

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

Жеребний Олександр Володимирович

**«Алгоритми виділення та опису об'єктів
цифрових зображень на основі аналізу контурної
лінії / Algorithm for detection and description of digital
image objects based on contour line analysis»**

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

Виконав студент групи КІм-21
О.В. Жеребний

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

Кваліфікаційну роботу допущено
до захисту:

" ____ " _____ 20____ р.

Завідувач кафедри
_____Л. О. Дубчак

ТЕРНОПІЛЬ – 2024

АНОТАЦІЯ

Кваліфікаційна робота на тему “ Алгоритми виділення та опису об'єктів цифрових зображень на основі аналізу контурної лінії ” зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «магістр» написана обсягом 84 сторінки та містить 19 ілюстрацій, 1 таблиці, 2 додатки та 50 джерел за переліком посилань.

Метою роботи є розробка алгоритму виділення та опису об'єктів цифрових зображеннях на основі контурного аналізу.

Методи досліджень. Для розв'язання поставлених задач у кваліфікаційній роботі використано методи: попередньої обробки цифрових зображень (для оцінки та обчислення параметрів вхідного зображення); алгоритми теорії контурного аналізу (для виділення контурної лінії об'єкта на зображенні); об'єктно-орієнтованого програмування (для програмної реалізації запропонованих алгоритмів та структури програмної системи).

Результати дослідження: алгоритм виділення та опису об'єкта на цифровому зображенні на основі елементів контурного аналізу, програмний додаток виділення та опису цифрових зображень на мові python.

Результати роботи можуть бути використані в створенні складних систем обробки, опису та синтезу цифрових зображень, охоронних системах та системах діагностики тощо.

Орієнтовні напрямки розвитку досліджень: розроблення алгоритмів виділення границь зображень з автоматичним визначенням параметрів їх роботи, створення нових програмних засобів та моделей для навчання.

КЛЮЧОВІ СЛОВА: ЦИФРОВІ ЗОБРАЖЕННЯ, КОНТУРНИЙ АНАЛІЗ, ВИДІЛЕННЯ ГРАНИЦЬ ЗОБРАЖЕНЬ, ДЕТЕКТОР КЕННІ.

ANNOTATION

Graduate qualification work on “ Algorithm for detection and description of digital image objects based on contour line analysis ” specialty 123 – Computer Engineering is 84 pages long and contains 19 illustrations, 1 tables, 2 appendices and 50 references.

The aim of the work is develop an algorithm for selecting and describing objects in digital images based on contour analysis.

Research methods. To solve the tasks set in the qualification work, the following methods were used: pre-processing of digital images (for evaluating and calculating the parameters of the input image); algorithms of the theory of contour analysis (for selecting the contour line of an object in the image); object-oriented programming (for software implementation of the proposed algorithms and the structure of the software system).

Research results: an algorithm for selecting and describing an object in a digital image based on contour analysis elements, a software application for selecting and describing digital images in the python language.

The results of the work can be used in creating complex systems for processing, describing and synthesizing digital images, security systems and diagnostic systems, etc.

Approximate directions of research development: development of image boundary selection algorithms with automatic determination of their operation parameters, creation of new software tools and models for training.

KEYWORDS: DIGITAL IMAGES, CONTOUR ANALYSIS, IMAGE BOUNDARY SEARCH, KENNEY DETECTOR.

ЗМІСТ

Вступ.....	5
1 Технології зберігання інформації на основі цифрових кольорових зображень ..	8
1.1 Цифрові кольорові растрові зображення.....	8
1.2 Основні визначення та поняття теорії контурного аналізу	17
1.3 Порівняльний аналізу програм обробки цифрових зображень.....	22
1.4 Постановка задач дослідження.....	27
1.5 Висновки до розділу	28
2 Методи та алгоритми виділення та опису елементів цифрових зображень	29
2.1 Алгоритми аналізу цифрових зображень	29
2.2 Алгоритми виділення контурної лінії.....	37
2.3 Алгоритм виділення та опису об'єктів на цифрових зображеннях	46
2.4 Висновки до розділу	48
3 Програмний додаток обробки та опису цифрових кольорових зображень	49
3.1 Архітектура реалізованого програмного додатку	49
3.2 Функції програмного модуля обробки растрових зображень	61
3.3 Тестування та аналіз реалізованого програмного модуля	65
3.4 Висновки до розділу	68
Висновки	69
Список використаної літератури	70
Додаток А Лістинг програмного коду з операторами підкреслення границь зображення.....	Помилка! Закладку не визначено.
Додаток Б Світлокопії виданих публікацій ..	Помилка! Закладку не визначено.

ВСТУП

Актуальність роботи. Обробка цифрових зображень є невід'ємною частиною сучасних технологій в ряді дисциплін, від медицини до інженерії, де важливу роль відіграє контурний аналіз. Контурний аналіз — це метод, що дозволяє виявляти межі об'єктів на зображеннях, і є одним із основних інструментів для виділення важливих характеристик зображення, таких як форма, розмір, орієнтація та інші геометричні особливості об'єктів. Актуальність контурного аналізу стає очевидною в контексті сучасних потреб автоматизації обробки зображень, що впливає на широкий спектр галузей, включаючи комп'ютерне бачення, медичну діагностику, промисловість, робототехніку, а також в обробці відео та фотознімків. Однією з основних причин популярності контурного аналізу є його здатність точно виділяти важливі елементи на зображенні. Контури, як сукупність пікселів, що мають суттєві зміни у яскравості, дозволяють ідентифікувати перехід між різними зонами зображення, наприклад, між об'єктами та фоном або між різними об'єктами. Це, в свою чергу, надає можливість точніше інтерпретувати інформацію, що міститься в зображенні, та здійснювати подальшу обробку, таку як сегментація, класифікація чи ідентифікація об'єктів.

Іншим важливим аспектом є використання контурного аналізу в автоматизації промислових процесів, зокрема в системах контролю якості. На виробництві, де часто потрібно контролювати геометричні параметри деталей чи виявляти дефекти на зображеннях продукції, контурний аналіз дозволяє швидко виявити будь-які відхилення від заданих стандартів, забезпечуючи високу точність і швидкість виявлення дефектів. Це особливо важливо в таких галузях, як автомобільна промисловість або виробництво електронних компонентів, де навіть незначні дефекти можуть призвести до серйозних наслідків. З технічної точки зору контурний аналіз може бути реалізований за допомогою різних алгоритмів та методів.

Однак, контурний аналіз не є безперешкодним і стикається з певними труднощами. Зокрема, однією з основних проблем є наявність шумів, які можуть призводити до спотворення результатів. Для їх зменшення часто використовуються додаткові етапи передобробки зображень, такі як фільтрація або корекція освітлення. Окрім того, на складність контурного аналізу може впливати і рівень контрасту між об'єктами та фоном, особливо якщо зображення має низьку якість або коли об'єкти мають схожі кольори та текстури.

Враховуючи всі ці фактори, контурний аналіз є необхідним елементом в обробці зображень, оскільки дозволяє не тільки виділяти важливі риси об'єктів, але й покращувати якість і точність інших етапів обробки, таких як сегментація, класифікація чи виявлення аномалій. Тому задача дослідження задачі виділення та опису об'єктів цифрових зображень є актуальною.

Метою роботи є розробка алгоритму виділення та опису об'єктів цифрових зображеннях на основі контурного аналізу.

Для досягнення даної мети ставились наступні завдання:

- проаналізувати та провести класифікацію цифрових зображень;
- дослідити методи та алгоритми теорії контурного аналізу;
- провести аналітичний огляд цифрових бібліотек та програмних засобів обпрацювання цифрових зображень;
- проаналізувати алгоритми виділення та опису цифрових зображень;
- розробити алгоритм опису об'єктів цифрових зображень на основі аналізу контурної лінії;
- реалізувати програмний додаток виділення та опису об'єктів на цифрових зображеннях, провести його тестування та порівняння з аналогами.

Об'єкт дослідження – процес обробки цифрових зображень.

Предмет дослідження – методи і алгоритми контурного аналізу растрових зображень.

Наукова новизна одержаних результатів визначається наступним чином:

- здійснено аналітичний огляд методів та алгоритмів контурного аналізу для виділення та опису об'єктів на цифрових зображеннях, виділено їх переваги та недоліки в порівнянні з іншими підходами;

- розроблено алгоритм виділення та опису елементів цифрових зображень на основі дослідження їх контурної лінії, що дало можливість отримувати компактний та зручний для подальших досліджень опис цифрових зображень.

Практична цінність одержаних результатів полягає в тому, що:

- спроектовано внутрішню архітектуру програмного додатку обробки та аналізу цифрових растрових зображень в використання елементів контурного аналізу, що дозволило програмно реалізувати запропонований алгоритм опису;

- здійснено програмну реалізацію системи обробки та аналізу цифрових кольорових зображень, проведено її тестування та порівняння з програмами аналогами.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційній роботі, підготовлено тези доповіді «Алгоритми контурного аналізу для опису та кодування об'єктів цифрових зображень» обсягом 2 сторінки на X Всеукраїнській науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ'2024), а також «Аналітичний огляд технологій розпізнавання рукописних текстів» обсягом 2 сторінки на X Всеукраїнській науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ'2024).

В першому розділі проаналізовано технології та характеристики цифрових зображень, досліджено методи та алгоритми контурного аналізу.

В другому розділі було розглянуто підходи до кодування об'єктів на цифрових зображеннях та запропоновано власний алгоритм опису об'єктів.

В третьому розділі приведено опис програмної системи розробленої на основі запропонованих алгоритмів, проведено тестування та порівняння за аналогами.

1 ТЕХНОЛОГІЇ ЗБЕРІГАННЯ ІНФОРМАЦІЇ НА ОСНОВІ ЦИФРОВИХ КОЛЬОРОВИХ ЗОБРАЖЕНЬ

1.1 Цифрові кольорові растрові зображення

Зображення – це планове представлення сцени або об’єкта, зазвичай розташованого в двовимірному просторі, воно виникає в результаті контакту променів світла, що виходять від об’єктів, що утворюють сцену, з датчик (камера, сканер, рентген тощо). Насправді це просторове відображення світла. Зображення розглядається як набір точок, яким присвоєна фізична величина (яскравість, колір). Ці величини можуть бути безперервними (аналогове зображення) або дискретними (цифрове зображення). Математично зображення представляє неперервну функцію IF , яка називається функцією зображення, двох просторових змінних, представлених $IF(x, y)$, що вимірюють відтінок рівня сірого зображення в координатах (x, y) . Оскільки зображення це структурований набір інформації, що характеризується такими параметрами, можна виділити основні з них (рисунок 1.1).

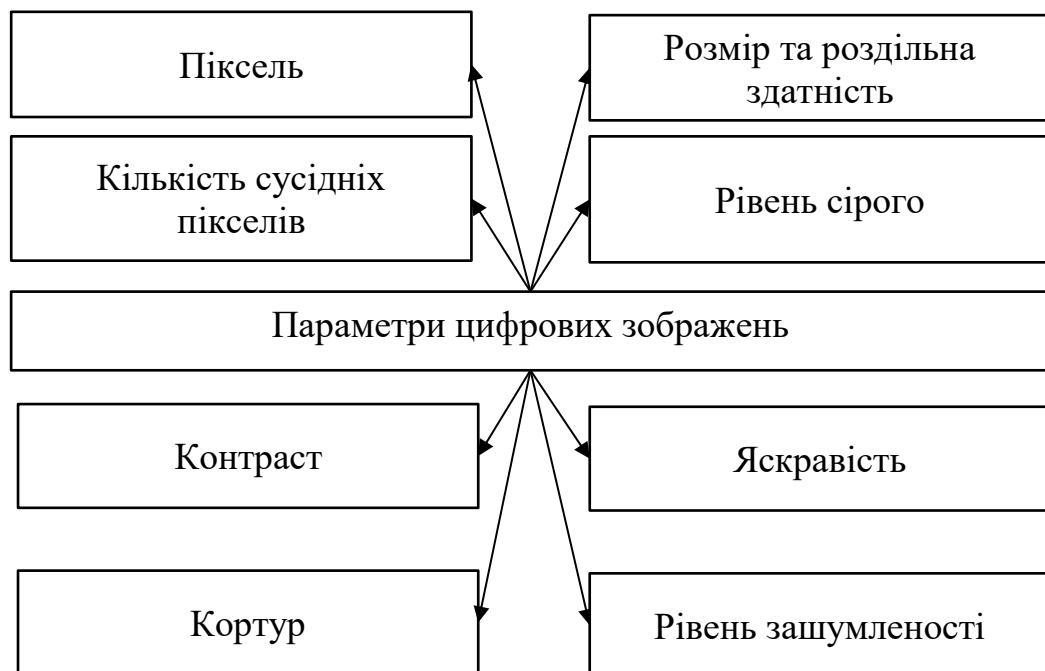


Рисунок 1.1 – Класифікація характеристики цифрових зображень

Піксель – це аббревіатура слова «елемент зображення», одиниця поверхні, яка використовується для визначити основу цифрового зображення. Він матеріалізує задану точку (x,y) площини зображення. Інформація, яку представляє піксель – це рівень сірого (або колір), взятий із відповідного місця на реальному зображенні. Різниця між монохромним і кольоровим зображеннями полягає в кількості інформації, що міститься в кожному пікселі, наприклад, у кольоровому зображенні (RGB: червоний, зелений, синій) значення пікселя представлено трьома байтами для кожного кольору.

Розмір – це розмір зображення. Він представлений у вигляді матриці, елементи якої є числовими значеннями, що представляють інтенсивність світла (пікселі). Кількість рядків у цій матриці, помножена на кількість стовпців, дає нам загальну кількість пікселів у зображенні. З іншого боку, роздільна здатність - це чіткість або тонкість деталей, досягнута монітором або принтером при створенні зображень. На комп'ютерних моніторах роздільна здатність виражається в кількості пікселів на одиницю вимірювання (дюйм або сантиметр). Слово роздільна здатність також використовується для позначення загальної кількості горизонтальних і вертикальних пікселів на моніторі. Чим більше це число, тим краща роздільна здатність.

Кількість сусідів для окремої точки може бути різною. Площина зображення розділена на прямокутні або шестикутні форми, що дозволяє використовувати поняття околиць. Околиці пікселя утворюють усі пікселі, розташовані навколо цього самого пікселя. Також визначаємо базу як набір пікселів, що визначають околиці, враховані навколо пікселя. Розрізняють два типи сусідства (рисунок 1.2):

- сусідство до 4 (беремо до уваги лише ті пікселі, які мають спільну сторону з розглянутим пікселем).
- сусідство до 8 (беремо до уваги всі пікселі, які мають принаймні одну точку, пов'язану з розглянутим пікселем).

	$P(i, j-1)$	
$P(i-1, j)$	$P(i, j)$	$P(i+1, j)$
	$P(i, j+1)$	

а)

$P(i-1, j-1)$	$P(i, j-1)$	$P(i+1, j-1)$
$P(i-1, j)$	$P(i, j)$	$P(i+1, j)$
$P(i-1, j+1)$	$P(i, j+1)$	$P(i+1, j+1)$

б)

Рисунок 1.2 – Приклад Визначення рівня сусідства між пікселями: (а) 4 сусіда та (б) 8 сусідів

Рівень сірого Це значення інтенсивності світла пікселя. Це значення може варіюватися від чорного (0) до білого (255), включаючи відтінки, що містяться в інтервалі $[0, 255]$. Це фактично відповідає кількості відбитого світла. Для 8 біт є 256 рівнів сірого, 40 з яких розпізнаються неозброєним оком. Чим більша кількість бітів, тим більше рівнів і точніше представлення.

Контраст – це помітне протиставлення між двома областями зображення. Контрастне зображення має хорошу динаміку в розподілі значень сірого по всьому діапазону можливих значень, з дуже світлими білими і глибокими чорними. Навпаки, зображення з невеликим контрастом має низьку динаміку, оскільки більшість пікселів мають дуже близькі значення сірого. Якщо L_1 і L_2 є ступенями освітленості відповідно двох сусідніх областей A_1 і A_2 зображення, контраст визначається співвідношенням:

$$C = (L_1 - L_2) / (L_1 + L_2).$$

Яскравість. Це ступінь яскравості точки на зображенні. Вона також визначається як частка інтенсивності світла поверхні на видиму площу цієї поверхні. Для віддаленого спостерігача слово яскравість замінюється словом блиск, що відповідає яскравості об'єкта. При створенні зображень слід враховувати наступні параметри, що хороша яскравість характеризується яскравими (сяючими) зображеннями. Для створення оптимального контрасту

слід уникати зображень, де контрастність має тенденцію до білого або чорного; ці зображення призводять до втрати деталей у темних або світлих областях.

Шум (зашумленість, сторонні артефакти) у зображенні розглядається як явище раптової зміни інтенсивності пікселя по відношенню до його сусідів, воно походить від освітлення оптичних та електронних пристроїв датчика. Це вплив, який представляє певні дефекти (пил, невеликі хмари, миттєве падіння інтенсивності електрики на датчиках тощо). Це призводить до появи невеликих плям, розподіл яких на зображенні випадковий.

Контури представляють межу між об'єктами на зображенні або межу між двома пікселями, рівні сірого кольору яких представляють значну різницю. У цифровому зображенні краї розташовані між пікселями, що належать областям з різною середньою інтенсивністю (це контури типу «стрибок амплітуди»). Контур також може відповідати локальній варіації інтенсивності, що представляє максимум або мінімум (тоді це контур «даху»).

Якщо проводити класифікацію зображень за кількістю кольорів, що використовуються під час його відображення то можна виділити такі три основні їх види (рисунк 1.3):



а) Кольорові



б) Градація сірого



в) Бінарні (чорно-білі)

Рисунок 1.3 – Класифікація цифрових зображень на основі кількості кольорів

Кольорове RGB зображення. Людське око аналізує колір за допомогою трьох типів фотоелементів. Ці клітини чутливі до низьких, середніх або високих частот (червоний, зелений, синій). Щоб представити колір пікселя, необхідно

надати три числа, які відповідають рівням трьох основних кольорів: червоного, зеленого, синього. Таким чином, можемо представити кольорове зображення трьома матрицями, кожна з яких відповідає базовому кольору.

Зображення інтенсивностей (градації сірого) – це матриця, в якій кожен елемент має дійсне значення від 0 (чорний) до 1 (білий). Ми також говоримо про зображення в градаціях сірого, оскільки значення від 0 до 1 представляють різні рівні сірого.

