

AI Assignment 2

Genetic algorithms

-Kamil Sabbagh-

Introduction:

In this task we are creating original art using AI and specifically genetic algorithms. A 512*512 pixels input image will be given and the aim is to create a program that will create an art using the given picture, various ways are applicable such as recreating the image using geometric shapes like triangle, circles, etc .. or using other images as a basis of the new image.

My approach was creating a mosaic from the given input and adding digital noise to the image, giving it a digital touch. Each piece in the mosaic shares the same (RGB) values.

Art:

Art is defined by **Aristotle** as the realization in the external form of a true idea, and is traced back to that natural love of imitation that characterizes humans, and to the pleasure which we feel in recognizing likenesses.

Aristotle defined art by the creativity and likenesses. My approach is to solve the image as a big puzzle using little mosaic pieces, and adding a touch of digital noise giving the sense of randomness created by the IA.

Chromosome Representation:

Each chromosome in the population is an image, each image consists of blocks of pixels sharing the same color. The number of blocks equals $(512*512)/(\text{area of the block})$. In my representation the area chosen for a block is 2*2 pixels. Therefore 65536 blocks in total. It is worth noting that the algorithms will consider only one pixel as a representative of the block making all the operations on the image much faster. All the other pixels will share the same color values as the representative pixel so there is no need to consider them until presenting the image.

Fitness function:

For the fitness function a chose to calculate the difference between each pixeles's (RBG) and the original photo pixel's (RBG). Therefore, each image will have an error score the sum of all differences of the individual pixels's (RGB) and the original's

Crossover:

Once two images were selected. A brand new image will be created.

For each block in the new image it will take the RGB values from one of the parents by random.

Mutation:

The Mutation can accrue during the crossover. Each block in the new image has a 3% chance to change one of its RGB colors to a random value.

Implementation of the algorithm:

1- Creating an initial population (Gen #0) with a 100 image. Each image consists of random blocks of pixels sharing the same random color.

2- Evaluating all the individuals by giving them the error score using the fitness function mentioned before.

3- Sorting the current population based on the error scores

4- the top ten images (10% of the total population) will be selected for breeding

5- two random images (from the top 10% selected individuals) will be chosen for breeding using the crossover function and new offspring will be adding the new population

Illustrations:

Example 1



Example 1 result 17617119 error score 86 generations (246 seconds)



