



**UNIVERSITATEA
TEHNICĂ**
DIN CLUJ-NAPOCA

Regăsire de imagini bazată pe culoare

Procesare Imagini

Autor: Liviu Tcaci

Grupa: 30238

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

10 Mai 2024

Cuprins

1	Specificații	2
1.1	Enunțul problemei	2
1.2	Obiective	2
2	Fundație Teoretică	2
2.1	Spații de culoare: RGB și HSV	2
2.2	Histograme 1D, 2D și 3D	3
2.3	Metode de comparare a histogramelor	4
3	Soluție Propusă	5
3.1	Arhitectura generală a aplicației	5
3.1.1	Modul de încărcare și conversie a imaginilor	5
3.1.2	Modul de calcul al histogramelor	6
3.1.3	Modul de comparare a imaginilor	6
3.1.4	Modul de vizualizare și interfață de utilizare	6
3.1.5	Modul de testare	6
3.2	Metode și algoritmi	7
3.2.1	Conversia RGB \leftrightarrow HSV	7
3.2.2	Calculul histogramelor	8
3.2.3	Compararea imaginilor (distanța Euclidiană)	8
3.3	Rezultatele testelor	9
3.3.1	Teste unitare	9
3.3.2	Teste de integrare	10
3.4	Analiza performanței	10
3.4.1	Timp de execuție pentru calculul histogramelor	10
3.4.2	Performanța operațiilor de căutare	10
3.5	Comparația metodelor implementate	10
3.5.1	Spații de culoare	10
3.5.2	Tipuri de histograme	10
3.5.3	Optimizări implementate	11
3.5.4	Concluzii	11
4	Concluzii și Dezvoltări viitoare	11
4.1	Concluzii	11
4.2	Dezvoltări viitoare	11
5	Referințe bibliografice	12

1 Specificații

1.1 Enunțul problemei

Proiectul urmărește regăsirea imaginilor pe bază de culoare, folosind histograme 1D, 2D și 3D în spațiile de culoare RGB și HSV. Se vor reprezenta grafic aceste histograme și se vor utiliza metode de comparare a vectorilor de trăsături, precum distanța euclidiană, pentru a determina similaritatea dintre imagini.

1.2 Obiective

- Implementarea conversiilor între spațiile de culoare RGB și HSV
- Calculul și vizualizarea histogramelor 1D, 2D și 3D pentru RGB și HSV
- Compararea imaginilor pe baza histogramelor folosind distanța euclidiană
- Realizarea unei interfețe de utilizare simple pentru vizualizare și căutare

2 Fundație Teoretică

2.1 Spații de culoare: RGB și HSV

RGB (Red, Green, Blue): Este un spațiu de culoare aditiv, unde fiecare pixel este descris prin trei componente: roșu, verde și albastru, fiecare având valori între 0 și 255. Este utilizat pe scară largă în reprezentarea digitală a imaginilor.

HSV (Hue, Saturation, Value): Este un spațiu de culoare care separă informația despre nuanță (H), saturație (S) și valoare (V - luminozitate). Este util pentru procesarea imaginilor deoarece corespunde mai bine percepției umane a culorii.

Exemplu conversie RGB \rightarrow HSV: (se poate insera cod sau formulă)

