

CONCEPTE SI APLICATII IN VEDEREA ARTIFICIALA

TEMA 1

MACIUCA GLORIA - RUXANDRA

GRUPA 334

a. Mozaic add_pieces_grid	1
Criteriul distantei euclidiene	
Numarul pieselor pe orizontala : 100, 75, 50, 25	
b. Mozaic add_pieces_random	8
Criteriul distantei euclidiene	
c. Mozaic add_pieces_grid	11
Criteriul distantei euclidiene	
Constrangere: piesele adiacente diferite	
d. Colectia de imagini CIFAR-10	13
e. Mozaic add_pieces_hexagon	19
Criteriul distantei euclidiene	
f. Mozaic add_pieces_hexagon	21
Criteriul distantei euclidiene	
Constrangere: piesele adiacente diferite	
DESPRE ALGORITM	23

a. Mozaic `add_pieces_grid`

Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '`.../data/imaginiTest/`' la rulara algoritmului pentru diferite valori ale parametrului `num_pieces_horizontal` (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.



100 piese



75 piese



50 piese



25 piese



adams original

Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '*.../data/imaginiTest/*' la rulara algoritmului pentru diferite valori ale parametrului *num_pieces_horizontal* (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.



100 piese



75 piese



50 piese



25 piese



ferrari original

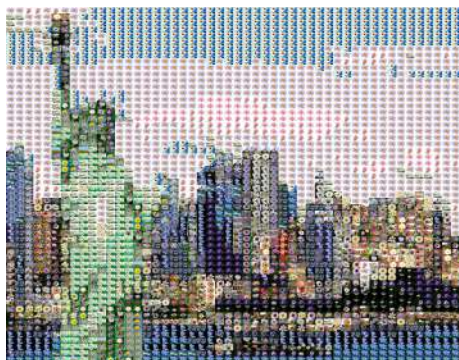
Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '*.../data/imaginiTest/*' la rularea algoritmului pentru diferite valori ale parametrului *num_pieces_horizontal* (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.



100 piese



75 piese



50 piese



25 piese



liberty original

Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '*.../data/imaginiTest/*' la rulara algoritmului pentru diferite valori ale parametrului *num_pieces_horizontal* (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.



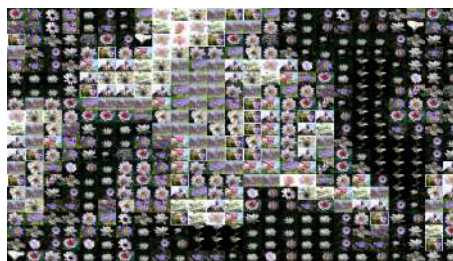
100 piese



75 piese



50 piese



25 piese



obama original

Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '*.../data/imaginiTest/*' la rulara algoritmului pentru diferite valori ale parametrului *num_pieces_horizontal* (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.



100 piese



75 piese



50 piese



25 piese



romania original

Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '*.../data/imaginiTest/*' la rularea algoritmului pentru diferite valori ale parametrului *num_pieces_horizontal* (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.



100 piese



75 piese



50 piese



25 piese



tomJerry original

Mozaicurile obtinute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '*.../data/imaginiTest/*' la rulara algoritmului pentru diferite valori ale parametrului *num_pieces_horizontal* (100, 75, 50, 25) si criteriul distantei euclidiene dintre culorile medii, comentand influenta lor asupra rezultatelor obtinute.

OBSERVATII:

- Mozaicurile realizate dintr-un numar mare de piese pe orizontala sunt mult mai apropiate de imaginile originale decat mozaicurile realizate dintr-un numar mic de piese.
- Imaginile originale sunt redimensionate functie de numarul de piese pe orizontala. Asadar, o portiune intr-un mozaic cu numar mare de piese reprezinta o zona de dimensiuni mici in imaginea originala, astfel pixelii respectivi vor avea valori apropiate.
- In momentul calcularii valorii medii a culorilor pentru o portiune intr-un mozaic cu numar mare de piese, valoarea obtinuta este apropiata de valoarea fiecarui pixel, astfel acoperirea cu o piesa va rezulta intr-o asemanare mare cu portiunea corespunzatoare in imaginea originala.
- Bineinteles, timpul necesar construirii unui mozaic cu numar mare de piese este mai crescut decat timpul necesar construirii unui mozaic cu numar mic de piese.

b. Mozaic `add_pieces_random`

Mozaicurile obtinute pentru imaginile din directorul '`.../data/imaginiTest/`' pentru modul de aranjare 'aleator' si criteriul distantei euclidiene dintre culorile medii. Pentru rezolvarea acestui punct trebuie sa scrieti functia `add_pieces_random`.

