

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра комп'ютерної інженерії

ЛЯХОВСЬКА Вікторія Олегівна

**Алгоритми автоматичної фільтрації біомедичних
зображень/ Algorithms of biomedical images automatic
filtration**

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконала студентка групи КІзм-21
В. О. Ляховська

Науковий керівник:
к.т.н., О. Й. Піцун

Кваліфікаційну роботу допущено
до захисту:

"__" _____ 20__ р.

Завідувач кафедри

_____ О. М. Березький

ТЕРНОПІЛЬ – 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему “Алгоритми автоматичної фільтрації біомедичних зображень” зі спеціальності 123 «Комп’ютерна інженерія» освітнього ступеня «магістр» написана обсягом 82 сторінки і містить 48 ілюстрацій, 7 таблиць, 3 додатки та 56 джерел за переліком посилань.

Метою роботи є розроблення адаптивного алгоритму фільтрації біомедичних зображень на низькому рівні комп’ютерного зору для опрацювання біомедичних зображень для підвищення якості сегментації та класифікації біомедичних зображень.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано: методи: комп’ютерного зору для попереднього оброблення зображень; об’єктно-орієнтованого програмування (для проектування програмних засобів аналізу зображень).

Результати дослідження: алгоритми автоматичної фільтрації зображень на основі показників вхідного зображення, програмний модуль фільтрації біомедичних зображень.

Результати роботи можуть бути використані в навчальному процесі під час виконання лабораторних, курсових робіт.

Орієнтовні напрямки розвитку досліджень: розроблення спеціалізованих систем автоматизованої мікроскопії для цитолога та патогістолога; розширення інформаційно-аналітичних систем, зокрема системи автмоатизованої мікроскопії «HIAMS».

КЛЮЧОВІ СЛОВА: ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ, БІОМЕДИЧНІ ЗОБРАЖЕННЯ, ПРОГРАМНИЙ МОДУЛЬ.

RESUME

The qualification work on "Algorithms for automatic filtering of biomedical images" in the specialty 123 "Computer Engineering" of the master's degree is written in 82 pages and contains 48 illustrations, 7 tables, 3 appendices and 56 sources according to the list of references.

The aim of the work is to develop an adaptive algorithm for filtering biomedical images at a low level of computer vision for processing biomedical images to improve the quality of segmentation and classification of biomedical images.

Research methods. To solve the tasks in the qualification work used: methods: computer vision for image pre-processing; object-oriented programming (for designing software for image analysis).

Research results: algorithm of automatic image filtering based on input image indicators, software module of biomedical image filtering.

The results of the work can be used in the educational process during laboratory and course work

Approximate directions of research development: development of specialized systems of automated microscopy for cytologist and pathohistologist; expansion of information-analytical systems, in particular the system of automated microscopy "HIAMS".

KEY WORDS: IMAGE FILTRATION, BIOMEDICAL IMAGES, SOFTWARE MODULE.

ЗМІСТ

Вступ.....	8
1 Аналіз біомедичних зображень	11
1.1 Класифікація біомедичних зображень	11
1.2 Алгоритми опрацювання зображень на низькому рівні комп'ютерного зору.....	16
1.3 Аналіз програмних засобів фільтрації зображень	28
1.4 Аналіз завдання магістерської роботи та постановка задач дослідження .	30
1.5 Висновки до розділу.....	31
2 МЕТОДИ І АЛГОРИТМИ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ	32
2.1 Аналіз алгоритмів фільтрації біомедичних зображень	32
2.2 Адаптивний алгоритм фільтрації	38
2.3 Нечітка база правил.....	43
2.4 Висновки до розділу.....	51
3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ АВТОМАТИЧНОЇ ФІЛЬТРАЦІЇ	52
3.1 Структура програмного модулю	52
3.2 Програмні методи реалізації алгоритму	57
3.3 Порівняльний аналіз експериментів	62
3.4 Висновки до розділу.....	71
Висновки.....	72
Список використаних джерел.....	73
Додаток А	Ошибка! Закладка не определена.
Додаток В Світлокопія виданої публікації ..	Ошибка! Закладка не определена.

ВСТУП

Із розвитком інформаційних технологій активно впроваджується їх використання в усіх сферах людського життя і зокрема в медицині. Лікарі активно використовують системи автоматизованої мікроскопії, системи підтримки прийняття рішень та системи обліку пацієнтів. Використання даних систем наразі не може замінити лікаря, адже це складна і відповідальна робота, що потребує наявності вузькоспеціалізованих знань та навиків. Однак, розробка спеціалізованих систем дозволить значно полегшити життя і роботу лікарів-діагностів.

Часто, при діагностуванні того чи іншого захворювання, лікарі акцентують увагу на результатах мікроскопічного дослідження зрізів тканин. Наприклад, при діагностуванні ракових захворювань здійснюють цитологічне, гістологічне та імуногістохімічне дослідження. Результати досліджень зберігають у вигляді зображень на компютері.

Лікарі – діагности використовують мікроскоп, фото- або відео-камеру для зберігання зображень у цифровому форматі. Під час формування зображення та його відправлення на компютер можуть формуватися додаткові шуми. В подальшому це негативно впливає на якість опрацювання даного зображення. Додатковим фактором, який ускладнює процес обробки зображень є те, що частини зображення (ядра клітин) можуть дотикатися один до одного, що у свою чергу не дозволяє чітко відділити ядра клітин для їх подальшого аналізу.

Етап попередньої обробки до якого входить і процес фільтрації зображень і одним із базовим. При якосному проходженні даного етапу можна значно покращити якість сегментації та класифікації зображень. Часто, при аналізі такого роду зображень необхідно виділити ядра клітин, щоби надалі після етапу сегментації можна було виділити їх кількісні характеристики, наприклад, площу, периметр, діаметр Ферета, окружність тощо. За допомогою цих показників можна класифікувати досліджувані ядра клітин для подальшої постановки

діагнозу. Наявність шумів на зображенні може спотворити результати, що зробить подальші дослідження недоцільними.

Сучасні системи автоматизованої мікроскопії характеризуються наявністю великої кількості алгоритмів для фільтрації зображень, однак дані системи не володіють алгоритмами, що можуть у автоматичному режимі покращити якість зображення. Робота з такими системами вимагає наявності спеціалізованих знань у лікарів, що є незручно.

Складність розробки адаптивного алгоритму опрацювання зображень характеризується ще й тим, що на даному етапі важко отримати кількісні вхідні дані для можливості формування правил. Тому, не зважаючи на велику кількість алгоритмів фільтрації, розробка автоматичного алгоритму є складним і актуальним завданням, яке в подальшому дозволить спростити роботу лікарів – діагностів.

Метою роботи є розроблення адаптивного алгоритму фільтрації біомедичних зображень на низькому рівні комп'ютерного зору для опрацювання біомедичних зображень для підвищення якості сегментації та класифікації біомедичних зображень.

Для розв'язання поставлених задач у кваліфікаційній роботі використано: методи комп'ютерного зору для попереднього оброблення зображень; об'єктно-орієнтованого програмування (для проектування програмних засобів аналізу зображень).

Орієнтовні напрямки розвитку досліджень: розроблення спеціалізованих систем автоматизованої мікроскопії для цитолога та патогістолога; розширення інформаційно-аналітичних систем шляхом створення розподілених баз даних для зберігання інформації про пацієнтів з можливістю інтернет доступу, створення засобів телемедицини.

Кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків. У першому розділі було проаналізовано типи біомедичних зображень, виділено їхні основні переваги та недоліки. Проведений порівняльний аналіз показав необхідність у застосуванні алгоритмів фільтрації та попереднього оброблення. Також проведено аналіз сучасних програмних

засобів, які використовують етап фільтрації зображень, що дозволило встановити необхідність розробки адаптивного алгоритму опрацювання зображень.

У другому розділі було розроблено адаптивний алгоритм фільтрації біомедичних зображень на основі показників вхідного зображення та запропоновану нечітку систему підбору алгоритмів фільтрації та їх параметрів.

У третьому розділі програмно реалізовано запропонований алгоритм, наведено структуру програмного модулю та проведено порівняльний аналіз результатів роботи розробленого алгоритму із аналогами.

1 АНАЛІЗ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ

1.1 Класифікація біомедичних зображень

У сучасній медицині медична візуалізація зазнала величезних успіхів і може отримати кілька біомедичних зображень пацієнтів. Після цього, щоб допомогти медичним фахівцям, ці зображення можна використовувати та навчати в інтелектуальній системі, щоб допомогти визначити різні захворювання, які можна ідентифікувати за допомогою аналізу цих зображень. Важливу роль у цьому відношенні відіграє класифікація; вона покращує групування цих зображень за категоріями захворювань і оптимізує наступний крок комп'ютерної системи діагностики.

Медичні зображення прийнято класифікувати за способом їх отримання та галузі, до якої вони належать. Виділяють наступні види зображень: анатомічні (фотографії, рентгенівські знімки, зображення УЗД і т.д. Крім того, останнім часом з'явилися області діагностики, які також можна віднести до цього списку, наприклад, іридіодіагностика, каріотипування. Легкість отримання та досить висока якість аналізу анатомічних зображень є передумовою існування великої кількості робіт з цієї тематики[1].

У той самий час коло робіт з гістологічним зображенням досить вузьке. Описується лише невеликий ряд щодо нових методів дослідження просторової організації клітинних структур: тривимірна реконструкція за знімками біологічних структур, що мають певні види симетрії, голографічний синтез об'ємних медичних та біологічних об'єктів за їх ракурсними знімками, отриманими в режимі вторинних електронів, моделювання гіпотетичної молекулярної структури об'єкта шляхом розрахунку плоского зображення та зіставлення з істинним зображенням, машинне дослідження геометричних та оптичних характеристик клітинних структур, що спостерігаються в оптичний мікроскоп, відновлення первинної структури біополімерів за набором їх фрагментів і т.д. Проекції та перерізи медичних та біологічних структур описуються двовимірними функціями оптичної щільності або традиційним

напівтоновим зображенням. Отже, у більшості методів дослідження просторової організації гістологічних структур одним із найважливіших етапів є сегментація гістологічних зображень на напівтонових зображеннях. Останнім часом з розвитком технічних можливостей світлова мікроскопія дозволяє отримувати кольорові зображення[2-4].

Класифікацію біомедичних зображень наведено у таблиці 1.1.

Таблиця 1.1 – Класифікація біомедичних зображень

Критерій класифікації		
За способом отримання зображення	За типом зображення	За розмірністю
<ul style="list-style-type: none"> - Цифрова радіологія -Комп'ютерна томографія - Ядерний магнітний резонанс - Ультразвук - Мікроскопія 	<ul style="list-style-type: none"> - RGB зображення (мікроскопія, еластографія, термографія) - Зображення в градаціях сірого (рентгенографія, томографія) - Чорно – біле (ультразвукове дослідження) 	<ul style="list-style-type: none"> - 2D (всі зображення незалежно від способу отримання) - 3D (послідовність радіологічних зображень, томографічне зображення динамічного об'єкта;)

Рентгенівське зображення починається з того, що пучок електронів високої енергії врізається в металеву мішень, і виникає рентгенівське випромінювання. Фільтр біля джерела рентгенівського випромінювання блокує ці промені низької енергії, що означає, що тільки високоенергетичні промені проходять через пацієнта до аркуша фотоплівки. Точка проникнення залежить від інтенсивності, якості та довжини хвилі рентгенівських променів. Чим сильніший промінь рентгенівського випромінювання, тим більше матеріалу він зможе проникнути, і навпаки. Ці електромагнітні випромінювання або рентгенівські промені діють на

поглинання випромінювання низького рівня частинами тіла з більшою щільністю, в результаті чого не поглинене випромінювання потрапляє на фотографічну «пластину», утворюючи «негативне зображення». Це означає, що чим вище щільність матеріалу, тим яскравіше воно буде зображено на фотоплівці[5].

МРТ (магнітно-резонансна томографія) – це неінвазивний діагностичний тест, який робить детальні зображення м'яких тканин тіла. На відміну від рентгена чи КТ, зображення створюються за допомогою магнітного поля, радіохвиль та комп'ютера. МРТ-сканування працює за допомогою потужного магніту, радіохвиль і комп'ютера для створення детальних зображень. Коли тіло знаходиться в магнітному полі, ці атоми вирівнюються з полем. Радіохвиля «збиває» атоми і порушує їх полярність. Датчик визначає час, необхідний для повернення атомів до початкового вирівнювання. МРТ вимірює вміст води (або характеристики рідини) різних тканин, яка обробляється комп'ютером для створення чорно-білого зображення. Зображення дуже деталізоване і може показати навіть найменші відхилення. МРТ можна використовувати для огляду артерій і вен. Стандартна МРТ не може побачити рідину, яка рухається, наприклад кров в артерії, і це створює «порожнечі потоку», які виглядають як чорні діри на зображенні[6-7].

Ультразвукове дослідження є одним з основних методів медичної візуалізації. В даному методі використовуються ультразвукові хвилі, які відбиваються від кордонів різних за щільністю середовищ. В основі методу лежить здатність ультразвукових хвиль проходити через різні тканини організму, по-різному відбиваючись від структур різної щільності. При збільшенні різниці між хвильовими опорами середовищ, що межують один з одним, амплітуда сигналу збільшується. Ці «відображення» фіксуються чутливим датчиком. Далі, після підсилення та перетворення на електричні сигнали, отримана інформація оцифровується за допомогою аналогово-цифрового пристрою і подається в комп'ютер. Зображення тканин обробляється за допомогою програмного забезпечення та на екран виводиться двовимірне зображення тканин, через які проходили ультразвукові хвилі. Позитронно-

емісійна томографія (ПЕТ) метод в якому Ін'єкційний радіоактивний індикатор виявляє хворі клітини. Комбінована ПЕТ-КТ дає тривимірні зображення для більш точної діагностики.

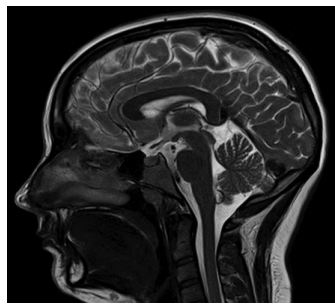
Ехокардіографія — процедура, яка представляє собою дослідження серця за допомогою ультразвуку. За допомогою цього методу діагностики можна виявляти функціональні аномалії органу і помічати зміни морфології. Він посилає, а потім вловлює ультразвукові сигнали, які відбиваючись від перешкод, що виникають у них на шляху, створюють зображення. Це дослідження має високу інформативність і дозволяє побачити навіть невеликі зміни в роботі серця.

Імуногістохімія є широко використовуваною методикою в патології для підтвердження експресії білка в зразках тканин. Однак ця техніка фарбування відома тим, що представляє міжсерійні варіації. Зображення цілого слайда при цифровій патології дає можливість подолати цю проблему за допомогою методів нормалізації зображення.

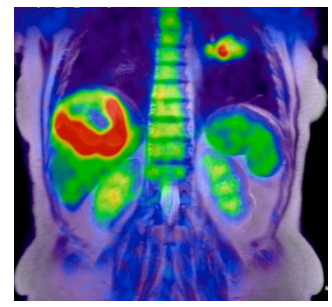
Гістологічні та цитологічні зображення – це RGB зображення, що формуються в результаті мікроскопічного дослідження. Гістологія - це важливий компонент діагностичної медицини та інструмент для вивчення патогенезу та біології таких процесів, як рак. Біоматеріалом для гістологічного дослідження можуть бути зразки різних тканин - шкіри, слизових оболонок, м'язів. Цитологічні зображення отримують на основі мазків. Даний клас зображень нагадує гістологічні зображення клітинного рівня. Основною його відмінністю є фон, який оточує клітини. Якщо в гістологічних зображеннях фон складається переважно з елементів тканини, то в цитологічних він не володіє явними поглинаючими світло властивостями. Його можна вважати оптично однорідним [8-10].



а) Рентгенографічне
зображення



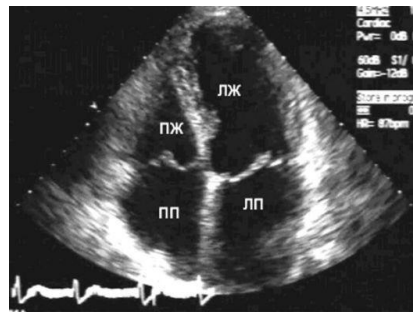
б) МРТ
зображення



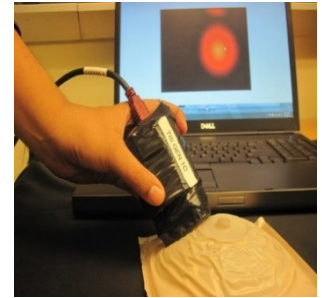
в) ПЕТ
зображення



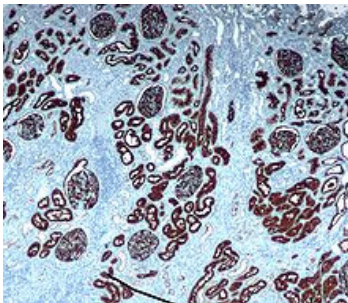
г) УЗД
зображення



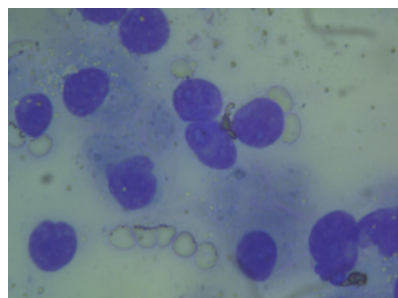
д) ехокардіографічне
зображення



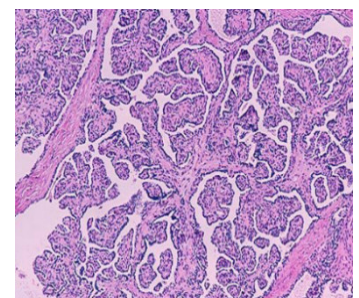
е) ТВ
зображення



є) імуногістохімічне
зображення



ж) цитологічне
зображення



з) гістологічне
зображення

Рисунок 1.1 – Приклади медичних зображень

Загальним недоліком даних зображень можна вважати нечіткість контурів досліджувальних об'єктів. Також такі типи зображень як УЗД, ехокардіографічне, цитологічне, гістологічне характеризується значним рівнем імпульсного шуму. Разом ці фактори негативно впливають на якість подальших етапів опрацювання зображень в системах автоматизованої мікроскопії [11].

1.2 Алгоритми опрацювання зображень на низькому рівні комп'ютерного зору

Протягом десятиліть вчені використовували комп'ютери для покращення та аналізу медичних зображень. Спочатку прості комп'ютерні алгоритми використовувалися для покращення зовнішнього вигляду цікавих елементів у зображеннях, допомагаючи людям краще їх читати та інтерпретувати. Пізніше були розроблені більш досконалі алгоритми, де комп'ютер не тільки покращував зображення, але й брав участь у розумінні їх змісту.

Опрацювання зображень – це метод покращення якості зображення після видалення невідповідних даних зображення із зображення в різних програмах і доменах. Медичні зображення містять багато невідповідних і небажаних частин у своєму фактичному форматі відсканованих зображень. Щоб видалити такі частини зображення, потрібні деякі методи попередньої обробки зображень для кращої візуалізації зображень перед виявленням захворювань, зокрема[12-14].

Пов'язані з цими етапами рівні опрацювання називаються відповідно: нижнім, середнім та високим рівнями відповідно. Класифікацію рівнів комп'ютерного зору наведено на рисунку 1.2.

