Part 1. Algorithm ♥

1) Chromosome representation:

In my algorithm a chromosome is a group of heart icons ♥ (optimal is 16*16 pixels). Each heart has a location in the picture and can change color (mutate). Since we have the original picture 512x512, then I have 512 / 16 = 32 hearts in one row (total rowes, respectively, also 32). So, chromosome is 32*32=1024 hearts.

2) Population size and selection technique:

Population size is 50 chromosomes.
To produce new generation I choose:

- 16 times chromosome with minimum fitness (top 1)
- 17 times top 2 chromosome
- 17 crossovered chromosomes between top two chromosomes with minimum fitness

3) Fitness function:

Mean square error (pixel-by-pixel comparison) between input and generated images:

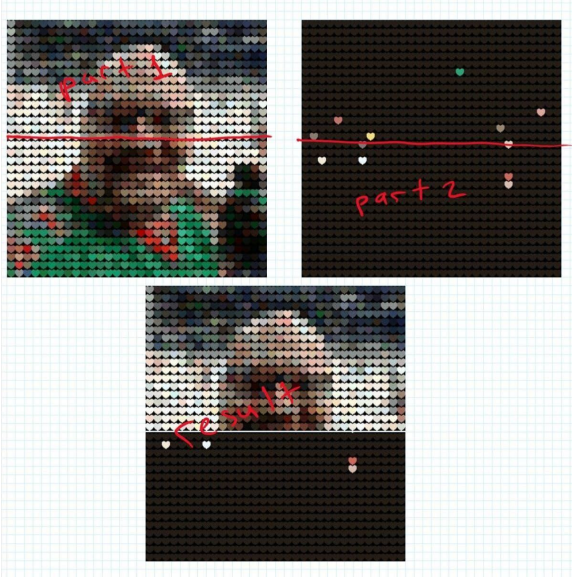
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

It turns out that the less fitness, the better

4) Mutation and Crossover:

As a mutation I change the color of some random heart icon in a chromosome.
As a crossover I choose two bests chromosomes and cross them in such a way:

Chromosome 1 Chromosome 2



Result

5) More information:

- In fact, the algorithm works with any single-color icon, just code it in svg format in this function:

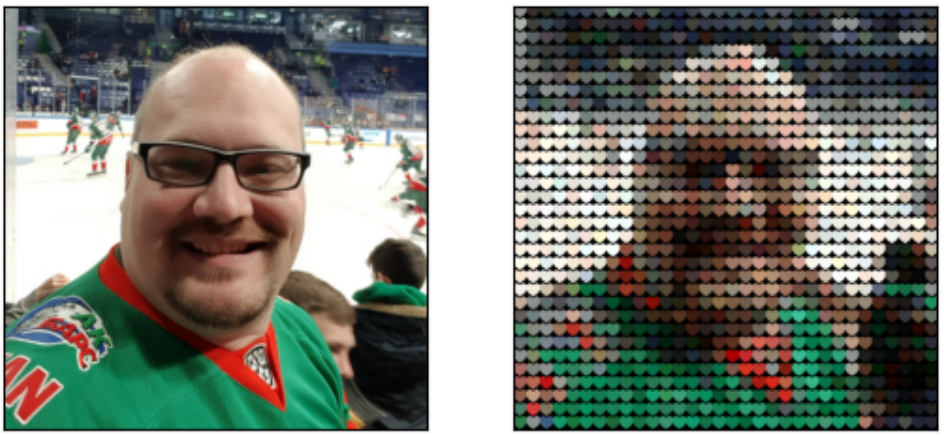
```
64  def recolor_heart():
65      random_color = random.choice(colors_list)
66      hex_color = rgb_to_hex(random_color[0], random_color[1], random_color[2])
67      dwg = svgwrite.Drawing('heart.svg', viewBox=f'0 0 512 512', size=(f'{ICON_SIZE}', f'{ICON_SIZE}'), fill=hex_color)
68      path = Path("M474.655,74.503C449.169,45.72,413.943,29.87,375.467,29.87c-30.225,0-58.5,12.299-81.767,35.566  c-15.522
69      dwg.add(dwg.path(path.d()))
70      path = Path("M160.959,401.243c-36.618-30.304-65.836-62.565-86.845-95.889  c-26.529-42.083-39.981-85.961-39.981-130.
71      dwg.add(dwg.path(path.d()))
72      dwg.saveas('./hearts/heart.svg')
73
74      cairosvg.svg2png(url="./hearts/heart.svg", write_to="./hearts/heart.png")
```

- Due to the fact that this is an svg, the icon size can change: 8x8, 16x16, 32x32, etc. Just change this variable:

