

Міністерство освіти і науки України  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

**БАГАН Ярослав Андрійович**

**«Алгоритми текстурної сегментації на основі  
статистичних ознак / Texture segmentation  
algorithms based on statistical features»**

Студент групи КІм – 21  
Баган Ярослав Андрійович.

---

Науковий керівник  
к.т.н., доцент Мельник Г.М.

---

**Тернопіль – 2020**

## РЕЗЮМЕ

Кваліфікаційна робота на тему «Алгоритми текстурної сегментації на основі статистичних ознак» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «магістр» написана обсягом 71 сторінок і містить 40 ілюстрацій, 3 таблиці, 2 додатки та 50 джерел за переліком посилань.

Метою кваліфікаційної роботи є розроблення алгоритмів текстурної сегментації зображень на основі статистичних ознак.

Методи дослідження: методи та алгоритми теорії графів, методи комп'ютерного зору, методи штучного інтелекту.

Результати дослідження. Розроблено алгоритми текстурної сегментації зображень на основі матриць суміжності пар пікселів та просторових моментів, що дозволило сегментувати на зображення складні об'єкти – поперечні переріз пластів тканин, частини кровоносних судин та протоків залоз.

Розроблено структуру системи аналізу біомедичних зображень, що складається з: блоку інтерфейсу користувача, блоку попереднього оброблення зображення, блоку сегментації, блоку виділення мікрооб'єктів, блоку обчислення кута орієнтації, блоку обчислення числових ознак та блоку зберігання опису зображення у БД. Розроблювані алгоритми реалізовано у вигляді модуля розширення до програмного пакета ImageJ на мові Java.

Орієнтовні напрямки розвитку досліджень. Можливими напрямками подальших досліджень є продовження робіт по забезпеченню швидкодії розроблюваних алгоритмів.

**КЛЮЧОВІ СЛОВА:** ТЕКСТУРА, , СЕГМЕНТАЦІЯ, КОНТУР, МАРИЦІ СУМІЖНОСТІ, ПРОСТОРОВІ МОМЕНТИ.

## RESUME

The qualification graduation thesis on «Алгоритми текстурної сегментації на основі статистичних ознак / Texture segmentation algorithms based on statistical features» in the specialty 123 "Computer Engineering" of the educational degree "Master" is written in the volume of 71 pages and contains 40 illustrations, 3 tables, 2 appendices and 50 sources according to the list of references.

The purpose of the qualification work is to develop algorithms for textural segmentation of images based on statistical features.

Research methods: methods and algorithms of graph theory, methods of computer vision, methods of artificial intelligence.

Research results. Algorithms for textural segmentation of images based on matrices of adjacency of pairs of pixels and spatial moments have been developed, which allowed to segment complex objects into images - cross-sections of tissue layers, parts of blood vessels and gland ducts.

The structure of the biomedical image analysis system has been developed, consisting of: user interface unit, image pre-processing unit, segmentation unit, micro-object selection unit, orientation angle calculation unit, numerical feature calculation unit and image description storage unit in the database. The developed algorithms are implemented in the form of an extension module to the ImageJ software package in Java.

Approximate directions of research development. Possible areas of further research are the continuation of work to ensure the speed of the developed algorithms.

**KEY WORDS:** TEXTURE,, SEGMENTATION, CONTOUR, ADJACENT MARITS, SPATIAL MOMENTS.

## ЗМІСТ

Перелік умовних скорочень.....	7
Вступ.....	8
1 Аналіз просторової структури біомедичних зображень .....	11
1.1 Систематизація методів сегментації зображень .....	11
1.2 Гістологічні зображення і системи опрацювання .....	20
1.4 Аналіз завдань роботи та постановка задач дослідження.....	29
1.5 Висновки до розділу .....	30
2 Алгоритми текстурної сегментації на основі статистичних ознак.....	31
2.1 Операції попереднього опрацювання зображення.....	31
2.2 Матриці суміжності .....	36
2.3 Алгоритми текстурної сегментації на основі моментів .....	42
2.4 Висновки до розділу .....	49
3 Експериментальне дослідження розроблених алгоритмів.....	50
3.1 Структура програмного забезпечення .....	50
3.2 Автоматизація повторюваних операцій і побудова методик.....	53
3.3 Реалізація бази даних.....	68
3.3 Експериментальні дослідження.....	74
3.4 Висновки до розділу .....	78
Висновки.....	79
Список використаних джерел.....	80
Додаток А Лістинг коду програми.....	86
Додаток Б Світлокопії виданих публікацій .....	93

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

МС – матриці суміжності

ПМ – просторові моменти

БМЗ – біомедичне зображення

ГЗ – гістологічне зображення

СКМ – системи комп'ютерної мікроскопії

## ВСТУП

Однією з основних задач в області комп'ютерного зору є задача отримання інформації і даних із цифрових зображень. Сегментація - це процес розбивки зображення на сегменти, які представляють собою множини пікселів, об'єднаних за тими або іншими ознаками. Сегментація дозволяє зменшити обсяг інформації на зображенні і полегшити його аналіз. Крім того, рішення задачі сегментації зображень є ключовим для багатьох інших задач і методів комп'ютерного зору, таких як: розпізнавання, ідентифікація та детекції об'єктів на зображеннях.

Поширеним типом сегментації зображень є текстурна сегментація, вона здійснюється за допомогою аналізу зображення на наявність певних патернів які періодично повторюються і поширюються на деяких областях зображення. Результатом такої сегментації є множина сегментів або областей виділених на зображенні. Всі пікселі в таких сегментах схожі за своєю текстурною характеристикою, тоді як сусідні сегменти своєю текстурою значно відрізняються.

На даний час ще не створено універсальних теоретичних основ і алгоритмічних рішень в області текстурної сегментації зображень, які б давали задовільні результати у всіх випадках практичного застосування. Зокрема, алгоритми текстурної сегментації поділяються в залежності від необхідності наявності попереднього знання про тип текстур, присутніх на зображенні. Іншою ознакою поділу алгоритмів текстурної сегментації є математичні процедури які використовуються для характеристики текстур в даних алгоритмах, за нею алгоритми поділяються на дві основні категорії, структурні та статистичні.

Структурні методи підходять для текстур, де патерни або текстурні примітиви можна описати, використовуючи більшу різноманітність властивостей, ніж просто тональні властивості окремих пікселів, зокрема опис

фігури і форми. Проте вони не ефективні коли розміри текстурних примітивів зображення співставні з розмірами пікселів. Актуальним вирішенням цієї проблеми є розробка і використання алгоритму для сегментації зображення за статистичними ознаками такими як математичне сподівання, дисперсія, просторова кореляція тощо.

Таким актуальним є розробка алгоритму текстурної сегментації на основі статистичних ознак, який дозволяє визначати набір текстурних ознак, що характеризують певний однорідний текстурний сегмент, наявний на зображенні. Надалі виявлений набір ознак використовується для ідентифікації та окреслення меж відповідного текстурного сегмента.

Мета і завдання дослідження. Метою кваліфікаційної роботи є розроблення алгоритмів текстурної сегментації зображень на основі статистичних ознак.

Об'єктом дослідження є процес сегментації цифрових зображень.

Предметом дослідження є методи і алгоритми текстурної сегментації зображень.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

1. систематизувати текстурні характеристики зображень;
2. провести аналіз методів сегментації зображень;
3. реалізувати алгоритми текстурної сегментації зображень;
4. розробити об'єктну модель та графічний інтерфейс системи сегментації зображень;
5. оцінити якість сегментації розробленими алгоритмами.

Методи дослідження базуються на використанні методів гістограмного аналізу зображень, методів комп'ютерного зору, положень теорії алгоритмів і аналітичної геометрії..

Наукова новизна одержаних результатів. Розроблено алгоритми текстурної сегментації зображень на основі матриць розподілу рівнів сірого та просторових моментів, що дозволило підвищити якість сегментації.

1. Практичне значення отриманих результатів. Реалізовано алгоритми текстурної сегментації зображень в складі системи опрацювання біомедичних зображень. Розроблено структуру системи аналізу біомедичних зображень, що складається з: блоку інтерфейсу користувача, блоку попереднього опрацювання зображення, блоку сегментації, блоку виділення мікрооб'єктів, блоку обчислення кута орієнтації, блоку обчислення числових ознак та блоку зберігання опису зображення у БД. Розроблені алгоритми реалізовано у вигляді модуля розширення до програмного пакета ImageJ на мові Java.

Публікації та апробація випускної кваліфікаційної роботи. Отримані результати апробовані в межах III науково-практичної конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Тернопільського національного економічного університету та опубліковано дві тези доповіді по темі роботи [1,2].

Кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків. У першому розділі систематизовано алгоритми аналізу просторової структури біомедичних зображень.

В другому розділі розроблено алгоритми текстурної сегментації на основі статистичних ознак.

У третьому розділі здійснено програмну реалізацію розроблених алгоритмів. Для оцінки якості сегментації обчислено відстань Хаусдорфа між контурами однорідних сегментів. Проведено експериментальне дослідження розроблених алгоритмів.



# 1 АНАЛІЗ ПРОСТОРОВОЇ СТРУКТУРИ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ

## 1.1 Систематизація методів сегментації зображень

При обробці зображень, сегментація зображень розглядається як дуже значущий і важливий процес. Щоб отримати будь-яку цінну інформацію із зображення, на ньому слід відокремити регіони з однаковими характеристиками, зокрема, знайти регіони з схожою текстурою або виділити об'єкти переднього та заднього плану. Процедурою для вирішення цих завдань є сегментація зображень.

В даному розділі ми оглянемо різні методи сегментації зображень. Можна виділити кілька основних підходів до класифікації методів сегментації текстур. Методи текстурної сегментації розробляються на основі двох основних підходів: виділення областей та виділення границь. Методи на основі границь намагаються виявити край текстур між пікселями, яким відповідають різні текстурні розподіли. Ми зосередимося на аналізі методів на основі областей. Вони намагаються знайти групи або кластери пікселів зі схожими текстурними властивостями. Залежно від використовуваних ознак текстурних областей зображень методи сегментації на основі аналізу областей можна розділити на статистичні, структурні, фрактальні, спектральні і комбіновані методи та інші. Повну класифікацію методів сегментації зображень зображено у вигляді деревної діаграми показано на рисунку 1.1.

Алгоритми сегментації зображень зазвичай класифікуються на основі двох властивостей: розрив та подібність. При сегментації на основі розриву сегментація зображень робиться в місцях, де відбувається швидка зміна інтенсивності, зокрема, краї зображення. На основі подібності зображення сегментуються на кілька областей, а пікселі в кожній області подібні на основі набору заздалегідь визначених критеріїв. Порогові показники, зростання регіонів, розділення та злиття регіонів - це деякі методи сегментації, засновані на подібності.



Рисунок 1.1 – Загальна класифікація методів сегментування зображення

Сегментація на основі порогів зазвичай використовується для сегментації зображень у градаціях сірого. Це є найпростішим методом сегментації. Обраний поріг повинен відокремлювати об'єкти переднього і заднього плану. Обробляються значення пікселів, які падають нижче порогового значення як фон, а значення пікселів, які падають вище порога, розглядаються як переднього плану. Порогова сегментація ефективна на зображеннях з яскравими предметами на темному фоні. Приклад роботи порогової сегментації зображено на рисунку 1.2.

У більшості випадків поріг вибирається вручну. Вихідним сигналом порогового зображення є двійкове (чорно-біле) зображення. У деяких випадках початкове порогове значення супроводжується поділом первинних двох категорій на кілька підкатегорій пов'язаних регіонів. У таких випадках для отримання більше двох сегментів використовується більше одного порога (декілька порогів). Ці кілька сегментів диференціюються за своїми сірими значеннями. У разі кількох порогів значення пікселів нижче початкового

порогу призначаються першому сегменту, а значення пікселів між першим і наступним порогом присвоюються другому сегменту.

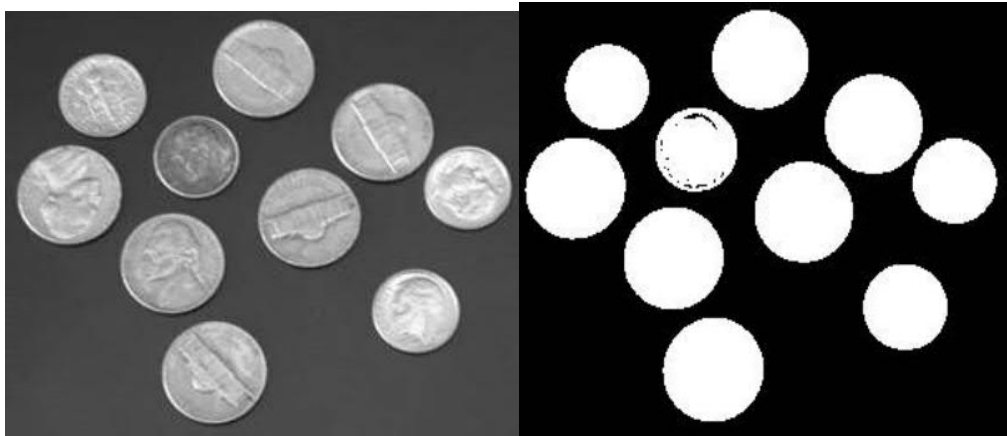


Рисунок 1.2 – Приклад порогової сегментації

#### Сегментація на основі виділення границь

Границя на будь-якому зображенні – це набір сусідніх пікселів, де відбувається швидка зміна значень інтенсивності (сірого або кольорового). Об'єкти на будь-якому зображенні можуть бути представлені їх границями. Отже, об'єкти, присутні на будь-якому зображенні, можуть бути сегментовані шляхом детекції їх границь. Границі можна класифікувати як ідеальні і справжні (не ідеальні, градієнтні). Модель ідеальної і справжньої границі наведена на рисунку 1.3

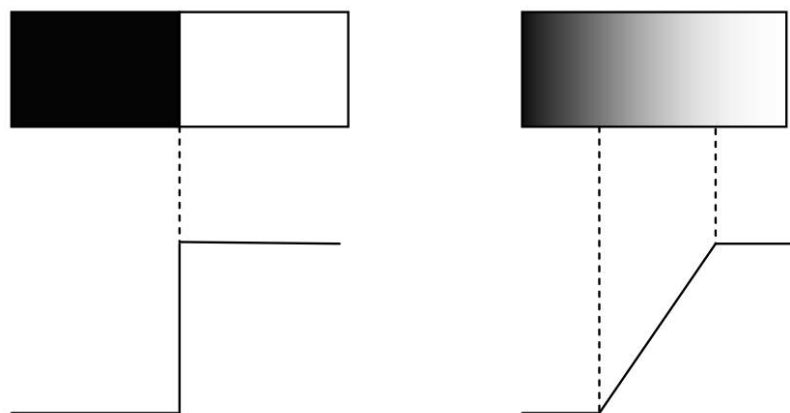


Рисунок 1.3 – Модель ідеальної і справжньої границі

Сегментація на основі виділення границь містить такі кроки:

- Обчислення границь об'єктів на заданому зображенні.
- Детекція границь які пов'язані між собою і утворюють замкнуті межі.

Загалом, для детекції границь використовується метод похідних. Тим не менше, метод похідних дуже чутливий до шуму. Похідна першого порядку обчислюється з використанням градієнта на зображенні. Похідна другого порядку обчислюється Лапласіаном на зображенні. Похідна другого порядку, як правило, більш чутлива, ніж похідна першого порядку. Результат використання методу похідних для детекції границь зображено на рисунку 1.4



Рисунок 1.4 – Приклад виділення границь методом похідних

Наявність країв на зображенні визначається за величиною похідної першого порядку. Знак похідної другого порядку використовується для визначення того, або знаходиться піксель краю на темній або світлій стороні границі.

Метод нарощування областей - це техніка, яка дозволяє безпосередньо визначати області або регіони на зображеннях. Нарощування областей - це найпростіша сегментація в основі якої лежить процес групування пікселів або субрегіонів у більші області на основі деяких заздалегідь визначених критеріїв. Концепція методу нарощування областей полягає в тому, щоб перевірити сусідні пікселі початкових точок, а потім визначити або ці сусідні пікселі подібні до початкових точок або ні. Приклад ітераційного процесу нарощування областей зображено на рисунку 1.5

1	1	9	9	9	1	1	9	9	9	1	1	9	9	9	1	1	9	9	9
1	1	9	9	9	1	1	9	9	9	1	1	9	9	9	1	1	9	9	9
5	1	1	9	9	5	1	1	9	9	5	1	1	9	9	5	1	1	9	9
5	5	5	3	9	5	5	5	3	9	5	5	5	3	9	5	5	5	3	9
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Рисунок 1.5 – Ітераційний процес нарощування областей

У сегментації, заснованій на вирощуванні областей, області формуються шляхом рекурсивного включення сусідніх пікселів, подібних та пов'язаних із початковими пікселями. Процес починається з початкового пікселя на області, також початковою може вибрана якась група пікселів. Початковим пікселем може бути якийсь заздалегідь визначений піксель, або його можна вибрати випадковим чином. Після до початкового пікселя додаються сусідні пікселі з подібними рисами з рівнем сірого, текстурою або кольором, цим самим утворюючи область.

## 1.2 Підходи до побудови векторів текстурних ознак зображень

Як і колір, текстура зображення вкрай важлива для зорового сприйняття людини. Вона містить інформацію про структуру поверхонь, об'єктів на зображенні. Кожний з нас здатний розпізнати текстуру, дати точне визначення цьому поняттю складніше. Текстура не визначаємо для окремого взятого пікселя, вона залежить від розподілу рівня яскравості в зображенні. Текстура має властивості періодичності й масштабу, може бути описана за допомогою виділення основних напрямків, визначення контрастності, різкості. Аналіз текстури відіграє істотну роль при порівнянні зображень, доповнюючи колірну характеристику. У науковій літературі представлена множина різних способів представлення інформації про текстуру зображення й порівняння

зображень на її основі. Текsturні ознаки розділяють на статистичні, геометричні, модельні й спектральні.

До спектральних ознак відносять ознаки, що описують текстуру в частотній області. Вони базуються на ідеї розкладання сигналу по деякому базису й використанні коефіцієнтів розкладання як елементів вектора ознак.

Наявність регіонів на зображенні, що містять текстури, є суттєвим джерелом інформації для вирішення задачі сегментації зображень. Текстури присутні на багатьох природних та штучних зображеннях, текстура описує різноманітні поверхні на зображенні, для визначення ознак текстур використовують різні методи побудови векторів текстурних ознак.

Побудова векторів текстурних ознак це процес характеристики регіонів на зображенні за їх текстурними властивостями, до яких можуть відноситися відносяться такі ознаки як однорідність, шорсткість і регулярність.

Методи отримання такого набору ознак можна розділити на чотири типи:

- статистичні (використовують кореляційні і коваріаційні характеристики текстур),
- геометричні (двовимірні соти Вороного і методи, що використовують структурні характеристики текстури),
- методи, що використовують фрактальні параметри і випадкові поля
- методи, засновані на цифровій обробці сигналів (просторовочастотна фільтрація і вейвлет-перетворення).

Коваріаційні матриці описують просторові зв'язки пар яскравостей елементів текстури (мозок людини для розпізнавання текстур використовує статистику лише першого і другого порядку). Для фіксованих відстані і кута розраховуються 14 ознак: другий кутовий момент, контрастність, ентропія, 3-кореляція та інші. Коваріаційні матриці обчислюються для кількох пар  $(r, \theta)$ , з метою зменшення обчислювальної складності кут і вектор відстаней обмежують. Для додавання інваріантності щодо повороту можливе використання усереднених значень таких матриць. На практиці в тому числі

використовуються суттєві для зорового сприйняття евристичні ознаки Тамури.

Зернистість — ознака, пов'язана з розміром утворюють текстуру примітивних елементів і з відстанню між значущими просторовими коливаннями відтінків сірого;

Контраст — міра того, наскільки сильно і різко може змінюватися колір на зображенні (ознака є об'єднанням впливів наступних факторів: розкид яскравості, відокремленість білого і чорного кольору на яркостній гистограмме, період затримки деякого шаблону).

Спрямованість - ознака, вимірюваний за допомогою гистограми локальних напрямків контурів.

Лінійність - ознака, що показує, наскільки прямолінійні контури на зображенні.

Регулярність - загальна мінливість перших чотирьох ознак між різними частинами зображення.

Грубість - суб'єктивна оцінка грубості переходів на зображенні (виразності текстури).