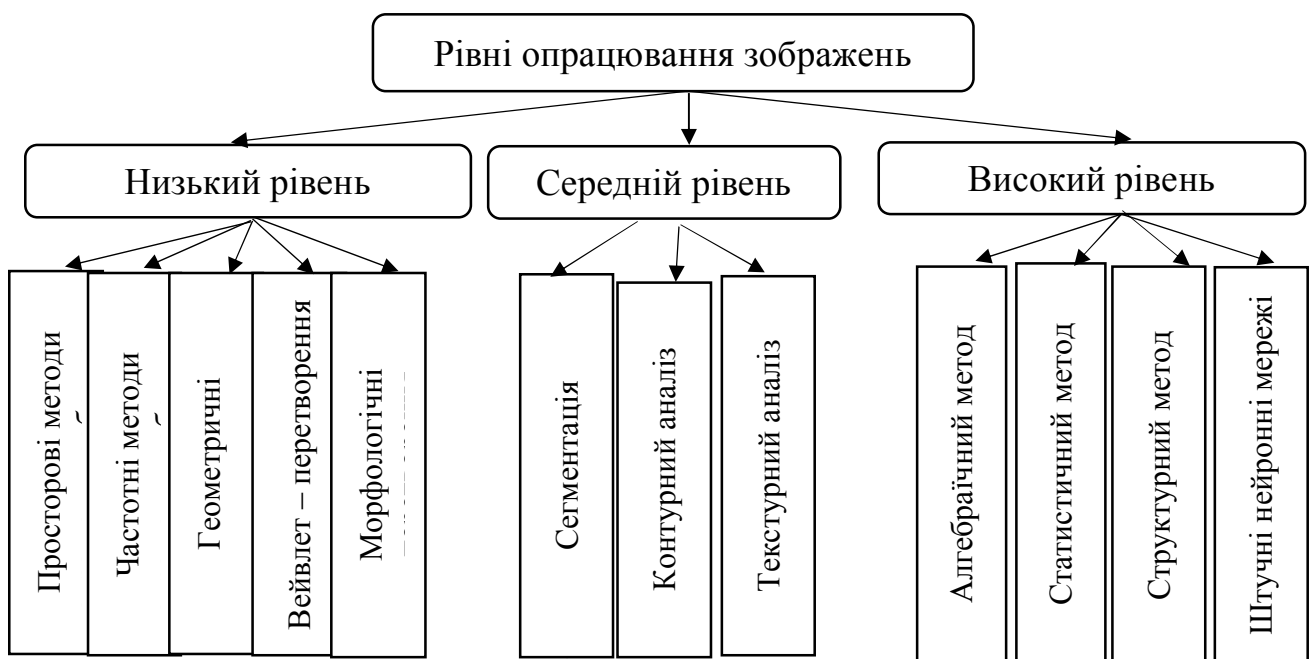


Рисунок 1.2 – Рівні опрацювання зображень

Розглянемо детальніше три рівні опрацювання зображень та їх алгоритми. Під низьким рівнем технічного зору розуміються такі процеси, які є простими з погляду здійснення автоматичних дій, які потребують штучного інтелекту. До низького рівня технічного зору належать методи зняття та попередньої обробки інформації. Таким чином, цей рівень охоплює процеси, починаючи безпосередньо від формування зображення та закінчуючи процесами компенсації, такими, як зменшення шуму, а також процесами виділення найпростіших параметрів зображення, такими, як розриви інтенсивності[15-17].

Методи обробки зображень низького рівня зорієнтовані в основному на апаратну реалізацію. Поліпшення якості зображення досягається шляхом застосування методів фільтрації, видалення шумів і підвищення контрастності. Процес підвищення інтенсивності пікселів і якості зображення також здійснюється після попередньої обробки. Після попередньої обробки якість зображення підвищується. Низькорівнева обробка зображень в основному пов'язана з вилученням описів із зображень (які зазвичай представлені як самі зображення). Аналіз зазвичай нічого не знає ні про те, які об'єкти насправді знаходяться на сцені, ні про те, де сцена знаходиться відносно спостерігача. Можуть бути численні, в основному незалежні описи, такі як фрагменти країв, плями, коефіцієнти відбиття, фрагменти ліній тощо.

Фільтрація – це операція, яка змінює:

- спостережувану якість зображення
- роздільну здатність
- контраст
- шум

Як правило, фільтрація передбачає застосування однакових або подібних математичних операцій до кожного пікселя зображення, наприклад, просторова фільтрація змінює інтенсивність кожного пікселя в зображенні, використовуючи деяку функцію сусідніх пікселів. Фільтрація є однією з найпростіших операцій обробки зображень[18-21].

Дуже простим прикладом просторового фільтра є середній фільтр. Замінює кожен піксель у зображенні середнім значенням околиці $N \times N$ навколо пікселя. На виході фільтра виходить зображення, яке виглядає більш «гладким» і менш «шумним», ніж вхідне зображення. Зображення згорнуте за допомогою середнього фільтра 7×7 наведено на рисунку 1.3.

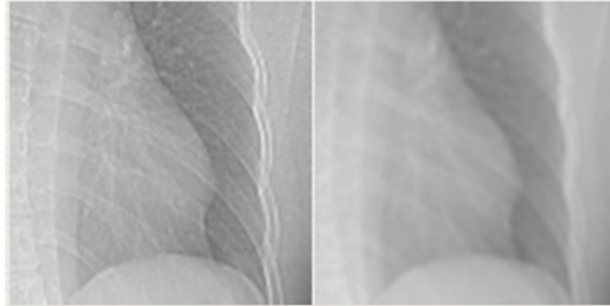


Рисунок 1.3 – Зображення згорнуте за допомогою середнього фільтра 7×7

Цифровий шум зображення – дефект зображення, що вноситься фотосенсорами та електронікою пристроїв, що їх використовують[22].

Для його придушення використовують такі наступні методи:

Класифікацію методів фільтрації зображень наведено на рисунку 1.4.

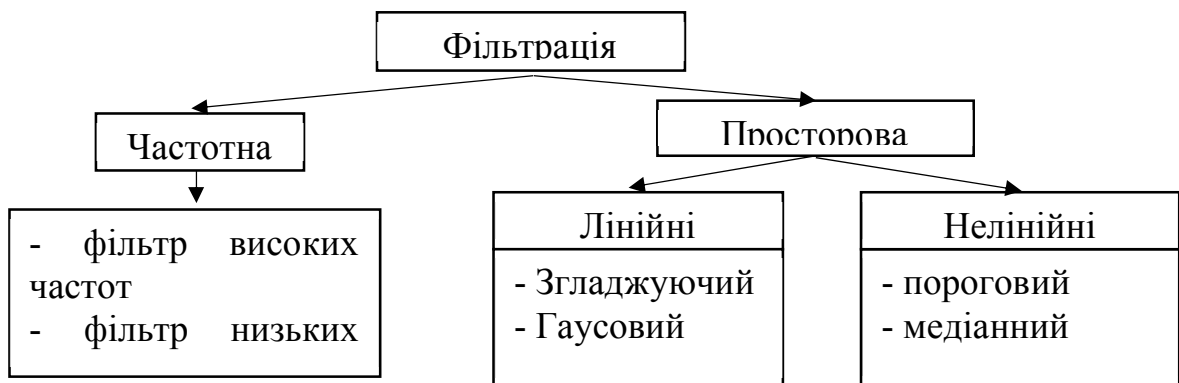


Рисунок 1.4 – Класифікація цифрових фільтрів

Лінійне усереднення точок сусідів – найпростіший вид алгоритмів видалення шуму. Основна ідея їх полягає в тому, щоб брати середнє арифметичне значення точок в деякій околиці як нове значення точки. Фізично

така фільтрація реалізується за допомогою обходу пікселів зображення матрицею згортки, що має такий вигляд як на рисунку 1.5.

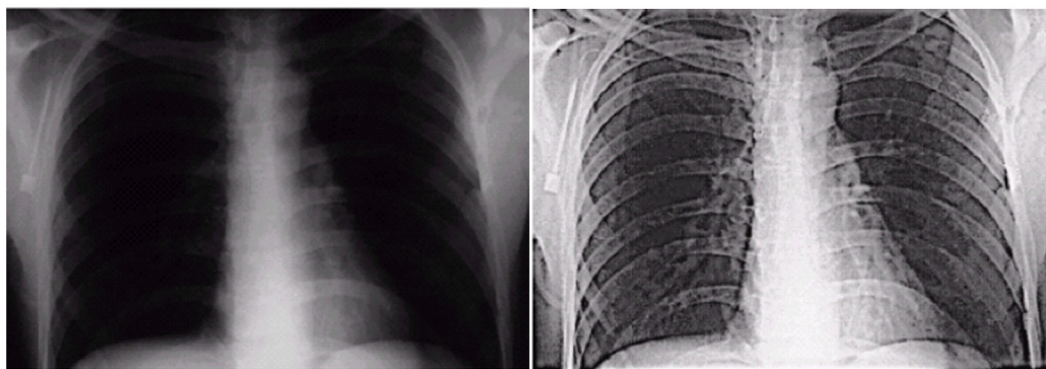


Рисунок 1.5 – Реалізація фільтрації матрицею розгортки

Зображення МРТ важко інтерпретувати. Тому ці зображення обробляються таким чином, щоб потім можна було використовувати їх для сегментації, що дає можливість лікарям визначити мікрокальцифікацію або утворення в легенях для подальшої діагностики. Попередня обробка включає видалення небажаних або невідповідних ділянок і виділення їх за рахунок збільшення контрасту. Запропоновано техніку підвищення роздільної здатності зображення та попередньої обробки для очищення або видалення шумів. Видалення з зображення частин, уражених шумом, без пошкодження країв зображень є завершальною частиною попередньої обробки[23].

Попередня обробка зображення. Число методів, придатних для попередньої обробки основних параметрів зображення, досить велике, вимогам по швидкості вирахувань та за експлуатаційною вартістю, що лежить в основі систем технічного зору, задовольняє лише певна підгрупа цих методів.

Існує два основних підходи до попередньої обробки інформації. Перший підхід заснований на методах просторової галузі, а другий - на методах частотної області з використанням перетворень Фур'є. Разом ці підходи охоплюють більшість з існуючих алгоритмів попередньої обробки інформації, що застосовуються в системах технічного зору роботів.

До просторової області відноситься сукупність пікселів. Методами просторової області є процедури, що оперують безпосередньо з пікселями, що

складають розглянуту просторову область. Функції попередньої обробки в просторовій області записуються у вигляді:

$$g(x, y) = H[f(x, y)],$$

де $f(x, y)$ - вхідне (вихідне) зображення, $g(x, y)$ - вихідне (оброблене) зображення, H - оператор функції f , визначений деякою просторовою областю, яка містить околицю точки (x, y) .

Основним підходом щодо околиці точки (x, y) є використання квадратної або прямокутної області - частини зображення з центром у точці (x, y) . Центр цієї області переміщається від пікселя до пікселю, починаючи наприклад, від лівого верхнього кута порядково, при цьому для отримання $g(x, y)$ оператор застосовується для кожного положення (x, y) . Квадратні форми просторової області більш переважні через простоту їх реалізацій. Один з найбільш поширених методів просторової області заснований на використанні так званих масок згортки (або шаблонів, вікон або фільтрів). Зазвичай маска є невеликою (наприклад, розмірністю 3 x 3) двовимірну систему, коефіцієнти якої вибирають таким чином, щоб виявити задану властивість зображення [24-25].

За допомогою маски, показаної на рисунку 1.6 можна виявити на зображенні з повільно змінюваною в просторі інтенсивністю окремо віддалені один від одного пікселі які значно відрізняється від фону інтенсивністю, що притаманно екстремального шуму.

	y_{i-1}	y_i	y_{i+1}
x_{i-1}	-1	-1	-1
x_i	-1	+8	-1
x_{i+1}	-1	-1	-1

Рисунок 1.6 – Маска для виявлення окремих точок, які відрізняються від постійного фону.

Потім інтенсивності пікселів просторової області підсумовуються із заданими ваговими коефіцієнтами. Шляхом порівняння отриманої суми з деяким

граничним значенням можна визначити світліший або темніший піксель області щодо фону. Цей оператор суми, по суті, масковий оператор, можна використовувати для зниження шуму на зображенні, отримання змінних порогів зображення для формування структури об'єкта або підрахунку параметрів зображення.

Гістограмне вирівнювання. Нехай змінна r позначає інтенсивність пікселів коригованого зображення. Спочатку припустимо, що r - нормалізована безперервна змінна, що знаходиться в діапазоні $0 \leq r \leq 1$. Для будь-якого r на інтервалі $[0, 1]$ виконується перетворення виду:

$$s = T(r),$$

яке визначає результуюче значення інтенсивності s для попереднього значення інтенсивності r кожного пікселя у вихідному зображенні. Це передбачає, що функція перетворення T задовольняє умовам:

1. $T(r)$ – однозначна та монотонно зростаюча функція на інтервалі $0 \leq T(r) \leq 1$.
2. $0 \leq T(r) \leq 1$ для $0 \leq r \leq 1$.

Перша умова виключає перехід з чорною шкалою інтенсивності на білу, а друга умова гарантує знаходження інтенсивності зображення в діапазоні від 0 до 1 для всіх значень пікселів. Зображена на рисунку 1.7 функція у разі застосування до вихідного зображення здійснює його «компресію», тобто зміщує сіре зображення, до якого найчастіше відноситься об'єкт, що цікавить, в область найбільшого дозволу по рівню інтенсивностей [26].

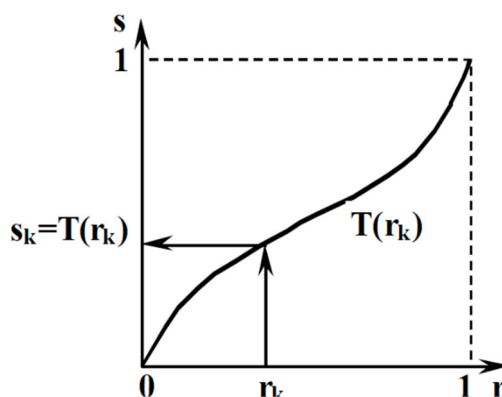


Рисунок 1.7 – Функція перетворення інтенсивності

Інтенсивності r і s розподілені на інтервалі $[0, 1]$ випадковим чином і, отже, можуть характеризуватись за допомогою відповідних функцій щільності розподілу ймовірності $p(r)r$ та $p(s)s$. Саме від характеру поведінки відповідної функції щільності розподілу вірогідності на інтервалі $[0, 1]$ залежить основна тональність зображення. Якщо більшість пікселів зображення мають інтенсивність близьку до нуля, то має місце зображення з темною тональністю. Якщо інтенсивності більшої кількості пікселів мають значення, близькі до одиниці, то тональність всього зображення світла. Функції щільності розподілення ймовірності для описаних випадків наведено на рисунку 1.8.

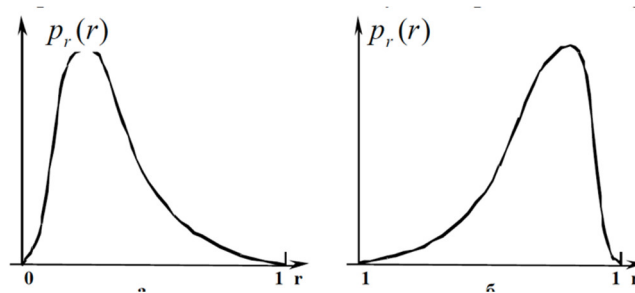


Рисунок 1.8 – Функції щільності розподілення ймовірності темного й світлого зображень

Основний ефект розглянутого перетворення полягають у вирівнюванні розподілу інтенсивностей пікселів зображення, отриманого системою технічного зору промислового робота. Це суттєво впливає на якість зображення і дозволяє позбавитися таких дефектів зображення, як постійне затемнення або, навпаки, засвічування зображення. Дане перетворення ідеально відповідає автоматичній корекції [27].

Обробка зображень середнього рівня. Середній рівень обробки зображень в основному стосується вилучення описів сцени з описів зображень, витягнутих на низькому рівні. Вихід зазвичай у більш символічній формі, що описує положення та форму частин сцени. Аналіз зазвичай нічого не знає про те, які об'єкти знаходяться в сцені, але використовує багато знань про форму сцени і те, як форма виглядає на зображенні. Обробка середнього рівня включає такі

завдання, як сегментація зображень, опис зображень, контурний аналіз, розпізнавання об'єктів тощо.

Сегментацією називається процес підрозділу сцени на складові частини чи об'єкти. Сегментація є одним з основних елементів роботи автоматизованої системи технічного зору, оскільки саме на цій стадії обробки об'єкти виділяються зі сцени для подальшого розпізнавання та аналізу. Алгоритми сегментації, як правило, ґрунтуються на двох фундаментальних принципах: розривності та подоби. У першому у разі основний підхід ґрунтується на визначенні контурів, а в другому – на визначенні порогового рівня та розширення області. Ці поняття застосовні як до статичних, і до динамічних сценам. Таким чином, формально процес сегментації - це процес виділення на зображенні об'єктів, що цікавлять. Виділення об'єктів може бути реалізовано багатьма способами в залежності від того, з якою метою це робиться. Так, наприклад, для розпізнавання об'єкта з ряду запропонованих можна скористатися лише контуром даного об'єкта, а можна провести порівняння за декількома пунктами, для чого необхідно мати повне зображення об'єкта[28-29].

В ідеальному випадку методи СТЗ низького рівня визначають пікселі, що лежать на межі між об'єктом та тлом. На практиці цей ряд пікселів рідко повністю характеризує кордон через шум, розривів на кордоні внаслідок нерівномірної освітленості та інших ефектів, що призводять до розмиття зображення. Таким чином, алгоритми виявлення контурів супроводжуються процедурами побудови меж об'єктів із відповідних послідовностей пікселів. Розрізняють локальний та глобальний аналіз зображення на предмет побудови безперервної межі зображення.

Локальний аналіз є одним із найпростіших підходів з'єднання точок контуру є аналіз характеристик пікселів невеликий в невеликій околиці (наприклад, в околиці розміром 3x3 або 5x5) кожної точки (x, y) образу, який вже зазнав процедури виявлення контуру. Усі точки, що є подібними (критерій подібності визначається нижче), з'єднуються, утворюючи кордон з пікселів, що володіють деякими загальними властивостями, що дозволили виділити необхідний критерій подоби. Дана процедура розширення області починає

роботу із заданої множини вузлових точок. Можна спочатку розбити образ на ряд довільних непересічних областей і потім об'єднувати та/або розбивати ці області з метою задоволення умов, сформульованих на початку розділу. Один із способів сегментації вихідного зображення полягає у розбиття площі образу на дедалі менші квадратні області. Потім кожен із квадратів піддається перевірці на виконання предикату, який відповідає за приєднання даного квадрата до образу. Як такий предикат може виступати величина, показує нерозривне заповнення квадрата, що перевіряється пікселями заданої інтенсивності [30-31].

Графічне представлення методів сегментації наведено на рисунку 1.9.



Рисунок 1.9 – Методи сегментації зображень

Контурний аналіз. Багато виготовлених об'єктів можна моделювати за допомогою жорстких примітивів, як контурів, поверхонь або об'ємних примітивів. Однак таке уявлення може не підійти для більш природних об'єктів, наприклад бананів. Кожен банан має схожу, впізнавану форму, але немає двох абсолютно однакових бананів. У медичній візуалізації подібна ситуація існує для частин тіла. Навіть у виробничій промисловості друкована плата може бути завантажена резисторами, які мають не зовсім однаковий об'ємний або контурний вигляд. В якості іншого прикладу розглянемо моделювання об'єкта, форма якого змінюється з часом, наприклад, відстеження губ або очей на

зображенні обличчя. Хоча губи зберігають характерну форму, вони не є жорсткими і не описують ту саму форму ні в 2D, ні в 3D просторі.

