

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

Шевчук Владислав Васильович

**«Алгоритм опису сцен на зображеннях на основі
контурного аналізу / Algorithm for describing a scene
on a digital image based on contour analysis
mechanisms»**

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

Виконав студент групи КІм-21
В.В. Шевчук

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

Кваліфікаційну роботу допущено
до захисту:

" ____ " _____ 20____ р.

Завідувач кафедри
_____ Л. О. Дубчак

Тернопіль – 2023

РЕЗЮМЕ

Кваліфікаційна робота на тему “Алгоритм опису сцен на зображеннях на основі контурного аналізу” зі спеціальності 123 «Комп’ютерна інженерія» освітнього ступеня «магістр» написана обсягом 85 сторінок та містить 26 ілюстрацій, 2 таблиці, 2 додатки та 60 джерел за переліком посилань.

Метою роботи є розробка алгоритму опису сцен на зображеннях на основі контурного аналізу.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано методи: обробки цифрових зображень (для попередньої опрацювання вхідних зображень); контурного аналізу (для опису окремих об’єктних областей на зображенні); об’єктно-орієнтованого програмування (для проектування програмного додатку аналізу та опису цифрових зображень).

Результати дослідження: алгоритм оцінки опису сцени на цифровому зображенні, програмний додаток обробки та опису цифрових зображень на основі механік контурного аналізу.

Результати роботи можуть бути використані в створенні нових систем розпізнавання даних та моніторингу, системах архівації даних, для наукових досліджень та в навчальному процесі.

Орієнтовні напрямки розвитку досліджень: розроблення алгоритмів автоматичного виділення шумів на цифрових зображеннях, створення нових програмних засобів та моделей для навчання.

КЛЮЧОВІ СЛОВА: КОНТУРНИЙ АНАЛІЗ, ГРАНИЦІ ОБ’ЄКТІВ, КОДУВАННЯ КОНТУРІВ, АПРОКСИМАЦІЯ ЛІНІЙ.

RESUME

Graduate qualification work on “Algorithm for describing a scene on a digital image based on contour analysis mechanisms” specialty 123 – Computer Engineering is 85 pages long and contains 26 illustrations, 2 tables, 2 appendices and 60 references.

The aim of the work is to develop an algorithm for describing scenes on images based on contour analysis.

Research methods. To solve the tasks in the qualification work, the following methods were used: digital image processing (for preliminary processing of input images); contour analysis (to describe individual abnormal areas in the image); object-oriented programming (for designing a software application for the analysis and description of digital images).

Research results: an algorithm for evaluating the description of a scene on a digital image, a software application for processing and describing digital images based on the mechanics of contour analysis.

The results of the work can be used in the creation of new data recognition and monitoring systems, data archiving systems, for scientific research and in the educational process.

Indicative directions of research development: development of algorithms for automatic selection of noise on digital images, creation of new software tools and models for learning.

KEY WORDS: CONTOUR ANALYSIS, BOUNDARIES OF OBJECTS, ENCODING CONTOURS, APPROXIMATION OF LINES.

ЗМІСТ

Вступ.....	7
1 Системи та засоби опрацювання цифрових зображень	10
1.1 Класифікація та характеристики цифрових зображень	10
1.2 Опис та класифікація зображень	18
1.3 Програмні засоби обробки та опису цифрових зображень	24
1.4 Постановка задач дослідження.....	28
1.5 Висновки до розділу	29
2 Методи та алгоритми обробки кольорових цифрових зображень	30
2.1 Методи та алгоритми обробки опису цифрових зображень.....	30
2.2 Алгоритми контурного аналізу цифрових зображень.....	39
2.3 Алгоритм опису об'єктів на основі виділення контурів	48
2.4 Висновки до розділу	50
3 Програмний додаток аналізу та опису цифрових зображень	51
3.1 Структура програмного додатку опрацювання цифрових зображень	51
3.2 Модулі програмного додатку аналізу та оцінки програмного коду	59
3.3 Тестування та аналіз реалізованого програмного додатку	64
3.4 Висновки до розділу	72
Висновки	73
Список використаної літератури	74
Додаток А Лістинг реалізації алгоритмів цифрової обробки зображень	79
Додаток Б Світлокопії виданих публікацій.....	82

ВСТУП

Актуальність роботи. Цифрове зображення – це зображення, інформація на яких записана камерою або датчиком і представлена візуально в друкованому або електронному вигляді. Зображення важливі, оскільки вони можуть бути надзвичайно ефективним засобом для зберігання та передачі інформації. Фотографії допомагають людям створювати постійні записи свого візуального досвіду, а також ділитися цим досвідом з іншими. Фотографія, як правило, надає необхідну інформацію, яка в іншому випадку може вимагати тривалого, нудного та неоднозначного словесного опису сфотографованої території. Зазвичай люди покладаються на очі, щоб отримати інформацію, пов'язану з навколишнім середовищем, а мозок вміє обробляти візуальні дані. Тому дуже влучно сказано, що фотографія чи малюнок варті тисячі слів.

Процес отримання фотографії подібний до процесу нормального людського зору. І в людському зорі, і в фотографії для освітлення сцени потрібне джерело світла. Світло взаємодіє з об'єктами на сцені, і відбите світло досягає спостерігача, після чого його виявляють очі у випадку людського зору або фотоапарат у разі фотографії. Інформація про об'єкти сцени записується як варіації інтенсивності та кольору виявленого світла. Важливо відзначити, що хоча сцена є тривимірною за своєю природою, зображення цієї сцени завжди є двовимірним.

Обробка зображень це загальний термін для широкого діапазону методів, які використовуються для маніпулювання, модифікації або класифікації зображень різними способами. Загалом, цифрове зображення, отримане через супутник або цифрову камеру, використовується для аналізу за допомогою комп'ютерів, і тому термін цифрова обробка зображень.

Загалом цифрова обробка зображень означає обробку двовимірного зображення в цифровій формі цифровим комп'ютером. Цифрове зображення можна визначити як масив дійсних або комплексних чисел, представлених

кінцевою кількістю бітів. Зображення можна отримати в процесі сканування існуючої фотографії за допомогою сканера, цифрової камери або цифрового датчика на борту супутника чи літака. Потім це цифрове зображення можна обробити та/або відобразити на моніторі комп'ютера з високою роздільною здатністю.

Тому обрана в кваліфікаційній роботі задача створення та реалізації алгоритму опису сцен на цифрових зображеннях за допомогою виділення контурів об'єктів є актуальною.

Метою роботи є розробка алгоритму опису сцен на зображеннях на основі контурного аналізу.

Для досягнення даної мети ставились наступні завдання:

- провести дослідження цифрових зображень та виділити їх основні характеристики;
- провести аналіз технологій опису цифрових зображень;
- провести дослідження сучасних програм обробки цифрових зображень;
- проаналізувати алгоритми виділення контурів об'єктів;
- розробити алгоритм опису сцен на цифрових зображеннях на основі виділення контурів об'єктів;
- реалізувати програмний додаток обробки та опису цифрових зображень.

Об'єкт дослідження – процес обробки цифрових зображень.

Предмет дослідження – методи і алгоритми виділення контурів об'єктів.

Наукова новизна одержаних результатів визначається наступним чином:

- проведено дослідження алгоритмів контурного аналізу на основі оцінки точності вибілення контурних ліній, що дозволило виділити ряд характеристик які максимально впливають на отримані результати;
- розроблено алгоритм опису сцен на цифрових зображеннях на основі виділення контурів окремих об'єктів, що дозволило зменшити розміри пам'яті для зберігання.

Практична цінність одержаних результатів полягає в тому, що:

- розроблено структуру та здійснено процес моделювання програмного додатку аналізу та опису сцен на цифрових зображеннях, що дозволило програмно реалізувати розроблені алгоритми та структури даних;

- реалізовано програму для аналізу та опису цифрових зображень з використанням механізмів контурного аналізу на основі об'єктно-орієнтованого підходу що дозволило провести тестування запропонованих програмних рішень.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційній роботі, підготовлено тези доповіді «Алгоритм опису об'єктів цифрових зображень на основі контурного аналізу» обсягом 1 сторінка на VIII Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», а також «Алгоритм сегментації цифрових зображень на основі аналізу характеристик однорідних областей» обсягом 1 сторінка на VIII Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі».

В першому розділі здійснено дослідження та опис характеристик цифрових зображень та програмних засобів аналізу та опису цифрових зображень.

В другому розділі було досліджено методи та алгоритми виділення та кодування контурів об'єктів, а також запропоновано алгоритм опису сцен на основі опису контурів.

В третьому розділі описаний процес проектування, моделювання, програмної реалізації та тестування програмного додатку аналізу та опису цифрових зображень.

1 СИСТЕМИ ТА ЗАСОБИ ОПРАЦЮВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1 Класифікація та характеристики цифрових зображень

У інформатиці розуміння представлення зображення є ключовим, оскільки комп'ютери сприймають зображення як масив значень пікселів. У цифровому всесвіті зображення можна описати як двовимірне представлення сцени, що складається з окремих елементів, які називаються пікселями. Кожен піксель містить інформацію про колір та інтенсивність певної точки зображення.

Піксель або «елемент зображення» є основною одиницею цифрового зображення, яке часто представлено за допомогою комбінацій червоного, зеленого та синього (RGB).

Якщо проаналізувати типи зображень, то в комп'ютерній техніці можна виділити три основних підходи до кодування зображень та один гібридний. Приклади класифікації зображення наведено на рисунку 1.1

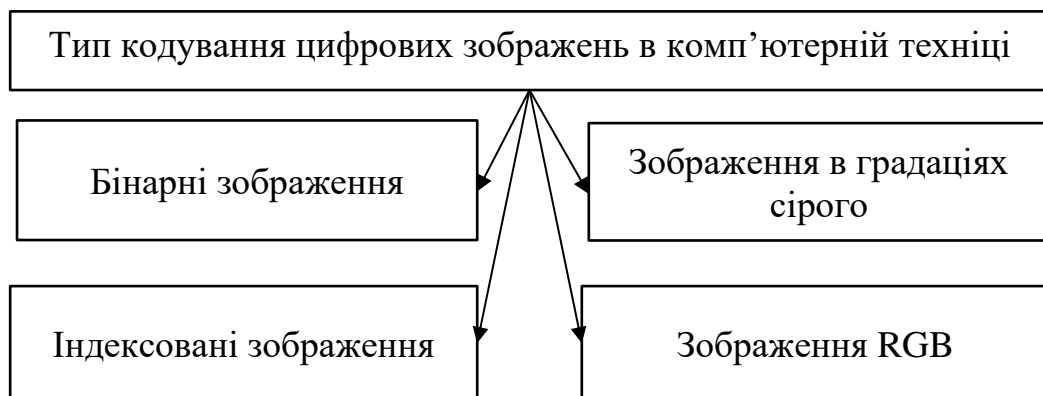


Рисунок 1.1 – Класифікація цифрових зображень

Двійкові зображення: двійкові зображення містять лише два значення (вимкнено або ввімкнено). Значення вимкнення зазвичай дорівнює 0, а значення увімкнення зазвичай дорівнює 1. Цей тип зображення зазвичай використовується як множник для маскуванню областей в іншому зображенні.

Зображення в градаціях сірого: зображення в градаціях сірого представляють інтенсивність. Пікселі варіюються від найменш інтенсивних (чорних) до найінтенсивніших (білих). Значення пікселів зазвичай коливаються від 0 до 255 або масштабуються до цього діапазону під час відображення.

Індексовані зображення: замість інтенсивності значення пікселя в індексованому зображенні пов'язане зі значенням кольору в таблиці пошуку кольорів. Оскільки індексовані зображення посилаються на таблиці кольорів, що містять до 256 кольорів, значення даних цих зображень зазвичай масштабуються в діапазоні від 0 до 255.

Зображення RGB: у тривимірному масиві зображення RGB два виміри визначають розташування пікселя в зображенні. Інший вимір визначає колір кожного пікселя. Розмір кольору завжди має розмір 3 і складається з червоних, зелених і синіх кольорових смуг (каналів) зображення.

Загальні моделі кольорів, які використовуються для представлення зображення, включають RGB (червоний, зелений, синій), HSV (відтінок, насиченість, значення/яскравість) і CMYK (блакитний, пурпуровий, жовтий, чорний). Різні мультимедійні додатки можуть віддати перевагу одній моделі над іншою на основі своїх окремих вимог. Наприклад, модель RGB зазвичай використовується в комп'ютерній графіці, тоді як модель CMYK переважно використовується для цілей друку.

- RGB: червоний (0-255), зелений (0-255), синій (0-255) - використовується в екранах комп'ютерів.

- CMYK: блакитний (0-100%), пурпуровий (0-100%), жовтий (0-100%), чорний (0-100%) - використовується для професійного друку.

- HSV: відтінок (0-360), насиченість (0-100%), значення/яскравість (0-100%) - використовується в телевізійному мовленні.

Розуміння представлення зображень є ключовим для ефективної обробки даних, особливо під час обробки зображень у прикладних областях, таких як комп'ютерне бачення, машинне навчання та графіка. Інформацією, отриманою з

оцифрованого зображення, можна маніпулювати різними способами, щоб виявити певні особливості або досягти певних ефектів.

Наприклад, у комп'ютерному зорі подання зображення відіграє життєво важливу роль у таких завданнях, як розпізнавання зображень, виявлення об'єктів, сегментація зображень тощо.

Зображення зберігаються в комп'ютерній системі як великий файл, що містить організовані цифрові дані. Файл цифрового зображення містить інформацію про властивості зображення, такі як його розміри та роздільна здатність, а також індивідуальні значення кольору кожного пікселя.

Зображення зберігаються в пам'яті комп'ютера в двійковій формі, як комбінація нулів і одиниць. Кожен піксель із зображення перетворюється на двійкове значення, і комп'ютер зчитує й обробляє ці цифрові дані.

Двійкові дані зображення використовують бітову глибину 1, тобто кожен піксель може приймати лише два значення: 0 (представлено білим) і 1 (представлено чорним).

Бітова глибина стосується інформації про колір, що зберігається в зображенні. Більшебітова глибина означає, що кожен колір може представляти ширший діапазон значень і, отже, більш нюансований колір.

Наприклад, у двійковому зображенні, якщо значення пікселя дорівнює 0, воно відобразатиметься як білий піксель, а якщо значення дорівнює 1, воно відобразатиметься як чорний піксель.

У обробці зображень двійкове представлення зображень особливо корисно для завдань, які потребують високого контрасту та меншої складності. Такі завдання, як виявлення країв, сегментація зображення та розпізнавання об'єктів, можуть отримати користь від представлення двійкового зображення.

Наприклад, в OCR (оптичне розпізнавання символів) представлення двійкового зображення може допомогти легко відрізнити букви від фону.

Підводячи підсумок, можна сказати, що представлення зображень – це обширне та фундаментальне поняття в інформатиці, яке дозволяє інтерпретувати зображення та маніпулювати ними в обчислювальних системах. Спосіб

представлення зображення може вплинути на результати завдань у таких сферах, як комп'ютерне бачення, машинне навчання та цифрова графіка.

У світі цифрових зображень існують два відомі типи зображень: растрові та векторні.

Представлення растрового зображення, яке також називають растровим зображенням, походить із сітки окремих пікселів, які разом візуалізують зображення. Кожен піксель у цій сітці має певне значення кольору, яке, якщо дивитися разом із відстані, формує повне зображення. Розуміння основ цієї концепції має важливе значення для вивчення таких галузей, як OpenGL, DirectX і загального 3D-програмування.

Растрові зображення характеризуються роздільною здатністю та глибиною кольору. Роздільна здатність зображення визначається кількістю пікселів, що містяться у фіксованій області, і зазвичай вимірюється в пікселях на дюйм (PPI). Що вища роздільна здатність растрового зображення, то більше деталей воно містить, що забезпечує високоякісний візуальний результат. Глибина кольору, з іншого боку, визначає максимальну кількість кольорів, які можна відобразити на зображенні. Вимірюється в бітах, де біт містить одне з двох значень: 0 або 1.

Основні характеристики растрових зображень включають:

Високий рівень деталізації: растрові зображення можуть відтворювати високий рівень деталізації, що ідеально підходить для складних дизайнів і фотографій.

Розмір файлу: Розміри растрових файлів зазвичай великі, особливо для зображень із високою роздільною здатністю. Розмір файлу можна зменшити за допомогою стиснення техніки.

Масштабованість: Растрові зображення втрачають якість при зміні розміру, явище, відоме як «піксельція».

Представлення растрового зображення широко використовується в цифровій фотографії, веб-графіці, піктограмах і детальних ілюстраціях, враховуючи його здатність відтворювати високий рівень деталей. Однак властиві обмеження растрового представлення, такі як втрата якості під час зміни розміру

та великі розміри файлів, роблять його менш ідеальним для певних програм. Наприклад, створення масштабованої векторної графіки, великомасштабних відбитків або графіки з прозорим фоном може бути складним завданням із растровим зображенням.

Формат растрового файлу визначає, як зображення зберігається у файлі. У ньому описано, як зберігаються кольори стиснення використовувані рівні та інші фактори, які впливають на загальну якість зображення та розмір файлу.

Давайте розглянемо деякі з цих прикладів форматів файлів:

JPEG широко використовується у фотографії завдяки його ефективним можливостям стиснення, які значно зменшують розмір файлу, зберігаючи прийнятну якість зображення. Це використовує стиснення з втратами і підтримує мільйони кольорів, що робить його ідеальним для фотографій.

