



PROIECT

Detecția cancerului mamar prin intermediul tumorilor descoperite în mamografii

Studenți: CĂLUȚ Anastasia-Maria

ICHIM Cosmin Daniel

LEONI Cesar Emanuele

Grupa: 443 Bb



CUPRINS

1. INTRODUCERE	4
1.1 Anatomia Sânelui	4
1.2 Cancerul mamar	4
1.3 Tipuri de cancer mamar	4
1.4 Metode de obținere a imaginilor medicale	6
2. STATE OF THE ART.....	7
2.1 Trecerea în revistă a metodelor utilizate pentru detectarea cancerului mamar	7
2.1.1. "Breast Cancer Detection in Mammography Images: A CNN-Based Approach with Feature Selection" [6].....	7
2.1.2. "Mammography with Deep Learning for Breast Cancer Detection" [7].....	8
2.1.3. "Breast Cancer Prediction Using Deep Learning Techniques" [8].....	8
2.1.4. "An Ensemble Approach for Breast Tumor Detection in Mammograms Using Hybrid CNN Models" [9]	8
3. DESCRIEREA METODELOR PROPUSE.....	10
3.1 Metode de optimizare a contrastului.....	10
a) Contrast liniar pe porțiuni	10
b) Contrast putere	10
c) Contrast logaritmic	11
d) Egalizare de histograma	11
3.2 Metode de segmentare a imaginilor	12
a) Binarizare.....	12
b) Slicing	12
c) Prăguire Otsu	12
3.3 Operații morfologice	13



UNIVERSITATEA POLITEHNICA BUCUREȘTI
Facultatea de Electronică, Telecomunicații și Tehnologia informației



a) Dilatare	13
b) Erodare	13
c) Închidere	13
d) Deschidere	14
4. REZULTATE OBȚINUTE	15
4.1 Etapa de optimizare a contrastului	15
4.1.1 Egalizarea de histogramă	16
4.2 Etapa de segmentare a zonei de interes.....	17
4.3 Etapa aplicării operațiilor morfologice	18
4.4 Extragerea de caracteristici.....	18
5. CONCLUZII	21
BIBLIOGRAFIE	22
ANEXA COD	23
Fișierul segmentare.py:.....	27
Fișierul Operatii_morfologice.py:	30
Fișierul extragere_caracteristici.py:.....	31

1. INTRODUCERE

1.1 Anatomia Sânului

Glanda mamară este o glandă exocrină prezentă la oameni și alte mamifere, situată bilateral în zona pectorală. Aceasta are o importanță majoră datorită funcției sale de a secreta lapte, asigurând hrana puilor după naștere. Sâniile femeilor sunt alcătuite din țesut glandular, responsabil de producerea laptelui, și din țesut gras, care influențează mărimea sânilui. Țesutul conjunctiv și ligamentele susțin structura sânilui și îi dau formă, iar nervii îi conferă sensibilitate. Sânul este, de asemenea, străbătut de vase de sânge, vase limfatice și ganglioni limfatici. [1]

1.2 Cancerul mamar

Cancerul de sân este o boală care se dezvoltă în **țesutul mamar** și poate afecta atât femeile, cât și bărbații. Totuși, este mult mai frecventă la femei, în special la cele de peste 50 de ani, deși poate apărea la orice vârstă. [2]

Este încurajator să observăm că ratele de supraviețuire pentru cancerul de sân sunt în creștere, iar numărul deceselor cauzate de această boală este în scădere. Aceasta se datorează, în principal, următorilor factori: [3]

- ❖ *identificarea bolii în stadii timpurii;*
- ❖ *tratamentele personalizate, adaptate stadiului bolii și sănătății fiecărui pacient;*
- ❖ *o înțelegere mai profundă a bolii în prezent.*

1.3 Tipuri de cancer mamar

Printre tipurile de cancer mamar se enumeră: [4]

1. Carcinomul ductal in situ (DCIS)

Este o formă neinvazivă de cancer care începe în ductele galactofore ale sânilui și nu s-a răspândit în țesutul mamar din jur. De obicei, este considerat o formă precoce de cancer de sân și poate fi tratat cu succes.

2. Carcinomul ductal invaziv (IDC)

Acesta este cel mai frecvent tip de cancer de sân și începe în ductele galactofore, dar poate invada și alte părți ale sânelui și se poate răspândi în alte zone ale corpului. IDC reprezintă aproximativ 80% din cazurile de cancer mamar.

3. Carcinomul lobular in situ (LCIS)

LCIS nu este considerat un cancer propriu-zis, ci o modificare precanceroasă. Apare în lobulii glandei mamare (unde se produce laptele) și indică un risc crescut de a dezvolta cancer invaziv în viitor.

4. Carcinomul lobular invaziv (ILC)

Al doilea tip de cancer de sân ca frecvență, ILC începe în lobuli și poate invada țesuturile din jur și alte părți ale corpului. Este mai greu de detectat prin mamografie decât IDC, dar poate fi tratat eficient dacă este depistat la timp.

5. Cancerul de sân inflamator

Este o formă rară și agresivă de cancer de sân, caracterizată prin blocarea vaselor limfatice din pielea sânelui, ceea ce duce la roșeață, umflare și un aspect inflamator al sânelui. Acest tip necesită tratament rapid, deoarece se poate răspândi rapid.

6. Boala Paget a sânelui

O formă rară de cancer care afectează mamelonul și areola. Este asociată adesea cu alte tipuri de cancer de sân invazive sau neinvazive. Simptomele includ iritații, roșeață și modificări ale pielii în zona mamelonului.

7. Cancerul de sân triplu negativ

Acest tip de cancer nu are receptori pentru estrogen, progesteron și HER2, ceea ce îl face mai dificil de tratat cu terapiile hormonale convenționale. Afectează adesea femeile tinere și necesită tratament agresiv.

8. Cancerul de sân HER2-pozitiv

Acesta este un tip de cancer de sân în care celulele canceroase au o supraexpresie a

proteinei HER2, care stimulează creșterea tumorală. Este un tip agresiv, dar poate fi tratat eficient cu terapii care țintesc proteina HER2.

Fiecare tip de cancer de sân necesită o abordare specifică de tratament, adaptată la caracteristicile sale și la starea de sănătate a pacientei.

1.4 Metode de obținere a imaginilor medicale

Mamografia este o metodă imagistică esențială în detectarea precoce a cancerului mamar. Folosind raze X la doze mici, mamografia permite obținerea de imagini detaliate ale țesutului mamar, ajutând la identificarea unor posibile anomalii, cum ar fi nodulii sau microcalcificările, care pot indica prezența cancerului. [5]

În timpul procedurii, sânul este poziționat și comprimat între două plăci, ceea ce permite o vizualizare mai clară și reduce doza de radiație necesară. Există două tipuri principale de mamografie: **mamografia convențională (analogică)** și cea **digitală**, ultima oferind imagini de o calitate superioară, care pot fi ajustate și stocate electronic.

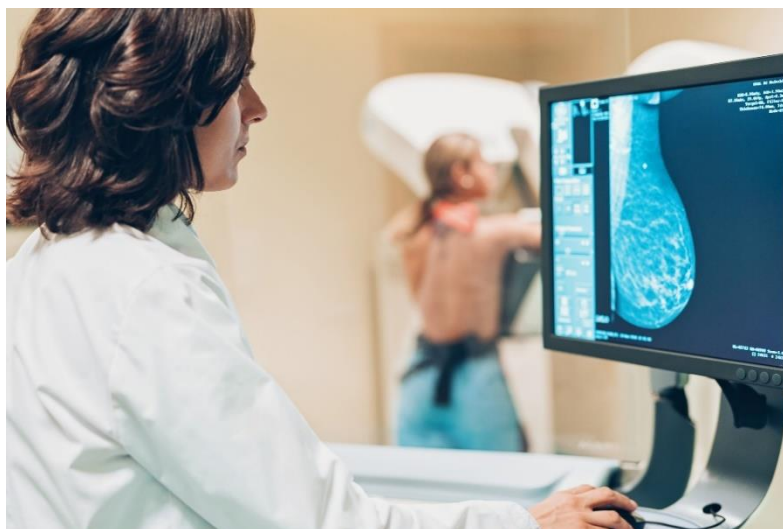


Figura 1.1: Mamograf

Mamografia este recomandată femeilor cu vârste de peste 40 de ani pentru screening regulat, deoarece poate detecta cancerul mamar în stadii incipiente, chiar înainte de apariția simptomelor, îmbunătățind astfel șansele de tratament eficient și reducând mortalitatea asociată.