Активні або деформовані моделі використовуються для представлення або класу об'єктів різної форми, або об'єктів, які змінюють форму. Деформованими моделями можуть бути тривимірні поверхні, наприклад повітряна куля, яку ми видавлюємо з форми, тривимірні просторові криві, як трубоочисник, який ми згинаємо, щоб утворити фігури, або 2D-контури, як креслярський сплайн, який він/вона використовує для малювання профілів. Тут ми розглянемо лише одне таке гнучке уявлення, Змію [32].

Змія є конкретним прикладом загальної деформованої моделі, яку можна зіставити з контуром зображення шляхом мінімізації енергії. Це сплайн, що мінімізує енергію; з заданої вихідної точки вона сама деформується, щоб відповідати найближчому виступаючому контуру. Вони не «знаходять» контури; початкове розташування має бути забезпечене або іншою обробкою, або знаннями вищого рівня.

У процесі середнього рівня вхідні дані, як правило, є зображеннями, але його вихідні дані, як правило, є атрибутами зображення.

Під високим рівнем технічного зору розуміються процеси, що відносяться безпосередньо до штучного інтелекту. Методом високого рівня можна вважати інтерпретацію - припущення про деталі визначення і ситуаціях, неповністю ідентифікуються на попередніх етапах. Труднощі, які можуть виникнути при обробці зображень на середньому рівні, полягають у частковому загородженні одних предметів іншими, наявності тіней і відблисків на зображенні, що виявляються в поганих умовах освітлення.

Обробка високого рівня передбачає отримання сенсу з групи розпізнаних об'єктів. Цей процес зазвичай пов'язаний з комп'ютерним зором.

Алгоритми регресії. Проблема регресії – це проблема керованого навчання, яка потребує моделі, яка потребує оцінки числа. Найпростіший і найшвидший алгоритм - це лінійна регресія (принаймні квадратна), але на цьому не варто зупинятися, тому що він часто дає поганий результат. Іншими поширеними алгоритмами регресії для машинного навчання (за винятком нейронних мереж)

є простий байєсовський алгоритм, дерево рішень, сусіди K-Close, LVQ (дослідження векторної величини), LARS Lasso, Elastic Net, Random Forest, AdaBoost та XGBoost. Ви помітите, що алгоритми машинного навчання суперечать один одному між регресією та класифікацією.

Алгоритми класифікації. Проблема класифікації, як правило, є проблемою керованого навчання, яка потребує ймовірності для кожного класу, вибираючи між двома чи більше класами. За винятком нейронних мереж та поглибленого вивчення, які вимагають набагато більш високого рівня обчислювальних ресурсів, найпоширенішими є алгоритми: наївний алгоритм байєсу, дерево рішень, логістична регресія, сусіди K-Close та еталонна векторна машина (SVM). Ви також можете використовувати методи ансамблю (приклади комбінацій), такі як випадковий ліс, інші методи завантаження та методи покращення, такі як AdaBoost та XGBoost.

Алгоритми кластеризації. Кластеризація - це неконтрольована проблема навчання, яка потребує моделі для пошуку груп у схожих точках даних. Найпопулярніший алгоритм – K-Means Clustering; серед інших - кластеризація в середній частині, DBSCAN (просторова кластеризація додатків, заснованих на щільності шуму), GMM (змішані моделі Гауса) і HAC (кластер ієрархічної агломерації).

Навчальні нейронні мережі. Управління нейронною мережею здійснюється так само, як і навчання інших машин: ви надаєте мережі набір навчальних даних, порівнюєте вихідний сигнал мережі з бажаним вихідним сигналом, створюєте вектор помилок і вносите виправлення до мережі. вектор помилок, які ви вводите. Об'єднані набори освітніх даних до внесення змін називаються періодами. Тепер, коли ми знаємо нейрони, нам потрібно знати про загальні топології нейронних мереж. У мережі з прямим підключенням нейрони поділяються на окремі рівні: вхідний рівень, n рівнів прихованої обробки та вихідний рівень. Вихід кожного шару переміщається лише наступний етап. У мережі з прямим з'єднанням для швидкого з'єднання деякі з'єднання можуть проходити через один або кілька проміжних рівнів. У нейронних мережах, що

повторюються, нейрони можуть впливати на себе прямо або опосередковано з наступної стадії[33-34].

1.3 Аналіз програмних засобів фільтрації зображень

Зараз існує великий вибір бібліотек і програмних пакетів для роботи з різноманітними типами зображень. Розглянемо найпопулярніші бібліотеки для роботи з графікою наведено на рисунку 1.10.

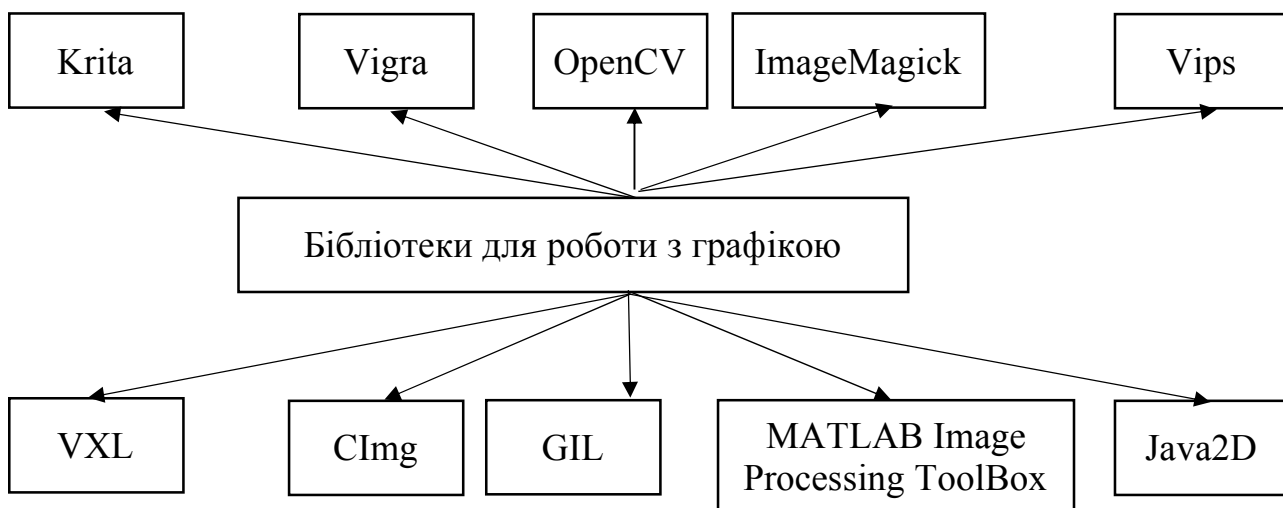


Рисунок 1.10 – Бібліотеки для роботи із зображеннями

CImg – бібліотека C++, що надає класи та функції обробки зображень. Це як елементарні функції (завантаження, збереження, перегляд), і алгоритми зміни розміру/обертання, застосування ефектів, малювання об'єктів (тексту, ліній, поверхонь, еліпсів, ...), тощо.

Java 2D API покращує графічні, текстові та графічні можливості Абстрактного набору інструментів для вікон (AWT), дозволяючи розробляти більш багаті користувацькі інтерфейси та нові типи програм Java. Java 2D API також дозволяє створювати вдосконалені графічні бібліотеки, такі як бібліотеки

CAD-CAM і бібліотеки графічних або графічних спеціальних ефектів, а також створення фільтрів читання/запису зображень і графічних файлів.

У поєднанні з Java Media Framework та іншими Java Media API, Java 2D API можна використовувати для створення та відображення анімацій та інших мультимедійних презентацій. API Java Animation і Java Media Framework покладаються на Java 2D API для надання підтримки.

OpenCV — це бібліотека C++ з відкритим вихідним кодом для обробки зображень і комп'ютерного зору, спочатку розроблена Intel, пізніше підтримана Willow Garage, а зараз підтримується Itseez. Він безкоштовний як для комерційного, так і для некомерційного використання. Це бібліотека, в основному націлена на обробку в реальному часі. Тепер він має кілька сотень вбудованих функцій, які реалізують обробку зображень і алгоритми комп'ютерного зору, які роблять розробку передових програм комп'ютерного зору легкою та ефективною.

VIGRA — це бібліотека для обробки та аналізу зображень, яка робить основний акцент на настроюваних алгоритмах і структурах даних. VIGRA особливо сильна для багатовимірних зображень, оскільки багато алгоритмів (наприклад, фільтри, обчислення ознак, суперпікселі) реалізовані для доволно високих розмірів. Використовуючи методи шаблонів, подібні до стандартної бібліотеки шаблонів C++, ви можете легко адаптувати будь-який компонент VIGRA до потреб вашої програми, не втрачаючи при цьому швидкості виконання. Починаючи з версії 1.7.1, VIGRA також надає великі прив'язки Python на основі популярного фреймворку numpy.

Image Processing Toolbox надає повний набір стандартних алгоритмів і програм для обробки зображень, аналізу, візуалізації та розробки алгоритмів. Програми Image Processing Toolbox дозволяють автоматизувати звичайні робочі процеси обробки зображень. Є можливість в інтерактивному режимі сегментувати дані зображень, порівнювати методи реєстрації зображень та обробляти великі набори даних в пакеті. Функції та програми візуалізації дозволяють досліджувати зображення, налаштувати контрастність; створювати гістограми; і маніпулювати регіонами інтересів (ROI) [35-39].

Порівняльну характеристику бібліотек для роботи з зображеннями наведено у таблиці 1.2.

Більшість бібліотек володіють стандартним набором алгоритмів низького та середнього рівнів комп'ютерного зору.

Таблиця 1.2 – Порівняльна характеристика бібліотек опрацювання зображень.

Бібліотека	Переваги	Недоліки
OpenCV	перетворення зображень в інші базиси; наявність методів та алгоритмів усіх рівнів комп'ютерного зору; наявність інтерфейсів для мов програмування: Python, Java, MATLAB, C++	відсутність стабільної версії
Krita	оптимізована для інтерактивної роботи;	не оптимізована; проблеми з великими зображеннями.
Vigra	працює з великими зображеннями; зручна документація.	орієнтована здебільшого на дослідницьку роботу
VXL	працює з великими за розміром зображеннями;	давно не оновлялась; оптимізована лише для простих зображень в RGB форматі
CImg	проста у застосуванні;	не оптимізована; неструктурована документація.
Java2D	об'єктно-орієнтована архітектура; повнофункціональна.	із закритим кодом.

Перевагою таких інструментів як OpenCV та MATLAB Image Processing Toolbox є ширший набір алгоритмів у порівнянні з іншими та постійний супровід зі сторони розробників.

1.4 Аналіз завдання магістерської роботи та постановка задач дослідження

Метою роботи є розроблення адаптивного алгоритму фільтрації біомедичних зображень на низькому рівні комп'ютерного зору для опрацювання біомедичних зображень для підвищення якості сегментації та класифікації біомедичних зображень. Для розв'язання поставлених задач у кваліфікаційній роботі використано: методи: комп'ютерного зору для попереднього оброблення зображень; об'єктно-орієнтованого програмування (для проектування програмних засобів аналізу зображень).

Для реалізації поставленої мети потрібно виконати наступні завдання:

- проаналізувати особливості біомедичних зображень;
- проаналізувати алгоритми фільтрації зображень;
- розробити алгоритм автоматичної фільтрації біомедичних зображень;
- здійснити програмну реалізацію розробленого алгоритму;
- провести тестування розробленого алгоритму з аналогами.

1.5 Висновки до розділу

На основі аналітичного підходу проведено порівняльний аналіз біомедичних зображень, зокрема гістологічних, цитологічних, імуногістохімічних, що дало змогу виділити їх переваги та недоліки на етапі опрацювання на низькому рівні комп'ютерного зору.

Проведено аналіз сучасних алгоритмів фільтрації зображень, що дозволило виділити алгоритми фільтрації та їх модифікації для подальшої розробки адаптивного алгоритму фільтрації.

З допомогою порівняльного аналізу виділено сучасні програмні засоби опрацювання зображень, зокрема в системах автоматизованої мікроскопії, де досліджуються зображення, отримані в результаті мікроскопічного дослідження.

2 МЕТОДИ І АЛГОРИТМИ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

2.1 Аналіз алгоритмів фільтрації біомедичних зображень

Існує велика кількість алгоритмів фільтрації зображень. Серед цих алгоритмів можна виділити такі:

- Медіанний фільтр;
- Білатеральний фільтр;
- Адаптивний білатеральний фільтр;
- Гаусовий фільтр.

Медіанний фільтр є одним із найпоширеніших видів фільтрів. Даний фільтр є нелінійним. Роботу медіанного фільтру можна навести з допомогою формули (2.1).

$$Im_{ij} = med \left[Im_{i+s, j+t}; (s, t) \in Z^2 \right] \quad (2.1)$$

На першому етапі усі значення пікселів в околиці конкретного пікселя сортуються у певному порядку, зазвичай від меншого до більшого. На наступних етапах відбувається обчислення медіанного значення. Після цього отримане медіанне значення замінює піксель, вибраний на першому кроці. Візуально принцип роботи медіанного фільтру наведено на рисунку 2.1.

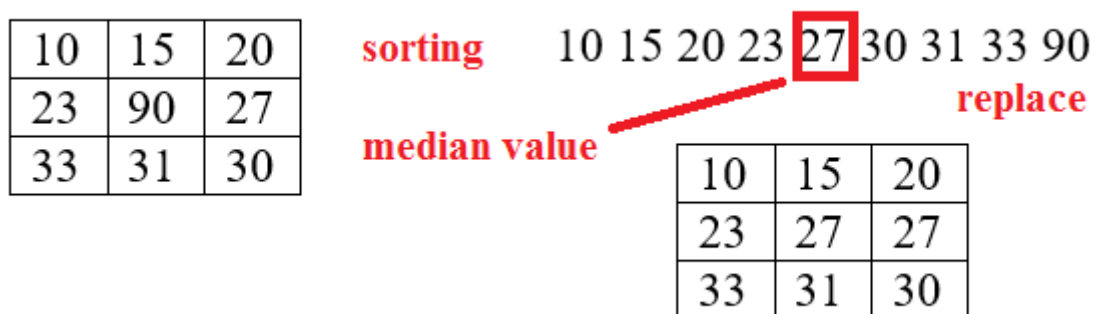


Рисунок 2.1 – Принцип роботи медіанного фільтру

Ідея середньої фільтрації полягає в тому, щоб просто замінити кожне значення пікселя в зображенні середнім ("середнім") значенням його сусідів, включаючи його самого. Це призводить до усунення значень пікселів, які не

відображають їх оточення. Середня фільтрація зазвичай розглядається як фільтр згортки. Як і інші згортки, він заснований навколо ядра, яке представляє форму та розмір околиці, яку потрібно відібрати під час обчислення середнього. Часто використовується квадратне ядро 3×3 , хоча більші ядра (наприклад, 5×5 квадратів) можна використовувати для більш серйозного згладжування. Мале ядро можна застосовувати більше одного разу, щоб отримати подібний, але не ідентичний ефект, як один прохід з великим ядром. Варіанти фільтра середнього згладжування, що обговорюються тут, включають усереднення порогового значення, при якому згладжування застосовується за умови, що значення центрального пікселя змінюється тільки в тому випадку, якщо різниця між його початковим значенням і середнім значенням перевищує попередньо встановлений поріг. Це призводить до того, що шум згладжується з меншою втратою деталей зображення. Результат роботи медіанного фільтра наведено на рисунку 2.2.

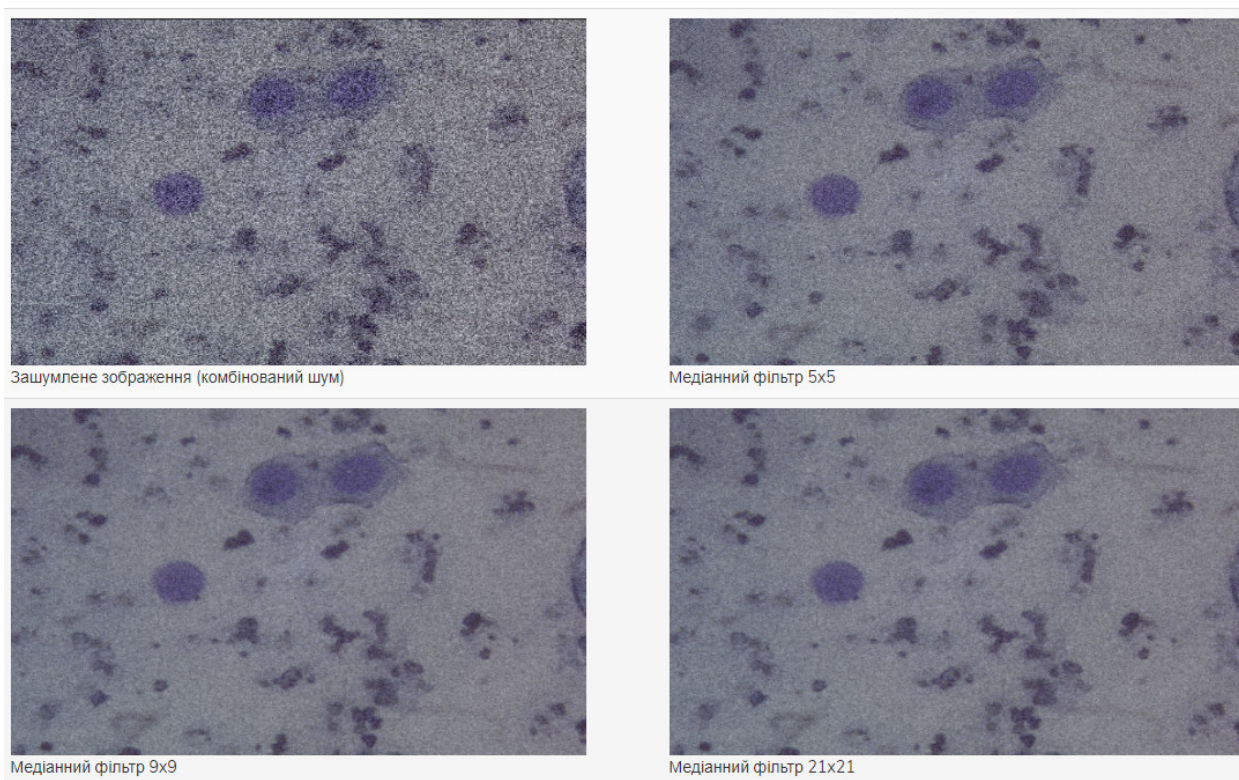


Рисунок 2.2 – Результати роботи медіанного фільтра з різним розміром вікна

Аналізуючи даний рисунок можна зробити висновок, що зі збільшенням розміру вікна зменшується рівень шуму. Однак, недоліком такого підходу є те, що втрачається якість зображення. Медіанна фільтрація — це нелінійна операція, яка часто використовується в обробці зображень, щоб зменшити шум «сіль і перець». Медіанна фільтрація є ефективнішою, ніж згортка, коли мета — одночасно зменшити шум і зберегти краї[40].