MENTIUNE:

- Mozaicurile construite prin metoda `add_pieces_random` sunt completate in proportie de 99%, din cauza timpului mare de executie.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

Mozaicurile obtinute pentru imaginile din directorul '*.../data/imaginiTest/*' pentru modul de aranjare 'aleator' si criteriul distantei euclidiene dintre culorile medii. Pentru rezolvarea acestui punct trebuie sa scrieti functia *add_pieces_random*.

MENTIUNE:

- Mozaicurile construite prin metoda *add_pieces_random* sunt completate in proportie de 99%, din cauza timpului mare de executie.

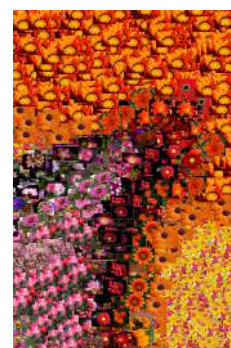
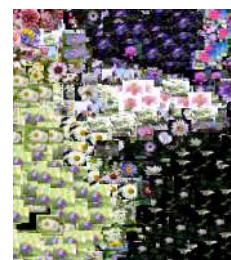


Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

Mozaicurile obtinute pentru imaginile din directorul `'.../data/imaginiTest/'` pentru modul de aranjare `'aleator'` si criteriul distantei euclidiene dintre culorile medii. Pentru rezolvarea acestui punct trebuie sa scrieti functia `add_pieces_random`.

MENTIUNE:

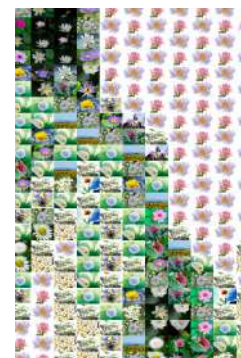
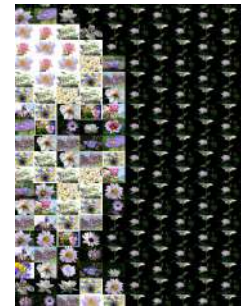
- Mozaicurile construite prin metoda `add_pieces_random` sunt completate in proportie de 99%, din cauza timpului mare de executie.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

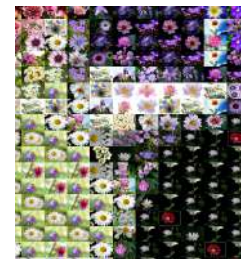
c. Mozaic `add_pieces_grid`

Mozaicurile obtinute pentru imaginile din directorul '`.../data/imaginiTest/`' astfel: implementati o modificare a functiei `add_pieces_grid` astfel incat mozaicul obtinut sa aiba proprietatea ca nu exista doua piese adiacente identice.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

Mozaicurile obtinute pentru imaginile din directorul '*.../data/imaginiTest/*' astfel: implementati o modificare a functiei *add_pieces_grid* astfel incat mozaicul obtinut sa aiba proprietatea ca nu exista doua piese adiacente identice.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

d. Colectia de imagini CIFAR-10

Inlocuiti colectia furnizata cu o colectie proprie. Realizati mozaicuri tematice, construind mozaicuri pentru obiecte din aceste clase cu piesele corespunzatoare.



Primele 20 de imagini din clasa 'airplane' din setul de test CIFAR-10.



Mozaic realizat prin modul de aranjare *caroiaj* si criteriul *distanței euclidiene* cu piese hexagonale si proprietatea ca piesele adiacente sunt diferite.



Mozaic realizat prin modul de aranjare *caroiaj* si criteriul *distanței euclidiene* cu piese hexagonale.

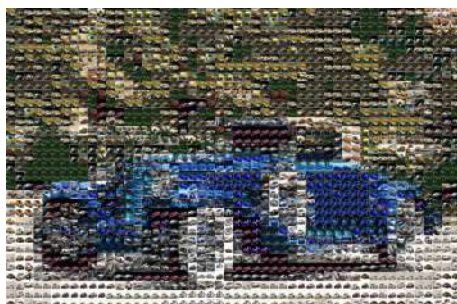


Imaginea originală.

