

Generare Arbori pentru Evaluarea de Binarizări Optime (GAEBO)

@Managementul Proiectelor Software

Raport dezvoltare Milestone 3 09.01.2023

Autor raport

Șerban Alexandru-George 344C2

Universitatea Politehnica București

Facultatea de Automatică și Calculatoare

Email: alexandru.serban00@stud.acs.upb.ro

ID Teams: Șerban Alexandru-George (110133)

Componența echipei

- + Project Manager / Manager de proiect: Șerban Alexandru-George 344C2
- + Team Leader / Liderul echipei: Marii Hristofor 344C4
 - + Software Developer / Dezvoltator software: Andrei Alin-Ionuț 344C3
 - + Software Developer / Dezvoltator software: Timpuriu Mircea 342C5
 - + Tester: Popa Marian-Elvis 342C5

Legături cu documentația anterioară

Detaliile privind descrierea proiectului, planificare, metodologie folosită de dezvoltare a soluțiilor software (Spiral) și alte detalii tehnice este inclusă în: [MILESTONE1/ RAPORT INITIAL MILESTONE 1 PROIECT GAEBO.pdf](#)

[MILESTONE2/ RAPORT DEZVOLTARE MILESTONE 2 PROIECT GAEBO.pdf](#)


Evoluția actualizată a lucrului efectuat este monitorizată în [TRACKING PROIECT GAEBO.pdf](#), dar și [DIAGRAMA GANTT PROIECT GAEBO.pdf](#).

Evidența tuturor documentelor scrise de Project Manager (Șerban Alexandru-George, 344C2) se găsește în [README](#)

Stadiu implementare și descriere sumară dezvoltare software

S-a finalizat proiectarea și implementarea procedurilor de binarizare globală și locală.

Etapele de dezvoltare programatică sunt de antrenare (train), validare (validation) și testare (test), dependente de prelucrarea de metadata asociate unor imagini în setul de fișiere CSV. Datele arborelui performant se obțin după stagiile de antrenare și validare, fiind salvate în fișiere de output, pentru a extrage informații de performanță și a compara cu efectele aplicării pragurilor clasice pe imagini.



Modul de construire a arborelui este generativ și progresiv (poate fi asemuit unui pipeline). Astfel, introducerea de noduri tip rădăcină (root-uri) este etapizată și dependentă aplicării rezultatului de funcții compuse. În urma etapei de antrenare, se preia ca ieșire a stadiului un arbore parțial de train/sau testare (A_t), fiind prelucrat în stagiul de validare, în urma căruia se obțin în final metadatele structurii arborescente definitive finale (A_v).

Diferența dintre antrenare și validare constă în comparațiile de performanță pentru actualizarea arborelui. Pentru prima, inserarea de noduri rădăcină este condiționată de obținerea unei metrice de performanță mai bună decât structura A_t în care nu s-a adăugat respectivul root, pe când cea din urmă (care preia A_t), nu mai apar în esență noi noduri, A_v obținându-se prin probarea lui A_t cu alți arbori aleatorii.

Evoluție dezvoltare proiect milestone 3

Întâlnirile în care s-au elaborat arhitectura și s-au dezbătut aspecte tehnice de implementare, precum obținerea efectului determinist al funcțiilor compuse utilizate pentru adăugarea root-urilor în arborele performant, au avut loc în cadrul laboratoarelor de MPS.

S-a păstrat o evidență de tip jurnal, reținându-se punctele succinte atinse ce relevă progresul software.

14 Noiembrie 2022

- Parsarea și analiza fișierelor CSV
- Împărțirea setului de date în 3 mulțimi: train, validation, test
(s-a folosit funcția `train_test_split` din biblioteca `sklearn.model_selection`)

21 Noiembrie 2022

- Generare funcții pentru determinare noduri
- Analiză algoritm generare globală arbore



Conceperea arhitecturală a algoritmului din care rezultă arborii globali

Generarea unui arbore:

1. Se alege un număr random de thresholduri folosite într-o listă, număr fix de noduri arbore (8)
2. Se mapează funcția folosită pentru determinarea nodurilor cu parametrii folosiți și valoarea pe care o returnează (pentru un număr fix inițial de noduri)
3. Se generează un număr de noduri initiale, folosind o funcție aleatoare
4. Pentru restul nodurilor, generăm aleator 10 funcții pretendente, din care aplicăm funcția potrivită de determinare, cu valoarea cea mai apropiată de threshold

Alegerea arborelui cel mai bun:

1. Generăm un număr de arbori cu threshold-uri diferite folosite, și un număr de noduri inițiale pe care se aplică o funcție aleatoare
2. La introducerea unui nou nod, se determină funcția potrivită pentru fiecare root al fiecărui arbore; Arborele cu nodul care are cea mai slabă performanță nu va mai fi luat în considerare pe viitor.
3. Arborele final este cel care rămâne după toate eliminările de la pasul 2.

