

Generare arbori de funcții pentru evaluarea de binarizări optime

1st Brinza Maria- Cristina
Software Developer

2nd Blaciotti Mihai
Tester

3rd Dranca Ștefana-Ioana
Team Leader

4th Dumitrescu Ioana-Teodora
Project Manager

5th Lovin Cosmin-Viorel
Software Developer

Abstract—Documentul de față abordează problema găsirii unor praguri optime pentru binarizarea globală, respectiv locală a imaginilor, pornind de la pragurile calculate de mai mulți algoritmi consacrați. Implementarea analizată implică un algoritm Monte Carlo.

Index Terms—binarizare, locală, globală, algoritm Monte Carlo

I. INTRODUCTION

Binarizarea imaginilor este procesul de conversie a unei imagini în tonuri de gri sau a unei imagini color într-o imagine binară. O imagine în tonuri de gri are un singur canal, care reprezintă intensitatea fiecărui pixel. O imagine color, pe de altă parte, are mai multe canale, cum ar fi roșu, verde și albastru. Pentru a converti o imagine color într-o imagine în tonuri de gri, imaginii i se aplică o conversie a spațiului de culoare.

Procesul de thresholding al unei imagini este utilizat pentru a separa imaginea în două regiuni, care sunt prim-planul și fundalul. Acest lucru se realizează prin definirea unei valori de prag, care este utilizată pentru a separa imaginea în două regiuni. Toți pixelii din imagine cu o valoare de intensitate mai mare decât pragul sunt setați la unu (alb), iar toți pixelii cu o valoare de intensitate mai mică decât pragul sunt setați la zero (negru). Se creează astfel o imagine binară în care singurele două valori posibile sunt alb sau negru.

Această tehnică de binarizare a imaginii este utilă în multe sarcini de eviziune computerizată, cum ar fi segmentarea imaginilor, recunoașterea obiectelor și recunoașterea textului, deoarece simplifică imaginea și face ca obiectul de interes să iasă în evidență.

II. APLICAȚII ȘI STUDII PRECEDENTE

O aplicație importantă a binarizării în acest domeniu este digitalizarea manuscriselor vechi, care conțin cunoștințe valoroase despre culturi și evenimente din trecut. În cazul acesta, o cauză foarte des întâlnită a distorsiunilor este degradarea materialelor în timp. Alt caz în care aceasta aduce valoare este radioterapia [3], unde s-a folosit metoda pentru detectarea creșterilor osoase în imaginile portal. De asemenea, codurile QR au devenit o metoda foarte rapida si de actualitate pentru transmiterea informatiei in cazul unui public numeros sau in locuri unnde nu exista interactiune umana intre consumator

si informator. Si aici, binarizarea imaginilor joaca un rol important, cu solutii bazate pe algoritmi Monte Carlo [4].

III. DESCRIERE SOLUȚIE

Există mulți algoritmi consacrați pentru determinarea pragurilor de binarizare, astfel că proiectul de față are ca scop generarea funcției pentru evaluarea binarizărilor optime, utilizând rezultatele obținute în urma rulării mai multor algoritmi clasici.

Combinând ieșirile acestor algoritmi, proiectul își propune generarea unui rezultat și mai bun decât rezultatele individuale ale acestor algoritmi.

Soluția propusă are la bază utilizarea unui algoritm de tip Monte Carlo care generează în mod aleator arbori de operații între pragurile algoritmilor dați, pentru ca în final să extragem cei mai buni doi arbori (pentru binarizarea globală, respectiv locală). Algoritmul este folosit atât pentru binarizare globală, în cadrul căreia se aplică un singur prag pentru întreaga imagine, dar și pentru binarizare locală care presupune alegerea unui prag pentru fiecare pixel, pe baza caracteristicilor din apropierea sa.

IV. ARHITECTURĂ

Pentru binarizarea globală, programul iterează prin fișierele CSV din care extrage pentru fiecare imagine threshold-urile obținute de algoritmii dați, pragul considerat optim și 256 de valori F-means normate în intervalul 0 - 1. Apoi, se aplică în mod repetat următorii pași: se alege un număr aleator de valori din fiecare vector cu pragurile date, pentru care se aplică câte o funcție matematică aleatoare, se adaugă în vectorul cu praguri rezultatul funcției și se adaugă scorul calculat pe baza valorii F-means la cel al iterației curente.