Двійкове зображення – це прямокутна матриця, елемент якої дорівнює 0 або 1. Під час перегляду такого зображення 0 позначаються чорним, а 1 – білим.

Формати зображень поділяються на дві основні категорії: растрові або матричні файли та векторні файли. Кожна категорія має своє призначення.

Векторна графіка використовує точки, криві та градієнти для представлення зображення. Вони найкраще підходять для простих геометричних фігур і ідеальні для логотипів і значків. Вектори можна нескінченно масштабувати без втрати якості, що робить їх ідеальними для екранів високої роздільної здатності та різних розмірів. Проте векторний формат не є найкращим форматом для збереження фотографій. Ось векторні формати файлів: SVG, PDF і PSE.

Растрова графіка використовує значення пікселів у прямокутній сітці для представлення зображення. Вони найкраще підходять для складних зображень, таких як фотографії. На відміну від векторних, ці типи файлів залежать від роздільної здатності та розміру, тобто їх неможливо масштабувати без втрати якості та пікселізації.

Використання різних типів зображень, зокрема растрових і векторних, є основою багатьох процесів в обробці зображень, дизайні та графічній роботі. Кожен з цих форматів має свої особливості, переваги і недоліки, що визначає сферу їхнього застосування. Розуміння різниць між растровими і векторними зображеннями є важливим аспектом у роботі з графічними файлами, адже правильний вибір формату може значно вплинути на ефективність роботи, а також на якість кінцевого результату.

Растрове зображення являє собою масив пікселів, кожен з яких містить інформацію про колір та яскравість. Кожен піксель на зображенні має своє власне значення, і це дозволяє створювати складні градієнти та текстури, що є основною перевагою растрових зображень. Застосування растрових зображень є незамінним при створенні складних і детальних зображень, таких як фотографії, де важливими є дрібні деталі, кольорові переходи та складні текстури. Растрове зображення дає змогу досягти високої реалістичності, адже кожен піксель може містити величезну кількість колірних варіацій, що дозволяє зображати найменші нюанси світла і тіні. Крім того, редагування растрових зображень дозволяє змінювати колір, контраст, а також додавати різноманітні текстурні ефекти, що робить цей формат ідеальним для обробки реалістичних зображень.

Проте растрові зображення мають і суттєві недоліки. Один з головних недоліків – це обмеження масштабу. Якщо растрове зображення збільшувати, пікселі стають чітко видимими, що призводить до втрати різкості та розмитості. Це явище, відоме як пікселізація, обмежує можливості масштабування растрових зображень без втрати якості. Крім того, растрові файли часто займають значний обсяг пам'яті, що може створювати труднощі при роботі з великими зображеннями або при обмежених ресурсах для збереження даних. У випадках, коли необхідно часто змінювати розмір зображення або використовувати його в різних масштабах, растрові зображення не є найкращим вибором, оскільки при масштабуванні вони можуть втратити свою чіткість і точність.

Векторні зображення, на відміну від растрових, складаються з математичних об'єктів, таких як лінії, криві та сегменти, які визначаються за допомогою математичних рівнянь. Завдяки цьому векторні зображення не залежать від пікселів і, отже, можуть бути масштабовані до будь-якого розміру без втрати якості. Це є основною перевагою векторних зображень, оскільки їх можна без обмежень збільшувати чи зменшувати, не боячись втратити чіткість і різкість зображення. Векторні формати ідеально підходять для створення графіки, де важливі точність ліній і геометричні форми, таких як логотипи, схеми, технічні креслення і інші роботи, де потрібна абсолютна точність.

Векторні зображення є значно легшими за розміром порівняно з растровими. Це зумовлено тим, що вони зберігають лише математичні рівняння, що описують елементи зображення, а не інформацію про кожен окремий піксель. Завдяки такій структурі векторні файли не потребують великої пам'яті для збереження, що робить їх ідеальними для використання в середовищах, де необхідно зберігати багато графічних елементів. Однак, незважаючи на численні переваги, векторні зображення мають і свої обмеження. Одним з головних недоліків є відсутність можливості створювати складні текстури і градієнти. Оскільки векторні зображення не працюють з пікселями, вони не можуть відтворювати складні переходи кольорів, тіні та текстури, що може бути обмеженням для певних типів графіки, наприклад, при роботі з фотографіями або іншими зображеннями, де необхідна велика кількість кольорових переходів.

Основною відмінністю між растровими і векторними зображеннями є їх здатність до масштабування. Растрові зображення, будучи побудованими на пікселях, стають розмитими при збільшенні, тоді як векторні зображення залишаються чіткими і чіткими незалежно від розміру. Ця властивість робить векторні зображення дуже корисними в ситуаціях, коли потрібна гнучкість у масштабуванні, наприклад, при створенні брендової графіки, дизайні логотипів або маркетингових матеріалів, які повинні бути придатними для різних розмірів від маленьких елементів до великих банерів. Векторна графіка дає можливість створювати ідеально симетричні та чисті лінії, що важливо для багатьох дизайнерських рішень. Водночас растрові зображення ідеально підходять для завдань, що вимагають високої деталізації, текстур і реалістичних ефектів, де точність відтворення кольору та текстури є основним завданням.

Файли растрових зображень використовують пікселі та називаються статичними. Кожен піксель має свій колір і визначений розмір, і ці зображення залежні від роздільної здатності, яка була обрана під час зйомки. При збільшенні растрового зображення програмне забезпечення розтягує кожен піксель для заповнення прогалів, що призводить до розмитості та спотворення зображення.

Серед найвідоміших форматів растрових зображень можна виділити jpeg, png, gif та tiff.

Формат JPEG (Joint Photographic Experts Group) є стандартом для обміну зображеннями в Інтернеті та друку. Цей формат використовується як цифровими камерами, так і на веб-сайтах завдяки його здатності стискати зображення без значної втрати якості та підтримці багатьма пристроями і програмами. JPEG-файли мають низьку вагу, що робить їх ідеальними для використання в Інтернеті та соціальних мережах. Крім того, формат JPEG підтримується всіма браузерами і дозволяє зберігати зображення високої якості за умови помірного стиснення. Однак варто зазначити, що надмірне стиснення JPEG-файлів може призвести до втрати якості. Також JPEG не підходить для зображень із низькою кількістю кольорових даних, таких як знімки екрану інтерфейсів або проста комп'ютерна графіка. До недоліків цього формату належить і те, що він не підтримує анімацію.

Формат PNG (Portable Network Graphics) є стандартом для зображень в Інтернеті, який підтримує прозорість і стиснення без втрат якості. Цей формат забезпечує високу якість зображень для графіки, але не для фотографій. На відміну від jpeg, png-файли не можна масштабувати без втрати якості. PNG підтримує до 16,7 мільйонів кольорів і є ідеальним для логотипів, інфографіки та банерів. Основною перевагою PNG є можливість збереження всіх деталей зображення навіть після стиснення. Однак цей формат має великий розмір файлу, що робить його менш підходящим для високоякісних веб-файлів та друку.

Формат GIF (Graphics Interchange Format) – це формат для анімованих зображень, який став популярним завдяки ефективному стисненню та швидкому завантаженню файлів. Цей формат підтримує до 256 кольорів та прозорий фон, що робить його ідеальним для невеликих анімацій в Інтернеті та соціальних мережах. Проте gif не підходить для зображень високої якості та фотографій через обмеження кольорової гами. Крім того, gif не підтримує часткову прозорість, що обмежує його використання у деяких випадках. Незважаючи на

ці недоліки, GIF залишається популярним форматом для створення простих анімацій та швидкого обміну зображеннями.

Файл TIFF (Tagged Image File Format) є растровим форматом без втрат, який зазвичай використовується професійними фотографами та дизайнерами. TIFF-файли підтримують стиснення без втрат і забезпечують високу якість зображень для комерційних цілей. Основна відмінність TIFF від інших форматів полягає в тому, що він не стискає зображення, що робить його ідеальним для обробки та створення високоякісних цифрових зображень. Однак формат TIFF має великий розмір файлу, що робить його непридатним для використання на веб-сайтах. TIFF підтримується більшістю браузерів, але через свій великий розмір не є зручним для веб-дизайну.

Растрові зображення залишаються важливим компонентом у цифровому світі, і кожен формат має свої переваги та обмеження залежно від конкретного застосування. Вибір правильного формату зображення залежить від багатьох факторів, таких як якість, розмір файлу, підтримка прозорості та анімації, а також вимоги до кінцевого використання. Jpeg є популярним вибором для фотографій завдяки своєму балансу між якістю та розміром файлу. Png підходить для графіки та логотипів через свою здатність підтримувати прозорість і зберігати високу якість зображення навіть після стиснення. Gif є ідеальним для створення простих анімацій і швидкого обміну невеликими файлами, але не підходить для фотографій та зображень із широкою кольоровою гамою. Tiff є найкращим вибором для професійної обробки зображень та друку, забезпечуючи високу якість без втрат, але його великий розмір файлу обмежує його використання у веб-дизайні.

Моделювання зображень за допомогою растрових форматів є важливим аспектом у галузях цифрової фотографії, веб-дизайну та графічного мистецтва. Розуміння характеристик та обмежень кожного формату дозволяє вибрати найбільш підходящий для конкретних завдань та забезпечувати високу якість кінцевого продукту. Використання растрових зображень залишається ключовою технологією у сучасному цифровому світі, і правильний вибір формату є

критично важливим для успішного проекту. Від фотографій і графіки до анімацій та високоякісного друку – растрові зображення продовжують відігравати важливу роль у створенні візуального контенту, забезпечуючи баланс між якістю, розміром файлу та функціональністю.

1.2 Основні визначення та поняття теорії контурного аналізу

Опис зображень є важливою задачею в області комп'ютерного зору та обробки цифрових зображень. Одним із найважливіших етапів цієї задачі є дослідження контурів об'єктів, які дозволяють виділити суттєві структурні особливості зображення. Контури, як лінії або межі, що відокремлюють різні частини зображення, є важливими ознаками для розпізнавання та класифікації об'єктів. Завдяки детекції контурів можна отримати точнішу та інформативнішу репрезентацію зображення, що дозволяє автоматизувати ряд процесів, пов'язаних з обробкою візуальної інформації.

Виявлення країв у цифрових зображеннях є нетривіальною операцією, оскільки існує багато детекторів, і кожен детектор має різні характеристики відповідно до характеристик зображення (шум, розмитість, однорідність, рівні деталізації та ентропії) і значень параметрів вибраних детекторів.

Контури об'єктів на зображеннях є лініями, що позначають межі різких змін інтенсивності пікселів. У більшості випадків, ці зміни вказують на перехід між різними об'єктами, що є особливо важливим при аналізі складних зображень, де важливо не тільки визначити наявність об'єктів, але й зрозуміти їхні взаємні розташування та взаємодії. Дослідження контурів дає змогу розв'язати важливі завдання в обробці зображень, зокрема в сегментації, розпізнаванні об'єктів, а також у роботах, пов'язаних із реконструкцією тривимірних сцен або оцінкою геометрії об'єктів.

Ідея опису зображень через аналіз контурів ґрунтується на припущенні, що межі об'єктів на зображеннях відповідають значним змінам у світлових характеристиках або кольорових відтінках, що дозволяє виділяти важливі структурні елементи. Процес опису зображення через контури передбачає не лише виявлення самих меж, але й їх класифікацію, а також визначення геометричних характеристик кожного об'єкта. Наприклад, контур може бути використаний для визначення форми, розміру та орієнтації об'єкта, що є критично важливим для багатьох додатків, таких як автоматичне розпізнавання об'єктів або візуальне відстеження.

Виявлення країв у зображенні є важливою функцією процесу пошуку зображень на основі вмісту. Зіткнувшись із великою кількістю методів виявлення країв і фільтрів, може бути важко вибрати підхід, який найбільше підходить для конкретної колекції зображень, особливо тому, що не існує оптимальної методики для всіх випадків. Крім того, методи визначення країв часто аналізують зображення в цілому, не беручи до уваги особливості компонентів зображення. Щоб виконати цю операцію виявлення країв і виправити цей стан речей, корисно спочатку вивчити набір методів і проаналізувати їх ефективність і адаптованість відповідно до характеристик зображень в цілому (зображення синтетичне або реальне, нечітке або чітке, шумно) і відповідно до їх компонентів (високочастотні проти низькочастотних зон). Потім буде запропоновано новий підхід до визначення краю та порівняно з набором вибраних детекторів. Нарешті, результати аналізу будуть представлені у вигляді правил вибору евристичних детекторів. Контурне виділення використовується в більшості випадків для полегшення вивчення деталей зображення. Найпростіше застосування – підкреслити контури, коли вони надто розмиті. Однак використання цих методів також використовується для пошуку зображень на основі вмісту, для визначення місцезнаходження певних об'єктів на зображеннях або навіть для реконструкції об'єктів у трьох вимірах. Для цього типу застосування важливо виділити контури присутніх об'єктів, щоб мати можливість їх розпізнати або реконструювати. Тому виділення контурів часто є

важливою попередньою обробкою зображення перед виконанням іншої обробки. Коли ми хочемо виділити контури, ми швидко помічаємо, що кількість існуючих методів дуже велика і що кожен детектор має свої особливості. Це тим більше обмеження, що продуктивність цих детекторів змінюється залежно від характеристик контурів зображення та параметрів, пов'язаних із детекторами. Тому ми повинні спробувати визначити найбільш відповідний детектор для даного зображення. Для цього необхідно визначити характеристики шуму, різкості або навіть безперервності контурів. Параметрів так багато, що визначення неозброєним оком може здатися суб'єктивним.

На зображенні контур можна розглядати різними способами. Для прикладу, проаналізуємо три основні способи (рисунок 1.4):

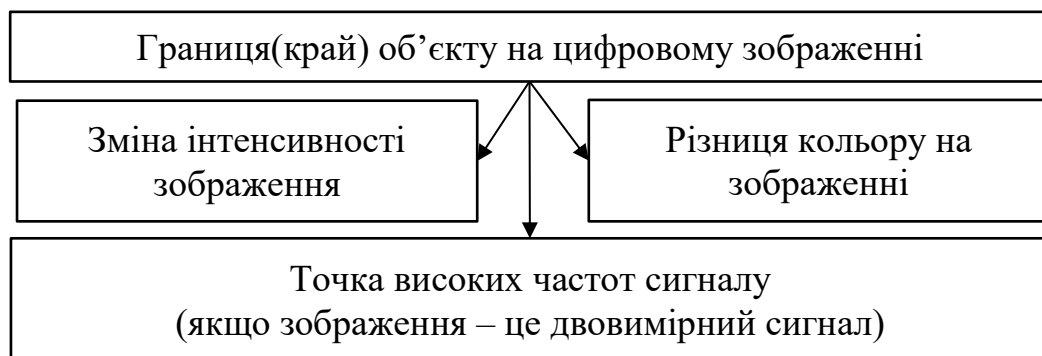


Рисунок 1.4 – Класифікація підходів виявлення країв

Границю можна розглядати як різку зміну інтенсивності зображення. Існує декілька типів варіацій. На рисунку 1.5 наведено різні профілі контурів: сходи́нка, пандус, дах, вершина.



Рисунок 1.5 – Приклади профілів границь об'єктів (сходи́нка, пандус, дах, вершина)

По-друге, спосіб, дуже близький до попереднього, але полягає в тому, щоб розглядати контури як різницю в кольорі. Ще одним варіантом для визначення границі може бути наступне. Якщо розглядати зображення як двовимірний сигнал, то можемо перейти в частотну область (наприклад, за допомогою Фур'є або вейвлет-перетворення). У цьому випадку край можна розглядати як точку високих частот сигналу. Однак цих уявлень інколи недостатньо для представлення контуру. Наприклад, іноді людський мозок здатний ідентифікувати невидимі контури на зображенні, оскільки він має здатність використовувати свої знання для вилучення форм із зображення.

Основним методом виявлення контурів є аналіз зміни градієнта інтенсивності пікселів. Один із найбільш популярних алгоритмів для цієї мети – алгоритм Канні, що виявляє контури шляхом визначення точок, де інтенсивність пікселів змінюється різко. Цей метод дозволяє отримати контури, які чітко відображають межі об'єктів, зокрема у випадках, коли зображення містить різні шари текстур або має складну структуру. Важливим аспектом є використання фільтрів для згладжування зображення перед виявленням контурів, що дозволяє зменшити вплив шумів і підвищити точність результатів.

Опис зображень на основі контурів може також включати такі етапи, як сегментація, де зображення поділяється на окремі області, кожна з яких відповідає певному об'єкту або частині об'єкту. Для цього можуть використовуватися методи, засновані на контурних ознаках, які дозволяють точно визначити межі об'єктів і поділити зображення на частини для подальшої обробки. Наприклад, у медичній діагностиці аналіз контурів може бути використаний для виділення об'єктів на зображеннях органів, що дозволяє точно оцінити їх стан або виявити патологічні зміни.

Крім того, опис зображень на основі контурів може включати в себе більш складні алгоритми, які поєднують контурний аналіз з іншими техніками, такими як виявлення текстур або кольорових характеристик. Зокрема, в випадках, коли контури можуть бути нечіткими через низьку якість зображення або складні освітлювальні умови, для підвищення точності опису використовуються методи,

які комбінують контури з іншими ознаками, наприклад, з характеристиками текстури або глибини.

Для отримання більш якісних описів при визначенні контурів слід виконувати ряд правил, серед яких:

- контури не повинні бути надто товстими, бажано, щоб товщина дорівнювала розміру пікселя;
- контури повинні мати певну безперервність. Наприклад, мінімальний розмір правильного контуру має становити два пікселі, тоді як контур розміром в один піксель представляє шум у зображенні.
- зв'язок між контурами має бути досить важливим. Припускаємо, що 90% контурів з'єднані з іншим контуром і лише 10% мають кінці, які завершують контур.
- проста класифікація контурів. Контури повинні бути легко ідентифіковані відносно один одного, два різних контури не повинні стикатися паралельно по всій довжині. Таким чином, два контури повинні бути розділені принаймні одним пікселем.

