CSI 2110 – Project Part 3- Golang

Shacha Parker 3002325525 – Joey Issa 300219818

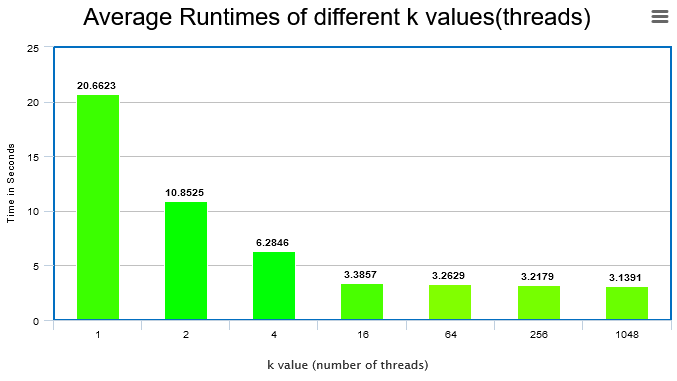
Our objective is to replicate the SimilaritySearch function in Go. It compares the histogram of an image and prints out the images that have the closest histograms to it.

Firstly, we will demonstrate the runtime of SimilaritySeach when given 7 different thread amounts. K is the number of allowed threads:

# Runtimes and Average Runtime in Seconds of SimilaritySearch on The Same Image Given Varying Thread Amounts

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **k** | **1st Try** | **2nd Try** | **3rd Try** | **4th Try** | **5th Try** | **Average** |
| **1** | 20.7134 | 20.5410 | 20.6555 | 20.6142 | 20.7874 | 20.6623 |
| **2** | 11.0102 | 10.7928 | 10.8192 | 10.8291 | 10.8111 | 10.8525 |
| **4** | 6.5728 | 6.3181 | 6.1619 | 6.1633 | 6.2070 | 6.2846 |
| **16** | 3.3368 | 3.4027 | 3.3672 | 3.3787 | 3.4433 | 3.3857 |
| **64** | 3.2343 | 3.2297 | 3.1176 | 3.1684 | 3.5647 | 3.2629 |
| **256** | 3.4927 | 3.1996 | 3.1229 | 3.2013 | 3.0728 | 3.2179 |
| **1048** | 2.9996 | 3.0548 | 3.1337 | 3.1314 | 3.3762 | 3.1391 |

**This data was run on q00.jpg. Numbers have been rounded to 4th decimal places.**



We can see that the runtime asymptotically becomes more similar to its last k value. Around k=16 do we see that the reduction in runtime plateaus.

This experiment was run on a Windows 10 system with an AMD Ryzen 5 1600, 6 Cores, 12 Threads, 65 W TDP, 3.2 GHz