5 Decembrie 2022

- Scriere funcții pentru crearea arborilor
- Algoritm (full) aleatoriu de generare și (pseudo) de adăugare nod în arbore
- Creat și aplicat metrica de evaluare a performanței
- Combinarea etapelor anterioare de train, validation, test
- Formatarea outputului în fișier

12 Decembrie 2022

- Discutarea și planificarea task-ului de binarizare locală
- Clarificări asupra etalonului de definire a pixelilor de tip alb-negru folosit în algoritm



19 Decembrie 2022

- Luarea deciziei de a schimba modul de parsare și de stocare a datelor prin folosirea unui pandas dataframe
- Luarea deciziei de a reduce numărul de arbori pretendenți pentru fiecare etapă la 2, pentru a ne adapta la anumite cerințe temporale

26 Decembrie 2022 - 3 Ianuarie 2023

- Adaptarea algoritmului de obținere a arborilor de binarizare la task-ul curent de binarizare locală, prin introducerea noii metode de calcul a valorii f-measure.

4 Ianuarie 2023 - 6 Ianuarie 2023

- Testarea algoritmului de binarizare locală, conform planului de testare.

Raport tehnic milestone 3

Articolul științific în care se ilustrează introducerea în domeniu, descrierea soluției, rezultate ale soluției software și concluziile, se regăsește în fișierul: [ARTICOL SOLUTIE DE GENERARE ARBORI PENTRU EVALUAREA DE BINARIZĂRI OPTIME.pdf](#)

Planul de testare și rezultate

A fost realizat [Planul de testare pentru proiectului GAEBO](#), structurat pe 12 paragrafe:

- Introducere
- Obiective și sarcini
- Cerințe hardware
- Cerințe mediu testare
- Strategia de testare
- Programul de testare
- Resurse, roluri și responsabilități
- Proceduri de control
- Caracteristici care trebuie testate

- Caracteristici care nu trebuie testate
- Livrabile
- Riscuri și asumări

Acest plan este însoțit de două anexe:

- [Formularul de urmărire a planului de testare, raportare a problemelor și verificare a efectuării corecțiilor;](#)
- [Lista modificărilor soluțiilor software.](#)

Planul de testare a fost structurat asemănător standardului [IEEE 829](#).

Analiza rezultatelor a avut loc pe un sistem PC cu procesor Intel Core i5-11400H, frecvență de 2.7 GHz și cu o memorie principală RAM de 8GB. Pentru arborele generat în cazul binarizării globale s-a obținut un rezultat f-measure mediu de aproximativ 70% la aproape 1s timp de rulare, iar pentru metoda locală o performanță de 78.6%, în jur de 1 minut timp de rulare.

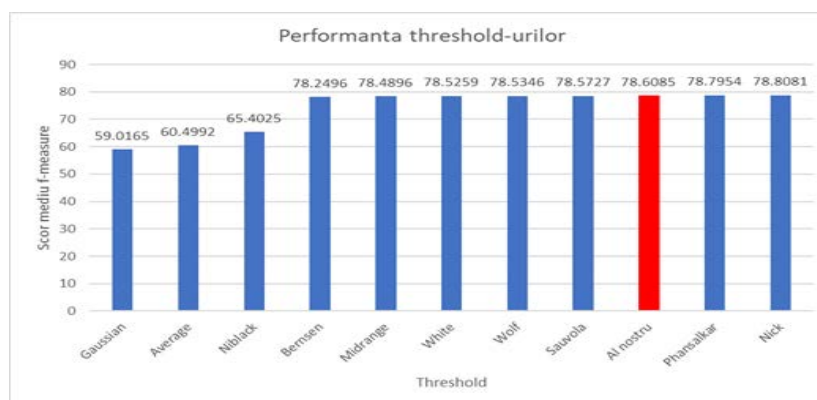



Fig 1. Comparație performanță arbore obținut pentru binarizare locală



Fig. 2. Comparație performanță arbore obținut pentru binarizare globală



Numărul de arbori concurenți pentru etapele de train și validation a fost redus la 2 în scopul adaptabilității temporale, constrânse deja de dimensiunea mare a setului de date oferit. De altfel, se folosesc structuri de tip dataframe ale modulului pandas pentru eficientizarea parsării metadatelor provenite de la intrările de tip CSV.

