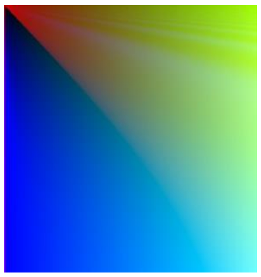
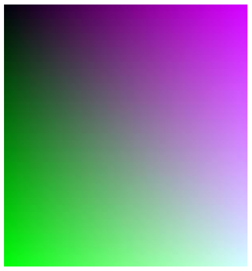
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 29: List of best solutions for target image 2 for 10 runs - reduced set 1.

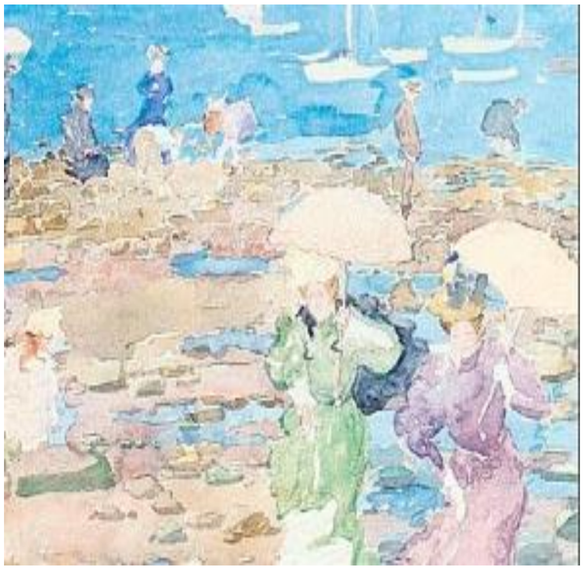
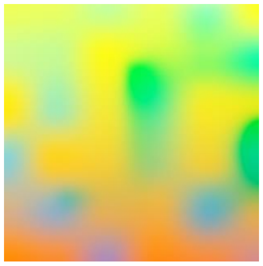
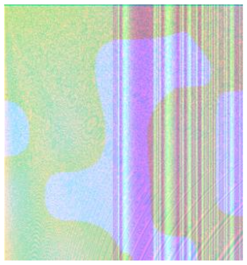
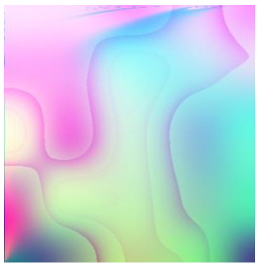
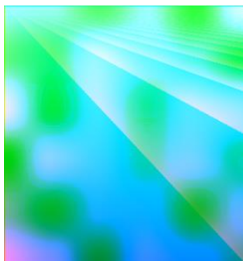
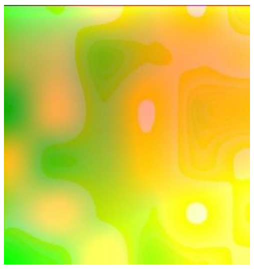
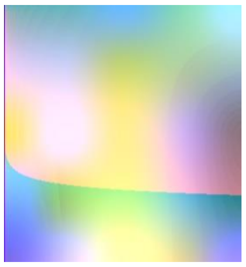
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		

Figure 30: List of best solutions for target image 3 for 10 runs - reduced set 1.

With respect to the plots achieved during the experiments, solutions show a good participation of each objective in the evolution. It is worth mentioning that the best solutions for run 1 and 2 for the target image 2, listed in Figure 29, are not the same. Magnifying both solutions, we figured out that solution in run 1 is more towards purple rather than pink in the right side of the texture and solution in run 2 tends to be brighter in the bottom right corner of the texture than the solution in run 1.

We used T-test analysis to compare the scores of features in the reduced set experiments for target image 1. We used a two-tail test with unequal variances and $P \leq 0.05$. There were no statistical significant difference in average values except for standard deviation green ($P = 0.00084$), which was superior in full set 1.

6.2.2 Reduced Set 2

The set of features chosen for these experiments are mean (H, S, V), standard deviation (H, S, V), intensity, entropy, and number of edge pixels. Plots obtained for these experiments show that all objectives all fully converging towards the base. Sobel edge pixel plots in the first two charts have risen a bit but converged at the end. The best convergence has happened for entropy, intensity, mean sat, and mean value for all targets.

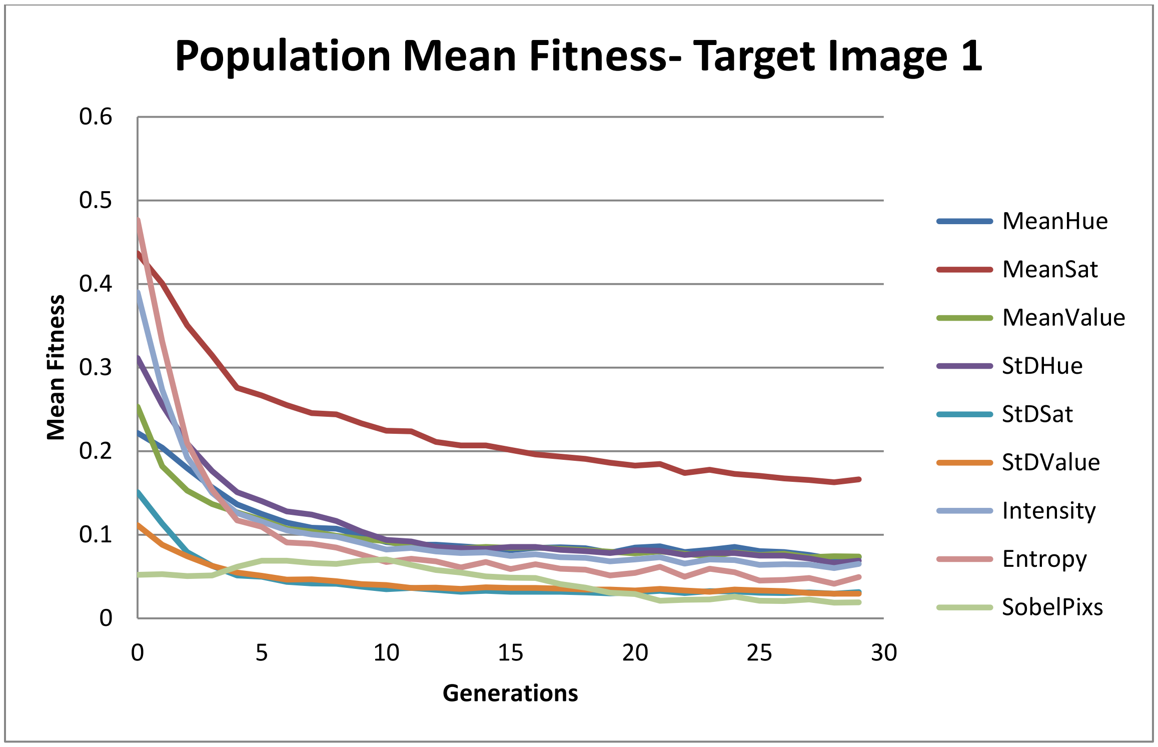


Figure 31: Population mean error for target image 1for 10 runs - reduced set 2.

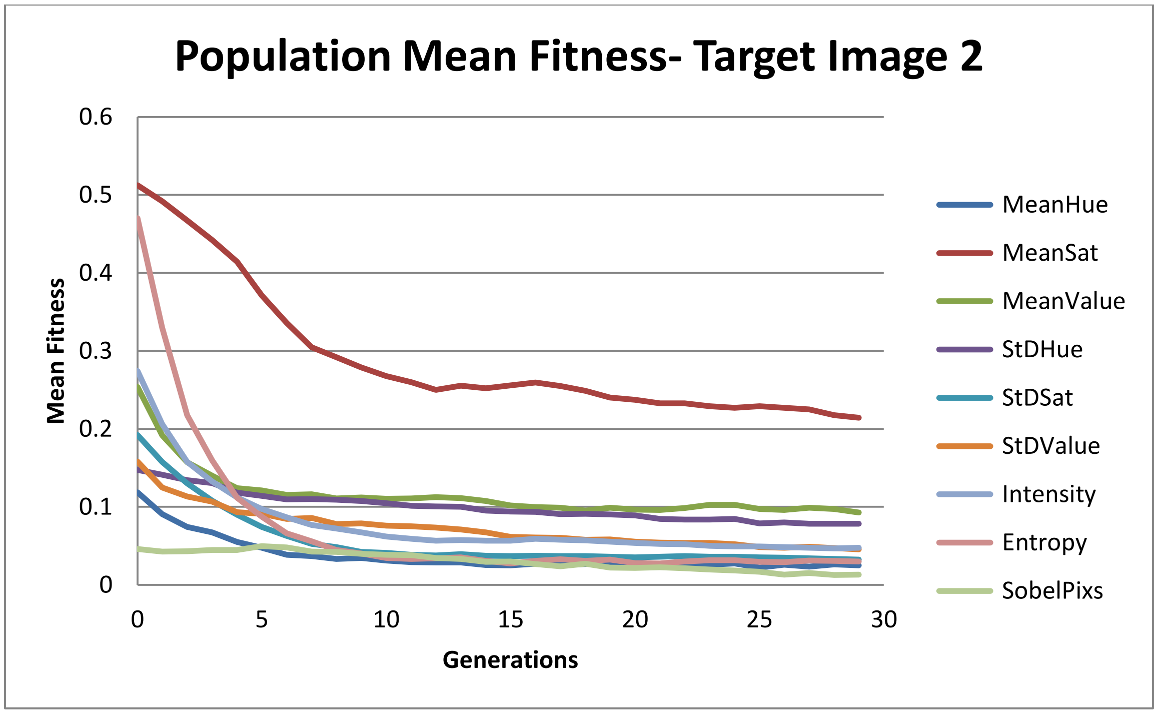


Figure 32: Population mean error for target image 2 for 10 runs - reduced set 2.

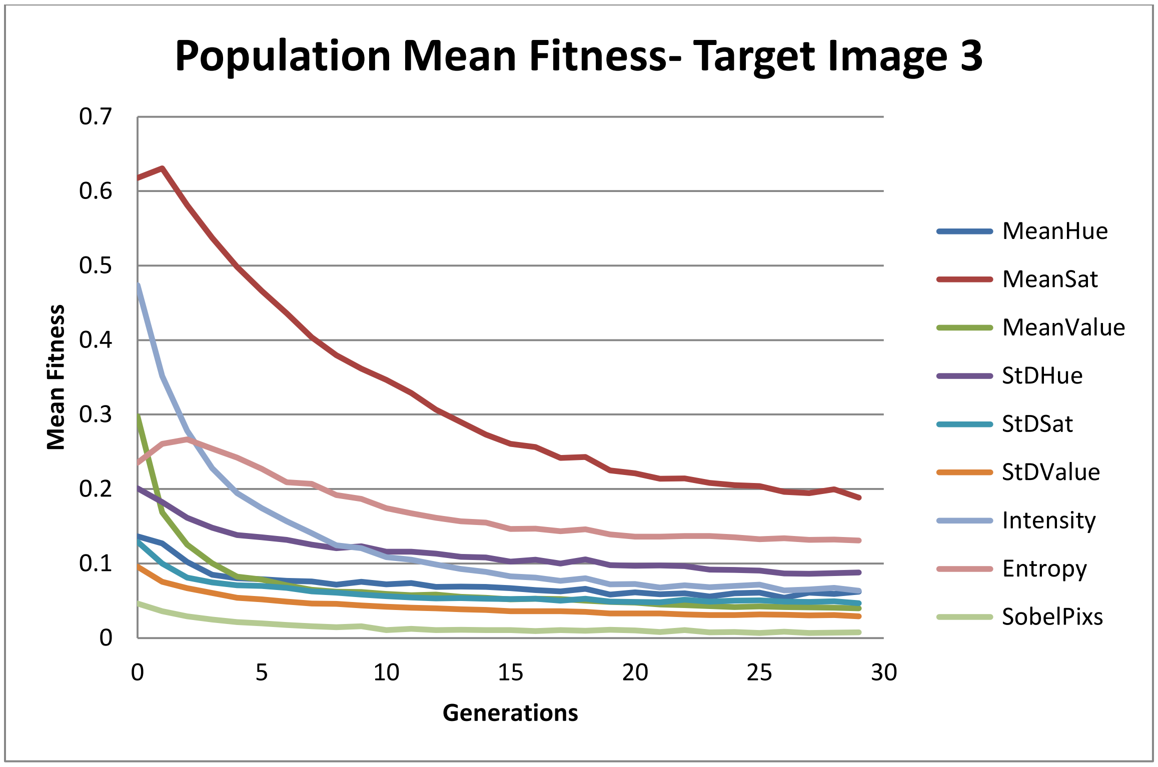


Figure 33: Population mean error for target image 3 for 10 runs - reduced set 2.

Figures 34 to 36 show best solutions for this set of experiments.

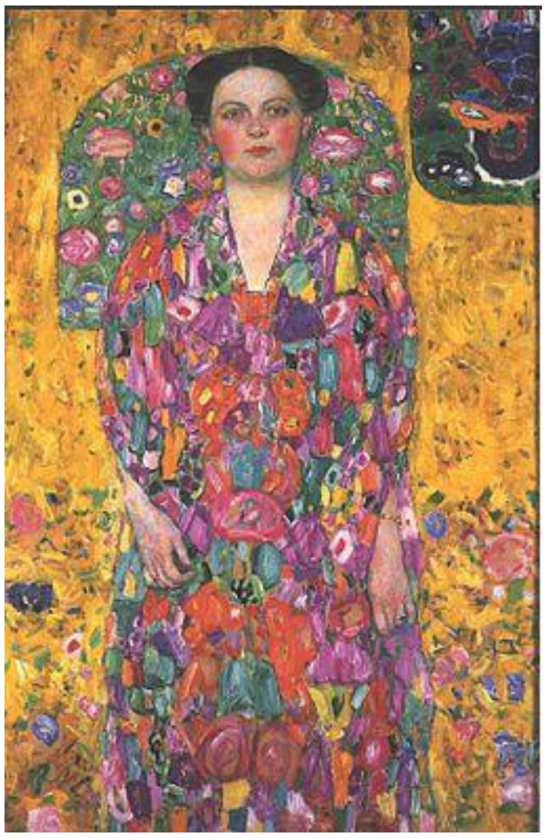
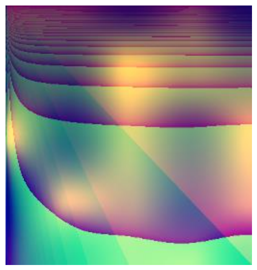
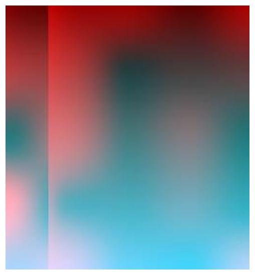
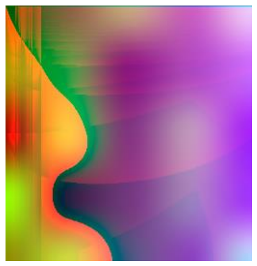
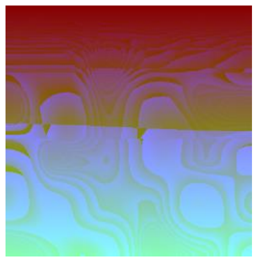
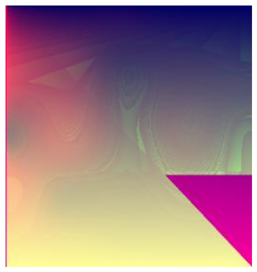
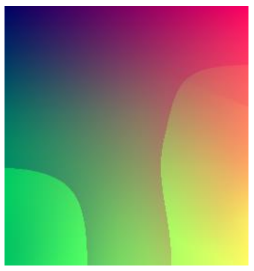
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		

Figure 34: List of best solutions for target image 1for 10 runs - reduced set 2.

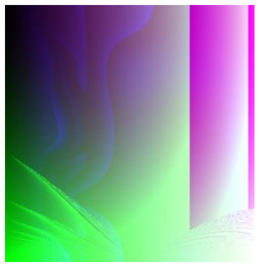
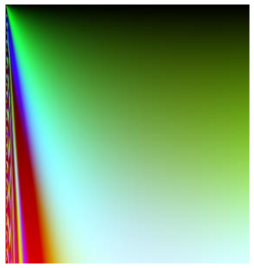
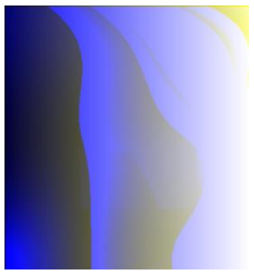
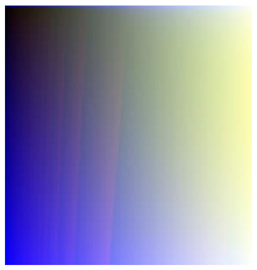
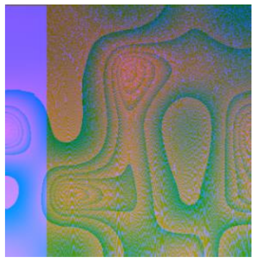
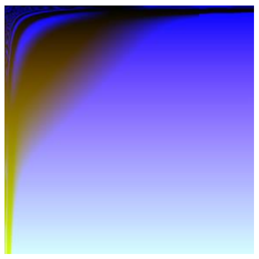
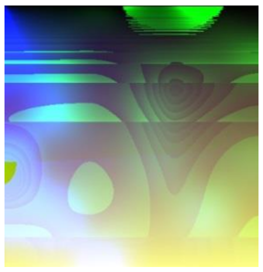
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 35: List of best solutions for target image 2 for 10 runs - reduced set 2.

