

Testausdokumentti Pihla Toivanen

Testaus oli projektini alkuvaiheessa hieman hankalaa, sillä TDD -lähestymistapaa on hankalaa soveltaa pakkausalgoritmeihin. Testasin kuitenkin mahdollisimman monta osaa omina yksikköinä ja tein kokonaisuutta testaavia kuvatestiohjelmia, joissa ihmissilmilläni arvioin onko pakkaus onnistunut.

Junit -yksikkötestit:

convertToYCbCr – convertToRgb: Tehdään yhdelle datasetille nämä metodit, tuloksena täytyisi olla alkuperäiset arvot. Jos arvot menevät muunnoksessa yli 255 tai alle 0, tuloksena on 255 tai 0.

huffman-koodaus: Kattaa koko Huffman-koodaukseen liittyvän ohjelmakoodin. Testaa, että blocks-taulukko on sama ennen huffman-koodausta ja sen jälkeen tiedostosta luettaessa.

kvantisointi: samat arvot kvantisoidessa ja dekvantisoidessa

minimikeko: laitetaan kekon alkioita, ja otetaan yksi pois, pitäisi olla oikea alkio.

Ensimmäinen kuvatesti

Diskreetin kosinitransformaation tehtyäni kirjoitin kuvatestin, joka ottaa tekee kuvalle muunnoksen, käänteismuunnoksen ja kirjoittaa arvot .rgb -tiedostoon. Sain sen avulla debugattua muunnoksen toimivaksi ennen Huffman-koodaukseen siirtymistä, muistona kyseisestä testistä valmiissa ohjelmassani on writeToRGBFile -metodi.

Pakkaussuhde:

Täysin mustilla kuvilla pakattu kuva on aina 0.02%, värillisillä kuvilla riippuu kuvasta. Testeissäni pienet kuvat olivat 256*256px ja isot kuvat 1024*1024px. Koodasin testipohjan, jolle syötetään input-tiedosto, output-tiedosto ja pakattavan tiedoston koko, testi palauttaa pakkaukseen menevän ajan.

Kuva1: Värillisen kuvan testikuva:



Kuva 2: Pakkausajat ja -prosentit:

| | |
|--|------------------------------------|
| Musta pieni : 0.5s, 2% alkup.koosta | Musta iso: 4s, 2% alkup. koosta |
| Värikäs pieni: 0.2s, 36% alkup. koosta | Värikäs iso: 4s, 22% alkup. koosta |

Kuva3: Pienen kuvan (256*256px) pakkausprosentit eri kuvilla, 100 on alkuperäisen kuvan koko.