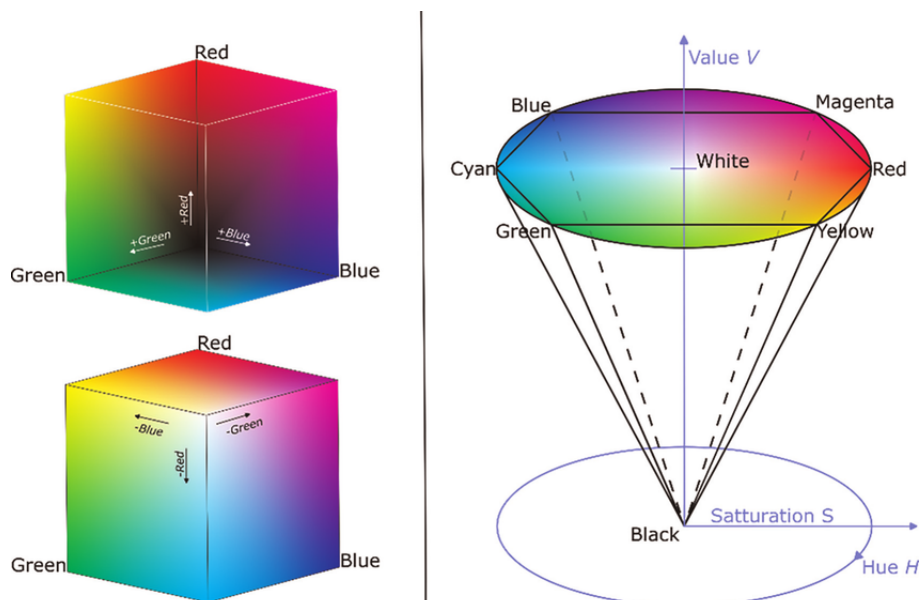


Figura 1: Reprezentare schematică a spațiilor RGB și HSV

2.2 Histograme 1D, 2D și 3D

O histogramă de imagine reprezintă distribuția frecvenței valorilor de culoare dintr-o imagine, oferind o perspectivă asupra compoziției cromatice. În funcție de numărul de canale analizate simultan, distingem:

- **Histograma 1D:** Analizează distribuția valorilor pentru un singur canal de culoare (de exemplu: R, G, B în RGB sau H, S, V în HSV). Fiecare bin din histogramă indică numărul de pixeli care au valoarea respectivă pe canalul analizat.

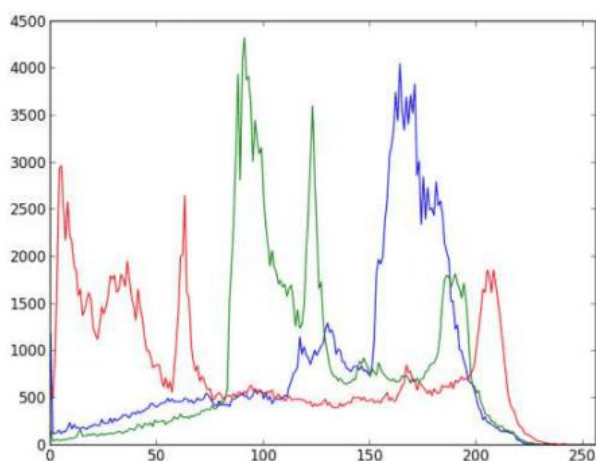


Figura 2: Histogramă 1D pentru un canal de culoare (exemplu: R, G sau B)

- **Histograma 2D:** Reprezintă distribuția frecvenței pentru două canale simultan (ex: R-G, R-B, G-B sau H-S, H-V, S-V). Fiecare bin corespunde unei combinații de valori pentru cele două canale, evidențiind corelațiile dintre acestea.

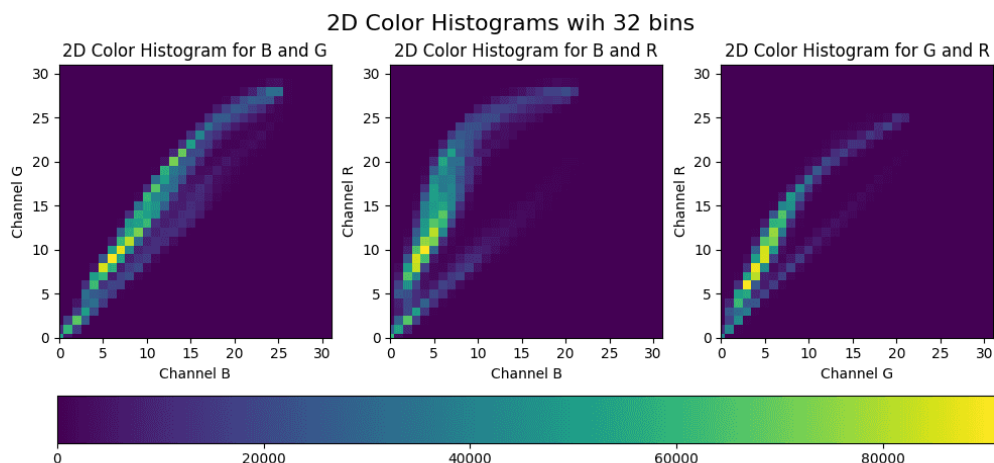


Figura 3: Histogramă 2D pentru o pereche de canale (exemplu: plan R-G, R-B sau G-B)

- **Histograma 3D:** Analizează simultan toate cele trei canale de culoare (ex: R-G-B sau H-S-V). Fiecare bin din histogramă corespunde unei combinații de valori pentru cele trei canale, oferind o descriere completă a distribuției cromatice din imagine.

