

Concepte și Aplicații în Vederea Artificială - Tema 1

Realizarea imaginilor mozaic

Obiectiv

Scopul acestei teme este realizarea imaginilor de tip mozaic (Figura 1). O imagine de tip mozaic (o numim *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 aproximeze cât mai bine imaginea de referință.

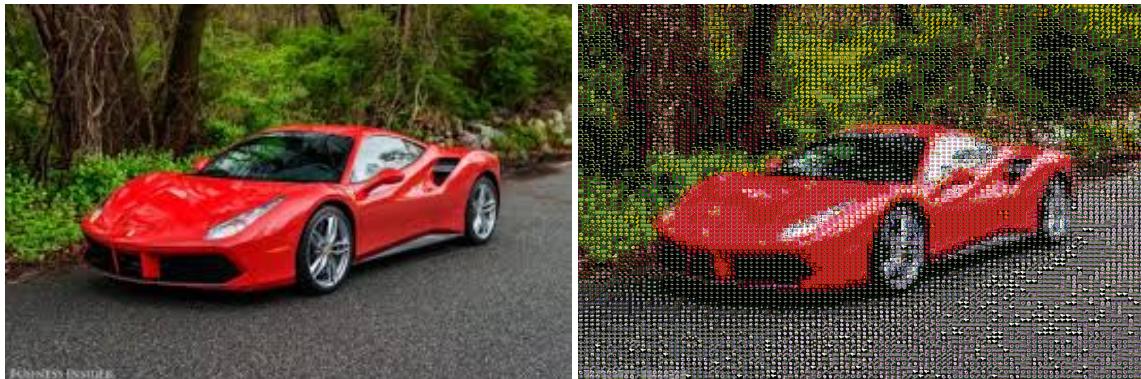


Figura 1: **Imagine de tip mozaic:** pentru imaginea de referință din stânga obținem mozaicul din dreapta. Pieselete mozaicului sunt alese dintr-o colecție dată cu 500 de imagini. Fiecare piesă reprezintă o floare (puteți mări documentul pentru a observa acest lucru).

Funcțiile Python care vă vor ajuta la implementarea temei sunt în directorul *cod*; imaginile pe care le veți folosi sunt în directorul *data*. **Tema este individuală.** Toate detaliile legate de predarea temei le găsiți la sfârșitul acestui document. Tema a fost prezentată și în slide-urile din cursul 2.

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 mozaicul dorit. În acest script se setează imaginea de referință pentru care se va realiza mozaicul și se setează diferenți parametri printre care:



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.

- numele directorului în care se află colecția de imagini de dimensiuni mici ce vor constitui 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. În acest proiect veți folosi două modalități de aranjare: (1) aranjare pe un caroaj, în care piesele sunt puse una după alta și se îmbină 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 urmatoarele funcții:

- `load_pieces` - funcția este scrisă parțial, trebuie să o completați voi;
- `compute_dimensions` - funcția trebuie completată în întregime de către voi;
- în funcție de opțiunea pentru aranjarea pieselor apelează `add_pieces_mosaic.py`. Funcția `add_pieces_grid` este scrisă parțial, trebuie să o completați voi. Funcția `add_pieces_random` și `add_pieces_hexagon` trebuie scrise în întregime de voi.

1.1 Încărcarea pieselor mozaicului

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

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 completată de către voi.

1.3 Adăugarea pieselor mozaicului

Piese sunt adăugate cu scopul de a obține un mozaic cât mai asemănător cu imaginea de referință redimensionată. În implementarea voastră, veți adăuga piesele 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 îmbină 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 aranjare conduce la suprapunerile de piese.

Fiecare piesă este selectată pe baza unui criteriu. În acest proiect există două criterii posibile:

- *aleator* - folosind acest criteriu veți înlocui blocurile de pixeli din imaginea de referință redimensionată cu piese alese în mod aleator. În consecință, mozaicul pe care îl veți obține nu va semăna cu imaginea de referință. Acest criteriu este deja implementat în Python însă merge numai pentru imaginile color (pentru imaginile grayscale nu funcționează).
- *pe baza distanței euclidiene dintre culorile medii* - folosind acest criteriu veți înlocui 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. Mozaicul din Figura 1 este obținut folosind acest criteriu. Va trebui să scrieți codul Python care implementează acest criteriu.

1.4 Predarea temei

Puneți într-o arhivă cu numele `tema1_cod.zip` codul vostru Python. Puneți într-un document cu numele `tema1_rezultate.pdf` următoarele:

