Report on Evolution Paint Algorithm.

by Uzbekova Katya, BS17-07

Description of the Algorithm:

- input:

  generate empty list for future picture and then generate 1st parent and then add to this parent other parents with some shapes at beginning

```
def initial_population(population_size, learning_rate):
    empty_img = np.ones((512, 512, 3)) * 255
    population = generate_random_shape(empty_img,
learning_rate).reshape((512 * 512 * 3))
    for i in range(population_size - 1):
        population = np.vstack((population,
generate_random_shape(empty_img, learning_rate)))
    return population
```

- Output: just picture, examples in Examples part
- Algorithm start when __main__ called and next Genetic Algorithms starts selection based on fitness_function,crossover, mutation and it will repeat for next 10000 iterations
- Fitness function just simple euclidean distance between output vector and population vector
- Selection a bit speed up by using numpy functions and select 5 out of 10 which is more closely related to original one

- Vector it means than every population picture transforms in one vector with size (512*512*3) as initial 3rd dimension picture
- Crossover in my implementation just simple doubling population which we will have after selection part
- Mutation part just simple add random circles at our population for now which we will have after crossover
- As a result of one iteration we have population which include population after genetic algorithm and population which we will have before mutation
- Also in this code exists learning_rate which decreases size of figures, at the beginning size is max and next goes to particular number
- Also there exist 2 functions: img_to_vector and vector_to_img which convert (512,512,3) to one vector of size(512*512*3) and vise versa.
- One more function, generate_random_shape which generates circle in random place at image with random color

At next section examples will present..

Examples

For this section I choose popular pictures to show how my algorithm will draw this picture with its skill.

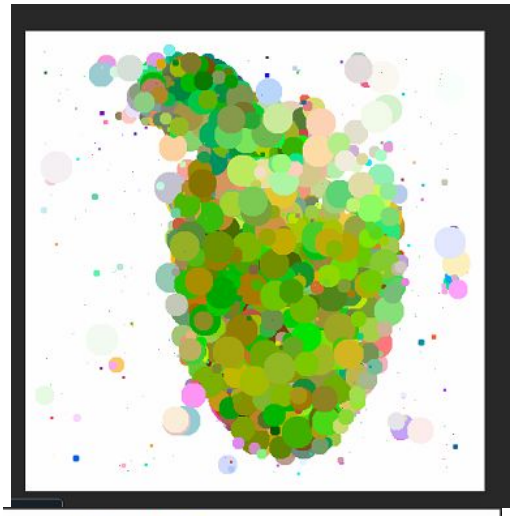
Mona Lisa

My Algo generation:

