



**Universitatea
Transilvania
din Brașov**

**FACULTATEA DE MATEMATICĂ
ȘI INFORMATICĂ**

Programul de studii:

Informatică Aplicată

LUCRARE DE LICENȚĂ

Absolvent: **Laurențiu-Adrian Andrei**

Coordonator științific: **Lect. Dr. Vlad Monescu**

Brașov, 2022



**Universitatea
Transilvania
din Brașov**

**FACULTATEA DE MATEMATICĂ
ȘI INFORMATICĂ**

Programul de studii:

Informatică Aplicată

LUCRARE DE LICENȚĂ

Segmentarea tumorilor cerebrale din RMN-uri

Absolvent: **Laurențiu-Adrian Andrei**

Coordonator științific: **Lect. Dr. Vlad Monescu**

Brașov, 2022

CUPRINS

LISTĂ DE FIGURI ȘI CODURI SURSĂ.....	5
LISTĂ DE ACRONIME	7
1. INTRODUCERE	8
1.1. Actualitatea temei.....	8
1.2. Scopul și obiectivele lucrării	8
1.3. Imagistica medicală.....	9
1.4. Structura temei.....	9
2. CREIERUL UMAN	11
2.1. Anatomia creierului	11
2.2. Tumorile cerebrale	12
3. MEDII ȘI CONCEPTE DE PROGRAMARE.....	13
3.1. Limbajul de programare C++.....	13
3.2. Biblioteca OpenCV.....	13
3.3. Biblioteca Algorithms.....	14
3.4. Biblioteca OpenMP	15
3.5. Mediul de dezvoltare Qt.....	15
3.5.1. Qt Designer.....	16
3.6. Platforma CMake	16
4. SEGMENTAREA TUMORILOR CEREBRALE	18
4.1. Setul de imagini utilizat	18
4.2. Preprocesarea imaginilor	20
4.2.1. Conversia imaginii în grayscale	20
4.2.2. Reducerea zgomotului	21
4.2.3. Eliminarea fundalului din imagine	26
4.2.4. Eliminarea craniului din imagine.....	27
4.3. Extragerea caracteristicilor	34
4.3.1. Algoritmul de grupare K-means	34

4.3.2.	Operațiunea morfologică de deschidere	36
4.3.3.	Algoritmul componentelor conexe.....	38
4.4.	Detectarea tumorii cerebrale.....	39
5.	INTERFAȚA GRAFICĂ.....	41
5.1.	Realizarea interfeței grafice	41
5.2.	Testare și simulare	44
6.	DIRECȚII VIITOARE DE CERCETARE.....	49
6.1.	Idei de dezvoltat	49
7.	CONCLUZII	53
7.1.	Concluzii generale	53
7.2.	Elemente de originalitate.....	54
8.	BIBLIOGRAFIE.....	56
	REZUMAT.....	59
	ABSTRACT.....	61
	ANEXA A COD SURSĂ IMPLEMENTARE ELIMINARE CRANIU DIN IMAGINE.....	62
	ANEXA B CON SURSĂ ESTIMARE ZGOMOT DINTR-O IMAGINE.....	64
	DECLARAȚIE PRIVIND ORIGINALITATEA.....	65

LISTĂ DE FIGURI ȘI CODURI SURSĂ

FIGURI

- Fig. 1. Anatomia creierului uman.
- Fig. 2. Tipuri de tumori cerebrale.
- Fig. 3. Componentele OpenCV.
- Fig. 4. Structura directorului sursă al proiectului.
- Fig. 5. Pași principali pentru segmentarea și diagnosticarea tumorilor cerebrale.
- Fig. 6. Exemple de imagini din setul de date Brain Tumor MRI Dataset: (a) Tumoră de tip gliom; (b) Tumoră de tip meningiom; (c) Tumoră pituitară.
- Fig. 7. Pași necesari pentru realizarea preprocesării.
- Fig. 8. Imaginea inițială (stânga); Imaginea după aplicarea filtrului mediu cu un nucleu de 5x5 (dreapta).
- Fig. 9. Imaginea inițială (stânga); Imaginea după aplicarea filtrului median (dreapta).
- Fig. 10. Imaginea inițială (stânga); Imaginea după aplicarea filtrului gaussian cu un nucleu de 5x5 (dreapta).
- Fig. 11. Imaginea inițială (stânga); Imaginea după aplicarea filtrului bilateral cu un nucleu de 5x5 (dreapta).
- Fig. 12. Rezultate medii ale nivelului de zgomot.
- Fig. 13. Timpul mediu de execuție, timpul median de execuție și deviația standard a algoritmilor de reducere a zgomotului din imagini.
- Fig. 14. Masca zonei de interes din imaginea de intrare.
- Fig. 15. Histograma cumulativă și evidențierea valorii de prag.
- Fig. 16. (a) Imaginea inițială; (b) Imaginea rezultată în urma partiționării; (c) Imaginea după eliminarea craniului.
- Fig. 17. (a) Imaginea inițială; (b) Imaginea după eliminarea craniului.
- Fig. 18. (a) Imaginea inițială; (b) Masca obținută prin diferența celor 2 măști calculate în algoritm; (c) Imaginea după eliminarea craniului.
- Fig. 19. (a) Imaginea inițială; (b) Poligonul estimat care cuprinde zona din imagine unde se află creierul; (c) Imaginea după eliminarea craniului.
- Fig. 20. (a) Imaginea de intrare; (b) Valorile centrozilor calculați prin algoritmul de grupare K-means; (c) Imaginea după aplicarea algoritmului.

Fig. 21. Imaginea de intrare (stânga); Imaginea obținută după aplicarea operației de deschidere morfologică (dreapta).

Fig. 22. Metoda de conectare de 4 pixeli (stânga); Metoda de conectare de 8 pixeli (dreapta).

Fig. 23. Imaginea de intrare (stânga); Imaginea după aplicarea algoritmului componentelor conexe (dreapta).

Fig. 24. (a) Imaginea inițială; (b) Imaginea finală, cu tumora cerebrală evidențiată.

Fig. 25. Qt Designer – zona de proiectare a interfeței grafice.

Fig. 26. Arhitectura elementelor interfeței grafice a utilizatorului.

Fig. 27. Interfața grafică.

Fig. 28. Meniurile disponibile în interfața grafică.

Fig. 29. Activarea acțiunilor din meniul de preprocesare.

Fig. 30. Interfața grafică completă, prezentând o tumoră cerebrală de tip gliom.

Fig. 31. Interfața grafică completă, prezentând o tumoră cerebrală de tip meningiom.

Fig. 32. (a) Astrocitom; (b) Carcinom al plexului coroid; (c) Craniofaringiom.

Fig. 33. Tumoră mamară (stânga); Tumoră pancreatică (dreapta).

Fig. 34. Segmentarea lobilor cerebrali.

LISTĂ DE ACRONIME

RMN – Rezonanță Magnetică Nucleară

DLL – Bibliotecă de Legături Dinamice

OpenCV – Open Computer Vision

OpenMP – Open Multi-Processing

GUI – Graphical User Interface

RGB – Roșu, Verde, Albastru

1. INTRODUCERE

1.1. Actualitatea temei

În conformitate cu informațiile publicate [1], în întreaga lume, oamenii suferă de forme diferite de tumori. Această afecțiune se află printre cele mai mari probleme ale secolului XXI, frecvența persoanelor care pot dezvolta această boală pe parcursul vieții ajungând până la aproximativ 50% [2]. Printre posibilele cauze ale creșterii numărului de persoane care dezvoltă de-a lungul vieții tumori putem enumera creșterea speranței de viață [2] sau creșterea nivelului de radiații din jurul nostru [3]

Unele dintre cele mai grave tipuri de tumori sunt tumorile cerebrale. Tumorile cerebrale sunt o colecție (o masă) de celule anormale aflate în creier care se înmulțesc într-un mod necontrolat. Craniul uman, cel care conține creierul, este foarte rigid și orice modificare a țesutului din interiorul său poate cauza probleme grave, din cauza spațiului restrâns din interiorul acestuia.

Unul dintre modurile în care aceste tumori cerebrale pot fi diagnosticate sunt prin rezonanță magnetică nucleară (RMN).

O scanare RMN reprezintă o tehnică nedureroasă de imagistică medicală care are avantajul că evită expunerea la radiații. Această tehnică utilizează câmpul magnetic puternic, undele radio și un computer pentru a produce imagini ale structurilor corpului.

1.2. Scopul și obiectivele lucrării

Scopul lucrării de diplomă este de a dezvolta o aplicație care să poată ajuta la detectarea tumorilor cerebrale. Pentru realizarea acestui proiect, a fost necesară segmentarea acestor tumori din creier, folosind procesarea de imagine.

De asemenea, necesitatea unui proiect de acest fel este evidentă, datorită elementelor menționate în subcapitolul anterior în legătură cu apariția tot mai deasă a persoanelor asupra cărora tumorile cerebrale acționează. Detectarea automată a tumorilor cerebrale tinde să joace un rol remarcabil în domeniul medical. Tumorile cerebrale au o evoluție foarte rapidă, iar orice secundă câștigată în diagnosticarea lor, poate reprezenta salvarea unui pacient. Probabilitatea de supraviețuire având o tumoră crește prin utilizarea detectării automate. Din aceste considerente, realizarea acestei aplicații are ca scop ajutorarea medicilor neurologi în identificarea tumorilor cerebrale.

Obiectivul principal al lucrării este cel menționat mai sus – ajutorul adus medicilor neurologi în identificarea tumorilor cerebrale. Alte obiective ale lucrării sunt:

- Încurajarea oamenilor de a merge la controale medicale din timp, pentru a împiedica apariția posibilelor boli.
- Evidențierea numărului mare de oameni predispuși la dezvoltarea unor boli.
- Prevenirea dezvoltării tumorilor cerebrale în organismul oamenilor.

1.3. Imagistica medicală

Tehnologiile de imagistică medicală au avansat foarte mult în ultimul deceniu, deoarece, fiind nevoie de un număr foarte mare de imagini de calitate preluate de la pacienți în timpul actului medical, aparatura specifică acestui domeniu a fost adusă la performanțe din ce în ce mai bune.

Imagistica medicală este una din ramurile ingineriei biomedicale. Datele care rezultă în urma utilizării unuia dintre aparatele specifice sunt sub formă de măsurători sau înregistrări și sunt, de asemenea, sursa de informație pentru această ramură. Prin evaluarea, compararea și amplificarea datelor primite, imagistica medicală oferă un rezultat vizual pentru o măsurătoare secvențială, acest lucru explicându-se mai precis prin faptul că sunt însumate proceduri și tehnici utilizate în generarea unor imagini cu caracter medical. Utilizarea imagisticii este clinică, dar și medical științifică [4].

1.4. Structura temei

Lucrarea este împărțită în 7 capitole, care vor fi prezentate pe scurt, în continuare.

Capitolul I – este capitolul introductiv. În acest capitol se pune accent pe actualitatea proiectului de diplomă, pe obiectivele și pe scopul lucrării.

Capitolul II – este destinat prezentării creierului uman, prin anatomia lui, dar și prezentării tumorilor cerebrale care pot apărea în creierul uman.

Capitolul III – este axat pe partea de implementare. Acest capitol conține prezentarea limbajului de programare folosit în realizarea proiectului și a conceptelor celorlalte care țin de partea de software, atât de design, cât și de procesul de compilare.

Capitolul IV – conține partea cea mai importantă a lucrării: descrierea în detaliu a fiecărei părți utilizate în realizarea proiectului. Astfel, sunt surprinse informații legate de procesarea necesară imaginilor investigate, de modul în care s-a realizat segmentarea tumorilor cerebrale, dar și de felul în care s-a oferit un diagnostic corect și valid.

Capitolul V – prezintă interfața grafică a aplicației implementate, cu detalii oferite pentru fiecare funcționalitate a butoanelor și imaginilor.

Capitolul VI – este capitolul în care sunt prezentate concluziile și direcțiile viitoare ale proiectului.

Capitolul VII – reprezintă biografia lucrării cu toate sursele utilizate.

De asemenea, proiectul conține, la finalul acestuia, un rezumat realizat în limba română și unul în limba engleză.

2. CREIERUL UMAN

Creierul uman este atât cel mai important organ al corpului, cât și cel mai complex. Creierul trimite, primește și procesează toate informațiile de la organele noastre de simț. Împreună cu măduva spinării, acestea alcătuiesc împreună sistemul nervos central [5].

2.1. Anatomia creierului

Părțile cele mai importante ale creierului sunt prezentate în Fig. 2.1.

Creierul este alcătuit din mai multe segmente [6] reprezentate de:

- Trunchiul cerebral – alcătuit din mai multe segmente: bulbul, puntea și mezencefalul alcătuit din pedunculii cerebrali și coliculii cvadrigemeni. Trunchiul cerebral este alcătuit atât din substanță cenușie, cât și din substanță albă.
- Cerebelul – localizat posterior de trunchiul cerebral, fiind separat de acesta prin intermediul ventriculului IV. Conexiunile sale cu trunchiul cerebral se realizează prin intermediul celor trei perechi de pedunculi cerebeloși: superiori, mijlocii și inferiori. Superior acestuia se găsește tentorium cerebelli și lobii occipitali ai emisferelor cerebrale.
- Diencefalul - alcătuit din: talamus, hipotalamus, epitalamus, subtalamus și metotalamus.
- Emisferele cerebrale – localizate la nivelul lojei supratentoriale, superior față de trunchiul cerebral și tentorium cerebelli.

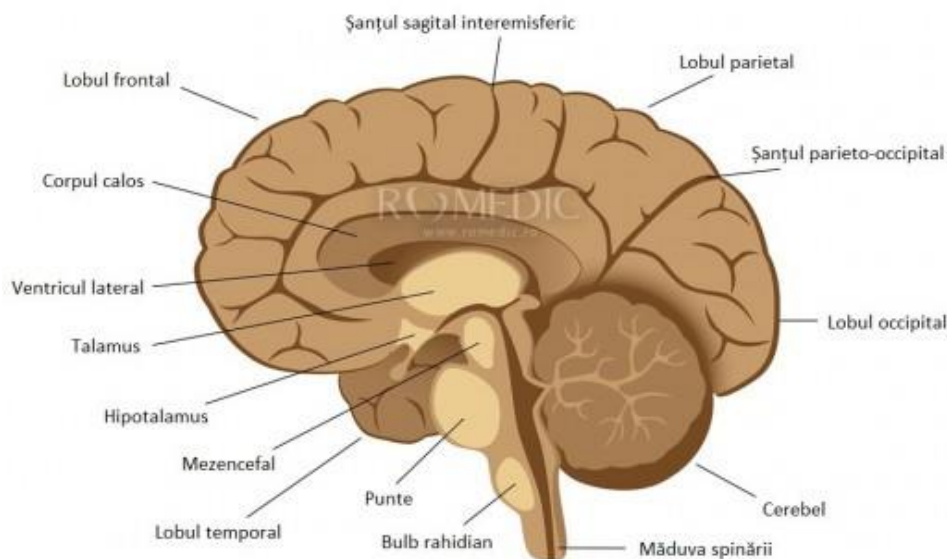


Fig. 1. Anatomia creierului uman.

2.2. Tumorile cerebrale

Tumora cerebrală este o masă de țesut sau o aglomerare de celule care se dezvoltă anormal la nivelul creierului sau în vecinătatea acestuia. Localizarea unei tumori și ritmul său de creștere determină modul în care aceasta va afecta funcția sistemului nervos.

Există mai multe tipuri de tumori cerebrale: benigne (necanceroase) sau maligne (canceroase). Pot exista tumori cerebrale care se dezvoltă în creier (primare) și tumori care au ca punct de plecare un alt organ, caz în care ele se numesc secundare sau metastatice.

Cauzele care duc la apariția tumorilor cerebrale, mai ales ale celor de tip malign, nu sunt pe deplin înțelese. Cercetările [7] au dovedit că au loc anumite modificări în structura celulelor cerebrale normale, care declanșează o multiplicare haotică a lor.

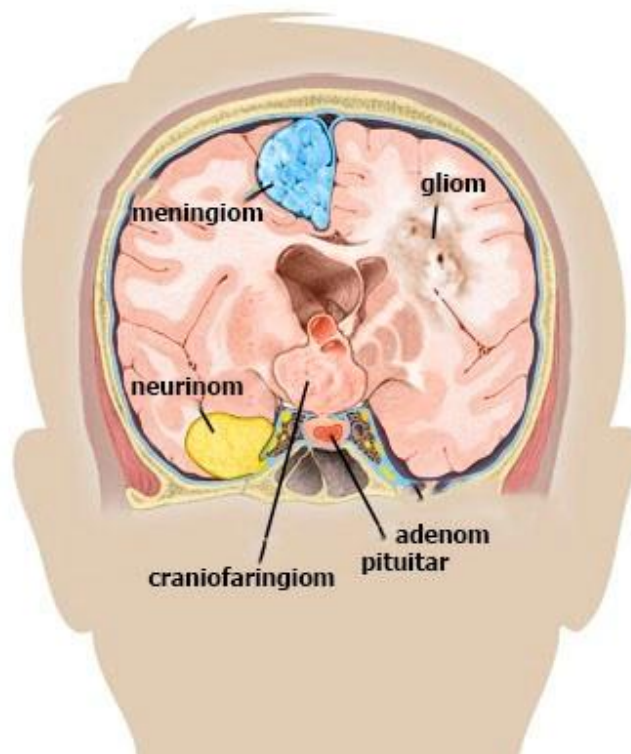


Fig. 2. Tipuri de tumori cerebrale.

3. MEDII ȘI CONCEPTE DE PROGRAMARE

3.1. Limbajul de programare C++

Limbajul de programare care este utilizat pentru implementarea codului sursă al aplicației este C++, inițial denumit „C cu clase”, fiindcă este o îmbunătățire a limbajului C. Acesta este unul dintre limbajele de nivel înalt cunoscute. Printre principalele caracteristici ale acestuia se numără:

- Programarea orientată pe obiecte – C++ este un limbaj de programare orientat pe obiecte, fapt care înseamnă că accentul este pus pe „obiecte” și nu pe manipularea acestora.
- Viteza de compilare – din punct de vedere al vitezei și timpului de compilare se poate specifica faptul că acest limbaj de programare este mult mai rapid decât majoritatea limbajelor utilizate.
- Suport pentru pointeri – C++ acceptă pointeri, care sunt utilizați frecvent în programare, dar care nu sunt disponibili în multe alte limbaje de programare.
- Suport bogat al bibliotecii – prin intermediul bibliotecii STL (Standard Template Library), C++ oferă funcții care ajută la scrierea rapidă a codului.

Datorită caracteristicilor specificate [8], C++ este unul dintre cele mai importante limbaje de programare, majoritatea celorlalte programe sau sisteme utilizate având la bază acest limbaj complex de programare.

3.2. Biblioteca OpenCV

OpenCV sau *Open Source Computer Vision Library* este o bibliotecă destinată sistemelor software pentru vedere computerizată și învățare automată. Aceasta permite achiziția, procesarea și reprezentarea imaginilor prin clasele de nivel înalt pe care le oferă. Biblioteca deține peste 2500 de algoritmi, ce pot fi utilizați în aplicații de procesare a imaginilor și vedere artificială, precum detectarea și recunoașterea facială, identificarea obiectelor, detectarea mișcării obiectelor în video. Biblioteca OpenCV este scrisă în limbajul de programare C++ și are interfețe pentru C++, Python, Java și Matlab [9].

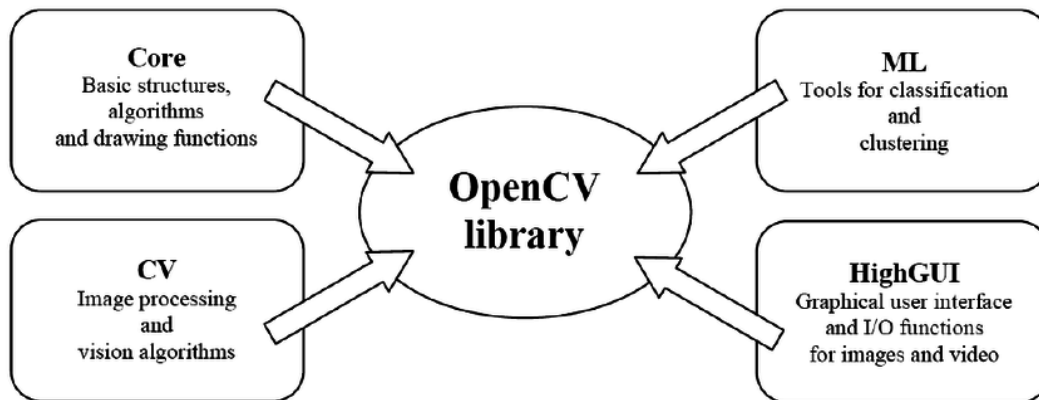


Fig. 3. Componentele OpenCV.