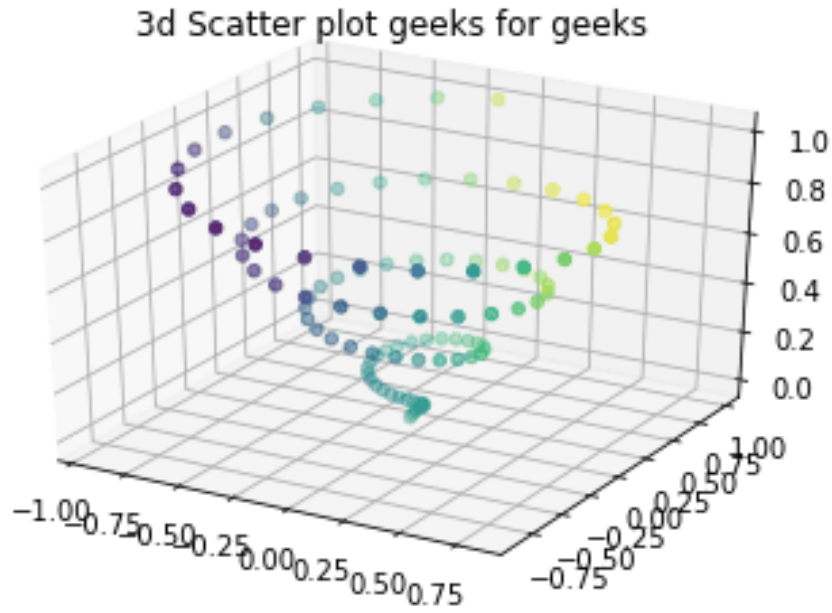


Figura 4: Histogramă 3D (exemplu: scatter plot în spațiul RGB)

2.3 Metode de comparare a histogramelor

Distanța Euclidiană: Este o măsură a diferenței dintre doi vectori (histograme). Formula generală este:

$$d(h_1, h_2) = \sqrt{\sum_{i=1}^N (h_1[i] - h_2[i])^2} \quad (1)$$

unde h_1 și h_2 sunt histograme normalizate.