Цікаво відзначити, що у файлах JPG ступінь втрат стиснення можна регулювати, дозволяючи вибрати компроміс між розміром пам'яті та якістю зображення. Однак недоліком формату JPEG є те, що при кожному збереженні зображення якість знижується. Коли зберігаєте файл у форматі JPEG, вам часто буде запропоновано вибрати шкалу якості. Якщо ви виберете нижчу якість, розмір файлу зменшиться, але ви втратите більше даних у процесі, через що зображення з часом може виглядати піксельним або розмитим.

На відміну від JPEG, PNG використовує без втрат стиснення даних і пропонує кращу підтримку прозорості. Це робить PNG придатним для зображень, які потребують прозорості, як-от логотипи, піктограми або графіка для веб-сайтів. З іншого боку, файли PNG зазвичай більші за файли JPEG і не підтримують анімацію, як файли GIF.

GIF підтримує до 256 кольорів, що робить його менш придатним для фотографій, але ідеальним для простих одноколірних зображень і анімації. Використання GIF-файлів стиснення без втратале через обмеження кольорів фотографії, збережені у форматі GIF, часто виглядають плоскими.

BMP – це старий і простий растровий формат, розроблений Microsoft. Він не стискається і тому не часто використовується для обміну зображеннями через

Інтернет через більший розмір файлу. Однак він простий і легкий у реалізації, що означає, що він часто використовується внутрішньо графічними програмами.

Векторні зображення принципово відрізняються від растрових тим, що вони не складаються з сітки пікселів. Замість цього векторні зображення створюються за допомогою математичних формул, що визначають шляхи та форми в зображенні. Масштабування та модифікація цих зображень обробляються по-різному, пропонуючи низку переваг, яких немає в растровому представленні.

В основі представлення векторних зображень лежать математичні формули, які створюють фігури через точки, лінії (також звані контурами) і криві. Кожна точка має певне положення на осях x і y у робочій площині та визначає напрямок шляху. Шлях може бути відкритим (наприклад, лінія) або замкнутим (наприклад, багатокутник), а криві створюються за допомогою математичних функцій, відомих як криві Безьє.

Ключові особливості векторних зображень:

- Масштабованість: векторні зображення можна збільшувати або зменшувати без втрати якості, оскільки вони не залежать від роздільної здатності. Це не стосується растрових зображень.

- Менший розмір файлу: розміри файлів векторних зображень часто менші, ніж їх растрові аналоги, особливо для великої складної графіки.

- Гнучкість: елементами векторного зображення можна маніпулювати індивідуально, що забезпечує більшу гнучкість дизайну.

Представлення векторних зображень є корисним для створення логотипів, типографіки та іншої графіки, яка вимагає точних ліній і контурів. Вони також підходять для анімації та інтерактивної графіки, оскільки окремими елементами можна керувати незалежно. Недоліком є те, що векторні зображення не здатні створювати такі детальні градієнти кольорів, як у растрових зображеннях, і тому зазвичай не використовуються для відтворення фотореалістичних зображень, складного затінення чи складного поєднання кольорів.

Кодування зображень є ключовим процесом в інформатиці, особливо в мультимедійних системах і комп'ютерних візуальних зображеннях. Він передбачає перетворення необроблених піксельних даних у формат, який можна зберігати та, за потреби, отримати та відобразити комп'ютерною системою.

Кодування зображення, інакше відоме як зображеннястиснення, це технічний процес перетворення файлу зображення в серію байтів і кодів. Це перетворення в першу чергу служить для зменшення розміру файлу зображення для зберігання або передачі, забезпечуючи оптимальне використання ресурсів комп'ютера.

Загальна мета кодування зображення полягає в тому, щоб більш ефективно представити те саме зображення, що зберігається як необроблені двійкові дані, і таким чином зменшити обсяг необхідних даних. Ці закодовані файли зображень займають менше місця в пам'яті та можуть набагато швидше передаватись через мережі, підвищуючи загальну продуктивність системи.

Як правило, алгоритм кодування бере растрове зображення та перетворює його в стиснутий формат. Цей перехід відбувається за допомогою двох основних типів стиснення зображення: без втрат і стиснення з втратами.

Стиснення без втрат: за допомогою цього методу стиснений файл зображення можна ідеально відновити до початкової форми. Він досягає скорочення даних без будь-яких втрат інформації, тому отримав назву "без втрат". Формати, які використовують стиснення без втрат, включають PNG, BMP і TIFF.

Стиснення з втратами: цей метод забезпечує ще більший рівень скорочення даних. Це усуває зайву або непотрібну інформацію, що призводить до втрати деяких даних у процесі. Цей прийом корисний, коли втрата даних не сильно впливає на якість зображення. Використання форматів зображень стиснення з втратами включають JPEG, GIF і певні типи TIFF.

Кодування зображень надзвичайно важливе для ефективного зберігання та транспортування даних, і воно приносить переваги багатьом галузям,

включаючи, але не обмежуючись цим: цифрову фотографію, графічний дизайн, Веб дизайн, комп'ютерне бачення та машинне навчання.

У цифровій фотографії та графічному дизайні кодування може зменшити розмір файлу зображення без значного погіршення якості зображення. Цей процес дозволяє ефективніше використовувати пам'ять і пришвидшити час завантаження/завантаження.

Веб-дизайнери регулярно використовують кодування зображень, щоб зменшити час завантаження веб-сторінок. Веб-сторінка зі зображеннями зменшеного розміру може завантажуватися значно швидше, тим самим покращуючи взаємодію з користувачем і потенційно підвищуючи рейтинг веб-сайту в результатах пошуку.

У комп'ютерному зорі та машинному навчанні закодовані зображення можуть сприяти більш ефективній обробці. Здатність зменшити обсяг даних, що обробляються, без значних втрат важливої інформації дозволяє цим системам працювати безперебійно та ефективно, підвищуючи загальну продуктивність.

Одним із методів є кодування Хаффмана, який є технікою без втрат, яка замінює значення, що найчастіше зустрічаються, коротшими кодами. Цей метод поширений у зображеннях PNG.

Стиснення LZ77 (Lempel-Ziv) – ще один метод кодування без втрат, який усуває повторне повторення даних. Ковзне вікно використовується для пошуку та заміни цих повторюваних входжень посиланнями.

DCT (дискретне косинусне перетворення) – це техніка стиснення з втратами, яка використовується в основному для зображень JPEG. Він розділяє зображення на частини з різними частотами, а менш важливі частоти відкидаються.

Обрана техніка відповідатиме вимогам вашої програми. Наприклад, якщо необхідно відновити зображення до точної вихідної форми, підійде метод без втрат, наприклад кодування Хаффмана або LZ77. І навпаки, якщо незначна втрата даних допустима для зменшення розміру, кращим вибором може бути метод стиснення з втратами, наприклад DCT.

Безперечно, кодування та декодування зображень є областю інформатики, яка робить значний внесок у повсякденні технологічні операції. Це підвищує ефективність, заощаджує пропускну здатність і суттєво покращує продуктивність системи, що робить його незамінним у цифровому світі.

1.2 Опис та класифікація зображень

Класифікація зображень – це процес класифікації та позначення пікселів або груп пікселів на цифрових зображеннях на основі їхніх спектральних значень. Цей процес дозволяє нам аналізувати та інтерпретувати складні шаблони та інформацію, що міститься на зображеннях, перетворюючи необроблені дані на більш значущу інформацію, яку можна використовувати для різних програм. Класифікація зображень допомагає отримувати цінну інформацію з цифрових даних. Ця інформація може бути використана для різних цілей, таких як землекористування та картографування ґрунтового покриву, міське планування, моніторинг сільського господарства, управління природними ресурсами та екологічні дослідження тощо. Розділяючи пікселі на різні класи, класифікація зображень спрощує дані та полегшує користувачам аналіз і розуміння просторових моделей, тенденцій і зв'язків.

У цифровій техніці використовуються два основних типи методів класифікації зображень (рисунок 1.2).



Рисунок 1.2 – Основні підходи до класифікації цифрових зображень

Контрольована класифікація зображень. Цей метод покладається на знання та досвід користувача, щоб забезпечити набір навчальних зразків для різних цікавих класів. Алгоритми класифікатора вивчають ці навчальні зразки та застосовують ці знання, щоб класифікувати все зображення за потрібними категоріями. Цей метод зазвичай забезпечує більшу точність, оскільки він включає знання домену користувача, але вимагає більше часу та зусиль для збору навчальних зразків.

Класифікація зображень без нагляду. У цьому методі процес класифікації здійснюється без попередньої інформації чи навчальних зразків. Натомість алгоритми класифікатора групують пікселі в різні кластери на основі їхніх спектральних значень і природної подібності. Потім користувач призначає значущі мітки цим кластерам на основі свого розуміння досліджуваної області. Неконтрольована класифікація займає менше часу, але може не досягти такого ж рівня точності, як контрольована, оскільки вона не включає жодних експертних знань.

Контрольована класифікація зображень – це метод, за якого користувач надає набір позначених навчальних зразків для кожного цікавого класу. Алгоритми класифікатора використовують ці навчальні зразки для вивчення характеристик кожного класу, а потім застосовують ці знання для класифікації всього зображення за вказаними категоріями. Цей метод покладається на досвід користувача та розуміння досліджуваної області та зазвичай забезпечує більшу точність порівняно з класифікацією без нагляду.

Етапи контрольованої класифікації зображень:

1) Попередня обробка даних: цей крок передбачає виправлення будь-яких спотворень або помилок у даних зображення, наприклад атмосферних, радіометричних або геометричних поправок.

2) Відбір навчальних зразків: користувач вибирає репрезентативні зразки для кожного цікавого класу на основі своїх знань про досліджувану область і даних зображення.

3) Вилучення функцій: цей крок передбачає вилучення релевантних характеристик, таких як спектральна, текстурна або контекстна інформація, із даних зображення для покращення ефективності класифікації.

4) Навчання класифікатора: вибраний алгоритм класифікатора вивчає характеристики кожного класу з наданих навчальних зразків.

5) Класифікація: навчений алгоритм класифікатора застосовується до всього зображення, класифікуючи кожен піксель до одного з указаних класів.

6) Оцінка точності та уточнення: результати класифікації оцінюються на точність, часто з використанням наземних правдивих даних або експертних знань. Якщо необхідно, процес класифікації може бути ітеративно вдосконалений шляхом коригування навчальних вибірок або параметрів класифікатора.

Серед відомих алгоритмів керованої класифікації є:

Класифікатор максимальної правдоподібності: статистичний алгоритм, який припускає, що кожен клас відповідає розподілу Гауса, і призначає пікселі до класу з найвищою правдоподібністю на основі їхніх спектральних значень.

Машини опорних векторів: алгоритм машинного навчання, який знаходить оптимальну гіперплощину для розділення різних класів шляхом максимізації запасу між ними.

Дерева рішень: ієрархічний підхід, який розбиває дані на підмножини на основі певних правил або умов у кожному вузлі дерева, доки піксель не буде призначено класу.

Серед переваг даного підходу можна відмітити наступні:

- Загалом вища точність порівняно з неконтрольованою класифікацією, оскільки вона включає знання домену користувача.

- Дозволяє більше контролювати процес класифікації, оскільки користувачі можуть вказувати цікаві класи та коригувати навчальні зразки.

Недоліки:

- Вимагає більше часу та зусиль для збору репрезентативних навчальних зразків.

- Може бути схильний до переобладнання, якщо навчальні зразки не відображають належним чином мінливість у кожному класі.

Класифікація зображень без нагляду. Класифікація зображень без нагляду – це метод, при якому процес класифікації здійснюється без будь-якої попередньої інформації чи навчальних зразків. Натомість алгоритми кластеризації групують пікселі в різні кластери на основі їхніх спектральних значень і природної подібності.

Потім користувач призначає значущі мітки цим кластерам на основі свого розуміння досліджуваної області. Цей метод займає менше часу, але може не досягти такого ж рівня точності, як контрольована класифікація, оскільки він не включає жодних експертних знань.

Етапи класифікації зображень без нагляду:

1) Попередня обробка даних: як і під час контрольованої класифікації, цей крок передбачає виправлення будь-яких спотворень або помилок у даних зображення, наприклад атмосферних, радіометричних або геометричних поправок.

2) Вилучення функцій: відповідні функції, такі як спектральна, текстурна або контекстна інформація, витягуються з даних зображення для покращення процесу кластеризації.

3) Кластерний аналіз: вибраний алгоритм кластеризації застосовується до попередньо оброблених даних зображення, групуючи пікселі у вказану кількість кластерів на основі їхніх спектральних значень і подібності.

4) Позначення кластерів: користувач призначає значущі мітки отриманим кластерам на основі свого розуміння області дослідження та даних зображення.

5) Оцінка точності та уточнення: результати класифікації оцінюються на точність, часто з використанням наземних правдивих даних або експертних знань. Якщо необхідно, процес класифікації може бути ітераційно вдосконалений шляхом коригування кількості кластерів або параметрів алгоритму кластеризації.

Загальні алгоритми неконтрольованої класифікації

K-means: популярний алгоритм кластеризації, який спрямований на мінімізацію суми квадратів усередині кластера шляхом повторного оновлення центроїдів кластера та призначення пікселів найближчому центроїду.

ISODATA (Техніка ітераційного самоорганізуючого аналізу даних): ітеративний метод кластеризації, який дозволяє об'єднувати та розбивати кластери на основі параметрів, визначених користувачем, що робить його більш гнучким, ніж K-середні.

Ієрархічна кластеризація: метод кластеризації, який створює деревоподібну структуру вкладених кластерів на основі метрики подібності, яку можна розрізати на певному рівні, щоб отримати бажану кількість кластерів.

Переваги неконтрольованої класифікації:

- Менш трудомісткий, оскільки не вимагає збору навчальних зразків.
- Може виявляти невідомі або неочікувані шаблони в даних зображення, оскільки не покладається на попередні знання.

Недоліки:

- Загалом нижча точність порівняно з контрольованою класифікацією, оскільки вона не включає жодних експертних знань.
- Отримані кластери можуть не мати чітких або значущих меж, через що користувачам буде важко призначати точні мітки.
- Вимагає від користувача визначити відповідну кількість кластерів, що може бути складним завданням і може вплинути на результати класифікації.

У деяких випадках може бути корисним комбінувати як контрольовані, так і неконтрольовані методи класифікації. Цей гібридний підхід може використовувати сильні сторони кожного методу, одночасно пом'якшуючи їхні слабкі сторони.

Наприклад, неконтрольована класифікація може бути використана спочатку для вивчення даних і виявлення шаблонів або потенційних класів. Отримані кластери можуть служити відправною точкою для збору репрезентативних навчальних вибірок, які можна використовувати в контрольованій класифікації для отримання більш точних і значимих

результатів. Цей підхід може заощадити час і зусилля, забезпечуючи при цьому високий рівень точності та сумісності.

У контексті класифікації зображень переобладнання означає ситуацію, коли алгоритм класифікатора вчиться надто точно підлаштовувати навчальні дані, фіксуючи шум і конкретні деталі навчальних зразків замість узагальнення базових шаблонів. У результаті класифікатор може працювати дуже добре на навчальних вибірках, але погано на нових, невидимих даних.

Переобладнання може статися під час контрольованої класифікації зображень, коли навчальні зразки не відображають належним чином мінливість у кожному класі або коли модель класифікатора є надто складною. У цих випадках класифікатор може навчитися розпізнавати конкретні характеристики навчальних зразків, а не загальні ознаки класу, що призводить до зниження точності класифікації при застосуванні до всього зображення або інших невидимих даних.

Щоб пом'якшити переобладнання, важливо переконатися, що навчальні вибірки є репрезентативними для різних класів, і розглянути можливість використання простіших моделей класифікаторів або методів регуляризації, які заохочують модель зосереджуватися на найбільш відповідних характеристиках.

Контрольована класифікація зображень спирається на досвід і знання користувача, щоб забезпечити набір навчальних зразків для різних класів інтересу. Алгоритми класифікатора навчаються на основі цих зразків і класифікують все зображення на основі цих знань. На відміну від цього, неконтрольована класифікація зображень не використовує попередньої інформації чи навчальних зразків. Натомість алгоритми кластеризації групують пікселі в різні кластери на основі їхніх спектральних значень і природної подібності. Вибір правильного методу класифікації зображень для конкретного проекту має вирішальне значення, оскільки це може суттєво вплинути на точність та інтерпретацію результатів.

1.3 Програмні засоби обробки та опису цифрових зображень

Розробки в напрямку створення програмних систем обробки та аналізу цифрових зображень на сьогоднішній день перебувають на піку свого розвитку. Все більше компаній інтегрують засоби комп'ютерного зору у свої програмно-апаратні комплекси. Тому необхідно виділити основні групи критеріїв для відбору максимально ефективних структурних модулів та функціональних рішень для систем цифрової обробки даних.

Існує багато варіантів програмного забезпечення для пакетної обробки залежно від ваших потреб і вподобань. Одними з найпопулярніших є Adobe Photoshop, Lightroom, Bridge і Camera Raw, які пропонують низку функцій і функцій для редагування та впорядкування ваших зображень. Інші альтернативи включають GIMP, IrfanView, FastStone Image Viewer і XnView, які є безкоштовними або недорогими програмами, які також підтримують пакетну обробку. Ви також можете використовувати такі онлайн-інструменти, як Pixlr, PicMonkey або Canva, які є веб-додатками, які дозволяють завантажувати та редагувати ваші зображення пакетами. Adobe Photoshop – стара, але дорога версія цих програм. GIMP, який дозволяє використовувати та обробляти всі шари зображення та має нульову вартість використання. Онлайн-програми легко використовувати на будь-якому комп'ютері, їх легше налаштувати як пакетні операції, але це може мати наслідки для втрати точності зображення.

