**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Західноукраїнський національний університет**

**Навчально-науковий інститут новітніх освітніх технологій**

Кафедра комп’ютерної інженерії

**Пастернак Віктор Мирославович**

**«Алгоритм підкреслення контурів об’єктів на основі об’єктно-орієнтованого підходу/** **Algorithm for tracing the objects contours based on an object-oriented approach»**

спеціальність: 123 - Комп’ютерна інженерія

освітньо-професійна програма - Комп’ютерна інженерія

Кваліфікаційна робота

Виконав студент групи КІзм-21

В.М. Пастернак

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Науковий керівник:

к.т.н., доц. Ю.М. Батько

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кваліфікаційну роботу допущено

до захисту:

"\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р.

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Л. О. Дубчак

**Тернопіль – 2022**

# Вступ

Актуальність роботи. Технологія обробки цифрових зображень є результатом швидкого розвитку науки й техніки, яка застосовувалася в багатьох галузях. Обробка зображень – це підмножина цифрової обробки сигналів, яка має різні застосування та переваги в різних сферах. Цифрова обробка – це фактично цифрова обробка зображень, яку можна виконати за допомогою інформатики, програмування та штучного інтелекту. Обробка зображень – це один із додатків і підмножин штучного інтелекту, який, як випливає з назви, обробляє цифрові зображення та відображає певний результат із конкретною інформацією на основі попередньо визначеного навчання. Сьогодні застосування технології обробки зображень у різних галузях науки, техніки викликало велику увагу з метою розширення можливостей артеріального інтелекту в різних інженерних завданнях.Технологія цифрової обробки зображень включає багато практичних технологій, і технологія сегментації зображень є однією з них. В даний час широко поширена технологія розпізнавання образів реалізована за допомогою технології сегментації зображення. Виявлено, що сегментація зображень має хороше застосування в автоматичному розпізнаванні номерних знаків, біомедичній інженерії, дистанційному зондуванні та запобіганні та виявленні пожеж.

Виявлення країв є однією з найбільш часто використовуваних операцій в аналізі зображень і тому в описано літературі велика кількість алгоритмів для покращення та виявлення країв, ніж по будь якому іншому напрямку в цифровій обробці зображень. Причиною цього є те, що краї формують контур об’єкта, а об'єкти – це предмети інтересу в аналізі зображень і системах зору. Границя – це межа між об’єктом і фоном, а також вона вказує на межу між об’єктами, що перекриваються. Це означає, що якщо краї на зображенні можна ідентифікувати точно, всі об’єкти можуть бути правильно розташовані, а основні властивості, такі як площа, периметр і форму можна виміряти. Оскільки комп’ютерний зір передбачає ідентифікація та класифікація об'єктів на зображенні, виявлення країв є необхідний інструмент. Тому задача проектування та створення алгоритму виділення країв об’єктів на цифрових зображення на основі об’єктно-орієнтованого підходу є актуальною.

Метою роботи є розробка алгоритму підкреслення контурів об’єктів на основі об’єктно-орієнтованого підходу.

Для досягнення даної мети ставились наступні завдання:

* провести класифікацію задач обробки цифрових зображень;
* проаналізувати існуючі формати зберігання цифрових зображень;
* провести аналітичний огляд наявних програмних засобів виділення та опису контурів об’єктів на цифрових зображеннях;
* проаналізувати існуючі методи та алгоритми проходження контуром об’єктів на цифрових зображеннях;
* розробити алгоритм підкреслення контурів об’єктів на основі об’єктно-орієнтованого підходу;
* реалізувати програмний додаток виділення та опису об’єктів на цифрових зображеннях, провести тестування та порівняти з програмами-аналогами.

Об’єкт дослідження – процес обробки цифрових зображень.

Предмет дослідження – методи і алгоритми проходження контуром об’єктів на цифрових зображеннях.

Наукова новизна одержаних результатів визначається наступним чином:

* проведено комплексний аналіз та класифікацію алгоритмів підкреслення границь однорідних областей на основі аналіз перепадів яскравості, що надало можливість видрати групу бетекторів границь для реалізації алгоритму виділення границь об’єктів на цифрових зображннях;
* розроблено алгоритм виділення границь об’єктів цифрових зображнь на основі обєднання сусідніх однорідних областей, що дозволило зменшити час опрацювання цифрових зображень та підвищити точність отриманих контурів обєктів цифрових зображень.

Практична цінність одержаних результатів полягає в тому, що:

* розроблено структуру та проведено теоретичне дослідження програмного додатку аналізу та опису об’єктів на цифрових зображеннях, що дозволило в подальшому програмно реалізувати та провести дослідження розроблених алгоритмів;
* реалізовано програмне забезпечення для підкреслення границь об’єктів цифрових зображень на основі об’єктно-орієнтованого підходу та з використанням алгоритмів цифрової обробки зображень.

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

1 Програмні системи аналізу та обробки цифрових зображень

## Цифрова обробка зображень

Обробка цифрових зображень складається з різних технік і методів, пов’язаних із маніпулюванням зображеннями на комп’ютері. Над зображеннями виконуються різні типи операцій, які становлять цифрову обробку зображень.

Зображення в основному є двовимірним сигналом. Сигнальною функцією є *f*(*x*,*y*), де значення *x* і *y* в точці генерує піксель у точці. Зображення в основному є двовимірним масивом, що складається з чисел від 0 до 255. В обробці зображень беруть участь різні фактори. Обробка зображень має кілька основних завдань. На рисунку 1.1 наведено основні групи завдань для яких використовуються алгоритми цифрової обробки зображень та розробляються програмні системи.

Завдання цифрової обробки зображень

Краща оптимізація зображення, що забезпечує ефективне зберігання та передачу інформації

Покращення цифрової інформації, що зберігається

Автоматизація роботи з зображеннями

Рисунок 1.1 – Завдання цифрової обробки зображення

Сьогодні використання комп’ютерів як ефективного інструменту полегшило контроль інженерних систем, збір даних, аналіз даних, обробка даних і навіть прийняття рішень на різних етапах аналізу цифрових даних. Обробка цифрових зображень – це розділ інформатики, який займається обробкою цифрових сигналів, які представляють зображення, зроблені цифровою камерою або відскановані сканером. Обробка зображень тепер називається обробкою цифрових зображень, який потребує знань комп’ютера та обробляє цифровий сигнал, який приймає камера або сканер. Обробка зображень є одним із основних компонентів інтелектуальних систем, де приймаються рішення. Ця обробка цифрових зображень застосовується комп’ютерними системами. Різноманітне застосування обробки зображень у різних технічних, промислових, міських, медичних і наукових галузях зробило її дуже активною темою серед галузей досліджень.

Обробка зображень має дві основні гілки: покращення зображень і машинний зір. Поліпшення зображень включає такі методи, як використання фільтра розмиття та підвищення контрастності для покращення візуальної якості зображень і забезпечення їх правильного відображення в цільовому середовищі (наприклад монітор комп’ютера), тоді як машинний зір має справу з методами. З їх допомогою значення і вміст зображень. У конкретному розумінні обробка зображень – це будь-який тип обробки вхідного сигналу якій відповідає деякій сцені, наприклад фотографії або сцени з фільму. Результатом обробки цифрового зображення може бути зображенням або набором спеціальних символів або пов’язаних із зображенням змінних. Обробка зображень часто відноситься до цифрової обробки зображень, але є також оптична та аналогова обробка зображень.

З роками обробка зображень значно вдосконалилася, і з’явилося багато сучасних комерційних застосувань обробки зображень. Окрім того цифрові системи опрацювання зображень використовуються в різних галузях. Широке розповсюдження програмних систем даного типу повязано з тим, що дані системи можуть виконувати великі обєми розрахунків за короткі проміжки часу при цьому необхідність присутності людей не завжди є обов’язковим. Серед основних сфер де використовуються системи обробки цифрових зображень слід відмітити наступні (рисунок 1.2):

Сфери використання програмних систем обробки цифрових зображень

Розпізнавання образів

Обробка відео

Медичні технології

Комп'ютер/машинний зір

Корекція зображення

Фільтри програмних додатків

Рисунок 1.2 – Сфери застосування систем цифрової обробки зображення

1. Корекція зображення, підвищення різкості та корекція роздільної здатності. Часто користувачі бажають покращити старі зображення. І це можливо в наш час. Масштабування, збільшення різкості, визначення країв, редагування з широким динамічним діапазоном – усе це відноситься до цієї категорії. Усі ці кроки допомагають покращити якість вхідного зображення та зменшити або взагалі видалити стороні шуми. Більшість програм для редагування та коду корекції зображень можуть легко виконувати ці дії. Відновлення зображень є неминучим компонентом цифрової обробки зображень. Це техніка попередньої обробки, яка зменшує (або) усуває шум і розмитість зображення. Шум – це небажана (або) нерелевантна інформація, яка погіршує якість зображення через обмеження в отриманні та передачі зображення. Цифрове зображення може бути забруднене імпульсом під час отримання (або) передачі. Навіть при ідеальних умовах фіксації на цифровому зображенні можуть утворюватись дефекти від фотофіксуючої апаратури.

2. Фільтри додатків для редагування та соціальних мереж. Сьогодні більшість програм для редагування та соціальних мереж пропонують фільтри. Використання системи фільтрів дозволяє підкреслити важливі елелементи зображення або внести в його структуру деякі елементи спотворення. Метою фільтрації є видалення імпульсів, щоб повністю відновити безшумне зображення з мінімальними спотвореннями сигналу. Найвідомішими та найбільш широко використовуваними нелінійними цифровими фільтрами, заснованими на порядковій статистиці, є медіанні фільтри. Середні фільтри відомі своєю здатністю видаляти імпульсний шум без пошкодження країв. Середні фільтри відомі своєю здатністю видаляти імпульсний шум, а також зберігати краї. Ефективне видалення імпульсу часто призводить до зображення з розмитими та спотвореними рисами. В ідеалі фільтрування слід застосовувати лише до пошкоджених пікселів, залишаючи непошкоджені пікселі недоторканими. Беззастережне застосування медіанного фільтра до всього зображення, як це практикується у звичайних схемах, неминуче змінило б інтенсивність і видалило деталі сигналу непошкоджених пікселів. Таким чином, процес виявлення шуму для розрізнення непошкоджених і пошкоджених пікселів перед застосуванням нелінійної фільтрації є дуже бажаним. Адаптивний медіанний фільтр – це «на основі рішень» або «перемикаючий» фільтр, який спочатку визначає можливі зашумлені пікселі, а потім замінює їх за допомогою медіанного фільтра або його варіантів, залишаючи всі інші пікселі незмінними. Цей фільтр добре виявляє шум навіть при високому рівні шуму. Адаптивна структура цього фільтра забезпечує виявлення більшості імпульсних шумів навіть при високому рівні шуму за умови достатньо великого розміру вікна. Існуючі нелінійні фільтри, такі як стандартний медіанний фільтр (SMF), адаптивний медіанний фільтр (AMF), алгоритм на основі прийняття рішень (DBA) і надійний алгоритм оцінки (REA), показують кращі результати при низькій і середній щільності шуму. При високій щільності шуму їх продуктивність погана. Даний механізм має широке використання в сучасних програмних системах по обробці цифрових зображень та широко використовується в різних галузях для підвищення функціональних можливостей. Приклад використання фільтрації для цифрових зображень наведено на рисунку 1.3.

|  |  |
| --- | --- |
|  |  |
| a) | b) |

Рисунок 1.3 – Приклад цифрового зображення (а) та результату використання цифрового фільтру (b)

Фільтри роблять зображення більш привабливим. Фільтри зазвичай являють собою набір функцій, які змінюють кольори та інші аспекти зображення, завдяки чому зображення виглядає інакше. Фільтри – це цікаве застосування обробки зображень.