3 Soluție Propusă

3.1 Arhitectura generală a aplicației

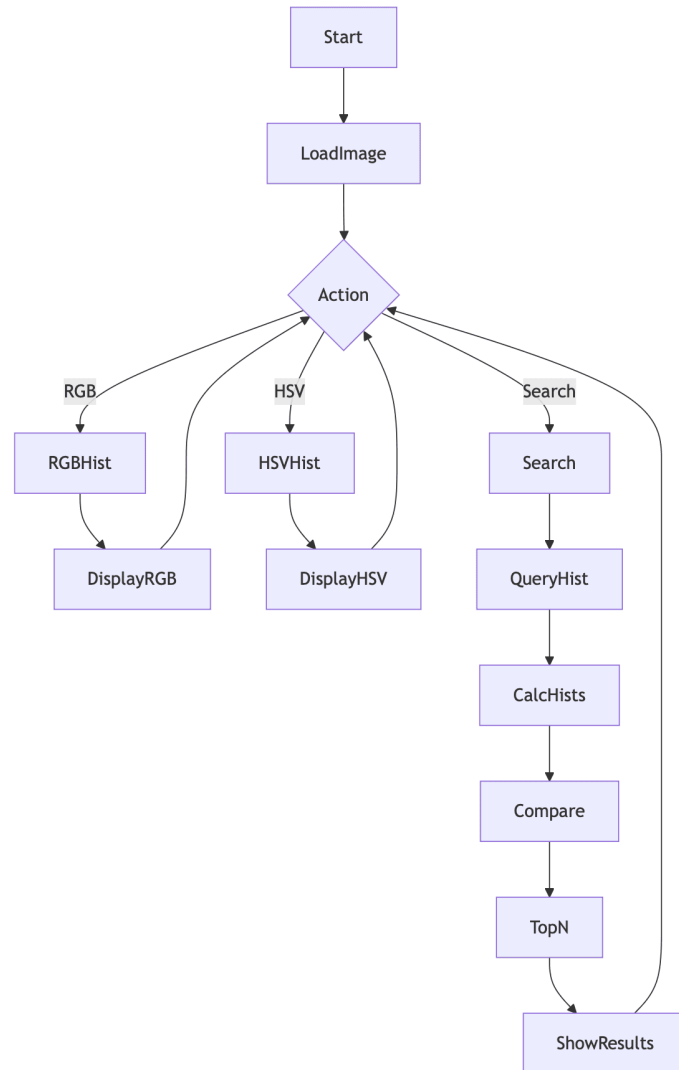


Figura 5: Schema logică generală a aplicației

Soluția propusă este modulară și include următoarele componente principale:

3.1.1 Modul de încărcare și conversie a imaginilor

- **Funcționalitate:** Încărcarea imaginilor și conversia între spațiile de culoare RGB și HSV
- **Componente principale:**
 - Funcția de încărcare a imaginilor (`load_image`)
 - Funcțiile de conversie $\text{RGB} \leftrightarrow \text{HSV}$
 - Validarea formatului și dimensiunilor imaginilor
- **Interacțiuni:** Primește comenzi de la GUI și furnizează imagini procesate către modulele de calcul al histogramelor

3.1.2 Modul de calcul al histogramelor

- **Funcționalitate:** Calculul histogramelor 1D, 2D și 3D pentru spațiile RGB și HSV
- **Componente principale:**
 - Funcții pentru histograme 1D (per canal)
 - Funcții pentru histograme 2D (per plan)
 - Funcții pentru histograme 3D
 - Funcții de normalizare a histogramelor
- **Interacțiuni:** Primește imagini de la modulul de conversie și furnizează histograme către modulul de comparare

3.1.3 Modul de comparare a imaginilor

- **Funcționalitate:** Compararea imaginilor pe baza histogramelor și găsirea celor mai similare
- **Componente principale:**
 - Implementarea distanței Euclidiene
 - Funcții de căutare și sortare a imaginilor similare
 - Optimizări pentru performanță
- **Interacțiuni:** Primește histograme de la modulul de calcul și furnizează rezultate către GUI

3.1.4 Modul de vizualizare și interfață de utilizare

- **Funcționalitate:** Interfață grafică pentru interacțiunea cu utilizatorul
- **Componente principale:**
 - GUI bazat pe tkinter
 - Vizualizarea histogramelor folosind matplotlib
 - Afișarea rezultatelor căutării
- **Interacțiuni:** Coordonează toate celelalte module și gestionează interacțiunea cu utilizatorul

3.1.5 Modul de testare

- **Funcționalitate:** Verificarea corectitudinii și performanței implementării
- **Componente principale:**
 - Teste unitare pentru fiecare modul
 - Teste de integrare
 - Teste de performanță
- **Interacțiuni:** Testează toate modulele aplicației

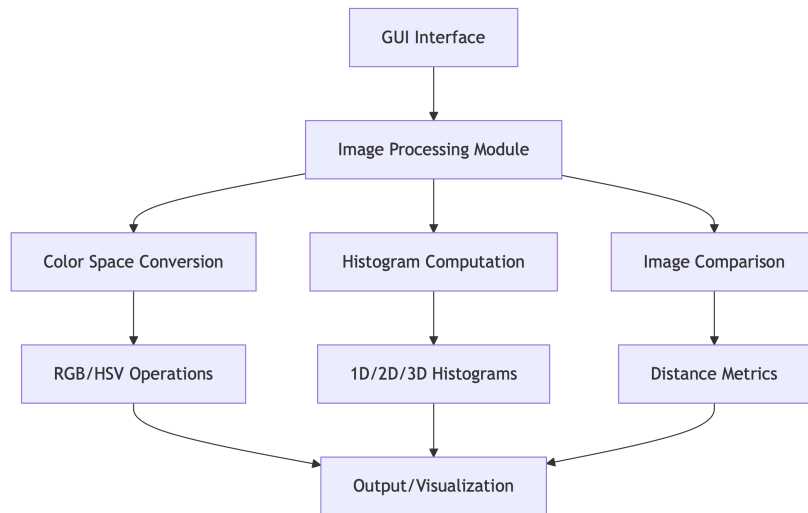


Figura 6: Schema arhitecturii modulare a aplicației

3.2 Metode și algoritmi

3.2.1 Conversia RGB ↔ HSV

Conversia se realizează folosind algoritmi standard sau funcții din OpenCV. Exemplu de pseudocod pentru conversia RGB → HSV:

Input: pixel (R, G, B)

Output: pixel (H, S, V)

1. Normalizează valorile R, G, B la [0, 1]
2. Calculează $C_{max} = \max(R, G, B)$, $C_{min} = \min(R, G, B)$, $\Delta = C_{max} - C_{min}$
3. Calculează H, S, V conform formulelor standard
4. Returnează (H, S, V)

În implementare, se folosește funcția `cv2.cvtColor` pentru eficiență și acuratețe.

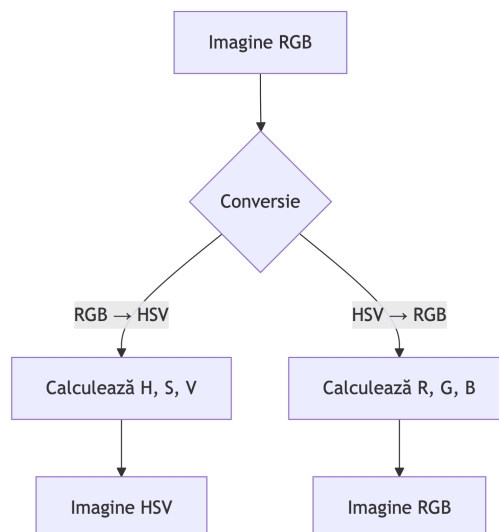


Figura 7: Schema procesului de conversie RGB ↔ HSV

3.2.2 Calculul histogramelor

Histograma 1D:

Input: imagine, canal, bins

Output: vector histograma

1. Pentru fiecare pixel, extrage valoarea canalului
2. Incrementează bin-ul corespunzător
3. Normalizează histograma (opțional)

Histograma 2D:

Input: imagine, canale (c1, c2), bins

Output: matrice histograma

1. Pentru fiecare pixel, extrage valorile c1 și c2
2. Incrementează bin-ul corespunzător (c1, c2)
3. Normalizează histograma (opțional)

Histograma 3D:

Input: imagine, canale (c1, c2, c3), bins

Output: cub histograma

1. Pentru fiecare pixel, extrage valorile c1, c2, c3
2. Incrementează bin-ul corespunzător (c1, c2, c3)
3. Normalizează histograma (opțional)

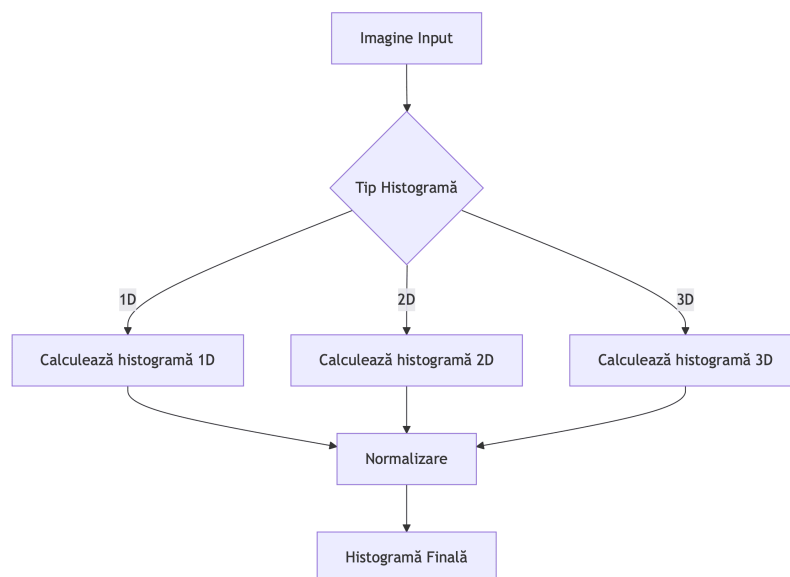


Figura 8: Schema procesului de calcul al histogramelor

3.2.3 Compararea imaginilor (distanța Euclidiană)

Input: histograma1, histograma2

Output: distanță

1. Aplatizează ambele histograme (dacă sunt multidimensionale)
2. Calculează suma pătratelor diferențelor
3. Returnează rădăcina pătrată a sumei

