

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Попович Віталій Васильович

**«Алгоритм сегментації кольорових зображень на
основі аналізу малоінформативних ділянок
зображення / Algorithm for segmentation of color
images based on the analysis of the image
uninformative areas»**

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

Виконав студент групи КІм-22
В.В. Попович

Науковий керівник:
к.т.н., доц. Ю.М. Батько

Кваліфікаційну роботу допущено
до захисту:

" ____ " _____ 20__ р.

Завідувач кафедри
_____ О. М. Березький

Тернопіль – 2021

ЗМІСТ

Вступ.....	3
1 Програмні та апаратні системи обробки цифрових зображень	6
1.1 Класифікація та характеристики цифрових зображень.....	6
1.2 Технології цифрової обробки зображень	14
1.3 Програмні засоби для обробки цифрових зображень	20
1.4 Постановка задач дослідження	25
1.5 Висновки до розділу.....	26
2 Методи та алгоритми обробки та опису цифрових зображень.....	27
2.1 Методи та алгоритми сегментації цифрових зображень.....	27
2.2 Алгоритми оцінки результатів сегментації.....	39
2.3 Алгоритм сегментації зображень на основі аналізу малоінформативних областей	43
2.4 Висновки до розділу.....	46
3 Програмна система обробки та аналізу цифрових зображень.....	47
3.1 Структура програмного додатку сегментації цифрових зображень....	47
3.2 Програмні функції сегментації зображень.....	59
3.3 Тестування та аналіз реалізованої системи.....	64
3.4 Висновки до розділу.....	69
Висновки.....	70
Список використаної літератури.....	71

ВСТУП

Актуальність роботи. Інтерес до методів цифрової обробки зображень випливає з двох основних сфер застосування: удосконалення графічної інформації для людської інтерпретації та обробки даних зображень для таких завдань, як зберігання, передача та вилучення зображальної інформації.

Область цифрової обробки зображень відноситься до обробки цифрових зображень за допомогою цифрового комп'ютера. Зауважимо, що цифрове зображення складається з кінцевої кількості елементів, кожен з яких має певне розташування та значення. Ці елементи називаються елементами зображення, елементами зображення, зображеннями та пікселями. Піксель – це термін, який найчастіше використовується для позначення елементів цифрового зображення. Зір є найрозвиненішим із наших почуттів, тому не дивно, що зображення відіграють найважливішу роль у людському сприйнятті. Проте слід зазначити, що людина обмежена візуальним діапазоном електромагнітного (ЕМ) спектру, а машини для обробки зображень охоплюють майже весь спектр ЕМ, починаючи від гамма-до радіохвиль.

Вони можуть оперувати зображеннями, створеними джерелами, які люди не звикли асоціювати із зображеннями. До них належать ультразвук, електронна мікроскопія та зображення, створені комп'ютером. Таким чином, цифрова обробка зображень охоплює широку та різноманітну область застосування.

Серед авторів немає загальної згоди щодо того, де зупиняється обробка зображень і починаються інші суміжні області, такі як аналіз зображень і комп'ютерний зір. Іноді розрізняють, визначаючи обробку зображень як дисципліну, в якій як вхід, так і вихід процесу є зображеннями. Наприклад, згідно з цим визначенням, навіть тривіальне завдання обчислення середньої інтенсивності зображення (яке дає єдине число) не буде вважатися операцією обробки зображення. З іншого боку, є такі галузі, як комп'ютерний зір, кінцевою метою яких є використання комп'ютерів для наслідування людського зору,

включаючи навчання та можливість робити висновки та виконувати дії на основі візуальних даних. Сама ця сфера є філією штучного інтелекту (ШІ), метою якого є наслідування людського інтелекту. Сфера штучного інтелекту знаходиться на ранній стадії зародження з точки зору розвитку, і прогрес був набагато повільнішим, ніж очікувалося спочатку. Область аналізу зображень (також називається розумінням зображень) знаходиться між обробкою зображень і комп'ютерним баченням. Але однією з найскладніших задач, яка стоїть на сьогоднішній день є задача розділення зображення на окремі складові об'єкти. Дана задача є одночасно і дуже простою, і дуже складною. Все залежить наскільки селадне зображення було обране для сегментації, наскільки об'єкти на зображенні є подібними одне до одного і наскільки границя між ними є невизначеною. Тому задача розробки алгоритму сегментації на основі малоінформаційних областей є актуальною.

Метою роботи є розробка алгоритму сегментації кольорових зображень на основі аналізу малоінформативних областей.

Для досягнення даної мети ставились наступні завдання:

- провести аналізу та класифікацію існуючих цифрових зображень;
- провести класифікацію задач цифрової обробки зображень;
- здійснити огляд програмного забезпечення для обробки зображень з відкритим кодом;
- проаналізувати алгоритми сегментації цифрових зображень;
- розробити алгоритм сегментації цифрових зображень на основі аналізу малоінформативних областей;
- реалізувати програмну систему аналізу та опису цифрових зображень, провести її моделювання та порівняти її функціональні можливості з подібними програмами.

Об'єкт дослідження – процес обробки цифрових даних.

Предмет дослідження – методи і алгоритми сегментації в системах обробки цифрових зображень.

Наукова новизна одержаних результатів визначається наступним чином:

– проведено аналітичне дослідження та класифікацію методів та алгоритмів сегментації цифрових зображень, що дозволило виділити основні фактори впливу на їх роботу, а також розробити власний алгоритм сегментації зображень на основі аналізу малоінформативних областей зображення;

– розроблено алгоритм сегментації зображень на основі аналізу малоінформативних областей зображення, що дозволило збільшити рівень автоматизації та підвищити швидкодію процесу обробки зображень.

Практична цінність одержаних результатів полягає в тому, що:

– розроблено та здійснено експериментальне моделювання програмної системи аналізу та опису цифрових зображень на основі методів та алгоритмів обробки цифрових зображень, що дозволило програмно реалізувати та дослідити розроблені структури та алгоритми;

– реалізовано програмний модуль сегментації зображень на основі аналізу малоінформативних областей зображення з використанням запропонованих алгоритмів та функцій бібліотеки OpenCV.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційній роботі, підготовлено тези доповіді «Аналіз алгоритмів сегментації для систем автоматизованого аналізу зображень» обсягом 1 сторінка на V Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», а також «Аналіз алгоритмів попередньої обробки зображень систем автоматизованого моніторингу» обсягом 1 сторінка на V Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі».

1 ПРОГРАМНІ ТА АПАРАТНІ СИСТЕМИ ОБРОБКИ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1 Класифікація та характеристики цифрових зображень

Процедура отримання зображення перетворює візуальне зображення фізичного об'єкта та його внутрішні характеристики в набір числових даних, які можуть бути оброблені та проаналізовані блоком обробки системи. Отримання складається з п'яти кроків:

- освітлення;
- формування та фокусування зображення;
- виявлення або виявлення зображення;
- форматування аналогового електричного сигналу камери;
- цифрування або перетворення аналогового електричного сигналу в набір числових даних, придатних для обробки в цифровому комп'ютері.

Освітленість є ключовим параметром, що впливає на якість формування цифрового зображення. Об'єкт, зображення якого необхідно аналізувати, повинен бути освітлений власним джерелом освітлення без впливу освітлення навколишнього середовища, яке здебільшого проявляється як шум у даних зображення. Під час використання з макрооб'єктивами або світловими мікроскопами важливо стабільне джерело холодного галогенного світла. Іноді для деяких додатків цифрових зображень у галузі автоматизації промисловості використовуються спеціальні методи освітлення, наприклад, освітлення заднього освітлення, яке покращує межі об'єкта, стробоскопічне освітлення, яке усуває вплив навколишнього світла, або структурне освітлення зі спеціальним візерунком або сіткою, що використовується для полегшення розпізнавання об'єкта.

Формування та фокусування зображення. Зображення об'єкта фокусується на чутливому елементі відеокамери через відповідний оптичний елемент. Це може бути звичайний відеооб'єктив, макрооб'єктив або зум-об'єктив, великий

мікроскоп або звичайний світловий мікроскоп. Найважливішими факторами, пов'язаними з оптичною системою та відеокамерою, є збільшення та глибина різкості. Збільшення - це міра відносного розміру візуального зображення об'єкта у фізичному світі до розміру зображення, відформатованого на датчику, розташованому в площині детектора в камері. Це збільшення зазвичай називають основним збільшенням, і воно важливо для калібрування вимірювань. Збільшення відеосистеми є мірою відносного розміру об'єкта на площині детектора та об'єкта, показаного на моніторі, а загальне збільшення є продуктом обох. Глибина різкості – це простір над і під площиною об'єкта, де оптична система підтримує фокус зображення в допустимих межах. Для зміни глибини різкості зазвичай використовується регульована діафрагма камери. Чим менший отвір об'єктива, тим більша глибина різкості, але тим менша кількість світла, що передається датчику. Виявлення зображень. Камера має електрооптичний пристрій, який перетворює електромагнітне випромінювання із зображення фізичного об'єкта в електричний сигнал, що використовується блоком цифровізації зору. Зображення фокусується лінзами на датчику в камері. Сенсорний елемент, який може бути відеотрубкою або твердим сенсорним елементом стану, розташований на площині датчика, виробляє аналоговий електричний сигнал, що представляє видиме зображення. Сьогодні в більшості випадків використовуються твердотільні датчики ПЗС (пристрій зарядового зв'язку). Вони характеризуються розміром діагоналі (1/3", 1/2" або 2/3"), кількістю елементів пікселів (як правило, 500 x 582 пікселів для стандартних камер PAL), чутливістю (зазвичай від 2 до 25 люкс), та роздільною здатністю (від 300 до 600 горизонтальних ліній). Цей аналоговий сигнал, генерований датчиком, потім форматується відповідно до одного із стандартів відео.

Форматування аналогового електричного сигналу камери. Кольорові камери можуть мати один з таких відеовиходів:

- відеовихід RGB;
- відеовихід Y-C (супер VHS);
- композитний відеовихід (NTSC або PAL).

Вихідний відеосигнал RGB є основним вихідним сигналом, оскільки він спочатку виробляється пристроєм прийому. Червоний (*R*), зелений (*G*) і синій (*B*) відеосигнали подаються окремо на власних контактах роз'єму. Сигнал синхронізації, який визначає кінець рядка та кінець одного зображення, може бути накладений на один із кольорових каналів або наданий окремо як горизонтальна та вертикальна синхронізація. Вихідний відеосигнал *Y-C (S-VHS)* має два виходи:

- сигнал люмінесценції *Y*;
- хромований сигнал *C*.

Сигнал люмінесценції можна використовувати як звичайний монохроматичний, чорно-білий сигнал, а сигнал кольоровості несе інформацію про колір. Композитний відеовихід має лише один вихідний сигнал, який несе інформацію про люмінесценцію, кольоровість та синхронізацію (сигнал *VBS*). Американський (*NTSC*) і європейський (*PAL*) стандарти суттєво відрізняються. Американський *NTSC* (Національний комітет телевізійної системи) старший (1953). Він має частоту горизонтальної синхронізації 15,734 кГц і частоту вертикальної синхронізації 59,94 Гц. Європейський *PAL* (*Phase Alternation Line*) має різні частоти синхронізації (по горизонталі 15,625 кГц, по вертикалі 50 Гц), але також він відрізняється тим, як складається вихідний відеосигнал *VBS*.

Оцифрування. Оцифрування означає перетворення аналогового відеосигналу в цифровий відеосигнал, придатний для обробки цифрового комп'ютера. Цей процес зазвичай виконується спеціальним блоком, який називається фреймграббером. У системах на базі персонального комп'ютера фреймграббер в більшості випадків підключається до карти, яка повинна бути встановлена в один із слотів вільної шини комп'ютера. Він приймає аналоговий відеосигнал, дискретує його, вирівнює і перетворює в цифровий сигнал. Зазвичай фреймграбери мають власну відеопам'ять, де зберігається це цифрове зображення. Обробка цього цифрового зображення може здійснюватися за допомогою головного комп'ютерного процесора або спеціального графічного процесора, який іноді вбудовується в захват кадру, щоб підвищити потужність

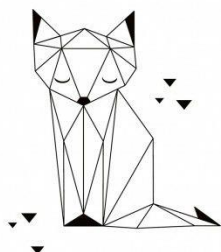
обробки всієї системи. Цифрове зображення з відеопам'яті може відображатися на моніторі комп'ютера та/або на додатковому 2-му допоміжному моніторі через спеціальний блок відображення відео кадрів. Під час отримання зображення можуть бути виконані деякі виправлення та оптимізація. Насамперед, це налаштування освітлення, контрасту та інтенсивності, корекція кольору насамперед червоним, зеленим та синім кольорами або за відтінком, насиченістю та інтенсивністю. Оскільки ці функції можуть бути застосовані також на етапі обробки зображень, ми пояснимо їх останнє, оскільки перед їх описом необхідно дати деякі пояснення основ цифрового зображення. Єдина відмінність між оптимізацією зображення під час отримання та під час обробки зображення полягає в тому, що в першому випадку змінюється аналоговий електричний відеосигнал, а в другому – перетворюється цифрове зображення, що зберігається у відеопам'яті.

Зображення може бути фотографією, картиною або чимось уявним, проте в світі комп'ютерів – це набір точок, які називаються «пікселями». Таке зображення зазвичай називають цифровим зображенням і формально визначається як масив пікселів, значення яких вказують на інтенсивність світлового потоку на елементі зображення, представленому цим пікселем.

Зображення можна визначити як двовимірну функцію $f(x,y)$, де x і y є просторовими (площинними) координатами, а амплітуда f у будь-якій парі координат (x,y) називається інтенсивністю або рівнем сірого зображення в цей момент. Коли x , y та значення інтенсивності f є кінцевими дискретними величинами, то це називається цифровим зображенням.

Кожен піксель має своє горизонтальне та вертикальне положення та своє значення. Позиції та значення є невід'ємними скалярами, діапазон яких залежить від характеристик одиниці цифровізації. Наприклад, засіб захоплення кадрів, що постачається з програмним забезпеченням Chronolab Color Vision, створює цифрове зображення з розміром 512x512 пікселів (у звичайному режимі), тому координати x і y можуть бути невід'ємними скалярами від 0 до 511. Також кожен канал захоплення кадрів має 8-бітовий аналог цифрове перетворення, тому

значення кожного пікселя може бути від 0 до 255, 0 представляє найтемнішу інтенсивність, а 255 — найсвітлішу інтенсивність. Можна виділити три різні види цифрових зображень (рисунок 1.1).



а) чорно-біле зображення



б) зображення сірого кольору



кольорове зображення

Рисунок 1.1 – Класифікація цифрових зображень за значенням кольору

Кожен піксель чорно-білого зображення, або його іноді називають двійковим зображенням, може мати одне з двох можливих значень, наприклад, для 0 або 255, але зазвичай це виражається логічними значеннями 0 і 1. Це найпростіше зображення, де об'єкти представлені своїми площами. Більшість вимірювань об'єкта можна було б виконати тільки на такому цифровому зображенні. Бінарне зображення можна отримати із зображення сірого або кольорового зображення за допомогою процесу, який називається.

Значення пікселів цифрового зображення зі значенням сірого можуть бути невід'ємним скаляром з діапазону, визначеного одиницею аналого-цифрового перетворення. Типовий діапазон – від 0 до 255 (256 значень сірого), де 0 означає чорний, а 255 – білий. Цифрові зображення сірого кольору можна отримати з монохромних камер або шляхом перетворення кольорів із кольорових цифрових зображень. Суб'єктивне сприйняття значень сірого не є лінійним і відповідає кривій, відомій як гамма-корекція (докладніше див. у розділі Обробка зображень). Кольорові цифрові зображення мають три площини зображення,

кожна з яких відповідає одному основному кольору: червоний (R), зелений (G) і синій (B). Кожен піксель кольорового зображення складається з трьох кольорових пікселів: червоного пікселя, зеленого пікселя і синього пікселя, кожен з яких має відповідне значення інтенсивності. Площини червоного, зеленого або синього кольорів мають таку ж структуру, як і площині зображення сірого зазвичай 256 інтенсивностей для кожного кольору. Поєднання різних інтенсивностей пікселів RGB, кожен піксель зображення може дати одне з $256 \times 256 \times 256 = 16777216$ різних значень кольору. При визначенні значення пікселя кольорового зображення в місці (x, y) потрібно задати триплет у вигляді (червоний, зелений, синій), наприклад $(0, 255, 255)$, що означає мінімальну інтенсивність для червоного та максимальну інтенсивність для зеленого і синій, що відповідає чисто жовтому відтінку. Кольоровий піксель зображення буде виглядати як піксель сірого кольору, якщо всі основні кольори мають однакове значення інтенсивності: чорний відповідає $(R, G, B) = (0, 0, 0)$, білий — $(R, G, B) = (255, 255, 255)$, і, наприклад, $(R, G, B) = (127, 127, 127)$ дасть деяку середню інтенсивність сірого. Іноді значення пікселів зображення зручніше виразити не основними кольорами червоного, зеленого та синього, а перетвореними значеннями відтінку (H), насиченості (S) та інтенсивності (I). Відтінок відповідає інформації про чистий колір. Весь спектр охоплюється значеннями відтінків від 0 до 360 (червоний 0, зелений 120, синій 240). Насиченість визначає близькість кольору до значень сірого. Чистий колір має насиченість, рівну 1, а значення сірого – 0. Інтенсивність або яскравість вимірюють загальну інтенсивність світла. Він може бути від 0 до 255, або в системі відносної інтенсивності від 0 до 1. Інтенсивність об'єкта не залежить від інтенсивності навколишніх об'єктів. Інтенсивність – це об'єктивна міра розподілу світла на зображенні. Суб'єктивним показником розподілу світла на зображенні є яскравість або її іноді називають видимою яскравістю. Яскравість об'єкта – це сприйнята яскравість і залежить від яскравості оточення. Два об'єкти з різним оточенням можуть мати однакову яскравість, але різну яскравість. Кольоровий простір RGB може бути схематично показано кольоровим кубом (рисунок 1.2).

