Image segmentation with Genetic Algorithm

Genetic Algorithm:

First we will explain Genetic Algorithm and then we will discuss how we have implemented it in our code.

- We have to make an initial generation that consists of chromosomes (possible answers). Chromosomes will be made randomly.
- Then we will use each chromosome as an answer and rate it.
- After we have rated all the chromosomes we will save the best chromosome (with best rating) and save it as the best chromosome we have found yet.
- Now we have to make children from the current generation. (crossover)
- After making each child, it has a chance to mutate. (mutation).
- Now we will replace some of the older chromosomes with new chromosomes made and now we have a new generation.
- With the new generation we will do the same thing until we have an answer (or until a certain amount of time).

Implementation:

We explained GA in the last part. Now we will explain the code.

- As said before, first we have to make an initial generation:

```
generation = []
for x in range(0, 100):
    chromosome = []
    for y in range(0, 3):
        gene = []
        for z in range(0, 3):
            n = random.randint(0, 255)
            gene.append(n)
        chromosome.append(gene)
    generation.append(chromosome)
```

In array generation we will store 100 chromosomes. Each chromosome has three genes and each gene has three values (RGB).

- Now from this generation, we will pick each chromosome and color the picture with it. For rating the chromosome we use it's euclidean distance from each pixel of the picture (img) and store it in array rate[100]. (rate[7] is rating for chromosome number 7 which is stored in generation[7]).

```
rate = []   #in rate we will have rating of 100 chromosomes
for i in range(0, 100):
    rate.append(394)
for c in range(0,100) : #we will color img with chromosome c and rate it
    rateforc = 0
    for x in range(0, 257):
        for y in range(0, 257):
            c0 = ((img[x][y][0]-generation[c][0][0])**2 + (img[x][y][1]-generation[c][0][1])**2 + (img[x][y][2]-generation[c][0][2])**2)**(0.5)
            c1 = ((img[x][y][0]-generation[c][1][0])**2 + (img[x][y][1]-generation[c][1][1])**2 + (img[x][y][2]-generation[c][1][2])**2)**(0.5)
            c2 = ((img[x][y][0]-generation[c][2][0])**2 + (img[x][y][1]-generation[c][2][1])**2 + (img[x][y][2]-generation[c][2][2])**2)**(0.5)
            if(c0 == min(c0,c1,c2)):
                rateforc = rateforc + c0
            elif(c1 == min(c0,c1,c2)):
                rateforc = rateforc + c1
            elif(c2 == min(c0,c1,c2)):
                rateforc = rateforc + c2
    rate[c] = rateforc/(257*257)
```

- Now we have to find the best chromosome in this generation. To do that we will sort them.

```
for i in range(0, 99):
    for j in range(i, 100):
        if (rate[i] < rate[j]):
            temp = rate[j]
            rate[j] = rate[i]
            rate[i] = temp
            temp2 = generation[j]
            generation[j] = generation[i]
            generation[i] = temp2
```

Here generation[99] contains the best chromosome and rate[99] is the rating for it. We will save them in two variables: “bestchromosome” and “bestrate”:

```
bestchromosome = generation[99]
bestrate = rate[99]
```

- Crossover and Mutation: in this part we want to make 50 new chromosomes. To do this, in a loop that repeats 50 times we will take to random chromosomes m and n and the average of their genes will be the genes for new chromosomes. After making each gene, we have to do mutation on them. There is a 10% chance for each new chromosome to have mutation.

```
children = []
for i in range(0, 50):
    n = random.randint(0, 99)
    m = random.randint(0, 99)
    gene0=[(generation[n][0][0]+generation[m][0][0])/2, (generation[n][0][1]+generation[m][0][1])/2, (generation[n][0][2]+generation[m][0][2])/2]
    gene1=[(generation[n][1][0]+generation[m][1][0])/2, (generation[n][1][1]+generation[m][1][1])/2, (generation[n][1][2]+generation[m][1][2])/2]
    gene2=[(generation[n][2][0]+generation[m][2][0])/2, (generation[n][2][1]+generation[m][2][1])/2, (generation[n][2][2]+generation[m][2][2])/2]
    # we have to do mutation on genes
    chance = random.randint(0, 9)
    if(chance == 0):
        gene0[0] = gene0[0]**(0.5)
        gene0[1] = gene0[1]**(0.5)
        gene0[2] = gene0[2]**(0.5)
        gene1[0] = gene1[0] ** (0.5)
        gene1[1] = gene1[1] ** (0.5)
        gene1[2] = gene1[2] ** (0.5)
        gene2[0] = gene2[0]**(0.5)
        gene2[1] = gene2[1]**(0.5)
        gene2[2] = gene2[2]**(0.5)
    child = []
    child.append(gene0)
    child.append(gene1)
    child.append(gene2)
    children.append(child)
```