2. STATE OF THE ART

2.1 Trecerea în revistă a metodelor utilizate pentru detectarea cancerului mamar

Detecția și segmentarea automată a tumorilor mamare sunt esențiale în diagnosticarea timpurie a cancerului mamar. Variabilitatea vizuală a tumorilor și structura complexă a țesutului mamar fac ca analiza imagistică să fie o provocare. În literatura de specialitate, metode avansate de învățare profundă și algoritmi de optimizare au fost aplicate pentru a îmbunătăți acuratețea detectării. Printre metodele cele mai comune se numără rețelele neuronale convoluționale (CNN), algoritmi de ansamblu, rețelele generative adversariale (GAN) și metodele bazate pe analiză statistică și prelucrare de imagini.

2.1.1. "Breast Cancer Detection in Mammography Images: A CNN-Based Approach with Feature Selection" [6]

- ❖ **Metode utilizate:** Studiul utilizează o rețea neuronală convoluțională (CNN) pentru extragerea caracteristicilor, selectând cele mai informative caracteristici pe baza informațiilor comune. Aceste caracteristici sunt apoi clasificate utilizând algoritmi precum rețele neuronale, k-nearest neighbor (kNN), păduri aleatorii (Random Forest) și mașini cu vectori de susținere (SVM).
- ❖ **Baza de date:** Studiul utilizează seturile de date **RSNA**, **MIAS** și **DDSM**, care conțin imagini mamografice cu caracteristici specifice și detalii ale pacienților.
- ❖ **Metrici de evaluare:** Modelul a obținut o acuratețe de 96% pe setul de date **DDSM**, cu îmbunătățiri semnificative în ceea ce privește sensibilitatea și acuratețea față de metodele tradiționale.
- ❖ **Rezumat:** Metoda bazată pe **CNN** a obținut o acuratețe ridicată prin combinarea selecției caracteristicilor și clasificarea cu învățare profundă, demonstrând performanțe îmbunătățite pe mai multe seturi de date.

2.1.2. "Mammography with Deep Learning for Breast Cancer Detection" [7]

- ❖ **Metode utilizate:** Articolul discută modelele de învățare profundă, în special **CNN**, învățarea prin transfer și metodele de ansamblu, pentru segmentarea și clasificarea tumorilor mamare. Modele precum **DenseNet**, **ResNet** și **VGGNet** sunt menționate.
- ❖ **Baza de date:** Seturi de date mamografice precum **DDSM** și seturi private din instituții au fost utilizate pentru antrenarea și evaluarea modelelor.
- ❖ **Metrici de evaluare:** Printre metrici se numără acuratețea, sensibilitatea, specificitatea și aria sub curbă (**AUC**), modelele **CNN** obținând scoruri **AUC** de până la 0,9468 și sensibilități mai mari decât radiologii.
- ❖ **Rezumat:** Studiul a constatat că rețelele **CNN** sunt deosebit de eficiente pentru detectarea cancerului mamar în mamografii, demonstrând sensibilitate și specificitate superioare, în comparație cu imaginile tradiționale.

2.1.3. "Breast Cancer Prediction Using Deep Learning Techniques" [8]

- ❖ **Metode utilizate:** Acest studiu combină tehnici **CNN** și învățare prin transfer, utilizând metode avansate de preprocesare a datelor, cum ar fi denoizarea prin undălet și difuzia anizotropă pentru a îmbunătăți calitatea imaginii.
- ❖ **Baza de date:** Au fost folosite seturi de date publice precum **BCDR** și **INBreast**.
- ❖ **Metrici de evaluare:** Metricile incluse sunt sensibilitatea, specificitatea și scorul **F1**. Modelul a obținut scoruri **F1** de peste 0,90, indicând o performanță de încredere în detectarea cancerului.
- ❖ **Rezumat:** Studiul subliniază importanța preprocesării datelor și selectării modelelor de învățare profundă, rezultând o îmbunătățire a acurateței predicției și segmentării cancerului.

2.1.4. "An Ensemble Approach for Breast Tumor Detection in Mammograms Using Hybrid CNN Models" [9]

- ❖ **Metode utilizate:** Această lucrare propune o abordare de ansamblu, combinând ieșirile din modele **CNN** precum **AlexNet**, **ResNet** și **VGG** pentru a îmbunătăți acuratețea

detectării tumorilor. Modelul de ansamblu a depășit modelele **CNN** individuale.

- ❖ **Baza de date:** Studiul a folosit seturile de date **MIAS** și **DDSM**, concentrându-se pe mamografii cu diagnostice confirmate de cancer.
- ❖ **Metrici de evaluare:** Printre metrici se numără acuratețea, specificitatea și sensibilitatea, iar modelul de ansamblu a obținut o acuratețe de până la 95% și o sensibilitate mai mare decât orice model **CNN** individual.
- ❖ **Rezumat:** Modelul de ansamblu s-a dovedit a fi foarte eficient în detectarea cancerului mamar din mamografii, cu o sensibilitate îmbunătățită și un număr redus de fals pozitive.

3. DESCRIEREA METODELOR PROPUSE

3.1 Metode de optimizare a contrastului

a) Contrast liniar pe porțiuni

Contrastul liniar pe porțiuni este o tehnică folosită pentru a ajusta contrastul unei imagini prin definirea mai multor intervale de intensitate a pixelilor, fiecare cu propria transformare liniară. În loc să aplicăm o transformare liniară

uniformă pe întreaga gamă de valori, imaginea este segmentată în intervale.

Fiecare interval este ajustat astfel încât să maximizeze contrastul în acea regiune specifică. Această tehnică este utilă pentru imagini care conțin detalii atât în regiuni foarte luminoase, cât și în cele întunecate. Prin adaptarea

funcției de contrast în funcție de interval, se poate evita pierderea detaliilor. Algoritmul este adesea aplicat în procese de prelucrare avansată, cum ar fi analiza medicală a imaginilor sau imagistica satelitară.

Modificarea contrastului, liniară pe porțiuni

2 puncte de control:

(a, T_a) și (b, T_b)

2 puncte fixe:

$(0,0)$ și $(L-1, L-1)$

$$T(x) = \begin{cases} \frac{T_a}{a} x, & x \in [0, a] \\ T_a + \frac{T_b - T_a}{b - a} (x - a), & x \in [a, b] \\ T_b + \frac{L - 1 - T_b}{L - 1 - b} (x - b), & x \in [b, L - 1] \end{cases}$$

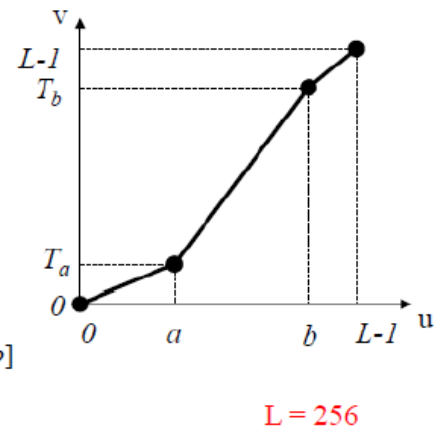


Figura 3.1.a) – Modificarea contrastului, liniară pe porțiuni

b) Contrast putere

Contrastul de tip putere utilizează o funcție non-liniară de forma $T(x) = c \cdot x^\gamma$, unde γ este exponențialul de ajustare, iar

ccc este o constantă de scalare.

Valoarea lui γ (gamma)

determină modul în care

contrastul este accentuat: dacă

$\gamma > 1$, valorile mai mari ale

pixelilor sunt întărite, iar dacă γ

Funcția putere

$$v = T(u) = (L-1) \left(\frac{u}{L-1} \right)^r$$

$r > 0$ parametru de reglaj

aceasta este corectia de gamma

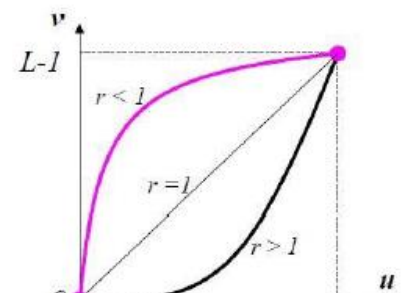


Figura 3.1.b) - Modificarea neliniară a contrastului - funcția putere

< 1 , valorile mai mici sunt accentuate. Această metodă este denumită și corecție gamma. Este frecvent folosită în imagini digitale pentru a îmbunătăți vizibilitatea detaliilor, mai ales în zonele întunecate sau luminoase ale unei imagini. În industria cinematografică și în afișarea grafică, corecția gamma este esențială pentru a adapta imaginile la sensibilitatea ochiului uman.

c) Contrast logarithmic

Contrastul logarithmic folosește o funcție matematică de forma $T(x) = c \cdot \log(1 + x)$, unde c este o constantă de scalare, iar x este valoarea intensității pixelilor. Această metodă este ideală pentru imagini cu o gamă largă de valori de

intensitate, cum ar fi cele

astronomice sau radiografii. Prin utilizarea unei funcții logaritmice, diferențele din intervalele mai mici de intensitate sunt amplificate, în timp ce intervalele superioare sunt

comprimate. Rezultatul este o

imagine cu un contrast îmbunătățit în zonele unde detaliile erau greu de observat. Contrastul logarithmic este util în situații unde este necesar să analizăm zone cu intensități subtile, cum ar fi în studii științifice.