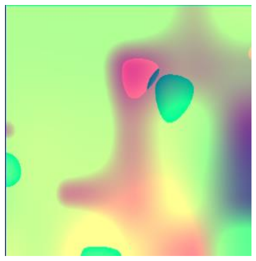
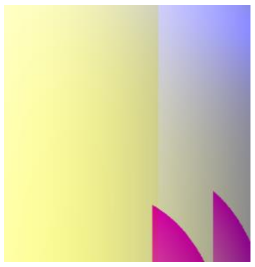
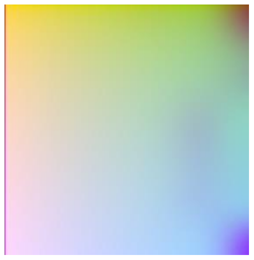
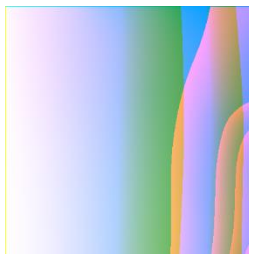
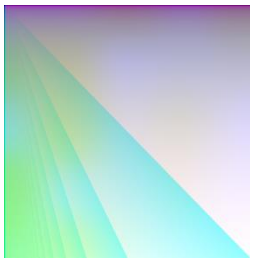
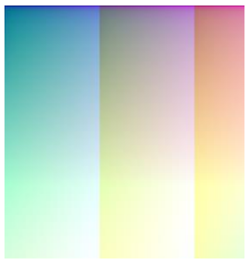
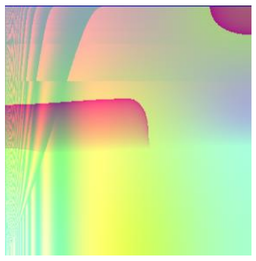
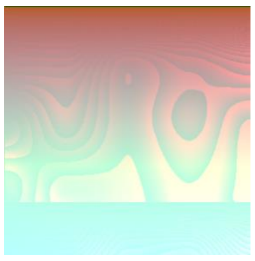
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 36: List of best solutions for target image 3 for 10 runs - reduced set 2.

Similar to the previous experiment set, it can be inferred from the lists of figures shown above that a reduced set of objectives gives us close results to the results of full set experiments. Moreover, fitness plots in both sets have improved for all used features.

We used T-test analysis to compare the scores of features in the reduced set experiments. We used a two-tail test with unequal variances and $P \leq 0.05$. There were no statistical significant difference in average values for this comparison and the only closest one was Sobel pixels with $P = 0.07$ which was superior in reduced set 2.

6.3 Mixed Sets

We decided to mix selected features (mean, standard deviation, unique pixels, intensity, entropy, and Sobel edge) of the two sets. What we expected to see is a proper convergence of plots similar to the previous experiments (reduction sets) as well as generating solutions close to the target images in terms of colors. We also wanted to compare RGB and HSV color models as the mean dependent variables, to see if there are noticeable difference in results. Therefore, pixel values and some simple shape features are tested to see how different the results will be from the full set.

6.3.1 Mixed Set 1

This set includes mean (R, G, B), standard deviation (R, G, B), unique pixels from the first set and intensity, entropy, and Sobel edge pixels from the second set. Error plots are shown in Figures 37 to 39 and best solutions for these experiments are shown in Figures 40 to 42.

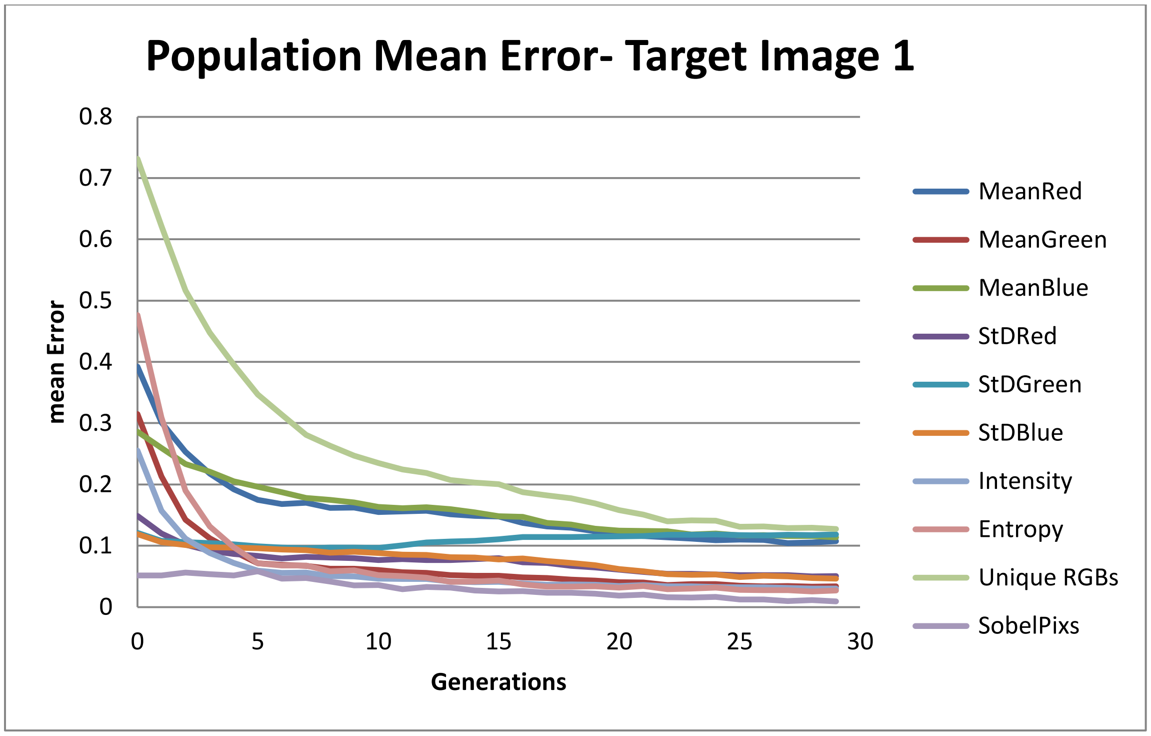


Figure 37: Population mean error for target image 1 for 10 runs - mixed set 1.

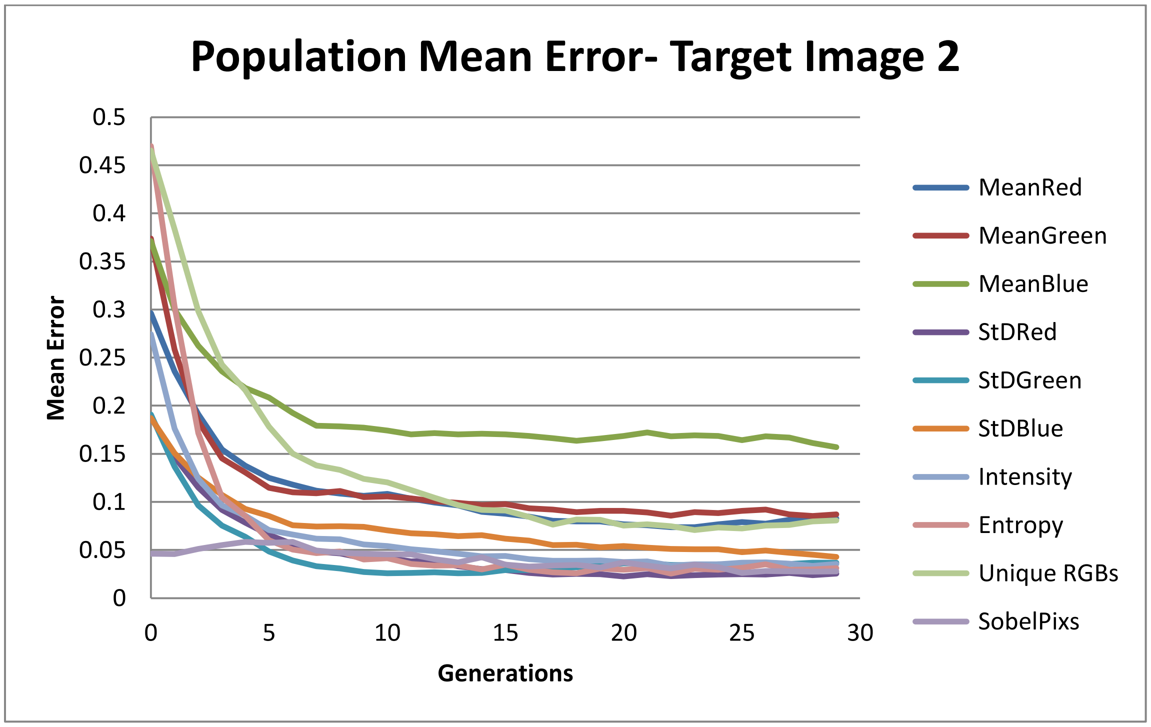


Figure 38: Population mean error for target image 2 for 10 runs - mixed set 1.

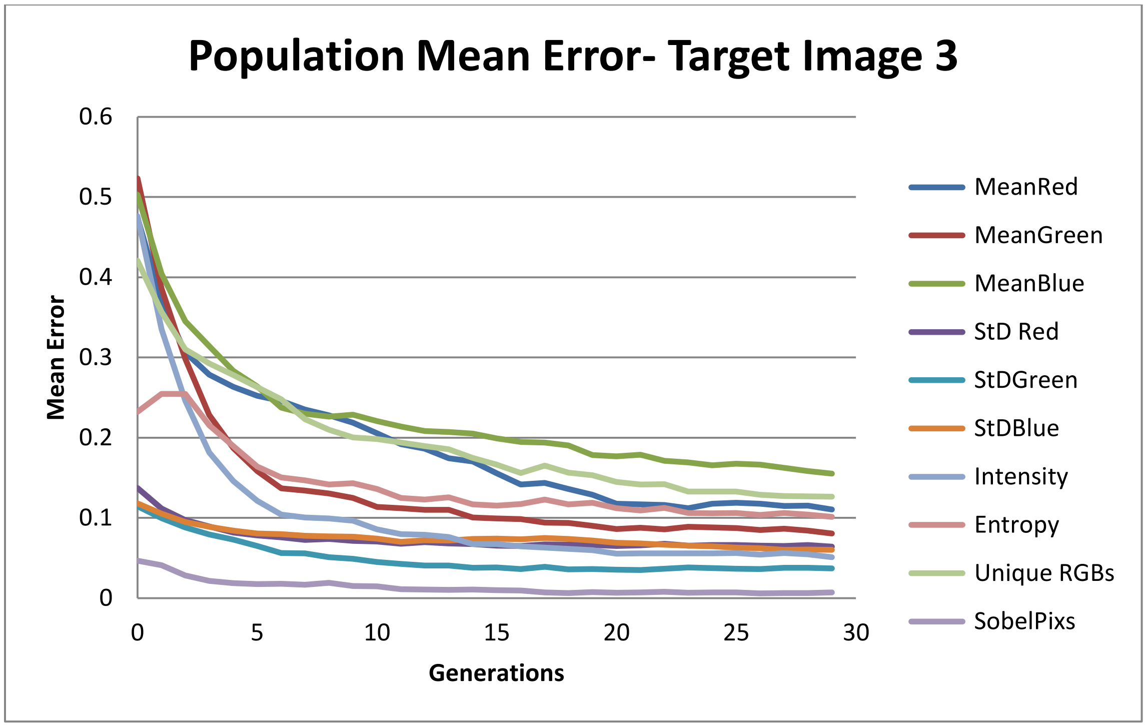


Figure 39: Population mean error for target image 3 for 10 runs - mixed set 1.

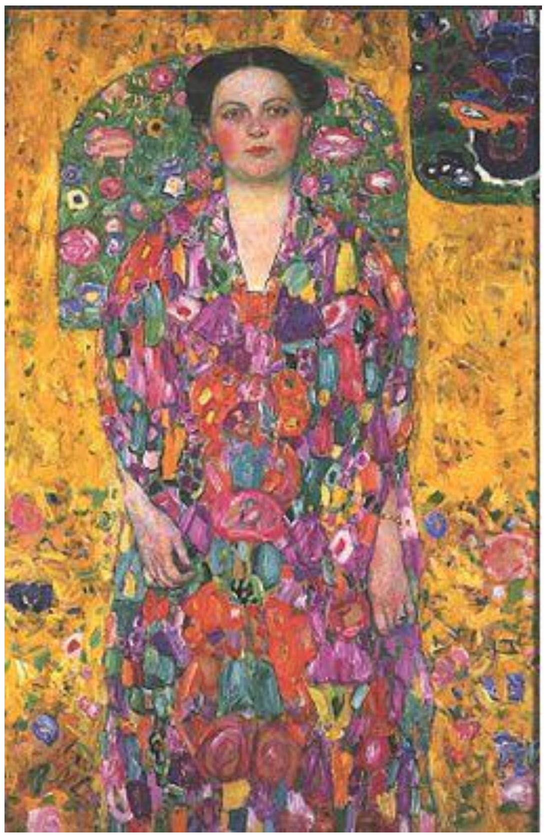
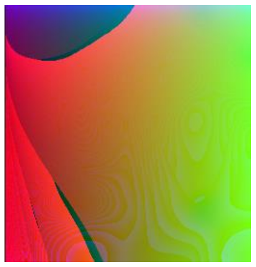
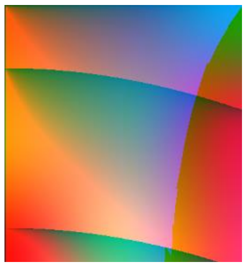
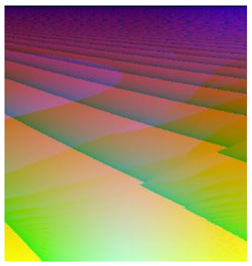
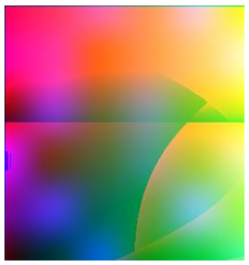
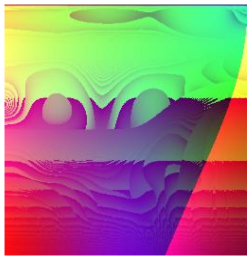
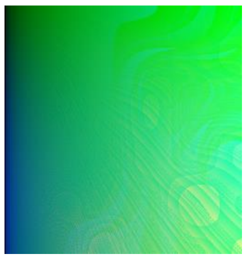
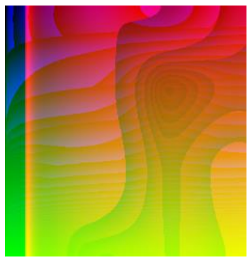
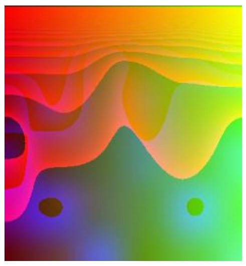
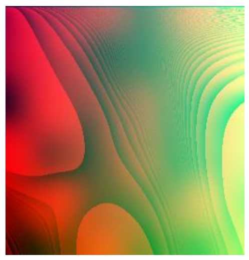
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 40: List of best solutions for target image 1for 10 runs - mixed set 1.

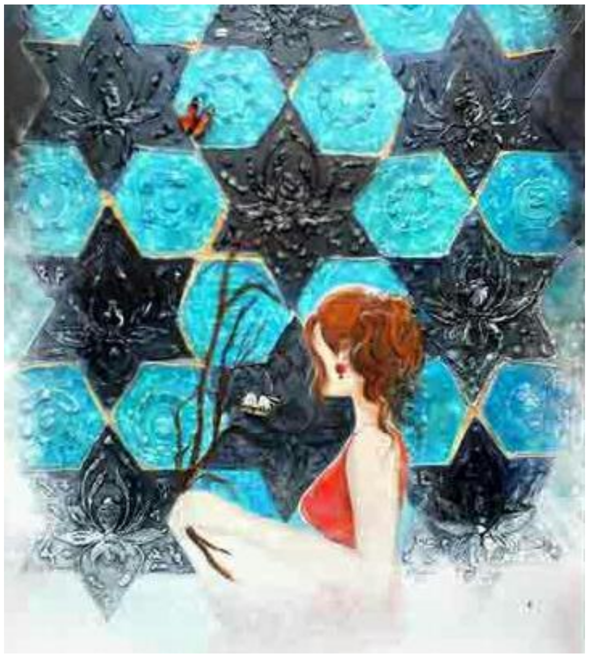
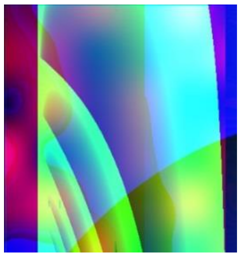
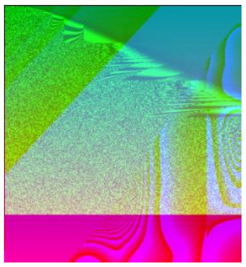
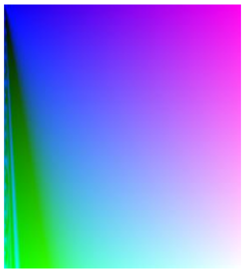
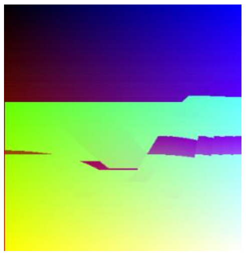
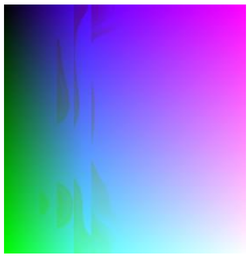
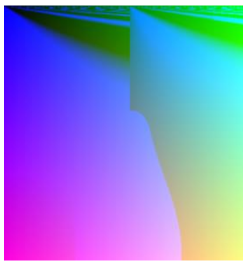
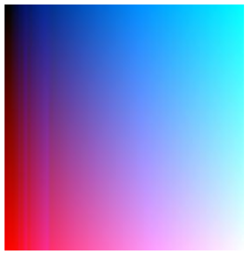
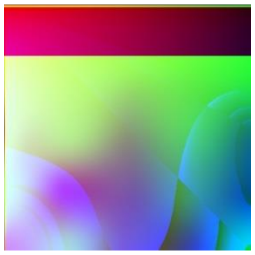
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 41: List of best solutions for target image 2 for 10 runs - mixed set 1.