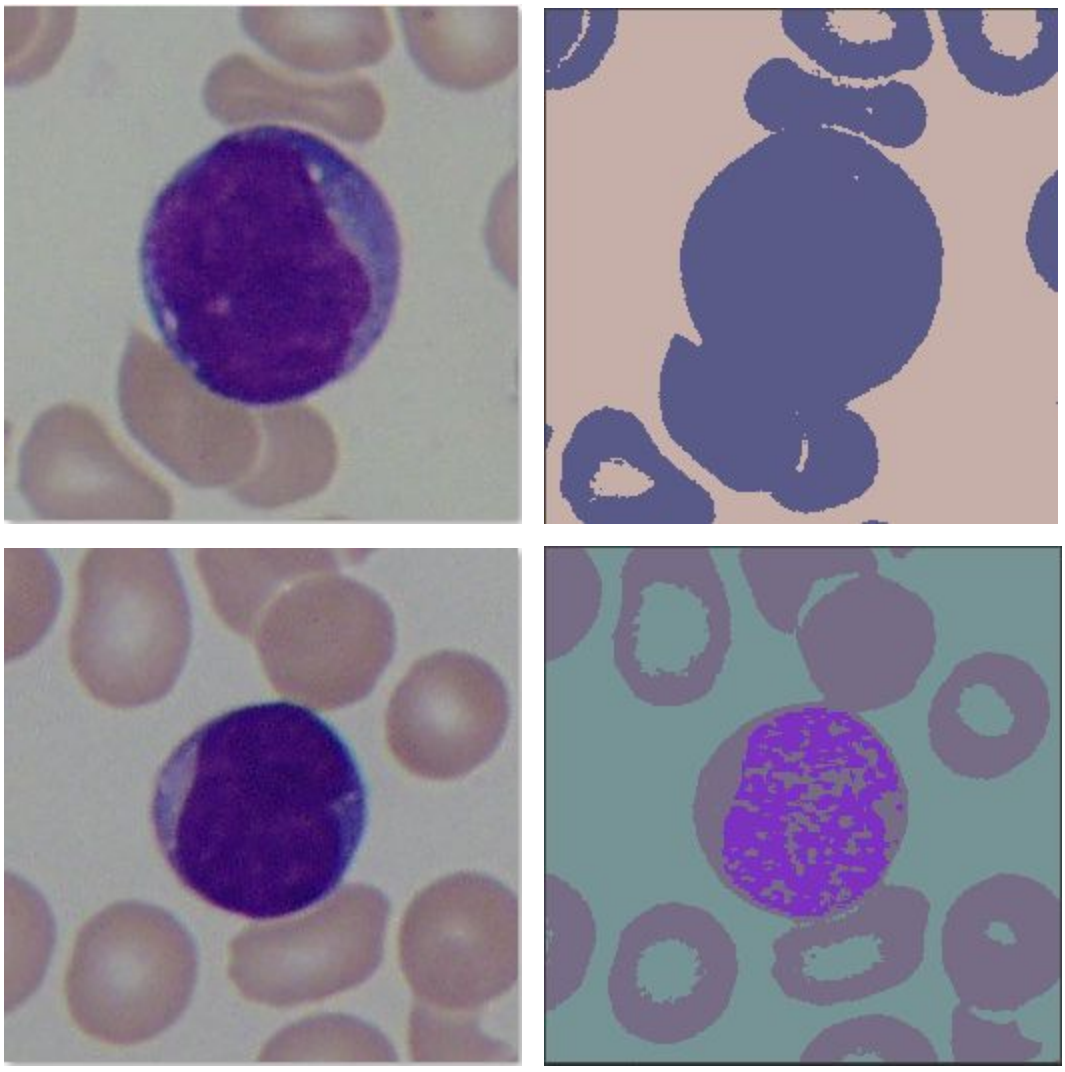
Here we have made 50 new chromosomes and each time we save them in array “child” and then add the new chromosome to array “children”. At the end of the loop, array children has the new 50 chromosomes.