### 1.3 Огляд методів структурного аналізу зображень

У структурних підходах текстурного аналізу очікується, що текстури складаються з простих текстурних примітивів згідно з визначеним правилом розміщення, а ці примітиви регулярно або майже регулярно повторюються[10]. Виділяють три основні підходи в структурному розпізнаванні: синтаксичний – орієнтоване розпізнавання, розпізнавання графових образів, розпізнавання деревовидних структур з модифікацією визначення граматики, визначенням функції ранжирування і зміною правил підстановки. Для того, щоб врахувати випадковий характер текстур, вводять

стохастичні мови і граматики. Отриманий опис відрізняється чутливістю до шумових спотворень і погано підходить для аналізу текстур з неявно вираженими періодичними властивостями (тобто в першу чергу реальних зображень). Яскравості елементів, розташованих уздовж рядка, представляються як дерево співвідношень, що описує рекурсивне розбиття для функції яскравості. Таке розбиття здійснюється в точці найменшого з відносних мінімумів. Такі точки і крайові значення функції яскравості використовуються на наступному кроці рекурсії. Існують ознаки, що базуються на довжинах серій (під терміном довжина серії розуміється число елементів рядка растру, що мають ту саму яскравість). Опишемо далі коротко ці ознаки:

1. вага лінії що має постійну оптичну щільність, характеризується тим, що для будь-якого рівня сірого вага кожної лінії збільшується в міру збільшення довжини;

2. розподіл рівнів сірого, ознака має мінімум в тих випадках, коли число ліній постійної оптичної щільності рівномірно розподілено за рівнями сірого;

3. розподіл довжини ліній постійної оптичної щільності, має мінімум при рівномірному розподілі;

4. відносна кількість ліній постійної оптичної щільності, значення ознаки максимально, коли всі лінії мають малі довжини.

Приклади регулярних текстур представлені на рисунку 1.6. Одним із методів в структурному є також є відносна густина екстремумів [9]. Вчені Розенфельд та Трой запропонували рахувати кількість екстремумів на одиницю для даної текстури. Екстремум виділяється тільки в одній площині по рядку і так по кожному з них.





Рисунок 1.6 – Приклад зображення з регулярною текстурою

В будь якому рядку піксель  $i$  є відносним максимумом, якщо його відтінок сірого задовільняє умову:

$$g(i) \leq g(i + 1) \text{ та } g(i) \leq g(i - 1), \quad (1.1)$$

де  $g(i)$  - відтінок сірого.

Піксель  $i$  відносний максимум якщо:

$$g(i) \geq g(i + 1) \text{ та } g(i) \geq g(i - 1). \quad (1.2)$$

Алгоритм запропонований вченими відмічає кожен піксель який задовільняє умову (2.2) та (2.3). В одновимірному випадку є дві властивості які можуть бути зв'язані з кожним екстремумом це висота і ширина. Висота максимума може бути визначена як різниця між значенням максимума і найвищим прилягаючим мінімумом. Висота (глибина) мінімуму може бути визначена як різниця між значенням мінімуму та найнижчим прилягаючим максимумом. Ширина максимума це відстань між двома прилягаючими

мінімумами. А ширина мінімуму це відстань між двома сусідніми максимумами. Це проілюстровано на рисунку 1.7.

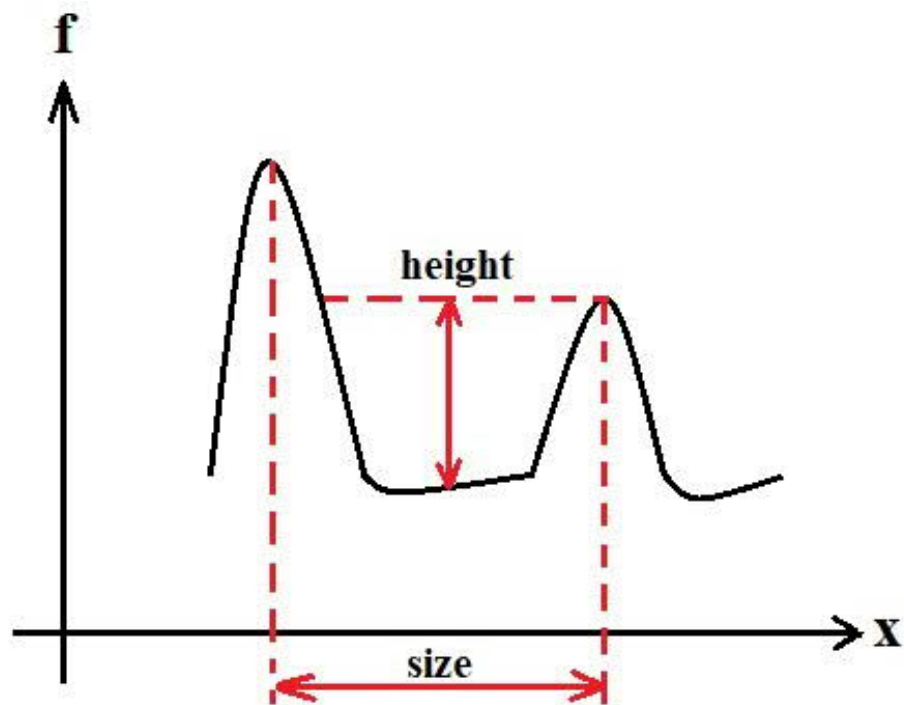


Рисунок 1.7 — Приклад екстремумів з однієї площини

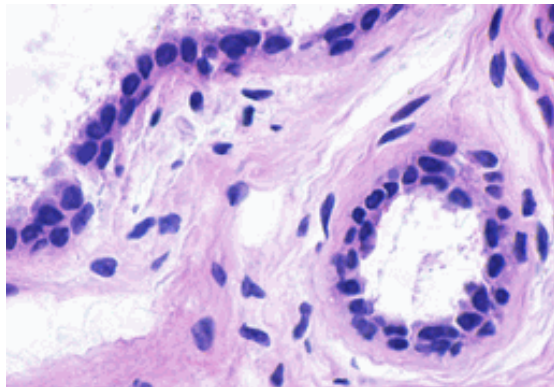
## 1.2 Гістологічні зображення і системи опрацювання

Дослідження корисних ознак для класифікації захворювань засновані на певних візуальних атрибутах, що названі клініцистами особливо важливими для діагностування та визначення стадії хвороби. Переважна більшість з цих ознак є ознаками ядер, і багато з них були визнані характерними при аналізі як цитопатологічних і гістопатологічних зображень (таблиця 1.5). Інші ознаки, які мають дискримінаційну (розділювальну) здатність включають опис вигляду границі різних груп мікрооб'єктів: протоків, строми, трубчастих і залізистих структур.

Таблиця 1.5 – Характерні ознаки мікрооб'єктів

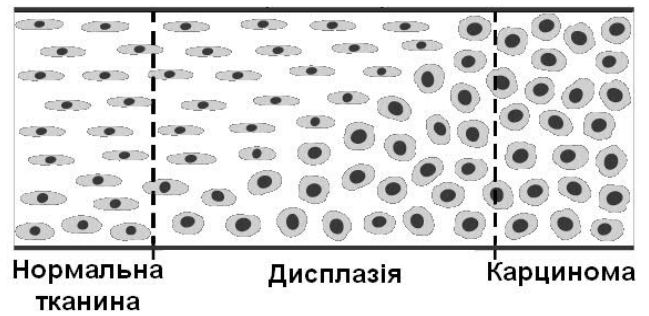
Категорія	Ознаки
Розмір і форма	Площа
	Еліптичні Особливості: довжина великої і малої осі, ексцентриситет, орієнтація, еліптичні відхилення
	Ознаки описаного багатокутника: опукла область, дефекти опуклості, щільність
	Ознаки заповненості зображення: Заповнена площа, число Ейлера
	Ознаки Обмежувального прямокутника: екстент, співвідношення сторін
	Ознаки границь: Периметр, радіуси, Фур'є енергії периметру, кривизна периметра, енергія вигину, фрактальної розмірності периметру
	Інші ознаки Форми: Еквівалентний діаметр, сферичність, компактність, форма інерції
	Центру мас
	Симетрія відображення
Радіометричні та денситометричні	Канали багатоканального зображення, Інтенсивність
	Оптична щільність, Оптична щільність інтегрована, і середня оптична
	Колірний тон
Текстура	Матриця розподілу рівнів сірого: інерція, енергія, ентропія, однорідність, максимальна ймовірність, відтінок кластеру, кластер
	Фрактальна розмірність
	Вейвлет ознаки: Енергії детальних зображень і зображень малої роздільної здатності
	Ентропія
Хроматин специфічні	Площа, інтегрований оптична щільність, середня оптична щільність, число областей, компактність, відстань, центр мас

На рисунку 1.5,а показано приклад гістологічного зображення (ГЗ), що використовується при діагностуванні ракових процесів. Характерним для ракових процесів є зміна структури (порядку) тканин (рисунок 1.5, б). Ця властивість є ключовою в задачах сегментації ГЗ.



- містять непохідний елемент;
- непохідні елементи мають структурні залежності;
- об'єкти оточені однорідним або текстурним фоном;
- відомий розподіл кольорів.

а)



Зміна структури тканини при злоякісному процесі

б)

Рисунок 1.5 – Гістологічне зображення (а) та зміна структури тканин при злоякісних процесах (б)

Розглянемо системи опрацювання БМЗ.

Біомедичне зображення – це структурно-функціональний образ органів людини і тварин, призначений для діагностики захворювань і вивчення анатомофізіологічної картини організму. Зображення отримані використанням технічних засобів візуалізації у медицині та біології. Завдяки візуалізації різних процесів, які відбуваються в живих організмах, вдається вивчати механізми функціонування клітин, тканин та органів людини і тварин [1,2].

Біомедична інженерія (англ. biomedical engineering) — галузь науки і техніки, яка поєднує інженерно-технічні та медико-біологічні знання, засоби і методи для створення, вдосконалення і дослідження природних і штучних біологічних об'єктів, техніки, матеріалів і виробів медичного призначення, технологій і технічних систем діагностики, лікування, реабілітації і профілактики захворювань людини, а також програмного забезпечення та інформаційних технологій для вирішення прикладних і фундаментальних проблем біології і медицини.

Перший крок в цифровому опрацюванні зображення - перетворення зображення в форму, яка може бути збережена в пам'яті машини. Ця трансформація виконується світлочутливої системою, званою електроннооптичний цифровий перетворювач, який видає закодовані числа, що є мірою інтенсивності світла. Алгоритм реєструє зображення з заданими регулярними часовими інтервалами, і виводить електричний струм, що перетворений в рядок дискретних числових значень. Значення представляють собою розподіл інтенсивності в зображенні оптичному. Цей процес іменують оцифровуванням, або квантуванням, а збережене числове представлення вихідного оптичного зображення іменують цифровим зображенням (ЦЗ). Цифрове зображення - це числова абстракція, якою можна оперувати, як і іншими даними, збереженими в пам'яті машини. Існує великий перелік операцій, які можуть бути виконані щодо цифрового зображення; проте всі такі трансформації можуть бути причислені до однієї з двох категорій: обробка зображення та аналіз зображення.

Словосполучення «аналіз зображення» відносимо до комп'ютерних процедур, які виробляють описову інформацію про ЦЗ. Типові приклади - визначення кількості клітин в полі зору мікроскопа або вимірювання діаметра ядра певної клітини. Такий тип числового аналізу називається виділенням ознак, а кожне число котре використовується, щоб описати цифрове зображення, називають характеристикою зображення. У найширшому сенсі, характеристика зображення - будь-який признак, який може використовуватися, щоб описати ЦЗ, на основі таких, зокрема, властивостей, як форма або площа ядра клітини, розподіл клітин в змішаній популяції за розмірами, і т.д. На додаток до виділення ознак, процедури, виконують аналіз зображення, можуть також бути використані для твору логічного аналізу інформації про проаналізованому зображенні, зокрема, або містить сегмент зображення (поле зору в мікроскопі), котрий повинен бути проаналізований. Інакше цей сегмент повинен бути відхилений і слід перейти до аналізу наступного сегмента. Процедури обробки зображень, які передують аналізу зображень, можуть

дуже відрізняться за складністю і, відповідно, за вимогами до устаткування для їх реалізації. Самі прості процедури аналізу ЦЗ розробляються, щоб істотно підвищити точність та продуктивність при вирішенні задач, які інакше мали б бути виконані вручну, зокрема підрахунок числа колоній бактерій в чашках Петрі. Типова система цього типу могла б складатися тільки з оптичного проектора, цифрового графічного планшета і мікрокомп'ютера, або, в іншому випадку, мінікомп'ютера, відеокамери та відеомонітора. Така система не потребує особливих витрат та може бути використана зразу ж нетренованим персоналом [3,4].

Другом крайнім випадком застосування аналізу зображень, що потребує досить складних технічних та ПЗ та високої кваліфікації користувача, можуть стати алгоритми автоматичного розпізнавання і класифікації аномальних клітин на гістологічних препаратах. Такі алгоритми потребують високошвидкісного вимірювання набору різних характеристик зображення препарату з метою виділення на складному фоні окремих клітин, отримання їх кількісних ознак і віднесення на основі заданих критеріїв до того або іншого класу. Створювані системи для складного аналізу зображень, вимагають, як мінімум, високоякісної системи уведення зображень в комп'ютер, великого об'єму оперативної пам'яті і надзвичайно високошвидкісного процесора. Більшість таких систем містять в собі спеціальні апаратні засоби (спецпроцесори), які виконують опрацювання зображення у багато разів швидше ніж звичайні програмовані універсальні комп'ютери. Сучасні системи містять в собі складні пристрої формування і введення зображень, зокрема автоматизовані мікроскопи, що дозволяють не тільки позиціонувати аналізовані клітини, а й виконують процедури настроювання оптичної системи приладу для отримання оптимальної якості зображення та його реєстрації з високим просторовим і фотометричним дозволом. До високої основної вартості згаданих систем додаються додаткові витрати, котрі пов'язані із їх обслуговуванням, розробкою спеціалізованого програмного забезпечення, навчанням персоналу.

Іншим класом процедур обробки зображень є процедури покращення якості зображення. Вони відрізняються від процедур аналізу зображення тим, що ціллю обробки є не отримання описової інформації про зображення, а трансформація зображення таким способом, щоб зробити зображення більш інформативним для людини. Приклади процедур покращення якості зображення: віднімання двох зафіксованих в різний час зображень, з метою детекції руху; покращення контрасту; усунення шуму, підкреслення меж об'єктів і т.п. Процедури покращення якості зображення можуть також використовуватися, щоб компенсувати деградацію оригінального зображення, викликаного дефектами або фізичними обмеженнями в системі формування зображення, зокрема, неоднорідністю освітлення поля зору в мікроскопі. Так як ЦЗ є абстракцією в пам'яті комп'ютера, то ним можна управляти способами, які не можуть бути реалізовані фізичними пристроями, типу призми, лінзи, або фільтру. Деякі процедури покращення якості зображення використовують корекцію цифрового зображення за допомогою цифрового аналога повної ретельно розрахованої оптичної системи, щоб виправити спотворення зображення [5-8].

Системи, що створюються спочатку для цілей покращення якості зображення, повинні задовольняти ряду додаткових вимог - вони повинні бути гнучкими і зручними для користувача, тобто дуже інтерактивними. Система покращення якості зображення - найбільш імовірний тип для лікаря, який використовує, зокрема, мікроскоп або рентгенівський апарат як щоденний дослідний інструмент, і чий вимоги до опрацювання можуть швидко змінюватися. Отже, система повинна бути досить гнучкою, щоб дозволити користувачеві випробувати багато різних процедур обробки і мати можливість спостерігати результати в максимально короткий час. Описані вище системи зазвичай мають можливість виконання опрацювання в реальному часі. Згадане означає, що в цифровій формі оброблене зображення буде відтворено на відеомоніторі без затримки. Очевидно, що різні застосування методів обробки

зображення висувають різні вимоги до необхідних для їх реалізації технічних засобів, методичного забезпечення та програмного.

Існуючі сьогодні методи візуалізації засновані на різноманітних фізичних взаємодіях електромагнітного випромінювання з структурами клітин і тканин живих організмів.

На даний час аналіз препаратів гістологічних і цитологічних мазків в діагностичних лабораторіях здійснюється візуально. Цей процес є трудомістким та рутинним. Тому з'явилися системи комп'ютерної мікроскопії (СКМ) – програмно-апаратні комплекси для цифрової обробки мікроскопічних зображень. СКМ є програмно-апаратними системами, до складу яких входять моторизований керований мікроскоп, відеокамера, комп'ютер, функціональні програми-методики.

САМ в цілому поділяють на дослідницькі та спеціалізовані [7, 17]. Дослідницькі САМ використовуються для розроблення нових методів діагностування. Спеціалізовані СКМ забезпечують виконання одного стандартизованого клінічного дослідження. Особливістю дослідницьких СКМ є використання багатофункціональних мікроскопів і камер. Камери зі збільшеною роздільною та чутливістю здатністю. Алгоритм дослідження не задана наперед, а формується користувачем.

Для задач мікробіології на світовому ринку пропонують СКМ або окремі їх компоненти такі фірми як Carl Zeiss (Німеччина), Leica, Olympus, Nikon, Микромед (Росія), Motic (Китай), Konus (Італія).

Опишемо процес дослідження з допомогою СКМ. На початку препарат встановлюється в мікроскоп на предметний стіл. На екран монітора виводиться зображення поля зору. Проводяться всі необхідні налаштування і захоплення зображення у реальному часі. Зображення може бути оброблене за допомогою фільтрів. На зображення можна розмістити текстові коментарі і калібрувальний маркер. СКМ дозволяє виділяти мікрооб'єкти на зображенні в ручному напівавтоматичному та автоматичному режимах. Детектовані об'єкти вимірюються автоматично і результати обчислень відображаються в



табличній формі. По будь-якому із вимірних параметрів мікрооб'єкти можуть класифікуватися. Результати дослідження виводяться на друк і зберігаються в базі даних.

Було досліджено склад і функції існуючих програмних засобів для побудови структури узагальненої СКМ (таблиця 1.1).

Порівняння проводилось за наступними параметрами:

- спосіб отримання інформації: зображення приймається із давача в реальному часі (підтримка технологій MCI і TWAIN), завантаження з диску;
- режими роботи алгоритмів сегментації: ручний (оператор в ручному режимі виділяє мікрооб'єкти), автоматизований (оператор попередньо проводить навчання алгоритму) або автоматичний (параметри алгоритму обираються автоматично);
- обчислення числових ознак мікрооб'єктів: площа, периметр, ядерно-цитоплазматичне відношення, діаметр, кут між двома відрізками, тощо;
- попереднє опрацювання зображень: видалення шумів, корекція контрасту, яскравості, а також фільтрація, детекція області інтересу, тощо;
- представлення результатів у вигляді діаграм, гістограм, графіків, тощо;
- обчислення статистичних признаков: математичне сподівання, середньоквадратичне відхилення, мінімальне (максимальне) значення, тощо;
- використання сценаріїв – присутність вбудованої мови для пакетного оброблення даних;
- взаємодія з іншим ПЗ: FoxPro, MS Access, MS Word, MS Excel,;
- детальна технічна документація.

Типова структура апаратної частини СКМ складається з мікроскопа, відеокамери або фотокамери, комп'ютера, принтера і монітора. Підсистема вводу зображення складається з мікроскопа прохідного світлова, фотоадаптера, камери, пристрою заміни об'єктива, засобу фокусування, пристрою переміщення предметного столика, пристрою подачі зразків та засобу освітлення.

Таблиця 1.1 – Характеристика СКМ

№	Назва програми	Підтримка додаткових форматів файлів	Трансформація файлу в інший формат	Редагування зображення	Конвертація в інші кольорові бази	Виділення мікрооб'єктів в ручному / автоматизованому / автоматичному режимах	Визначення координат обмежуючого прямокутника	Використання сценаріїв	Формування звітів	Передача даних в програмне забезпечення 3-х сторін	Друк	Налаштування параметрів роботи програми	Калібрування камери	Наявність системи допомоги
1.	ImageTool в.3.00	+	+	+	+	+/-/-	-	+	+	-	+	+	-	+
2.	«ИМАДЖЕР-ЦГ»	-	+	+	-	+/-/-	-	-	+	-	+	+	-	+
3.	ВидеоТест-Морфо 3	+	+	+	+	+/+/-	-	+	+	-	+	+	-	+
4.	Micromed Images	+	+	+	-	+/-/-	-	+	+	-	+	+	-	+
5.	ScreenMeter	-	-	-	-	+/-/-	-	-	+	-	-	+	-	+
6.	ImageExpert Pro 3	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
7.	ImageExpert™ Gauge	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
8.	AnalySIS Five	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
9.	BioVision	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
10.	MetaMorph 8	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
11.	ВидеоТест-Морфология 7.0	+	+	+	+	+/+/+	-	+	+	+	+	+	+	+
12.	ImageWarp	-	+	+	-	+/-/-	-	+	+	-	+	+	-	+
13.	QCapture PRO 6.0	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
14.	Motic Images Advanced 3.2	+	+	+	+	+/+/+	+	+	+	+	+	+	-	+
15.	MCID™ Core	-	+	+	+	+/+/-	-	+	+	-	+	+	-	+
16.	Image-Pro Plus 7.2	+	+	+	+	+/+/-	-	+	+	-	+	+	-	+
17.	ImageJ (NIH, USA)	-	+	+	+	+/-/-	-	+	+	-	+	+	-	+
18.	Pixels	-	-	+	-	+/-/-	-	-	+	-	+	+	-	+
19.	NIH Image	-	+	+	+	+/+/+	+	+	+	+	+	+	-	+
20.	Xite v3.45	-	-	+	-	+/-/-	-	-	+	-	+	+	-	+
21.	ДиаМорф	-	+	+	+	+/+/+	+	+	+	+	+	+	-	+

Мікроскоп прохідного світла складається зі штатива і прикріплених основних вузлів: система освітлення, предметний стіл, системи відтворення та візуалізації зображення. Основними характеристиками мікроскопу є його клас, кратність збільшення, роздільна оптична здатність, розмір лінійного

поля на предметі і ступінь виправлення аберацій. У СКМ для цитології і гістології застосовують прямі біологічні мікроскопи дослідницького, лабораторного та робочого класів. Розглянуто структуру та функції програмного забезпечення аналізу біомедичних зображень. Підсумовано переваги і недоліки систем комп'ютерної мікроскопії.

