

Plan de testare proiect GAEBO

Întocmit, manager de proiect GAEBO: **ȘERBAN Alexandru-George 344C2**

1. INTRODUCERE

Prezentul plan de testare se referă la Proiectul de generare arbori pentru evaluarea de binarizări optime (GAEBO) și are în vedere cele două soluții software rezultate, bazate pe conceptele de binarizare globală, respectiv locală, astfel încât prin combinarea lor să se obțină un rezultat optim. Prezentul plan de testare urmărește standardul [IEEE 829](#).

2. OBIECTIVE ȘI SARCINI

2.1 Obiective

Obiectivele urmărite prin planul de testare al proiectului GAEBO sunt:

2.1.1. Testarea soluției software de binarizarea globală prin care se generează un arbore și care returnează pragul de binarizare unic, optim pentru imaginea respectivă, cu următoarele sub-obiective:

- a) Verificarea corectitudinii extragerii datelor din fisierele CSV;
- b) Asigurarea diversității funcțiilor generate random;
- c) Verificarea corectitudinii output-ului generat de funcții pe exemple precalculate;
- d) Împărțirea threshold-urilor către funcții și validarea numărului total de threshold-uri: `generate_starting_tree_structure()`;
- e) Verificarea modificării nodului rădăcină al arborelui care corespunde noului nod generat: `generate_new_node_in_tree()`;
Pentru etapa de antrenare:
 - f) Respectarea restricțiilor legate de numărul de arbori care vor fi generați;
 - g) Respectarea restricțiilor legate de numărul de opțiuni de funcții ce vor fi generate;
 - h) Asigurarea trimiterii în etapa de validare a celui mai bun arbore;
Pentru etapa de validare:
 - i) Respectarea restricțiilor legate de numărul de arbori care vor fi generați;
 - j) Respectarea restricțiilor legate de numărul de opțiuni de funcții ce vor fi generate;
 - k) Asigurarea trimiterii în etapa de finală a celui mai bun arbore.

2.1.2. Testarea soluției software de binarizare locală prin care se generează praguri de binarizare diferite pentru diverse ferestre din imagine, cu următoarele sub-obiective:

- a)** Verificarea corectitudinii extragerii datelor din fisierele CSV;
- b)** Asigurarea diversității funcțiilor generate random;
- c)** Verificarea corectitudinii output-ului generat de funcții pe exemple precalculate;
- d)** Împărțirea threshold-urilor către funcții și validarea numărului total de threshold-uri: `generate_starting_tree_structure()`;
- e)** Verificarea modificării nodului rădăcină al arborelui care corespunde noului nod generat: `generate_new_node_in_tree()`;
Pentru etapa de antrenare:
- f)** Respectarea restricțiilor legate de numărul de arbori care vor fi generați;
- g)** Respectarea restricțiilor legate de numărul de opțiuni de funcții ce vor fi generate;
- h)** Asigurarea trimiterii în etapa de validare a celui mai bun arbore;
Pentru etapa de validare:
- i)** Respectarea restricțiilor legate de numărul de arbori care vor fi generați;
- j)** Respectarea restricțiilor legate de numărul de opțiuni de funcții ce vor fi generate;
- k)** Asigurarea trimiterii în etapa de finală a celui mai bun arbore.

2.2 Sarcini

Sarcinile identificate pentru fiecare obiectiv și sub-obiectiv sunt:

- testarea propriu-zisă;
- raportarea problemelor;
- corecția problemelor;
- post-testarea (după corecție).

3. CERINȚE HARDWARE

- 4 Calculatoare PC cu următoarele dotări minime: procesor Intel i5, 2,9GHz sau AMD Ryzen 5, 3,9GHz, 16GB DDR4, 1TB SSD.

4. CERINȚE MEDIU DE TESTARE

- instalare mediu de dezvoltare Python 3 și soluție conexă Jupyter Notebook pe calculatoarele pentru testare și dezvoltare;
- seturi de fișiere în format CSV pentru imagini diverse, fiecare fișier fiind asociat unei imagini din data set și conține ca date pixelii din cadrul imaginii și valorile lor; fiecărui pixel îi este asociat ca intrare în fișier pragurile obținute de algoritmi

5. STRATEGIA DE TESTARE

Se bazează pe aplicarea seturilor de fișiere programelor software implementate în cadrul proiectului și urmărirea rezultatelor prin acuratețea pragurilor de binarizare obținute, timp de rulare.