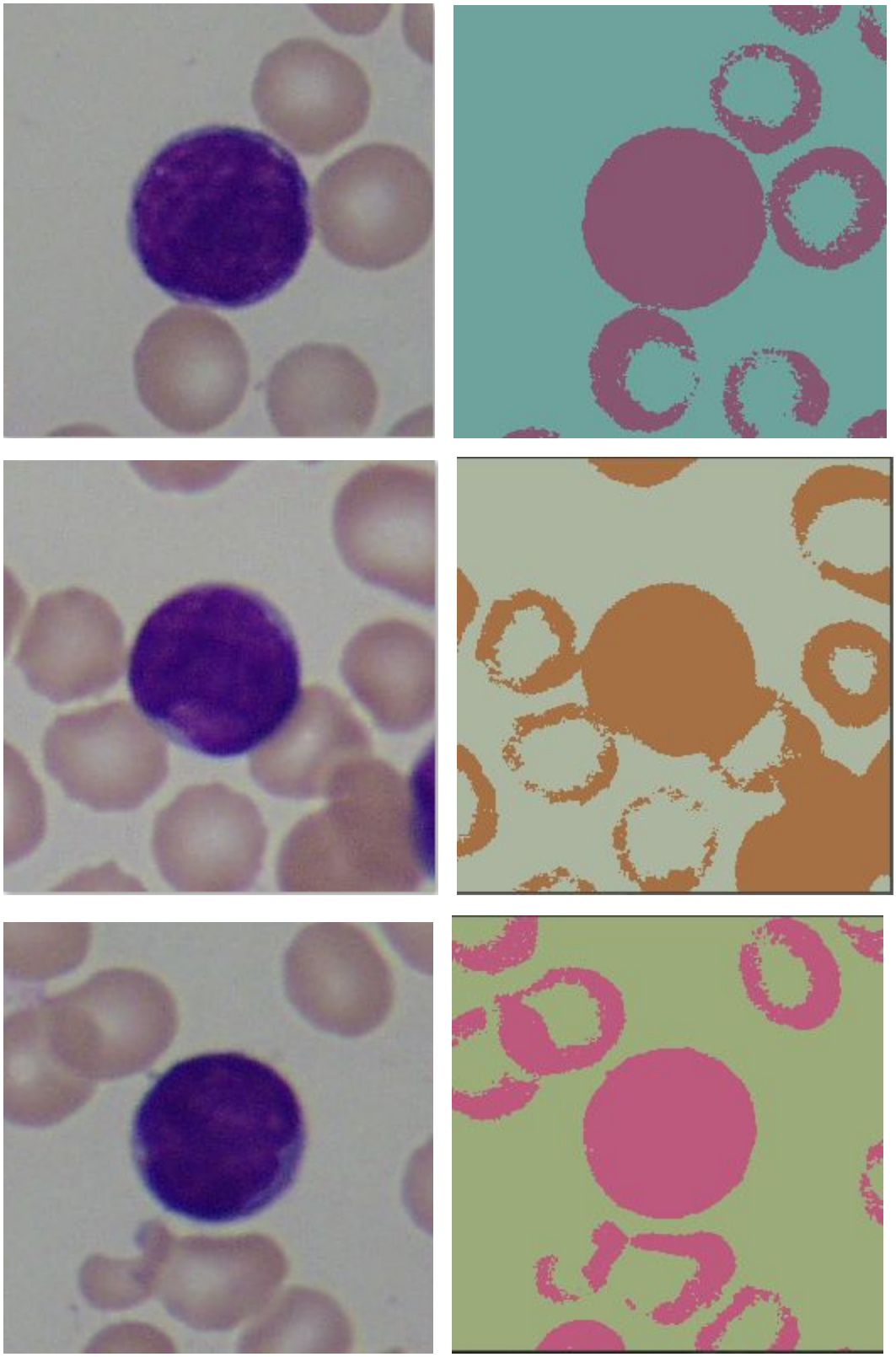
- Here we will make the new generation by replacing the new chromosomes with some of the old ones. We will replace them instead of chromosomes with the lowest ratings.