Важливим аспектом опису зображень на основі контурів є також питання масштабування та адаптації алгоритмів до різних типів зображень. Оскільки контури є функцією інтенсивності пікселів, що залежить від роздільної здатності зображення та його контексту, важливо мати змогу коригувати алгоритми детекції контурів для різних умов. Наприклад, на зображеннях з низькою роздільною здатністю або на зображеннях, що містять багато шумів, алгоритми виявлення контурів можуть потребувати додаткових етапів попередньої обробки або використання спеціальних фільтрів для поліпшення результатів.

Один із найбільш цікавих аспектів опису зображень за допомогою контурного аналізу полягає в його потенціалі для подальшого використання в системах штучного інтелекту та машинного навчання. Зокрема, сучасні методи глибокого навчання використовують контурний аналіз як одну з ключових ознак для автоматичної класифікації та розпізнавання об'єктів. Застосування нейронних мереж для виявлення та опису контурів дозволяє покращити точність

і швидкість обробки зображень, відкриваючи нові можливості для створення складних систем комп'ютерного зору, які здатні не тільки розпізнавати контури, але й робити глибший аналіз зображень на основі виявлених структурних елементів.

Таким чином, опис зображень на основі дослідження контурів є важливою і складною задачею в обробці цифрових зображень. Використання контурного аналізу дозволяє отримати точнішу репрезентацію об'єктів, що зображені на зображеннях, і надає можливість для подальших кроків у їх класифікації, розпізнаванні та інтерпретації. З розвитком алгоритмів і технологій, контурний аналіз продовжує розвиватися, надаючи нові можливості для покращення ефективності та точності обробки зображень у різних галузях, від медицини до промисловості та робототехніки.

1.3 Порівняльний аналіз програм обробки цифрових зображень

Раніше призначене для професіоналів програмне забезпечення для редагування цифрових зображень останнім часом стало більш популярним. Сьогодні велика кількість використовують цифрові інструменти, щоб змінити зовнішній вигляд своїх цифрових зображень відповідно до своїх вимог. Для визначення основних функціональних можливостей програмних систем проведено аналітичний огляд систем які широко представлені на сучасному ринку програмних засобів.

Adobe Photoshop Lightroom – це потужне та універсальне програмне забезпечення для редагування фотографій, яке є частиною набору Adobe Creative Cloud. Він пропонує багато розширених функцій для професійних фотографів і фотографів-аматорів. Серед цих функцій знаходимо інтегрований менеджер фотографій, який полегшує керування та впорядкування ваших файлів. Це полегшує пошук і навігацію великою колекцією фотографій, що важливо для

фотографів, які працюють над багатьма різними проектами. Крім того, Lightroom підтримує файли RAW, що означає чудову якість зображення та більшу гнучкість редагування порівняно зі звичайними файлами JPEG. Окремо слід виділити можливість створювати та використовувати попередні налаштування. Попередні налаштування – це попередні налаштування, які дозволяють швидко застосувати попередньо визначені ефекти та налаштування до фотографії. Це може бути дуже корисним для економії часу при обробці великої кількості схожих фотографій. Наприклад, якщо потрібно застосувати той самий фільтр або колірну корекцію до кількох фотографій, то можна просто застосувати відповідний стиль замість редагування кожної фотографії окремо. Окрім цих основних функцій, Lightroom також пропонує розширені інструменти для редагування фотографій. У ньому є інструменти для виправлення дефектів, наприклад Spot Removal Tool, який допомагає усунути недоліки на фотографії, а також інструменти для корекції рівня кольору та яскравості. Lightroom також пропонує низку локальних інструментів редагування, які дозволяють робити точні налаштування окремих частин фотографії, наприклад видаляти непотрібні об'єкти або додавати спеціальні ефекти. Однак основним недоліком Lightroom є його бізнес-модель, оскільки це платне програмне забезпечення з місячною або річною підпискою. Він сумісний з Windows, macOS, iOS і Android. Зауважте, що якщо ви дійсно прихильник продуктів Adobe, існує безкоштовна програма для редагування фотографій - Adobe Photoshop Express. Функціональність редактора, очевидно, більш обмежена, але для початківців, які хочуть почати роботу з обробкою та редагуванням зображень, інструмент ідеально підходить.

PhotoWorks – це програмне забезпечення для редагування фотографій, призначене для користувачів, які хочуть здійснити поступовий перехід між спрощеними програмами на смартфонах і професійними рішеннями, такими як Photoshop. PhotoWorks насамперед прагне бути простим у використанні програмним забезпеченням для редагування фотографій. Дійсно, спосіб, у який він був розроблений, не залишає місця для сумнівів, цільовою метою є користувачі початківців або середнього рівня. Доказом цього є те, що

PhotoWorks пропонує певну кількість автоматичних та інтелектуальних удосконалень, які вказують користувачеві шляхи покращення своєї роботи, а також численні відеоуроки французькою мовою (досить рідко, щоб це було підкреслено). Також відзначимо наявність пресетів для рівнів і кривих, а також допомогу в розрізанні за допомогою застосування відомих сіток (правило однієї третини, золотий переріз тощо). Тому користувачі не залишаються напризволяще, коли починають використовувати програмне забезпечення PhotoWorks. Що стосується інтерфейсу, то він зрозумілий і досить швидкий в освоєнні. Зображення займає 2/3 лівої частини екрана, а основні доступні меню доступні клацанням трохи вище. Вміст кожного меню розкривається в правій частині, як і більшість професійних програм для редагування. З точки зору функціональності програмне забезпечення PhotoWorks пропонує багато можливостей, але залишається менш повним, ніж Photoshop. Під час редагування фотографій PhotoWorks допомагає покращити якість зображення, змінюючи експозицію та баланс білого, виправляючи розмиття та дефекти перспективи, а потім усуваючи спотворення. Програмне забезпечення, очевидно, має багато варіантів для зміни кольорів на ваш смак і налаштування відтінку, насиченості, різкості та яскравості. Є всі основні функції професійного програмного забезпечення для редагування фотографій. Крім того, PhotoWorks надає прості, але інтелектуальні інструменти (інтелектуальний коригувальний пензель і штамп клонування), які спрямовані на чисте видалення об'єктів із зображення кількома клацаннями миші. Вирізка значною мірою автоматизована, і користувачеві доведеться втручатися лише в деталі контуру об'єкта, якщо це необхідно. Портрети не забуті, адже можна змінити текстуру шкіри, відбілити зуби, змінити колір очей завдяки ряду «чарівних пензлів», які нагадують нам, що PhotoWorks — це проста у використанні програма для редагування фотографій.

Нарешті, програмне забезпечення цілком логічно включає велику кількість фільтрів і ефектів, щоб надати більшого характеру вашій роботі: чорно-білий, HDR, легенди, водяні знаки тощо. Список довгий, і PhotoWorks пропонує як

«класичні» ефекти для досвідчених аматорів або професіоналів, так і інші, більш легкі, які сподобаються новачкам.

Darktable – це надійне програмне забезпечення, яке пропонує широкий спектр функцій для розробки фотографій. Він розроблений, щоб імітувати функціональність традиційного світлового столу та темної кімнати, надаючи фотографам повний набір інструментів для покращення своїх зображень. Однією з головних особливостей darktable є його можливість неруйнівного редагування. Це означає, що користувачі можуть редагувати свої фотографії, не змінюючи вихідний файл зображення, що забезпечує більшу гнучкість у постобробці. Програмне забезпечення також пропонує різноманітні інструменти для корекції кольору, налаштування експозиції та зменшення шуму, що дає фотографам можливість точно налаштувати свої зображення до досконалості. Ще одна примітна особливість darktable – це організація робочого процесу та система керування. Програмне забезпечення надає користувачам віртуальний світловий стіл, де вони можуть легко переглядати та сортувати свої фотографії. Це дозволяє фотографам швидко переміщатися по своїй бібліотеці зображень і вибирати найкращі знімки для редагування. Darktable також пропонує потужні параметри пошуку та фільтрації, що полегшує пошук певних зображень у великій колекції. На додаток до функцій редагування та організації, darktable також надає ряд інструментів для розширеної обробки зображень (темна кімната).

RawTherapee також є одним із еталонних програм для редагування фотографій завдяки функціям проявлення фотографій. Завдяки передовим інструментам і функціям він дозволяє фотографам покращувати та редагувати свої зображення з точністю та контролем. Однією з найвизначніших функцій RawTherapee є менеджер фотографій, який забезпечує зручний спосіб перегляду та впорядкування фотографій. Це дозволяє користувачам легко переглядати свою колекцію зображень і вибирати ті, над якими вони хочуть працювати. Ця функція полегшує фотографам порівняння різних знімків, вибір і спрощення робочого процесу. Іншою визначною особливістю RawTherapee є повний набір інструментів редагування. Він пропонує широкий спектр налаштувань і

покращень, які можна застосувати до фотографій, дозволяючи фотографам досягти бажаного вигляду своїх зображень. Від базових налаштувань, таких як експозиція, контраст і баланс білого, до більш розширених функцій, таких як шумозаглушення та підвищення різкості, RawTherapee надає фотографам інструменти, необхідні для точного налаштування своїх зображень і підкреслення їх. Окрім можливостей редагування, RawTherapee також пропонує неруйнівний робочий процес. Це означає, що будь-які зміни, внесені до зображення, зберігаються окремо від оригінального файлу, гарантуючи збереження вихідних даних.

ON1 Photo RAW – це комплексне рішення, яке пропонує широкий спектр функцій для покращення фотографій і керування ними. Однією з його головних особливостей є надійна система каталогізації, яка дозволяє користувачам легко впорядковувати та шукати свої колекції зображень. Завдяки можливості створювати спеціальні ключові слова та теги, користувачі можуть швидко знаходити конкретні зображення на основі теми, розташування чи будь-яких інших критеріїв, які вони вибирають. Ця функція каталогізації особливо корисна для професійних фотографів або тих, хто має велику бібліотеку фотографій, заощаджуючи час і гарантуючи, що жодне зображення не буде втрачено в цифровій безодні. Ще одним вражаючим аспектом ON1 Photo RAW є використання технології штучного інтелекту (AI). Ця система на основі ШІ здатна автоматично аналізувати та класифікувати зображення на основі їхнього вмісту. Використовуючи алгоритми машинного навчання, програмне забезпечення може розпізнавати об'єкти, обличчя та пейзажі, дозволяючи користувачам легко знаходити та сортувати свої фотографії. Ця функція AI особливо корисна для фотографів, які часто знімають різноманітні об'єкти та потребують швидкого доступу до певних зображень. На додаток до можливостей каталогізації та штучного інтелекту, ON1 Photo RAW також пропонує широкий спектр фільтрів і ефектів для покращення візуальної якості фотографій. Програмне забезпечення надає повний набір інструментів для редагування та покращення зображень, починаючи з базових налаштувань, таких як експозиція

та контраст, і закінчуючи більш просунутими ефектами, такими як емуляція вінтажної плівки чи креативне градування кольорів. Ці фільтри та ефекти можна повністю налаштувати, що дозволяє користувачам точно досягти бажаного вигляду.

Pixlr X програмне забезпечення, призначене для початківців. Його інтерфейс англійською мовою, але його можна вивчити за кілька хвилин. Pixlr Він дозволяє змінювати розмір фотографії, повертати її, інвертувати, вибирати частини, які потрібно зберегти або видалити, регулювати насиченість, контраст, яскравість, експозицію та грати з тінями. Він також пропонує налаштувати фільтри щодо різкості, розмиття, зернистості, згладжування або застосувати такі ефекти, як пікселізація чи мозаїка. Він також пропонує можливість вибору ефекту з попередньо визначеної палітри (ретро, кольори, відтінки тощо), додавання тексту чи малюнків, вибору рамки чи прикріплення наклейки.

Pixlr також пропонує версію Pro, яка надає доступ до численних шаблонів PSD і редактора з додатковими розширеними функціями. Pixlr також доступний на мобільних пристроях Android та iOS.

Проведений огляд програмного забезпечення дозволив виділити основні функції, що притаманні програмним системам обробки цифрових зображень та надають максимальний функціонал для користувачів з різним рівнем досвіду роботи.

1.4 Постановка задач дослідження

Перший розділ кваліфікаційної роботи був присвячений аналізу технологій створення кодування, обробки та зберігання цифрових зображень. Було проаналізовано основні характеристики цифрових зображень та сфери їх використання. Проведений аналіз технологій теорії контурного аналізу з метою визначити основні задачі, що можуть бути вирішені з її використанням. Окрім

того було досліджено ряд програмних систем, що використовуються для обробки цифрових растрових зображень, що дозволило проаналізувати їх основні функціональні та архітектурні особливості. В результаті проведеного аналізу для успішного вирішення поставлених завдань було виділено наступні задачі:

- проаналізувати та провести класифікацію цифрових зображень;
- дослідити методи та алгоритми теорії контурного аналізу;
- провести аналітичний огляд цифрових бібліотек та програмних засобів оброблення цифрових зображень;
- проаналізувати алгоритми виділення та опису цифрових зображень;
- розробити алгоритм опису об'єктів цифрових зображень на основі аналізу контурної лінії;
- реалізувати програмний додаток виділення та опису об'єктів на цифрових зображеннях, провести його тестування та порівняння з аналогами.

1.5 Висновки до розділу

Проведено аналіз та здійснено класифікацію цифрових растрових зображень на основі дослідження їх характеристик, що дозволило обрати цифрових зображень для програмної реалізації системи обробки зображень та провести класифікацію цифрових зображень.

Досліджено методи та алгоритми теорії контурного аналізу на основі їх функціональних можливостей виділення границь на цифрових зображеннях, що дозволило вибрати набір фільтрів та алгоритмів кодування для опису об'єктів на цифрових зображеннях.

Проведено аналітичний огляд програмних систем оброблення цифрових зображень на основі аналізу їх функціональних можливостей та архітектурних рішень, що дозволило виділити основні блоки для систем обробки цифрових зображень.

2 МЕТОДИ ТА АЛГОРИТМИ ВИДІЛЕННЯ ТА ОПИСУ ЕЛЕМЕНТІВ ЦИФРОВИХ ЗОБРАЖЕНЬ

2.1 Алгоритми аналізу цифрових зображень

Аналіз зображення – це просто розпізнавання та пояснення змісту самого зображення. Аналіз виконується точно та з врахуванням найменших деталей, тому що зображення містить багато інформації. Важливо мати достатні знання та відповідну кваліфікацію, щоб мати можливість пояснити, проаналізувати та точно інтерпретувати зображення. Щоб зрозуміти, для чого використовується аналіз зображень, цікаво зрозуміти, навіщо проходити цей аналіз. Кожне зображення, незалежно від його типу, має своє повідомлення. Це повідомлення може бути різних типів і може містити повідомлення, переконати громадськість, підкреслити елемент, переконати, критикувати тощо. Аналіз зображення дозволяє розшифрувати повідомлення, яке воно надсилає, і зрозуміти, які елементи та які методи вибрано для передачі бажаного повідомлення.

Аналіз зображення – це тривала робота, яка вимагає певної організації. Виділення великої кількості інформативних ознак приведе до збільшення часу опрацювання цифрового зображення, що може негативно повпливати на підсумковий результат. Навпаки ігнорування – може призвести до некоректного опису сцени чи її елементу. Це важливо для того, щоб розробити аналіз чітко та стисло. Аналіз зображення можна розділити на три частини (рисунок 2.1):

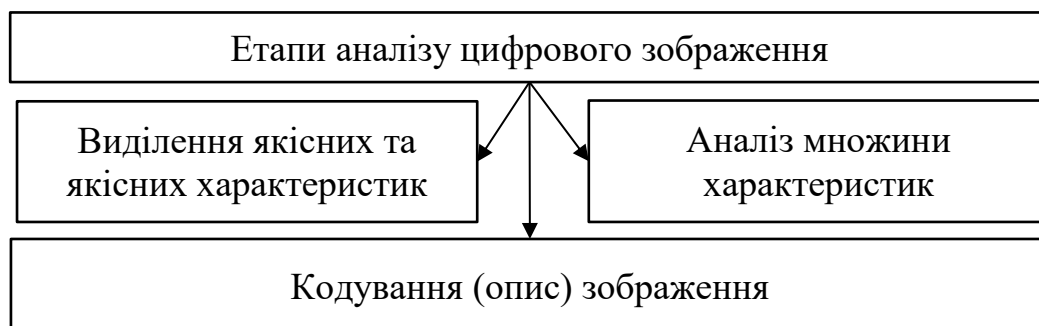


Рисунок 2.1 – Переваги використання технології інтелектуалізованого

Ці три етапи складають інтегральний аналіз зображення. Деякі з цих кроків самі по собі включають різні кроки, які слід виконати, щоб розробити насичений і повний аналіз.

Для більшого розуміння поняття аналізу зображень необхідно виділити характеристики які можуть бути визначені під час обробки та аналізу цифрового зображення. Наприклад, якщо почати аналіз зображення на якому розміщений деякий об'єкт (рисунок 2.2), то одразу можна виділити набір ознак які визначають деяку оцінку вхідному зображенню.



Рисунок 2.2 – Цифрові зображення з об'єктом «Ейфелева вежа»

Як видно з наведеного прикладу, хоча всі зображення містять об'єкт одного типу, проте всі вони кардинально відрізняють один від одного, багато елементів можуть відрізнитися від одного зображення до іншого. Проте враховуючи те, що цифрові зображення це лише відображення деякого об'єкту, то можна вважати, що, два зображення одного класу, тобто ті, що представляють одне й те саме, пов'язані трансформацією. Однак точна характеристика цього перетворення нам невідома.

Визначення типу трансформації, яка дозволяє переходити від одного зображення до іншого це класична проблема комп'ютерного зору, яка називається відповідністю зображень. На сьогоднішній день відома велика кількість програмних додатків, які містять функції для створення панорам,

виявлення зображень, схожих на інші (візуальний пошук), або навіть розпізнавання об'єктів на зображенні (розпізнавання об'єктів).

У контексті комп'ютерного зору та обробки зображень важливими аспектами є різноманітні фактори, які можуть впливати на результат аналізу. Кожен з цих факторів має своє значення і може мати вплив на точність та ефективність обробки зображень. При аналізі цифрового зображення можна виділити ряд характеристик які дозволяють визначати тип трансформації яка була застосована до сцени під час фотографування (рисунок 2.3).

