# Introduction to Artificial Intelligence

# Assignment 2

# Report

# On topic “Evolutionary algorithms”

**Report by:Ruslan Nurutdinov**

**Group number:2**

**Student’s ID:BS19543**

**Used stack of technologies: Python with PIL and Numpy libraries.**

## 1.Chromosome representation

**1)** Firstly,I implemented simple data structure named “square” that stores coordinates of left upper corner and right lower corner of square, and 3 values from 0 to 255 that specify color in RGB.

```
class square:
    def __init__(self,x1,y1,x2,y2,red,green,blue):
        self.x1=x1
        self.y1=y1
        self.x2=x2
        self.y2=y2
        self.red=red
        self.green=green
        self.blue=blue
```

**2)**Chromosome is an array of squares and genes are the squares.

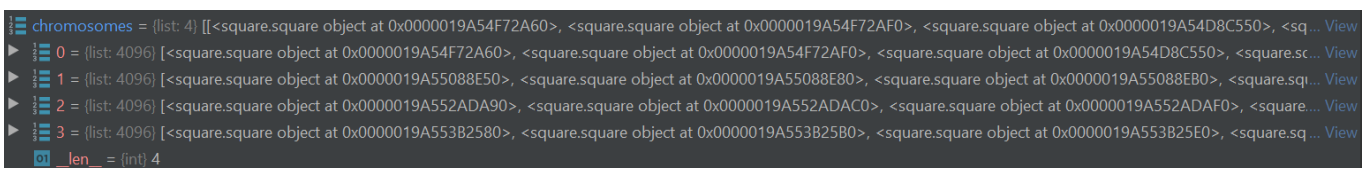
Every chromosome is an array of such squares, that initially generated corresponding to image size and length of the side of the square. Value of length of the side of the square may vary and it can changed directly with editing it’s value in code by user. Increasing this value leads to increasing of accuracy of obtained picture to initial one, especially if initial image have a lot of small details.

Note:for correct processing of the image it is preferable to use values that is 2 in some non negative power.

```
square_size=8
```

In this implementation the size of population is static and equals to 4.

Overall structure of array that stores data of all 4 chromosomes is following:



## 2.Description of algorithm

**1)**As it was mentioned, population size equals to 4 and it can't be changed.

Firtsly,we do a mutation by selecting random pixel in initial image and getting it's values of color in RGB. Then we select random square in image that should mutate and change it's values on obtained ones.

```
def mutation():
    global chromosomes
    for chromosome in chromosomes:
        colors = initial_image.getpixel((randint(0,511),randint(0,511)))
        index_of_mutation=randint(0,(int(512/square_size)*int(512/square_size))-1)
        chromosome[index_of_mutation].red=colors[0]
        chromosome[index_of_mutation].green=colors[1]
        chromosome[index_of_mutation].blue=colors[2]
```

**2)**After a mutation a fitness function compares "mutated images" and initial image and fills array called "fit" with corresponding fit values.Then it return indexes of 2 chromosomes that pretend to be parents for next generation

```
def fitness():
    for i in range(len(chromosomes)):
        fit[i]=sum(ImageStat.Stat(ImageChops.difference(Image.fromarray(chromosomesImg[i]), initial_image)).rms)

    sorted_array=sorted(fit)
    if fit.index(sorted_array[0])!=fit.index(sorted_array[1]):
        return (fit.index(sorted_array[0]),fit.index(sorted_array[1]))
    else:
        arr = numpy.array(fit)
        t=numpy.flatnonzero(arr == arr.min())
        return (t[0],t[1])
```

**3)**After receiving 2 indexes crossover function starts to operate. It copy first and second parent to first 2 positions of "chromosomes" array. Then its splits each parent on 2 halfes:bottom and top parts. First child will be compiled using top part of first parent and bottom part of second, second child will have top part of second parent and bottom part of first parent. First child will be placed at third index of array,second will occupy fourth cell of an array. Then algorithm proceeds to first step until it reaches specified number of generations.