C) Funcția logaritmică

$$v = T(u) = \frac{L-1}{\log L} \log(u+1)$$

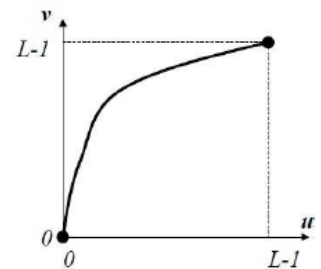


Figura 3.1.c) - Modificarea neliniară a contrastului - funcția logaritmică

d) Egalizare de histograma

Egalizarea de histogramă este o tehnică de procesare a imaginilor care redistribuie uniform intensitățile pixelilor pentru a obține o histogramă cât mai plată. Algoritmul funcționează prin aplicarea unei transformări cumulative asupra

intensităților inițiale, ajustând astfel distribuția pixelilor. Această metodă este extrem de eficientă pentru îmbunătățirea contrastului global, în special în imagini cu iluminare slabă sau distribuite neuniform. Egalizarea este frecvent folosită în imagistica medicală,

în fotografii subacvatice sau în orice context unde detaliile ascunse în umbre sau zone luminoase trebuie

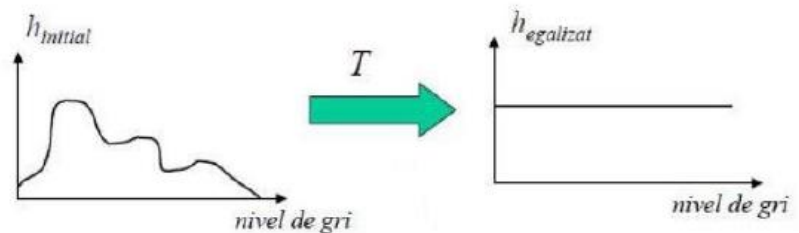


Figura 3.1.d) - egalizare de histogramă

evidențiate. Dezavantajul metodei poate fi amplificarea zgomotului în zone cu puține variații de intensitate.

3.2 Metode de segmentare a imaginilor

a) Binarizare

Binarizarea este procesul de transformare a unei imagini gri în alb-negru, reducând fiecare pixel la două valori posibile: 0 (negru) și 255 (alb). Aceasta se realizează prin aplicarea unui prag fix: dacă valoarea unui pixel este mai mare decât pragul, pixelul devine alb; altfel, devine negru. Este folosită frecvent pentru extragerea obiectelor de interes dintr-o imagine, cum ar fi textul în procesarea documentelor sau contururile în imagini. Simplitatea metodei o face rapidă, dar performanța depinde de alegerea corectă a pragului, mai ales în condiții de iluminare variabilă.

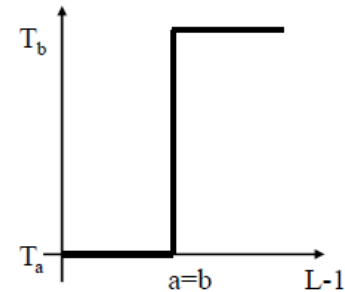


Figura 3.2.a) - binarizarea

b) Slicing

Slicing-ul este o tehnică de procesare a imaginilor care permite evidențierea unui interval specific de valori de intensitate. De exemplu, valorile de intensitate care cad între două limite sunt evidențiate (setate la alb), în timp ce restul devin negre sau sunt atenuate. Această metodă este utilă pentru a izola regiuni de interes în imagini complexe, cum ar fi detectarea organelor în imagistica medicală. Este un instrument puternic pentru analiza segmentată, dar nu modifică distribuția generală a contrastului din imagine.

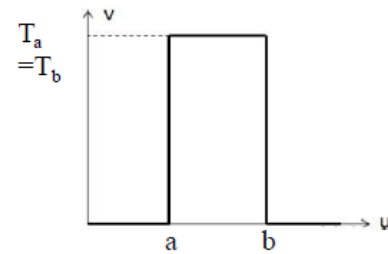


Figura 3.2.b) - Slicing

c) Prăguire Otsu

Prăguirea Otsu este o metodă avansată de binarizare care alege automat un prag optim pe baza distribuției histogrammei unei imagini. Algoritmul maximizează separarea între clasele de pixeli (alb și negru) prin minimizarea varianței intraclase și maximizarea varianței între clase. Este ideal pentru imagini cu două regiuni distincte (obiect și fundal). Spre deosebire de binarizarea simplă, Otsu este mai robust în condiții de iluminare neuniformă. Totuși, metoda poate avea performanțe mai scăzute în

imagini cu zgomot ridicat sau distribuție complexă a pixelilor.

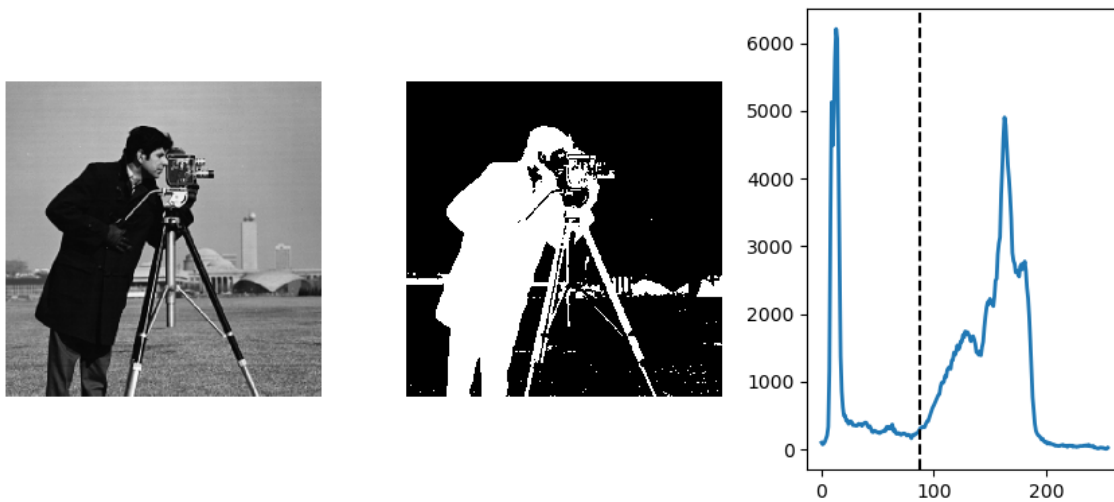


Figura 3.2.c) - Prăguirea Otsu

3.3 Operații morfologice

a) Dilatare

Dilatarea este o operație morfologică care extinde zonele luminoase (albe) dintr-o imagine binară. Scopul este de a mări regiunile obiectelor și de a închide mici goluri sau spații. Se aplică folosind un element structural care determină cât de mult se extind aceste regiuni. Este utilă pentru conectarea componentelor apropiate sau pentru îngroșarea conturilor.

b) Erodare

Erodarea reduce zonele luminoase (albe) dintr-o imagine binară, micșorând regiunile obiectelor. Acest proces elimină zgomotul și conturile subțiri. Este utilizată frecvent pentru a separa obiectele conectate sau pentru a elimina detalii mici și neimportante.

c) Închidere

Închiderea este o combinație de dilatare urmată de erodare. Este folosită pentru a umple golurile mici din obiecte, pentru a conecta componente apropiate și pentru a elimina zgomotul negativ (negru) fără a modifica forma generală a obiectelor.

d) Deschidere

Deschiderea constă în erodare urmată de dilatare. Este folosită pentru a elimina zgomotul pozitiv (alb) și pentru a izola structuri de dimensiuni specifice. Spre deosebire de închidere, aceasta netezește marginile și îndepărtează componentele mici fără a le conecta.