Однією з переваг пакетної обробки є те, що можете легко перейменувати та сортувати файли відповідно до власних критеріїв. Наприклад, можете використовувати узгоджену конвенцію про найменування, яка включає дату, місце, подію або тему ваших зображень. Також можете додавати префікси, суфікси або цифри до імен файлів, щоб зробити їх більш описовими та доступними для пошуку. Крім того, ви можете сортувати файли в папки або вкладені папки на основі їхніх категорій, тегів, рейтингів або ключових слів. Це допоможе зберегти зображення впорядкованими та доступними.

Іншим поширеним завданням, яке можна виконувати за допомогою пакетної обробки, є зміна розміру та обрізання зображень. Це особливо корисно, якщо потрібно підготувати зображення для різних цілей, наприклад для друку, публікації в Інтернеті або спільного використання в соціальних мережах. Можете використовувати пакетну обробку, щоб налаштувати розміри, роздільну здатність, співвідношення сторін або орієнтацію ваших зображень одним рухом. Також можете обрізати зображення, щоб видалити непотрібні елементи, покращити композицію або зосередитися на головному об'єкті.

Редагування та покращення зображень. Пакетна обробка також може допомогти покращити якість і зовнішній вигляд ваших зображень шляхом застосування різноманітних змін і покращень. Наприклад, можете використовувати пакетну обробку, щоб налаштувати експозицію, контраст, колір, різкість, шум або спотворення ваших зображень. Також можете застосувати фільтри, ефекти чи стилі до своїх зображень, щоб створити різні настрої, тони чи образи. Деякі програмні засоби також дозволяють використовувати дії, сценарії або макроси для автоматизації складних або повторюваних завдань редагування. Однак ви повинні бути обережними, щоб не перестаратися або не втратити оригінальність ваших зображень під час їх пакетного редагування.

Метадані. Пакетна обробка може захистити та ідентифікувати зображення шляхом додавання водяних знаків і метаданих. Водяний знак – це видимий або невидимий знак, який показує право власності, авторські права або джерело зображення. Метадані – це набір інформації, яка описує вміст, контекст або характеристики зображення. Можете використовувати пакетну обробку, щоб додати до своїх зображень водяні знаки тексту, логотипу або зображення в різних положеннях, розмірах або непрозорості. Також можете використовувати пакетну обробку для додавання або редагування метаданих, таких як назва, опис, автор, дата, місце або ключові слова до ваших зображень.

Перетворення та стиснення зображень. Перетворення та стиснення зображень важливі, якщо потрібно змінити формат або зменшити розмір

зображень для різних платформ, пристроїв або цілей. Можете використовувати пакетну обробку, щоб конвертувати зображення з одного формату в інший, наприклад JPEG, PNG, TIFF або RAW. Також можете використовувати пакетну обробку для стиснення зображень для оптимізації їх якості та продуктивності. Деякі програмні інструменти також дозволяють оптимізувати зображення для використання в Інтернеті чи на мобільних пристроях, регулюючи такі параметри, як якість, колірний простір або роздільна здатність.

Серед представлених на сьогоднішній день програмних продуктів можна виділити п'ять систем які найчастіше використовуються для обробки зображень:

Adobe Photoshop є одним з найвідоміших і найпотужніших інструментів для обробки та редагування цифрових зображень. Він надає широкий спектр функцій, які додатково включають корекцію кольорів, ретуш, атакве та повне редагування тексту та створення та корекція складних композицій. Photoshop також підтримує роботу з різними форматами файлів та надає широкі можливості для маніпуляції пікселями.

GIMP (GNU Image Manipulation Program) – це надзвичайно популярний вихідно-кодовий редактор зображень. Окремою перевагою даного програмного засобу є те, що для нього існує безкоштовний варіант використання. Він забезпечує безліч інструментів для корекції, монтажу та редагування оцифрованих зображень. GIMP підтримує механіку шарів, пензлі, фільтри та різноманітні розширення, що в підсумку робить його функціональним аналогом різних платних рішень.

Lightroom, спроектований та реалізований Adobe, спеціалізується на обробці фотографій. Програмна розробка надає інструменти для налаштування експозиції, колірного балансу, покращення деталей та організації фотоколекцій. Lightroom також забезпечує управління, налаштування та додавання різних метаданих та ключовими словами.

Capture One – професійне рішення для обробки фотографій, орієнтоване на фотографів. Він надає точні інструменти для корекції кольору, корекції тонів і

ретуші, а також підтримує роботу з RAW-файлами. Capture One часто використовується у сфері фотостудій та ретушування модних знімків.

PaintShop Pro від Corel – це графічний редактор, який пропонує безліч інструментів для обробки фотографій та створення графічних дизайнів. Включаючи функції для редагування, ретушування, накладання ефектів та керування кольором, PaintShop Pro також підтримує роботу з різними шарами та форматами файлів.

Ці програми надають широкий спектр можливостей обробки цифрових зображень і адаптовані до різних потреб користувачів – від професійних фотографів до тих, хто шукає доступні і прості у використанні інструменти. Окрім того, проаналізовані програмні системи містять основні структурні елементи які притаманні усім сучасним програмним комплексам обробки цифрових зображень, а саме в них реалізовані різні підходи для отримання зображень, пакетні та одинарні цикли опрацювання цифрових зображень, можливість редагувати зображення за шаблонами та вносити корективи в значення параметрів алгоритмів обробки та опису цифрових зображень. Окрім того дані програмні комплекси дозволяють користувачам самостійно розробляти та інтегрувати різні плагіни для збільшення функціоналу.

Для проведення порівняння існуючих програмних засобів було обрано ряд узагальнюючих критеріїв та проведено їх оцінювання за п'ятибальною шкалою. Результати порівняльного аналізу було згруповано та наведено в таблиці 1.1.

Таблиця 1.1 – Програмні засоби для обробки цифрових зображень

Назва програмного засобу	Безкоштовна версія	Зручність користування	Оцінка отриманого програмного коду	Підтримка від розробників	Загальна оцінка
Adobe Photoshop	-	5	5	5	5
GIMP	+	4	4	4	4
Lightroom	+	4	4	4	4
Capture One	+	4	5	4	4,6
Corel Paint Shop Pro	-	4	4	4	4

1.4 Постановка задач дослідження

На основі проведеного аналізу цифрових зображень та технологій їх обробки можна зробити наступні висновки. Цифрові зображення відіграють значну роль в сучасних технологічних процесах, та дозволяють автоматизувати все більше складних та трудомістких процесів. Основними характеристиками цифрових зображень є спосіб його кодування та роздільна здатність. Характеристика кольору не у всіх задачах відіграє вирішальне значення, інколи достатньо і бінарного зображення. Опис зображень є важливим етапом в процесі обробки та класифікації зображення. Для технологій опису зображень важливими є не тільки точність, але і кількість ознак на основі яких він здійснюється. Надмірне використання малоінформативних ознак призведе до збільшення часу опрацювання зображення. На сьогоднішній день використання технологій обробки зображень все частіше використовується в різних програмно-апаратних системах. Це можна пояснити стрімким розвитком комп'ютерної техніки та розробки швидких і точних алгоритмів розпізнавання та класифікації цифрових зображень. При цьому, використання технологій комп'ютерного зору та штучного інтелекту дозволяє інтегрувати новий функціонал в звичні програмні комплекси.

Для досягнення поставленої мети необхідно розв'язати наступні задачі.

- провести дослідження цифрових зображень та виділити їх основні характеристики;
- провести аналіз технологій опису цифрових зображень;
- провести дослідження сучасних програм обробки цифрових зображень;
- проаналізувати алгоритми виділення контурів об'єктів;
- розробити алгоритм опису сцен на цифрових зображеннях на основі виділення контурів об'єктів;
- реалізувати програмний додаток обробки та опису цифрових зображень.

1.5 Висновки до розділу

Проведено дослідження цифрових зображень на основі технологій їх отримання та кодування, що дозволило провести їх класифікації та виділити основні характерні ознаки.

Проведено аналіз технологій опису цифрових зображень на основі аналізу внутрішніх об'єктів, що дозволило виділити основні характерні ознаки з максимальною інформативною складовою.

Проведено дослідження огляд програмних додатків аналізу та опису цифрових зображень на основі аналізу їх внутрішньої архітектури, що дозволило визначити основні структурні елементи які реалізуються в сучасних програмних системах обробки цифрових даних.

2 МЕТОДИ ТА АЛГОРИТМИ ОБРОБКИ КОЛЬОРОВИХ ЦИФРОВИХ ЗОБРАЖЕНЬ

2.1 Методи та алгоритми обробки опису цифрових зображень

Зображення визначають світ, кожне зображення має свою власну історію, воно містить багато важливої інформації, яка може бути корисною в багатьох відношеннях. Цю інформацію можна отримати за допомогою техніки, відомої як обробка зображень. Це основна частина комп'ютерного зору, яка відіграє вирішальну роль у багатьох прикладах реального світу, таких як робототехніка, безпілотні автомобілі та виявлення об'єктів. Обробка зображень дозволяє перетворювати та маніпулювати тисячами зображень одночасно та отримувати з них корисну інформацію. Вона має широкий спектр застосування майже в усіх сферах. Обробка зображень означає обробку зображення, і це може включати багато різних методів, поки не досягнуто потрібної мети. Кінцевий результат може бути у формі зображення або відповідної функції цього зображення. Це можна використовувати для подальшого аналізу та прийняття рішень.

Обробка зображень – це процес покращення й вилучення корисної інформації із зображень. Зображення розглядаються як двовимірні сигнали, а вхідними даними для цього процесу є фото чи відео. Вхідними даними є зображення, а виходом може бути покращене зображення або пов'язані з ним характеристики/функції. Існує багато способів обробки зображення, але всі вони мають схожу схему. Спочатку виділяються червоний, зелений і синій інтенсивності зображення. Новий піксель створюється з цих інтенсивностей і вставляється в нове порожнє зображення в тому самому місці, що й оригінал. Крім того, пікселі градації сірого створюються шляхом усереднення інтенсивності всіх пікселів. Після цього їх можна перетворити на чорні або білі за допомогою порогу.

Важливо знати, що алгоритми обробки зображень відіграють найважливішу роль у обробці цифрових зображень. Розробники використовують

і впроваджують кілька алгоритмів обробки зображень для вирішення різних завдань, включаючи виявлення цифрових зображень, аналіз зображень, реконструкцію зображення, відновлення зображення, покращення зображення, стиснення даних зображення, оцінку спектрального зображення та оцінку зображення. Іноді алгоритми можуть бути прямо з книги або більш налаштованою об'єднаною версією кількох функцій алгоритму.

У випадку повного захоплення зображення алгоритми обробки зображень зазвичай класифікуються :

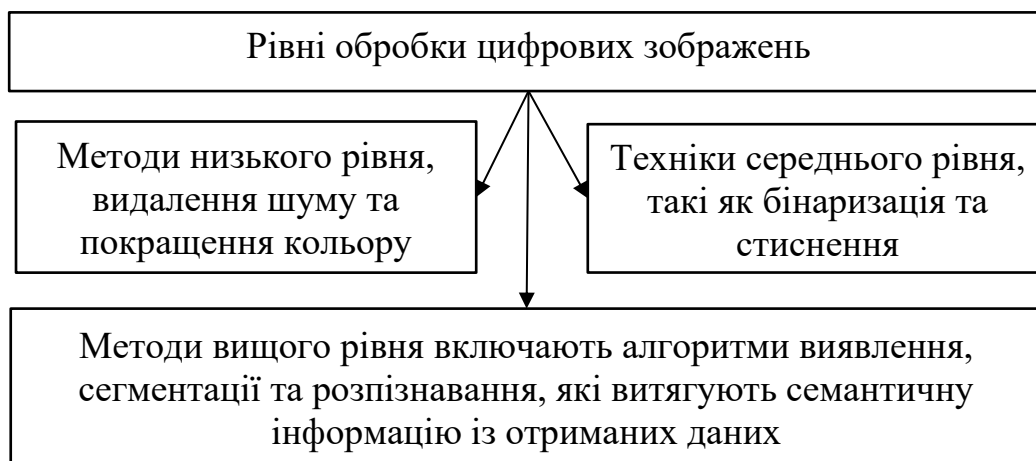


Рисунок 2.1 –Класифікація алгоритмів обробки цифрових зображень

Існують різні типи алгоритмів обробки зображень. Методи, які використовуються для обробки зображень, це створення та аналіз зображень. Основна ідея цього полягає в перетворенні зображення з його оригінальної форми на цифрове зображення з однорідним макетом. Ось деякі з традиційних алгоритмів обробки зображень:

- Алгоритми підвищення контрастності:
 - Алгоритм вирівнювання гістограми: використання гістограми для покращення контрастності зображення;
 - Адаптивний алгоритм вирівнювання гістограми: це вирівнювання гістограми, яке адаптується до локальних змін контрасту;

- Алгоритм маркування зв'язаних компонентів: йдеться про пошук і маркування непересічних областей.
- Алгоритми виявлення ознак складається:
 - Алгоритм Марра–Хілдрета: це алгоритм раннього виявлення країв;
 - Алгоритм детектора країв Canny: детектор країв Canny використовується для виявлення широкого діапазону країв на зображеннях.
- Алгоритми сегментації:
 - Алгоритм випадкового проходження;
 - Алгоритм нарощування регіону;
 - Алгоритм перетворення вододілу.

Наведені алгоритми ілюструються велику кількість алгоритмів обробки цифрових зображень які використовуються в різних галузях. На рисунку 2.2 наведено класифікацію алгоритмів на основі підходів опрацювання цифрових зображень.

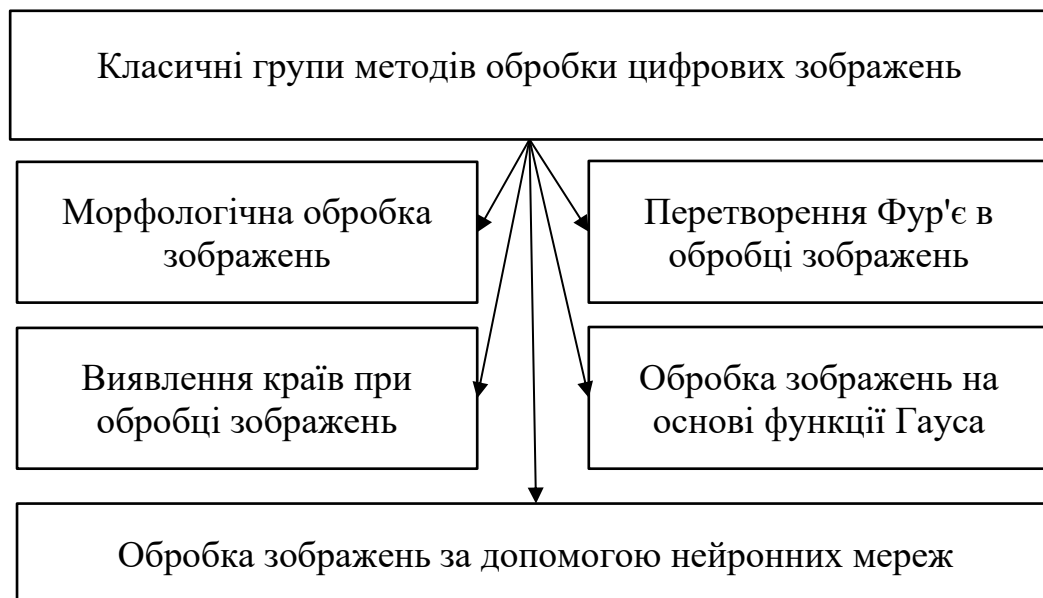


Рисунок 2.2 – Класичні групи методів обробки цифрових зображень

Слід зазначити, що окрім алгоритмів, згаданих вище, галузі також створюють індивідуальні алгоритми для задоволення своїх потреб. Вони можуть бути з нуля або комбінацією різних алгоритмічних функцій. Можна з упевненістю сказати, що з розвитком комп'ютерних технологій алгоритми обробки зображень надали багатьом дослідникам і розробникам достатньо можливостей досліджувати, класифікувати, характеризувати та аналізувати різноманітні орди зображень.

Морфологічна обробка зображень намагається усунути недоліки бінарних зображень, оскільки бінарні області, створені простим пороговим визначенням, можуть бути спотворені шумом. Це також допомагає згладити зображення за допомогою операцій відкриття та закриття.

Морфологічні операції можна поширити на зображення в градаціях сірого. Він складається з нелінійних операцій, пов'язаних зі структурою ознак зображення. Це залежить від відповідного порядку пікселів, але від їхніх числових значень. Ця техніка аналізує зображення за допомогою невеликого шаблону, відомого як структурний елемент, який розміщується в різних можливих місцях на зображенні та порівнюється з відповідними сусідніми пікселями. Структурним елементом є маленька матриця зі значеннями 0 і 1.

Давайте розглянемо дві фундаментальні операції обробки морфологічного зображення, розширення та ерозію:

- операція розширення додає пікселі до меж об'єкта на зображенні
- операція ерозії видаляє пікселі з меж об'єкта.

Кількість пікселів, видалених або доданих до оригінального зображення, залежить від розміру структурного елемента.

Структуруючий елемент - це матриця, що складається лише з нулів і одиниць, яка може мати будь-яку довільну форму та розмір. Він розташовується в усіх можливих місцях зображення та порівнюється з відповідним оточенням пікселів.

