Universitatea din Bucureşti
Facultatea de Matematică şi Informatică
Departamentul de Informatică

Iuliana Georgescu
Bogdan Alexe
Radu Ionescu
Informatică , anul 3, opţional

Concepte şi Aplicaţii 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 obţine prin înlocuirea blocurilor de pixeli dintr-o imagine de referinţă cu imagini de dimensiuni mici (le numim *piesele mozaicului*) dintr-o colecţie dată. Înlocuirea blocurilor de pixeli cu piese se realizează astfel încât mozaicul rezultat să aproximeze cât mai bine imaginea de referinţă.

Figura 1: **Imagine de tip mozaic:** *pentru imaginea de referinţă din stânga obţinem mozaicul din dreapta. Piesele mozaicului sunt alese dintr-o colecţie dată cu* 500 *de imagini. Fiecare piesă reprezintă o floare (puteţi mări documentul pentru a observa acest lucru).*

Funcţiile Python care vă vor ajuta la implementarea temei sunt în directorul *cod*; imaginile pe care le veţi folosi sunt în directorul *data*. ***Tema este individuală***. Toate detaliile legate de predarea temei le găsiţi la sfârşitul acestui document. Tema a fost prezentată şi in slide-urile din cursul 2.

Punct de pornire

Scriptul *RunProject.py* vă oferă un punct de pornire în implementarea voastră. Completând toate funcţiile proiectului veţi rula acest script şi veţi obţine mozaicul dorit. În acest script se setează imaginea de referinţă pentru care se va realiza mozaicul şi se setează diferiţi parametri printre care:

Figura 2: **Colecţia de piese pentru realizarea mozaicului:** *Primele* 15 *piese din colecţie care sunt folosite pentru realizarea mozaicului. Fiecare piesă din colecţie este color şi are dimensiunile* 28×40 *pixeli.*

- numele directorului în care se află colecţia de imagini de dimensiuni mici ce vor constitui piesele mozaicului;
- formatul imaginilor din colecţie ('jpg', 'jpeg', 'png', 'tiff', etc) ale pieselor mozaicului. Toate piesele mozaicului trebuie să aibă acelaşi format.
- numărul de piese pe orizontală ale mozaicului (setat implicit cu valoarea 100), ce defineşte dimensiunile mozaicului rezultat;
- modalitatea de aranjare a pieselor în mozaic. În acest proiect veţi folosi două modalităţi de aranjare: (1) aranjare pe un caroiaj, în care piesele sunt puse una după alta şi se îmbină perfect, fără suprapuneri; (2) la întâmplare, în care piesele sunt puse la poziţii aleatoare în mozaic, cu suprapuneri.
- criteriul după care se aleg piesele pentru realizarea mozaicului. În acest proiect veţi folosi două criterii: (1) aleator; (2) după culoarea medie. Fiecare criteriu este descris în detaliu în secţiunea 1.3.

După setarea parametrilor, scriptul *RunProject.py* apelează scriptul *BuildMosaic.py* care la rândul lui apelează urmatoarele funcţii:

- *load_pieces* - funcţia este scrisă parţial, trebuie să o completaţi voi;
- *compute_dimensions* - funcţia trebuie completată în întregime de către voi;
- în funcţie de opţiunea pentru aranjarea pieselor apelează *AddPiecesMosaic.py*. Funcţia *add_pieces_grid* este scrisă parţial, trebuie să o completaţi voi. Funcţia *add_pieces_random* şi *add_pieces_hexagon* trebuie scrise în întregime de voi.

1.1 Încărcarea pieselor mozaicului

Realizaţi mai întâi operaţia de citire a pieselor mozaicului din directorul ce conţine colecţia de piese, completând funcţia *load_pieces*. Inspiraţi-vă în scrierea ei din fragmentul de cod folosit în Laboratorul 1 pentru citirea imaginilor cu extensia '.jpg' din directorul 'set1'. Funcţia afişează la final într-o figură primele 100 de piese ale mozaicului citite din directorul corespunzător. În Figura 2 sunt afişate primele 15 astfel de piese.

1.2 Calculul dimensiunilor mozaicului

Pentru obţinerea unui mozaic care să fie cât mai asemănător cu imaginea de referinţă vom construi un mozaic de dimensiuni modificate. Parametrul *num_pieces_horizontal* (setat implicit cu valoarea 100) defineşte numărul de piese pe orizontală ale mozaicului. Pe baza acestui parametru, ale dimensiunilor iniţiale ale imaginii de referinţă şi ale dimensiunilor pieselor (28×40 pixeli) putem determina dimensiunea mozaicului pe care îl vom construi. În construcţia mozaicului vom folosi imaginea de referinţă redimensionată corespunzător. Funcţia *compute_dimensions* trebuie completată de către voi.

