Introduction to Artificial Intelligence

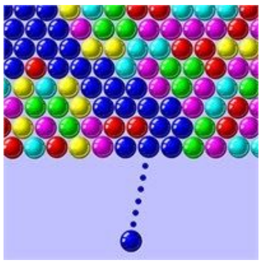
Assignment 2

Computers Can Do Art

Margarita Peregudova
BS18-05

Introduction

Artificial intelligence is developing rapidly. At the moment, it is used in many areas. In this assignment, we will write a genetic algorithm for creating a photo filter.
In our case art will be created from circles with random radius in range(0,X) and average color of polygons 3x3. Coordinate of centre of the circle will be in the centre of polygon.
Let's look at an example of the result of our algorithm.
When we created this photo filter, there was a question: why is this art?
In the definition from wikipedia **Art** is a diverse range of human activities in creating visual, auditory or performing artifacts (artworks), expressing the author's imaginative, conceptual ideas, or technical skill, intended to be appreciated for their beauty or emotional power.
So, we made set of shapes and colors, then it is art. The main idea of our photo filter it is made new image from initial image from colored circles looks like "bubbles" in old game.



Theoretical part

Fitness function

The distance is calculated between two colors. In our case, the distance between the colors of two points A1(red,green,blue,alpha) and A2(red,green,blue,alpha) will be calculated. The alpha value is not taken into account, since the image has a background, so it will always be 255.

$$Distance = \sqrt{(red_1 - red_2)^2 + (green_1 - green_2)^2 + (blue_1 - blue_2)^2}$$

Fitness function represent percentage ratio of distance.
Each color in range (0,255).

$q = 256 \times \sqrt{3} \approx 440$

$$Fitness = \frac{Distance}{q} == \sqrt{\frac{(red_1 - red_2)^2}{256^2} + \frac{(green_1 - green_2)^2}{256^2} + \frac{(blue_1 - blue_2)^2}{256^2}} = \frac{\sqrt{(red_1 - red_2)^2 + (green_1 - green_2)^2 + (blue_1 - blue_2)^2}}{256 \times \sqrt{3}} \approx \frac{\sqrt{(red_1 - red_2)^2 + (green_1 - green_2)^2 + (blue_1 - blue_2)^2}}{440}$$

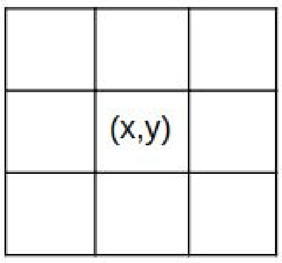
Population and mutation

In my case population consists of one parent and one child. Only one of the two survives and becomes the parent. We select the most suitable instance from the parent and child according to the fitness function. In my case, only the color of the circle mutates from parent to child. The mutation of the location and radius of the circle is senseless and ineffective. In my case I stored children and parents as set of objects of class Circle.

Practical part

Algorithm description

- We take .png image with size 512x512.
- We get some parameters of new image (maximum radius of circles on image α, quality of similarity of initial and results images β, number of circles on new image γ).
- After it we made γ circles with random color , random radius in range(0, α) and random location in range x and y in range (0, 512).
- Image with size 512x512 consist of 262144 pixels (512x512 = 262144). And we will change number of circles on canvas.
- After it we evaluate all circles. So, we take circle(color, radius, location).
    - We compare color of circle with color of points on initial image. Take a point (x,y) and calculate average color of polygon [(x-1,y-1),(x-1,y+1),(x+1,y+1),(x+1,y-1)] from initial image.
    - If value of fitness function of circle color and average color of polygon less than value which we set then we made child of this parent. Child has the same location and radius. Color was chosen randomly.



- If fitness function for child more than value which we set then we replace child with parent and take next circle, else we made new child.
  Kind of, for dog in picture 3.png and quality we have 262144 circles. And we made 793612202 childs. In average it 3027 childs for one parent.

- After it we made canvas from our results circles.

Requirements:

1) python3
2) library Pillow in python3
3) PNG image .png with size 512x512
4) time for computing image

How run code:

In Linux system we can run the code

```
>> python3 genetic_algorithm.py {name of image} {quality} {maximum radius of circles}
```

For example,

```
>> python3 genetic_algorithm.py 3.png 97 5
```

Details of code:

We take picture from folder **data** and save result to folder **results.** Logs of run the file will saved to file .log .

