Exploration of Human Design with Genetic Algorithms as Artistic Medium for Color Images

Aidan F. Schmelzle
EECS
University of Kansas
Lawrence, USA
schmelzle@ku.edu

***Abstract*—Genetic Algorithms (GAs), a subclass of evolutionary algorithms, seek to apply the concept of natural selection to promote the optimization and furtherance of “something” designated by the user. GAs generate a population of chromosomes represented as value strings, score each chromosome with a “fitness function” on a defined set of criteria, and mutate future generations depending on the scores ascribed to each chromosome. In this case, each chromosome is a bitstring representing one canvased artwork. Artworks are scored with a variety of design fundamentals and user preference. The artworks are then evolved through thousands of generations and the final piece is computationally drawn for analysis. While the rise of gradient-based optimization has resulted in more limited use-cases of GAs, genetic algorithms still have applications in various settings such as hyperparameter tuning, mathematical optimization, reinforcement learning, and black box scenarios. Neural networks are favored presently in image generation due to their pattern recognition and ability to produce new content; however, in cases where a user is seeking to implement their own vision through careful algorithmic refinement, genetic algorithms still find a place in visual computing. The debated obsoletion of GAs in image generation is negated in this case as the genetic algorithm itself becomes a medium for artistic expression.**

***Index Terms*—evolutionary algorithms, genetic algorithms, image generation**

I. Introduction

While the time for the application of genetic algorithms (GAs) in training machine learning models has mostly come to pass, their multimodality has preserved them from extinction. GAs can be applied to almost anything that a user can quantify and excel at finding global optima in large, potentially discontinuous search spaces. Though they have been found to possess sub-optimal image generation capabilities in comparison to advanced techniques, there is an argument for returning to GAs as a mode to incorporate more human involvement. The intersection between art and technology has experienced much unrest in recent times as many believe that the removal of humans from ”art” poses a dangerous question for our future. Others argue that machine-generated art opens doors and makes the creation of art accessible. The rapid progression of AI techniques begs us to take a look backward to consider what may have been left in the wake of this technological boom. The demand for careful construction of the GA elements reintroduces human touch to program output. This research explores the potential to quantify design elements in order to iterate generations of artworks.

II. Background

A. Genetic Algorithms

GAs draw upon the concept of natural selection to find optimal problem solutions. The algorithm works by generating a population of candidate solutions (chromosomes), evaluating candidates on a defined fitness function, selecting the fittest individual chromosomes and using them to produce offspring. If a stopping criteria is met, the algorithm reaches completion, otherwise, evolution continues with the new generation of children.

1) Chromosome: Each candidate solution is represented by a value string. In this case, the string is a 736-bit binary string. The chromosome contains all the information about the artwork.

2) Gene: Genes are components of GA chromosomes. One or more bits provide information about the attributes of the chromosome. For example, in this research, the first 3 bits of every 23-bit sequence in a chromosome represent a shape.

3) Population: A population is a set of possible solutions. Initial GA populations can be either partially or fully manually-constructed or partially or fully random. In this research, the initial population is composed of 300+, random bit strings.

4) Selection: GAs utilize various selection methodologies to determine which chromosomes will produce offspring. Here, roulette wheel selection (RWS) is utilized. In this selection method, more highly fit chromosomes occupy a larger ”slice” of the roulette wheel, and thus, a larger chance of being selected to reproduce. RWS offers a realistic selection process as all chromosomes are still eligible to reproduce as chromosomes receiving lower fitness scores are not eliminated from the gene pool, they just have a lower chance of reproducing.

5) Crossover: Crossover constitutes the means by which chromosomes are combined to generate new offspring. Single-point crossover is utilized in this experiment. In single-point crossover, a point between 2 bits is chosen at random, and the tails are swapped to create two new offspring.

6) Mutation: Mutations are incorporated into GA evolutions the way one would expect to see in the natural world. Mutations introduce random changes to chromosomes to promote diverse populations and to prevent the algorithm from reaching premature convergence. Mutations rarely occur, but to prevent early convergence, mutations occur as often as 5-10% of the time.