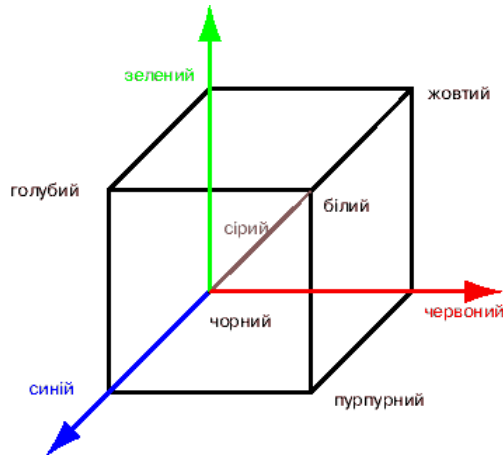


Рисунок 1.2 – Просторова модель кольорового простору RGB

Кути куба відповідають елементарним кольорам (червоний, зелений, синій), їх доповненням (блакитний, пурпурний, жовтий) та чорному та білому. Значення сірого кольору розташовані на головній діагоналі. Колірний простір HSI може бути схематично показано подвійними конусами.

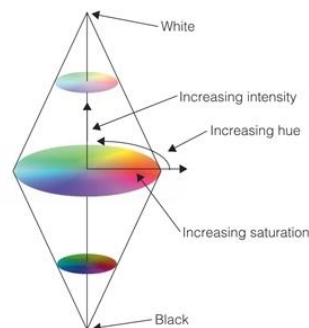


Рисунок 1.3 – Просторова модель кольорового простору HSI

Нижня і верхня вершини відповідають чорному і білому, а значення сірого розташовуються на лінії між ними (насиченість = 0). Чисті кольори розташовані на зовнішніх колах. Кожне значення пікселя кольорового зображення можна знайти всередині куба RGB або подвійного конуса HSI. Якщо зображення має значення одиниці сірого кольору, усі значення пікселів лежать на лініях значень сірого кольору. Чисті кольори розташовані на зовнішніх поверхнях куба RGB або подвійного конуса HSI. Одним із недоліків моделі RGB і частково моделі HSI

для визначення кольорового зображення є її неоднорідність; однакова відстань у колірному просторі RGB зазвичай не відповідає однаковій різниці у сприйнятті кольору. Тому призначені деякі інші кольорові моделі. Однією з них є колірна модель YIQ американського стандарту NTSC для композитного відеосигналу, яка, є лінійним перетворенням моделі RGB з інформацією про яскравість, закодованою в компонент Y , а кольоровістю в I і Q . Це дозволяє відобразити кольоровий відеосигнал на чорно-білих моніторах за допомогою компонента Y . Параметр I містить інформацію про оранжево-блакитний відтінок, тоді як параметр Q містить зелено-пурпурні відтінки. Досить схожою є колірна модель європейського стандарту кольорового відео PAL. Відеопам'ять для зберігання кольорових цифрових зображень має бути втричі більша за відеопам'ять для зберігання цифрових зображень із значенням сірого з такою ж кількістю рівнів інтенсивності. Наприклад, для одного цифрового зображення 512x512 пікселів із значенням сірого з 256 рівнями сірого потрібно приблизно 262 Кбайт відеопам'яті, а для кольорового цифрового зображення – 786 Кбайт відеопам'яті. Також час обробки кольорових зображень утричі довший, і через це системи кольорового цифрового відео є складнішими та дорожчими. Кольорові зображення, описані раніше, відомі як зображення істинного кольору. Їм потрібно багато місця для зберігання. Щоб зменшити обсяг пам'яті, що зберігається, пропонується інший тип кольорових зображень, іменованих кольоровими зображеннями. Вони зменшили колірну палітру з 16 777 216 кольорів до 256 кольорів. Це означає, що для зберігання кожного значення пікселя кольорових зображень потрібно лише 8 біт пам'яті, а справжнім кольорам потрібно втричі більше 24 біт. Для зображень із кольоровою картою потрібен той самий простір пам'яті, що й зображення рівня сірого. У зображеннях істинного кольору кожне значення пікселя задається червоним, зеленим, синім триплетом, а в кольорових зображеннях – індексом палітри (або його іноді називають номером палітри). Кожен індекс палітри має відповідне значення кольору, яке відображається на екрані дисплея. Коли визначають палітру кольорів і хочуть перетворити зображення істинного кольору в

зображення з відображенням кольорів, спеціальна процедура шукає колір палітри, який є найближчим до кольору істинного зображення.

1.2 Технології цифрової обробки зображень

Технологія цифрових зображень швидко розвивається, мультидисциплінарна технологія, яка використовує знання з різних галузей для створення системи, яка може бути використана в різних сферах людської діяльності, наприклад, у медицині, біології, хімії, інженерії, промисловій автоматизації та інспекції, безпеці, кримінології. Технологія цифрових зображень включає отримання зображення, оцифровку зображення, що означає його перетворення в цифрове зображення, цифрову обробку зображень та цифровий аналіз зображень. Цю технологію можна використовувати у всіх тих випадках, коли рішення ґрунтуються на якісній або кількісній інформації, отриманій із зображення. Одним із типових прикладів є класифікація частинок, де частинки на зображенні мають бути класифіковані на основі морфологічного параметра окремої частинки (площа, периметр, індекс круговості або щось подібне). Вхідною інформацією до такої системи є зображення різних частинок, які знімає відеокамера через відповідний об'єктив камери. Вихідною інформацією є розподіл параметрів частинки. Більше того, система могла б розпізнати кожну окрему частинку, якщо деякі знання про значення типових параметрів частинки впровадити в базу знань системи. Для класифікації та розпізнавання матеріалу можна було б використовувати інформацію про текстуру, можна було виявити та розпізнати патологічні клітини за допомогою інформації про колір і форму, можна було ідентифікувати нерівності об'єктів на виробничій лінії, порівнюючи їх зі звичайними, щоб виключити фальшиву продукцію з виробничої лінії. Усі ці завдання сьогодні можна було б виконати за допомогою складного апаратного та програмного забезпечення з більшою

швидкістю та точністю. Важливо знати, що вартість такого складного і складного обладнання вже не така висока, як була донедавна.

Весь процес застосування цифрових зображень можна розділити на три основні види діяльності (рисунок 1.4).

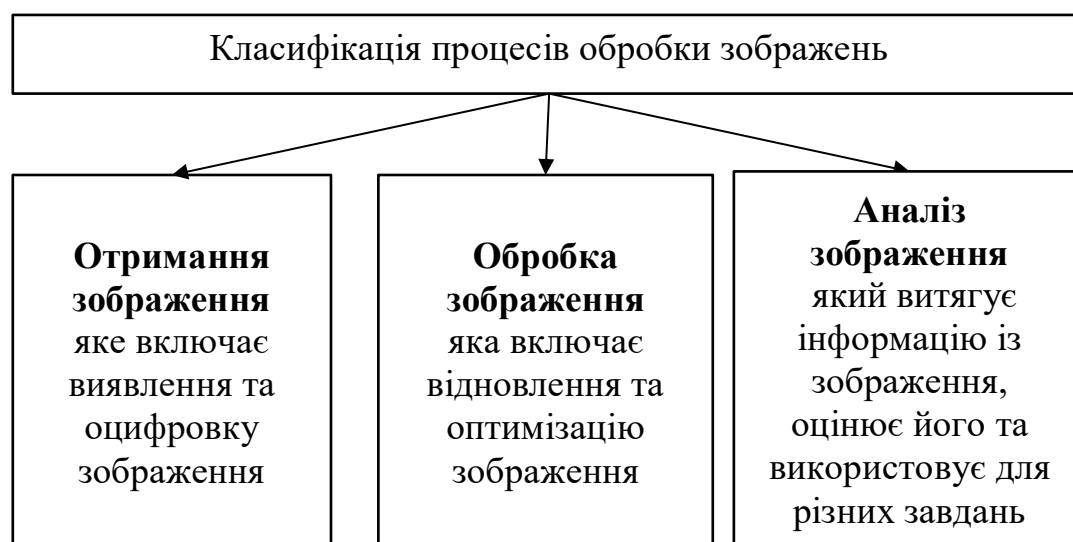


Рисунок 1.4 – Класифікація процесів обробки зображень

Ця послідовність функцій зазвичай використовується під час виконання типової процедури аналізу зображення. На основі цієї послідовності функцій можна було б «запрограмувати» багато конкретних завдань. Наприклад, його можна використовувати для моніторингу патологічних змін в органах шляхом оцінки біологічних зрізів, диференціації лейкоцитів в аналізі мазка крові, аналізу матеріалу шляхом виявлення та вимірювання відсотка включень у металевому зразку, аналізу фертильності шляхом аналізу кінетики сперматозоїдів, ангіографічний аналіз послідовності рентгенівських зображень після введення контрастної речовини, макроскопічний аналіз бактеріальних колоній у чашках Петрі тощо. Різні окремі функції можуть бути пов'язані разом, щоб сформувати повне завдання аналізу зображення, яке потім можна було б використовувати для прийняття відповідних рішень або використовувати для автоматичного формування відповідних дій. Але щоб оптимально використовувати цю нову

дивовижну технологію, принципи формування та отримання зображень, обробки зображень та аналізу зображень мають бути зрозумілі й відомі.

Під час запису зображень процедуру оптимізації та покращення можна виконати під час завантаження зображення в пам'ять зображення. Основними процедурами оптимізації є:

- основні кольори;
- яскравість;
- контраст;
- насиченість;
- корекція відтінку.

Ця обробка зображення виконується на аналоговому відеосигналі в порівнянні з обробкою цифрового зображення, яка виконується для цифрових зображень, що зберігаються в цифрових відеопам'яті. За допомогою корекції основних кольорів можна підкреслити або збільшити інтенсивність червоного, зеленого або синього. Ефект такий же, як і застосування кольорових фільтрів перед об'єктивом камери.

Корекція яскравості робить зображення світлішим або темнішим. Використовується для виправлення неправильного освітлення, але краще спробувати покращити систему освітлення, ніж використовувати цю функцію. Найнижча корекція яскравості дає повністю чорне зображення, а найвища корекція яскравості дає повністю біле зображення.

Корекція контрасту посилює або збільшує різницю в інтенсивності світла двох областей. Зображення є контрастним, якщо ця різниця більша, і менш контрастним, якщо вона менша. Чорно-біле (бінарне) зображення є результатом застосування максимальної корекції контрасту, а повністю однорідне сіре зображення є результатом застосування мінімальної корекції контрасту.

Корекція насиченості використовується лише для кольорових зображень. Зображення з високою насиченістю має високу інтенсивність кольорів, а зображення з низькою насиченістю досить близьке до зображення із значенням сірого. Інколи застосовується корекція відтінку, яка використовується для зміни

палітри відтінків кольорового зображення. Завдяки цьому червоний стає зеленим, зелений стає синім і подібні. Подібний ефект можна отримати за допомогою корекції колірною діапазону/зміщення. Іншою можливою оптимізацією зображення під час отримання є усереднення зображення та інтеграція зображення. Усереднення зображення використовується у випадках, коли зображення пошкоджене шумом. Шум у зображенні розпізнається за значеннями інтенсивності випадкових пікселів зображення, які відхиляються від реального значення. Завдяки усередненню зображення випадковий шум можна значно зменшити. Інтеграція зображень використовується для слабких сигналів (наприклад, темне поле або флуоресцентна мікроскопія). У цій процедурі підсумовується кількість окремих зображень, тому інтенсивність кінцевого зображення набагато вища, ніж для окремого кадру зображення.

Немає чітких меж у просторі від обробки зображень на одному кінці до комп'ютерного зору на іншому. Однак одна корисна парадигма полягає в тому, щоб розглянути три типи комп'ютеризованих процесів у цьому континуумі:

- процеси низького рівня;
- процеси середнього рівня;
- процеси високого рівнів.

Процеси низького рівня включають примітивні операції, такі як попередня обробка зображення для зменшення шуму, підвищення контрастності та різкості зображення. Низькорівневий процес характеризується тим, що і його входи, і виходи є зображеннями.

Обробка зображень середнього рівня включає такі завдання, як сегментація (поділ зображення на області або об'єкти), опис цих об'єктів, щоб привести їх до форми, придатної для комп'ютерної обробки, і класифікація (розпізнавання) окремих об'єктів. Процес середнього рівня характеризується тим, що його вхідні дані зазвичай є зображеннями, але його вихідними є атрибути, витягнуті з цих зображень (наприклад, ребра, контури та ідентичність окремих об'єктів). Нарешті, обробка вищого рівня передбачає «осмислення» ансамблю розпізнаних об'єктів, як при аналізі зображень, і, на дальньому кінці

континууму, виконання когнітивних функцій, які зазвичай пов'язані з людським зором.

Виходячи з попередніх коментарів, очевидно, що логічним місцем перекриття між обробкою зображень та аналізом зображень є область розпізнавання окремих областей або об'єктів на зображенні.

Цифровий аналіз зображень може:

- візуально покращити та оптимізувати зображення,
 - підготувати зображення для кількісного аналізу та вимірювання,
 - усунути артефакти зображення,
 - автоматично оцінювати результати аналізу зображень;
 - дає об'єктивну, кількісну оцінку та вимірювання зображення,
 - підвищує точність вимірювань,
 - запроваджує стандартизацію процедур вимірювання,
 - це збільшує швидкість, з якою можна проводити вимірювання,
 - дозволяє повторювати та перевіряти всі вимірювання, оскільки аналізоване зображення можна легко зберігати та
- кінцеві результати вимірювань можна було б автоматично та легко оцінити, наприклад, за допомогою статистичних даних та графічного представлення.

Цифрова обробка зображень змінює значення цифрових зображень, що зберігаються у відеопам'яті, щоб підготувати їх до аналізу зображення. Обробку даних цифрового зображення можна розділити на дві основні категорії:

- унарні операції, які діють на значення одного пікселя зображення в один момент часу;
- бінарні операції, які діють на декілька значень пікселів одного зображення або кількох зображень одночасно. момент.

Унарні операції точки за точкою, або іноді їх називають програмним LUT, передбачають створення нового цифрового зображення шляхом зміни значення пікселя в одному місці на основі глобального правила, застосованого до кожного місця на вихідному зображенні. Процес включає в себе взяти значення пікселя

$p(x,y)$ заданого (x,y) розташуванням, модифікувати його простим лінійним або нелінійним оператором F і помістити нове значення пікселя $q(x,y)$ на те саме (x,y) розташування. Цей процес повторюється для значення пікселя наступного розташування і продовжується на всьому зображенні або його області інтересу (ROI), що тепер називається обробкою ROI .

Підвищення контрастності – це лінійна операція по точках, яка використовується для покращення якості зображення, коли контраст зображення занадто слабкий. У цьому випадку значення інтенсивності не рівномірно розподілені по всьому діапазону інтенсивності. Вони обмежені частиною можливого діапазону. Динамічний діапазон вихідного зображення, який є результатом розрахунку, що розподіляється на максимально можливі значення інтенсивності системи. Для кольорових зображень покращення контрасту з різними значеннями відсікання для кожного основного кольору можна було б виконувати окремо, або однакові значення відсікання, розраховані з гістограми інтенсивності, можна було б застосувати для всіх площин основного кольору. Ефект застосування оператора значення сірого та підсилення колірною контрасту показано на таблицях в кінці цього вступу. Підвищення контрастності є більш потужною операцією, ніж просте регулювання контрасту, яке можна виконати за допомогою подібної функції оператора. За рахунок налаштування контрасту діапазон значень пікселів зображення розтягується або стискається навколо середини. Значення відсікання посилення контрасту можуть бути будь-якими, враховуючи, що $p1 < p2$. На відміну від коригування контрасту $p1$ і $p2$, симетрично розташовані навколо середньої вхідної інтенсивності. Наприклад, якщо у нас є 256 різних вхідних інтенсивностей (вхідне зображення 8-бітового сірого), ніж інтенсивність середньої точки дорівнює 128. Якщо $p1$ вибрано як 50, то $p2$ має бути $128 + (128 - 50) = 206$. Регулювання контрасту збільшує контраст в центрі та покращує контраст навколо будь-якого значення вхідної інтенсивності. Також регулювання контрасту має можливість зниження контрастності. Підвищення контрасту (розтягування) використовується, коли зображення виглядає занадто плоским, без яскравих світлих або темних

кольорів. Зменшення контрасту (зтискання) використовується, коли світлі та темні зображення занадто екстремальні.