#### 1.4 Аналіз завдань роботи та постановка задач дослідження

Актуальним є розробка алгоритму текстурної сегментації на основі статистичних ознак, який дозволяє визначати набір текстурних ознак, що характеризують певний однорідний текстурний сегмент, наявний на зображенні. Надалі виявлений набір ознак використовується для ідентифікації та окреслення меж відповідного текстурного сегмента.

Мета і завдання дослідження. Метою кваліфікаційної роботи є розроблення алгоритмів текстурної сегментації зображень на основі статистичних ознак.

Об'єктом дослідження є процес сегментації цифрових зображень.

Предметом дослідження є методи і алгоритми текстурної сегментації зображень.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

1. систематизувати текстурні характеристики зображень;
2. провести аналіз методів сегментації зображень;
3. реалізувати алгоритми текстурної сегментації зображень;
4. розробити об'єктну модель та графічний інтерфейс системи сегментації зображень;
5. оцінити якість сегментації розробленими алгоритмами.

## 1.5 Висновки до розділу

Систематизовано текстурні характеристики біомедичних зображень. Біомедичні зображення мають складну структуру. Вони містять шари тканин утворені окремими клітинами котрі розміщені в певному порядку. Клітини оточені різнорідним фоном.

Проведено аналіз методів сегментації зображень. Оскільки біомедичні зображення володіють текстурними характеристиками то для їх аналізу застосуємо обчислення статистичних ознак.

## 2 АЛГОРИТМИ ТЕКСТУРНОЇ СЕГМЕНТАЦІЇ НА ОСНОВІ СТАТИСТИЧНИХ ОЗНАК

### 2.1 Операції попереднього опрацювання зображення

Яскравість точки й гистограми зображення. Людське око більш чутливе до зеленого кольору, ніж до червоного й синього разом узятих. Надлишковість по зелених пікселях дає зображення, яке видається менш шумним і більш чітким, ніж при рівній кількості кольорів. Яскравість кожного пікселя знаходиться по формулі:

$$Y:=0.3*R+0.59*G+0.11*B.$$

Гистограма - це представлена залежність числа повторень кожного значення яскравості на всьому зображенні від цього самого значення. Для зображення можна розглядати 4 виду гистограм: 3 по каналах(RGB) і одну по обчисленій яскравості.

Зміни яскравості й контрастності можна узагальнити виразом «зміна балансу зображення», тому що яскравість і контрастність схожі, і відносяться до сфери сприйняття зображення людиною, відмінність полягає в тому, що яскравість збільшує або зменшує кількість білого (знебарвлює) зображення, а контраст, навпаки, регулює кількість чорного (тіней).

Підвищення/зниження яскравості – це, відповідно, додавання/вирахування значення кожного каналу з деяким фіксованим значенням у межах від 0 до 255. Обов'язково необхідно контролювати вихід нового значення каналу за межі діапазону 0..255.

Підвищення/зниження контрастності – це, відповідно, множення/ділення значення кожного каналу на деяке фіксоване значення. Це веде до зміни співвідношень між квітами й, відповідно, до більш чітких колірних границь. На практиці існує такий принцип: зміна контрастності не

повинна приводити до зміни середньої яскравості по зображенню, тому користуються наступною формулою:

$$NewY:=K*(OldY-AveY)+AveY.$$

У цій формулі  $NewY$  – нове значення одного з каналів,  $K$  – коефіцієнт контрастності ( $K=(0..1)$  – зниження,  $K<1$  – підвищення контрастності),  $OldY$  – поточне значення того ж каналу,  $AveY$  – середнє значення того ж каналу по зображенню. Корекція нового значення при виході його за межі 0..255 обов'язкова.

Бінаризація – це трансформація зображення, у загальному випадку, до одноколірного (найчастіше до чорно-білого). При виконанні бінаризації вибирається якийсь поріг (зокрема, 155), усі значення нижче якого перетворюються в колір фону, а вище – в основний колір. Саме трансформація можна здійснювати по каналах, але в цьому випадку результуюче зображення не буде в буквальному значенні бінарним (чорно-білим), а буде містити 8 чистих кольорів, що представляють собою комбінації чистих червоного, зеленого й блакитного кольорів, тобто буде бінарним по каналах. Тому краще проводити перетворення над «повним» кольором точки.

Перетворення зображення до відтінків сірого полягає в одержанні яскравості кожної точки по відомій раніше формулі  $Y:=0.3*R+0.59*G+0.11*B$ , а також у наступному копіюванні отриманого значення в кожний з каналів RGB.

Одержання негатива відбувається простою заміною кожного значення каналу на його доповнення до 255. Зокрема,  $R=255-R$ .

Фільтрація дозволяє позбутися існуючого на зображенні шуму. Для фільтрів характерне поняття апертури фільтра – розміру тієї частини зображення, з якої безпосередньо працює фільтр у цей момент часу. Це вікно послідовне пересувається по зображенню, поки не буде оброблено все зображення. Ковзне вікно визначає особливий вид пересування апертури

фільтра, коли при досягненні бічної границі зображення вікно опускається на 1 піксель униз і починає рух у зворотному напрямку. При цьому на наступних кроках будуть використовуватися так само вже перетворені раніше пікселі. Нове значення яскравості або насиченості колірних каналів центрального у вікні пікселя встановлюється рівним середині між максимальним і мінімальними значеннями яскравості або насиченості його сусідів.

Реалізація обраних вище операцій.

Розрахунки таблиці яскравостей. Побудова гістограм. Щоб побудувати таблицю яскравостей необхідно розрахувати рівні яскравості для кожного пікселя й занести отримані значення в матрицю (DataGridView у нашому випадку). Для кожного пікселя застосовуємо формулу

$$Br_{ij} = 0,3R_{ij} + 0,59G_{ij} + 0,11B_{ij}$$

де  $i, j$  – координати пікселя, для якого знаходиться рівень яскравості,

$R_{ij}, G_{ij}, B_{ij}$  – рівні відповідних каналів для даного пікселя.

Для побудови гістограм необхідно підрахувати кількість пікселів для кожного рівня яскравості й для кожного рівня насиченості по каналах RGB. При побудові гістограм яскравостей можна скористатися розрахованою таблицею яскравості й підрахувати кількість пікселів для кожного рівня яскравості. Після чого виконаємо побудова гістограм. Для гістограм по каналах RGB необхідно проробити аналогічні операції, тільки над рівнями насиченості кожного з каналів.

Зміна балансу зображення. Зміна яскравості проводиться по наступному алгоритму: необхідно зчитати рівні кожного з каналів пікселя, після чого до кожного із цих каналів додати деяке фіксоване значення, яке буде показувати тип перетворення яскравості: при збільшенні яскравості – значення буде позитивним, а при зменшенні – негативним.

Після підсумовування поточного рівня кожного каналу з фіксованим значенням необхідно помістити отримані значення у відповідні канали, попередньо відкоригувавши значення, якщо воно вийшло за межі 0..255.

Для зміни контрасту зображення необхідно для початку розрахувати

середнє значення по кожному з каналів RGB для всього зображення.

$$\bar{R} = \frac{1}{N} \sum_{i=0}^n \sum_{j=0}^m R_{ij};$$
$$R \frac{1}{N} \sum_{i=0}^n \sum_{j=0}^m G_{ij};$$
$$B = \frac{1}{N} \sum_{i=0}^n \sum_{j=0}^m B_{ij} ,$$

де  $N$  – кількість пікселів,

$n$  – ширина,

$m$  – висота зображення,

$R_{ij}, G_{ij}, B_{ij}$  – рівні відповідних каналів для пікселя по координатах  $ij$ .

Після цього необхідно визначити коефіцієнт перетворення  $K$ , а потім скористатися для значень кожного каналу для кожного з пікселів формулами:

$$R_{ij\text{new}} = R(R_{ij} - \bar{R}) + \bar{R}$$

$$R = K * + \bar{G}$$

$$B_{ij\text{new}} = G(B_{ij} - \bar{B}) + \bar{B} .$$

На наступному кроці необхідно помістити отримані значення у відповідні канали, попередньо відкоригувавши значення, якщо воно вийшло за межі 0..255. Наведені вище формули потрібно застосувати для кожного пікселя в зображенні.

При виконанні бінаризації необхідно для початку знайти поріг бінаризації. У цих цілях спочатку будуємо гістограми зображення, після чого по отриманій гістограмі вибираємо деяке значення яскравості, яке умовно буде відокремлювати фон зображення від об'єктів. Поріг необхідно вибирати так, щоб втратити якнайменше деталей бажаного об'єкта й при цьому



позбутися як можна більшої кількості фонових об'єктів, які не несуть у собі корисної інформації.

Після перетворення зображення буде складатися тільки з пікселів двох кольорів. Перетворення зображення в півтонове полягає в тому, щоб кожному пікселю по всіх трьом каналам RGB привласнити значення яскравості цього пікселя. Тому спочатку необхідно розрахувати яскравість пікселя по формулі:

$$B = 0,3R_{ij} + 0,59 G_{ij} + 0,11B_{ij}.$$

Потім отримане значення присвоюється всім трьом каналам:

$$R_{ij} = R, G_{ij} = G, B_{ij} = B_{ij}.$$

У наведених формулах  $i, j$  – координати пікселя,  $R_{ij}$ ,  $G_{ij}$ ,  $B_{ij}$  – рівні відповідних каналів для пікселя по координатах  $ij$ .

Дані перетворення необхідно виконати для кожного пікселя для одержання півтонового зображення.

Метод рангової фільтрації з оцінкою середини діапазону. Для виконання фільтрації необхідно для початку вибрати розмір апертури фільтра. Звичайно вона береться розмірами 3x3, але іноді є зміст вибрати апертуру й більшого розміру. У загальному випадку вікно фільтра буде виглядати в такий спосіб:

$$\begin{array}{ccc} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{array}.$$

Маска для рангової фільтрації має такий вигляд (можна її й не враховувати):

$$H \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Для кожного кроку фільтр буде знаходити мінімум і максимум, і знаходити середнє між ними, по всіх трьом каналам.

Після чого вікно буде зміщатися на наступну позицію. При наступних обчисленнях фільтр буде працювати з вікном, що полягають не тільки з елементів вихідного зображення, але й з елементів, що раніше подвергнувшись перетворенню.

## 2.2 Матриці суміжності

Статистичні методи описують просторовий розподіл рівнів сірого зображення. Матриці суміжності (МС) відображають статистику другого порядку. У МС  $P_d$  для вектора зміщення  $\mathbf{d} = (dx, dy)$  значення елемента  $p_{i,j}$  є кількістю входжень пари значень рівнів сірого  $i$  та  $j$ , які розміщені на відстані  $\mathbf{d}$  [3]. Кожній точці зображення  $(x, y)$  відповідає матриця суміжності  $P_d$ , яка характеризує розподіл яскравості в квадраті розміром  $W \times W$  з центром в точці  $(x, y)$ . Елементи матриці  $P_d$  визначаються таким чином:

$$P_d(i, j) = \sum_{(m,n) \in D} f_{i,j}(x_{m,n}; x_{m+d,n+d}), \quad (2.1)$$

де

$$f_{i,j}(x_{m,n}; x_{m+d,n+d}) = \begin{cases} 1, & (x_{m,n} = i \text{ та } x_{m+d,n+d} = j) \text{ або } (x_{m,n} = j \text{ та } x_{m+d,n+d} = i) \\ 0, & \text{інакше} \end{cases}, \quad (2.2)$$

де  $D$  – квадрат розміром  $W \times W$  ( $W$  - непарне),

$i, j = 0..255$  – значення яскравості точок,

$x_{m,n}$  – значення яскравості точки з координатами  $(m,n)$  .

Функція  $f_{i,j}(x_{m,n}; x_{m+d,n+d})$  є означає що точки котрі знаходяться на заданій відстані, володіють певними рівнями яскравості. Параметр  $d$  визначає відстань, на якій проводиться аналіз сусідніх точок. На основі матриць суміжності визначаються такі текстурні ознаки як енергія, ентропія, контраст, однорідність, кореляція, затінення. МС використовуються для вирішення задач класифікації текстури.

Функція  $f_{i,j}(x_{m,n}; x_{m+d,n+d})$  визначається так:

$$f_{i,j}(x_{m,n}; x_{m+d,n+d}) = \begin{cases} 1, & (x_{m,n} = i \text{ та } x_{m+d,n+d} = j) \\ & \text{або } (x_{m,n} = j \text{ та } x_{m+d,n+d} = i) \\ 0, & \text{інакше} \end{cases}$$

Функція є індикатором, що точки, котрі знаходяться на заданій відстані, мають конкретні рівні яскравості. Параметр  $d$  визначає відстань, на якій проводиться дослідження сусідніх точок.

По МС  $P_d$  (2.2), що описує розподіл яскравості всередині сегменту з центром в точці з координатами  $(x, y)$ , обчислюються признаки текстур [3], перераховані нижче ( $F1, \dots, F9$ ). Після опрацювання цілого зображення для кожного признаку формується матриця, що зберігає значення признаку у всіх оброблюваних точках – простір текстурних ознак. Нижче наведені вирази для обчислення ознак [3]:

1) сумарне середнє значення:

$$F_1 = \sum_{i=1}^W m_i p_i,$$

2) інерція:

$$F_2 = \sum_{i=1}^W \sum_{j=1}^W (i - j)^2 P(i, j),$$

3) другий кутовий момент:

$$F_3 = \sum_{i=1}^W \sum_{j=1}^W P^2(i, j),$$

4) сумарна кореляція:

$$F_4 = \sum_{i=1}^W \sum_{j=1}^W m_i m_j,$$

5) ентропія:

$$F_5 = - \sum_{i=1}^W \sum_{j=1}^W \ln(P(i, j)) P(i, j),$$

6) кореляція:

$$F_6 = \sum_{i=1}^W \sum_{j=1}^W (i - M)(j - M) P(i, j),$$

7) затінення:

$$F_7 = \sum_{i=1}^W \sum_{j=1}^W (i + j - 2M)^3 P(i, j),$$

8) контраст:

$$F_8 = \sum_{i=1}^W \sum_{j=1}^W |i - j| P(i, j),$$

9) сумарна ентропія:

$$F_9 = \sum_{i=1}^W \ln(p_i) p_i,$$

де  $m_i = \sum_{j=1}^W j P(i, j)$ ,  $p_i = \sum_{j=1}^W P(i, j)$ ,  $M = \sum_{i=1}^W i p_i$ .

Отже, обчислення для кожної точки зображення матриць суміжності і текстурних ознак, по наведеному вище визначенню, займатиме тривалий час. Варто обчислювати матрицю і ознаки рекурсивно. Крім того варто не нормувати матриц. Оскільки нормування (ділення на загальне число оброблених пар точок всередині границь вікна) необхідне при обчисленні

лише певних ознак ( $F_5, F_6, F_7, F_9$ ). Щоб імплементувати рекурсивне обчислення ознак, спочатку необхідно обчислити матрицю  $P_y^x$  і текстурні признаки для першої точки ( $x = L, y = L$ ), із  $L = \frac{W-1}{2}$ . Ковзаюче вікно по якому обчислюється матриця для наступної точки ( $x+1, y$ ), зсувається вправо, що рівнозначно наступному перетворенню:

$$P_y^{x+1}(i, j) = P_y^x(i, j) - \sum_{(m,n) \in D1} f_{i,y}(x_{m,n}; x_{m+d,n+d}) + \sum_{(m,n) \in D2} f_{i,y}(x_{m,n}; x_{m+d,n+d}),$$

де  $D1 = \{(m, n) \mid m \in [x - W, x + W], n = y - W\}$ ,

$D2 = \{(m, n) \mid m \in [x - W, x + W], n = y + W\}$ .

Перехід на подальшу стрічку зображення проводиться аналогічно: при цьому вікно зсувається вниз для першої точки попереднього рядка і матриця обчислюється знову для нового положення вікна. Нове обчислення признаков здійснюється наступним чином: під час кожного кроку, коли до елементу матриці  $P(i, j)$  додається (віднімається) одиниця, признаки текстури перетворюються відповідно до наведених нижче формул:

$$m_i = m_j \pm j$$

$$p_i = p_i \pm 1$$

$$F_2 = F_2 \pm (i - j)^2$$

$$F_5 = F_5 + \ln\left(\frac{P(i, j)}{N}\right)P(i, j) + \ln\left(\frac{P(i, j) \pm 1}{N}\right)(P(i, j) \pm 1)$$

$$F_8 = F_8 \pm |i - j|, \quad Mx = Mx \pm \frac{i}{N}, \quad N = 2S^2.$$

Якщо  $i = j$ , то  $F_3 = F_3 \pm 4P(i, j) + 4$ , у протилежному випадку  $F_3 = F_3 \pm 2P(i, j) + 1$ . Далі в кожній матриці значень текстурних признаков

знаходяться максимум і мінімум, і здійснюється нормування до стандартного діапазону градацій сірого кольору 0..255 по виразу:

$$g = 255 \frac{f - \min}{\max - \min},$$

де  $f$  - елемент матриці;

$g$  - відповідне значення яскравості;

$\max$  і  $\min$  - максимальний і мінімальні елементи матриці відповідно.

Після цього матриці текстурних признаков відображаються візуально - створюється зображення простору текстурних признаков. Розмір вікна, всередині якого обчислюються текстурні признаки, впливає на чіткість зображення об'єктів інтресу: при збільшенні  $S$  границі стають більш розмитими, тому цю величину необхідно вибирати з урахуванням розміру оброблюваного зображення і об'єктів, що виділяються. Параметр  $d$  вказує на відстань між точками, значення яскравості котрих враховуються при складанні матриці розподілу рівнів сірого. Схема алгоритму обчислення матриць розподілу рівнів сірого на мові MATLAB наведена на рисунку 2.1.

Алгоритм текстурної сегментації на основі матриць суміжності

Запропонований алгоритм для сегментації текстурного поля складається з наступних кроків:

1. Побудова текстурного поля  $G$ , кожна точка  $g \in G$  відповідно (2.1)

$$g(i,j) = m_{1,1} = \sum_{-W/2}^{W/2} \sum_{-W/2}^{W/2} f(m,n) x_m y_n$$

2. Нормалізація  $G$ ,  $g \in [0, 255]$ .

3. Пошук порогів  $t_1, t_2, \dots, t_n$  за допомогою наступних кроків:

а) Встановлення інтервалу  $R=[a,b]$ ;  $a=0$  і  $b=255$ .

б) Знаходження математичного сподівання  $\mu$  і середнього квадратичного відхилення  $\sigma$  всіх пікселів на інтервалі  $R$ .

в) Обчислення порогів  $t_1$  і  $t_2$  як  $t_1 = \mu - k\sigma$  і  $t_2 = \mu + k\sigma$ . Параметр  $k$  впливає на величину інтервалу  $R$ ;

г) Обчислення нових інтервалів  $a = t_1 + 1$ ,  $b = t_2 - 1$ ;

д) Повторювати кроки а-г,  $n/2$  разів ( $n$  - число порогів), встановлюючи нові межі інтервалів  $a = t_1 + 1$ ,  $b = \mu$  і  $a = \mu + 1$ ,  $b = t_2 - 1$ .

4. Сегментація текстурного поля  $G$ , в результаті чого отримуємо  $n+1$  бінарних масок  $s_i$ ,  $i=\{1, \dots, n+1\}$ :

$$s_i(x, y) = \begin{cases} 1 & \text{якщо } t_{i-1} \leq g(x, y) \leq t_i \\ 0 & \text{інакше} \end{cases},$$

де  $t_0 = 0$ ,  $t_{n+1} = 255$ .

5. Сегментація вхідного зображення з метою отримати  $n+1$  зображень  $\{s_i\}$ ,  $i=\{1, \dots, n+1\}$

$$\{s_i\}(x, y) = \begin{cases} f(x, y) & \text{якщо } s_i(x, y) = 1 \\ 0 & \text{інакше} \end{cases}.$$

Для тестування МС (5) як текстурної ознаки потрібно виконати даний алгоритм, побудувавши на 1 кроці  $g(i, j) = F_5$ . Параметр  $k$  служить для керування інтервалом між найменшим і найбільшим порогом.

Оптимальне число порогів  $n$  (відповідно число ітерацій алгоритму) можна задавати апріорно, виходячи із застосування або обирати на основі зміни значення відношення сигнал/шум  $\rho$ . Значення  $\rho$  можна обчислювати між вихідним зображенням та сегментованим з усередненими значеннями пікселів в середині сегментів.