Componentele CV conțin procesarea de bază a imaginii și algoritmi de vedere artificială la nivel superior. Biblioteca ML (machine learning) include clasificatoare statistice și instrumente de grupare a datelor. HighGUI conține rutine I/O și funcții pentru stocarea și încărcarea imaginilor, iar Core conține structurile de date și conținutul de bază.

Figura 3.1. nu include componenta CvAux, care conține recunoașterea chipului încorporat HMM (Hidden Markov model), cât și algoritmi experimentali (segmentare, fundal/prim-plan).

În proiectul realizat s-a utilizat versiunea 4.2.0 a bibliotecii, cea mai actuală versiune fiind 4.5.5, acesta cuprinzând și o nouă parte, care este utilă pentru realizarea aplicațiilor cu învățare automată. Pentru ceea ce a fost dezvoltat în această lucrare de diplomă, învățarea automată nu este inclusă, proiectul axându-se pe procesarea imaginilor, utilizându-se algoritmi adecvați în ceea ce se urmărește a fi detectat, și anume: tumorile cerebrale prezente în creier. Astfel, OpenCV este utilizat de la începutul proiectului, atât în preprocesare, cât și în segmentarea tumorilor cerebrale.

3.3. Biblioteca Algorithms

Orice proiect care se dorește a fi apreciat are și părți originale. Pentru a îndeplini acest punct, în actualul proiect se remarcă realizarea unei biblioteci proprii, denumită „Algorithms”. S-a dorit implementarea unei astfel de biblioteci pentru a se pune accent pe învățare. Pe scurt, „Algorithms” este rezultatul învățării și înțelegerii a ceea ce se întâmplă în spatele unor funcții din biblioteca OpenCV, prezentată anterior. Bineînțeles, realizarea clasei și funcțiilor din bibliotecă au fost gândite și implementate pe baza celor din OpenCV.

Biblioteca „Algorithms” conține funcții de conversie a culorilor, precum metode de conversie în gri [10] și metode de aplicare a unui filtru de blur asupra unei imagini [11]. De asemenea, sunt implementate și codurile corespunzătoare calculării și creării histogramei uzuale și a celei cumulative, dar și a afișării lor.

În plus, utilizarea DLL-urilor este un plus adus computerului, deoarece ajută la reutilizarea codului, a modularizării acestuia, utilizează eficient memoria, iar spațiul de pe disc este redus prin utilizarea acestui tip de biblioteci. Din acest motiv, Algorithms este o bibliotecă de tip DLL. O bibliotecă de legături dinamice conține cod și date care pot fi folosite, în același timp, de mai multe programe. Unul dintre avantajele importante ale acesteia este că utilizează mai puține resurse.

3.4. Biblioteca OpenMP

Biblioteca OpenMP sau „*Open Multi-Processing*” acceptă multiprocesarea cu memorie partajată. Modelul de programare OpenMP este SMP (procesoare cu memorie partajată), adică atunci când se programează cu această bibliotecă, toate firele își împart memoria și datele [12].

Prin utilizarea acestei biblioteci, firul de execuție principal creează alte fire de execuție. Ele rulează același cod, dar în paralel, fiecare executând independent secțiunea paralelizată a codului. În momentul în care un fir termină de executat porțiunea sa de cod, acesta se alătură firului principal. Pentru a utiliza OpenMP, este nevoie să se utilizeze directive de preprocesare pentru controlul paralelismului, de exemplu *#pragma omp parallel for*.

În acest proiect este folosită biblioteca OpenMP, deoarece, datorită dimensiunii mari a imaginilor, parcurgerea acestora putea îngreuna timpul de execuție al programului. Sunt folosite directivele **#pragma omp parallel for**, precum și **#pragma omp parallel num_threads(omp_get_max_threads)**, parametrul din paranteze fiind utilizat pentru a seta automat numărul de procesoare pe care le folosește computerul.

3.5. Mediul de dezvoltare Qt

Qt nu este un limbaj de programare, ci este un mediu de dezvoltare scris, de asemenea, în limbajul de programare C++. Un preprocesor, MOC (*Meta-Object Compiler*), extinde limbajul C++ prin adăugarea de caracteristici precum semnale și sloturi. Platforma este utilizată pentru crearea aplicațiilor cu interfață grafică, dar și pentru programele care nu au interfață, precum serverele. Cu ajutorul Qt au fost realizate, de exemplu, browserul web Opera, Google Earth, Skype [13].

3.5.1. Qt Designer

Proiectarea și crearea de GUI se face folosind unul dintre instrumentele Qt, numit Qt Designer. Elementele grafice sunt atașate codului prin folosirea semnalelor Qt și a mecanismului de sloturi. Deși prin realizarea directă a GUI sunt setate anumite caracteristici, acestea pot fi modificate cu ușurință în cadrul codului. Totodată, în cazul în care se dorește să se folosească propriile widgeturi, ca de exemplu propria clasă de labeluri, Qt Designer permite promovarea acestora la propriile componente [14]. Utilizarea Qt Designer implică patru etape de bază:

- Alegerea propriei interfețe și a obiectelor dorite;
- Așezarea obiectelor acolo unde se dorește pe interfață;
- Conectarea semnalelor la sloturile corespunzătoare;
- Vizualizarea interfeței.

3.6. Platforma CMake

Gruparea într-un mod ordonat și curat a fișierelor componente ale unui program este foarte utilă pentru a găsi cu ușurință ceea ce este nevoie atunci când se dorește acest lucru. CMake este sistemul extensibil care poate realiza cele menționate. Astfel, acesta gestionează procesul de construire într-un sistem de operare și într-o manieră independentă de compilator.

Fișierele numite „CMakeLists.txt” sunt plasate în fiecare director sursă și sunt utilizate pentru generarea de fișiere de compilare standard care sunt folosite în mod obișnuit. CMake produce un mediu de construire nativ care va compila codul sursă, va crea biblioteci, va genera pachete și va putea construi executabile. Ceea ce este special la CMake este faptul că acesta generează un fișier cache. Când rulează acest program, de exemplu, el localizează fișiere, biblioteci și executabile, și poate întâmpina directive opționale de construire. Informațiile sunt adunate în memoria cache și pot fi modificate de către utilizator înainte de generarea fișierelor native, denumite „build” [15].

CMake este proiectat pentru a susține ierarhii de directoare complexe și aplicații dependente de mai multe biblioteci. De exemplu, în figura 3.2 este surprinsă componența directorului sursă al proiectului și utilizarea fișierului „CMakeLists.txt”.

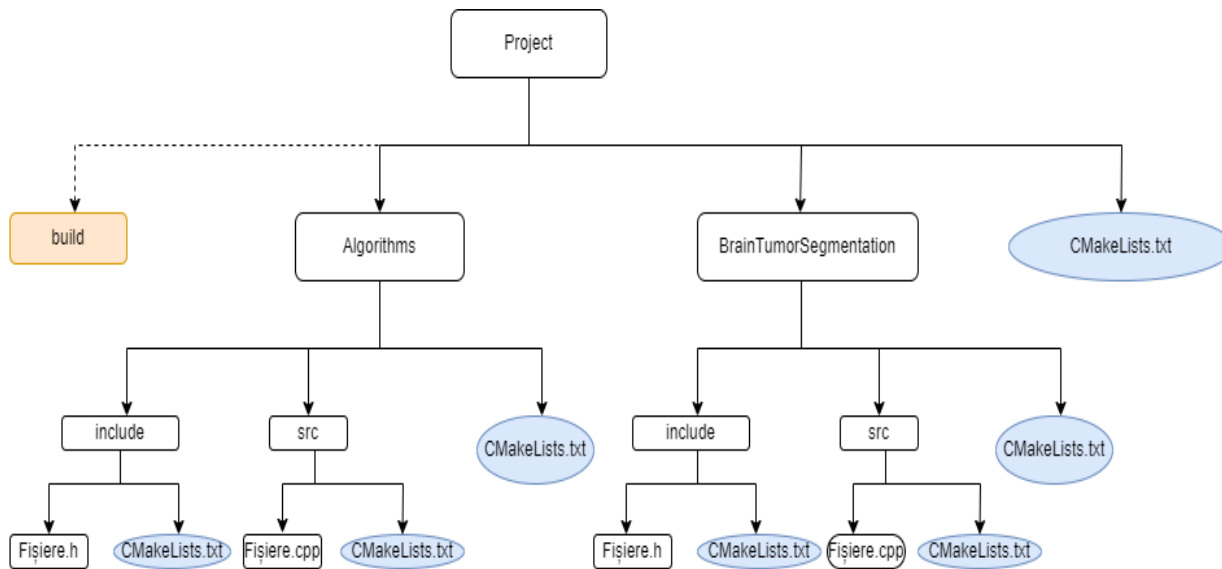


Fig. 4. Structura directorului sursă al proiectului.

În urma configurării și a generării proiectului, este creat directorul „build”, care cuprinde informațiile necesare rulării proiectului. Directorul „Algorithms” conține o bibliotecă dinamică proprie, realizată pe baza OpenCV. „BrainTumorSegmentation” reprezintă directorul în care implementarea proiectului a fost realizată în totalitate cu ajutorul celorlalte directoare prezentate.

4. SEGMENTAREA TUMORILOR CEREBRALE

Aplicația realizată a fost aleasă astfel încât rezultatul să fie de ajutor persoanelor specializate în acest sens, adică medicilor neurologi. Utilizarea acestora poate duce la ușurarea diagnosticării persoanelor, prin analiza imaginilor de tip RMN. Imagistica medicală ajută la îmbunătățirea detectării unor tumori cerebrale.

În decursul implementării programului, datorită documentării conforme asupra proiectului, s-a ajuns la concluzia că se vor urma trei pași principali, surprinși în figura 4.1 și prezentați în continuare.

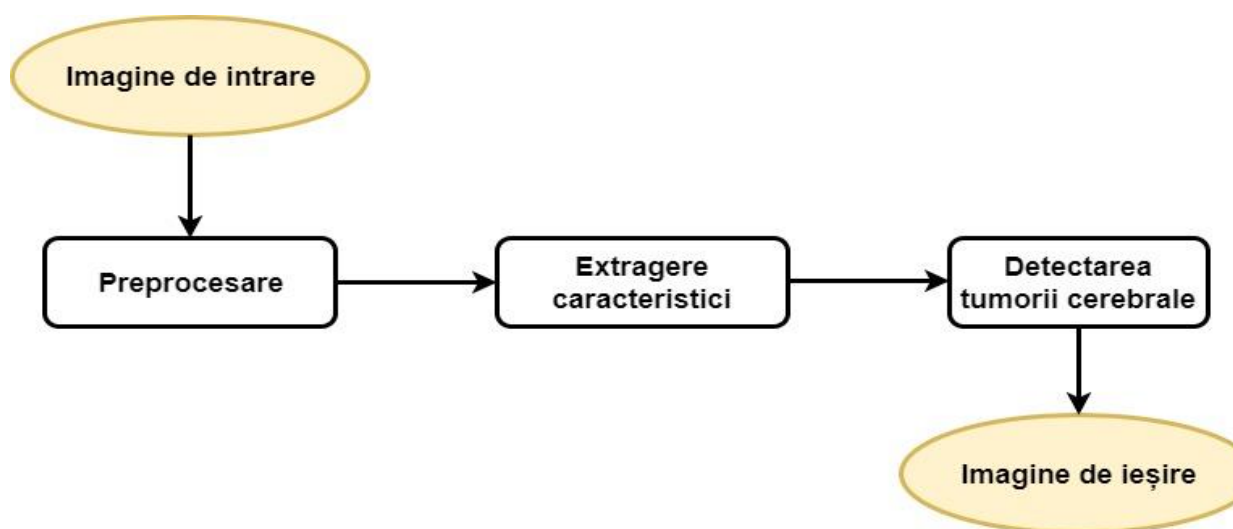


Fig. 5. Pași principali pentru segmentarea și diagnosticarea tumorilor cerebrale.

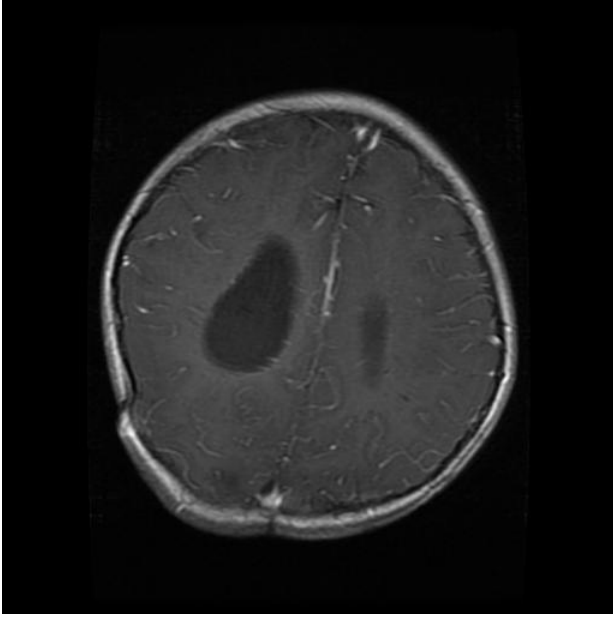
4.1. Setul de imagini utilizat

În metoda prin care s-a evaluat segmentarea tumorilor cerebrale s-au folosit imagini din setul de date public Brain Tumor MRI Dataset [16].

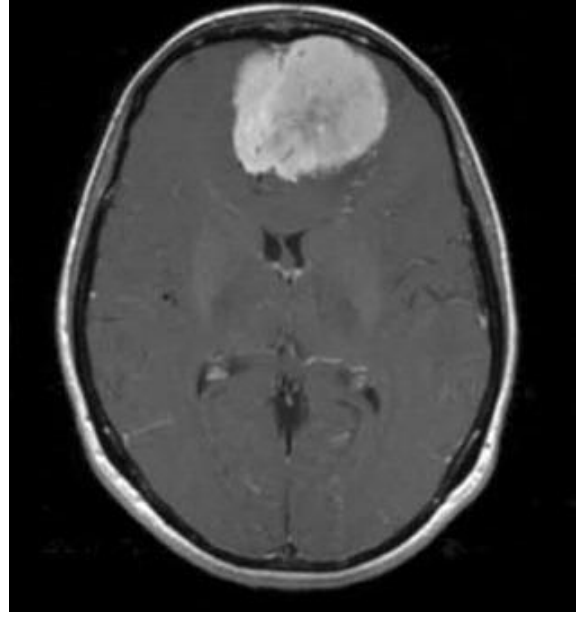
Setul de date conține, la momentul actual, patru seturi de imagini de tip RMN, din care unul cu 300 de imagini care conțin tumori de tip gliom, unul cu 306 imagini care conțin tumori de tip meningiom, unul cu 300 de imagini care conțin tumori pituitare și ultimul care conține 405 imagini fără tumori cerebrale.

Pentru aplicația noastră, vom folosi doar cele trei seturi de imagini care conțin tumori cerebrale, cel fără tumori cerebrale urmând a fi folosit în dezvoltarea ulterioară a aplicației.

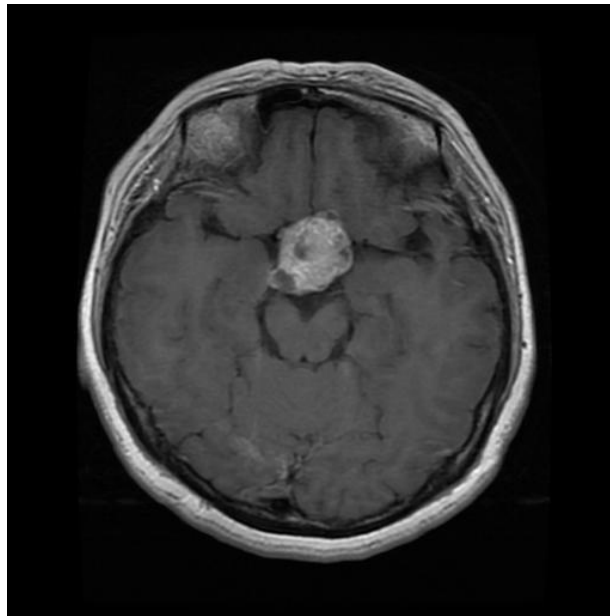
În figura 4.2 sunt surprinse 3 imagini din setul de date Brain Tumor MRI Dataset, câte una cu fiecare tip de tumoră cerebrală regăsit în setul de date.



(a)



(b)



(c)

Fig. 6. Exemple de imagini din setul de date Brain Tumor MRI Dataset: (a) Tumoră de tip gliom; (b) Tumoră de tip meningiom; (c) Tumoră pituitară.

4.2. Preprocesarea imaginilor

Primul pas al proiectului este constituit din preprocesare, adică pregătirea imaginii suport pentru a putea fi segmentată. Și în acest punct al realizării lucrării s-a ținut cont de câteva idei de bază, care au fost preluate din documentația de specialitate. Partea de preprocesare a fost realizată conform ideilor din articolul [17], care au fost de mare ajutor în acest pas al realizării lucrării. În urma preprocesării, imaginea rezultat este gata să fie supusă operațiilor necesare de segmentare.

Prin analiza imaginilor color RMN se observă că acestea au zgomot și, mai ales, prezența craniului în acestea poate reprezenta o mare problemă în partea de segmentare. Pentru reducerea imperfecțiunilor care se găsesc în majoritatea acestora și pentru generarea unor imagini convenabile în segmentarea tumorilor cerebrale, preprocesarea urmărește și ea anumiți sub-pași, surprinși în figura 4.3.

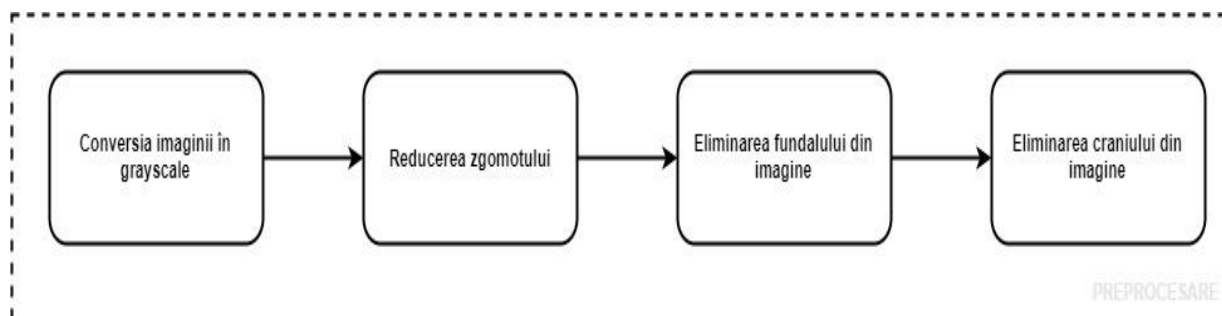


Fig. 7. Pași necesari pentru realizarea preprocesării.

4.2.1. Conversia imaginii în grayscale

Imaginile care se regăsesc în setul de date utilizat în dezvoltarea aplicației sunt imagini de tip RGB, adică imagini care conțin trei canale de culori: canalul roșu, canalul verde și canalul albastru. Pentru aplicația noastră, imaginile utilizate în segmentarea tumorilor cerebrale trebuie să fie în tonuri de gri, adică să conțină un singur canal de culoare. Astfel, asupra imaginii inițiale este aplicat un filtru în urma căruia imaginea va fi convertită la o imagine grayscale.

Formula pentru această transformare este următoarea:

$$p_{gri} = \frac{p_{roșu} + p_{verde} + p_{albastru}}{3}, \quad (4.1)$$

4.2.2. Reducerea zgomotului

Zgomotul într-o imagine este o variație aleatorie a intensității imaginii sau a informației unei culori. Acesta poate fi produs ca efect al naturii fotonice a luminii sau din cauza energiei termice din interiorul senzorilor de imagine.

Algoritmul de eliminare a zgomotului este procesul de eliminare sau de reducere al zgomotului dintr-o imagine. Algoritmii de eliminare a zgomotului reduc sau elimină vizibilitatea zgomotului prin omogenizarea întregii imagini, lăsând zone apropiate de limitele de contrast, însă aceste metode pot ascunde detalii fine ale contrastului scăzut.

Tipurile comune de zgomot care apar în imagini sunt:

- Zgomot de impuls;
- Zgomot aditiv;
- Zgomot multiplicativ.

Diferitele tipuri de zgomot au propriile lor caracteristici care le fac să se distingă de altele.

Zgomotul într-o imagine este un produs secundar nedorit al imaginii capturate, de aceea, pentru a obține informații corecte dintr-o imagine, zgomotul ar trebui să fie cât mai mic.

Pentru reducerea zgomotului din imaginile regăsite în setul de date am comparat patru algoritmi din biblioteca OpenCV, implementați individual pentru aplicația dezvoltată:

- Filtrul mediu;
- Filtrul median;
- Filtrul gaussian;
- Filtrul bilateral.