Налаштування інтенсивності та яскравості наступні досить часто використовувані поточкові операції. Вони застосовуються, коли зображення занадто темне або занадто світле. Обидва вони реалізуються шляхом зсуву вихідної операторної функції. У разі налаштування інтенсивності функція оператора зміщується вгору (для освітлення зображення) або вниз (для затемнення зображення). Математично коригування інтенсивності реалізується шляхом додавання або віднімання постійного позитивного значення до кожного вхідного пікселя інтенсивності. Регулювання яскравості також реалізується шляхом зсуву функції оператора, але цього разу вліво (для освітлення зображення) або вліво (для затемнення зображення). Звичайно, діапазон значень в обох випадках обмежений, наприклад, інтервалом $[0,255]$. Важливо підкреслити, що після застосування коригування контрасту, інтенсивності або яскравості деякі інтенсивності вхідного зображення будуть втрачені, чого не буде у випадку, якщо їх налаштувати під час отримання зображення. Тому, якщо зображення потребує корекції інтенсивності, яскравості або контрасту, краще застосовувати їх під час отримання зображення, а не під час обробки зображення, але найкращим рішенням, особливо для корекції інтенсивності або яскравості, є ретельне регулювання освітлення.

1.3 Програмні засоби для обробки цифрових зображень

Створення структури програмного забезпечення передбачає аналіз існуючих програмних рішень в даному напрямку, тому було проаналізовано безкоштовні програми-аналоги з відкритим кодом. Це надасть можливість визначити основні функціональні вимоги та структурні особливості програм даного типу.

ImageJ – це програма обробки зображень на основі Java. ImageJ, мабуть, найвідоміше і найдавніше програмне забезпечення з відкритим кодом для аналізу зображень. Незважаючи на те, що програма настільки широко використовується, ImageJ є експериментальною системою і розробники не несуть жодної відповідальності за її використання іншими сторонами. Для використання в регульованому середовищі кінцевому користувачеві знадобиться окрема перевірка та перевірка.

Протягом багатьох років було розроблено великий набір плагінів ImageJ, що охоплюють більшість областей аналізу зображень. Багато з них мають простий графічний інтерфейс користувача (GUI), але деякі вимагають навичок програмування для використання.

ImageJ сумісний з більш ніж 150 різними форматами зображень, які називаються. Планове використання плагінів – це в основному швидке створення прототипів і тестування алгоритмів обробки цифрових слайдів, спрямованих на дослідження.

Fiji – це дистрибутив ImageJ. Це «готовий до використання» пакет плагінів ImageJ для використання в науках та дослідницьких цілях. Проект Fiji був створений для підтримки встановлення та обслуговування одного з найскладніших плагінів ImageJ, TrakEM2, який надає комплексні рішення для керування, реєстрації, сегментації та анотації великих наборів даних електронної мікроскопії. Згодом до Fiji приєдналися інші вдосконалені плагіни для аналізу зображень у галузі науки про життя. На відміну від ImageJ, ці плагіни підбираються та організовані за категоріями в меню плагінів, що робить їх більш цілеспрямованими та простішими у використанні. Програмне забезпечення в першу чергу орієнтоване на дослідників з мінімальними навичками роботи з комп'ютером, але оскільки функціональні можливості ImageJ можна легко розширити за допомогою плагінів, воно також було привабливим для дослідників з навичками розробки програмного забезпечення.

QuPath – це комплексна безкоштовна настільна програма з відкритим вихідним кодом, розроблена спеціально для аналізу цифрових зображень. Вона включає в себе зручний засіб перегляду з розумними інструментами анотації, які використовують інформацію про пікселі на слайді для прискорення процесу та підвищення точності анотації та широкі інструменти візуалізації.

QuPath пропонує як готові до використання алгоритми штучного інтелекту для вирішення таких поширених завдань як пошук та виділення об'єктів (наприклад, автоматичне виявлення однорідних областей), так і будівельні блоки, які можна з'єднати для створення користувацьких робочих процесів і пакетної обробки зображень. QuPath дозволяє розробникам додавати власні розширення для вирішення нових завдань, впровадження нових додатків та обміну даними з існуючими інструментами, такими як ImageJ і MATLAB.

Функціональні можливості QuPath включають:

- створення мультислайд-проектів;
- виявлення об'єктів та обчислення функцій;
- класифікація об'єктів, що піддається навчанню з використанням класифікатора випадкових дерев машинного навчання;
- пакетна обробка;
- обмін даними з існуючим програмним забезпеченням і платформами (наприклад, ImageJ і Matlab);
- скриптовий аналіз даних;
- швидке створення, візуалізація та експорт просторових, морфологічних та інтенсивних ознак.

Ilastik – це простий у використанні безкоштовний інструмент з відкритим кодом, який дозволяє користувачам без досвіду обробки зображень виконувати сегментацію та класифікацію 2, 3 та 4D зображень уніфіковано. За допомогою класифікатора випадкових лісів ilastik вивчає мітки, надані користувачем через

зручний графічний інтерфейс. На основі цих міток ilastik застосовує сегментацію для певної проблеми.

Pastik підтримує до трьох просторових плюс один спектральний вимір і використовує всі виміри при розрахунку ознак. Програма забезпечує зворотний зв'язок у режимі реального часу, що дозволяє користувачеві в інтерактивному режимі визначати результати сегментації на льоту.

Програма здатна покращити класифікацію на основі відгуків користувачів. Прогноз класифікатора вказує, де класифікатор все ще може помилятися. Це показано на накладенні, яке називається картою невизначеності, яка виділяє регіони, у яких прогноз є найбільш неоднозначним. Він спрямовує користувача до складних областей зображення, де класифікатор вважає нові мітки найбільш інформативними. Після навчання ilastik на наборі репрезентативних зображень його можна використовувати для автоматичної обробки дуже великої кількості зображень у режимі пакетної обробки або за допомогою конвеєра CellProfiler. Є можливість інтегрувати нові модулі аналізу через механізм плагіна.

Програма не призначена для обробки цілих зображень слайдів, але при патології її використовували для виявлення **мітотичних фігур у полях високої потужності**.

Розроблено для дослідників без навичок програмування на комп'ютері, але розуміння обробки зображень корисно.

Orbit Image Analysis - це безкоштовне програмне забезпечення з відкритим вихідним кодом для кількісної оцінки широкоформатних зображень, таких як цілі слайдові зображення тканин. Воно може завантажувати зображення з локального диска або підключатися до сервера зображень та може обробляти зображення на локальному комп'ютері або в кластері.

Орбіту можна використовувати для визначення регіону інтересу. Обробка можлива завдяки конвеєру обробки на основі плиток і роботі з різними роздільними можливостями зображення. Пізніше розбиті підобласті можна надіслати для обробки окремих комірок у кожній плитці. Особливості

програмного забезпечення включають багато вбудованих алгоритмів аналізу зображень для кількісної оцінки тканин із застосуванням методів машинного навчання, сегментації об'єкта/клітини та класифікації об'єктів.

Виявлення об'єктів засноване на навчанні класів переднього/фонового плану. Об'єкти, що перекриваються, наприклад, кластери клітин можна розділити пізніше. Характеристики об'єктів (наприклад, коефіцієнти форми, площа, інтенсивність) можуть бути обчислені та повідомлені або використані для класифікації об'єктів.

Сегментовані об'єкти можна класифікувати за такими ознаками, як розмір, фактори форми чи інтенсивність. Наприклад, він може розрізняти забарвлені та незабарвлені клітини або між круглими та гострими об'єктами. Ця класифікація заснована на машинному навчанні; користувач може вказати класи, вибравши приклади ROI можна визначити за допомогою анотацій вручну або за допомогою карти виключень, яку можна навчати, яка визначає «хороші» регіони, де слід виконувати аналіз. Обидва способи визначення рентабельності інвестицій можна комбінувати.

Для моделі фіброзу легенів у щурів і для виявлення клубочків у щурів і мишей Orbit має готові до використання, **попередньо навчені моделі**.

Користувачі з навичками розробки програмного забезпечення можуть створювати сценарії та розширення та застосовувати їх до всіх плиток у межах ROI. Програма також сумісна з ImageJ і її плагіни можна використовувати для аналізу.

Незважаючи на те, що Orbit оптимізовано для WSI (наприклад, SVS, NDPI, NDPIS, SCN, AFI, CZI, IMS, VSI, ETS, SLD, TIF, TIFF, TF2, TF8, BTF), програмне забезпечення сумісне з усіма форматами від бібліотеки Bioformats.

Створено для дослідників без навичок роботи з комп'ютером, але дослідники з навичками розробки програмного забезпечення можуть додавати власні розширення.

Ісу, безкоштовне програмне забезпечення з відкритим вихідним кодом, засноване Institut Pasteur та France-BioImaging. Як зазначено на його веб-сайті, це

«спільний фотошоп, присвячений аналізу зображень». Спільна робота означає, що користувачі з навичками розробки програмного забезпечення можуть додавати функції до програми. У Ісу користувачі можуть візуалізувати, анотувати та кількісно оцінювати дані біовізуалізації. Він розроблений як спільна платформа для вчених з аналізу зображень, які можуть розробляти нові алгоритми та вчених, які шукають інтуїтивно зрозумілий інструмент для аналізу зображень.

GUI розроблений на базі стандартних компонентів з якими користувачі вже знайомі, наприклад, основна панель «Стрічка» нагадує за своїм виглядом офіс Suite (Microsoft) і забезпечує доступ до всіх функціональних можливостей Ісу.

Аналіз зображень стає все більш затребуваним у дослідженнях. Багато компаній пропонують цінне програмне забезпечення та/або послуги в цій ніші. Але аналіз зображень не повинен бути дорогим. Існує багато безкоштовних програм з відкритим вихідним кодом зі складними можливостями аналізу зображень.

1.4 Постановка задач дослідження

Під час аналітичного огляду відомих технологій, алгоритмів та засобів обробки зображень було досліджено та проведено класифікацію основних типів цифрових зображень. Проведено аналізу сфер застосування та основних принципів функціонування підходів цифрової обробки зображень. Проведено огляд програмного забезпечення з відкритим кодом для вирішення задач обробки зображень.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- провести аналізу та класифікацію існуючих цифрових зображень;
- провести класифікацію задач цифрової обробки зображень;
- здійснити огляд програмного забезпечення для обробки зображень;

- проаналізувати алгоритми сегментації цифрових зображень;
- розробити алгоритм сегментації цифрових зображень на основі аналізу малоінформативних областей;
- реалізувати програмну систему аналізу та опису цифрових зображень, провести її моделювання та порівняти її функціональні можливості з подібними програмами.

1.5 Висновки до розділу

Проаналізовано та проведено класифікацію існуючих типів цифрових зображень на основі їх характеристик, що дозволило виділити групи зображень для обробки в системі аналізу цифрових зображень.

Проаналізовано основні задачі цифрової обробки зображень та проведено їх групування на основі подібності отриманих результатів, що дозволило виділити основні напрямки розвитку цифрової обробки зображень.

Проведено огляд програмних систем обробки зображень з відкритим кодом на основі аналізу їх структури та набору функціональних можливостей, що дозволило виділити структурні блоки та алгоритми які є головними в програмах даного класу.

2 МЕТОДИ ТА АЛГОРИТМИ ОБРОБКИ ТА ОПИСУ ЦИФРОВИХ ЗОБРАЖЕНЬ

2.1 Методи та алгоритми сегментації цифрових зображень

Основною метою сегментації зображення є незалежне від домену розділення зображення на набір незалежних областей, які візуально відрізняються і мають різні значення щодо деяких характеристик або обчислюваних властивостей, таких як рівень сірого, текстура або колір, щоб забезпечити легкий аналіз зображення. Двома основними властивостями пікселів по відношенню до їх локального сусідства є розривність і подібність, які використовуються в багатьох методах сегментації. Значення пікселів змінюється від 0 до 1, якщо воно належить об'єкту. Методи сегментації, які відносно пов'язані з властивістю розриву пікселів, приймаються як граничні або «методики на основі країв», а ті, що ґрунтуються на подібності або однорідності, є методами на основі регіонів. На жаль, технології на основі краю та регіону не завжди дають бажаних результатів. Використання сегментації зображень здійснюється на різних платформах або в різних програмах, але використання одного методу не може дати бажаного результату. Це все тому, що зображення, що містять різну властивість, а також деякі інші фактори, такі як звук, яскравість тощо, роблять акцент на зображеннях, і неможливо застосувати єдиний метод сегментації, а також єдину техніку оцінки для всіх типів зображень. Сегментація зображень знизу вгору залишається складною проблемою, незважаючи на постійні дослідження, які тривають кілька десятиліть. За останні роки було запропоновано і також продемонстровано багато алгоритмів на кількох зображеннях, але справжнє питання полягає в тому, як це порівняння цих алгоритмів один з одним не було розглянуто належним чином. Здебільшого це пов'язано з недоступністю алгоритмів протягом цього часу в порівнянні з останніми роками відповідного стандартного набору зображень і пов'язаної реальної інформації, а також через відсутність загальнодоступних реалізацій

основних алгоритмів сегментації. Класифікація існуючих алгоритмів сегментації на дена на рисунку 2.1.

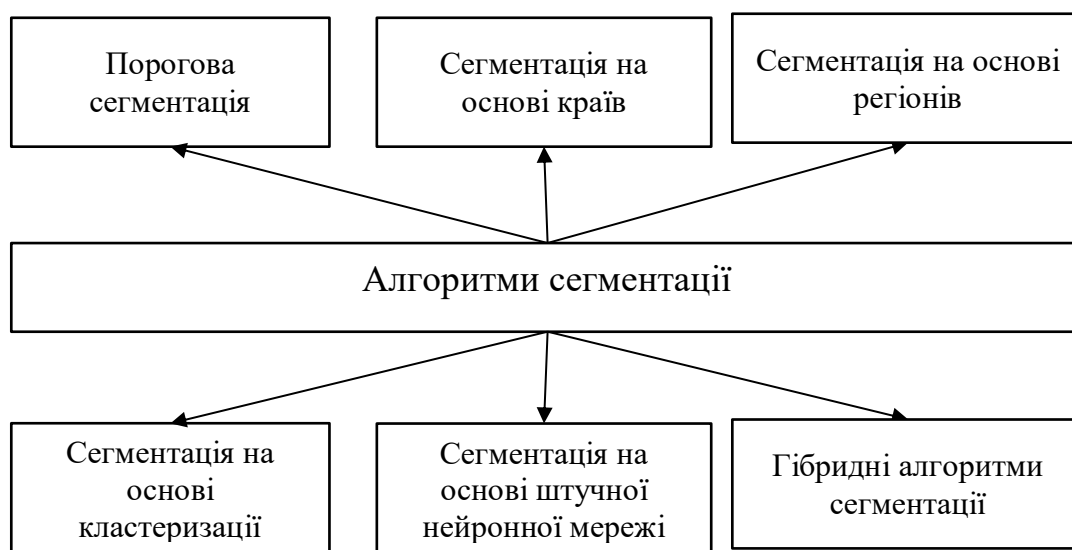


Рисунок 2.1 – Класифікація алгоритмів сегментації

Порогова сегментація зображень – це проста форма сегментації зображень. Це спосіб створити двійкове або багатоколірне зображення на основі встановлення порогового значення інтенсивності пікселів вихідного зображення.

У цьому процесі порогового значення розглянемо гістограму інтенсивності всіх пікселів на зображенні. Потім встановимо поріг для поділу зображення на частини. Наприклад, враховуючи пікселі зображення в діапазоні від 0 до 255, встановлюємо поріг 60. Таким чином, усі пікселі зі значеннями, меншими або рівними 60, отримають значення 0 (чорний), а всі пікселі зі значенням більше ніж 60 буде надано значення 255 (білий).

Розглядаючи зображення з фоном і об'єктом, можемо розділити зображення на області на основі інтенсивності об'єкта та фону. Але цей поріг має бути ідеально встановлений, щоб розділити зображення на об'єкт і фон.

Глобальне порогове значення: у цьому методі використовуємо бімодальне зображення. Бімодальне зображення – це зображення з 2 піками інтенсивності на графіку розподілу інтенсивності. Один для об'єкта, інший для фону. Потім виводимо порогове значення для всього зображення і використовуємо цей

глобальний поріг для всього зображення (рисунок 2.2). Недоліком цього типу порогу є те, що він дуже погано працює при поганому освітленні зображення.

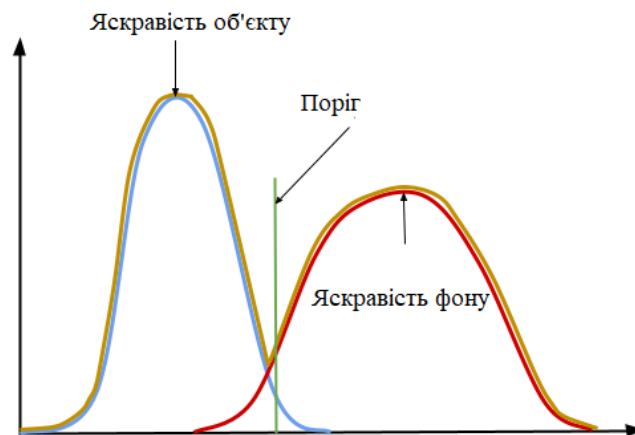


Рисунок 2.2 – Встановлення автоматичного порогу

Встановлення порогу вручну: При виборі порогового значення в ручному режимі не завжди можна правильно підрахувати кількісні властивості того чи іншого об'єкту, в загальному випадку даний підхід оптимально підійде при умові, що параметри об'єктів які знаходяться на зображенні та параметри фону кардинально відрізняються один від одного. Наприклад, чорна деталь на білому фоні полотна транспортера. В інших випадках встановлення фону відбувається за рахунок багаторазового повторення експериментів з метою отримання оптимального, на погляд користувача результату.