1.3 Adăugarea pieselor mozaicului

Piesele sunt adăugate cu scopul de a obţine un mozaic cât mai asemănător cu imaginea de referinţă redimensionată. În implementarea voastră, veţi adăuga piesele prin două modalităţi de aranjare a lor: (1) pe caroiaj, adăugând piesele de la stânga la dreapta şi de sus în jos. Piesele astfel adăugate se îmbină perfect, fără a se suprapune; (2) în mod aleator, adăugând de fiecare dată o piesă într-o poziţie aleasă la întâmplare în mozaic. Un asemenea mod aranjare conduce la suprapuneri de piese.

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

- *aleator* - folosind acest criteriu veţi înlocui blocurile de pixeli din imaginea de referinţă redimensionată cu piese alese în mod aleator. În consecinţă, mozaicul pe care îl veţi obţine nu va semăna cu imaginea de referinţă. Acest criteriu este deja implementat în Python însă merge numai pentru imaginile color (pentru imaginile grayscale nu funcţionează).
- *pe baza distanţei euclidiene dintre culorile medii* - folosind acest criteriu veţi înlocui blocurile de pixeli din imaginea de referinţă redimensionată cu piese alese astfel încât culoarea medie a piesei să fie cât mai apropiată (în sensul distanţei euclidiene) de culoarea medie a blocului. Mozaicul din Figura 1 este obţinut folosind acest criteriu. Va trebui să scrieţi codul Python care implementează acest criteriu.

1.4 Predarea temei

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

(a) **(3 puncte)** mozaicurile obţinute 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) şi criteriul distanţei euclidiene dintre culorile medii comentând influenţa lor asupra rezultatelor obţinute.

(b) **(1 punct)** mozaicurile obţinute pentru imaginile din directorul *'.../data/imaginiTest/'* pentru modul de aranjare 'aleator' şi criteriul distanţei euclidiene dintre culorile medii. Pentru rezolvarea acestui punct trebuie să scrieţi funcţia *add_pieces_random*. Pentru imaginea iniţială din Figura 1 ar trebui să obţineţi ceva similar cu imaginile din Figura 3.

(c) **(1 punct)** mozaicurile obţinute pentru imaginile din directorul *'.../data/imaginiTest/'* astfel: implementaţi o modificare a funcţiei *add_pieces_grid* astfel încât mozaicul obţinut să aibă proprietatea că nu există două piese adiacente (stânga, dreapta, jos, sus) identice. Pentru imaginea iniţială din Figura 1 ar trebui să obţineţi ceva similar cu imaginile din Figura 4.

(d) **(1 punct)**: înlocuiţi colecţia furnizată de noi (cu cele 500 de piese ce reprezintă flori) cu o colecţie proprie. O posibilitate ar fi să downloadaţi setul de date $CIFAR-10$ de la adresa `https://www.cs.toronto.edu/~kriz/cifar.html`. $CIFAR-10$ conţine 60000 de imagini color de dimensiuni 32×32 pixeli cu obiecte din 10 clase: avion, automobil, etc. Realizaţi mozaicuri tematice, construind mozaicuri pentru imagini conţinând obiecte din aceste clase cu piesele corespunzătoare: realizaţi un mozaic pentru o imagine cu un automobil folosind piese cu automobile. Includeţi în pdf-ul vostru cel puţin 5 exemple de mozaicuri tematice.

(e) **(1 punct)**: modificaţi proiectul vostru astfel încât mozaicurile obţinute să aibă piese hexagonale. Includeţi în pdf-ul vostru mozaicurile obţinute astfel pentru imaginile din directorul *'.../data/imaginiTest/'*. Pentru imaginea iniţială din Figura 1 ar trebui să obţineţi ceva similar cu imaginile din Figura 5.

(f) **(1 punct)**: modificaţi punctul anterior astfel încât mozaicurile obţinute să aibă proprietatea că nu există două piese hexagonale adiacente (sus, jos, stânga sus, stânga jos, dreapta sus, dreapta jos) identice. Includeţi în pdf-ul vostru mozaicurile obţinute astfel pentru imaginile din directorul *'.../data/imaginiTest/'*.

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

Oficiu: 1 punct.

Figura 3: **Modalitatea de aranjare a pieselor în mod aleator.** *(a) imagine de tip mozaic obţinuta pentru imaginea din Figura 1 pentru modalitatea de aranjare a pieselor în mod aleator; (b) zoom al imaginii obţinute.*

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

Trimiteţi cele două fişiere (*tema1_cod.zip* şi *tema1_rezultate.pdf*) la adresa de email a Iulianei Georgescu, **georgescu_lily@yahoo.com**.

Termenul limită de predare a proiectului este duminică, 27 octombrie 2019, 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 obţinuta pentru imaginea din Figura 1 ce are proprietatea că fiecare piesă este hexagonală; (b) zoom al imaginii obţinute.*