

Tema 1 - Metode Numerice

Procesare de imagine. Captcha

Data si ora ultimei modificari: 17 Martie 2013, 12:30

Responsabili tema:

Andrei Sfrent
Roxana Istrate

Descriere generala

Scopul acestei teme este familiarizarea studentilor cu limbajul de programare GNU Octave prin implementarea de functii pentru procesarea imaginilor.

Arhiva de pornire cuprinde un schelet de cod ce contine semnaturile functiilor pe care trebuie sa le implementati in cadrul temei. Fisierele pe care va trebui sa le completati sunt:

- rgb2gray.m
- histogram.m
- brightness.m
- salt_pepper.m
- captcha.m

Punctaj

Tema consta in rezolvarea a trei taskuri (Basic, Salt & Pepper si Captcha) si valoreaza in total 120p. Vor fi acordate, de asemenea, 10 puncte pentru coding style si README.

Task 1.1 (Basic - RGB to Gray)

10p

Fisier .m: rgb2gray.m

Cerinta: Trebuie sa implementati transformarea unei imagini RGB intr-o imagine gri. Formula folosita pentru transformarea unui singur pixel din imagine este $\text{Gray} = 0.2989 * \text{Red} + 0.5870 * \text{Green} + 0.1140 * \text{Blue}$, unde Gray este valoarea corespunzatoare tonului de gri aferent tripletului (Red, Green, Blue). Functia va primi drept parametri calea catre fisierul ce contine imaginea color, precum si locul in care trebuie salvata imaginea alb-negru obtinuta. Toate valorile pixelilor trebuie sa se gaseasca in intervalul 0-255. Folositi functia floor pentru conversia valorilor reale la valori intregi.

Hints: imread, imwrite, operatii cu matrici

Exemplu transformare:



Imagine originala, color



Imagine transformata, tonuri de gri

Task 1.2 (Basic - Histogram)

10p

Fisier .m: histogram.m

Cerinta: Realizati histograma unei imagini in tonuri de gri. Pentru aceasta trebuie sa inregistrati intr-un vector numarul de aparitii al fiecarei valori a pixelilor din imagine, astfel: pe pozitia i in vector se va gasi numarul de aparitii ale valorii $i - 1$. Functia va primi ca parametru calea catre fisierul ce contine imaginea si va intoarce vectorul cerut. Pentru mai multe detalii, urmariti scheletul de cod.

Hints: imread, find

Task 1.3 (Basic - Brightness)

10p

Fisier .m: brightness.m

Cerinta: Modificati luminozitatea unei imagini in tonuri de gri. Functia primeste doi parametri: calea catre imaginea originala si o valoare intreaga x cu care trebuie modificata luminozitatea imaginii. Valoarea fiecarui pixel se modifica dupa formula $V = V + x$. **Atentie**, valorile finale ale pixelilor trebuie sa fie in intervalul 0-255 (cele negative se vor considera 0, iar cele ce depasesc limita superioara se vor considera 255). Functia va intoarce valoarea medie a pixelilor din imaginea modificata.

Hints: imread, min, max, mean

Task 2 (Salt & Pepper)

30p

Fisier .m: salt-pepper.m

Zgomot in imagini: Zgomotul reprezinta stricarea valorilor anumitor pixeli din imaginea originala. In cadrul acestui task ne intereseaza un anumit tip de zgomot, numit "Salt and Pepper", care afecteaza pixeli aleatori, schimbandu-le valoarea in ALB sau NEGRU (de aici si numele).

Cerinta: Reparati fiecare pixel stricat din imagine. Valoarea sa va fi data de media valorilor celor 8 pixeli ce-l inconjoara pe fiecare canal in parte (Red, Green, Blue). Se garanteaza faptul ca pixelii de pe marginea imaginii nu vor fi niciodata afectati de zgomot. De asemenea, nu vor exista cazuri in care doi pixeli afectati de zgomot sunt vecini (ie. orice pixel stricat este inconjurat de pixeli nemodificati). Functia salt_pepper primeste trei parametri:

- calea catre imaginea originala (color)
- o masca ale carei elemente au semnificatia: $\text{mask}(i, j) = 1$ daca pixelul de coordonate (i, j) din imagine a fost afectat de zgomot si 0 altfel.
- calea catre fisierul in care trebuie salvata imaginea corectata



Imagine afectata de zgomot



Imagine corectata folosind procedeul de mai sus

Task 3 (Captcha)

50p

Fisier .m: captcha.m

Captcha: Captcha este o metoda automata de a determina daca un anumit utilizator este o persoana sau un program de calculator si se bazeaza pe dificultatea scrierii unui program care poate recunoaste corect literele dintr-o imagine. In general, textul generat pentru CAPTCHA este redat inghesuit, distorsionat sau amestecat cu alte elemente care fac fragmentarea sa in litere separate aproape imposibila cu tehnologia actuala. Totusi, exista pe Internet multe exemple de CAPTCHA nesigure, in care textul poate fi fragmentat, iar literele recunoscute cu usurinta.

t h | s | S | n s e C U r e C a P t C h A

Cerinta: Pentru rezolvarea acestui task trebuie sa identificati corect textul dintr-o imagine de tip CAPTCHA. Functia din fisierul captcha.m primeste ca parametru calea catre un fisier .png color care contine un text, ca in exemplu. Functia trebuie sa intoarca un sir de caractere care corespunde textului identificat. Pentru generarea imaginilor CAPTCHA am folosit urmatoarea procedura: pentru fiecare litera din text am ales aleator o imagine din directorul corespunzator din (letter_images) pe care am plasat-o aleator intr-un bloc de dimensiune prestabilita, colorand-o, de asemenea, aleator; in final am alipit blocurile obtinute, pe orizontala (adaugand, de asemenea, un spatiu de cel putin o coloana alba intre ele pentru a usura fragmentarea).

t e m a m E T o d E n u M e r I C e
t E m a M e t O D E N u m E r I c e
t e M a m E t O D e r U m E R I c E

README, coding style

10p

Se acorda 10 puncte in functie de README si coding style.

Resurse

In directorul letter_images gasiti mostre de imagini ce sunt folosite in generare testelor CAPTCHA, iar in directorul examples gasiti imaginile continute in acest enunt. De asemenea, puteti incarca masca folosita pentru a genera imaginea cu zgomot data ca exemplu folosind comanda

```
mask = load("-ascii", "examples/salt_pepper_mask.ascii");
```

Istoric modificari enunt & arhiva

13 martie 2013 - a fost publicata versiunea initiala.

14 martie 2013 - am adaugat in arhiva masca ce corespunde imaginii cu zgomot data ca exemplu.

17 martie 2013 - am adaugat in arhiva fisierul histogram_gray.ascii care contine histograma nivelelor de gri corespunzatoare fisierului example_gray.png.