B. Design Concepts

Again, the question of this research is not whether a computer can produce art but rather what kind of art can a human create with GAs. As the fitness function must be implemented by a human, there is demand for human design and discernment. The structure of both the chromosome and the fitness function were constructed with design concepts in mind. Thought was given to the number of elements and colors needed to create interest as well as how to preserve the standard color wheel. Some of the general design principles include composition, layout, color theory, focal point, hierarchy, and repetition. In order to score artworks, design elements must be quantified which is where the user must make decisions about how to reward or punish chromosome characteristics based on their own preference. For example, there is no absolute way to define what constitutes a perfect composition, but there are general rules of thumb. Further specifications are derived from individual preference. The design elements selected for usage here are symmetry, color harmony, and shape diversity.

1) Symmetry: The design concept of symmetry is often thought of as providing a sense of balance to a piece and aids in drawing a viewer in towards an image's focal point. Analyzing this characteristic will involve examining the mirror-point of each pixel in the piece along one or more axes.

2) Color Harmony: Harmonious color palettes provide interest to pieces in a variety of ways. Monochromatic, or single-hue, pieces draw viewers in as the lack of color diversity encourages you to look more closely at what is hidden in the variety of values. Complementary palettes help to make opposing colors "pop" and can add an element of extremity due to the rich saturation. The algorithm will use the chromosome attributes as well as the standard color wheel to analyze each piece.

3) Shape Diversity: The range of shape diversity can add interest to a piece. Whether all shapes or the same or all types of shapes are included on the canvas, interest is created in the overlapping elements of the piece.

III. Related Work

A. GAs in Binary Image Evolution

Genetic algorithms have been applied to the evolution of binary images [3]. Binary images in [3] are represented by 16x16 matrices. These images are evaluated on their fitness and reproduce to create the next generations of children. Fitness is calculated using various "masks" of 3x3 or 4x4 size which traverse the matrix looking for features such as solid squares, hollow, squares, vertical lines, etc. In this case, population size was 1000 and each experiment ran for 120 generations. Final matrices were tiled to more clearly show the resulting patterns of the high-scoring images.

B. GAs in Art with Optimized Operators

The reproductive step of GA evolution is limited by the variety and methodology of its operators. In [1], authors propose an "Improved Genetic Algorithm" and explore crossover and mutation optimization to promote a more diverse set of candidate solutions. Parameter-Free GAs (PfGA) automatically adapt their parameters over the duration of a run to more closely match desired results.

C. GAs Incorporated with Artistic Decision-Making

When incorporating GAs in image evolution and creation, some try to increase human involvement while others seeks to automate processes mitigating human intervention [2]. In this work, the authors utilize a "sitter image" which the algorithm utilizes to create derivative works based on some designated aspects of the human creative process. The authors acknowledge that the artistic choices of humans are based both on structural limitations and an understanding of form as well as incorporation of breaking or manipulating those constructs with associative thought.

IV. Methodology

In order to construct results complex enough to showcase the effectiveness of the GA, but avoiding excessive computational overload, 736 bits were designated to each chromosome. Each canvas would contain 32 shapes with 23 bits of attribute values divided into shape, size, color, the existence of a tint or shade and which, rotation and position. This means that each chromosome is 736 bits. Each canvas is 512x512 pixels.

A. Chromosome Components

1) Shape: The first 3 bits of the chromosome indicate which of 8 shapes will be drawn onto the Canvas. The first shape in the chromosome will be drawn on the "bottom layer" and each shape in succession will be overlayed in a new layer on top of its predecessors. This is important to note as fitness function scoring rewards interesting shapes formed with this methodology and it punishes some situations. For example: the top shape at maximum size covers all shapes below. Each shape is represented as follows:

TABLE I
Shape Representation

Shape	Bit Representation
Line	000
Circle	001
Triangle	010
Rectangle	011
Pentagon	100
Hexagon	101
Septagon	110
Octagon	111

2) Size: 6 bits of the chromosome are dedicated to shape size. Size applies to both the x and y axes each of which receive 3 bits of information. This means that equivalent values will produce a symmetric shape, whereas differing bit values will produce an oblong or skewed shape. The maximum shape size is 512 pixels meaning that a square of the largest size could cover the canvas entirely. A chromosome's final shape with this property would produce a "blank" or mono-colored canvas while the first shape with this property could provide an interesting background color for the rest of the shapes on the canvas.