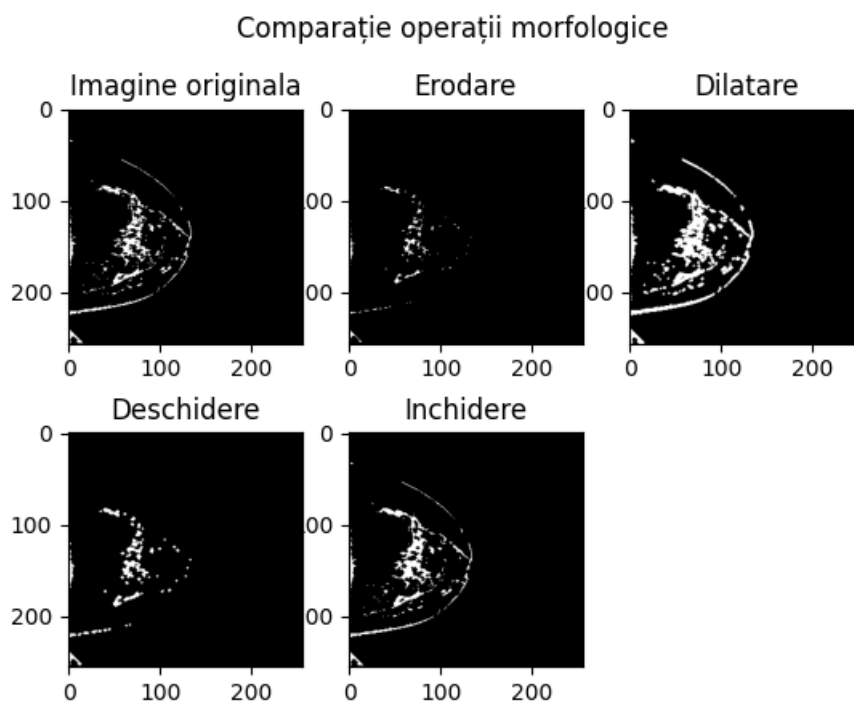


Figura 3.3 - Comparație operații morfologice

4. REZULTATE OBȚINUTE

4.1 Etapa de optimizare a contrastului

Am comparat metodele propuse pentru îmbunătățirea contrastului, aplicându-le pe rând tuturor imaginilor din baza de date. Am ales imaginea 2 din setul de date, oferind o viziune foarte clară asupra eficienței operațiilor de procesare a imaginii aplicate. Se poate observa din figura 4.1 că varianta de contrast liniar pe porțiuni oferă cele mai bune rezultate, accentuând zona canceroasă de la nivelul sânelui. Cu ajutorul acestui contrast am reușit să pregătim imaginea pentru o segmentare mai eficientă la pasul următor. Parametrii utilizați rezultă din codul Python de mai jos:

```
image = Image.fromarray(contrast_liniar_portiuni(m, L=L, a=80, b=175, Ta=50, Tb=205))
```

Atât funcția `contrast_liniar_pe_porțiuni`, cât și toate celelalte funcții utilizate sunt prezentate în ANEXA COD.

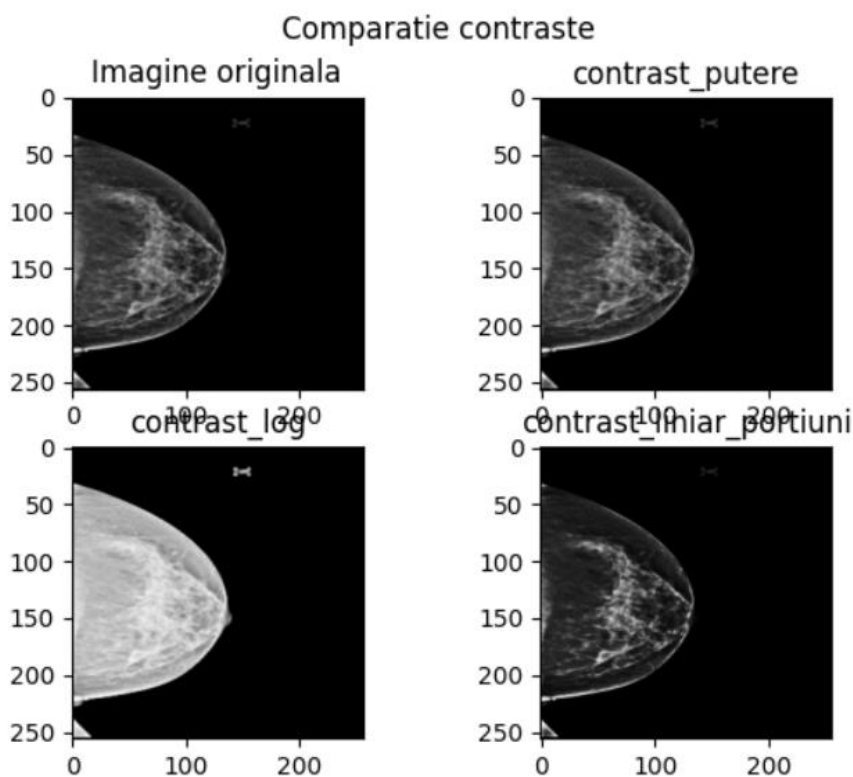


Figura 4.1 - Comparație contraste

Astfel am salvat toate imaginile după aplicarea acestui contrast într-un nou folder separat. Codul complet poate fi găsit în ANEXA COD.

4.1.1 Egalizarea de histogramă

Am aplicat egalizarea de histogramă pe imaginile din baza de date pentru a înțelege mai clar funcționarea ei. După testarea și analizarea imaginilor am ajuns la concluzia că nu este metoda ideală în cazul nostru, dar am ales să prezentăm și rezultatele obținute cu aceasta. Se poate observa în figura 4.1.1 că egalizarea de histogramă luminează prea mult zone care nu sunt de interes, pentru a compensa zona foarte mare neagră din imagine (exterioară sânului).

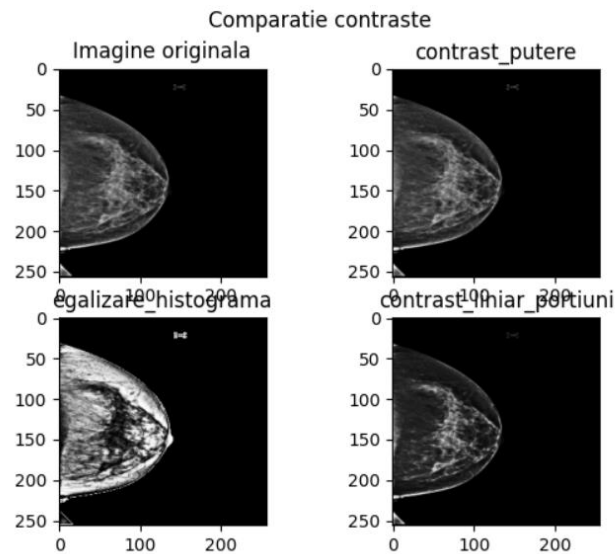


Figura 4.1.1 - Egalizare histogramă

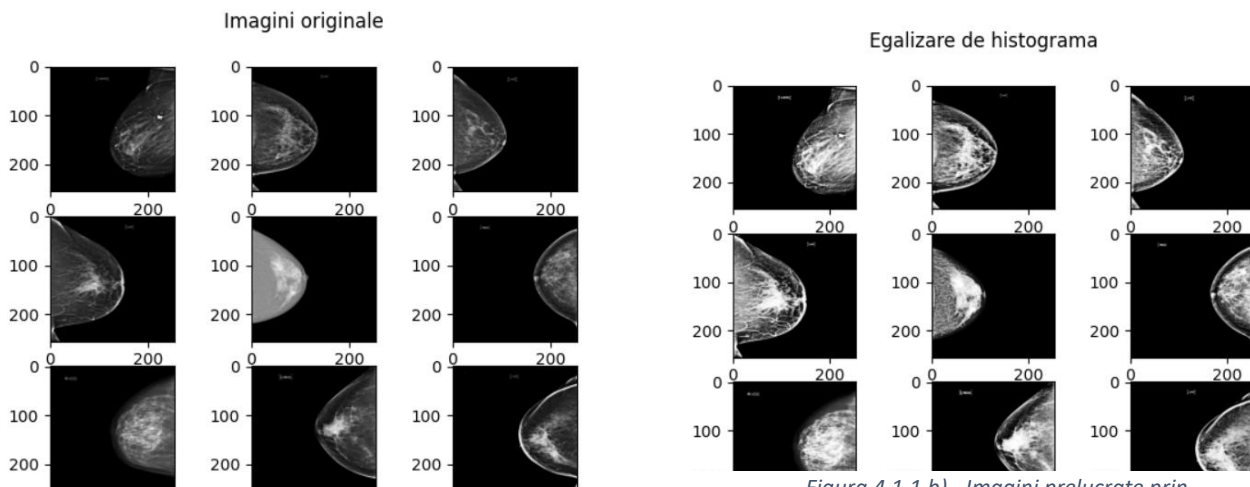


Figura 4.1.1.a) - Imagini originale

Figura 4.1.1.b) - Imagini prelucrate prin
egalizare de histogramă

4.2 Etapa de segmentare a zonei de interes

Pentru segmentarea zonei de interes am ales binarizarea, deoarece în cazul imaginilor noastre zonele de interes prezentau valori de intensitate luminoasă peste un anumit prag, nu între anumite valori (caz în care ar fi fost mai utilă metoda de slicing. Pentru alegerea pragului am trecut prin câteva iterații, ajungând la valoarea de 110, pentru care putem observa în toate imaginile din Figura 4.2.a) că există masă tumorală.