Адаптивне порогове значення: щоб подолати ефект освітлення, зображення поділяється на різні субрегіони, і всі ці області сегментуються за допомогою порогового значення, розрахованого для всіх цих регіонів. Потім ці субрегіони об'єднуються для відображення повного сегментованого зображення. Це допомагає до певної міри зменшити ефект освітлення.

Оптимальне порогове значення. Метод оптимального порогового значення можна використовувати, щоб мінімізувати неправильну класифікацію пікселів, що виконується сегментацією.

Локальне адаптивне порогове значення: через різницю в освітленні пікселів на зображенні глобальне порогове значення може мати труднощі з сегментацією зображення. Таким чином зображення розбивається на менші підгрупи, а потім виконується адаптивне порогове значення цих окремих груп. Після окремої сегментації цих підгруп усі вони об'єднуються, щоб утворити завершене сегментоване зображення вихідного зображення. Отже, гістограма підгруп допомагає забезпечити кращу сегментацію зображення.

Сегментація на основі порогів. Для сегментації зображення використовуються пороги та зрізи. Вони застосовуються безпосередньо до зображення, але також можуть бути об'єднані з техніками попередньої та післяобробної обробки. Сегментація на основі порогів не вимагає попередньої інформації про зображення. Це просто в порівнянні з іншими. Сегментація на основі порогів, яка використовується для реалізації, є порівняно швидкою та простою. А також його можна використовувати для реалізації в реальному часі, але сегментація на основі порога не буде добре працювати для зображення з широкими та плоскими долинами або без правильного піку. Пороговий метод дуже чутливий до шуму. А також неправильний вибір порогу може призвести до надмірної або недостатньої сегментації.

Сегментація на основі країв – ця методика використовується для виявлення країв у зображенні, які, як передбачається, представляють межі об'єкта, і використовується для ідентифікації об'єктів.

Сегментація на основі регіонів – тут метод на основі країв може спробувати знайти межі об'єкта, а потім знайти сам об'єкт, заповнивши їх, а техніка на основі регіонів використовує протилежний підхід, починаючи з середини об'єкта. а потім «зростає» назовні, поки не буде досягнуто межі об'єкта. Дає найкращий результат у порівнянні з будь-якими іншими методами сегментації. Також дуже гнучкий вибір між інтерактивною та автоматичною технікою для сегментації зображення, яка перетікає від внутрішньої області до зовнішньої, чітко генеруючи межі об'єкта. Це також дає точний результат, ніж будь-який інший метод, коли все насіння відібрано належним чином. Це

порівняно дороге в пам'яті або також за часом обчислень. Важко визначити критерії зупинки для сегментації.

Хоча техніка кластеризації іноді використовується як синонім для методів сегментації зображень, її також використовують, щоб виразити методи, які в основному використовуються в дослідницькому аналізі даних високовимірних моделей, які підлягають вимірюванню. У цьому контексті техніка кластеризації намагатиметься згрупувати однакові шаблони разом, які подібні в деяких формах. Цей тип цілі дуже схожий на те, що намагаємося зробити, коли маємо виконати сегментацію зображень, а також деякі методи кластеризації можна легко застосувати для сегментації зображення. Метод K -середніх обчислювально швидше для порівнянь малих значень. Він зменшує помилкові вкраплення та усуває зашумлені плями, а також стає більш однорідним, отримують чіткі області результату. Важко дати значення k , коли є фіксована кількість кластерів. Він також дуже чутливий до умови ініціалізації номера та центру кластера. Але помилка в тому, що він погано працює, коли використовуються широкі кластери. Це відносна складний та трудомісткий метод обробки цифрових зображень.

Алгоритми кластеризації - це неконтрольовані алгоритми, які допомагають знайти на зображенні приховані дані, які можуть бути невидимими для звичайного зору. Ці приховані дані містять таку інформацію, як кластери, структури, затінення тощо.

Як впливає з назви, алгоритм кластеризації поділяє зображення на кластери (непересікаючі групи) пікселів, які мають схожі характеристики. Це розділить елементи даних на кластери, де елементи в кластері більш схожі на елементи, присутні в інших кластерах.

Деякі з популярних алгоритмів кластеризації включають нечіткі c -середніх (FCM), k -середніх та покращені алгоритми k -середніх. У сегментації зображень ви здебільшого використовували б алгоритм кластеризації k -середніх, оскільки він досить простий та ефективний. З іншого боку, алгоритм FCM поміщає пікселі в різні класи відповідно до їх різного ступеня належності.

Найважливішими алгоритмами кластеризації для сегментації в обробці зображень є:

K-means – це простий алгоритм машинного навчання без нагляду. Він класифікує зображення за певною кількістю кластерів. Він починає процес, розділяючи простір зображення на k пікселів, які представляють k центроїдів групи.

Потім вони відносять кожен об'єкт до групи на основі відстані між ними та центроїдом. Коли алгоритм призначив усі пікселі всім кластерам, він може переміщати та перепризначати центроїди.

Нечіткі C середні. За допомогою методу кластеризації нечітких c -середніх пікселів на зображенні можна об'єднати в кілька кластерів. Це означає, що піксель може належати більш ніж одному кластеру. Однак кожен піксель буде мати різні рівні подібності з кожним кластером. Алгоритм нечітких c -середніх має функцію оптимізації, яка впливає на точність результатів. Алгоритми кластеризації можуть вирішити більшість завдань у сегментації зображень.

Сегментація на основі країв спирається на краї, знайдені на зображенні за допомогою різних операторів виявлення країв. Ці краї позначають місця розриву зображення в рівнях сірого, кольорі, текстурі тощо. Коли ми переходимо від однієї області до іншої, рівень сірого може змінитися. Отже, якщо зможемо знайти цей розрив, ми зможемо знайти це ребро. Доступні різні оператори виявлення країв, але отримане зображення є проміжним результатом сегментації, і його не слід плутати з кінцевим сегментованим зображенням. Потім потрібно виконати подальшу обробку зображення, щоб сегментувати його. Додаткові кроки включають об'єднання отриманих сегментів країв в один сегмент, щоб зменшити кількість сегментів, а не шматків малих меж, які можуть перешкоджати процесу заповнення області. Це робиться для отримання межі об'єкта. Метою сегментації краю є отримання проміжного результату сегментації, до якого можемо застосувати сегментацію на основі регіонів або будь-який інший тип сегментації, щоб отримати остаточне сегментоване зображення (рисунок 2.3).



Рисунок 2.3 – Класифікація меж об’єктів на основі аналізу яскравості

Краї зазвичай асоціюються з поняттями «Величина» і «Напрямок». Деякі детектори країв дають обидва напрямки і величину. В процесі роботи можна використовувати різні детектори країв, як-от оператор краю Собеля, детектор краю, оператор краю Кірша, оператор краю Превітта, оператор краю Роберта тощо.

Оператор Робертса. Даний метод чудово підходить для зображень з чітко виділеними краями, оскільки він є достатньо простим та швидким в роботі. Його імплементація в програмні додатки не вимагає стороннього втручання або додаткових бібліотек. Для виявлення границь об’єктів на зображенні використовуються два простих фільтра розміром 2x2, які призначені для підкреслення областей зображення де відбуваються різкі перепади яскравості.

$$H_1^R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

$$H_2^R = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Оператори напрямку. Даний підхід фільтрації з метою виділення контурів включає в себе набір фільтрів які поодиноці здатні вловити стрибки яскравості в

одному з основних напрямків (в орієнтації 45 градусів у восьми напрямках). Проте поєднання декількох фільтрів в результаті дають досить точні результати:

$$\begin{aligned}
 H_0^S &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, & H_4^S &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \\
 H_1^S &= \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}, & H_5^S &= \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, \\
 H_2^S &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, & H_6^S &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \\
 H_3^S &= \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}, & H_7^S &= \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}.
 \end{aligned}$$

Силу краю та орієнтацію також потрібно розрахувати, але вони по-різному. Міцність краю визначається максимальним градієнтним зображенням із восьми фільтрів.

Сегментація на основі границь використовується для визначення розташування пікселів на зображенні, які завжди відповідають краям об'єктів. Тоді приймається, що все-таки це межа області або об'єкта, а потім воно закривається і якщо номери сусідніх об'єктів будуть такими ж, то вони об'єднуються. Для точності сегментації периметр виявлених кордонів повинен трохи дорівнювати периметру об'єкта на введеному зображенні. Дані методи мають проблеми із зображенням, які мають:

- слабо виражені краї;
- зашумлені;
- області дуже подібні між собою.

Інші проблеми цих методів пов'язані з складністю керування функцією градієнта, що призводить до небажаних результатів.

Нарощування регіону. Структури даних, які використовуються для відстеження набору центральної точки або так званого «насіння», зазвичай є «стеком». У стеку визначено дві операції: *push*, який переміщує піксель на верхню частину стека, і *pop*, який бере піксель з вершини стека. В алгоритмі зображення позначається f , «насіння» має координати (x,y) і значення сірого $g=f(x,y)$. Збільшення області здійснюється шляхом встановлення значення сірого для кожного об'єднаного пікселя на значення h (яке не повинно дорівнювати g). Область на зображенні можна отримати, вибравши всі пікселі з сірим значенням h . Щоб переконатися, чи є результат правильним чи ні, перед запуском алгоритму повинні мати можливість вибрати h як значення, якого немає на реальному зображенні. Оператор, який вирішує, чи потрібно об'єднати піксель («якщо $f(a,b)=g$ »), можна змінити, щоб використовувати інший критерій об'єднання. Проста модифікація полягала б у тому, щоб дозволити об'єднання пікселів, які знаходяться в певному діапазоні значень сірого («якщо $l < h$ »).

У цьому методі алгоритм починає свою роботу з невеликого набору пікселів, а потім починає ітеративно об'єднувати більше пікселів відповідно до конкретних умов подібності. Алгоритм зростання регіону вибирає довільний початковий піксель на зображенні, порівнює його з сусідніми пікселями і починає збільшувати область, знаходячи збіги з початковою точкою.

Коли конкретна область не може розвиватися далі, алгоритм вибере інший початковий піксель, який може не належати жодному існуючому регіону. Одна область може мати занадто багато атрибутів, через що вона займає більшу частину зображення. Щоб уникнути такої помилки, алгоритми вирощування регіонів розвивають кілька регіонів одночасно.

Ви повинні використовувати алгоритми збільшення області для зображень, які мають багато шуму, оскільки шум ускладнить пошук країв або використання порогових алгоритмів.

Розділ та злиття регіонів. Як випливає з назви, сфокусований метод розділення та об'єднання областей виконував би дві дії разом – розділяючи та об'єднуючи частини зображення.

Спочатку зображення розбивається на області, які мають подібні атрибути, і об'єднує сусідні частини, схожі одна на одну. При розподілі регіонів алгоритм розглядає все зображення, тоді як при зростанні області алгоритм фокусується на конкретній точці.

Метод розділення та злиття регіонів дотримується методології «розділяй і владарюй». Він ділить зображення на різні частини, а потім узгоджує їх відповідно до заданих умов. Інша назва алгоритмів, які виконують це завдання, — алгоритми розділеного злиття

Сегментація вододілу. При обробці зображень вододіл – це перетворення зображення у відтінках сірого. Він відноситься до геологічного вододілу або вододілу. Алгоритм вододілу обробляв би зображення так, ніби це була топографічна карта. Він розглядає яскравість пікселя як його висоту і знаходить лінії, які проходять уздовж вершини цих хребтів.

Вододіл має багато технічних визначень і має кілька застосувань. Окрім визначення хребтів пікселів, він зосереджується на визначенні улоговин (протилежних від хребтів) і заливає улоговини маркерами, поки вони не зустрінуться з лініями вододілу, що проходять через хребти.

Оскільки басейни мають багато маркерів, а гребені – ні, зображення ділиться на кілька областей відповідно до «висоти» кожного пікселя.

Метод вододілу перетворює кожне зображення на топографічну карту. Метод сегментації вододілу відображатиме топографію через сірі значення пікселів.

Тепер ландшафт з долинами та хребтами, безумовно, мав би тривимірні аспекти. Вододіл розглядатиме тривимірне відображення зображення і відповідно створюватиме регіони, які називаються «басейнами водозбору».

Він має багато застосувань у медичній галузі, наприклад, МРТ, медичне зображення тощо. Сегментація вододілу є важливою частиною сегментації

медичних зображень, тому, якщо ви хочете увійти в цей сектор, вам слід зосередитися на вивченні цього методу для сегментації, зокрема, при обробці зображень.

Нейронні мережі для сегментації. Використання AI для аналізу зображення та визначення його різних компонентів, таких як обличчя, об'єкти, текст тощо. Згорткові нейронні мережі досить популярні для сегментації зображень, оскільки вони можуть ідентифікувати та обробляти дані зображення дуже швидко й ефективно. Експерти Facebook AI Research (FAIR) створили архітектуру глибокого навчання під назвою Mask R-CNN, яка може створити піксельну маску для кожного об'єкта, присутній на зображенні. Це розширена версія архітектури виявлення об'єктів Faster R-CNN. Швидший R-CNN використовує дві частини даних для кожного об'єкта на зображенні, координати рамки та клас об'єкта. З Mask R-CNN ви отримуєте додатковий розділ у цьому процесі. Маска R-CNN виводить маску об'єкта після виконання сегментації.

У цьому процесі спочатку необхідно передати вхідне зображення до ConvNet, яка генерує карту ознак для зображення. Потім система застосовує мережу регіональних пропозицій (RPN) до карт об'єктів і генерує пропозиції об'єктів з їх оцінками об'єктивності.

Після цього до пропозицій застосовується шар об'єднання інвестицій, щоб зменшити їх до одного розміру. На останньому етапі система передає пропозиції зв'язаному шару для класифікації та генерує вихід з обмежувачими рамками для кожного об'єкта.

Порівняльне дослідження цих методів із використанням деяких стандартних параметрів, таких як: швидкість, складність обчислень, автоматичність, шумостійкість, виявлення кількох об'єктів та точність, проведено для методів на основі порогів, методів на основі регіонів та кластерних методів. Експериментальні результати показують, що метод на основі порогів і кластерів є порівняно швидшим, ніж метод на основі регіонів, і при перевірці на складність обчислень є швидким для регіону та кластера, де він менший для порогового значення. Шумостійкість помірна для кластерів, де вона

менша для решти. Алгоритми на основі кластерів автоматично дає результати, тоді як решта є напівавтоматичними. Точність сегментації хороша і хороша лише з методами на основі регіонів. Описані експериментальні аналізи наведені в таблиці 2.1.

Таблиця 2.1– Порівняльний аналіз алгоритмів сегментації

Параметр	Пороговий метод	Метод на основі виділення регіонів	Метод на основі кластеризації
Швидкість	швидкий	повільний	швидкий
Складність обчислень	низька	швидкий	швидкий
Автоматичність	напівавтоматичний	напівавтоматичний	автоматичний
Шумостійкість	низька	низька	помірна
Точність	помірний	точний	помірний
Виявлення кількох об'єктів	поганий	допустимий	допустимий

Сегментація зображень є важливою частиною комп'ютерного зору, і вона використовується в багатьох різних сферах бізнесу. Сегментація зображень використовується технологією розпізнавання обличчя у вашому телефоні та передовими системами безпеки для розпізнавання вашого обличчя. Система дорожнього руху може розпізнати автомобіль та отримати інформацію про його власника за допомогою технологій ідентифікації номерних знаків. Алгоритми сегментації зображень використовуються Google та іншими пошуковими системами на основі зображень, щоб розпізнати елементи на вашому зображенні та порівняти їхні висновки з відповідними фотографіями, які вони знаходять, щоб надати вам результати пошуку.

2.2 Алгоритми оцінки результатів сегментації

Одним з ключових параметрів для створення систем обробки зображень з елементами автоматизації є можливість проведення оцінки результатів процедури сегментації. Оскільки етап розділення зображення на однорідні області або об'єкти має відчутний вплив на якість кінцевих висновків. Якість сегментації може визначатись на основі декількох різних критеріїв серед яких спектральні характеристики, яскравість областей, статистичні параметри тощо. Самим простим підходом до визначення якості сегментації є порівняння результатів які були отримані за допомогою програмної системи та за допомогою експертної сегментації. Міра близькості цих двох зображень і визначає наскільки точно була проведена сегментація. Дана оцінка може бути як суб'єктивною, оскільки результати може візуально оцінювати експерт, який має власний досвід та власні погляди на можливість поділу зображення на окремі області. Так і об'єктивно, при якій програмно підраховується кількість точок які розташовані в одній області на двох зображеннях.

Для отримання кількісної оцінки сегментації можна проводити дослідження використовуючи критерії схожості результатів сегментації за допомогою системи та за допомогою людини. Проте слід враховувати, що при такій оцінці на результат роботи буде впливати людський фактор. Для усереднення результатів проведеної оцінки можна використати результати сегментації декількох експертів з подальшим обранням середнього результату. Для проведення такої оцінки використовуються такі критерії:

- критерій пребору сегментації:

$$BSeg = \frac{S_1 \cap S_2^e}{S_1 \cup S_2}$$

- критерій недобору сегментації:

$$NSeg = \frac{S_1^e \cap S_2}{S_1 \cup S_2}.$$

де S_1 – результати проведеної сегментації на основі деякого алгоритму сегментації;

S_2 – результати проведеної сегментації на основі роботи експерта;

S_1^e – контурна лінія на основі деякого алгоритму сегментації;

S_2^e – експертна контурна лінія.

