

Concepte și Aplicații în Vederea Artificială - Tema 2

Redimensionarea imaginilor cu păstrarea conținutului

Obiectiv

Scopul acestei teme este implementarea și testarea unei versiuni a algoritmului de redimensionare a imaginilor cu păstrarea conținutului (Figura 1) propus de S. Avidan și A. Shamir în articolul "Seam Carving for Content-Aware Image Resizing" (găsiți articolul atașat în materialele pentru proiect).

Funcțiile Python care vă vor ajuta la implementarea proiectului sunt în directorul *cod*; imaginile pe care le veți folosi sunt în directorul *data*.

Pentru a înțelege tema citiți articolul în limba engleză și consultați slide-urile de la curs. **Tema este individuală.** Toate detaliile legate de predarea temei le găsiți la sfârșitul acestui document.

Punct de pornire

Scriptul *run_project.py* vă oferă un punct de pornire în implementarea voastră. În acest script se setează imaginea ce urmează a fi redimensionată și sunt setați parametri folosiți. Acest script apelează funcția *resize_image* care la rândul ei apelează funcții pentru micșorarea în lățime sau înălțime a imaginii, amplificarea conținutului sau eliminarea unui obiect.

1.1 Micșorarea imaginii pe lățime

Implementați mai întâi operația de micșorare pe lățime a unei imagini. Funcția *decrease_width* realizează operația de eliminare a pixelilor de pe drumurile verticale ce conectează prima linie cu ultima linie din imagine. Această funcție apelează următoarele funcții:

- *compute_energy* - funcția trebuie completată de către voi folosind ecuația (1) din articolul atașat;
- *select_path.py* - scriptul este scris parțial, trebuie să-l completați voi;
- *show_path* - funcția este scrisă în întregime;



Figura 1: **Redimensionare imaginilor.** (a) Imaginea inițială; (b) Imaginea inițială redimensionată la o imagine cu 50 de pixeli mai puțini în lățime folosind algoritmul de păstrare al conținutului; (c) Imaginea inițială redimensionată la o imagine cu 50 de pixeli mai puțini în lățime cu algoritmul de redimensionare uzuală (folosind funcția 'resize' din OpenCV) ce scalează tot conținutul din imagine.

- *delete_path* - funcția este scrisă parțial, trebuie să o completați voi.

Drumul optim vertical este drumul de cost minim și se obține prin metoda programării dinamice (vedeți slide-urile de la curs). În script este scris codul pentru o metodă care alege aleator drumuri, fără să țină cont de funcția cost (scriptul *select_path.py*).

Completați pentru început restul funcției *delete_path* și observați cum rulează scriptul *run_project* pentru cazul drumurilor alese aleator. Apoi completați restul scriptului *select_path.py* pentru cazurile 'greedy' și 'programareDinamica' și testați codul vostru pe imaginea 'castel.jpg' furnizată eliminând 50 de pixeli în lățime (Figura 1).

1.2 Micșorarea imaginii pe înălțime

Realizați operația de eliminare a pixelilor de pe drumurile orizontale ce conectează prima coloană cu ultima coloană dintr-o imagine scriind funcția *decrease_height*. Inspirați-vă în scrierea ei din funcția *increase_width*. Testați-vă codul pe imaginea 'praga.jpg' furnizată eliminând 100 de pixeli în înălțime.

1.3 Amplificarea conținutului imaginilor

Secțiunea 4.4 din articol descrie cum puteți amplifica conținutul unei imagini păstrând dimensiunile inițiale ale imaginii. Scrieți funcția *amplify_content* care implementează această operație prin scalarea imaginii inițiale cu un anumit factor și apoi eliminarea de drumuri. Testați algoritmul pe imaginea 'arcTriumf.jpg' furnizată folosind diverși factori la scalarea imaginii inițiale.

1.4 Eliminarea unui obiect din imagine

Secțiunea 4.6 din articol descrie cum puteți elimina un obiect dintr-o imagine marcat de un utilizator. Scrieți funcția *delete_object* care micșorează imaginea cu scopul de a elimina

obiectul delimitat. Folosiți funcția *selectROI(image)* din OpenCV pentru a delimita obiectul. Testați-vă codul pe imaginea 'lac.jpg' furnizată.

1.5 Predarea proiectului

Puneți într-o arhivă cu numele *tema2_cod_nume_prenume.zip* codul vostru Python. Puneți într-un document cu numele *tema2_rezultate_nume_prenume.pdf* următoarele:

- (a) **(1.5 puncte)** rezultatele obținute la punctul (1.1) pentru imaginea 'castel.jpg' ;
- (b) **(1.5 puncte)** rezultatele obținute la punctul (1.2) pentru imaginea 'praga.jpg' ;
- (c) **(1.5 puncte)** rezultatele obținute la punctul (1.3) pentru imaginea 'arcTriumf.jpg' ;
- (d) **(1.5 puncte)** rezultatele obținute la punctul (1.4) pentru imaginea 'lac.jpg'.
- (e) **(2 puncte)** rezultatele obținute de voi pentru alte imagini (cel puțin 5) decât cele furnizate folosind cele 3 metode de selectare a drumurilor: metoda de selecție aleatoare, metoda greedy și metoda programării dinamice. Puteți exemplifica rezultatele obținute pentru oricare din operațiile enumerate mai sus (în secțiunile 1.1 - 1.4). Includeți cel puțin 3 exemple reușite și cel puțin 2 exemple nereușite explicând de fiecare dată de ce algoritmul vostru a reușit sau nu a reușit.

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

Oficiu: 1 punct.

Pentru fiecare exemplu de la punctele (a) - (d) includeți 3 imagini: imaginea inițială, imaginea obținută cu algoritmul implementat și imaginea obținută cu redimensionarea uzuală - folosind funcția *resize* (acolo unde e posibil). La punctul (e) includeți pentru fiecare exemplu 5 imagini: imaginea inițială, imaginea obținută cu algoritmul implementat pentru toate cele 3 metode (aleatoare, greedy, programare dinamică) și imaginea obținută cu redimensionarea uzuală - folosind funcția *resize*.

Trimiteți cele două fișiere (*tema2_cod_nume_prenume_nume_prenume.zip* și *tema2_rezultate_nume_prenume.pdf*) la adresa de email a lui Bogdan, bogdan.alex@fmi.unibuc.ro.

Termenul limită de predare a proiectului este marți, 1 decembrie 2020, ora 23:59. Fiecare zi de întârziere în predarea temei se penalizează cu 1 punct în minus. Temele primite după data de 4 decembrie nu vor fi luate în calcul.