Приклад коду на мові програмування Java для реалізації медіанної фільтрації:

```
for(int i=1;i<img.getWidth()-1;i++)
    for(int j=1;j<img.getHeight()-1;j++)
    {
        pixel[0]=new Color(img.getRGB(i-1,j-1));
        pixel[1]=new Color(img.getRGB(i-1,j));
        pixel[2]=new Color(img.getRGB(i-1,j+1));
        pixel[3]=new Color(img.getRGB(i,j+1));
        pixel[4]=new Color(img.getRGB(i+1,j+1));
        pixel[5]=new Color(img.getRGB(i+1,j));
        pixel[6]=new Color(img.getRGB(i+1,j-1));
        pixel[7]=new Color(img.getRGB(i,j-1));
        pixel[8]=new Color(img.getRGB(i,j));
        for(int k=0;k<9;k++){
            R[k]=pixel[k].getRed();
            B[k]=pixel[k].getBlue();
            G[k]=pixel[k].getGreen();
        }
        Arrays.sort(R);
        Arrays.sort(G);
        Arrays.sort(B);
        img.setRGB(i,j,new Color(R[4],B[4],G[4]).getRGB());
    }
    ImageIO.write(img,"jpg",output);
```

Фільтр Гауса усереднює по закону Гауса пікселі довкола точки.

- Згортка з самим собою - фільтр Гауса

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

- Згортка з радіусом δ двічі надасть результат аналогічний згортці з радіусом $\delta \sqrt{2}$.

Ядро сепарабельне - розкладається в суму одномірних ядер.

Приклад згортки наведено на рисунку 2.3.

1	2	1
2	4	2
1	2	1

*

2	3	3
3	5	5
4	4	6

Рисунок 2.3 - Приклад згортки

Розкладання фільтру на 2 1D фільтри наведено на рисунку 2.4.

1	2	1
2	4	2
1	2	1

=

1
2
1

*

1	2	1
---	---	---

Рисунок 2.4 - Розкладання фільтру на 2 1D фільтри

Згортка по стрічках наведено на рисунку 2.5.

1	2	1
---	---	---

*

2	3	3
3	5	5
4	4	6

=

	11	
	18	
	18	

Рисунок 2.5 - Згортка по стрічках

Цей метод застосовує фільтр розмиття за Гауссом до зображення. Ефект націлений на пікселі в колі, визначеному радіусом, і використовує розподіл по Гауссу, щоб розмивати зображення від центру. Фільтр розмиття за Гауссом використовує такі властивості:

- Радіус
- Вхідне зображення.

Приклад коду для реалізації Гаусового фільтру:

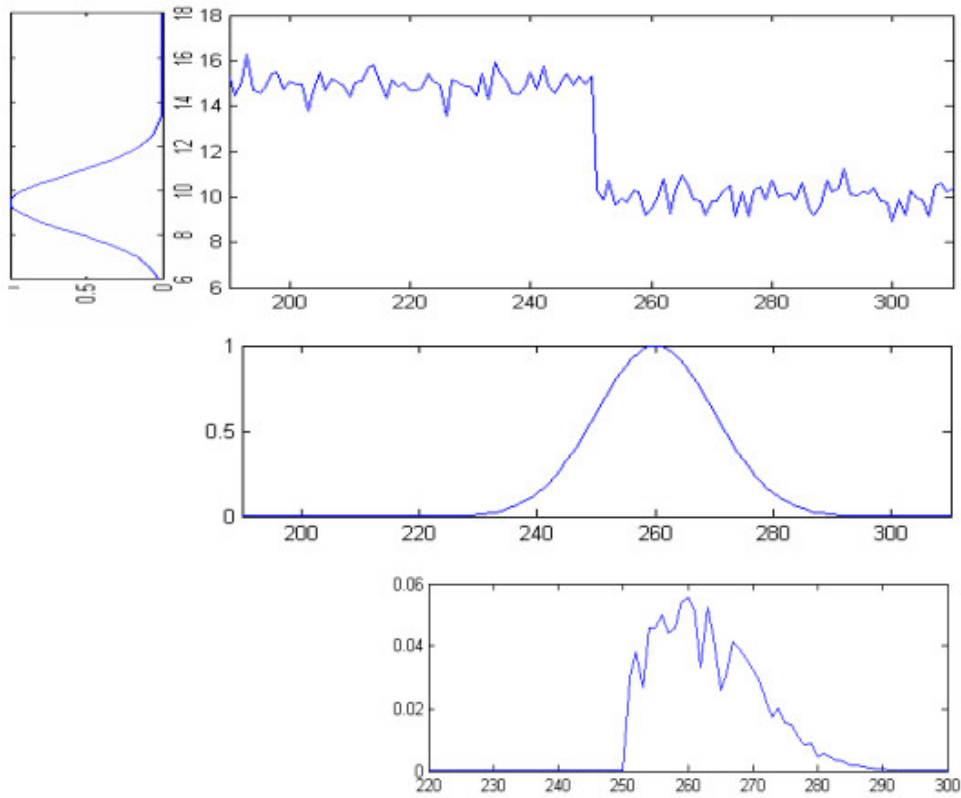


Рисунок 2.6 - Ілюстрацію 1D двостороннього фільтра

Інша найшвидша реалізація двостороннього фільтра, складність обчислення якої збігається до $O(\log n)$ (n – загальна кількість пікселів), була розроблена Вайссом з використанням ієрархії часткових розподілених гістограм з використанням підходу на основі рівнів. Хоча складність має було знижено, простота була втрачена через конкретні вимоги щодо розміру фільтра та оптимальної кількості гістограм. Цей метод обмежується прямокутними просторовими ядрами та коробочними фільтрами. Інша проблема – це недосконала частотна характеристика їх просторового фільтра.

З точки зору підвищення різкості зображення, USM залишається найпоширенішим інструментом для підвищення різкості, незважаючи на недоліки, які він має: по-перше, USM підвищує різкість зображення, додаючи перевищення та заниження до країв, що створює ореол навколо країв. По-друге, при застосуванні до зображення із зашумленням USM посилює шум у гладких областях, що значно погіршує якість зображення. Для вирішення першої проблеми запропоновано декілька алгоритмів відновлення схилів [5, 6]. Вони або змінюють профілі країв у нормальному напрямку до краю, або, щоб спростити

проблему, змінюють одновимірну горизонтальну/вертикальну проекцію профілів кромки. Ці одновимірні методи відновлення схилів, як правило, створюють артефакти на природних зображеннях, і їх ефективність недостатньо продемонстрована. Адаптивний двосторонній фільтр (ABF), який ми пропонуємо, також має на меті відновити нахил країв, але без необхідності знаходити нормалі краю або отримувати профілі країв. Таким чином, ABF ефективний у реалізації. Ми покажемо в розд. 4, що ABF створює чисті, чіткі краї без артефактів [42].

Фільтр Вінера (ФВ) є зворотним фільтром. Тому ФВ має низьку завадостійкість. PV використовується для відновлення розфокусованих зображень. Цей фільтр робить зображення чіткішим. Але це може додати (усунути) точки в порівнянні з вихідним зображенням. PV також використовується для зменшення випадкового шуму. При використанні методів фільтрації слід враховувати особливості медичних зображень. Серед цих особливостей – розмір об'єктів на зображенні (SOI). Тому ми також можемо розглянути метод фільтрації, який враховує геометрію об'єктів на зображенні.

2.2 Адаптивний алгоритм фільтрації

Сьогодні опрацювання біомедичних зображень є важливим напрямком застосування сучасної медичної техніки. Задачами опрацювання зображень є опис, аналіз та оброблення зображень. Проблеми аналізу біомедичних зображень включаючи класичну задачу розпізнавання фігур заданої форми, важлива також експертна оцінка, яка зараз є дорогою. Виникають проблеми, які зумовлені новими завданнями опису зображення та пошуком закономірностей або наборів закономірностей, що одночасно зустрічаються в багатьох наборах. Оскільки наборів може бути велика кількість необхідно здійснювати цей пошук

автоматично. Тому актуальною задачею є розробка власного алгоритму фільтрації біомедичних зображень.

У 3D-зображенні числове визначення зображення в термінах кількості пікселів зображення в горизонтальній і вертикальній площинах зображення. Образи допомагають сфокусувати уявлення особистості. Зображення легше розуміти та інтерпретувати на відміну від текстової інформації. Таким чином, через велику кількість застосувань цифрових зображень, необхідно, щоб деталі зображення були чітко описані і не містили будь-яких спотворення або артефактів. Роздільна здатність зображення – це найважливіша ознака зображення, яка зображує деталі, які містить зображення і в цьому контексті; Покращення роздільної здатності зображення є важливим підходом, який широко використовується для покращення роздільної здатності зображення та візуального входу зображення [43].

Опис алгоритму фільтрації біомедичних зображень.

- 1) Нехай \overline{lm} – вхідне зображення, представлено $g(1)$

$$\overline{lm} = \begin{bmatrix} a_{0,1} & a_{0,N-1} \\ a_{M-1} & a_{M-1,N-1} \end{bmatrix}$$

- 2) На даному рівні потрібно визначити рівень шуму на зображенні. Для цього спершу обчислимо середньоквадратичне відхилення.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |lm(i,j) - \overline{lm}(i,j)|^2$$

де \overline{lm}^l та lm – профільзоване та оригінальне зображення відповідно .

\overline{lm}^l -формується в результаті попередньої фільтрації зображення Гаусовим фільтром.

У статистиці середня квадратична помилка (MSE) або середнє квадратичне відхилення (MSD) оцінювача (процедури оцінки неспостережуваної величини) вимірює середнє квадратів помилок, тобто середню квадратичну різницю між розрахунковими і фактичними значеннями. MSE є функцією ризику, що відповідає очікуваному значенню квадрата втрати помилки. Той факт, що MSE майже завжди є строго додатним (а не нульовим), пояснюється випадковістю або тому, що оцінювач не враховує інформацію, яка могла б дати більш точну оцінку. MSE є мірою якості оцінювача. Оскільки воно виводиться з квадрата евклідової відстані, воно завжди є додатним значенням, яке зменшується, коли похибка наближається до нуля.

MSE — це другий момент (щодо походження) помилки, і, таким чином, включає в себе як дисперсію оцінювача (наскільки широко поширені оцінки від однієї вибірки даних до іншої), так і її зміщення (наскільки далеко від середнє оцінене значення є від справжнього значення). Для неупередженого оцінювача MSE є дисперсією оцінювача. Як і дисперсія, MSE має ті ж одиниці виміру, що й квадрат оцінюваної величини. За аналогією зі стандартним відхиленням, взяття квадратного кореня з MSE дає середньоквадратичну помилку або середньоквадратичне відхилення (RMSE або RMSD), які мають ті самі одиниці, що й оцінювана величина; для неупередженої оцінки RMSE є квадратним коренем дисперсії, відомою як стандартна помилка [44].

3) На даному етапі визначаємо значення параметру PSNR

$$PSNR = 10 \log_{10} \left(\frac{MAX_i^2}{MSE} \right)$$

Де MAX_i - це максимальне значення яке приймається тінями зображення,
 MSE -середньоквадратичне відхилення.

Блок PSNR обчислює пікове відношення сигнал/шум у децибелах між двома зображеннями. Це співвідношення використовується як вимірювання якості між оригіналом і стиснутим зображенням. Чим вищий PSNR, тим краща якість стиснутого або відновленого зображення. Середньоквадратична помилка (MSE) і пікове відношення сигнал/шум (PSNR) використовуються для порівняння якості стиснення зображення. MSE представляє кумулятивну квадратичну помилку між стисненим і вихідним зображенням, тоді як PSNR представляє міру пікової помилки. Чим нижче значення MSE, тим менша похибка. Щоб обчислити PSNR, блок спочатку обчислює середньоквадратичну помилку. Існують різні підходи для обчислення PSNR кольорового зображення. Оскільки людське око є найбільш чутливим до інформації про яскравість, ви можете обчислити PSNR для кольорових зображень, перетворивши зображення в колірний простір, який розділяє канал інтенсивності (яскравості), наприклад YCbCr. Y (люм) в YCbCr представляє середнє значення R, G і B. G надається найбільша вага, знову ж таки, тому що людське око сприймає його найлегше. Для порівняння та аналізу методів фільтрації можна використовувати різні інструменти порівняння зображень (оригінальне зображення та зображення, яке було оброблено). Серед цих інструментів можна виділити: стандартну помилку (immse), модель оцінки якості зображення (niqe), модель оцінки просторової якості зображення (brisque), індекс структурної схожості (ssim), пікове відношення сигнал/шум (psnr). Ssim використовується для вимірювання подібності двох зображень. Якщо маємо значення $assim = 1$, то ми говоримо про повну структурну схожість. Значення 0 вказує на відсутність структурної подібності. Psnr показує співвідношення між максимально можливою потужністю вихідного зображення і потужністю спотворення шуму, що впливає на точність зображення. Показник psnr легко обчислити, але він погано відповідає сприйманій якості. Чим вище значення psnr, тим кращий показник.

На основі експериментальних досліджень з Гістологічними, цитологічними, імуногістохімічними зображеннями, підібрано такі параметри фільтрів:

$\left\{ \begin{array}{l} \text{median kernel} = 7 * 7, \text{adapt Bilator kernel} = 11 * 11, \text{adapt Bilaten Sigma Space} = 50; \text{PSNR} \leq 12 \\ \text{median kernel} = 5 * 5, \text{adapt Bilator kernel} = 9 * 9, \text{adapt Bilaten Sigma Space} = 30; 12 < \text{RSNK} \leq 20 \\ \text{median kernel} = 3 * 3, \text{adapt Bilator kernel} = 9 * 9, \text{adapt Bilaten Sigma Space} = 20 * 20; \text{PSNR} > 20 \end{array} \right.$

4) Виходячи із результатів отриманих на етапі 4 здійснюємо безпосередньо фільтрацію зображень з допомогою медіанного фільтру.

$$lm'' = \text{median kernel} * lm'$$

5) Здійснюємо фільтрацію зображень з використанням адаптивного білатерального фільтру.

$$lm''^v = lm'''(\text{adaptive Bilateral Kernel}, \text{adaptive Bilateral Sigma Space})$$

Візуально даний алгоритм можна відобразити у вигляді схеми (рисунок 2.7).

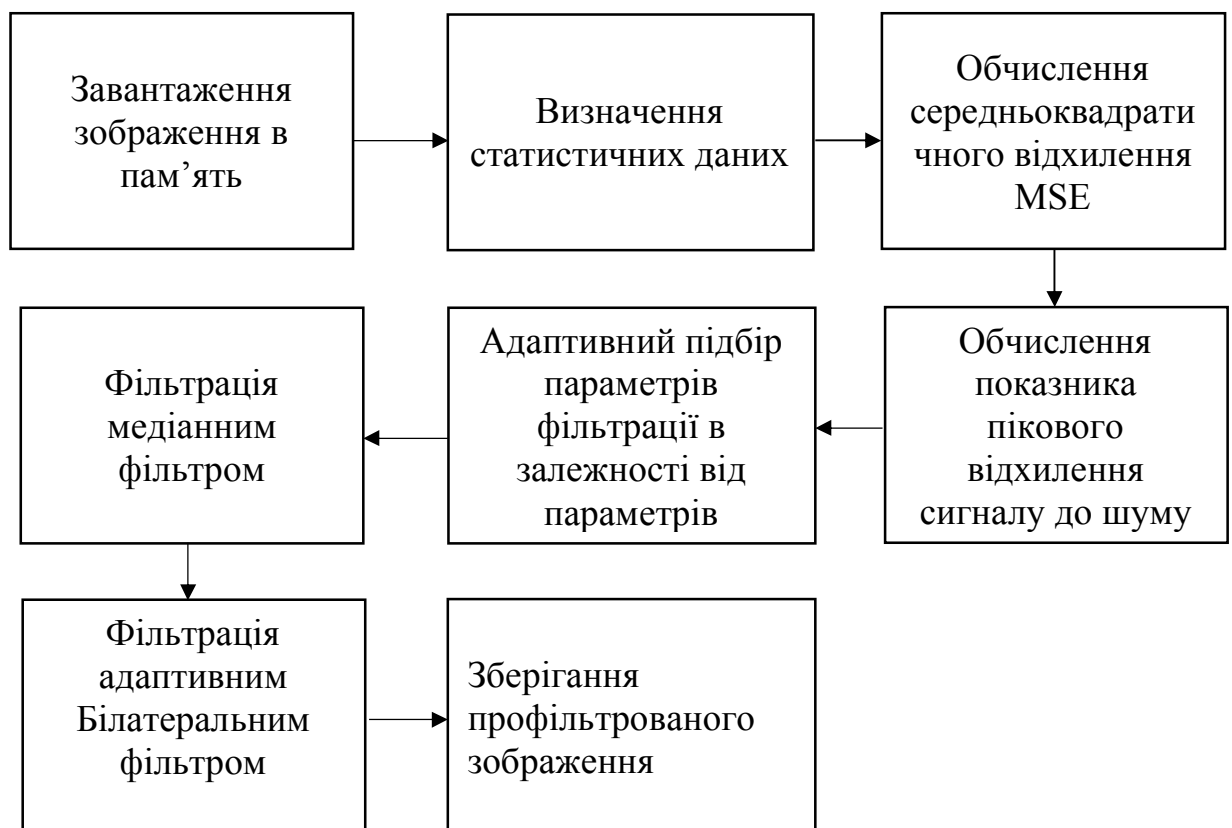


Рисунок 2.7 – Алгоритм адаптивної фільтрації зображень

Зменшення шуму відіграє важливу роль у широкому спектрі додатків, таких як реєстрація зображень, сегментація зображень, класифікація зображень і візуальне відстеження. Зображення також може погіршуватися через шум від будь-якого джерела на певному етапі отримання або передачі через будь-який носій, що є зображенням, може бути пошкоджено від внутрішнього чи зовнішнього джерела. Тому під час різних процесів, як зазначено, в зображення вводяться різні типи шумів. Різні види шумів мають свої особливості і по-різному притаманні зображенням [45].

2.3 Нечітка база правил

Для кожної групи вхідних параметрів на основі експертних оцінок було виділено лінгвістичні оцінки розподілу алгоритмів фільтрації та типів зображень в залежності від вхідних параметрів. У таблиці 2.2 наведені лінгвістичні оцінки значень фільтрації зображень, де Н- низька якість, С – середня, В – висока якість фільтрації.

Таблиця 2.2 – Лінгвістичні оцінки значень фільтрації зображень

Алгоритм фільтрації / Тип зображення	Цитологічні	Гістологічний	Імуногістохімічні
Медіанний 7x7	Н	С	С
Медіанний 5x5	С	С	В
Медіанний 3x3	В	В	В
Адаптивний білатеральний	С	С	В
Гаусовий	Н	С	С
Адаптивний білатеральний	В	В	В

Система нечітких запитів — це інтерфейс для користувачів для отримання інформації з бази даних за допомогою (квазі) речень природної мови. Було запропоновано багато реалізацій нечітких запитів, що призвели до дещо інших мов. Хоча існують деякі варіації відповідно до особливостей різних реалізацій, відповідь на речення нечіткого запиту, як правило, є списком записів, ранжованих за ступенем відповідності [46].

