CMKV

Introduction

Imagine a world where color does not exist anymore, as IT engineers decided to remove it from every piece of image that still exist, for their own financial benefits (like the proverb says, "quantity, not quality"). As aspirant IT engineers, we were recruited to create algorithms able to sweetly remove color from images, without removing the "perception" of it.

In this project for the CMKV course, we had to use Markov chain to convert rgb image to binary black-white images, representing the result in an artistic way.

Build

Run `make` then `./cmkv` specifying input and output image's paths.

Explanations

According to the course, we decided to implement the Metropolis-Hastings algorithm, as it seemed to be both efficient and easy to implement.

Using this notebook as an example, we first decided to shift the data array from range 0|255 to -1|1 but the result was not efficient. Then we moved to range -127|128 (0 - median|255 - median) to perform the algorithm, and we got different kind of result depending on the approach. The different approachs are commited in the branches `threshold`, `linear1` and `linear2`.

- `threshold` : In this approach, we apply Metropolis algorithm on pixels randomly selected for `image.size*6` iterations. We then apply a mask (corresponding to the first quartile of the values) on the result to color the darkest pixels uniformly.
- `linear1` : Here, the Metropolis algorithm is processed in a linear way, by walking through every line. In the output, it leads to vertical line patterns.
- `linear2` : This time, the algorithm is processed by walking through every columns, which leads to horizontal line patterns. Both approach are artistic.

In each approach, we compute a delta from 8-connected neighbors. From this score, we generate a random number between 0 and 1 to randomly apply or not the pixel coloration according to the probability law.

The values used for `T`, `beta` and `gamma` were chosen according to our observations of the results.

Limits

We could have used simulated annealing for better performances but we decided not to do so. It can be a way to improve the project.