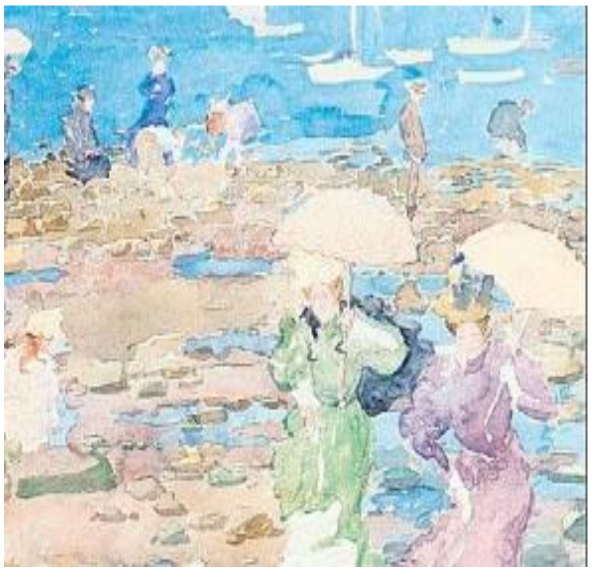
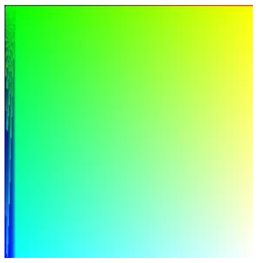
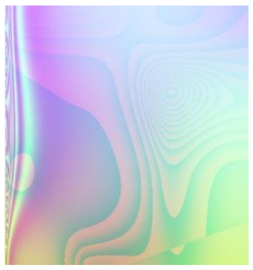
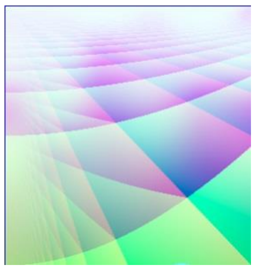
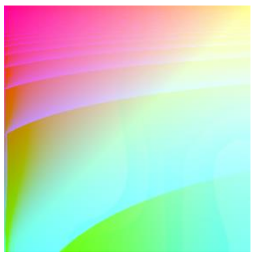
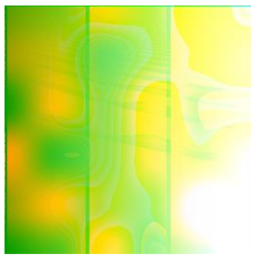
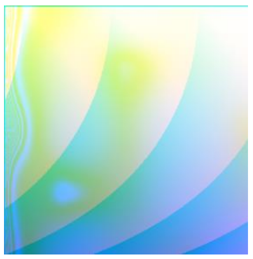
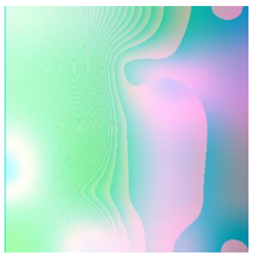
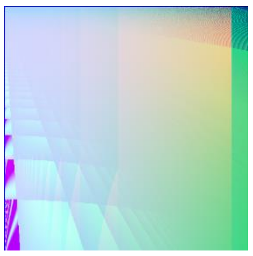
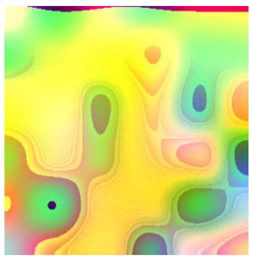
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 42: List of best solutions for target image 3 for 10 runs - mixed set 1.

These experiments did not produce visually improved results, especially for the first and second target images. An assumption for having these results might rely on the fact that having higher entropy and only pixel values cannot be strong enough to create that interesting results. Because in these experiments all the information that the system owns are mean and standard deviation of pixel values and the rest are counting pixels (unique and edge) and a bit of brightness. It can be inferred that having higher entropy with these features can cause complications.

6.3.2 Mixed Set 2

The second set includes mean (H, S, V), standard deviation (H, S, V), intensity, entropy, and Sobel edge pixels from second set and unique RGBs from the first set. In other words, HSV color model replaces RGBs. Figures 43 to 45 show plots obtained for three target images. The plots show a great convergence for all of the objectives specifically mean sat, mean value, unique pixels, std sat, std hue, and intensity. Sobel edge pixels, std value, and std sat have converged very close to the base line. Figures 46 to 48 show lists of best solutions for experiments of this set on three target images.

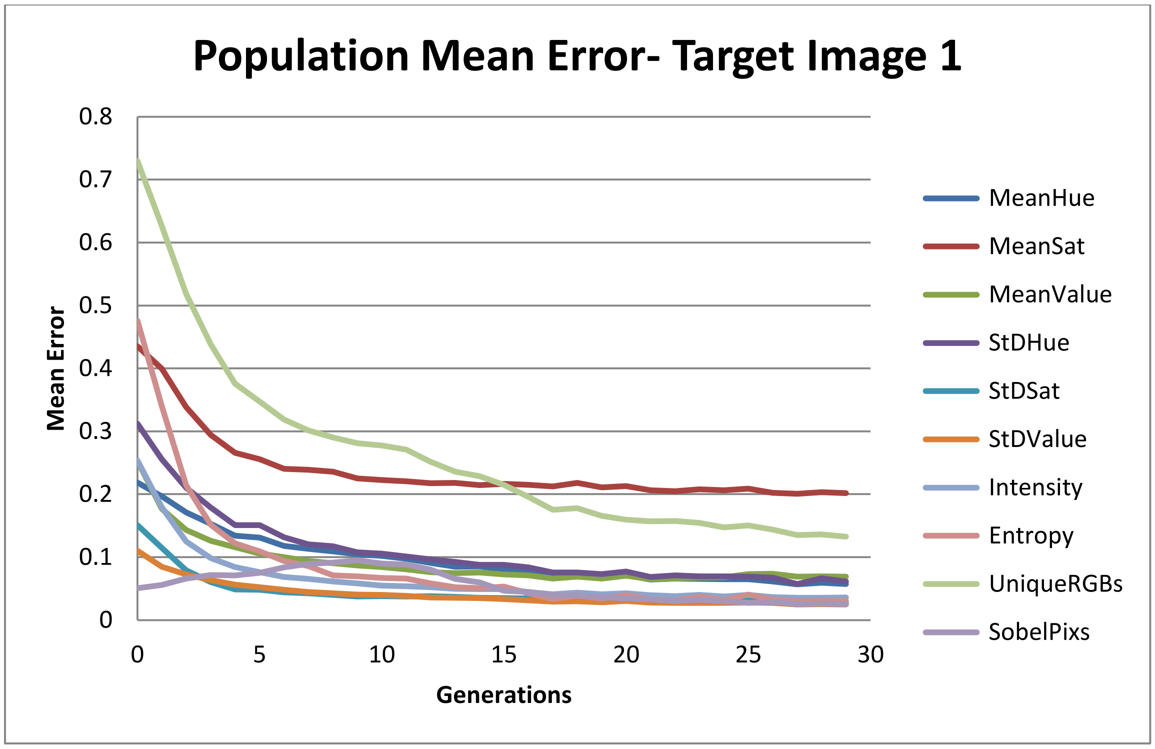


Figure 43: Population mean error for target image 1 for 10 runs - mixed set 2.

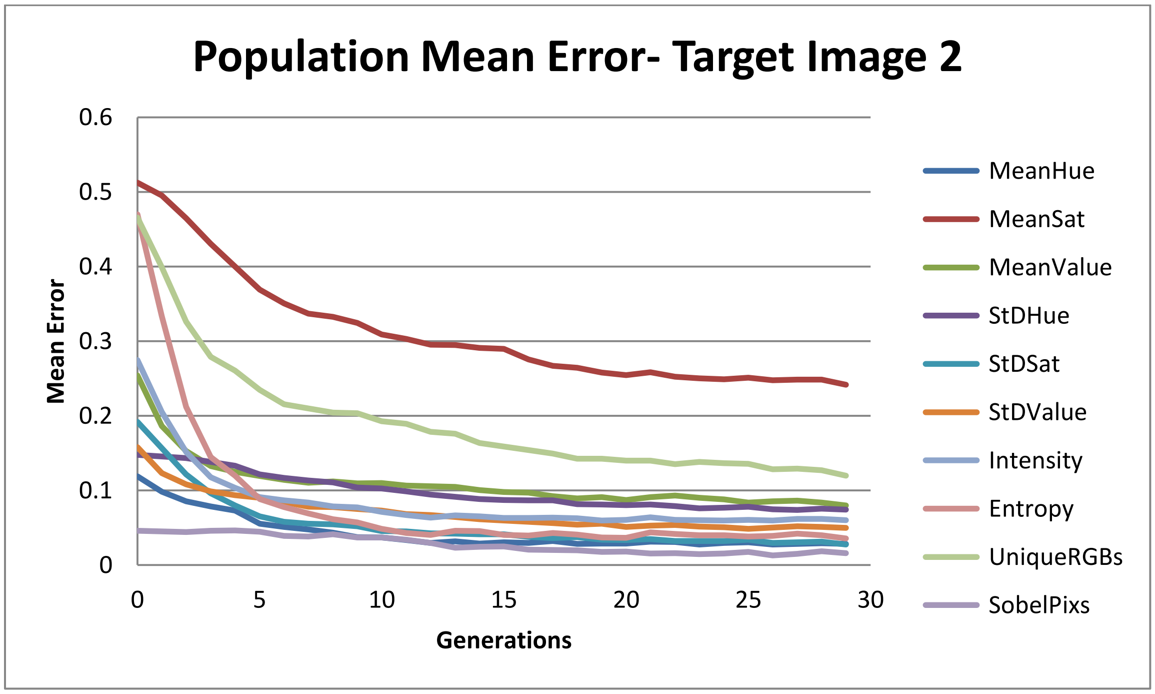


Figure 44: Population mean error for target image 2 for 10 runs - mixed set 2.

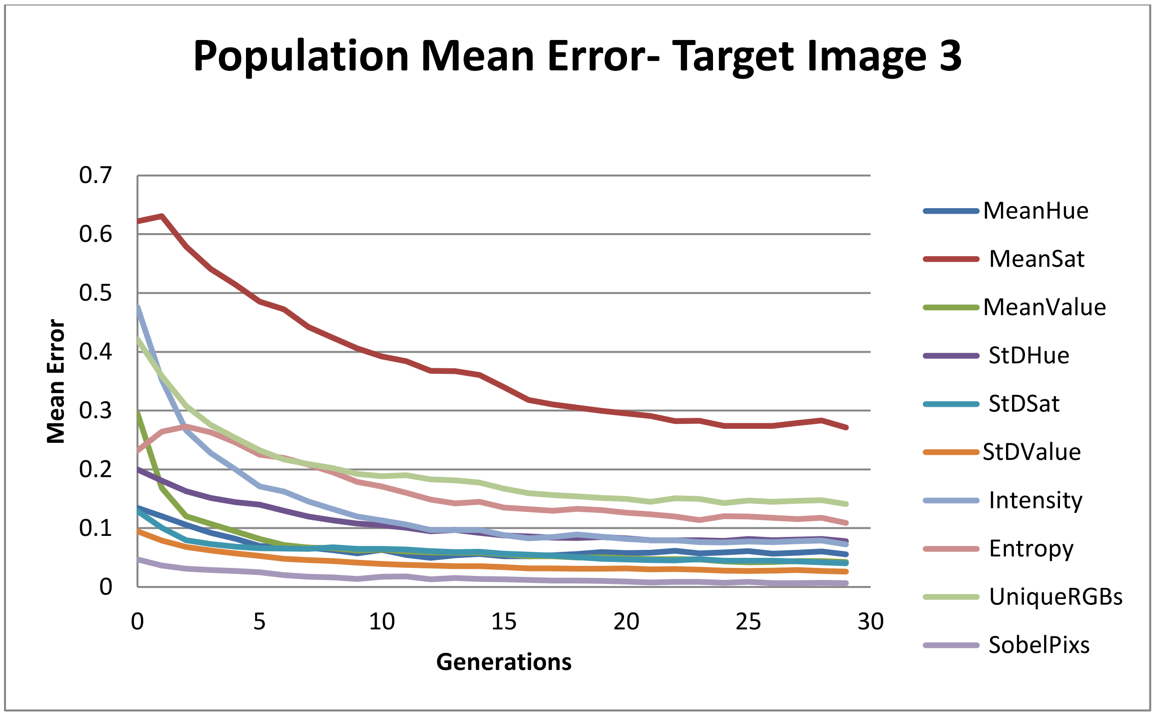


Figure 45: Population mean error for target image 3 for 10 runs - mixed set 2.

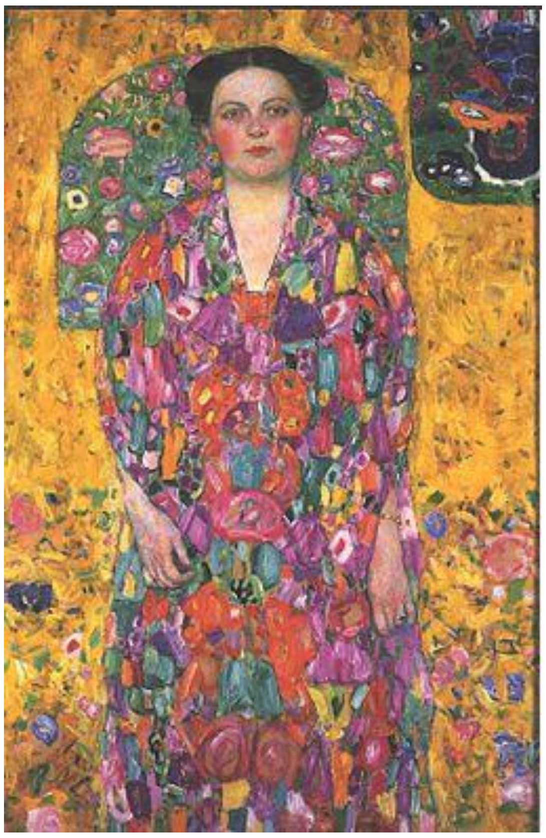
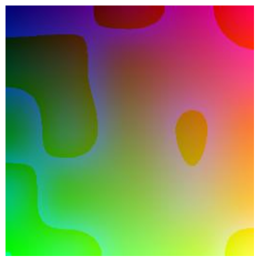
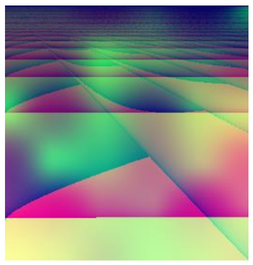
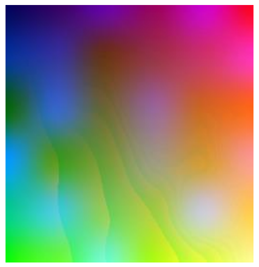
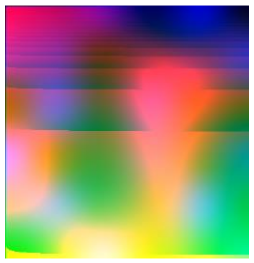
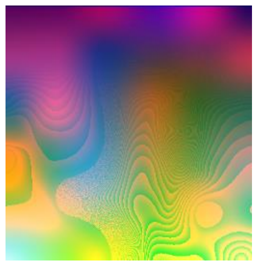
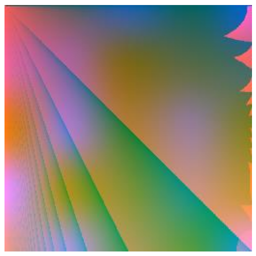
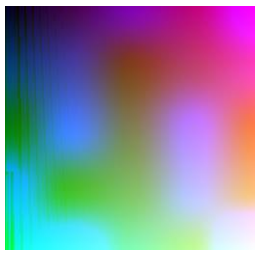
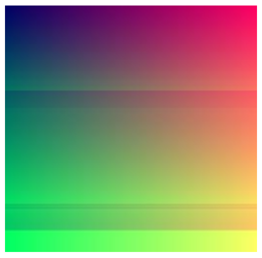
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 46: List of best solutions for target image 1for 10 runs - mixed set 2.