Impact metodologie software aleasă


Evaluarea metodologiei de dezvoltare software Spiral în cadrul echipei formate ad-hoc pentru soluționarea proiectului GAEB0 la disciplina MPS este una aproximativă, fiind o emulare a situației reale în care se organizează un grup de dezvoltatori sub conducerea și îndrumarea unui management corporatist. Astfel, în cadrul acestei simulări au lipsit mijloace de coerciție (exemplu, sancționarea salarială și/sau demiterea membrilor care nu au îndeplinit funcțiile comun stabilite la începutul proiectului), care ar fi impactat fluxul programatic și reliefat satisfacerea clientului, metrică esențială în determinarea eficienței metodei alese.

Tehnica de dezvoltare Spiral a mitigat majoritar riscurile de scriere a surselor cod (source code), precum potențialele blocaje de programare logică ale dezvoltatorilor software.

Acest fapt s-a datorat prin lucrul modularizat, funcționalitățile de bază (precum parsarea setului CSV, generarea mulțimii de praguri de binarizare frunză pentru arbore, adăugarea unui root, returnarea f-measure-ului pentru o imagine, formatarea output-ului, etc.) fiind bine delimitate, interfețele de lucru fiind izolate pentru software developeri.

Totodată, și împărțirea etapizată, generativă, asemănătoare unui pipeline, asociate stagiile de antrenare, validare respectiv testare, au înlesnit scrierea individuală a programelor din ierarhia de fișiere.

Comunicarea săptămânală din cadrul întâlnirilor de colectare a progresului software iterativ a fost localizată la laboratoarele de MPS, pe 14, 21 Noiembrie, 5 Decembrie 2022. S-au putut explicita modalitățile de combinare ale interfețelor pentru integritatea arhitecturală.




Prin urmare, izolabilitatea conferită la un timp fixat de metodologia Waterfall, aplicat succesiv (spiralat) din săptămână în săptămână, a conferit dezvoltarea prezentată la milestone 2, reușindu-se implementarea binarizării globale.

Determinarea unor membri ai echipei a fost diferită față de a altora în vederea finalizării proiectului, fapt ameliorat de metodologia Spiral, avându-se ca scop eliminarea dependențelor între porțiunile de cod asignate dezvoltatorilor software diferiți și a testerului. S-a ținut cont ca sursele critice să fie implementate de persoane diferite (interfațarea aplicațiilor de binarizare globală și locală folosind ca și soluție Jupyter Notebook de către un programator, generarea de funcții folosite pentru inserarea de noi noduri tip rădăcină pentru construcția arborilor de către altcineva).

Feedback acordat membrilor echipei (puncte negative)

- Team Leader / Liderul echipei: [Marii Hristofor 344C4](#)
 - Colaborare inexistentă cu managerul de proiect, comunicare ineficientă cu dezvoltatorii software și testerul
 - Nu a contribuit în dezvoltarea documentației proiectului la milestone 3 (aspecte care au fost comunicate și de care nu s-a ținut cont, menționate la [feedback milestone 2](#)); Poziția de Team Leader presupunea comunicare a soluției software între Tester și Software Developer, și deci și posibilitatea de aduna date privind elaborarea soluției respectiv rezultate teste (era natural să fi existat o contribuție la [planul de testare al proiectului GAEBO](#), dar care a fost absentă), pe care oricum după Project Manager-ul le-ar fi adaptat pentru a fi lesne de înțeles clientului
- Software Developer / Dezvoltator software: [Andrei Alin-Ionuț 344C3](#)
 - Același aspect care a fost amintit la feedback-ul de la milestone 2
 - Contribuție neclară asupra scrierii/eficientizării de cod la milestone 3 și implicare insuficientă

Mulțumesc prin această cale, o dată cu finalizarea proiectului GAEBO, în mod special lui **Timpuriu Mircea (342 C5)** și **Popa Marian-Elvis (342 C5)** pentru contribuțiile decisive în scrierea surselor cod, analiza pertinentă a rezultatelor testării și comunicarea eficientă avută cu mine. Totodată, lui **Timpuriu Mircea (342 C5)** pentru ideile creative în elaborarea



arhitecturii software atât la binarizarea globală, cât și la cea locală, fapt evidențiat de rezultatele deosebite obținute: performanță de aproape 70% pentru arborele global generat (mai mare decât cea obținută pornind de la toate celelalte threshold-uri) și f-measure mediu de aproape 79% pentru arborele local obținut.