Для визначення кінцевого результату сегментації використаємо формулу:

$$QSeg = BSeg + NSeg .$$

На основі даного критерію вже можна робити деякі висновки про отримані результати сегментації. Проте слід пам'ятати, що ці результати будуть насити суб'єктивний характер.

Іншим фактором, на сонові якого можна проводити кількісну оцінку результатів роботи алгоритмів сегментації є оцінка помилок сегментації. Під час сегментації можуть виникнути помилки двох типів:

- хибне віднесення до межі області;
- хибне не віднесення до межі області.

На основі цих двох похибок сегментації теж можна виділити два критерії подібності які будуть ґрунтуватись на обчисленні значення цих помилок в розрізі цілого зображення.

Перший критерій *FOM* – це критерій подібності контурної лінії ідеально сегментованої області (за допомогою експерта), який буде визначчатись на основі формули:

$$FOM(f, g) = \frac{1}{\max\{card(f), card(g)\}} \cdot \sum_{i=1}^{card(g)} \frac{1}{1 + d^2(i)},$$

де $card(g)$ - кількість пікселів в ідеальному контурі g ;

$card(f)$ - кількість пікселів в отриманому контурі f ;

$d(i)$ – відстані між відповідними точками контуру g та f .

Перший критерій RMS – це критерій подібності контурної лінії ідеально сегментованої області (за допомогою експерта), який буде визначатись на основі формули:

$$RMS(f, g) = \sqrt{\frac{1}{card(X)} \cdot \sum_{x \in X} (f(x) - g(x))^2}$$

де $f(x), g(x)$ – значення інтенсивності пікселів;

X – множина точок які належать відповідній області.

Іншою групою критерії оцінки результатів роботи алгоритмів сегментації є суб'єктивні. Дані критерії ґрунтуються на стрийнятті людиною результатів сегментації. Дані критерії складніше внести в роботу програмних додатків, проте для налаштувань загальних параметрів даний підхід є достатньо швидким та надійним. На сьогоднішній день є різні шкали для провдення суб'єктивної оцінки, більшість з них ґрунтується на звичних для людини поняттях:

- схоже/не схоже;
- подобається/ не подобається;
- оцінки за звичними критеріями від дуже добре до дуже погано;
- подібно до ідеалу/ несхоже тощо.

Всі ці критерії можна оцінювати з цізною точністю і деталізацією, проте результати такої оцінки будут напряду залежети від експерта який проводить оцінку. Приклади шкал порівняння наведено в таблиці 2.2.

Таблиця 2.2 – Шкали порівня для суб'єктивної оцінки

Нормальзована	П'ятибальна шкала		Сембальна шкала погіршення
	Шкала якості	Шкала похибки	
1	5 (відмінно)	5 (відсутня)	1 (відсутня)
0,875	–	–	2 (частково помітна)
0,75	4 (добре)	4 (ледь помітна)	3 (помітна, на результат не впливає)
0,625	–	–	4 (помітна, на результат частково впливає)
0,5	3 (допустимо)	3 (помітна)	5 (результат частково спотворений)
0,25	2 (підійде з допрацюванням)	2 (впливає на результат)	6 (значні помилки)
0	1 (не підходить)	1 (значно спотворює результат)	7 (негативний результат)

Оскільки даний спосіб оцінювання базується на принципах опитування декількох людей, то при аналізі його результатів доцільно провести процедуру усереднення отриманих даних. Усереднення проводиться за наступною формулою:

$$ser = \frac{1}{N} \sum_{i=1}^y i \cdot n_i,$$

де N – загальна кількість проведених опитувань;

n_i – кількість оцінок з значенням i ;

y – розкид оцінок отриманих під час оцінювання.

Інша група оцінок які вже можна інтернувати в системи обробки зображень є об'єктивними. Дані критерії дозволяють отримати характеристику зображення на основі різницевого сигналу, що просто можна обчислити та використати в подальших дослідженнях. Середньоквадратичний критерії належить саме до цієї

групи критеріїв. По ньому можна визначити міру невідповідності між двома зображеннями $f_1(x,y)$ та $f_2(x,y)$. Для цифрових зображень які, задані для меж $x[0,M]$ та $y[0,M]$, значення середньоквадратичного відхилення можна обчислити за формулою:

$$\sigma^2 = \int_0^M \int_0^N [f_1(x,y) - f_2(x,y)]^2 dx dy.$$

У інколи доцільно використати критерій максимальної помилки, який дозволяє встановити значення максимальної помилки перетворення:

$$\varepsilon_{\max} = \max_{x,y} |f_1(x,y) - f_2(x,y)|.$$

Управління інформацією в автоматизованих системах обробки даних тісно пов'язане з вирішенням задач оцінки отриманих результатів, накопичення даних та організацію обміну інформаційних між окремими модулями, а також вироблення нових критеріїв оцінки якості алгоритмів сегментації зображень.

2.3 Алгоритм сегментації зображень на основі аналізу малоінформативних областей

При проведенню аналізу груп алгоритмів сегментації було виділено основні проблеми з якими стикаються користувачі в процесі роботи. Серед основних пролем це присутність невеликих областей які можуть бути віднесені до декількох однорідних областей з приблизно однаковим рівнем ймовірностей. Дана ситуація може виникнути у випадках коли зображення містить дрібні об'єкти які до того ж мають подібні характеристики. Існуючі алгоритми

сегментації при виникненні подібної ситуації або проводять автоматичну корекції параметрів своєї роботи або просто створюють новий клас об'єктів на зображенні. У першому випадку зміна параметрів роботи не завжди може призвести до вирішення ситуації оскільки це може поробити нові області. При цьому немає гарантії що кількість нових областей не буде більшою ніж при попередніх параметрах. Інша проблема, що у деяких випадках неможливо провести зміну параметрів роботи алгоритму. Причини можуть бути різними: технічні обмеження алгоритмів, значення чітко регламентовані, не передбачено механізм адаптивної корекції параметрів. При розгляді причин другої групи, при якому алгоритм сортування створює нові класи, то очевидним стає висновок, що збільшення кількості класів негативно впливатиме на результати роботи системи. Це пояснюється, тим що у випадку коли користувач точно знає що на зображенні присутня деяка кількість областей, а алгоритм виділяє іншу кількість, то користувач зробить висновок про хибність роботи алгоритму і неточність отриманих результатів. Відповідно провівши такий аналіз було запропоновано алгоритм сегментації на основі малоінформативних областей.

Розроблений алгоритм сегментації базується на послідовній сегментації зображення декількома алгоритмами з різними параметрами роботи. З визначення поняття сегментації випливає, що якщо точки зображення будуть мати однакові або близькі характеристики то вони формуватимуть однорідну область. А отже якщо провести сегментації декількома алгоритмами, то точки з подібними характеристиками будуть предувати в одні області назважаючи на спосіб розподілення вхідного зображення. У процесі сегментації точки які сформують малоінформативні області будуть аналізуватись наступним чином: від загального результату сегментації будуть відніматись послідовно варіанти сегментації які вносять максимальну неясність. Тобто якщо точки формують однорідну область у більшості випадків, то результати сегментації для цієї конкретної області в яких отримані інші варіанти будуть ігноруватись в кінцевому результаті. Це дозволить зменшити кількість малоінформативних областей, а також підвищити точність результатів сегментації.

Блок-схема спроектованого алгоритму сегментації зображень проілюстрована на рисунку 2.4.

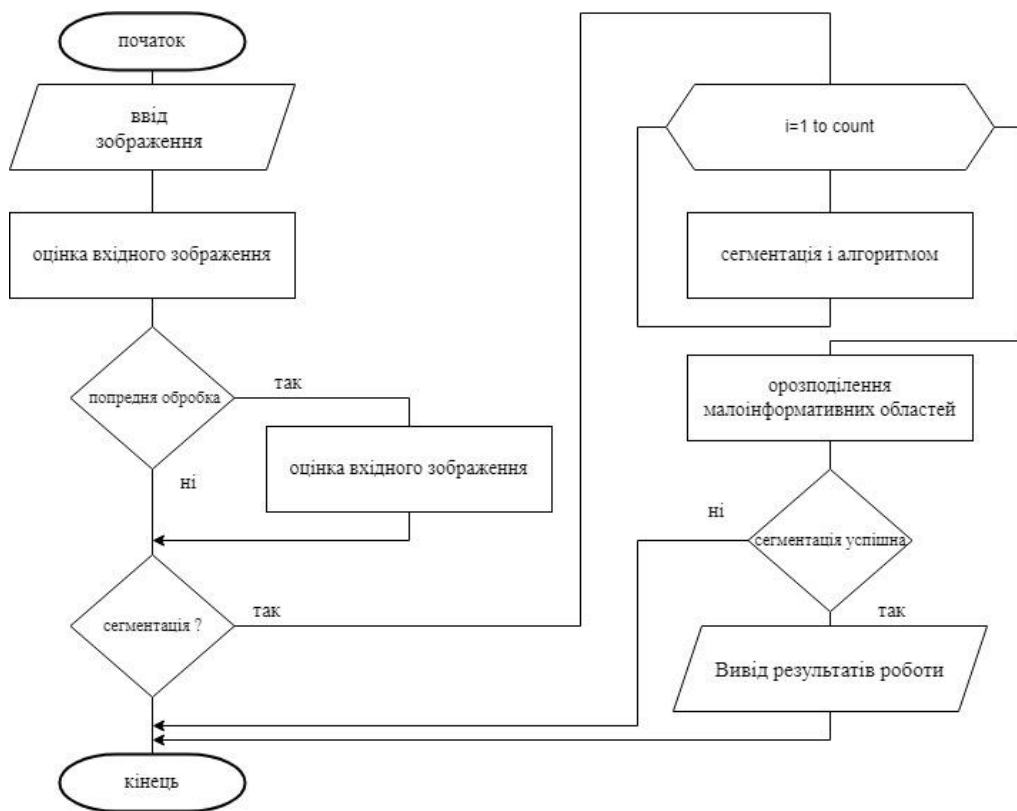


Рисунок 2.4 – Блок-схема алгоритму сегментації зображень на основі аналізу малоінформативних областей

Запропонований алгоритма має ряд переваг:

- результати роботи базуються на різних характеристиках, що підвищує точність обробки зображення;
- відсутність малоінформативних областей на зображенні
- автоматична оцінка результатів сегментації зображення
- алгоритм може працювати в автоматичному режимі.

Проте окрім яких переваг запропонований алгоритм містить ряд недоліків:

- велика кількість додаткових обчислень;
- оцінка ґрунтується на попередньому експертному аналізі
- складність обробки зображень з великою кількістю дрібних областей.

Незважаючи на наведені недоліки алгоритм показує високу точність отриманих результатів сегментації та розрахований на програмні системи для яких час опрацювання не відіграє суттєвої ролі.

Зображення це один з найбільш інформативних способів передічі та зберігання інформації. Тому розробка нових алгоритмі їх обробки та аналізу є та буде залишатись пріорітетним завдання для розробників.

2.4 Висновки до розділу

Досліджено алгоритми сегментації зображень та поведено їх оцінку на основі суб'єктивни та об'єктивних критеріїв, щозволило виділити групи алгоритмів на основі яких і було спроектувано алгоритм сегментації на основі аналізу малоінформативних областей.

Розроблено алгоритм сегментації на основі аналізу малоінформативних областей, що надало можливість в подальшому спроектувати структуру системи оброки та аналізу зображень з мобулем автоматизованої сегментації.

3 ПРОГРАМНА СИСТЕМА ОБРОБКИ ТА АНАЛІЗУ ЦИФРОВИХ ЗОБРАЖЕНЬ

3.1 Структура програмного додатку сегментації цифрових зображень

При старті будь-якого проекту одонією з перших задач на яку необхідно дати відповідь це питання обрання технології та засобів для реалізації. Оскільки на сьогоднішній день є безліч варіантів, тому необхідно провести детальний аналіз щодо вибору.

Об'єктно-орієнтована парадигма – це новий і інший спосіб мислення про програмування. В даному підході, що все має бути об'єктом Метод або іншими словами методологія – це набір процесів і евристик, які використовуються для визначення складності проблеми програмування. Багато методів ООП було сформульовано з самого початку об'єктно-орієнтованого програмування. ООП є полем для багатьох експериментів, тому важливо зрозуміти, яку проблему намагається вирішити метод, перш ніж розглянути його. Особливо це стосується C++, в якому мова програмування призначена для зменшення складності (у порівнянні з C), пов'язаної з вираженням програми. Це може фактично зменшити потребу в дедалі складніших методологіях.

Ів цьому напрямку слід розуміти, що строк життя об'єкта не обмежується часом, коли його тільки безпосередньо використовують. Натомість дизайн об'єкта з'являється в послідовності етапів. Тому при розробці та проектування класів програмної системи було розглянуто їх в пересективі п'яти етапів. Відкриття об'єкта. Цей етап відбувається під час початкового аналізу програми. Об'єкти можуть бути виявлені шляхом пошуку зовнішніх факторів і кордонів, дублювання елементів у системі та найменших концептуальних одиниць. Спільність між класами, що пропонують базові класи та успадкування, може з'явитися відразу або пізніше в процесі проектування. Збірка об'єкта. Внутрішні потреби об'єкта можуть вимагати інших класів для його підтримки. Побудова системи. Потреба в комунікації та взаємозв'язку з іншими об'єктами в системі

може змінити потреби класів або потребувати нових класів. Розширення системи. Послідовне додавання нових функції до системи, для підвищення функціональних можливостей системи, проте це може потребувати преробки інтерфейсу або ієрархії класів. Повторне використання об'єктів. Типи багаторазового використання, як правило, менш поширені, і вони повинні вирішувати більш загальні проблеми, щоб бути повторно використаними.

Метою використання саме C++ є підвищення продуктивності. Ця продуктивність досягається багатьма способами, але мова розроблена, щоб допомогти якомога більше, водночас якнайменше заважаючи довільними правилами або будь-якими вимогами щодо використання певного набору функцій. C++ розроблено, щоб бути практичним; Рішення щодо дизайну мови C++ були засновані на наданні максимальної вигоди для програміста (принаймні, зі світогляду C). В C++ забезпечує кращу перевірку типів і аналіз під час компіляції. Потреба в препроцесорі практично зникла для заміни значень і макросів, що усуває набір проблем, які важко знайти. C++ має функцію, яка називається посиланнями, яка дозволяє зручніше обробляти адреси для аргументів функцій і значень, що повертаються. Обробка імен покращена за допомогою функції, яка називається перевантаженням функцій, яка дозволяє використовувати одне ім'я для різних функцій. Функція, яка називається просторами імен, також покращує контроль над іменами. Існує безліч менших функцій, які покращують безпеку C.

Тому вибір мови програмування C++ та методології об'єктно-орієнтованого підходу є очевидним та має ряд значних переваг при реалізації програмної системи обробки цифрових зображень.

На наступному кроці реалізації програмної розробки необхідно розробити її внутрішню структуру та принципи обміну інформацією між різними її програмними складовими. Для реалізації даного проекту було обрано класичний модульний підхід, що в подальшому гарантує простоту адаптації системи під різні завдання. Запропоновану структуру системи наведено на рисунку 3.1.

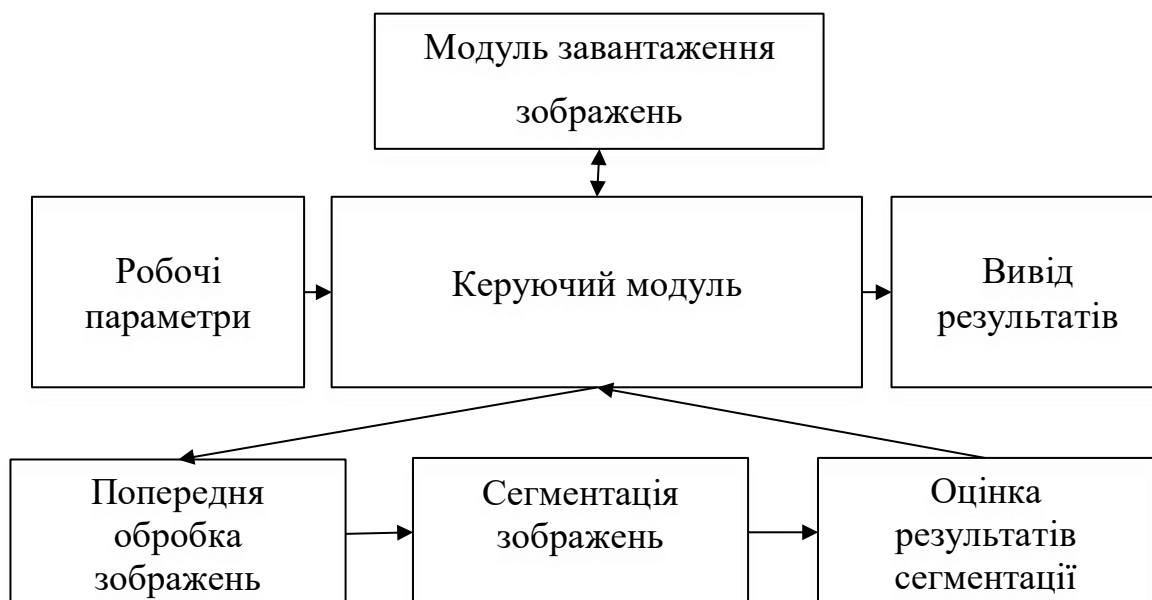


Рисунок 3.1 – Узагальнена структура програмної системи обробки зображень

Спроектowana структура програмного засобу володіє рядом переваг. Однією з головних переваг є те, що такий підхід дозволяє змінювати структурні елементи програми не завдаючи ніякої шкоди цілісності та функціональним можливостям розробленого програмного забезпечення. Додавання, модифікація або видалення окремого модуля просто унеможливить виконання окремо взятого етапу, проте програма зможе успішно виконати усі інші задачі. Окрім того модульна структура дозволяє використати можливості сторонніх цифрових бібліотек та надає можливість розпаралелити процес написання програмного коду. В цьому випадку частину маловагомих функцій можна використати з зовнішніх джерел, що надасть більше часу для реалізації важливих функцій, наприклад функції сегментації. Проведемо аналіз окремих функціональних модулів системи.

