An Exploration of Different Techniques for Artificially Producing Art

Matt Chapeski, Dalon Lobo, Donald Jewkes
St. Francis Xavier University
Department of Computer Science
Antigonish, Nova Scotia, Canada
x2018yov@stfx.ca, x2020fyh@stfx.ca, x2017uzp@stfx.ca

***Abstract*—Art has historically always been thought of as a subjective, creative outlet. It has long been a defining factor of the human experience, that because we are sentient beings, and can feel and process emotion, we are the only beings that can truly produce what we commonly know as ”art”. However, an interesting question that arose in the 1990s was the idea of computer generated art, using Darwinian evolutionary operands. This idea was proposed by Karl Sims in a paper titled ”Artificial Evolution for Computer Graphics” [8]. Evolutionary Art can be implemented with many strategies, and with different goals. In this project, we seek to perform an exploratory analysis of different approaches towards evolutionary art, to see what results can be yielded from various approaches. For this project, multiple fitness based approaches are to be examined, where different interpretations of crossover and mutation will be applied to various images, to see what kind of images form as a result of the experimentations.**

***Index Terms*—Evolutionary Computation; Evolutionary Art, Genetic Algorithms, Fitness-based Search**

I. Introduction

Art has been around for as long as human beings, and human-related ancestors have been around. Art has been a key aspect of culture, communication, and the overall human experience. Much like human beings, it has been argued that art has evolved over the course of millennia to become much more complex and deep in meaning, and that this evolution can be thought to be happening in tandem with human evolution.[1] Human evolution, and on a broader scale, biological evolution, as defined by Charles Darwin, takes place over several generations, and as such, it can be difficult to encapsulate and take note of evolutionary processes taking place in one lifetime. [2] However, in 1958, Computer Scientist R.M Friedberg described in an article the usage of evolutionary processes for solving computing problems. [3] This article laid the ground work the process of evolutionary computation, a field of study in Computer Science that uses evolutionary processes such as mutation, crossover and selection, to evolve a population of data over the course of multiple generations to try and find an optimal solution to a problem.

This discovery pioneered the field of evolutionary computation, and many subfields of evolutionary computation have been created, such as genetic programming, the study of utilising genetic operators to find the best set of operands to solve a problem, [4] genetic algorithms, which is the process of utilising various genetic operators such as crossover and mutation to find an optimal solution in a population of possible solutions to a problem, [6] and particle swarm optimisation, which seeks to mimic the patterns of birds and other similar ”swarming” species to find a local optimum for some function or problem by using a swarming technique. [7]

It was in the 1990s that a Computer Scientist by the name of Karl Sims first used one of these ideas from Evolutionary Computation, the Genetic Algorithm, to try and create pieces of art that utilised evolutionary operators, such as mutation and crossover, to produce novel works of art, entirely artificially.[5][8]

This method proved to be successful in creating artwork, however other methods have been explored in the past, as most implementations are fitness based, the resulting artworks always converge towards an optimal solution, and do not explore the different, more novel outcomes that could have been produced over the set amount of generations. This is where Vinhas et al. in 2016 developed a novel approach of creating evolutionary art. The paper described a hybrid approach between a novelty search and a fitness based search for finding an optimal image, and was able to produce a wide range of phenotypically diverse images, outperforming the fitness based algorithm in this category. [9].

For this project, multiple fitness based approaches will be explored in detail, to examine what kind of images are produced, and if the images themselves are aesthetically pleasing.

II. Methods

For this project, multiple fitness based approach was used. For the genetic operations, crossover and mutation were defined in terms of a coloured 2D image. Coloured images have three colour channels, one for red, one for green, and one for blue. With regards to crossover and mutation, all implementations followed a similar structure. The crossover/mutation was to be conducted by modifying the values within an RGB channel.

Because channel values simply set the colour of the pixel for an image, most implementations of crossover do not need to worry about breaking the integrity of the image within the population.