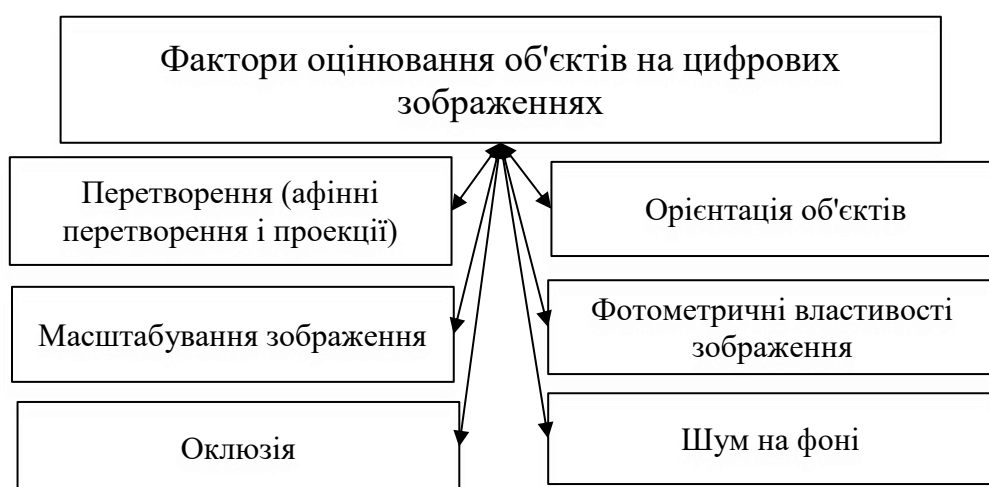


Рисунок 2.3 – Класифікація факторів, які впливають на аналізі цифрових зображень

Перетворення (афінні перетворення і проекції) є базовими математичними операціями, які використовуються для зміни координатної системи зображення. Афінне перетворення включає в себе такі операції, як масштабування, повороти, зсуви та інші лінійні перетворення, які можуть зберігати прямолінійність і паралельність елементів на зображенні. Це може бути важливим для відновлення об'єктів з різних перспектив чи зміщених координат, наприклад, при відновленні зображень із зміненою орієнтацією або при перетворенні одних зображень у інші для порівняння. Результати афінних перетворень або проекцій повинні зберігати

достатній рівень точності, щоб забезпечити коректне подальше використання зображення, зокрема у задачах реконструкції або порівняння.

Орієнтація об'єктів на зображенні також є важливим аспектом для обробки зображень. Поворот зображення або його частини може змінити розташування об'єктів на ньому. Врахування орієнтації особливо важливе в таких задачах, як автоматичне розпізнавання об'єктів, де об'єкти можуть з'являтися під різними кутами. Наприклад, у випадку зі штучним інтелектом та системами комп'ютерного зору необхідно, щоб алгоритми могли враховувати повороти та правильно класифікувати об'єкти, незважаючи на їх різну орієнтацію. Цей аспект є критичним при обробці зображень, де об'єкти можуть змінювати своє положення у просторі, наприклад, під час обертання камери чи зміни кута огляду.

Масштабування зображення – це важлива операція, що дозволяє змінювати розмір об'єктів на зображенні. Збільшення масштабу зазвичай використовується для виділення деталей або уточнення зображення. Водночас важливо зберігати пропорції та якість зображення при зміні його розміру, що дозволяє уникнути спотворень. Масштаб також може бути використаний для аналізу об'єктів на різних рівнях деталізації або для зменшення обсягу даних в задачах з обмеженими ресурсами. Однак надмірне збільшення масштабу може призвести до втрати інформації або появи артефактів, таких як пікселізація, що може негативно вплинути на подальшу обробку зображення.

Фотометричні властивості зображення, такі як яскравість і контраст, суттєво впливають на сприйняття зображення людиною та ефективність його обробки. Зміни в освітленні або варіації через погодні умови, час доби або використання спалахів можуть призвести до значних змін у вигляді зображення. Зокрема, зміни контрасту можуть приховати важливі деталі або створити ефект затемнення, що ускладнює точне розпізнавання об'єктів. Наприклад, у нічних умовах або під час дощу, зображення може стати більш розмитим, що впливає на точність детекції контурів і на подальші етапи обробки. Тому коригування

фотометричних властивостей є важливим етапом для підвищення якості зображень у таких умовах.

Оклюдія – це явище, коли частина об'єкта на зображенні стає прихованою або перекритою іншим об'єктом. Це може бути результатом взаємодії об'єктів у реальному світі або ж результатом обмежень камери. Оклюдія є однією з основних проблем в комп'ютерному зорі, оскільки часто не можна побачити всю необхідну інформацію про об'єкт через те, що частина його перекрита іншим об'єктом або елементом на зображенні. Це особливо важливо в таких задачах, як розпізнавання об'єктів або відстеження, де неповне зображення може призвести до помилок у класифікації або інтерпретації даних. Рішення таких проблем потребує застосування алгоритмів, які можуть відновлювати інформацію, що була частково втрачена через оклюдію, або, принаймні, коректно враховувати її на всіх етапах обробки.

Шум на фоні є ще однією суттєвою проблемою для обробки зображень. Коли частини зображення зливаються з елементами заднього плану, це може ускладнити відокремлення об'єктів або правильне їх розпізнавання. Такий ефект може бути викликаний схожістю кольору або текстури об'єктів на передньому та задньому плані. У таких випадках потрібна додаткова обробка зображення для відокремлення об'єктів від фону, що зазвичай включає в себе такі етапи, як сегментація, фільтрація або корекція контрасту. Шум на фоні часто зустрічається в природних або міських пейзажах, де складна текстура фону може значно вплинути на точність аналізу зображення.

Таким чином, кожен з цих факторів має суттєвий вплив на процес обробки цифрових зображень. Рішення проблем, що виникають внаслідок афінних перетворень, поворотів, масштабування, зміни фотометричних властивостей, оклюдій та безладу на фоні, є ключовими для досягнення високої точності та ефективності в задачах комп'ютерного зору та автоматичної обробки зображень.

Обробка зображень є однією з найважливіших інженерних дисциплін, що займається аналізом і обробкою цифрових зображень для покращення їх якості або отримання важливої інформації. З розвитком технологій комп'ютерного

зору, штучного інтелекту та машинного навчання, обробка зображень стала надзвичайно важливою у багатьох сферах, включаючи медицину, автоматичне управління, відеоспостереження, промисловість, а також у розважальній індустрії та цифровому дизайні. Основною метою цієї дисципліни є перетворення та інтерпретація візуальних даних для різноманітних цілей, що варіюються від поліпшення якості зображення до витягання важливої інформації або автоматичного розпізнавання об'єктів на зображеннях. Завдяки застосуванню інноваційних підходів і методів, обробка зображень дозволяє отримати точні результати навіть у складних і великих наборах даних, що не могли б бути оброблені вручну.

Етапи обробки зображень є основою процесу роботи з цифровими зображеннями. Першим етапом є отримання зображення, яке може бути здійснене різними способами, наприклад, за допомогою сканерів, фотокамер, сенсорів або інші пристрої, здатні створювати цифрові зображення. Отримане зображення часто має певний рівень шуму або спотворення, тому наступним етапом є попередня обробка. На цьому етапі використовуються різноманітні методи фільтрації, покликані усунути небажані елементи, такі як шум, дефекти або інші спотворення, які можуть знижувати якість зображення. Такі методи включають фільтрацію за допомогою масок або спеціальних алгоритмів для корекції контрасту, яскравості, різкості або інших властивостей зображення.

Наступним етапом є сегментація зображення, яка полягає в поділі зображення на окремі області, що мають спільні характеристики. Це дозволяє виділити важливі елементи зображення, такі як контури об'єктів або інші значущі частини зображення, що необхідні для подальшого аналізу. Сегментація є важливим кроком, оскільки вона забезпечує точне виділення об'єктів або інформації, яку необхідно проаналізувати. Після сегментації наступним етапом є представлення та опис зображення. На цьому етапі застосовуються методи, що дозволяють структурувати інформацію з отриманих сегментів для подальшої інтерпретації. Використовуються різні методи представлення зображень у вигляді математичних моделей, таких як гістограми, вектори, матриці, що

дозволяють детально описати характеристики зображення. Опис зображення може включати інформацію про його геометрію, текстуру, колір, форми, розміри та інші ознаки, які можуть бути використані для подальшого аналізу або розпізнавання. Останнім етапом є розпізнавання, що включає ідентифікацію та класифікацію об'єктів на зображенні. Алгоритми розпізнавання зображень використовують отриману інформацію про сегменти та їхні характеристики для визначення, до якого класу належить об'єкт на зображенні. Це може включати як просте порівняння з попередньо відомими шаблонами, так і складніші методи, такі як глибоке навчання, де нейронні мережі використовуються для автоматичного розпізнавання та класифікації об'єктів на зображенні. Розпізнавання зображень є основою для багатьох сучасних технологій, від систем відеоспостереження до автономних транспортних засобів.

Методи обробки зображень мають велике значення для покращення якості зображень, що отримуються на різних етапах. Один з таких методів – покращення зображення, яке може включати різноманітні техніки для підвищення контрасту, видалення шуму або зменшення розмитості. Ці методи дозволяють поліпшити якість зображення, роблячи його більш чітким і деталізованим, що особливо важливо в таких сферах, як медицина, де чіткість зображень є критично важливою для правильної діагностики. Іншим важливим методом є просторове фільтрування. Просторова фільтрація використовується для обробки пікселів зображення за допомогою різноманітних фільтрів, що дозволяють видаляти або послаблювати деякі частини зображення, зберігаючи важливі деталі. Це може включати фільтрацію для зменшення шуму, розмиття або покращення контрасту. Просторові фільтри можуть бути реалізовані за допомогою математичних операцій, таких як згортка або маски, що застосовуються до зображення. Перетворення частотної області є ще одним важливим методом обробки зображень. В даному контексті цей метод полягає в перетворенні зображення з просторової області в частотну. Частотна область дозволяє аналізувати і змінювати зображення в термінах його частотних компонентів, що особливо корисно для покращення таких характеристик, як

різкість, контраст та виявлення деталей. Частотне перетворення часто використовується для фільтрації, де низькочастотні компоненти можуть бути видалені для згладжування зображення або високочастотні компоненти для підвищення різкості. Стиснення зображень є ще одним важливим методом, який використовує алгоритми для зменшення обсягу зображень без значної втрати їхньої якості. Це особливо актуально в умовах обмежених ресурсів зберігання даних або в ситуаціях, коли зображення необхідно передавати через мережу з обмеженою пропускнуою здатністю. Алгоритми стиснення, такі як JPEG або PNG, дозволяють значно зменшити розмір файлів, зберігаючи при цьому прийнятний рівень якості. Техніка обробки зображень поєднує в собі різні аспекти інформатики, математики та електроніки для досягнення ефективної візуальної обробки даних. Комп'ютерне зображення обробляється за допомогою спеціальних алгоритмів та математичних моделей, що дозволяє автоматично обробляти великі обсяги даних, витягувати важливу інформацію та здійснювати складні операції з обробки зображень, таких як фільтрація, трансформація або розпізнавання.

Алгоритми, що використовуються в обробці зображень, є серцем цієї дисципліни. Фільтрація, сегментація і визначення країв є одними з основних алгоритмів, які допомагають виявляти важливі елементи зображення та покращувати його якість. Фільтрація дозволяє змінювати пікселі зображення за допомогою спеціальних фільтрів для видалення шуму або покращення контрасту. Сегментація дозволяє поділити зображення на окремі частини, що мають спільні властивості, що є важливим кроком для подальшого аналізу та розпізнавання. Визначення країв є методом, який допомагає виявляти контури об'єктів, що важливо для подальшої ідентифікації та класифікації об'єктів на зображенні.

Обробка зображень включає численні етапи, від отримання зображення до його розпізнавання та класифікації, і використовує широкий спектр алгоритмів, що дозволяють поліпшувати якість зображень та отримувати важливу інформацію для різних застосувань.

2.2 Алгоритми виділення контурної лінії

Виявлення контурів є однією з найважливіших задач у галузі комп'ютерного зору, а градієнт грає в цьому процесі ключову роль. Контури на зображеннях визначають межі між різними об'єктами або між об'єктами та фоном, а також вказують на зміни інтенсивності кольору чи текстури. Використання градієнта дозволяє виявляти ці межі, оскільки саме в таких місцях відбуваються різкі зміни інтенсивності пікселів. Основна ідея виявлення контурів за допомогою градієнта полягає в тому, що він дозволяє знаходити зони, де ці зміни є найбільшими, і таким чином виокремлювати об'єкти або їх частини на зображенні.

Градієнт є одним із фундаментальних понять в обробці зображень та комп'ютерному зорі, що визначає напрямок і величину зміни функції інтенсивності пікселів зображення. У простих термінах, градієнт є вектором, який показує напрямок найбільш різкої зміни інтенсивності пікселів та величину цієї зміни. У контексті цифрових зображень, які складаються з пікселів, градієнт дозволяє виявляти межі або контури, що виникають між різними областями зображення, де спостерігаються суттєві зміни інтенсивності кольору чи світла. Це робить градієнт важливим інструментом для аналізу структурних елементів зображення. Градієнт зображення є математичним об'єктом, який вказує на найшвидшу зміну значення пікселя в певній точці. Вектор градієнта вказує напрямок цієї зміни, а його величина відображає швидкість зміни інтенсивності. Важливою особливістю градієнта є те, що він є скалярною величиною, що визначається за допомогою часткових похідних зображення по координатах. У двовимірному просторі зображення, де кожен піксель має координати (x, y) , градієнт можна виразити через похідні по x та y , що вказують на зміни інтенсивності пікселів по горизонталі та вертикалі. Це дозволяє визначати контури об'єктів на зображенні, де зміна інтенсивності різко відрізняється від навколишнього середовища.

Гradient зображення є дуже корисним інструментом для виявлення особливостей. Це вектор, що складається з часткових похідних функції інтенсивності та обчислюється для кожного пікселя:

$$\nabla J(x,y) = (\partial J(x,y) \partial x, \partial J(x,y) \partial y),$$

$$\nabla I(x,y) = (\partial I(x,y) \partial x, \partial I(x,y) \partial y).$$

Часткові похідні формально визначаються як межі темпів зростання. Gradient, будучи вектором, має норму (або амплітуду) Gradient вказує в напрямку, в якому інтенсивність змінюється найбільше в околицях пікселя (x,y) . Амплітуда показує, наскільки важлива ця варіація: чим вона вища, тим різкіша зміна.

У цифровому зображенні функція інтенсивності визначається кінцевою кількістю пікселів. Тому розрахувати часткові похідні як межі дуже важко. Часткові похідні потім апроксимуються методом кінцевих різниць:

$$\partial J(x,y) \partial x \approx J(x+1,y) - J(x,y) \quad \partial I(x,y) \partial x \approx I(x+1,y) - I(x,y),$$

$$\partial J(x,y) \partial y \approx J(x,y+1) - J(x,y) \quad \partial I(x,y) \partial y \approx I(x,y+1) - I(x,y).$$

Маски можуть використовувати інформацію, що надходить з обох боків пікселя, а не лише з одного боку. Наприклад, застосування першої в $I(x+1,y) - I(x-1,y)$ повертається $J(x+1,y) - J(x-1,y)$, $I(x+1,y) - I(x-1,y)$, що є нічим іншим, як сумою $J(x+1,y) - J(x,y)$, $I(x+1,y) - I(x,y)$ (наближення до $\partial J(x,y) \partial x$ справа) і $J(x,y) - J(x-1,y)$, $I(x,y) - I(x-1,y)$ (наближення зліва). Існує багато інших ядер згортки для апроксимації градієнта. Найпопулярніші характеризують фільтр Sobel. Наближення обчислюються з урахуванням кількох інших пікселів у сусідстві. Тоді більша вага надається сусіднім пікселям у напрямку похідної.

Застосування градієнта для виявлення контурів зазвичай включає в себе кілька етапів. На початковому етапі відбувається обчислення градієнта для кожного пікселя зображення. Це дозволяє визначити напрямок і величину зміни

інтенсивності пікселів, що допомагає виявити найбільш важливі переходи між об'єктами. Одним з найбільш популярних методів для обчислення градієнта є використання фільтрів, таких як оператори Собеля або Прюїтта, які застосовують маски для визначення змін інтенсивності пікселів у горизонтальному та вертикальному напрямках. Ці операції дозволяють виділити місця з найбільшими змінами інтенсивності, що відповідають контурним лініям на зображенні.

На наступному етапі визначається значення порогу для градієнта. Після обчислення градієнта, необхідно вирішити, які значення градієнта повинні бути визнані контурними лініями. Це зазвичай робиться шляхом застосування порогового значення, яке відокремлює важливі зміни інтенсивності від менш значущих. Всі пікселі з градієнтом, що перевищує це значення, вважаються частиною контурів. Порогові методи є критичними для контролю за кількістю виділених контурів, оскільки дуже низький поріг може призвести до надмірного виявлення шуму, а занадто високий поріг може пропустити важливі контури.

Третій етап полягає у подальшому очищенні та уточненні виявлених контурів. Оскільки в процесі обчислення градієнта можуть виникати помилки або артефакти, важливо використовувати додаткові фільтри або алгоритми, які допомагають згладжувати результати і зменшувати вплив шуму. Наприклад, часто використовуються методи хапа (hysteresis thresholding), які дозволяють з'єднати не зовсім чітко визначені контури і виключити фрагменти, що є шумами або непотрібними деталями.

Градієнт також може бути використаний в більш складних методах виявлення контурів, таких як методи на основі статистичних підходів або глибокого навчання. В останні роки спостерігається зростання інтересу до використання нейронних мереж та алгоритмів глибокого навчання для виявлення контурів, які можуть використовувати градієнт як одну з ознак для ідентифікації об'єктів на зображеннях. У таких системах зазвичай градієнт використовується разом із іншими характеристиками зображення для

покращення точності виявлення контурів і зменшення помилок, пов'язаних із освітленням, фоном або іншими складними умовами.

Краї або контури надають багато інформації про зображення. Вони розмежовують об'єкти, присутні на представленій сцені, тіні або навіть різні текстури. Одним із способів виявлення країв було б сегментувати зображення на об'єкти, але це складна проблема. Фільтр Canny, розроблений в 1986 році, є більш простим рішенням, яке засноване на вивченні градієнта. Краї знаходяться в областях зображення, де помітні сильні зміни. Дійсно, контури об'єктів відповідають змінам глибини (переходимо від одного об'єкта до іншого, розташованого на задньому плані), а тіні і різні текстури – змінам освітлення. З математичної точки зору виявлення країв зводиться до пошуку точок на зображенні, де функція інтенсивності змінюється раптово. Однак відомо, що висока амплітуда градієнта вказує на сильну зміну інтенсивності. Тому метою є пошук локальних максимумів.