3. Медичні технології. У галузі медицини обробка зображень використовується для виконання різних завдань, таких як ПЕТ-сканування, рентгенівське зображення, медична КТ, УФ-зображення, обробка зображень ракових клітин тощо (рисунок 1.4).



Рисунок 1.4 – Приклад цифрових біомедичних зображень

Магнітно-резонансна томографія є найбезпечнішою серед рентгенологічних методик. Техніка для отримання двовимірних МРТ-зображень із синтетичним шумом окремого суб’єкта із семантичними областями. Використовуються модальності T1, T2, PD, і кожна модальність містить зображення коронального виду, сагітального вигляду, поперечного вигляду для кількох суб’єктів. Пухлини головного мозку включають всі пухлини всередині центральної епітеліальної протоки. На основі центральної епітеліальної протоки, яка утворюється аномальним і неконтрольованим органічним процесом асоційованого ступеня в головному мозку, в лімфатичній тканині, кровоносних судинах, в межах несучих нервів. Впровадження обробки зображень у галузі медичних технологій значно покращило процес діагностики. Зображення ліворуч є оригінальним зображенням. Зображення праворуч є обробленим зображенням. Як видно з наведеного прикладу (див. рисунок 1.4) оброблене зображення набагато чіткіше та на ньому більш детально видимі та дозволяють провести візуальни аналіз без використання додаткової техніки. Відповідно результати діагностики будуть більш точнішими, що дозволоить обрати оптимальний протокл для лікування пацієнта.

4. Комп'ютерний/машинний зір. Одним із найцікавіших і найкорисніших застосувань обробки зображень є комп’ютерне бачення. Комп’ютерний зір використовується для того, щоб змусити комп’ютер бачити, ідентифікувати речі та обробляти все середовище в цілому. Важливим використанням комп’ютерного бачення є самокеровані автомобілі, дрони тощо. Комп’ютерний зір допомагає виявляти перешкоди, розпізнавати шлях і розуміти навколишнє середовище. Дані механіки набувають все більшого застосування в автомобільній галузі, оскільки дозволяж впроваджувати додаткові механізми контролю дій водіїв та зменшити кількість аварійних ситуацій. В критичних ситуаціях внутрішній комп’ютер автомобіля може проводити додаткові маніпуляції з механізмами управління транспортним засобом. Наприклад, при екстреному тормозінні, внутрішній комп’ютер не дозволяє різко максимально нажати на педаль тормозу, а поступово послаблює її, це дозволяє уникнути ситуації заклинення коліс. Приклад роботи системи автопілота в сучасних автомобілях наведено на рисунку 1.5.

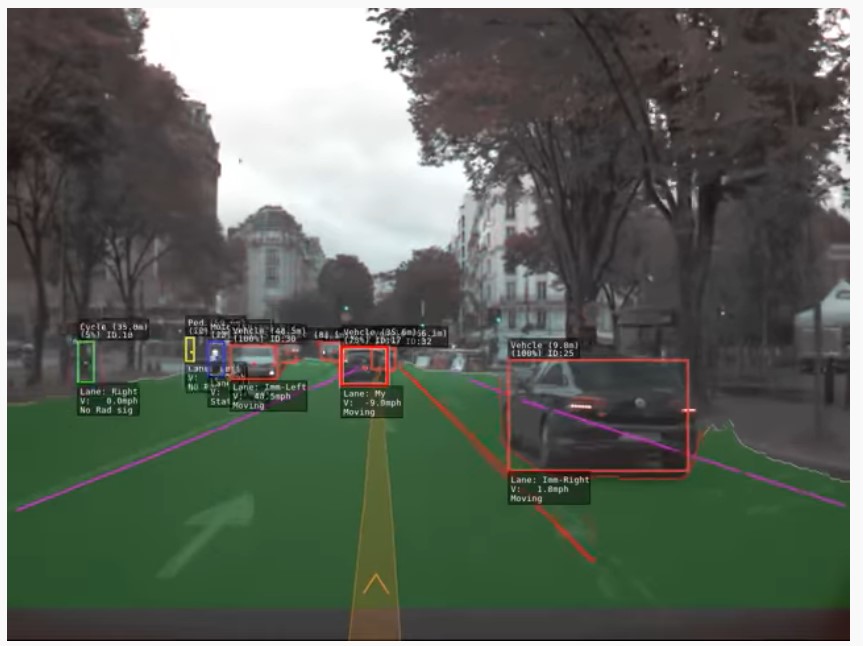


Рисунок 1.5 – Приклад автоматизованого виділення рухомих об’єктів у відеопотоці в режимі реального часу

На прикладі проілюстровано провдення типового аналізу зовнішньої ситуації в системах автопілотів. Вона дозволяє в режимі реального часу визначати перешкоди, проводити їх класифікацію та дозволяє приймати рішення про подільші дії автомоділя на основі цифрової обробки кадру з відепотоку внутрішньої камери. Комп’ютер знімає живі кадри та аналізує інші автомобілі, дорогу та інші перешкоди. Це дозволяє уникати аварійних ситуацій які можуть виникнути у випадку коли водій тимчасово не може примати рішення про подальші дії автомобіля, наприклад проблеми зі здоровям або сторонні фактори які можуть відволікати водія тощо.

5. Розпізнавання образів. Розпізнавання образів є частиною обробки зображень, яка включає штучний інтелект і машинне навчання. Обробка зображень використовується для виявлення різних закономірностей і аспектів у зображеннях. Розпізнавання образів використовується для аналізу рукописного тексту, розпізнавання зображень, комп’ютерної медичної діагностики та багато іншого.

6. Обробка відео. Відео – це в основному швидкий рух зображень. У обробці відео використовуються різні методи обробки зображень. Деякі методи обробки відео – видалення шуму, стабілізація зображення, перетворення частоти кадрів, покращення деталей та багато іншого.

7. Зонування зображення – це поділ пікселів зображення на окремі однакові за змістом ділянки: яскравість, текстуру чи колір, або максимально корельовані. Зонування зображення є необхідною потребою для початку обробки в багатьох програмах опрацювання зображень, таких як медичні системи аналізу зображень, машинний зір, стиснення зображення, аналіз наукових об’єктів.

8. Розпізнавання обличчя є одним із кількох біометричних методів, який має високу точність і відрізняється від інших методів перевірок, користувач може легко вивчити в процес перевірки обличчям. Розпізнавання обличчя за допомогою комп'ютера є однією з найперспективніших галузей біометричних досліджень, яка охоплює різні наукові галузі такі як машинний зір, обчислювальний інтелект, розпізнавання образів і психологія. Тому використання програмного забезпечення, оснащеного системою розпізнавання обличчя людини, може значно збільшити надійність систем контролю руху, особливо на кордонах і пунктах в'їзду в країну. Крім того, розпізнавання облич широко використовується в робототехніці для порятунку людського життя в небезпечних умовах. Кожна система розпізнавання облич складається з чотирьох основних частин:

* отримання зображення;
* попередня обробка;
* обчислення характерних ознак;
* прийняття рішення.

9. Розпізнавання країв є одним із найефективніших і найкорисніших методів обробки зображень, особливо у відокремленні та ідентифікації основного кадру зображення. Існують різні способи виявлення границь на зображенні проте складність даного процесу полягає в тому що зображення може містити або навпаки можуть бути відсутні (погано видимі) різні структурні елементи, як-от втрата вихідних даних зображення та неможливість знайти край на різних кутах тощо. Метою виявлення країв є визначення меж об’єктів на зображенні, які є основою аналізу зображень і машинного зору.

Щоб застосувати методи обробки зображень до різних типів зображень, то зображення розглядають як двовимірний сигнал та застосувують стандартні методи обробки сигналів.

Існує два типи аналогових і цифрових методів обробки зображень. Аналогова техніка використовується для обробки зображень у випадку жорстких версій, таких як друк фотографій. Цифрова обробка зображень використовується для різноманітність програмних систем та завдань, від аналізу супутникових зображень до мікроскопічного контролю розмірів. Обробку зображень можна розділити на три основні категорії:

1. Обробка низького рівня (попередня обробка): включає базову обробку, таку як усунення шумів, фільтрація зображень і контраст.

2. Обробка проміжного рівня (аналіз зображння): Особливістю цього процесу є те, що йому на вхід зазвичай подається цифрове зображення, а на його виході є атрибутами об'єктів зображення, такі як краї, контури та розпізнавання об'єктів.

3. Обробка високого рівня: цей процес передбачає розуміння зв’язку між виявленими об’єктами, висновок та інтерпретація сцени, а також виконання інтерпретацій і діагностики, які виконує зорова система людини.

## Цифрові зображення, класифікація та основні характеристики

Зображення – це двовимірне зображення об’єктів у реальній сцені. Зображення дистанційного зондування – це зображення частин земної поверхні, які видно з космосу. Зображення можуть бути аналоговими або цифровими. Цифрове зображення – це двовимірний масив пікселів (рисунок 1.6). Кожен піксель має значення інтенсивності (представлене цифровим числом) і адресу розташування (на яку посилаються номери рядків і стовпців).

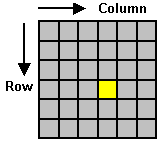


Рисунок 1.6 – Схема організації доступу до елементів цифрового зображення

Цифрове зображення складається з двовимірного масиву окремих елементів зображення, які називаються пікселями, розташованих у стовпцях і рядках. Кожен піксель представляє область на поверхні. Піксель має значення інтенсивності та адресу розташування в двовимірному зображенні.

Значення інтенсивності представляє виміряну фізичну величину, таку як сонячне випромінювання в заданому діапазоні довжин хвиль, відбите від об’єкта, випущене інфрачервоне випромінювання або інтенсивність радіолокатора, розсіяного назад. Це значення зазвичай є середнім значенням для всієї площі сцени, покритої пікселем.

Інтенсивність пікселя оцифровується та записується у вигляді цифрового числа. Завдяки обмеженій ємності зберігання цифрове число зберігається з кінцевою кількістю бітів (двійкових цифр). Кількість біт визначає роздільну здатність зображення. Наприклад, 8-бітне цифрове число знаходиться в діапазоні від 0 до 255 (тобто 2 8 - 1), тоді як 11-бітне цифрове число в діапазоні від 0 до 2047. Виявлене значення інтенсивності потрібно масштабувати та квантувати, щоб воно відповідало цьому діапазону. значення.