Шаблон нуль-один визначає конфігурацію структурного елемента. Це залежить від форми об'єкта, який ми хочемо вибрати. Центр структурного

елемента ідентифікує піксель, що обробляється. Приклад операції ерозії наведено на рисунку 2.3.



Рисунок 2.3 – Приклад виконання операції ерозії: вхідне зображення (a) результат операції ерозія (b)

Розмиття за Гаусом, також відоме як згладжування за Гаусом, є результатом розмиття зображення за допомогою функції Гауса. Воно використовується для зменшення шуму зображення та зменшення деталей. Візуальний ефект цієї техніки розмивання схожий на перегляд зображення через напівпрозорий екран. Він іноді використовується в комп'ютерному зорі для покращення зображення в різних масштабах або як техніка збільшення даних у глибокому навчанні. Основна функція Гауса виглядає так:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}.$$

На практиці найкраще скористатися властивістю відокремлюваності розмиття за Гаусом, розділивши процес на два проходи. Під час першого проходу одновимірне ядро використовується для розмиття зображення лише в горизонтальному або вертикальному напрямку. Під час другого проходу те саме одновимірне ядро використовується для розмиття в решті напрямку. Отриманий ефект такий самий, як згортання з двовимірним ядром за один прохід. Давайте

розглянемо приклад, щоб зрозуміти, що фільтри Гауса роблять із зображенням. Якщо є фільтр, який зазвичай розподіляється, і коли його застосувати до зображення, результати виглядатимуть так (рисунок 2.4):



a)



b)

Рисунок 2.4 – Приклад використання фільтра Гауса для розмивання: вхідне зображення (a) результат розмивання зображення (b)

Перетворення Фур'є розбиває зображення на синусну та косинусну складові. Воно має кілька напрямів використання, таких як реконструкція зображень, стиснення зображень або фільтрація зображень. Оскільки зараз використовуються зображення, то маємо на увазі дискретне перетворення Фур'є.

Виявлення країв – це техніка обробки зображень для знаходження меж об'єктів на зображеннях. Вона працює шляхом виявлення розривів яскравості. Це може бути дуже корисним для отримання корисної інформації із зображення, оскільки більшість інформації про форму укладено по краях. Класичні методи визначення країв працюють шляхом виявлення розривів яскравості.

Виявлення країв може швидко реагувати, якщо на зображенні буде виявлено шум під час виявлення варіацій рівнів сірого. Краї визначаються як локальні максимуми градієнта.

Найпоширенішим алгоритмом виявлення краю є алгоритм виявлення краю Sobel . Оператор виявлення Sobel складається зі згорткових ядер 3×3 . Просте ядро G_x і повернуте на 90 градусів ядро G_y .

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Окремі вимірювання проводяться шляхом нанесення ядра окремо на зображення.

Вейвлет-обробка зображень. Попередньо проаналізувавши перетворення Фур'є, але воно обмежене лише частотою. Вейвлети враховують час і частоту. Це перетворення підходить для нестационарних сигналів. Відомо, що краї є однією з важливих частин зображення, під час застосування традиційних фільтрів було помічено, що шум видаляється, але зображення стає розмитим. Вейвлет-перетворення розроблено таким чином, що отримуємо хорошу частотну роздільну здатність для низькочастотних компонентів.

Обробка зображень за допомогою нейронних мереж. Нейронні мережі - це багаторівневі мережі, що складаються з нейронів або вузлів. Ці нейрони є основними процесорними одиницями нейронної мережі. Вони створені, щоб діяти як людський мозок. Вони беруть дані, навчаються розпізнавати закономірності в даних, а потім прогнозують результат. Основна нейронна мережа має три рівні:

- Вхідний шар.
- Прихований шар.
- Вихідний шар.

У Базовій нейронній мережі вхідні шари отримують вхідні дані, вихідний рівень передбачає вихідні дані, а приховані шари виконують більшість обчислень. Кількість прихованих шарів можна змінити відповідно до вимог. У нейронної мережі повинен бути принаймні один прихований шар. Основна робота нейронної мережі полягає в наступному:

Давайте обробимо зображення, кожен піксель подається на вхід до кожного нейрона першого шару, нейрони одного шару з'єднані з нейронами наступного шару через канали. Кожному з цих каналів присвоюється числове

значення, відоме як вага. Вхідні дані множаться на відповідні ваги, і ця зважена сума потім подається як вхідні дані до прихованих шарів. Вихід із прихованих шарів пропускається через функцію активації, яка визначатиме, чи буде активований певний нейрон чи ні. Активовані нейрони передають дані на наступні приховані шари. Таким чином дані поширюються через мережу, це відомо як пряме поширення. На вихідному рівні нейрон із найвищим значенням прогнозує вихід. Ці виходи є значеннями ймовірності. Прогнозований вихід порівнюється з фактичним, щоб отримати похибку. Ця інформація потім передається назад через мережу, процес відомий як зворотне поширення. На основі цієї інформації коригуються ваги. Цей цикл прямого та зворотного розповсюдження виконується кілька разів на кількох входах, доки мережа в більшості випадків правильно передбачить вихід.

На цьому процес навчання нейронної мережі закінчується. Час, витрачений на навчання нейронної мережі, у деяких випадках може бути високим.

Конволюційна нейронна мережа, коротко ConvNets, має три рівні:

Згортковий рівень (*CONV*): він є основним будівельним блоком CNN, він відповідає за виконання операції згортки. Елемент, який бере участь у виконанні операції згортки на цьому рівні, називається ядром /фільтром (матрицею). Ядро виконує горизонтальні та вертикальні зсуви на основі швидкості кроку, доки не буде пройдено повне зображення.

Рівень об'єднання (*POOL*): цей рівень відповідає за зменшення розмірності. Це допомагає зменшити обчислювальну потужність, необхідну для обробки даних. Існує два типи об'єднання: максимальне об'єднання та середнє об'єднання. Max pooling повертає максимальне значення з області, охопленої ядром на зображенні. Об'єднання середніх значень повертає середнє значення всіх значень у частині зображення, охопленій ядром.

Повністю зв'язаний рівень (*FC*): повністю зв'язаний рівень (FC) працює на плоскому вході, де кожен вхід з'єднаний з усіма нейронами. Якщо присутні, шари FC зазвичай знаходяться в кінці архітектур CNN.

CNN в основному використовується для вилучення ознак із зображення за допомогою його шарів. CNN широко використовуються в класифікації зображень, де кожне вхідне зображення проходить через серію шарів, щоб отримати імовірнісне значення від 0 до 1.

У генеративних моделях використовується підхід до неконтрольованого навчання (є зображення, але немає міток).

GAN складаються з двох моделей Generator і Discriminator. Генератор вчиться створювати фальшиві зображення, які виглядають реалістично, щоб обдурити дискримінатора, а Дискримінатор вчиться відрізнити фальшиві зображення від справжніх (він намагається не бути обдуреним).

Генератору заборонено переглядати реальні зображення, тому він може дати погані результати на початковій фазі, тоді як дискримінатору дозволено переглядати реальні зображення, але вони змішуються з підробленими, створеними генератором, які він повинен класифікувати як справжні або підробка.

Деякий шум подається як вхідний сигнал до генератора, щоб він міг щоразу створювати різні приклади, а не зображення одного типу. На основі балів, передбачених дискримінатором, генератор намагається покращити свої результати, через певний момент часу генератор зможе виробляти зображення, які буде важче розрізнити, у цей момент часу користувач буде задоволений його результати. Discriminator також вдосконалюється, оскільки отримує все більш реалістичні зображення з кожним раундом від генератора.

Популярними типами GAN є Deep Convolutional GAN (DCGAN), Conditional GAN (cGAN), StyleGAN, CycleGAN, DiscoGAN, GauGAN тощо.

GAN чудово підходять для створення зображень і обробки зображень. Деякі програми GAN включають: старіння обличчя, змішування фотографій, суперроздільність, малювання фотографій, переклад одягу.

Дані технології є основними підходами при використанні цифрових технологій в сучасних програмно-апаратних комплексах.

2.2 Алгоритми контурного аналізу цифрових зображень

Сучасні системи обробки інформації характеризуються високим рівнем використання різноманітних алгоритмів і технологій інтелектуалізації процесів обробки даних. Один із важливих напрямків у сфері інтелектуалізації інформаційних технологій є обробка інформації у вигляді фото- та відеоданих, зокрема цифрових зображень.

Технології інтелектуалізації процесів обробки зображень, затребувані при створенні систем відеоспостереження та контролю доступу. Процес обробки зображень складається з ряду процедур, серед яких однією з найважливіших є процедури сегментації об'єктів на записаних зображеннях. Деталі відносно простої форми характеризуються однозначно контуром предметів на зображеннях.

Ребра мають бути пов'язані в представлення меж регіону. Це зображення називається контуром. Контур може бути відкритим і закритим. Замкнуті контури відповідають межах області, а пікселі в області можна знайти за допомогою алгоритму заповнення. Розімкнутий контур може бути частиною межі області. На межі області можуть виникати прогалини, оскільки контраст між областями може бути недостатнім, щоб детектор країв міг знайти краї вздовж межі. Можливо, встановлено занадто високий поріг виявлення країв, або контраст уздовж певної частини межі може бути настільки слабким відносно інших ділянок зображення, що жодне порогове значення не працює всюди на зображенні. Відкриті контури також виникають, коли фрагменти лінії з'єднані між собою, наприклад, коли фрагменти лінії з'єднані вздовж штриха на малюнку чи зразку почерку. Контур може бути представлений у вигляді впорядкованого списку ребер або кривої. Крива — це математична модель контуру. Приклади кривихліній або відрізків включають відрізки ліній і кубічні сплайни.

Є кілька критеріїв для якісного представлення контуру. Саме на цих критеріях і відбувається загальна оцінка отриманих контурів (рисунок 2.5):

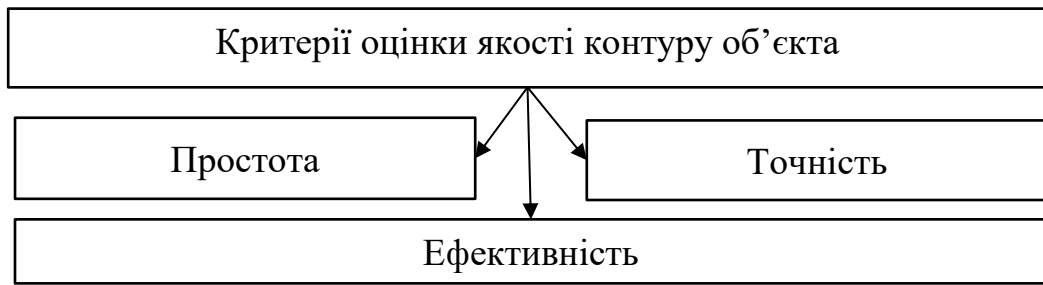


Рисунок 2.5 – Критерії для проведення оцінювання отриманих контурних функції

Простота: контур має бути простим, компактним зображенням.

Точність: контур повинен точно відповідати особливостям зображення.

Ефективність: Ефективність контура визначається його придатністю для подальших операцій, і точність представлення грає ключову роль у забезпеченні успішного нанесення.

Одним з базових способів представлення контуру є використання впорядкованого списку його ребер. Це просте та точне представлення, але воно може бути недостатньо компактним та неефективним для подальшого аналізу.

Вдосконалення точності контура досягається за допомогою підгонки відповідної моделі кривої до ребер контура. Це підвищує точність шляхом усереднення та зменшення помилок в розташуванні країв. Крім того, це забезпечує більш компактне та зручне представлення для подальших операцій.

Наприклад, для ефективного представлення набору ребер, які лежать уздовж лінії, можна використовувати підгонку лінії до цих ребер. Це спрощує обчислення орієнтації та довжини лінії і підвищує точність, оскільки помилка від підгонки лінії буде меншою, ніж помилка від використання ребер.

Важливим концептом є також розрізнення між інтерполяцією та апроксимацією. Інтерполяція вимагає, щоб крива точно проходила через всі точки, тоді як апроксимація дозволяє кривій проходити близько до точок, не обов'язково точно через них.

Загальною метою є збалансоване використання точності та ефективності представлення контуру для оптимальної обробки та аналізу зображень.

Краї, отримані за допомогою визначення країв, застосованого до реальних зображень, не будуть точними. Буде деяка помилка в розрахунковому розташуванні краю.

Список ребер – це впорядкований набір точок або фрагментів ребер.

Контур – це список ребер або крува, яка використовувалася для представлення списку ребер.

Границя – це замкнутий контур, який оточує область.

Орієнтація краю не використовується більшістю алгоритмів підгонки кривої. У кількох випадках, коли алгоритм використовує орієнтацію краю, з контексту буде зрозуміло, що термін «ребра» відноситься до фрагментів краю.

Планарні криві можна представити трьома різними способами:

- явною формою $y = f(x)$;
- неявною формою $f(x, y) = 0$;
- параметричною формою $(x(\kappa), y(\kappa))$ для деякого параметра u .

Явна форма рідко використовується в машинному зорі, оскільки крива в площині x - y може закручуватися таким чином, що на кривій може бути більше однієї точки для даного x . Параметрична форма кривої використовує дві функції, $x(\kappa)$ і $y(\kappa)$, параметра u для визначення точки вздовж кривої від початкової точки кривої в $P_1 = (X(U_1), y(U_1))$ до кінцевої точки $P_2 = (X(U_2), y(U_2))$.

Основна ідея полягає в тому, щоб оцінити орієнтацію дотичної за допомогою точок ребер, які не є суміжними у списку ребер. Це дозволяє отримати більший набір можливих орієнтацій дотичних. Нехай $P_i = (X_i, Y_i)$ – координати ребра i у списку ребер. K -нахил – це вектор напрямку (кута) між точками, відстань яких становить k країв. Лівий k -нахил - це напрямок від P_{i-k} до P_i , а правий k -нахил - це напрямок від P_i до P_{i+k} . K -кривизна – це різниця між лівим і правим k -нахилами. Припустимо, що в списку ребер є n граничних точок

$(X_l, Y_l), \dots, (X_n, Y_n)$. Довжина цифрової кривої може бути приблизно визначена додаванням довжин окремих сегментів між пікселями:

$$S = \sum_{i=2}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

Для зручного представлення контурної функції в системах обробки цифрових зображень використовується декілька підходів. Класифікацію алгоритмів контурів наведено на рисунку 2.6.

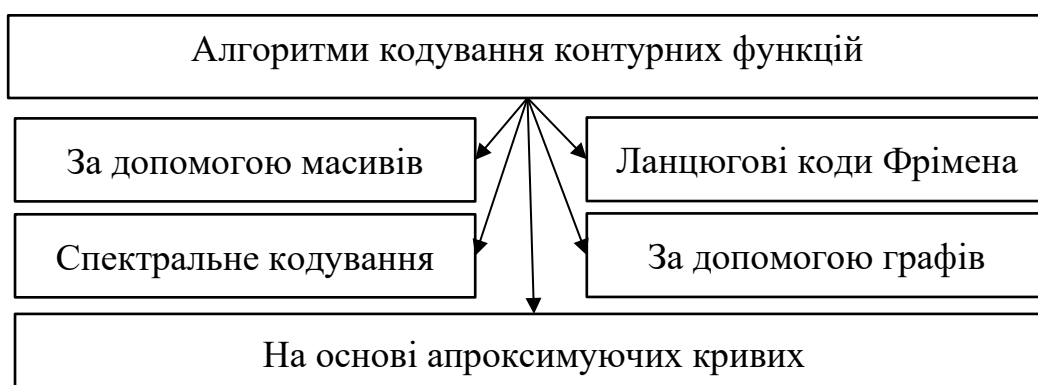


Рисунок 2.6 – Алгоритми кодування контурів об’єктів на цифровому зображенні

Ланцюгові коди – це позначення для запису списку крайових точок уздовж контуру. Код ланцюга вказує напрямок контуру на кожній грані в списку ребер. Напрямки квантуються в одному з восьми напрямків, як показано на рисунку 2.7.

2	3	4
1	.	5
8	7	6

Рисунок 2.7 – Схема кодування напрямів під час кодування за допомогою ланцюгових кодів

Починаючи з першого ребра в списку та рухаючись за годинниковою стрілкою навколо контуру, напрямок до наступного краю вказується за допомогою одного з восьми кодів ланцюга. Напрямок є ланцюговим кодом для 8-сусіда ребра. Ланцюговий код представляє список ребер за координатами першого ребра та список кодів ланцюга, що ведуть до наступних ребер. Крива та її код ланцюга наведені на рисунку 2.8.