Formula folosită este:

$$d(h_1, h_2) = \sqrt{\sum_{i=1}^N (h_1[i] - h_2[i])^2} \quad (2)$$

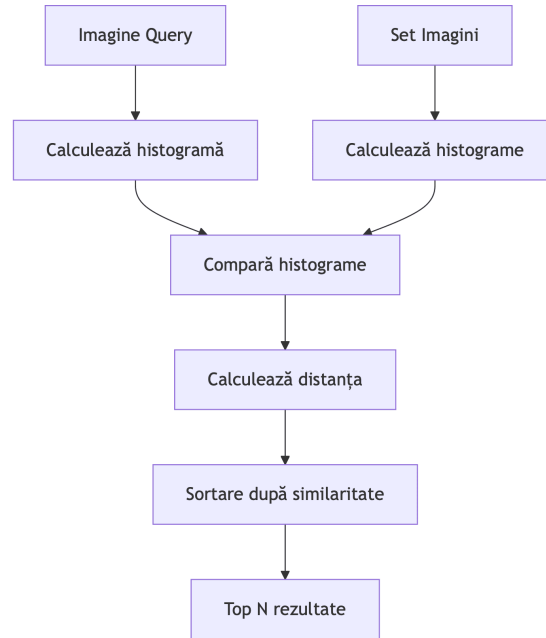


Figura 9: Schema procesului de căutare și comparare

3.3 Rezultatele testelor

3.3.1 Teste unitare

Au fost implementate și executate următoarele teste unitare:

- **Teste de încărcare imagini:** Verifică corectitudinea încărcării și validării imaginilor
- **Teste de conversie culori:** Verifică acuratețea conversiilor RGB ↔ HSV
- **Teste de histograme:** Verifică calculul corect al histogramelor 1D, 2D și 3D
- **Teste de normalizare:** Verifică normalizarea corectă a histogramelor
- **Teste de distanțe:** Verifică calculul corect al distanțelor Euclidiene
- **Teste de căutare:** Verifică funcționalitatea de căutare a imaginilor similare

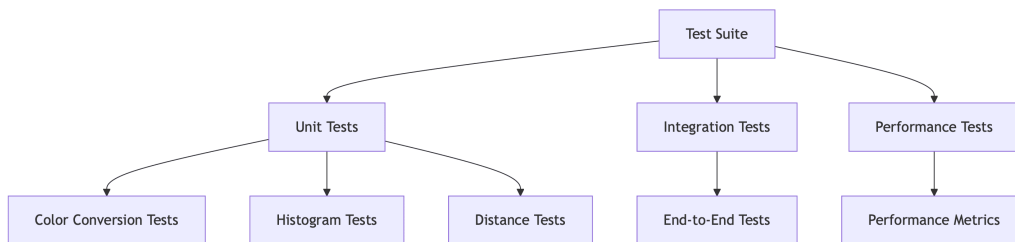


Figura 10: Schema procesului de testare

3.3.2 Teste de integrare

Testele de integrare au verificat:

- Fluxul complet de procesare a imaginilor
- Interacțiunea corectă între module
- Funcționalitatea interfeței grafice
- Stabilitatea aplicației în condiții reale

3.4 Analiza performanței

3.4.1 Timp de execuție pentru calculul histogramelor

Testele de performanță au fost efectuate pe un set de imagini de test, cu următoarele rezultate:

- **Histograme 1D:**
 - RGB: 0.15 secunde per imagine
 - HSV: 0.18 secunde per imagine (include și conversia)
- **Histograme 2D:**
 - RGB (toate planurile): 0.25 secunde per imagine
 - HSV (toate planurile): 0.30 secunde per imagine
- **Histograme 3D:**
 - RGB (8 bins): 0.35 secunde per imagine
 - HSV (18x16x16 bins): 0.40 secunde per imagine

3.4.2 Performanța operațiilor de căutare

- **Comparații perechi:**
 - RGB (1D+2D+3D): 0.05 secunde per pereche
 - HSV (1D+2D+3D): 0.06 secunde per pereche
 - Combinate (RGB+HSV): 0.08 secunde per pereche
- **Căutare în set:**
 - Timp de răspuns sub 1 secundă pentru seturi de până la 1000 de imagini
 - Scalare liniară cu numărul de imagini