Filtrul mediu

Filtrul mediu constă în înlocuirea elementului central din masca aleasă cu media tuturor pixelilor din zona nucleului. Masca sau filtrul de dimensiune 3x3 arată astfel:

$$K = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (4.2)$$

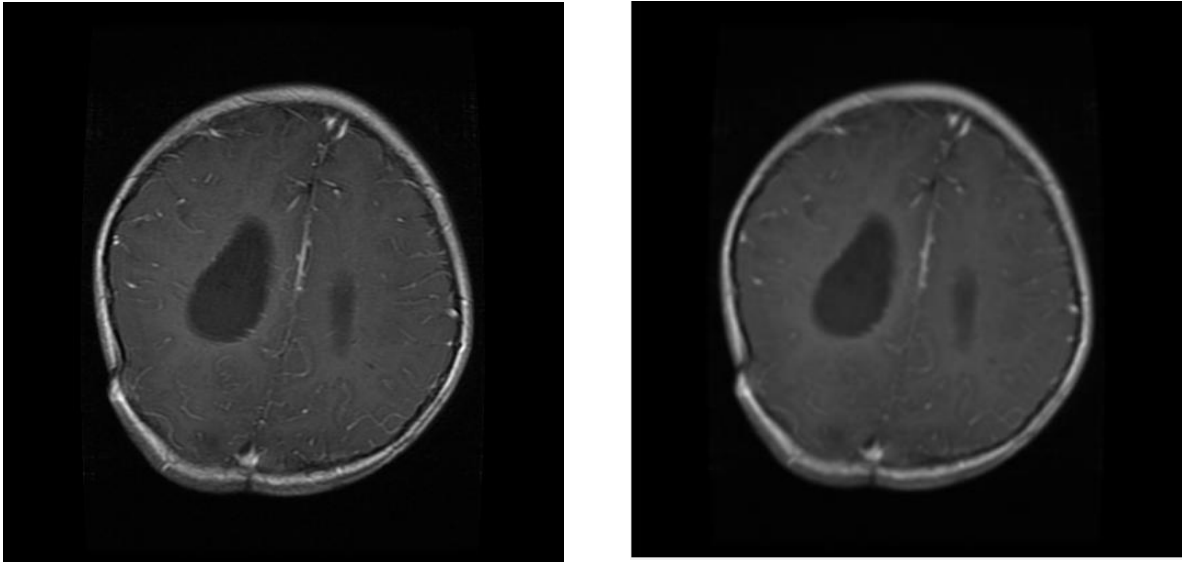


Fig. 8. Imaginea inițială (stânga); Imaginea după aplicarea filtrului mediu cu un nucleu de 5x5 (dreapta).

Filtrul median

În acest algoritm, elementul central al nucleului este înlocuit cu valoarea mediană a celorlalți pixeli din aria nucleului. Acest procedeu este foarte folositor împotriva zgomotului de tip salt-and-pepper dintr-o imagine.

În implementarea acestui algoritm nu e nevoie de declararea unei mărimi a unui nucleu, deoarece implementarea a fost făcută astfel încât algoritmul să fie adaptiv.

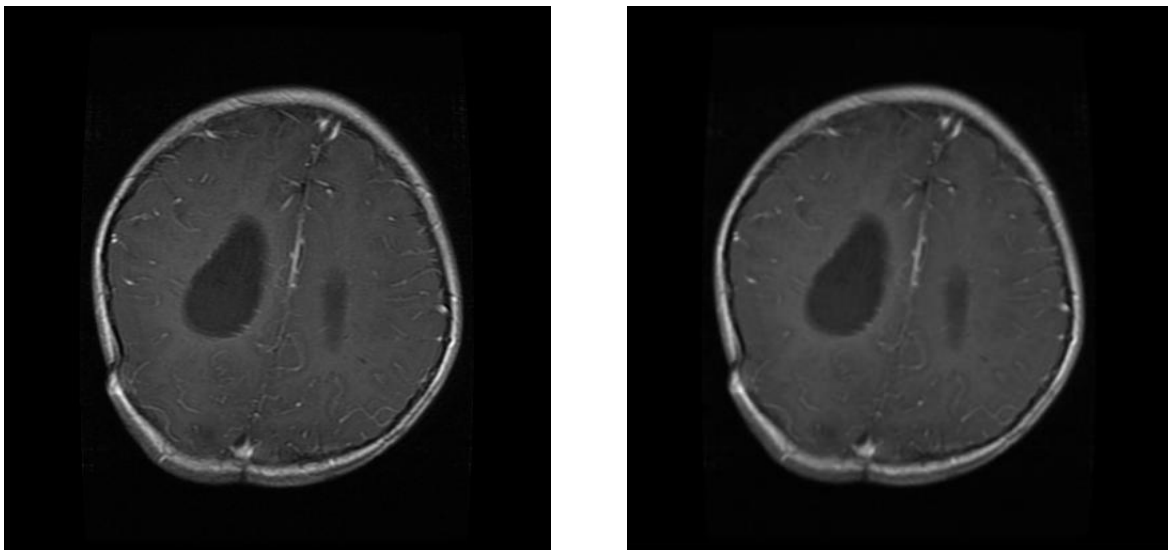


Fig. 9. Imaginea inițială (stânga); Imaginea după aplicarea filtrului median (dreapta).

Filtrul gaussian

Filtrul gaussian folosește, în locul unui filtru normalizat, un nucleu de tip Gaussian. Acest filtru este dat de formula:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}, \quad (4.3)$$

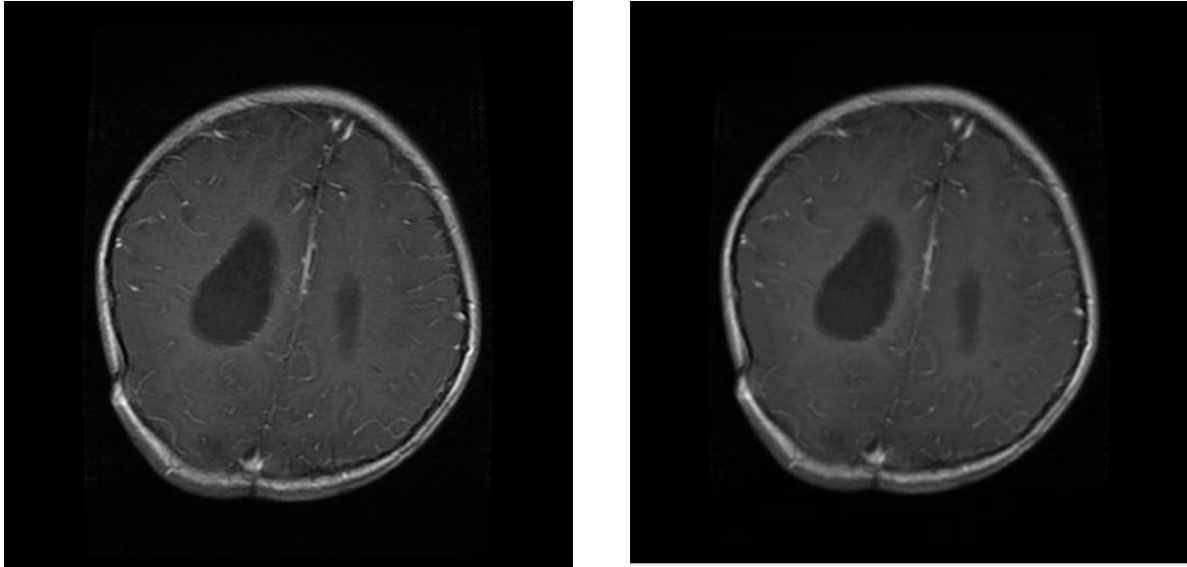


Fig. 10. Imaginea inițială (stânga); Imaginea după aplicarea filtrului gaussian cu un nucleu de 5x5 (dreapta).

Filtrul bilateral

Filtrul bilateral folosește un filtru gaussian pentru spațiu, însă folosește și un al doilea filtru gaussian pentru intensitate. Astfel, primul filtru se asigură că doar pixeli apropiați se iau în considerare, iar al doilea filtru garantează că doar pixelii cu intensități asemănătoare sunt luați în considerare. În acest mod, marginile sunt păstrate, deoarece pixelii de pe margini vor avea variații mari ale intensității.

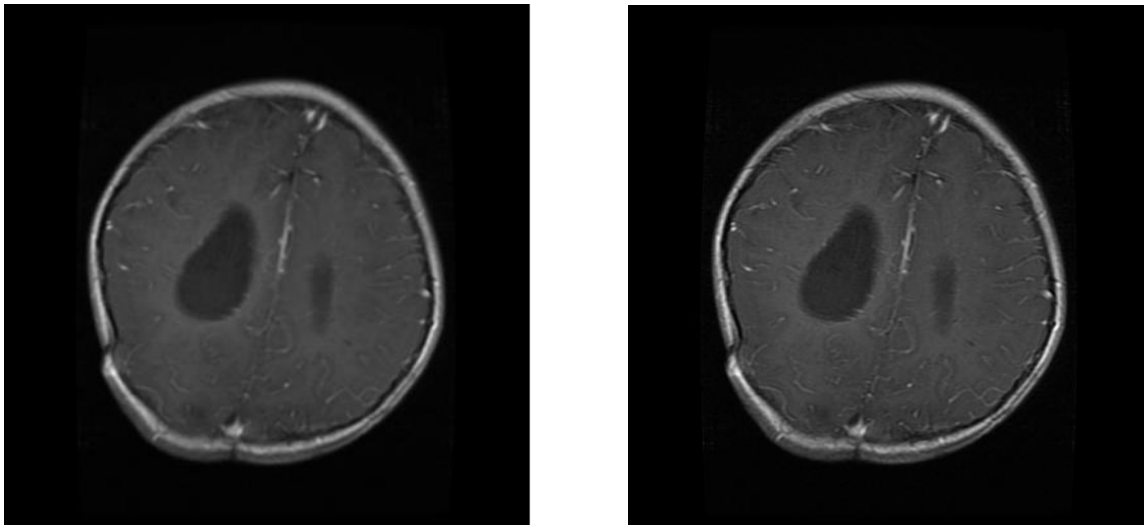


Fig. 11. Imaginea inițială (stânga); Imaginea după aplicarea filtrului bilateral cu un nucleu de 5x5 (dreapta).

Pentru măsurarea nivelului de zgomot dintr-o imagine am utilizat câteva idei, conform articolului [18], pentru calcularea unei valori σ corespunzătoare nivelului de zgomot al imaginii.

Primul pas al acestei metode este suprimarea structurii imaginii prin folosirea următorului operator Laplacian:

$$N = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \quad (4.4)$$

Apoi, valoarea σ poate fi calculată astfel:

$$\sigma = \sqrt{\frac{\pi}{2}} \frac{1}{6(W-2)(H-2)} \sum_{image I} |I(x, y) * N|, \quad (4.5)$$

unde W și H sunt lățimea, respectiv înălțimea imaginii.

După aplicarea acestei metode asupra întregului set de date, se obțin următoarele rezultate medii:

		Rezultate medii nivel de zgomot										
		Initial	Filtru mediu			Filtru median	Filtru gaussian			Filtru bilateral		
Mărimea nucleului			3x3	5x5	7x7	Adaptiv	3x3	5x5	7x7	3x3	5x5	7x7
		0.5591	0.1591669	0.1135	0.1015	0.49791984	0.1781	0.1303	0.1082	0.4655	0.4713	0.4621

Fig. 12. Rezultate medii ale nivelului de zgomot.

Putem observa că valorile medii sunt cele mai mici în cazul aplicării filtrului mediu folosind mărimea nucleului de 7, însă nici valorile filtrului gaussian nu sunt foarte mari și ar putea fi utilizate.

Aplicația dezvoltată urmează a fi rulată de un număr mare de ori în cabinetele medicilor neurologi, de aceea ar trebui ales un algoritm care duce la o execuție mai rapidă.

Timpul de rulare al unui algoritm depinde de următorii factori:

- Datele de intrare;
- Calitatea codului generat de compilator;
- Natura și viteza de execuție a instrucțiunilor programului;
- Complexitatea algoritmului care stă la baza programului.

Astfel, au fost efectuate și măsurători asupra timpului de execuție al fiecărui filtru, pentru fiecare mărime a nucleului.

Rezultatele obținute sunt următoarele:

		Timpul mediu de execuție (in microsecunde)									
		Filtru Average			Filtru Median	Filtru Gaussian			Filtru Bilateral		
Marimea nucleului		3x3	5x5	7x7	Adaptiv	3x3	5x5	7x7	3x3	5x5	7x7
		638.5496	797.9792	432.7212	1718.844369	140.055	173.0685	217.9852	379.4229	938.9376	3726.232
		Timpul median de execuție (in microsecunde)									
		Filtru Average			Filtru Median	Filtru Gaussian			Filtru Bilateral		
Marimea nucleului		3x3	5x5	7x7	Adaptiv	3x3	5x5	7x7	3x3	5x5	7x7
		691	915	479	14766	121	146	210	387	940	3532
		Deviatia standard									
		Filtru Average			Filtru Median	Filtru Gaussian			Filtru Bilateral		
Marimea nucleului		3x3	5x5	7x7	Adaptiv	3x3	5x5	7x7	3x3	5x5	7x7
		356	429.025	246.2986	7105.697186	103.6087	84.59617	87.17496	140.1814	273.9766	1159.057

Fig. 13. Timpul mediu de execuție, timpul median de execuție și deviația standard a algoritmilor de reducere a zgomotului din imagini.

Observăm din figura 4.9 că rezultatele cele mai bune raportate la timpul de execuție sunt obținute de filtrul gaussian.

Având în vedere atât rezultatele observate în Fig. 4.8, cât și rezultatele din figura 4.9, în aplicație, pentru reducerea zgomotului de fundal, va fi utilizat filtrul gaussian cu o mărime a nucleului de 5x5.

4.2.3. Eliminarea fundalului din imagine

Iluminarea neuniformă datorată de variația pixelilor de fundal este un dezavantaj atunci când se dorește o procesare de imagine corectă. Intensitățile pixelilor de fundal pot fi diferite, astfel valorile acestora pot fi ușor comparate cu pixelii din interiorul creierului. Acesta este un impediment des întâlnit în realizarea unei segmentări corecte a tumorilor cerebrale.

Prin eliminarea fundalului din imagine se înțelege că pixelii de fundal sunt uniformizați, adică se ajunge la o valoare a intensității lor care să nu se influențeze segmentarea tumorilor cerebrale.

Pentru a ajunge la imaginea dorită, în primul rând vom calcula valoarea de prag a pixelilor imaginii folosind metoda Otsu [19]. Metodele uzuale de calcul al valorii de prag folosesc o valoare arbitrară a acestei valori de prag. În contrast cu acestea, metoda Otsu evită alegerea arbitrară a unei valori și o calculează în mod automat.

În continuare, asupra imaginii obținute după aplicarea metodei Otsu, vom aplica o operație de deschidere morfologică, pentru a fi siguri că toți pixelii din interiorul creierului sunt în masca astfel obținută. În cele din urmă, imaginea cu fundalul eliminat se obține printr-o diferență dintre pixelii celor două imagini, și anume: cea obținută după aplicarea pașilor anteriori de preprocesare și masca rezultată în urma aplicării metodei Otsu și a operației de deschidere morfologică.

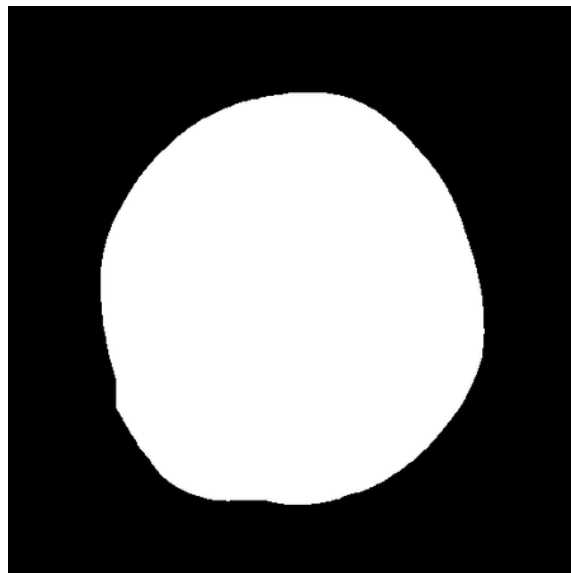


Fig. 14. Masca zonei de interes din imaginea de intrare.

4.2.4. Eliminarea craniului din imagine

Imaginile obținute prin tehnica rezonanței magnetice, indiferent din ce unghi al craniului uman sunt realizate, au o mare problemă, și anume faptul că, pe lângă zona creierului, în ele apare și conturul cranian, contur care, de cele mai multe ori, este foarte proeminent în aceste imagini.

Pentru a putea realiza o segmentare corectă a tumorilor cerebrale, este nevoie de eliminarea acestui contur cranian din imaginile pregătite pentru segmentare. Intensitățile pixelilor din acest contur cranian sunt foarte mari și pot fi cu ușurință comparate cu intensitățile pixelilor din interiorul zonei creierului.

Pentru realizarea acestui pas al preprocesării imaginii, am implementat patru algoritmi pentru eliminarea craniului din imagine, pentru a compara rezultatele obținute și a putea alege algoritmul optim pentru aplicația dezvoltată:

- Algoritmul de eliminare a craniului din imagine folosind o valoare de prag calculată în mod dinamic;
- Algoritmul de eliminare a craniului din imagine folosind o fereastră adaptivă pentru calcularea valorii de prag;
- Algoritmul de eliminare a craniului din imagine folosind operații morfologice și măști ale imaginii;
- Algoritmul de eliminare a craniului din imagine folosind algoritmul K-Means.

Algoritmul de eliminare a craniului din imagine folosind o valoare de prag calculată dinamic.

Primul pas în implementarea acestui algoritm este de a aplica segmentarea imaginii după un anumit prag (engl. *thresholding*). Pentru pasul prezentat, este nevoie să se găsească o valoare a pragului, corespunzătoare imaginii. Valoarea nu este aleasă aleatoriu, ea fiind determinată pe baza pixelilor din imaginea inițială.

Pentru găsirea acestor valori optime de prag este utilizat algoritmul triunghiului. Acesta este un algoritm automat de partiționare și este considerat ca fiind o metodă bazată pe formă, pentru că analizează structura histogramei. El funcționează în câteva etape. Primul pas este să se creeze histograma, iar pe baza acesteia, histograma cumulativă, care se determină după formula 4.6.

$$h_{cum}[u] = \sum_{i=0}^u h[i], \quad (4.6)$$

unde, h_{cum} este histograma cumulativă, h este histograma imaginii, iar u reprezintă nivelul de gri. A doua etapă este de a se trasa o linie între valoarea minimă și cea maximă ale histogramei. Al treilea pas se rezumă la calcularea distanței dintre linia găsită și vârful liniei corespunzătoare

fiecărui nivel de gri. Ultima etapă constă în alegerea valorii de prag, care este reprezentată de nivelul de gri corespunzător distanței maxime găsite la pasul anterior [20].

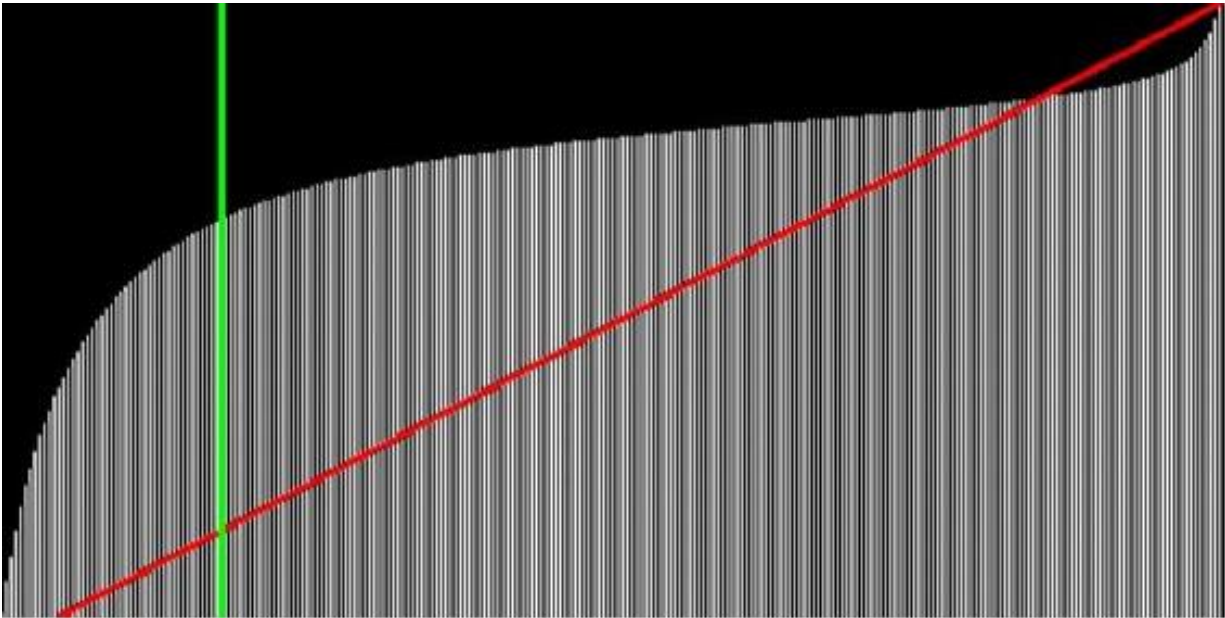


Fig. 15. Histograma cumulativă și evidențierea valorii de prag.