Метод виявлення краю фільтра Canny складається з п'яти кроків:

Крок 1. Зменшення шуму. Шум зображення може ввести нас в оману: пікселі з відмінними значеннями викликають сильні зміни інтенсивності, навіть якщо вони не належать жодному контуру. Таким чином, зображення необхідно попередньо зняти шум за допомогою відповідного фільтра. Для зменшення рівня зашумленості можна використовувати, наприклад, фільтр Гаусса.

Крок 2. Розрахунок градієнта зображення. Градієнт знешумленого зображення апроксимується згортковою фільтрацією, найчастіше за допомогою масок Собеля. Потім обчислюється амплітуда та напрямок градієнта в будь-якій точці зображення. Краї визначаються точками високої амплітуди.

Крок 3. Видалення немаксимумів. Краї, виявлені на попередньому кроці, занадто товсті. Щоб зробити їх більш точними, вибираємо лише ті точки, для яких амплітуда градієнта локально максимальна в його напрямку. Для цього кількісно знаходимо двох сусідів кожного пікселя, дотримуючись напрямку його градієнта. При цьому визначаємо кількісне значення кута напрямку градієнта. Наприклад, піксель (x,y) для якого $\theta = 85^\circ \approx \pi/2$ матиме сусідні пікселі з

координатами $(x, y+1)$ та $(x, y-1)$. Поточний піксель зберігається, лише якщо його амплітуда більша, ніж амплітуда двох його сусідів.

Крок 4. Пороги. Серед точок, вибраних на попередньому кроці необхідно вибрати лише ті, амплітуда градієнта яких перевищує певний поріг. Важко підібрати значення «хорошого» порогу. Тому як правило віддають перевагу для використання порогу гістерезису, який використовує два пороги, позначені як низький *sbas* та високий *shaut*. Даний підхід використовує наступні правила:

- якщо $\nabla J(x, y) \geq \text{високий}$ та $I(x, y) \geq \text{високий}$, то піксель (x, y) розташований на одному краю;
- якщо $\nabla J(x, y) \geq \text{низький}$ та $I(x, y) \geq \text{низький}$, то піксель (x, y) знаходиться на ребрі тоді і тільки тоді, коли один із його восьми сусідів також;
- в іншому випадку піксель (x, y) не на межі.

Цей детектор простий у застосуванні, але має два основні обмеження:

- вибір параметрів (дисперсія фільтра Гаусса і два пороги). Вони дуже сильно впливають на час обчислення та якість результатів, але немає автоматичного методу визначення найкращих значень для кожного зображення.
- виявлені ребра є точками, а не кривими, і їх важко «зв'язати».

Іншим важливим елементом під час опису контурує є кути. Кути – це інші функції, багаті інформацією. Вони розташовані в регіонах, де інтенсивність сильно змінюється принаймні в двох напрямках. В області зображення де знаходиться кут чітко відслідковується, що інтенсивність раптово змінюється в кількох напрямках. Детектор Харріса-Стівенса, є дуже популярною технікою для визначення кутів зображення. Він заснований на детекторі Моравеса, який використовує градієнт. Детектор Моравеса використовується для визначення змін інтенсивності навколо певного пікселя. Ідея полягає в тому, щоб розглянути околиці з центром у цьому пікселі, щоб трохи зрушити його в кількох напрямках, а потім обчислити зміну інтенсивності для кожного зміщення. Ця функція застосовується в трьох основних ситуаціях (рисунок 2.4).

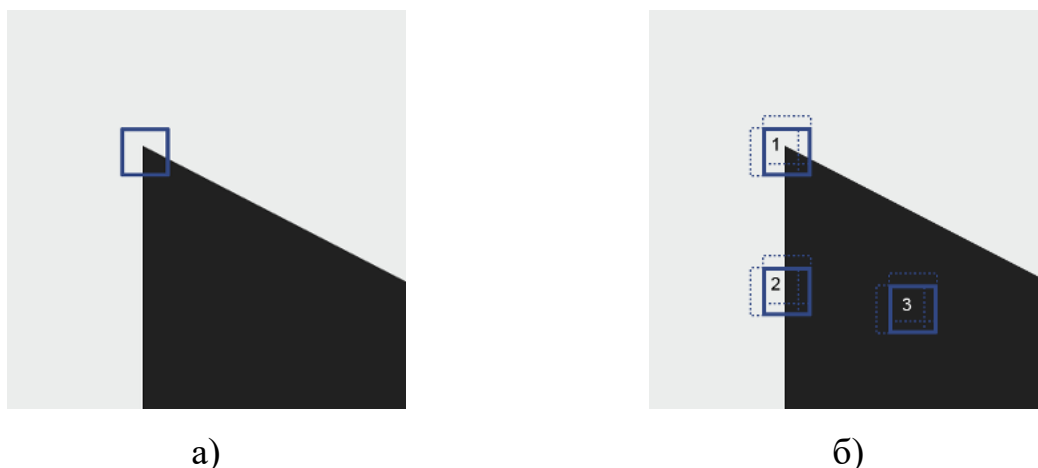


Рисунок 2.4 – Приклад виділення кута на цифровому зображенні: (а) приклад кутової функції та (б) можливі ситуації при визначенні кута за допомогою детектора Moravec

При виконанні детектора Moravec можуть виникати наступні ситуації:

Ситуація 1: область містить кут, інтенсивність змінюється різко в кількох напрямках, тому функція E приймає сильні значення в цих напрямках;

Ситуація 2: область містить контур, інтенсивність раптово змінюється, якщо ми рухаємось горизонтально та дуже мало вертикально. Отже, E приймає високі значення, якщо рухаємось перпендикулярно контуру, і низькі значення для рухів по контуру.

Ситуація 3: відсутність зміни інтенсивності: область однорідна. E потім приймає низькі значення в усіх напрямках.

Піксель (m,n) відповідає куту, якщо найменша зміна інтенсивності навколо нього є максимальною порівняно з іншими пікселями. Іншими словами, ми шукаємо локальні максимуми $F(m,n) = \min(u,v) E_{m,n}(u,v)$.

Детектор Гарріса-Стівенса є вдосконаленням детектора Моравека. Він вносить три основні зміни:

1. Квадратне вікно W_m, W_n з центром в (m,n) замінюється вікном Гауса;
2. $I(x+u, y+v)$ апроксимується розкладанням Тейлора в околицях (x,y) ;
3. Кути визначаються за новим критерієм: власними значеннями λ_1 та λ_2 .

Таким чином, виявлення кутів за допомогою детектора Харріса-Стівенса виконується в чотири етапи:

- розрахунок матриці для кожного пікселя;
- розрахунок значення параметра R для кожного пікселя;
- вибір порогового значення;
- видалення не максимумів (кути відповідають локальним максимумам).

Як і візуальні характеристики кольору та текстури, контури також мають свої особливості. З одного боку, вони можуть бути різних типів, а з іншого боку, не всі сповіщувачі мають однакову продуктивність для кожного типу контуру. Під час виявлення границь контурів якість роботи алгоритмів можна оцінити за наступними критеріями (рисунк 2.5):

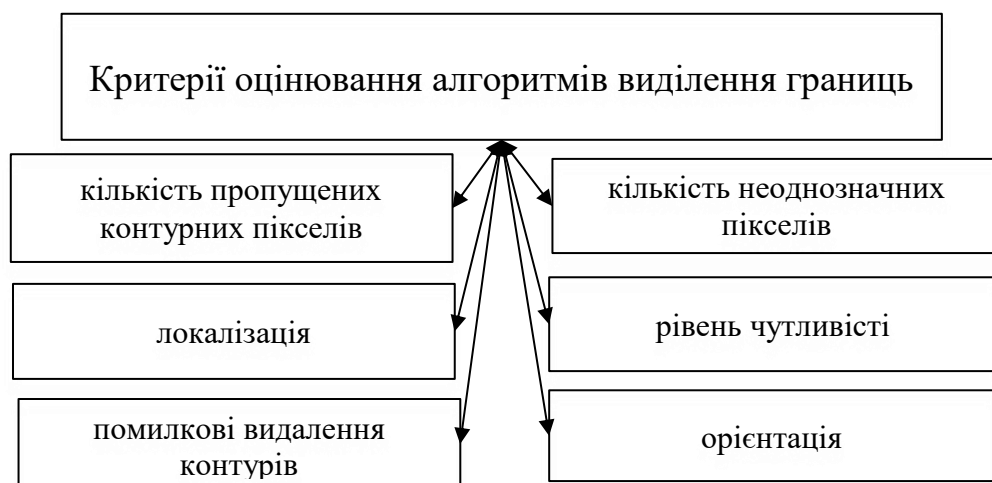


Рисунок 2.5 – Критерії оцінки алгоритмів виділення границь на цифрових зображеннях

- кількість пропущених контурних пікселів, які потрібно виявити. Вимірюється шляхом підрахунку кількості забутих пікселів по відношенню до загальної кількості пікселів ідеального контуру;
- кількість неоднозначних пікселів (піксель віднесений до декількох контурів). Вимірюється шляхом підрахунку кількості неоднозначних пікселів порівняно з тими, які такими не є.

- локалізація (ситуація коли однозначний піксель контуру знаходиться віддалено від контуру(не в правильному місці)). Вимірюють шляхом підрахунку загальної відстані між виявленим контуром та ідеальним контуром.

- рівень чутливості. Описує ситуацію коли було виявлено помилковий контур біля ідеального контуру. Вимірюють шляхом підрахунку кількості помилкових контурів і загальної кількості виявлених контурів

- помилкові видалення контурів, як правило, після порогу. Цієї помилки можна уникнути, переконавшись, що інтенсивність світла справжнього контуру є більшою, ніж інтенсивність світла помилкового контуру.

- орієнтація: ця помилка виникає, коли орієнтація контуру неправильна. Вимірюється шляхом обчислення різниці між розрахунковою орієнтацією та справжньою орієнтацією для всіх справжніх контурів.

Основним завданням методів детекції країв є пошук точок з найбільшими змінами інтенсивності пікселів, що свідчить про наявність межі між об'єктами або їх частинами. У загальному випадку детекція країв є попереднім етапом у складних задачах, таких як класифікація об'єктів, розпізнавання, відстеження об'єктів або побудова 3D-моделей. Залежно від специфіки завдання та властивостей зображення, використовуються різні алгоритми для ефективного виявлення та відображення країв.

Одним з найвідоміших і найширше використовуваних методів детекції країв є оператор Собеля. Цей метод заснований на обчисленні градієнта інтенсивності пікселів, що дозволяє знаходити найбільші зміни інтенсивності, які зазвичай відповідають краям об'єктів. Оператор Собеля використовує два фільтри для обчислення градієнта в напрямках по осі x і y . Потім, обчислюючи величину градієнта, визначаються місця на зображенні, де відбувається різка зміна інтенсивності. Це дозволяє виділити контури та межі між різними об'єктами на зображенні.

Іншим популярним методом є оператор Канні, який є вдосконаленим варіантом оператора Собеля. Оператор Канні включає кілька етапів обробки, серед яких є фільтрація для зменшення шуму, обчислення градієнтів, а також

застосування порогових значень для виділення чітких країв. Оператор Канні дозволяє отримувати більш точні і менш зашумлені контури, що робить його популярним у багатьох практичних застосуваннях, зокрема у медичній діагностиці та відеоспостереженні.

Оператор Лапласа є ще одним відомим методом детекції країв. У порівнянні з операторами Собеля та Канні, оператор Лапласа використовує метод другого похідного для виявлення країв. Оператор Лапласа визначає точки, де відбуваються різкі зміни у вигині інтенсивності, що дозволяє ефективно виявляти контури, особливо в областях, де градієнт не є максимальним, але є значні зміни в кривизні інтенсивності. Хоча оператор Лапласа має здатність виявляти більше контурів, він часто генерує більше шуму, ніж оператори Собеля або Канні, тому вимагає додаткових кроків фільтрації для досягнення високої точності.

Окрім класичних методів, існують також більш сучасні та складні підходи, засновані на методах машинного навчання та глибинного навчання. Ці методи використовують нейронні мережі та інші алгоритми, які можуть навчатися на великих наборах даних, щоб виявляти контури та межі зображень з високою точністю. Наприклад, згорткові нейронні мережі (CNN) застосовуються для автоматичного виявлення країв і контурів, особливо коли зображення мають складні структури або є значний рівень шуму. Такі підходи дозволяють досягти кращих результатів у завданнях, де традиційні методи не завжди ефективні.

Методи детекції країв можна умовно поділити на два основних підходи: методи, засновані на градієнті, і методи, засновані на інших математичних характеристиках зображення. Методи, засновані на градієнті, включають такі алгоритми, як оператори Собеля, Канні та Лапласа. Вони є досить простими та швидкими для обчислення, що робить їх популярними для реального часу та вбудованих систем. Однак ці методи можуть бути чутливими до шуму та вимагають додаткових етапів фільтрації для покращення результатів.

Інші методи детекції країв використовують більш складні математичні моделі. Наприклад, методи, засновані на аналізі сегментації, дозволяють більш

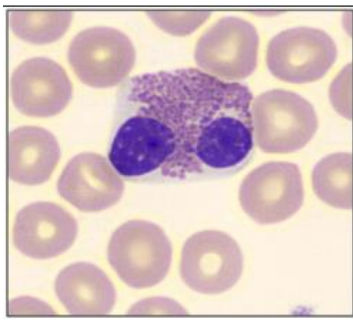
точно відокремлювати об'єкти на зображеннях, навіть коли вони мають складні контури або перекриття. Ці методи зазвичай є більш точними, але й більш обчислювально затратними. Вони включають такі підходи, як методи на основі рівняння рівноваги, рівняння енергії чи рівняння Гіббса.

Особливо важливою є обробка країв на зображеннях з низьким контрастом або з великою кількістю шуму. Для таких зображень використовуються спеціальні методи, що поєднують детекцію країв з фільтрацією, щоб зменшити вплив шуму. Прості фільтри, такі як гаусовий фільтр, часто використовуються для згладжування зображення перед застосуванням алгоритмів детекції країв. Це дозволяє знизити кількість помилкових спрацьовувань і покращити точність виявлення контурів.

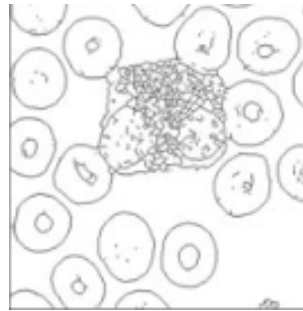
Детекція країв є важливим етапом в багатьох реальних застосуваннях, таких як автоматичне розпізнавання об'єктів, медична діагностика, системи безпеки та автономні транспортні засоби тощо. У таких випадках важливою є точність і швидкість алгоритмів детекції країв.

2.3 Алгоритм виділення та опису об'єктів на цифрових зображеннях

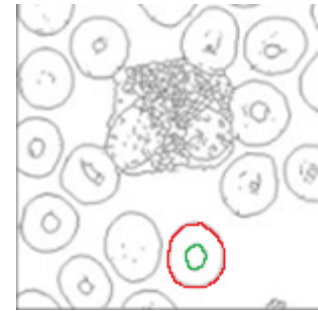
Проведений аналіз алгоритмів підкреслення меж на цифрових зображеннях дозволив розробити алгоритм виділення та опису елементів цифрового зображення на основі використання засобів контурного аналізу. Запропонований алгоритм використовує інформацію про перепади яскравості на зображенні для виявлення місць переходів від одного елемента до іншого. Кольорові зображення перекодовуються у відтінки сірого для прискорення їх опрацювання та зменшення необхідної кількості операцій в процесі обробки. Приклад етапів обробки цифрового зображення наведено на рисунку 2.6. В результаті обробки об'єкти на цифровому зображенні отримують виділений контур, на основі аналізу якого відбувається опис об'єкта.



а) Вхідне зображення



б) Підкреслені границі



в) Виділений контур

Рисунок 2.6 – Схема опису цифрового зображення на основі детекторів границь

Візуально етапи роботи запропонованого алгоритму наведено у вигляді блок-схеми алгоритму на рисунку 2.7.

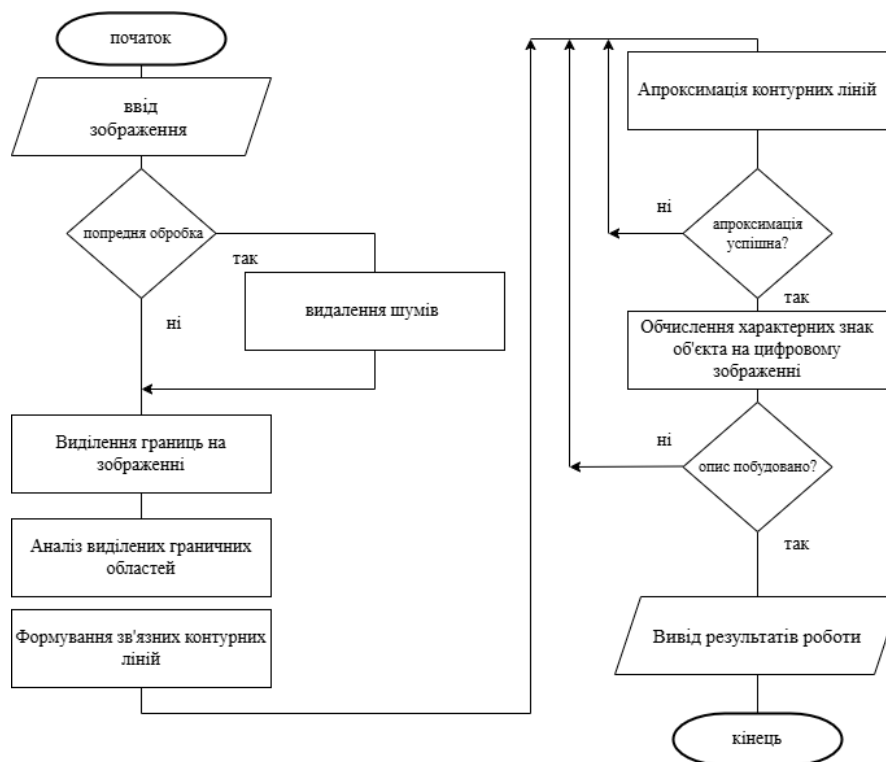


Рисунок 2.7 – Блок-схема алгоритму виділення та опису об'єктів на цифрових зображеннях з використання елементів контурного аналізу

Даний алгоритм може використовуватись для обробки та опису сен на цифрових зображень як самостійно так і бути інтегрованим в складні системи

аналізу цифрових зображень. Детальна послідовність кроків розробленого алгоритму наведена нижче:

- 1) завантаження цифрового зображення I_m .
- 2) Визначення рівня зашумленості, якщо рівень зашумленості $noise$ більший за порогове значення T , то використовуємо фільтри для зменшення зашумленості зображення.
- 3) Використання алгоритмів підкреслення границь (можуть використовуватись оператори Собеля, Превіта, Кенні тощо).
- 4) Аналіз результату роботи алгоритмів операторів підкреслення границь.
- 5) Якщо виділені границі інформативні, то утворюємо зв'язний контур для окремих об'єктів.
- 6) Проводим опроксимацию контурних ліній об'єктів.
- 7) Обчислення характеристик об'єктів на основі аналізу апроксимованої контурної лінії (координати розташування, периметр, площа, колоподібність, орієнтація тощо).
- 8) Формування опису цифрового зображення на основі отриманих описів окремих його елементів та вивід результатів аналізу зображення.