6. PROGRAMUL DE TESTARE

Conține toate etapele necesare testării obiectivelor și sub-obiectivelor urmărite, precizate în paragraful 2. Normarea de timp propusă pe sub-obiective pentru efectuarea testelor este următoarea:

ID	Activitate	Timp alocat
a	Verificarea corectitudinii extragerii datelor din fisierele CSV	1h
b	Asigurarea diversității funcțiilor generate random	1h
c	Verificarea corectitudinii output-ului generat de funcții pe exemple precalculate	3h
d	Împărțirea threshold-urilor către funcții și validarea numărului total de threshold-uri: <code>generate_starting_tree_structure()</code> ;	3h
e	Verificarea modificării nodului rădăcină al arborelui care corespunde noului nod generat: <code>generate_new_node_in_tree()</code> ;	2h
	Pentru etapa de antrenare	

f	Respectarea restricțiilor legate de numărul de arbori care vor fi generați	1h
g	Respectarea restricțiilor legate de numărul de opțiuni de funcții ce vor fi generate	1h
h	Asigurarea trimerii în etapa de validare a celui mai bun arbore	1h
	Pentru etapa de validare	
i	Respectarea restricțiilor legate de numărul de arbori care vor fi generați	1h
j	Respectarea restricțiilor legate de numărul de opțiuni de funcții ce vor fi generate;	1h
k	Asigurarea trimerii în etapa de finală a celui mai bun arbore	1h

7. RESURSE, ROLURI ȘI RESPONSABILITĂȚI

În proiectul de testare sunt implicați:

- Testerii SW: [Popa Marian-Elvis 342C5](#), [Marii Hristofor 344C4](#)
- Dezvoltatori corectori SW: [Timpuriu Mircea 342C5](#), [Popa Marian-Elvis 342C5](#).
- Supervizare și alocare sarcini: [Șerban Alexandru-George 344C2](#)

8. PROCEDURI DE CONTROL

Va fi utilizat un *Formular de urmărire a planului de testare, raportare a problemelor și verificare a efectuării corecțiilor*. Managerul de proiect va superviza și va alocă resursele umane realizării planului de testare prin acest formular.

[ANEXA 1 - FORMULAR URMĂRIRE PLAN DE TESTARE, RAPORTARE PROBLEME ȘI VERIFICARE CORECȚII](#)

Raportarea problemelor

La întâlnirea unui incident în timpul procesului de testare acesta va fi semnalat corespunzător de către testerul SW în *Formularul de urmărire a planului de testare, raportare a problemelor și verificare a efectuării corecțiilor*.

Managerul de proiect va alocă un dezvoltator corector SW pentru rezolvarea problemei.

După corecție, testerul va reverifica (post-testa) soluționarea problemei.

Tot fluxul de acțiuni se notifică și se urmărește prin formularul de mai sus, în regim online.

Descrierea câmpurilor din *Formular de urmărire a planului de testare, raportare a problemelor și verificare a efectuării corecțiilor*:

Nr. – se precizează numărul de ordine al incidentului;

Nume tester software – numele testerului care a realizat testarea;

Identificator obiectiv / sub-obiectiv testat – numărul de identificare al obiectivului / sub-obiectivului testat;

Corecție necesară (da/nu) – se completează cu nu dacă totul este ok, altfel cu da;

Data semnalării problemei – se completează data identificării problemei;

Descrierea problemei – descrierea problemei, dacă este necesară corecție;

Nume dezvoltator corector SW – numele dezvoltatorului care realizează corecția;

Data finalizării corecției - se completează data corecției problemei;

Validare corecție de către tester (da/nu) - se completează cu da dacă totul este ok, altfel cu nu;

Data validării corecției - se completează data validării corecției problemei;

Alte observații – se completează cu alte observații (dacă sunt).

Urmărirea modificărilor soluțiilor software

Urmărirea modificărilor soluțiilor software se va realiza prin documentul *Lista modificărilor soluțiilor software*.

Descrierea câmpurilor din *Lista modificărilor soluțiilor software*:

Nr. – numărul curent;

Denumire modificare – se completează cu v.1, v.2, ...;

Data modificării – se completează cu data de la care intră în producție;

Nume manager proiect – se completează cu numele managerului care a validat modificarea.

Tot fluxul de acțiuni se notifică și se urmărește prin documentul de mai sus, în regim online.

[ANEXA 2 – LISTA MODIFICĂRI SOLUȚII SOFTWARE](#)

9. CARACTERISTICI CARE TREBUIE TESTATE

Se testează toate obiectivele și sub-obiectivele precizate anterior.

10. CARACTERISTICI CARE NU TREBUIE TESTATE

Nu este cazul.

11. LIVRABILE

Documentele livrabile sunt următoarele:

- Planul de testare
- Formularul de urmărire a planului de testare, raportare a problemelor și verificarea efectuării corecțiilor
- Rapoarte sumare ale testelor (format liber la alegerea testerului SW)
- Lista modificărilor soluțiilor software

12. RISCURI ȘI ASUMĂRI

Riscurile implementării planului de testare identificate sunt:

- Indisponibilitatea membrilor echipei;
- Indisponibilitatea resurselor hardware;

Soluționarea va fi asumată de managerul de proiect și team lider și se va baza pe realocarea resursei umane, respectiv pe alocarea partajată a resurselor hardware (ambele cu riscul nefinalizării la termen a procesului de testare).