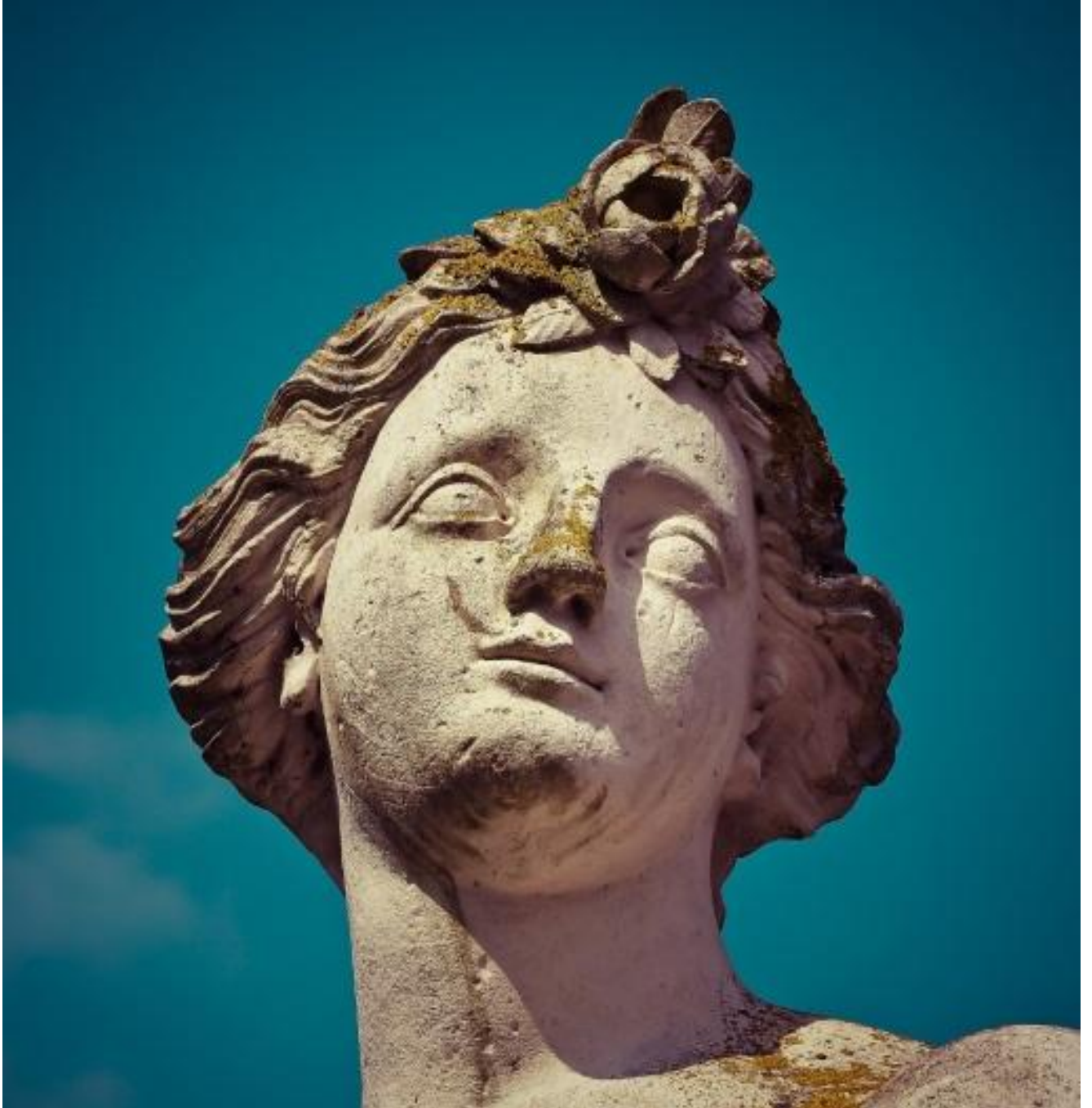
Example 2



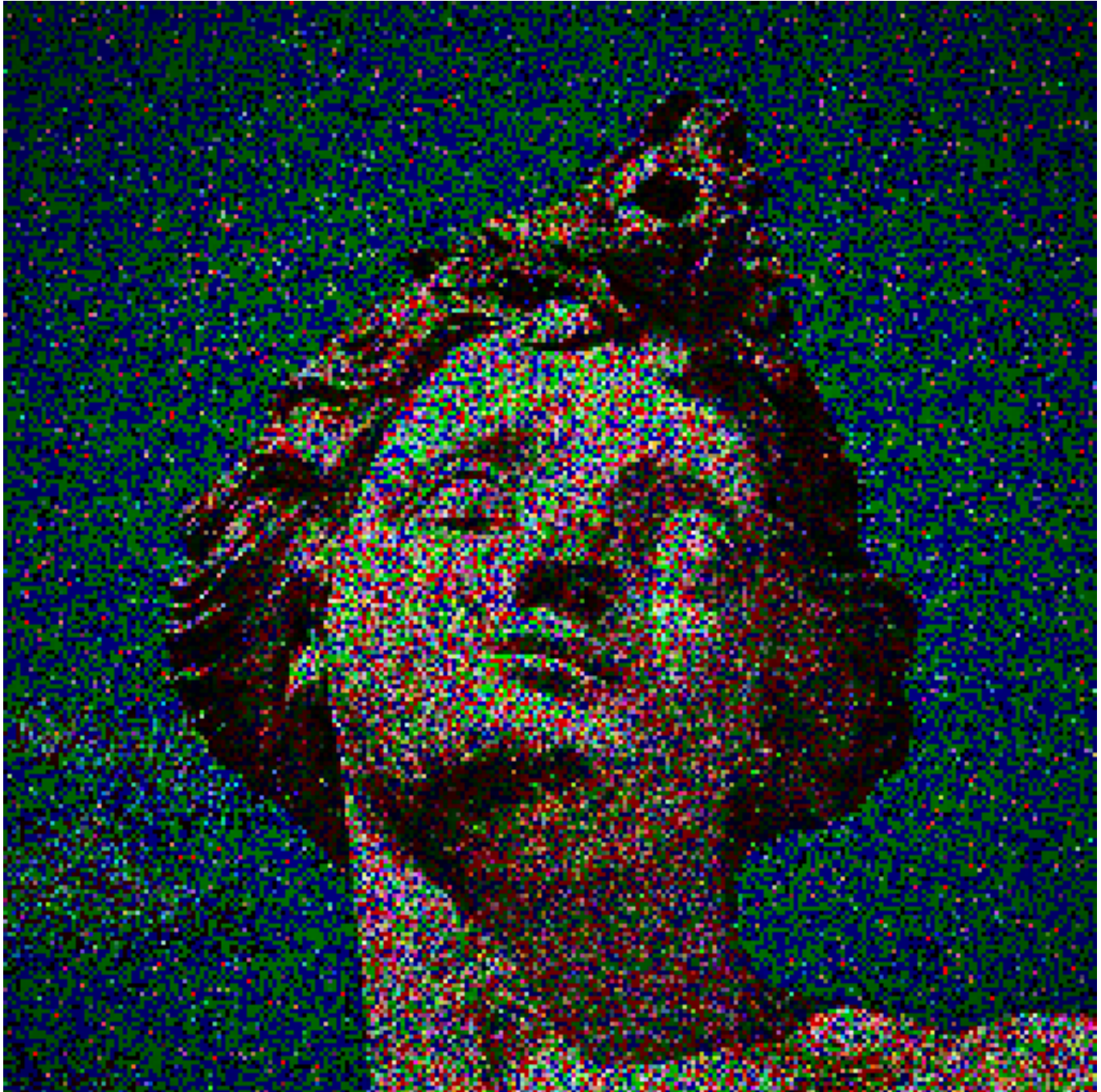
Example 2 result 27702009 error score 100 generations (266 seconds)



Example 3



Example 3 result 15758667 error score 78 generations (210 seconds)



Conclusion and thoughts by the creator:

We were asked to create a program that generates art. While there is no concrete method to determine how much things are artistic, I like to believe that anything creative, beautiful, and can “WOW” people should be considered as an art

Mosaic is one of the most beautiful arts, simple small pieces that alone is not worth looking at. But, put all these pieces together and you get an outstanding art that can be immortalized and looked for through eons.

In my approach I took this principle to the complete extreme. Each subpart of the image is completely random color blocks and here is where the beauty of the digital noise comes in place adding another layer of randomness and worthlessness to any small part (even 5% percent of the photo sometimes) as it will look completely random. In my humble opinion that gives the greater picture much much greater value.