Odată aflată valoarea optimă de prag, se poate aplica partiționarea, astfel:

$$img_{th}(x, y) = \begin{cases} 0, & \text{dacă } i(x, y) < p \\ 255, & \text{dacă } i(x, y) \geq p' \end{cases} \quad (4.7)$$

unde, img_{th} reprezintă imaginea care rezultă în urma partiționării (prezentată în figura 4.12.b), $i(x, y)$ este intensitatea pixelului de la poziția (x, y) , iar p este valoarea de prag.

Prin efectuarea unei diferențe între pixelii imaginii obținute folosind pașii anteriori ai preprocesării și imaginea rezultată în urma partiționării, obținem imaginea finală, cu craniul eliminat din aceasta.

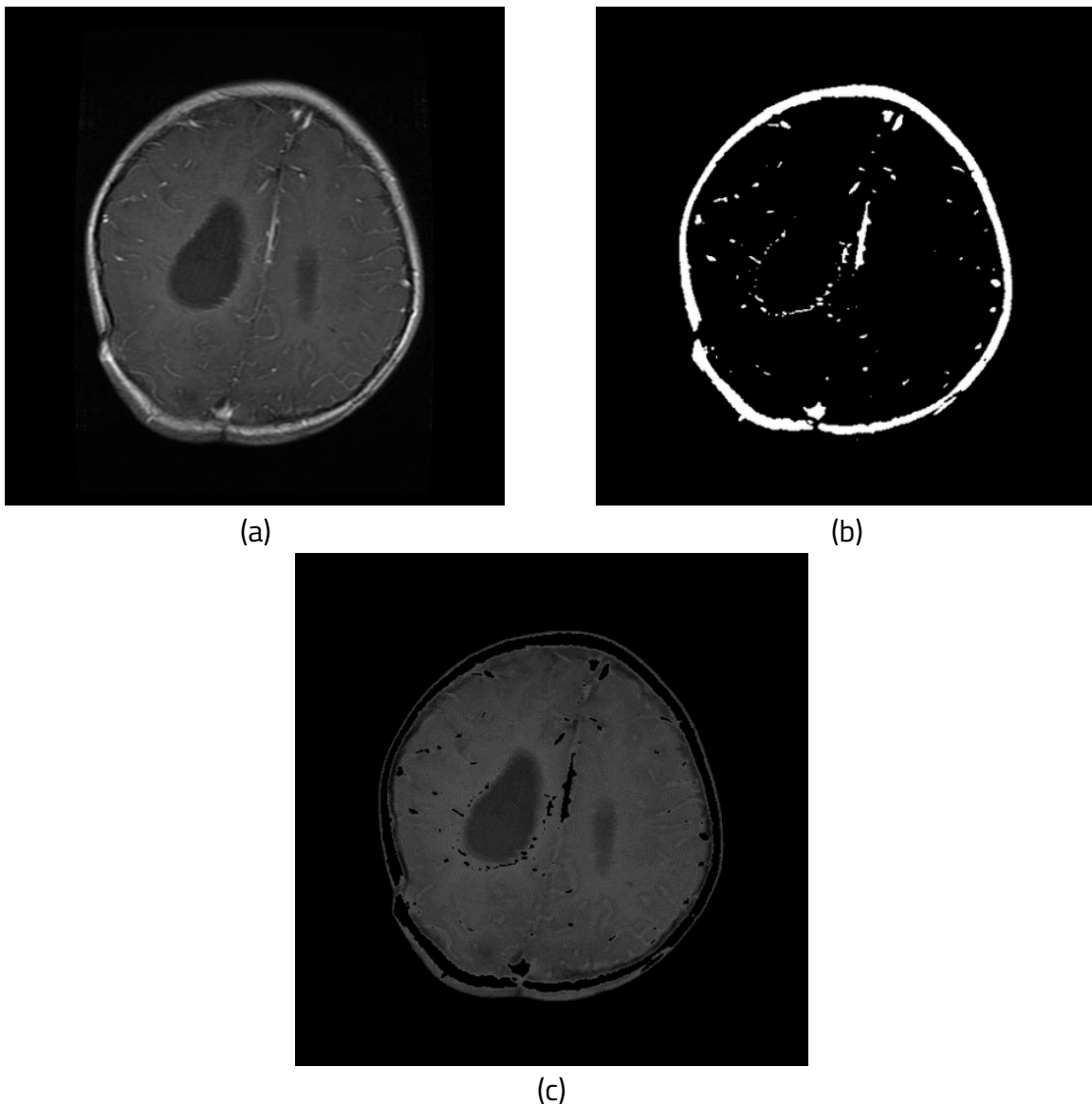


Fig. 16. (a) Imaginea inițială; (b) Imaginea rezultată în urma partiționării; (c) Imaginea după eliminarea craniului.

Algoritmul de eliminare a craniului din imagine folosind o fereastră adaptivă pentru calcularea valorii de prag.

Implementarea algoritmului a fost făcută conform ideilor din articolul [21]. Astfel, primul pas al algoritmului este ca, pentru fiecare pixel al imaginii, să se calculeze suma nivelelor de gri ale pixelilor aflați într-o fereastră dreptunghiulară definită de pixelul curent și pixelul din colțul din stânga de sus al imaginii. Următoarea etapă a algoritmului este ca, pentru fiecare pixel, să creștem mărimea ferestrei până când produsul dintre deviația standard al ferestrei curente și logaritm din mărimea ferestrei este mai mic decât valoarea calculată pentru

fereastra anterioară. Apoi este nevoie să calculăm media valorilor pixelilor din fereastra curentă și să calculăm valoarea locală de prag folosind ecuația:

$$T = mean, \quad (4.8)$$

unde T este valoarea de prag, iar $mean$ este valoarea medie a pixelilor din fereastra curentă.

Ultimul pas este, asemănător formulei 4.7, să setăm valoarea pixelilor din imagine la 0, dacă valoarea lor inițială este mai mică decât valoarea de prag calculată și la 255, dacă valoarea inițială este mai mare sau egală decât valoarea de prag calculată de algoritm.

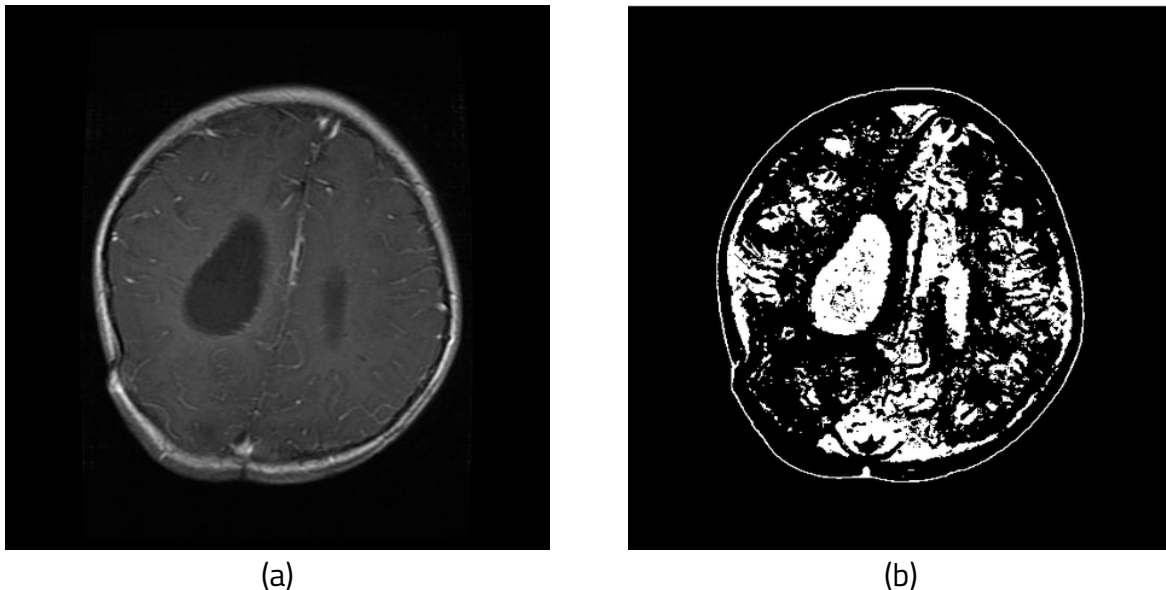


Fig. 17. (a) Imaginea inițială; (b) Imaginea după eliminarea craniului.

Algoritmul de eliminare a craniului din imagine folosind operații morfologice și măști ale imaginii.

Primul pas în implementarea algoritmului este calcularea gradientului imaginii de intrare, pentru a distinge conturul din imagine. Algoritmul continuă prin calcularea unei valori de prag pentru imaginea gradient obținută folosind algoritmul triunghiului descris anterior. Următorul pas este partiționarea imaginii folosind valoarea de prag calculată.

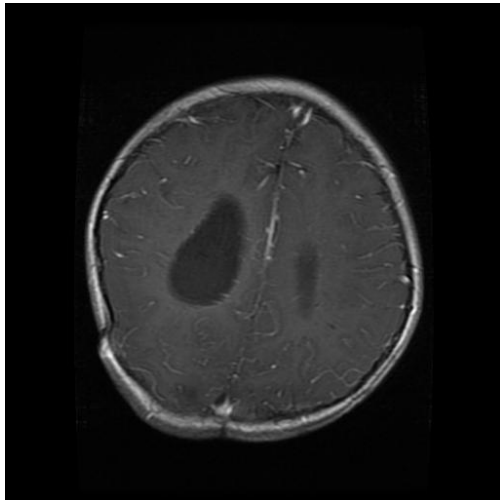
Folosind noua imagine partiționată, se calculează valoarea medie a pixelilor și deviația standard a imaginii, apoi, imaginea de intrare este partiționată folosind ca valoare de prag o valoare calculată conform ecuației:

$$T = mean - 1.5 * stdDev, \quad (4.9)$$

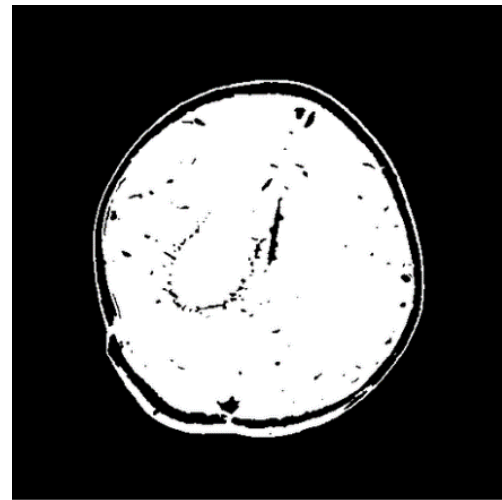
unde $mean$ reprezintă media pixelilor imaginii, iar $stdDev$ deviația standard a imaginii.

În urma acestei prime partiționări, obținem o primă mască a imaginii ce va fi de folos în continuare.

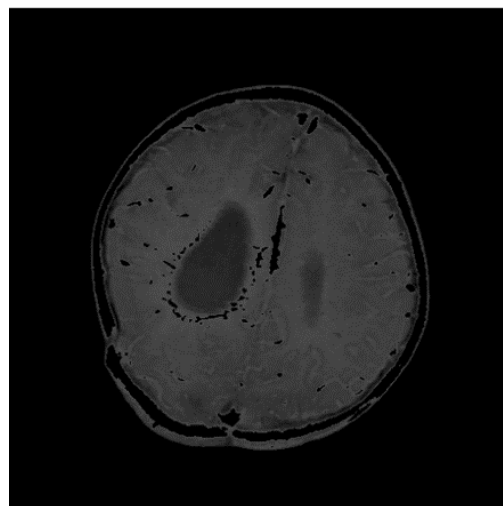
Următoarea etapă a algoritmului este o nouă partiționare a imaginii inițiale, folosind ca și valoare de prag, de această dată, valoarea medie a pixelilor din imagine. Astfel obținem o a doua mască. Prin efectuarea diferenței dintre pixelii primei măști calculate și a celei de-a doua, obținem o nouă mască, pe care, în ultimul pas al algoritmului, o folosim pentru a obține imaginea finală cu craniul eliminat.



(a)



(b)



(c)

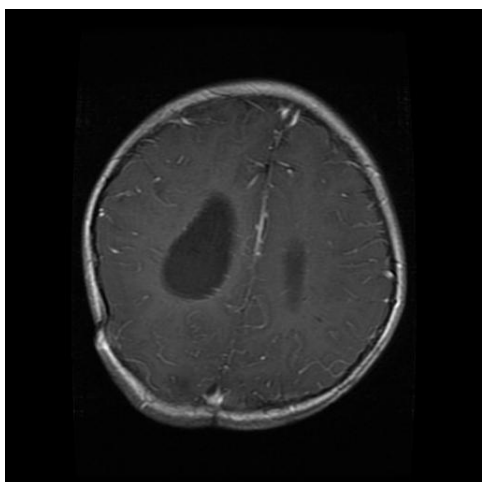
Fig. 18. (a) Imaginea inițială; (b) Mască obținută prin diferența celor 2 măști calculate în algoritmul; (c) Imaginea după eliminarea craniului.

Algoritmul de eliminare a craniului din imagine folosind algoritmul K-Means.

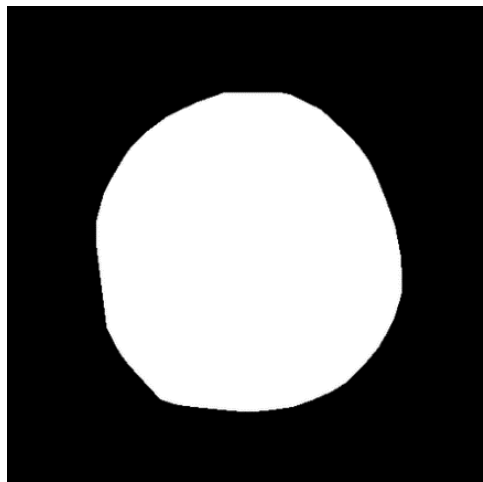
Implementarea acestui algoritm are ca punct de plecare imaginea obținută folosind algoritmul descris mai sus, asupra căreia a fost aplicată o operație morfologică de deschidere (engl. *opening*) pentru a scăpa de zonele din imagine unde zona craniului este unită de zona creierului. Folosind imaginea obținută, aplicăm algoritmul K-Means [22] pentru a separa imaginea în trei zone distincte: zona de background, zona craniului și zona creierului. Utilizând această separare a zonelor distincte, putem elimina din imagine zona găsită de algoritmul K-Means ca fiind zona craniului.

Totuși, pentru a ne asigura ca nu vom mai avea părți din craniu în imaginea finală, aplicăm algoritmul de găsim a componentelor conexe dintr-o imagine [23] pentru a extrage componenta cu aria cea mai mare, componentă care reprezintă zona din imagine care cuprinde encefalul. După extragerea acestei componente, se estimează conturul acesteia, apoi se calculează coordonatele centrului de greutate al poligonului obținut din conturul componentei extrase.

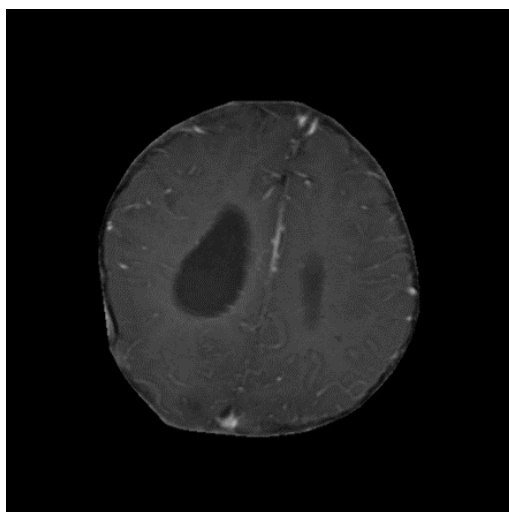
După determinarea coordonatelor centrului de greutate al poligonului, se încearcă aproximarea unui poligon care să cuprindă doar zona creierului din imagine, excluzând zona craniului. Primul pas al acestui proces este calcularea distanței maxime dintre centrul de greutate calculat anterior și poligonul obținut prin conturarea componentei extrase din imagine. Având toate aceste informații, se poate estima un poligon care să cuprindă zona din imagine unde se află creierul, iar mai apoi, folosind acest poligon, zona de interes să fie extrasă din imaginea de intrare.



(a)



(b)



(c)

Fig. 19. (a) Imaginea inițială; (b) Poligonul estimat care cuprinde zona din imagine unde se află creierul; (c) Imaginea după eliminarea craniului.

În urma implementării celor patru algoritmi descriși anterior, putem observa din figurile 4.12.c, 4.13.b și 4.14.c că în imaginea finală încă există un contur fin al craniului, contur care poate afecta segmentarea tumorilor cerebrale. Astfel, cele mai bune rezultate se obțin folosind algoritmul de eliminare a craniului din imagine utilizând algoritmul K-Means, algoritm care va fi folosit în continuare în dezvoltarea aplicației.

4.3. Extragerea caracteristicilor

Identificarea tumorilor cerebrale este o sarcină dificilă din cauza faptului că imaginile prezintă ori un contrast slab, ori anomalii în fundalul lor. Majoritatea lucrărilor care urmăresc acest subiect utilizează, pe lângă o puternică procesare a imaginii, inteligența artificială prin învățarea automată. De asemenea, există și anumite probleme întâmpinate până acum în segmentarea tumorilor, precum realizarea unei segmentări manuale, semi-automatizate sau automatizate, dar și apariția erorilor de segmentare atunci când setul de date este schimbat. Scopul segmentării tumorilor cerebrale este de a evidenția întreaga arie a tumorii, celelalte componente ale creierului fiind ignorate.

Principala parte pe care se axează proiectul este extragerea caracteristicilor. Aceasta presupune segmentarea tumorilor cerebrale și este, de asemenea, și ținta pentru care această aplicație a fost realizată. Există numeroase metode prin care se pot extrage caracteristici dintr-o imagine, precum segmentarea după un anumit prag (engl. *thresholding*), gruparea K-means (engl. *K-means clustering*), metoda Otsu, threshold adaptiv etc. Unele dintre ele au fost chiar încercate în proiect, dar, neavând rezultate bune, s-a optat pentru utilizarea algoritmului de grupare K-means, urmat de aplicarea unei operații morfologice de deschidere, iar în final o identificare a ariei tumorale din imagine folosind metoda componentelor conexe (engl. *connected components*).

4.3.1. Algoritmul de grupare K-means

Gruparea (engl. *clustering*), o tehnică introdusă în anul 1932 de H.E. Driver și A.L. Kroeber [24], este un tip de învățare nesupervizată unde referințele trebuie extrase din seturi de date neetichetate. Această tehnică presupune o divizare a populației în mai multe grupuri, astfel încât indivizii din același grup sunt similari. Pe scurt, tehnica reprezintă o colecție de obiecte bazate pe similarități și diferențe.

Există numeroși algoritmi pentru grupare și înainte de a alege un algoritm pentru utilizare, este important să fie ales cel mai potrivit pentru cazul de utilizare. Trebuie, de asemenea, luat în considerare faptul că alegerea unui algoritm pentru grupare depinde inclusiv de mărimea setului de date.

Seturile de date pot conține milioane de înregistrări și nu toți algoritmi sunt eficienți în aceste cazuri. Algoritmul de grupare K-means [25] este un algoritm popular și este, totodată, și eficient, având o complexitate de liniară.

Algoritmul de grupare K-means este un algoritm bazat pe gruparea în funcție de niște puncte numite centroizi, unde se calculează distanța dintre fiecare punct de date și un centroid pentru atribuirea punctului de date într-un grup. Este un proces iterativ de atribuire a fiecărui punct de date unui anumit grup, astfel datele fiind grupate pe baza similarității. Obiectivul este

de a minimiza suma distanțelor dintre punctele de date și centroidul grupului pentru identificarea corectă a grupului de care ar trebui să aparțină fiecare punct de date.

