

Tema 2 - Mini OCR

- **Responsabili:** Anca Morar, Alexandru Grădinaru
- **Lansare:** 17 decembrie 2021
- **Termen de predare:** 23 ianuarie 2022, ora 23:55
- **Regulament:** https://ocw.cs.pub.ro/courses/spg/regulament_general
[https://ocw.cs.pub.ro/courses/spg/regulament_general]
- **Notă:** Orice informație ce nu a fost acoperită în acest document este la latitudinea voastră!

În cadrul acestei teme va trebui să implementați un OCR simplu.

Descrierea aplicației

Aplicația va permite încărcarea unei imagini de intrare, identificarea literelor și a cuvintelor individuale și afișarea textului din acea imagine. Se va pune la dispoziție de asemenea o imagine șablon cu literele alfabetului și un fișier text care realizează corespondența dintre caractere și literele din imaginea șablon. Imaginea de intrare este următoarea (aceasta trebuie descărcată la rezoluția originală, 1920×1080):



Imaginea șablon cu literele alfabetului este următoarea:



Fișierul text care realizează corespondența între caractere și literele din șablon conține pe fiecare linie următoarele elemente:

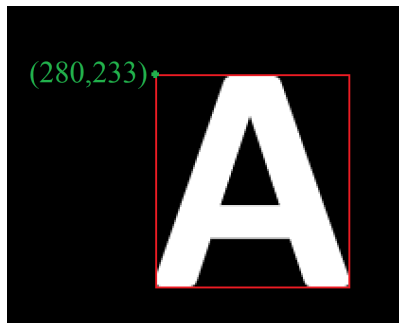
caracterul x y

x și y reprezintă adresa pixelului colțului stânga-sus al dreptunghiului încadrator al caracterului, în imaginea șablon. Fișierul text poate fi descărcat de aici: [sablon.txt](#)

De exemplu, dacă pe prima linie în fișier avem următoarele elemente:

A 280 233

Înseamnă că în imaginea șablon, la rezoluția originală (1920 x 1080), imaginea cu caracterul "A" se găsește într-un dreptunghi încadrator care are colțul din stânga-sus la pixelul (280,233):



Textul din imaginea de intrare se va afișa în consolă: MERRY CHRISTMAS AND A HAPPY NEW YEAR !

Pași

Pașii algoritmului de identificare a textului sunt următorii:

- **Aplicarea unui filtru median pentru eliminarea zgomotului din imagine**

- Filtrul median constă în ordonarea intensităților pixelilor dintr-o vecinătate a pixelului curent și selectarea intensității de pe poziția din mijloc (de exemplu, dacă alegem o vecinătate 3×3 , cu 9 pixeli, se va alege intensitatea de pe poziția a 5-a în vectorul ordonat de intensități).
- Filtrul median va fi aplicat pe fiecare canal (R,G,B), independent
- Rezultatul filtrului median aplicat pe imaginea de intrare este următorul:



- **Conversia pixelilor imaginii filtrate din spațiul RGB în spațiul HSV (Hue, Saturation, Value)**

- **Segmentarea imaginii utilizând nuanța (Hue) pe post de criteriu de similaritate**

- Puteti folosi orice metodă de segmentare a imaginii, dar sunteți încurajați să aplicați un algoritm simplu de segmentare prin extindere (region growing):
 1. Se consideră că toți pixelii imaginii sunt nevizitați
 2. Se parcurg (și se bifează ca fiind vizitați) de sus în jos și de la stânga la dreapta pixelii imaginii. Dacă în parcurgerea scan-line se identifică un pixel de foreground (care are nuanța verde) nevizitat, atunci acesta primește un ID neutilizat până atunci, și este inserat într-o coadă, el reprezentând un pixel germen. Urmează o parcurgere în lățime (BFS), în care se procesează elementele din coadă (cât timp aceasta conține elemente). Procesarea unui element din coadă constă din următorii pași:
 - a. se identifică toți pixelii vecini nevizitați, care au o nuanță similară, și se introduc în coadă
 - b. pixelii vecini se marchează ca fiind vizitați și primesc același ID cu pixelul germen (acesta reprezentând chiar ID-ul regiunii)
 - c. se scoate din coadă elementul care a fost procesat
- Parcurgerea în lățime conduce la identificarea unei regiuni de foreground, a unei litere individuale din imaginea de intrare. Algoritmul de mai sus conduce la identificarea și

etichetarea cu ID-uri diferite a tuturor regiunilor individuale din imaginea de intrare. În figura de mai jos se poate observa rezultatul segmentării:



MERRY CHRISTMAS
AND
A HAPPY NEW YEAR !

