

Programul de studii:

Informatică Aplicată

**LUCRARE DE LICENȚĂ**

Absolvent: **Laurențiu-Adrian Andrei**

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

Brașov, 2022



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**

Cuprins...................................................................................................................................................................3

Listă de figuri și coduri sursă............................................................................................................................5

Listă de acronime................................................................................................................................................6

1. Introducere.......................................................................................................................................................7

1.1 Actualitatea temei....................................................................................................................................7

1.2 Scopul și obiectivele lucrării...................................................................................................................7

1.3 Imagistica medicală..................................................................................................................................8

1.4 Structura temei.........................................................................................................................................8

2. Creierul uman................................................................................................................................................10

2.1 Anatomia creierului................................................................................................................................10

2.2 Tumorile cerebrale.................................................................................................................................11

3. Medii și concepte de dezvoltare...............................................................................................................12

3.1 Limbajul de programare C++...............................................................................................................12

3.2 Biblioteca OpenCV..................................................................................................................................12

3.3 Biblioteca Algorithms............................................................................................................................13

3.4 Biblioteca OpenMP................................................................................................................................14

3.5 Mediul de dezvoltare Qt.......................................................................................................................15

3.5.1 Qt Designer........................................................................................................................................15

3.6 Platforma CMake...................................................................................................................................15

4. Segmentarea tumorilor cerebrale...........................................................................................................17

4.1 Setul de imagini utilizat

4.2 Preprocesarea imaginilor

4.3 Extragerea caracteristicilor

4.4 Detectarea tumorii cerebrale

5. Interfața grafică

5.1 Crearea interfeței grafice

5.2 Testarea și simularea interfeței grafice

6. Direcții viitoare de cercetare

6.1 Idei pentru dezvoltarea ulterioară a aplicației

7. Concluzii

7.1 Concluzii generale

7.2 Elemente de originalitate

8. Bibliografie

Rezumat

Abstract

Anexe

DECLARAȚIE PRIVIND ORIGINALITATEA

**Listă de figuri și coduri sursă**

**Listă de acronime**

RMN – Rezonanță Magnetică Nucleară

DLL – Bibliotecă de Legături Dinamice

OpenCV – Open Computer Vision

OpenMP – Open Multi-Processing

GUI – Graphical User Interface

1. **Introducere**
   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. **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 creste 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. **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. **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.

1. **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].

* 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 metatalamus.
* Emisferele cerebrale – localizate la nivelul lojei supratentoriale, superior față de trunchiul cerebral și tentorium cerebelli.

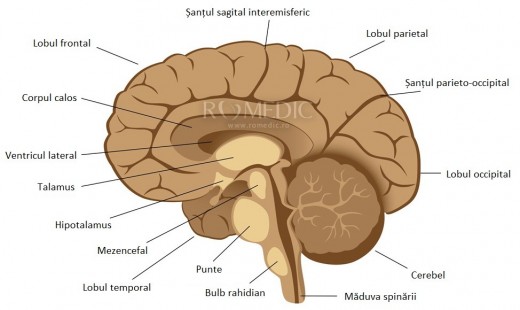


Fig. 2.1. Anatomia creierului uman

* 1. **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.

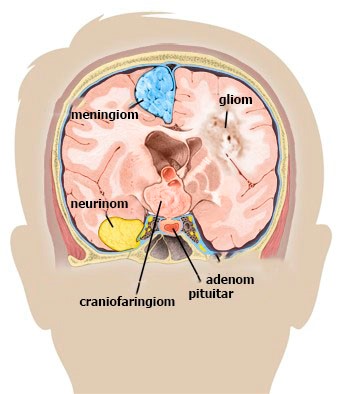


Fig. 2.2. Tipuri de tumori cerebrale

1. **Medii și concepte de programare**
   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.

* 1. **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].

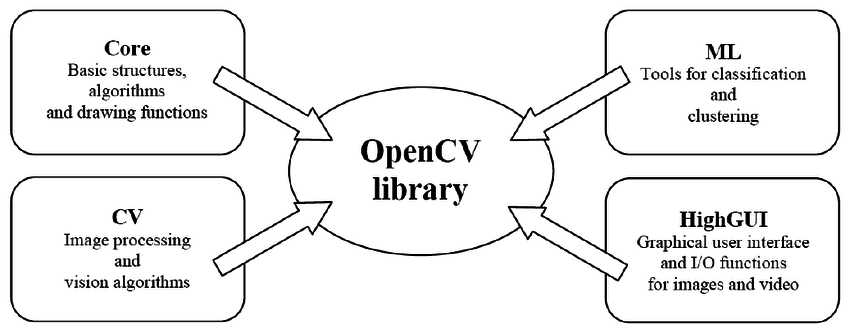


Fig. 3.1. Componentele OpenCV

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 algoritmii 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.