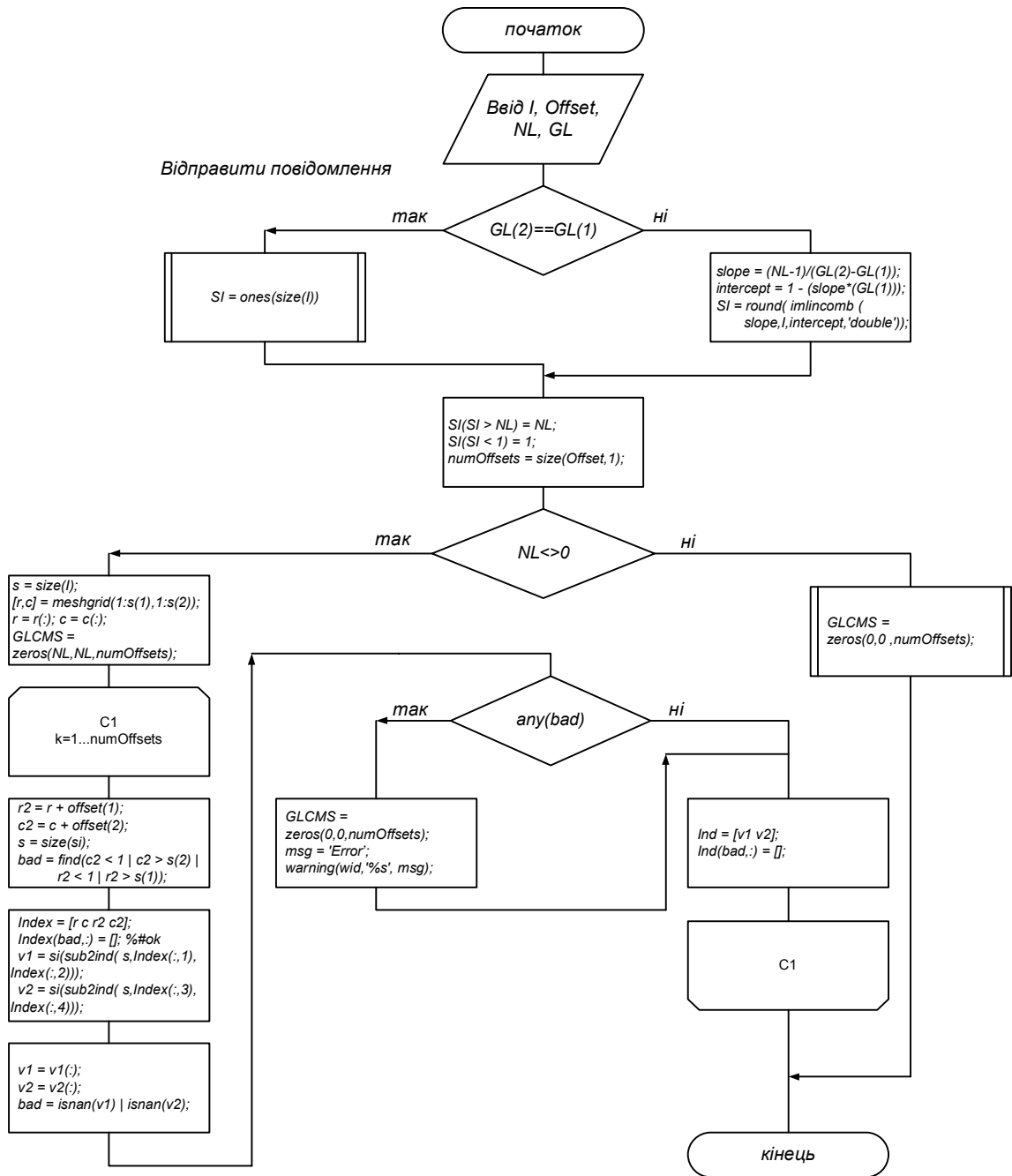


Рисунок 2.1 – Алгоритм обчислення матриць розподілу рівнів сірого

### 2.3 Алгоритми текстурної сегментації на основі моментів

Реалізовано алгоритм текстурної сегментації на основі моментів включає наступні кроки [19]:



а) обчислення текстурних признаков для кожної точки зображення в границях ковзаючого вікна  $W \times W$  розміром;

б) сегментація створеного текстурного простору.

Розглянемо текстурні признаки на основі просторових моментів області та матриць суміжності рівнів сірого.

Текстуру на зображенні кількісно описують за допомогою відносно простих статистичних характеристик, зокрема, математичного сподівання, дисперсії і моментів вищого порядку [17]. Назва «просторові моменти» (ПМ) походить з механіки. Застосовуючи до зображень обчислення ПМ, отримують відображення розподілу рівнів сірого зображення вздовж осей зображення. На основі ПМ можна обчислити признаки області інваріантні до переносу, повороту або масштабу.

В точці з координатами  $(x,y)$  і значенням  $f(x,y)$  функції рівнів сірого ПМ для області обчислюються так :

$$m_{p,q} = \iint x^p y^q f(x,y) dx dy, \quad (2.3)$$

де  $p+q = 0,1,2,3$ .

Розглянемо зображення як функцію двох змінних  $f(x,y)$  і обчислимо ряд моментів нижчого порядку для кожного пікселя на зображенні ( $p+q \leq 2$ ). Моменти обчислюються навколо кожного пікселя в границях вікон із  $W \times W$  розміром.

В дискретному варіанті при нормалізованих координатах  $(x_m, y_n)$  ПМ в границях вікна, із центром в пікселі  $(i,j)$ , обчислюються так:

$$m_{p,q} = \sum_{-W/2}^{W/2} \sum_{-W/2}^{W/2} f(m,n) x_m^p y_n^q \quad (2.4)$$

де  $m,n$  - координати точки відносно ковзаючого вікна.

Ми використовуємо рядок-стовпчик момент інерції  $m_{1,1}$ .

Ознаки текстури на основі локальних бінарних шаблонів.

Використаємо локальні бінарні шаблони (ЛБШ, ЛБШ) для побудови текстурних ознак для зображень в колекції. ЛБШ, уперше описаний в [14], являє собою простий і потужний інструмент для розпізнавання різних елементів зображення. У якості простору ознак використовуються гістограми так званих ЛБШ кодів. Завдяки високій швидкості обчислення деяких типів кодів ЛБШ став широко застосовуватися для розпізнавання образів.

ЛБШ являє собою фільтр, що позначається як ЛБШ  $R,P(x, y)$ , який для кожної точки зображення розраховує код на основі значень точок у деякій околиці цієї точки. У цьому випадку  $P$  – число точок,  $R$  – радіус околиці

Точки околиці позначимо як  $g_i$ , де  $i = \overline{0, P-1}$ . При цьому координати точки розраховуються як

$$(R \cdot \cos(\frac{i}{2\pi}); R \cdot \sin(\frac{i}{2\pi})).$$

Позначимо зображення  $f(x,y)$  або  $f(g)$ , якщо  $g$  це точка. Нехай

$$s_i(x, y) = \begin{cases} 1, f(g_i) > f(g_c) \\ 0, \text{інакше} \end{cases}$$

де  $g_c$  - точка з координатами  $(x, y)$ . Тоді

$$LBP_{P,R}(x, y) = \sum_{i=0}^{P-1} 2^i \cdot s_i(x, y)$$

Таким чином, послідовність  $s_i(x, y)$ , де  $i = \overline{0, P-1}$ , являє собою двійкову послідовність коду ЛБШ. Отже,  $LBP_{P,R}(x,y) \in [0, 2^P - 1]$ .

Для порівняння двох зображень як векторів ознак використовуються гістограми кодів ЛБШ. У загальному випадку для кожного зображення будується гістограма  $H(l)$  для значень  $LBP_{P,R}(x,y)$ , де  $l = \overline{0, P-1}$ . Існує кілька методів розрахунків відстані між гістограмами. Зокрема, відстань Хі-квадрат:

$$\chi^2(H_1, H_2) = \sum_{i=0}^{B-1} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)},$$

де  $B$  - число кодів.

Існують також модифікації фільтра, описані в роботах [10, 13-15]:

Деякі коди несуть у собі більше інформації, ніж інші. Коди, у двійковому циклічному записі яких число переходів між послідовностями «1» і «0» не перевищує двох, позначаються як «uniform», що відповідає слову «рівномірний» [3]. Для заданого  $P$  існує  $P \cdot (P-1) + 2$  рівномірних значень. Модифікований фільтр  $LBP_{P,R}^{u2}$  у цьому випадку повертає коди рівномірних значень, додаючи тільки один код для нерівномірних значень.

$$LBP_{P,R}^{u2}(x, y) = \begin{cases} \text{індекс коду, якщо рівномірний} \\ P \cdot (P-1) + 2, \text{ інакше} \end{cases},$$

4. Т.к. околиця являє собою коло, то можна знайти групи кодів, інваріантних до повороту. Для кожного коду ЛБШ існує  $P$  кодів, інваріантних до повороту, одержуваних шляхом циклічного зсуву  $P$  -бітового числа. Для кожної такої групи у фільтр попадає мінімальне значення кодів даної групи. Задача визначення кількості кодів, інваріантних до повороту є нетривіальною. Фільтр позначається як  $LBP_{P,R}^{ri}$ .

5. З врахуванням попередніх двох властивостей визначається також рівномірний фільтр, інваріантний до повороту. ЛБШ кодів, що володіють одночасно двома властивостями всього  $P + 2$ , які відрізняються друг від друга

числом біт, рівних «1». У цьому випадку фільтр  $LBP_{P,R}^{riu2}$  задається в такий спосіб:

$$LBP_{P,R}^{riu2}(x, y) = \begin{cases} \text{число одиниць, якщо рівномірний} \\ P + 1, \text{ інакше} \end{cases}.$$

У роботі [14] був запропонований метод порівняння гістограм мікроб'єктів. Зображення особи розбивається на  $k \times k$  ділянок, для кожного з яких обчислюється гістограма. Підсумкова гістограма зображення мікроб'єктів визначається як конкатенація гістограм ділянок зображення.

У роботі [26] був запропонований розширений метод порівняння гістограм мікроб'єктів, заснований на зваженій матриці. Задається матриця ваг  $k \times k$ , кожний елемент якої відповідає ділянці зображення. Позначимо гістограму  $j$ -ї ділянки зображення як  $H^j$ ,  $j = \overline{0, k^2 - 1}$ . Для кожної  $j$ -ї ділянки задається вага  $w_j$ . Тоді можна визначити модифіковану (зважену) відстань Хі-квадрат у такий спосіб:

$$\chi_w^2(H_1, H_2) = \sum_{j=0}^{k^2-1} w_j \cdot \sum_{i=0}^{B-1} \frac{(H_1^j(i) - H_2^j(i))^2}{H_1^j(i) + H_2^j(i)},$$

де  $B$  - число кодів ЛБШ.

Процес офлайнового обчислення дескриптора текстур, який у загальному випадку містить у собі наступні кроки:

1. Задається набір текстур, кожна з яких належить одному з декількох заданих класів.
2. Кожному класу присвоюється набір міток.
3. Для кожного зображення виконується пошук текстур із заданого набору.

4. Для кожної знайденої текстури до зображення додається набір міток відповідного класу.

Для виконання текстурного пошуку потрібен класифікатор. У даній роботі пропонується класифікатор на основі методу моментів.

У роботі [17] був запропонований метод сегментування текстур на основі моментів і описаний відповідний математичний апарат. Нехай  $f(x, y)$  являє собою зображення довжиною  $W$  і шириною  $H$ . Область значень  $f(x, y)$  -  $[0;1]$ , що відповідає області  $[0; 255]$  звичайного монохромного зображення. Метод моментів полягає в тому, що для кожної точки зображення  $f(x, y)$  розраховується набір моментів і похідних від них характеристик у межах деякого вікна із центром у даній точці. Таким чином, формується набір зображень, відповідних до набору ознак.

Момент  $M_{p,q}$  з розміром вікна  $W_M$  розраховується в такий спосіб:

$$M_{p,q}(i, j) = \sum_{(a,b) \in W_{i,j}^M} f(a,b) \cdot x_a^p \cdot y_b^q$$

де

$$x_a = \frac{a-i}{W_M/2}; \quad y_b = \frac{b-i}{W_M/2}$$

де  $W_{i,j}^M$  - вікно розміром  $W_M$  із центром у точці  $(i,j)$ .

У тій же роботі [17] стверджувалося, що набір моментів в «чистому» виді не годиться для сегментування, тому автором був запропонований покращений набір ознак:

$$F_{p,q}(i, j) = \frac{1}{W_F^2} \sum_{(a,b) \in W_{i,j}^F} \left| \tanh(\sigma(M_{p,q}(a,b) - \overline{M_{p,q}})) \right|,$$

$$\overline{M_{p,q}}(i, j) = \frac{1}{W_F^2} \sum_{(a,b) \in W_{i,j}^F} M_{p,q}(a,b),$$

де  $W_{i,j}^F$  - вікно розміром  $W_F$  із центром у точці  $(i,j)$ .

Ступені моментів  $p$  і  $q$  задаються таким чином, щоб їх сума не перевищувала деякого значення  $O$ , яке являє собою порядок моментів. Число ознак, таким чином, залежить від  $O$ . У даній роботі розглядалися моменти порядку не вище 2-го.

Іншими параметрами для розрахунків характеристик є розміри вікон  $W_M$  і  $W_F$  і коефіцієнт  $C$ . Авторами даного методу запропоновані значення даних параметрів: 9, 49 і 0.01 відповідно. На жаль, дані значення були обґрунтовані методом «проб і помилок» без якого-небудь аналітичного або практичного обґрунтування. Тому в даній роботі був проведений аналіз впливу значень даних параметрів на результат сегментації.

Для експериментів використовувався алгоритм сегментації, похідний від алгоритму, запропонованого в [17].

1. Тестове зображення «склеюється» з декількох текстур (приклад на рисунку 4). Кожна текстура представлена квадратною ділянкою. Площі ділянок текстур однакові.

2. Розраховуються характеристики, що представляють собою  $N$  зображень, де  $N$  - число характеристик.

3. Випадково відбираються 6 % точок, рівномірно розподілених по зображенню.

4. Для обраних точок проводиться кластеризація методом k-середніх. Число кластерів відповідає числу текстур. Автори методу запропонували алгоритм кластеризації CLUSTER [12].

5. Отримані центри кластерів як результат роботи алгоритму k-середніх використовуються для сегментації всього зображення. Кожному пікселю ставиться у відповідність номер того кластера, до якого ближче всього розташовується відповідний вектор ознак.

6. Визначимо точність сегментації  $p$  як частка правильно сегментованих пікселів. Оскільки алгоритм k-середніх доволіно визначає початкові центри

кластерів, до алгоритму додається проміжний крок, на якому визначається відповідність номеру текстури й номеру кластера. Будемо вважати, що кластер з номером  $i$  буде відповідати текстурі  $j$ , якщо більша частина пікселів класу  $i$  буде розташовуватися усередині ділянки текстури  $j$ .

#### 2.4 Висновки до розділу

Розроблено алгоритми текстурної сегментації зображень на основі матриць суміжності пар пікселів та просторових моментів, що дозволило сегментувати на зображення складні об'єкти – поперечні переріз пластів тканин, частини кровоносних судин та протоків залоз.

## 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНИХ АЛГОРИТМІВ

### 3.1 Структура програмного забезпечення

Розроблено структуру системи аналізу біомедичних зображень, що складається з таких компонентів: блоку інтерфейсу користувача, блоку попереднього опрацювання зображення, блоку сегментації, блоку виділення мікрооб'єктів, блоку обчислення кута орієнтації, блоку обчислення числових ознак та блоку зберігання опису зображення у БД (рисунок 3.1).

Блок попереднього опрацювання зображення призначений для максимального наближення досліджуваного зображення до еталонного або нормованого. Найчастіше для медичних зображень це просторово-інваріантні операції, зсув, зміна яскравості, зміна контрасту, квантування і геометричні перетворення (зміна масштабу, поворот осі). Теорія цих перетворень добре розроблена і як правило, не викликає труднощів при використуванні сучасних комп'ютерів.

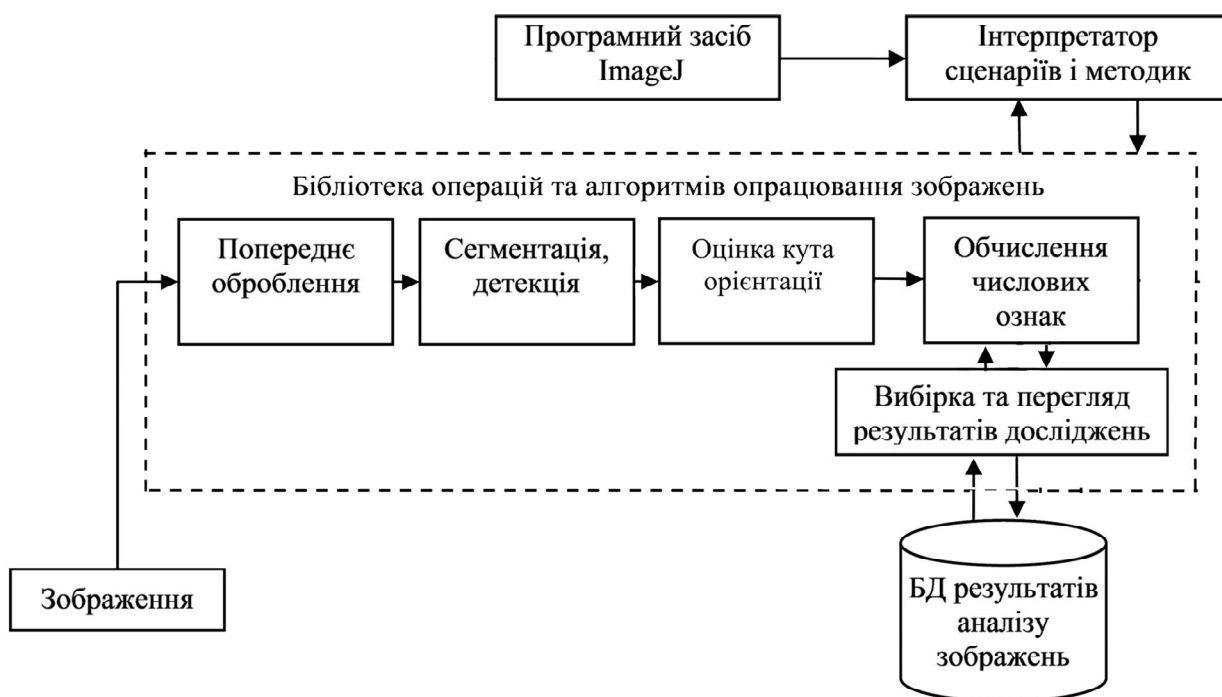


Рисунок 3.1 - Структура системи аналізу біомедичних зображень



Блок визначення текстурних ознак та блок текстурної сегментації призначені для сегментації вхідного зображення. Процес текстурної сегментації можна розділити на 3 етапи:

1) перетворення вихідного зображення для отримання набору особливостей (характеристичних ознак), що утворюють багатомірний простір ознак зображення, кожен вектор простору характеризує особливості певної текстури;

2) класифікація, кожна точка зображення співставляється з певним класом на основі оцінки вектора признаков відповідного цієї точці в просторі ознак;

3) етап сегментації вихідного зображення ґрунтується на інформації отриманої після класифікації.

Ознака (feature) обчислюється як функція від значень, що містяться в одному або більше пікселях, і обчислюється щоб чисельно виражати деяку значиму характеристику об'єкту. Узагальнені признаки (текстура, форма, колір) поділяють на ознаки піксельного рівня (координата, колір), локальні признаки (обчислюються у околі області або по вікну), глобальні ознаки (обчислюються для цілого зображення, зокрема, дисперсія, гістограма).

Блок виділення мікрооб'єктів призначений для виділення контурів мікрооб'єктів та обчислення їх ознак.

Блок формування таблиць числових характеристик призначений для збереження значень ознак виділених мікрооб'єктів.

Блок зберігання опису зображення у БД призначений для зберігання біомедичних зображень та опису їх вмісту.

Розроблювана система реалізована у вигляді плагіну до ImageJ. Оскільки програмі не потрібне зображення в якості вхідного аргументу то плагін реалізує інтерфейс PlugIn. Після того як плагін викликається із меню плагінів “Plugins/Diagnosis/AMS Diagnosis” (рисунок 3.1) запускається метод run.

При першому зверненні до класу AMS\_Diagnosis виконується його статична ініціалізація, де виконується необхідні налаштування та ініціалізація.

Після запуску у методі run показується головне вікно програми, яке наведено на рисунку 3.2.

Перед початком роботи із програмою необхідно запуснути базу даних і вказати налаштування для підключення (рисунок 3.3).

Нижче наведений детальний опис про URL підключення до БД:

- mysql – драйвер бази даних MySQL;
- localhost – адрес сервера бази даних;
- 3306 – порт сервера бази даних;
- Diagnosis – назва бази даних.