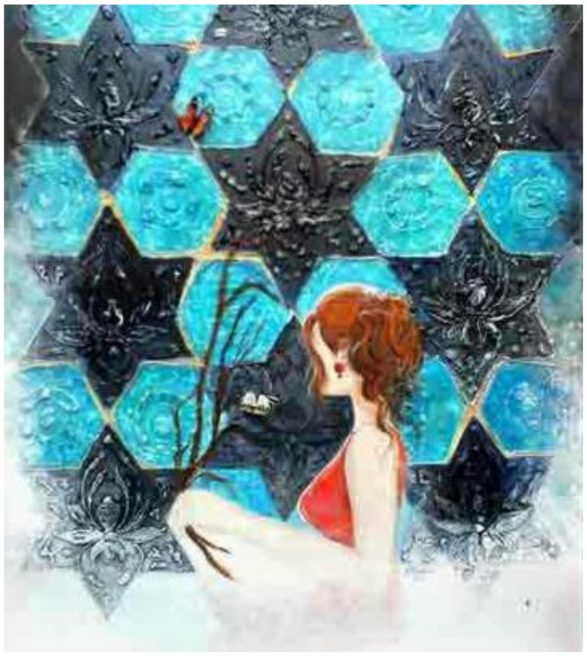
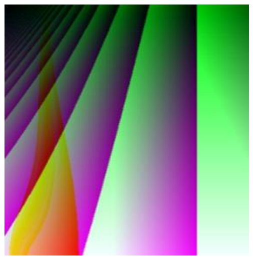
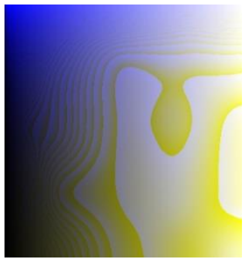
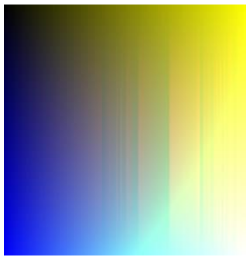
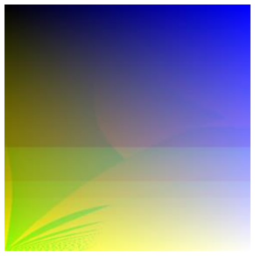
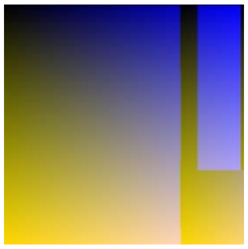
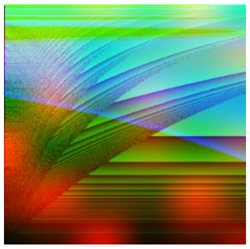
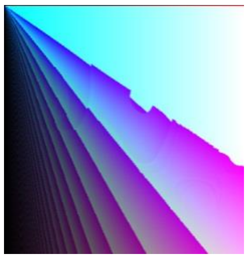
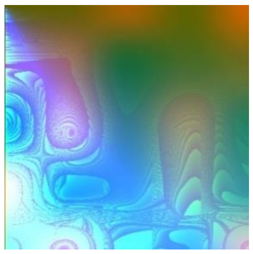
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 47: List of best solutions for target image 2 for 10 runs - mixed set 2.

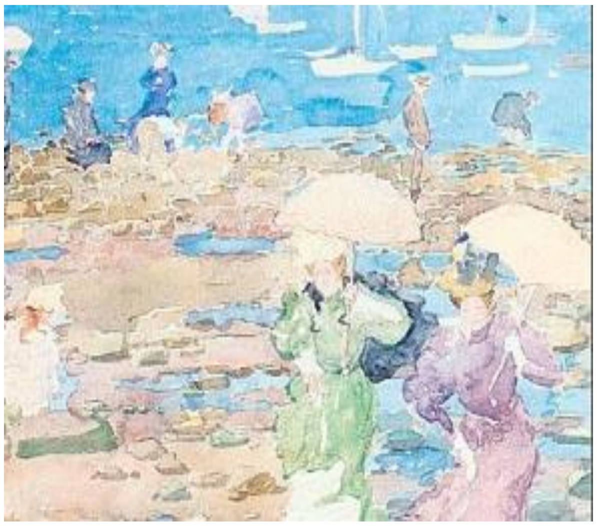
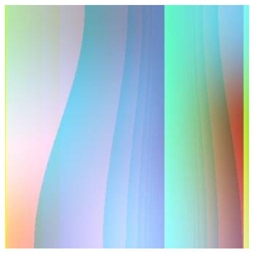
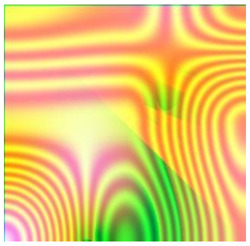
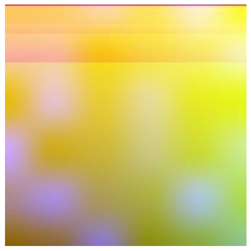
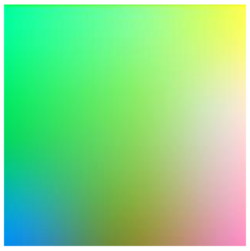
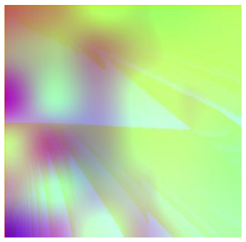
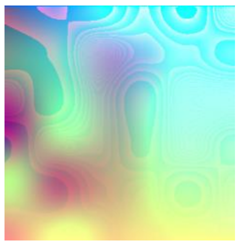
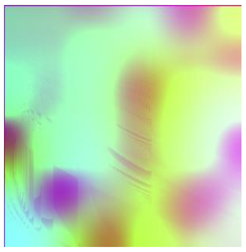
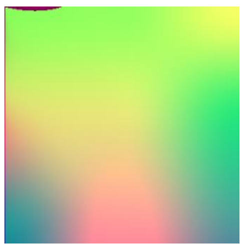
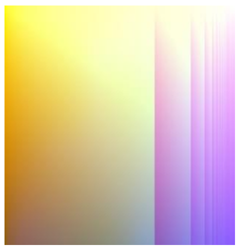
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 48: List of best solutions for target image 3 for 10 runs - mixed set 2.

In our opinion more reasonable color combination can be seen in these solutions than the first set.

6.4 Distance Full Sets

In this set of experiments, the goal is to compare how the system would function without using sum of ranks with when it uses sum of ranks in the first experiment set. Similar to first experiment set, the objectives are divided into two sets. Instead, we use distance formula as used in Lombardi *et al*’s work. So the problem will be a single-objective problem.

6.4.1 Distance Set 1

This set includes objectives such as min (R, G, B), max (R, G, B), mean (R, G, B), standard deviation (R, G, B), and number of unique RGB pixels. The following charts show results for this set on three target images.

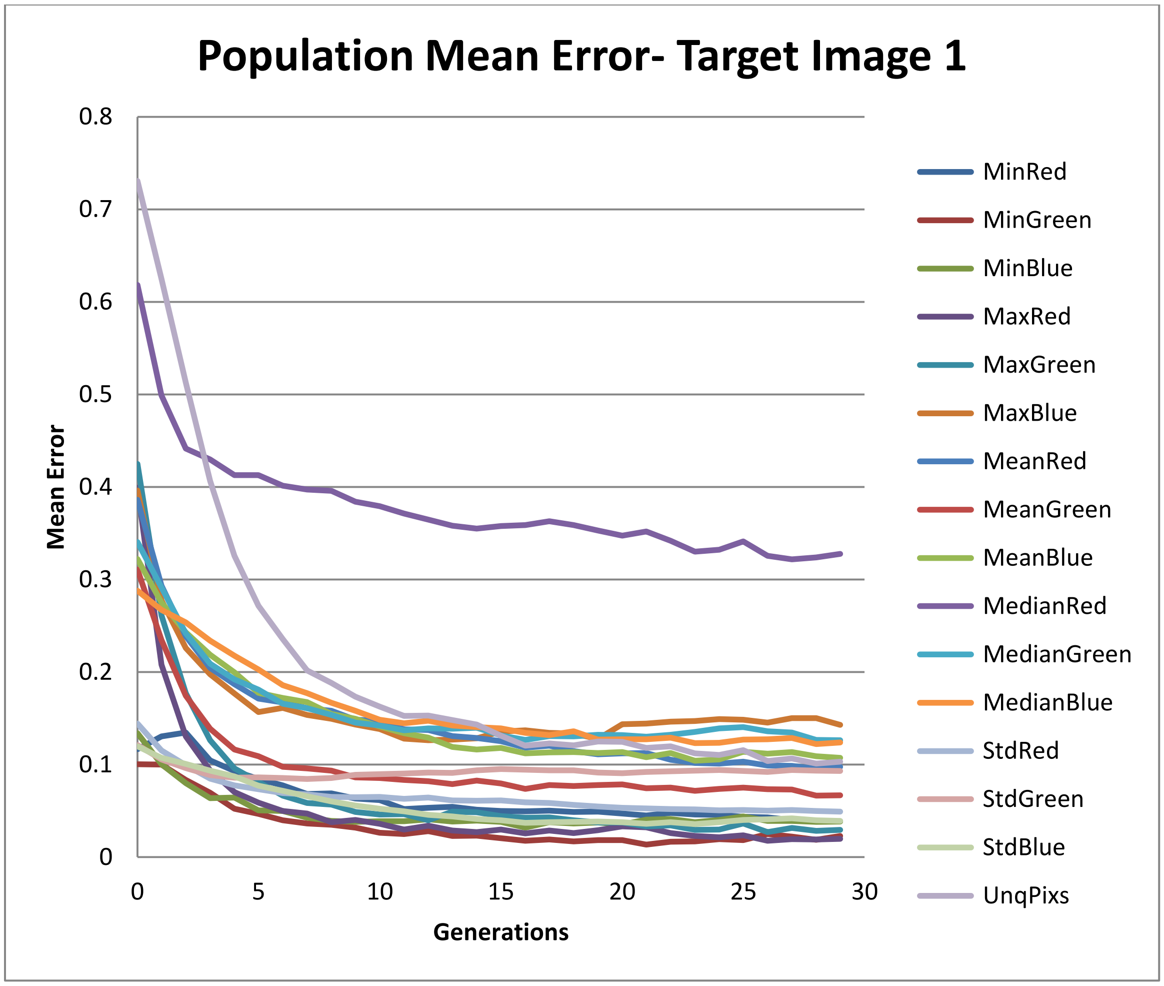


Figure 49: Population mean error for target image 1for 10 runs – distance set 1.

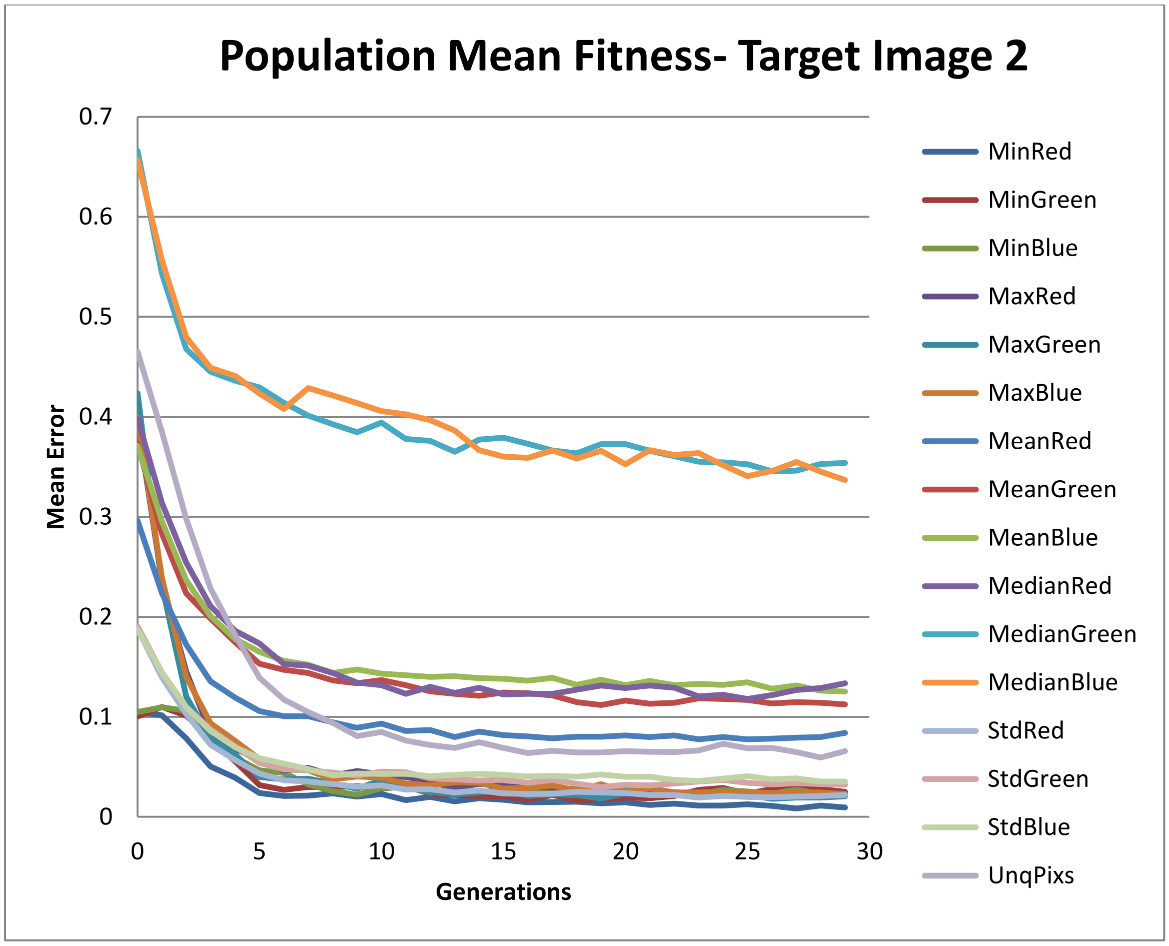


Figure 50: Population mean error for target image 2 for 10 runs – distance set 1.

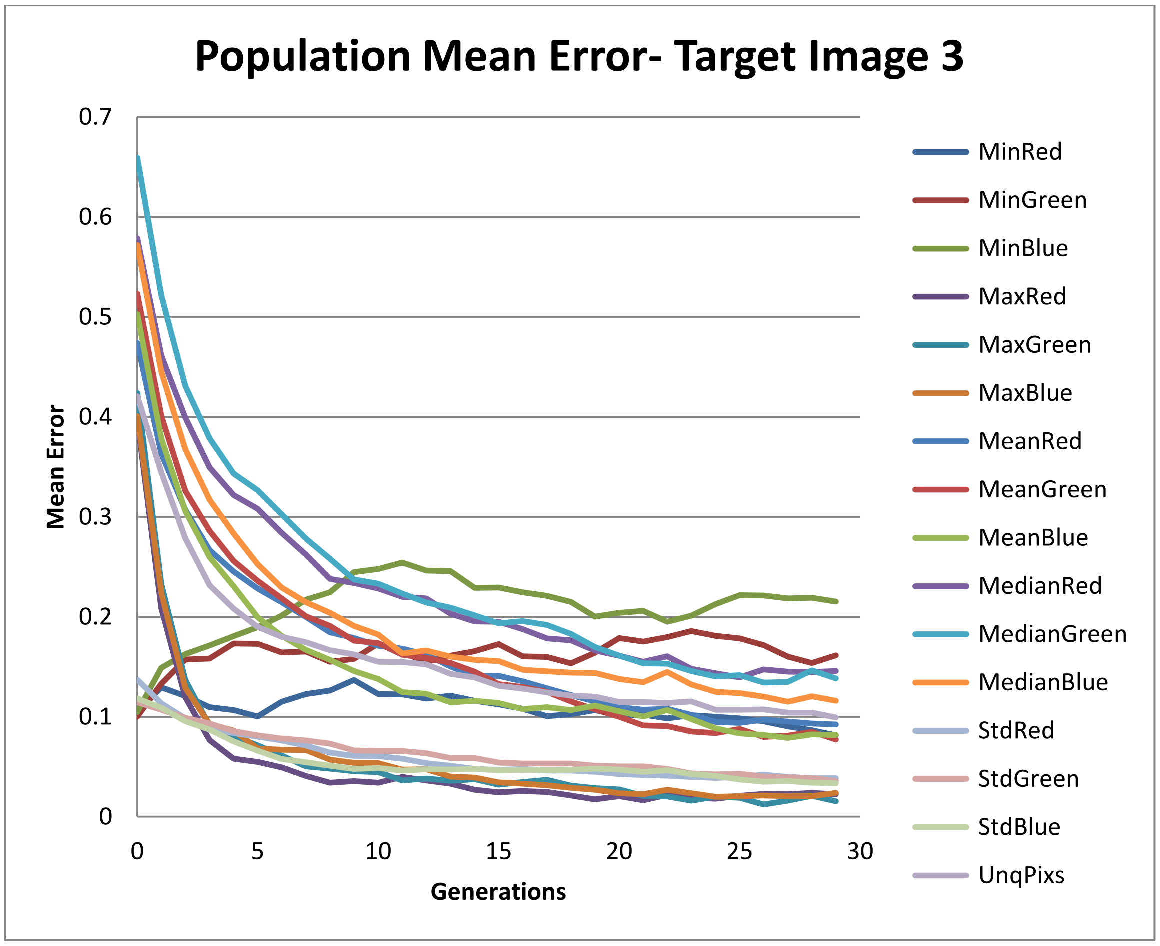


Figure 51: Population mean error for target image 3 for 10 runs – distance set 1.

Plots show that the system in this test has obtained good results. As mentioned earlier, sum of ranks helps the system to rank all objectives in a way that all of them are in an optimal average of fitness and participation. For example in the last target image, considering min blue as a plot that rises in both experiments in full set with sum of ranks and this experiment set of full set without sum of ranks. Min blue in this set rises to a higher level above 0.2 than the other experiment, which is below 0.2 at its highest rise. Solution examples of this experiment set are shown below.

Same as the full set experiments, we used the two-tail T-test to compare statistical results of full set experiments and distance experiments. Similar to previous examination considering $P \leq 0.05$ we only found standard deviation green ($P = 0.007$) to be significantly different in full set 1 experiments.

