

Concepte și Aplicații în Vederea Artificială

Tema de laborator 1 - Realizarea imaginilor mozaic

Obiectiv

Scopul acestei prime teme de laborator este realizarea imaginilor de tip *mozaic* (Figura 1). Un *mozaic* se obține prin înlocuirea blocurilor de pixeli dintr-o imagine de referință cu imagini de dimensiuni mici (le numim *piesele mozaicului*) dintr-o colecție dată. Înlocuirea blocurilor de pixeli cu piese se realizează astfel încât mozaicul rezultat să semene cât mai mult cu imaginea de referință.



Figura 1: **Imagine de tip mozaic:** pentru imaginea de referință din stânga obținem mozaicul din dreapta. Pieselete mozaicului sunt alese dintr-o colecție dată cu 500 de imagini. Fiecare piesă reprezintă o floare (puteți mări documentul pentru a observa acest lucru).

Prima parte - punct de pornire

Scriptul *run_project.py* vă oferă un punct de pornire în implementarea voastră. Completând toate funcțiile proiectului veți rula acest script și veți obține un mozaic (pe baza parametrilor setați). În acest script setăm imaginea de referință pentru care realizăm mozaicul și dăriți parametri:

- numele directorului în care se află colecția de imagini de dimensiuni mici ce constituie piesele mozaicului;

- formatul imaginilor din colecție ('jpg', 'jpeg', 'png', 'tiff', etc) ale pieselor mozaicului. Toate piesele mozaicului trebuie să aibă același format.
- numărul de piese pe orizontală ale mozaicului (setat implicit cu valoarea 100), ce definește dimensiunile mozaicului rezultat;
- modalitatea de aranjare a pieselor în mozaic. Folosim două modalități de aranjare: (1) aranjare pe un caroaj dreptunghiular, în care piesele sunt puse una după alta și se potrivesc perfect, fără suprapunerii; (2) la întâmplare, în care piesele sunt puse la poziții aleatoare în mozaic, cu suprapunerii.
- criteriul după care se aleg piesele pentru realizarea mozaicului. În acest proiect veți folosi două criterii: (1) *aleator*; (2) *după culoarea medie*. Fiecare criteriu este descris în detaliu în Secțiunea 1.3.

După setarea parametrilor, scriptul *run_project.py* apelează scriptul *build_mosaic.py* care la rândul lui apelează următoarele funcții:

- *load_pieces* - folosim această funcție pentru a încărca piesele colecției (detalii în Secțiunea 1.1), funcția este scrisă parțial;
- *compute_dimensions* - folosim această funcție pentru a calcula numărul de piese ale mozaicului (detalii în Secțiunea 1.2), funcția trebuie scrisă complet;
- *add_pieces_mosaic.py* - folosim această funcție pentru a implementa diversele opțiuni de aranjare a pieselor (detalii în Secțiunea 1.3). Funcția *add_pieces_grid* este scrisă parțial. Funcțiile *add_pieces_random* și *add_pieces_hexagon* (vedeți partea a doua) trebuie scrise complet.

1.1 Încărcarea pieselor mozaicului - **Pasul 1**

Realizați mai întâi operația de citire a pieselor mozaicului din directorul ce conține colecția de piese, completând funcția *load_pieces*. Inspirați-vă în scrierea ei din fragmentul de cod folosit în Laboratorul 1 pentru citirea imaginilor cu extensia '.jpg' din directorul 'set1'. Funcția afișează la final într-o figură primele 100 de piese ale mozaicului citite din directorul corespunzător. În Figura 2 sunt afișate primele 15 astfel de piese.

1.2 Calculul dimensiunilor mozaicului - **Pasul 2**

Pentru obținerea unui mozaic care să fie cât mai asemănător cu imaginea de referință vom construi un mozaic de dimensiuni modificate. Parametrul *num_pieces_horizontal* (setat implicit cu valoarea 100) definește numărul de piese pe orizontală ale mozaicului. Pe baza acestui parametru, ale dimensiunilor inițiale ale imaginii de referință și ale dimensiunilor pieselor (28×40 pixeli) putem determina dimensiunea mozaicului pe care îl vom construi. În construcția mozaicului vom folosi imaginea de referință redimensionată corespunzător. Funcția *compute_dimensions* trebuie scrisă complet.



Figura 2: Colecția de piese pentru realizarea mozaicului: Primele 15 piese din colecție care sunt folosite pentru realizarea mozaicului. Fiecare piesă din colecție este color și are dimensiunile 28×40 pixeli.