Pentru binarizarea locală, programul iterează prin fișierele CSV din care se extrag, pentru fiecare pixel, pragurile obținute de algoritmi, valoarea și clasa sa în imaginea de ground truth (0 sau 1). Pentru a manipula cu ușurință datele am implementat o clasă Pixel, iar fiecare reader salvează într-o listă toți pixelii săi. Apoi, se aplică în mod repetat următorii pași: pentru fiecare reader se inițializează cu 0 variabilele în care se contorizează numărul de pixeli true positive, false positive, true negative și false negative; se alege un număr aleator de valori din fiecare listă cu pragurile date pentru fiecare pixel, pentru care se aplică câte o funcție matematică

aleatoare, se adaugă în lista cu praguri rezultatul funcției; se încadrează cu ajutorul acestui nou prag pixelul curent într-una dintre categoriile menționate; după parcurgerea tuturor pixelilor dintr-un reader se calculează valoarea F-means pentru iterația curentă (asociată cu un arbore) și se adaugă la scorul total. Funcțiile care pot fi aplicate în algoritmul nostru (atunci când se alege una aleatoare) sunt următoarele: media aritmetică, media armonică, media geometrică, media pătratică, minimul, maximul și mediana. Pentru binarizarea globală, s-a elaborat o funcție în C++ pentru traducerea soluției într-un format comun pentru toate echipele de dezvoltare. Se citesc valorile din input-ul global direct într-o clasă, apoi se citesc datele arborelui care va genera cel mai mare scor din script-ul de python. Ulterior, se iterează prin valorile de input global și se aplică pașii necesari pentru a ajunge la rădăcina arborelui, printându-se scorul obținut.

V. REZULTATE

În urma rulării programului, am reușit să oscilăm în jurul unui scor de aproximativ 72% pentru binarizarea globală, numărul de iterații fiind de 1000. Inițial ne propusesem atingerea unui scor peste 90%, însă dat fiind volumul foarte mare de date și comportamentul iterațiilor (scorurile parțiale obținute din fișierele de log) am ajuns la concluzia că va trebui să continuăm cu acuratețea obținută. Pentru a detalia, am observat că se creează clustere de valori ce se află în intervalul 60-70. În ceea ce privește binarizarea locală, scorul obținut este de 76%, rezultat obținut prin numai 100 de iterații.

VI. CONCLUZII

Binarizarea poate fi folosită pentru a reduce o imagine la un format mai simplu, ușurând astfel procesări și analize ulterioare ale acesteia. Totuși, o problemă a folosirii binarizării este data de faptul că aceasta nu va funcționa conform așteptărilor pentru orice tip de imagine. Dacă aceasta are un background complex sau condiții variate de lumină ar trebui folosite tehnici mai avansate. Pentru a obține un arbore bun cu ajutorul rezultatelor algoritmilor aplicați în contextul binarizării locale trebuie efectuate mai multe operații, ceea ce face procesul mai intensiv computațional. Însă am reușit să obținem un scor mai mare în cadrul binarizării locale decât în cadrul binarizării globale, chiar și cu mai puține iterații.

REFERENCES

- [1] P. Lech and K. Okarma, "Optimization of the Fast Image Binarization Method Based on the Monte Carlo Approach," ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 20, NO. 4, 2014.
- [2] Hubert Michalak and Krzysztof Okarma, "Optimization of Degraded Document Image Binarization Method Based on Background Estimation"
- [3] Sund, Torbjørn, and Karsten Eilertsen. "An algorithm for fast adaptive image binarization with applications in radiotherapy imaging." IEEE Transactions on Medical Imaging 22.1 (2003): 22-28.
- [4] Okarma, Krzysztof, and Piotr Lech. "Fast statistical image binarization of colour images for the recognition of the QR codes." Elektronika Ir Elektrotechnika 21.3 (2015): 58-61.