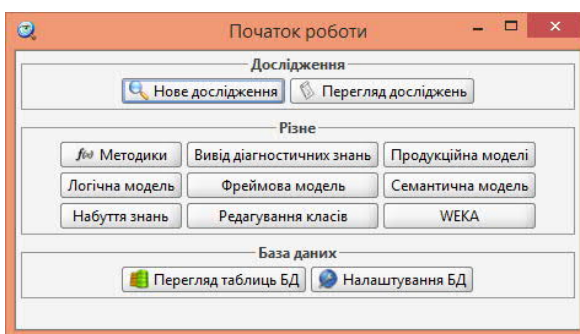


Рисунок 3.2 - Головне вікно плагіну AMS Diagnosis

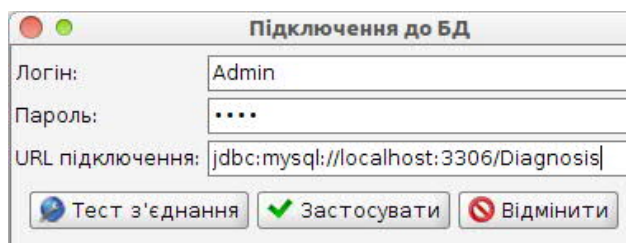


Рисунок 3.3 - Налаштування підключення до бази даних

Після введення параметрів підключення до БД можна протестувати або вдасться встановити з'єднання (рисунок 3.4).

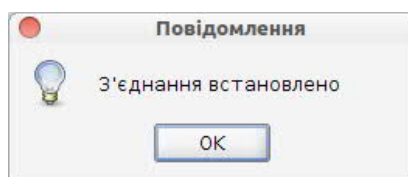
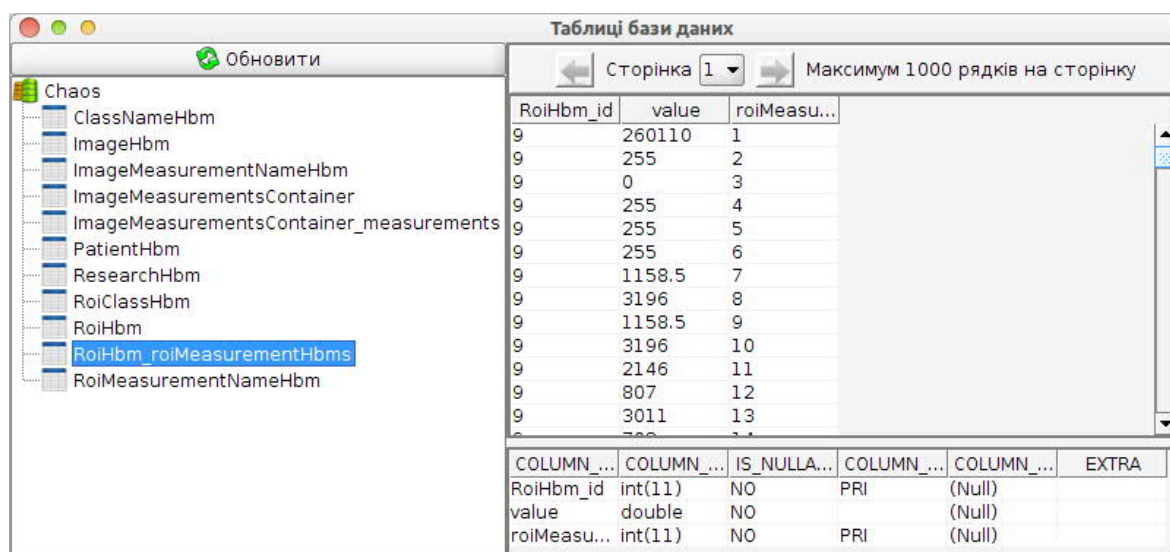


Рисунок 3.4 - Повідомлення про успішне встановлення з'єднання із БД

Також існує можливість переглянути структуру БД, а також даних які вона зберігає (рисунок 3.5).



RoiHbm_id	value	roiMeasu...
9	260110	1
9	255	2
9	0	3
9	255	4
9	255	5
9	255	6
9	1158.5	7
9	3196	8
9	1158.5	9
9	3196	10
9	2146	11
9	807	12
9	3011	13

COLUMN_...	COLUMN_...	IS_NULLA...	COLUMN_...	COLUMN_...	EXTRA
RoiHbm_id	int(11)	NO	PRI	(Null)	
value	double	NO		(Null)	
roiMeasu...	int(11)	NO	PRI	(Null)	

Рисунок 3.5 - Перегляд структури та даних таблиць БД

## 3.2 Автоматизація повторюваних операцій і побудова методик

### 3.2.1 Опис функцій графічного інтерфейсу з точки зору користувача

Вікно для побудови методик (рисунок 3.6) являє собою візуальний засіб і використовується для створення методик та логіки їх виконання. З наведеного рисунку можна побачити наступні елементи меню:

1. запуск методики або функції на виконання для тестування роботи (доступно лише для методик та функцій без вхідних параметрів);
2. виклик діалогового вікна для управління методиками;
3. назва змінної у яку присвоюється результат функції;
4. функція, що нічого не повертає;
5. результат функції ігнорується;
6. панель, що містить послідовність функцій для виконання;
7. виклик діалогового вікна для створення функції з уже існуючих функцій;
8. виклик діалогового вікна для додання ImageJ плагіну у список

функцій;

### 9. дерево доступних функцій.

Для побудови методики потрібно перемістити необхідні функції з правої частини вікна у ліву. Для цього необхідно натиснути ліву кнопку мишки на потрібній функції та перемістити її у ліву панель.

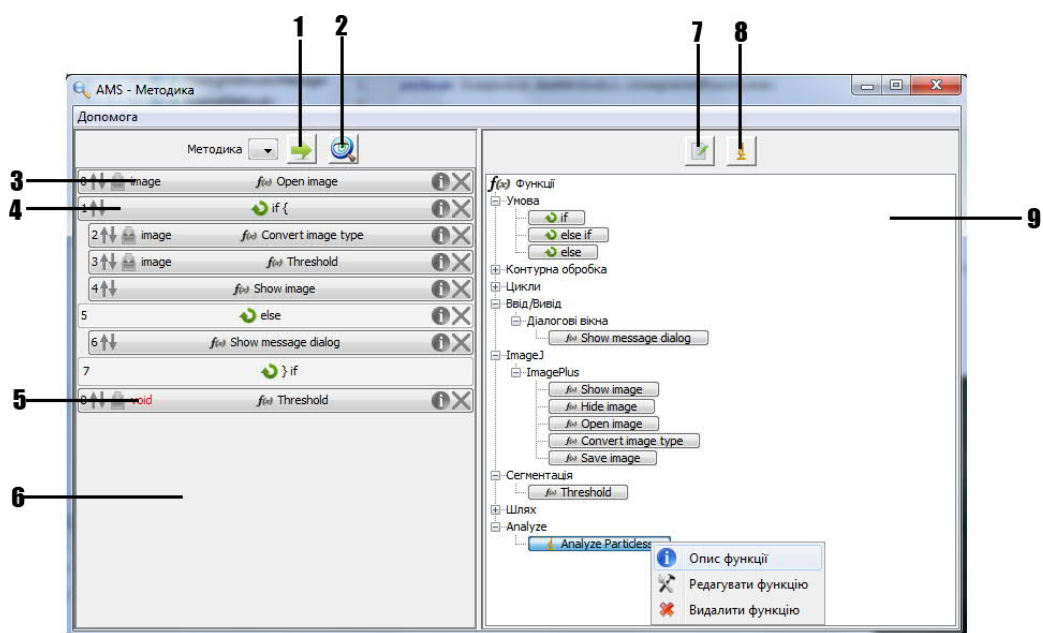


Рисунок 3.6 – Вікно для побудови методик та функцій

Задання вхідних параметрів функції здійснюється за допомогою діалогового вікна (рисунок 3.7), для його виклику потрібно натиснути лівою кнопкою мишки по функції. Значення вхідного параметра “Не вибрано” не дозволяє запуснути методику або функцію на виконання. Константи – це наперед задані можливі варіанти для вхідного параметра. Змінні містять значення, що отримуються як результат виконання функції. Якщо тип параметра з наступних: String, boolean, char, byte, short, int, long, float, double або їх клас “обгортка”, то їх значення можна ввести вручну.

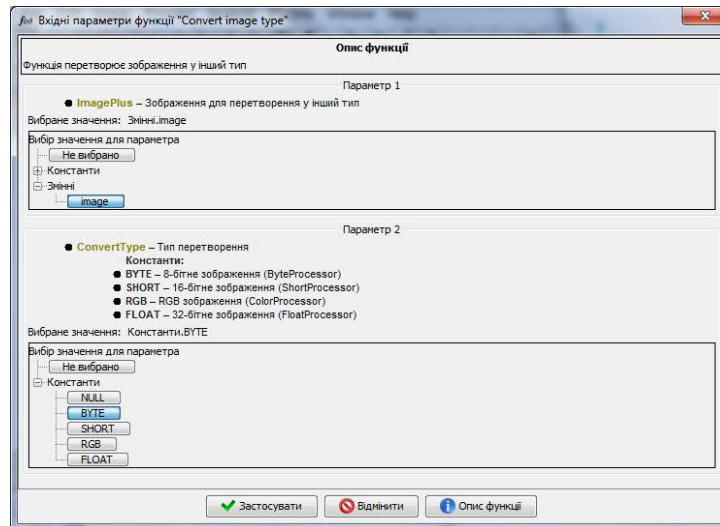


Рисунок 3.7 - Діалог задання вхідних параметрів для функції

Щоб задати назву змінної для присвоєння, потрібно натиснути лівою кнопкою мишки на відповідну іконку (пункт 3 на рисунку 3.6). Після чого відкриється діалогове вікно для задання назви змінної (рисунок 3.8).

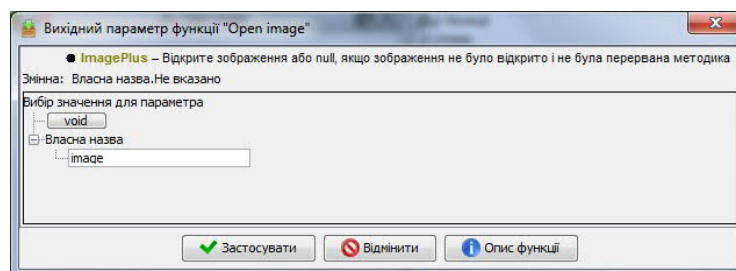


Рисунок 3.8 - Діалог задання імені змінної для вихідного параметра функції

Для повного опису функції потрібно натиснути на відповідну кнопку після чого відкриється вікно з повним описом функції (рисунок 3.9). Щоб додати ImageJ плагін існує спеціальне діалогове вікно (рисунок 3.10) для генерування функції “обгортки” (пункт 8 на рисунку 3.6). Для цього необхідно вказати назву плагіну, яка використовується для його виклику, цю назву можна дізнатися за допомогою вбудованого в ImageJ записувача макросів (рисунок 3.11).

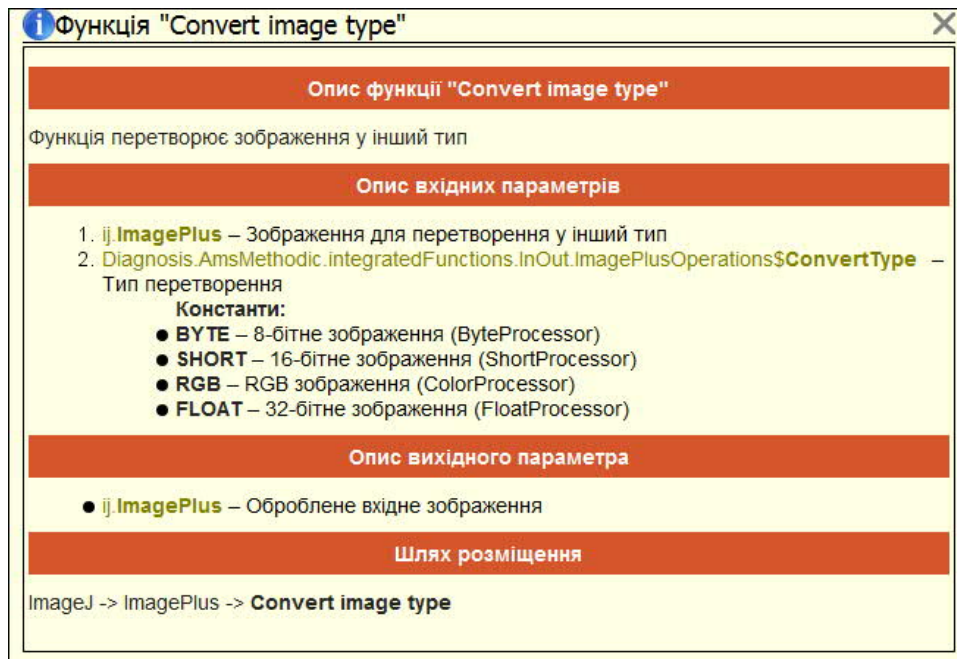


Рисунок 3.9 - Вікно з повним описом функції

Для цього потрібно запустити його та викликати ImageJ плагін. Якщо ImageJ плагін містить діалогове вікно для задання вхідних параметрів, то необхідно поставити галку “Містить параметри”. Якщо ImageJ плагін потребує вхідного зображення, то це теж потрібно відмітити. Також, якщо плагін повертає зображення, то це теж потрібно відмітити. Якщо ImageJ плагін містить діалогове вікно з параметрами, то необхідно зробити так, щоб воно відкрилося при натисненні на кнопку “Згенерувати функцію”. Зокрема, для відкриття діалогового вікна плагіна “Analyze Particles...” потрібно, щоб над зображенням була виконана сегментація. Діалогове вікно для редагування згенерованої функції “обгортки” для ImageJ плагіну наведено на рисунку 3.12.

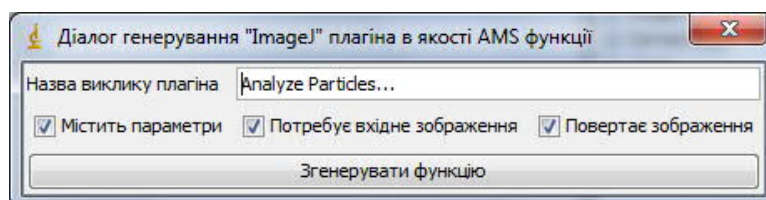


Рисунок 3.10 - Діалог генерування функції “обгортки” для ImageJ плагіна

Функція для задання умови має особливе діалогове вікно для

редагування умов (рисунок 3.13). Якщо поставити галку на “Інвертувати результат”, то результат цього вираження буде інвертований (якщо результат істина, то стане не істиною і навпаки).

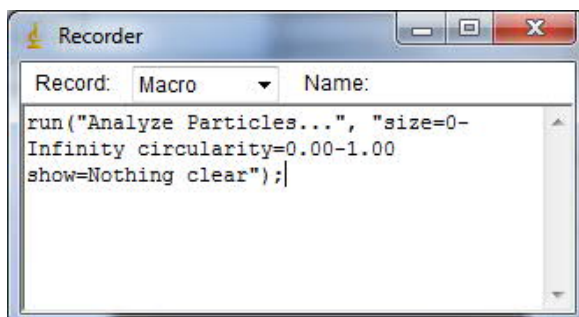


Рисунок 3.11 – Записувач макросів

Кнопки “+” та “-” додають та видаляють дужки, які можна використовувати для задання пріоритету. Кнопка “Добавити вираження” додає нижче цієї кнопки ще одне вираження, а кнопка “закрити” відповідно його видаляє. Діалогове вікно контролює правильність задання дужок і в разі їх неправильного задання буде виведена відповідна помилка (рисунок 3.14).

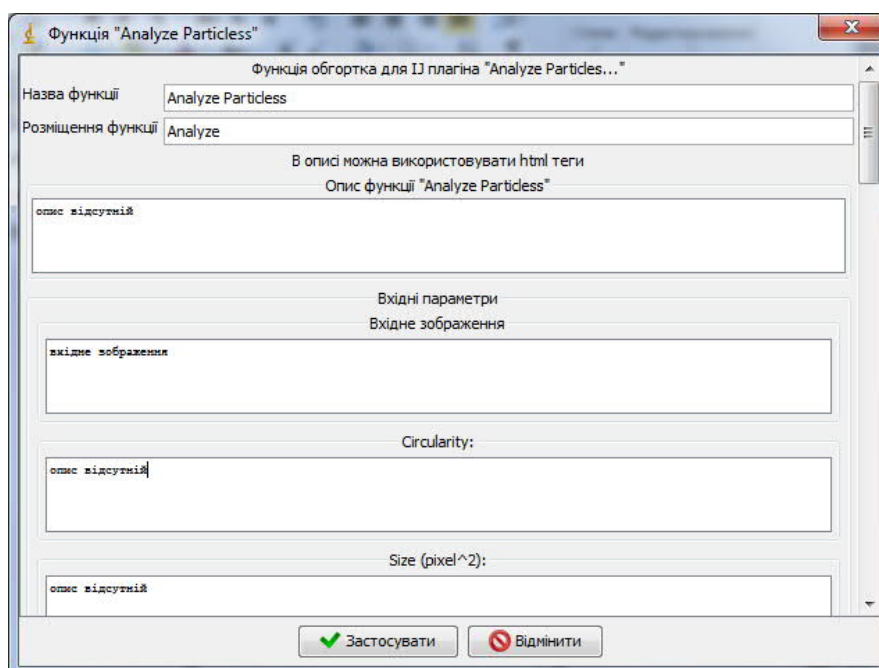


Рисунок 3.12 – Діалогове вікно для редагування згенерованої функції “обгортки” для ImageJ плагіна

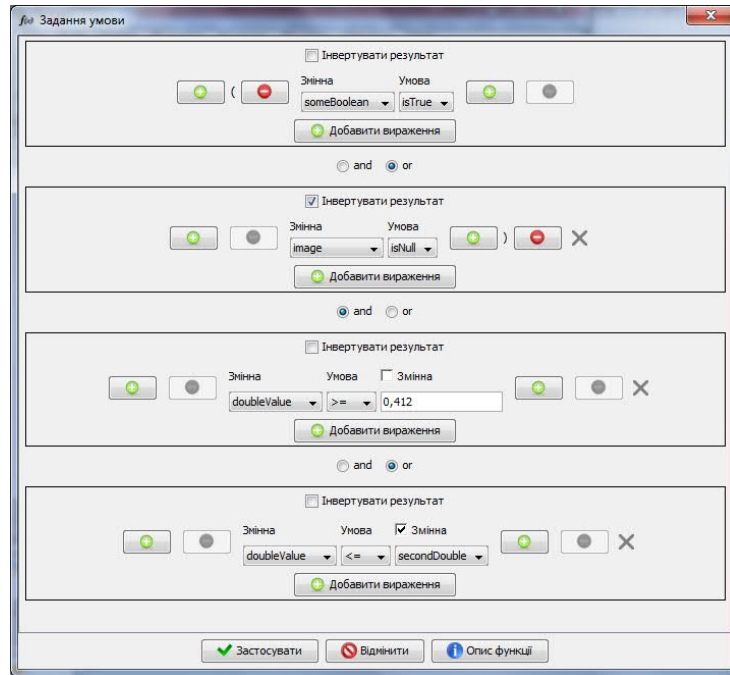


Рисунок 3.13 – Діалогове вікно для задання умови

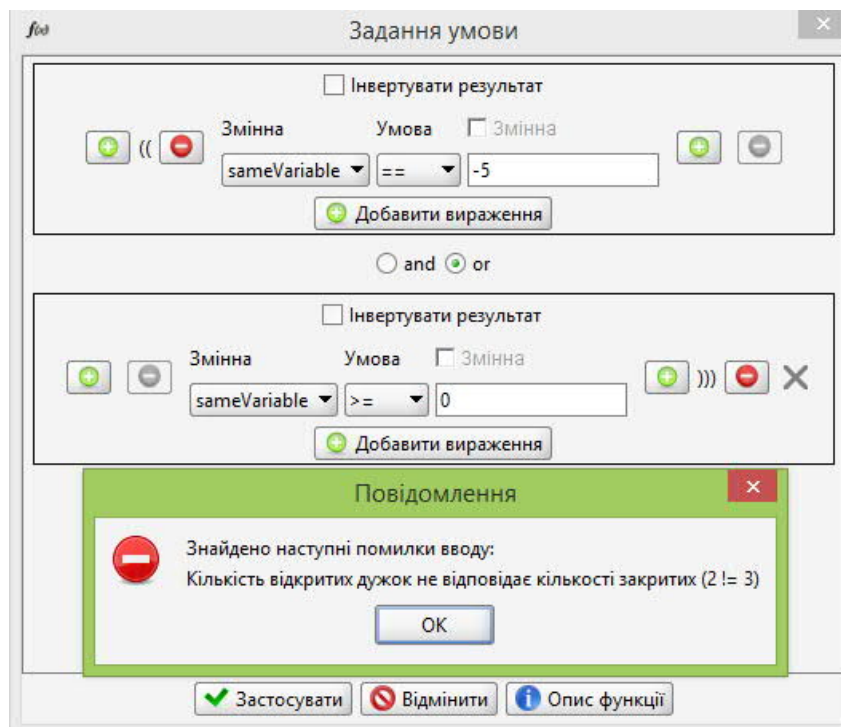


Рисунок 3.14 – Повідомлення про нерівність відкритих дужок закритим

### 3.2.2 Класи для побудови інтерфейсу користувача (вікон)

Діаграма основних класів, що відносяться до інтерфейсу користувача наведена на рисунку 3.15.