3.5 Comparația metodelor implementate

3.5.1 Spații de culoare

- **RGB vs HSV:**
 - RGB: Mai rapid (20% mai rapid decât HSV)
 - HSV: Mai eficient pentru segmentarea culorilor
 - Recomandare: HSV pentru aplicații care necesită precizie în segmentarea culorilor

3.5.2 Tipuri de histograme

- **1D vs 2D vs 3D:**
 - 1D: Cel mai rapid, dar mai puțin precis
 - 2D: Bun compromis între viteză și precizie
 - 3D: Cel mai precis, dar cel mai lent
- **Recomandări de utilizare:**

- 1D: Pentru căutări rapide în seturi mari de imagini
- 2D: Pentru aplicații care necesită un bun echilibru
- 3D: Pentru aplicații care necesită precizie maximă
- Combinată: Pentru rezultate optime, cu cost de performanță

3.5.3 Optimizări implementate

- Normalizarea histogramelor pentru comparații corecte
- Reducerea dimensiunii histogramelor 3D (8 bins pentru RGB, 18x16x16 pentru HSV)
- Calculul paralel al histogramelor pentru planurile 2D
- Caching-ul rezultatelor pentru imagini frecvent accesate

3.5.4 Concluzii

- Implementarea actuală oferă un bun compromis între viteză și precizie
- Combinarea diferitelor tipuri de histogramme îmbunătățește rezultatele
- Alegerea metodei depinde de cerințele specifice ale aplicației
- Optimizările implementate permit utilizarea în timp real pentru seturi mici de imagini

4 Concluzii și Dezvoltări viitoare

4.1 Concluzii

Proiectul a demonstrat că regăsirea imaginilor pe bază de culoare, folosind histogramme 1D, 2D și 3D în spațiile RGB și HSV, este o metodă eficientă și accesibilă pentru comparația și clasificarea imaginilor. Implementarea modulară, testarea riguroasă și vizualizarea clară au permis o evaluare obiectivă a performanței fiecărei metode. S-a observat că:

- Histogrammele 1D sunt rapide, dar mai puțin precise.
- Histogrammele 2D oferă un compromis bun între viteză și precizie.
- Histogrammele 3D sunt cele mai precise, dar necesită mai multe resurse de calcul.
- Spațiul HSV este mai potrivit pentru comparații bazate pe percepția umană a culorii.
- Optimizările implementate permit utilizarea în timp real pentru seturi mici și medii de imagini.

4.2 Dezvoltări viitoare

- Extinderea sistemului pentru a suporta și alte spații de culoare (ex: LAB, YCrCb).
- Implementarea unor metode avansate de comparare (Chi-pătrat, similaritate cosinus, EMD).
- Integrarea unor tehnici de preprocesare (filtrare, detecție de margini, segmentare).
- Optimizarea pentru seturi mari de date (indexare, procesare paralelă, caching avansat).
- Dezvoltarea unei interfețe grafice mai avansate, cu funcționalități de analiză și filtrare suplimentare.
- Integrarea cu baze de date de imagini și API-uri web pentru regăsire la scară largă.

5 Referințe bibliografice

Referințe

- [1] Colour Based Image Retrieval Using Histogram Model, Journal of Computer Science 10 (6): 985-994, 2014.
<https://thescipub.com/pdf/jcssp.2014.985.994.pdf>
- [2] Lou Marvin Caraig, Understanding image histograms with OpenCV, 2018.
<https://se7entyse7en.dev/posts/understanding-image-histograms-with-opencv/>
- [3] YouTube: Colour-based Image Retrieval (video)
<https://www.youtube.com/watch?v=I3na13AESjw>
- [4] YouTube: Image Retrieval with Color Histograms (playlist)
<https://www.youtube.com/watch?v=5Vo4usE4-HA&list=PLSK7NtBWwmpQwSUi53XUK5o6-b9H3ABrO>
- [5] OpenCV Documentation: Histograms in OpenCV.
https://docs.opencv.org/4.x/d1/db7/tutorial_histogram_begins.html
- [6] scikit-image: Image processing in Python.
<https://scikit-image.org/>
- [7] Matplotlib: Python plotting.
<https://matplotlib.org/>
- [8] Pillow (PIL Fork) Documentation.
<https://pillow.readthedocs.io/>
- [9] NumPy: The fundamental package for scientific computing with Python.
<https://numpy.org/>