Inlocuiti colectia furnizata cu o colectie proprie. Realizati mozaicuri tematice, construind mozaicuri pentru obiecte din aceste clase cu piesele corespunzatoare.



Primele 20 de imagini din clasa 'automobile' din setul de test CIFAR-10.



Mozaic realizat prin modul de aranjare *caroiaj* si criteriul *distanței euclidiene* cu 50 piese pe orizontala.



Mozaic realizat prin modul de aranjare *caroiaj* si criteriul *distanței euclidiene* cu 100 piese pe orizontala.



Imaginea originala.

Inlocuiti colectia furnizata cu o colectie proprie. Realizati mozaicuri tematice, construind mozaicuri pentru obiecte din aceste clase cu piesele corespunzatoare.



Primele 20 de imagini din clasa 'bird' din setul de test CIFAR-10.



Mozaic realizat prin modul de aranjare *aleator* si criteriul *distanței euclidiene*.



Mozaic realizat prin modul de aranjare *carotid* si criteriul *distanței euclidiene*.



Imaginea originală.

Inlocuiti colectia furnizata cu o colectie proprie. Realizati mozaicuri tematice, construind mozaicuri pentru obiecte din aceste clase cu piesele corespunzatoare.



Primele 20 de imagini din clasa 'dog' din setul de test CIFAR-10.



Mozaic realizat prin modul de aranjare *carotiaj* si criteriul *distanței euclidiene*



Mozaic realizat prin modul de aranjare *carotiaj* si criteriul *distanței euclidiene* cu piese hexagonale.



Imaginea originala.

Inlocuiti colectia furnizata cu o colectie proprie. Realizati mozaicuri tematice, construind mozaicuri pentru obiecte din aceste clase cu piesele corespunzatoare.



Primele 20 de imagini din clasa 'horse' din setul de test CIFAR-10.



Mozaic realizat prin modul de aranjare *aleator* si criteriul *distanței euclidiene*.



Mozaic realizat prin modul de aranjare *carotid* si criteriul *distanței euclidiene*.



Imaginea originala.

Inlocuiti colectia furnizata cu o colectie proprie. Realizati mozaicuri tematice, construind mozaicuri pentru obiecte din aceste clase cu piesele corespunzatoare.



Primele 20 de imagini din clasa 'ship' din setul de test CIFAR-10.



Mozaic realizat prin modul de aranjare *carioaj* si criteriul *distanței euclidiene*.



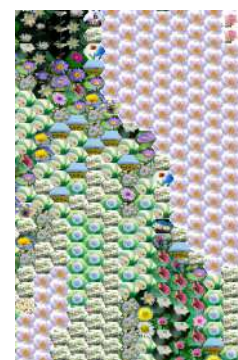
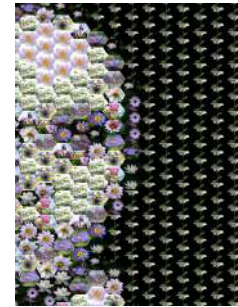
Mozaic realizat prin modul de aranjare *carioaj* si criteriul *distanței euclidiene* si proprietatea ca piesele adiacente sunt diferite.



Imaginea originala.

e. Mozaic `add_pieces_hexagon`

Modificati proiectul astfel incat mozaicurile obtinute sa aiba piese hexagonale. Includeti mozaicurile obtinute astfel pentru imaginile din directorul `'.../data/imaginiTest/'`.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

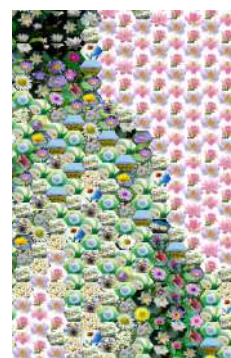
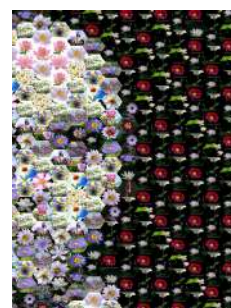
Modificati proiectul astfel incat mozaicurile obtinute sa aiba piese hexagonale. Includeti mozaicurile obtinute astfel pentru imaginile din directorul `'.../data/imaginiTest/'`.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

f. Mozaic `add_pieces_hexagon`

Modificati punctul anterior astfel incat mozaicurile obtinute sa aiba proprietatea ca nu exista doua piese hexagonale adiacente identice. Includeti mozaicurile obtinute astfel pentru imaginile din directorul `'.../data/imaginiTest/'`.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

Modificati punctul anterior astfel incat mozaicurile obtinute sa aiba proprietatea ca nu exista doua piese hexagonale adiacente identice. Includeti mozaicurile obtinute astfel pentru imaginile din directorul '*.../data/imaginiTest/*'.