Якщо розглядати зображення як відображення кольорів, то зображення можна поділити за такими критеріями:

a) Світло та спектри. Колір є результатом сприйняття світла у видимій області спектру, що знаходиться в області від 400 нм до 700 нм, що падає на сітківку. Видиме світло – це форма електромагнітної енергії, що складається зі спектру частот із діапазоном довжин хвиль від приблизно 400 нм для фіолетового світла до приблизно 700 нм для червоного світла. Більшість світла, яке бачать люди, є комбінацією багатьох довжин хвиль.

b) Основні принципи формування цифрового зображення. Будь-який колір можна поєднати за допомогою відповідних пропорцій трьох складових кольорів, які називаються основними. Найпоширенішими первинними є червоний, синій і зелений.

Визначення кольорового світла:

1. Яскравість або яскравість: це кількість світла, яку отримує око незалежно від кольору.

2. Відтінок: це переважаючий спектральний колір у світлі.

3. Насиченість: це вказує на спектральну чистоту кольору у світлі.

Основні етапи обробки зображень:

* Створення зображення: отримання цифрового зображення.
* Попередня обробка зображення: покращення зображення таким чином, щоб підвищити шанси на успіх інших процесів.
* Сегментація зображення: для поділу вхідного зображення на його складові частини або об'єкти.
* Представлення зображення: для перетворення вхідних даних у форму, придатну для комп’ютерної обробки.
* Опис зображення: щоб виділити ознаки, які призводять до певної кількісної інформації, що цікавить, або ознак, які є основними для диференціації одного класу об’єктів від іншого.
* Розпізнавання зображень: для призначення мітки об’єкту на основі інформації, наданої його дескрипторами.
* Інтерпретація зображення: надати значення множині розпізнаних об'єктів.

Типи цифрових зображень:

* двійкові (бінарні);
* градації сірого;
* кольорові;
* багатоспектральний.

Бінарні зображення. Двійкові зображення є найпростішим типом зображень і можуть приймати два значення, як правило, чорно-білі або 0 і 1. Посилається на бінарне зображення як 1-бітне зображення, оскільки для представлення кожного потрібно лише 1 двійкова цифра піксель. Ці типи зображень часто використовуються в програмах, де необхідна лише інформація про загальну форму або контур, наприклад оптичний розпізнавання символів (OCR).

Бінарні зображення часто створюються із зображень у градаціях сірого за допомогою, a порогова операція, де повертається кожен піксель вище порогового значення білі («1»), а ті, що знаходяться під ним, чорні («0»). На рисунку 1.7 наведено приклади бінарних зображень.

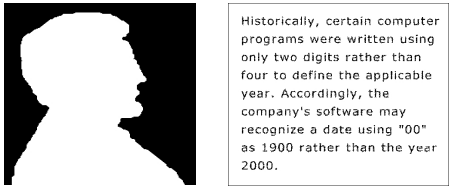


Рисунок 1.7 – Приклади бінарних зображень

Зображення в градаціях сірого називають монохромними (однокольоровими). Вони містять інформацію про рівень сірого, але не містять інформації про колір. Кількість бітів, що використовуються для кожного пікселя, визначає кількість різних рівнів сірого доступний. Типове зображення у градаціях сірого містить дані 8 біт/піксель, які дозволяє мати 256 різних рівнів сірого. Даний формат використовується для передач інформації коли необхідно предати або зберігати інформацію, коли необхідний мінімальний розмір і при цьому все ж таки важливі дрібні деталі. На рисуноку 1.8 наведено приклади зображень у відтінках сірого.

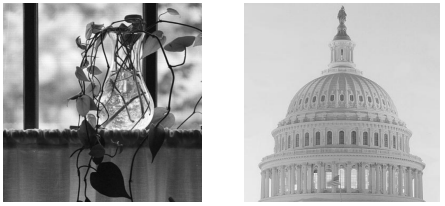


Рисунок 1.8 – Приклади зображень у відтінках сірого

У програмах, таких як медична візуалізація та астрономія, 12 або 16 біт/піксель використовуються зображення. Ці додаткові рівні сірого стануть корисними, коли вони малі частина зображення значно збільшується, щоб розрізняти деталі.

Кольорові зображення можна моделювати як трисмугові дані монохромного зображення, де кожна смуга даних відповідає різному кольору. Фактичний інформація, що зберігається в даних цифрового зображення, є інформацією про рівень сірого в кожній спектральній смузі.

Типові кольорові зображення представлені як червоний, зелений і синій (RGB зображення). Використовуючи 8-бітний монохромний стандарт як модель, відповідне кольорове зображення матиме 24 біти/піксель (8 бітів для кожного з три кольорові смуги: червона, зелена та синя). Дане представлення містить максимальну інформацію про відмінності кольорових відтінків які присутні на на фіксованій сцені. Проте такий формат зберігання інформації вимагає максимального об’єму пам’яті, оскільки для зберігання однієї точки на зображенні необхідно 4 байти пам’яті. На малюнку 1.9 приклад типового кольорового RGB зображення.



Рисунок 1.9 – Приклади кольорового RGB зображення

Стиснення зображення означає відкидання фрагментів інформації та десятків. Що зменшує необхідну ширину смуги пропуску та частоти, що в свою чергу полегшує зберігання зображень та їх передачу. Ступінь стиснення – це число, яке вказує швидкість, з якою інформація відкидається.

Метод jpeg. Ця назва розшифровується як Joint Photographic Expert Group, яка використовується для стиснення нерухомої графіки у форматі цифрового зображення. JPEG – це перший і найпростіший спосіб стиснути зображення, яке намагається стиснути кадр за кадром для рухомих зображень, а для з’єднання цих зображень використовувався Motion JPEG, який мав проблеми.

Метод mpeg. Метод розшифровується як Moving Picture Expert Group, у ньому дані зображення передавалися приблизно з швидкістю 5,1 Мбіт/с, яка використовувалася для відеозображень. Цей спосіб дозволяє зберігати близько 650 Мб даних еквівалентно приблизно 70 хвилинам рухомого зображення на диску. У MPEG інформація надсилається послідовно, разом з керуючими та координатними бітами, які визначають розташування та розміщення інформаційних бітів для запису інформації.

Метод mpeg 2. Цей метод використовує вищий ступінь стиснення, а доступ до даних складає від 3 до 15 Мбіт/с, що використовується в сьогоднішні DVD-диски. Кожен кадр зображення містить декілька рядків цифрової інформації.

Метод mpeg 4. Цей метод використовується для обладнання, яке має справу зі швидкою або повільною передачею даних, але має можливість компенсувати помилки та забезпечити високу якість зображення. У комп'ютерних мережах зображення має бути добре відображатись для всіх користувачів, які працюють з різними швидкостями модемів, що підходить для методу MPEG 4. Основна ідея цього методу полягає в тому, щоб розділити відеокадр на один або кілька предметів, які є виділені за певним правилом, і кожен з них містить сюжет і може копіюватися або передаватися окремо. Корисутвачі можуть використовувати загальні компоненти лише один раз і посилатися на них під час створення теми і навіть можуть створити нову колекцію, об’єднавши теми. Це призвело до гнучкості та широкого використання методу MPEG 4.

Роздільна здатність зображення. У візуальних науках – здатність системи розрізняти деталі зображення в сигналі зображення. Часто цей тип роздільної здатності залежить від розміру пікселів зображення та роздільної здатності зображення та його можна виміряти в одиницях ліній на одиницю довжини.

Якщо критерієм роздільної здатності є кількість пікселів, то на при відображенні однієї і тієїж сцени результати бубуть різнитись (рисунок 1.10).



Рисунок 1.10 – Приклад впливу роздыльноъ здатносты на вызуальне стрийняття сцени

З наведеного прикладу можна зробити наступний висновок, що чим більше пікселів, тим чіткішим буде зображення. Проте слід враховувати, що в такому випадку зображення може суттєво збільшуватись в розмірі. Роздільна здатність стосується розміру найменшого об’єкта, який можна розрізнити на зображенні. У цифровому зображенні роздільна здатність обмежена розміром пікселя, тобто найменший роздільний об’єкт не може бути меншим за розмір пікселя. Внутрішня роздільна здатність системи візуалізації визначається насамперед миттєвим полем зору датчика, яке є мірою площі сцени, яку переглядає один елемент детектора в певний момент часу. Однак ця внутрішня роздільна здатність часто може погіршуватися іншими факторами, які спричиняють розмиття зображення, такими як неправильне фокусування, атмосферне розсіювання та рух цілі. Розмір пікселя визначається відстанню дискретизації.

Зображення "високої роздільної здатності" означає зображення з невеликою роздільною здатністю. Дрібні деталі можна побачити на зображенні з високою роздільною здатністю. З іншого боку, зображення з низькою роздільною здатністю – це зображення з великою роздільною здатністю, тобто на зображенні можна спостерігати лише грубі деталі.

## Програмні засоби аналізу та опису цифрових зображень

Сучасні програмні додатки обробки та аналізу інформації є широко представлені як на Укріїнському так і на світовому ринках. Це можна пояснити тим, що на сьогодні значно збільшились апаратні моливості створення цифрових зображень. Цифрові камери, камери відеоспостереження, мобільні телефони генерують величезні об’єми цифрових зображнь, що в свю чергу збільшує навантаження для програмних систем обробки інформації. Тому про проектуванні майбуюньої системи детекції об’єктів на цифрових зображення необхідно провести аналіз існуючих програмних систем представлених на ринку.

PhotoDirector Essential дає змогу користувачам із будь-яким рівнем досвіду редагувати зображення до студійної якості. Він поєднує в собі функції обробки та організації зображнь Lightroom із потужними інструментами редагування, покращення та зміни Photoshop. Керовані модулі скорочують час опанування даним програмним засобом є мінімальним та допоможе створювати високоякісні ефекти всього за кілька натисків. Це безкоштовне програмне забезпечення для редагування зображень дозволяє групово сортувати та впорядковувати назви тегів, розташування та інші метадані. Його технологія розпізнавання облич на базі штучного інтелекту також допоможе оптимізувати організаційний процес. Коритсувачі можуть комбінувати зображення за допомогою автоматичного створення слайд-шоу. PhotoDirector також містить велику кількість параметрів налаштування фотографій та інструментів для редагування фотографій. Застосовуються маски, фільтрація шарів, використовуються ефекти розмиття, коригування вирівнювання або перетворення 360° зображень на панорамні знімки. Це також чудовий вибір для базових редагувань, таких як додавання тексту або обрізання, обертання чи гортання фотографій. PhotoDirector Essential – найкраще безкоштовне програмне забезпечення для редагування фотографій у списку. Програмний додаток дуже зручний для початківців, але також має всі розширені інструменти редагування для більш досвідчених редакторів. Не кажучи вже про те, що завдяки регулярним оновленням він завжди в авангарді останніх фототенденцій і технологій.