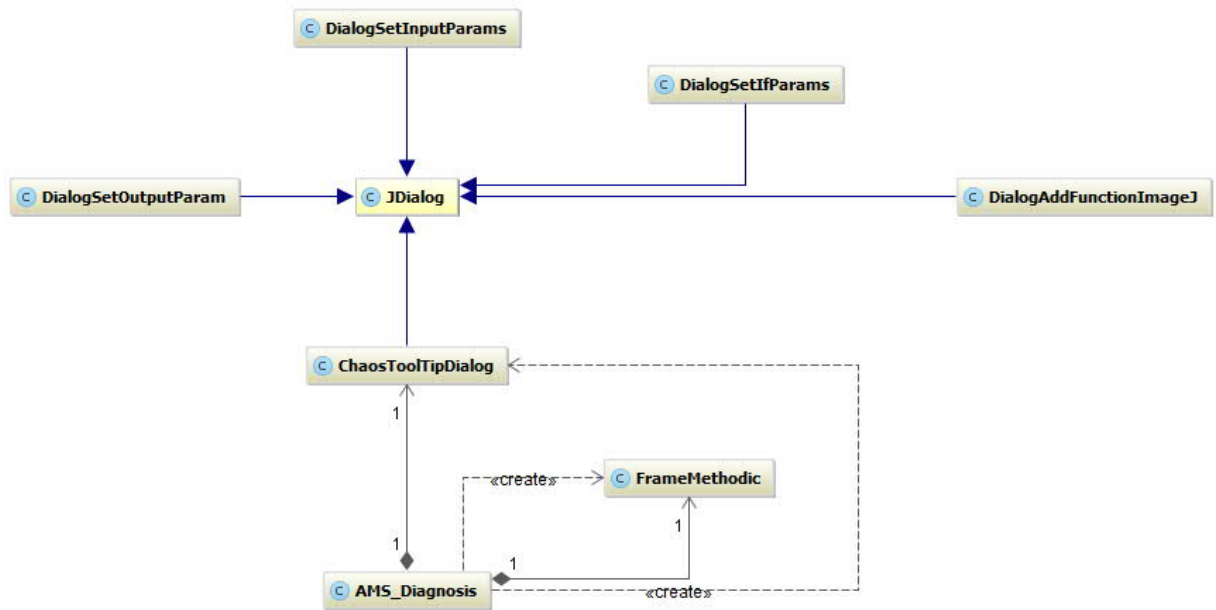


Рисунок 3.15 – Діаграма основних класів, що відносяться до інтерфейсу користувача

На рисунку можна побачити наступні класи:

1. “AMS\_Diagnosis” – початкове вікно програми (рисунок 3.2);
2. “FrameMethodic” – вікно для побудови методик та функцій (рисунок 3.6);
3. “DialogSetInputParams” – діалогове вікно для задання вхідних параметрів функції (рисунок 3.7);
4. “DialogSetOutputParam” – діалогове вікно для задання назви змінної, що повертається функцією (рисунок 3.8);
5. “ChaosToolTipDialog” – діалогове вікно з підказками або описом функцій (рисунок 3.9);
6. “DialogAddFunctionImageJ” – діалогове вікно для генерування функції обгортки для ImageJ плагіна (рисунок 3.10);
7. “DialogSetIfParams” – діалогове вікно для задання умови (рисунок 3.13).

### 3.2.3 Опис класів, що відносяться до функцій методики

На рисунку 3.16 наведена діаграма класів, які пов’язані з функціями

МЕТОДИКИ.

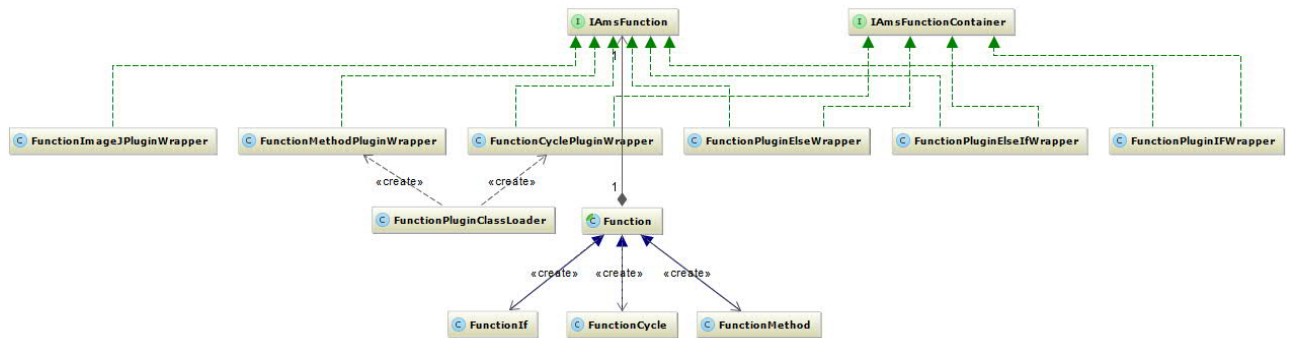


Рисунок 3.16 – Діаграма класів, які відносяться до функцій методики

З рисунку можна побачити, що наступні класи реалізують інтерфейс “IAmsFunction”: “FunctionImageJPluginWrapper”; “FunctionMethod Plugin Wrapper”; “FunctionCyclePluginWrapper”; “FunctionPluginElseWrapper”; “Function PluginElseIfWrapper”; “FunctionPluginIfWrapper”.

Інтерфейс “IAmsFunction” описує базові методи, які повинні бути реалізовані для того, щоб функція появилася у списку доступних (права частина методики із списком доступних функцій, яку можна побачити з рисунку 3.6). Кожен з вищенаведених класів реалізує методи цього інтерфейсу згідно із своїми уподобаннями. Повний код інтерфейсу з коментарями наведений нижче:

```
/* Клас повинен реалізовувати даний інтерфейс, щоб добавитись у
список доступних функцій методики. */
public interface IAmsFunction {
    /* @return повний опис функції (включає опис функції, опис
вхідних
    * параметрів, опис вихідного параметру...) */
    public String getFullFunctionDescription();
    /**      * @return загальний опис функції      */
    public String getFunctionDescription();
    / * @return шлях, де буде розміщена функція. В якості
розділення
    * використовується символ крапки ".".<br>
    * Для прикладу, якщо getTreePath повертає "Відкриття
файлів.Зображення", то функція буде розміщена у "Відкриття
файлів->Зображення->Назва функції" */
    public String getTreePath();
}
```

```

        /          * @return назва функції, що буде відобразитися у
програмі          */
    public String getFunctionName();
    /   * @return клас у якому знаходиться функція(метод)          */
    public Class<?> getFunctionClass();
    /   * @return вхідні параметри функції          */
    public FunctionParameter[] getFunctionInputParams();
    /**          * @return параметер, що функція повертає          */
    public FunctionParameter getFunctionReturnParam();
    /**          * @return іконка функції          */
    public ImageIcon getImageIcon();
    /* Метод, що буде викликатися інтерпретатором на виконання
    *   @param inputParams вхідні параметри, що будуть
передаватися
    *   інтерпретатором
    *   @return Результат, що повертає функція          */
    public Object invoke(Object... inputParams) throws Exception;
}

```

Також з рисунку 3.16 можна побачити, що наступні функції реалізують інтерфейс “IAmsFunctionContainer”: “FunctionCyclePluginWrapper”; “FunctionPluginElseWrapper” ; “FunctionPluginElseIfWrapper”; “Function PluginIf Wrapper”.

Даний інтерфейс не містить жодних методів і лише є знаком того, що функція є контейнером для інших функцій, тобто в її тілі можуть бути розміщені інші функції.

Клас “FunctionPluginClassLoader” призначений для завантаження плагінів із носія інформації, їх розпізнавання та додання у список доступних функцій. Як можна побачити з рисунку 3.16 в даний момент він розпізнає лише 2 плагіни, “FunctionMethodPluginWrapper” та “FunctionCyclePluginWrapper”.

Також з рисунку 3.16 можна побачити, що інтерфейс “IAmsFunction”, а отже і усі класи, що його реалізують пов’язані з абстрактним класом “Function”. Клас “Function” і всі його похідні класи будують уже саму методику (рисунок 3.6 ліва частина методики). Тобто користувачем задається порядок їх виконання, їх вхідні та вихідні параметри. Сам клас реалізує багато функцій, зокрема будує кнопки, їх представлення та дії, що відбуваються по їх натисненню, а класи, що походять від нього задають ці дії, коректують

представлення функції у лівому вікні або роблять усе, що їм потрібно.

### 3.2.4 Функція “обгортка” для існуючих плагінів ImageJ

На рисунку 3.17 – зображена діаграма класів пов’язаних з функцією “обгорткою” для існуючих ImageJ плагінів.

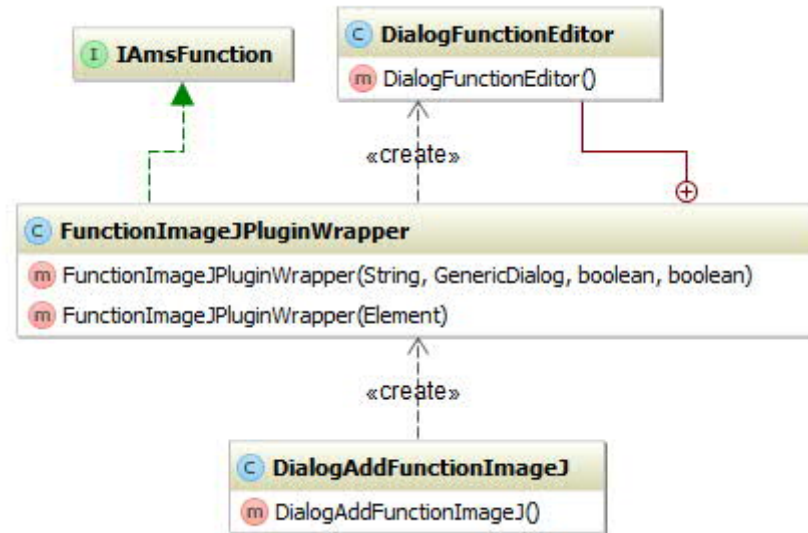


Рисунок 3.17 – Діаграма класів функції “обгортки” для існуючих ImageJ плагінів

Після того як користувач ввів відповідні дані і не відбулося помилок у діалоговому вікні “DialogAddFunctionImageJ” (рисунок 3.10), то генерується функція “обгортка” “FunctionImageJPluginWrapper” за допомогою введених користувачем параметрів та діалогового вікна самого ImageJ плагіну. Для цього використовується конструктор класу “обгортки” для генерування параметрів плагіну із діалогового вікна, який виглядає наступним чином:

```
/ * Даний конструктор викликається для генерування параметрів
функції із
* діалогового вікна ImageJ плагіна
* @param gd Вікно з параметрами плагіна для аналізу. Якщо ==
null, то
* плагін без параметрів
* @param pluginNameToRun Назва плагіна для виклику
* @throws NoSuchFieldException
```

```

        * @throws IllegalAccessException      */
    public FunctionImageJPluginWrapper(String pluginNameToRun,
        GenericDialog gd,
            boolean isImageRequired, boolean isImageReturn) throws
                NoSuchFieldException,
        IllegalAccessException {...}

```

Для розпізнавання параметрів використовується Java Reflection API, що дозволяє аналізувати клас під час виконання програми, а також отримувати доступ навіть до “private” полів та методів.

Згенерована функція “обгортка” зберігається у носії інформації в XML форматі для відновлення згенерованих функцій при наступному запуску програми. Конструктор для відновлення функції із XML формату виглядає наступним чином:

```

/* Даний конструктор використовується для загрузки функцій із
файлу
 * @param elemFunc Кореневий XML-тег (Function) функції, для
відновлення із
 * XML */
public FunctionImageJPluginWrapper(Element elemFunc) throws
        ClassNotFoundException {...}

```

Клас “DialogFunctionEditor” як можна побачити з рисунку 3.12 використовується для редагування користувачем функції “обгортки” (здебільшого опис параметрів).

### 3.2.5 Опис класів, що реалізують інтерпретатор

На рисунку 3.18 зображена діаграма класів із їх методами, що відносяться до інтерпретатора.

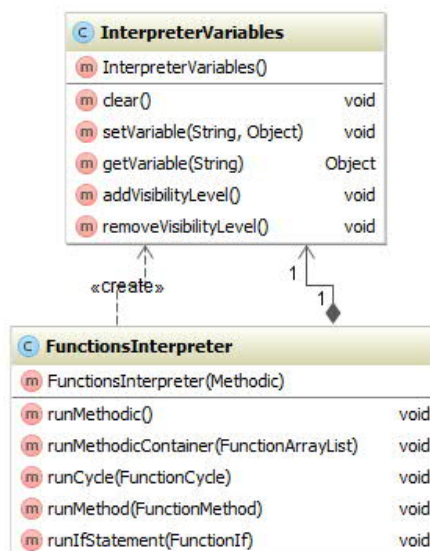


Рисунок 3.18 – Діаграма класів із методами, що відноситься до інтерпретатора

Клас “InterpreterVariables” містить змінні, що отримуються під час виконання інтерпретатором. Клас “FunctionsInterpreter” запускає методику на виконання.

3.2.6 Необхідні класи для додання функцій написаних на Java у методику На рисунку 3.1 наведена діаграма класів, про які необхідно знати користувачеві, щоб додати власну функцію, яка написана на Java.

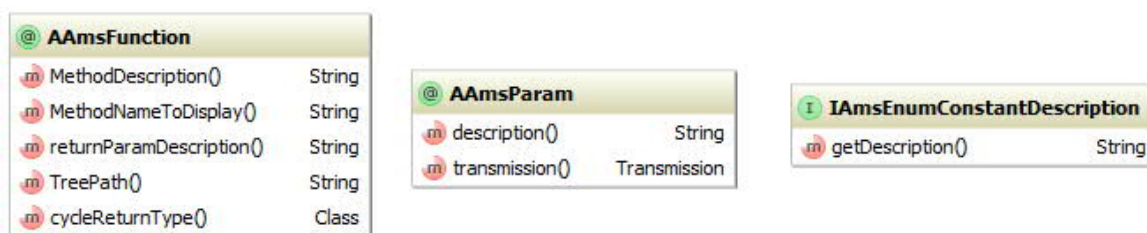


Рисунок 3.19 - Діаграма класів для додання функцій написаних на Java

Частина коду та повний опис наведено нижче. Метод, що помічається даною анотацією появляється у вікні "AMS - Методика" в якості функції. Якщо анотація використовується для не статичного метода то клас повинен реалізовувати конструктор без параметрів (не обов'язково public). Конструктор без параметрів буде використовуватися для створення об'єкту

кожен раз, коли в методиці буде викликатися його функція (дійсно для не статичних методів), Якщо анотація використовується для конструктора, то даний клас повинен реалізовувати інтерфейс Iterable. Використання анотації над конструктором значить, що дана функція появиться в якості цикла.

Файл AAmsFunction.java:

```
package Diagnosis.AmsMethodic.functions.forUsers.anoations;
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
@Target(value = {ElementType.METHOD, ElementType.CONSTRUCTOR})
@Retention(value = RetentionPolicy.RUNTIME)
public @interface AAmsFunction {
    /* @return Опис методу, що буде відобразатись у програмі.<br>
    * Допускається використання html тегів */
    String MethodDescription();
    /* @return методу, що буде відобразатись у програмі */
    String MethodNameToDisplay();
    /* @return Опис значення, що повертається (якщо метод типу
void, то ця
    * строка ігнорується)<br>
    * Допускається використання html тегів */
    String returnParamDescription();
    /* @return шлях, де буде розміщена функція. В якості
розділення
    * використовується символ крапки ".".<br>
```

Для прикладу, якщо `getTreePath` повертає "Відкриття файлів.Зображення", то функція буде розміщена у "Відкриття файлів->Зображення->Назва функції".

```
String TreePath();
    /* Якщо анотація використовується для конструктора (цикл), то
цей параметр
    * мусить вказати дійсний тип значення (клас об'єкта), що
повертається
    * ітератором.
    * Для методів ігнорується */
    Class cycleReturntype() default Object.class;
}
```

Файл AAmsParam.java:

```
package Diagnosis.AmsMethodic.functions.forUsers.anoations;
import java.lang.annotation.*;
/* Опис параметра */
```

```

@Target(ElementType.PARAMETER)
@Retention(RetentionPolicy.RUNTIME)
public @interface AAMSParam {
    /* Спосіб передачі параметра */
    enum Transmission {
        /* Передача параметра допускається лише по силці */
        BY_LINK,
        /* Передача параметра допускається лише за значенням */
        BY_VALUE,
        /* Спосіб передачі задається користувачем при введенні
вхідних параметрів
* функції (використовується по замовчуванням) */
        USER_SELECT}
    / * @return Опис параметра, що буде відобразитись у
програмі<br>
* допускається використання html тегів */
    String description();
    / * @return Спосіб передачі параметра у функцію */
    Transmission transmission() default Transmission.USER_SELECT;
}

```

### 3.2.7 Приклад створення користувацьких функцій

Нижче наведений приклад написання Java метода для його додання у список доступних функцій методики:

```

@AAMSFunction(
    MethodDescription = "Функція перетворює зображення у
інший тип",
    MethodNameToDisplay = "Convert image type",
    returnParamDescription = "Оброблене вхідне
зображення",
    TreePath = "ImageJ.ImagePlus"
)
public static ImagePlus convertImageType(
    @AAMSParam(description = "Зображення для перетворення
у інший тип")
    ImagePlus imagePlus,
    @AAMSParam(description = "Тип перетворення")
    ConvertType convertType
)
{
    final ImageProcessor ip;
    if (convertType == ConvertType.BYTE)
        ip = imagePlus.getProcessor().convertToByte(false);
    else if (convertType == ConvertType.SHORT)
        ip = imagePlus.getProcessor().convertToShort(false);
    else if (convertType == ConvertType.RGB)
        ip = imagePlus.getProcessor().convertToRGB();
    else if (convertType == ConvertType.FLOAT)
        ip = imagePlus.getProcessor().convertToFloat();
}

```



```

        else throw new IllegalArgumentException("Не відомий тип
зображення");
        imagePlus.setProcessor(ip);
        return imagePlus;
    }

```

Анотацій “AAmsParam” можна і не використовувати, але у цьому випадку параметр не буде описаним у методиці. Обов’язковою є лише анотація “AAmsFunction” без якої метод не буде додано у список доступних функцій методики. У даному методі є параметр “ConvertType”, який є перелічуваним типом (може приймати тільки визначенні значення). Перелічуваний тип може реалізувати інтерфейс “IAmsEnumConstantDescription” для опису своїх перелічуваних значень.

Його реалізація не є обов’язковою і впливає лише на опис його значень. Нижче наведений приклад його реалізації перелічуваним типом “ConvertType”:

```

public enum ConvertType implements IAmsEnumConstantDescription {
    BYTE, SHORT, RGB, FLOAT;
    @Override public String getDescription() {
        if (this == BYTE) return "8-бітне зображення
(ByteProcessor)";
        else if (this == SHORT) return "16-бітне зображення
(ShortProcessor)";
        else if (this == RGB) return "RGB зображення
(ColorProcessor)";
        else if (this == FLOAT) return "32-бітне зображення
(FloatProcessor)";
        else throw new IllegalArgumentException("Unknown type of
image");
    }
}

```

Після цього, щоб метод появився у списку доступних функцій методики, потрібно архівувати ці скомпільовані класи у “.jar” архів та помістити його у папку “plugins” плагіна “AMS Diagnosis”. Результат додання вищенаведеного Java методу наведений на рисунку 3.20.