2.4 Висновки до розділу

Проведено аналіз алгоритмів виділення та опису цифрових зображень на основі використання контурного аналізу, що дозволило виділити групу операторів підкреслення границь цифрових зображень для подальшої їх інтеграції в систему опрацювання цифрових кольорових зображень.

Розроблено алгоритм опису об'єктів цифрових зображень на основі аналізу контурної лінії, що дозволило програмно реалізувати додаток опису елементів на цифрових зображеннях шляхом виділення границь об'єктів на цифрових зображеннях.

3 ПРОГРАМНИЙ ДОДАТОК ОБРОБКИ ТА ОПИСУ ЦИФРОВИХ КОЛЬОРОВИХ ЗОБРАЖЕНЬ

3.1 Архітектура реалізованого програмного додатку

Актуальність систем обробки цифрових зображень з інтегрованими алгоритмами контурного аналізу в важко переоцінити, оскільки вони є основою багатьох важливих завдань у галузі комп'ютерного зору та обробки зображень. Аналіз контурів об'єктів є одним із першочергових етапів у процесі автоматичної інтерпретації зображень, що включає виявлення меж між різними частинами зображення, їх виділення та подальше класифікування. Виявлення контурів має велике значення в таких областях, як медична діагностика, автономні системи, робототехніка, відеоспостереження та багато інших. Контури в даних системах служать ключовими індикаторами меж між різними частинами зображення, дозволяючи точно локалізувати і розпізнавати об'єкти. Без чіткої детекції контурів не можна ефективно здійснювати жодні подальші етапи обробки зображень, такі як сегментація, класифікація або навіть відстеження об'єктів у відеопотоці. Сучасні алгоритми контурного аналізу базуються на різних математичних принципах, включаючи операції фільтрації, градієнтні методи, аналіз змін інтенсивності та застосування складних геометричних моделей для опису контурів. Класичні методи детекції країв хоча і дуже швидкі, проте не завжди дають бажану точність, особливо коли об'єкти на зображенні мають складні форми або погано виражені межі. З цієї причини все більше поширення набувають методи, засновані на машинному навчанні, зокрема, нейронні мережі. Інтеграція цих сучасних методів з класичними підходами відкриває нові можливості для більш точного та швидкого аналізу контурів в реальному часі. Це особливо важливо для таких застосувань, як відеоспостереження, автономні транспортні засоби та робототехніка, де необхідна висока точність та швидкість обробки зображень для ефективної взаємодії з навколишнім середовищем. Наприклад, в автономних транспортних засобах аналіз контурів дозволяє точно виявляти межі між дорогою і об'єктами навколо, що є важливим для прийняття

правильних рішень щодо маневрування. Системи обробки цифрових зображень, які застосовують алгоритми контурного аналізу, можуть значно покращити ефективність у різних галузях, таких як автоматичне управління, безпека, охорона здоров'я та наукові дослідження. Завдяки розвитку технологій машинного навчання, алгоритми контурного аналізу стають все більш точними, швидкими та адаптивними до різних умов, що робить їх актуальними для вирішення складних завдань у реальному часі.

Поєднання OpenCV та Python для обробки цифрових зображень є надзвичайно потужним інструментом для дослідження та розв'язання завдань комп'ютерного зору. Ці дві технології поєднують у собі численні переваги, зокрема, простоту використання, ефективність, відкритість і гнучкість, що робить їх ідеальними для розробки як простих, так і складних програмних рішень у галузі обробки зображень. Аналіз переваг цього поєднання дає змогу глибше зрозуміти, чому саме OpenCV та Python так популярні серед дослідників, розробників і інженерів, які займаються цифровою обробкою зображень.

Однією з головних переваг використання OpenCV разом із Python є зручність і швидкість розробки. Python, будучи високорівневою мовою програмування, надає простий і зрозумілий синтаксис, що дозволяє розробникам швидко писати ефективний код без необхідності вручну працювати з низькорівневими деталями, які часто присутні в інших мовах програмування. Це особливо важливо для задач обробки зображень, де розробникам необхідно швидко протестувати різні алгоритми та модифікувати параметри без занурення у складність реалізації низькорівневих операцій. Водночас OpenCV, будучи потужною бібліотекою для обробки зображень і відео, пропонує багатий набір функцій для виконання численних операцій з цифровими зображеннями, таких як фільтрація, виявлення контурів, трансформація зображень, аналіз текстур і багато інших. Завдяки Python розробники можуть швидко взаємодіяти з цими функціями через прості та інтуїтивно зрозумілі виклики, що значно скорочує час розробки.

Ще однією суттєвою перевагою є відкритість і доступність обох технологій. OpenCV є відкрите програмне забезпечення, що означає, що кожен може вільно використовувати, модифікувати і поширювати його. Бібліотека активно підтримується спільнотою розробників, що дозволяє постійно покращувати функціональність і додавати нові можливості. Крім того, величезна кількість прикладів, документації та навчальних матеріалів доступні в Інтернеті, що значно полегшує навчання і вирішення специфічних завдань. Python, у свою чергу, є популярною мовою програмування з відкритим кодом, і його екосистема включає численні бібліотеки для аналізу даних, штучного інтелекту, чисельних методів та обробки зображень. Завдяки цьому комбінація Python і OpenCV дозволяє легко інтегрувати алгоритми комп'ютерного зору з іншими аспектами програмних систем, такими як машинне навчання, обробка даних або веб-розробка.

Python і OpenCV також надають гнучкість при роботі з різними типами зображень і відео. Вони підтримують широкий спектр форматів зображень, що дозволяє працювати з даними з різних джерел, таких як камери, сканери, медичні пристрої або навіть відеофайли. Крім того, завдяки інтеграції з іншими бібліотеками Python, такими як NumPy, SciPy та Pandas, можна ефективно обробляти та маніпулювати великими масивами даних. Наприклад, NumPy дозволяє працювати з зображеннями як з багатовимірними масивами, що полегшує маніпуляцію пікселями або виконання математичних операцій на зображеннях, таких як фільтрація, трансформації або обчислення градієнтів. Це дозволяє значно спростити складні завдання обробки зображень і отримати високопродуктивні рішення за короткий час.

Обрана комбінація OpenCV та Python є надзвичайно потужним і гнучким інструментом для обробки цифрових зображень. Це поєднання дозволяє розробникам створювати ефективні, масштабовані та високопродуктивні рішення для різноманітних завдань, від базових операцій з обробки зображень до складних застосувань в комп'ютерному зорі та машинному навчанні. Переваги, такі як простота використання, відкритість, підтримка широкого

спектра форматів і можливість інтеграції з іншими технологіями, роблять OpenCV і Python одними з найбільш популярних інструментів у галузі обробки зображень.

Для проектування внутрішньої структури програмного модуля було обрано модальний підхід з чіткими розмежуваннями між різними структурними елементами програмного додатку. Модульна архітектура програмних систем є одним з найбільш важливих підходів у розробці складних програмних продуктів. Вона передбачає розбиття програмної системи на окремі, незалежні або слабо зв'язані компоненти, що називаються модулями. Кожен модуль виконує свою специфічну задачу і може бути розроблений, протестований, оновлений або замінений окремо від інших частин системи. Це дає можливість значно спростити процес розробки, тестування та підтримки програмних систем, особливо коли йдеться про великі проекти. Основною перевагою модульної архітектури є її здатність до масштабування. Коли система складається з незалежних модулів, кожен з яких має чітко визначену функцію, це дає змогу командам розробників паралельно працювати над різними частинами проекту, що значно скорочує час на розробку. Крім того, таке розбиття дозволяє здійснювати модифікацію або вдосконалення лише конкретних модулів, не порушуючи роботи інших частин системи. Це робить підтримку програмного продукту більш ефективною і менш затратною, оскільки зміни в одному модулі не тягнуть за собою необхідність оновлення всієї системи. Ще одним важливим аспектом модульної архітектури є підвищення надійності та безпеки програмних систем. Оскільки модулі є відокремленими одиницями, кожен з них можна тестувати і перевіряти на наявність помилок незалежно від інших частин програми. Це забезпечує можливість виявлення та усунення помилок на ранніх етапах розробки. Крім того, модульна архітектура дозволяє застосовувати принципи безпеки більш ефективно, оскільки доступ до кожного модуля можна обмежити і контролювати окремо. Це знижує ризики, пов'язані з небажаними взаємодіями між частинами програми і дає можливість краще захищати систему від потенційних атак. Обраний модульний підхід є практично виправданим,

оскільки дозволяє знижувати складність і підвищувати надійність програмного забезпечення. Додатково він забезпечує ефективність, гнучкість і масштабованість, проте вимагає ретельного планування і організації для того, щоб уникнути проблем з взаємодією між модулями

Для реалізації програмної системи по обробці цифрових зображень з інтеграцією запропонований модуль було розроблено внутрішню архітектуру. Принципи побудови внутрішніх зв'язків та набір відповідних модулів був побудований на основі попереднього аналізу структур систем подібного типу. Також при проектуванні архітектури були враховані негативні рішення, а також було розділено окремі великі блоки з метою спрощення процесу реалізації та тестування програмного засобу. Отримана архітектура оформлена у вигляді взаємопов'язаних блоків та представлена на рисунку 3.1.

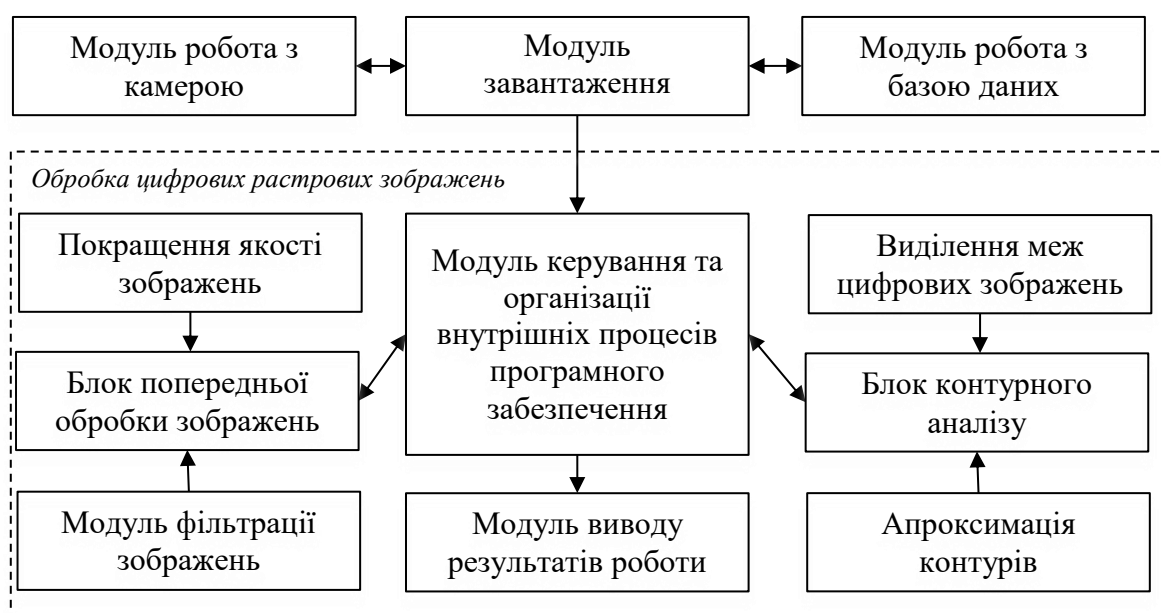


Рисунок 3.1 – Узагальнена структура програмного додатку обробки цифрових зображень з використанням елементів контурного аналізу

Наведена структура програмної системи (див. рисунок 3.1) обробки цифрових зображень є багатофункціональною та модульною. Вона включає кілька компонентів, що взаємодіють для забезпечення ефективної обробки

цифрових растрових зображень. Для більш детального аналізу функціональних можливостей кожного блоку проведемо їх опис.

Модуль роботи з камерою. Відповідає за отримання цифрових зображень безпосередньо з камер або інших пристроїв захоплення зображень. Це первинне джерело даних, яке може включати обробку сигналу від сенсорів камери та передавання отриманих зображень у систему. Функції даного модуля дозволяють проводити налаштування параметрів роботи фотофіксуючої апаратури для отримання максимально якісних вхідних зображень. Даний модуль дозволяє отримувати дані в режимі реального часу, що значно підвищує функціональні можливості та зручність використання програмного додатку.

Модуль завантаження. Забезпечує імпорт зображень із різних джерел, таких як файлові системи, мережеві ресурси чи бази даних. Цей модуль є початковим етапом обробки, який формує вхідний потік даних для подальшого аналізу. В ньому реалізовані функції по управлінню та оцінці вхідних зображень з подальшим визначенням параметрів роботи програмної системи. Додатково функції даного модуля можуть проводити операції по встановленню метаданих для окремо взятого цифрового зображення. Для цього користувачеві буде надаватись відповідних функціонал для встановлення міток.

Модуль роботи з базою даних. Забезпечує інтеграцію системи з базами даних, у яких зберігаються зображення та метадані, пов'язані з ними. Використовується для завантаження або збереження результатів обробки. Даний модуль носить допоміжний характер та дозволяє організувати взаємодію не тільки з стандартними базами даних, а надає можливості для конфейєрної обробки набору цифрових зображень.

Покращення якості зображень. Цей модуль займається оптимізацією якості вхідних зображень. Використовує методи підвищення чіткості, корекції кольорів, зменшення шумів та інших дефектів. Функції даного модуля дозволяють визначити рівень зашумленості вхідного зображення та за допомогою фільтрації зменшити рівень різких перепадів яскравості на зображенні.

Блок попередньої обробки зображень. Виконує базові операції з обробки цифрових зображень. В даному модулі реалізовано функції масштабування, обертання, нормалізації значень пікселів на цифрових зображеннях. В результаті обробки цифрового зображення буде отримуватись зображення яке максимально підходить для ефективної роботи усіх інших модулів програмної системи.

Модуль фільтрації зображень. Відповідає за застосування фільтрів, таких як низькочастотна чи високочастотна фільтрація, для видалення шумів, згладжування або підвищення деталізації. Окрім стандартних фільтрів в програмі реалізована можливість корекції параметрів ядра фільтраційної маски, що дозволяє користувачу вносити корекції в процес роботи програмного додатку з метою підвищення рівня якості отриманих результатів.

Модуль керування та організації внутрішніх процесів програмного забезпечення. Цей модуль є центральною частиною системи, яка забезпечує координацію роботи всіх інших модулів. Вона управляє потоками даних, регулює порядок виконання завдань та забезпечує узгодженість між етапами обробки. Окрім того, в даному модулі закладені функції по перевірці цілісності та наявності необхідних програмних модулів, їх коректна робота та можливість проведення корекції при виявленні помилкових ситуацій. Реалізовані принципи взаємодії окремих модулів через буферний блок дозволяє підвищити рівень надійності роботи програмної системи, оскільки під час виявлення нештатних ситуацій блок зможе коректно припинити процес виконання обробки цифрового зображення, зберегти поточний стан та сформулювати відповідне повідомлення для користувача з описом помилки.

Виділення меж цифрових зображень. Виконує ідентифікацію контурів та сегментацію зображень шляхом аналізу градієнтів або інших характеристик пікселів. Один з основних елементів структури програмного додатку, оскільки саме в ньому зібрані функції для вибілення областей з різкими перепадами яскравості на цифровому зображенні. Для виідлення використовуються класичні алгоритми виділення, проте за необхідності можуть бути додані і нові алгоритми.

Блок контурного аналізу. Реалізує детальне дослідження виділених контурів для визначення їх властивостей, таких як форма, розмір або взаєморозташування. Даний блок є організаційним та дозволяє коректно організувати взаємодію між блоками додаткової обробки зображень та алгоритмами контурного аналізу.

Аппроксимація контурів. Здійснює спрощення контурів або їх опис математичними моделями (лініями, кривими або багатокутниками), що полегшує подальшу обробку та аналіз. Даний підхід дозволяє отримати компактний опис елемента з можливістю отримання додаткових характеристичних ознак для отримання більш детального опису елемента цифрового зображення. Серед додаткових параметрів можна відзначити площу, периметр, колоподібність, орієнтацію, центр мас тощо.

Модуль виводу результатів роботи. Забезпечує візуалізацію або збереження результатів роботи системи. Включати створення зображень, графіків, звітів, які показують оброблені дані та статистичні висновки.

Структура організована таким чином, що дані проходять через послідовність обробки: від захоплення та завантаження до виводу результатів. Центральний модуль керування забезпечує узгоджену роботу всіх блоків, гарантуючи ефективність і надійність системи. Ця архітектура є прикладом модульного підходу до розробки програмного забезпечення, що дозволяє легко адаптувати або розширювати функціональність системи.

Моделювання архітектури програмних систем є важливим етапом розробки, оскільки дозволяє детально спланувати структуру та взаємодію компонентів системи ще на початкових етапах проєкту. Одними з основних інструментів для цього є діаграми послідовності та прецедентів, які сприяють чіткому визначенню взаємозв'язків між різними частинами системи.