* 1. **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.

* 1. **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.

* 1. **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 si 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].

* + 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.
  1. **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”.

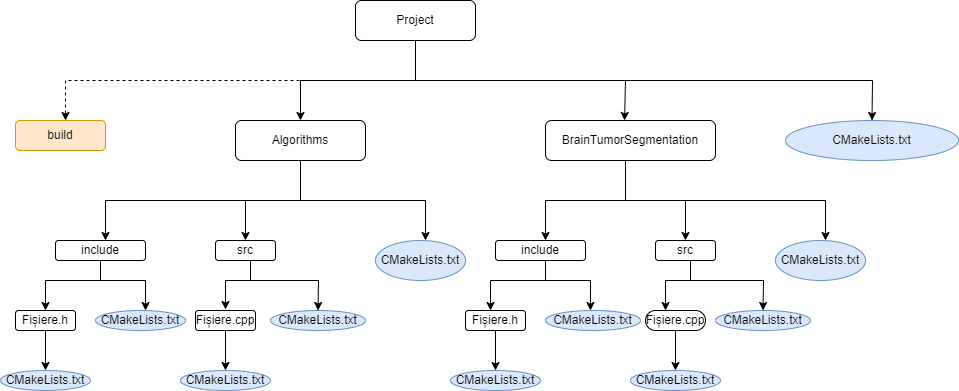


Fig. 3.2. Structura directorului sursă al proiectului.

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.

1. **Segmentarea tumorilor cerebrale**
   1. **Reducerea zgomotului din imaginile inițiale**

Zgomotul într-o imagine este o variație aleatorie a luminozității sau a informației unei culori. Acesta poate fi produs de senzorul de imagine al unei camere digitale.

Zgomotul imaginii poate varia de la valori aproape neobservabile într-o imagine digitală făcută cu lumină bună, până la valori foarte mari, în imagini optice sau radioastronomice, imagini din care se pot extrage informații puține prin procesare sofisticată.

Zgomotul într-o imagine ar trebui să fie cât mai mic pentru a obține informații corecte din aceasta.

Pentru reducerea zgomotului în imaginile din aplicație am comparat 4 algoritmi din biblioteca OpenCV:

* Average Blurring
* Median Blurring
* Gaussian Blurring
* Bilateral Filtering

**Average Blurring**

Acest algoritm este realizat prin strângerea unei imagini folosind un filtru normalizat. Pur și simplu, algoritmul preia media tuturor pixelilor de sub aria nucleului (kernel) și înlocuiește elementul central. Valorile nucleului trebuie specificate (lungimea și lățimea). Un filtru normalizat de mărimea 3x3 ar arăta astfel:

**Median Blurring**

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.

**Gaussian Blurring**

Algoritmul acesta folosește, în locul unui filtru normalizat, un nucleu de tip Gaussian. Acest algoritm este foarte eficient în eliminarea zgomotului de tip Gaussian dintr-o imagine.

**Bilateral Filtering**

Acest algoritm este eficient în reducerea zgomotului dintr-o imagine păstrând marginile din imagine. Bilateral Filtering 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.

Pentru măsurarea nivelului de zgomot dintr-o imagine am folosit o metodă numită [”Fast Estimation”](https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.1644&rep=rep1&type=pdf) pentru calcularea unei valori σ.

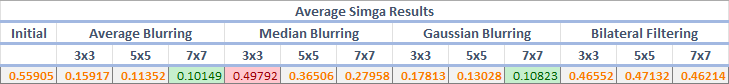
Primul pas al acestei metode este suprimarea structurii imaginii prin folosirea următorului operator Laplacian:

Apoi, valoarea σ poate fi calculată astfel:

,

unde W și H sunt lățimea și înălțimea imaginii.

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



Putem observa că valorile medii sunt cele mai mici în cazul aplicării filtrului Average Blurring folosind mărimea kernelul-ui de 7.

Astfel, vom implementa algoritmul pentru folosirea acestuia în dezvoltarea aplicației.

**Bibliografie**

5 – Niculescu, Cezar Th., *Anatomia și Fiziologia Omului. Compendiu – Ediția a II-a* , Editura Educațională Corint, 2014, p. 150.