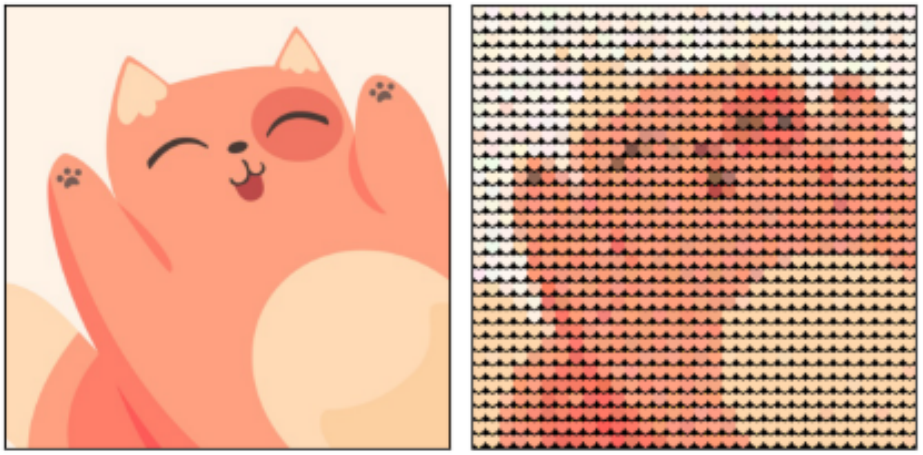
```
13      ICON_SIZE = 16
```

Part 2. Examples

Unfortunately, when I ran my algorithm, I forgot to add a function to save the resulting image, I only plot a comparison of the original and the resulting image (see below). Only at the end I added this feature, so not all images are in the original 512x512 format.

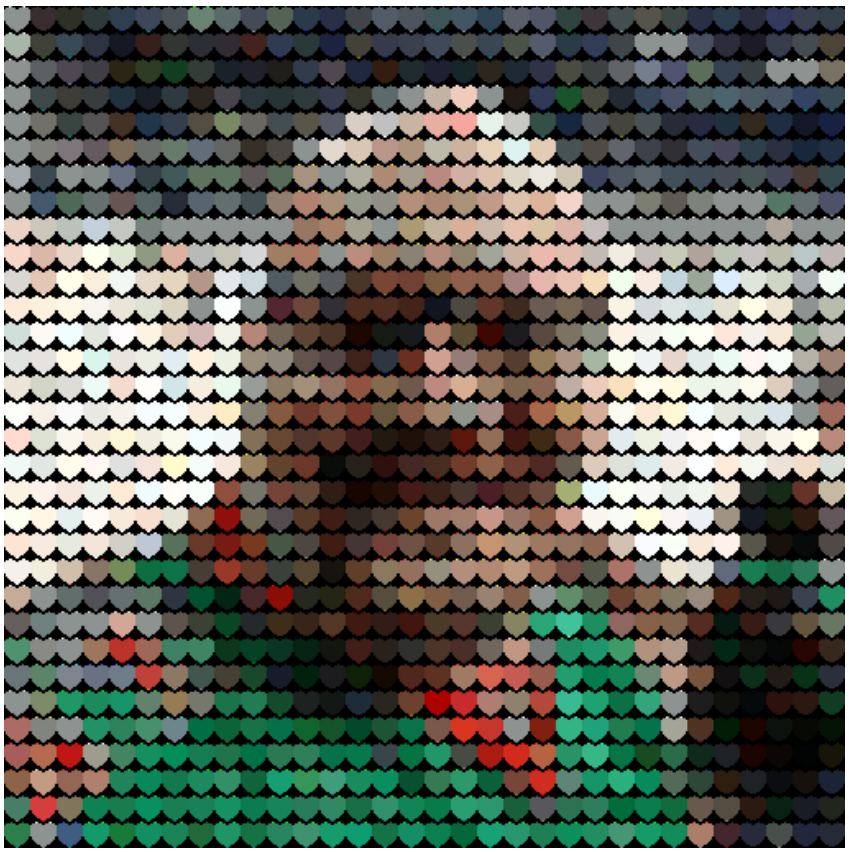


Iteration 1185



Iteration 1100

Iteration 750



Belov Timur, BS19-01

Part 3. What is art?

Art is the expression or application of human creative skill and imagination, typically in a visual form, art is what a person creates. A person can draw a picture, can make a sculpture, and so on. And even the image generated by the program can be considered art and is art in fact, because the program was written by the person himself. Moreover, for such people there is a term algorist[1] and the generated art is algorithmic art.

The most famous form of such art is the fractals. I think looking at fractals, everyone thinks about how beautiful and fascinating it is, because it is art! However, I repeat, fractals are just one type. The other type can be seen, for example, in my works - works made of hearts.

All in all, I think that my work is art, as I explained it above. ♥

[1] https://en.wikipedia.org/wiki/Algorist, https://en.wikipedia.org/wiki/Algorithmic_art