Imagini segmentate

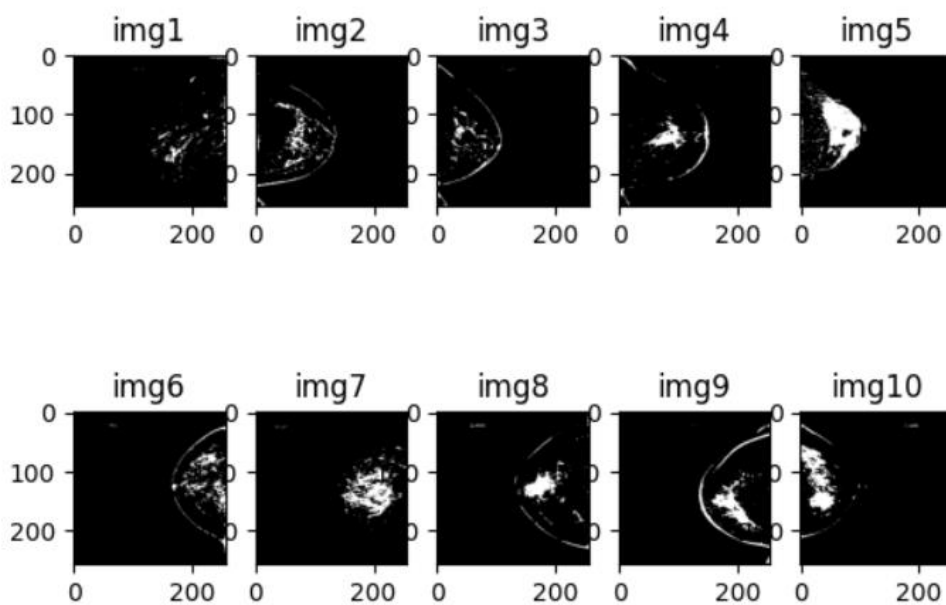


Figura 4.2.a) - Imaginile segmentate

Imaginea 1 este singura în care nu se vede foarte clar faptul că există masă canceroasă, însă comparativ cu un sân sănătos se vede că poate exista o problemă. În baza de date am ales doar imagini cu sâni în care s-a detectat problema de sănătate.

În figura 4.2.b), păstrând ca reper imaginea 2 din setul de date putem observa efectul produs de aplicarea binarizării asupra imaginii obținute după aplicarea contrastului de la punctul anterior.

Codul aferent poate fi consultat în ANEXA COD

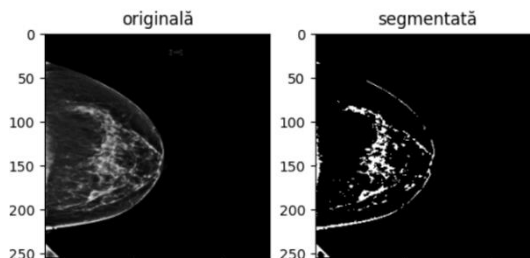


Figura 4.2.b) - Img2 după segmentare

4.3 Etapa aplicării operațiilor morfologice

În urma binarizării am obținut imagini binare ce pot fi acum procesate prin aplicarea de operații morfologice precum cele prezentate în capitolul 3. Mai jos, în figura 4.3.a) am realizat o comparație între cele 4 operații, concluzionând ca închiderea este singura care produce un efect pozitiv semnificativ în îmbunătățirea imaginilor. Trebuie precizat că imaginea aleasă este imaginea numărul 2, iar codul utilizat va fi prezentat în ANEXA COD.

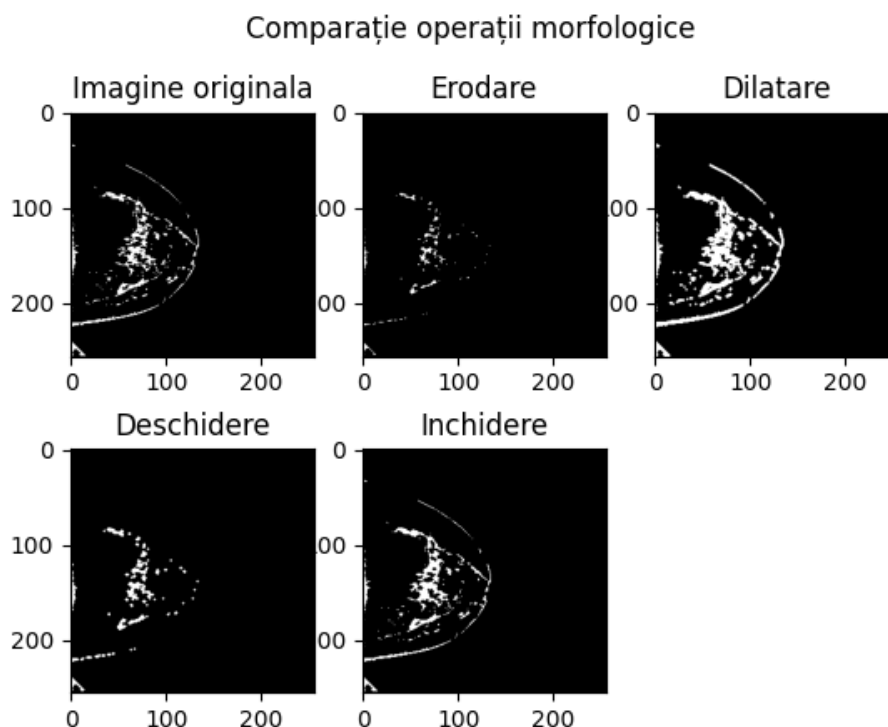


Figura 4.3.a) - Comparație operații morfologice

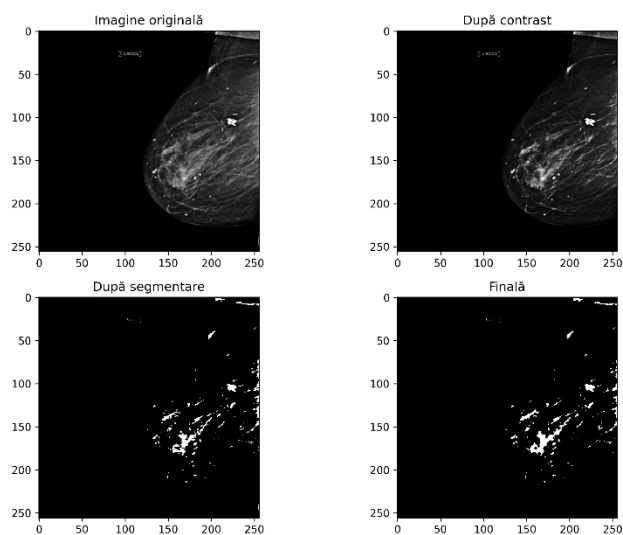
Prin aplicarea operației morfologice de închidere, am reușit să obținem imaginea finală, pregătită pentru extragerea de caracteristici.

4.4 Extragerea de caracteristici

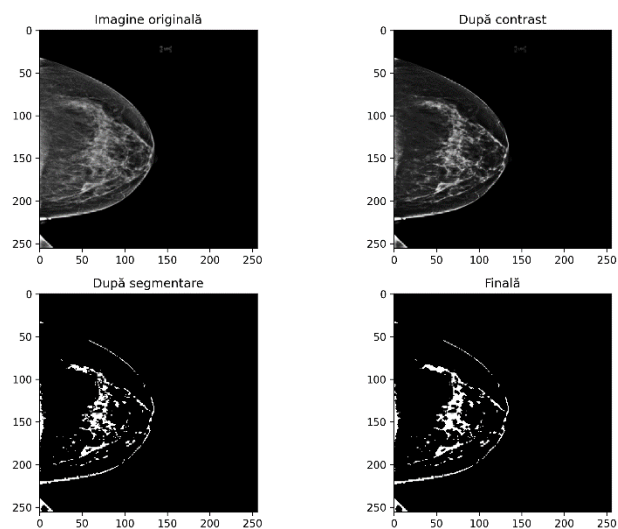
În urma prelucrării tuturor imaginilor am obținut rezultate satisfăcătoare, conform imaginilor de mai jos, unde am prezentat parcursul prelucrării fiecărei imagini din dataset. Analizând rezultatele, observăm că în 9 din 10 imagini se observă foarte clar prezența unei tumori, pe când în una din cele 10

imagini este mai puțin clar acest lucru, deci putem spune că operațiile folosite pentru procesarea imaginilor au avut o rata de succes de peste 90%.