TABLE II
SIZE REPRESENTATION

Size (X or Y)	Bit Representation
4 px	000
8 px	001
16 px	010
32 px	011
64 px	100
128 px	101
256 px	110
512 px	111

3) Color: Overall, 5 bits are allotted to color values. The first 3 bits represent the primary (red, yellow, blue) and secondary (orange, green, purple) colors as well as black and white. The following sections describe how tints and shades are incorporated with 2 extra bits. This approach allows us to include a more diverse range of colors without filling extra bit representations randomly. The 8 base colors and their respective hex values are represented as follows:

TABLE III
COLOR REPRESENTATION

Color	Hex Value	Bit Representation
Black	#000000	000
Red	#ff0000	001
Orange	#ff6600	010
Yellow	#ffff00	011
Green	#008000	100
Blue	#0000ff	101
Violet	#6600ff	110
White	#ffffff	111

4) Tint Shade (Binary Y/N): Creating tints and shades by adding white or black, respectively, is a tool used by artists and allows us to program the fitness function more effectively to reward chromosomes with interesting color palettes (such as complementary or monochromatic palettes). 1 bit is allotted to give a binary yes or no. If the value here is a "yes," or "1," the following bit, discussed next, will indicate which value (black or white) will be mixed with the base color identified earlier. If the value is "no," or "0," the base shade will not be made into a tint or shade and the subsequent bit value can be disregarded.

5) Tint or Shade: 1 bit is allotted to the black and white values which will be added to a base color given that the previous bit is a "1."

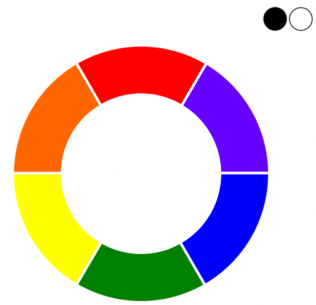

Fig. 1. Color Wheel Representing 8 Base Colors.

TABLE IV
TINT OR SHADE REPRESENTATION

Tint or Shade (Yes or No)	Bit Representation
NO	0
YES	1

TABLE V
TINT AND SHADE REPRESENTATION

Black or White	Bit Representation
Black	0
White	1

Now, we have created the tints and shades of each base color. Their respective hex values are represented in Table VI.

TABLE VI
SHAPE REPRESENTATION

Color	Shade Hex Value	Tint Hex Value
Black	#000000	#7d7d7d
Red	#7f0000	#f7787f
Orange	#7f3300	#f7aa7f
Yellow	#7f7f00	#f7f67f
Green	#004000	#79b77f
Blue	#00007f	#7978fd
Violet	#33007f	#ab78fd
White	#7d7d7d	#ffffff

It is worth noting that the 8 base colors shown in Figure 14 have a statistically higher chance of being chosen due to the chromosome structure. If the 10th bit which decides if a base color will be mixed with a tint or shade is a "0," then the shape's color is limited to 1 of 8 colors meaning each has a 6.25% chance of being selected. If the 10th bit is a "1," then the base color will be mixed with a tint or shade (designated by the 11th bit) and will be one of "15" colors meaning each color on the extended palette has only a 3.125% chance of being selected. The number 15 in parentheses here as on the extended palette, the grey shade can be selected 2 ways, either if the base color is white and a shade is added, or if the black color is the base and a tint is selected. This means that while the other colors in the extended palette have a 3.125% chance of being chosen, gray has slightly higher odds at a 6.25%

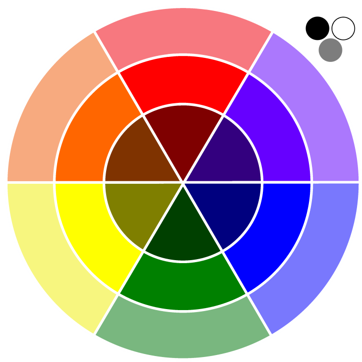

Fig. 2. Color Wheel Representing Extended Palette with Tints and Shades.

chance. Lastly, white and black are capable of being chosen in both palettes as black with the addition of a shade is still black, and white with the addition of a tint is still white. This means that black and white ultimatly have a 9.375% chance of being chosen. Grey, as well as the primary and seconday colors have a 6.25% chance of being chosen, and all other colors with either a tint or shade have a 3.125% chance of being chosen.

6) Rotation: 3 bits are allotted to shape rotation. PyCairo rotation works by rotating the entire user space. This means that without recalibrating the center of each shape, the context is rotated around the origin and shapes drawn near the canvas edges are rotated out of scope once the rotation is returned to 0 after drawing a shape. To combat this, before drawing each shape, their centerpoint is calculated and utilized so that the rotation is only applied to that shape in that specific loop iteration. The center is reset to the origin after completion. Degrees of rotation are represented as follows in Table VII.