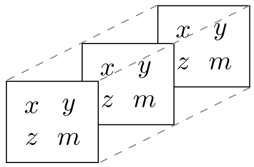

Fig. 1: An example of an RGB Matrix of some image, where $x, y, z, m \in \mathbf{N}^0 \ni x, y, z, m \leq 255$

Above is a mathematical representation of a hypothetical RGB matrix for some image. For our first implementation of crossover, the crossover simply swapped the values of one RGB channel, with the values of the same channel from another image. This implementation is rather basic, however still drastically changes the image, depending on the difference between the values being swapped. For the second crossover implementation, much like the first implementation, the RGB values from some channel for some image are swapped with the RGB values from the channel of another image, however, for this version of crossover, a bounding box was introduced, which served to isolate parts of the image that would be susceptible to being crossed over. The third implementation of crossover, crosses over an entire row, or column, of pixels between two images. The fourth crossover is a sort of meta implementation of crossover. It chooses one of the three previous implementations of crossover to perform on the two images.

For mutation, similarily, the modification was done with regards to the values inside of the RGB channels of the image. For the first mutation, values of two RGB channels within a single image are swapped. This is a simple mutation, much like the first implementation of crossover, however, depending on the values within the channels, it could produce a significant change. For the second mutation implementation, much like the first implementation, the RGB values from some channel for some image are swapped with the RGB values from the channel of another image, however much like with the second crossover implementation, for this version of mutation, a bounding box was introduced, which served to isolate parts of the image that would be susceptible to being mutated, and thus have RGB channel values swapped. The third implementation of mutation is the same as the second in structure, however, instead of swapping the values between two RGB channels, the values are changed to random values between 0 and 255. The final implementation of mutation selected 100 pixels, and randomly changed their values to a valid RGB value.

Two different fitness metrics were used, one calculates the similarity between the reference image, and the image that has been evolved using the Structural Similarity Index Measure (SSIM) formula, [10] where a higher score, is an indicator of being more closely resembling the original image, and thus, being selected. The other fitness metric used evaluated the global average distance between pixels, and attempting to minimise that distance, using the mean squared error metric. Selection was done using simple tournament selection. The populations of images were either defined randomly, where RGB values were filled at random, or by using a pre-determined collection of images.

III. Results and Discussion

For each experimentation, the genetic algorithm (GA) was run for 500 generations, It was found that when crossover was run over the images, without a bounding box in place, that convergence would take place after approximately 300 generations. Seeing as there is no bounding box selecting a portion of the image to perform crossover over, the crossover function can take advantage of the entire image, which is what likely yielded an early convergence from using the first crossover implementation. It was further noted that diagonal trends tend to exist in the newly formed images that had been evolved over the 500 generations. Often it is the case that there is some symmetry between the top left, and the bottom right halves of the image, and this likely is the result of the way in which the crossover function selects x and y values.

Fig. 2: A piece of abstract art passed through the GA using the first crossover function