- (a) **(3 puncte)** mozaicurile obținute pentru modul de aranjare 'caroiaj' pentru imaginile din directorul '`../data/imaginiTest/`' la rularea algoritmului vostru pentru diferite valori ale parametrului `num_pieces_horizontal` (100, 75, 50, 25) și criteriul distanței euclidiene dintre culorile medii comentând influența lor asupra rezultatelor obținute.
- (b) **(1 punct)** mozaicurile obținute pentru imaginile din directorul '`../data/imaginiTest/`' pentru modul de aranjare 'aleator' și criteriul distanței euclidiene dintre culorile medii. Pentru rezolvarea acestui punct trebuie să scrieți funcția `add_pieces_random`. Pentru imaginea inițială din Figura 1 ar trebui să obțineți ceva similar cu imaginile din Figura 3.
- (c) **(1 punct)** mozaicurile obținute pentru imaginile din directorul '`../data/imaginiTest/`' astfel: implementați o modificare a funcției `add_pieces_grid` astfel încât mozaicul obținut să aibă proprietatea că nu există două piese adiacente (stânga, dreapta, jos, sus) identice. Pentru imaginea inițială din Figura 1 ar trebui să obțineți ceva similar cu imaginile din Figura 4.
- (d) **(1 punct):** 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. Includeți în pdf-ul vostru cel puțin 5 exemple de mozaicuri tematice.
- (e) **(1 punct):** modificați proiectul vostru astfel încât mozaicurile obținute să aibă piese hexagonale. Includeți în pdf-ul vostru mozaicurile obținute astfel pentru imaginile din directorul '`../data/imaginiTest/`'. Pentru imaginea inițială din Figura 1 ar trebui să obțineți ceva similar cu imaginile din Figura 5.
- (f) **(1 punct):** modificați punctul anterior astfel încât mozaicurile obținute să aibă proprietatea că nu există două piese hexagonale adiacente (sus, jos, stânga sus, stânga jos, dreapta sus, dreapta jos) identice. Includeți în pdf-ul vostru mozaicurile obținute astfel pentru imaginile din directorul '`../data/imaginiTest/`'.

Se va nota cu 1 punct prezentarea proiectului în format pdf. Vom lua în calcul aspecte precum: așezarea în pagină, comentariile ce însotesc imaginile prezentate, exemplele alese.

Oficiu: 1 punct.



Figura 3: **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.

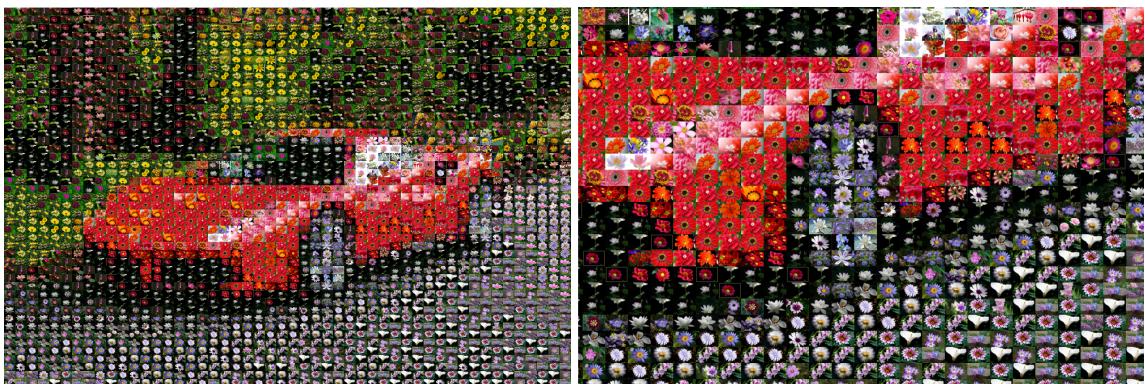


Figura 4: **Mozaic cu piese adiacente diferite.** (a) imagine de tip mozaic obținuta pentru imaginea din Figura 1 ce are proprietatea că fiecare piesă are toți cei 4 vecini diferenți de ea; (b) zoom al imaginii obținute.

Trimiteți cele două fișiere (*tema1.cod.zip* și *tema1_rezultate.pdf*) la adresa de email a lui Bogdan, bogdan.alexe@fmi.unibuc.ro.

Termenul limită de predare a proiectului este duminică, 8 noiembrie 2020, ora 23:59. Fiecare zi de întârziere în predarea proiectului se penalizează cu 1 punct în minus.

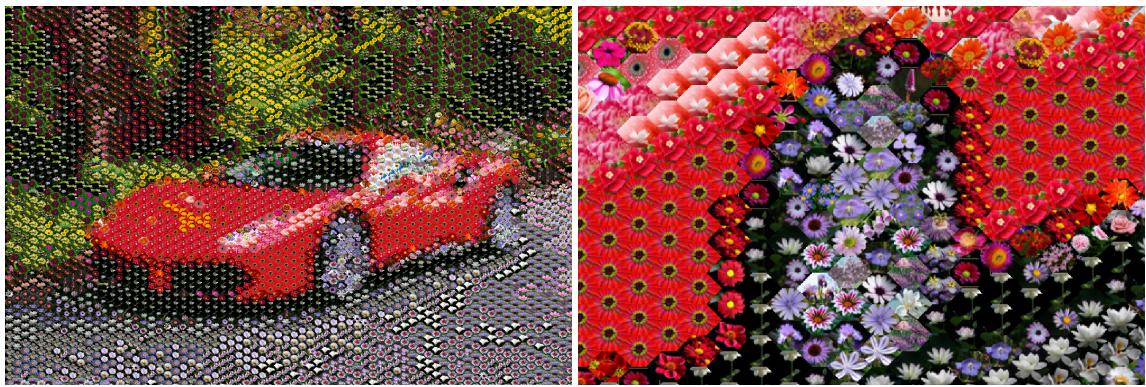


Figura 5: **Mozaic cu piese hexagonale.** (a) imagine de tip mozaic obținuta pentru imaginea din Figura 1 ce are proprietatea că fiecare piesă este hexagonală; (b) zoom al imaginii obținute.