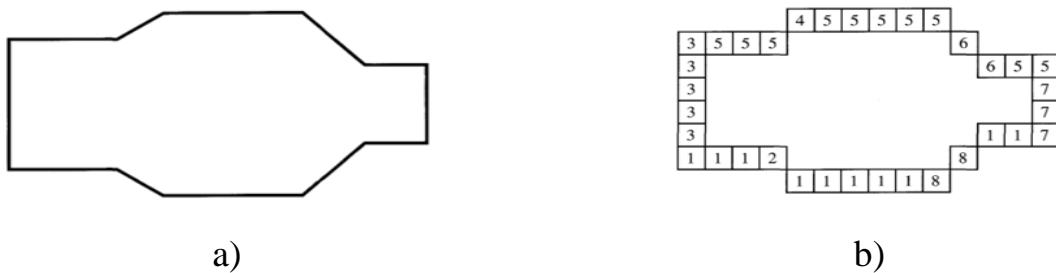


Рисунок 2.8 – Приклад кодування контурної функції: контур об'єкта (а) кодування контуру ланцюговим кодом Фрімена (b)

Ланцюговий код вражає своїм унікальним підходом до представлення об'єктів та забезпеченням зручних можливостей для їх обробки та аналізу. Одна з його властивостей - легкість реалізації поворотів об'єкта на конкретний кут, наприклад, 45° . Це можливо завдяки простому правилу: якщо об'єкт повернуто на $n \times 45^\circ$, код повернутого об'єкта отримується додаванням $n \bmod 8$ до вихідного коду. Такий підхід робить роботу з поворотами простою та ефективною. Ланцюговий код також має різницевий код, який служить описом межі та є інваріантним до поворотів. Це спрощує аналіз та роботу з кривими, що підлягають обертанню. Ще однією перевагою ланцюгового коду є можливість обчислення різних характеристик регіону, таких як площа та кути, безпосередньо з коду. Це дозволяє швидко та ефективно отримувати важливі геометричні величини для подальшого аналізу.

Однак важливо враховувати обмеження ланцюгового коду, пов'язане із фіксованим набором напрямків для представлення дотичної в точці. Це обмеження може бути подолано за допомогою розширених методів обробки

зображень. Алгоритм розбиття зверху вниз додає вершини рекурсивно, виходячи з початкової кривої. Це може бути особливо ефективним для апроксимації контурів об'єкта та зменшення похибки. Усі ці характеристики роблять ланцюговий код цікавим та потужним інструментом для обробки та аналізу цифрових об'єктів. Приклад операції апроксимації наведено на рисунку 2.9.

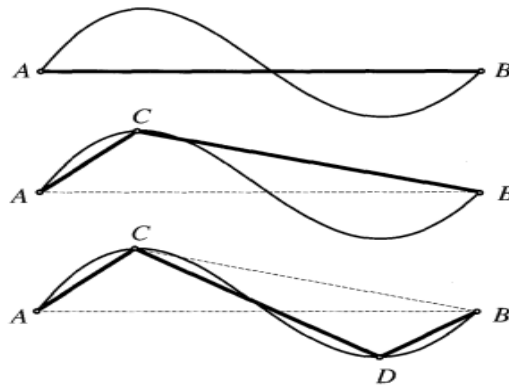


Рисунок 2.9 – Приклад апроксимації контурної лінії

Початкова крива представляє собою відрізок між першою та останньою точками ребра, позначеними як A і B. Основна ідея полягає в тому, щоб знайти точку в списку ребер, яка знаходиться найдалекше від прямої лінії. Якщо нормалізована максимальна похибка перевищує задане порогове значення, то вставляється нова вершина в крайову точку, яка знаходиться найдалекше від сегмента лінії, позначеного як точка C. Далі алгоритм розбиття рекурсивно застосовується до двох нових відрізків лінії та списку ребер. Список ребер розбивається на два підсписки, відповідно до двох відрізків лінії. Для кожного нового сегмента лінії знаходяться крайові точки в списку, які знаходяться найдалше від кожного сегмента, і вводяться нові вершини, якщо точки знаходяться занадто далеко від сегментів лінії.

Алгоритм завершується, коли нормалізована максимальна похибка для всіх крайових точок уздовж полілінії стає менше порогового значення. Ця рекурсивна процедура виявляється дуже ефективною. Підхід злиття також відомий як підхід "знизу вгору" для підгонки поліліній. Є ряд методів для

визначення того, чи крайова точка знаходиться занадто далеко від сегмента лінії. Один з таких методів використовує послідовний метод найменших квадратів, який підганяє сегмент лінії до країв за методом найменших квадратів і оновлює параметри сегмента лінії по мірі обробки кожної нової точки ребра. Алгоритм підгонки обчислює квадратичну відмінність між відрізком лінії та крайовими точками. Коли помилка перевищує порогове значення, вводиться вершина, і новий сегмент починається з кінцевої точки останнього сегмента.

Алгоритм зони допуску використовує інший метод для визначення положення вершин. Обчислюються два відрізки лінії, які паралельні відрізку лінії, що наближає крайові точки на відстані E від центрального відрізка лінії. Тут E представляє собою допустиму абсолютну величину відхилення від встановленої лінії. Ребра додаються до поточного сегмента лінії, доки нові ребра знаходяться в межах смуги допуску. Параметри сегмента лінії можуть бути перераховані, коли до сегмента додаються нові ребра. Апроксимуючий відрізок лінії не залишається паралельним сторонам смуги допуску, і вершина на його кінці стає початковою точкою для наступного відрізка. Цей метод часто приводить до великої кількості сегментів, і точність визначення розташування кутів і кутів не завжди є точною, оскільки вершина не створюється, доки алгоритм не обробить ребра до межі смуги допуску.

Безсумнівно, найкориснішою інформацією в багатьох задачах ідентифікації ознак об'єктів на цифровому зображенні є інформація про контури зображення (тобто про лінії, що проходять на кордонах однорідних областей). Крім того, дані про контурні лінії зображень дуже корисні для систем розпізнавання об'єктів, представлених у вигляді зображень.

В даний час для вирішення задачі виділення контурів об'єктів на зображеннях в основному використовуються такі три підходи:

- 1) підхід, заснований на обчисленні дискретних похідних;
- 2) підхід, заснований на статистичному аналізі яскравості поблизу кожного пікселя зображення;
- 3) підхід, заснований на використанні апарату теорії нечітких множин.

Підхід, заснований на обчисленні дискретних похідних. Усі методи, що розробляються в рамках цього підходу, базуються на обчисленні похідних першого та другого порядків і складаються з двох груп. Перша група методів покладається на оператори градієнта, а друга група покладається на оператори Лапласа. Основна ідея методів виділення градієнтних контурів полягає в пошуку набору пікселів, що відповідає істотним (максимальним) змінам дискретного диференціала першого порядку. У цьому випадку контури об'єктів визначаються на основі пошуку максимального модуля градієнтних векторів. Контурне виділення за допомогою лапласіана зводиться до визначення дискретної похідної другого порядку. Основна ідея методів, заснованих на операторах Лапласа, полягає в тому, щоб підкреслити розриви рівнів яскравості в зображенні і придушити область зі слабкою зміною яскравості. Основна перевага методів на основі операторів Лапласа полягає в тому, що вони працюють дуже швидко. Недоліками цих процедур є те, що вони чутливі до шумів на зображенні. Тому методи виділення контурів на основі операторів Лапласа практично не використовуються при розв'язуванні прикладних задач. Спеціальні лінійні інваріантні в часі (LTI) фільтри є найпоширенішою процедурою, що застосовується до проблеми вибору контуру, і такою, яка призводить до найменших обчислювальних зусиль. У випадку фільтрів першого порядку край інтерпретується як різка зміна рівня сірого між двома сусідніми пікселями. У цьому випадку мета полягає в тому, щоб визначити, в яких точках зображення перша похідна від рівня сірого як функція положення має велике значення. Застосовуючи порогове значення до нового вихідного зображення, виявляються краї в довільних напрямках. Ранні методи виявлення країв, такі як детектори Sobel і Prewitt, базувалися на концепції фільтрації просторової похідної, де оператори локального градієнта використовуються лише для виділення контурів певної орієнтації. Похідні фільтри страждають, коли межі розмиті, шумні та негнучкі. Марр і Хілдрет запропонували алгоритм, який знаходить ребра на нульовому перетині лапласіана зображення. Кенні запропонував метод протидії проблемам шуму від градієнтних операторів, де зображення згортається за

допомогою похідних першого порядку фільтра Гаусса для згладжування в напрямку локального градієнта з подальшим виділенням контурів за пороговим значенням. Більшість алгоритмів виділення контурів застосовують оператор локальної похідної першого та другого порядку, а потім певний метод видалення шуму для зменшення шуму. Слід зазначити, що методи виділення градієнтних контурів мають певні переваги (за чутливістю) порівняно з методами, заснованими на лапласівських розрахунках. Хоча ці методи найкраще працюють на реальних зображеннях порівняно з методом, заснованим на розрахунках Лапласа, вони не можуть забезпечити необхідну точність при обробці зображень низької якості. За цих умов можна використовувати статистичні методи для виділення контурів об'єктів на зображеннях.

Підхід, заснований на використанні апарату теорії нечітких множин. За останні кілька років теорія нечітких множин стала ще одним, але потужним інструментом для прийняття рішень. Заде запропонував концепцію нечітких множин, і незабаром ця концепція набула популярності в області обробки зображень. Багато методів було запропоновано дослідником для виявлення країв на основі апарату теорії нечітких множин. Розвиваються методи, засновані на апараті теорії нечітких множин і нечіткої логіки. Незважаючи на успіхи, досягнуті в області цифрової обробки зображень, існує ряд невирішених проблем. До них відносяться проблеми адекватного відображення предметної області в нечітку систему, вибір моделей нечіткого висновку та їх інтеграція в єдину інтелектуальну систему. У той же час багато з розроблених методів засновані на теорії нечітких множин вимагає великих обчислювальних ресурсів, що ускладнює їх застосування в прикладних системах, наприклад, в біометричних системах контролю доступу. Таким чином, питання обробки цифрових зображень за допомогою апарату нечітких множин досліджені недостатньо. Тому розробка та вдосконалення методів обробки цифрових зображень на основі теорії нечітких множин є дуже актуальними.

2.3 Алгоритм опису об'єктів на основі виділення контурів

На даний час розроблено низку алгоритмів, орієнтованих на виділення контуру об'єктів на зображеннях. Використання того чи іншого алгоритму при виділенні контурів здійснюється з урахуванням характеристик вихідного зображення, швидкості його обробки та з точки зору досягнення необхідного рівня якості. Підсвічування контурів є дуже корисним інструментом низькорівневої обробки зображень для аналізу зображень у полі зору комп'ютера та розпізнавання образів. На зображенні краї містять важливу інформацію про об'єкт, який цікавить зображення, оскільки вони розділяють різні області зображення. Тому виділення контурів є важливою частиною багатьох систем розпізнавання об'єктів, визначених як зображення. Методи, які використовуються в таких системах розпізнавання, передбачають існування набору ознак, виділених на основі аналізу та обробки відповідних зображень. Виділення ознак та формування опису зображення об'єкта на основі цифрової обробки зображень є одним із найважливіших етапів вирішення прикладних задач із використанням системи розпізнавання й аналізу зображень.

В основі запропонованого алгоритму пропонується використати алгоритм проходження контуром з зворотнім ходом, для отримання цілісного зв'язного коду. Після отримання усіх контурів на цифровому зображенні буде проведена процедура опису цілісної сцени шляхом встановлення рівнів взаємного розташування та відношення до самого фону зображення. Для однозначного опису зображень будемо розглядати контур цифрового зображення як границю першого рівня, на якій розташовані усі інші об'єкти. Контури об'єктів в свою чергу будуть формувати границю другого рівня для цілого зображення але будуть границями першого рівня для усіх об'єктів-деталей кі розміщені всередині об'єкта. Такий підхід дозволить створити компактний опис цифрового зображення. Запропонований алгоритм описаний за допомогою елементів блок-схеми та наведений на рисунку 2.10.

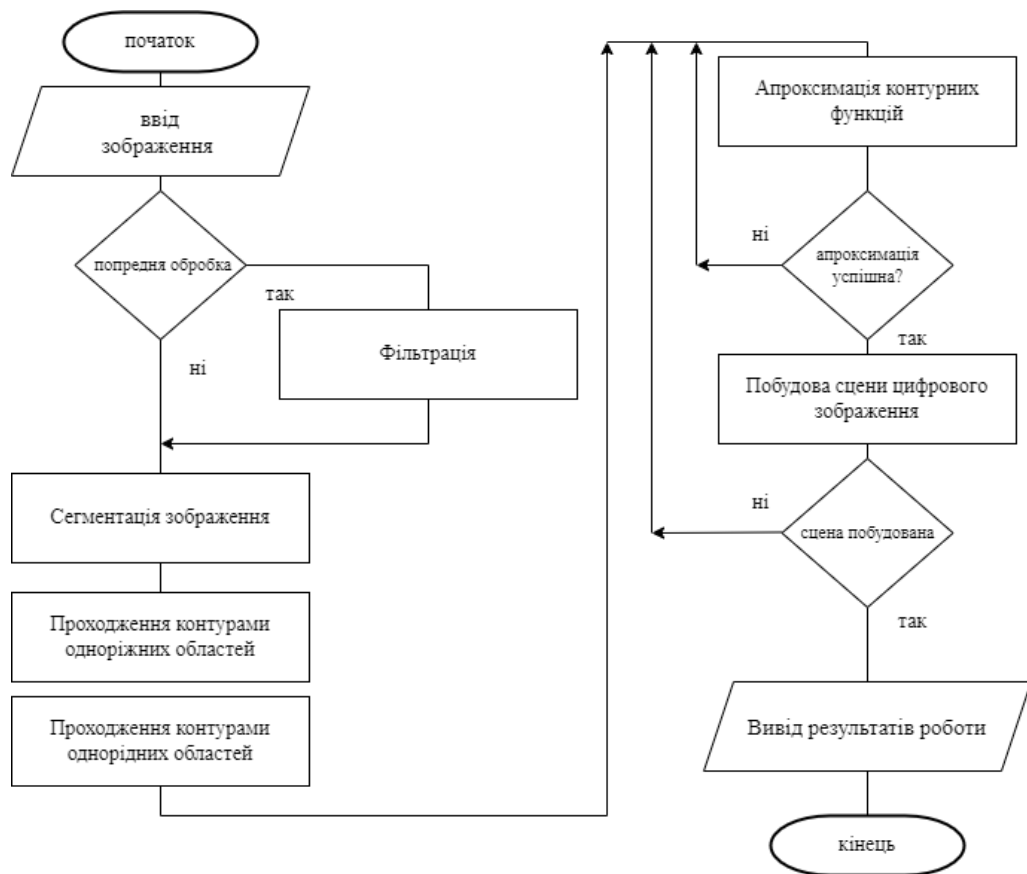


Рисунок 2.10 – Блок-схема алгоритму опису сцени цифрового зображення на основі контурного аналізу

Алгоритм опису сцени цифрового зображення на основі контурного аналізу включає ряд проміжних операцій над вхідним зображенням та містить наступну послідовність кроків:

- 1) Отримання вхідного зображення.
- 2) Попередня обробка цифрового зображення.
- 3) Сегментація вхідного зображення на основі обраного алгоритму.
- 4) Аналіз однорідних областей та проходження контуром з метою отримання контурної функції кожної з них у вигляді двовимірного масиву дани.
- 5) Апроксимація контурних функцій.
- 6) Маркування окремих контурів окремих однорідних областей.
- 7) Вибір критеріїв опису цифрового зображення.
- 8) Формування графа опису цифрового зображення.
- 9) Вивід отриманих результатів роботи алгоритму.

Запропонований алгоритм опису сцени цифрового зображення на основі контурного аналізу дозволяє опрацьовувати цифрові зображення в режимі реального часу та виконує усі поставлені в дослідженні задачі. До переваг даного алгоритму слід віднести:

- Простота реалізації та швидкість роботи;
- Можливість опрацювання зображень з об'єктами різної деталізації;
- Мінімальні вимоги до апаратних складових систем обробки та аналізу зображень;
- Можливість корекції результатів роботи в залежності від поставлених завдань.

Серед недоліків є:

- Складність опису об'єктів з невеликими внутрішніми областями;
- Необхідні коректні стартові параметри роботи.

В загальному запропонований алгоритм відповідає усім сучасним вимогам які ставляться до алгоритмів обробки цифрових зображень, а саме: працює в режимі реального часу, адаптується під параметри вхідного зображення може бути розпаралелений на етапі проходження контурами одноріжних областей.

2.4 Висновки до розділу

Провдений аналіз алгоритмів виділення та кодування контурів об'єктів на основі дослідження типів вихідних контурних функцій, що дозволило виділити підхід на основі апроксимації як максимально ефективний під час опису сцен на цифрових зображеннях;

Розроблено алгоритм опису сцен цифрових зображеннях на основі виділення контурів об'єктів та їх градації, що дозволило реалізувати програмний додаток обробки та опису цифрових зображень на основі контурного аналізу.

3 ПРОГРАМНИЙ ДОДАТОК АНАЛІЗУ ТА ОПИСУ ЦИФРОВИХ ЗОБРАЖЕНЬ

3.1 Структура програмного додатку опрацювання цифрових зображень

У сучасному світі програмні системи опису цифрових зображень на основі контурного аналізу відіграють важливу роль у різних галузях, починаючи від медичних досліджень до комп'ютерного зору та штучного інтелекту. Контурний аналіз надає потужний інструмент виділення форм, структур та меж об'єктів на зображеннях. Системи опису цифрових зображень на його основі дозволяють автоматизувати процеси розпізнавання та класифікації об'єктів, що є критичним у сучасній науці та технології. У медицині контурний аналіз знайшов застосування у галузі діагностики та обробки медичних зображень. Програмні системи, що використовують цей метод, здатні виділяти контури органів, визначати зміни в тканинах та допомагають лікарям у більш точних діагнозах. Це особливо важливо у галузі онкології, де раннє виявлення змін може врятувати життя пацієнтів. У комп'ютерному зір контурний аналіз є ключовим елементом для створення систем розпізнавання образів. Він дозволяє комп'ютерам розуміти форму та структуру об'єктів, що необхідно для автоматичного розпізнавання осіб, транспортних засобів, предметів та інших елементів. Ці технології широко використовуються в системах відеоспостереження, автономних автомобілях та інших сферах. У сфері штучного інтелекту контурний аналіз стає важливим інструментом на навчання машинного зору. Моделі, навчені аналізувати контури, здатні краще розуміти зміст зображень і робити точніші висновки. Це особливо корисно у завданнях, де важливі деталі та структури об'єктів.