Algoritmul de grupare K-means funcționează după următorii pași:

- Primul pas este definirea a unui număr de K grupuri în care datele să fie grupate.
- Al doilea pas este inițializarea centroizilor. La începutul algoritmului, se selectează puncte de date aleatorii care sunt definite ca centroizi ai fiecărui grup.
- Al treilea pas constă în plasarea fiecărui punct de date X_n în grupul care conține cel mai apropiat centroid C_k . Pentru acest lucru se va calcula distanța dintre punctul de date X și centroidul C folosind formula distanței Euclidiene:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (4.10)$$

și apoi prin alegerea grupului unde distanța dintre punctul de date și centroid este minimă.

- Al patrulea pas este reinițializarea centroizilor prin calcularea mediei tuturor punctelor de date al fiecărui grup.

$$C_i = \frac{1}{|N_i|} \sum x_i, \quad (4.11)$$

- Ultimul pas constă în repetarea pașilor 3 și 4 până când centroizii sunt plasați corespunzător și punctele de date nu își mai schimbă grupurile din care fac parte.

Pentru aplicație, algoritmul de grupare K-means a fost utilizat pentru separarea în trei grupuri distincte a pixelilor imaginii de intrare: un grup pentru pixelii din background, un grup pentru pixelii imaginii care nu ne sunt folositori și un ultim grup pentru pixelii din imagine care pot compune tumora. Astfel, asupra imaginii de intrare, reprezentată de imaginea obținută după toți pașii preprocesării, se aplică algoritmul de grupare K-means din OpenCV cu un număr $k = 3$, care reprezintă numărul de grupuri diferite care trebuie separate din imagine.

Algoritmul continuă prin separarea din imaginea de intrare a pixelilor care se găsesc în grupul cu pixelii care pot compune tumora, obținând astfel o imagine nouă care conține doar acești pixeli.

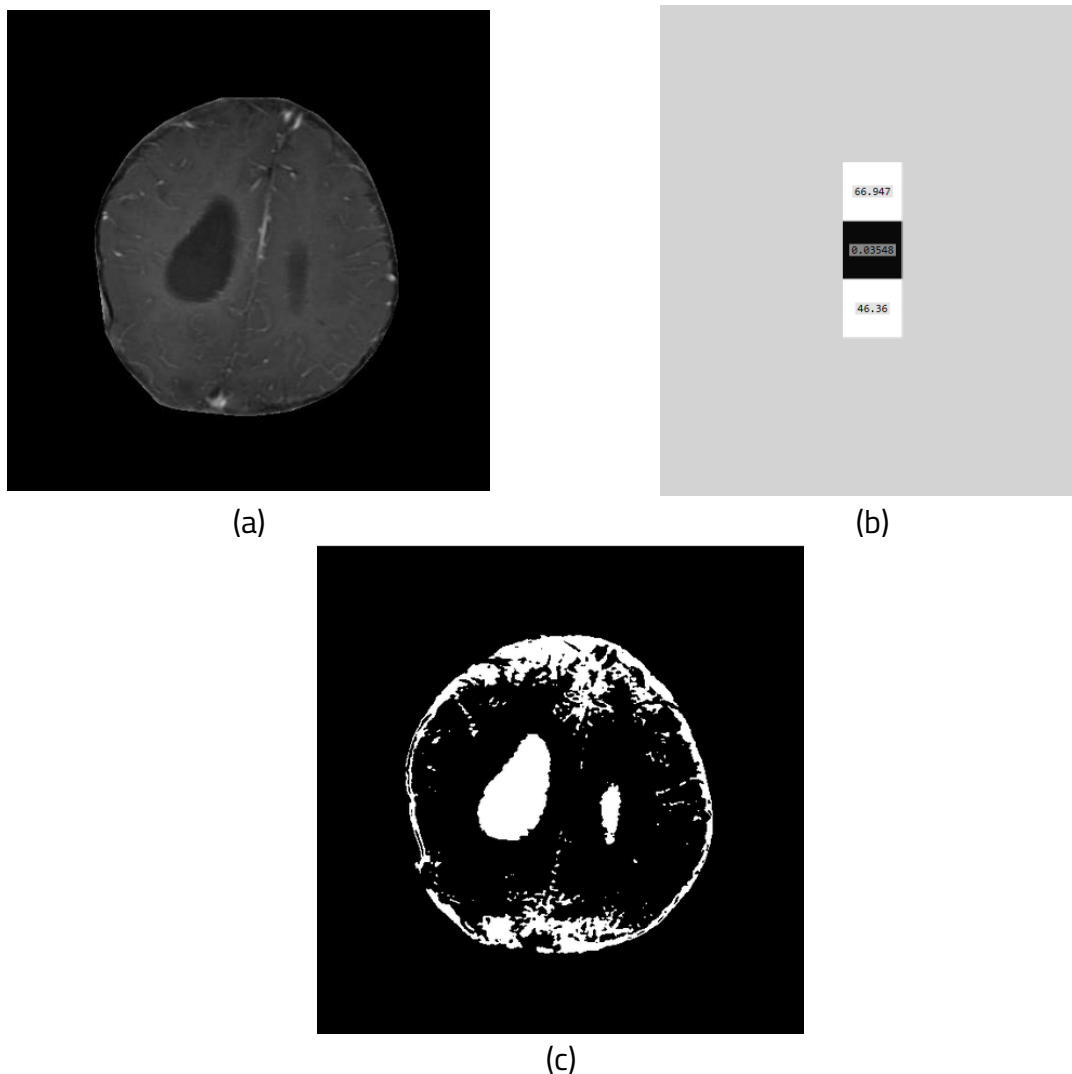


Fig. 20. (a) Imaginea de intrare; (b) Valorile centrozilor calculați prin algoritmul de grupare K-means; (c) Imaginea după aplicarea algoritmului.

4.3.2. Operațiunea morfologică de deschidere

Transformările morfologice sunt operații simple, bazate pe dimensiunile imaginii, aplicate, de obicei, pe imagini binare. Aceste transformări au nevoie de doi parametri de intrare: primul este reprezentat de imaginea originală, iar al doilea se numește nucleu (engl. *kernel*) care decide natura operației aplicate. Există două operații morfologice de bază care sunt eroziunea și dilatarea, apoi existând forme care variază de la acestea cum ar fi deschiderea, închiderea, gradientul etc.

Eroziunea

Ideea de bază a eroziunii este, exact ca și eroziunea solului, aceea de a eroda limitele unui obiect din prim plan. Principiul de funcționare al acestei operațiuni este că nucleul merge prin toată imaginea, iar un pixel din imaginea originală (cu valorile 1 sau 0) va fi considerat ca având valoarea 1 dacă toți pixelii din interiorul nucleului au valoarea 1, altfel el va fi erodat (îi va fi atribuită valoarea 0). Eroziunea este utilă pentru eliminarea punctelor albe dintr-o imagine, pentru detașarea a două componente conexe, etc.

Dilatarea

Operațiunea de dilatare este exact opusul eroziunii. În acest caz, un pixel este considerat cu valoarea egală cu 1 dacă cel puțin un pixel din interiorul nucleului are valoarea 1, altfel, acestui pixel îi este atribuită valoarea 0.

În mod normal, în cazuri de eliminare a zgomotului, eroziunea este urmată de dilatare, deoarece eroziunea elimină zgomotul alb, dar micșorează obiectul, iar dilatarea mărește aria obiectului, fără însă a reda zgomotul.

Deschiderea

Operația morfologică de deschidere, reprezintă practic o operație de eroziune, urmată de o operație de dilatare.

După aplicarea algoritmului de grupare K-means, se observă că, pe lângă pixelii care compun tumora cerebrală, algoritmul a mai păstrat și pixelii de la marginea zonei cerebrale, pixeli care, prin intensitatea lor apropiată de cea a pixelilor din tumora cerebrală, au „păcălit” algoritmul că ar face parte din tumora cerebrală.

Astfel, pentru eliminarea a mare parte din acești pixeli care încurcă segmentarea tumorii cerebrale, se aplică operația morfologică de deschidere asupra imaginii obținută după aplicarea algoritmului de grupare K-means folosind un nucleu de formă dreptunghiulară cu mărimea de 5 pe 5. Nucleul va avea forma următoare:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (4.12)$$

Mărimea nucleului a fost aleasă în mod optim pentru a putea elimina majoritatea pixelilor care nu sunt utili din imagine, fără a afecta însă zona în care se găsește tumora cerebrală.

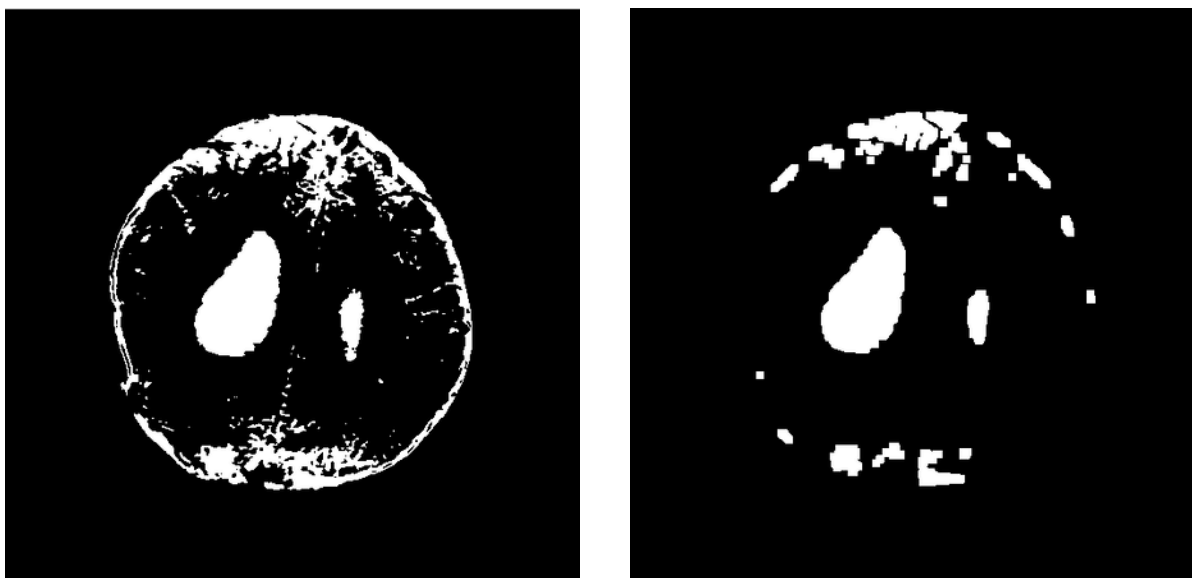


Fig. 21. Imaginea de intrare (stânga); Imaginea obținută după aplicarea operației de deschidere morfologică (dreapta).

4.3.3. Algoritmul componentelor conexe

Componentele conexe sau componentele din teoria grafurilor sunt subgrafuri ale unui graf conex în care oricare două vârfuri sunt conectate între ele prin drumuri și care nu sunt conectate de niciun alt vârf din supergraf.

Componentele conexe într-o imagine sunt seturi de pixeli cu aceeași valoare conectați prin metoda de conectare de 4 pixeli, care conectează toți pixelii care au aceeași valoare de-a lungul marginilor, sau prin metoda de conectare de 8 pixeli, care conectează toți pixelii care au aceeași valoare de-a lungul marginilor și colțurilor.

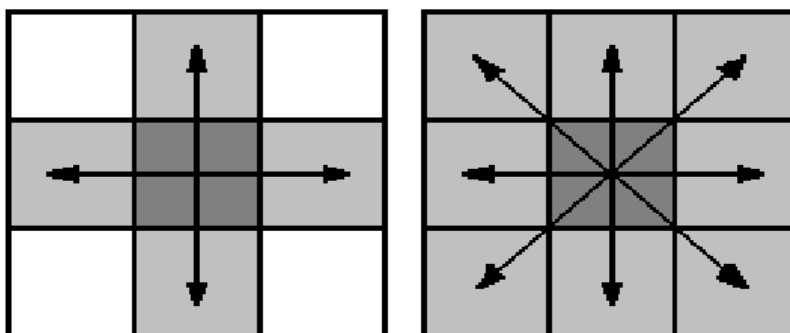


Fig. 22. Metoda de conectare de 4 pixeli (stânga); Metoda de conectare de 8 pixeli (dreapta).

Folosind ca imagine de intrare imaginea obținută după aplicarea operației morfologice de deschidere, se aplică funcția de componente conexe. După aplicarea acestei funcții, se alege componenta cu cea mai mare arie găsită. Pentru că rezultatele în urma aplicării tuturor algoritmilor precedenți sunt foarte bune, se așteaptă ca tumora cerebrală să fie în totalitate conectată. Astfel, componenta cu aria cea mai mare este, de fapt, tumora cerebrală care se dorește a fi detectată de aplicație.

După detectarea componentei conexe cu cea mai mare arie, aceasta poate fi extrasă din imaginea de intrare, obținând astfel o imagine care va conține doar acei pixeli care compun tumora cerebrală.

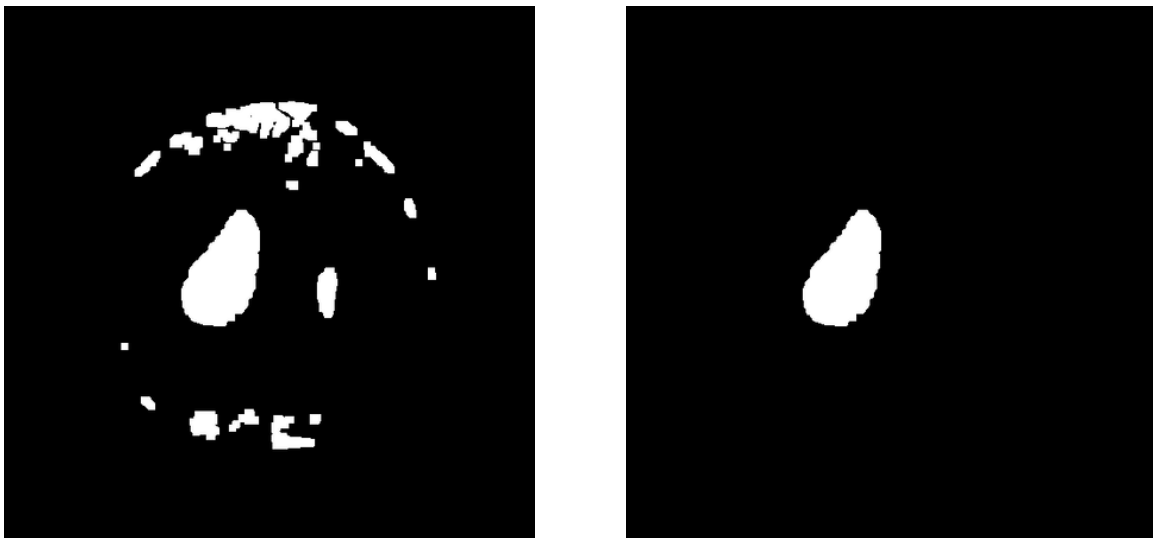


Fig. 23. Imaginea de intrare (stânga); Imaginea după aplicarea algoritmului componentelor conexe (dreapta).

4.4. Detectarea tumorii cerebrale

Proiectul prezentat are ca scop detectarea tumorilor cerebrale, prin evidențierea acestora în imaginile de tip RMN. După aplicarea tuturor pașilor anteriori, s-a ajuns la o imagine care conține doar pixelii care compun tumora cerebrală. Tot ceea ce a mai rămas de făcut este ca, folosind această imagine obținută la acest punct al proiectului, să fie evidențiată tumora cerebrală pe imaginea originală.

Pentru realizarea acestui ultim pas, în primul rând imaginea originală trebuie transformată din nou dintr-o imagine gri, cu un singur canal de culoare, într-o imagine RGB, cu trei canale de culoare, astfel încât să poată fi aplicat un contur colorat care să evidențieze tumora cerebrală.

După efectuarea acestei transformări, având imaginea care conține doar pixelii tumorii cerebrale, putem calcula conturul tumorii, contur care va fi evidențiat cu culoarea roșie în imaginea originală.

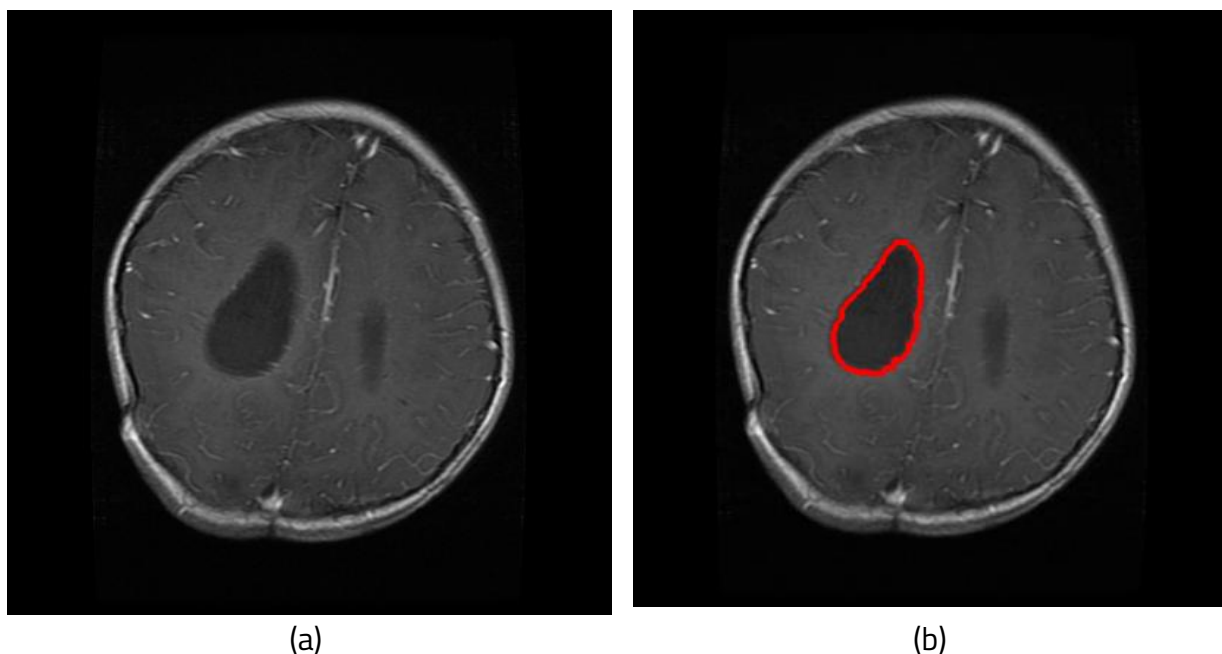


Fig. 24. (a) Imaginea inițială; (b) Imaginea finală, cu tumora cerebrală evidențiată.

5. INTERFAȚA GRAFICĂ

Rolul interfeței grafice este acela de a ajuta utilizatorul să folosească aplicația. Informațiile și acțiunile disponibile în interfață sunt prezentate prin anumite texte sugestive pentru ceea ce urmează să realizeze fiecare componentă. Interfața grafică a utilizatorului are o istorie care se leagă de ecranele bidimensionale cu rezoluție mare, ce pot afișa orice informație.

5.1. Realizarea interfeței grafice

Mediul de dezvoltare Qt este folosit pentru realizarea interfeței grafice a proiectului. Acesta se mai ocupă și cu realizarea programelor care nu au interfață, precum serverele, dar în cazul de față ele nu sunt relevante. Există două instrumente foarte des folosite pentru a facilita crearea și utilizarea unui GUI. Acestea sunt Qt Creator și Qt Designer. Diferența dintre cele două instrumente este că Qt Creator este un mediu de dezvoltare integrat pentru dezvoltarea de aplicații Qt. El acceptă, pe lângă crearea unei interfețe pentru utilizator, și scrierea unui cod într-o zonă destinată, pentru testarea acesteia. În schimb, Qt Designer se ocupă, după cum sugerează și denumirea sa, doar cu proiectarea unui GUI.

În proiect este utilizat instrumentul Qt Designer pentru crearea interfeței utilizatorului. Ceea ce este obținut în urma finalizării GUI este un fișier cu extensia „*.ui”, ce poate fi ușor integrat în orice program se dorește folosirea lui. Figura 5.1 surprinde ceea ce este afișat odată cu rularea Qt Designer, ceea ce se observă în mijloc fiind interfața proiectată pentru proiect. Vor fi prezentate în continuare și elementele care sunt vizibile în fereastra instrumentului de dezvoltare folosit, cât și cele de control utilizate în GUI.

- Zona numerotată cu **1** cuprinde toate elementele ce pot fi accesate și adăugate în interfața creată, precum: elemente specifice aranjamentului automat, spațiere, butoane, containere, elemente de control de intrare și de ieșire și altele.
- Zona centrală, **2**, reprezintă porțiunea care este deservită afișării GUI-ului realizat, în figură fiind prezentată cea din proiect.
- Regiunea numerotată cu **3** conține lista cu elemente adăugate interfeței create. Acestea sunt grupate în funcție de spațiul unde a fost pus, de exemplu, elementul central (în imagine: *centralwidget*) cuprinde o grilă (engl. *grid*) care, la rândul său, are incluse toate elementele destinate afișării imaginilor și a textelor (engl. *labels*).
- Porțiunea **4** este destinată detaliilor despre fiecare element selectat. Acestea se referă la dimensiunea componentei, la numele acesteia, la culorile și designul destinate acesteia, etc.