1.3 Adăugarea pieselor mozaicului - **Pasul 3**

Piese sunt adăugate cu scopul de a obține un mozaic cât mai asemănător cu imaginea de referință redimensionată. În implementarea voastră, veți adăuga piesele inițiale (drept-unghiulare) prin două modalități de aranjare a lor: (1) pe caroaj, adăugând piesele de la stânga la dreapta și de sus în jos. Piese astfel adăugate se potrivesc perfect, fără a se suprapune; (2) în mod aleator, adăugând de fiecare dată o piesă într-o poziție aleasă la întâmplare în mozaic. Un asemenea mod de aranjare conduce la suprapunerile de piese.

Fiecare piesă este selectată pe baza unui criteriu. În acestă temă de laborator considerăm două criterii posibile:

- *aleator* - folosind acest criteriu înlocuim blocurile de pixeli din imaginea de referință redimensionată cu piese alese în mod aleator. În consecință, mozaicul pe care îl obținem nu va semăna cu imaginea de referință.
- *pe baza distanței euclidiene dintre culorile medii* - folosind acest criteriu înlocuim blocurile de pixeli din imaginea de referință redimensionată cu piese alese astfel încât culoarea medie a piesei să fie cât mai apropiată (în sensul distanței euclidiene) de culoarea medie a blocului înlocuit. Mozaicul din Figura 1 este obținut folosind acest criteriu.

Completați funcția *add_pieces_grid* astfel încât piesele să fie așezate pe caroaj folosind cele două criterii:

- (a) aleator;
- (b) pe baza distanței euclidiene dintre culorile medii (Figura 3);

Scrieți funcția `add_pieces_random` astfel încât piesele să fie așezate în mod aleator în imagine (Figura 4).

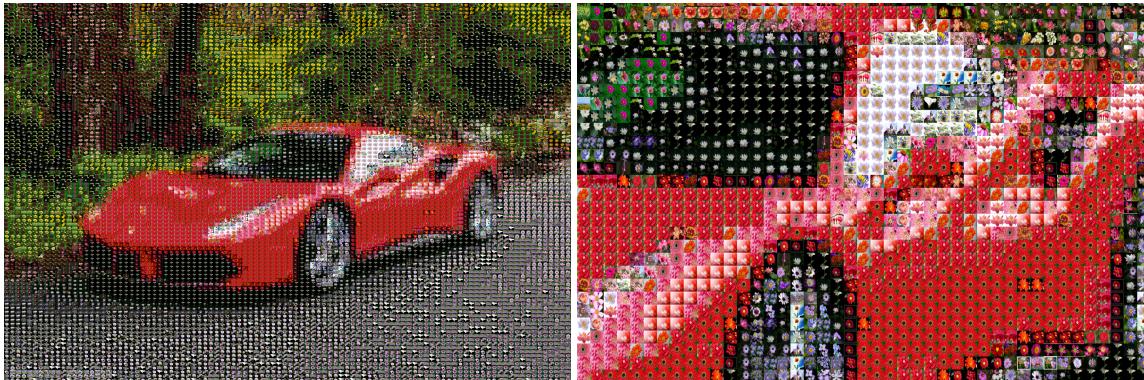


Figura 3: **Mozaic cu piese aranjate pe caroaj.** (a) imagine de tip mozaic obținuta pentru imaginea din Figura 1 (b) zoom al imaginii obținute.



Figura 4: **Modalitatea de aranjare a pieselor în mod aleator.** (a) imagine de tip mozaic obținuta pentru imaginea din Figura 1 pentru modalitatea de aranjare a pieselor în mod aleator; (b) zoom al imaginii obținute.

Pentru prima parte a temei, implementați pașii 1, 2 și 3 descriși mai sus și realizați următoarele:

- (a) implementați algoritmul de obținere a unui mozaic pentru o imagine de referință folosind adăugarea pieselor pentru modul de aranjare *caroaj* și ambele criterii: criteriu selecției pieselor *aleator* și criteriu distanței euclidiene dintre culorile medii.
Asigurați-vă că implementarea voastră merge bine și în cazul imaginilor grayscale (un singur canal).
- (b) rulați algoritmul pentru modul de aranjare *caroaj* pentru imaginile din directorul '`../data/imaginiTest/`' pentru diferite valori ale parametrului `num_pieces_horizontal` (100, 75, 50, 25) și criteriu distanței euclidiene dintre culorile medii comentând influența lor asupra rezultatelor obținute.

- (c) implementați algoritmul de obținere a unui mozaic pentru o imagine de referință folosind adăgarea pieselor pentru modul de aranjare *aleator* și criteriul distanței euclidiene dintre culorile medii. Rulați acest algoritm pentru imaginile din directorul '*../data/imaginiTest/*' (Figura 4).

