Yushuo Ruan

CSCI 5722 Computer Vision

HW1 – Task 1.7 Analysis

Due 1/26/19

There are two tolerance, and they serve for two different fitness calculation methods. I first use 10 for both tolerance, but the result was not good. The first two fitness could get up to 0.5 while the last two was still below 0.1. This situation would cause the last two fitness (rate of change on pixel values) to be ignored. The solution to this is to increase the second tolerance. If the difference between two images is 10, then the difference between the rate of change of two images should be 20. So changing the second tolerance to 20 will improve the program.

As the image gets bigger, there is a huge amount of pixels. Therefore, if these pixels’ ticket only vary from 0 to 10, there would not be much difference between “good pixels” and “bad pixels”. So we can increase the factor to make the pixels receive 0 to 20 tickets. Cooperating with the exponential factor, it would improve the program.

When I use the first breeding method, it was very likely to get into some situation which the whole population are very similar and the max fitness is very close to the average fitness, and the fitness grows very slowly or stop growing. Second breeding method does a better job.

Larger image needs larger population size since it is harder to find a fix. The 10x10 image runs pretty fast on a 200 population size, where the 25x25 image needs a larger (I use 1200) to have its fitness constantly grow.

In the assignment, the Pythagoras way of getting the fitness somehow causes problem and the fitness grows very slowly. The traditional mean does better job in my case.