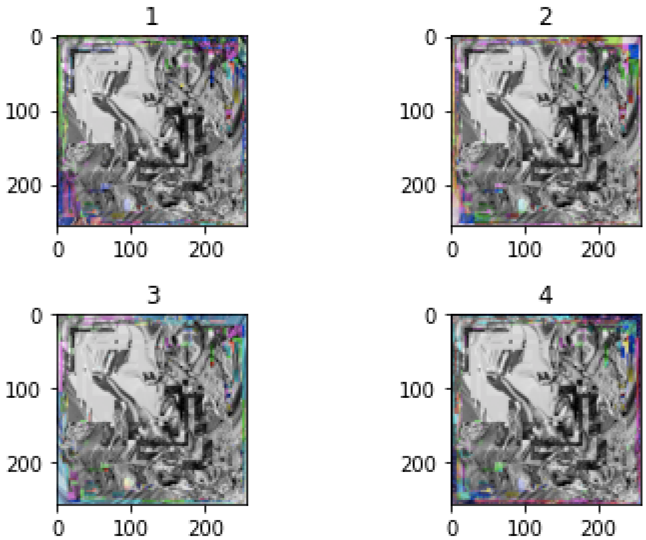

Fig. 3: Collection of final images generated using the first crossover function

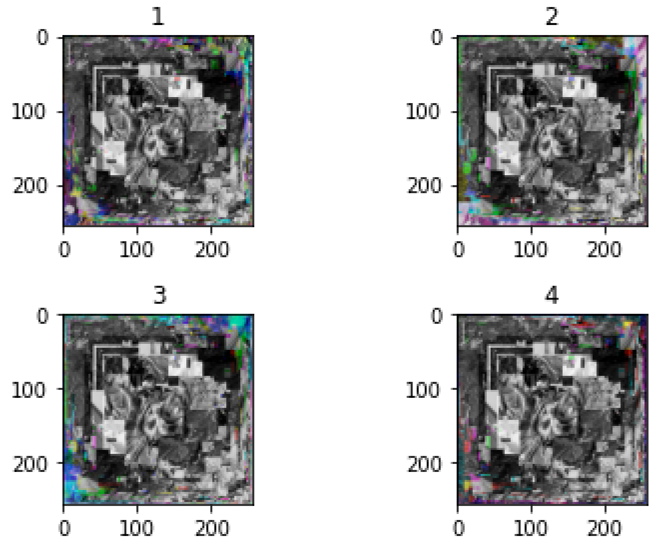

Fig. 5: Collection of final images generated using the first crossover and mutation functions

As seen in the two images included above, we can see a loose capture of certain general features present from the initial population that tend to be converged upon, and furthermore, it is noted that by using all channel crossover, the GA tended to converge on black and white images.

Next, mutation was added, and the first mutation function was utilised in conjunction with the first crossover function. The mutation rate was set such that the probability P of the image mutating was 0.30. ($P(\text{Mutation}) = 0.30$)

The results were quite similar to when just crossover was employed to evolve the image population, however unlike when simply using crossover, the images tended to form their own unique features that happen to be more independent than when simple crossover was used. This leads us to believe that the introduction of mutation has successfully introduced more genetic diversity into the population. Furthermore, there is still a strong symmetry on the diagonal, much like with the usage of only crossover.

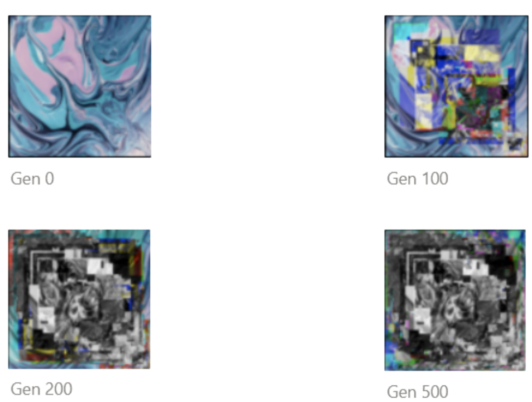

Fig. 4: Evolution of an image using first crossover and mutation function

The next experiment was based on the introduction of a boundary box to the crossover and mutation functions, meaning that crossover and fitness could only be applied to specific parts of the image, and not to any pixel in the image as a whole.

This experiment utilised the second crossover and mutation functions, which are identical to the first crossover and mutation functions described in the Methods section, however as explained above, a bounding box on where crossover and mutation could occur was applied to the images. It was noted that the images did not converge nearly as quickly as before, where the genetic operators could be applied to any pixel in the image, and for a convergence to occur, more generations would be needed to converge on an ideal image. Another result of note, is that there was a much better maintenance of colour with images evolved through the GA using the second crossover and mutation functions, which lead to more interesting outputs.

First, parameters including a mutation rate of 0.3 was set, with a bounding box of 30 to evolve over 500 generations.