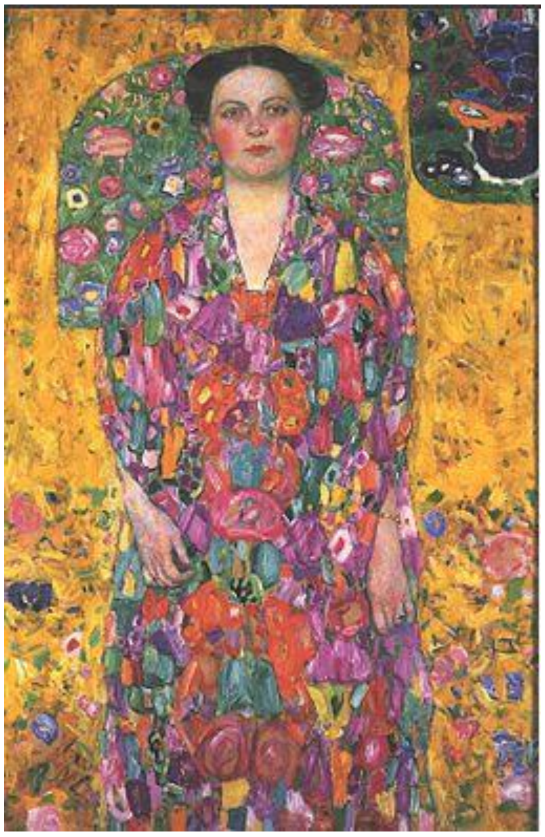
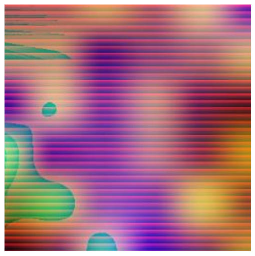
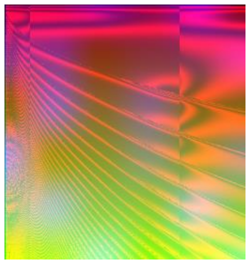
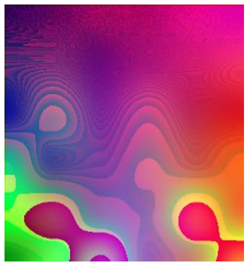
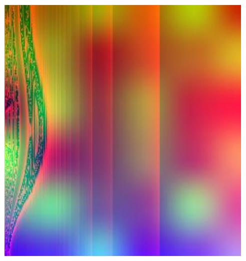
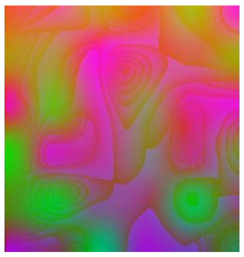
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		

Figure 52: List of best solutions for target image 1 for 10 runs – distance set 1.

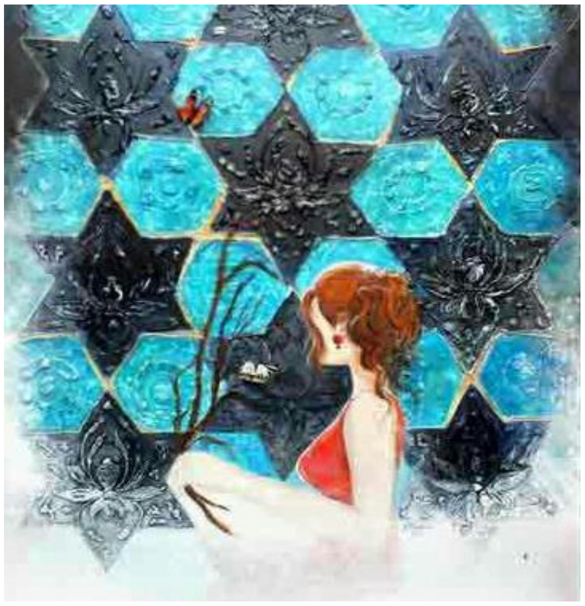
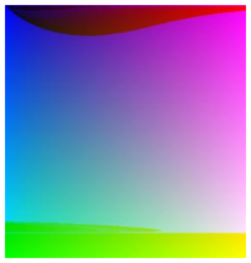
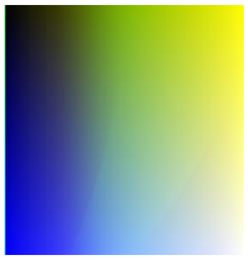
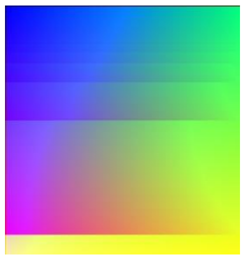
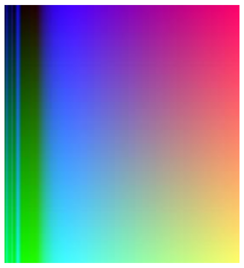
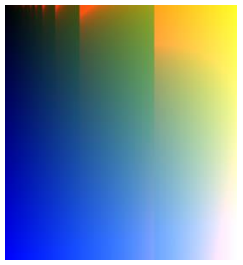
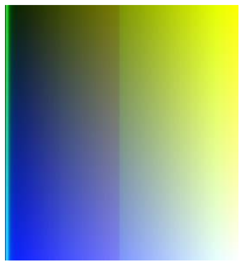
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		

Figure 53: List of best solutions for target image 2 for 10 runs – distance set 1.

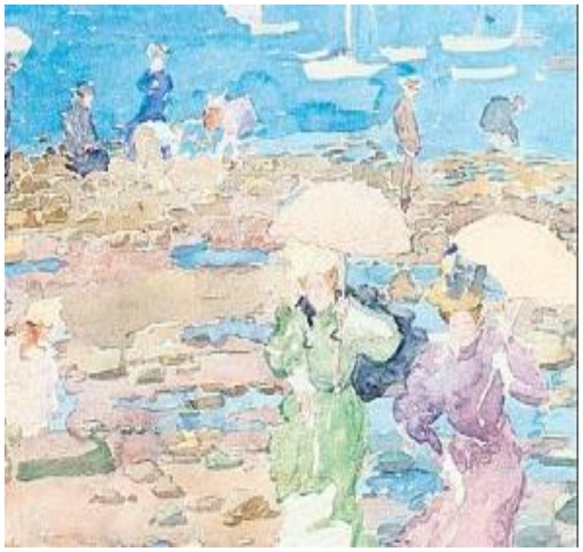
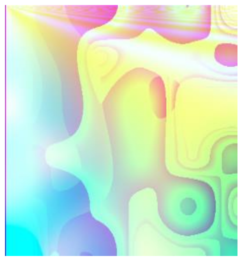
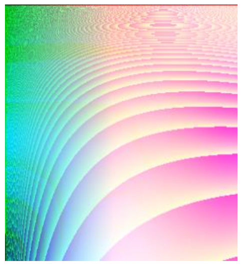
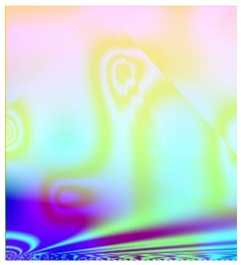
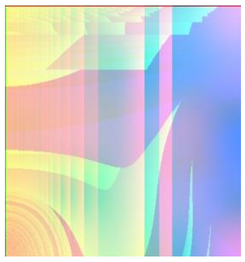
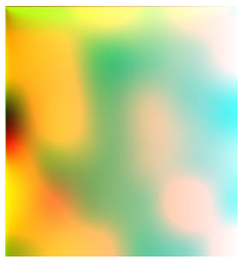
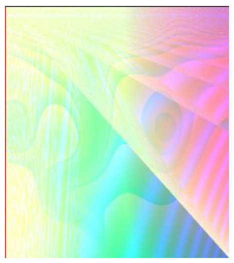
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 54: List of best solutions for target image 3 for 10 runs – distance set 1.

6.4.2 Distance Set 2

This set includes min (H, S, V), max (H, S, V), mean (H, S, V), standard deviation (H, S, V), intensity, entropy, and Sobel edge pixels. The following charts show the results per target image.

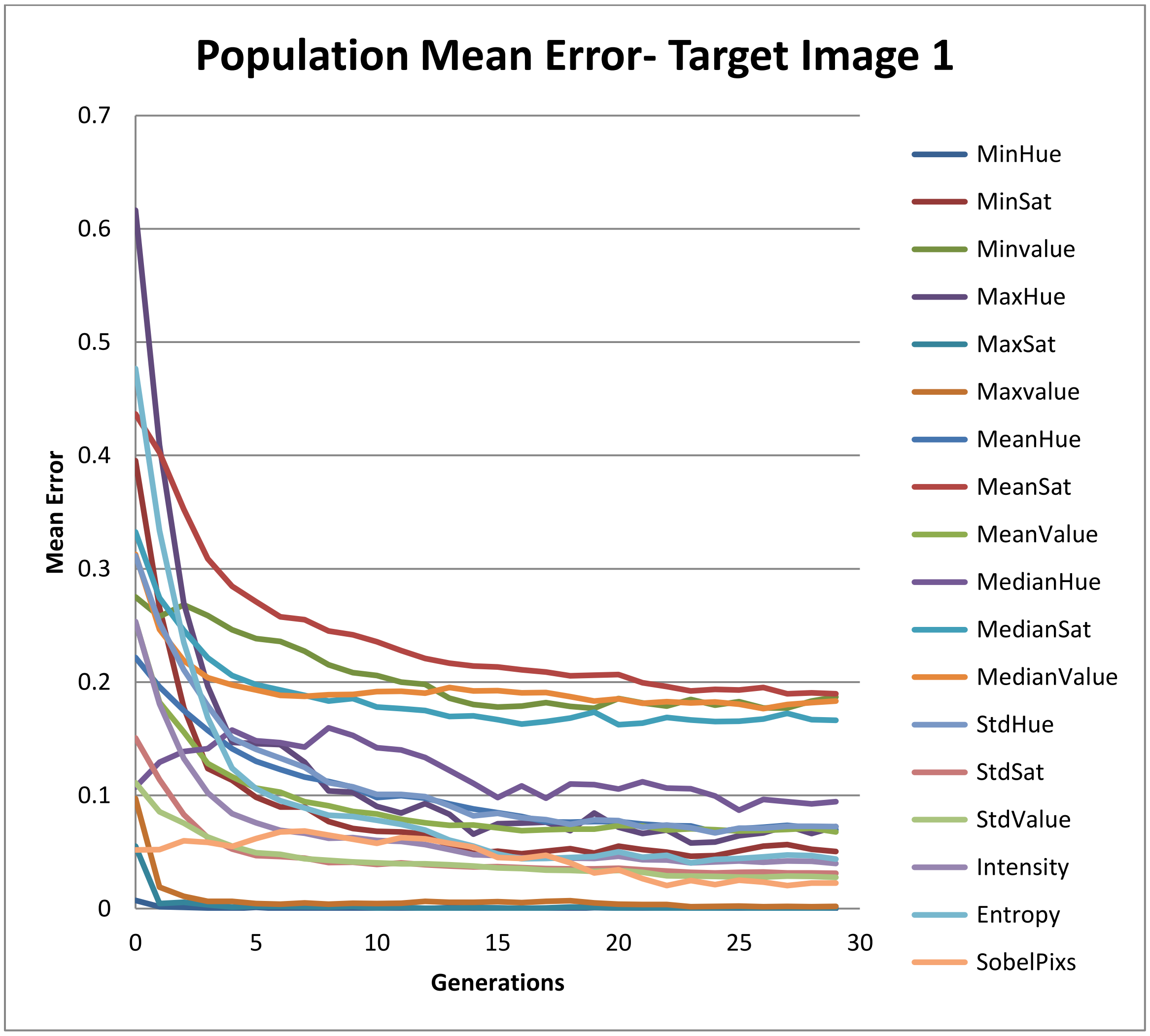


Figure 55: Population mean error for target image 1for 10 runs – distance set 2.

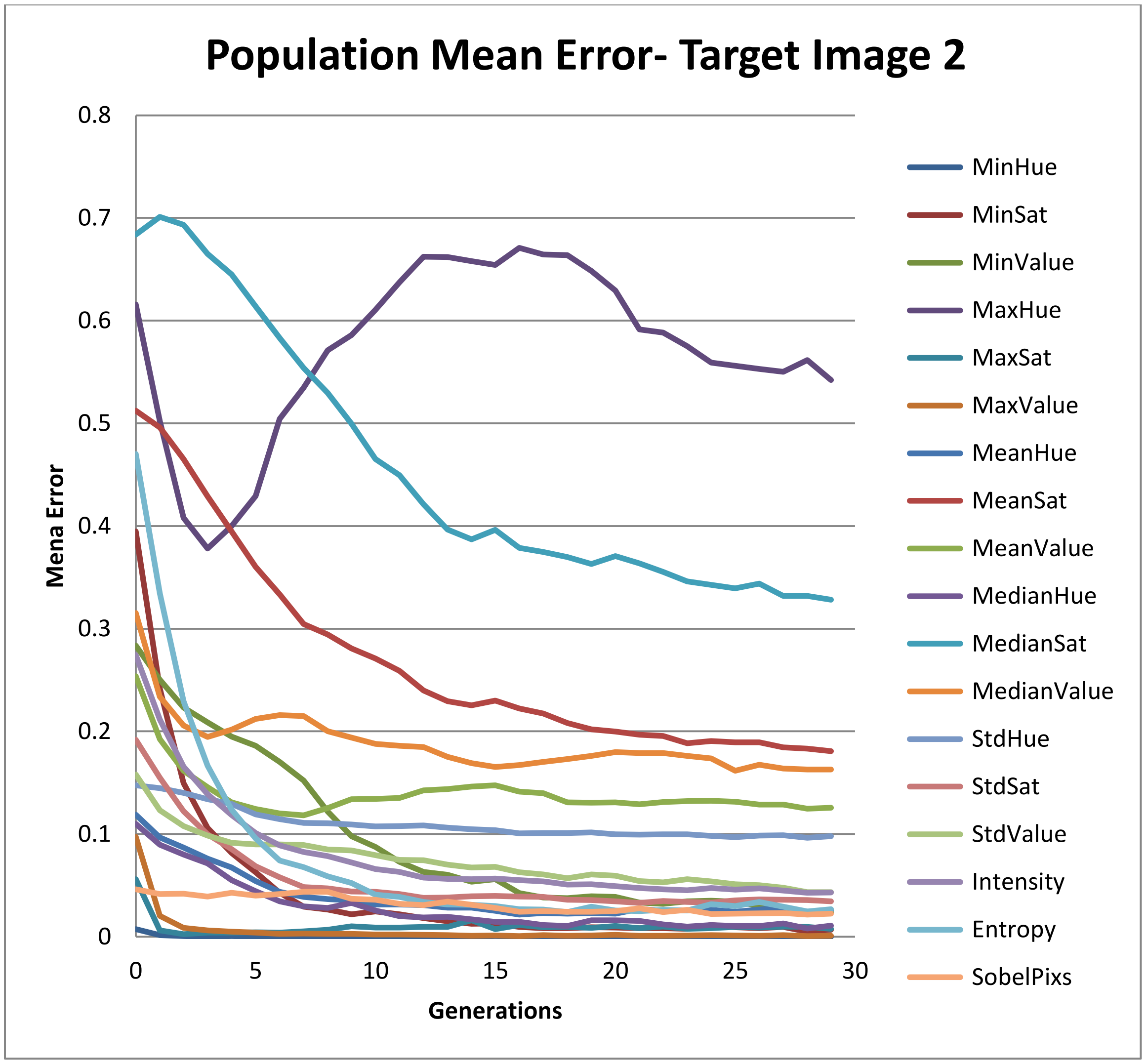


Figure 56: Population mean error for target image 2 for 10 runs – distance set 2.