Наступні п'ять функціональних блоків показують структуру нечіткої системи:

- база правил — містить нечіткі правила IF-THEN. Прийняття клінічних рішень, що є наріжним каменем якісної медичної допомоги, було і залишається складним завданням. Найпершими спробами інтегрувати штучний інтелект (ШІ) в охорону здоров'я були системи підтримки прийняття діагностичних рішень (DDSS). DDSS підтримує діагностичний процес шляхом створення диференціальних діагнозів на основі наданих спостережень. Перші DDSS були натхненні міркуваннями експертів-людей і зберігали медичні знання в структурованих базах знань. Однак ці системи не знайшли широкого визнання. За останні десятиліття системи штучного інтелекту, засновані на знаннях, були замінені платформами машинного навчання (ML), які навчаються на великих обсягах даних. Прогрес у сфері ML в охороні здоров'я свідчить про те, що добре підібрані бази медичних знань більше не потрібні, і ми можемо покладатися на аналіз існуючих медичних підручників, публікацій] або великомасштабних неструктурованих даних пацієнтів. Ми стверджуємо, що методи ML і бази медичних знань доповнюють один одного і що нам не вистачає діагностичних баз знань з відкритим кодом для інтеграції обох підходів до нових DDSS, які поєднують їх сильні сторони. Передбачувані системи підтримки прийняття рішень будуть інтегрувати штучний інтелект на основі ML, структуровані алгоритми на основі знань та евристики, подібні до подвійної системної теорії людського пізнання, яка розрізняє швидке та несвідоме мислення та аналітичне, повільне та свідоме мислення. ML забезпечує розпізнавання образів, яке в даний

час сприяє прогресу в розпізнаванні зображень і голосу, але ці досягнення не перекладаються безпосередньо на DDSS, оскільки «кожна програма вимагає років цілеспрямованих досліджень і ретельної, унікальної конструкції». Системи, засновані на знаннях, були натхненні діагностичними методами, які навчали в медичній школі (наприклад, байєсівськими міркуваннями). База знань зберігає медичну інформацію в структурований спосіб, щоб комп'ютер автоматично рекомендував діагнози, а людина могла зрозуміти відмінності в цих виборах. Не на відміну від людського пізнання, ML та системи, засновані на знаннях, мають свої сильні та слабкі сторони, але, ймовірно, найкраще працюють у поєднанні.

- база даних — визначає функції належності нечітких наборів, які використовуються в нечітких правилах. Можна виділити наступні приклади баз даних:

Попередній перегляд BIOSIS – створений Web of Science Group, BIOSIS Previews є розширеним показником наук про життя та біомедичних досліджень із журналів, зустрічей, книг і патентів. База даних охоплює доклінічні та експериментальні дослідження, методи та інструменти, дослідження на тваринах тощо. Включає індексацію BIOSIS та розширені терміни захворювання MeSH.

Ця колекція є важливим джерелом високоякісних даних про охорону здоров'я, корисних не тільки для медичних працівників і пацієнтів, але й для тих, хто відповідає за дослідження, навчання, фінансування та адміністрування на всіх рівнях медичної професії [47].

Dentistry & Oral Sciences Source — це повна повнотекстова база даних досліджень для лікарів-стоматологів та дослідників. Він пропонує найкращі журнали з стоматологічних та оральних наук, які зазвичай купують стоматологічні школи та суміжні установи.

Food Science Source — це велика повнотекстова база даних, розроблена для підтримки потреб у дослідженні харчової промисловості. Він надає сотні повнотекстових журналів, монографій, журналів і торгових видань, а також велику кількість звітів про харчову промисловість і ринок. Теми включають переробку харчових продуктів, безпеку, обслуговування, доставку та інновації.

GIDEON (Глобальна мережа інфекційних захворювань та епідеміології) надає поточний, заснований на доказах ресурс для діагностики, лікування та навчання в галузі тропічних та інфекційних захворювань, епідеміології та мікробіології.

Глобальне здоров'я – створена CABI, ця база даних була створена для того, щоб ключова література з усіх джерел могла бути доведена до відома тих, хто працює в галузі охорони здоров'я. База даних охоплює всі аспекти громадського здоров'я як на міжнародному рівні, так і на рівні громади, а також багатий матеріал з інших біомедичних та природничих галузей.

International Pharmaceutical Abstracts – ця всеосяжна база даних, створена Web of Science Group, надає індексацію та реферати для фармацевтичних та медичних журналів, що публікуються по всьому світу. Це важливо для всіх, хто хоче бути в курсі сучасної літератури, пов'язаної зі здоров'ям.

Ізабель – лікарі всього світу використовують Ізабель, щоб допомогти побудувати або розширити диференційну діагностику. Вводячи інформацію, яка зазвичай отримується під час обстеження пацієнта, Ізабель надає лікарям список можливих діагнозів, щоб допомогти їм сформулювати своє мислення та швидко й легко поставити точний остаточний діагноз у місці надання допомоги.

Medcom – за допомогою спеціально розробленої системи управління навчанням Medcom надає високоякісний онлайн-освітній контент для медсестер, щоб задовольнити потреби в освіті та навчанні установ і систем охорони здоров'я по всьому світу.

MEDLINE— це надійна база даних, що містить повний текст для сотень провідних біомедичних і медичних журналів, індексованих у MEDLINE. Багато журналів доступні без ембарго, тому лікарі, медсестри, медичні працівники та дослідники можуть отримати доступ до інформації, щойно її опублікують.

Природні та альтернативні методи лікування - Ця база даних зосереджена на альтернативних і цілісних підходах до охорони здоров'я та здоров'я. Природні та альтернативні методи лікування, розроблені для дослідників здоров'я споживачів і клініцистів, є важливим інструментом, який задовольняє постійно зростаючий попит на точну, неупереджену інформацію про природне здоров'я.

Довідковий центр медсестер Плюс – час медсестри є цінним товаром, і для того, щоб забезпечити найкращий догляд, їм щоразу потрібно швидко знаходити правильні відповіді на свої клінічні запитання. Завдяки Nursing Reference Center Plus, заснованому на доказах інформаційному ресурсі, розробленому медсестрами для медсестер, користувачі можуть зробити саме це. Неперевершені за обсягом медсестри звертаються до Nursing Reference Center Plus за всіма своїми практичними, освітніми та дослідницькими потребами.

APA PsycInfo – ця база даних є найбільшим у світі ресурсом, присвяченим рецензованій літературі з поведінкової науки та психічного здоров'я. Створений Американською психологічною асоціацією, він є незамінним інструментом для відкриття глобальних наукових досліджень.

Джерело реабілітації та спортивної медицини – створений для студентів і дослідників спортивної медицини та фізичної терапії, Rehabilitation & Sports Medicine Source пропонує повнотекстові журнали з медицини та реабілітації, книги, торгові видання та журнали.

Наукова та медична ART Imagebase – ідеально підходить для студентів і викладачів анатомії, фізіології, біології та інших наук про життя, Scientific & Medical ART Imagebase (SMART Imagebase) містить медичні ілюстрації, анімацію та інтерактивні мультимедіа, які можна завантажити.

SPORTDiscus з повним текстом є остаточною базою даних для досліджень спорту та спортивної медицини. Надаючи сотні повнотекстових журналів спортивної медицини, він є важливим інструментом для медичних працівників та дослідників, які вивчають фітнес, здоров'я та спорт.

ViDAL Consult – створений ViDALGroup, ViDAL Consult покращує безпеку пацієнтів, пропонуючи найповніший доступ до іспаномовної інформації про ліки, пов'язаної з довідковим клінічним вмістом, в рамках одного інтерфейсу.

- Блок прийняття рішень — він виконує операції за правилами.
- Модуль інтерфейсу фазифікації – він перетворює чіткі величини в нечіткі величини.

- Модуль інтерфейсу дефазифікації – він перетворює нечіткі величини в чіткі величини. Нижче наведена блок-схема системи нечітких перешкод (рисунок 2.8).

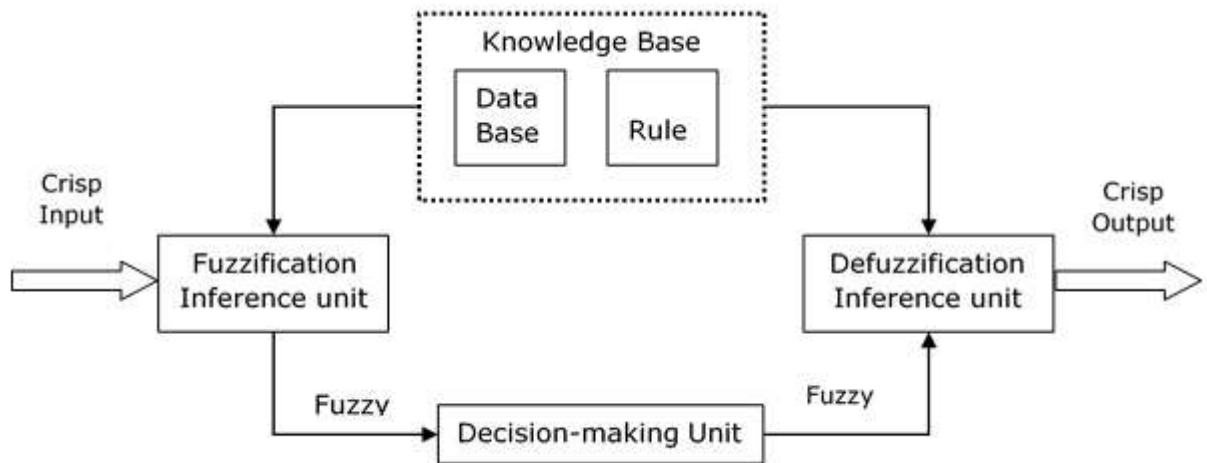


Рисунок 2.8 – Структура нечіткої системи

Класифікація сцен, регіонів або об'єктів у зображенні є фундаментальним завданням у багатьох областях, особливо пов'язаних із пошуком зображень та дистанційним зондуванням. Зокрема, класифікація сцен на знімках дистанційного зондування є активною темою дослідження в області аналізу аерофотознімків і супутникових знімків, метою якого є категоризація зображень у дискретний набір змістовних класів відповідно до типів регіонів на поверхні Землі. Перша стаття під назвою «Класифікація земного покриття на основі мультиспектральних даних з використанням засобів обчислювальної інтелекту: порівняльне дослідження» розглядає проблему злиття зображень у класифікації земного покриття з мультиспектральних супутникових знімків. Автори пропонують підхід нечіткого синтезу для злиття спектральних зображень дистанційного зондування в зображення вищого рівня розподілу земного покриття. У порівнянні з іншими методами обчислювального інтелекту, цей нечіткий підхід демонструє його придатність для об'єднання просторово-часових зображень [48].

Якщо звичайна або класична база даних є структурованою колекцією записів або даних, які зберігаються в комп'ютері, то нечітка база даних — це база

даних, яка може працювати з невизначеною або неповною інформацією за допомогою нечіткої логіки. Існує багато форм додавання гнучкості в нечітких базах даних. Найпростіший метод — додати нечіткий ступінь належності до кожного запису, тобто атрибут у діапазоні [0,1]. Однак існують інші види баз даних, які дозволяють зберігати нечіткі значення в нечітких атрибутах, використовуючи нечіткі набори, розподіли можливостей або нечіткі ступені, пов'язані з деякими атрибутами та з різними значеннями (ступінь членства, ступінь важливості, ступінь виконання тощо). Звичайно, нечіткі бази даних повинні дозволяти нечіткі запити з використанням нечітких або нечітких даних, і є деякі мови, які дозволяють виконувати такі види запитів, наприклад FSQL або SQLf. Узагальнено дослідження в нечітких базах даних включає такі чотири сфери: гнучкі запити в класичних або нечітких базах даних, розширення класичних моделей даних з метою створення нечітких баз даних (нечіткі реляційні бази даних, нечіткі об'єктно-орієнтовані бази даних тощо), нечіткий аналіз даних. методи та застосування цих досягнень у реальних базах даних.

На основі експериментальних досліджень розроблено наступний набір правил:

IF $0 < \text{PSNR} < 4$ **THEN** $\overline{\text{median kernel} = 7 * 7, \text{adapt Bilator kernel} = 11 * 11}$

IF $4 < \text{PSNR} < 7$ **THEN** $\overline{\text{median kernel} = 7 * 7, \text{adapt Bilator kernel} = 9 * 9}$

IF $7 < \text{PSNR} < 12$ **THEN** $\overline{\text{median kernel} = 5 * 5, \text{adapt Bilator kernel} = 7 * 7}$

IF $12 < \text{PSNR} < 15$ **THEN** $\overline{\text{median kernel} = 5 * 5, \text{adapt Bilator kernel} = 9 * 9}$

IF $15 < \text{PSNR} < 17$ **THEN** $\overline{\text{median kernel} = 7 * 7, \text{adapt Bilator kernel} = 9 * 9}$

IF $17 < \text{PSNR} < 20$ **THEN** $\overline{\text{median kernel} = 5 * 5, \text{adapt Bilator kernel} = 9 * 9}$

IF $\text{PSNR} > 20$ **THEN** $\overline{\text{median kernel} = 3 * 3, \text{adapt Bilator kernel} = 9 * 9}$

Теорія правил IF-THEN, запропонована Лотфі А. Заде, приваблювала багатьох дослідників і практиків своєю простотою та елегантністю. Цей внесок є спробою створити всеосяжну логічну теорію нечітких правил ЯКЩО-ТО, на основі предикатної VL-нечіткої логіки Гаєка. Формальна логічна теорія, представлена в цьому внеску, підкреслює, що: - система нечітких правил ЯКЩО-ТО з композиційним правилом висновку характеризує частково задану нечітку

функцію. - У будь-якій моделі теорії правил IF-THEN відповідна система нечітких правил IF-THEN є розв'язною.

Метою базових правил нечіткої логіки є формалізація та застосування людського висновку. Отже, нечітка логіка є частиною штучного інтелекту. Бази правил нечіткої логіки є найчастіше використовуваних інструментів у додатках з нечіткою логікою. Бази правил нечіткої логіки є набором правил, які зазвичай використовуються паралельно, але в деяких додатках можуть бути об'єднані.

Бази правил нечіткої логіки, подібно до традиційних експертних систем, ґрунтуються на базі знань, побудованої на основі людського досвіду. У той же час існує суттєва різниця в обробці та характеристиках цих знань.

Найчастіше використовують механізм нечіткого логічного висновку, званий механізмом Мамдані. Він є спрощенням більш загального механізму, який базується на “нечіткому висновку” та узагальненому правилі дедукції. Мамдані не підходять для застосування в медицині, де результати неоднозначні. Теорія ймовірностей, запропонована Лотфі Заде, описує відповідну методіку цих цілей.

Нечітка логіка є розвитком та узагальненням булевої логіки. Вона дозволяє уявити градуированность як понять, які раніше були або істинними, або хибними. Можливості, крім виникнення подій двійкового характеру (істинно чи хибно) дозволяють керувати невизначеністю цих обставин. На кордоні між цими підходами знаходиться теорія ймовірностей (винайдена Лотфі Заде), яка враховує і градуированість, і невизначеність .

Для застосування автоматизованого управління/регулювання, бази нечітких правил часто порівнюють з нейронними мережами та традиційним автоматизованим управлінням. Три зазначені вище підходи вимагають відповідно: наявності практичного досвіду, даних щодо аналізу та динамічної моделі процесу. Ці підходи можна порівнювати лише тоді, коли можливе використання всіх трьох підходів, що часто буває в теоретичних дослідженнях, але рідко на практиці. Якщо реалізація всіх підходів можлива, то пріоритетне

значення набувають практичних аспектів. Зокрема, нечітка логіка має перевагу у простоті її освоєння операторами [49].

2.4 Висновки до розділу

У даному розділі наведено опис сучасних алгоритмів фільтрації зображень, які активно використовуються в сучасних системах автоматизованої мікроскопії, зокрема медіанний фільтр, фільтр Гауса, усереднюючий фільтр, білатеральний тощо.

На основі експериментальних досліджень покроково наведено адаптивний алгоритм фільтрації біомедичних зображень на основі вхідних параметрів зображення. У вигляді блок-схеми наведено узагальнену структуру адаптивного алгоритму фільтрації зображень.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ АВТОМАТИЧНОЇ ФІЛЬТРАЦІЇ

3.1 Структура програмного модулю

У програмній інженерії процес розробки програмного забезпечення — це процес поділу роботи з розробки програмного забезпечення на менші, паралельні або послідовні кроки або підпроцеси для покращення дизайну, управління продуктом. Він також відомий як життєвий цикл розробки програмного забезпечення (SDLC). Методологія може включати попереднє визначення конкретних результатів і артефактів, які створюються та доповнюються командою проекту для розробки або підтримки програми [50].

У цьому випадку діаграми UML використовуються для передачі різних аспектів і характеристик системи. Однак це лише вид системи верхнього рівня і, швидше за все, не буде включати всі необхідні деталі для виконання проекту до самого кінця.

Передній дизайн – дизайн ескізу виконується перед кодуванням програми. Це зроблено для кращого уявлення про систему або робочий процес, який ви намагаєтеся створити. Можна виявити багато проблем або недоліків дизайну, покращуючи таким чином загальне здоров'я та благополуччя проекту.

Зворотний дизайн – після написання коду діаграми UML малюються як форма документації для різних видів діяльності, ролей, акторів і робочих процесів.

UML не є окремою мовою програмування, як Java, C++ або Python, однак, за допомогою правильних інструментів, він може перетворитися на псевдомову програмування. Щоб досягти цього, вся система повинна бути задокументована в різних діаграмах UML і, використовуючи правильне програмне забезпечення, діаграми можуть бути безпосередньо переведені в код. Цей метод може бути корисним лише в тому випадку, якщо час, необхідний для малювання діаграм, займе менше часу, ніж написання фактичного коду.

Існує кілька типів діаграм UML, і кожна з них служить різним цілям, незалежно від того, розробляється вона до впровадження чи після (як частина документації).

Дві найширші категорії, які охоплюють усі інші типи, - це поведінкова діаграма UML і структурна діаграма UML. Як випливає з назви, деякі діаграми UML намагаються проаналізувати та зобразити структуру системи чи процесу, тоді як інші описують поведінку системи, її акторів та її будівельних компонентів.

UML – діаграму класів програмного модулю для фільтрації зображень наведено на рисунку 3.1.

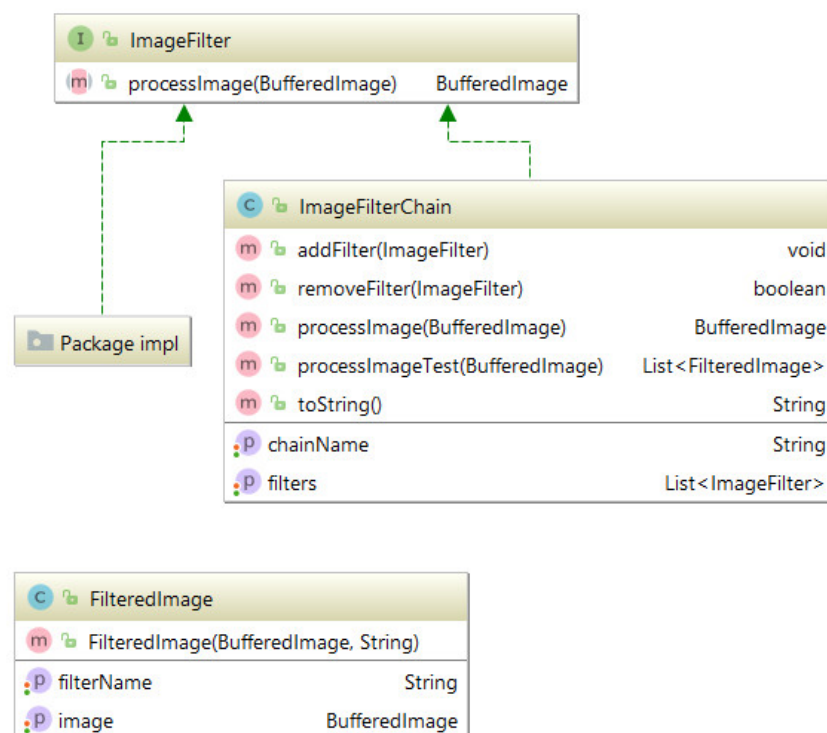


Рисунок 3.1 – UML – діаграму класів програмного модулю для фільтрації зображень

Інтерфейс «ImageFilter» призначений для задання методів, що повинні використовувати всі класи, які розроблені для здійснення фільтрації. Клас, який реалізує будь-який алгоритм фільтрації повинен перевизначити метод processImage. Цей метод обов'язкового повинен бути присутнім у класі. Даний підхід дозволяє реалізувати поліморфізм. Інтерфейс в Java - це план класу. Він має статичні константи та абстрактні методи.