Stanga: imaginile originale.
Centru: mozaicurile obtinute.
Dreapta: zoom mozaic.

DESPRE ALGORITHM

Parameters.py

Am adaugat in clasa *'Parameters'*:

```
self.constraint = None
    precizeaza daca se doreste proprietatea ca piesele adiacente sa fie diferite

self.small_images_avg_color = None
    matrice care memoreaza valorile culorilor medii ale imaginilor din colectia de piese, pentru
    a evita calcule repetate

self.output_path = './../data/output/'
    adresa la care se vor salva imaginile mozaic construite
```

BuildMosaic.py

Am modificat functia in *'load.pieces'* modul in care se face memorarea imaginilor din colectie (din matrice $H \times W \times C \times N$ in matrice $N \times H \times W \times C$), deoarece functiile deja completate din scriptul *'AddPiecesMosaic.py'* prelucreaza piesele pentru mozaic pornind de la matrici cu forma $N \times H \times W \times C$. Toate functiile au fost scrise considerand forma modificata.

Am adaugat functia:

```
set_mosaic_name
    construiesc numele imaginii in care este salvat un mozaic
```

ProcessMosaic.py

In acest script am adaugat functii pentru calculele necesare alegerii pieselor pentru mozaic.

```
compute_average_color_small_img , compute_average_color_ref_img
    calculeaza culorile medii ale imaginilor din colectie si a sectiunii specificate din imaginea de
    referinta

compute_euclidean_distance
    calculeaza distanta euclidiană între culorile medii ale unei imagini din colectie si a unei sectiuni
    din imaginea dereferinta

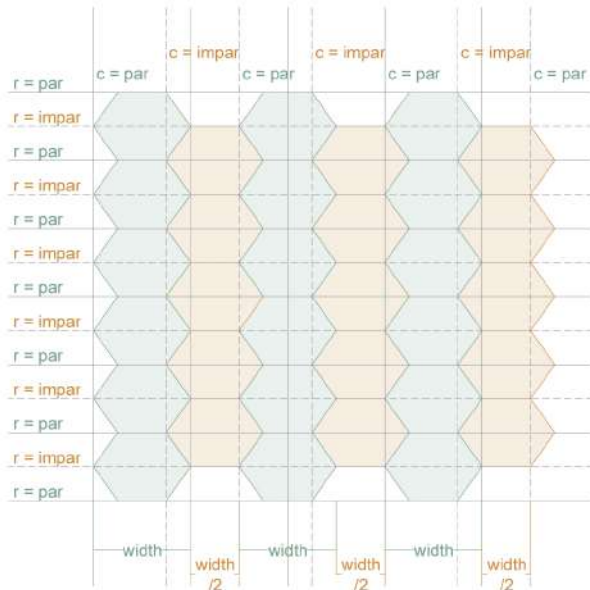
get_neighbours_grid , get_neighbours_hexa
    returneaza piesele vecine unei pozitii in mozaic, pentru caroiă rectangular, respectiv hexagonal

num_filled_pixels
    folosita pentru modul de construire 'aleator' a mozaicului, calculeaza numarul de pixeli nou
    acoperiti de o piesa adaugata pe o pozitie specificata

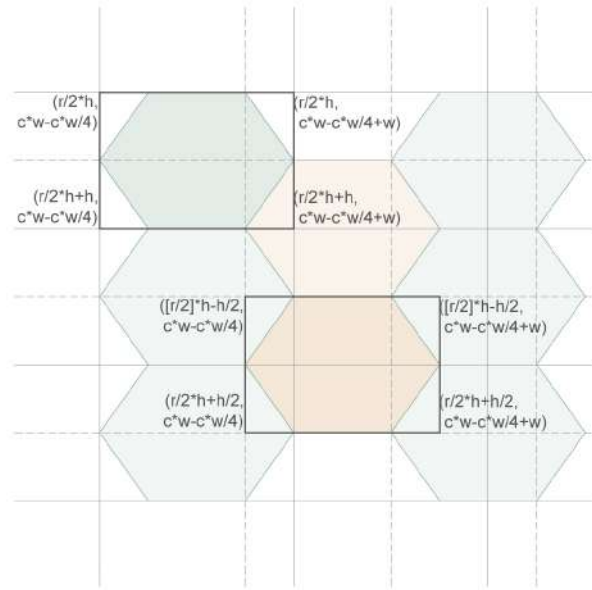
create_mask
    construiesc o masca hexagonală de dimensiunile unei piese de mozaic
```

AddPiecesMosaic.py

add_pieces_hexagon



(a)



(b)

Ilustrarea modului de reprezentare a pozitiilor pieselor hexagonale pe un grid rectangular.