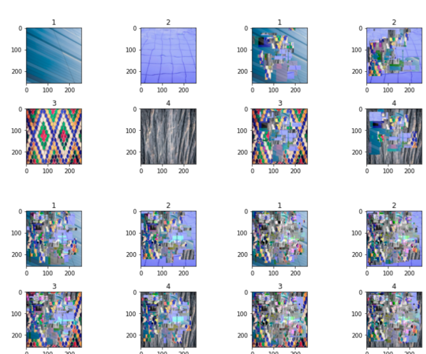

Fig. 6: Sample of images evolved using crossover and mutation with the inclusion of a bounding box

Next, a bounding box of 60 was defined with a mutation rate of 0.5, to be run over the course of 800 generations. These parameters seemingly had more feature capture, however unlike with the smaller bounding box and smaller chance of mutation, the image tended to want to evolve towards grey-scale again. Interestingly, one of the runs produced an image of what looks to be a smile with one eye.

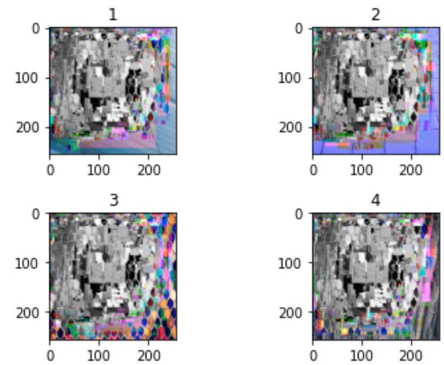

Fig. 7: Second crossover/mutation function, mutation rate of 0.5, Bounding box of 60

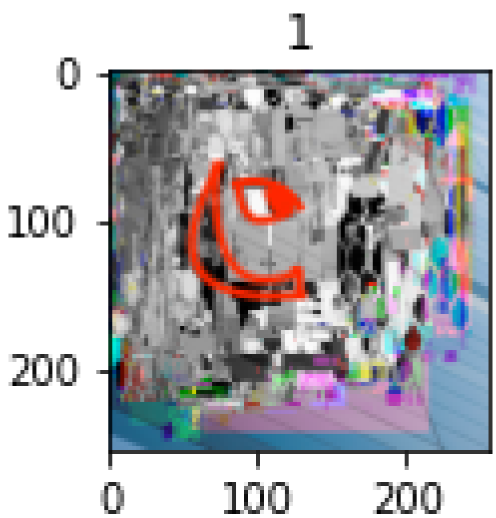

Fig. 8: The eye and mouth in the image

For the last run with the second crossover/mutation function, a bounding box of 70 was set, with a mutation rate of 0.8, and importantly maximum generation size of 10 000 was set. Furthermore, the single channel modification was enabled, meaning only one channel could be modified, instead of all channels being free for modification.

With these parameters, there was a convergence on an image from the starting population, however some novel features are still present. Interestingly however, the bottom right corner of all the images is seemingly less affected with these parameters in place because of the crossover implementation.

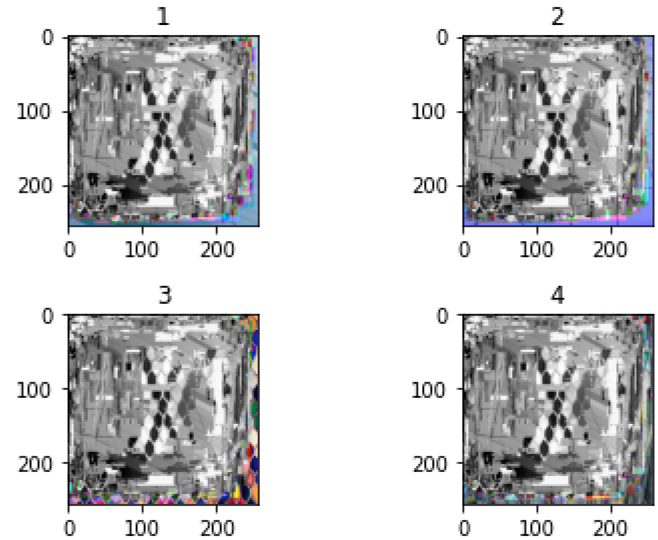

Fig. 9: Second crossover/mutation function, mutation rate of 0.8, Bounding box of 70, Single Channel Modification enabled