- **Identificarea dreptunghiurilor încadratoare pentru regiunile de foreground individuale și afișarea acestora**
 - Se calculează minimul și maximul pe x și pe y ale adreselor pixelilor dintr-o regiune
 - Se afișează dreptunghiurile încadratoare pe imaginea segmentată. În imaginea de mai jos se poate observa rezultatul acestui pas (dreptunghiurile încadratoare sunt desenate cu linii roșii subțiri):



MERRY CHRISTMAS
AND
A HAPPY NEW YEAR !

* **Identificarea cuvintelor**

- Se va identifica fiecare caracter. Se va procesa fiecare regiune individuală, R , astfel:
 - Se va identifica lungimea și lățimea dreptunghiului încadrator al regiunii, D
 - Pixelii din dreptunghiul încadrator se vor compara cu pixelii din imaginea șablon. Pentru fiecare linie din fișierul text:
 1. Se va determina o fereastră în imaginea șablon, F , de rezoluția dreptunghiului încadrator, care va avea colțul din stânga-sus egal cu adresa menționată pe linia din fișierul text
 2. Se va compara fiecare pixel din dreptunghiul încadrator, D , cu pixelul corespunzător din fereastra F (la aceeași adresă (x,y) , relativ la colțul din stânga-sus al dreptunghiului / ferestrei)
 3. Se va calcula un cost de potrivire (de exemplu, numărul de pixeli corespunzători care au aceeași valoare, relativ la dimensiunea dreptunghiului încadrator)
 - Linia din fișierul text care corespunde la fereastra din imaginea șablon cu costul de potrivire cel mai bun reprezintă linia caracterului regiunii R
- Se va identifica fiecare cuvânt. Pentru aceasta puteți stabili niște reguli simple de identificare a spațiilor în text. Exemplu:
 - Dacă între două regiuni individuale de pe același rând există un spațiu mai mare decât un prag dat (de exemplu, 50 de pixeli), se consideră că între acele două litere există un spațiu (o literă corespunde unui cuvânt iar cealaltă literă, următorului cuvânt din text)
 - Dacă o regiune individuală începe pe un nou rând (nu mai există altă regiune în stânga ei), atunci această regiune face parte dintr-un cuvânt nou

* **Afișarea mesajului în consolă**

- Se vor afișa cuvintele din imaginea de intrare, conform corespondenței între caracterele din fișierul text, literele din șablon, și potrivirea dintre regiunile individuale din imaginea de intrare și literele șablonului.

Barem (din 100 puncte)

- Filtrul median: 15 puncte
- conversia RGB - HSV: 15 puncte
- segmentarea (unde fiecare regiune are un ID diferit): 30 puncte
- determinarea dreptunghiurilor încadratoare regiunilor (literelor): 10 puncte
- identificarea caracterelor corespondente din fișierul text și a spațiilor din text: 30 puncte

Bonusuri Posibile

- Funcționarea pentru litere cu dimensiunea fontului variabilă
- Funcționarea pentru mai multe fonturi
- Funcționarea pentru text rotit

Indicații Suplimentare

Tema se poate implementa în orice limbaj sau framework. Dacă folosiți biblioteci ajutătoare (ex, OpenCV), acestea pot fi utilizate numai pentru încărcarea/afișarea imaginii sau pentru operații de citire/scriere pe pixelii imaginii. Nu se pot folosi funcții deja implementate în biblioteci pentru:

- implementarea filtrului median
- trecerea din spațiul RGB în spațiul HSV
- segmentarea imaginii și identificarea regiunilor de foreground (a literelor individuale)
- determinarea dreptunghiurilor încadratoare regiunilor (literelor)
- compararea pixelilor din dreptunghiurile încadratoare din imaginea de intrare cu informația din șablonul cu alfabetul

Arhivarea proiectului

Arhiva trebuie să conțină toate resursele necesare compilării și rularii. Dacă folosiți framework-ul de la laborator, realizați următorii pași:

- înainte de a face arhiva asigurați-vă că ați curățat proiectul Visual Studio:
 - click dreapta pe proiect în **Solution Explorer** → **Clean Solution**
 - și stergeți folderul **/build/.vs** (dacă nu îl vedeți, **este posibil să fie ascuns**)
- SAU stergeți complet folderul **/build**
- în cazul în care arhiva tot depășește limita de 50MB (nu ar trebui), puteți să stergeți și folderul **/deps** sau **/assets** întrucât se pot adăuga la testare. Nu este recomandat să faceți acest lucru întrucât îngreunează mult testarea în cazul în care versiunea curentă a librăriilor/resurselor diferă de versiunea utilizată la momentul scrierii temei.

spg/teme/2021/02.txt · Last modified: 2021/12/17 16:17 by anca.morar