Documentație Proiect AWJ

Converting Color Image to Gray-Scale Image using Weighted method (luminosity method)

Breazu Laura-Andreea

332AA

Introducere

Acest proiect își propune convertirea unei imagini RGB de tip 24 bit BMP in Gray-Scale folosind Weighted method/Luminosity method.
Proiectul este reprezentat de o aplicație Java care realizează o procesare a unei imagini. Aplicația nu are interfață grafică. Aceasta a fost realizată folosind Java 8, Eclipse Mars 2 și Windows x64.

Descrierea Aplicației

Aplicația are rolul de a converti o imagine primită ca input din RGB in Gray-Scale
și de a o salva imaginea convertita intr-un fisier nou.
Fișierul sursă este o imagine în format 24 bit BMP RGB. Pentru procesare am folosit doar algoritmi și secvențe de cod low-level. Aplicația își propune să includa conceptele POO. Am restricționat accesul claselor la date declarându-le pe cele din urmă "private" și am generat setteri și getteri publici acolo unde a fost cazul (principiul de încapsulare). De asemenea m-am folosit și de alte principii precum moștenire, polimorfism sau abstractizare.
Codul respecta „Coding Standards". În cadrul moștenirii am folosit tag-ul @Override.
Aplicația conține operații de lucru cu fișiere, fișierul input care este o imagine pe care vreau sa o convertesc și fișierul output care este imaginea convertita.
Aplicația primește numele fisierului care contine imaginea in initiala, în linie de comandă. Aplicația este împarțită în mai multe clase, scopul fiind multimodularizarea și încercarea de a face fiecare clasă să realizeze un singur obiectiv specific, clar.
Am folosit constructori, blocuri de initializare. Aplicatia contine Interfata si alta clasa care o implementeaza. Aplicatia include Clase Abstracte cu metode abstracte si clase concrete care extind clasele abstracte.
Am tratat exceptiile si am folosit 2 pachete: packTest care contine aplicatia de test si packWork care contine restul claselor.
Aplicatia contine Producer-Consumer. Un nou thread este alocat citirii din fisier a imaginii sursa – Producer Thread. Intra in Not Runnable dupa citirea a fiecarui sfert (1/4) de informatie. Un nou thread (Consumer Thread) este alocat consumului informatiei furnizate de Producer Thread. Se utilizeaza "multithread communication" (notify). Se insereaza output la consola si sleep (1000) pentru a evidentia etapele comunicarii. Se folosesc elementele de sincronizare pentru protectia la o eventuala interferenta cu alte posibile threaduri. Dupa terminarea consumului intregii informatii de imagine sursa, se incepe procesarea imaginii.
Aplicatia contine comunicatie prin Pipes pentru a transmite imaginea procesata de la Consumer catre un obiect de tipul WriterResult.

Partea teoretică

Pentru realizarea rotirii m-am folosit de Weighted method/Luminosity method. Clasa BufferedImageGrayscaleConverter este cea care primeste imaginea de tip RGB data ca input si o converteste intr-o imagine Gray-Scale de tip BMP. Algoritmul ia fiecare pixel al imaginii in parte si memoreaza valorile R, G si B in 3 variabile diferite. Dupa care se calculeaza valoarea pixelului folosind formula 0.3*R+0.59*G+0.11*B si se actualizeaza imaginea cu valolarea noului pixel.

Descrierea implementării

Aplicația este implementată în Java 8, folosind IDE-ul Eclipse Mars 2 pe un Windows 11 x64. Exportarea s-a facut ca jar pentru Eclipse Mars 2.

Descrierea arhitecturală

Aplicația contine urmatoarele clase:

MyMain – clasa principala, se ocupa cu citirea numelui fisierului input care contine imaginea initiala si apoi citirea imaginii, instantiaza clasele Producer si Consumer si porneste executia thread-urilor Producer si Consumer

ProducerConsumerInterface – o clasa interfata pentru clasele Producer si Consumer.

ProducerConsumer – o clasa abstracta implementata de interfata ProducerConsumerInterface si Runnable si contine metoda start()..

ProducerThread – o clasa de tip Thread care citește imaginea data si o memoreaza intr-un vector, iar apoi începe trimiterea informației către Consumer. Intra in Not Runnable dupa citirea a fiecarui sfert de informatie.

ConsumerThread - o clasa de tip Thread care preia informatia furnizata de ProducerThread, instantiaza clasa BufferedImageGrayscaleConverter care se ocupa cu aplicarea algoritmului de conversie si transmite imaginea procesata printr-un Pipe catre clasa WriterResult

ImageProcessor – clasa abstracta folosita in conversie.

GrayscaleConverter – clasa care mosteneste ImageProcessor.

BufferedImageGrayscaleConverter – clasa care mosteneste GrayscaleConverter si care se ocupa cu aplicarea algoritmului de conversie

WriterResult – clasa care primeste imaginea prin intermediul unui Pipe de la ConsumerThread si care se ocupa cu scrierea acesteia intr-un fisier nou

Descrierea funcțională a aplicației

Aplicația pornește imediat după apăsarea butonului run. In arhiva trimisă fișierul de intrare se află în același folder cu folderul src deci am pus ca argumente input.bmp și result.bmp. Fișierul result.bmp nu trebuie neapărat să existe, el va fi creat sau în caz că există va fi suprascris.
După rulare trebuie sa introducem in linia de comanda numele fisierului care contine imaginea initiala. Daca fisierul nu exista apare un mesaj de eroare si se asteapta o noua introducere. Se citeste imaginea si se porneste executia Thread-ului Producer. Acesta citeste imaginea, cate un sfert pe rand si o memoreaza intr-un vector. Consumer preia apoi imaginea, de asemenea impartita in 4 parti. Cele 2 thread-uri se executa in paralel. Dupa ce toata imaginea a fost preluata de Consumer aceasta este convertita cu Metoda Weighted in clasa BufferedImageGrayscaleConverter. Imaginea convertita este apoi tranmisa prin intermediul unui Pipe din Consumer in WriterResult si apoi este scrisa intr-un fisier nou.

Concluzii

Operația de conversie in Gray-Scale este foarte folosită în cadrul procesării imaginilor. Conversia unei imagini are loc printr-un proces în care se analizeaza fiecare pixel al imaginii si se modifica dupa formula data de Metoda Weighted. Aplicația construită are rolul de a citi o imagine primită ca input și de a o converti in Gray-Scale.

Bibliografie

Curs.acs.pub.ro, Cursul de AWJ
https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.html
https://www.tutorialspoint.com/java_nio/java_nio_pipe.html
https://www.geeksforgeeks.org/java-threads/