Системи опису цифрових зображень на основі контурного аналізу активно застосовуються в обробці графіки та дизайну. Вони допомагають автоматично виділяти грані об'єктів, створювати візуально привабливі композиції та оптимізувати процеси обробки зображень у графічних редакторах. З розвитком технологій і появою складніших алгоритмів контурний аналіз стає дедалі

точнішим і ефективнішим. Системи можуть автоматично коригувати параметри аналізу залежно від складності зображення. Однак, незважаючи на всі переваги, існують виклики. Точність аналізу може знижуватися в умовах низького освітлення, за наявності шуму на зображенні або за наявності об'єктів зі схожими контурами.

Внутрішня архітектура програмних систем опису цифрових зображень, заснованих на контурному аналізі, є складною відносно технологічних рішень та інновацій які в них використовуються. На початку розробки будь-якої системи опису цифрових зображень стоїть вибір підходів до контурного аналізу. Це може включати різні методи виділення контурів, такі як оператори Собеля, Кенні, або використання алгоритмів межі об'єкта. Такий вибір залежить від конкретних вимог системи та характеристик зображень, з якими вона працюватиме. Для роботи будь-якої системи даного типу необхідно отримання вихідних даних. Зображення можуть надходити з різних джерел: камери відеоспостереження, медичні сканери, знімки з супутників тощо. Важливо, щоб система ефективно обробляла різноманітні формати зображень та адаптувалася під різні умови зйомки. Після завантаження цифрового зображення в програмну систему активуються функції попередньої обробки зображень. На цьому етапі проводиться усунення шумів, нормалізація яскравості та контрастності. Це важливо для отримання більш точних та послідовних результатів контурного аналізу. Внутрішній механізм системи повинен бути гнучким і настроюваним, щоб адаптуватися до різних типів зображень.

Процес виділення контурів – один з ключових елементів внутрішньої архітектури. Для його роботи використовуються різні математичні методи визначення переходів між областями різної яскравості. Може застосовуватися детекція кордонів за градієнтом, виділення країв за кольором чи текстурою. Система повинна бути здатна адаптуватися під різні типи об'єктів і сцен. Наступним етапом є побудова та аналіз контурів. Внутрішня архітектура системи включає алгоритми, здатні обробити згенеровані контури і витягти з них потрібну інформацію. Тут застосовуються техніки фільтрації та класифікації

виділення значних контурів і відкидання зайвих. Системи можуть використовувати різні методи опису контурів, включаючи цифрові ланцюги, апроксимації поліномами, або дескриптори Фур'є. Цей етап внутрішньої архітектури відіграє важливу роль у тому, як інформація інтерпретуватиметься та використовуватиметься на наступних етапах. Важливим елементом архітектури механізм обробки помилок. Контурний аналіз може зіткнутися з різними викликами, такими як наявність артефактів на зображенні, невизначеності у формі об'єктів або проблеми з освітленістю. Система має бути стійкою до таких факторів та надавати механізми корекції помилок. Внутрішні алгоритми системи також мають бути ефективними з погляду обчислювальних ресурсів. Це важливо для забезпечення високої продуктивності системи, навіть при обробці великих обсягів даних.

На основі проведеного дослідження та виділення основних етапів та структурних модулів відомих програмних систем аналізу та опису цифрових зображень було запропоновано структуру програмного додатку опису цифрових зображень на основі алгоритмів контурного аналізу (рисунок 3.1).

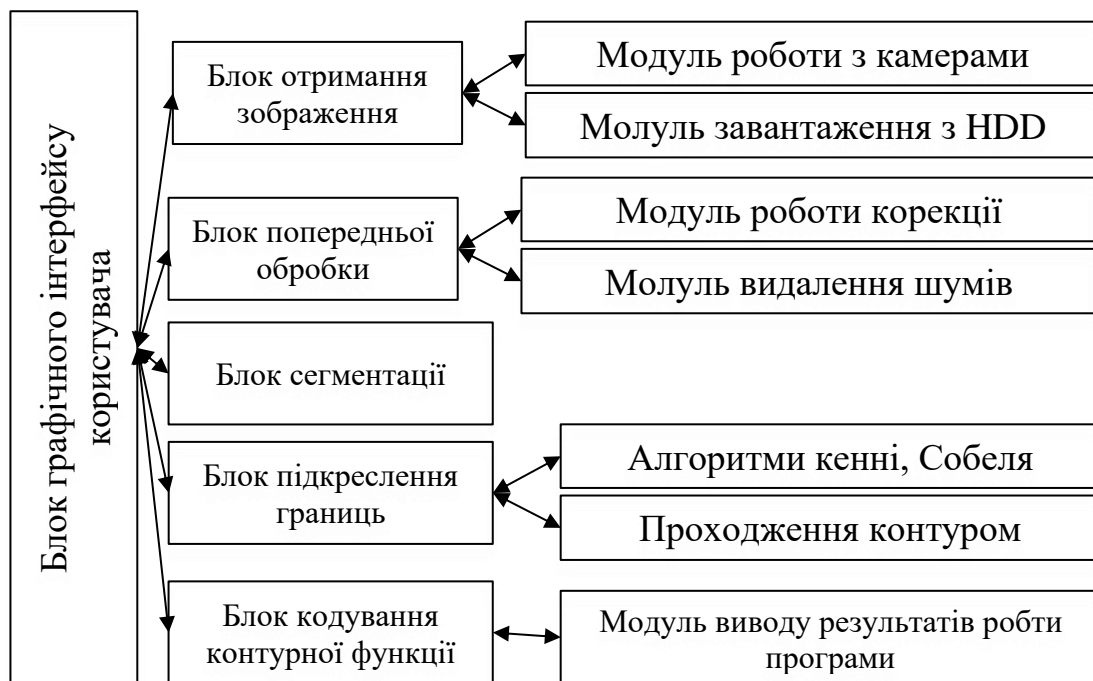


Рисунок 3.1 – Узагальнена структура додатку опису цифрових зображень

Запропонована структура програмного додатку передбачає можливість його адаптації під більш вузькі задачі аналізу та опису цифрових зображень. Окрім того вона дозволяє спростити розуміння внутрішніх потоків та процесів які відбуватимуться в програмній розробці. Розглянемо функції, вхідні та вихідні типи даних для кожного з виділених програмних модулів.

Блок введення зображення є першим етапом програмної системи аналізу цифрового зображення. Він включає механізми, що забезпечують збір даних з різних джерел, таких як камери або сховища зображень. Для зчитування зображень камери використовуються бібліотеки для взаємодії з обладнанням, забезпечуючи стабільний потік вхідних даних. Крім того, блок повинен підтримувати завантаження зображень з різних форматів, що зберігаються в сховищі, для забезпечення гнучкості в роботі із заздалегідь зібраними даними. Важливо, щоб цей блок забезпечував уніфікацію вхідних даних для наступних етапів обробки, гарантуючи їхнє коректне подання в програмі.

Блок попередньої обробки є ключовою ланкою у ланцюжку обробки цифрового зображення. Його основне завдання – підготовка вхідних даних для більш глибокого аналізу. У цьому блоці реалізуються процеси, такі як усунення шумів, корекція освітленості та контрастності, а також приведення зображення до стандартного формату. Алгоритми фільтрації можуть бути використані для поліпшення якості зображення та видалення артефактів, які могли виникнути на етапі введення. Важливо, щоб цей блок налаштовувався, враховуючи різноманітність умов зйомки та особливостей зображень.

Виділення кордонів відіграє важливу роль в аналізі цифрових зображень, і цей блок є відповідальним за застосування операторів Собеля та Кенні. Оператори Собеля використовуються виявлення градієнтів інтенсивності у зображенні, що дозволяє виділити краю об'єктів. Оператор Кенні, у свою чергу, доповнює процес, покращуючи деталізацію та придушення шумів. У цьому блоці можуть включатися механізми вибору оптимальних параметрів операторів залежно від конкретних особливостей оброблюваних зображень. Гнучкість та

ефективність цих операторів відіграють важливу роль як передумови для точного подальшого аналізу.

Сегментація зображень - наступний етап, який спрямований на виділення окремих об'єктів та областей у зображенні. У цьому блоці можуть використовуватися різні методи, такі як методи граничної сегментації або сегментації на основі кольору та текстури. Це дозволяє виділити різні структури та компоненти зображення для більш детального аналізу на наступних етапах. Гнучкість у виборі методів сегментації важлива для обробки зображень з різними характеристиками та змістом.

Блок кодування контуру. Коли контури об'єктів виділено, блок кодування контуру перетворює їх у формат, зручний для аналізу та зберігання. Це може включати використання цифрових цепів, описів поліномами або інших методів. Важливим елементом є мінімізація обсягу даних при збереженні інформації про форму та структуру об'єктів. Результат роботи цього блоку стане основою для наступних етапів аналізу та розпізнавання.

Блок виведення результатів на екран і файл. Останній блок включає механізми для візуалізації результатів аналізу на екрані і збереження їх у файл. Візуальне подання може включати відображення виділених контурів, сегментованих областей та інших важливих елементів. Це важливо для візуальної перевірки правильності роботи системи. Збереження результатів у файлі забезпечує можливість подальшого використання даних для додаткового аналізу або інтеграції з іншими програмами.

Запропонована внутрішня архітектура програмної системи має ряд видатних переваг, які зробили її ефективною та затребуваною в сучасному світі цифрової обробки зображень. Завдяки блоку введення, спроектованого для роботи з камерами та сховищами зображень, архітектура демонструє гнучкість в обробці різноманітних джерел даних. Це дозволяє легко інтегрувати систему з різними пристроями та джерелами зображень, що особливо цінно в контексті зростання різноманітності цифрових пристроїв. Використання алгоритмів попередньої обробки дозволяє зменшити час опрацювання цифрових зображень

на пізніх етапах аналізу. Усунення шумів, корекція яскравості і контрастності як поліпшують якість вхідних даних, а й створюють основу точнішого аналізу наступних етапах. Виділення в окремий блок функції підкреслення границь з використанням операторів Собеля та Кенні демонструє високу точність у виявленні градієнтів інтенсивності. Це важливо для правильного виділення контурів об'єктів і забезпечує більш точні результати аналізу. Технологія сегментації забезпечує точне виділення окремих однорідних областей у зображенні. Це сприяє детальному аналізу та дозволяє системі працювати з різноманітними структурами зображень. Кодування контуру контуру дозволяє отримати компактні описи об'єктів, зберігаючи при цьому необхідну інформацію для подальшого аналізу. Не лише оптимізує використання пам'яті, а й спрощує зберігання даних. Завершуючи цикл обробки, блок виведення забезпечує зручний механізм візуалізації результатів та їх збереження у файл. Це не тільки робить систему прозорою та зрозумілою для кінцевого користувача, але й надає можливість подальшого аналізу та обміну даними.

Таким чином, архітектура програмної системи аналізу цифрових зображень стає оптимальним інструментом для широкого спектру завдань, де потрібна висока гнучкість, точність та ефективність обробки візуальних даних. Внутрішня архітектура програмних систем опису цифрових зображень на основі контурного аналізу є складним поєднанням математичних методів, алгоритмів обробки, аналізу та опису цифрових зображень.

Після проведення процедури проектування внутрішньої архітектури програмної системи та вибору усіх алгоритмів які будуть використовуватись для реалізації окремих модулів було проведено моделювання. Під час моделювання структури програми основну увагу було приділено аналізу протікання внутрішньопрограмних процесів та моливості доступу користувачів до функціоналу програмного додатку. Для проведення модулювання було використано апарат універсальних мов модулювання UML. Вибір саме такого тежнічного апарату був зобумовлений тим, що UML є широко поширеним серед розробників, надає усі необхідні технічні моливості для проведення процедури

моделювання. Крім того, UML надає інструменти для візуалізації динамічних аспектів системи, таких як взаємодія об'єктів та зміна станів. Це особливо корисно при роботі зі складними системами, де важливо розуміти внутрішню взаємодію компонентів. Динамічні діаграми, такі як діаграми послідовності та діаграми станів, забезпечують краще моделювання та передбачення поведінки системи. Також, використання UML підтримує принципи модульності та повторного використання коду, сприяючи прискореному процесу розробки та зменшенню можливих помилок. Модель, створена із застосуванням UML, може бути основою для документації та автоматичної генерації коду, що в кінцевому підсумку сприяє більш ефективному та легко підтримуваному життєвому циклу проекту.

Діаграма прецедентів дозволяє оцінити можливості доступу груп користувачів до окремих функціональних груп розроленого програмного додатку. Також дана діаграма дозволяє оцінити повноту засобів які отримують користувачі в процесі експлуатації програмної розробки. Приклад самої діаграми наведено на рисунку 3.2.



Рисунок 3.2 – Діаграма прецедентів програмної системи

Отримані результати моделювання, що для роботи з програмним додатком було виділено одну групу користувачів яка отримала повний доступ до усіх систем програмної розробки. Це пов'язано з тим, що запропонований програмний додаток позиціонується к проскте рішення для отримання описів сцен на цифрових зображень та може використовуватись для ввирішення різного спектру питань або бути інтегрованим у більш складні системи.

Діаграма послідовностей призначена для проведення аналізу внутрішньої архітектури програмного додатку та перевірки коректності протікання та взаємодії внутрішніх процесів в середині самої системи. Результат моделювання наведено на рисунку 3.3.

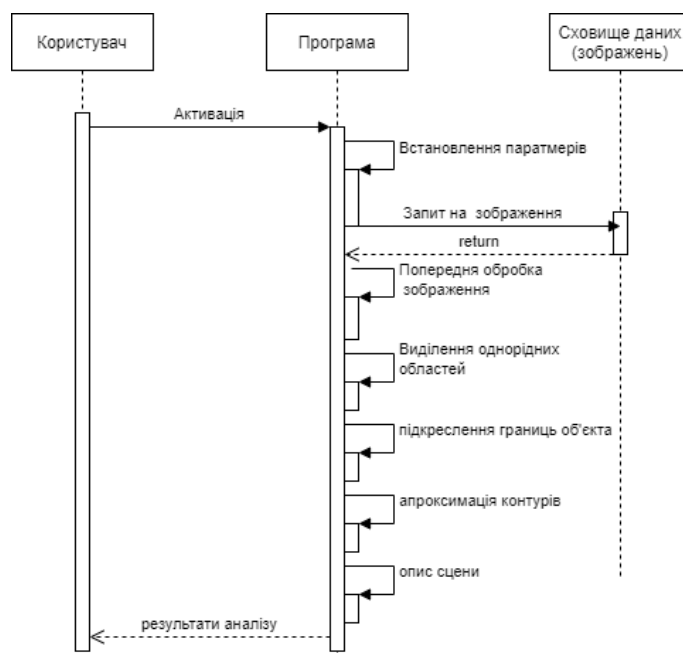


Рисунок 3.3 – Діаграма послідовності програмної системи

Основном завдання моделювання за допомогою діаграми посідовностей є аналізу запуску процесів, що відбуваються в середині програмного додатку. Отримані результати підтвердили, що внутрішні процеси програмного додатку запускаються у потрібній послідовності та не викликають внутрішніх конфліктів. Отже запропонована архітектура є коректною та дозволяє на її основі реалізувати програмну систему обробки цифрових зображень.

3.2 Модулі програмного додатку аналізу та оцінки програмного коду

Комп'ютерні системи зору, що обробляють зображення отримані з електронної камери, відтворює принципи людського зору, де обробка зображень відбувається аналогічно тому, як мозок обробляє отримані від ока враження. В сучасному світі, коли камери стали доступними та вартісно ефективними, а обчислювальні можливості та обсяг пам'яті комп'ютерів вражають своєю потужністю, комп'ютерний зір знаходить широке застосування в різноманітних галузях. Зокрема, системи бачення використовуються в промисловому виробництві для автоматизації процесів якості, а також в астрономії для обробки та аналізу зображень космосу. Значний внесок робиться в медицині та біометрії, де застосовуються методи розпізнавання обличчя та «текстури» райдужної оболонки очей для ідентифікації осіб. Ці технології виявляються дуже ефективними в сучасному світі, де забезпечення безпеки та ідентифікація осіб стають все більше актуальними завданнями. Важливо відзначити, що розвиток комп'ютерного зору відбувається паралельно з дослідженнями біологів та психологів, які досліджують роботу людської системи зору, що дозволяє постійно вдосконалювати існуючі алгоритми технології обробки зображень. Вивчення того, як працює наш орган зору та як взаємодіє з навколишнім світом, є ключовим елементом для подальшого вдосконалення та розвитку комп'ютерних систем зору.