The next set of experiments utilised the third crossover function. As described in the Methods section, it produced a sort of linear crossover, with the crossover affecting an entire row or column of pixels. This crossover was paired with the third mutation function, which swapped RGB values with a random integer between 0 and 255, ($0 \leq x \leq 255$). With a newly defined starting population, the mutation rate was set to 0.1 and the bounding box was set to a size of 5. The GA was to run for a total of 1500 generations, with the single channel modification ON. With these parameters, it was found that the images simply turned white by the end of the 1500 generations.

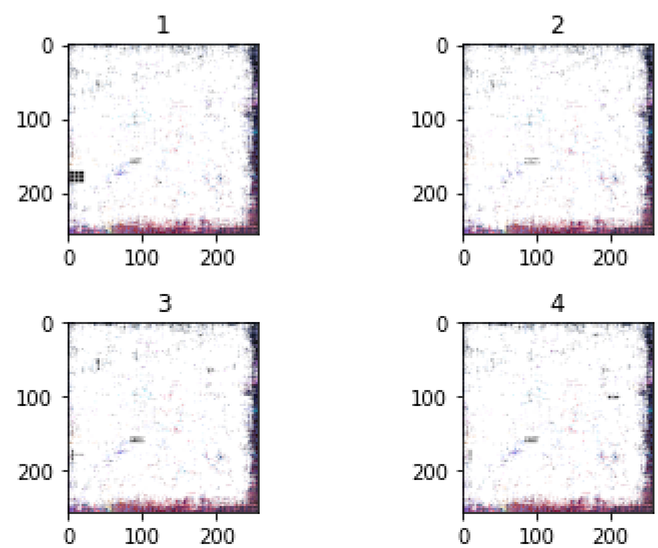

Fig. 10: Third crossover/mutation function, mutation rate of 0.1, Bounding box of 5, Single Channel Modification enabled

The third mutation function was exchanged for the fourth and final mutation function, as it proved to be quite difficult to get interesting results with the third mutation function. For this run of the GA, the third crossover function and fourth mutation function, as defined in the methods section was used, with a mutation rate of 0.1, bounding box with a defined size of 5, single channel crossover/mutation was enabled, and the maximum generation size was defined to be 5000. The usage of linear crossover appeared to ”collect” dominant features of all images within the population, and merged them all together. The best amalgamation of the images appears after approximately 3000 generations, where soon after, the image becomes much more noisy.

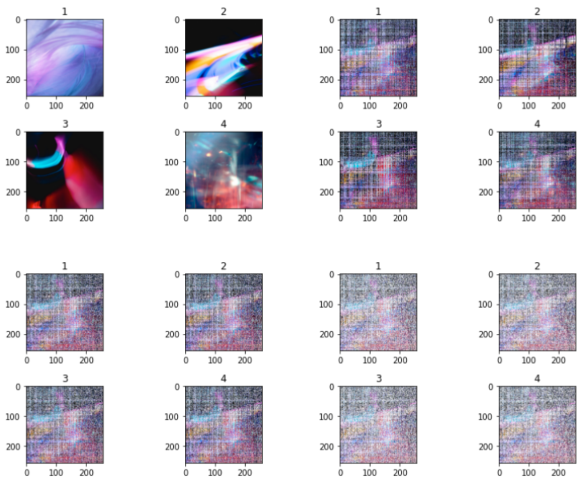

Fig. 11: Third crossover, fourth mutation function, mutation rate of 0.1, Bounding box of 5, Single Channel Modification enabled

By decreasing the generations to 1000, with a different starting population we start to get a more dominant image emerging by the end of the evolution run.