Photoscape X – це одне з найкращих безкоштовних програмних продуктів для обробки розражень для тих, кому потрібно редагувати кілька зображень одночасно. Хоча користувач можете пакетно редагувати зображення за допомогою будь-якого інструменту Photoscape, він постачається зі спеціальними інструментами для найпоширеніших інструментів пакетної обробки, таких як пакетна зміна розміру, пакетна перейменування та пакетна зміна формату. Photoscape X це не лише безкоштовний фоторедактор, але й програма для перегляду зображень, створення GIF-файлів, пристрій для захоплення екрана, конвертер зображень RAW тощо. Він поставляється з декількома автоматичними колірними ефектами, такими як інструмент кольорового сплеску, вибіркове коригування кольору та ефект заміни кольору. Користувачі обирають Photoscape X, тому що він дуже швидкий у використанні. Однак заплутаний макет ускладнює пошук необхідних інструментів, додаючи додатковий час під час редагування фотографій. Ми також виявили, що він не підтримує формати, і він зависав під час обробки файлів RAW. З огляду на це, це дуже надійне програмне забезпечення для безкоштовного редагування фотографій і зручне для початківців, якщо ви витратите час на вивчення макета.

Fotor є одним із найкращих безкоштовних програм для редагування зображень завдяки набору інструментів для налаштування кольорів. Більшість інструментів налаштування кольору доступні безкоштовно, включаючи налаштування тіней, температури, відтінку, яскравості та навіть кілька ефектів кольорових плям. Усі ці інструменти прості у використанні та пропонують повзунки для точного редагування. Для тих, хто віддає перевагу рішення в один клік, багато попередньо встановлених фільтрів Fotor також доступні безкоштовно. Це безкоштовне програмне забезпечення для редагування фотографій просте в навігації, і новачок без проблем знайде потрібні функції. Кожен інструмент, який ви натискаєте, має інформаційне вікно, яке з’являється та пояснює, що це за інструмент і як ним користуватися.

Pixlr's – це безкоштовне, легке безкоштовне програмне забезпечення для редагування зображень від Pixlr добре підходить для початківців і більш досвідчених, яким потрібне швидке рішення для редагування та аналізу зображень. Не маючи повноцінного пакета редагування для пошуку, можна просто використовувати такі інструменти, як пензлі та штампи для відновлення плям, або внести незначні зміни до кольору та освітлення зображення. Спрощений макет Pixlr і мінімалістичний набір інструментів допомагають початківцям спростити процес редагування фотографій. Програмне забезпечення підтримує шари та маски та має кілька основних інструментів для редагування фотографій, таких як настроювані пензлі та візуальні ефекти. Досвідчені користувачі, знайомі з Photoshop, знайдуть панелі інструментів програми знайомими. Серед недоліків даного програмного забезпечення є те, що програма відображається повільно, а це призводить до розчарування, непослідовних рухів маніпулятора миші або сенсорної панелі. Звичайні користувачі можуть насолоджуватися внесенням невеликих змін до своїх зображень або створенням колажів у програмному забезпеченні Pixlr.

BeFunky – найкраще для швидкого та простого редагування. За допомогою десяти модулів опрацювання зображень можна швидко переглядати інтерфейс редагування програмного забезпечення, змінювати макети шаблонів, завантажувати зображення та вносити незначні налаштування. Безкоштовний редактор пропонує багато інструментів, функцій і засобів, необхідних початківцям для покращення зображень, ручного налаштування кольору та освітлення та створення настроюваних колажів або графіки. У безкоштовній версії можна редагувати зображення лише за допомогою приблизно однієї третини інструментів платформи, користувачі-початківці можуть користуватися спрощеною функціональністю.

Даний перелік звісно не є остаточним проте дозволяє зробити висновок про основні архітектури та функціональні рішення які використовуються при реалізації програмних систем даного типу. В основі більшості архітектур знаходиться центральне ядро, а всі інші функції винесені в окремі модулі які мають властивість комбінуватись в залежності від поставленої задачі.

## 1.4 Постановка задач дослідження

Під час проведених аналітичних досліджень в даному розділі було проілюстровано результати аналізу завдань які вирішуються засобами цифрової обробки зображень, а також сфери та реальні приклади їх використання. Досліджено внутрішні структури та принципи створення цифрових зображень та проведено їх класифікацію відносно кількості інформації про кольорову складову на кожного пікселя на цифровому зображенні та проведенно їх класифікацію на основі основі кількості інформації про кольоровий компонент. Досліджено програмні засоби для виділення та опису об’єктів на цифрових зображеннях, виділено основіні архітектурні та функціональні рішення, що застосовуються під час реалізації програмних систем даного типу.

Для досягнення поставленої мети необхідно розв’язати наступні задачі.

* провести класифікацію задач обробки цифрових зображень;
* проаналізувати існуючі формати зберігання цифрових зображень;
* провести аналітичний огляд наявних програмних засобів виділення та опису контурів об’єктів на цифрових зображеннях;
* проаналізувати існуючі методи та алгоритми проходження контуром об’єктів на цифрових зображеннях;
* розробити алгоритм підкреслення контурів об’єктів на основі об’єктно-орієнтованого підходу;
* реалізувати програмний додаток виділення та опису об’єктів на цифрових зображеннях, провести тестування та порівняти з програмами-аналогами.

## 1.5 Висновки до розділу

Проаналізовано основні завдання та сфери застосування механізмів цифрової обробки зображень, що дозволило провести класифікацію основних задач при проектуванні системи опрацювання та опису об’єктів на цифрових зображеннях.

Проаналізовано формати кодування та зберігання цифрових зображень, що дозволило обрати програмні структури даних підвищення швидкодії проходження контуром об’єкта.

Проведено дослідження програмних систем для аналізу та опису об’єктів на цифрових зображеннях, що дало можливість підкреслити основні архітектурні рішення які використовуються під час реалізації систем даного типу.

# 2 Методи та алгоритми виділення контурів об’єктів на зображеннях

## 2.1 Об’єктно-орієнтований підхід в цифровій обробці даних

Об'єктно-орієнтована парадигма вважається природнім і ефективний підхідом для обробки та аналізу зображень. Тим не менш, незважаючи на інтуїтивну зручність, використання об'єктної орієнтації у цій складній сфері не зазнало швидкого зростання, як це сталося в інших сферах, де така парадигма була запроваджена. Стосовно зображень і, загалом, просторової інформації, об’єктно-орієнтований (O-O) підхід в основному використовувався для зберігання та пошуку, щоб подолати типові недоліки реляційної моделі даних у обробці 2-D даних. Представлення об’єкта для графічної інформації використовує той факт, що будь-який просторовий образ має певну форму та структуру та може бути зображена як окремий об'єкт, таким чином захоплюючи структуру даних зображення в ієрархічному порядку. Об'єкти можна створювати на різних рівнях простору, а інформації про ієрархічні зв'язки між об'єктами можна зберігати разом із самими об'єктами. На цій основі моделі ОО можуть бути точно використані, а абстрактні формалізми для представлення просторових знання можуть бути змоделювані. На жаль, велика робота, про яку повідомляється в цій галузі застосування, мало допомагає якщо підходити до зображень з точки зору аналізу та розпізнавання. Тому що при роботі з аналізом зображень і комп’ютерним зором доменів, виникають набагато інші складні питання, ніж ті, що ставляться в базі даних поле.

З одного боку існує своєрідна обробка вимоги до виконання. Кілька типів операцій над зображеннями створюють інші зображення, або структури, пов’язані із зображенням. Прикладом є сегментація, яка може дати початок довільній кількість нових об'єктів, які можуть або не можуть належати до того самого класу вхідних даних об'єкт. Створення об’єктів є частою діяльністю в обробці та аналізі зображень (IP&A), підкреслюючи необхідність знищення об’єктів, ведення записів, контроль зв’язків. Велике значення мають складені об'єкти, а також методи, що працюють на кількох об'єктах одночасно, як, наприклад, розширення структурним елементом, згортка за матрицею чисел, знаходження багатокутника Вороного, визначеного набіру предметів тощо.

Аналіз сцени пов’язаний з обробкою одного або кількох зображень. Однією з кінцевих цілей аналізу є отримання інформації про об’єкти, представлені в сцені: синтез за допомогою процесу міркування, отже, фіксує значення зображення (або послідовності зображення), виконуючи розпізнавання та класифікацію [24]. Більшість сучасних модулів автоматичного розпізнавання є невеликими та призначені для конкретних програм. Особливості об'єктної моделі полягає в тому, що при проектуванні додатків уся інформація розбивається на окремі блоки або класи які описують ту чи іншу частину поставленої задачі. Однак, щоб здійснити розпізнавання чи інший аналіз, виникає потреба у великих базах даних для зберігання моделей об’єктів (зрештоюці прості моделю будуть представляти складні об’єкти), тому швидко зростає потреба в створенні принципі які будуть орієнтовувати в меншій мірі на моделі та бути більш «орієнтованими на дані». Це призводить до зростаючого значення механізми структурування даних, а також дослідження відповідних мовних парадигми для моделювання об’єктів, якими потрібно керувати.

Фактично, коли системи стають більшими, інтеграція кількох різних компонентів системи, таких як моделі знань, процес розпізнавання, мови програмування та бази даних, стає центральною проблемою для дизайнера, ніж у звичайних системах програмного забезпечення. Це відмінний підхід від типового першого покоління систем, в яких не існує відповідності між плоским представленням моделей у файловій системі та абстракції або ієрархії абстракцій, інтерпретовані прикладній програмі. Значна частина роботи в області штучного інтелекту стосувалася структурування людських знань. Різних видів структурування знань і технік представлення були запропоновані раніше:

* логіка першого порядку;
* операції на основі правил;
* семантичні мережі тощо.

Каркасні системи, зокрема, були дуже популярні в штучному інтелекті, оскільки вони дають можливість як організувати інформацію в семантичних мережах і пов’язувати її з будь-яким значенням атрибутів об'єкта, так і виокристати процедури для створення об'єктів, що здатний відповісти на відповідне запитання. Останнім часом парадигма об’єктно-орієнтованого програмування (ООП) змінилася та отримала широке визнання як у розробці програмного забезпечення, проектуванні баз даних та інших сферах. Об'єктно-орієнтована парадигма підтримує іншу версію фреймів під назвою «Класи». класи представляють визначення об’єктів і виділяються «Об’єктами», що представляють, таким чином, екземпляри класу. Обидва знання представляють і проблеми програмної інженерії які присутні в об'єктно-орієнтованому підході, навіть якщо його основні концепції з'являються частково як концепції програмування. ООП підтримує моделі сутностей реального світу на різних рівнях абстракції, таким чином інтерес для більшості програм, орієнтованих на дані, які сильно пов’язані з моделюванням реального світу. Даний підхід також допомагає в проектуваннш, впровадженні та обслуговуванні складних систем завдяки підтримці модульності, багаторазового використання коду та структур даних. На відміну від того, що насправді відбувається зі звичайними системами, орієнтованими на записи, об’єктно-орієнтовані бази даних створюють стійкі об’єктні структури та в них використовується семантична інформація. Таким чином, можна використовувати єдину об’єктну модель у мові та в базі даних. Парадигма ООП - це узгоджена основа для конструктора, в якій попередні поняття про інтеграцію можуть бути однозначно формалізовані [6].

Центральними в парадигмі ООП є поняття «Об’єкт», «Клас» і «Нащадок». Загальновизнано, що ці три поняття визначають підмножину мов програмування об'єктно-орієнтовані мови. На сьогодні не завжди є консенсус на даній моделі представлення даних, що підтримує такі базові поняття, тому і мовах програмування теж є різні варіації даного піходу (наприклад, існує різна підтримка об’єкта, використовується концепції: пасивна, автономна або заснована на слотах) і це призводить до різної потужності доступного моделювання. Навіть якщо аналіз і розпізнавання образів природно об'єктно-орієнтовані, однак об'єктно-орієнтовані концепції, представлення, мови і бази даних не мають тут широкого поширення. Проте сучасні системи все більше проектуються на основі гібридних підходів, що об’єднують деякі об’єкти та підходи на основі правил.