Для прискорення процесу реалізації програмного коду для реалізації стандартних процесів обробки зображення була використана цифрова бібліотека роботи з цифровими зображеннями OpenCV (Open Source Computer Vision). Функції даної бібліотеки використовувались для реалізації алгоритмів вводу/виводу цифрових зображень, алгоритмів попередньої оцінки та обробки, розбиття цифрових зображень на однорідні області. Окрім того в програмній реалізації було використано структури даних які дозволяють описувати цифрові зображення у зручному для опрацюванні вигляді. Вибір програмної бібліотеки

OpenCV пов'язаний з декількома чинниками які роблять дану бібліотеку одним з найпотужніших засобів роботи з цифровими зображеннями. Дана бібліотека для алгоритмів комп'ютерного зору, була розроблена та використовується з відкритим вихідним кодом. Її основна мета – надати усім потенційним розробникам інструменти для створення програмних засобів, що зможуть проводити маніпуляції з цифровими зображення та відео в реальному часі. І якщо на початкових етапах свого розвитку вона була орієнтована на C і C++, то тепер існує багата підтримка для інших мов, таких як Python та Java. OpenCV пропонує широкий набір функцій для маніпуляції вхідними цифровими зображеннями. Це включає фільтрацію, зміна розміру, перекодування кольірних просторів, а також більш складні операції, такі як морфологічні перетворення. Іншим важливим напрямком реалізованих в бібліотеці алгоритмів є процеси виявлення та розпізнавання об'єктів на вхідних зображеннях. Це включає детекцію осіб, підкреслення контурів і зіставлення шаблонів. Окремо слід виділити можливості даної бібліотеки при роботі з цифровими камерами. OpenCV забезпечує можливості калібрування камер, оптимізацію параметрів цифрової камери, а також надає можливість виконання різних геометричних трансформацій, що є невід'ємним етапом для коректної обробки цифрових зображень з камер. Для оптимізації продуктивності OpenCV інтегрує підтримку графічних процесорів, що дозволяє виконувати деякі обчислювально інтенсивні операції в режимі реального часу. OpenCV має активну спільноту розробників, що забезпечує постійне оновлення бібліотеки. OpenCV є невід'ємним інструментом для розробників, які працюють в галузі комп'ютерного зору, машинного зору та робототехніки. Її часто інтегрують в системи які працюють з медичними зображеннями, автоматизованому відеоспостереженні, аналізі зображень та відео, ну і звісно у різних дослідницьких напрямках.

Для аналізу якості написання програмного доду розглянемо декілька методів які були реалізовані під час програмування та тестування ситеми обробки та опису цифрового зображення.

Для отримання більш якісного зображення над вхідним зображенням на етапі попередньої обробки проводиться ряд операцій, одна з таких операцій – це вирівнювання рівня яскравості на цифровому зображенні. Програмний код функції `correctBrightness()` наведено на рисунку 3.4.

```
void correctBrightness(const std::string& inputPath, const
std::string& outputPath) {
    cv::Mat image = cv::imread(inputPath, cv::IMREAD_GRAYSCALE);
    if (image.empty()) {
        std::cerr << "ERROR." << std::endl;
        return;}
    cv::equalizeHist(image, image);
    cv::imwrite(outputPath, image);
    cv::imshow("Original Image", cv::imread(inputPath));
    cv::imshow("Corrected Image", image);
    cv::waitKey(0);
    cv::destroyAllWindows();
}
```

Рисунок 3.4 – Функція корекції рівня яскравості вхідного зображення

Реалізована функція використовує механізми бібліотеки OpenCV та дозволяє швидко та на високі якості провести процедуру корекції вхідного зображення. Програмний код функції включає декілька етапів роботи. Визначаємо функцію корекції `Brightness()`, яка приймає шляхи до вхідного та вихідного зображень. Проводимо завантаження зображення у відтінках сірого за допомогою методу `cv::imread`. Для отримання відкорегованого зображення використовуємо метод для гістограмного вирівнювання `cv::equalizeHist`. Відкоректоване зображення зберігаємо за допомогою `cv::imwrite`. В результаті даний етап попередньої обробки завершив свою роботу.

Іншим, не менш важливим етапом обробки цифрового зображення це є отримання набору контурних функці. Дане завдання може бути вирішеним за допомогою одного з двох підходів. Перший підхід ґрунтується на основі використання операторів підкреслення границь. В програмі реалізовано декілька алгоритмів, проте одні з найкращих результатів було отримано під час використання оператора виділення границь Кенні. Даний алгоритм достатньо стабільно працює з сильно зашумленими зображеннями тому його доцільно

використовувати в системах даного типу. Оскільки він буде видавати максимально якісні результати. Приклад реалізації алгоритму підкреслення границь Кенні наведено на рисунку 3.5.

```
void applyCannyEdgeDetection(const std::string& inputPath,
const std::string& outputPath) {
    cv::Mat image = cv::imread(inputPath, cv::IMREAD_GRAYSCALE);
    if (image.empty()) {
        std::cerr << "Error." << std::endl;
        return;}
    cv::Mat edges;
    cv::Canny(image, edges, 50, 150);
    cv::imwrite(outputPath, edges);
    cv::imshow("Original Image", image);
    cv::imshow("Canny Edges", edges);
    cv::waitKey(0);
    cv::destroyAllWindows();
}
```

Рисунок 3.5 – Функція підкреслення границь на основі алгоритму Кенні

Реалізація обраного алгоритму включає декілька етапів. Визначення функції `applyCannyEdgeDetection()`, яка приймає шляхи до вхідного та вихідного зображень. Завантаження зображення у відтінках сірого за допомогою `cv::imread`. Застосування оператора Кенні за допомогою `cv::Canny` для виділення кордонів. Алгоритм `cv::Canny` виконується в декілька кроків. Спочатку зображення конвертується у відтінки сірого, щоб спростити обчислення. Потім застосовується оператор Гауса для згладжування зображення та усунення шуму. Далі обчислюються градієнти інтенсивності пікселів, що дозволяє визначити напрямок та силу зміни інтенсивності. На основі градієнтів визначаються краї, які вважаються межами об'єктів зображення. Оператор Кенні використовує порогові значення для визначення, які краї слід розглядати як межі, а які як шум або неважливі деталі. Фінальним кроком є застосування техніки придушення немаксимальних значень, що дозволяє зберегти лише локальні максимуми у напрямку градієнта. Це сприяє отриманню тонких, однопіксельних кордонів. Збереження зображення з виділеними межами за допомогою `cv::imwrite`.

Наступною задачею яку необхідно зробити під час реалізації процесу обробки цифрового зображення є процес апроксимації отриманих контурів. Дана

підзадача включена в загальний процес обробки і опису цифрових зображень для збільшення компактності кінцевого опису сцени на цифровому зображенні. При цьому задається порогове значення точності проведеної апроксимації, що дозволяє зберігати попустимий рівень похибки під час відтворення цифрового зображення на основі отриманого опису. Реалізація функції апроксимації контурної функції наведена на рисунку 3..

```

void approximateContour(const cv::Mat& inputMatrix, double
epsilon, cv::Mat& outputMatrix) {
    std::vector<std::vector<cv::Point>> contours;
    cv::findContours(inputMatrix, contours,
cv::RETR_EXTERNAL, cv::CHAIN_APPROX_SIMPLE);
    std::vector<std::vector<cv::Point>>
approxContours(contours.size());
    for (size_t i = 0; i < contours.size(); ++i) {
        cv::approxPolyDP(contours[i], approxContours[i],
epsilon, true);
    }
    outputMatrix = cv::Mat::zeros(inputMatrix.size(),
CV_8UC3);
    for (size_t i = 0; i < approxContours.size(); ++i) {
        cv::drawContours(outputMatrix, approxContours,
static_cast<int>(i), cv::Scalar(255, 255, 255), 2);
    }
}

```

Рисунок 3.6 – Функція підкреслення границь на основі алгоритму Кенні

Під час реалізації процесу апроксимації використовується функція `cv::findContours` для знаходження контурів у бінарному зображенні. Після чого ативується функція `cv::approxPolyDP`. Дана функція в бібліотеці OpenCV призначена для апроксимації кривої за допомогою алгоритму Дугласа-Пекера. Цей алгоритм дозволяє згладити злами в кривій, зменшивши кількість точок, зберігаючи при цьому її загальну форму. Алгоритм Дугласа-Пекера працює в такий спосіб: на вході в нього є набір точок, які мають криву. Алгоритм ітеративно вибирає найвіддаленішу точку від прямої, що з'єднує початкову та кінцеву точки набору. Якщо ця точка знаходиться на відстані більшій, ніж задана похибка (параметр `epsilon`), то вона додається до апроксимованої кривої. Процес повторюється рекурсивно двох нових наборів точок, утворених обраною точкою.

Функція `cv::approxPolyDP` приймає на вхід контур (список точок) і повертає новий контур, що представляє апроксимовану криву. Важливим аспектом є можливість керування ступенем апроксимації за допомогою параметра `epsilon`. Великі значення `epsilon` призведуть до більш грубої апроксимації, зберігаючи лише основні риси кривої, у той час як менші значення враховуватимуть дрібніші деталі. Цей алгоритм корисний для обробки контурів зображень, особливо у комп'ютерному зорі, де зменшення шуму і спрощення форми контурів може поліпшити продуктивність подальших обчислень. Наприклад, у розпізнаванні образів, візуальній обробці, або в інших завданнях, де апроксимація кривих відіграє ключову роль. Значення `epsilon` задає допустиму похибку для апроксимації. Окрім того в реалізації було використано стандартну функцію `cv::threshold`. Вона дозволяє швидко виконати просту обробку цифрового зображення з розподілом міток точок в залежності від обраних вхідних параметрів.

3.3 Тестування та аналіз реалізованого програмного додатку

Автоматизовані системи аналізу та опису сцен на цифрових зображеннях стають все більш важливим елементом сучасних технологій, що забезпечує низку переваг у багатьох галузях від розробки програм до медицини та безпеки. Ці системи використовують різноманітні техніки комп'ютерного зору та обробки зображень для розпізнавання об'єктів, сцен та подій. Одним із ключових переваг є швидкість та точність, з якими вони можуть аналізувати великі обсяги даних у реальному часі. Також автоматизовані системи відіграють важливу роль у відомчих дослідженнях, наукових проектах та архітектурних дослідженнях. За допомогою аналізу сцен та об'єктів на зображеннях, вони дозволяють швидше та ефективніше отримувати інформацію, що є важливою умовою для прогресу в цих галузях. У світі програмування та розробки програм, системи аналізу сцен

широко використовують для роботи з великими наборами даних, виявлення паттернів та створення інтелектуальних програм. Вони сприяють автоматизації завдань, що раніше вимагали значного зусилля та часу для вирішення. Загалом автоматизовані системи аналізу та опису сцен на цифрових зображеннях інтегруються в нашу повсякденну діяльність, полегшуючи багато аспектів нашої життя та удосконалюючи ефективність роботи в різних галузях.

На етапі тестування програмної системи важливим є вибір параметрів робочої станції для виконання тестових завдань. Низькі параметри робочої станції призведуть до спотворення результатів роботи самої розробленої програми. І навпаки, якщо технічні параметри будуть занадто високими, то результати будуть спотворені за рахунок високої швидкості опрацювання цифрового зображення. Технічні параметри наведено на рисунку 3.7.

Назва параметра	Значення
Процесор (CPU):	Intel Core i7-11700K
Оперативна пам'ять (RAM):	Corsair Vengeance LPX 32GB DDR4 3200MHz
Жорсткий диск:	Samsung 970 EVO Plus 1TB NVMe SSD
Графічний процесор (GPU):	NVIDIA GeForce RTX 2080
Операційна система:	Windows 10 Pro
Материнська плата:	ASUS ROG Strix Z590-E Gamin
Бездротові технології:	Wi-Fi 6 (802.11ax), Bluetooth 5.2
Порти та роз'єми:	USB 3.2 Gen 2, Thunderbolt 4, HDMI, DisplayPort, Ethernet
Корпус:	NZXT H510 Mid Tower
Блок живлення:	EVGA SuperNOVA 750 G5, 80 Plus Gold

Рисунок 3.7 – Технічні параметри робочої станції для тестування

Обраний комп'ютер має досить високий рівень продуктивності і призначений для використання у великому спектрі завдань. Переваги такої конфігурації можна описати наступними тезами. В нього потужний процесор і графічна картка тому він ідеально підходить для обробки цифрових зображень. Збільшений обсяг оперативної пам'яті і швидкий SSD дозволяють зручно працювати з великими обсягами даних у графіці, відео- та аудіоредагуванні. Завдяки потужній графічній карті і великому обсягу пам'яті, цей ПК може бути

використаний для відтворення відео високої якості, потужних аудіосистем і стрімінгу мультимедійного контенту. Враховуючи ці параметри, комп'ютер підходить для широкого спектру завдань і відповідає потребам високопродуктивного використання. Тому такі параметри у повній мірі задовільняють усі поставлені перед тестерами потреби та дозволить отримати об'єктивні результати в процесі тестування з при різних елементах складності вхідних цифрових зображень.

Окрім того під час тестування було виділено чотири типи можливого опису сцени на вхідному зображенні. Дані групи підходів було виділено з розрахунком на те, як саме буде трактуватись кожна границя в порівнянні з сусідніми. Для прикладу проаналізуємо опис сцени для зображення, яке містить пару об'єктів, що мають внутрішні однорідні області. Приклад такого зображення наведено на рисунку 3.8.

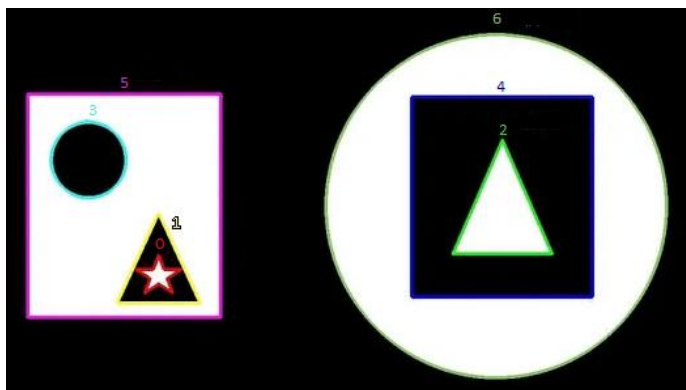


Рисунок 3.8 – Приклад цифрового зображення з парою об'єктів які мають внутрішні однорідні області

Опишемо усі варіанти можливого опису сцени на цифровому зображенні:

Режим списку. Під час опису сцени за допомогою даної піходу усі границі вважаються рівноцінними і не утворюють жодної ієрархії. Даний спосіб може використовуватись і випадку простих об'єктів без внутрішніх мікрообластей. Приклад такого опису сцени наведено на рисунку 3.9.

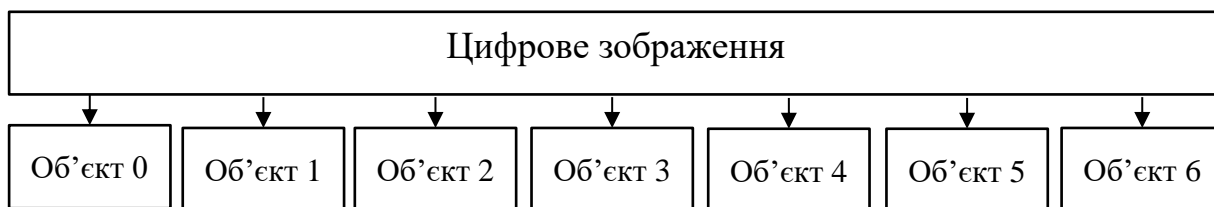


Рисунок 3.9 – Приклад опису сцени на цифровому зображенні в режимі списку

Режим основних об'єктів. Якщо використовувати даних підхід. То для опису сцени будуть обиратись тільки ті границі, які знаходяться на найвищому рівні. Тобто буде враховуватись тільки зовнішні границі об'єктів, в то й же час вся внутрішня архітектура таких об'єктів буде ігноруватись. Даний підхід доцільно використовувати у випадку якщо необхідно вибілити тільки основні великі об'єкти, наприклад, у випадку систем моніторингу, де основною задачею є виділення порушника, а не його детальни опис. Приклад такого опису сцени наведено на рисунку 3.10.



Рисунок 3.10 – Приклад опису сцени на цифровому зображенні в режимі основних об'єктів

Режим опису з фоном. Під час даного опису якщо в середині об'єкта існує одноріжна область яка має подібні кольорові характеристики, що і фон, та такі області прирівнюються до фону. А це означає, що об'єкти мають «порожнини, діри». В такому випадку створюється дворівнева ієрархія. Такий спосіб опису сцени доцільно використовувати, якщо наперед відомо, що об'єкти на зображення дійсно містять «дірки». Для прикладу, на основі такого підходу можна описувати

зображення деталей транспортних засобів, які мають в собі пустоти. Приклад такого опису сцени наведено на рисунку 3.11.



Рисунок 3.11 – Приклад опису сцени на цифровому зображенні в режимі опису з фоном

Режим опису на основі дерева. Даний спосіб формує ієрархію у формі дерева яка дозволяє описати сцену у вигляді дерева взаємопов'язаних об'єктів області які включаються одна в одну. Даний підхід є максимально інформативним для опису внутрішньої структури будь якої сцени на цифровому зображенні. Проте для його отримання необхідно затратити більше часу на проведення аналізу та оцінки цифрового зображення. Приклад такого опису сцени наведено на рисунку 3.12.