Settings:

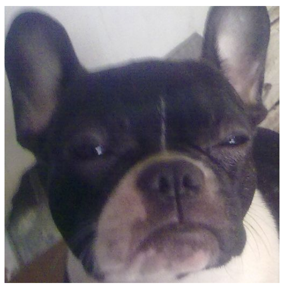
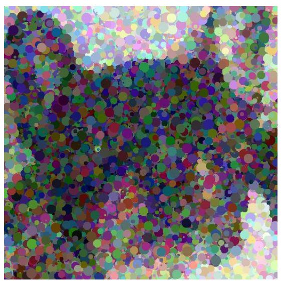
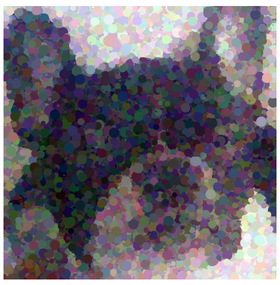
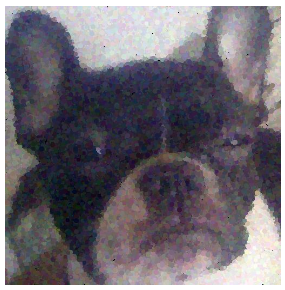
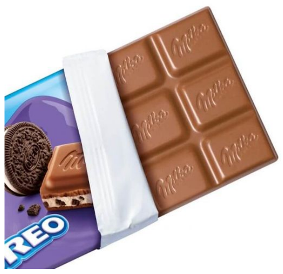
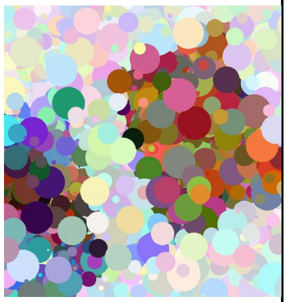
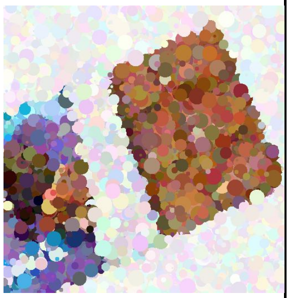
MAX_CIRCLE_RADIUS	- maximum radius of circles on image
SIZE	- size of image sides
NUMBER_OF_CIRCLES	- number of circles on new image
QUALITY	- quality of similarity of initial and results images
INPUT_NAME	- input name of image
OUTPUT_NAME	- output name of results image
OUTPUT_DIRECTORY	- directory for saving image
INPUT_DIRECTORY	- directory of input image
LOGS	- name of file for saving logs of computing

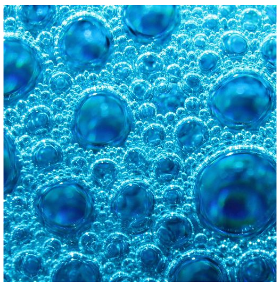
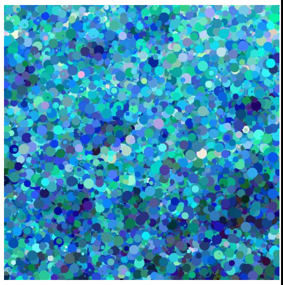
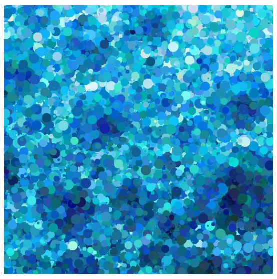
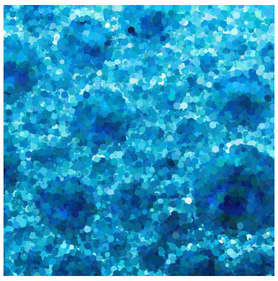
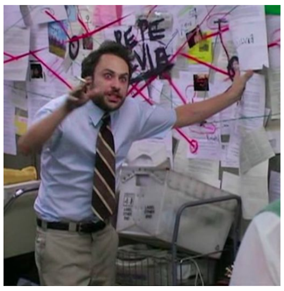
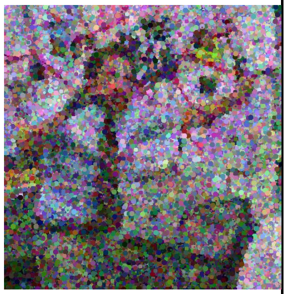
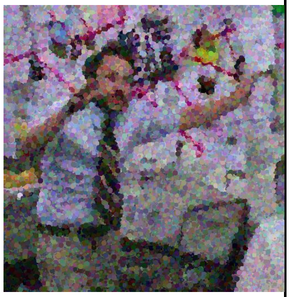
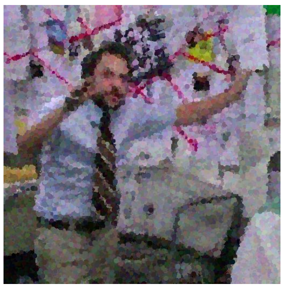
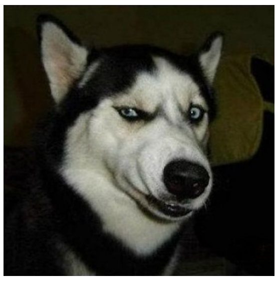
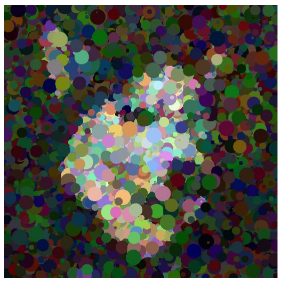
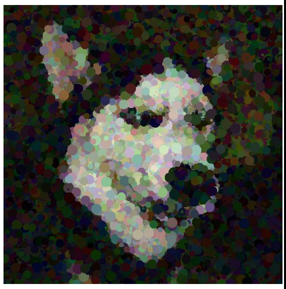
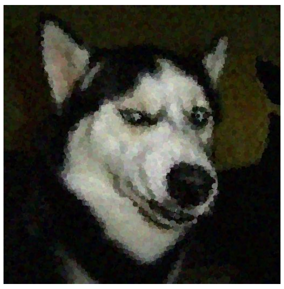
These parameters will affect to result image:

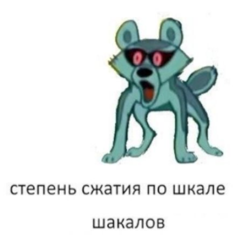
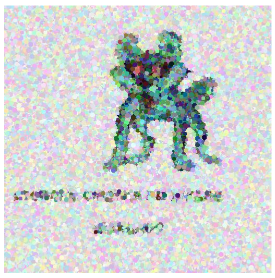
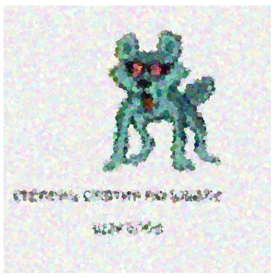
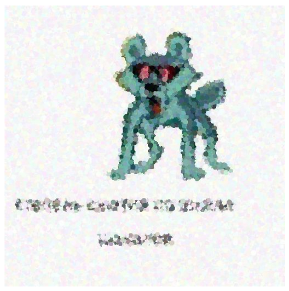
- NUMBER_OF_CIRCLES
- QUALITY (in percents)
- MAX_CIRCLE_RADIUS (in pixels)

So, we will change these parameters , check result image and running time.

Table for compare initial images and results images.

Initial image	Result 1	Result 2	Result 3
			
7.png	* running time = 18sec * radius of circles in range(0,10) * number of circles 65536 * quality 80% * number of childs 554087	* running time = 1min 25sec * radius of circles in range(0,10) * number of circles 65536 * quality 90% * number of childs 3283553	* running time = 10min 49sec * radius of circles in range(0,5) * number of circles 65536 * quality 95% * number of childs 24805430
			
chocolate.png (Made milka bubble from simple milka)	* running time = 54sec * radius of circles in range(0,30) * number of circles 65536 * quality 80% * number of childs 1593423	* running time = 5min 29sec * radius of circles in range(0,15) * number of circles 65536 * quality 90% * number of childs 11734777	* running time = 30min 15sec * radius of circles in range(0,5) * number of circles 65536 * quality 95% * number of childs 85247651

			
bubble.png (Made bubbles from bubbles)	* running time = 24sec * radius of circles in range(0,10) * number of circles 65536 * quality 80% * number of childs 743214	* running time = 2min 6sec * radius of circles in range(0,10) * number of circles 65536 * quality 90% * number of childs 4755721	* running time = 48min 23sec * radius of circles in range(0,7) * number of circles 100 000 * quality 95% * number of childs 129340185
			
8.png	* running time = 34sec * radius of circles in range(0,5) * number of circles 131072 * quality 80% * number of childs 988460	* running time = 2min 50sec * radius of circles in range(0,5) * number of circles 131072 * quality 90% * number of childs 6754801	* running time = 37min 22sec * radius of circles in range(0,5) * number of circles 262144 * quality 95% * number of childs 102859688
			
3.png	* running time = 1min 25sec * radius of circles in range(0,15)	* running time = 14min 35sec * radius of circles in range(0,10)	* running time = 4h 28min 58sec * radius of circles in range(0,5)

	* number of circles 262144 * quality 80% * number of childs 2890800	* number of circles 262144 * quality 90% * number of childs 35352026	* number of circles 262144 * quality 97% * number of childs 793612202
			
shakal.png (let's simulate jackal scale compression ratio from memes)	* running time = 3min 56sec * radius of circles in range(0,5) * number of circles 262144 * quality 80% * number of childs 10103934	* running time = 27min 24sec * radius of circles in range(0,5) * number of circles 262144 * quality 90% * number of childs 77641758	* running time = 5h 13min 19sec * radius of circles in range(0,5) * number of circles 262144 * quality 95% * number of childs 577767212

Analyze results

We change 3 parameters(quality, radius and number of circles). Let's analyze results.

When we compare photos we can check that:

- Increase in *quality* leads to an increase *running time* , *number of childs* per circle and *color similarity* of initial image and result image.
- Decrease in *radius of circles* leads to an increase shape *similarity of images.*
- Increase in *number of circles* leads to an *variety of colors* on result image.

So, we made the *art* then we need to *decrease similarity* of initial and result image. When quality is too big , result image is looks like initial image with *Blur* photo filter. Good example of art is *3.png* and *chocolate.png* with quality 80% .