Керуючий модуль. При проектуванні структури програмної системи було прийнято рішення згрупувати допоміжні функції в один блок, та надати даному блоку можливість управління програмним додатком від старту до завершення процесу роботи програми. Таке рішення дозволило зменшити можливість виникнення помилкових ситуацій на різних етапах роботи програми. Зокрема

додавання функції перевірки наявності всіх складових елементів програми дозволило вже на початковій стадії провести оцінку програми та виявити неполадки. Іншим завдання даного блоку є формування меню дозволених програмних команд. Елементи меню, блоки яких є пошкодженими просто будуть неактивними, що дозволить користувачу одразу оцінити можливості програмної системи та можливості виконання поставлених завдань. Приклад набору функції керуючого модуля наведено на рисунку 3.2.

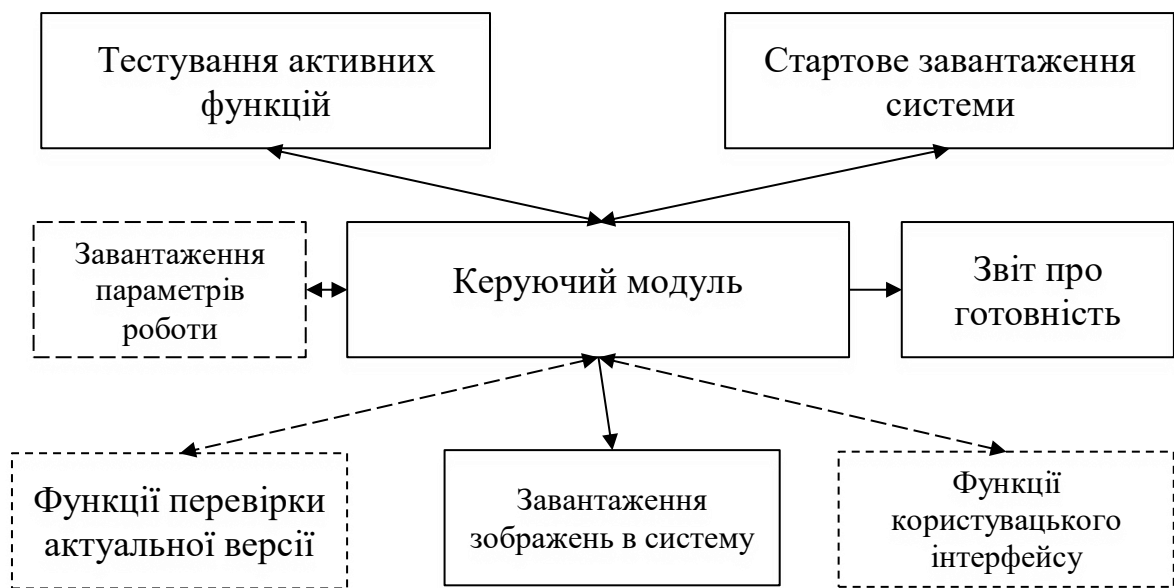


Рисунок 3.2 – Узагальнена структура керуючого модуля

Модуль завантаження зображення – дозволяє користувачеві завантажити необхідне зображення для аналізу. В даному блоці реалізовано декілька варіантів завантаження зображення. Перший варіант – це завантаження зображення з локального носія інформації. Зображення повинно попередньо бути створене та збережене на вінчестері. Другий варіант завантаження передбачає наявність камери для отримання зображення в режимі реального часу. Така можливість дозволяє опрацьовувати не тільки цифрові зображення, а й звичайні, оскільки їх можна буде оцифрувати використовуючи камеру. Інший варіант завантаження зображення включає можливість пошуку зображень на віддалених носіях інформації, що дозволить використовувати можливості програмної розробки

віддалено. Окрім того в даному модулі передбачені можливості по переконвертації вхідних зображень в формати які зручні для роботи з програмою. Вхідні дані – назва файла для імпорту зображень, а на виході – цифрове зображення переконвертоване в формат bmp.

Модуль сегментації цифрових зображень – один з ключових модулів програмної системи обробки та аналізу цифрових зображень. В ньому реалізовані функції середнього рівня обробки зображень, а саме сегментації зображень різними алгоритмами. Користувач може обрати як окремих алгоритм сегментації так і набір алгоритмів як для порівняння отриманих результатів так і для оцінки алгоритмів за іншими параметрами: час, складність операцій тощо. Схема взаємодії функцій наведена на рисунку рисунок 3.3. В результаті використання функцій даного модуля буде отримано зображення з виділеними на ньому однорідними областями, а також структуру яка буде описувати вхідне зображення для можливого подальшого аналізу.

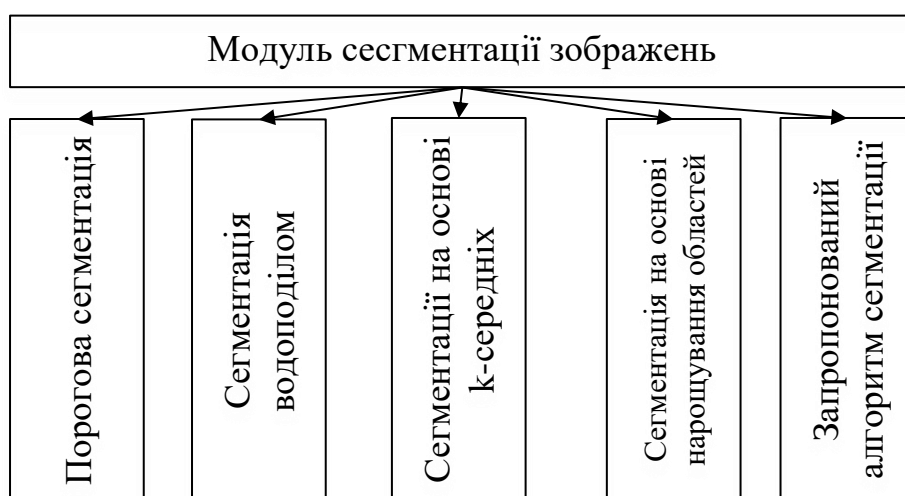


Рисунок 3.3 – Узагальнена структура слока сегментації зображень

Модуль налаштування параметрів роботи програмного додатку – це один з допоміжних програмних модулів призначений для надання користувачеві встановлення більш зручних та комфортних параметрів роботи з програмною системою. Встановлення шрифтів, кольору деяких графічних елементів,

параметрів відображення візуальних елементів та завантаження нових та зереження опрацьованих файлів. Відсутність модуля ніяк не вплине на роботу програми в цілому проте для підвищення зручності він був включений в кінцевий варіант програмного додатку.

Модуль попередньої обробки. Для прискорення роботи програмного додатку було проведено дослідження яке показало, що найбільші часові затримки виникають під час сегментації зображень у випадках коли вхідні зображення мають велику кількість дефектів: зашумленість, висока/низька яскравість, погана чіткість зображення тощо. Тому в структуру програмного додатку було включено елементи поперньої обробки зображення. В результаті роботи функцій даного блоку зображення будуть більш підготовлені до етапу сегментації, що в свою чергу зменшить часові затрати та підвищить точність отриманих результатів в цілому. Серед алгоритмів які були реалізовані в даному блоці можна відмітити:

- корекція яскравості;
- корекція контрасту;
- фільтрація;
- бінаризація;
- гама-корекція;
- виділення областей інтересу;
- преконвертування цифрового зображення.

На вході даний модуль отримує цифрове зображення, після проведення відповідних алгоритмів обробки дане зображення є більш адаптованим для його подальшого аналізу та обробки.

Для відображення отриманих результатів у зручному для користувача сорматі в структуру програмного додатку було додано модуль формування звітів про цифрове зображення. Даний модуль не є ключовим в роботі системи обробки зображень, але без нього зручність роботи з програмою буде нижча. Функції даного модуля дозволяють користувачеві переглянути кількісні характеристики отриманих однорідних оластей, отримати загальну статистику про проведені

операції, налаштувати параметри візуалізації отриманої інформації. При потребі передати отриману статистичну інформацію іншим користувачам реалізована можливість збереження отриманих даних в файлі. Налаштування формату звіту та спосіб його зберігання можна задати за допомогою модуля встановлення параметрів роботи програми.

Запропонована структура програмної системи для сегментації кольорових цифрових зображень є мінімально достатньою для вирішення поставлених завдань. Проте при необхідності збільшення функціональних можливостей при постановці нових задач, структура можна оновити без додаткових затрат. В даному випадку нові класи та структури даних просто інтегруються в ситсему та під'єднуються через керуючий блок команд.

Для проведення оцінки структури спроектованої системи сегментації цифрових зображень було проведено ряд тестових моделювань. В процесі моделювання було досліджено:

- можливості роботи програмної системи в випадках некоректних дій користувачів
- перевірка доступності окремих функцій програмного додатку для користувачів
- перевірка наявності проблем при запуску послідовності команд при виконання процесу тестування.

Результати проведених тестувань були оформленні у вигляді UML діаграм на наведені в роботі. Використання саме механізму UML діаграм для процесу моделювання пояснюється присутністю там таких механізмів як:

- Діаграма прецедентів. Як найбільш відомий тип діаграм поведінкових типів UML, діаграми варіантів використання дають графічний огляд акторів, задіяних у системі, різних функцій, необхідних цим акторам, і того, як ці різні функції взаємодіють. Це чудова відправна точка для будь-якого обговорення проекту, оскільки можна легко визначити головних учасників та основні процеси системи.

– Діаграми послідовності в UML показують, як об'єкти взаємодіють один з одним і порядок цих взаємодій. Важливо зазначити, що вони показують взаємодії для конкретного сценарію. Процеси представлені вертикально, а взаємодії – стрілками.

– Діаграми класів є основним будівельним блоком будь-якого об'єктно-орієнтованого рішення. Він показує класи в системі, атрибути та операції кожного класу та відносини між кожним класом. У більшості інструментів моделювання клас складається з трьох частин. Ім'я вгорі, атрибути посередині та операції чи методи внизу. У великій системі з багатьма пов'язаними класами класи групуються разом для створення діаграм класів. Різні відносини між класами показані різними типами стрілок.

По-перше проводилось моделювання для оцінки можливостей взаємодії користувача з різними блоками програми. Приклад діаграми прецедентів наведено на рисунку 3.4.

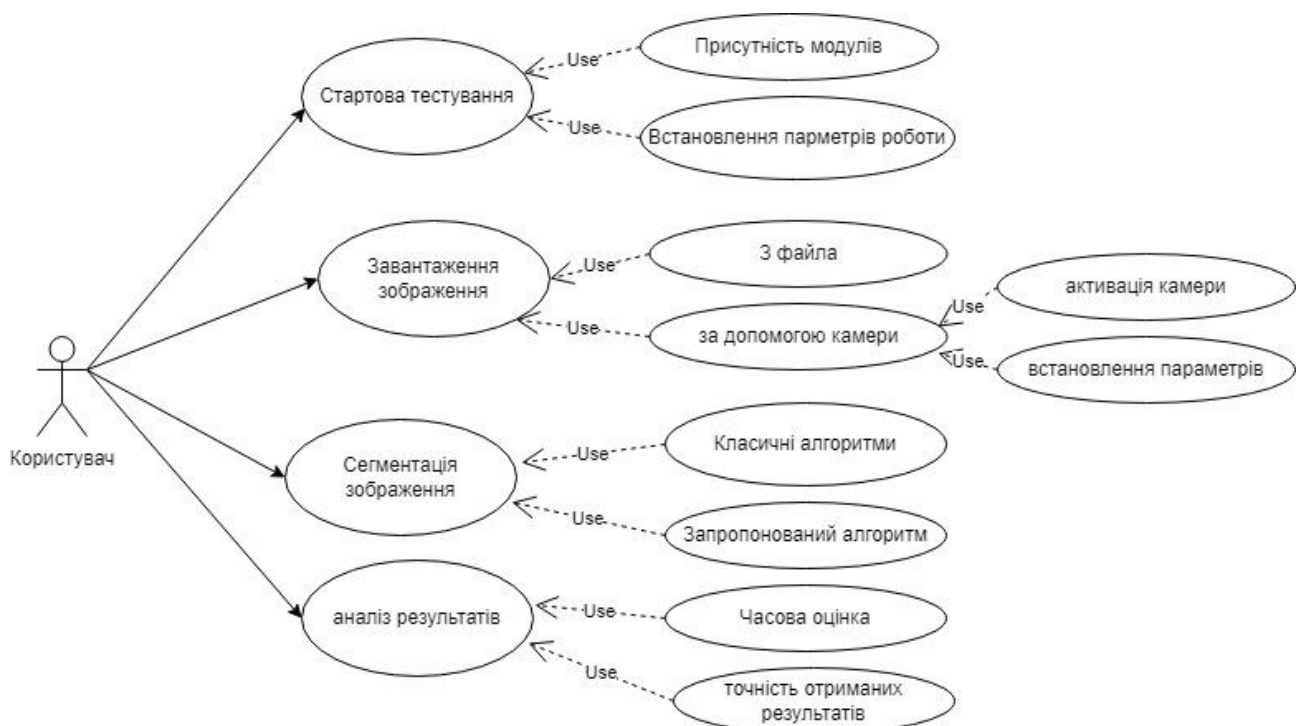


Рисунок 3.4 – Діаграма прецедентів програмної системи

На основі отриманих результатів моделювання можна зробити висновки, що група акторів «Користувач» має доступ до усіх необхідних функцій при роботі з програмним додатком. Проте, в цієї групи є немає доступу до основних системних функцій розробленого програмного додатку. Це зроблено з тією метою, щоб підвищити рівень стабільності роботи програмного додатку. Оскільки користувачі можуть мати різних рівень досвіду роботи з програмними додатками такого типу, то очевидним є спроба збільшення автоматизації деяких процесів в середині програмного додатку. Проте, для користувачів які мають підвищені вимоги до програми та розуміють суть параметрів в реалізованих алгоритмах є доступ до більшої кількості налаштувань та можливість впливати на результати процесу сегментації.

Наступний аналіз структури програмного забезпечення був спрямований на перегляд кроків, які необхідно здійснити програмі для виконання поставлених завдань. Зокрема результати проведенного моделювання наведені на рисунку 3.5 та ілюструють покроковий процес сегментації цифрового зображення.

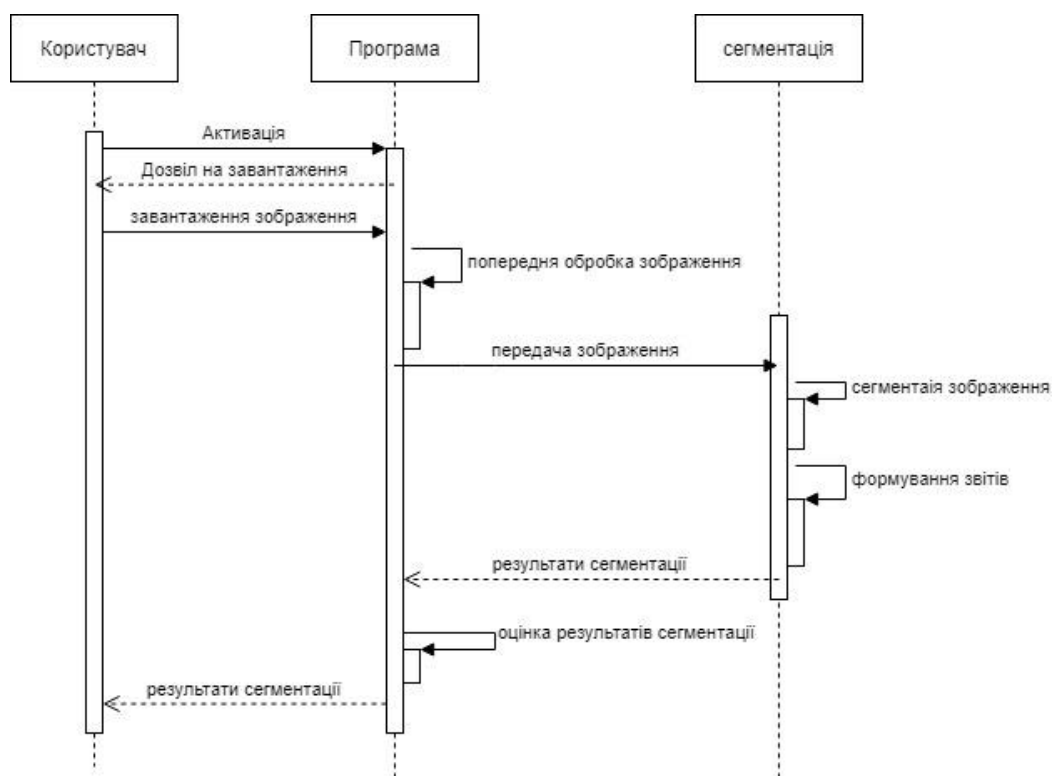


Рисунок 3.5 – Діаграма послідовності програмної системи

На даному етапі дослідження було відбувалась перевірка можливості роботи наявних функцій та не виникнення ситуації з конфліктом між ними. Окремі програмні функції стартують у чіткій послідовності не створюючи між собою конфліктів в структурі програмного додатку. Розроблені програмні інтерфейси дозволяють без проблем передавати зображення між окремими блоками програми без додаткової обробки. При виникненні ситуацій збою в керуючому блоці передбачений механізм примусового перекодування зображення. Проте дана ситуація може виникнути в випадку додавання нових модулів або структур даних. Процес обробки зображення на 90% автоматизований, користувач тільки спостерігає за отриманими результатами та проводить візуальну оцінку. Зменшення впливу людини на процес обробки дозволяє прискорити дану обробку та створювати цілі групи для послідовної обробки файлів..