Системах обробки зображеннь опис та розпізнавання образів можуть бути засновані на основі парадигм ООП. У результаті це дозволяє поєднання петеваги об'єктно-орієнтованих мов і технологій баз даних із обробкою зображень і розпізнаванням шаблонів, що є важливою перевагою, коли система має працювати з великі набори високоструктурованої інформації (моделі) або система повинна обслуговувати різні програми одночасно. Крім того, це дозволяє реалізацію на слабко пов’язаній багатопроцесорні архітектурі, щоб дозволить значно зменшити вартості розроблених систем.

Щоб зрозуміти та використовувати об'єктно-орієнтоване програмування, необхідно знати такі ключові поняття:

Клас – це фундаментальна одиниця для опису предметів, явищ, подій, яка прокладає шлях до об’єктно-орієнтованого програмування. Це визначений користувачем тип даних, до якого можна отримати доступ і використовувати, створивши екземпляр цього класу. Він має власні члени даних і функції-члени. Клас можна порівняти з планом об'єкта. І функції‑члени, і члени-дані знаходяться в класах. Членами даних всередині класу маніпулюють за допомогою цих функцій-членів.

У момент створення класу опис є першим об'єктом, який визначається. Екземпляр класу існує в об’єкті. Примітно, що система не виділяє жодного простору пам’яті, коли вказано клас, але воно виділяється, коли його екземпляр створюється, тобто коли формується об’єкт. Реальні речі мають спільний стан і поведінку, пару особливостей. Об’єкт приховує свою поведінку за допомогою методів і зберігає свою інформацію в атрибутах.

Інкапсуляція – це процес групування функцій і даних в одну сутність. Щоб отримати доступ до цих елементів даних, область дії функції-члена має бути встановлена на «загальнодоступну», а область членів даних має бути встановлена на «приватну». Відповідно до цієї теорії, предмет містить всю важливу інформацію; тільки невелика підмножина стає доступною для зовнішнього світу. Кожен об’єкт має приватний клас, який містить його реалізацію та стан.

Кілька класів можуть використовувати те саме ім’я методу за допомогою поліморфізму, який також передбачає перевизначення методів для похідних класів. Поліморфізм під час компіляції та поліморфізм під час виконання є двома різними типами поліморфізму. Окрім наявності кількох форм, об’єкти мають спільну поведінку. Щоб уникнути написання повторюваного коду, програмне забезпечення визначатиме, яке використання або значення потрібно для кожного разу, коли використовується об’єкт із батьківського класу.

У найширшому розумінні успадкування означає процес набуття властивостей. Один об’єкт в ООП успадковує властивості іншого. Розробники можуть повторно використовувати загальні функції, зберігаючи при цьому чітку ієрархію, призначаючи зв’язки та підкласи між елементами. Ця характеристика ООП прискорює розробку та забезпечує більшу точність, вимагаючи більш глибокого дослідження даних. Відносини батьків і дітей символізуються через спадщину.

Однією з концепцій ООП є абстракція, яка є актом представлення ключових функцій без включення допоміжної інформації. Це метод проектування абсолютно нового типу даних, що підходить для певної задачі. Він уникає надання сторонніх або безглуздих фактів і відображає лише ту частину, яку запитував користувач. Це вкрай важливо, оскільки це заважає вам виконувати те саме завдання більше одного разу.

Згуртованість класу визначається тим, наскільки тісно та змістовно пов'язані один з одним його методи та властивості, а також тим, наскільки цілеспрямовано вони зосереджені на досягненні єдиної, чітко визначеної мети для системи. Це показник того, наскільки вузькими є обов’язки класу. Оскільки їхні методи та властивості не пов’язані один з одним логічно, класи з низьким рівнем зв’язку важко підтримувати.

Асоціація — це зв’язок між двома різними класами, які встановлюються за допомогою їхніх об’єктів. Можливі асоціації «один до одного», «один до багатьох», «багато до одного» та «багато до багатьох». Асоціація - це зв'язок між двома речами. Різноманітність між об'єктами визначається однією з концепцій ООП. У цій концепції ООП немає власника, і кожен об’єкт має окремий життєвий цикл.

Агрегація зображує зв’язок між об’єктом, який містить інші об’єкти, і є слабкою асоціацією. У цьому методі кожен об’єкт має окремий життєвий цикл. Проте право власності не дозволяє дочірньому об’єкту бути частиною іншого батьківського об’єкта. Це ілюструє зв’язок між компонентом і цілим, де частина може існувати без цілого. Унікальний тип семантично слабкого зв’язку, який називається агрегацією, виникає, коли об’єднуються непов’язані речі.

Композиція - це асоціація, яка зображує зв'язок між частиною та цілим, у якому частина не може існувати без цілого. Агрегація може приймати різні форми, включаючи композицію. Оскільки дочірні об’єкти не мають життєвого циклу, усі вони автоматично зникають разом із батьківським об’єктом. Один об’єкт не може існувати без іншого в будь-якій композиції між двома сутностями. У результаті обидві сутності залежать один від одного за своїм складом.

## 2.2 Методи та алгоритми виділення контурів об’єктів на зображеннях

Сьогодні цифрове відео та зображення створюються з дуже високою швидкістю. Обчислювальна потужність та сучасні комп'ютери дозволяють розробляти нові методи автоматизації процесу аналізу даних. Швидкість отримання корисної інформації з цих цифрових даних дуже важлива. Методи та засоби отримання і обробка цифрових зображень продовжує розвиватися. Відеокамери високої роздільної здатності збільшує кількість контурних точок, це призводить до значного збільшення часу на опрацювання зображень. Опис контурів об’єктів за допомогою кривих різних порядків або масиву точок із заданою похибкою, що фактично визначає рівень відповідності реального контуру та отриманого опису (точність опису) значно скорочує час обробки. Спосіб отримання граничних точок із заданою точністю виділяють об'єкти з різним рівнем деталізації.

Будучи однією з найважливіших частин розпізнавання об’єктів, нса сьогоднішній день методи сегментації та виявлення країв все ще перебувають в стані розвитку і багато науковців працюють в даному напрямі. Існує багато різних методів отримання інформації про контурні точки об'єктів у задачах розпізнавання об'єктів. Один з найпопулярніших з це підгонка кривої на основі граничних точок з використанням методів апроксимації або інтерполяції. Підгонка кривої методом апроксимації, коли деякі точки повинні належати кривій або бути згладженими в деякому представлені.

Використання камери з високою роздільною здатністю з одного боку підвищує якість зображення, а з іншого боку дає більший масив точок для обробки. Підгонка кривих і заміна масиву граничних точок іншими точками дозволяє скоротити час обробки, збільшити швидкість розпізнавання об'єкта. Під час визначення ланцюга граничних точок при методі геометричного моделювання межі об'єкта визначаються як масив вихідних точок який замінює набір приближених точок, що забезпечує необхідну точність опису границь об'єкта для подальшої обробки. Після отримання граничних точок користувачі можуть отримати більше інформації про особливості об’єкта, такі як площа, периметр тощо. Однією з головних переваг є плавність кривої. Виконання наближення межі об'єкта з дугою кола дає набір даних про значення кривизни для еталонних об'єктів. Кривизна є постійною характеристикою, яка не залежить від розташування об'єкта, обертання та інших просторових перетворень, а отже, може використовуватися в задачах розпізнавання. Контури розпізнавання і еталонні об'єкти порівнюються за їх кривизною.

Подання країв зображення значно зменшує кількість даних, які потрібно обробити, але воно зберігає важливу інформацію щодо форм об’єктів у сцені. Це представлення зображення легко включити у велику кількість алгоритмів розпізнавання об’єктів, які використовуються в комп'ютерному зорі разом з іншими програмами обробки зображень. Основна властивість в техніки виявлення краю полягає в його здатності виділити точну лінію краю з хорошою орієнтацією. На сьогодні є багато алгоритмів виділення країв зображень, але з іншого боку, ще немає загального каталогу продуктивності, щоб оцінити продуктивність методів виявлення країв. Завжди оцінюється продуктивність методів визначення краю в цілому та окремо залежно від його застосування.

Виявлення країв є основним інструментом для сегментації зображення. Методи виявлення країв перетворення оригінальних зображень на краєві зображення показує найкращі резельтати при кодуванні зображення у градаціях сірих тонів у зображення. Це легко поястини тим, що при використанні бінарних зображень кількість інформації є мінімальною, що дозволяє хіба що виділити об’єкти які чітко відокремлені на зображенні. У випадку використання кольорових зображень. Навпаки кількість інформацієї є великою, що призводить до додаткових обрахунків під час процесу виділення. У обробці зображень, особливо в комп’ютерному зорі, методи виявлення країв обробляють локалізації важливих змін рівня сірого зображення та виявлення фізичних і геометричні властивості об'єктів сцени. Це фундаментальний процес, який виявляє та окреслює об’єкти та межі між об’єктами та фоном на зображенні. Виявлення краю є найбільш звичним підходом для виявлення значних розривів у значеннях інтенсивності.

Границі – це локальні зміни інтенсивності зображення. Краї зазвичай виникають на межі між двома областями, що мають велику різицю в інтенсивності. Основні риси можна виділити з країв зображення. Виявлення країв використовується як основна функція для аналізу зображень. Ці функції використовуються вдосконаленим систем, що використовують алгоритми комп’ютерного зору. Виявлення країв використовується для виявлення об’єктів, які обслуговують різні програми, наприклад обробка медичних зображень, біометрія тощо. Виявлення країв є активною сферою досліджень, що полегшує аналіз зображень вищого рівня. Є три різні типи розривів рівеня сірого:

* точковий розрив;
* лінійний розрива;
* розриви на границі.

Просторові маски можна використовувати для виявлення всіх трьох типів розриви зображення.

Методи детекції людських облич

Детектор Робертса

Детектор Собеля

Алгоритм LoG

Canny Edge Detection

Детектор Робінсона

Детектор Марра-Хілдрета

Детектор Превітта

Детектор Кірша

Рисунок 2.1 – Класифікація алгоритмів підкреслення країв об’єктів на цифровому зображенні

У літературі існує багато методів визначення країв для сегментації зображення. Найбільший часто використовуватиним є методи виявлення країв на основі розривів. Такими методами є виявлення краю Робертса, виявлення краю Собеля, виявлення краю Превітта, виявлення краю Кірша, виявлення краю Робінсона, виявлення краю Марра-Хілдрета, краю LoG виявлення та Canny Edge Detection (рисунок 2.1).

Детектор країв Робертса. Виявлення країв на основі детектора Робертса введено Лоуренсом Робертсом (1965). Він виконує просту, швидке обчислення, двовимірного вимірювального просторового градієнта на зображенні. Цей метод підкреслює області високої просторової частоти, які часто відповідають краям. На вхід даного оператора передається зображення у градаціях сірого, таке ж, як і на буде отримано на виході. Це є одна з найпоширеніших технік, що використовується на сьогодні. Піксель значення в кожній точці на виході представляють оцінену повну величину просторового градієнта вхідного зображення в цій точці. На зображенні 2.2 Наведено ядро для оператора Робертса.