TABLE VII
ROTATION REPRESENTATION

Rotation	Bit Representation
0	000
45	001
90	010
135	011
180	100
225	101
270	110
315	111

7) Position: 6 bits of the chromosome identify the shape's position on the canvas with 3 bits identifying position on the x-axis and 3 bits identifying position on the y-axis. The canvas can be thought of as a grid and the center of each shape can be snapped to the grid along the x and y axes. This is shown in Table VIII.

Shapes are prevented from being lost in the canvas edges by padding the grid with 32 pixels on each side. This creates interest by allowing larger shapes to extend off the canvas but prevents smaller shapes from being lost on the edges.

TABLE VIII
POSITION REPRESENTATION

Position (X or Y)	Bit Representation
32	000
96	001
160	010
224	011
288	100
352	101
416	110
480	111

Additionally, artworks attempt to pull you inward towards a focal point which is not often on the canvas edges. The positional grid is shown in Figure 3. It is of note that the surface origin is located in the top left corner as opposed to the bottom left corner. Many of the built-in Cairo shape-drawing functions utilize the positional parameters to represent the top corner of a shape rather than its center which had to be factored in to calculations in this project.

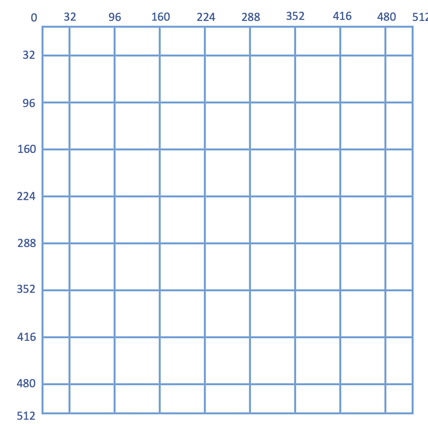

Fig. 3. Context Surface Grid.

B. Initializing the Population

The initial population is random. Each chromosome is 736 bits and the population size is 300 chromosomes. After each generation, 2% of the parents will be retained in the next generation meaning that each generation after the initial population will contain 306 chromosomes. The fitness scores of the retained parents are discarded and recalculated in the next round of scoring.

C. Drawing the "Artworks"

"Artwork" construction happens primarily within DrawShapes.py. The primary import library is Cairo, or more specifically PyCairo. This library allows for the creation of a context surface and for ease of shape generation. It has some built-in shapes like circle and rectangle and supports the visualization of other polygons by using point-to-point calculations. Given the chromosome structure, attributes such as position and size are utilized to draw the shapes. After the canvas has been drawn, it is saved as a PNG and converted into a Numpy array whose features are examined and scored by the fitness function. This feature also allows canvases to be saved periodically for examination throughout fine-tuning. In final runs, the most optimal chromosome is saved as a PNG.

D. Scoring the Pieces Utilizing GA Fitness Function

The fitness function is written to consider both the chromosome itself and the numpy array representing the actual canvas. The fitness function scores on the design elements from Section II B. Symmetry, color harmony, and shape diversity are applied and the weighted importance of each is toggled to examine impact on GA results.

1) Symmetry: Symmetry is the element that is scored using a numpy array. The function takes the array as input and calculates its symmetry across the vertical, horizontal, and diagonal axes. The final symmetry score can be toggled to weight components more or less strongly and otherwise is calculated as a mean of each score. The grey background color is not shared by the grey in our color wheel. It is disregarded when computing symmetry scores to prevent error in calculation.

2) Color Harmony: Interest can be created in pieces by utilizing various color palettes. The color palette function takes a chromosome as input and calculates scores for usage of either a monochrome palette, a complementary palette, and a palette consisting of the most diversity. The weights can be adjusted across palettes to promote a specific type of harmony, but the final returned score is not an average as a piece that receives a highly-scoring monochrome palette will also have a low diversity and averaging the scores would negate the purpose of

3) Shape Diversity: In order to promote complex compositions, a score identifies how homogeneous or diverse a set of shapes is by using Shannon Entropy. This score can be toggled up or down to encourage compositions to utilize more or less shape diversity.

V. Results

The results of this project can be seen through the progression from a completely random chromosome's display to a chromosome that has undergone genetic evolution.