The next experiment was run using the fourth, meta crossover function, in conjunction with the fourth mutation function. The mutation rate was set to 0.1, bounding boxes were set to size 10, single channel crossover/mutation was used, and the GA was set to run for 500 generations. This experiment was run over a set of abstract art pieces, and it was found that there was substantially better colour retention.

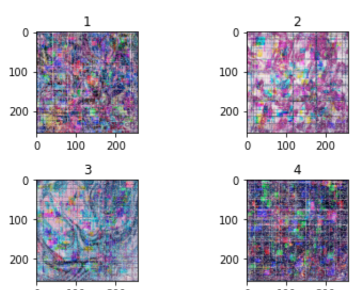

Fig. 12: Fourth crossover/mutation function, mutation rate of 0.1, Bounding box of 10, Single Channel Modification enabled

Increasing the mutation rate to 0.3, the bounding box to size 35 and increasing max generations to 1000 allowed for better organic feature development, where novel features seemed to have an easier time forming, however the final images were still quite noisy.

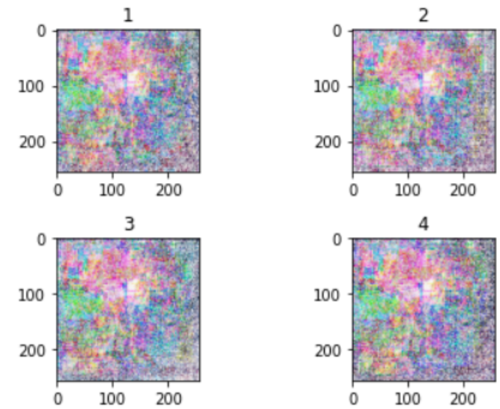

Fig. 13: Fourth crossover/mutation function, mutation rate of 0.3, Bounding box of 35, Single Channel Modification enabled, Max. gen of 1000

Outside of the defined experiments, an interesting result appeared when the second crossover function was combined with the fourth mutation function. The resulting pair seemingly evolved to give a sort of ”stained glass”, much like what is found in Christian churches and other houses of worship. The results were found when the mutation rate was set to 0.3, and the bounding box size was set to 25.

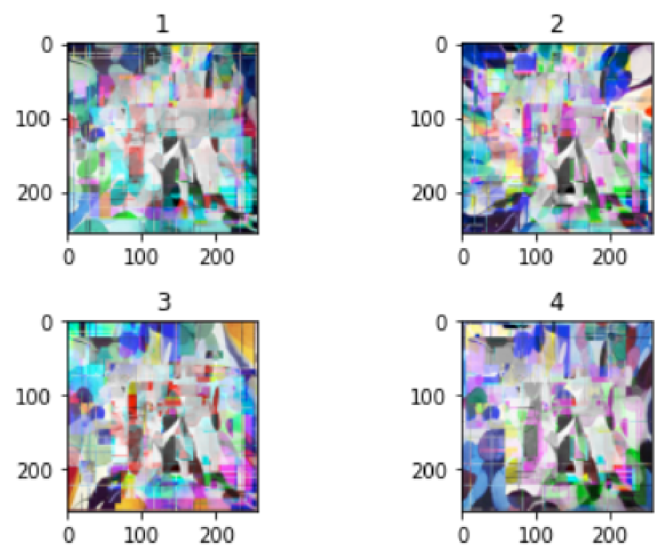

Fig. 14: Observed Stained Glass Effect

The final experiment involved testing the difference in outcome between two different fitness functions, Mean Squared Error, and the SSIM similarity function [10] with regards to feature extraction. The first test was run for 10 000 generations, and used the second crossover function, in conjunction with the fourth mutation function. This test utilised the mean squared error fitness function. We see some interesting novel feature formation, however the images do not seem to extract many features from the original image.