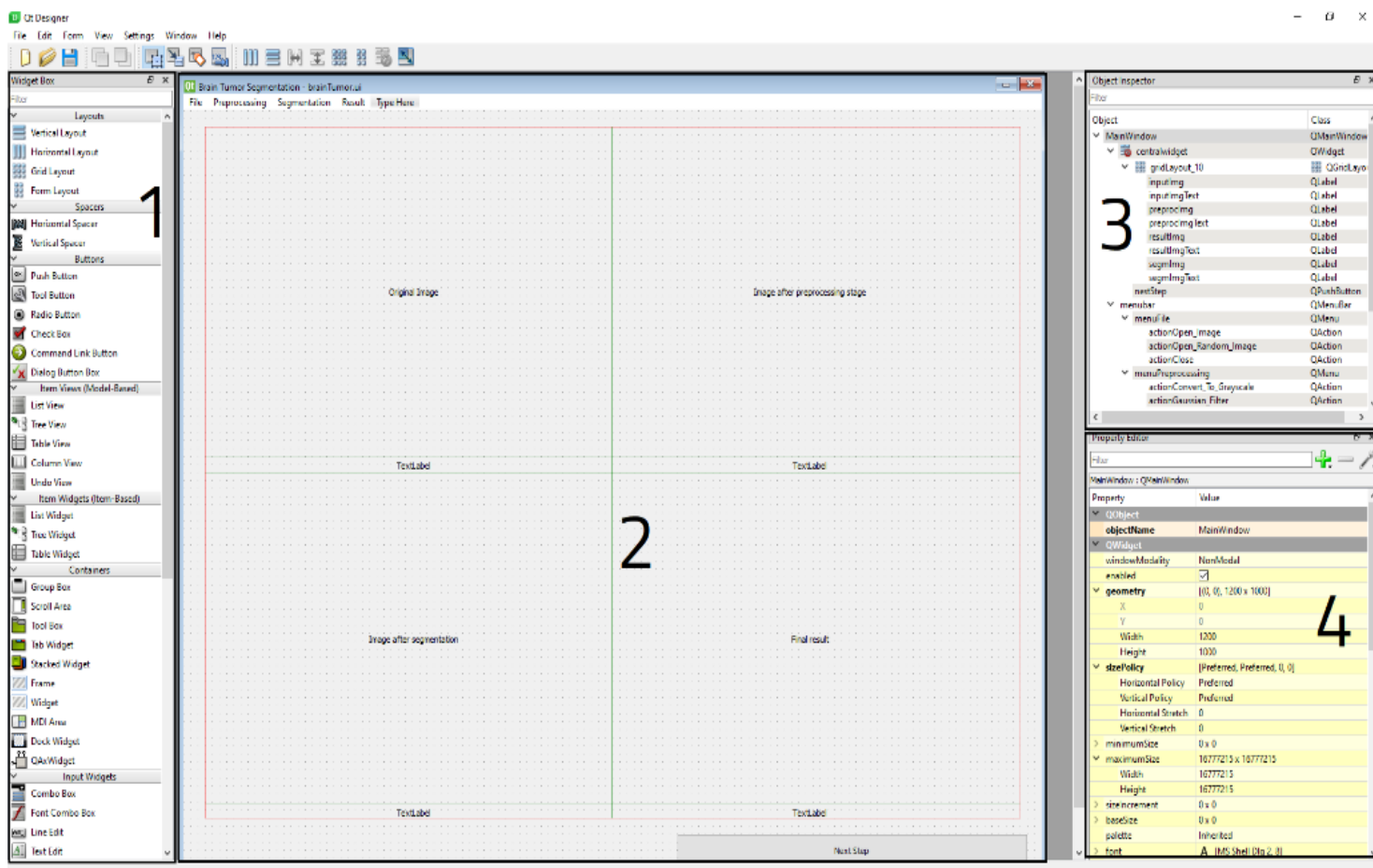


Fig. 25. Qt Designer – zona de proiectare a interfeței grafice.

Interfața utilizatorului, realizată în proiect, cuprinde câteva elemente de bază (engl. *widgets*). Oricare dintre aceste componente poate fi adăugată pe o fereastră sau poate fi deschisă într-una nouă. În figura 5.2 este surprinsă arhitectura elementelor incluse pe interfața grafică [26].

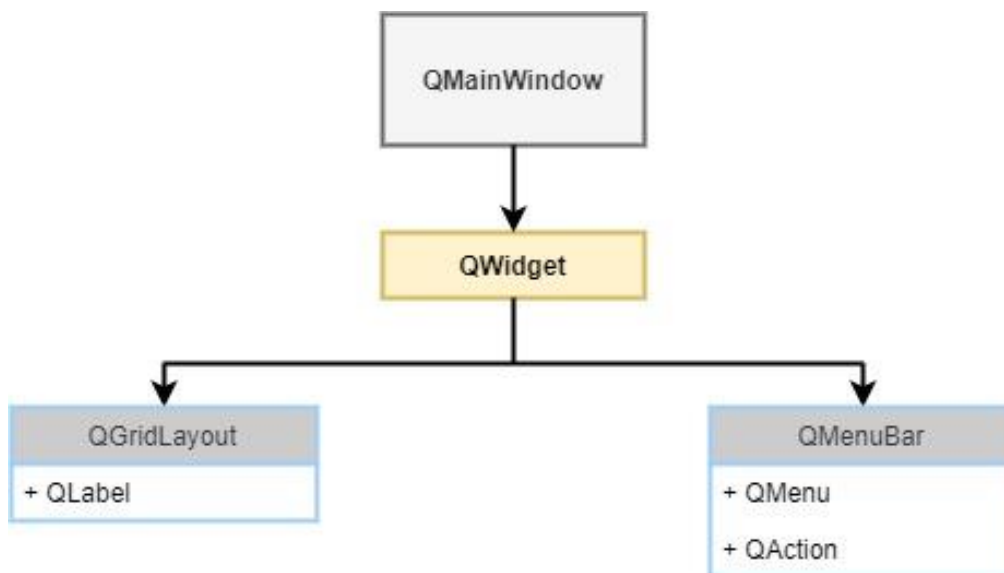


Fig. 26. Arhitectura elementelor interfeței grafice a utilizatorului.

QMainWindow este fereastra principală care oferă un cadru pentru adăugarea tuturor elementelor necesare proiectării interfeței dorite.

QWidget este clasa moștenită de obiectele din interfața grafică. Aceasta poate primi anumite evenimente de la tastatură și mouse, cărora să le răspundă prin îndeplinirea unor acțiuni.

Sistemul de layout-uri al Qt se ocupă cu aranjarea într-un mod automat a tuturor widget-urilor copil în zona destinată unui widget părinte, în așa fel încât tot spațiul disponibil să fie ocupat. Rolul layout-urilor este de a poziționa și redimensiona elementele componente din interfață în momentul în care dimensiunea acestora se schimbă, dar și de a asigura rearanjarea constantă a widget-urilor. Clasa **QGridLayout** aranjează elementele din interfață într-un tablou bidimensional, acestea putând ocupa mai multe celule. De regulă, când un element este inclus în **QGridLayout**, acestuia i se atribuie o anumită dimensiune pentru a ocupa spațiul disponibil în funcție de ce alte componente mai sunt adăugate.

QLabel este elementul cu ajutorul căruia se pot afișa texte, imagini sau filmulețe în GUI. Unui label i se pot adăuga conținuturi astfel: pentru text se utilizează funcția **setText()**, care cere ca parametru un **QString**, iar pentru afișarea și manipularea unei imagini se utilizează **setPixmap()**, ce are ca și parametru **QPixmap**. În proiect se utilizează aceste două elemente pentru text și imagini. **QPixmap** are proprietatea că atunci când imaginea este schimbată, nu se umple memoria, cele două imagini nefiind suprapuse și ținute în memorie. Astfel, programul este optim și poate fi utilizat pentru oricâte încărcări de imagini.

QMenuBar este elementul care conține bara de meniuri. Proiectul utilizează un astfel de meniu pentru a oferi mai mult loc afișării imaginilor rezultate în urma implementării. Dacă se foloseau butoane, se pierdea destul de mult loc din zona destinată imaginilor, acestea având nevoie de un spațiu mai mare, deoarece este prezentată câte una aproape la fiecare pas din proiect.

QMenu creează un meniu care este utilizat în bara de meniuri. Poate fi un meniu de tip derulant (afișat în jos) sau poate fi unul de sine stătător. Proiectul are patru meniuri de tip pull-down.

În aplicații, multe dintre comenzi sunt invocate prin meniuri, butoane sau chiar comenzi rapide de la tastatură. De obicei, utilizatorul se așteaptă ca fiecare comandă să fie executată la fel, iar acest lucru este posibil dacă acestea sunt reprezentate ca acțiuni. Acțiunile pot fi create ca obiecte independente, dar ele sunt făcute automat în timpul adăugării de meniuri. **QAction** poate conține atât o pictogramă, cât și o comandă rapidă sau un text. Odată ce un **QMenu** a fost adăugat, implicit și **QAction**, această acțiune trebuie conectată la un slot care o va efectua.

Qt prezintă o tehnică de apelare inversă: folosirea semnalelor și a sloturilor. **Semnalul** (engl. *signal*) este emis atunci când are loc un eveniment. Widget-urile Qt au mai multe semnale predefinite. Unul dintre ele este corespunzător pentru **QAction**, semnalul declanșat (engl. *triggered*), care este activat la fiecare conexiune a unei acțiuni cu slotul corespunzător. De asemenea, se pot adăuga în orice moment semnale proprii. De exemplu, în evenimentul creat în proiect, atunci când pe imagine se face click cu mouse-ul, se emite un semnal ce este recepționat atunci când se conectează cu funcția corespunzătoare îndeplinirii acțiunii. Un **slot** este o funcție ce reprezintă răspunsul unui semnal. Slotul `close()` din Qt este unul predefinit, care permite închiderea interfeței grafice, în cazul proiectului prezentat. Pentru toate celelalte acțiuni sunt create funcții proprii, astfel încât să fie gestionate semnalele care sunt de interes la momentul respectiv.

Odată prezentate toate elementele care compun interfața grafică din proiect, se poate trece la o evidențiere a funcționalității acesteia, prin testarea și simularea ei.

5.2. Testare și simulare

După ce programul este implementat, el este rulat și sunt verificate funcționalitățile lui. În momentul rulării, pe ecran apare interfața grafică disponibilă utilizatorului. Aceasta este prezentată, în formă inițială, în figura 5.3.

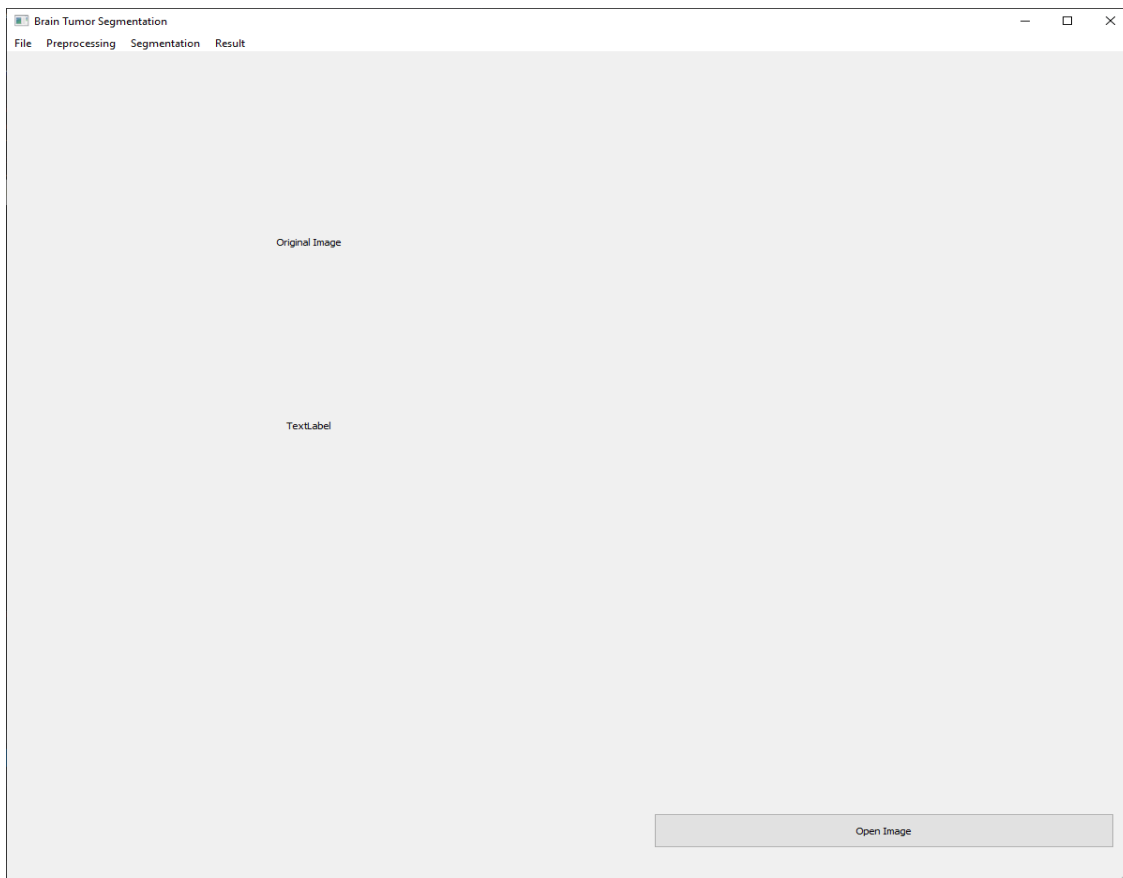


Fig. 27. Interfața grafică.

Interfața prezentată cuprinde o bară de meniuri, cu 4 meniuri: *File*, *Preprocessing*, *Segmentation*, *Result*, 4 label-uri destinate imaginilor afișate în urma alegerii unei acțiuni de executat și 4 label-uri pentru afișarea de text pentru informații cu privire la imaginile din label-uri. Când imaginea este vizibilă în interfață, sub acesta este afișat un text, într-un label.

Fiecare meniu este de tip pull-down și conține una sau mai multe acțiuni. Primul meniu din bara destinată este meniul *File*. La apăsarea acestuia, apar acțiunile pentru alegerea și afișarea unei imagini, atât de către utilizator, cât și în mod aleatoriu, din setul de date, dar și acțiunea pentru închiderea interfeței grafice. Al doilea meniu este *Preprocessing* și conține acțiuni pentru convertirea pixelilor din color în gri, aplicarea filtrului gaussian, eliminarea fundalului din imagine și pentru aplicarea operației de eliminare a craniului din imagine. Meniul al treilea are numele *Segmentation* și are ca acțiuni asociate: aplicarea algoritmului de grupare K-Means, aplicarea operației de deschidere asupra imaginii pentru eliminarea pixelilor neimportanți și pentru aplicarea algoritmului componentelor conexe pentru extragerea zonei din imagine în care se regăsește tumora cerebrală. Ultimul meniu, denumit *Result*, este destinat rezultatului

final, iar acțiunea creată este de afișare a rezultatului. Aceste meniuri sunt prezentate în figura 5.4.

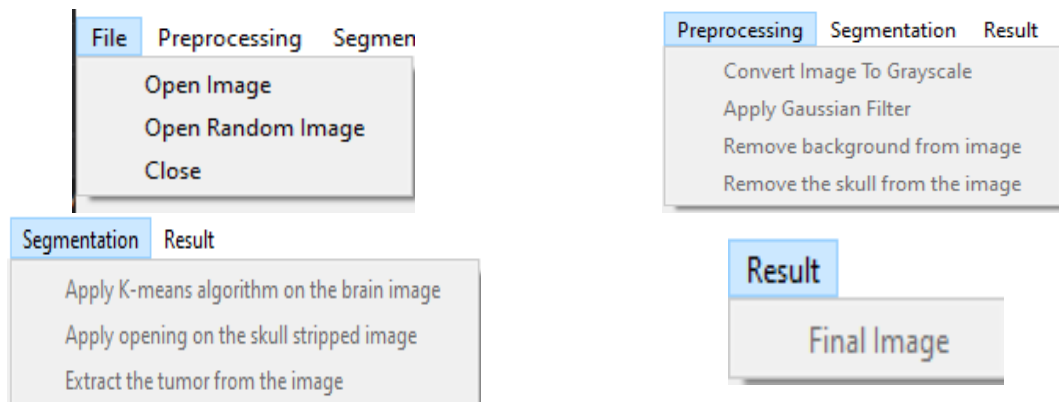


Fig. 28. Meniurile disponibile în interfața grafică.

Se observă că în afară de primul meniu ale cărui acțiuni apar cu negru, celelalte acțiuni ale meniurilor sunt gri. Acest lucru se explică prin faptul că toate cele trei meniuri au acțiunile dezactivate. Activarea lor are loc odată cu alegerea și afișarea pe interfață a unei imagini supuse procesării și cu aplicarea fiecărui pas în ordine, așa cum este exemplificat în imaginea din figura 5.5 pentru meniul de preprocesare.

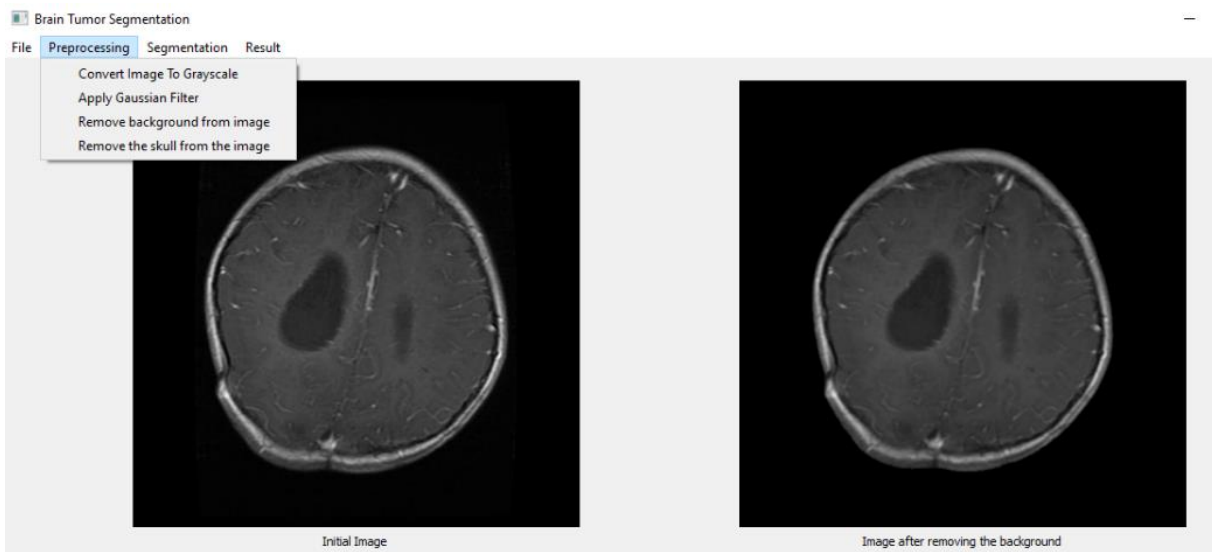


Fig. 29. Activarea acțiunilor din meniul de preprocesare.

S-a optat pentru menținerea acestor acțiuni dezactivate inițial pentru a nu crea erori în funcționarea aplicației. Oricare dintre acțiunile din proiect au nevoie de imaginea de intrare, cea color, pentru realizarea funcționalității corespunzătoare. Din acest motiv acțiunile sunt activate treptat, după executarea fiecărui pas.

Slot-urile corespunzătoare fiecărei acțiuni sunt implementate independent. Este de menționat faptul că, dacă se dorește, de exemplu, aplicarea algoritmului de grupare K-Means, imediat după ce imaginea de intrare este aleasă, acest lucru nu va fi posibil fără parcurgerea în prealabil a tuturor pașilor anteriori pasului dorit de utilizator. S-a ales acest mod de utilizare, deoarece, pentru a putea executa orice pas al algoritmului imediat după încărcarea imaginii de intrare ar fi dus la un timp de execuție prelungit, pentru că ar fi trebuit incluse în acțiunile fiecărui pas și metodele anterioare pasului respectiv.

Zona destinată afișării imaginii finale, în care tumora cerebrală este evidențiată cu un contur roșu pe imaginea de intrare, este reprezentată de un label în care se încarcă imaginea finală, împreună cu un label pe care este afișat text.

Figura 5.6 surprinde interfața grafică corespunzătoare detectării unei tumori cerebrale de tip gliom.