Figures 4 is a display of a randomly-generated chromosome from the initial population. There is no obvious color harmony and any elements of symmetry or diversity are coincidental at this stage.

Surprisingly, high-symmetry weights produced somewhat "poor" results. As symmetry performs point-to-point calculations this made sense. At first, final evolutions often produced primarily solid canvases or extremely narrow shapes as either the largest shapes or the background color were unintentionally producing the most symmetry. This can be seen in Figures 5 and 6. Symmetry scoring was reprogrammed to disregard the background color when calculating symmetry. This change is reflected in all figures after Figure 6.

The most interest was found in evolutions where the color harmony and shape diversity were heavily weighted while symmetry received minimal importance. Such instances can be seen in Figures 7 and 8. The color harmony creates interest as overlapping shapes of the same hue form complex forms. Shape diversity promotes interest by including a mixture of rounded and hard edges.

As the images were evolved over 300 generations, the program saved the high-scoring chromosome after every 30

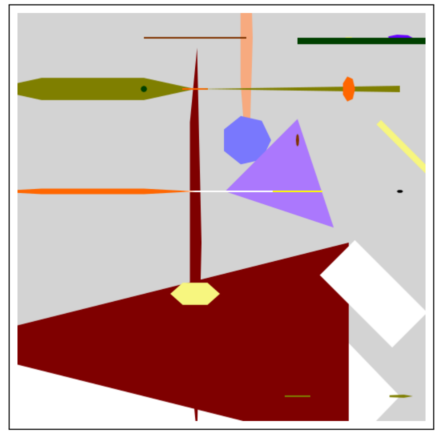

Fig. 4. Randomly Generated Chromosome from Initial Population

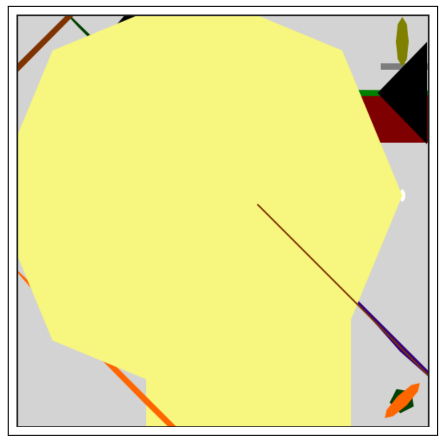

Fig. 5. Symmetry Weight 0.8

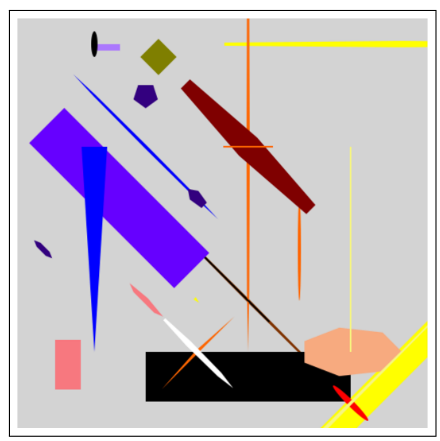

Fig. 6. Symmetry Weight 0.8

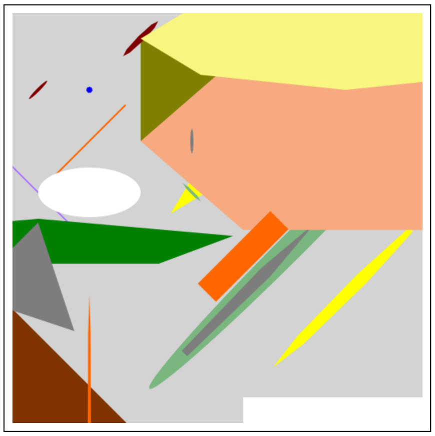

Fig. 7. Color Harmony Weight 0.5 and Shape Diversity Weight 0.5

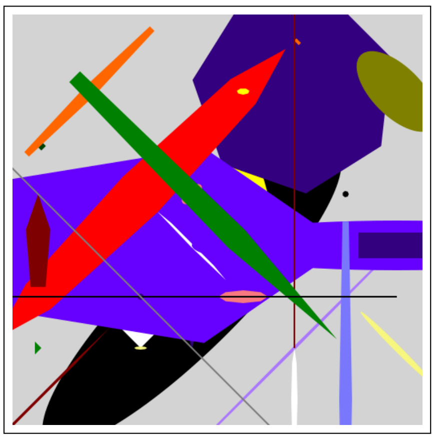

Fig. 8. Color Harmony Weight 0.5 and Shape Diversity Weight 0.5

generations until completion. This can be seen in Figures 10-14. The generational number as well as the fitness score are denoted for each canvas in the figure. It is of note that not only do scores converge to 1 almost immediately, but even exceed 1 which should not be the case as fitness function scores are normalized in both their individual components and their final score. The early convergence is likely due to the simplicity of the fitness function. Even with a small population size, the likelihood of one of the chromosomes receiving a high score across all categories and weights is exceedingly high.