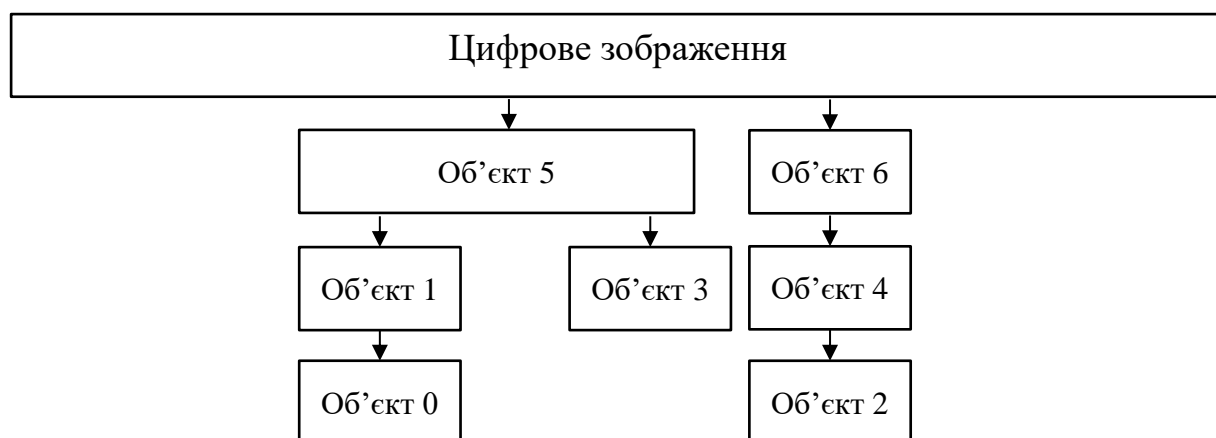


Рисунок 3.12 – Приклад опису сцени на цифровому зображенні в режимі опису на основі дерева

Для оцінки якості запропонованого алгоритму було проведено ряд досліджень. Одним з критеріїв якості результатів роботи запропонованого алгоритму було проаналізовано часові затримки які виникають під час роботи програмного додатку. Тестову вибірку для проведення дослідження було розбито на дві групи. В першій групі були зображення різного розміру та з простими об'єктами на них. Дослідження проводилось з метою оцінки збільшення часу опрацювання цифрового зображення під час збільшення його розмірів. Оскільки, запропонований алгоритм виконує обчислення дуже швидко, то для отримання результатів в алгоритм опрацювання було додатково закладено затримки в 10мс для після обробки кожного пікселя зображення. При менших затримках заміряти значення часу є проблематичним, а при збільшенні, загальний час виконання програми значно зростає, проте на загальні результати не впливає. Результати тестування наведено на рисунку 3.13.

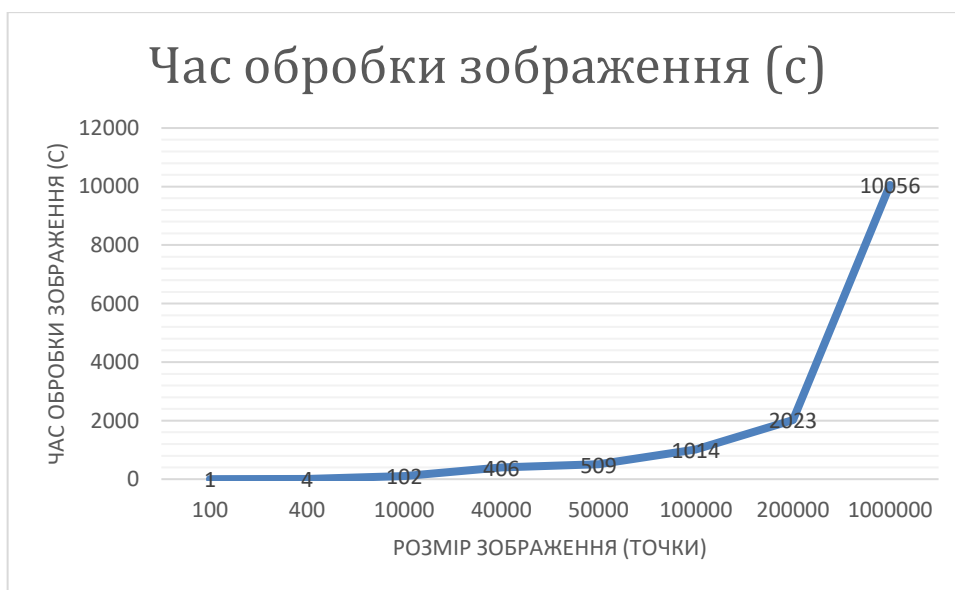


Рисунок 3.13 – Порівняння часових затримок при зміні розміру зображення

Як продемонстрували результати проведеного аналізу розмір зображення прямопропорційно впливає на час обробки цифрового зображення, оскільки чим більше точок на зображенні тим операцій необхідно провести для отримання кінцевого результату.

Окрім того було проведено дослідження роботи запропонованого алгоритму для опису сцен на цифрових зображення на основі контурного підходу для різних підходів під час опису. Порівняння здійснювалось на основі наперед визначеного набору критеріїв порівняння та особливостей вхідних зображень. Критерії порівняння:

- Складність внутрішньої структури однорідних областей. Однорідні області містять велику/малу кількість однорідних областей та дані характеристики необхідно враховувати під час опису сцен.
- Кількість однорідних областей на зображенні.
- Зашумленість вхідного зображення.
- Розмір вхідного зображення.
- Чіткість границь.

В результаті отримані в процесі тестування результати були згруповані та представлені у вигляді таблиці 3.1

Таблиця 3.1 – Узагальнена таблиця результатів тестування

Тестові групи	Складність внутрішньої структури	Кількість однорідних областей	Зашумленість	Розмір вхідного зображення	Чіткість границь
Режим списку	Вплив мінімальний	Пряма кореляція	Пряма кореляція	Пряма кореляція	Відчутний вплив
Режим основних об'єктів	Вплив мінімальний	Вплив мінімальний	Пряма кореляція	Пряма кореляція	Відчутний вплив
Режим опису з фоном	Сильний вплив на результати	Пряма кореляція	Пряма кореляція	Пряма кореляція	Відчутний вплив
Режим опису на основі дерева	Сильний вплив на результати	Пряма кореляція	Пряма кореляція	Пряма кореляція	

В результаті проведеного тестування програмної системи обробки та опису цифрових зображень на основі використання алгоритмів контурного аналізу було підтверджено початкові теоретичні оцінки, виконані усі поставлені завдання та доведено доцільність використання механік контурного аналізу під

час проектування та реалізації програмних систем даного типу. Серед переваг реалізованої програмної системи слід відзначити наступні:

- Виконання усіх операцій над цифровими зображеннями відбувається в режимі реального часу, що мінімазує час затрачений на отримання кінцевого результату.

- Можливість опрацювання цифрових зображень довільної складності, розміру, формату.

- Зображення можна отримувати як за допомогою камери в реальному часі, так і завантажувати з сховищ даних, що дозволяє значно підвищити функціональні можливості по отриманню вхідного зображення.

- В програмній системі реалізовано етап попередньої обробки вхідних зображень, що дозволило підвищити точність отриманих результатів та зменшити час опрацювання вхідних зображень.

- Програмна система дозволяє здійснювати опис за одним з чотирьох підходів, що дозволяє користувачеві більш ефективно використовувати отримані результати.

- В програмі реалізовано простий, інтуїтивно зрозумілий інтерфейс з мінімально видимими керуючими елементами, що дозволило значно спростити процес роботи з програмою та зменшити час навчання для її використання.

Серед недоліків розробленої системи:

- Реалізований механізм боротьби з зашумленістю не завжди в повній мірі дозволяє нівелювати вплив зовнішніх факторів на кінцевий результат роботи програми.

- В програмі можна опрацювати зображення обмеженої кількості форматів.

В загальному розроблений програмний додаток дозволяє виконати увесь цикл опрацювання цифрового зображення від етапу отримання до етапу збереження результатів опрацювання цифрового зображення за допомогою алгоритмів цифрової обробки зображень та контурного аналізу.

3.4 Висновки до розділу

Спроектовано та апроаналізовано внутрішню структуру програмної розробки для аналізу та опису цифрових зображень на основі використання контурного аналізу, що в подальшому надало можливість програмно реалізувати додаток обробки, аналізу та опису цифрових зображень для опису сцен на них.

Проведено дослідження програмного додатку для опису цифрових зображень на основі опису контурів об'єктів на них та проведено тестування з метою оцінки часових затримок, обчислювальної складності та порівняння з програмами аналогами запропонованого алгоритму та програмної розробки.

ВИСНОВКИ

На основі отриманих результатів при аналізі систем обробки цифрових зображень, а також розробленому алгоритму та спроектованій структурі програмного додатку можна зробити такі висновки:

1. Проведено дослідження цифрових зображень на основі технологій їх отримання та кодування, що дозволило провести їх класифікації та виділити основні характерні ознаки.

2. Проведено аналіз технологій опису цифрових зображень на основі аналізу внутрішніх об'єктів, що дозволило виділити основні характерні ознаки з максимальною інформативною складовою.

3. Проведено дослідження огляд програмних додатків аналізу та опису цифрових зображень на основі аналізу їх внутрішньої архітектури, що дозволило визначити основні структурні елементи які реалізуються в сучасних програмних системах обробки цифрових даних.

4. Провдений аналіз алгоритмів виділення та кодування контурів об'єктів на основі дослідження типів вихідних контурних функцій, що дозволило виділити підхід на основі апроксимації як максимально ефективний під час опису сцен на цифрових зображеннях.

5. Розроблено алгоритм опису сцен цифрових зображеннях на основі виділення контурів об'єктів та їх градації, що дозволило реалізувати програмний додаток обробки та опису цифрових зображень на основі контурного аналізу.

6. Проведено дослідження програмного додатку для опису цифрових зображень на основі опису контурів об'єктів на них та проведено тестування з метою оцінки часових затримок, обчислювальної складності та порівняння з програмами аналогами запропонованого алгоритму та програмної розробки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шевчук В. В., Гриненко А.А. Алгоритм опису об'єктів цифрових зображень на основі контурного аналізу. Збірник тез VIII науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», Тернопіль, 05 грудня 2023 р. с. 15.
2. Горішній Я. І., Шевчук В. В. Алгоритм сегментації цифрових зображень на основі аналізу характеристик однорідних областей. Збірник тез VIII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», Тернопіль, 05 грудня 2023 р. с. 17.
3. Belongie, S., Carson, C., Greenspan, H., and Malik, J. Color and texture-based image segmentation using EM and its application to content-based image retrieval. In Proc. 6th Int. Conf. Computer Vision, Bombay, India, pp. 675–682.
4. Belongie, S. and Malik, J. Finding boundaries in natural images: A new method using point descriptors and area completion. In Proc. 5th Euro. Conf. Computer Vision, Freiburg, Germany, pp. 751–766.
5. Binford, T. Inferring surfaces from images. *Artificial Intelligence*, 205–244.
6. Canny, J. A computational approach to edge detection. *IEEE Trans. Pat. Anal. Mach. Intell.*, 8(6):679–698.
7. Chung, F. *Spectral Graph Theory*, AMS. Providence, RI. DeValois, R. and DeValois, K. *Spatial Vision*. Oxford University Press. New York, N.Y.
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. *PAMI* 22(8) (2000) 888–905
9. Stella, X.Y.: Segmentation using multiscale cues. *CVPR*. (2004) I: 247–254
10. Fogel, I. and Sagi, D. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113.
11. Geman, S. and Geman, D. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.

12. Gersho, A. and Gray, R. Vector Quantization and Signal Compression, Kluwer Academic Publishers, Boston, MA. Heeger, D.J. and Bergen, J.R.
13. Pyramid-based texture analysis/synthesis. In Proceedings of SIGGRAPH '95, pp. 229–238.
14. Jacobs, D. Robust and efficient detection of salient convex groups. IEEE Trans. Pattern Anal. Mach. Intell., 18(1):23–37.
15. Jones, D. and Malik, J. Computational framework to determining stereo correspondence from a set of linear spatial filters. Image and Vision Computing, 10(10):699–708.
16. Gonzalez R., Wood R. Digital image processing, 2005. – 1072 p.
17. Belim S.V., Kutlunin P.E. Selection of contours in images using the clustering algorithm // Computer optics 2015. - № 39(1). - Pp. 119-124.
18. Mirzoev N. Model of formation of features of objects presented in the form of images // Dynamics of systems, mechanisms and machines, , 2016. – Volume 4. – No. 1. – pp. 150-154.
19. Marufova, Z.. EXPRESSION OF ORIENTAL BEAUTY THROUGH ZONYMS. Theoretical & Applied Science, (7), 13-16.
20. Burger W., Burge M. J. Digital Image Processing: An Algorithmic Introduction using Java. –New York: Springer, 2008. –564 p.
21. Marr, D., and Hildreth, E.C., “Theory of edge detection”, Proc. of the Royal Society of London, b207, , 187- 217.
22. Canny, J.F., “A computational approach to edge detection”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(6), 679-698.
23. Hou, T. H. and Kuo, W.L., 1997. “A New Edge Detection method for Automatic Visual Inspection”, International Journal of Advanced Manufacturing Technology, Springer-Verlag London Limited.
24. Jiang, X. and Bunke, H. “Edge Detection in Range Images based on Scan Line Approximation”, Computer Vision and Image Understanding ,Vol. 73, No. 2, February, pp. 183– 199,

25. Caragea S.. Fondater, Administrator and Chief Editor, IntelliProject, “Difference Edge Detection“, Licensed under IntelliProject open License, Romania.
26. Axbutayevich T. S., Abdumalikovich Q. N. Image contour separation algorithms based on the theory of fuzzy sets // International Journal of Contemporary Scientific and Technical Research. – 2022. – C. 120-125.
27. Genming, C. and Bouzong, Y.,. “A New Edge Detector with Thinning and Noise resisting Abilities”, Journal of Electronics, China, Volume 6, No. 4, 314-319
28. Zadeh . L.A. Fuzzy sets // Information and Control. № 8. - pp. 338-353. Texas Journal of Engineering and Technology ISSN NO: 2770-4491 <https://zienjournals.com>
Date of Publication: 12-12-2022
29. Kaufmann A. Introduction to the Theory of Fuzzy Subsets Fundamentals Theoretical Elements, Vol. 1. Academic Press, New York,
30. Kuo Y., Lee C., Liu C. A New Fuzzy Edge Detection Method for image Enhancement // IEEE Inter. Conf. on Fuzzy Systems, 1997, pp. 1069-1074.
31. El-Khamy S., Modified Fuzzy Sobel Edge Detector // Seventeenth National Radio Science Conference (NRSC'2000), February 22-24, Minufia, Egypt, 2000.
32. N. Apostoloff and A. Fitzgibbon. Bayesian estimation of layers from multiple images. In Proc. CVPR, volume 1, pages 407–414, 2004.
33. A. Blake, C. Rother, M. Brown, P. Prez, and P. H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In Proc. ECCV, p428–441, 2004.
34. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In Proc. ICCV, pages 105–112, 2001.
35. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI, 124–137, 2004.
36. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? PAMI, 26(2):147–159, 2004. 6. J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. IJCV, 43(1):7–27, 2001.
37. D. R. Martin, C. C. Fowlkes, Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI, 26(5):530–549, 2004.

38. B. Micusik and A. Hanbury. Steerable semi-automatic segmentation of textured images. In Proc. Scandinavian Conference on Image Analysis (SCIA), 2005.
39. N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: A level set approach. In Proc. ECCV, volume II, pages 224–240, 2000.
40. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. IJCV, 46(3):223–247, 2002.
41. M. Ruzon and C. Tomasi. Alpha estimation in natural images. In Proc. CVPR, volume 1, pages 18–25, 2000.
42. J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI, 22(8):888–905, 2000.
43. X. Y. Stella. Segmentation using multiscale cues. In Proc. CVPR, volume 1, pages 247–254, 2004.
44. Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In Proc. ECCV, volume 3, pages 487–501, 2002.
45. R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In Proc. CVPR, volume 2, pages 437–444, 2004.
46. Micusik, B., Hanbury, A.: Supervised texture detection in images. In: Proc. Conference on Computer Analysis of Images and Patterns (CAIP). - 2005 - 441–448
47. Estrada, F.J., Jepson, A.D.: Quantitative evaluation of a novel image segmentation algorithm. In: Proc. CVPR. (2005) II: 1132–1139 .
48. Gevers, T., Smeulders, A.: Color-based object recognition. Pattern Recognition 32(3) (1999) 453–64.
49. C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches. In Proc. of Conf. on Comp. Vision and Pattern Rec., pages 54–61, 2003.
50. A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In Proc. of European Conf. on Comp. Vision, 2006.
51. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and

measuring ecological statistics. In Proc. of 8th Int. Conf. on Comp. Vision, pages 416–423, 2001.

52.J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Proc. of British Machine Vision Conf., pages 384–393, 2002.

53.B. Micusik and A. Hanbury. Automatic image segmentation by positioning a seed. In Proc. of the European Conf. on Comp. Vision, volume 2, pages 468–480, 2006.

54.K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. Int. Journal of Comp. Vision, 65(1-2):43–72, 2005.

55.J. C. Pichel, D. E. Singh, and F. F. Rivera. Image segmentation based on merging of sub-optimal segmentations. Pattern Recognition Letters, 27(10):1105–1116, 2006.

56.J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

57.O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In Proc. of European Conf. on Comp. Vision, pages 589–600, 2006.

58.M. Vasconcelos, N. Vasconcelos, and G. Carneiro. Weakly supervised top-down image segmentation. In Proc. of Conf. on Comp. Vision and Pattern Rec., pages 1001–1006, 2006.

59.Березький О.М., Дубчак Л.О., Мельник Г.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 - Комп’ютерна інженерія. Магістерська програма - Комп’ютерна інженерія". Тернопіль: ЗУНУ, 2022. 32 с.

60.Гураль І.В., Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» Тернопіль: ТНЕУ, 2019. 33 с.