Generare de arbori de binarizare optimi

1st Brutaru Bogdan
Software Developer

2nd Capragiu David
Project Manager

3rd Ioniță Ștefăniță
Team Leader

4th Măciucă Alexandru
Tester

5th Niță Maria
Software Developer

6th Velea Florian
Software Developer

***Abstract*— Acest document abordează problema de a determina pragurile optime pentru binarizarea globală și locală a imaginilor. Se bazează pe pragurile calculate de mai mulți algoritmi consacrați. Un algoritm Monte Carlo este folosit în implementarea examinată.**

Keywords—binarizare, algoritmi, Monte Carlo

I. Introducere

Procesul de a transforma o imagine într-o imagine binară care are tonuri de gri sau o imagine color este cunoscut sub numele de binarizare an imaginilor. O imagine gri are un canal care măsoară intensitatea fiecărui pixel. În schimb, o imagine color are mai multe canale: roșu, verde și albastru. O conversie a spațiului de culoare este folosită pentru a transforma o imagine color într-un ton gri.

Thresholding-ul unei imagini împarte imaginea în două zone distincte: prim-planul și fundalul. Acest proces este realizat prin stabilirea unei valori de prag, care împarte imaginea în două zone distincte. Se pune unu, sau alb, la fiecare pixel din imagine cu o intensitate mai mare decât pragul, iar zero, sau negru, la fiecare pixel cu o intensitate mai mică decât pragul. În consecință, se creează o imagine binară cu doar două valori posibile: alb sau negru.

Multe sarcini de tip “computer vision”, cum ar fi segmentarea imaginilor, recunoașterea obiectelor și recunoașterea textului, se bazează pe această metodă de binarizare a imaginii, deoarece simplifică imaginea și face obiectul de interes evident.

II. Aplicații

Digitalizarea manuscriselor vechi, care conțin cunoștințe importante despre culturi și evenimente din trecut, este o aplicație semnificativă a binarizării în acest domeniu. În acest caz, deteriorarea progresivă a materialului este o cauză frecventă a distorsiunilor. Acest lucru este util și în alte cazuri de radioterapie, unde s-a folosit tehnica pentru an identifica crestele osoase în imaginile portal. De asemenea, codurile QR sunt o modalitate modernă de a transmite informații într-un public mare sau în locuri în care nu există interacțiune umană între consumator și informator.

III. Descriere soluție

Întrucât există o mulțime de algoritmi bine cunoscuți pentru a determina pragurile de binarizare, scopul acestui proiect este de a crea o funcție care să permită evaluarea binarizărilor optime, folosind rezultatele obținute prin utilizarea mai multor algoritmi clasici. Se generează în mod aleator operațiile care urmează să fie aplicate pe imagine pentru a obține un rezultat mai performant decât dacă am fi utilizat tehnici clasice abordate în binarizarea imaginilor. În urma rulării soluției noastre, vom avea generați cei mai buni arbori de binarizare, numărul acestora fiind ales arbitrar. Acestora, le vom analiza f measure-urile obținute împreună cu operațiile aplicate. După acumularea acestor informații, putem aplica operațiile pe imaginile corespunzătoare.

IV. Binarizare globală – ARHITECTURĂ

Am creat inițial niște matrici pentru a stoca datele necesare din fișierele CSV oferite după cum urmează”:

- În “GlobalTrain” fiecare linie este o imagine și sunt definite threshold-uri normate în intervalul [0, 1] pentru fiecare algoritm dat
- În “LUTTrain” au fost extrase valorile de tip f_means care sunt în intervalul [0, 100]

Mai departe, se aplică următoarea secvență de pași:

- Pentru fiecare imagine se generează un număr aleator de indecși care sunt folosiți pentru a extrage thresholdurile din fisiere. Aceasta se face cu ajutorul modului random
- Se grupează frunzele două câte două și se aplică o funcție random pe acestea. În cazul în care numărul de frunze este impar, o frunză va urca în arbore mai departe cu no_op.
- Se repetă pasul precedent până ajungem la o singură frunză în lista noastră de lucru.
- Pe baza rezultatului mai exact a frunzei precedent menționate, calculăm indexul în fișierul ”LUTTrain”.
- Adăugăm informații despre numărul iterației, numărul imaginii, threshold-urile folosite, f measure obținut și funcția aplicată într-un fișier json
- În final se calculează media de f measures a celor mai buni arbori aleși de către algoritm
- Am scris cel mai bun arbore într-un fișier JSON pentru a fi folosit mai departe în etapa locală.

V. Binarizare locală – Arhitectură

- Spre deosebire de etapa globală, acum fiecare fișier este o imagine, iar fiecare rând din fișier este un pixel, prima coloană fiind valoarea thresholdului inițial și a doua coloană este clasa pixelului, iar în continuarea acestora avem thresholduri aplicate de diferiți algoritmi.
- Am parsat arborele generat de la etapa globală, mai exact am obținut indecșii pe care îi vom aplica în continuare pe thresholdurile fiecărui pixel. Am preluat și funcția aplicată pe perechi de indecși.

- Am preluat indecșii din arbore și i-am aranjat corespondent cu thresholdurile pixelului, astfel să putem păstra aplicarea funcțiilor pe aceleași perechi de frunze. După aplicarea fiecărei funcție pe perechile de tresholduri, ajungem la o valoare finală, valoare ce trebuie comparată cu threshold-ul inițial al pixelului.
- În funcție de rezultatul comparării, acesta se compară cu valoarea imaginii ground truth și în funcție de tipul de valoare (true positive, false positive, true negative, false negative) incrementăm acea valoare.
- La sfârșitul aplicării algoritmului pe fișierul nostru, calculăm f_measure în funcție de formulă (f_measure = TP/(TP + 0.5*(FP + FN)). Se aplică această logică pentru fiecare fișier/imagine.
- La sfârșitul aplicației calculăm un f_measure average.

VI. Rezultate Finale

Algoritmul prezentat la binarizarea globală iterează prin toate imaginile și își alege la întâmplare un număr de frunze inițial. În continuare frunzele se grupează două câte două și se alege o funcție la întâmplare, astfel rezultând f_measure-ul. În urma rulării algoritmului obținem persistent un f_measure average de 95.

Algoritmul prezentat la binarizarea locală folosește operațiile celui mai bun arbore de la binarizarea globală. Astfel, iterând prin toate imaginile, obținem și scoruri de 0.99, dar și scoruri de 0.0. În final avem un f_measure average de 0.179.

VII. Concluzii

O imagine poate fi redusă la un format mai simplu prin utilizarea binarizării, ceea ce face procesarea acesteia mai ușoară. O problemă semnificativă atunci când este folosită este faptul că nu va funcționa conform așteptărilor noastre pentru orice tip de imagine. Dacă există diverse elemente complexe în imagine sau condiții de iluminare dificile, pentru an obține rezultate cu acuratețe mai mare va trebui să fie folosite tehnici mai avansate.

References

[1]: “An algorithm for fast adaptive image binarization, with applications in radiotherapy imaging”. Torbjørn Sund and Karsten Eilertsen. IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 22, NO. 1, JAN 2003

[2]: “Adaptive Binarization of QR Code Images for Fast Automatic Sorting in Warehouse Systems”. Rongjun Chen Yongxing Yu , Xiansheng Xu , Leijun Wang , Huimin Zhao and Hong-Zhou Tan.