The initial population size was set to be 1000 and the convergence can be seen in Figure 10. The time taken to evaluate one generation of this population size was approximately 68.79 seconds meaning it would taken about 19.1 hours to iterate 1000 generations. Due to the fact that the high-scoring chromosome of the first generation already displayed convergence led to scaling down the population size and number of generations.

It is of note that to combat the high symmetry scores of much larger shapes, the scale of the shapes was scaled down for some tests. In Figure 14, instead of shape sizes mapping to 4 px, 8 px, 16 px, 32 px, 64 px, 128 px, 256 px, 512 px, they instead were mapped to 2 px, 4 px, 8 px, 16 px, 32 px, 64

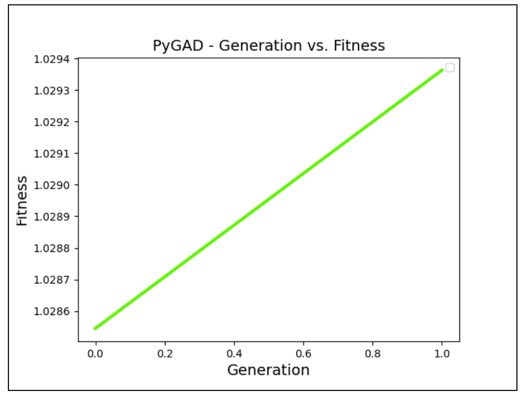

Fig. 9. Convergence of Fitness Score with Initial Population Size of 1000

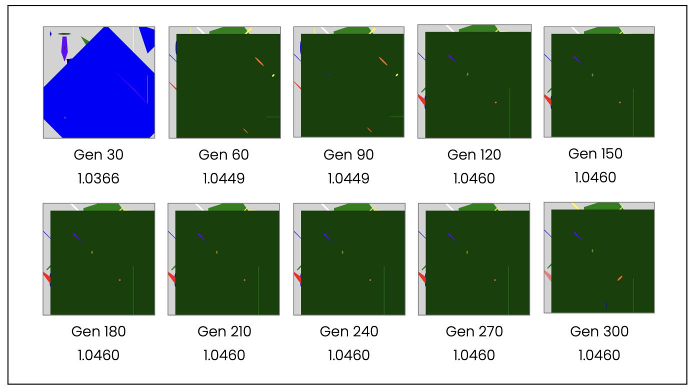

Fig. 10. Symmetry (0.33) Color Harmony (0.34) Shape Diversity (0.33)

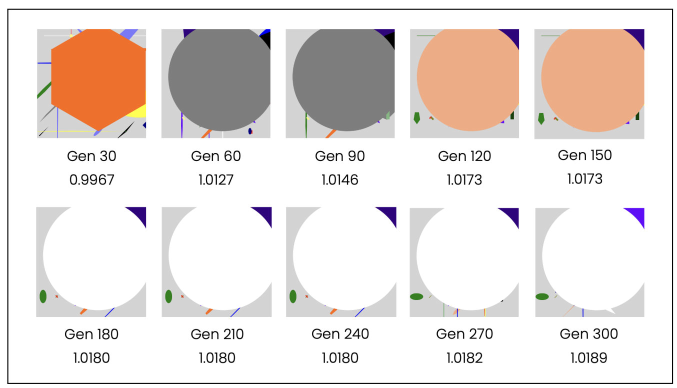

Fig. 11. Symmetry (0.8) Color Harmony (0.1) Shape Diversity (0.1)

px, 128 px, 256 px. This arguably produced more "interest" in the resulting canvases but did not solve the problem of early convergence.