Рисунок 2.2 – Ядро для оператора Робертса

Детектор країв Собеля (Sobel). Метод виявлення країв Собела був представлений Собелом у 1970 році. Метод Собеля для виявлення країв для сегментації зображення знаходить краї за допомогою наближення до похідної. Він підкреслює краї у тих точках, де є градієнт найвищий. Метод Sobel виконує 2-D просторовий градієнт зображення що виділяє області високої просторової частоти, які відповідають краям. Загалом необхідно знайти розрахункову абсолютну величину градієнта в кожній точці *n* вхідного зображення в градаціях сірого. Реалізація детектора Собеля представляє собою пари складних ядер 3x3. Одне ядро - це просто інше, повернуте на 90o. Це дуже схоже на перехресний оператор Робертса. Приклад ядра для оператора виділення країв Собеля наведено на рисуноку 2.3.



Рисунок 2.3 – Ядро для оператора Собеля

Детектор країв Превітта (Prewitt). Алгоритм виявлення країв Превітта запропоновано Превіттом у 1970 році. При умові, що необхідно оцінити величину та орієнтацію ребер детектор Превітта є одним з найкращих варіантів. Хоч і різні детектори країв градієнта потребує досить тривалих обчислень, щоб оцінити напрямок зміни величини в напрямках *x* і *y*, визначення напрямку за допомогою даного алгоритму дозволяє визначити напрямок безпосередньо з ядра з найвищим відгуком. Обмежується 8 можливими напрямками; проте дослідження показують, що більшість прямих оцінок напрямку не є набагато досконалішими. Цей градієнт на основі детектора краю оцінюється в околиці 3x3 для восьми напрямків. Всі вісім обчислюються за допомогою маски згортки. Потім вибирається одна маска ускладнення, а саме з обчисленим найбільшим модулем. На рисунку 2.4 представлено ядра для використання оператора Превітта для детекції границь об’єктів на зображенні.

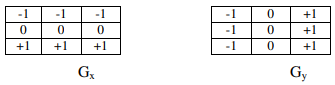


Рисунок 2.4 – Ядро для оператора Превітта

Детектор Превітта трохи простіше реалізувати обчислювально, ніж виявлення Собеля, але це має тенденцію давати дещо більш зашумлених результатів.

Детектор країв Кірша (Kirsch). Виявлення країв Кірша запропоновано Кіршем у 1971 році. Маски цієї техніки Кірша визначаються шляхом розгляду однієї маски та обертання її у восьми основних напрямках компаса: північ, північний захід, захід, південний захід, південь, південний схід, схід і північний схід. На зображенні 2.5 представлено маски які використовуються при роботі з детектором границь Кірша

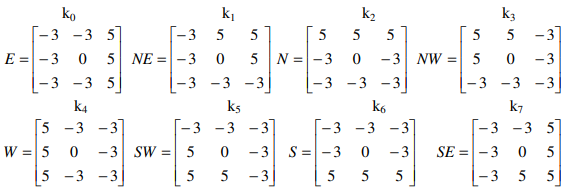


Рисунок 2.5 – Маски для оператора Кірша

Гранична величина визначається як максимальне значення, знайдене згорткою кожної маски на зображення. Напрямок визначається маскою, яка створює максимальну величину. Наприклад, маска *k*0 відповідає вертикальному краю, тоді як маска *k*5 відповідає діагональному краю. Зліт таокж зазаначити, що останні чотири маски фактично такі ж, як перші чотири, але повернуті навколо центральної осі.

Детектор країв Робінсона (Robinson). Метод Робінсона запропонований у 1977 році подібний до масок Кірша, але його легше реалізувати оскільки вони покладаються лише на коефіцієнти 0, 1 і 2. Маски є симетричними щодо своїх напрямних осей та осі з нулями. Потрібно лише обчислити результат на чотирьох масок, а результат з інших чотирьох можна отримати шляхом заперечення результату з перших чотирьох. Приклади масок які використовуються під ча реалізації детектора Робінсона наведено на рисунку 2.6.

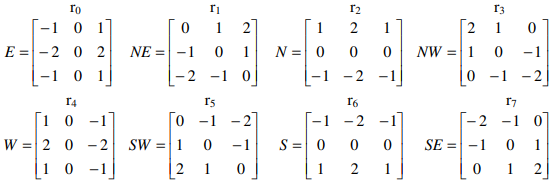


Рисунок 2.6 – Маски для оператора Робінсона

Величина градієнта – це максимальне значення, отримане в результаті застосування всіх восьми масок в околиці пікселя, а кут градієнта можна апроксимувати як кут рядка нулів у масці, що дає максимальний відгук.

Детектор країв Марра-Хілдрета (Marr-Hildreth). Підхід Марра-Хілдрета (1980) – це метод виявлення країв у цифрових зображеннях, який підкреслює безперервні криві скрізь, де є чіткі та швидкі зміни яскравості зображення. Це простий алгоритм працює шляхом згортання зображення за допомогою функції LoG або, як швидкої апроксимації DoGs. Згодом у відфільтрованих результатах виявляються переходи через нуль, що доволяє знайти краї. Метод LoG іноді також називають вейвлетом мексиканського капелюха через форму зображення, що перевернуте догори дном. Алгоритм роботи детектора на основі підходу Марра-Хілдрета включає наступні кроки:

* згладжування зображення за допомогою підходу Гауса;
* використання двовимірної моделі Лапласа до згладженого зображення (часто перші два кроки об'єднані в одну операцію);
* аналіз результату знаходження місць зміни знака. Якщо є зміна знака плюс та нахил зміни знака більший за деякий поріг, то дане місце позначається як край.

Для покращення результатів, можна провести отриманих результат зо дапомогою лапласіана через гістерезис схожий на виявлення країв від Canny, хоча в початкових версіях даний детектор країв не передбачав такої функції. Дане покращення було реалізовано пізніше.

Детектор країв LoG (Laplacian of Gaussian). Лапласіан Гаусса (LoG) був запропонований Марром у 1982 році. LoG зображення *f*(*x,y*) дорівнює похідній другого порядку, та визначений як:

.

Дана формула має два ефекти:

* згладжує зображення
* обчислює лапласіан, що дає двостороннє зображення.

Розташування ребер тоді визначається на основі пошуку нульових перетинів між дублем краю. Цифрова реалізація функції Лапласа зазвичай здійснюється через маску яка проілюстрована на рисунку 2.7.

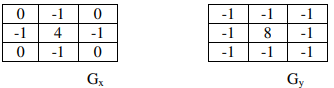


Рисунок 2.7 – Маски для детектора країв LoG

Детектор країв Кенні (Canny) У промисловості метод виявлення країв Canny є одним із стандартних методів виявлення країв під час створення автоматизованих систем. Його вперше створив Джон Кенні для своєї магістерської дисертації в MIT у 1983 році, і він досі перевершує багато новіших алгоритмів, які були розроблені. Щоб знайти ребра за за допомогою детектора Кенні, необхідно спочатку максимально видалити/придушити шуми на зображені. Метод Canny є кращим методом оскільки при своїй роботі він не порушуючи особливості країв об’єктів на зображення дозволяє алгоритму знаходити місця потенційної границі та послідовно віднаходить оптимальний рівень порогу для процесу обробки зображень.

Алгоритму детектора Кенні включає в себе наступні кроки:

* згортання зображення *f*(*r*, *c*) за допомогою функції Гауса, щоб отримати гладке зображення *f*^(*r*, *c*):

*f*^(*r*, *c*)=*f*(*r*,*c*)\**G*(*r*,*c*,6);

* використання оператора градієнта першої різниці, щоб обчислити міцність краю, а величину краю та напрямок отримуютью Дана операції виконуються на основі наступний формул:

,

.

* використання немаксимальноно або критичного придушення до величини градієнта;
* використання порогового значення до зображення немаксимального придушення.

На відміну від операторів Робертса і Собела, оператор Canny не дуже чутлива до шуму.

На сьогоднішній де найкращі результати виділення границь об’єктів можна побачити під час використання аператора Кенні, проте і якість вхідного зображення та тип обєків які розташовані на ньому теж мають суттєве значення.

Властивості контурів можуть допомогти у створенні програм комп’ютерного зору. Наприклад аналізуючи властивості контурів можна легко виконувати ідентифікацію об’єктів для простих об’єктів.

Співвідношення сторін. Першою «розширеною» властивістю контуру є співвідношення сторін. Фактичне визначення співвідношення сторін контуру:

*співвідношення сторін = ширина зображення / висота зображення*

Співвідношення сторін – це просто співвідношення ширини зображення до його висоти. Фігури зі співвідношенням сторін <1 мають висоту, більшу за ширину – такі фігури виглядатимуть більш «високими» та витягнутими. Наприклад, більшість цифр і символів на номерному знаку мають співвідношення сторін менше 1.

А фігури зі співвідношенням сторін >1 мають ширину, більшу за висоту. Сам номерний знак є прикладом об’єкта, який матиме співвідношення сторін більше 1, оскільки ширина фізичного номерного знака завжди більша за висоту.

Нарешті, фігури зі співвідношенням сторін = 1 (плюс або мінус деякі \epsilon, звичайно), мають приблизно однакову ширину та висоту. Квадрати та кола є прикладами форм, які матимуть співвідношення сторін приблизно 1.

Розмір фігури або контуру – це відношення площі контуру до площі обмежувальної рамки:

*екстент = площа форми / площа обмежувальної рамки*

Площа фактичної форми – це кількість пікселів у контурній області. Інакше, прямокутна область контуру визначається обмежувальною рамкою:

*площа обмежувальної рамки = ширина обмежувальної рамки х висота обмежувальної рамки*

У всіх випадках екстент буде меншим за 1 – це тому, що кількість пікселів усередині контуру не може бути більшою за кількість пікселів у обмежувальній рамці фігури.

## 2.3 Алгоритм підкреслення контурів на основі об’єктно-орієнтованого підходу

При розробці підкреслення контурів об’єктів на цифрових зображення з основу було прийнято твердження, що будь який об’єкт чи предмет може бути сформований з набору об’єктів чи предметів меншого розміру які будоть відповідати його структурним частинам. Це в свою чергу дозволяє побудувати ієрархію простик об’єктів, при цьому враховуючи просторове розташування різних складових частин основного об’єкта. Окрім просторових параметрів, додатково враховуються і інші параметри внутрішніх об’єктів, наприклад значення параметру кольру. Іншим важливим параметром для аналізу та опису об’єктів є значення взаємодії між його структурними частинами. В даному контексті можемо розглядати взаємодію як:

* повне входження;
* дотикання;
* обов’язкова відсутність контакту.