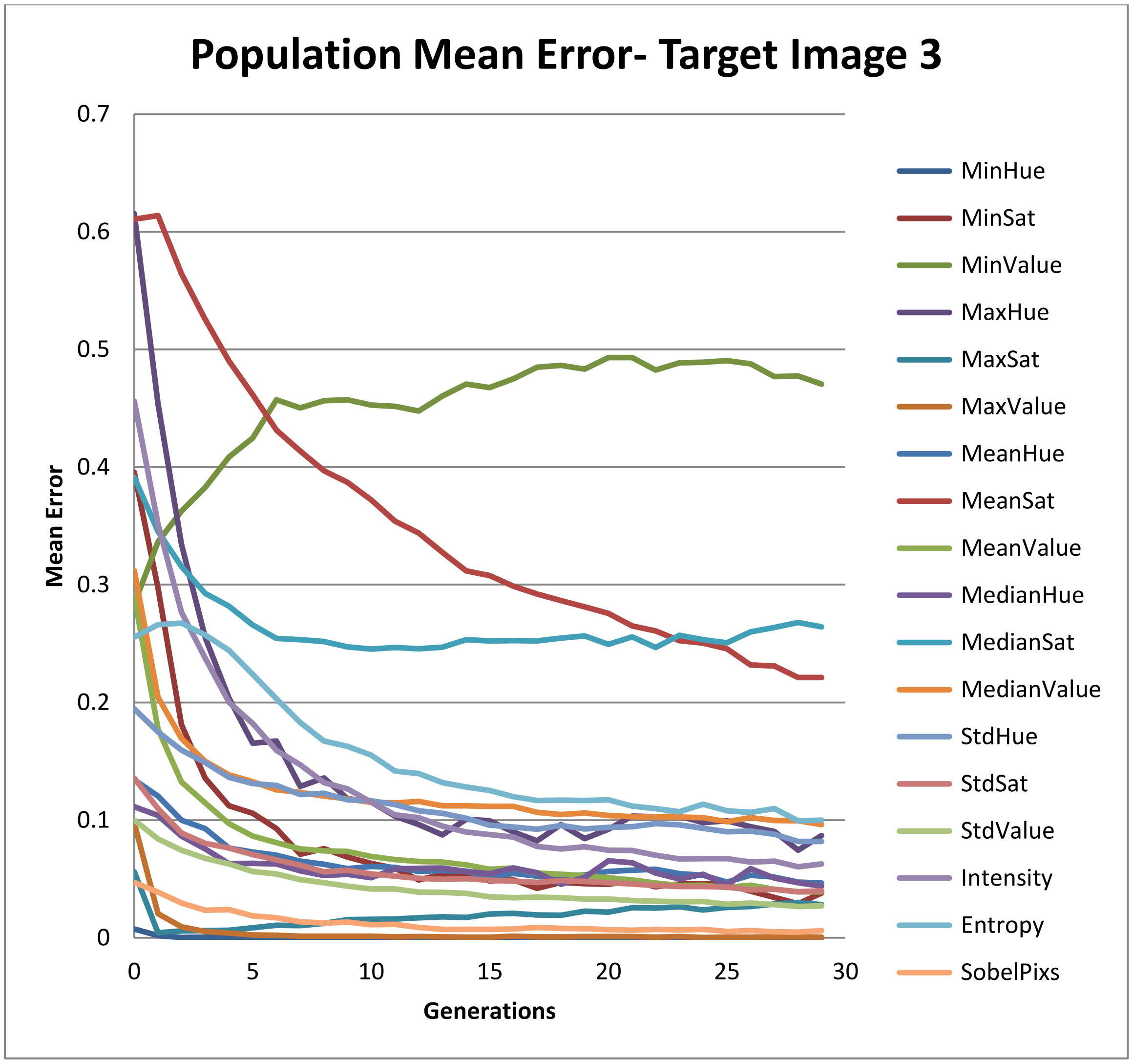


Figure 57: Population mean error for target image 3 for 10 runs – distance set 2.

Comparing plots for the first target image in both experiments show that sum of ranks has kept the good ones as they are and has helped the ones with sudden rises to improve by either converging or getting closer to the fitness level of other objectives. For example, it has improved Sobel pixel, median hue, and max hue in this case.

For the second target image, comparisons show that sum of ranks has improved the overall functionality of participation of objectives and helped some objectives to converge. For example, Median value has improved from generations 5 to 15 and has converged after generation 15 while where sum of ranks is not used it’s not the same. Other examples are mean value that has improved from generations 10 to 20. Standard deviation hue and max sat have improved in their convergence.

In addition, it can possibly be inferred from the plots that sum of ranks can help to smoothen the graph. It means that it has helped objectives with sudden decrease and increases regarding to their previous and later generations. Some of the examples of this claim are max hue from generations 10 to 20, median sat and mean at their generation 15. In the last graph, not many changes have happened in either case. We noticed that plots for each objective have similar curves in distance experiments and full set experiments. For example, for the first target image in set 1 experiments median red, unq pixels, and max red have similar curves in both full set 1 and distance set 1 experiments.

T-test analysis for comparing statistical results of distance set 2 and full set2 show that Sobel pixels (P = 0.04), median hue (P = 0.0001), min value (P = 4.93E-07) were significantly different in distance set 2, and min hue (P = 6.53E-05) was significantly different in full set 2 experiments.

In all experiments, our system shows better performance for some target images but not that efficient for others for instance solutions evolved for target image 1 and target image 3 seem to be closer to their target comparing to target image 2 and its solutions. In addition, we were expecting to see inefficient results in distance experiments but statistical results show that many-objective ranking is perhaps unnecessary in this problem although it was effective.

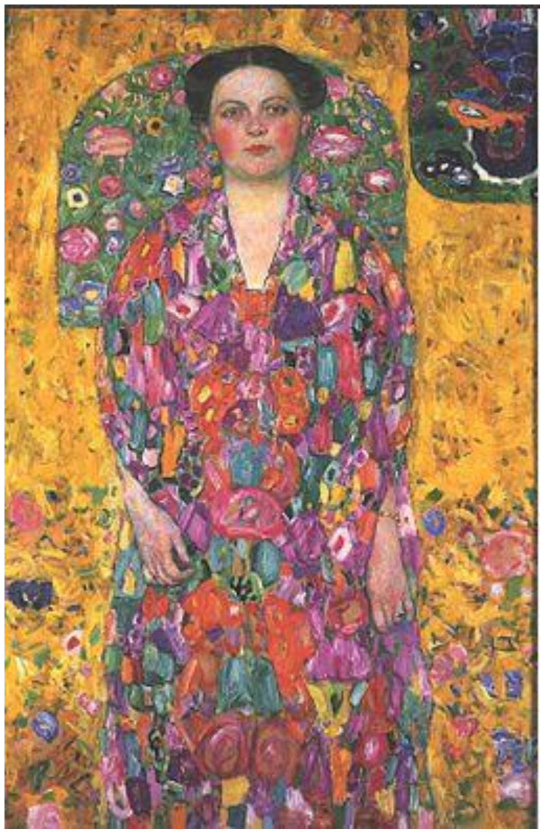
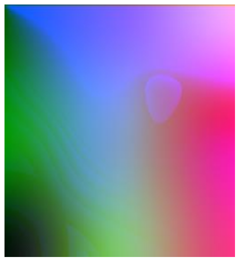
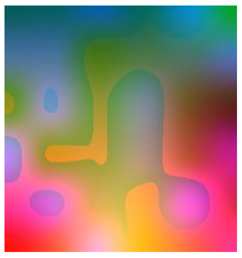
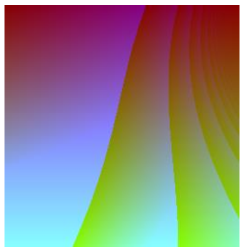
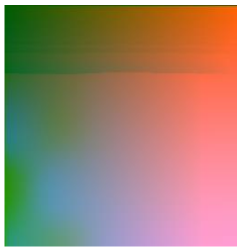
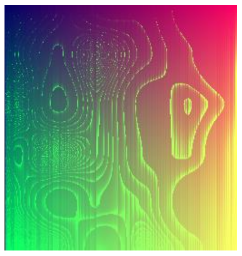
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 58: List of best solutions for target image 1for 10 runs – distance set 2.

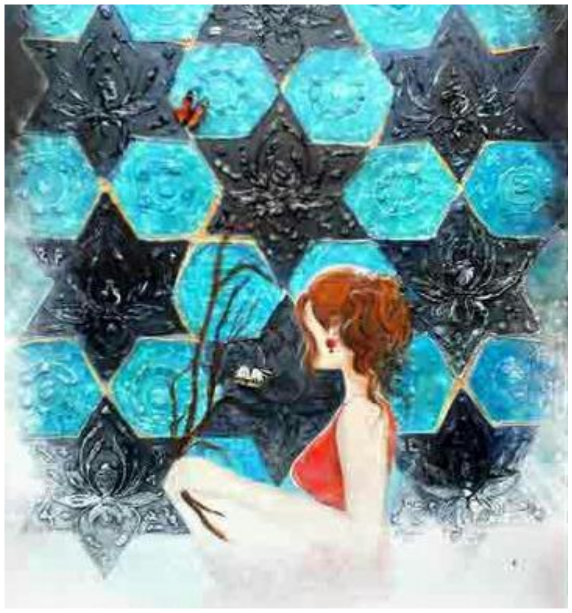
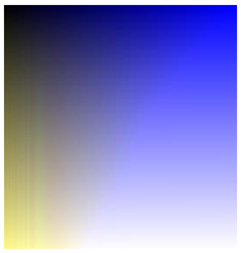
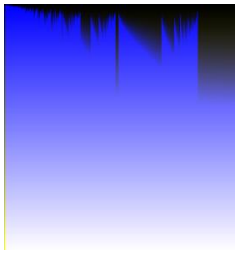
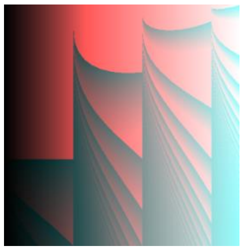
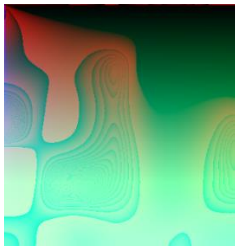
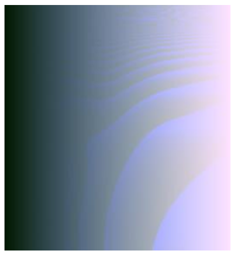
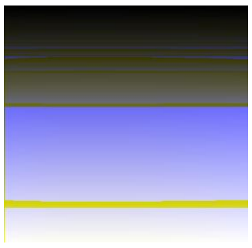
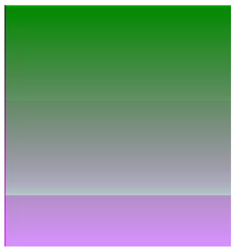
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 59: List of best solutions for target image 2 for 10 runs – distance set 2.

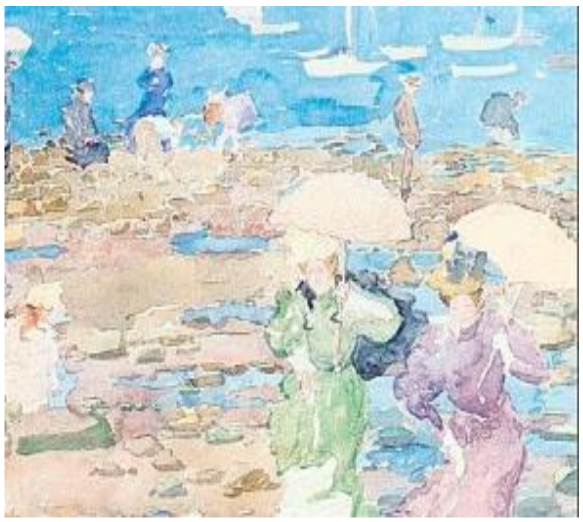
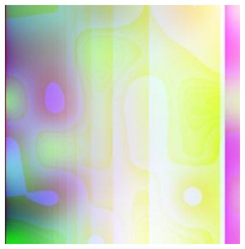
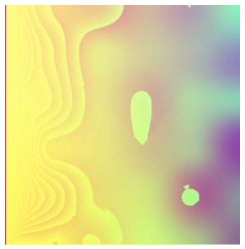
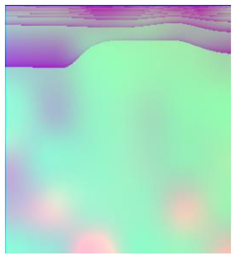
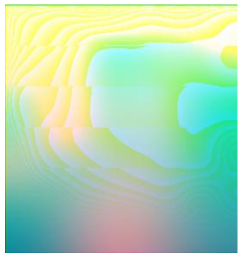
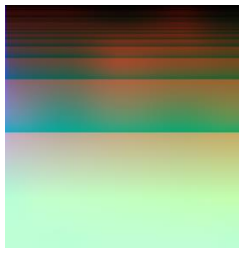
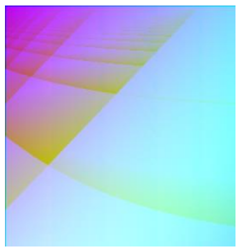
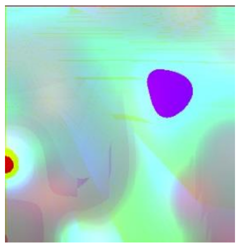
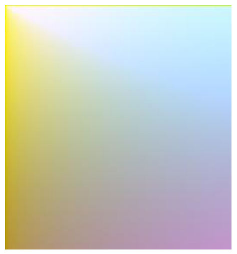
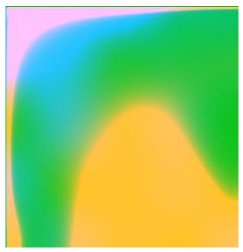
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 60: List of best solutions for target image 3 for 10 runs – distance set 2.

Chapter 7

Human Survey

7.1 Survey Design

The last experiment that we did was a human survey. We want to test whether the system's results are close to human's opinions. In other words, we would like to see if users think a result image from a GP run compared to a target image, looks like a match to that target. Therefore, we designed a survey of 7 randomized questions. In each question a target image is compared to two solutions. The first 3 questions of the survey were testing target image 1, 2 were for target image 2 and 2 for target image 3. We used survey monkey [56] for implementing the survey.

The answer choices were chosen from the best solutions of reduced set experiments mentioned in Section 6.2. We used a random number generator for each time choosing a solution as an answer and we had to shuffle solutions in order to maintain randomization of the selections in our survey. For the first answer choice, we picked a solution randomly from sets 1 and 2 of reduced experiments for the target image and for the second answer choice, we randomly chose among the solutions of the other target images of the reduced sets experiments. In each iteration, the picked solutions are removed from the answers pool to keep our selection random. In each question, the user is asked to choose one of the answer choices, which is the closest to the given image in terms of color closeness. Figure 61 shows a sample question from the survey for target image 1. See Appendix II for complete survey approved by Brock University's ethics board.

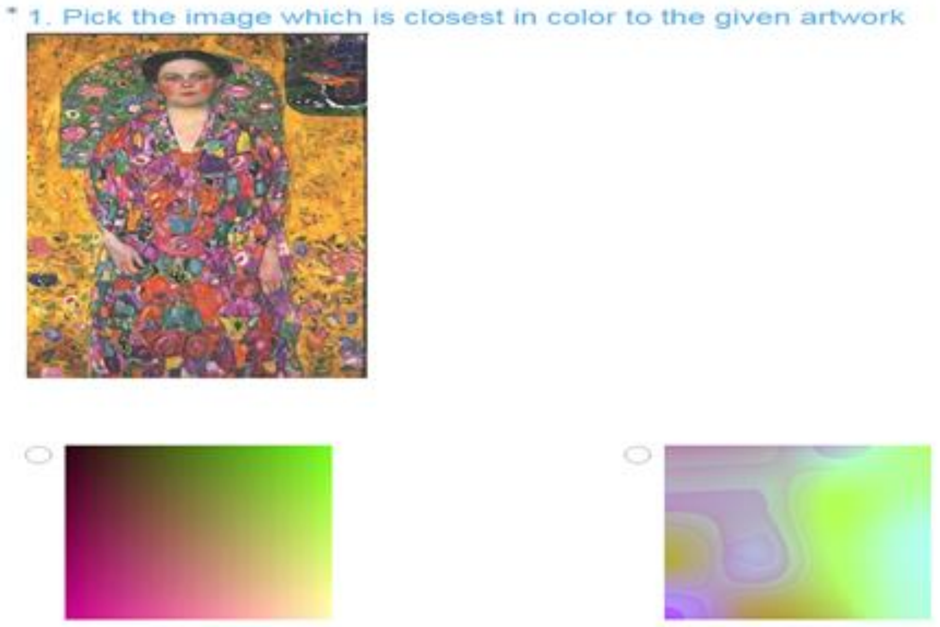

* 1. Pick the image which is closest in color to the given artwork

Figure 61: Human survey sample question for target image 1.

7.2 Results

81 participants completed the survey during the time specified for our work. Participants were Master's student at Brock University and others were publicly invited through social media. All the surveys were complete and the obtained results were analyzed using a signed rank test for a mean (a non-parametric test) [57].

The purpose of this test is to find the difference between mean of a population and a defined significance value where the data is not normally distributed but population is symmetrical. In our problem, $n = 7$ and null hypothesis $P \leq 0.05$. Table 12, shows the answer choices, A and B, and the raw values listed for them are the percentage of each answer chosen by participants. Correct answers for each question are marked with asterisk sign (*) in Table 12.

Table 12: Human survey answers (%).

A	B
* 61.73	38.27
* 88.89	11.11
18.52	* 81.48
2.47	* 97.53
* 83.95	16.05
* 9.88	90.12
* 13.58	86.42

Considering μ_0 the null hypothesis condition as μ_0 being 50% (i.e. correct answers have a random chance of being correct), we found that using 0.05 of significance, the null hypothesis is rejected and results are not significant.

7.3 Conclusion

The survey was done by a high number of participants but resulted in inconclusive results. The reason for not getting strong results is likely that we used too few questions (7), since surveymonkey limited the number of questions we could use in the survey. We would have preferred to have 20. In addition, the image tests are not guaranteed to give perfect matches. A similar case is mentioned in Lombardi *et al.* [4] where the classification accuracy for overall image retrieval was 49.2% in an interactive test on 10 images from 50 artists and the overall performance of their system was 56.3%. Moreover, it could be that humans are more discriminative in these tests than what the lightweight tests are capable of evaluating.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this research, we used lightweight image features to evolve textures using a GP system. Our main motivation was research by Lombardi *et al.* [4], in which they used lightweight features for image retrieval purposes. Our main contribution was to use them as fitness criteria to compare evolved textures with target images. The goal is to use the lightweight tests as a measurement for evolving textures that match characteristics of a target image. To handle the high number of objectives, we used normalized sum of ranks because Pareto ranking is not helpful when the number of objectives go beyond 4. In fact sum of ranks was effective in experiments with up to 17 objectives.

Several experiments were conducted on target images to test the system's results with variations of objectives. Based on the results from the experiments, we could see good convergence for most objectives e.g. Sobel pixels, entropy, unique pixels. However, distance experiment results did well too which shows many-objectivity was unnecessary in this problem.

We noticed that our system works better for some target images, and not as effective for others. T-test analysis results did not show significant differences each objective between two sets of experiments. The overall speed of our system was fast, although we were expecting it to be slower (average of 5 minutes per run depending on the tree size).