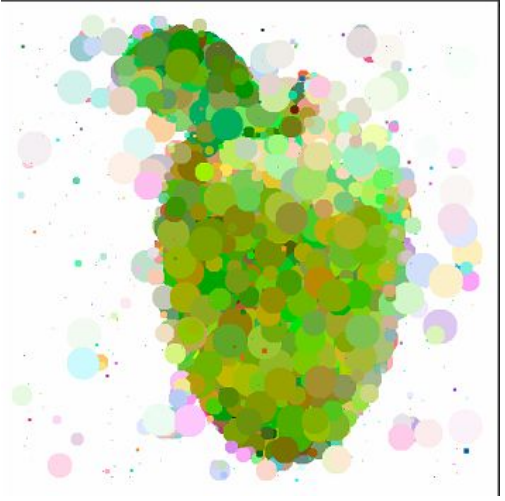
-3000 algo iterations

-10000 algo iterations

Apple

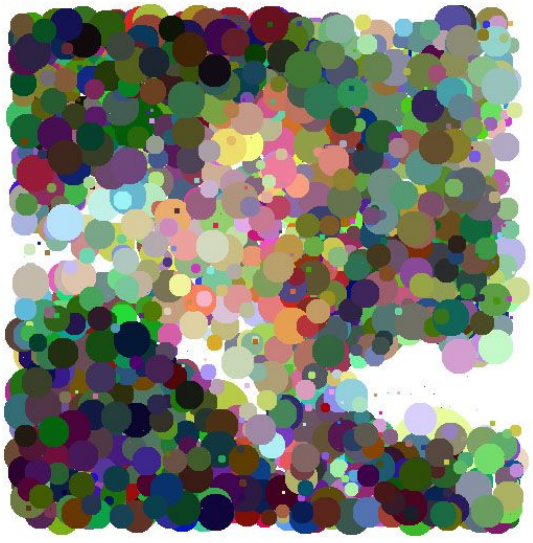


- 5000 iterations of algo

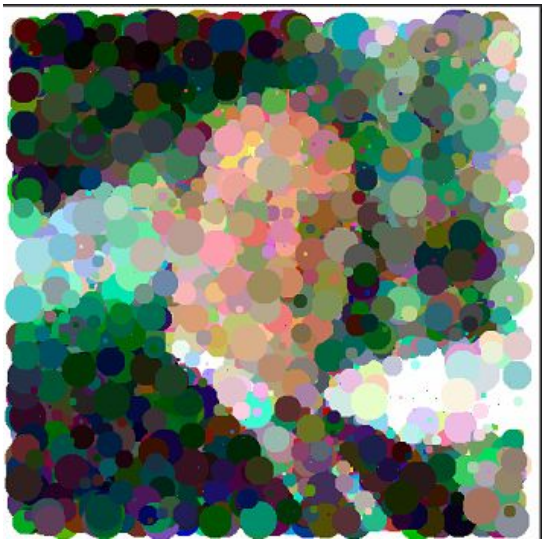


- 10000 iterations

Joseph Brown

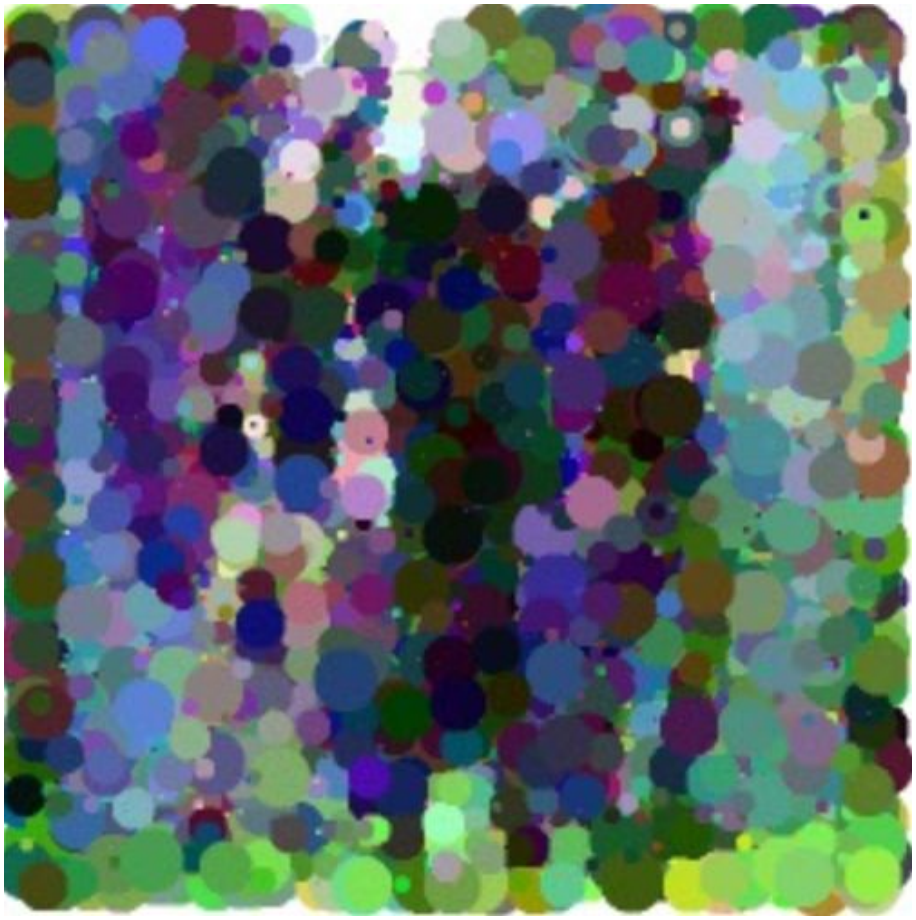


-3000 iterations



-1000 iterations

Photo From Halloween(reaction on photo)



-1000 iterations

Why artistic?

One picture drawing take a lot of time(up to 2 hours for just 10000 iterations), but, it also so artistic and I can prove it.

My painter draw similar picture and with more iterations picture will be more closer to original but it will be with few modification thanks to circle-drawing. Picture from code softer on contorous. So this just small difference adds charm to art.