Для прикладу розглянемо формування об’єкту світлофор. Його складовими елементами є сам корпус, та світлові ліхтарі, які показують стан світлофора в даний момент часу (рисунок 2.8).

|  |  |
| --- | --- |
|  |  |
| a) | b) |

Рисунок 2.8 – Представлення класу «Світлофор» (а) як об’єднання класів «Корпус» та «Світловий ліхтар» (b)

Дане представлення дозволяє описати об’єкт цілком тільки у випадку якщо на зображенні буду виділено три об’єкти типу «світловий ліхтар» та один об’єкт типу «корпус». При цьому «ліхтарі» повинні бути розташовані вертикально та бути розміщені в середині «корпусу», а також самі «ліхтарі» не повинні дотикатись один до одного. Окремо слід відзначити про встановлення параметру колір в обох типах класів. Для класу «корпус» в даному випадку буде значення – чорний, а для кламу «ліхтар» - червоний, жовтий, зелений, чорний. Окремо слід прописати, що одночасно може бути активований тільки один колір. На сонові даного підходу можна створити опис обдь якого об’єкту складної структури. На основі даної моделі розроблено алгоритм підкреслення контурів на основі об’єктно-орієнтованого підходу. Блок-схема алгоритму наведена на рисунку 2.6.



Рисунок 2.6 – Блок-схема алгоритму підкреслення контурів на основі об’єктно-орієнтованого підходу

Розроблений алгоритм підкреслення та опису контурів на основі моделях опису об’єктів складної структури за допомогою об’єктно-орієнтованого підходу містить наступну послідовність кроків:

1. Завантаження вхідного цифрового зображення.
2. Оцінка вхідного зображення на присутніст зашумленості. Якщо вхідне ображення має велику кількість перепадів рівня яскравості, то проводимо процедуру попередньої обробки зображення на основі фільтрації.
3. Застосування алгоритмів підкреслення контуррів одноріжних областей.
4. Виконання процесу проходження контурами окремих однорідних областей з метою отримання звязного опису границь.
5. Обчислення кількісних характерних ознак одноріжних областей: опис форми, кольору, просторового розміщення на зображенні.
6. На основі отриманих характерних ознак проводиться аналіз та опис рівня взаємодії окремих однорідних областей з сусідніми областями.
7. Формування масиву опису контурів об’єктів які виділенні на зображені на основі масиву однорідних областей та масиву моделей.
8. Послідовно проводиться пошук однорідних областей які можуть відповідати класам які знаходяться на верху ієрархії моделі об’єктів. Якщо не було виявлено об’єкта відповідного класу, то така модель видаляється з подальшого аналізу. Якщо відповідність була встановлена то проводиться послідовна перевірка відповідності структури та внутрішньої їєрархії моделі.
9. Якщо модель повністю співпала з структурою та параметрами одноріжних областей то границя області яка знаходиться на верху ієрархії однорідних областей визнається границею цілого об’єкта.
10. Якщо було проаналізовано усі моделі, то алгоритм виводить отримані результати та завершує свою роботу.

Виділення та опис контурів складних об’єктів є складною задачею. На результат роботи процесу виділення впливає велика кількість сторонніх факторів, які в кінцевому випадку можуть значно вплинути на результат роботи програми. Запропонований підхід на представлені складних об’єктів з набором власних параметрів та з можливістю додаткового аналізу взаємодії з сусідіми областями показав хороші результати роботи та міімальні похибки під чс аналізу цифрових зображень.

Серед плюсів запропонованого алгоритму відносяться:

* можливість аналізу та виділення складних об’єктів на зображенні;
* низькі технічні вимоги до робочих станцій на яких буде використовуватись програмне забезпечення з даним алгоритмом;
* процес реалізації даного алгоритму не є трудоємим та може бути виконаний на любій мові програмування

Недоліки:

* для коректної роботи розробленого алгоритму необхідно попередньо свормувати масив моделей обкремих складних об’єктів.

В результаті моделювання спроектованого алгориму було визначено його переваги та недоліки, а також доведено моливість його використання для детекції об’єктів на цифрових зображеннях.

## 2.4 Висновки до розділу

Проведно аналітичний огляд алгоритмів виділення контурів однорідних областей на основі підкресленні парепадів яксравості, що дало можливість визначити множину детекронів границь для проектування алгоритму виділення контурів складних об’єктів.

Розроблено алгоритм підкреслення контурів на основі об’єктно-орієнтованого підходу, що дало можливість розробити та провести моделювання програмного додатку виділення та опису складних об’єктів на цифрових зображеннях.

# 3 Програмний додаток детекції контурів об’єктів на цифрових зображеннях

## 3.1 Структура програмного додатку підкреслення контурів об’єктів

Виявлення країв є надзвичайно популярним завданням у таких сферах, як комп’ютерне бачення та обробка зображень. Неважко зрозуміти чому люди сприймають основну інформацію про зовнішній світ на основі аналізу зорових об’єктів, тому завжди були залежиними від виявлення країв для таких завдань, як сприйняття глибини та виявлення об’єктів у полі зору.

Мабуть, найпоширенішим застосуванням виявлення країв є розпізнавання об’єктів у комп’ютерному зорі. Щоразу, коли на зображенні присутні об’єкти, між межею об’єкта та фоном за об’єктом буде межа. Ці алгоритми визначають порогові значення, де існує великий градієнт між сусідніми пікселями, що вказує на значну зміну інтенсивності зображення. Навіть сьогодні прості, але потужні алгоритми, такі як Canny edge detection, широко використовуються як етап попередньої обробки для багатьох систем комп’ютерного зору.

Усюди, де системи комп’ютерного зору можуть отримати вигоду від ідентифікації країв (тобто різких змін величини градієнта) то дані алгоритми використовуються завжди. Серед основиних завдань де використовується даний підхід можна завжди знайти розпізнавання окремих елементів зображення. Наприклад розпізнавання відбитків пальців: під час розпізнавання відбитків пальців корисно попередньо обробити зображення, виконавши визначення країв. У цьому випадку «краї» – це контури відбитка пальця, які контрастують із фоном, на якому зроблено відбиток. Це допомагає зменшити шум, щоб система могла зосередитися виключно на формі відбитка пальця. Медична візуалізація: як і розпізнавання відбитків пальців, медична візуалізація є ще однією сферою, де системи комп’ютерного зору можуть підвищити продуктивність шляхом усунення шуму та сторонньої інформації та покращити теплообслуговування за допомогою штучного інтелекту. Медичні зображення чутливі до різних типів шуму під час процесу збору даних. Застосування визначення країв полегшує системам комп’ютерного зору (і лікарям) виявлення аномалій у медичних зображеннях. Виявлення транспортних засобів. При аналізі одного з поширених сьогодні напрямків важливим є виділення та опис об’єктів, що оточують об’єкт керування, а саме безпілотні автомобілі. Ці системи залежать від здатності швидко й автоматично виявляти інші транспортні засоби на дорозі. Ця технологія може допомогти значно зменшити складність зображень, які повинні аналізувати безпілотні автомобілі, зберігаючи при цьому впізнавані форми інших транспортних засобів.

При проектуванні структури програмного додатку підкреслення границь об’єктів було використано модульну архітектуру та об’єктно-орієнтований підхід в програмуванні. Об’єктно-орієнтований підхід на согоднішнй день має ряд переваг які забезпечують спрощену процедуру ствронення та тестування програмних розробок. Серед переваг використання даної технологіє є:

Вирішує проблеми на ранній стадії. Це одна з головних перевага об’єктно-орієнтованого програмування, що полягає в тому, що може ефективно вирішувати проблеми, розділяючись на менші компоненти. Хорошою практикою програмування стає розкладання складної задачі на простіші частини чи компоненти. З огляду на цю інформацію, OOП використовує функцію, яка ділить програмний код на менші, більш керовані частини, розроблені по черзі. Після того, як проблему було розібрано, можна знову використовувати окремі частини для вирішення додаткових проблем. Можна використовувати модулі з тим самим інтерфейсом і деталями реалізації для заміни менших кодів.

Надає переваги дизайну архітектури програмного рішення. Значною подією в інженерії програмного забезпечення стала об’єктно-орієнтована розробка. Серед іншого, вона дозволяє скоротити тривалість розробки та дати фірмам конкурентну перевагу. Перевага дизайну, яку користувачі відчують від ООП, полягає в легкості, з якою вони можуть проектувати та виправляти речі, а також у зменшенні ризиків, якщо такі є. Тут об’єктно-орієнтовані програми вимагають від дизайнерів тривалої та ретельної фази проектування, яка створює кращі проекти з меншою кількістю недоліків. Простіше запрограмувати всі не-ООП незалежно після певного моменту, коли програма натрапила на деякі фундаментальні обмеження.

Знижує фінансові витрати на розробку проекту. Повторне використання програмного забезпечення також знижує ціну розробки. У більшості випадків на об’єктно-орієнтований аналіз і проектування витрачається більше часу та зусиль, що знижує загальну вартість розробки. Загальна вартість удосконалення зменшується, оскільки зазвичай докладається більше зусиль для оцінки конкретної статті та плану. Вартість розробки зазвичай знижується, оскільки більше часу та зусиль витрачається на об’єктно-орієнтований аналіз і проектування.

Взявши до уваги особливості роботи запропонованого алгоритму підкреслення контурів об’єктів на зображенні та базуючись на принципах побудови програмних додатків з модульною структурою було спроектовано архітектуру програмної системи аналізу та опису об’єктів на цифрових зображеннях Спрощену структуру програмного додатку детекції контурів складних об’єктів наведено на рисунку 3.1.

Блок центрального управління програмною системою

Блок завантаження цифрового зображення

Блок попередньої обробки за допомогою фільтрів

Блок детекції контурів областей

Блок проходження та опису контурів однорідних областей

Блок визначення контурів складних об’єктів

Блок визначення параметрів

Блок показу результатів роботи

Рисунок 3.1 – Узагальнена структура програмного додатку для детекції людських облич

На етапі проектування архітектури програмного додатку необхідно враховувати різні фактори майбутнього функціонування програмного додатку. Також слід закладати можливості для модифікації та нарощування фанкціональних можливостей програми в залелжності від поставлених завдань. Даний етап є важливим та вимагає від розробника чіткого розуміння усіх особливостей роботи окремих структурних елементів та принципів обміну інформацією між ними в процесі роботи. Проаналізувавши архітектури програмних систем які присутні на ринку окремі функції та структури даних було сформовано в наступні логічні блоки. Серед основних блоків програмної системи можла відмітити такі:

Блок центрального управління програмною системою. Даний блок був спроектований для вирішення основної задачі управління усіма іншими модулями програмної розробки. Саме тут проводиться перевірка цілісності програми та підключення усіх необхідних для коректного функціонування зовнішніх пристроїв. При виявленні помилкових ситуацій або некоректної роботи окремих структурних модулів або відсутності зовнішніх пристроїв в даному блоці буде згенеровано повідомлення для користувача про це. Окрім того функції даного блоку спробують продовжити роботу програмної системи відключивши доступ до пошкоджених частин програмного додатку. Наприклад при відсутності доступу до віддалених серверів баз даних зображень програма запропонує користувачеві завантажити вхідне зображення з локального носія даних. Іншим прикладом роботи даного блоку може бути відключення функцій попередньої обробки зображення у випадку якщо відсутній файл з масками для проведення фільтрації. Окрім того ще однією задачею даного блоку є переконвертування даних в формт який необхідних для коректного функціонування зовнішніх пристроїв чи стороннього програмного забезпечення, наприклад конвертація зображення у формат який обере користувач для зберігання та пересилки даних. Вхідними значеннями для роботи даного блоку є набір параметрами функціонування та внутрішніх інструкцій для встановлення початкових параметрів роботи програми. Результатом роботи є підготовлена до роботи програмна система обробки та аналізу цифрових зображень.