The last experiment was the human survey. Our purpose of doing this survey was to see if human's opinions are close to the system's results, i.e. we want to see if humans think that the results from our system match the target images in terms of colors. Analysis reports showed that our survey results were inconclusive, even though we saw good color matches in our experiments results. We decided that the number of questions (7) in our survey could be one reason for insignificant results.

8.2 Future Work

While doing the experiments and analyzing our results, we came up with some new ideas that we would like to cover in our future work on this system.

- The lightweight test concentrates on color characteristics. Adding shape features such as segmentation and wavelet analysis to the objective sets help us getting textures closer to the target images in terms of shapes and visual features outside of color.
- Paying more attention to reduced and mixed sets experiments and analyzing their results is our next goal in future experiments. We want to analyze differences between the effects of using RGB or HSV models on resulting textures in these experiments.
- Examine other color comparison methods used in other research such as quadratic histogram matching [24] [27] [13], color indexing using color histograms [58], and clustering [59]. We would like to investigate what kind of images seems to be

having better results using different color matching methodologies, and how they impact texture evolution.

- We would like to further test different GP parameters, e.g. texture coordinates and GP languages, to see their impact on the results. We thought adding geometrical shapes or fractal features could produce textures that are more interesting.
- Add aesthetic models as new objectives e.g. models used in NPR [13].
- A more comprehensive user survey is necessary that entails new tests such as comparing the effect of HSV and RGB color models on textures.
- There are several ranking methodologies, other than sum of ranks, which can be tested along with our experiments. Kukkonen and Lampinen [60] have introduced some of the approaches that are based on the concept of ranking dominance in their paper. One optimized version of this method is used in a work by Ross et al. [61] [62] in which they have tried to maintain diversity of the population. In [12] another ranking method called sum of ratio is introduced in which the distribution of individual objectives is considered as the ranking factor. Another non-Pareto approach is call L-dominance relation [63], in which fitness values of all individuals are sorted in increasing order.

Bibliography

[1] P. Bentley, D. W. Corne. *Creative Evolutionary Systems*, Morgan Kaufmann, 2002.

[2] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu and G. Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, vol. 5, Springer Science & Business Media, pp. 1-49, 2006.

[3] E. d. Heijer, A. Eiben."Comparing aesthetic measures for evolutionary art". *Proceedings of the EvoMusArt*, vol. 2, pp. 311-320, 2010.

[4] T. Lombardi, S.-H. Cha, and C. Tappert."A lightweight image retrieval system for paintings". *Proceedings of the Storage and Retrieval Methods and Applications for Multimedia*, paper 5682-24, 2005.

[5] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

[6] R. Poli, W. B. Langdon, and N. F. McPhee. *A Field Guide to Genetic Programming*, Published via Lulu.com, 2008.

[7] "geneticprogramming," [Online]. Available: www.geneticprogramming.com. [Accessed 12 October 2016].

[8] B. L. Miller, D. E. Goldberg."Genetic algorithms, tournament selection, and the effects of noise". *Complex Systems*, vol. 9, no. 3, pp. 193-212, 1995.

[9] Branke, Jürgen, K. Deb, K. Miettinen, R. Słowiński, (Eds.). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer, vol. 5252, 2008.

[10] Deb, Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, 2009.

[11] E. Zitzler, L. Thiele."Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach". *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 9, pp. 257-271, 1999

[12] P. J. Bentley, J. P. Wakefield."Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms". *Soft Computing in Engineering Design and Manufacturing*, pp. 231-240, 1998.

[13] M. Baniasadi."Genetic programming for non-photorealistic rendering". *Master's Thesis*.

Brock University, St.Catharines, Ontario, Canada, 2013.

[14] D. G. Stork."Computer vision and computer graphics analysis of paintings and drawings: an introduction to the literature". *International Conference on Computer Analysis of Images and Patterns,* pp. 9-24, 2009.

[15] Y. Rui, T. S. Huang, and S. F. Chang."Image retrieval: current techniques, promising directions, and open issues". *Journal of Visual Communication and Image Representation,* vol. 10, no. 1, pp. 39–62, 1999.

[16] P. L. Stanchev, D. J. Green, and B. Dimitrov."Some issues in the art image database systems". *Journal of Digital Information Management,* vol. 4, no. 4, pp. 227-232, 2006.

[17] P. Kamavisdar, S. Saluja, and S. Agrawal."A survey on image classification approaches and techniques". *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 2, no. 1, pp. 1005-1009, 2013.

[18] M. A.Stricker, M. Orengo."Similarity of color images". *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology,* pp. 381-392, 1995.

[19] J. R. Smith, S. F. Chang."Tools and techniques for color image retrieval". *Electronic Imaging: Science & Technology,* pp. 426-437, 1996.

[20] S. M. Singh, K. Hemachandran."Content based image retrieval based on the integration of color histogram, color moment and Gabor texture". *International Journal of Computer Science,* vol. 59, no. 17, pp. 299-309, 2012.

[21] O. Marques, B. Furht."Content-based visual information retrieval". *Distributed Multimedia Databases: Techniques and Applications*, Idea Group Inc, pp. 37-57, 2001.

[22] R. C. Gonzalez, R. E. Woods. *Digital Image Processing*, Prentice-Hall, 2006.

[23] R. C. Johnson, E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma, and J. Z. Wang."Image processing for artist identification". *IEEE Signal Processing Magazine,* vol. 25, no. 4, pp. 37- 48, 2008.

[24] J. R. Smith, S. F. Chang."VisualSEEk: a fully automated content-based image query system". *Proceedings of the Fourth ACM International Conference on Multimedia,* ACM, pp. 87-98, 1997.

[25] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. "Image retrieval using scene graphs*". IEEE Conference on Computer Vision and Pattern 119 Recognition (CVPR),* pp. 3668-3678, 2015.

[26] K. Sims."Artificial evolution for computer graphics". *SIGGRAPH*, vol. 25, no. 4, pp. 319-

328, 1991.

[27] A. L. Wiens and B. J. Ross.“Gentropy: evolving 2D textures”. *Computer and Graphics*, vol. 26, no. 1, pp. 75- 88, 2002.

[28] A. E. M. Ibrahim.“Evolutionary techniques for procedural texture automation”. *Advances in Visual Computing*, pp. 623-632, 2013.

[29] B. J. Ross, H. Zhu.“Procedural texture evolution using multi-objective optimization”. *New Generation Computing*, vol. 22, no. 3, pp. 271-293, 2004.

[30] S. Gupta, S. G. Mazumdar.“Sobel edge detection algorithm”. *International Journal of Computer Science and Management Research*, vol. 2, no. 2, pp. 1578-1583, 2013.

[31] R. Maini, H. Aggarwal.“Study and comparison of various image edge detection techniques”. *International Journal of Image Processing*, vol. 3, no. 1, pp. 1-11, 2009.

[32] R. Muthukrishnan, M.Radha.“Edge detection techniques for image segmentation”. *International Journal of Computer Science & Information Technology*, vol. 3, no. 6, pp. 259-267, 2011.

[33] "Sobel Operator," [Online]. Available: https://en.wikipedia.org/wiki/Sobel_operator. [Accessed 12 October 2016].

[34] F. Alamdar, M. Keyvanpour .“a new color feature extraction method based on dynamic color distribution entropy of neighborhoods”. *IJCSI International Journal of Computer Science*, vol. 8, no. 5, pp. 42-48, 2011.

[35] P. Bromiley, N. Thacker, and E. Bouhova-Thacker.“Shannon entropy, Renyi entropy, and information”. *Statistics and Inf. Series*, 2004.

[36] "Entropy(information theory)," [Online]. Available: https://en.wikipedia.org/wiki/Entropy_(information_theory). [Accessed 7 October 2016]

[37] "Entropy," [Online]. Available: http://www.johnloomis.org/ece563/notes/basics/entropy/entropy.html. [Accessed 7 October 2016]

[38] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan.“Local shannon entropy measure with statistical tests for image randomness”. *Information Sciences*, vol. 2, no. 39, pp. 62-65, 2012.

[39] "Stackoverflow," [Online]. Available: http://stackoverflow.com/questions/22265872/calculation-of-entropy-of-an-image-using-java . [Accessed 12 October 2016].

[40] "Grayscale," [Online]. Available: https://en.wikipedia.org/wiki/Grayscale. [Accessed 12 October 2016].

[41] "Number 11, (1952)," [Online]. Available https://en.wikipedia.org/wiki/Number_11,_1952_(painting). [Accessed 16 July 2016].

[42] "Number5, (1948)," [Online]. Available: https://en.wikipedia.org/wiki/No._5,_1948. [Accessed 11 June 2016].

[43] "The Stone Bridge," [Online]. Available: https://en.wikipedia.org/wiki/The_Stone_Bridge. [Accessed 8 August 2016].

[44] "The Storm on the Sea of Lake Galilee," [Online]. Available: https://en.wikipedia.org/wiki/The_Storm_on_the_Sea_of_Galilee. [Accessed 8 August 2016].

[45] "The Starry Night," [Online]. Available: https://en.wikipedia.org/wiki/The_Starry_Night. [Accessed 27 August 2016].

[46] "Café Terrace at Night," [Online]. Available: https://en.wikipedia.org/wiki/Caf%C3%A9_Terrace_at_Night. [Accessed 31 July 2016].

[47] "Keelmen Heaving in Coals by Moonlight," [Online]. Available: https://en.wikipedia.org/wiki/Keelmen. [Accessed 17 March 2016].

[48] "Modern Rome," [Online]. Available: https://en.wikipedia.org/wiki/Modern_Rome_%E2%80%93_Campo_Vaccino. [Accessed 28 March 2016].

[49] S. Bergen, B. Ross.“Evolutionary art using summed multi-objective ranks”. *Genetic Programming Theory and Practice VIII*, Springer, pp. 227-244, 2010.

[50] S. Luke,"the ECJ owner’s manual," [Online]. Available: https://cs.gmu.edu/~eclab/projects/ecj/docs/manual/manual.pdf . [Accessed 12 October 2016]

[51] "Perlin Noise," [Online]. Available: https://en.wikipedia.org/wiki/Perlin_noise. [Accessed 31 July 2016]

[52] "Portrait of Eugenia Primavesi,". [Online]. Available: https://commons.wikimedia.org/wiki/File:Klimt_-_Bildnis_Eugenia_Primaesi.jpg .[Accessed 12 October 2016]

[53] "Maurice Summer Visitors," [Online]. Available: https://en.wikipedia.org/wiki/Maurice_Prendergast. [Accessed 12 October 2016].

[54] "RGB Color Model," [Online]. Available: https://en.wikipedia.org/wiki/RGB_color_model. [Accessed 25 February 2016]

[55] "HSV Color Model," [Online]. Available: https://en.wikipedia.org/wiki/HSL_and_HSV. [Accessed 13 August 2016].

[56] "Survey Monkey," [Online]. Available: https://www.surveymonkey.com/. [Accessed 12 October 2016].

[57] G. K. Kanji. *100 Statistical Tests*, Sage Publication, p. 95, 2006.

[58] D. Neumann, K. R. Gegenfurtner.“Image retrieval and perceptual similarity”. *ACM Transactions on Applied Perception*, vol. 3, no. 1, pp. 31-47, 2006.

[59] M. S. Kankanhalli, B. M. Mehtre, and J. K. Wu.“Cluster-based color matching for image retrieval”. *Pattern Recognition*, vol. 29, no. 4, pp. 701-708, 1996.

[60] J. L. S. Kukkonen.“Ranking-dominance and many-objective optimization”. *IEEE International Congress on Evolutionary Computation*, pp. 3983-3990, 2007.

[61] B. R. W. Barry.“Virtual photography using multi-objective particle”. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp. 285-292, 2014.

[62] B. Ross.“Evolution of stochastic bio-networks using summed rank strategies”. *IEEE International Congress on Evolutionary Computation*, pp. 773-780, 2011.

[63] X. Zou, Y. Chen, M. Liu, & L. Kang.“a new evolutionary algorithm for solving many-objective optimization problems”. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, no. 5, pp. 1402-1412, 2008.

Appendix I

Best tree for solution 10 in full set experiments set 1 target image 1:

```
(render-texture (rgb (cos (+ (* ERCFloat[d4605821809130995712|0.84894198179245|]
v) (noise ERCFloat[d4605404023166599168|0.802558422088623|])))) (noise
ERCFloat[d4606471661842071552|0.9210901260375977|]) (power (% (power (% (%
(noise ERCFloat[d4603693188979359744|0.6126176714897156|]) u) (+ (noise
ERCFloat[d4606471661842071552|0.9210901260375977|]) (/ (%
ERCFloat[d4606858979308470272|0.9640910029411316|]
ERCFloat[d4602284488561524736|0.47811025381088257|]) (noise
ERCFloat[d4607037591965925376|0.9839209914207458|]))))) (power (+
ERCFloat[d4605821809130995712|0.84894198179245|]
ERCFloat[d4596334424479498240|0.19890767335891724|]) (noise
ERCFloat[d4603486603199905792|0.5896820425987244|]))) (+ (noise
ERCFloat[d4605404023166599168|0.802558422088623|])
ERCFloat[d4596334424479498240|0.19890767335891724|])) (power (+
ERCFloat[d4594111628579962880|0.13721269369125366|]
ERCFloat[d4596334424479498240|0.19890767335891724|]) (power (noise
ERCFloat[d4606471661842071552|0.9210901260375977|]) (noise
ERCFloat[d4605404023166599168|0.802558422088623|]))))))
```

Best tree for solution 10 in full set experiments set 2 target image 1:

```
(render-texture (rgb u (+ (* (* (noise
ERCFloat[d4588053502047027200|0.05476647615432739|]) (power (/ (noise
ERCFloat[d4606053933322862592|0.8747129440307617|]) v) (* (% (% u
ERCFloat[d4598684879691972608|0.27829182147979736|]) (noise
ERCFloat[d4603273060110303232|0.5659739971160889|])) (noise
ERCFloat[d4601513868349407232|0.4353322386741638|]))))) (* (/ (% (noise
ERCFloat[d4601513868349407232|0.4353322386741638|]) (sin v)) (+ v (power (% u
ERCFloat[d4598684879691972608|0.27829182147979736|]) (% u
ERCFloat[d4598684879691972608|0.27829182147979736|])))) (cos (power (* (+ u v)
(noise ERCFloat[d4601513868349407232|0.4353322386741638|])) (+ v (noise
ERCFloat[d4603273060110303232|0.5659739971160889|])))))) v)
ERCFloat[d4600638822449938432|0.38675743341445923|]))
```

Best tree for solution 10 in reduced set experiments set 1 target image 1:

(render-texture (rgb v u (sin (% (% (sin (% (noise ERCFloat[d4606544337520558080|0.9291587471961975|]) (/ (noise ERCFloat[d4605995535726280704|0.8682295083999634|]) u))) (cos v)) (cos (cos (cos (noise ERCFloat[d4606544337520558080|0.9291587471961975|]))))))))

Best tree for solution 10 in reduced set experiments set 2 target image 1:

(render-texture (rgb u (+ ((* (% (sin v) v) (* ERCFloat[d4600638822449938432|0.38675743341445923|] (/ (+ (* u u) ERCFloat[d4587790804667334656|0.052943646907806396|]) ERCFloat[d4606541319232290816|0.9288236498832703|]))) (* (% (* ERCFloat[d4600638822449938432|0.38675743341445923|] u) v) (/ (* u (sin v)) (+ (power (power ERCFloat[d4595175240723595264|0.16673386096954346|] ERCFloat[d4606541319232290816|0.9288236498832703|]) ERCFloat[d4604575136354402304|0.7105334997177124|]) (noise ERCFloat[d4606298068390772736|0.9018173813819885|]))))) (sin (+ (* v (% (noise ERCFloat[d4601885675078287360|0.4559716582298279|]) (noise ERCFloat[d4598323349746089984|0.25822287797927856|]))) (sin v)))) ERCFloat[d4600638822449938432|0.38675743341445923|]))*

Appendix II

Project Title: Statistical Image Analysis for Image Evolution

Date: 2/29/2016

Principal Investigator (PI), Faculty Supervisor:

Dr. Brian Ross

Department of Computer Science

Brock University

(905) 688-5550 Ext. 4284

bross@brocku.ca

Student Investigator:

Elham Salimi

Department of Computer Science

Brock University

es13dz@brocku.ca

INVITATION

You are invited to participate in a study that involves research. The research will involve your participation in an online survey. My MSc thesis research is in the area of computer generated art. The goal of this research is to contribute improved techniques for automating the generation of artistic images with computer software. My research focuses on artificial intelligence software that generates abstract computer images. The computer software uses mathematical measurements to help generate images that are visually similar to predefined examples of art. The predefined examples might be famous artworks found online. The purpose of this survey is to obtain some insight into human opinions on the quality of the images generated by my system. In particular, I wish to see whether the computer generated images are indeed felt to be similar to existing examples of images of art from a human's point of view. Our goal is to statistically determine whether the mathematical analyses used by the software matches with human's opinions about the generated images.

WHAT'S INVOLVED

In this online survey, you will be presented with an existing art work, and two computer generated images.

POTENTIAL BENEFITS AND RISKS

There are no known or anticipated risks associated with participation in this study.

CONFIDENTIALITY

The users do not have to include any personal identification, such as name or email address. All recorded data will be anonymous. No information will be retained that will link your responses to your identity. Data collected during this study will be stored as part of my research and will be added to the appendix of my thesis. Note that incomplete surveys will not be used.