Partea a doua - folosirea pieselor hexagonale - **Pasul 4**

În a doua parte a temei de laborator veți lucra cu aceleași piese dreptunghiulare asupra cărora aplicați o mască binară astfel încât forma lor să pară hexagonală (Figura 5). Scrieți funcția *add_pieces_hexagon* astfel încât piesele adăugate să aibă formă hexagonală și să se potrivească perfect, ținând cont că puteți acoperi o suprafață plană (în cazul nostru, o imagine) cu hexagoane regulate (Figura 6). În Figura 5 puteți observa modalitatea de obținere și alăturare a pieselor hexagonale.

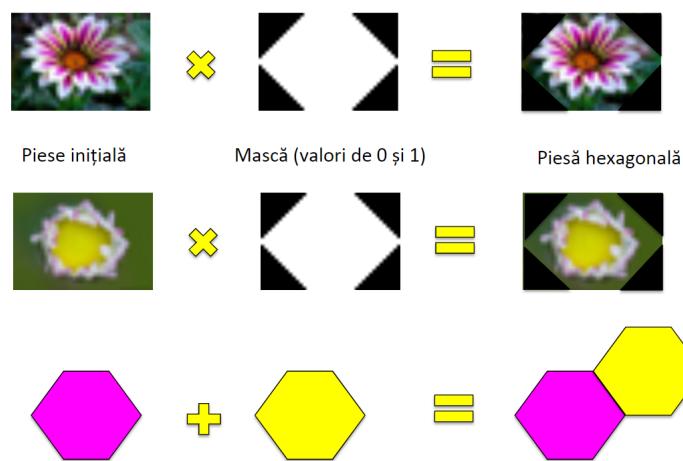


Figura 5: Obținerea pieselor hexagonale și modalitatea de alăturare

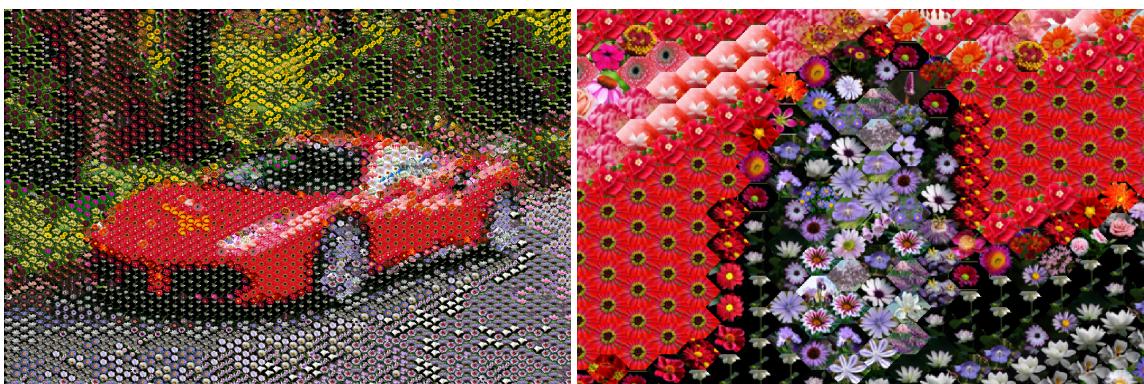


Figura 6: **Mozaic cu piese hexagonale.** (a) imagine de tip mozaic obținuta pentru imaginea din Figura 1 ce are proprietatea că fiecare piesă este hexagonală; (b) zoom al imaginii obținute.

Pentru a doua parte a temei, implementați pasul 4 și realizați următoarele:

- (a) rulați algoritmul implementat pentru imaginile din directorul `'.../data/imaginiTest/'` pentru piese hexagonale pentru diferite valori ale parametrului `num_pieces_horizontal` (100, 75, 50, 25)
- (b) înlocuiți colecția furnizată de noi (cu cele 500 de piese ce reprezintă flori) cu o colecție proprie. O posibilitate ar fi să downloadați setul de date *CIFAR – 10* de la adresa <https://www.cs.toronto.edu/~kriz/cifar.html>. *CIFAR – 10* conține 60000 de imagini color de dimensiuni 32×32 pixeli cu obiecte din 10 clase: avion, automobil, etc. Realizați mozaicuri tematice, construind mozaicuri pentru imagini conținând obiecte din aceste clase cu piesele corespunzătoare: realizați un mozaic pentru o imagine cu un automobil folosind piese cu automobile.