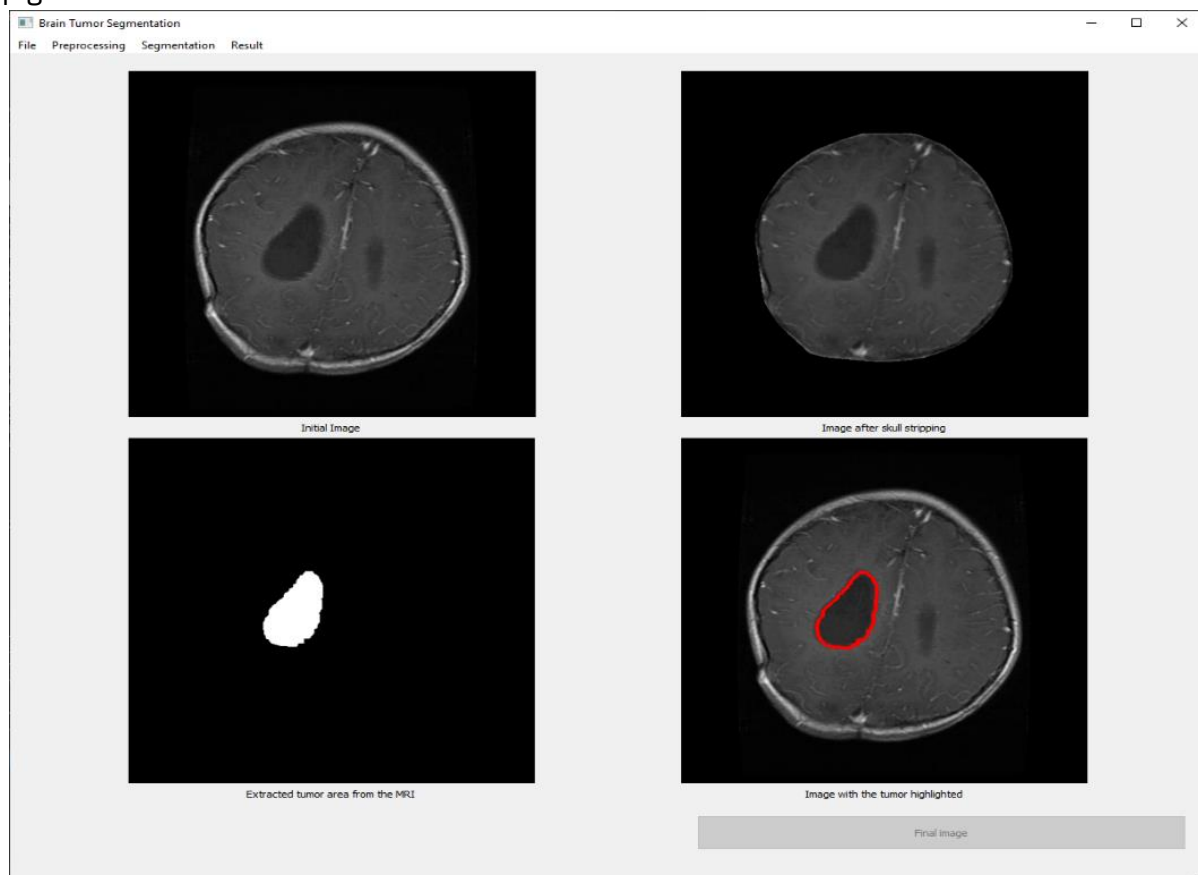


Fig. 30. Interfața grafică completă, prezentând o tumoră cerebrală de tip gliom.

De asemenea, în figura 5.7 este prezentată interfața grafică corespunzătoare detectării unei tumori cerebrale de tip meningiom.

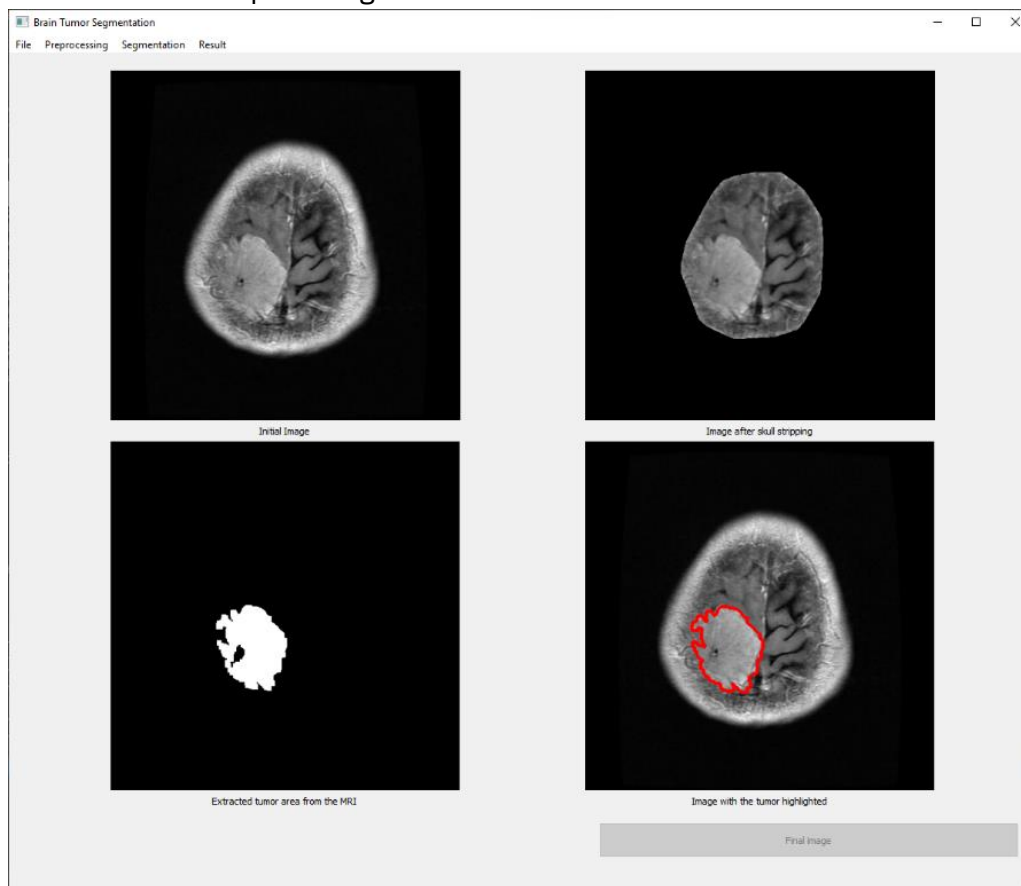


Fig. 31. Interfața grafică completă, prezentând o tumoră cerebrală de tip meningiom.

6. DIRECȚII VIITOARE DE CERCETARE

În ultima perioadă, unele dintre cele mai dezvoltate proiecte software sunt cele care includ medicina, deoarece imagistica medicală s-a dezvoltat și ea la rândul său. Bineînțeles, un astfel de proiect pornește de la a fi minimal, apoi poate continua și poate cuprinde diferite părți care se axează pe subiectul principal. Făcând referire la proiectul prezentat, acesta este realizat pentru a detecta o afecțiune cerebrală, care poate genera o multitudine de alte complicații pentru o persoană.

Imaginea proiectului dezvoltat a început să se contureze în jurul unei boli tot mai des întâlnite, tumora cerebrală. Din acest punct, s-a urmărit să se găsească o posibilă metodă care să ajute personalul specializat în detectarea anumitor probleme ce pot apărea, referitoare la boala respectivă. Cum pentru tratarea tumorilor detectate în alte părți ale corpului se lucrează la din ce în ce mai multe programe și aparate utile, a fost necesară o documentare pentru detectarea altor probleme care pot apărea la persoanele care suferă de tumori cerebrale. Fiind o afecțiune care se manifestă atât de sever, s-a optat pentru detectarea acesteia pe baza imaginilor craniene preluate din aparatele speciale utilizate de medicii neurologi. Acest subiect principal examinat nefiind singura complicație posibilă care poate să apară, se dorește ca proiectul realizat să fie continuat și să poată fi și mai util pentru medicii neurologi, și nu numai.

În acest sens, vor fi prezentate câteva idei care se doresc a fi implementate în proiectul actual pentru dezvoltarea acestuia ulterioară.

6.1. Idei de dezvoltat

Se urmărește ca proiectul realizat să fie cât mai corect posibil, pentru că, odată ce este dezvoltat în domeniul medical, nu sunt permise erori de diagnosticare, acesta trebuind să fie cât mai precis și să se poată folosi în cât mai multe cazuri. Astfel, există câteva idei de abordat în viitor pentru a face ca precizia segmentării tumorilor cerebrale să fie mai bună, dar și pentru a dezvolta utilizarea proiectului de către personalul specializat în detectarea afecțiunilor oculare. Câteva exemple de astfel de idei de dezvoltat sunt prezentate în continuare:

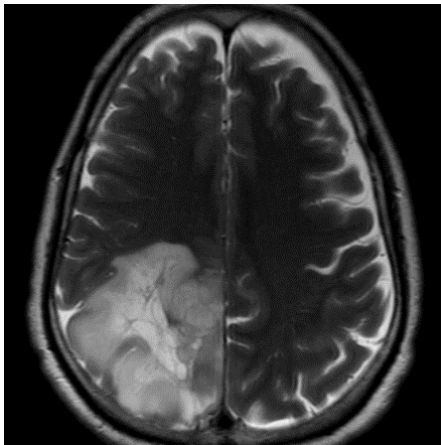
- Îmbunătățirea contrastului imaginilor

În proiect, imaginile pe care s-a realizat detectarea tumorilor cerebrale sunt preluate din setul de date Brain Tumor MRI Dataset, iar acestea sunt îmbunătățite pe cât de mult posibil. Există însă și imagini, din alte seturi de date, care nu au un contrast la fel de bun ca cele folosite, și pentru care testarea lor în proiectul prezentat nu produce niciun rezultat. Pentru a mări aria de dezvoltare și testare, este necesar să fie implementat un algoritm prin care

contrastul oricărei imagini să fie îmbunătățit, astfel proiectul putând fi utilizat pentru orice imagine cu craniul uman. S-a avut în vedere până la momentul actual documentarea în legătură cu ceea ce trebuie implementat în acest punct, iar cea mai bună alegere este egalizarea histogramei cu metoda CLAHE, adică egalizarea adaptivă a histogramei cu contrast limitat [27].

- Detectarea mai multor tipuri de tumori cerebrale

Proiectul trebuie dezvoltat pentru a detecta mai multe tipuri de tumori cerebrale, deoarece ar putea ajuta și mai mult în instituțiile care se ocupă cu o astfel de detectare. Deși, în momentul actual, se pot detecta trei tipuri de tumori cerebrale din cele mai comune reprezentate de gliom, meningiom și de tumorile pituitare, se urmărește ca algoritmi să fie optimizați pentru a putea detecta și alte tipuri de tumori cerebrale cum ar fi astrocitomul, craniofaringiomul sau carcinomul plexului coroid.



(a)



(b)



(c)

Fig. 32. (a) Astrocitom; (b) Carcinom al plexului coroid; (c) Craniofaringiom.

- Detectarea tumorilor de orice tip (de exemplu: tumori mamare, tumori pancreatice etc.)

La momentul actual, proiectul dezvoltat se axează pe segmentarea tumorilor cerebrale din imaginile de tip RMN, fiind destinat medicilor neurologi specializați în detectarea unor diagnostice de acest tip. O altă idee de dezvoltat în viitor este ca toate tumorile, indiferent de organele în care se găsesc, să poată fi detectate, tot printr-o procesare amănunțită a imaginilor. Pentru aceste lucruri este nevoie de algoritmi individuali pentru fiecare organ, care să țină cont de caracteristicile fiecărui organ, astfel încât segmentarea tumorilor să se producă fără erori. O astfel de dezvoltare ulterioară ar putea face ca proiectul să nu rămână destinat doar medicilor neurologi, ci să poată fi utilizat de medicii de orice tip, poate chiar și de medicii de familie, acest lucru ducând la o detecție mai rapidă a tumorilor și, în consecință, la o intervenție mai rapidă a medicilor specialiști în tratarea afecțiunilor detectate.

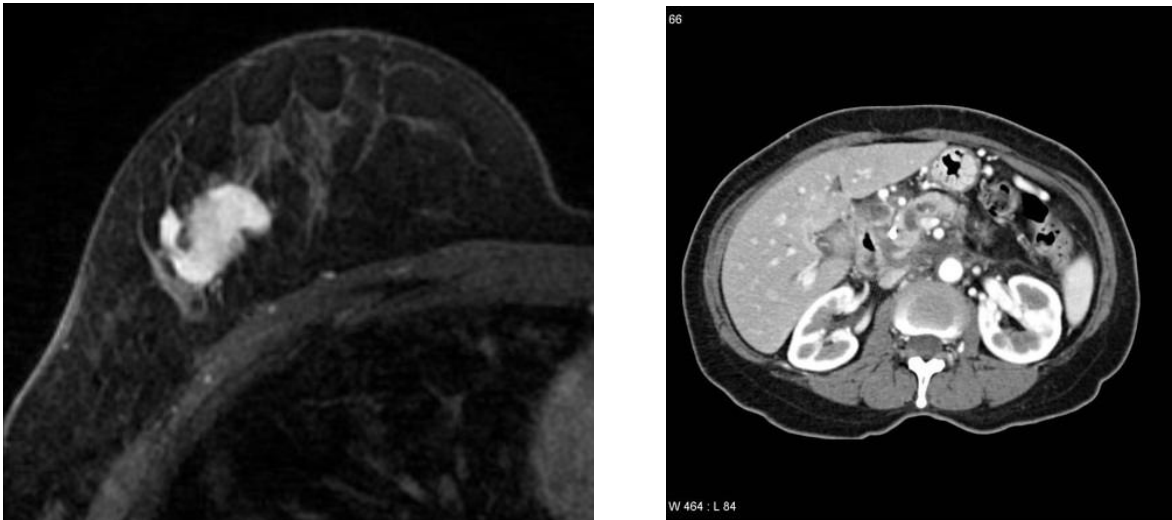


Fig. 33. Tumoră mamară (stânga); Tumoră pancreatică (dreapta).

- Segmentarea encefalului pentru detectarea poziției tumorii

Un amănunt foarte important pentru tratarea tumorilor cerebrale este determinarea cu exactitate a poziției unde se găsesc acestea. În funcție de poziționarea lor, medicul specialist poate lua o decizie în legătură cu tratamentul optim pentru afecțiunea detectată. Astfel, o idee pentru dezvoltarea ulterioară a proiectului este segmentarea creierului în funcție de lobi și calcularea poziției tumorii cerebrale detectate în funcție de acești lobi.

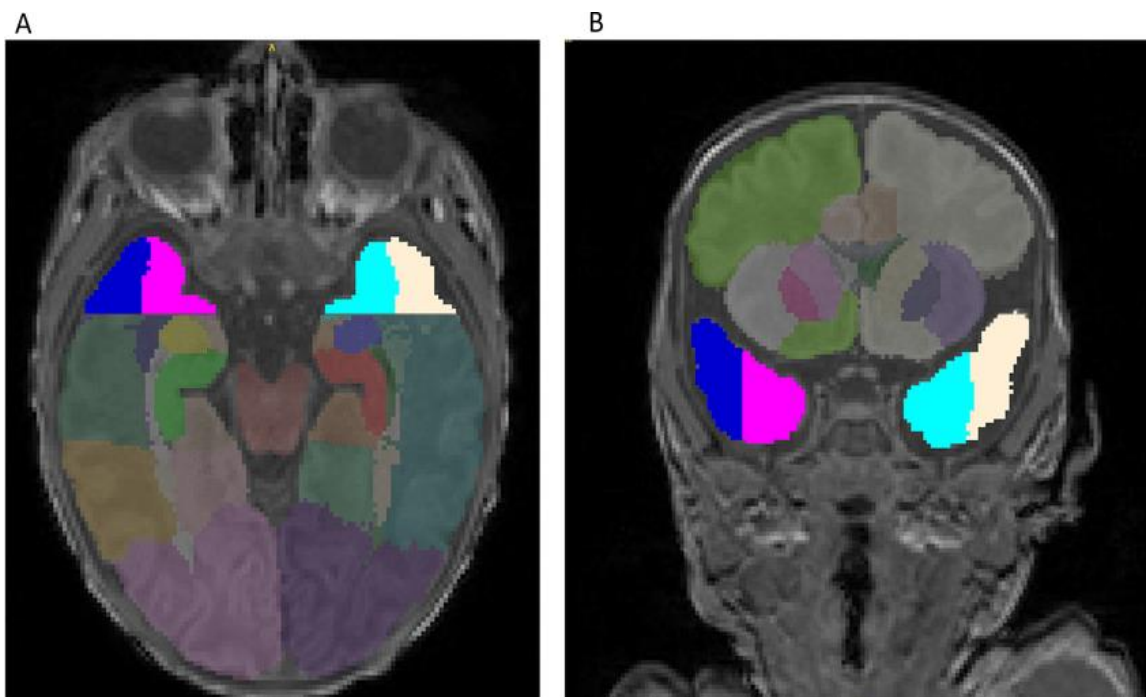


Fig. 34. Segmentarea lobilor cerebrali.

- Realizarea unui program bazat pe inteligența artificială, pentru a face o comparație între rezultatele obținute din acesta și cele din procesarea simplă de imagini, realizată în proiect

Foarte mulți programatori aleg să nu mai implementeze cod pur pentru proiectele care presupun procesare de imagini. Ei se axează pe învățarea automată a programelor cu inteligență artificială. După cum a fost amintit, proiectul realizat nu a utilizat AI deloc, pentru a se accentua faptul că rezultatele în urma procesării imaginilor sunt unele corecte și sunt generate mai rapid decât în cazul programelor care utilizează inteligența artificială. Comparația dintre aceste două posibile metode se dorește să fie surprinsă prin calcule clare din punct de vedere al timpului de rulare. Astfel, este necesar ca proiectul prezentat să fie implementat cu ajutorul AI. Din acest lucru se pot trage concluzii și în ceea ce privește acuratețea segmentării vaselor de sânge și detectarea corectă a afecțiunii oculare.

7. CONCLUZII

7.1. Concluzii generale

Proiectul prezentat urmărește punerea unui diagnostic asupra persoanelor care cred că suferă de tumori cerebrale. Acestea sunt afecțiuni foarte grave care pot provoca multiple complicații grave bolnavilor respectivi. Prin detectarea afecțiunii, personalul medical specializat poate ajuta pacienții în remedierea problemei sau stagnarea acesteia.

Proiectul software este realizat pentru a detecta dacă encefalul pacientului este afectat de o tumoră cerebrală. Acest rezultat a fost posibil doar în urma segmentării imaginilor de tip RMN.

Pentru ca produsul final să fie unul pe măsura așteptărilor, au fost urmați câțiva pași care au avut rezultate bune. Deși au fost încercate multe variante pentru implementarea anumitor algoritmi, proiectul prezintă în desfășurarea sa doar acele metode care au generat rezultatele corecte. În acest sens, primul pas urmărit a fost cel de găsire a unor imagini corespunzătoare pentru realizarea proiectului. Acestea au fost alese din setul de date Brain Tumor MRI Dataset, fiind de mare ajutor pentru aplicarea tuturor etapelor următoare. În continuare, pe baza imaginilor suport, s-a făcut o preprocesare, care a avut ca scop eliminarea zgomotului de fundal, dar și eliminarea craniului uman din imaginile suport. Pasul acesta a fost unul reușit, eliminarea craniului efectuându-se cu mare exactitate, fiind de mare ajutor pentru segmentarea ce urmează. Apoi, pentru segmentare s-a recurs la aplicarea algoritmului de grupare K-means, care s-a rezumat la găsirea pixelilor cu intensitatea necesară pentru a face parte din aria tumorală, urmat de aplicarea unei operații morfologice de deschidere, pentru eliminarea pixelilor care nu sunt utili în segmentarea tumorii. Rezultatul astfel obținut a fost supus identificării componentelor conexe. După ce acestea au fost detectate, etapa de segmentare a fost finalizată.

Având imaginea obținută după etapa de segmentare, imagine ce conține componenta conexă ce reprezintă tumora cerebrală detectată în imaginea suport, ultimul pas al proiectului a fost de a evidenția tumora cerebrală în imaginea originală din setul de date, prin evidențierea acesteia printr-un contur de culoare roșie.

De asemenea, interfața grafică realizată în proiect a fost făcută în așa fel încât informațiile determinate să fie vizibile și, totodată, accesibile. GUI este ușor de utilizat, având denumiri clare ale acțiunilor ce urmează să fie efectuate. Simplitatea interfeței grafice, aceasta prezentând doar un singur buton prin care pot fi accesate, pe rând, toate funcțiile, este utilă, deoarece, o mare parte din medicii care urmează să utilizeze proiectul nu s-ar pricepe în a folosi

o interfață complicată, cu meniuri multiple. Prin utilizarea unui singur buton, proiectul asigură utilizarea cu ușurință a funcțiilor sale de către orice medic.