Діаграми прецедентів, у свою чергу, слугують для моделювання статичних аспектів системи, зокрема взаємодії між користувачами та системою. Вони допомагають визначити основні функціональні вимоги та сценарії використання, що необхідно врахувати при розробці програмного продукту. Прецеденти

дозволяють розробникам зрозуміти, як користувачі взаємодіють з системою, що є основою для подальшого визначення функціональних модулів та їх взаємодії. Приклад процесу моделювання на основі діаграми преценентів виконувався з метою оцінювання можливості користувача впливати на результати роботи програмного додатку на різних етапах його роботи. Отримані результати моделювання були представлені на рисунку 3.2.

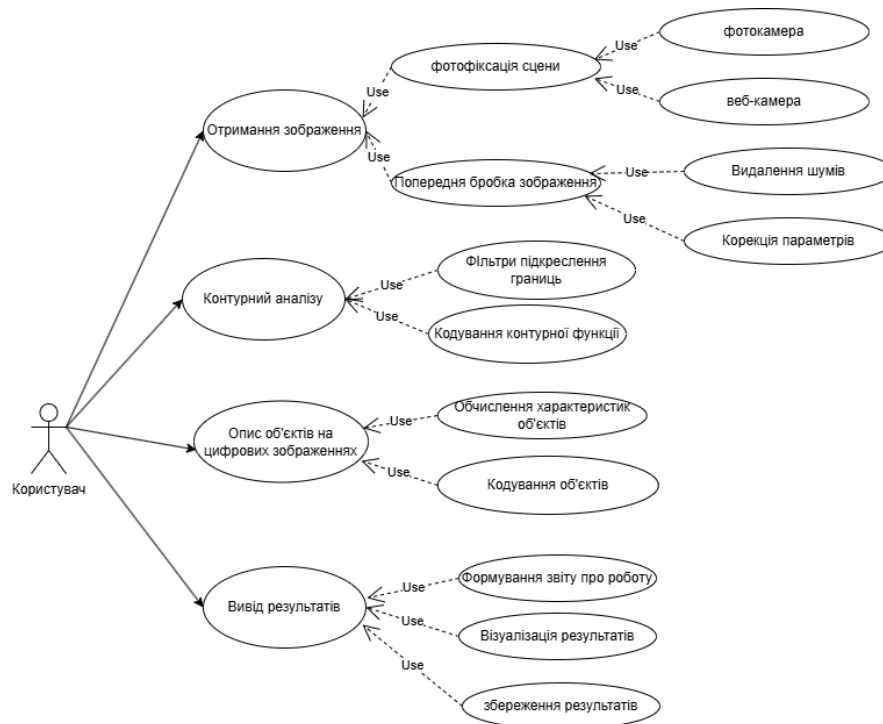


Рисунок 3.2 – Діаграма преценентів програмної системи

Під час моделювання було доведено, що користувачі мають можливість доступу до усіх функціонально важливих елементів програмної розробки. Частина функцій може виконуватись як в ручному так і в автоматичному режимах. Для вирішення окремих задач по обробці та аналізу цифрового зображення користувачу надають декілька альтернативних шляхів, що значно підвищує функціональні моливості та рівень зручності використання програми.

Діаграми послідовності є важливими для опису динамічних аспектів роботи програмної системи. Вони дозволяють уявити, як об'єкти взаємодіють між собою протягом часу, що особливо корисно при проєктуванні складних

систем з багатьма компонентами. Завдяки таким діаграмам розробники можуть виявити потенційні проблеми з синхронізацією, часом відгуку або ефективністю взаємодії компонентів, що дозволяє коригувати архітектуру на ранніх етапах розробки, до початку кодування. В системах обробки цифрових зображень основні проблеми можуть виникати під час некоректної пердачі параметрів або вхідних даних. Для дослідження та підхилення даної проблеми уже на етапі моделювання було проведене дослідження на основі діаграми послідовності. З результатами моделювання можна ознайомитись на рисунку 3.3.

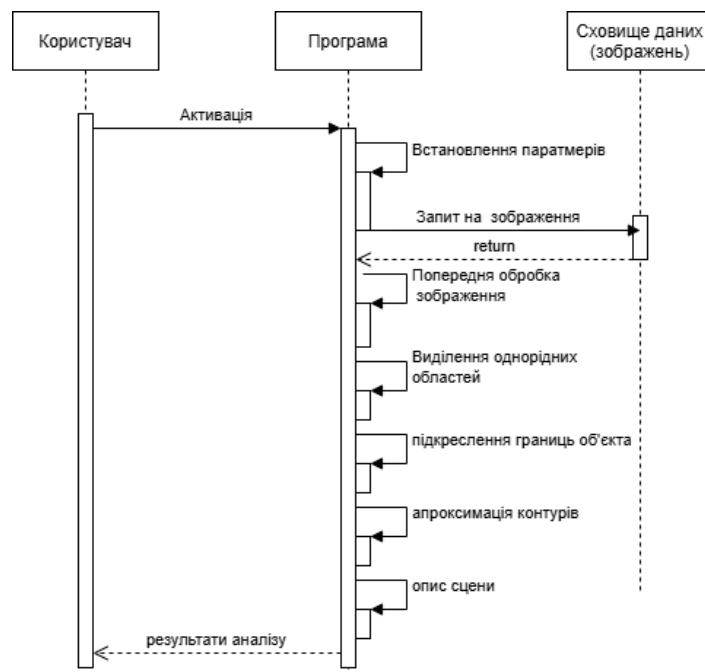


Рисунок 3.3 – Діаграма послідовності програмної системи

Проведене моделювання довело, що в середині програмної системи дані передаються у строгій послідовності, а функції запускаються тільки після завершення попереднього етапу обробки. Дана ситуація забезпечується використання додаткового модуля керування внутрішніми процесами. Використання даного модуля з однієї сторони незначно зповільнює процеси борацювання зображення, проте дозволяє уникати аварійних ситуацій. Тому в кінцевому випадку інтеграція даного блоку є правильним рішенням на етапі проектування структури програмного додатку.

Моделювання архітектури програмної системи з використанням діаграм послідовності та прецедентів забезпечує структурований підхід до проектування, дозволяючи заздалегідь виявити недоліки у плануванні та мінімізувати ризики на етапі реалізації. Це сприяє створенню більш надійних і ефективних програмних продуктів.

Програмна реалізація була реалізована на основі використання множини класів та структур даних які надасть цифрова бібліотека OpenCV та мова python. Проте стандартні засоби не дозволяють у повні мірі реалізувати спроектовані алгоритм та структуру даних. Тому додатково було спроектовано набір класів, який містить набір методів та атрибутів, що значно спрощує процеси написання функціоналу по обробці цифрових зображень. Ієрархія запропонованих класів обробки вхідних зображень наведена на рисунку 3.4.

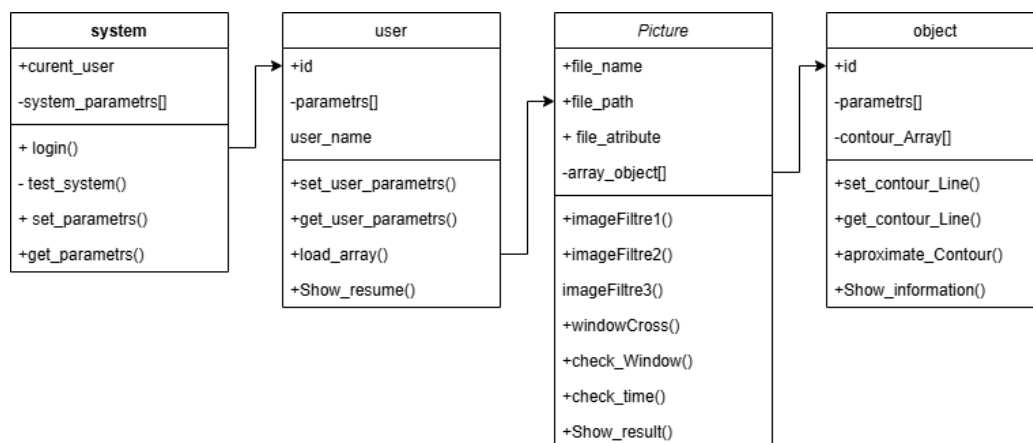


Рисунок 3.4 – Діаграма класів програмної розробки

Ще одним важливим елементом розробленої програмної системи є проектування інтерфейсу для взаємодії користувача з елементами програмної системи. В першочерговій реалізації інтерфейс було реалізовано у вигляді тестового меню, проте даний підхід є мало ефективним. Використання текстового меню значно сповільнює процес навчання та використання запропонованого програмного засобу та знижує рівень зручності. Тому для усунення даних

недоліків було спроектовано та програмно реалізовано графічний інтерфейс користувача який наведено на рисунку 3.5.

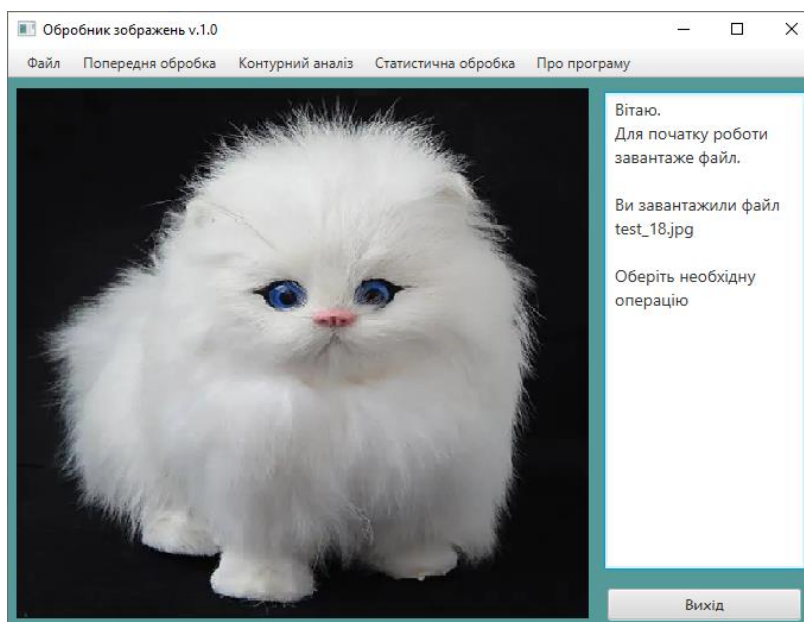


Рисунок 3.5 – Головне вікно програми

Запропонований дизайн програмного додатку обробки цифрових зображень подібний до інтерфейсів більшості реалізацій, що представлені на ринку. Це дозволить користувачам значно швидше опанувати та виконувати поставлені завдання. Основне вікно програми розделено за принципом максимального місця для відображення результатів роботи. Це зроблено з метою максимізувати можливість користувача по оцінці отриманих в процесі роботи результатів. Для виводу інформації про поточний стан роботи системи та візуалізації технічних даних додатков інтегроване текстову вікно. В ньому відображається інформація про цифрові зображення, що є особливо корисним підчас конфеєрної обробки зображень. Додатково тут відображаються повідомлення/підказки від системи у випадку виникнення помилок. Запропонований дизайн головного вікна виконаний в мінімалістичному стилі, що дозволяє зменшити відволікання уваги користувача на додаткові процеси. Основна увага буде зосереджена на процесі обробки цифрових зображень та аналізу отриманих статистичних та візуальних даних.

3.2 Функції програмного модуля обробки растрових зображень

Однією з головних переваг використання мови python є простота реалізації. Код використовує популярні бібліотеки, такі як OpenCV та imutils, що забезпечує легкий доступ до потужних інструментів для обробки зображень, таких як виявлення контурів, фільтрація країв та перспективна трансформація. Ці інструменти добре оптимізовані і мають широке використання в комп'ютерному зорі, що дозволяє зосередитись на основній задачі без необхідності розробляти складні алгоритми з нуля. Ще однією перевагою є адаптивність цього коду до різних вхідних зображень. Завдяки використанню методів для корекції перспективи та порогової обробки, код здатний обробляти зображення з різними характеристиками, такими як документ із складним фоном або нахилена фотографія. Використання методів, таких як апроксимація контурів і фільтрація країв, дозволяє виділяти основні об'єкти та коригувати їх вид так, щоб поліпшити подальшу обробку зображення.

Одним з основних модулів, які були інтегровані в програмну систему був модуль виділення границь цифрових зображень, який містить реалізації операторів з різними ядрами. Для коректності роботи функцій в даному модулі були підключені додаткові бібліотеки. У першому рядку імпортується функція `four_point_transform` із модуля `transform`, яка реалізує перспективну трансформацію з чотирьох точок для коригування вигляду зображення. Бібліотека `skimage.filters` імпортує метод `threshold_local`, що використовується для порогової фільтрації, яка допомагає в отриманні бінарного зображення. Інші імпортовані бібліотеки включають: `numpy` (для роботи з масивами даних), `cv2` (основна бібліотека для обробки зображень OpenCV), `imutils` (набір утиліт для спрощення роботи з OpenCV, включаючи функції для масштабування зображень).

Для початку роботи програми необхідно завантажити вхідне зображення, для цього використовують наступні функції. Функція `load_image(path)` виконує

завантаження зображення з диска за допомогою функції `cv2.imread()`. Зображення масштабуються таким чином, щоб його висота дорівнювала 500 пікселям (це робиться для того, щоб полегшити подальшу обробку зображення). Важливий момент: зображення зберігається в оригінальному вигляді в змінній `orig`, що дозволяє у подальшому порівнювати оригінал із зміненою версією. Потім зображення відображається на екрані за допомогою `cv2.imshow()`.

```
def load_image(path):  
    image = cv2.imread(path)  
    ratio = image.shape[0]/500.0  
    orig = image.copy()  
    image = imutils.resize(image, height=500)  
    cv2.imshow('image', image)  
    cv2.waitKey(0)  
    return image
```

Настпним етапом виділення границь об'єктів є переконвертація вхідного зображення в бінарне. Таке перекодування необхідне для коректної роботи оператора підкреслення границь. У функції зображення конвертується з кольорового (BGR) в чорно-біле за допомогою функції `cv2.cvtColor()`. Потім на нього застосовується гаусове розмивання для зменшення шуму. В кінці застосовується детектор країв `cv2.Canny()`, який знаходить контури об'єктів на зображенні. Результатом є зображення, яке містить лише краю об'єктів, позначені білим кольором на чорному фоні.

```
def convert_image(image):  
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
    gray = cv2.GaussianBlur(gray, (5, 5), 0)  
    edged = cv2.Canny(gray, 75, 200)  
    return edged
```

Пошук контурів є важливим етапом у процесі обробки зображень, оскільки саме контури дозволяють визначити межі об'єктів на зображенні, що дає змогу виконувати подальші операції, такі як сегментація або ідентифікація. Ось детальний опис цього кроку:

```

def find_contour(edged, perc=0.01):
    cnts = cv2.findContours(edged.copy(), cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
    cnts = cnts[0] if imutils.is_cv2() else cnts[1]
    cnts = sorted(cnts, key=cv2.contourArea,
reverse=True)[:5]
    sapproxes = np.empty(0)
    for countour in cnts:
        peri = cv2.arcLength(countour, True)
        approx = cv2.approxPolyDP(countour, perc * peri,
True)
        if len(approx) == 4:
            screenCnt = approx
            break

```

Цей фрагмент коду відповідає за пошук контурів на зображенні, що було оброблене (зазвичай після застосування фільтрації країв, як у функції `convert_image`). Функція `cv2.findContours()` – це стандартна функція OpenCV для пошуку контурів на бінарних зображеннях. Вона приймає три аргументи:

- вхідне зображення, у якому потрібно знайти контури (після застосування детектора країв, такого як Canny);
- режим пошуку контурів (в прикладі використовується режим `cv2.RETR_LIST`, який дозволяє знаходити всі контури в зображенні без ієрархії (не пов'язує контури один з одним);
- метод для збереження контурів. (в прикладі це `cv2.CHAIN_APPROX_SIMPLE`, що означає використання алгоритму для спрощення контурів, зберігаючи лише основні точки, що визначають контури).

Функція повертає список знайдених контурів (у форматі координат пікселів, що утворюють контур).

Після етапу виділення контури необхідно додатково опрацювати шляхом їх сортування. Контури сортуються за площею в порядку спадання, використовуючи функцію `cv2.contourArea()`, щоб вибрати найбільші контури. Вибираються лише перші 5 контурів. Дана кількість може бути зміненою в процесі роботи програми, оскільки це є допоміжний параметр роботи програми. Це корисно, оскільки часто саме найбільші контури на зображенні можуть бути тими, що містять основний об'єкт.

Для кожного з найбільших контурів виконується апроксимація за допомогою методу `cv2.approxPolyDP()`. Цей метод спрощує контур, зменшуючи кількість точок, необхідних для його опису. Параметр `perc` використовується для налаштування точності апроксимації. Це дозволяє задавати точність апроксимації відносно периметра контуру.

Якщо знайдено контур, то на оригінальному зображенні малюється цей контур зеленим кольором. Контур малюється через функцію `cv2.drawContours()`, де `screenCnt` контур об'єкта, а параметр `(0, 255, 0)` задає колір малювання (зелений). Результат показується у вікні, використовуючи `cv2.imshow()`.

```
cv2.drawContours(image, [screenCnt], -1, (0, 255, 0), 1)
cv2.imshow('Outline', image)
cv2.waitKey(0)
```

Точки контуру зберігаються в змінній `screenCnt`. Оскільки зображення було змінено за висотою до 500 пікселів (на етапі `load_image()`), для коректної масштабованої перспективи координати точок множаться на коефіцієнт, який враховує масштабування зображення.

Після виконання перспективної трансформації, що дозволяє виправити перспективу об'єкта, зображення перетворюється в чорно-біле. Це робиться за допомогою `cv2.cvtColor()`, яка конвертує зображення з кольорового в відтінки сірого. Після цього застосовується локальна порогова фільтрація за допомогою функції `threshold_local()` з бібліотеки `skimage.filters`. Цей метод застосовує порогову обробку до кожного пікселя зображення на основі локальної статистики (розміру сусіднього вікна та відступу), що дозволяє покращити контраст між текстом і фоном на відсканованому зображенні.

Параметри функції `threshold_local()`: `warped` (вхідне зображення, яке буде оброблено), `11` (розмір вікна для локальної фільтрації), `offset=10` (параметр корекції порогу, який може бути змінений для адаптації до різних типів зображень), `method='gaussian'` (метод для обчислення порогу).