### 3.3 Реалізація бази даних

В програмі використовується база даних MySQL для зберігання даних про дослідження. Також, щоб спростити взаємодію з БД використовується JPA, точніше одна з його реалізацій – Hibernate. Нижче наведено призначення таблиць:

- PatientHbm – таблиця із даними про пацієнта;
- ResearchHbm – таблиця із даними про дослід;
- ImageHbm – таблиця із даними про зображення;
- ImageAttributeHbm – таблиця із даними про атрибути зображення (назва та значення);

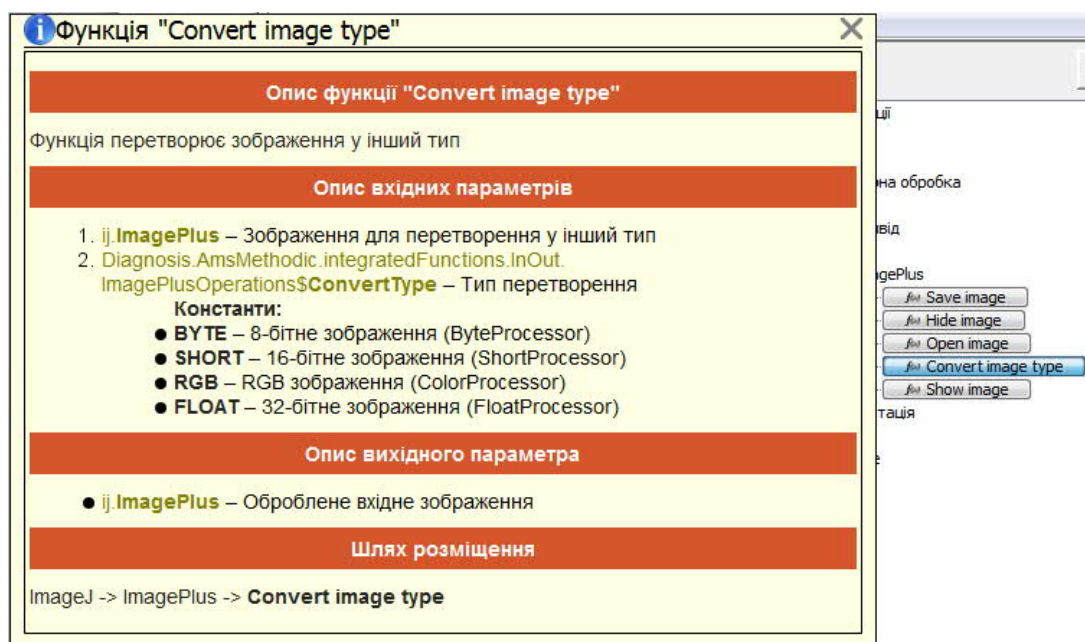


Рисунок 3.20 – Доданий Java метод у список доступних функцій методики

Результат додання цього користувацького плагіну-циклу наведений на рисунку 3.21.

- ImageAttributeNameHbm – таблиця, що містить назви атрибутів зображень;
- ImageAttributeValueHbm – таблиця, що містить значення можливі

значення для атрибутів зображень;

– ImageMeasurementNameHbm – таблиця, що містить назви числових вимірювань для зображень;

– ImageMeasurementsContainer – таблиця, що містить значення числових вимірювань певної назви для зображень;

– ImageMeasurementsContainer\_Measurements – таблиця, що містить значення для вищенаведеного контейнера;

– RoiClassHbm – таблиця, що містить класи об'єктів, які належить зображенню;

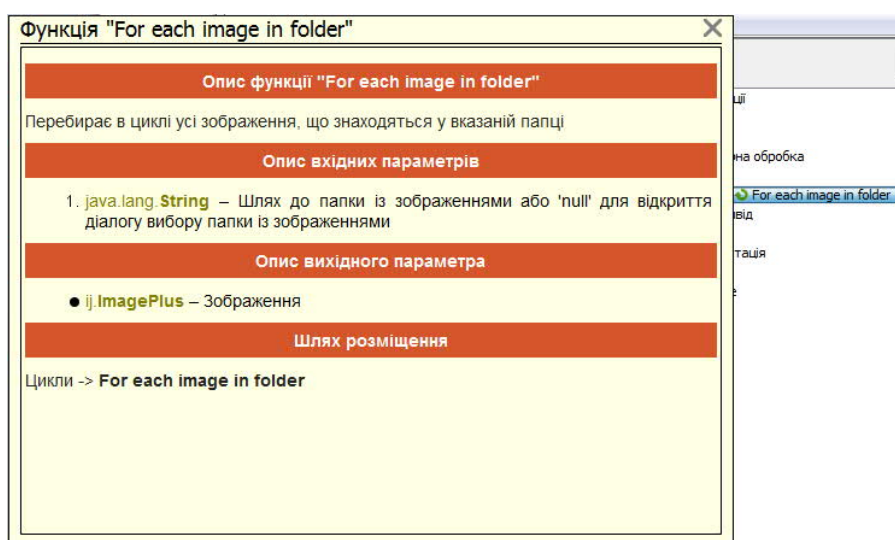


Рисунок 3.21 – Результат додавання користувацького циклу у список доступних функцій методики

– ClassNameHbm – таблиця, що містить назви класів об'єктів;

– ClassName\_ClassName – таблиця, що містить ієрархія назв класів об'єктів (надкласи, підкласи);

– RoiGroupHbm – таблиця, що містить групи об'єктів (одна група може мати багато об'єктів, ця група вважається як цілий об'єкт);

– RoiGroupContainer – таблиця, що містить контейнер підкласів об'єктів для надкласу;

– roi\_group\_container\_roi\_group – таблиця, що містить детектовані об'єкти



(рисунок 3.24). На рисунку 3.25 наведено вікно для вибору методики по якій буде виконуватись дослідження. На рисунках 3.26 - 3.27 наведено інші частини вікна.

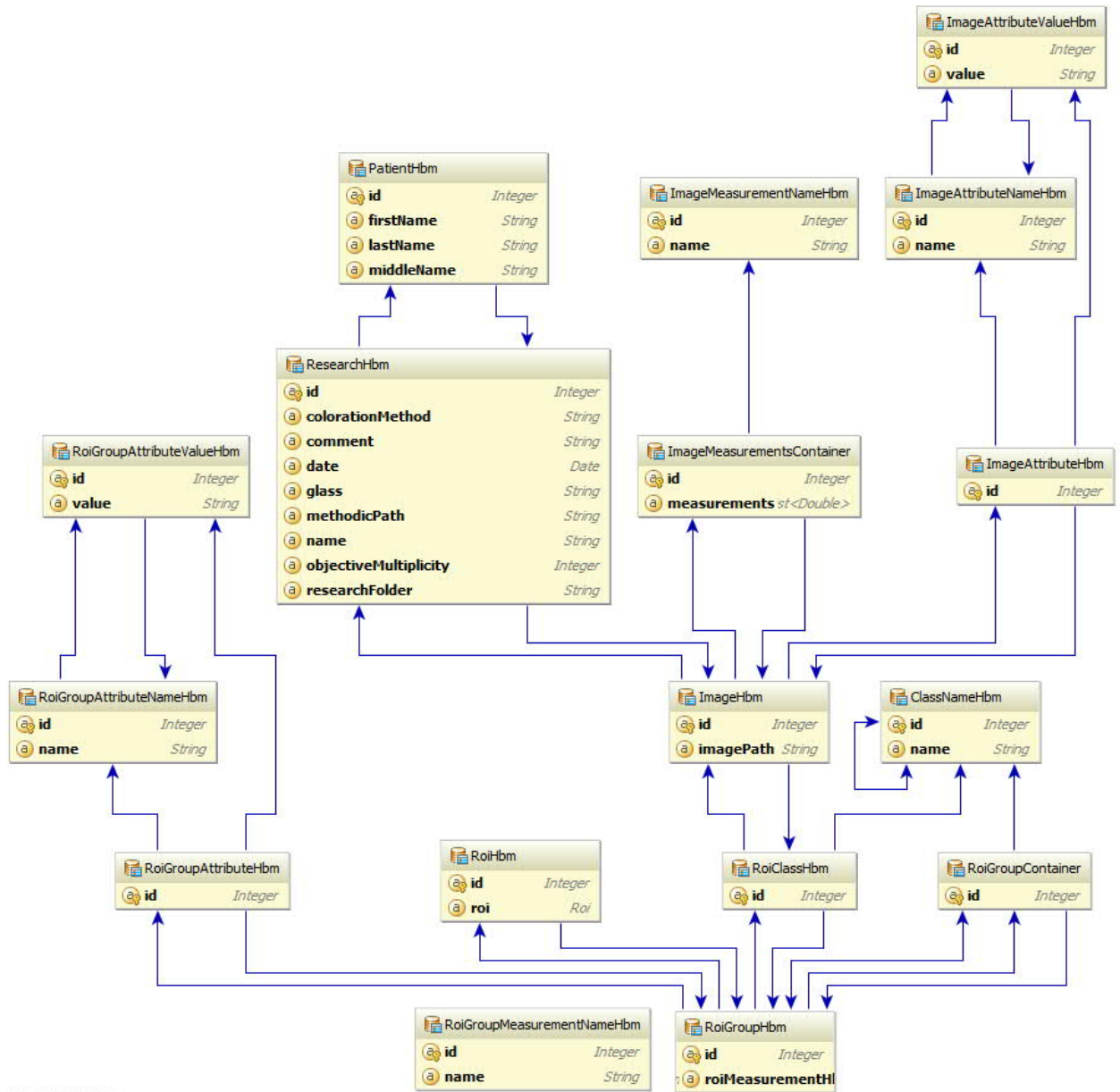


Рисунок 3.23 – Діаграма зв'язків класів сутностей, що представляють БД

З рисунку 3.24 можуть викликати деякі запитання про поля “Папка досліджу” та “Методика”. “Папка досліджу” – це директорія з якої будуть вибрані зображення для досліджу. “Методика” – це скрипт, що буде викликаний після натиснення на кнопку “Розпочати дослід”.

У вікні, що наведено на рисунку 3.26 потрібно вибрати об’єкти для

дослідження. За допомогою кнопки “Добавити” можна додати нову назву класу об’єктів (рисунк 3.28).

У вікні, що наведено на рисунку 3.27 здійснюється вибір ознак для обчислення. Питання можуть викликати поля “Клас ядер” та “Клас цитоплазми”. У цих полях потрібно вказати клас ядер та цитоплазми, по яких буде обчислюватися ядерно-цитоплазматичне відношення.

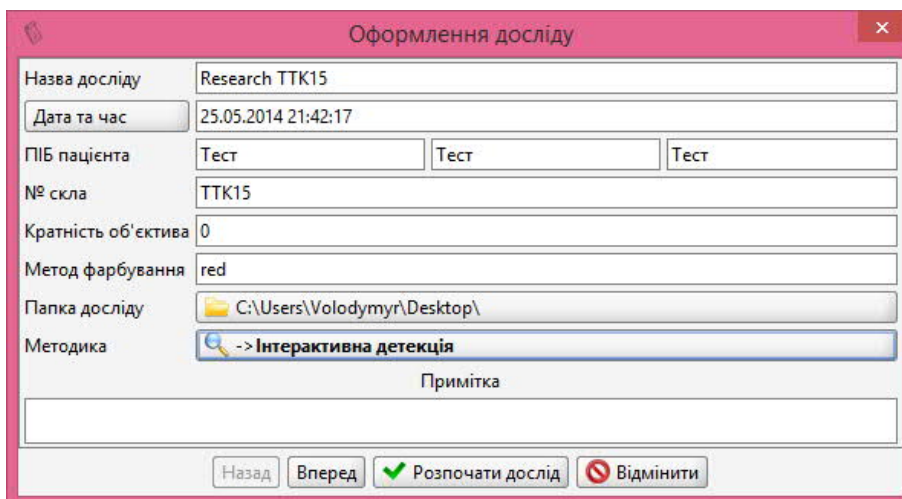


Рисунок 3.24 – Введення загальної інформації про дослідження

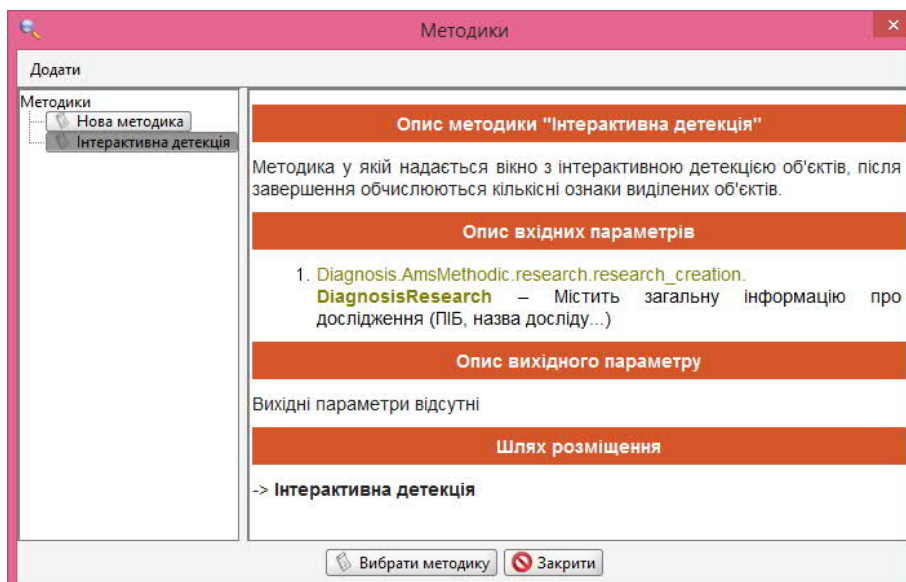


Рисунок 3.25 – Вікно для вибору методики

Після запуску дослідження запускається вибраний скрипт. В даному випадку для прикладу був вибраний скрипт “Інтерактивна детекція”. Даний

скрипт показує користувачеві вікно, що наведено на рисунку 3.29 для ручного виділення об'єктів. Також якщо дослід був перезапущений то за допомогою даного вікна можна відредагувати раніше виділенні об'єкти.

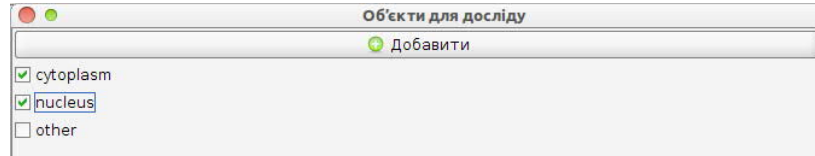


Рисунок 3.26 – Вибір об'єктів для дослідження

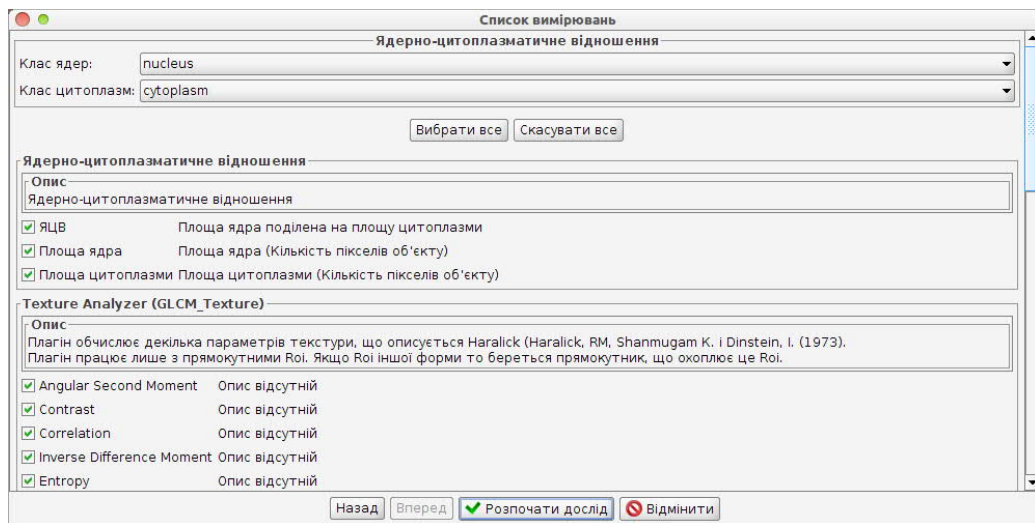


Рисунок 3.27 - Вибір ознак для обчислення

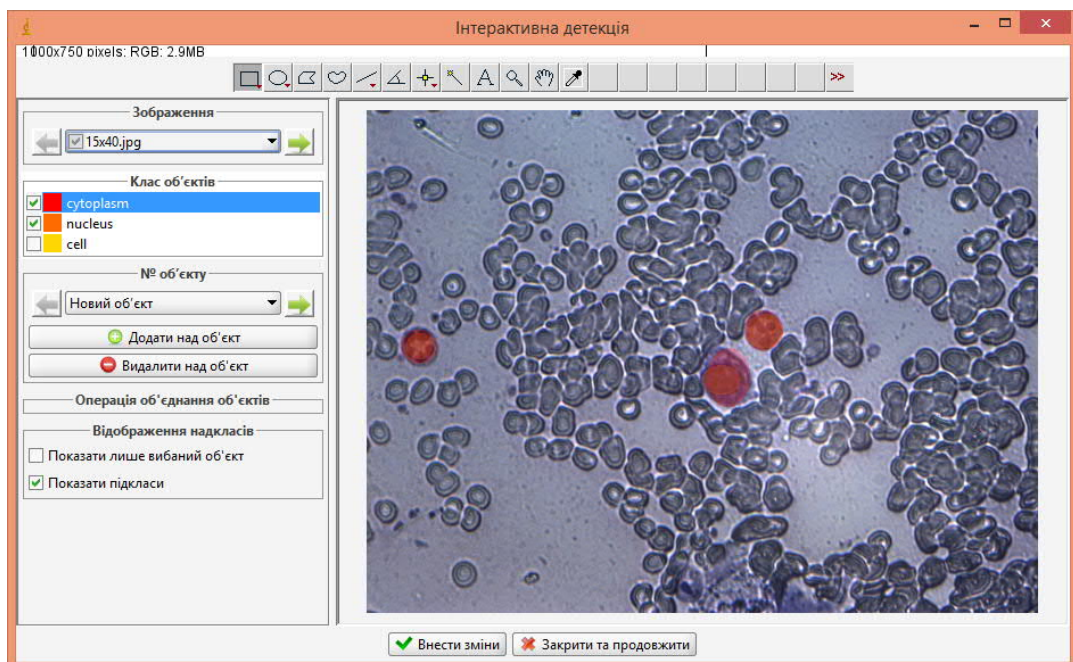
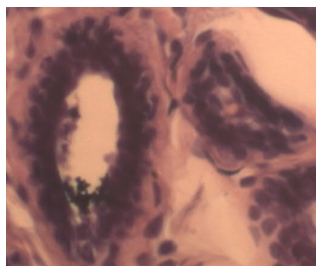


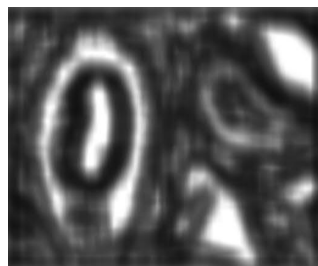
Рисунок 3.28 – Інтерактивна детекція

### 3.3 Експериментальні дослідження

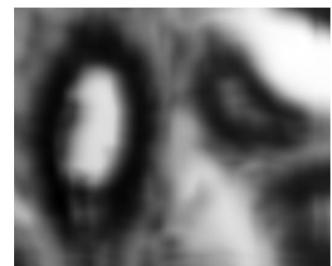
На рисунку 3.29 наведено фрагмент зображення гістологічного зрізу тканини молочної залози [16]. На рисунок 3.10,а наведено поле текстурних ознак побудоване на основі обчислення значень матриць розподілу рівнів сірого (МС), на рисунку 3.10,б – на основі геометричних моментів-ознак. Для тестування МС як текстурної ознаки використано значення ентропії  $g(i,j) = F_5$ .



а) Вихідне зображення



б) поле ознак МРРС



в) поле ознак ПМ

Рисунок 3.29 – Перетворення вхідного зображення

Приклади сегментації зображень і виділення дольок молочної залози наведені на рисунку 3.30. Зміна  $\rho$  при зміні параметру  $k$  при сталій кількості ітерацій наведена на рисунку 3.12, а. Зміна  $\rho$  при різній кількості порогів наведена на рисунку 3.12,б.

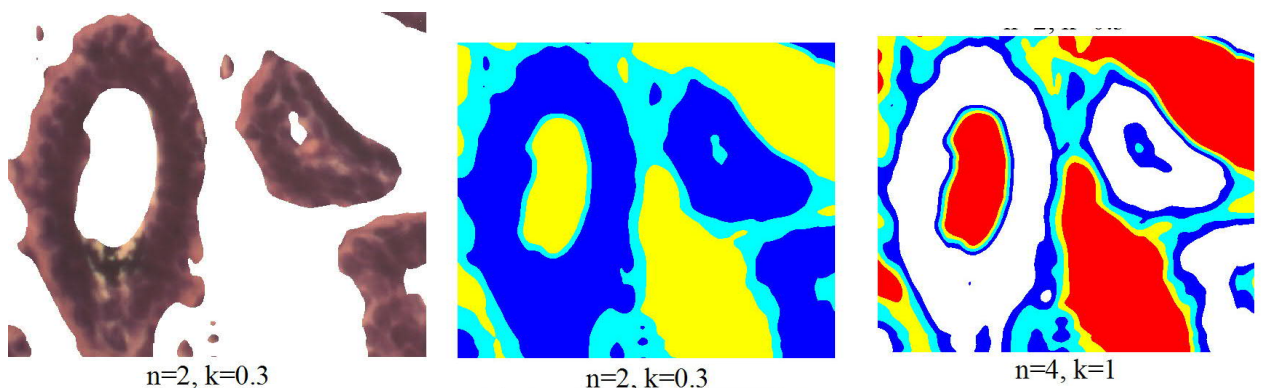


Рисунок 3.30 – Сегментація текстурного поля ПМ

Параметр  $k$  служить для керування інтервалом між найменшим і найбільшим порогом. Оптимальне число порогів  $n$  (відповідно число ітерацій



алгоритму) можна задавати апріорно, виходячи із застосування або обирати на основі зміни значення відношення сигнал/шум  $\rho$ . Значення  $\rho$  можна обчислювати між вихідним зображенням в градаціях сірого  $I_g$  та сегментованим  $I_{segm}$  з усередненими значеннями пікселів в середині сегментів:

$$\rho = PSNR(I_g, I_{segm}).$$

Алгоритм демонструє вище значення відношення сигнал/шум при заданій кількості порогів і використанні текстурних ознак на основі ПМ.