Функції є фундаментальними для організації коду; вони існують у всіх мовах програмування вищого порядку. Загалом, функціональне програмування означає використання функцій з найкращим ефектом для створення чистого та підтримуваного програмного забезпечення. Точніше, функціональне програмування — це набір підходів до кодування, який зазвичай описується як парадигма програмування.

Функціональне програмування іноді визначається як протилежність об'єктно-орієнтованого програмування (ООП) і процедурного програмування. Це вводить в оману, оскільки ці підходи не є взаємовиключними, і більшість систем, як правило, використовують усі три.

Функціональне програмування дає очевидні переваги в певних випадках, воно активно використовується в багатьох мовах і фреймворках, і це помітно в сучасних тенденціях програмного забезпечення. Це корисний і потужний інструмент, який повинен бути частиною концептуального та синтаксичного інструментарію кожного розробника[51].

Інтерфейс в Java є механізмом досягнення абстракції. В інтерфейсі Java можуть бути лише абстрактні методи, а не тіло методу. Він використовується для досягнення абстракції та множинного успадкування в Java. Ідеалом у функціональному програмуванні є те, що відомо як чисті функції. Чиста функція — це функція, результати якої залежать лише від вхідних параметрів, і чия операція не ініціює жодного побічного ефекту, тобто не робить жодного зовнішнього впливу, крім повернутого значення.

Перевага чистої функції полягає в її архітектурній простоті. Оскільки чиста функція зводиться лише до аргументів і повертається значення (тобто до її API), її можна розглядати як тупик складності: єдина взаємодія із зовнішньою системою, в якій вона працює, здійснюється через визначений API.

Це на відміну від ООП, де об'єктні методи розроблені для взаємодії зі станом об'єкта (членів об'єкта), і на відміну від коду процедурного стилю, де зовнішнім станом часто маніпулюють зсередини функції.

Іншими словами, можна сказати, що інтерфейси можуть мати абстрактні методи та змінні. Він не може мати тіло методу. Інтерфейс Java також представляє відносини IS-A. Його не можна створити так само, як абстрактний клас. Починаючи з Java 8, ми можемо мати стандартні та статичні методи в інтерфейсі. Починаючи з Java 9, ми можемо мати приватні методи в інтерфейсі.

Найпростіший спосіб уникнути мутаційного стану - використовувати незмінні структури даних. Clojure надає набір незмінних списків, векторів, наборів і карт. Оскільки їх не можна змінити, «додавання» чи «вилучення» чогось із незмінної колекції означає створення нової колекції, як і стару, але з необхідними змінами. Постійність — це термін, що використовується для опису властивості, в якій стара версія колекції все ще доступна після «зміни», і що колекція зберігає свої гарантії продуктивності для більшості операцій.

Конкретно це означає, що нову версію не можна створити за допомогою повної копії, оскільки для цього знадобиться лінійний час. Неминуче, постійні колекції реалізуються за допомогою зв'язаних структур даних, так що нові версії можуть спільно використовувати структуру з попередньою версією. Однозв'язані списки та дерева є основними функціональними структурами даних, до яких Clojure додає хеш-карту, набір і вектор на основі спроб хешування зі відображенням масиву.

Ім'я конструктора завжди збігається з ім'ям класу. Клас у наведеному вище прикладі має ім'я `Person`, тому конструктору також потрібно буде назвати `Person`. Конструктор також надає як параметр ім'я об'єкта `person`, який буде створено. Параметр укладено в дужки і слідує за назвою конструктора. За дужками, які містять необов'язкові параметри, йдуть фігурні дужки. Між цими дужками знаходиться вихідний код, який програма виконує під час виклику конструктора (наприклад, `new Person («Ада»)`). Об'єкти завжди створюються за допомогою конструктора. Конструктор містить вираз `this.age = 0`. Цей вираз встановлює вікову змінну екземпляра для щойно створеного об'єкта (тобто вік «цього» об'єкта) на 0. Другий вираз `this.name = initialName` також призначає рядок, переданий як параметр, імені змінної екземпляра створеного об'єкта.

Клас ImageFilterChain складається з методів, які використовуються для для опрацювання фільтром зображення. Наприклад, метод addFilter() – додає новий фільтр до списку задач.

Клас «FilteredImage» складається з методу, який дозволяє повернути профільтроване зображення для подальшої його обробки.

На рисунку 3.2 наведено діаграму класу «MeanFilter» та його залежності.

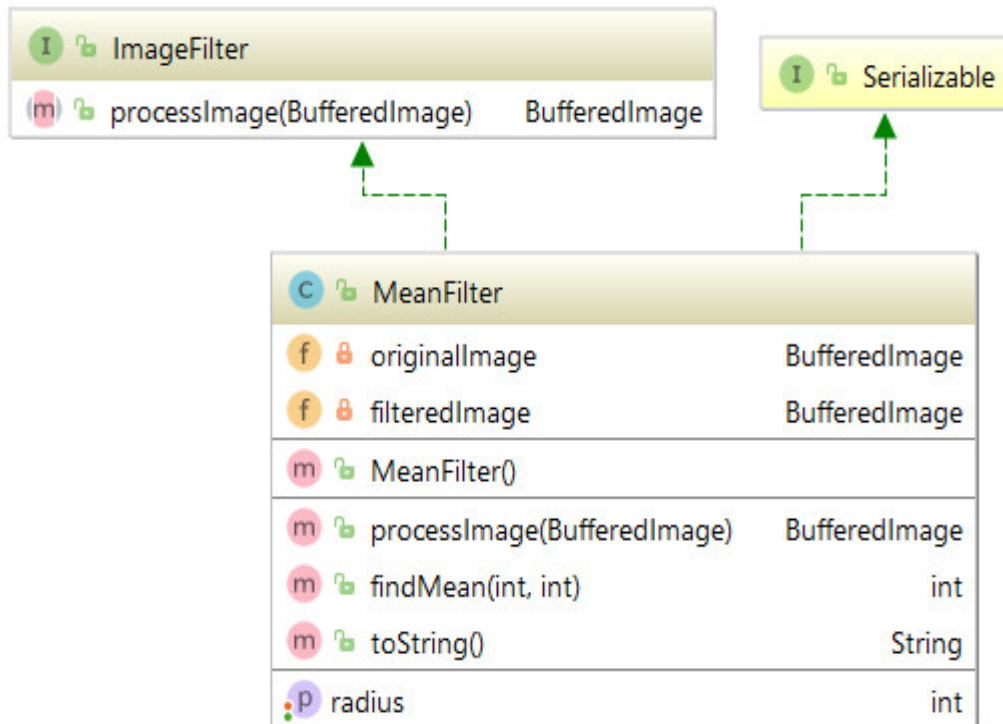


Рисунок 3.2 – Діаграма класу «MeanFilter»

Клас MeanFilter імплементує інтерфейс imageFilter, який описаний вище та інтерфейс Serializable. Клас — це визначений користувачем план або прототип, на основі якого створюються об'єкти. Він представляє набір властивостей або методів, загальних для всіх об'єктів одного типу. Загалом, оголошення класів можуть включати ці компоненти. Конструктори використовуються для ініціалізації нових об'єктів. Поля — це змінні, які надають стан класу та його об'єктів, а методи використовуються для реалізації поведінки класу та його об'єктів.

3.2 Програмні методи реалізації алгоритму

У деяких медичних зображеннях можна досягти посилення контрасту на різних рівнях, щоб виділити різні структури, присутні на них. Це може бути корисно медичним фахівцям для постановки більш конкретних діагнозів, на рентгенограмах грудної клітки та мамографії, де можна виділити різні деталі під час посилення контрасту. Запропоновано налаштування параметрів для Contrast Limited Adaptive Histogram Equalization (CLAHE) за допомогою багатоцільової мета-евристики (SMPSO), де цільовими функціями є максимізація обсягу доступної інформації (через ентропію) та мінімізація спотворення в отриманих зображеннях. (Індекс структурної схожості, SSIM) одночасно. Результати показують, що наш підхід обчислює набір рішень без домінування або набір Парето, який представляє зображення з різними рівнями контрасту та різними рівнями прихильності між ентропією та індексом структурної схожості. Зокрема, ці цільові функції є суперечливими. Ці покращені зображення надають корисну інформацію для прийняття рішень спеціалістами [52-53].

Для реалізації даного проекту було обрано мову програмування Java, оскільки характеристики і синтаксис даної мови дозволяє реалізувати поставлене завдання. На рисунку 3.3 наведено лістинг коду для реалізації медіанного фільтру.

Ключове слово `Java implements` є ще одним зарезервованим словом в Java, яке передає компілятору особливе значення. Ключове слово `implements` корисно, коли ми хочемо використовувати інтерфейс у класі. Коротше кажучи, ключове слово `implements` корисно для реалізації інтерфейсів у класах.

Ми знаємо, що інтерфейс у Java визначається як особливий тип класу, який містить лише абстрактні методи; методи, які не мають жодної реалізації чи тіла всередині себе. Він просто надає план, контракт або прототип, яким повинні дотримуватися класи.

Ключове слово `implements` використовується класом, щоб він міг слідувати або дотримуватися контракту, наданого інтерфейсом. Клас, який реалізує

інтерфейс, повинен забезпечити конкретну реалізацію всіх методів, оголошених в інтерфейсі, шляхом надання тіла методу методам.

```
MedianFilter.java x
25  * @author Mihailo Stupar
26  */
27  public class MedianFilter implements ImageFilter, Serializable{
28
29      private transient BufferedImage originalImage;
30      private transient BufferedImage filteredImage;
31
32      private int radius;
33
34      public MedianFilter () { radius = 1; }
35
36
37
38
39      @Override
40      public BufferedImage processImage(BufferedImage image) {
41
42          originalImage = image;
43
44          int width = originalImage.getWidth();
45          int height = originalImage.getHeight();
46
47          filteredImage = new BufferedImage(width, height, originalImage.getType());
48
49          int [] arrayOfPixels;
50          int median;
51          int alpha;
52          int newColor;
```

Рисунок 3.3 – Лістинг коду для реалізації медіанного фільтру

У даному класі передбачено такі методи: processImage, getArrayOfPixels, findMedian, setRadius, getRadius, toString.

Метод processImage – приймає параметром значення типу BufferedImage і використовуючи цикл формує матрицю зі значеннями пікселів.

Метод getArrayOfPixels – призначений для формування масиву пікселів результуючого зображення і повертає його у вигляді двовимірного масиву.

Метод findMedian - приймає параметром значення типу int [] і обчислює медіанне значення для масиву.

Метод setRadius – задає значення розміру вікна фільтру.

Метод toString – це перевизначений метод, який повертає значення назви фільтру.

Цикл `Java for` використовується для запуску блоку коду певну кількість разів. Початковий вираз ініціалізує та/або оголошує змінні та виконується лише один раз. Стан оцінюється - якщо умова істинна, виконується тіло циклу `for`. `UpdateExpression` оновлює значення `initialExpression`. Стан знову оцінюється - процес триває до тих пір, поки умова не стане хибною.

Змінні зберігають дані. Але виникає фундаментальна проблема щодо того, які дані зберігає змінна. У обчисленні програми може виникнути велика плутанина, якщо окремі типи даних змінних невідомі. Звідси виникає поняття типів даних. У цьому підручнику ми вивчатимемо типи даних `Java` на прикладах.

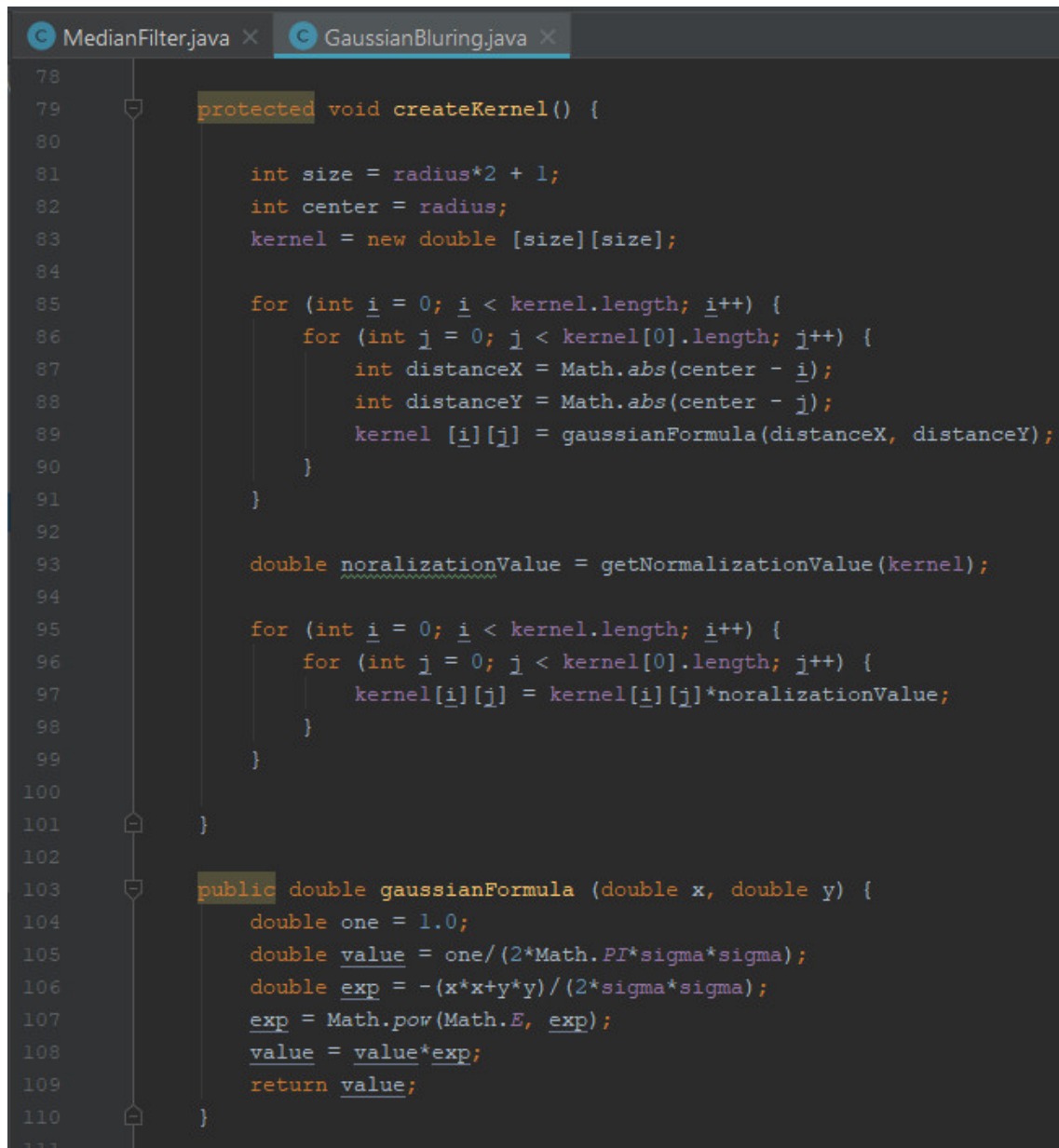
Кожен окремий біт даних, який обробляється щодня, поділяється на типи. Тип даних відомий як тип даних. `Java` використовує різні типи даних. Однак типи даних в основному поділяються на дві категорії:

- Примітивні типи даних – ці типи даних вже жорстко закодовані в компіляторі для розпізнавання під час виконання програми. Приклади: `int`, `float` тощо.
- Непримітивні типи даних – ці типи даних є спеціальними типами даних, які визначаються користувачем, тобто програма містить їх визначення. Деякі приклади - класи, інтерфейси тощо.

Модифікатор доступу `Java` визначає, які класи можуть отримати доступ до даного класу та його полів, конструкторів і методів. Модифікатори доступу можна вказати окремо для класу, його конструкторів, полів і методів. Модифікатори доступу `Java` також іноді називають у щоденній мові специфікаторами доступу `Java`, але правильна назва - це модифікатори доступу `Java`. Класи, поля, конструктори та методи можуть мати один з чотирьох різних модифікаторів доступу `Java`:

- приватний;
- за замовчуванням (пакет);
- захищений;
- публічний.

На рисунку 3.4 наведено частину лістингу коду для реалізації методів Гаусового фільтру.



```
78
79     protected void createKernel() {
80
81         int size = radius*2 + 1;
82         int center = radius;
83         kernel = new double [size][size];
84
85         for (int i = 0; i < kernel.length; i++) {
86             for (int j = 0; j < kernel[0].length; j++) {
87                 int distanceX = Math.abs(center - i);
88                 int distanceY = Math.abs(center - j);
89                 kernel [i][j] = gaussianFormula(distanceX, distanceY);
90             }
91         }
92
93         double noralizationValue = getNormalizationValue(kernel);
94
95         for (int i = 0; i < kernel.length; i++) {
96             for (int j = 0; j < kernel[0].length; j++) {
97                 kernel[i][j] = kernel[i][j]*noralizationValue;
98             }
99         }
100
101     }
102
103     public double gaussianFormula (double x, double y) {
104         double one = 1.0;
105         double value = one/(2*Math.PI*sigma*sigma);
106         double exp = -(x*x+y*y)/(2*sigma*sigma);
107         exp = Math.pow(Math.E, exp);
108         value = value*exp;
109         return value;
110     }
111
```

Рисунок 3.4 – Лістинг коду з методами для реалізації Гаусового фільтру

Об'єкт є базовою одиницею об'єктно-орієнтованого програмування і представляє реальні сутності. Типова програма Java створює багато об'єктів, які, як відомо, взаємодіють, викликаючи методи. Об'єкт складається з:

Стан: представлений атрибутами об'єкта. Він також відображає властивості об'єкта.

Поведінка: представлена методами об'єкта. Він також відображає реакцію об'єкта на інші об'єкти.

Ідентичність: дає унікальне ім'я об'єкту і дозволяє одному об'єкту взаємодіяти з іншими об'єктами.

Коли створюється об'єкт класу, кажуть, що клас створений. Усі екземпляри поділяють атрибути та поведінку класу. Але значення цих атрибутів, тобто стану, є унікальними для кожного об'єкта. Один клас може мати будь-яку кількість екземплярів.

Оскільки ми оголошуємо змінні типу (ім'я типу;). Це сповіщає компілятор, що ми будемо використовувати name для посилань на дані, типом яких є тип. З примітивною змінною це оголошення також резервує належний обсяг пам'яті для змінної. Отже, для опорної змінної тип має бути строго конкретним ім'ям класу. Загалом, ми не можемо створювати об'єкти абстрактного класу чи інтерфейсу.