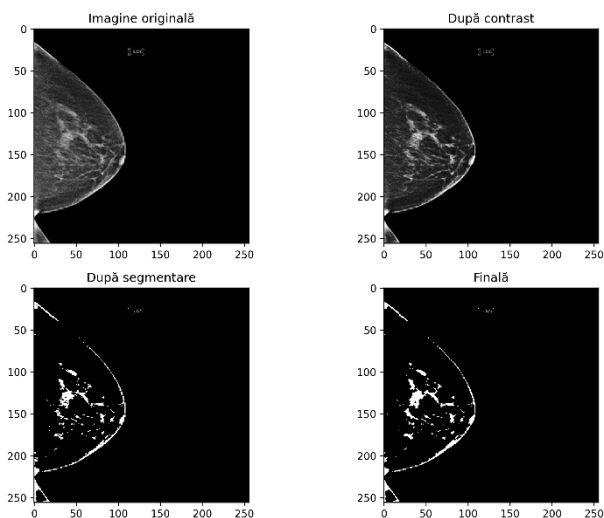
Parcurs imagine 1



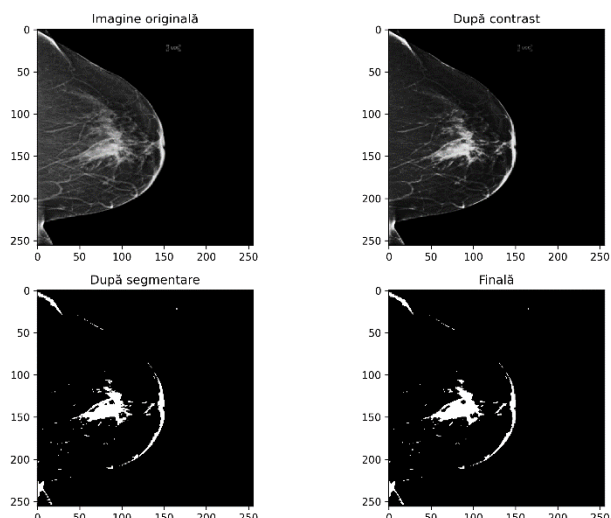
Parcurs imagine 2



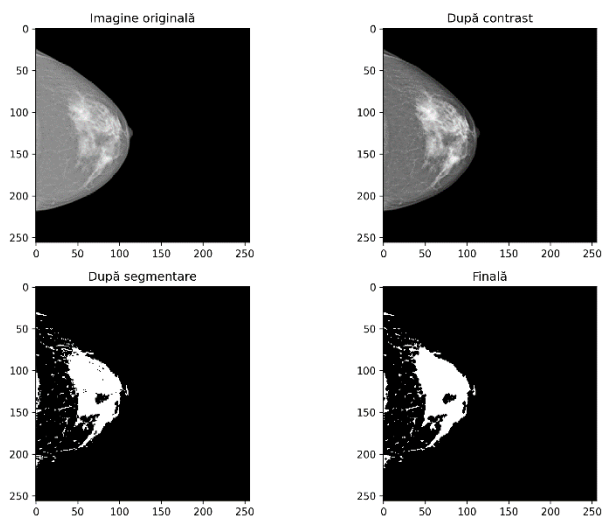
Parcurs imagine 3



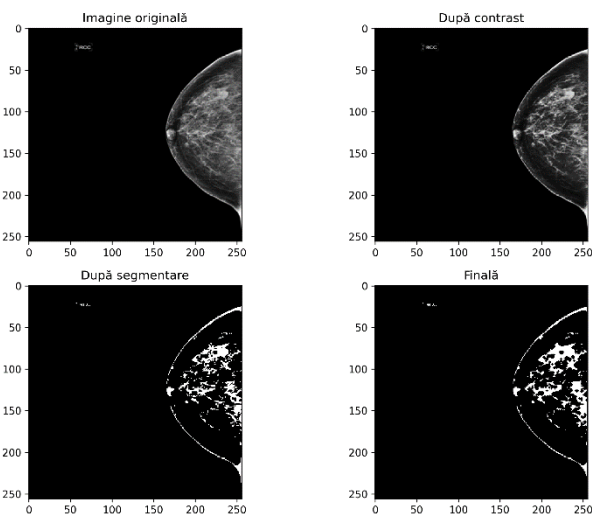
Parcurs imagine 4



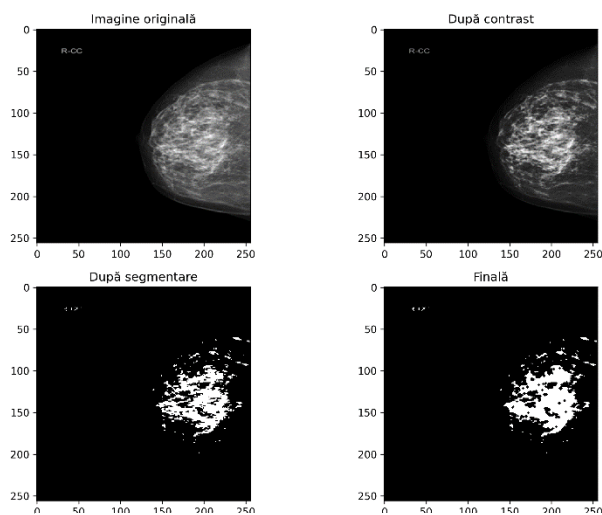
Parcurs imagine 5



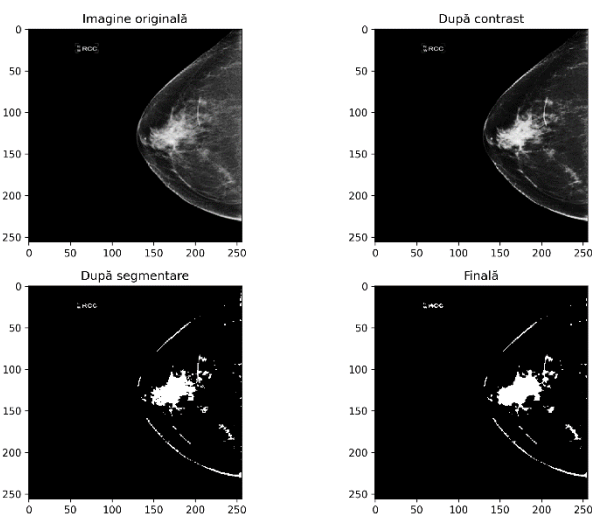
Parcurs imagine 6



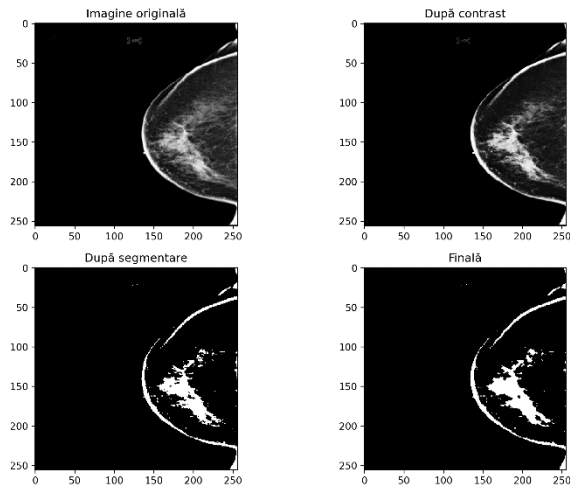
Parcurs imagine 7



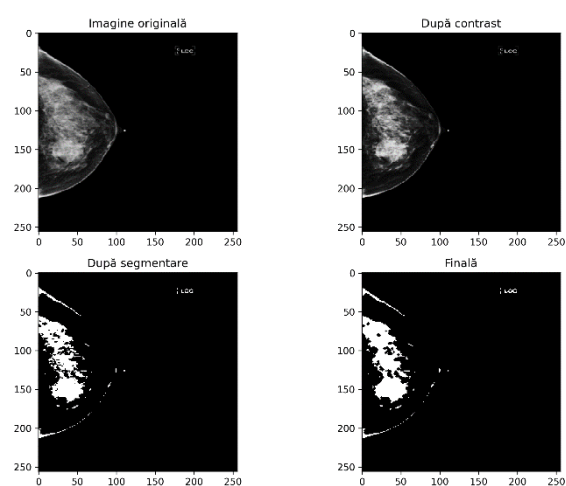
Parcurs imagine 8



Parcurs imagine 9



Parcurs imagine 10



5. CONCLUZII

În cadrul acestui proiect, am acumulat cunoștințe și experiențe valoroase în domeniul prelucrării imaginilor medicale și al utilizării algoritmilor de procesare a datelor. Principalele concluzii rezultate sunt următoarele:

- **Procesarea imaginilor mamografice:** Am aprofundat utilizarea metodelor de optimizare a contrastului, segmentare și aplicarea operațiilor morfologice pentru evidențierea și delimitarea tumorilor. Aceste tehnici contribuie la o mai bună vizibilitate a zonelor de interes, esențială pentru diagnosticare.
- **Eficiența metodelor aplicate:** Compararea tehnicilor de contrast (cum ar fi prăguirea Otsu sau egalizarea de histogramă) a demonstrat că anumite metode sunt mai potrivite pentru imaginile cu structuri complexe, precum cele din mamografii. În special, contrastul liniar pe porțiuni a oferit cele mai bune rezultate.
- **Rolul operațiilor morfologice:** Prin aplicarea operațiilor de închidere, am reușit să eliminăm zgomotul și să obținem o segmentare precisă, ceea ce a facilitat analiza ulterioară a caracteristicilor tumorale.
- **Abilități tehnice consolidate:** Proiectul ne-a oferit oportunitatea de a consolida cunoștințele de programare în Python, în special utilizarea bibliotecilor precum OpenCV și Matplotlib pentru procesarea imaginilor.
- **Impactul asupra diagnosticării:** Rezultatele obținute arată că tehnicile propuse au o rată de succes de peste 90%, evidențiind tumori în 9 din cele 10 imagini analizate. Acest fapt subliniază potențialul algoritmilor dezvoltați în îmbunătățirea detectării cancerului mamar.

Prin aplicarea metodelor analizate, am demonstrat că automatizarea procesării imaginilor medicale poate susține în mod eficient diagnosticarea precoce și precisă a cancerului mamar, cu potențial semnificativ pentru implementarea clinică.

BIBLIOGRAFIE

- [1] „Cancerul de san”. Data accesării: 12 decembrie 2024. [Online]. Disponibil la:
<https://www.reginamaria.ro/utile/dictionar-de-afectiuni/cancerul-de-san>
- [2] F. D. Max, „Cancerul de san: Cauze, manifestari si optiuni de tratament”, Dr.Max Farmacie. Data accesării: 12 decembrie 2024. [Online]. Disponibil la: <https://www.drmax.ro/articole/cancerul-de-san-cauze-manifestari-tratament>
- [3] „American Cancer Society Guidelines for Breast Cancer Screening: Update 2003 - Smith - 2003 - CA: A Cancer Journal for Clinicians - Wiley Online Library”. Data accesării: 12 decembrie 2024. [Online]. Disponibil la: <https://acsjournals.onlinelibrary.wiley.com/doi/full/10.3322/canjclin.53.3.141>
- [4] J. Ewing, „CLASSIFICATION OF MAMMARY CANCER”, *Ann. Surg.*, vol. 102, nr. 2, pp. 249–252, aug. 1935.
- [5] T. Cebanu, „Diagnosticul cancerului glandei mamare prin mamografie”, 2018, doi: 10/9963.
- [6] Z. Jafari și E. Karami, „Breast Cancer Detection in Mammography Images: A CNN-Based Approach with Feature Selection”, *Information*, vol. 14, nr. 7, Art. nr. 7, iul. 2023, doi: 10.3390/info14070410.
- [7] L. Wang, „Mammography with deep learning for breast cancer detection”, *Front. Oncol.*, vol. 14, feb. 2024, doi: 10.3389/fonc.2024.1281922.
- [8] A. Allada, G. R. K. Rao, P. Chitturi, H. Chindu, M. S. N. Prasad, și P. Tatineni, „Breast Cancer Prediction using Deep Learning Techniques”, în *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, mar. 2021, pp. 306–311. doi: 10.1109/ICAIS50930.2021.9395793.
- [9] A. Sahu, P. K. Das, și S. Meher, „High accuracy hybrid CNN classifiers for breast cancer detection using mammogram and ultrasound datasets”, *Biomed. Signal Process. Control*, vol. 80, p. 104292, feb. 2023, doi: 10.1016/j.bspc.2022.104292.

ANEXA COD

Fișierul contrast.py:

```
import os
from functools import partial
from PIL import Image

import numpy as np
import matplotlib.pyplot as plt
import cv2

def rgb2gri(img_in, format):
    img_in=img_in.astype('float')
    s=img_in.shape
    if len(s)==3 and s[2]==3:
        if format=='png':

img_out=(0.299*img_in[:, :, 0]+0.587*img_in[:, :, 1]+0.114*img_in[:, :, 2])*255
        elif format=='jpg':

img_out=0.299*img_in[:, :, 0]+0.587*img_in[:, :, 1]+0.114*img_in[:, :, 2]
        img_out=np.clip(img_out, 0,255)
        img_out=img_out.astype('uint8')
        return img_out
    else:
        print('Conversia nu a putut fi realizata deoarece imaginea de intrare
nu este color!')
        return img_in
def contrast_liniar_portiuni(img_in, L, a, b, Ta, Tb):
    s = img_in.shape
    img_out = np.empty_like(img_in)
    img_in = img_in.astype(float)
    for i in range(0, s[0]):
        for j in range(0, s[1]):
            if (img_in[i, j] < a):
                img_out[i, j] = (Ta / a) * img_in[i, j]
```

```
        if (img_in[i, j] >= a and img_in[i, j] <= b):
            img_out[i, j] = Ta + ((Tb - Ta) / (b - a)) * (img_in[i, j] -
a)

        if (img_in[i, j] > b):
            img_out[i, j] = Tb + ((L - 1 - Tb) / (L - 1 - b)) *
(img_in[i, j] - b)

img_out = np.clip(img_out, 0, 255)
img_out = img_out.astype('uint8')
return img_out

def contrast_putere(img_in, L, r):
    s = img_in.shape
    img_out = np.empty_like(img_in)
    img_in = img_in.astype(float)
    for i in range(0, s[0]):
        for j in range(0, s[1]):
            img_out[i, j] = (L-1) * (img_in[i, j] / (L-1))**r

img_out = np.clip(img_out, 0, 255)
img_out = img_out.astype('uint8')
return img_out

def contrast_log(img_in, L):
    s = img_in.shape
    img_out = np.empty_like(img_in)
    img_in = img_in.astype(float)
    for i in range(0, s[0]):
        for j in range(0, s[1]):
            img_out[i, j] = ((L-1)/np.log(L)) * np.log(img_in[i, j]+1)

img_out = np.clip(img_out, 0, 255)
img_out = img_out.astype('uint8')
return img_out

def contrast_exp(img_in, L):
    s = img_in.shape
```



```
img_out = np.empty_like(img_in)
img_in = img_in.astype(float)
for i in range(0, s[0]):
    for j in range(0, s[1]):
        img_out[i, j] = L*(img_in[i,j]/(L-1)) - 1

img_out = np.clip(img_out, 0, 255)
img_out = img_out.astype('uint8')
return img_out

def contrast_putere_cu_pct_fix(img_in, L, r,a):
    s = img_in.shape
    img_out = np.empty_like(img_in)
    img_in = img_in.astype(float)
    for i in range(0, s[0]):
        for j in range(0, s[1]):
            if(img_in[i,j]<=a):
                img_out[i,j] = a*(img_in[i,j]/a)**r
            else:
                img_out[i,j] = L-1 - (L-1-a) * ((L-1-img_in[i,j])/(L-1-a))**r

    img_out = np.clip(img_out, 0, 255)
    img_out = img_out.astype('uint8')
    return img_out

def afisare(img,f1,f2,f3):
    img_1 = f1(img)
    img_2 = f2(img)
    img_3 = f3(img)

    plt.suptitle("Comparatie contraste")
    plt.subplot(2,2,1),plt.imshow(img,cmap="gray"),plt.title("Imagine
originala")

plt.subplot(2,2,2),plt.imshow(img_1,cmap="gray"),plt.title(f1.func.__name__)

plt.subplot(2,2,3),plt.imshow(img_2,cmap="gray"),plt.title(f2.func.__name__)
```

```
plt.subplot(2,2,4),plt.imshow(img_3,cmap="gray"),plt.title(f3.func.__name__)
plt.show()

def egalizare_histograma(img):
    # Verifică dacă imaginea este una color (RGB/BGR)
    if len(img.shape) == 3: # Imagine color
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Asigură-te că imaginea este în format de 8 biți (CV_8UC1)
    img = np.uint8(img * 255) if img.dtype != np.uint8 else img

    # Aplică egalizarea histogramei
    img_equalizat = cv2.equalizeHist(img)

    return img_equalizat

cale = os.path.join(os.path.dirname(__file__), "img")

save_directory = os.path.join(os.path.dirname(__file__), "contrast_img")
os.makedirs(save_directory, exist_ok=True)

L = 256
for i in range(1,11):
    cale_img = os.path.join(cale, f"{i}.png")
    try:
        m = plt.imread(cale_img)
        m = m * (L - 1)
        print(f"Image {i}.png loaded successfully!")
        afisare(
            m,
            partial(contrast_putere,L=L,r=1),
            #partial(contrast_log,L=L),
            partial(egalizare_histograma),
            partial(contrast_liniar_portiuni,L=L,a=80,b=175,Ta=50,Tb=205)
```

```
)  
    save_path = os.path.join(save_directory, f"contrast_{i}.png")  
    image =  
Image.fromarray(contrast_liniar_portiuni(m, L=L, a=80, b=175, Ta=50, Tb=205))  
    image.save(save_path)  
    print(f"Image saved at {save_path}")  
except FileNotFoundError:  
    print(f"Image {i}.png not found in {cale}")  
  
print("Contrasted images saved")  
  
plt.suptitle("Imagini originale")  
for i in range(1,10):  
    cale_img = os.path.join(cale, f"{i}.png")  
    m = plt.imread(cale_img)  
    plt.subplot(3,3,i), plt.imshow(m, cmap="gray")  
plt.show()  
  
plt.suptitle("Egalizare de histograma")  
for i in range(1,10):  
    cale_img = os.path.join(cale, f"{i}.png")  
    m = plt.imread(cale_img)  
    plt.subplot(3,3,i), plt.imshow(egalizare_histograma(m), cmap="gray")  
plt.show()
```