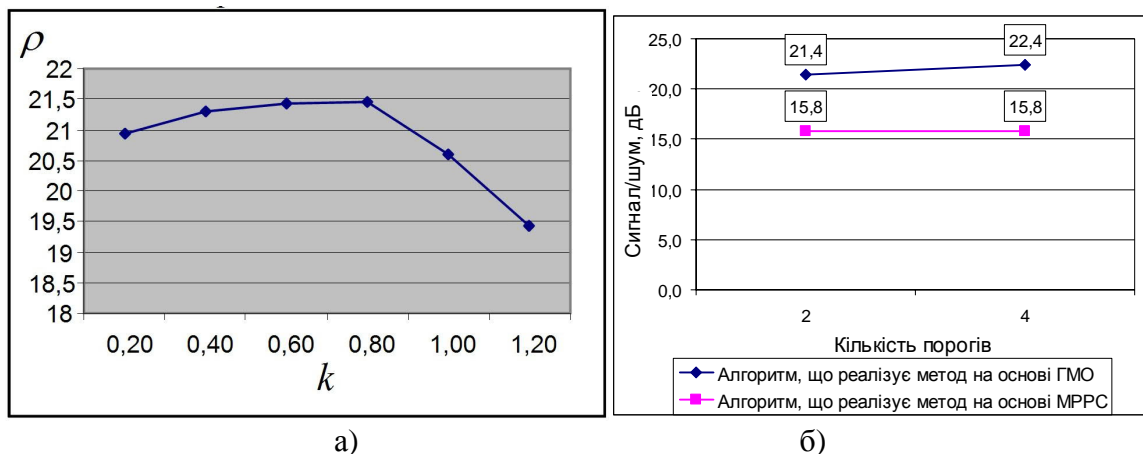


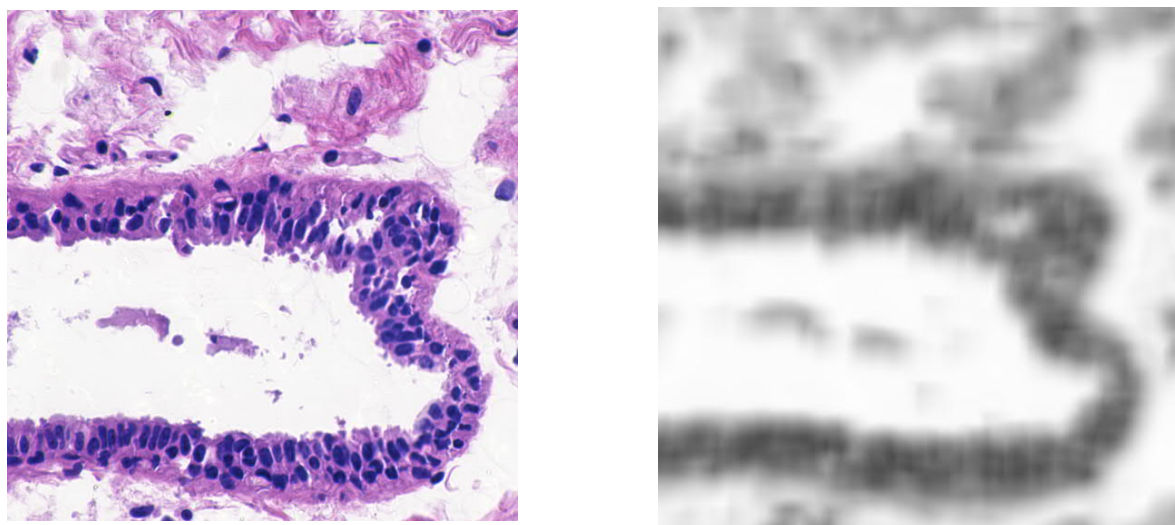
Рисунок 3.31 - Залежність значення відношення сигнал-шум від зміни  $k$

При підвищенні кількості порогів якість сегментації зростає, але вибір їх кількості залежить і від застосування алгоритму. При сегментації гістологічних зображень кількість порогів обирається в залежності від кількості типів тканин які потрібно виділити.

Продовжимо досліді.

Гістологічні зображення містять такі складні мікрооб'єкти, як поперечні перерізи залоз і протоків. На рисунку 3.32,*a* зображений фрагмент гістологічного зображення тканини молочної залози. Для експериментальних досліджень алгоритмів текстурної сегментації використано базу із 100

гістологічних зображень. Проведено сегментацію на основі алгоритму ПМ. На рис. 3.32,б показано простір текстурних ознак.

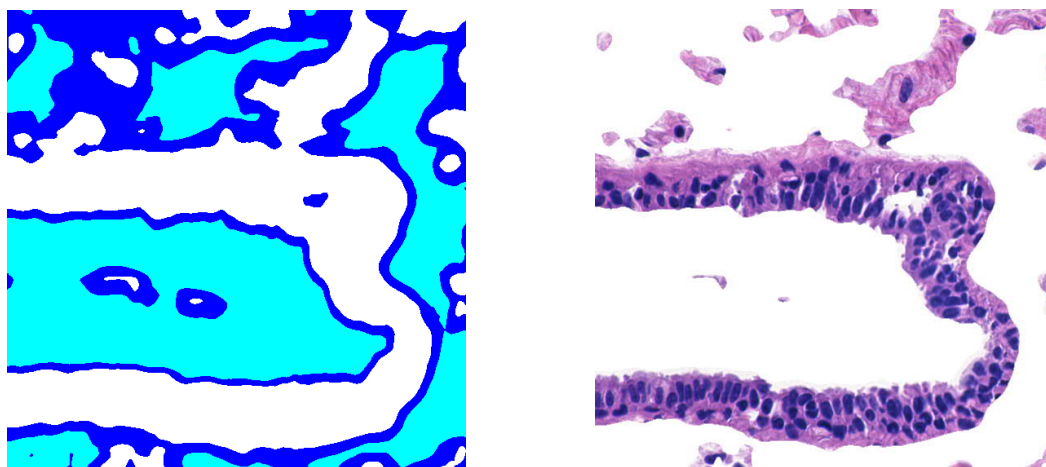


*a*

*б*

Рисунок 3.32 - Фрагмент вихідного зображення та простір текстурних ознак: *a* - вихідне зображення; *б* - простір ознак текстури

Простір текстурних ознак розбивається на області шляхом обчислення порогів. Пронумеровані області утворюють розмітку (рисунок 3.33,а). Одна із однорідних областей відповідає протоці молочної залози показану на рисунок 3.33,б.



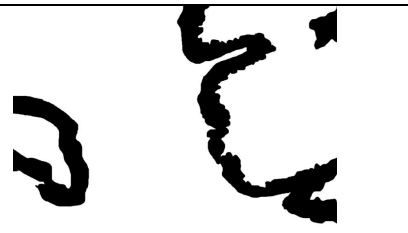




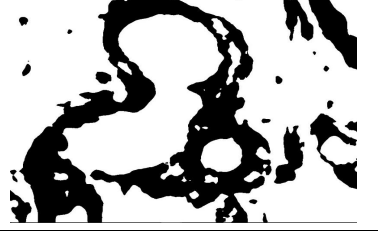


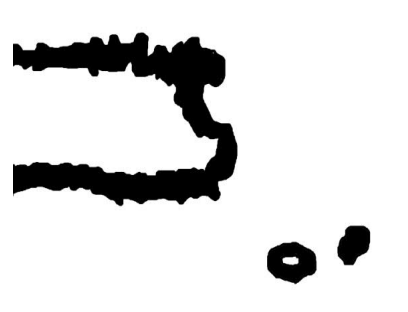
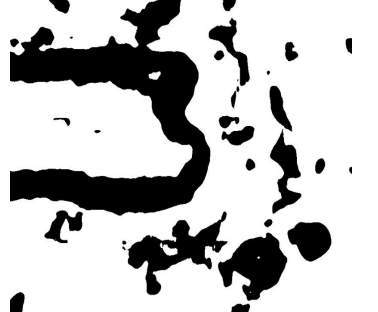
*a*

*б*

Рисунок 3.33 - Сегментація зображення та виділення однорідної області: *a* - розмічене зображення, *б* – область протоки молочної залози

Результати порівняння якості текстурної сегментації на основі відстані Хаусдорфа для контурів показано у таблиці 3.2 .

Таблиця 3.2 - Оцінка алгоритмів текстурної сегментації

Експертна сегментація	Текстурна сегментація	Значення відстані	
		Просторові моменти	Матриці суміжності
		122,14	158,78
		335,62	436,31
		111	155,40
		422,9	549,77
		170,35	221,46

Розроблений алгоритм текстурної сегментації на основі просторових моментів дозволяє сегментувати групу клітин (тканину). Перевагою

алгоритму є більша точність сегментування порівняно з алгоритмами на основі матриць розподілу рівнів сірого на 30% та менша кількість параметрів для налаштування.

Наступні дослідження потрібно проводити в напрямку розрахунку розміру ковзаючого вікна обчислення просторових моментів. Це дозволить проводити автоматичне налаштування параметрів алгоритму відповідно до кратності збільшення мікроскопа або масштабу зображення.

### 3.4 Висновки до розділу

Застосування ознак просторових моментів для сегментації текстур дозволило ідентифікувати складні мікрооб'єкти, такі як шари клітин, частини кровоносних судин та протоків молочної залози. Для оцінки якості сегментації обчислено відстань Хаусдорфа між контурами однорідних сегментів.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи одержані наступні результати:

2. Систематизовано текстурні характеристики біомедичних зображень. Біомедичні зображення мають складну структуру. Вони містять шари тканин утворені окремими клітинами котрі розміщені в певному порядку. Клітини оточені різнорідним фоном.

3. Проведено аналіз методів сегментації зображень. Оскільки біомедичні зображення володіють текстурними характеристиками то для їх аналізу застосуємо обчислення статистичних ознак.

4. Розроблено алгоритми текстурної сегментації зображень на основі матриць суміжності пар пікселів та просторових моментів, що дозволило сегментувати на зображення складні об'єкти – поперечні переріз пластів тканин, частини кровоносних судин та протоків залоз.

5. Розроблено структуру системи аналізу біомедичних зображень, що складається з: блоку інтерфейсу користувача, блоку попереднього опрацювання зображення, блоку сегментації, блоку виділення мікрооб'єктів, блоку обчислення кута орієнтації, блоку обчислення числових ознак та блоку зберігання опису зображення у БД. Розроблювані алгоритми реалізовано у вигляді модуля розширення до програмного пакета ImageJ на мові Java.

6. Оцінено якість сегментації проведеної розробленими алгоритмами. Застосування ознак просторових моментів для сегментації текстур дозволило підвищити якість сегментації в середньому на 30 відсотків.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баган Я.А., Бабій Ю.В. Алгоритми текстурної сегментації на основі статистичних ознак. III Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020, Тернопіль, Україна. Тернопіль: ЗУНУ, 2020. с.23
2. Бабій Ю.В., Баган Я.А. Алгоритми розпізнавання дорожніх знаків для мобільних пристроїв. III Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020, Тернопіль, Україна. Тернопіль: ЗУНУ, 2020. с.24
3. Мельник Г.М. Метод і алгоритми аналізу симетричних зображень. Штучний інтелект. 2010. № 4. С. 253–261.
4. Мельник Г.М. Метод знаходження відповідних точок на контурах мікрооб'єктів біомедичної природи. Вісник Національного університету "Львівська політехніка" "Комп'ютерні науки та інформаційні технології" 2012., №732. С. 343–350.
5. Мельник Г.М. Інформаційна технологія опрацювання гістологічних зображень. Вісник Хмельницького національного університету. Технічні науки. 2012. № 5. С.154–161.
6. Мельник Г.М. Аналітичний огляд методів аналізу та синтезу текстурних зображень. Вісник Хмельницького національного університету" 2007. Т.1, №2 С. 110–114.
7. Березький О.М., Мельник Г.М., Батько Ю.М. Текстура сегментація біомедичних зображень на основі просторових моментів. Матеріали 4-ї міжнародної науково-технічної конференції "Комп'ютерні науки та інформаційні технології". 15-17 жовтня, 2009, Львів, Україна. Львів: ПП "Вежа і Ко", 2009 С.42–45.
8. Veltkamp R. C. Hagedoorn M. State of the art in shape matching. Principles of visual information retrieval. London, UK: Springer, 2001. 87-119 p.

9. Towards efficient automated characterization of irregular histology images via transformation to frieze-like patterns / B. A. Canada, G. K. Thomas, K. C. Cheng, James Z. Wang, Y. Liu. CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval 2008. P.581-590.

10. Коксетер Г.С.М., Мозер У.О.Дж. Порождающие элементы и определяющие соотношения дискретных групп: Пер. с англ. М.: Наука, 1980. 240 с.

11. Liu Y. W.-C. Lin, J. H. Hays Near-regular Texture Analysis and Manipulation. ACM Transactions on Graphics (SIGGRAPH 2004). P. 368-376.

12. Liu Y. Robert T. Collins, Y. Tsin Computational Model for Periodic Pattern Perception Based on Frieze and Wallpaper Groups. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). 2004. Т. 1, № 26. С. 354– 371

13. Leung T., Malik J. Detecting, localizing and grouping repeated scene elements. 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings. 1996. P. 546-555.

14. Schaffalitzky F., Zisserman A. Geometric grouping of repeated elements within images . In Proc. 9th British Machine Vision Conference, Southampton - Southampton, 1998. P. 13-22.

15. Lin H.C. , Wang Ling-Ling, Yang Shi-Nine Extracting periodicity of a regular texture based on autocorrelation functions. Pattern Recognition Letters. 1997. V. 5, № 18. P. 433-443

16. Schaffalitzky F., Zisserman A. Geometric Grouping of Repeated Elements within Images. Shape, Contour and Grouping in Computer Vision. Lecture Notes In Computer Science. 1999. Num 1681. p. 165-181

17. Liu Y. , Collins R. Frieze and Wallpaper Symmetry Groups Classification under Affine and Perspective Distortion. Robotics Institute. Pittsburgh, PA.,US, 1998. 56 p.

18. Bieri M. Wethmar A., Wey N. Quantitative analysis of Alzheimer plaques in mice using virtual microscopy. 1st European Workshop on Tissue Imaging and Analysis 2009. P. 23-29.

19. Berezsky, O.M. Design of computer systems for biomedical image analysis / O.M. Berezsky, K.M. Berezska, Y.M. Batko, G.M. Melnyk // Proceedings of the X-th International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics" CADSM'2009, 24-28 February 2009, Lviv-Polyana, Ukraine. - 2009. - P. 186-192.
20. Russ J. The Image Processing Handbook, second Edition. US: CRC Press, 1995. 688 p.
21. Tuceryan M. Jain A. Texture Analysis. Handbook of Pattern Recognition and Computer Vision (2 nd Ed). World Scientific Publishing Co., 1998. P. 207-249.
22. Pratt W. K. Digital Image Processing: PIKS Scientific Inside. NY, USA: John Wiley and Sons, Inc., 2007. 782 p.
23. Автандилов Г.Г. Основы количественной паталлогической анатомии. М.: Медицина , 2002. 240 с.
24. Petushi S. Garcia F. Haber M. Katsinis C., Tozeren A. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer . BMC Medical Imaging. 2006. V. 1, № 6. p. 14
25. Argyris, G. 3D Terrain Modelling of the Amyntaio–Ptolemais Basin / G. Argyris, I. Kapageridis, A. Triantafyllou // 2nd International Workshop in “Geoenvironment and Geotechnics”, Milos island, Greece, 2008.
26. Berezsky, O.M. Vision-based medical expert system / O.M. Berezsky, K.M. Berezska, Y.M. Batko, G.M. Melnyk // 6th International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT'2011, Lviv, Ukraine, November 16-19) - 2011 – P. 49-50.
27. Bieri M. Quantitative analysis of Alzheimer plaques in mice using virtual microscopy / M. Bieri, A. Wethmar, N. Wey //First European Workshop on Tissue Imaging and Analysis, - 2009 – P. 31-38.
28. Muralidhar, G. Snakules: A model-based active contour algorithm for the annotation of spicules on mammography./ Muralidhar G., Bovik A., Giese J. // IEEE Transactionson Medical Imaging. – 2010 - V. 29. - N. 10. - P.1768-1780.



29. Berezsky, O.M. Texture segmentation of biomedical images based on spatial moments/ O.M. Berezsky, Y.M. Batko, G.M. Melnyk //Proceedings of the 4th International Scientific-Technical Conference "Computer Science and Information Technology 2009", (October 15-17, 2009, Lviv, Ukraine) - Lviv, Ukraine: "Tower and Co", 2009. - P. 42-45.
30. Berezsky, O. Segmentation of Cytological and Histological Images of Breast Cancer Cells / O. Berezsky, Yu. Batko, G. Melnyk, S. Verbovy, L. Haida // Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015), 24-26 September 2015, Warsaw, Poland. – Warsaw, 2015. – V.1. – P. 287-292.
31. Jipkate, B.R. A Comparative Analysis of Fuzzy C-means Clustering and K-means Clustering Algorithms. / B.R. Jipkate, V.V. Gohokar // International Journal of computational Engineering Research - 2012 - V.2.
32. Thilagamani, S. A Survey on image segmentation through clustering / S. Thilagamani, N. Shanthi // International journal of research and Information sciences, March - 2011. - V. 1 - 1.
33. Saini, K. A Study Analysis on the Different Image Segmentation Techniques. /Saini K., Arora S. //International Journal of Information & Computation Technology. - International Research Publications House. – 2014. - V. 4. - N. 14. - P. 1445-1452.
34. Abirami, S. Analysis of Image Segmentation Techniques for Medical Images. / Abirami, T. Sheel, // Proceedings of International Conference on Emerging Research in Computing, Information,Communication and Applications (ERCICA-14). - 2014.
35. Li, C. A Level Set Method for Image Segmentation in the Presence of Intensity Inhomogeneities With Application to MRI. / C. Li, R. Huang, Z.H. Ding, J.C. Gatenby, D.N. Metaxas, J.C. Gore // IEEE Transactions On Image Processing, - 2011. - V. 20. - N. 7. - P.2007-2016.

36. Arbelaez P. Contour detection and hierarchical image segmentation. / P. Arbelaez, M. Maire, C.Fowlkes, J.Malik // IEEE TPAMI - 2011. - V. 33. - N. 5.
37. Berezsky O.N. Quantitative evaluation of the quality of the segmentation of images based on metrics. / Berezsky O.N., Berezskaya E.N. // Upravlyayushchie Sistemy i Mashiny. - 2015. - V. 6. - P.59-65.
38. Joshi, V.S., Shire A.N. A Review of an Enhanced Algorithm for Color Image Segmentation. International Journal of Advanced Research in Computer Science and Software Engineering. 2013. V. 3. N. 3. P.435-441.
39. Berezsky O., Pitsun O. Automated Processing of Cytological and Histological Images. Proceedings of the XIIth International Conference “Perspective Technologies and Methods in MEMS Design”, MEMSTECH’2016, Lviv-Polyana. 2016. P. 51-53.
40. Haddadnia J., Ahmadi M., Faez K. An efficient feature extraction method with pseudo-zernike moment in RBF neural network-based human face recognition system. EURASIP Journal on Applied Signal Processing. 2003.p. 890-901.
41. Haralick R. M., Shanmugam K.7 Dienststein I. Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics. 1973. November. Vol. 3, no. 6. p. 610-621.
42. Hateren J. H. V., der Schaaf A. V. Independent component filters of natural images compared with simple cells in visual cortex. Transactions of Royal Society of London. 1998. Vol. B265. p. 359-366.
43. Heller K. A., Ghahramani Z. simple bayesian framework for content-based image retrieval. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006). 2006. p. 2110-2117.
44. Hew P. Geometric and zernike moments. Diary, Department of Mathematics, The University of Western Australia. 1996. October. URL: <http://citeseer.ist.psu.edu/hew96geometric.html>.
45. Hoqve S. K., Sirlantzis M. C. F. A new chain-code quantization approach enabling high performance handwriting recognition based on multi- classifier schemes . Seventh International Conference on Document Analysis and

Recognition (ICDAR'03). Vol. 2. 2003. 843 p.

46. Howarth P., Riiger S. Evaluation of texture features for content-based image retrieval. Proceedings of CIVR'04. 2004. p. 326-334.

47. Howarth P., Riiger S. Robust texture features for still image retrieval. IEEE Proceedings of Vision, Image and Signal Processing. Vol. 152. 2005. p.868-874.

48. Efficient and effective querying by image content: Tech. rep. / С. Faloutsos, W. Equitz, M. Flickner et al.: IBM Research, 1993. p. 179-187

49. Березький О.М., Дубчак Л.О., Мельник Г.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня «Магістр». Спеціальність: 123 - Комп'ютерна інженерія. Магістерська програма - Комп'ютерна інженерія"/ Під ред. О.М. Березького. Тернопіль:ЗУНУ,2020.32 с.

50. Гураль І.В., Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія»/Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.