VOLUNTARY PARTICIPATION

Participation in this study is voluntary. If you wish, you may decline to answer any questions or participate in any or all components of the study. Incomplete surveys will not be included in the research analysis. Further, you may decide to withdraw from this study during the course of the survey, and may do so without any penalty (e.g. you may close your browser at any time). However, it will not be possible to withdraw from participation after the survey has been completed, since it will not be possible to identify and remove a specific participant's responses in the data afterwards.

PUBLICATION OF RESULTS

Participants can read about the results of this research in my completed thesis upon submission (anticipated in 2016), via the library's online collection. Results of this study may also be published in professional journals and presented at conferences. Additional information about this study will be available at www.cosc.brocku.ca/~bross.

CONTACT INFORMATION AND ETHICS CLEARANCE

If you have any questions about this study or require further information, please contact Elham Salimi and Dr. Brian Ross using the contact information provided above. This study has been reviewed and received ethics clearance through the Research Ethics Board at Brock University. If you have any comments or concerns about your rights as a research participant, please contact the Research Ethics Office at (905) 688-5550 Ext. 3035, reb@brocku.ca.

This study has been reviewed and received ethics clearance through Brock University's Research Ethics Board (file
#15-249).

Thank you for your assistance in this project. Please keep a copy of this form for your records.This research is partially supported by NSERC Discovery Grant 138467.

CONSENT FORM

By clicking the Next button below, I indicate that:

• I have read and understood the above information.

• I agree to participate in this study. I have made this decision based on the information I have read in the

Information-Consent Letter.

• I have had the opportunity to receive any additional details I wanted about the study and understand that I

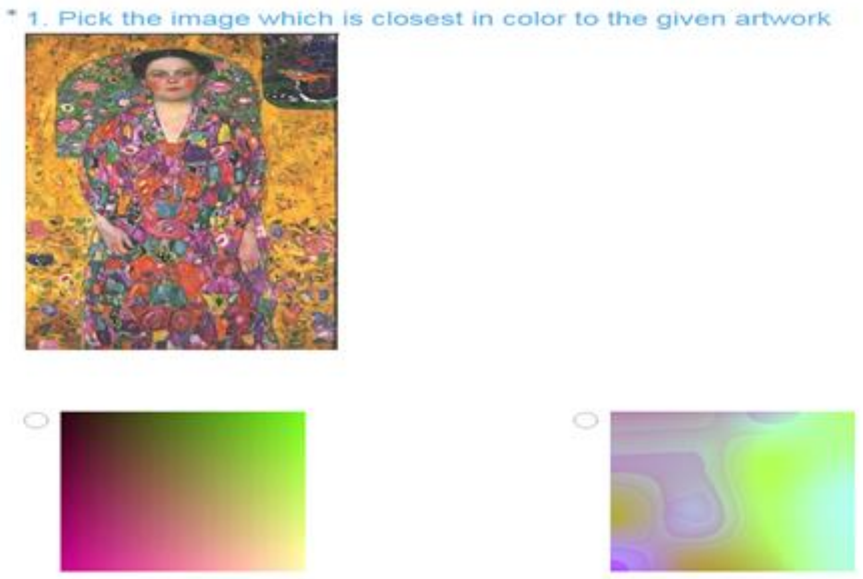
may ask questions in the future.

• I understand that I may withdraw this consent at any time during the survey.

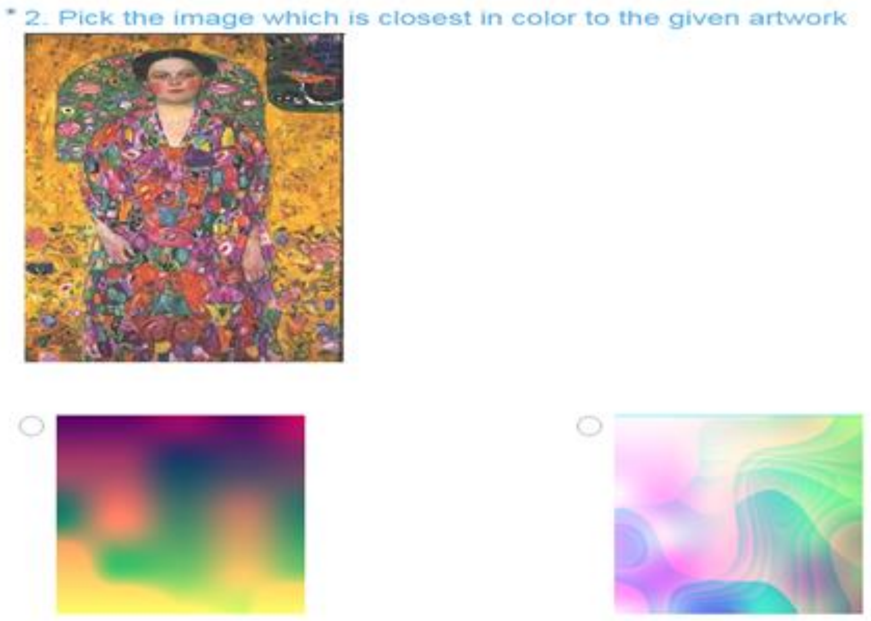
• I should keep a copy of this consent form (web page) for my records.

By clicking the Exit link at the upper right corner of the page, you can exit the survey at any time.

Survey Questions :

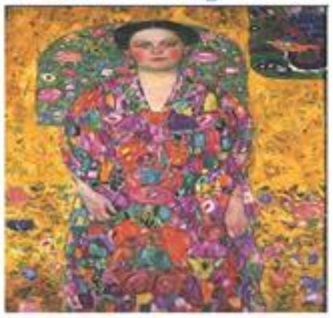


* 1. Pick the image which is closest in color to the given artwork

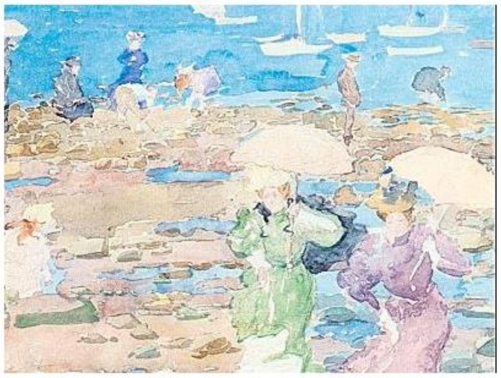


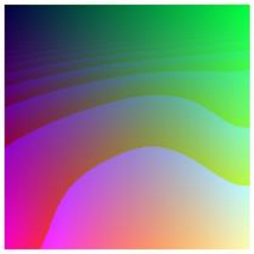
* 2. Pick the image which is closest in color to the given artwork

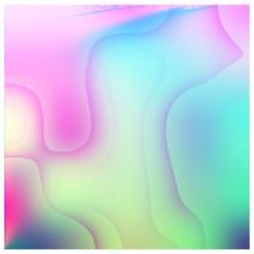
* 3. Pick the image which is closest in color to the given artwork



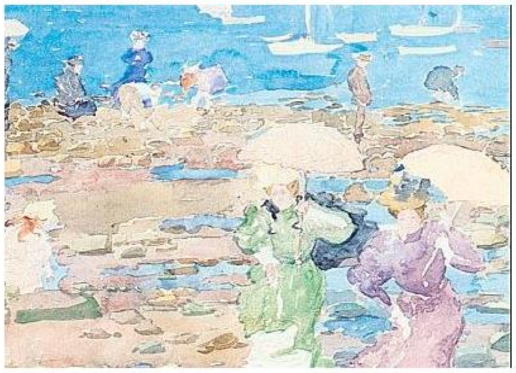
* 4. Pick the image which is closest in color to the given artwork

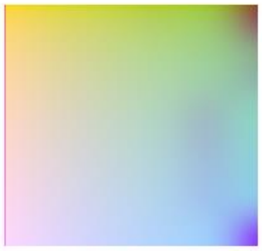


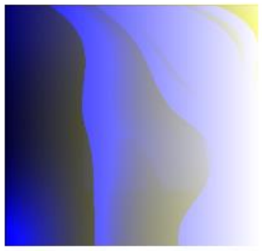




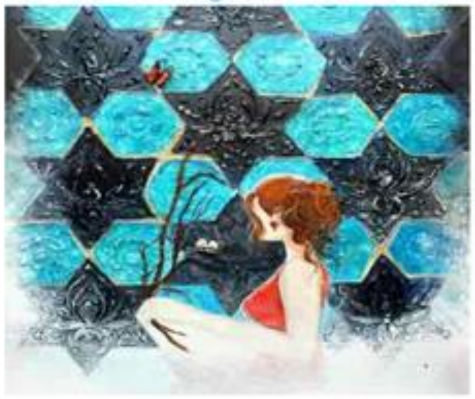
* 5. Pick the image which is closest in color to the given artwork



○ 

○ 

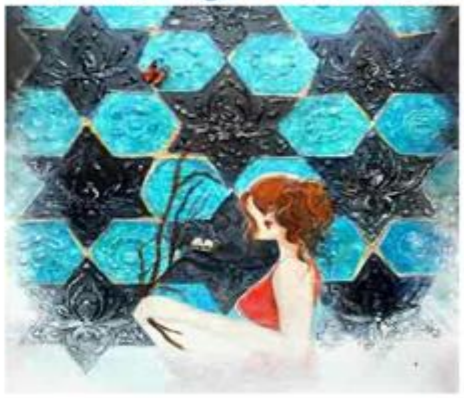
* 6. Pick the image which is closest in color to the given artwork



○

○

* 7. Pick the image which is closest in color to the given artwork



○

○

Appendix III

No Color Experiment:

In this experiment, we selected objectives that are not related to colors and tried to run the GP system with those objectives and the language set similar to all previous experiments. The solutions for target image 1 are shown in the following figure.

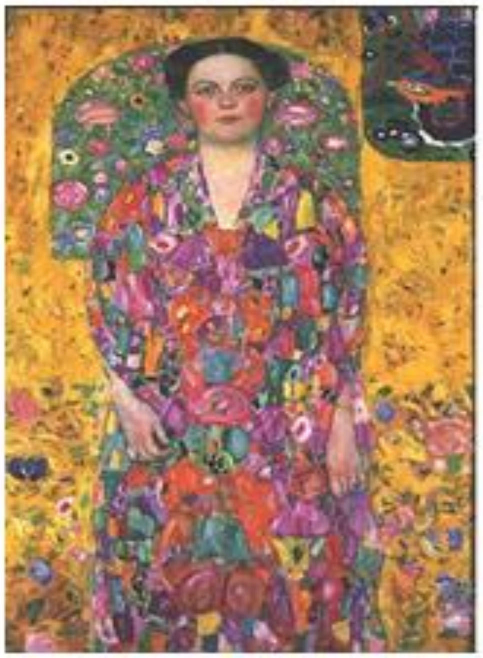
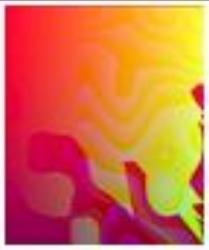
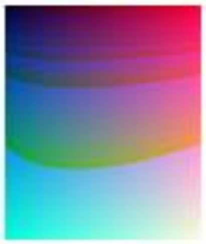
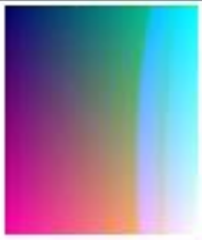
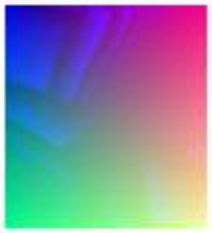
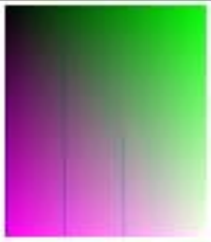
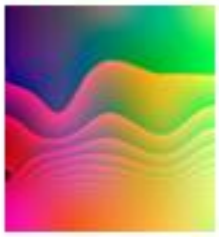
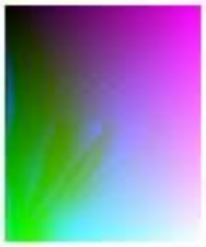
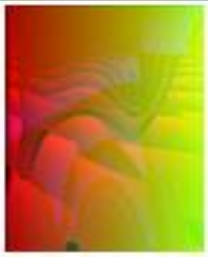
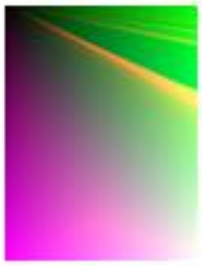
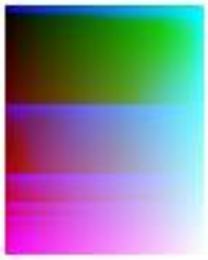
Target	Run 1 to Run 5	Run 6 to Run10
		
		
		
		
		

Figure 62: List of best solutions for target image 1 – No color experiment.

Appendix IV

Tables 13 and 14 show feature values calculated using the GUI shown in Figure 9 in Section 4.6.

Table 13: Feature values for target images set 1.

Feature items	Target Image 1	Target Image 2	Target Image 3
Min Red	0.023529	0	0
Max Red	1	1	1
Mean Red	0.63823	0.445338	0.762892
StD Red	0.220724	0.291432	0.198141
Med Red	0.654902	0.364706	0.815686
Min Green	0	0	0
Max Green	1	1	1
Mean Green	0.47796	0.589495	0.808224
StD Green	0.15916	0.291609	0.130864
Med Green	0.466667	0.647059	0.831373
Min Blue	0	0	0
Max Blue	1	1	1
Mean Blue	0.367229	0.606359	0.804311
StD Blue	0.148064	0.290492	0.139519
Med Blue	0.333333	0.682353	0.831373
unique pix	0.845138	0.553027	0.514718

Table 14: Feature values for target images set 2.

Feature items	Target Image 1	Target Image 2	Target Image 3
Min Hue	0	0	0
Max Hue	0.999997	0.999998	0.999989
Mean Hue	0.206044	1.48E-02	0.054375
Med Hue	3.03E-04	1.44E-03	7.56E-04
StD Hue	0.403869	0.115275	0.224967
Min Sat	0	0	0
Max Sat	1	1	1
Mean Sat	0.439782	0.339418	0.200706
Med Sat	0.435897	0.273973	0.151639
StD Sat	0.204904	0.26942	0.157238
Min Value	0.047059	0	0
Max Value	1	1	1
Mean Value	0.65203	0.632519	0.870286
Med Value	0.662745	0.717647	0.905882
StD Value	0.207199	0.28529	0.125661
Int Mean	0.997765	0.898231	0.011957
Entropy	0.692816	0.685526	0.144401
Sobel pixels	0.01059	1.49E-02	7.28E-03

Appendix V

Table 15 shows average of best solutions for set 1 of full sets experiments for all three target images and Table 16 shows average of best solutions for set 2 pf full sets experiments for three target images.

Table 15: Average best solutions feature values - full set 1.

	MinRed	MinGreen	MinBlue	MaxRed	MaxGreen	MaxBlue	MeanRed	MeanGree	MeanBlue	MedRed	MedGren	MediBlue	StDRed	StDGreen	StDBlu	UniqueRG
Target 1	0.0235	0	0	1	1	1	0.6382	0.4779	0.3672	0.2207	0.1591	0.1480	0.6549	0.4666	0.3333	0.8444
evolved 1	0	0.0121	0.0227	0.9949	0.9756	0.8835	0.6163	0.4600	0.3587	0.2399	0.2253	0.1732	0.6372	0.4537	0.3498	0.8387
Target 2	0	0	0	1	1	1	0.4453	0.5894	0.6063	0.2914	0.2916	0.2904	0.3647	0.6470	0.6823	0.5530
evolved 2	0	0	0	0.9968	0.9984	0.9996	0.4867	0.5944	0.6111	0.4858	0.6572	0.6176	0.2827	0.2926	0.2864	0.5553
Target 3	0	0	0	1	1	1	0.7628	0.8082	0.8043	0.1981	0.1308	0.1395	0.8156	0.8313	0.8313	0.5147
evolved 3	0.0596	0.1082	0.0890	0.9980	0.9996	0.9996	0.7369	0.8169	0.8186	0.7760	0.8360	0.8749	0.2001	0.1368	0.1703	0.5169

Table 16: Average best solutions feature values - full set 1.

	MinHue	MinSat	MinValu	MaxHue	MaxSat	MaxValu	MeanHu	MeanSat	MeanVal	MedHue	MedSat	MedValu	StDHue	StDSat	StDValu	Intensity	Sobel	Entropy
Target 1	0	0	0.0470	0.9999	1	1	0.2059	0.4397	0.6520	3.03E-0	0.4358	0.6627	0.4038	0.2049	0.2072	0.9977	0.0105	0.6928
evolved 1	0	0.0067	0.2356	0.9999	1	0.9996	0.2052	0.5774	0.6792	0.0011	0.5688	0.6682	0.4027	0.2171	0.1952	0.7975	0.0073	0.6918
Target 2	0	0	0	0.9999	1	1	0.0147	0.3394	0.6325	0.0014	0.2739	0.7176	0.1152	0.2694	0.2852	0.8982	1.49E-0	0.6855
evolved 2	0	0	0	0.7004	0.9933	0.9996	0.0077	0.4829	0.7118	0.0016	0.4660	0.8258	0.0604	0.2711	0.2905	0.9498	0.0123	0.6884
Target 3	0	0	0	0.9999	1	1	0.0543	0.2007	0.8702	7.56E-0	0.1516	0.9058	0.2249	0.1572	0.1256	0.0119	7.28E-0	0.1444
evolved 3	0	0	0.5321	0.9999	1	1	0.0515	0.3034	0.8842	0.0015	0.2987	0.9337	0.2179	0.1489	0.1240	0.6438	0.0047	0.1187