Fișierul segmentare.py:

```
import os  
import numpy as np  
import matplotlib.pyplot as plt  
from functools import partial  
from PIL import Image  
import cv2  
  
def binarizare(img_in, L, a):  
    s = img_in.shape  
    img_out = np.empty_like(img_in)
```

```
img_in = img_in.astype(float)
for i in range(0, s[0]):
    for j in range(0, s[1]):
        if (img_in[i, j] < a):
            img_out[i, j] = 0
        else:
            img_out[i, j] = L-1

img_out = np.clip(img_out, 0, 255)
img_out = img_out.astype('uint8')
return img_out

cale = os.path.join(os.path.dirname(__file__), "contrast_img")

save_directory = os.path.join(os.path.dirname(__file__), "segmentare_img")
os.makedirs(save_directory, exist_ok=True)

L=256
prag = 110

plt.suptitle("Imagini segmentate")

for i in range(1,11):
    cale_img = os.path.join(cale, f"contrast_{i}.png")
    try:
        m = plt.imread(cale_img)
        m = m * (L - 1)
        print(f"Image {i}.png loaded successfully!")

        save_path = os.path.join(save_directory, f"segmentare_{i}.png")
        image = Image.fromarray(binarizare(m,L,prag))
        image.save(save_path)

        plt.subplot(2, 5, i), plt.imshow(image, cmap="gray"),
plt.title(f"img{i}")
```

```
print(f"Image saved at {save_path}")
except FileNotFoundError:
    print(f"Image {i}.png not found in {cale}")

print("Segmented images saved")
plt.show()

def praguire_otsu(img):
    # Verifică dacă imaginea este una color (RGB/BGR)
    if len(img.shape) == 3: # Imagine color
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Convertim imaginea în 8 biți dacă nu este deja
    if img.dtype != np.uint8:
        img = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX) #
        Normalizează la intervalul [0, 255]
        img = np.uint8(img) # Conversie la 8 biți

    # Aplică metoda Otsu pentru praguire
    _, img_praguita = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY +
    cv2.THRESH_OTSU)

    return img_praguita

cale_img = os.path.join(cale, f"contrast_2.png")
m = plt.imread(cale_img)
m = m * (L - 1)

plt.suptitle("Exemplu otsu")
plt.plot(), plt.imshow(praguire_otsu(m), cmap="gray"), plt.title(1)
plt.show()

plt.suptitle("Img2 segmentare")
plt.subplot(1,2,1), plt.imshow(m, cmap="gray"), plt.title("originală")
plt.subplot(1,2,2), plt.imshow(binarizare(m,L,prag), cmap="gray"), plt.title("se
```

```
gumentată")  
plt.show()
```

Fișierul Operatii_morfologice.py:

```
import os  
import numpy as np  
import matplotlib.pyplot as plt  
from functools import partial  
from PIL import Image  
import cv2  
import scipy.ndimage as sc  
  
def afisare(img):  
    img_1 = sc.binary_erosion(img)  
    img_2 = sc.binary_dilation(img)  
    img_3 = sc.binary_opening(img)  
    img_4 = sc.binary_closing(img)  
  
    plt.suptitle("Comparație operații morfologice")  
    plt.subplot(2,3,1),plt.imshow(img,cmap="gray"),plt.title("Imagine  
originala")  
    plt.subplot(2,3,2),plt.imshow(img_1,cmap="gray"),plt.title("Erodare")  
    plt.subplot(2,3,3),plt.imshow(img_2,cmap="gray"),plt.title("Dilatare")  
    plt.subplot(2,3,4),plt.imshow(img_3,cmap="gray"),plt.title("Deschidere")  
    plt.subplot(2, 3, 5), plt.imshow(img_4, cmap="gray"),  
plt.title("Inchidere")  
plt.show()
```

```
cale = os.path.join(os.path.dirname(__file__), "segmentare_img")

save_directory = os.path.join(os.path.dirname(__file__), "img_finale")
os.makedirs(save_directory, exist_ok=True)

L=256
for i in range(1,11):
    cale_img = os.path.join(cale, f"segmentare_{i}.png")
    try:
        m = plt.imread(cale_img)
        m= m/(L-1)
        print(f"Image segmentare_{i}.png loaded successfully!")

        afisare(m)

        save_path = os.path.join(save_directory, f"final_{i}.png")
        m=m*L
        image = Image.fromarray(sc.binary_closing(m))
        image.save(save_path)

        print(f"Image saved at {save_path}")
    except FileNotFoundError:
        print(f"Image {i}.png not found in {cale}")

print("Segmented images saved")
```

Fișierul extragere_caracteristici.py:

```
import os
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import cv2
```

```
# Directorii de lucru
cale = os.path.join(os.path.dirname(__file__), "img")
cale_contrast = os.path.join(os.path.dirname(__file__), "contrast_img")
cale_segmentare = os.path.join(os.path.dirname(__file__), "segmentare_img")
cale_final = os.path.join(os.path.dirname(__file__), "img_finale")

cale_procesare = os.path.join(os.path.dirname(__file__), "img_proces")
os.makedirs(cale_procesare, exist_ok=True)

# Procesarea imaginilor
for i in range(1, 11):
    try:
        # Creează o figură pentru imaginea curentă
        plt.figure(figsize=(10, 8))

        # Imagine Originală
        cale_img = os.path.join(cale, f"{i}.png")
        try:
            originala = plt.imread(cale_img)
            plt.subplot(2, 2, 1), plt.imshow(originala, cmap="gray"),
plt.title("Imagine originală")
        except FileNotFoundError:
            print(f"Fișierul nu a fost găsit: {cale_img}")

        # Imagine după ajustarea contrastului
        contrast_img = os.path.join(cale_contrast, f"contrast_{i}.png")
        try:
            contrast = plt.imread(contrast_img)
            plt.subplot(2, 2, 2), plt.imshow(contrast, cmap="gray"),
plt.title("După contrast")
        except FileNotFoundError:
            print(f"Fișierul nu a fost găsit: {contrast_img}")

        # Imagine segmentată
        img_segmentata = os.path.join(cale_segmentare, f"segmentare_{i}.png")
        try:
```



```
segmentare = plt.imread(img_segmentata)
plt.subplot(2, 2, 3), plt.imshow(segmentare, cmap="gray"),
plt.title("După segmentare")

except FileNotFoundError:
    print(f"Fișierul nu a fost găsit: {img_segmentata}")

# Imagine finală
img_final = os.path.join(cale_final, f"final_{i}.png")
try:
    final = plt.imread(img_final)
    plt.subplot(2, 2, 4), plt.imshow(final, cmap="gray"),
plt.title("Finală")

except FileNotFoundError:
    print(f"Fișierul nu a fost găsit: {img_final}")

# Adaugă un titlu general și ajustează layout-ul
plt.suptitle(f"Parcurs imagine {i}", fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.95]) # Lasă spațiu pentru titlu

# Salvează și afișează figura
cale_salvare = os.path.join(cale_procesare, f"proces_{i}.png")
plt.savefig(cale_salvare, dpi=300, bbox_inches='tight')
plt.show()

print(f"Imaginea {i}.png a fost procesată și salvată în
{cale_salvare}")

except Exception as e:
    print(f"Eroare la procesarea imaginii {i}: {e}")
```