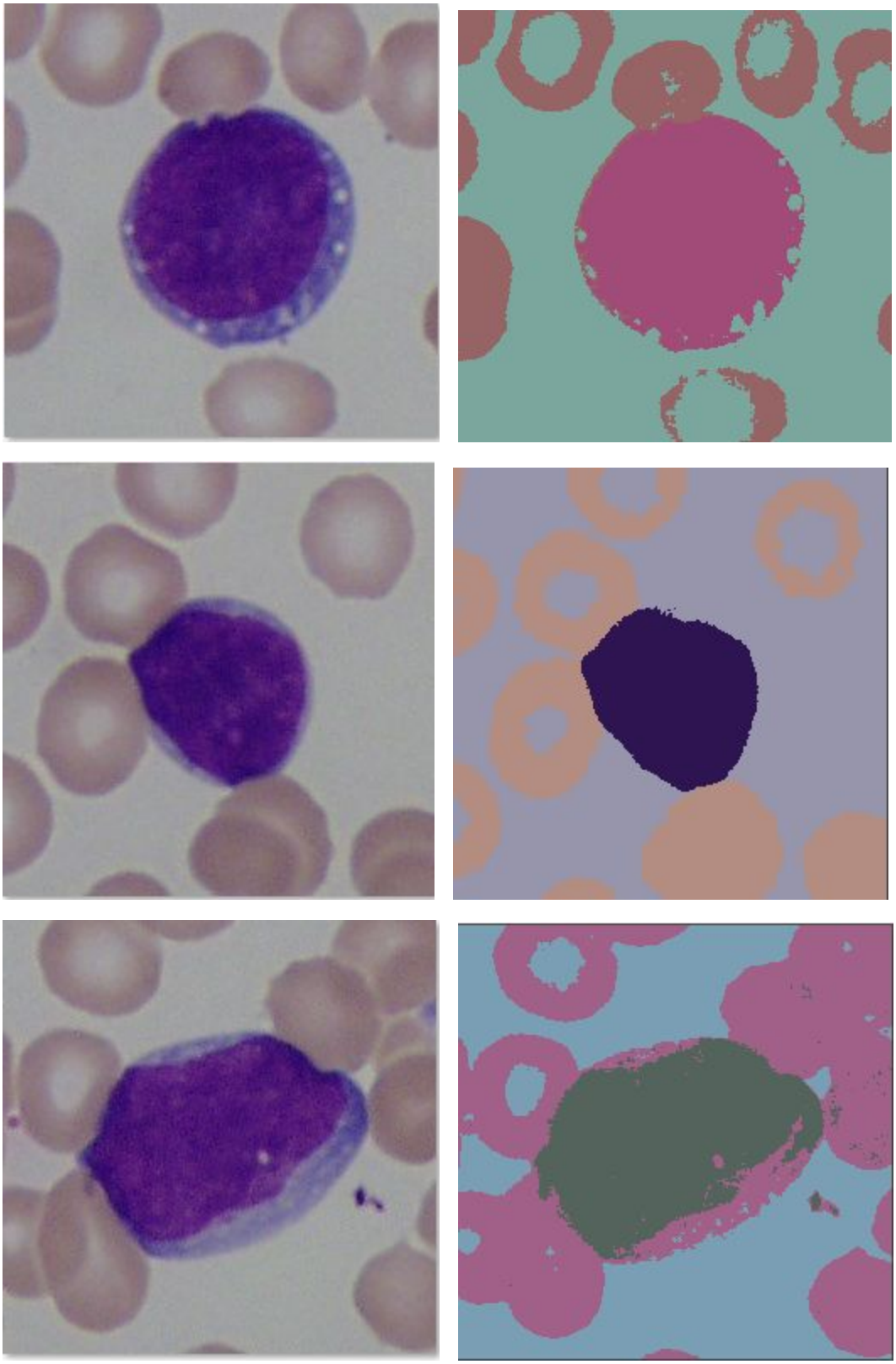
```
for i in range(0, 50):
    generation[i] = children[i]
```

- Now we have to continue doing this for 20 more generations and each time if the best chromosome of the new generation is better than the previous best chromosome, we will replace it and in the end we will have the best chromosome to segmentate the picture.

Here we have some examples:







One of the problems i had with this was that as you can see some of the pictures are coloured with only two colours so only two of the genes of the best chromosome have been used because the other one had more distance from the pixels than the other two.