

TEMA AWJ

Image Contrast Modification

I. Introducere

Aplicatia Java realizeaza o procesare de imagini. Aceasta modifica contrastul unei imagini in functie de un factor de contrast ales de user.

II. Partea teoretica

Contrastul poate fi explicat simplu ca diferenta dintre maximul si minimul intensitatii pixelilor dintr-o imagine.

Pentru a schimba contrastul unei imagini trebuie sa modificam valoarea fiecarui pixel. Acesta are 3 parametri: red, green si blue. Pentru a modifica contrastul, formulele sunt urmatoarele:

1. $\text{red} = \text{contrastFactor} * (\text{red} - 128) + 128$
2. $\text{green} = \text{contrastFactor} * (\text{green} - 128) + 128$
3. $\text{blue} = \text{contrastFactor} * (\text{blue} - 128) + 128$

III. Descrierea implementarii

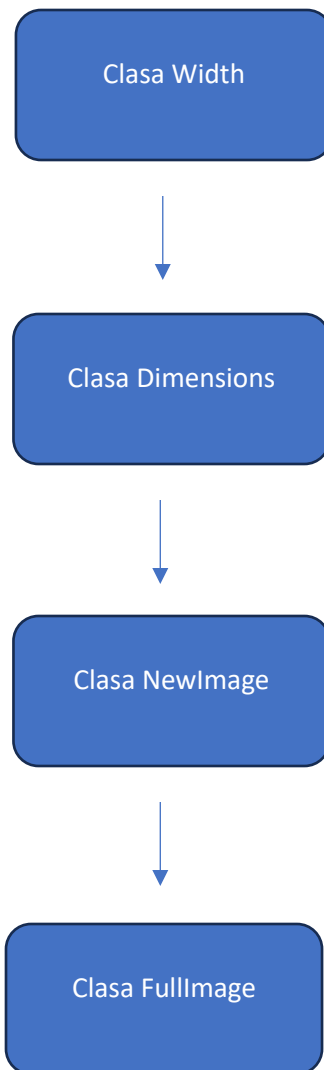
Am realizat implementarea utilizand 9 clase. Pentru a descrie implementarea, voi explica rolul fiecareia.

1. Clasa Test – aceasta clasa imbina functionalitatea aplicatiei. Ea primeste parametri din linia de comanda sau de la tastatura. In prima faza, verifica daca exista parametri din linia de comanda. In cazul in care nu exista, utilizatorul este nevoit sa ii introduca de la tastatura. Tot in metoda main a clasei test sunt initializate buffer-ul, producer-ul si consumer-ul, cat si pornite (producer-ul si consumer-ul).
2. Clasa Buffer – buffer-ul poate fi vazut ca locul in care este plasata informatia, canalul de legatura dintre Producer si Consumer. Acesta este gol pana cand Producer-ul trimite o imagine sau Consumer-ul o ia. Aceasta are 2 metode: put (care pune informatia in buffer) si get (care trimite informatia din buffer)
3. Clasa Producer – clasa Producer trimite imaginea citita in Buffer, cu scopul ca aceasta sa fie luata de Consumer si procesata. Aceasta are o singura metoda: start (care porneste Producer-ul)
4. Clasa Consumer – clasa Consumer preia imaginea din buffer si o proceseaza, aplicand modificarea contrastului. Aceasta are o singura metoda: start (care porneste Consumer-ul)
5. Clasa Width – clasa folosita pentru punerea in evidenta a mostenirii si abstractizarii. Aceasta retine latimea unei imagini si are ca metode getWidth (care returneaza latimea) si setWidth (care seteaza latimea), plus o metoda abstracta getHeight care va fi folosita in urmatoarea clasa.
6. Clasa Dimensions – clasa care mosteneste clasa Width si adauga si inaltimea imaginii. Similara cu clasa Width, aceasta contine toate metodele ei, plus setHeight (seteaza inaltimea) si getHeight (returneaza inaltimea)

7. Clasa NewImage – clasa care mosteneste clasa Dimensions si adauga ca attribute imaginea propriu-zisa si calea sa. Aceasta are in plus metodele de citire si scriere ale imaginii, cat si o metoda care returneaza imaginea retinuta in obiect.
8. Clasa FullImage – clasa care mosteneste clasa NewImage si adauga metoda de modificare a contrastului.
9. Clasa Interface – clasa de interfata care retine metodele de scriere si citire a imaginii

IV. Descrierea arhitecturala

Aplicatia este multimodulara. Aceasta contine 4 niveluri de mostenire:



V. Descrierea functionala

Dupa ce aplicatia primeste toti parametrii, Buffer-ul, Producer-ul si Consumer-ul sunt pornite. Producer-ul citeste imaginea din path-ul ales de utilizator si o trimite in Buffer. De acolo, consumer-ul o ia si modifica contrastul acesteia, salvand-o la o cale aleasa tot de utilizator. Mai mult, utilizatorul alege si factorul de contrast, cat si numele noii imagini.

VI. Evaluare performate

Aplicatia functioneaza si modifica contrastul in functie de factorul de contrast ales de utilizator.



Figure 1. Poza originala



Figure 2. Poza cu factor de contrast 33



Figure 3. Poza cu factor de contrast 12

```
Buffer created
Image created
Image read execution time: 117 ms
Producer is about to send the image
Producer sent the image
Consumer recieved the image
Image saved at: D:\UPB\AWJ\ModificareaContrastului\dsdsds.bmp
Contrast modify execution time: 161 ms
```

Figure 4. Performante factor de contrast 33

```
Buffer created
Image created
Image read execution time: 119 ms
Producer is about to send the image
Producer sent the image
Consumer recieved the image
Image saved at: D:\UPB\AWJ\ModificareaContrastului\dsdsds.bmp
Contrast modify execution time: 158 ms
```

Figure 5. Performante factor de contrast 12

Se pare ca factorul de contrast nu modifica performantele, dar cu siguranta dimensiunea imaginii da. Cu cat sunt mai multi pixeli, cu atat citirea, scrierea si modificarea imaginilor dureaza mai mult. Java, in concluzie, este un limbaj de programare care faciliteaza procesarea de imagini si are performante foarte bune.

VII. Bibliografie

1. https://curs.upb.ro/2023/pluginfile.php/190473/mod_resource/content/2/Java05.pdf
2. https://curs.upb.ro/2023/pluginfile.php/206313/mod_resource/content/1/Java07.pdf
3. https://curs.upb.ro/2023/pluginfile.php/217822/mod_resource/content/3/Java08.pdf
4. https://curs.upb.ro/2023/pluginfile.php/224091/mod_resource/content/2/Java09.pdf
5. https://curs.upb.ro/2023/pluginfile.php/226298/mod_resource/content/1/Java10.pdf
6. https://curs.upb.ro/2023/pluginfile.php/237529/mod_resource/content/1/Java11.pdf
7. <https://ie.nitk.ac.in/blog/2020/01/19/algorithms-for-adjusting-brightness-and-contrast-of-an-image/>