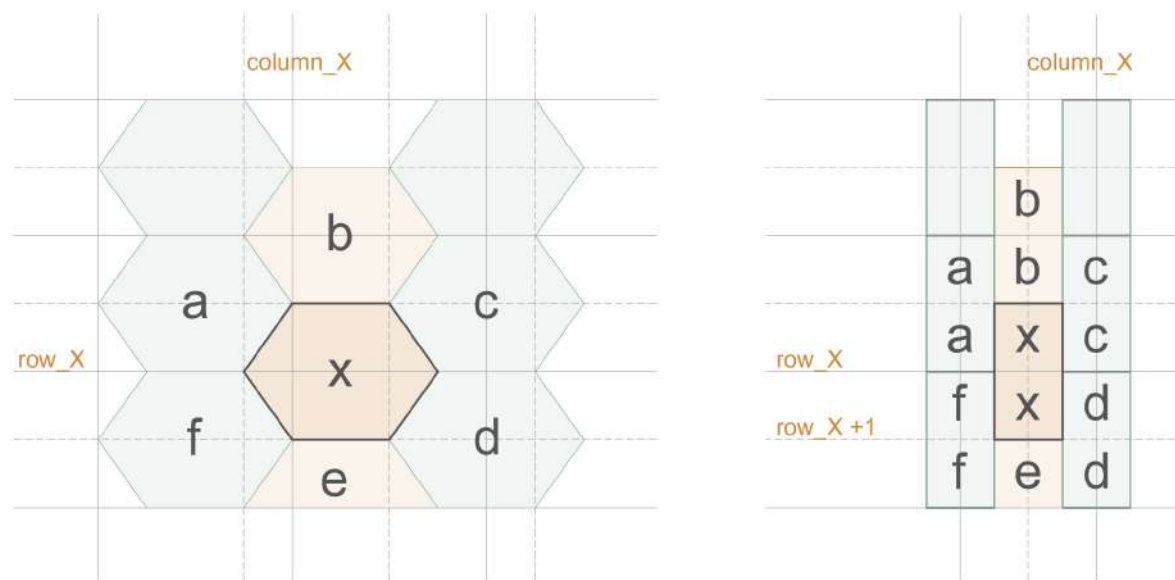
Cunoscand dimensiunile w si h ale unei piese de mozaic rectangulare si alegand un numar N de piese pe orizontala si M piese pe verticala, obtinem dimensiunile W si H ale mozaicului. Observam ca in cazul lucrului cu piese hexagonale, pe latimea mozaicului intalnim o insiruire de dimensiuni $w + w/2$ (a).

Asadar:

$$\begin{aligned} W &= 100 \cdot w \\ W &= w + w/2 + w + w/2 + w + w/2 + \dots \\ W &= x \cdot 3w/2 \\ x &= 2 \cdot N/3 \end{aligned}$$

Am obtinut $4/3 \cdot N$ pozitii pe orizontala si $2 \cdot M$ pozitii pe verticala.

De asemenea, observam si modul de tratare a pozitiei pe linie, functie de paritatea acesteia (b).



Ilustrarea modului de reprezentare a vecinilor.

Pentru fiecare piesa asezata in mozaic, in matricea-evidenta a pieselor, notam piesa respectiva pe pozitia coloana curenta si linia curenta, dar si pe coloana curenta si urmatoarea linie.

ProcessCifar10.py

Scriptul contine functii pentru prelucrarea setului de imagini CIFAR-10:

load_cifar10_test_batch

incarca informatii despre imagini, denumirile lor si clasele de care apartin

create_cifar10_collections

salveaza informatiile citite sub forma de imagini '.png'

plot_cifar

plotarea primelor 20 de imagini din clasa specificata

RunProjectCifar.py

Scriptul pentru rularea proiectului, modificat pentru a folosi date din colectia CIFAR-10.