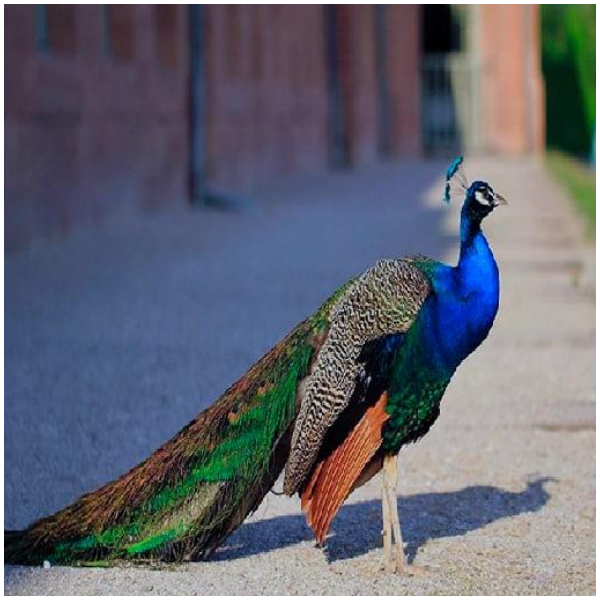
```
def crossover(first_parent_index,second_parent_index):
    first_parent=chromosomes[first_parent_index]
    second_parent=chromosomes[second_parent_index]
    for i in range(len(chromosomes[0])):
        chromosomes[0][i].red=first_parent[i].red
        chromosomes[0][i].blue=first_parent[i].blue
        chromosomes[0][i].green=first_parent[i].green
    for i in range(len(chromosomes[1])):
        chromosomes[1][i].red=second_parent[i].red
        chromosomes[1][i].blue=second_parent[i].blue
        chromosomes[1][i].green=second_parent[i].green

    for i in range(0,int(len(chromosomes[3])/2)):
        chromosomes[2][i].red=first_parent[i].red
        chromosomes[2][i].blue=first_parent[i].blue
        chromosomes[2][i].green=first_parent[i].green
    for i in range(int(len(chromosomes[2])/2),len(chromosomes[3])):
        chromosomes[2][i].red=second_parent[i].red
        chromosomes[2][i].blue=second_parent[i].blue
        chromosomes[2][i].green=second_parent[i].green

    for i in range(0,int(len(chromosomes[3])/2)):
        chromosomes[3][i].red=second_parent[i].red
        chromosomes[3][i].blue=second_parent[i].blue
        chromosomes[3][i].green=second_parent[i].green
    for i in range(int(len(chromosomes[3])/2),len(chromosomes[3])):
        chromosomes[3][i].red=first_parent[i].red
        chromosomes[3][i].blue=first_parent[i].blue
        chromosomes[3][i].green=first_parent[i].green
```

## 3.Demonstrations of obtained images

1)Square size=16 pixels. Final fit=80




2) Square size=16 pixels. Final fit=78


3) Square size=16 pixels. Final fit=82

4) Square size=8 pixels. Final fit=72.Decreasing square size shows increasing accuracy

## 4. My perception of art and how do I define it.

Art is beauty that touches the inner feelings of a person and makes us experience truly strong feelings. Of course, everyone defines beauty in their own way. Architects admire graceful buildings, composers enjoy the masterpieces of the classics ... But scientists find beauty in the patterns of this world: formulas, laws. So, the most beautiful formula of mathematics is Euler's formula. For a person who is not familiar with mathematics, this formula will seem just an incoherent set of symbols, but for mathematicians it is a standard of beauty and elegance. A person who looks at the result of my work may not feel anything, but the true beauty lies in the idea that a machine can create something new from the old with just an algorithm. Just 100 years ago, the thoughts of robots and powerful computing machines were fantastic, but today there is nothing surprising about it. Even today machines can analyze our search queries and use them to create an image of our personality and try to guess what we could like. I am sure that in the future, machines will be able to create something completely new that will give joy and unite huge mass of people.