VI. Future Work

Corrections to current work include fine-tuning the symmetry function to promote more subtle aspects of shape symmetry as opposed to rewarding canvases with extreme amounts of solid color. It is also possible to reward large shapes in the "background" (those which appear as one of the first shapes in a chromosome) while punishing large shapes appearing in the foreground which leads to the coverage of any interesting

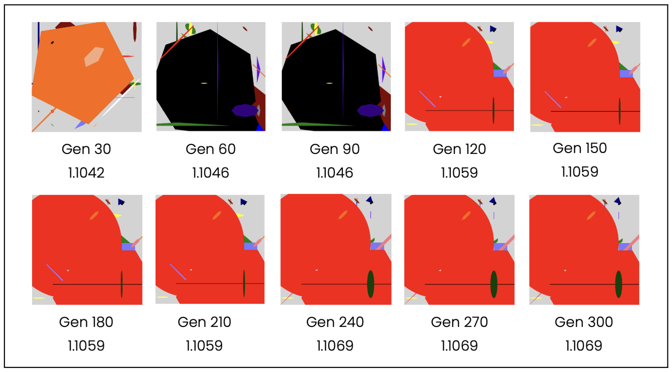

Fig. 12. Symmetry (0.1) Color Harmony (0.8) Shape Diversity (0.1)

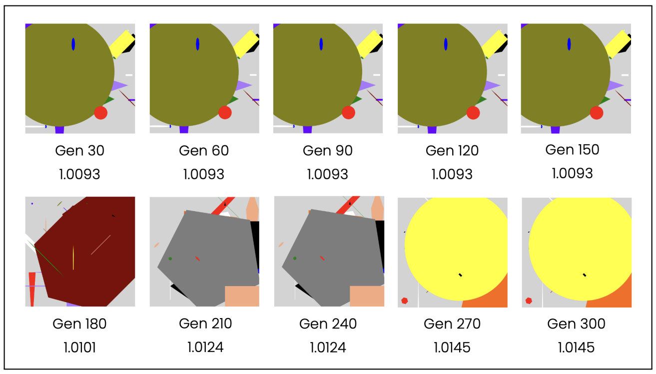

Fig. 13. Symmetry (0.1) Color Harmony (0.8) Shape Diversity (0.1)

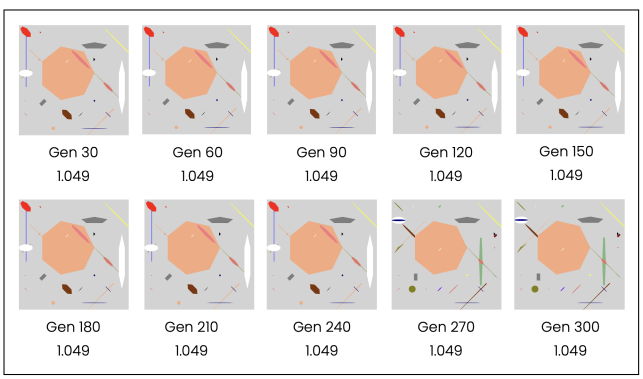

Fig. 14. Symmetry (0.4) Color Harmony (0.4) Shape Diversity (0.2)

elements lost underneath.

Future work consists of extending the scope of design elements considered by the GA fitness function. When more elements are considered, it becomes more difficult to discern how toggling different element weights affects the overall composition; however, too few fitness function components lead to premature convergence as was seen here.

It is also possible to increase the amount of attributal information in each chromosome. Some potential implementable features include stroke, color gradients, and background fill colors. Increasing the variety of shapes, colors, positions, etc. is another way to advance the current project scope . Exploring parameter alterations is another avenue for future work. Utilizing other selection or crossover types as well as a much higher mutation rate may produce more optimal results.

Acknowledgment

The author would like to thank Dr. Arvin Agah for his support throughout the duration of this project and for providing foundational knowledge about genetic algorithms.

References

[1] Chen, Shaoqiang. (2024). Genetic Algorithm Based on Operator Optimization in Illustration Art Design. Journal of Electrical Systems. 20. 337-342. 10.52783/jes.2647.

[2] Dipaola, Steve & Gabora, Liane. (2010). Incorporating characteristics of human creativity into an evolutionary art algorithm. Genetic Programming and Evolvable Machines, 10(2), 97-110. Genetic Programming and Evolvable Machines. 10. 10.1007/s10710-008-9074-x.

[3] Jones, Mara E. & Agah, Arvin. Evolution of Digital Images. IEEE. 10.1109/TSMCC.2002.80449.