Assignment 2

Mark Nicholson BS19-05

Description of my implementation:

I have decided to create generic algorithm to create a new Image from emojis (or any other images or even color

Chromosome representation:

Let's start with genes. Gen can be one of an image in the pull of images (it contains about 800 photos). I represent gen as a string(name or path to image) or a number(number of the image in the certain pull of images).

A chromosome contains genes as lists of genes. I divide the photo into chunks and each chunk has its gen (which contains a certain image to paste in the resulting photo). So my chromosome is a list of lists of strings (names of images).

Chromosome example

Aa 0	1	2	3 1
image_245	image_182	...	image_82
image_673	image_354	...	image_639
...	...	...	...
image_714	image_628	...	image_482

Population size and selection technique

I was thinking about how do we select population size. I figured that if we will make too big a population size generic algorithm will check many chromosomes that will be not accurate. On the other hand for the next generations, it will be more accurate if we will have more variations of genes. So by experimenting a bit I decided to have a population size 128.

I am using the Tournament Selection technique. So I am choosing one image from the pool of images (set of images) by my fitness function. If a value from the fitness function (error)is the lowest I am choosing this image.

Fitness function

As I mentioned above I am calculating error (deviation of given image) in the fitness function. So, I am calculating the mean color of all images in the image pool let's call it mce, and the mean color of a chunk in an image let's call it mci. And error will be deviation by each color:

mci_r is red of second image, mce_b is blue color of the first image etc...

$$error = |mce_r - mci_r| + |mce_b - mci_b| + |mce_g - mci_g|$$

Crossover & mutation

Getting child from 2 parents. First of all I choose random number from 0 to 1:

- if probability less than 0.45 I choose gen from the first parent
- if 0.45 ≤ probability < 0.9 than I choose gen from the second parent
- if probability ≥ 0.9 I choose random gen from

How did I collected this emoji images:

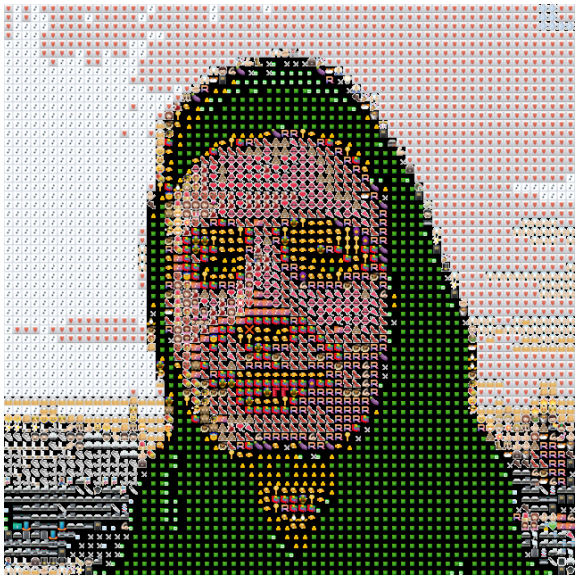
I was seeking on the internet the images. First time I found a pack of png images without background. It was very comfortable to use it in my image pool. But there were only about 250 images and they were without clear colors.

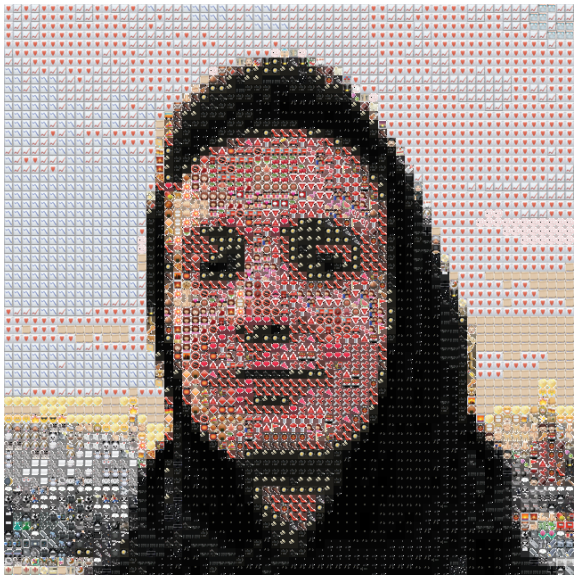
I decided that it is not the result I want to get, so I started to search once again. After some time I found 800+ pack of jpg images with white background. It was what I needed, but it had a white background which would make an image in white dots which I found unacceptable.