Після застосування порогової обробки зображення стає бінарним: пікселі, які перевищують поріг, стають білими, інші – чорними. Це дозволяє досягти ефекту, який часто застосовується при роботі з цифровими зображеннями для виділення різних елементів на них.

Запропонована реалізація має ряд переваг, проте для підвищення ефективності можна проводити корекції параметрів роботи по замовчуванню, наприклад при обробці зображень великого розміру. Хоча програма зменшує розмір зображення до 500 пікселів по висоті, що дозволяє значно прискорити обробку, у випадку з більшими чи високоякісними зображеннями можливе значне зниження точності виявлення контурів або візуалізації через масштабування. Це може призвести до втрати інформації, особливо якщо деталі зображення мають значення для подальшої обробки. Іншим важливим обмеженням є залежність від попереднього налаштування порогових значень. Проте незважаючи на перелічені недоїлки в програмі реалізований елемент налаштування параметрів роботи як в ручному так і в автоматичному режимі, що дозволяє як підвищити загальний рівень точності опрацювання зображень так і проводити аналіз зображень різного типу та складності.

3.3 Тестування та аналіз реалізованого програмного модуля

Оцінка функціональних можливостей розробленого програмного забезпечення була проведена на основі ряду тестів з використанням реальних цифрових зображень. Для отримання об'єктивного результату тестування програмного засобу необхідно забезпечити дотримання двох параметрів: апартна складова робочої станції не повинна нести вирішальний вплив на швидкість опрацювання зображення та тестова вибірка повинна в повній мірі відображати можливу варіативність цифрових зображень відповідного типу.

Для тестування було обрано робочу станцію з середніми ціновими та функціональними характеристиками. Конфігурація комп'ютера, що використовувався на етапі завершального тестування приведена на рисунку 3.6.

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 3.6 – Параметри робочої станції для тестування

Обрана конфігурація відповідає параметрам офісного комп'ютера, що виконує завдання середньої складності. Такі комп'ютери є поширеними, тому можна стверджувати, що отримані результати можна повторно отримати в використання загальнодоступних технологій.

Для максимального охоплення варіативності контурів об'єктів було виділено ряд критеріїв, які будуть однозначно відрізняти контури різних груп. Серед запропонованих критеріїв є:

- рівень зашумленості зображення (кількість областей на яких відбуваються різкі скачки рівня яскравості);
- чіткість границі (рівень відмінності рівня значення кольору між сусідніми точками на зображенні);
- рівень дотикання/накладання об'єктів (відносна кількість об'єктів, які мають точки які можна віднести як до одного так і другого контуру);
- складність контуру (кількість апроксимуючих ліній, які необхідні для опису контуру об'єкта з заданим пороговим значенням).

Да основі даного набору характеристик було сформовано ряд груп зображень які були використані для тестування програмної системи та оцінки ефективності її роботи під час опрацювання складних зображень.

Тестові зображення були розбиті на наступні тестові групи:

- Прості зображення. Зображення даної групи містять невелику кількість об'єктів, які легко візуально ідентифікуються. Контурні лінії окремих елементів не пересікаються, границі чітко видимі та легко прослідковуються. Об'єкти мають форму близьку до простих геометричних фігур. Під час апроксимації в основному використовуються прямі лінії та криві першого порядку.

- Звичайні зображення. Для зображень цієї групи характерним є велика кількість елементів на них, незначна кількість об'єктів може дотикатись/накладатись зі своїми сусідами, контури більшості елементів можна чітко виділити та апроксимувати. Зображення можуть мати деякий рівень зашумленості. До даної групи зображень відносяться зображення отримані в реальних умовах.

- Складні зображення будуть містити велику кількість подібних елементів, що важко розрізняються навіть візуально, форма контурної лінії є складною, має велику кількість додаткових відхилень та виступів. Границі між окремими елементами мають мінімальні відходження, що утруднює її виділення. На зображенні присутні області зашумленості, які можуть покривати більшу частину вхідного зображення.

Аналіз отриманих в процесі тестування результатів показав, що розроблений програти додаток у повній мірі виконав усі поставлені завдання. При обробці складних зображень програма видавала для користувача запити для встановлення параметрів своєї роботи, проте при обробці простих та свичайних зображень робота проводиться в автоматичному режимі. Час опрацювання зображень різних груп відрізнявся мінімально, основні часові витрати виникали в результаті, коли програмна система очікувала на підказки оператора для продовження процесу аналізу. Кількість об'єктів на вхідному зображенні мав невеликий вплив, оскільки основна кількість операцій залежала від розмірів зображень, що напряму впливало на кількість точок до яких необхідно було застосувати фільтри. Основним параметром який впливав на результати виділення границь був рівень шуму. Присутність на зображенні великої кількості шумів фактично унеможливило коректне виділення контурів (рисунки 3.7)

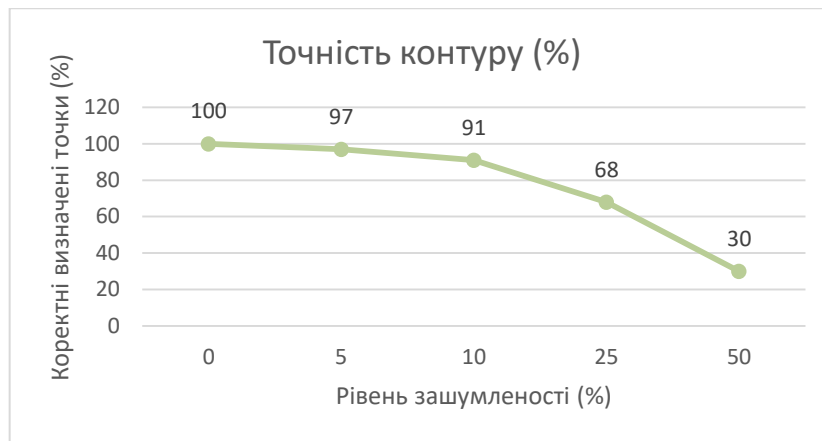


Рисунок 3.7 – Графік залежності точності виокремлення контурних ліній від рівня зашумленості цифрового зображення

В результаті проведеного моделювання, тестування та порівняння з аналогічними системами можна зробити наступні висновки:

- запропонований алгоритм є простим та швидким засобом виділення та опису елементів на цифрових зображеннях;
- програмна система працює з різними типами зображень та може працювати як в ручному так і автоматичному режимах.

3.4 Висновки до розділу

Запропоновано внутрішню архітектуру програмного додатку обробки цифрових зображень та здійснено її дослідження на основі проведення моделювання, що дозволило провести оцінку протікання внутрішніх процесів в середині системи та описати рівні доступу користувачів до функціоналу системи.

Здійснено програмну реалізацію системи обробки цифрових зображень на основі запропонованого алгоритму виділення та опису об'єктів на цифрових зображеннях, що дозволило провести порівняння розробки з програмами аналогами.

ВИСНОВКИ

На основі теоретичних та практичних результатів, які були отримані під час виконання задач кваліфікаційної роботи можна зробити наступні висновки:

1. Проведено аналіз та здійснено класифікацію цифрових растрових зображень на основі дослідження їх характеристик, що дозволило обрати цифрових зображень для програмної реалізації системи обробки зображень та провести класифікацію цифрових зображень.

2. Досліджено методи та алгоритми теорії контурного аналізу на основі їх функціональних можливостей виділення границь на цифрових зображеннях, що дозволило вибрати набір фільтрів та алгоритмів кодування для опису об'єктів на цифрових зображеннях.

3. Проведено аналітичний огляд програмних систем опрацювання цифрових зображень на основі аналізу їх функціональних можливостей та архітектурних рішень, що дозволило виділити основні блоки для систем обробки цифрових зображень.

4. Проведено аналізу алгоритмів виділення та опису цифрових зображень на основі використання контурного аналізу, що дозволило виділити групу операторів підкреслення границь цифрових зображень для подальшої їх інтеграції в систему опрацювання цифрових кольорових зображень.

5. Розроблено алгоритм опису об'єктів цифрових зображень на основі аналізу контурної лінії, що дозволило програмно реалізувати додаток опису елементів на цифрових зображеннях шляхом виділення границь об'єктів на цифрових зображеннях.

6. Здійснено програмну реалізацію системи обробки цифрових зображень на основі запропонованого алгоритму виділення та опису об'єктів на цифрових зображеннях, що дозволило провести порівняння розробки з програмами аналогами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Жеребний О.В. , Тропак О.Ю. Алгоритми контурного аналізу для опису та кодування об'єктів цифрових зображень. X Всеукраїнська науково-практична конференція «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ'24). 05 листопада 2024, Тернопіль, Україна. Тернопіль: ЗУНУ, 2024.
2. Жеребний О.В. , Тропак О.Ю. Аналітичний огляд технологій розпізнавання рукописних текстів. X Всеукраїнська науково-практична конференція «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ'24). 05 листопада 2024, Тернопіль, Україна. Тернопіль: ЗУНУ, 2024.
3. Дубчак Л. О., Гураль І. В. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / ред. О. М. Березький. Тернопіль : ТНЕУ, 2019. 33 с.
4. 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" / ред. О. М. Березький. Тернопіль: ЗУНУ, 2020. 32 с.
5. Соколенко Д. Г., Корнага Я. І. “Система розпізнавання писемних символів за допомогою нейронної мережі”, Вчені записки ТНУ імені В.І. Вернадського. Серія Технічні науки. Том 29 (68) Ч. 2 № 5 2018 с. 56-58
6. Карпович А. В. “Використання згорткових нейронних мереж для задачі класифікації текстів”, International scientific journal «Internauka», № 14(54), 2018 с. 69-73.
7. Content Based Image Retrieval Using Image Preprocessing Techniques. Strad Research. 2020. Vol. 7, no. 12.
8. Contour enhanced image super-resolution / L. Kong et al. Journal of Visual Communication and Image Representation. 2022. P. 103659.

9. Depth image restoration method based on color image contour / D. SU et al. Chinese Journal of Liquid Crystal and Displays. 2021. Vol. 36, no. 3. P. 456–464.
10. Dixit U. D., Shirdhonkar M. S. Preprocessing framework for document image analysis. International journal of advanced networking and applications. 2019.
11. Masters T. Image Preprocessing. Deep Belief Nets in C++ and CUDA C: Volume 2. Berkeley, CA, 2018. P. 85–101.
12. Prem Jacob T, Polakam Sukanya, Thatiparthi Madhavi. Brain Tumor Discovery Using Image Preprocessing. International Journal of Research in Pharmaceutical Sciences. 2020. Vol. 11, no. 4. P. 6870–6875.
13. Rasche C. Rapid contour detection for image classification. IET Image Processing. 2018. Vol. 12, no. 4. P. 532–538.
14. Vazquez-Corral J., Bertalmio M. Angular-Based Preprocessing for Image Denoising. IEEE Signal Processing Letters. 2018. Vol. 25, no. 2. P. 219–223.
15. M. Abadi, P. Barham, J. Chen et al., "TensorFlow: a system for large-scale machine learning" in Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16), Berkeley, CA, USA:USENIX Association, pp. 265-283,
16. S. Shi, Q. Wang, P. Xu and X. Chu, "Benchmarking State-of-the-Art Deep Learning Software Tools", 2016 7th International Conference on Cloud Computing and Big Data (CCBD), 2016, pp. 99-104,
17. Y. Kochura, S. Stirenko, O. Alienin, M. Novotarskiy and Y. Gordienko, "Comparative analysis of open source frameworks for machine learning with use case in single-threaded and multi-threaded modes", 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 2017, pp. 373-376,
18. Z. Yang, Y. "Parallel Image Processing Based on CUDA", 2008 International Conference on Computer Science and Software Engineering, 2008, pp. 198-201.
19. I. K. Park, N. Singhal, M. H. Lee, S. Cho and C. Kim, "Design and Performance Evaluation of Image Processing Algorithms on GPUs", IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 1, Jan. 2011, pp. 91-104.

- 20.J. Lawrence, J. Malmsten and A. Rybka, Comparing TensorFlow Deep Learning Performance Using CPUs GPUs Local PCs and Cloud, 2017.
- 21.S. Bahrampour, N. Ramakrishnan "Study of Caffe Neon Theano and Torch for Deep Learning", CoRR, vol. abs/1511.06435, 2015.
- 22.K. Kavitha "A Hybrid Biometric Authentication Algorithm", International Journal of Engineering Trends and Technology, no. 3, 2012, pp. 311-319.
- 23.P. Suja and Shikha Tripathi, "Real-time emotion recognition from facial images using Raspberry Pi II", international conference on Signal Processing and Integrated Networks (SPIN) , 2016., pp. 666-670
- 24.Urvashi Bakshi and Rohit Singhal, "A survey on face detection methods and feature extraction techniques of face recognition", International Journal of Emerging Trends Technology in Computer Science (IJETTCS), no. 3, 2014, pp. 233-237.
- 25.M. Saranya, S. Padmavathi "Face Tracking in Video by Using Kalman Filter", International Journal of Engineering Research, no. 6, 2014, pp. 54-58.
- 26.Liton Chandra Paul and Abdulla AlSumam, "Face recognition using principal component analysis method", International Journal of Advanced Research in Computer Engineering Technology (IJARCET), no. 9, 2012, pp. 135.
- 27.Jie Yang, Hua Yu and William Kunz, "An efficient LDA algorithm for face recognition", Proceedings of the International Conference on Automation Robotics and Computer Vision ICARCV, 2000, pp. 34-47.
- 28.Hari Kumar Singh, Prashant Kr Maurya, Khushboo Singh and Pooja Singh, "Discrete cosine transform and discrete fourier transform of RGB image", International Journal of Computer Applications, 2012.
- 29.Andre Teixeira Lopes et al., "Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order", Advances in Pattern Recognition, pp. 610-628, 2017.
- 30.A.K. Kushsairy "Enhancing embedded platform for face detection system", Instrumentation and Measurement Technology Conference Proceedings, pp. 1-5.

31.A.K. Sandeep, S. Nithin and K.I. Ramachandran, "An Image Processing Based System for Driver Assistance", International Journal of Computer Technology and Applications (IJCTA), pp. 7369-7375, 2016.

32.Subham Mukherjee et al., "Convolutional Neural Network based Face Detection", International Conference on Electronics Materials Engineering and Nano Technology, 2017.

33.Jayasudha, "Facial Expression Recognition Using Neural Network and Its Correlative Process under Video Analytics", the International of Advanced Research in Computer Science and Software Engineering, 2014.

34.Vikas Maheshkar, Sushila Kamble, Suneeta Agarwal and Vinay Kumar Srivastava, "DCT-based reduced face for face recognition", International Journal of Information Technology and Knowledge Management, vol. 5, no. 1, pp. 97-100, 2012.

35.Enrique G. Ortiz and Brian C. Becker, "Face recognition for web-scale datasets", Computer Vision and Image Understanding, vol. 188, pp. 153-170, 2014.

36. Moyano, J.; Nieto-Julián, J.E.; Bienvenido-Huertas, D.; Marín-García, D. Validation of Close-Range Photogrammetry for Architectural and Archaeological Heritage: Analysis of Point Density and 3D Mesh Geometry. 2020, 12, 3571.

37.Lauria, G.; Sineo, L.; Ficarra, S. A Detailed Method for Creating Digital 3D Models of Human Crania: An Example of Close-Range Photogrammetry Based on the Use of Structure-from-Motion (SfM) in Virtual Anthropology. 2022, 14, 42.

38.Murtiyoso, A.; Pellis, E.; Grussenmeyer, P.; Landes, T.; Masiero, A. Towards Semantic Photogrammetry: Generating Semantically Rich Point Clouds from Architectural Close-Range Photogrammetry. Sensors 2022, 22, 966.

39.Iglhaut, J.; Cabo, C.; Puliti, S.; Piermattei, L.; O'Connor, J.; Rosette, J. Structure from Motion Photogrammetry in Forestry: A Review. 2019, 5, 155–168.

40.Shang, H.; Liu, C.; Wang, R. Measurement Methods of 3D Shape of Large-Scale Complex Surfaces Based on Computer Vision: A Review. Measurement 2022, 197, 111302.

41.Ahn, S.J.; Rauh, W.; Kim, S.I. Circular Coded Target for Automation of Optical 3D-Measurement and Camera Calibration. Pattern Recogn. 2001, 15, 905–919.

- 42.Chen, L.; Rottensteiner, F.; Heipke, C. Feature Detection and Description for Image Matching: From Hand-Crafted Design to Deep Learning. 2021, 24, 58–74.
- 43.Sharma, S.K.; Jain, K.; Shukla, A.K. A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching. *Appl. Sci.* 2023, 13, 6015.
- 44.Remondino, F. Detectors and Descriptors for Photogrammetric Applications. *International Archives of the Photogrammetry. Remote Sens.* 2006, 36, 49–54.
- 45.Forero, M.G.; Mambuscay, C.L.; Monroy, M.F.; Miranda, S.L.; Méndez, D.; Valencia, M.O.; Gomez Selvaraj, M. Comparative Analysis of Detectors and Feature Descriptors for Multispectral Image Matching in Rice Crops. *Plants* 2021, 10, 1791.
- 46.Liu, W.C.; Wu, B. An Integrated Photogrammetric and Photoclinometric Approach for Illumination-Invariant Pixel-Resolution 3D Mapping of the Lunar Surface. *ISPRS J. Photogramm. Remote Sen.* 2020, 159, 153–168.
- 47.Karami, A.; Menna, F.; Remondino, F.; Varshosaz, M. Exploiting Light Directionality for Image-Based 3d Reconstruction of Non-Collaborative Surfaces. *Photogramm. Rec.* 2022, 37, 111–138.
- 48.Tang, C.H.; Tang, H.E.; Tay, P.K. Low Cost Digital Close Range Photogrammetric Measurement of an As-Built Anchor Handling Tug Hull. *Ocean Eng.* 2016, 119, 67–74.
- 49.Liu, Y.; Su, X.; Guo, X.; Suo, T.; Yu, Q. A Novel Concentric Circular Coded Target, and Its Positioning and Identifying Method for Vision Measurement under Challenging Conditions. *Sensors* 2021, 21, 855.
- 50.Kniaz, V.V.; Grodzitskiy, L.; Knyaz, V.A. Deep Learning for Coded Target Detection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 2021, 44, 125–130.