În concluzie, proiectul conceput este util pentru detectarea tumorilor cerebrale folosind imagini obținute prin metoda rezonanței magnetice nucleare. Domeniul de dezvoltare fiind cel medical, lucrarea are ca direcție oferirea unui ajutor personalului medical care se ocupă cu persoanele care suferă din cauza tumorilor cerebrale. De asemenea, rezultatul generat a urmărit scopul inițial ales. Un lucru important de menționat este acela că proiectul poate fi dezvoltat ulterior pentru detectarea mai multor tipuri de tumori, fie ele cerebrale, sau de alt tip, dar și pentru a oferi sugestii de tratament medicilor în funcție de poziționarea și de mărimea tumorilor detectate.

7.2. Elemente de originalitate

Proiectul realizat a pus accent pe elementele de originalitate încă de la început. Astfel, viziunea asupra temei este considerată ca fiind originală, deoarece, majoritatea proiectelor care se ocupă cu segmentarea de tumori se axează pe folosirea inteligenței artificiale. Alegând evitarea utilizării AI se pune prima amprentă proprie.

De asemenea, implementarea software aduce un plus elementelor de originalitate, prin faptul că se utilizează procesarea de imagine în forma ei brută, fără să se recurgă la inteligența artificială. În prezent, majoritatea aplicațiilor dezvoltate în domeniul medical sunt realizate utilizând AI, deoarece datele existente sunt numeroase, iar rezultatele sunt precise destul de corect în cele mai multe cazuri. Cu toate acestea, în proiectul prezentat nu se folosește deloc AI, pentru a accentua timpul de extragere a unui diagnostic, timp care este mult mai mic față de o implementare cu inteligență artificială. Cea din urmă necesită foarte mult timp pentru antrenarea datelor, acesta fiind un factor prețios în zona software, deoarece nu este indicat să se piardă mult timp degeaba.

Totodată, s-a utilizat procesarea de imagini în forma ei inițială pentru a strânge mai multe cunoștințe, adică a învăța ceea ce se află în spatele anumitor funcții. Pentru a accentua acest aspect, s-au realizat metodele de convertire a imaginii color în gri, metodele pentru aplicarea filtrelor pentru eliminarea zgomotului de fundat, urmând să se realizeze și alte funcții corespunzătoare celor din biblioteca OpenCV, precum identificarea componentelor conexe.

Tot în zona elementelor de originalitate se poate încadra și modul de segmentare utilizat. Tumorile cerebrale sunt identificate utilizând algoritmul de grupare K-means, urmat de operația morfologică de deschidere, iar mai apoi de identificarea componentelor conexe, ceea ce în proiectele găsite în urma documentării de specialitate nu era folosit.

Totodată, realizarea bibliotecii DLL proprii, Algorithms, reprezintă un punct important de evidențiat. Această bibliotecă este utilizată pentru algoritmi de procesare de imagine utilizați în proiect, separându-i pe aceștia de componentele din view, cu scopul de a nu suprapune OpenCV cu mediul de dezvoltare Qt. În concluzie, biblioteca Algorithms este elementul de conexiune dintre biblioteca OpenCV și mediul de dezvoltare Qt.

8. BIBLIOGRAFIE

- [1] "Brain Cancer Forum," eMedicineHealth, n.d.. [Online]. Available: https://comments.emedicinehealth.com/brain_cancer/viewer-comments_em-26.htm. [Accessed 15 02 2022].
- [2] G. Jones, "Cancer Research UK," 2015. [Online]. Available: <https://news.cancerresearchuk.org/2015/02/04/why-are-cancer-rates-increasing/>. [Accessed 15 02 2022].
- [3] „Background Radiation,” Science Direct, n.d.. [Interactiv]. Available: <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/background-radiation>. [Accesat 15 02 2022].
- [4] "Imagistică medicală," Wikipedia, 2021. [Online]. Available: https://ro.wikipedia.org/wiki/Imagistic%C4%83_medical%C4%83. [Accessed 18 02 2022].
- [5] C. T. Niculescu, în *Anatomia și Fiziologia Omului. Compendiu - Ediția a II-a*, Editura Educațională Corint, 2014, p. 150.
- [6] O.-R. Neagu, "Encefalul (creierul)," ROMedic, 22 05 2013. [Online]. Available: <https://anatomie.romedic.ro/encefalul-creierul>. [Accessed 18 02 2022].
- [7] "Tumorile cerebrale," Regina Maria, n.d.. [Online]. Available: <https://www.reginamaria.ro/utile/dictionar-de-afectiuni/tumorile-cerebrale>. [Accessed 18 02 2022].
- [8] Albatross, cplusplus.com, n.d.. [Online]. Available: <https://cplusplus.com/info/description/>. [Accessed 22 02 2022].
- [9] OpenCV, n.d.. [Online]. Available: <https://opencv.org/>. [Accessed 22 02 2022].
- [10] T. Helland, "Seven grayscale conversion algorithms," tannerhelland.com, 1 10 2011. [Online]. Available: <https://tannerhelland.com/2011/10/01/grayscale-image-algorithm-vb6.html>. [Accessed 22 02 2022].

- [11] "2D Convolution (Image Filtering)," OpenCV, n.d.. [Online]. Available: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html. [Accessed 22 02 2022].
- [12] "OpenMP," Wikipedia, 5 05 2022. [Online]. Available: <https://en.wikipedia.org/wiki/OpenMP>. [Accessed 15 05 2022].
- [13] "About Qt," Qt, 23 03 2022. [Online]. Available: https://wiki.qt.io/About_Qt. [Accessed 22 05 2022].
- [14] "Qt Designer Manual," Qt, n.d.. [Online]. Available: <https://doc.qt.io/qt-5/qtdesigner-manual.html>. [Accessed 22 05 2022].
- [15] "About CMake," CMake, n.d.. [Online]. Available: <https://cmake.org/overview/>. [Accessed 22 05 2022].
- [16] M. Nickparvar, "Brain Tumor MRI Dataset," Kaggle, 2021. [Online]. Available: <https://doi.org/10.34740/KAGGLE/DSV/2645886>. [Accessed 14 03 2022].
- [17] B. Devkota, A. Abeer, P. Prasad, A. Singh and A. Elchouemi, "Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction," *Procedia Computer Science*, 2018.
- [18] S.-C. Tai and S.-M. Yang, "A fast method for image noise estimation using Laplacian operator and adaptive edge detection," *2008 3rd International Symposium on Communications, Control and Signal Processing*, pp. 1077-1081, 2008.
- [19] "Otsu's method," Wikipedia, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Otsu%27s_method. [Accessed 21 03 2022].
- [20] E. Hodneland, "Segmentation of Digital Images," pp. 17-18, 2013.
- [21] C.-A. Boiangiu, A. Olteanu, A. Stefanescu, D. Rosner and N. Tapus, "Local Thresholding Algorithm Based on Variable Window Size Statistics," *Proceedings of the 18th International Conference on Control Systems and Computer Science (CSCS)*, pp. 647-652, May 2011.

- [22] "Clustering," OpenCV, n.d.. [Online]. Available: https://docs.opencv.org/4.x/d5/d38/group__core__cluster.html. [Accessed 21 05 2022].
- [23] Y. Joshi, "Connected Component Labeling," OpenGenus IQ: Computing Expertise & Legacy, n.d.. [Online]. Available: <https://iq.opengenus.org/connected-component-labeling/>. [Accessed 21 05 2022].
- [24] H. Driver and A. Kroeber, "Quantitative Expressions of Cultural Relationships," in *University of California Publications in American Archaeology and Ethnology - Volume 31, No. 4*, 1932, pp. 211-256.
- [25] N. Sharma, "K-Means Clustering Explained," Neptune Blog, 4 11 2021. [Online]. Available: <https://neptune.ai/blog/k-means-clustering>. [Accessed 21 05 2022].
- [26] "Widgets and Layouts," Qt, n.d.. [Online]. Available: https://wiki.qt.io/Widgets_and_Layouts/. [Accessed 16 03 2022].
- [27] H. Kaur and J. Rani, "MRI brain image enhancement using Histogram Equalization techniques," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 770-773.

REZUMAT

Proiectul realizat are ca scop identificarea afecțiunilor cerebrale denumite tumori cerebrale la persoanele suspecte de această boală. Acesta se adresează tuturor persoanelor care suspectează că ar suferi de o asemenea afecțiune și le îndeamnă să intre în contact cu personalul medical de specialitate pentru un control neurologic amănunțit. De asemenea, proiectul vine în ajutorul medicilor neurologi, oferind o aplicație ușor de utilizat, în care imaginea craniană extrasă folosind aparatele de rezonanță magnetică nucleară este procesată și analizată cu scopul de a detecta dacă encefalul pacientului este afectat sau nu de tumorile cerebrale.

Pentru îndeplinirea proiectului, este necesară parcurgerea a trei etape, ce vor fi prezentate pe scurt în continuare.

Etapă de preprocesare este cea inițială. După ce datele necesare au fost introduse, în cazul de față fiind nevoie de imaginea RMN ce conține craniul uman, se începe procesarea acesteia. Această etapă incipientă constă în conversia imaginii dintr-o imagine color într-una în tonuri de gri, urmată de eliminarea zgomotului de fundal, care împiedică o bună segmentare a creierului. De asemenea, tot în această fază a implementării codului, pixelii de fundal ai imaginii sunt eliminați, iar zona creierului este extrasă din imagine prin eliminarea craniului care poate genera rezultate false în etapa de segmentare.

Etapă de extragere a caracteristicilor este cea care realizează segmentarea tumorilor cerebrale. Inițial, cu ajutorul algoritmului de grupare K-Means, pixelii a căror intensitate se poate încadra în intervalul intensității pixelilor care compun tumora sunt extrași din imaginea RMN. Următorul pas este eliminarea pixelilor care nu sunt necesari în segmentare prin aplicarea unei operații morfologice de deschidere, ultimul pas fiind aplicarea algoritmului componentelor conexe pentru identificarea componentei cu cea mai mare arie din imagine, componentă care, pentru proiect, este reprezentată de tumora care se dorește a fi detectată.

Ultima etapă necesară îndeplinirii scopului proiectului este aceea de evidențiere a tumorii cerebrale în imaginea suport încărcată la primul pas al aplicației. Această etapă constă în transformarea imaginii înapoi dintr-o imagine în tonuri de gri într-una color, apoi, folosind conturul componentei conexe calculată la pasul anterior, evidențierea acestui contur cu roșu în imaginea originală.

De asemenea, proiectul pune la dispoziție și o interfață grafică ușor de utilizat de către cei care o folosesc. Aceasta oferă posibilitatea de a fi vizibile imaginile corespunzătoare mai multor etape din desfășurarea proiectului, lucru ce ajută la analiza amănunțită a lor. Fluxul

aplicației este următorul: utilizatorul alege o imagine de intrare, care constă în cea extrasă cu ajutorul aparatului RMN, în proiect fiind utilizate imagini din setul de date Brain Tumor MRI Dataset. Următorul pas este de a efectua pașii de preprocesare pe rând, până se ajunge la o imagine fără zgomot de fundal și cu craniul eliminat din aceasta, urmat de segmentarea propriu-zisă a tumorii cerebrale, iar mai apoi de generarea rezultatului final.

ABSTRACT

The project aims to identify brain conditions called brain tumours in people suspected of having this disease. It is aimed at anyone who suspects they may have such a condition and urges them to contact specialist medical staff for a thorough neurological check-up. The project also helps neurologists by providing an easy-to-use application in which images of the skull taken using nuclear magnetic resonance machines are processed and analysed to detect whether or not the patient's brain is affected by brain tumours.

Three steps are required to complete the project, which will be briefly described below.

The pre-processing stage is the initial one. Once the necessary data have been entered, in this case the MRI image containing the human skull, processing begins. This initial stage consists of converting the image from a colour image to a greyscale image, followed by the removal of background noise, which prevents good segmentation of the brain. Also in this phase of the code implementation, the background pixels of the image are removed and the brain area is extracted from the image by removing the skull which can generate false results in the segmentation step.

It is the feature extraction stage that performs the segmentation of brain tumours. Initially, using the K-Means clustering algorithm, pixels whose intensity can fall within the intensity range of the pixels that make up the tumour are extracted from the MRI image. The next step is to remove the pixels that are not needed in the segmentation by applying a morphological opening operation, the last step is to apply the connected components algorithm to identify the component with the largest area in the image, which for the project is represented by the tumour to be detected.

The last step required to achieve the project goal is to highlight the brain tumour in the support image loaded in the first step of the application. This step consists of transforming the image back from a greyscale image to a colour image, then, using the contour of the related component calculated in the previous step, highlighting this contour with red in the original image.

The project also provides an easy-to-use graphical user interface. This makes it possible to see images of several steps in the project, which helps to analyse them in detail. The flow of the application is as follows: the user chooses an input image, which is the one extracted with the MRI machine, and images from the Brain Tumor MRI Dataset are used in the project. The next step is to perform the pre-processing steps one at a time, until an image without background noise and with the skull removed from it is reached, followed by the actual segmentation of the brain tumour and then the generation of the final result.

ANEXA A COD SURSĂ IMPLEMENTARE ELIMINARE CRANIU DIN IMAGINE

```
ALGORITHMSLIBRARY_API cv::Mat SkullStripping_KMeans(cv::Mat& image)
{
    cv::Mat erodedImg;
    image.copyTo(erodedImg);
    cv::dilate(erodedImg, erodedImg, cv::getStructuringElement(cv::MORPH_RECT,
cv::Size(13, 13)));
    cv::erode(erodedImg, erodedImg, cv::getStructuringElement(cv::MORPH_RECT,
cv::Size(11, 11)));
    cv::Mat result = GradientTest(erodedImg);
    cv::Mat maskSkullStripping = SkullStripping_UsingMask(erodedImg);
    // K-means clustering for skull stripping
    cv::Mat samples = erodedImg.reshape(1, erodedImg.rows * erodedImg.cols);
    samples.convertTo(samples, CV_32FC1);
    cv::Mat bestLabels, centers;
    cv::kmeans(samples, 3, bestLabels, cv::TermCriteria(cv::TermCriteria::EPS +
cv::TermCriteria::MAX_ITER, 10, 1.0), 3, cv::KMEANS_PP_CENTERS, centers);

    cv::Mat labelsImg = bestLabels.reshape(1, erodedImg.rows);
    labelsImg.convertTo(labelsImg, CV_8U);

    float maxIntensityCenter = -5000;
    int threshValue = 0;
    for (int col = 0; col < centers.cols; ++col)
    {
        float* imgCenterCol = centers.ptr<float>(0);
        for (int row = 0; row < centers.rows; ++row)
        {
            maxIntensityCenter = std::max(maxIntensityCenter,
imgCenterCol[row]);
            if (imgCenterCol[row] == maxIntensityCenter)
                threshValue = row;
        }
    }
    cv::Mat skullImage = cv::Mat::zeros(labelsImg.rows, labelsImg.cols,
labelsImg.type());
    for (int row = 0; row < labelsImg.rows; ++row)
    {
        uchar* currentRow = labelsImg.ptr<uchar>(row);
        uchar* skullImageRow = skullImage.ptr<uchar>(row);

        for (int col = 0; col < labelsImg.cols; ++col)
        {
            if (currentRow[col] == threshValue)
                skullImageRow[col] = 255;
            else
                skullImageRow[col] = 0;
        }
    }
    cv::Mat copy;
    erodedImg.copyTo(copy);
    for (int row = 0; row < copy.rows; ++row)
    {
        uchar* currentRow = skullImage.ptr<uchar>(row);
        uchar* copyRow = copy.ptr<uchar>(row);
    }
}
```

```

        for (int col = 0; col < copy.cols; ++col)
        {
            if (currentRow[col] == 255)
                copyRow[col] = 0;
        }
    }
    // Connected components on skull stripped image and finding the max area component
    std::pair<cv::Mat, int> connectedComp_result = ConnectedComponents(copy);
    int max_index = connectedComp_result.second;
    cv::Mat labeledImg = connectedComp_result.first;
    // Finding the contour of the brain region
    cv::Mat creier = ExtractTumorFromImage(labeledImg, max_index);
    std::vector<std::vector<cv::Point>> contours;
    std::vector<cv::Vec4i> hierarchy;
    cv::findContours(creier, contours, hierarchy, cv::RETR_EXTERNAL,
cv::CHAIN_APPROX_SIMPLE);
    // Finding the center of mass of the max area component
    cv::Point center;
    int sumofx = 0, sumofy = 0;
    for (int i = 0; i < contours[0].size(); ++i) {
        sumofx = sumofx + contours[0][i].x;
        sumofy = sumofy + contours[0][i].y;
    }
    center.x = sumofx / contours[0].size();
    center.y = sumofy / contours[0].size();

    double maxDist = 0.0;
    double minDist = 5000.0;
    for (int i = 0; i < contours[0].size(); ++i)
    {
        double distCentrePoint = cv::norm(center - contours[0][i]);
        maxDist = std::max(maxDist, distCentrePoint);
        minDist = std::min(minDist, distCentrePoint);
    }
    // Aproximate a contour for the entire brain region
    std::vector<std::vector<cv::Point>> full_contour(contours.size());
    cv::convexHull(contours[0], full_contour[0]);
    cv::Mat drawn_contour = cv::Mat::zeros(image.size(), CV_8UC1);
    cv::drawContours(drawn_contour, full_contour, -1, 255, 1); // , cv::noArray(), 1);
    cv::fillConvexPoly(drawn_contour, full_contour[0], 255);

    // Extracting the brain region from the original image
    cv::Mat brain_image = cv::Mat::zeros(cv::Size(image.rows, image.cols), CV_8UC1);
    for (int row = 0; row < image.rows; ++row)
    {
        uchar* imgAfterMaskRow = image.ptr<uchar>(row);
        uchar* drawnContourRow = drawn_contour.ptr<uchar>(row);
        uchar* brain_imageRow = brain_image.ptr<uchar>(row);

        for (int col = 0; col < image.cols; ++col)
        {
            if (drawnContourRow[col] == 255)
                brain_imageRow[col] = imgAfterMaskRow[col];
        }
    }

    return brain_image;

```

ANEXA B CON SURSĂ ESTIMARE ZGOMOT DINTR-O IMAGINE

```
ALGORITHMSLIBRARY_API double EstimateNoise(const cv::Mat& img)
{
    cv::Mat greyImg;
    cv::cvtColor(img, greyImg, cv::COLOR_BGR2GRAY);
    int height = greyImg.rows;
    int width = greyImg.cols;

    cv::Mat transformMat = (cv::Mat_<int>(3, 3) <<
        1, -2, 1,
        -2, 4, -2,
        1, -2, 1);
    cv::filter2D(greyImg, greyImg, -1, transformMat);

    cv::Scalar sigma = cv::sum(cv::sum(cv::abs(greyImg)));
    sigma = sigma * sqrt(0.5 * CV_PI) / (6 * (width - 2) * (height - 1));

    return sigma[0];
}
```


**DECLARAȚIE PRIVIND ORIGINALITATEA
LUCRĂRII DE LICENȚĂ / PROIECTULUI DE DIPLOMĂ / DISERTAȚIEI**

UNIVERSITATEA TRANSILVANIA DIN BRAȘOV
FACULTATEA MATEMATICĂ ȘI INFORMATICĂ
PROGRAMUL DE STUDII INFORMATICĂ APLICATĂ
NUMELE ȘI PRENUMELE ANDREI LAURENȚIU-ADRIAN
PROMOȚIA 2019-2022
SESIUNEA Iunie 2022
TEMA LUCRĂRII / PROIECTULUI / DISERTAȚIEI
..... SEGMENTAREA TUMORILOR CEREBRALE DIN RMN-URI
.....
CONDUCĂTOR ȘTIINȚIFIC lect. univ. dr. VLAD MONESCU

Declar pe propria răspundere că lucrarea de față este rezultatul muncii proprii, pe baza cercetărilor proprii și pe baza informațiilor obținute din surse care au fost citate și indicate conform normelor etice, în textul lucrării/proiectului, în note și în bibliografie.

Declar că nu s-a folosit în mod tacit sau ilegal munca altora și că nici o parte din teză/proiect nu încalcă drepturile de proprietate intelectuală ale altcuiva, persoană fizică sau juridică.

Declar că lucrarea/ proiectul nu a mai fost prezentat(ă) sub această formă vreunei instituții de învățământ superior în vederea obținerii unui grad sau titlu științific ori didactic.

În cazul constatării ulterioare a unor declarații false, voi suporta rigorile legii.

Data: 16.06.2022

Absolvent
ANDREI LAURENȚIU-ADRIAN
(nume, prenume, semnătură)