Додатковим елементом в процесі модулювання програмного додатку було проведення дослідження та опис створених власних класів для опису користувача та самої системи. Запропоновані класи дозволяють в повній мірі створити гнучку систему блоків та дозволити більш простішу зміну архітектури програмного додатку. На рисунку 3.6 наведена діаграма класів реалізованої програми. На зображенні наведено лише структурно значимі класи які були розроблені в процесі проектування програмного засобу. Для реалізації другорядних елементів програмної системи було використано набір стандартних структур, класів та інтерфейсів, які реалізовані в OpenCV та Visual Studio. Запозичені класи та структури в основному використовувались для створення графічного інтерфейсу користувача, налаштування взаємодії з операційною системою (на рівні завантаження сторонніх файлів) та організації роботи зовнішніми апаратними пристроями (на рівні запуску та отримання цифрових зображень з камери). Використання для другорядних операцій вже готових методів дозволило зменшити час на реалізацію програми.

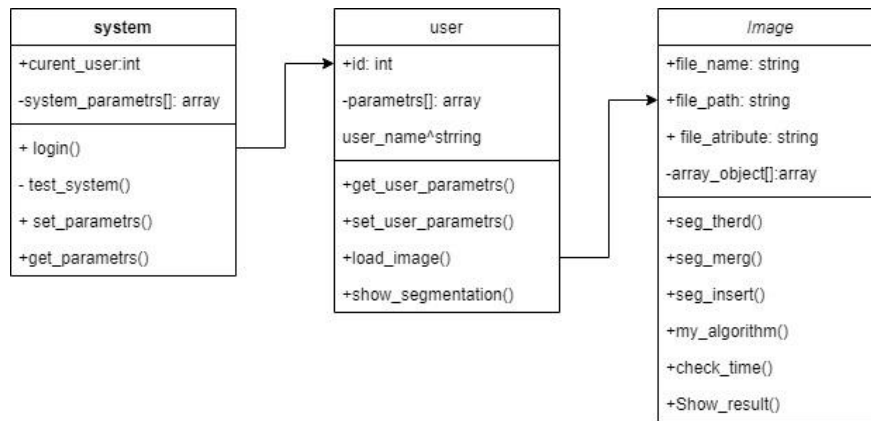


Рисунок 3.6 – Діаграма класів програмної системи

В процесі модулювання, що включав в себе аналіз програмної структури на трьох рівнях було доведено правильність обраної ієрархічно-модульної структури, розробленої власно системи класів та структур даних а також можливості роботи користувача з програмною системою. Моделювання показало, що усі програмні модулі виконують поставлені завдання не створюють конфліктних ситуацій та можуть використовуватись під час програмної реалізації програмної системи сегментації цифрових зображень. Усі отримані результати були відповідним чином проілюстровані, оцінені та наведені в роботі.

Окремою задачею яка ставилась в процесі проектування програмної системи була розробка графічного інтерфейсу користувача. Під час роботи над даною задачею було проаналізовано вже існуючі програми-аналоги з метою оцінки програмних інтерфейсів за різними критеріями, серед яких:

- зрічність роботи;
- швидкість доступу до функцій;
- простота роботи з системою;
- загальний зовнішній вигляд.

В результаті проведеного аналізу було запропоновано власний дизайн графічного інтерфейсу користувача для програмного модуля сегментації зображень (рисунок 3.7). Даний дизайн є одночасно і простим і функціонально повним.

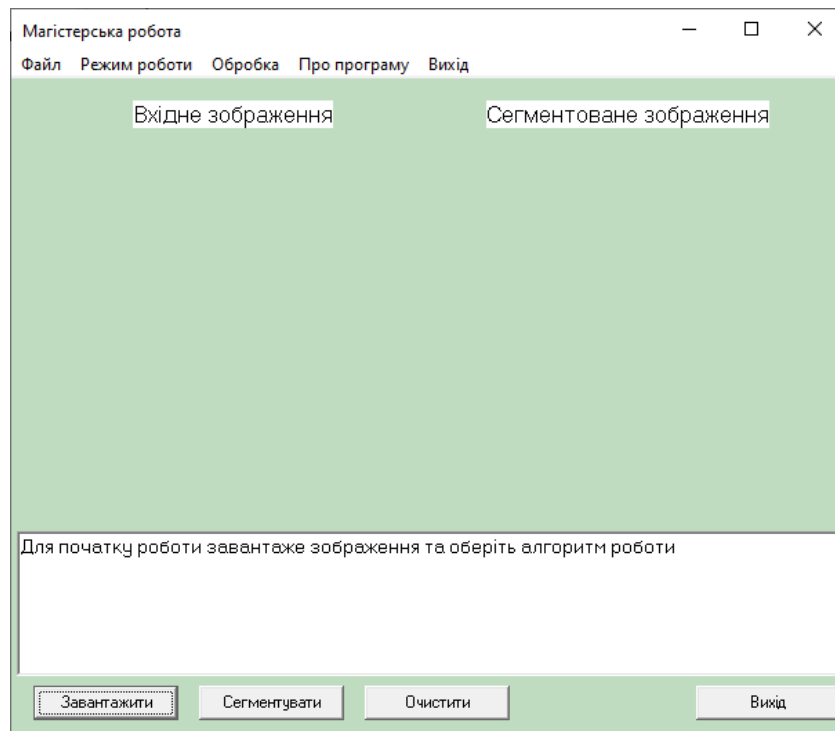


Рисунок 3.7 – Графічний інтерфейс користувача програмної системи обробки цифрових зображень

Під час проектування на головному вікні програмної системи було розміщено тільки функціонально важливі елементи, які найчастіше буде використовувати потенційний користувач. Це з однієї сторони дозволило збільшити корисну площу для розширення місця для відображення зображень, а з іншої сторони зменшили кількість елементів які повинен опанувати користувач для роботи з запропонованою системою. Якщо користувачеві має досвід роботи з системами даного типу то для активації інших пунктів меню він може скористатись випадаючим меню, що є цілком зручно і не обмежує користувача в функціональних можливостях.

Розроблена структура передбачає мінімізацію стручання користувача в робочий процес. Оскільки в програмі реалізовані спеціалізовані алгоритми обробки зображень, то для їх правильного і повного функціонування потрібно мати певний рівень досвіду та розуміти призначення усіх параметрів функціонування. Послідовність мінімально необхідних кроків для роботи з програмним додатком візуально відображено на рисунку 3.8:

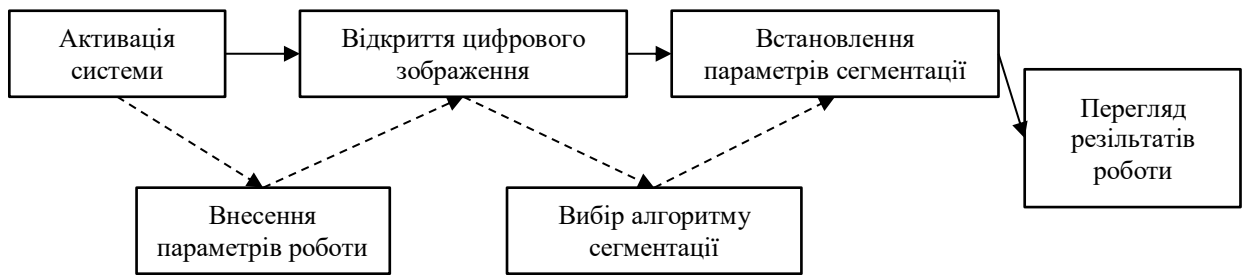


Рисунок 3.8 – Послідовність вказівок користувача для досягнення поставлених задач розбиття цифрового зображення на однорідні області

В системі реалізовано дружній до користувача графічний інтерфейс, передбачено систему підказок, а також введено мінімальну кількість активних елементів на головному вікні програмної розробки. Такий дизайн сприятиме високій зручності роботи програмного додатку, швидкому опануванню навичок роботи при обробці масивів даних, а також мінімалізації впливу користувача на процес роботи програми.

3.2 Програмні функції сегментації зображень

При програмній реалізації розробленої структури та алгоритмів окрім основних алгоритмів було реалізовано і ряд додаткових. Вони носять допоміжний характер та виконуються в більшій мірі на етапах попередньої обробки зображень. Серед таких алгоритмів можна відмітити алгоритми порогової обробки зображення. Порогова обробка – це дуже популярна техніка сегментації, яка використовується для відокремлення об'єкта від фону.

Процес визначення порогу включає порівняння кожного пікселя зображення (інтенсивності пікселя) із заданим порогом. Це ділить всі пікселі вхідного зображення на 2 групи:

- пікселі зі значенням інтенсивності, нижчим за поріг;
- пікселі зі значенням інтенсивності більше ніж порогове значення.

Ці 2 групи тепер мають різні значення, залежно від різних типів сегментації. OpenCV підтримує 5 різних схем порогової обробки на зображеннях у відтінках сірого (8-бітових) за допомогою функції:

```
Double threshold(InputArray src, OutputArray dst, double
thresh, double maxval, int type)
```

Параметри даної функції:

- InputArray src: вхідне зображення (матове, 8-розрядне або 32-розрядне);
- OutputArray dst: вихідне зображення (такий же розмір, що і вхідний);
- double thresh: встановити порогове значення;
- double maxval: maxVal, максимальне значення порогу;
- int type*: Визначає тип порогу, який буде використано.

Приклад програмного коду реалізації алгоритму порогової обробки на мові програмування C++ наведено на рисунку 3.9.

```
int threshold_value = 0;
int threshold_type = 2;
int maxVal = 255;
Mat src = imread(argv[1], 1);
cvNamedWindow("Original", CV_WINDOW_NORMAL);
imshow("Original", src);
Mat src_gray, dst;
cvtColor(src, src_gray, CV_BGR2GRAY);
cvNamedWindow("Result", CV_WINDOW_NORMAL);
createTrackbar("Threshold", "Result", &threshold_value, 255);
while (1)
{
    threshold(src_gray, dst, threshold_value, maxVal, threshold_type);
    imshow("Result", dst);
    waitKey(1);
}
```

Рисунок 3.9 – Програмна реалізація алгоритму порогової сегментації на мові C++ з використанням бібліотеки OpenCV

В даній реалізації було створено дві функції:

`cvNamedWindow()` – створює вікно, яке можна використовувати як заповнювач для зображень і трекбарів. Створені вікна посилаються за їх іменами.

`imshow` – Відображає зображення у зазначеному вікні. Якщо вікно було створено з прапорцем `cv::WINDOW_AUTOSIZE`, зображення відображається з його оригінальним розміром, однак воно все ще обмежене роздільною здатністю екрана. В іншому випадку зображення масштабується відповідно до вікна. Функція може масштабувати зображення залежно від його глибини:

- Якщо зображення 8-розрядне без знака, воно відображається як ϵ .
- Якщо зображення 16-бітове без знака, пікселі діляться на 256. Тобто діапазон значень $[0, 255 * 256]$ відображається на $[0, 255]$.
- Якщо зображення є 32-бітним або 64-бітним з плаваючою комою, значення пікселів помножуються на 255. Тобто діапазон значень $[0, 1]$ відображається на $[0, 255]$.
- 32-розрядні цілі зображення більше не обробляються через неоднозначність необхідного перетворення. Перетворіть у 8-розрядну беззнакову матрицю за допомогою спеціальної попередньої обробки, яка залежить від контексту зображення.

Перетворює зображення з одного колірного простору в інший.

`cvtColor()` – Функція перетворює вхідне зображення з одного колірного простору в інший. У разі перетворення з колірного простору RGB порядок каналів має бути вказано явно (RGB або BGR). Зауважте, що формат кольору за замовчуванням в OpenCV часто називають RGB, але насправді це BGR (байти змінюються на інше). Таким чином, перший байт у стандартному (24-бітовому) кольоровому зображенні буде 8-бітовим синім компонентом, другий байт буде зеленим, а третій байтом буде червоним. Четвертий, п'ятий і шостий байти будуть тоді другим пікселем (синім, потім зеленим, потім червоним) і так далі.

Звичайні діапазони значень каналів R, G і B:

- від 0 до 255 для зображень `CV_8U`;
- від 0 до 65535 для зображень `CV_16U`;

- від 0 до 1 для зображень CV_32F.

`createTrackbar` – Функція `createTrackbar` створює панель відстеження (повзунок або елемент керування діапазоном) із зазначеним ім'ям та діапазоном, призначає значення змінної для положення, синхронізоване з панеллю треків, і визначає функцію зворотного виклику `onChange`, яка буде викликатися при зміні положення панелі треків. Створена трекбар відображається у вказаному вікні `winname`.

`Threshold` – Першим аргументом є вихідне зображення, яке має бути зображенням у відтінках сірого. Другий аргумент — це порогове значення, яке використовується для класифікації значень пікселів. Третій аргумент — це `maxVal`, який представляє значення, яке потрібно надати, якщо значення пікселя більше (іноді менше) порогового значення. `OpenCV` надає різні стилі порогового значення, і це визначається четвертим параметром функції. Різні типи:

- `cv2.THRESH_BINARY`;
- `cv2.THRESH_BINARY_INV`;
- `cv2.THRESH_TRUNC`;
- `cv2.THRESH_TOZERO`;
- `cv2.THRESH_TOZERO_INV`.

Іншим алгоритмом сегментації який використаний в програмній системі є сегментація зображень на основі алгоритму водоподілу. На зображенні 3.10 наведено програмний код підготовчого етапу роботи даного алгоритму. Зображення проілюстровано процес завантаження зображення та створення додаткових масивів для зберігання тимчасової інформації. Також в програмі використана функція `findContours()` яка використовується для отримання масиву контурів майбутніх однорідних областей. При чому в програмі реалізовано додаткова перевірка виділення контурів, якщо контурів не виявлено, то процес виводиться повідомлення та користувачу пропонується внести корективи в роботу програми.

```

srcImage.copyTo(g_srcImage);
imshow("image", g_srcImage);
}

int i, j, compCount = 0;
vector<vector<Point> > contours;
vector<Vec4i> hierarchy;

findContours(g_maskImage, contours, hierarchy, RETR_CCOMP, CHAIN_APPROX_SIMPLE);
if (contours.empty())
    continue;

Mat maskImage(g_maskImage.size(), CV_32S);
maskImage = Scalar::all(0);
Mat watershedImage(maskImage.size(), CV_8UC3);

```

Рисунок 3.10 – Програмна реалізація алгоритму водоподілу на мові C++ з використанням бібліотеки OpenCV

Концепція вододілу заснована на тривимірній візуалізації зображень: два – координати, а інший – рівень сірого. Виходячи з такого тлумачення «топографії», ми розглядаємо три типи точок:

- Точки, що належать до локального мінімуму, також можуть мати мінімальну поверхню, в якій усі точки є мінімальними.
- Коли крапля води поміщається в точку, вода повинна впасти до однієї мінімальної точки.
- Коли вода знаходиться в точці, вона з однаковою ймовірністю тече в більш ніж одну з цих мінімальних точок.

Сукупність точок, що задовольняють другій умові для даного регіонального мінімуму, називається водозбірним басейном або вододілом мінімуму. Сукупність точок, що задовольняють третій умові, становить лінію піку топографічної поверхні, яку називають «лінією поділу» або «лінією поділу».

