Project 2: Image Reconstruction with Genetic Algorithms

Eduardo Barros Innarelli: RA 17016
Victor Ferreira Ferrari: RA 187890
Vinícius Couto Espindola: RA 188115

This project is a study of genetic algorithm-based solutions for **image reconstruction**. This problem consists of, given the original image, recreating an image from scratch. The original is only used for fitness purposes (how good is the image generated). This is not particularly useful in many applications, but can be used as basis for other problems, and is good for the purpose of this project: test different methods of reproduction, mutation, etc.

The modeling of the problem as a GA problem is almost direct: the image is the individual (chromosome) and each of its pixels is a gene. The chosen way of representation is an unidimensional array, as this facilitates operations, especially crossover.

Due to the huge number of combinations in a regular RGB image (2553 possiblities per pixel), greyscale images are used in this project. High resolution images were rescaled to a lower resolution using OpenCV to speed up the testing process.

Overall, the best results achieved for the small images were of good quality and high similarity. In some cases, it was possible to considerably reduce the fitness value to quite low values. Some of the final results and their original versions can be seen below. With the big images, it was possible to achieve results that resemble the original, with a lot of noise.

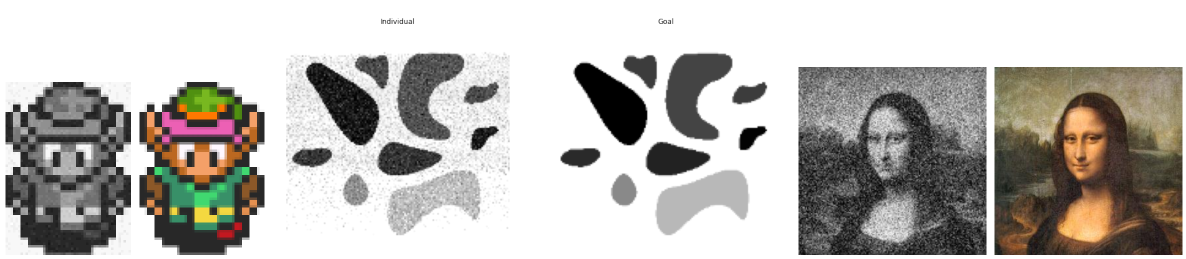
Two major problems that this project ran into were, on one hand, premature convergence, and on the other hand, slow convergence speed. Sometimes, both of these problems were present at once, when a lot of iterations were needed to keep optimizing the fitness value, and the convergence speed slowed down with still a high error value.

With a good computer or using the Google Colab environment, it takes a few hours to run 5 million iterations on the larger images, with resolution reduction factor of 0.6. Since this problem wasn't solved so far, the best course of action is to avoid it, using higher reduction factors or smaller images.

As expected, the amount of possibilities makes this a hard problem to solve, especially for higher resolution images. The initial error values can be astronomical, and the optimization can be slow, even if regular.

Still, the results achieved with small images are really similar to the original, even in full resolution, so this project was a success in that front. With really small images, the a very similar result could be achieved in just a few thousand generations. Most of those images, though, reached minimum point after which there was no improvement found, even after hundreds of thousands of generations.

Some of the best results for some test cases are available in figure 1. `Link` is 50x69 px, `Objects` is 238x238 px, and `Monalisa` is 256x256 px. The latter two were run with 0.6 resolution reduction factor. More results (final and partial) can be found in this link.



(a) Link (b) Objects (c) Monalisa

Figure 1: Results