That I had one more task. Convert all these images to png without background. I found many sites that can do it. But they did it only for one picture. Write a bot that will make requests for all 800 images would be not that simple, so I searched for another solution.

Example of image of old image pool:

Example of image using new image pool:





Not generic implementation:

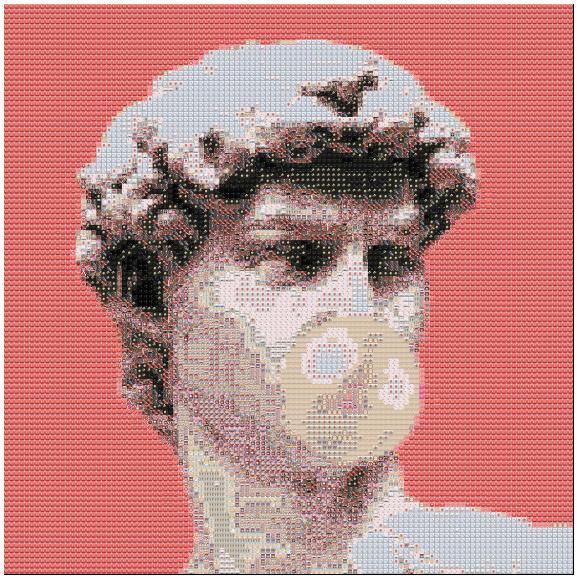
Since this problem can be solved not only by a genetic algorithm. I have implemented it in another way. The other file that I called "fast.py" is dividing the initial photo into chunks and search in all images the most suitable image and saves it. It working much faster than the generic algorithm (in about 1000×1000 about 10 seconds). Also, it can change the size of images so it can create an image from 10×10 small images, 20×20, and 30×30.

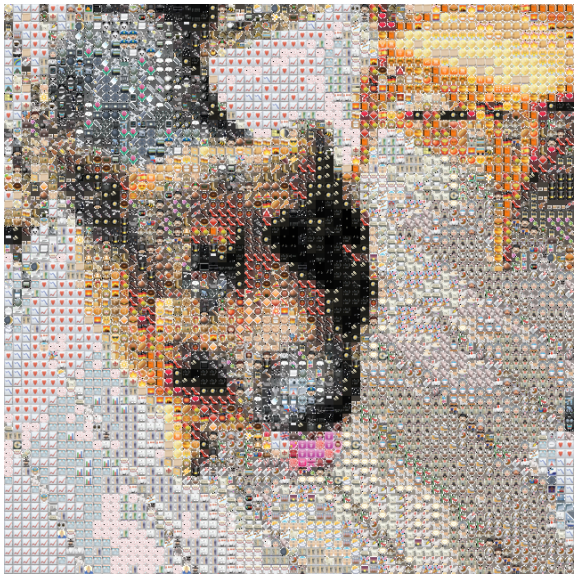
So, if you don't want to wait for 1-2 hours you can run this program and it will do the same but in 10 seconds.

Here a some examples of the algorithm

Original photo:

Converted:

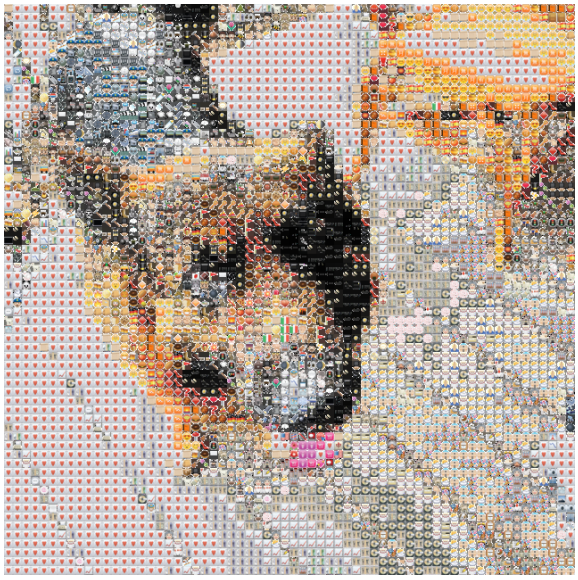


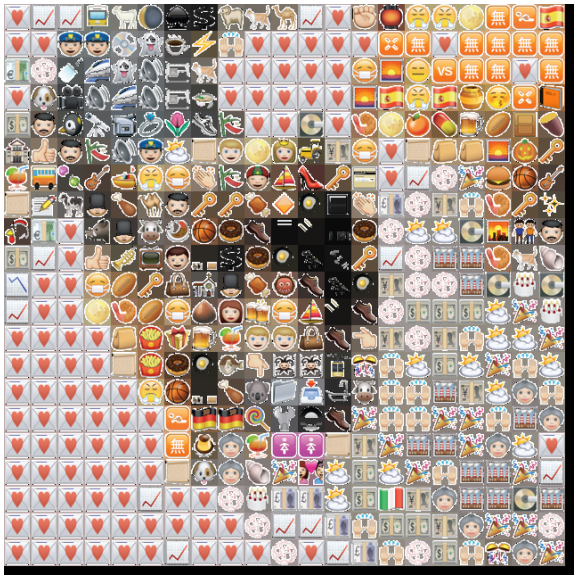


Also, non-generic implementation can generate 20×20 and 30×30.

Converted to 20×20:

Converted to 30×30:





Here is examples on large photo to see the changes in details:

Original:

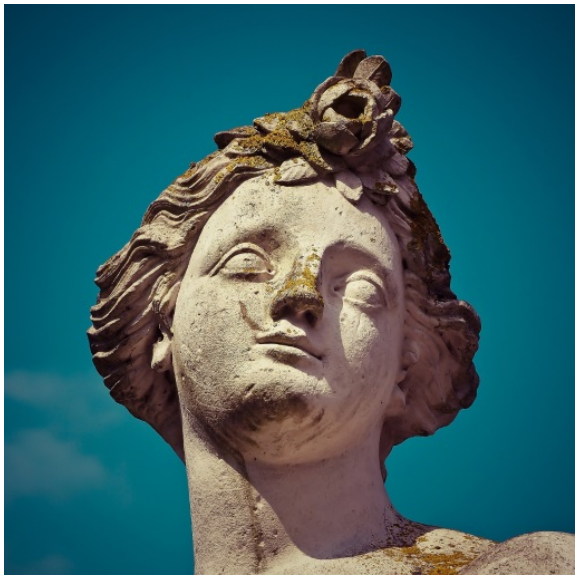
Converted to 10×10:

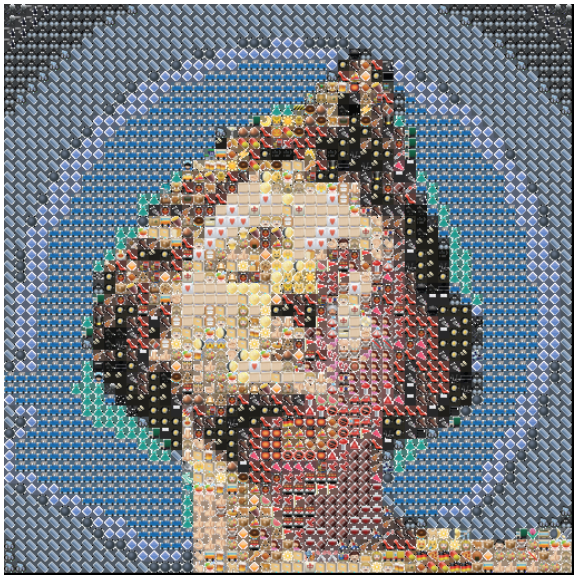
Converted to 20×20:

Converted to 30×30

Original:

Converted to 10×10:





How to run this programs:

For both scripts you may change image_name" on 5-th line to image name you want to convert. Default is *image.jpg*

To run fast.py (not generic implementation) Also you can choose to create or not images of emoji size 10×10, 20×20 or 30×30. And run script

To run main.py (generic solution) you need to specify name and run script.

What is Art?

Art for me is an object/action that can be done in many ways and the way an artist does is a special (most likely unique) way to do this. So, art can be a dance, code, building, photo, painting, etc...