Сегментація вододілу – метод математичної морфологічної сегментації, заснований на топологічній теорії. Сірий колір кожного пікселя на зображенні представляє висоту точки. Основна ідея алгоритму моделювання повені полягає в тому, щоб припустити, що свердловина просвердлена в положенні мінімального значення в кожній області, і вода виливається з свердловини з рівномірною швидкістю, занурюючи всю місцевість від низького до високого. Коли вода в різних конвергентних басейнах має бути агрегована, споруджені

греблі запобігатимуть агрегації. Вода досягне точки, де на ватерлінії можна буде побачити лише вершини дамб. Межа цих дамб відповідає лінії розмежування вододілу. Таким чином, вони є (безперервними) граничними лініями, виділеними за алгоритмом вододілу (рисунок 3.11).

```
for (i = 0; i < maskImage.rows; i++)
    for (j = 0; j < maskImage.cols; j++)
        {
            if(maskImage.at<int>(i, j)>index1)
                index1 = maskImage.at<int>(i, j);
        }
for (i = 0; i < maskImage.rows; i++)
    for (j = 0; j < maskImage.cols; j++)
        {
            int index = maskImage.at<int>(i, j);
            if (index == -1)
                watershedImage.at<Vec3b>(i, j) = Vec3b(255, 255, 255);
            else if (index <= 0 || index > compCount)
                watershedImage.at<Vec3b>(i, j) = Vec3b(0, 0, 0);
            else if (index ==index1)
                watershedImage.at<Vec3b>(i, j) = Vec3b(255, 255, 255);
            else
                watershedImage.at<Vec3b>(i, j) = Vec3b(index*10, 0, 0);
        }
imshow(WINDOW_NAME2, watershedImage);
```

Рисунок 3.11 – Програмна реалізація алгоритму вододілу на мові C++ з використанням бібліотеки OpenCV

Реалізовані в даній роботі алгоритми сегментації дозволяють у повній мірі провести процес порівняння результатів роботи запропонованого алгоритму з відомими аналогами. А використання бібліотеки OpenCV забезпечує високий рівень застосованих алгоритмів для більш об'єктивної оцінки отриманих результатів.

3.3 Тестування та аналіз реалізованої системи

На останньому етапі роботи з програмою обробки та аналізу цифрових зображень відбувся процес тестування та порівняльного аналізу запропонованого алгоритму з загальновідомими алгоритмами сегментації зображень. Для проведення даних експериментів було використано робочу станцію з такими технічними характеристиками (рисунок 3.12):

Процесор: Intel Core i3-7100 1 шт.
Чіпсет: Intel® B250.
ОЗУ: 8GB (1x4GB) DIMM DDR4 (частота шини 2400MHz) Crucial, об'єм пам'яті (макс.) 32GB, слотів пам'яті 2шт.
Диски: 500GB (1x500GB) SATA 6Gb/s 3.5" Seagate.
Відеокарта: ASUS, серія: GeForce GT 730, об'єм відео-пам'яті 2GB, GT730, встановлено 1 шт.
Мережа: на мат. платі: 1Gb/s 1шт.
Блок живлення: 1шт. 450W.
Виробник: Gigabyte.

Рисунок 3.12 – Технічні параметри персонального комп'ютера для тестування програмної системи

Оскільки отримані результати будуть додатково оцінюватись візуально, то для більшої зручності під час етапу тестування було використано монітор з технічними характеристиками наведеними на рисунку 3.13.

```
матриця: TN;  
роздільна здатність: 1600×900 (16:9);  
частота поновлення кадрів: 75 Гц;  
яскравість: 250 кд/м²;  
підсвічування: WLED;  
входи: HDMI 1.4, DisplayPort 1.2, VGA (D-Sub);  
розміри: 570 * 446 * 226 мм.
```

Рисунок 3.13 – Технічні параметри монітора для тестування програмної системи

Технічні особливості обраної тестової станції дозволили провести усі заплановані тести з усіма групами тестових вибірок та не мали вирішальний вплив на отримані результати. Це дозволило провести оцінку запропонованих алгоритмів та програмних структур ігноруючи вплив сторонніх факторів. А також додатково провести порівняння реалізованої програми з програмами такого ж сегмента які присутні сьогодні на ІТ ринку.

Під час проведення тестування було використано декілька варіантів різних груп цифрових зображень, що відрізняються між собою певним набором технічних особливостей. Для оцінки було запропоновано використати групи зображень з такими параметрами:

– зображення з чітко відмінними однорідними областями. Однорідні області на зображеннях повинні відрізнятися за всіма характеристиками кольору, яскравості тощо. Дана тестова група призначена не скільки для оцінки роботи різних алгоритмів, скільки для отримання стартових значень для подальшого порівняння та аналізу, оскільки дані розраження будуть мати ідеальні хірикеристики для проведення сегментації: чітко виражені границі між областями та велику різницю в значенні кольору;

– зображення з подібними кольорами але границі між областями додатково підкреслені. На зображеннях даної групи додатково буде нанесено маркування по лініях меж між областями. Дана група зображень використовуватиметься для перевірки роботи на зображеннях з складнішими умовами сегментації;

– зображення з подібними областями з слабо вираженими границями. Дана тестова вибірка повинна проілюструвати можливість роботи запропонованого алгоритму у складних випадках сегментації. Оскільки більшість алгоритмів слабо справляється з сильно подібними областями, найбільша кількість хибно виділених результатів сегментації припадає саме на них.

Вибір саме таких груп зображень для формування тестових вибірок базується на тому, щоб отримати об'єктивні результати оцінки роботи запропонованого алгоритму та алгоритмів аналогів. Велика відмінність між тестовими екземплярами призведе до того, що проведені тестові сегментації у повній мірі зможуть продемонструвати можливості запропонованого алгоритму, виділити його сильні та слабкі сторони.

Іншими чинниками які мігли повпливати на результати тестування є зовнішні фактори роботи апаратних засобів. Для зменшення впливу та для більшої об'єктивності отриманих даних процеси сегментації запускались для кожного зображення по 10 разів, а отримані результати піддались усередненню.

Порівняння відбувалось з оцінкою точності проведеної процедури сегментації, при якій обчислювався відмокок коректно розмічений точок в порівнянні з експертною оцінкою. Отримані результати тестування різних груп зображень було узагальнено та представлено у вигляді таблиці (таблиці 3.1).

Таблиця 3.1 –Результати тестування алгоритм/тип зображення

Алгоритм сегментації/ тип зображень	Перший тип зображень	Другий тип зображень	Третій тип зображень
Порогова сегментація	99	83	68
Сегментація водоподілом	99	94	82
Сегментація k-means	99	95	86
Сегментація нарощуванням областей	99	91	80
Запропонований алгоритм	99	94	85

При аналізі отриманих результатів можна зробити наступні висновки. Зображення першої групи були відсегментовані без значних похибок, а невелика похибка носить скоріш випадковий характер, оскільки під час сегментації зображень експертом траплялись випадки випадкового зафарбування точок з сусідніх областей. Тому всі алгоритми показали фактично 100% подібність з експертними результатами. При обробці зображень другої групи результати різних алгоритмів відрізняються, оскільки області на зображеннях були подібними то очевидно що алгоритм порогової сегментації знаходив поріг не завжди коректно, що в загальному вплинуло на отримані результати. Інші алгоритми показали приблизно рівні результати. Під час обробки зображень третьої групи рівень хобно виділених точок значно зріс, це можна пояснити тим фактором, що однорідні області були дуже подібними за своїми характеристиками і алгоритми не завжди могли коректно налаштувати параметри своєї роботи. Візуально результати тестування представлені у вигляді графіка (рисунок 3.14).

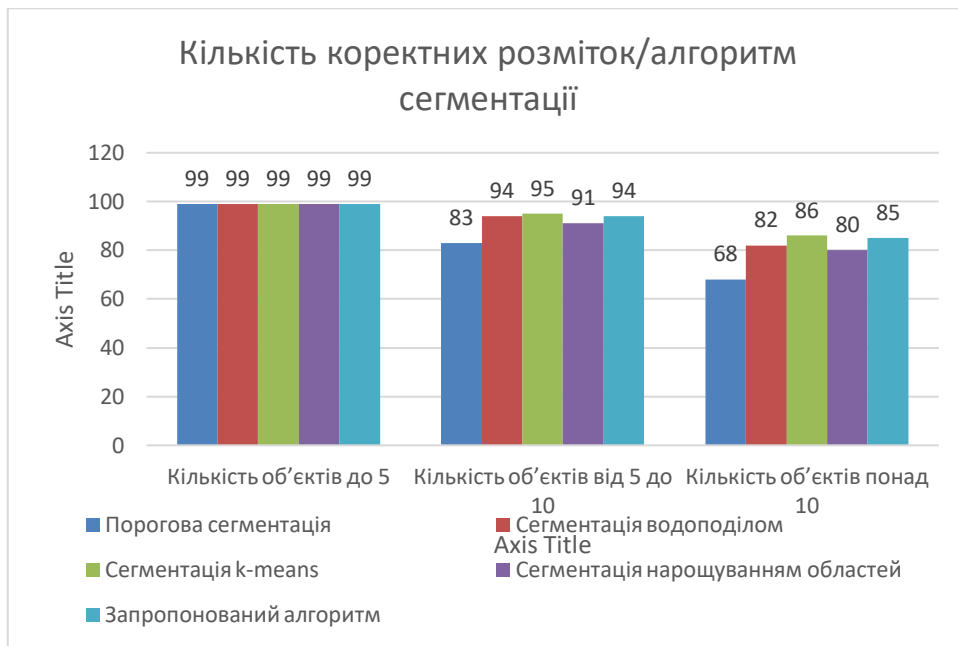


Рисунок 3.14 – Порівняння результатів роботи алгоритмів сегментації

Аналізуючи результати роботи програмної системи було відмічено ряд факторів які вносили корективи в роботу програмної системи. Дані фактори та рівень їхнього впливу наведено в таблиці 3.2.

Таблиця 3.2 – Узагальнена таблиця факторів впливу на результати роботи

Фактори впливу/ тип масиву	Чіткість границь	Зашумленість зображення	Попередня обробка
Порогова сегментація	сильно впливає	сильно впливає	сильно впливає
Сегментація водоподілом	сильно впливає	сильно впливає	має вплив
Сегментація k-means	впливає	впливає	впливає критично
Сегментація нарощуванням областей	сильно впливає	сильно впливає	має вплив
Запропонований алгоритм	впливає	сильно впливає	впливає

Отримані результати під час тестування підтвердили гіпотези, що були зроблені під час моделювання. Отримані дані та результати є подібними до результатів програм з подібним спектрним вирішуваних задач та набором закладених функцій. Тому можна зробити висновок, що реалізована програмна

система обробки та аналізу цифрових зображень з використання алгоритму аналізу малоінформативних областей є актуальною та має практичну цінність.

3.4 Висновки до розділу

Запропоновано блокову структуру програмного модуля сегментації та оцінки цифрових зображень, на основі ієрархічно-модульного підходу, що в подальшому дозволило програмно реалізувати систему обробки та аналізу цифрових зображень.

Здійснено випробування реалізованої програмної системи на основі використання різних груп цифрових зображень та алгоритмів сегментації, що дозволило проаналізувати точність результатів роботи запропонованого алгоритму та провести порівняльну оцінку з аналогами.

ВИСНОВКИ

На основі аналізу сучасних програмних систем обробки, аналізу та зберігання цифрових зображень та на основі аналізу алгоритмів сегментації можна зробити такі висновки:

1. Проаналізовано та проведено класифікацію існуючих типів цифрових зображень на основі їх характеристик, що дозволило виділити групи зображень для обробки в системі аналізу цифрових зображень.

2. Проаналізовано основні задачі цифрової обробки зображень та проведено їх групування на основі подібності отриманих результатів, що дозволило виділити основні напрямки розвитку цифрової обробки зображень.

3. Проведено огляд програмних систем обробки зображень з відкритим кодом на основі аналізу їх структури та набору функціональних можливостей, що дозволило виділити структурні блоки та алгоритми які є головними в програмах даного класу

4. Досліджено алгоритми сегментації зображень та проведено їх оцінку на основі суб'єктивних та об'єктивних критеріїв, що дозволило виділити групи алгоритмів на основі яких і було спроектовано алгоритм сегментації на основі аналізу малоінформативних областей.

5. Розроблено алгоритм сегментації на основі аналізу малоінформативних областей, що надало можливість в подальшому спроектувати структуру системи обробки та аналізу зображень з модулем автоматизованої сегментації.

6. Здійснено випробування реалізованої програмної системи на основі використання різних груп цифрових зображень та алгоритмів сегментації, що дозволило проаналізувати точність результатів роботи запропонованого алгоритму та провести порівняльну оцінку з аналогами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Bradski G. Learning OpenCV - Computer Vision with the OpenCV Library / G. Bradski.,2018 – 580 с.
2. Гонсалес Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс. - М.: Техносфера, 2016. – 616 с.
3. Сафонов В. О. Параметризованные типы данных. История, теория, реализация и применение / В.О. Сафонов. – М.: Издательство Санкт-Петербургского университета, 2013. – 116 с.
4. Секунов Н.Ю. Самоучитель Visual C++ 6.0 / Н.Ю. Секунов. – М.: СПб: ВHV, 2014. - 960 с.
5. Страуструп Б. Язык программирования C++ / Б. Страуструп. – М.: Радио и связь,2019. – 350 с.
6. Страуструп Бьерн Дизайн и эволюция C++ / Бьерн Страуструп. – М.: ДМК Пресс, 2016. – 446 с.
7. Сойфер В.А. Методы компьютерной обработки изображений / В.А. Сойфер. – М.: Физматлит,2013. – 784 с.
8. Оппенгейм А. Цифровая обработка сигналов / А. Оппенгейм, Р. Шафер. – М.: Техносфера,2006. – 856 с.
9. Прэтт У. Цифровая обработка изображений. В 2-х книгах: пер. с англ. / У. Прэтт. – М.: Мир,2009. – 792 с.
10. Грузман И.С. Цифровая обработка изображений в информационных системах: учеб, пособие / И.С. Грузман, В.С. Киричук., 2002. – 352 с.
11. Яне Б. Цифровая обработка изображений / Б. Яне. - М.: Техносфера, 2007. – 584 с.
12. Селянкин В.В. Компьютерное зрение. Анализ и обработка изображений: учебное пособие / В. В. Селянкин.,2019 – 152с.
13. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт.,2004. – 928 с

14. Поляков А. Методы и алгоритмы компьютерной графики / А. Поляков., 2003. – 545 с.
15. Безрядин С. Н. Преобразование яркости в программном обеспечении / С. Н. Безрядин., 2006 – 211с.
16. Баженова И. Ю. Язык программирования С++ / И. Ю. Баженова., 2017. – 366 с.
17. Бартлетт Н. Программирование на Delphi Путеводитель / Н. Бартлетт., 2016. – 116с.
18. Вебер Дж. Технология С++ в подлиннике / Дж. Веберю., 2017. – 256с.
19. Волш А. И. Основы программирования на С++ для World Wide Web / А. И. Волш., 2016. – 458с.
20. Абрамов С. А. Задачи по программированию / С. А. Абрамов., 2018. – 256 с.
21. Березин Б.И. Начальний курс Delphi / Б. И. Березин., 2016. – 331с.
22. Бондарев В.М. Основі програмування / В. М. Бондарев., 2017. – 446с.
23. Вирт Н. Алгоритмі и структурі даних / Н. Вирт, 2019. – 345с.
24. Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие / В. П. Гладков., 2014. – 516с.
25. Грогоно П. Программирование на языке Delphi / П. Грогоно., 2012. – 216с.
26. Дагене В.А. 100 задач по программированию / В. А. Дагене., 2013. – 106с.
27. Джамса К. Библиотека программиста Java / К. Джамса., 2016. – 656с.
28. 2D Image Features Detector And Descriptor Selection Expert System / I. Merino, J. Azpiazu, A. Remazeilles, A. Sierra, 2019. – С. 51–61.
29. Заварикин В.М. Основі информатики и вычислительной техники / В. М. Заварикин., 2019. – 556с.
30. Кен А. Язык программирования С++ / А. Кен, 2017. – 378с.
31. Керниган Б. Язык программирования Delphi. / Б. Керниган., 2012. – 391с.

32. Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники. / В.Ф. Ляхович., 2014. – 127с.
33. Мейнджер Дж. С++ Основы программирования / Дж. Мейнджер., 2017. – 346с.
34. Миков А. И. Информатика. Введение в компьютерные науки / А. И. Миков., 2018. – 442с.
35. Могилев А. В. Информатика: Учеб. пособие для студ. пед. Вузов / А. В. Могилев., 2019. – 629с.
36. Нотон П. JAVA:Справ.руководство / П. Нотон., 2016. – 447с.
37. Нотон П. Полный справочник по Java / П. Нотон.,2017. – 556с.
38. Ренеган Э.Дж. 1001 адрес WEB для программистов :Новейший путеводитель программиста по ресурсам World Wide Web / Э. Дж. Ренеган., 2017. – 512с.
39. Родли Дж. Создание Java-апплетов / Родли Дж., 2016. – 466с.
40. Томас М. Секреты программирования для Internet на Java / М. Томас., 2017. – 396с.
41. Семакина И. Г. Информатика. Задачник-практикум: В 2 томах / И. Г. Семакина., 2019. – 476с.
42. Сокольский М.В. Все об Intranet и Internet / М. В. Сокольский.,2018. – 254с.
43. Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ / Д. Тассел,2011. – 56с.
44. Тюрин Ю.Н. Анализ данных на компьютере / Ю. Н. Тюрин., 2015. – 384с.
45. Флэнэген Д. Java in a Nutshell / Д. Флэнэген, 2018. – 473с.
46. Чен М.С. Программирование на С++:1001 совет:Наиболее полное руководство по С++ и Visual С++ / М. С. Чен., 2017. – 640с.
47. Эферган М. С++: справочник. QUE Corporation / М. Эферган.,1998. – 256с.

48.Sinha P. Perceiving and Recognizing threedimensional forms / P. Sinha., 2016. – 278p.

49.Pablo F. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces / F. Pablo.,2013. – 187с.

50.Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня “Магістр”. Спеціальність „Комп’ютерні системи та мережі”/О.М. Березький, Л.О. Дубчак, Г.М. Мельник /Під ред. О.М. Березького Тернопіль: ТНЕУ, 2018. – 41с.