Метод `createKernel` – відповідає за формування вікна Гаусового фільтру відповідно до алгоритму Гаусової фільтрації. Ключовим у даному методі є локальна змінна `size`, що визначає розмір вікна.

Метод `gaussianFormula(double x, double y)` відповідає за обчислення згідно алгоритму. На вхід метод потрапляють дві змінні типу `double`. В середині методу наявна змінна `sigma`. Значення числа Π береться зі стандартної бібліотеки `java` у вигляді константи `Math.PI`.

Метод `checkConditios` призначений для перевірки на коректність застосування фільтрів.

На рисунку 3.5 наведено лістинг коду методу `findMean` для розрахунку вікна Усереднюючого фільтру.

```
MedianFilter.java x GaussianBlurring.java x MeanFilter.java x
48     int rgb = ImageUtilities.colorToRGB(alpha, color, color, color);
49     filteredImage.setRGB(i, j, rgb);
50
51     }
52 }
53
54
55     return filteredImage;
56
57
58 }
59
60 public int findMean (int x, int y) {
61     double sum = 0;
62     int n = 0;
63     for (int i = x-radius; i <= x+radius; i++) {
64         for (int j = y-radius; j <= y+radius; j++) {
65             if (i>0 && i<originalImage.getWidth() && j>0 && j<originalImage.getHeight()) {
66                 int color = new Color(originalImage.getRGB(i, j)).getRed(); // why we use only red component here?
67                 sum = sum + color;
68                 n++;
69             }
70         }
71     }
72     return (int) Math.round(sum/n);
73 }
```

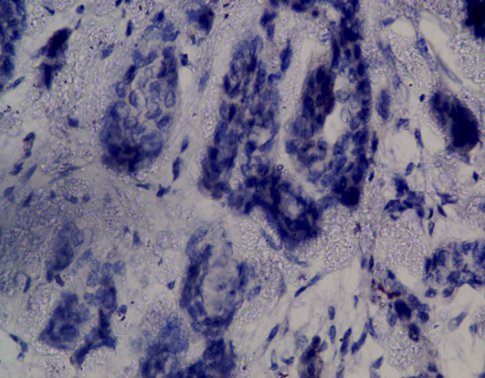
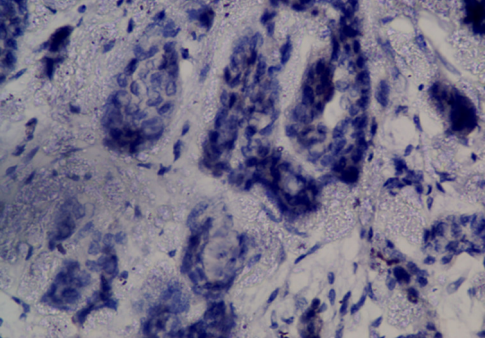
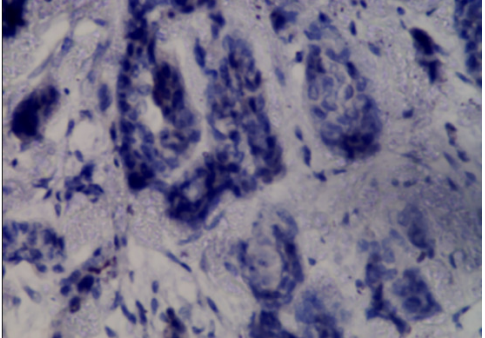
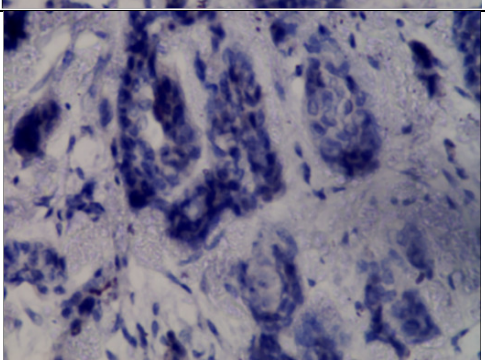
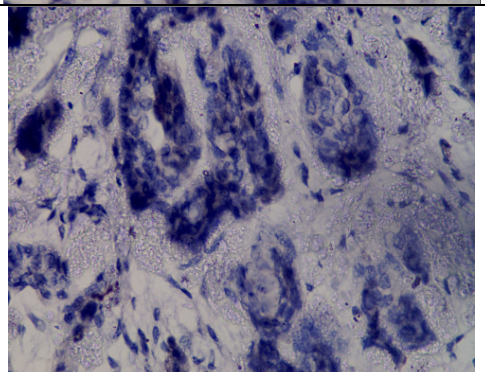
Рисунок 3.5 – Лістинг коду методу findMean для розрахунку вікна
Усереднюючого фільтру

Для реалізації усереднюючого фільтру згідно формули розрахуємо кінцеве значення використовуючи вкладений цикл для того, щоби зробити перебір по усіх елементах матриці, які представляє зображення.

3.3 Порівняльний аналіз експериментів

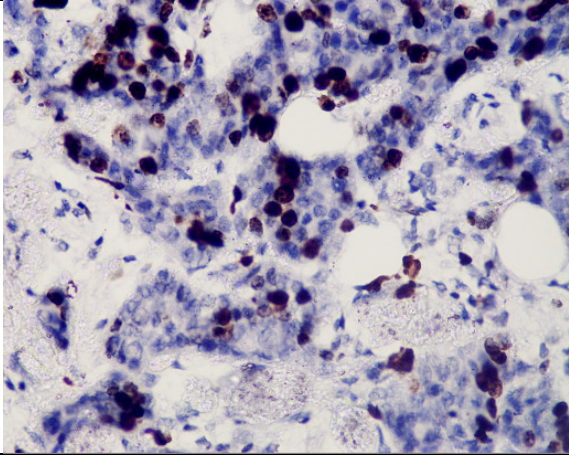
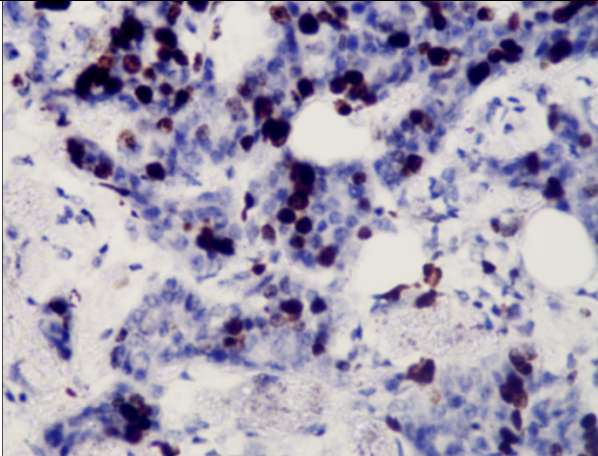
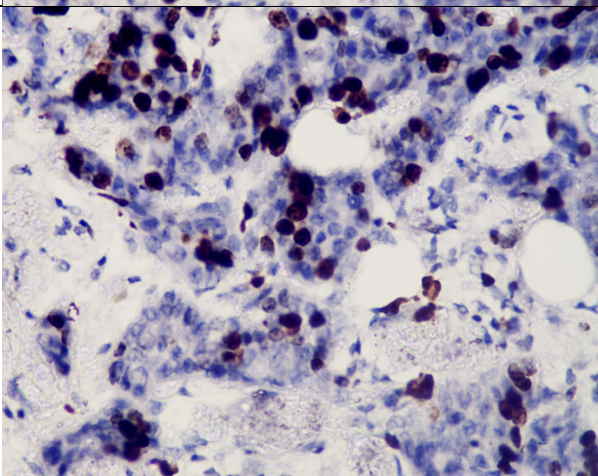
У таблиці 3.1 наведено результати застосування різного роду фільтрів до імуногістохімічного зображення. Візуально змін між цими зображеннями помітити важко, однак на етапі сегментування зображень можна відчутти реальний вплив етапу попереднього оброблення на кінцевий результат.

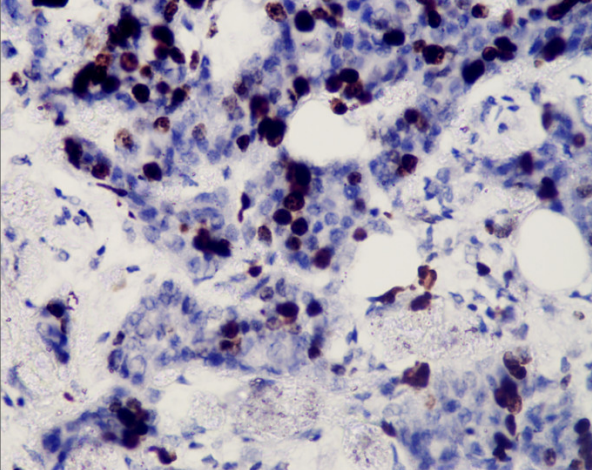
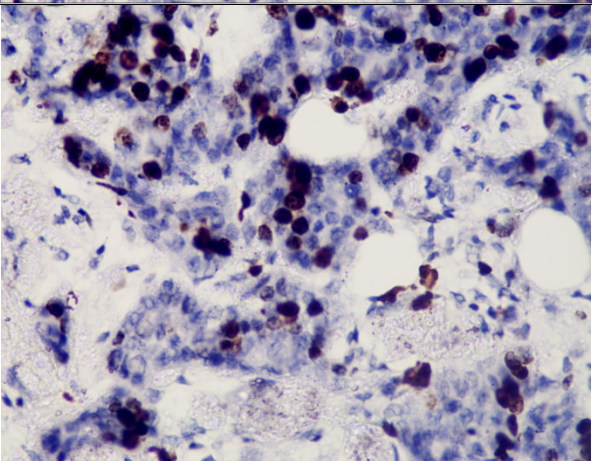
Таблиця 3.1 – Фільтрація імуногістохімічного зображення (експеримент 1)

Тип зображення	Приклад зображення	
Оригінальне		
Гаусовий фільтр		
Медіанний фільтр 7x7		
Медіанний фільтр 3x3		
Розроблений алгоритм		

У таблиці 3.2 наведено результати застосування різного роду фільтрів до імуногістохімічного зображення (експеримент 2). Для тестування обрано наступні фільтри: Гаусовий, Медіанний фільтр з розміром вікна 7x7, Медіанний фільтр з розміром вікна 3x3, Усереднюючий фільтр.

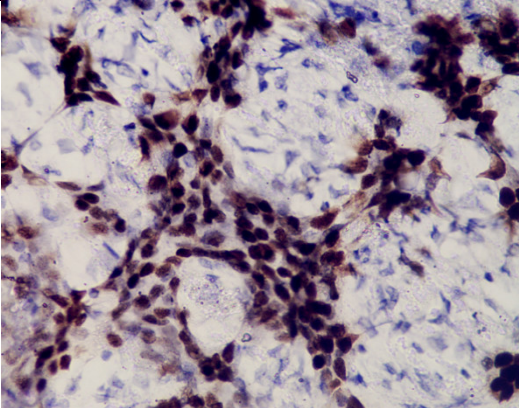
Таблиця 3.2 – Фільтрація імуногістохімічного зображення (експеримент 2)

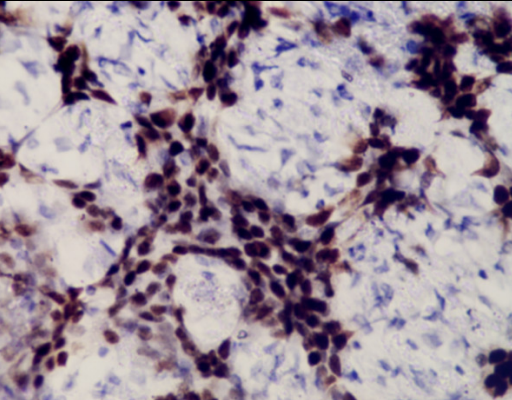
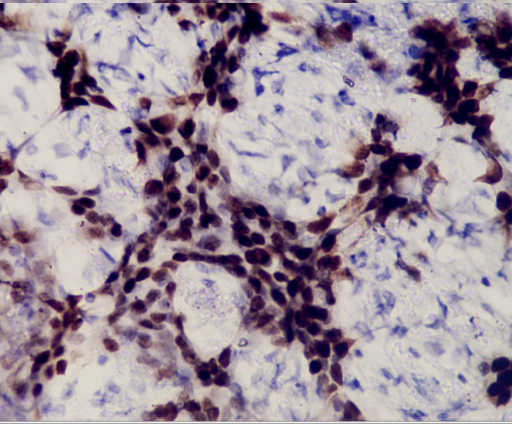
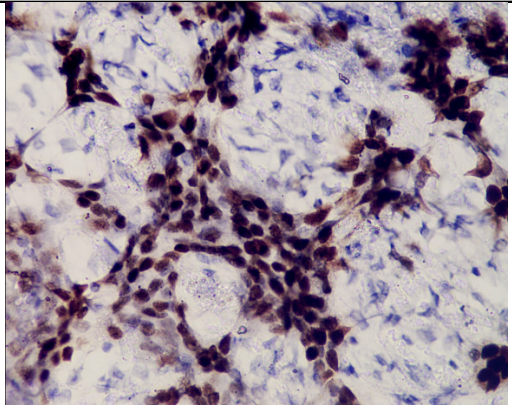
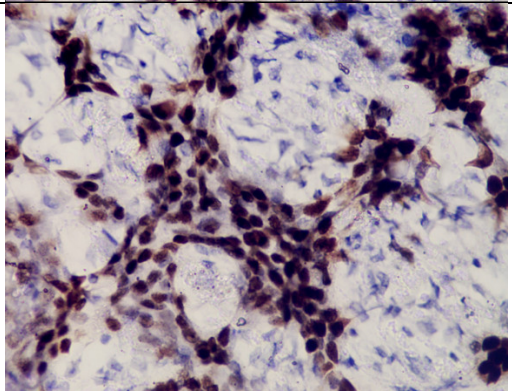
Тип зображення	Приклад зображення	
Оригінальне		
Гаусовий фільтр		
Медіанний фільтр 7x7		

Медіанний фільтр 3x3	
Розроблений алгоритм	

У таблиці 3.3 наведено результати застосування різного роду фільтрів до імуногістохімічного зображення (експеримент 3).

Таблиця 3.3 – Фільтрація імуногістохімічного зображення (експеримент 3)

Тип зображення	Приклад зображення
Оригінальне	

Гаусовий фільтр			
Медіанний фільтр 7x7			
Медіанний фільтр 3x3			
Розроблений алгоритм			

Сегментація зображення – це процес поділу цифрового зображення на кілька сегментів (об’єктів). Мета сегментації – змінити представлення зображення на щось більш значуще та легше для аналізу. Ми можемо зібрати різні методи сегментації, як показано нижче:

- класичні підходи на основі комп’ютерного зору;

- порогове значення;
- методи на основі регіонів;
- методи на основі краю/границі;
- підходи на основі штучного інтелекту;
- методи на основі кластеризації;
- нейронні мережі.

Результати сегментації досліджуваних зображень наведено на рисунку 3.6.

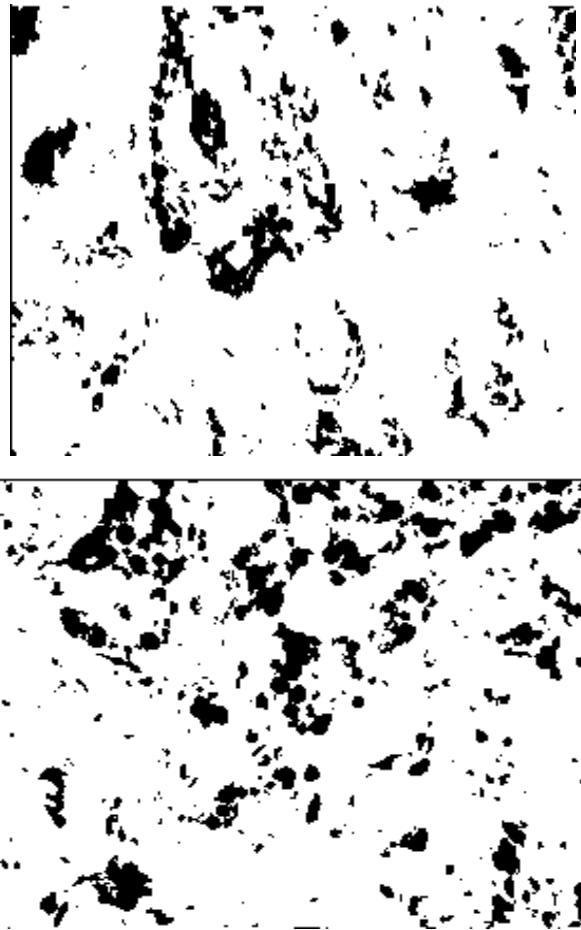


Рисунок 3.6 - Результати сегментації досліджуваних зображень

Ручна сегментація медичних зображень призводить до двох основних проблем: для розмежування потрібно багато часу, а відтворюваність ставиться під сумнів. По-перше, час, необхідний для сегментації, не стискається, і він корелює з кількістю та розміром зображень. Оскільки розмір цих двох параметрів збільшується через полегшення доступу до медичних зображень та вдосконалення технологій отримання, ручна сегментація стає важкою. По-друге,

відтворюваність відповідає узгодженню між результатами множинних вимірювань даних (тут результати сегментації) за однією методологією. У сегментації медичних зображень добре відомо, що існує мінливість між та інтраоператорами. Перший стосується спостережуваних відмінностей у результатах сегментації, отриманих двома різними операторами, а другий — до спостережуваних відмінностей між двома результатами завдань сегментації, які виконуються одним оператором у два різні моменти часу. Через вирішальну роль сегментації в медичній діагностиці та лікуванні відтворюваність методу є принципово важливою.

В останні роки широко використовується автоматична сегментація на основі методів глибокого навчання (DL), коли нейронна мережа може автоматично вивчати особливості зображення, що різко контрастує з традиційним методом ручного навчання. U-net є однією з найважливіших структур семантичної сегментації для згорткової нейронної мережі (CNN). Він широко використовується в області аналізу медичних зображень для сегментації ураження, анатомічної сегментації та класифікації. Перевага цієї мережевої структури полягає в тому, що вона може не тільки точно сегментувати потрібну цільову функцію, ефективно обробляти й об'єктивно оцінювати медичні зображення, але також допомагає підвищити точність діагностики за медичними зображеннями. Тому в цій статті представлений огляд літератури щодо сегментації медичних зображень на основі U-net, зосереджено на успішному досвіді сегментації U-net для різних областей ураження в шести системах медичної візуалізації. Поряд з останніми досягненнями в DL, ця стаття знайомить з методом поєднання оригінальної архітектури U-net з глибоким навчанням і методом покращення мережі U-net. Згорткові нейронні мережі (CNN) відіграють важливу роль у сфері сегментації медичних зображень. Серед багатьох видів CNN архітектура U-мережі є однією з найвідоміших повністю згорткових мережевих архітектур для завдань медичної семантичної сегментації. Останні роботи показують, що мережа U-net може бути значно глибшою, що призведе до покращення продуктивності при виконанні завдань сегментації. Хоча додавання більше шарів безпосередньо в мережу є популярним способом

зробити мережу глибшою, це може призвести до зникнення градієнта або надлишкових обчислень під час навчання[54-55].

Порівняльна характеристика роботи алгоритмів автоматичного покращення якості зображень за критерієм SSIM наведено у таблиці 3.4.