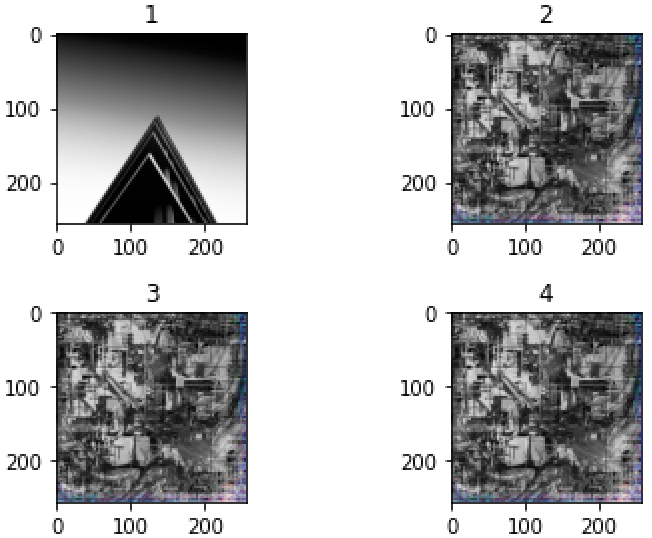

Fig. 15: Feature Extraction with MSE, 1st image is the target image

With using the SSIM similarity function [10], we were able to, with reasonable accuracy, map colours from the target image to the evolved image.

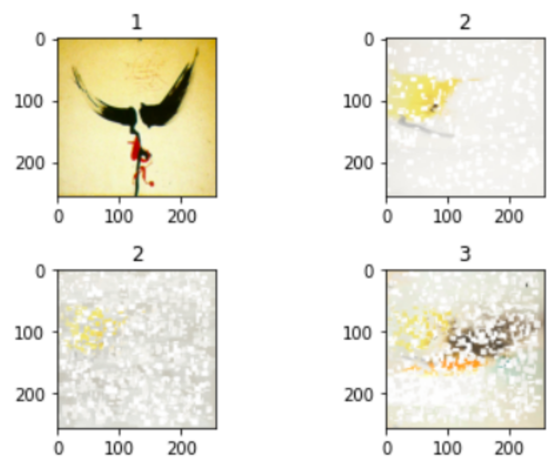

Fig. 16: Feature Extraction with SSIM, 1st image is the target image

IV. Conclusions and Future Work

In conclusion, these exploratory findings helped us understand that by changing how the crossover and mutation functions are defined, we can radically change the outputted image not only in terms of basic visual convergence, but also in detail, quality and overall aesthetic appeal. It is not possible to conclude which implementation is the best because evolutionary art is a subjective field. There is no definition of what ”good art” is, hence the creation of multiple art schools, and the study of aesthetics as a field of philosophy. Further work includes the introduction of a novelty search implementation, to be compared with the four fitness based approaches developed for this project.

References

[1] Noël Carroll. Art and human nature. *The journal of aesthetics and art criticism*, 62(2):95–107, 2004.

[2] Charles Darwin. *The origin of species*. PF Collier & son New York, 1909.

[3] Kenneth De Jong, David Fogel, and Hans-Paul Schwefel. *A history of evolutionary computation*, pages A2.3:1–12. 01 1997.

[4] John R Koza and Riccardo Poli. Genetic programming. In *Search methodologies*, pages 127–164. Springer, 2005.

[5] Nicholas Lambert, William Latham, and Frederic Fol Leymarie. The emergence and growth of evolutionary art: 1980–1993. In *ACM SIGGRAPH 2013 Art Gallery*, pages 367–375. 2013.

[6] Seyedali Mirjalili. Genetic algorithm. In *Evolutionary algorithms and neural networks*, pages 43–55. Springer, 2019.

[7] Seyedali Mirjalili. Particle swarm optimisation. In *Evolutionary Algorithms and Neural Networks*, pages 15–31. Springer, 2019.

[8] Karl Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 319–328, 1991.

[9] Adriano Vinhas, Filipe Assunção, João Correia, Aniko Ekárt, and Penousal Machado. Fitness and novelty in evolutionary art. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*, pages 225–240. Springer, 2016.

[10] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multi-scale structural similarity for image quality assessment. In *in Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers, (Asilomar*, pages 1398–1402, 2003.