AI Assignment 2: Report

by Irek Nazmiev, GitHub

What is the definition of Art?

This is one of the most controversial questions. After many centuries of the existence of human civilization, people still cannot come to an unequivocal conclusion.

Maybe the real Art is the eternal beauty or in a moment of explosion? For one the Art is the Life, for another - death. Someone sees the pure Art in drawings, anyone else - in music, cooking, admiring the nature, **anything**!

Oscar Wilde told: *"The beauty is in the eye of the beholder"*, and I completely agree with his deservedly great words.

\- *The art can be anything* -

But what is the art specifically for me, and more importantly - for my algorithm?

As the result I expect to get several geometrical figures: rectangle, triangle or circle of the different colors and positioned in some certain way.

The figures should be positioned in a such way that their colors are similar or close to the ones existing at certain positions on the origin image. But also there must be added some kind of new and unique things to the picture. It might be whether triangle or circle of a random color.

My art is not about restoring the origin image. My art is something **completely different** comparing to that picture, but **based on** that image.

My idea is not to provide the output with similar picture which looks like the processed origin one - **this is not art for me**! I want something unique, I want to let people to see different creatures/things in my picture, I want to provide the ability to see the image from different angles.

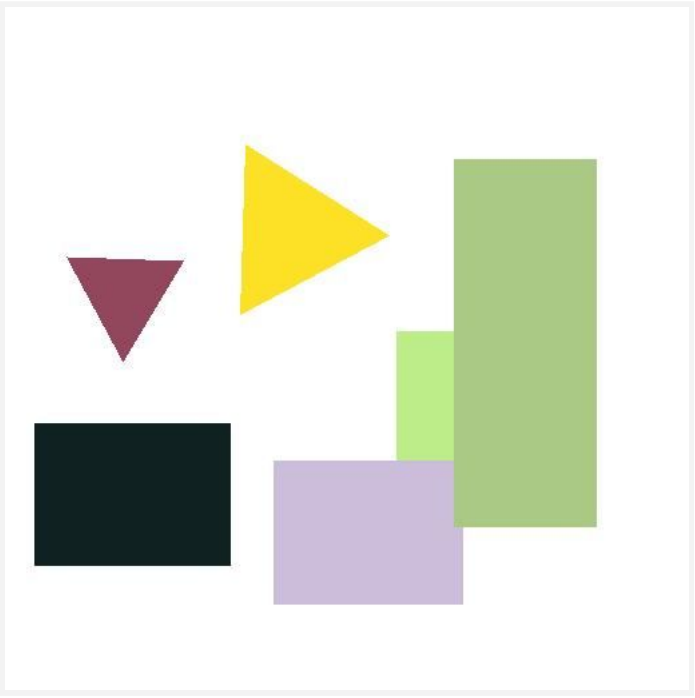
And I received exactly what I expected to get.

Initial input

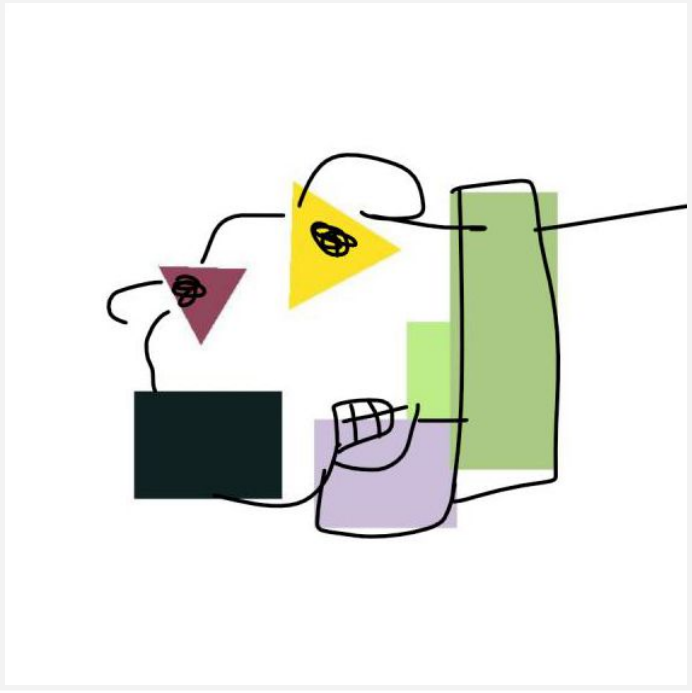
As the initial input I used the picture with a dog that I chose randomly from somewhere in my folders:

Result

As a result I've got the following picture:



This is exactly what I wanted to get. The picture is abstract and looks like a piece of suprematism, but this is incredible! The colors are similar to the ones from the origin image, adding a new one - lavender rectangle. I made a picture based on some origin one, but looking completely different from it. Also I brought something new.



This is how I see received picture. Someone else may see the image completely different, but that's the idea!

Implementation

Initial population

On this stage, the necessary amount (specified initially) of individuals is generated.

The individual's chromosome is the picture with the black or white background (chosen randomly) and the rectangle of some random color on it.

The rectangle's coordinates and color are saved as the individual's genes. Rectangles for my algorithm are the same as the brush strokes on the paintings.

Fitness function

On this stage, comparisons with the origin image happen.

From the individual, I take genes with the information about coordinates of the last added rectangle and its color. Then I compare pixels on the origin picture lying on the appropriate area of that rectangle (lying on individual's chromosome) with its color. I find the difference between each pixel on that area and rectangle color, and them sum it to the individual's fitness score.

Selection

On this stage, the whole population is sorted increasingly by the fitness score and then reduced by the survival coefficient specified initially.

Crossover

On this stage, the population is restored by crossing the best survived individuals.

Pairwisely, all the combinations of survived individuals are crossed in following way: the area of last added rectangle is copied from one individual to another (from worse individual to the better one). If the amount of combinations is not enough to restore the population to its initial amount, it's finally complemented by random creatures already existing.

Mutation

On this stage, the new figures are added with some probability to each individual.

The rectangle is added with 100% probability.

The triangle is added with 40% probability.

The circle is added with 15% probability.

All this stages (except the first one) are being repeated until the specified number of iterations is done.

All the code is available on my **GitHub**.