Індекс структурної схожості (SSIM) — це перцептивний показник, який кількісно визначає погіршення якості зображення, викликане обробкою, наприклад стисненням даних, або втратами при передачі даних. Це повна опорна метрика, яка вимагає двох зображень з одного і того ж зображення — еталонного та обробленого зображення. Оброблене зображення зазвичай стискається. Його можна, наприклад, отримати, зберігши еталонне зображення у форматі JPEG (у будь-якому рівні якості), а потім прочитавши його назад. SSIM найбільш відомий у відеоіндустрії, але має потужні можливості для фотозйомки. Можна використовувати будь-яке зображення, включно з тестовими шаблонами Imatest, такими як розлиті монети або журнал F-Contrast.

Кластеризація k-середніх — це неконтрольований алгоритм кластеризації, який розділяє інтенсивності зображення на основі центроїдів кластера з подальшим розрахунком відстані між кожним пікселем зображення та відповідним центроїдом. Алгоритм призначає кожному пікселю певний центроїд на основі мінімального значення відстані. Алгоритм також оновлює центроїди, знаходячи середні значення відстані призначених пікселів центроїдів. Значення відстаней оновлюються щодо нових центроїдів, а пікселі повторно призначаються. Алгоритм продовжується до тих пір, поки не відбудеться суттєвих змін відстаней від центроїдів [56]. Ефективна кластеризація k-середніх досягається шляхом мінімізації дисперсії всередині кластера, яка є сумою квадратів у кожному кластері (SSW), і максимізації дисперсії між кластерами, яка є сумою квадратів між кластерами.

Таблиця 3.4 – Порівняльна характеристика роботи алгоритмів автоматичного покращення якості зображень за критерієм SSIM

Алгоритм	1	2	3	4	5	6	7
Сегментація без попередньої обробки	0,5	0,52	0,47	0,68	0,7	0,62	0,65
HE	0,61	0,75	0,64	0,71	0,75	0,7	0,79
CLANE	0,63	0,75	0,6	0,69	0,8	0,81	0,85
MSR	0,65	0,75	0,7	0,75	0,81	0,83	0,88
Розроблений алгоритм на основі медіанного та адаптивного білатерального фільтрів	0,73	0,81	0,84	0,83	0,83	0,87	0,91

На рисунку 3.7 наведено порівняльний аналіз результатів роботи алгоритмів автоматичного покращення якості зображень.

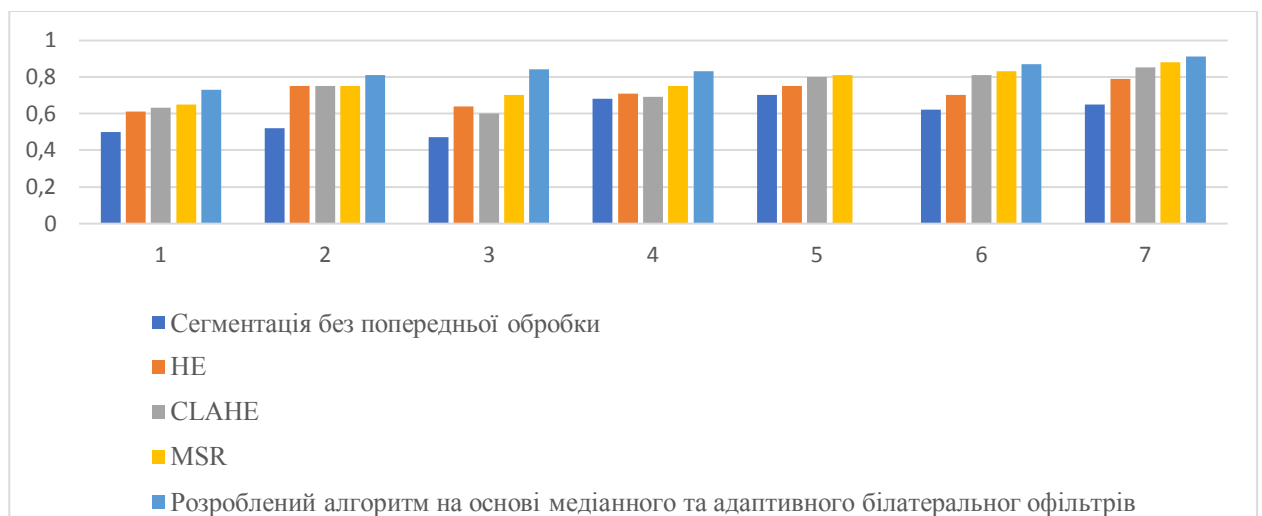


Рисунок 3.7 - Порівняльний аналіз результатів роботи алгоритмів автоматичного покращення якості зображень

Виходячи із результатів наведених у таблиці 3.4 та рисунку 3.1 можна зробити висновок, що розроблений алгоритм показав кращі результати у порівнянні з аналогами. Найгірший результат показали зображення, які не були попередньо оброблені. Приріст якості сегментації із застосуванням попередньої обробки зображень на основі розробленого алгоритму складає орієнтовно 15-20% у порівнянні із зображеннями, до яких не застосовувалась попередня обробка. Приріст якості сегментації із застосуванням попередньої обробки зображень на основі розробленого алгоритму складає орієнтовно 5-10% у порівнянні із відомими алгоритмами.

3.4 Висновки до розділу

У даному розділі наведено UML – діаграму класів розроблено модулю, який реалізує розроблений алгоритм адаптивної фільтрації зображень. Проведено порівняльний аналіз результатів роботи алгоритмів фільтрації. Приріст якості сегментації із застосуванням попередньої обробки зображень на основі розробленого алгоритму складає орієнтовно 15-20% у порівнянні із зображеннями, до яких не застосовувалась попередня обробка. Приріст якості сегментації із застосуванням попередньої обробки зображень на основі розробленого алгоритму складає орієнтовно 5-10% у порівнянні із відомими алгоритмами.

ВИСНОВКИ

На основі аналітичного підходу проведено порівняльний аналіз біомедичних зображень, зокрема гістологічних, цитологічних, імуногістохімічних, що дало змогу виділити їх переваги та недоліки на етапі опрацювання на низькому рівні комп'ютерного зору.

Проведено аналіз сучасних алгоритмів фільтрації зображень, що дозволило виділити алгоритми фільтрації та їх модифікації для подальшої розробки адаптивного алгоритму фільтрації.

З допомогою порівняльного аналізу виділено сучасні програмні засоби опрацювання зображень, зокрема в системах автоматизованої мікроскопії, де досліджуються зображення, отримані в результаті мікроскопічного дослідження.

Наведено опис сучасних алгоритмів фільтрації зображень, які активно використовуються в сучасних системах автоматизованої мікроскопії, зокрема медіанний фільтр, фільтр Гауса, усереднюючий фільтр, білатеральний тощо.

На основі експериментальних досліджень покроково наведено адаптивний алгоритм фільтрації біомедичних зображень на основі вхідних параметрів зображення. У вигляді блок-схеми наведено узагальнену структуру адаптивного алгоритму фільтрації зображень.

Наведено UML – діаграму класів розроблено модулю, який реалізує розроблений алгоритм адаптивної фільтрації зображень. Проведено порівняльний аналіз результатів роботи алгоритмів фільтрації. Приріст якості сегментації із застосуванням попередньої обробки зображень на основі розробленого алгоритму складає орієнтовно 15-20% у порівнянні із зображеннями, до яких не застосовувалась попередня обробка. Приріст якості сегментації із застосуванням попередньої обробки зображень на основі розробленого алгоритму складає орієнтовно 5-10% у порівнянні із відомими алгоритмами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Королюк И.П. Медицинская информатика : Учебник / И.П. Королюк. – 2 изд., перераб. и доп. – Самара : ООО «Офорт»: ГБОУ ВПО «СамГМУ». 2012. – 244 с.
2. Візуалізація медико-біологічних даних. Обробка й аналіз медичних зображень «Медична інформатика» / упор. Рисована Л.М., Радзішевська Є.Б. – Харків : ХНМУ, 2016. – 23 с.
3. Кузнецова Л.А. Анализ проблематики использования изображений в цифровом формате в медицинской практике / Л.А. Кузнецова, М.А. Мельник, К.А. Катилова // Системи обробки інформації. 2016. – № 4. – 169-171 с.
4. Дороничева А.В. Методы Распознавания Медицинских Изображений Для Задач Компьютерной Автоматизированной Диагностики / А.В. Дороничева, С.З. Савин // Современные проблемы науки и образования. – 2014. – № 4. – 10 – 18 с.
5. Wang L., Ker J., Rao J., Lim T. Deep learning applications in medical image analysis. IEEE Special Section on Soft Computing Techniques for Image Analysis in the Medical Industry Current Trends, Challenges and Solutions. 2018;6:9375–9389.
6. Javed U., Riaz M. M., Ghafoor A., Cheema T. A. MRI Brain Classification Using Texture Features, Fuzzy Weighting and Support Vector Machine. Progress in Electromagnetics Research B. 2013;53:73–88.
7. Гонсалес Р., Вудс Р. Цифровая обработка изображений: Пер. с англ. М.: Техносфера, 2012. 1104 с.
8. Березький О.М. Методи, алгоритми та програмні засоби опрацювання біомедичних зображень: монографія / [О.М. Березький, Ю.М. Батько, К.М. Березька, С.О. Вербовий, Т.В. Дацко, Л.О. Дубчак, І.В. Ігнатєв, Г.М. Мельник, В.Д. Николюк, О.Й. Піцун]; під наук. ред. Березький О.М., Тернопіль . ТНЕУ «Економічна думка», 2017. - 330 с.
9. Барінов Е. Ф. Гістологія, цитологія та ембріологія. Кн.1: Цитологія і загальна ембріологія: навч. посіб. для студ. вищ. мед. навч. закл. IV рівня акредитації / Е. Ф. Барінов, Ю. Б. Чайковський . – К. : Медицина — 2010. — 216 с.

- 10.Потапов А. А. Новейшие методы обработки изображений / А. А. Потапов, А. А. Пахомов, С. А.. Никитин.— М.: Физматлит, 2008. — 496 с.
- 11.Luo J., Wu M., Gorukumar D., Zhao Y. Big data application in biomedical research and health care: a literature review. *Biomedical Informatics Insights*. 2016;8:1–10.
- 12.Шапиро Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман — М. : Бином. Лаборатория знаний, 2006. — 752 с.
- 13.Chitradevi B. An Overview on Image Processing Techniques / B. Chitradevi, P.Srimathi // *International Journal of Innovative Research in Computer and Communication Engineering*. – Vol. 2 (11). – 2014. – pp.
- 14.Jayashree Kalpathy-Cramer. Evaluating performance of biomedical image retrieval systems – an overview of the medical image retrieval task at ImageCLEF 2004–2013 / Alba García Seco de Herrera, Dina Demner-Fushman, Sameer Antani, Steven Bedrick, and Henning Müller // *Comput Med Imaging Graph*. – 2015. – pp. 55–61.
- 15.Aswathy M. A. Detection of breast cancer on digital histopathology images: Present status and future possibilities / M. A. Aswathy, M. Jagannath // *Informatics in Medicine Unlocked*. – 2017. – Vol. 8. – P. 74-79
- 16.Lyashenko V. Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology / V. Lyashenko, O. Kobylin, R. Matarneh, Y. P. Putyatin // *International Journal of Advance Research in Computer Science and Management Studies*. – 2016. – Vol. 4(1) . – P. 85-94.
- 17.James A.P., Medical Image Fusion: A survey of the state of the art / A.P. James, B. V. Dasarathy// *Information Fusion*. – 2014.
- 18.Waseem K. M. A Survey: Image Segmentation Techniques / K. M. Waseem // *International Journal of Future Computer and Communication*, Vol. 3, No. 2, April 2014. – P. 89 -93.
- 19.Бондіна Н.М. Порівняння алгоритмів фільтрації медичних зображень за оцінками їх якості / Н.М. Бондіна, О.С. Калмичков, О.А. Козіна // *Вісник НТУ "ХПІ"*. Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2013. – № 39 (1012). – С. 15 – 21.

20. Khan S., Yong S. P. A deep learning architecture for classifying medical images of anatomy object. Proceedings of APSIPA Annual Summit and Conference; December 2017; Kuala Lumpur, Malaysia. APSIPA.
21. Хрящев Д.А. Об одном методе анализа цифрового изображения с применением гистограмм / Д.А. Хрящев / - Вестник АГТУ. Сер.: Управление вычислительная техника и информатика – 2010 - №1 – С. 109-113.
22. Колючкин В.Я., Нгуен К.М. Выбор рабочего словаря признаков в алгоритмах распознавания изображений на основе фурье-дескрипторов// Электронное научно-техническое издание «Наука и Образование». № 3. С. 188 - 199.
23. Manogaran G., Lopez D. A survey of big data architectures and machine learning algorithms in healthcare. International Journal of Biomedical Engineering and Technology. 2017;25(2/3/4):182–211.
24. Li Q., Cai W., Wang Z., Zhou Y., Feng D. G., Chen M. Medical image classification with convolutional neural network. Proceedings of the 13th International Conference on Control, Automation, Robotics & Vision Marina Bay Sands (ICARCV); December 2014; Singapore.
25. Chowdhary C. L., Acharjya D. P. Segmentation and feature extraction in medical imaging: a systematic review. Procedia Computer Science. 2020;167:26–36.
26. Колючкин В.Я., Нгуен К.М., Чан Т.Х. Алгоритмы обработки информации в системах технического зрения промышленных роботов // Труды московского государственного технического университета имени Н.Э. Баумана. Современные проблемы оптотехники, 2013. №610. С.139-150.
27. Нгуен К.М, Чан Т.Х. Алгоритмы обработки информации в системах технического зрения промышленных роботов//устный доклад. Сборник трудов VIII Международной конференции молодых ученых и специалистов «Оптика-2013».- 187- 189.
28. Adatrao S. An analysis of different image preprocessing techniques for determining the centroids of circular marks using hough transform / S. Adatrao, M. Mittal // 2nd International Conference on Frontiers of Signal Processing (ICFSP). - 15-17 Oct. 2016. –pp.

29. Kaur P. Contour Detection of Gradient Images Using Morphological Operator and Transform Domain Filtering / P. Kaur, A. Gupta // 2015 IEEE International Conference on Computational Intelligence & Communication Technology – 2015. – P. 107-111.
30. PHAM, D.L., XU, C., PRINCE, J.L., Current methods in medical image segmentation, *AnnuRevBiomedEng*2 (2000)315-37.
31. Rizwan I Haque I., Neubert J. Deep learning approaches to biomedical image segmentation. *Informatics in Medicine Unlocked*. 2020;18.
32. KASS, M., WITKIN, A., TERZOPOULOS, D., Snakes: active contour models, *InternationalInt.J.Comput.Vision*1 (1988)321-331.
33. COOTES, T.F., TAYLOR, C.J., COOPER, D.H., GRAHAM, J., Active shape models -- their training and application, *Computer Vision and Image Understanding*61 (1995)38-59.
34. Giger M. L. *Machine Learning in Medical Imaging*. Reston, VA, USA: American College of Radiology; 2017.
35. GONZALEZ, R.C., WOODS, R.E., EDDINS, S., *Digital image processing using MATLAB*, Repr. with corr. [i.e. 3rd] edn, Prentice Hall, Upper Saddle River, NJ (2004) xvi, 716 pp.
36. Zhu Y. An Improved Median Filtering Algorithm for Image Noise Reduction / Youlian Zhu, Cheng Huang // *Physics Procedia*. – V. 25. – 2012. – P. 609 – 616
37. Reza A.M. Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement / A.M. Reza // *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology* (2004) Vol. 38(1). – P. 35–44.
38. Jang C.Y. A fast multi-scale retinex algorithm using dominant SSR in weights selection. / C.Y. Jang, J.H. Lim, Y.H. Kim // *ISOC-2012*. – P. 37-40.
39. Nakonechny A. Adaptive wavelet thresholding for image denoising using sure minimization and clustering of wavelet coefficients / A. Nakonechny, V. Fedak // *Technical Transactions. Electrical Engineering, Wydawnictwo Polytechniki Krakowskiej* vol. 2-E/2015, pp. 197-210.

40. MAINTZ, J.B., VIERGEVER, M.A., A survey of medical image registration, *Med Image Anal* 2 1 (1998) 1-36.
41. Chowdhary C. L., Shynu P. G., Gurani V. K. Exploring breast cancer classification of histopathology images from computer vision and image processing algorithms to deep learning. *International Journal of Advanced Science and Technology*. 2020;29(3):43–48.
42. LINDEBERG, T., Edge Detection and Ridge Detection with Automatic Scale Selection, *Int. J. Comput. Vision* 30 (1998) 117-156.
43. PLUIM, J.P., MAINTZ, J.B., VIERGEVER, M.A., Mutual-information-based registration of medical images: a survey, *IEEE transactions on medical imaging* 22 8 (2003) 986-1004.
44. Колучкин В.Я., Нгуен К.М., Чан Т.Х. Алгоритмы обработки изображений в системах машинного зрения роботизированных производственных линий // *Нейрокомпьютеры: разработка, применение*. М.: Издательство «Радиотехника». 2014. №3. С.44-51.
45. Tchagna Kouanou A., Tchiotsop D., Kengne R., Zephirin D. T., Adele Armele N. M., Tchinda R. An optimal big data workflow for biomedical image analysis. *Informatics in Medicine Unlocked*. 2018;11:68–74.
46. Miranda E., Aryuni M., Irwansyah E. A survey of medical image classification techniques. *Proceedings of the International Conference on Information Management and Technology (ICIMTech)*; November 2016; Bandung, Indonesia. pp. 56–61.
47. Masood S. A Survey on Medical Image Segmentation / S. Masood, M. Sharif, A. Masood, M. Yasmin // *Current Medical Imaging Reviews*. – 2015. – Vol. 11. – P. 3-14.
48. Ma B., Li X., Xia Y., Zhang Y. Autonomous deep learning: a genetic DCNN designer for image classification. *Neurocomputing*. 2019;379.
49. Wackerly Dennis; Mendenhall William; Scheaffer Richard I. (2008). *mathematical statistics with applications* (7 ed.). belmont, ca, usa: thomson higher education. isbn 978-0-495-38508-0.

50. Wu X., Sahoo D., Steven C., Hoi H. Recent advances in deep learning for object detection. *Neurocomputing*. 2020;396.
51. Gareth James; Witten Daniela; Hastie Trevor; Tibshirani Rob (2021). *An introduction to statistical learning: with applications in R*. Springer. ISBN 978-1071614174.
52. Zhu C., Song F., Wang Y., et al. Breast cancer histopathology image classification through assembling multiple compact CNNs. *BMC Medical Informatics and Decision Making*. 2019;19:p. 198.
53. Chowdhary C. L., Shynu P. G., Gurani V. K. Exploring breast cancer classification of histopathology images from computer vision and image processing algorithms to deep learning. *International Journal of Advanced Science and Technology*. 2020;29(3):43–48.
54. Bergamasco L. C. C., Nunes Fatima L. S. Intelligent retrieval and classification in three-dimensional biomedical images—a systematic mapping. *Computer Science Review*. 2019;31:19–38.
55. Sun Y., Li L., Zheng L. Image classification based on PCA of multi-view deep representation. *Journal of Visual Communication and Image Representation*. 2019;62:253–258.
56. Khan S., Yong S. P. A deep learning architecture for classifying medical images of anatomy object. *Proceedings of APSIPA Annual Summit and Conference*; December 2017; Kuala Lumpur, Malaysia. APSIPA.