Блок детекції контурів областей. В даній структурній одиніці зібрано класи та структури даних для провдення етапу підкреслення контурів однорідних областей на основі відомих детекторів границь. Користувач може обрати один з запропонованих детекторів або програма проведе обробку вхідного зображення за допомогою алгоритма виділення границь на основі детектора Кенін. Даний алгоритм був обраний по замовчуванню через те, що на етапі аналізу раніше запропонованих детекторів він має найбільшу точність виділення та найменш відчутний до впливу зовнішніх шумів. Окрім того, користувач матиме можливість порівняти результати роботи різних детекторів, що надасть йому можливість обрати той з них який на його думку максимально коректно провів процес виділення окремих однорідних областей. Результати роботи даного блоку напряму впливають на якість роботи програмної ситеми в цілому, тому функції та парметри його роботи в основному є прихованими від користувачів, щоб зменшити вплив людського фактора. Вхідними даним для роботи даного блоку є цифрові зображення в градаціях сірого, а вихідним – цифровезображення з підкреслиними границями однорідних областей.

Блок детекції контурів однорідних областей

Детектор Кенні

Детектор Превітта

Детектор Собеля

Детектор Кірше

Детектор Робінсона

Рисунок 3.2 – Узагальнена структура блоку детекції контурів однорідних областей

Модуль попередьої обробки цифрових зображень за допомогою фільтрів. В загальному випадку приймаємо за основу, що вхідні зображення можуть бути зроблені не в ідеальних умовах та містити зовнішні шуми які вносять спотворення в форму обнорідних областей. Дані шуми є неінформативними та потребують видалення перед початком аналізу зображення. Для видалення зашумленості використовуються статистичні фільтри. Дані фльтри дозволяють за рахунок усереднення значень пікселів на основі аналізу та обчислення параметрів сусідніх пікселів зменшити перепади яскравості між сусідніми областями. У випадку якщо такі області є невеликого розміру вони злиються з своїми більшими сусідати, чим забезпечеться видалення малоінформативних областей. Під час фільтрації великі однорідні області не зазнають суттєвих змін, оскільки сусідні пікселі і так мають подібні характеристики. В результаті провденої обробки на зображенні зменшується кількість одноріжних областей за рахунок вливання малоінформативних областей в більші за розміром сусідні. При необхідності отримання інформації про дрібні елементи зображння використовуються фільтри іншої групи, які навпаки дозволяють підвищити рівень різниці яскравості на окремих областях. Дана опція використовуєтсья у випадку якщо кінцевий шуканий обєкт має велику кількість областей невеликого розміру. Інший варіант вирішення задачі детекції об’єктів з невеликими внутрішними структурами є повторне створення цифрового зображення проте з більшою роздільною здатністю.

Модуль визначення параметрів – в програмному додатку встановлені параметри роботи по замовчуванню. Вони були обрані експерементальним шляхом та дозволяють отримати результати з високою точністю. Проте якщо результати проведеного аналізу не є достатньо точними, то користовачам надається можливість налаштувати параметри роботи самостійно. Проте дану опцію рекомендуєтсья використовувати користувачам з достатнім досвідом рооти з системами даного типу. Оскільки встановлення некоректних параметрів робти може призвести до помилкового завершення проботи програми, отримання некоректних результатів або втрити даних.

Блок проходження та опису контурів однорідних областей. В результаті роботи детекторів підкреслення контурів на зображення наноситься додаткові мітки які відповідають місцям значного перепада яскравості на зображенні і які описують границі однорідних областей. Проте дані точки не можуть бути використані для подальшого опрацювання оскільки вони є не звязними і в деяких випадках мають розриви між собою. Тому для отримання звязних контурів необхідно пройти процедуру проходження контурами однорідних областей. Складність полягає в тому що дані оласті на границях областей не завжди мають товщину в один піксель і тому потребують використання алгоритмів морфологічного стоншення для отримання більш точного опису границь області. З іншої сторони граничні області можуть мати розриви. Тому додатково необхідно використати функцію морфометричного нарощування. В результаті цих операції контур буде налижений до одноріжного звязного ланцюга. Після чого використовуються алгоритми проходження контуром для послідовного обєднання окремих граничних точок в загальний контру однорідної області.

Блок визначення контурів складних об’єктів. Для реалізації алгоритму підкреслення контурів об’єктів на зображенні використовуються функції послідовної перевірки та порівняння виділених однорідних областей з шаблонами які зберігаються в базі даних моделей. Даний процес є ітеративним та вимагає максимальних часових затрат при роботі з програмою. Це пояснюється тим, що перед початком процесу порівняння програма повинна обчислити характерні ознаки для кожної з однорідних областей, а вже тільки після цього проводити процес порівняння. Для порівняння використовуються такі параметри як колоподібність, площа та колір. Наший набір параметрі може бути збільшений в налаштування програми, проте це збільшить час роботи програмного блоку. При цьому додавання нових характерних ознак не завжди буде впливати на кінцевий результат. Після знаходження співпадіння проводиться процес аналізу областей які граничать з виділеною областю з метою визначення їх відповідності до одраної моделі. Результати даного етапу обробки цифрового зображення напряму залежить від якості та точності опису моделей об’єктів.

Блок завантаження цифрового зображення – даний блок є допоміжним і реалізує функції для отримання зображення як з локальних так і глобальних сховищ даних. В ньому можна обрати формати та розміри вхідних даних. Додатково в даному блоці реалізована можливість отримання зображень з цифрової фотофіксуючої апаратури (камери, сканери). Проте під час другого варіанту отримання цифрових зображень слід враховувати технічні можливості та обмеження апаратних засобів.

Блок показу результатів роботи – даний набір функцій призначений для візуалізації результатів роботи програмної системи у зручному для користувача форматі. Користувач може борати колі для підкреслення обремого об’єкта, відображати обєкти одночасно або послідовно. Результати робти програми можна отримати в табличному представлені, де контур відображається як послідовність пікселів які відповідають границям обєктів.

Використання модульного підходу при проектуванні значно спростило процес реалізації програмної системи. Оскільки розбиття цілісної задачі на окремі складові дозволило розпаралелити процес написання програмного коду. Окремі структурні блоки омжна було редагувати не залежно один від одного, що значно спростило процес тестування та відлагодження роботи програми. Ще однією перевагою використання такої архітектури є можливість використання попередньо розроблени блоків та цифрових бібліотек від відомих компаній. Зовнішні програмні блоки використовували для вирішення допомімжних задач і не мали прямого впливу на кінцевий результат аналізу зображення. Серед задач які були вирішені за допомогою зовнішніх бібліотек функцій можна виділити задачі переконвертації зображень з різних форматів в формат bmp, підключення зовнішніх пристроїв отримання зображень тощо.

Для оцінки запропонованої структури програмного додатку та для аналізу повноти та цілісності роботи в кінцевому результаті було проведено моделювання мйбутньої програмної системи. Для моделювання було використано механізми універсальної мови моделювання (UML). Даний механіз дозволяє провести теоретичну оцінку пограмного додатку яяк в цілому так і окремих структурних елементів. В процесі моделювання було використано ряд діаграм серед яких діаграми прецедентів, послідовності та класів. Даний етап вимагає певних затрат, проте у випадку виявлення критичних помилок процес їх виправлення буде займати менше часу ніж модифікації на етапі програмної реалізації. Процес моделювання має безліч переваг, проте є ряд недоліків, а саме:

* додатковий час на розробку. Оскільки процес моделювання вимагає детального вивчення вхідних даних та аналізу взаємодії між окремими структуриними одиницями, то на його проведення потрібен час. Під час якого фактично не проводить розробка основного програмного продукту.
* можливі втрати інформації. Під час аналізу проводиться розгляд основних вузлів та структур даних можливі ситуації не врахування випадкових ситуацій. Тому під час процесу моделювання оцінюється загальні принципи функціонування програмної розробки, а не враховуються варіанти для окремих випадків. Для усунення даного недоліку можна проводити моделювання не цілої системи, а окремих її складових.
* Фінансові затрати. Як і бульякий проце роботи над програмним продуктом, моделювання потребує залучення фахівців з відповідною кваліфікацією, а також програмно-апаратних засобів моніторингу та моделювання. Все це потребує фінансових витрат які в кінцевому результаті можуть здаватись не виправданими. Проте при виявлені критичних помилок на ранніх етапах фінансові витрати на модифікацію та переробку проекту будуть відчутно меншими ніж у випадку знаходження помилок на завершальних етепах

Результати моделювання за допомогою діаграми прецедентів дозволяють провести технічну оцінку рівня взаємодії між окремими програмними структурами та акторами які будуть використовувати дані методи під час роботи. Додатково розглядається функціональні можливості які отримає користувач при роботі з системою. Приклад діаграми прецеденів наведено на рисунку 3.3.



Рисунок 3.3 – Діаграма прецедентів програмної системи

Результати першого етапу моделювання показали, що реалізовані групи функціональних можливостей у повній мірі надають користувачу засоби для вирішеня поставлених перед нихм завданб. Серед можливих дій користувача є можливість завантаження/створення вхідного цифрового зображення, можливість преегляду результатів обробти програмної системи на окремих етапах роботи, корекція параметрів системи для отримання точніших результатів. Окрім того дуло відмічено приховування деяких функціональних можливостей роботи системи від користувача з метою зменшення впливу людського фактору на кінцевий результат роботи програмної системи. Окрім того було чітко окреслено ті задачі до яких буде мати доступ кожна з груп акторів, набір функцій та обовязків під час роботи з програмою та загальні принципи функціонування. Додатково виділено групу акторів «Користувач» та проаналізовано його можливості по взаємодії з окремими елементами системи.

Наступний етап моделювання – це аналіз запропонованих кроків та їх коректність які необхідно зробити користувачеві для досягнення результатів. Дане модулювання відбувалось за допомогою діаграми послідовностей, яка якраз і дозволяє оціними та виділити основні та додаткові кроки при роботі з програмомю. Результат моделювання наведено на рисуноку 3.4.

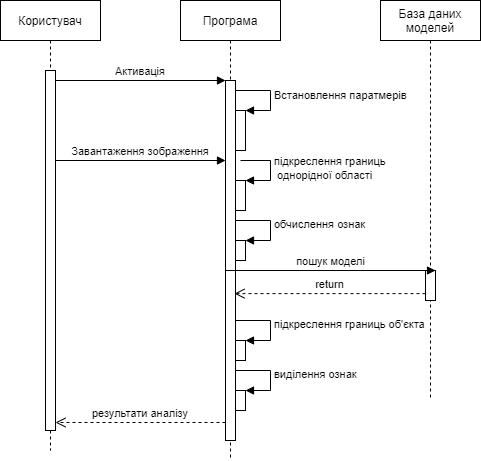


Рисунок 3.4 – Діаграма послідовності програмної системи

Аналіз отриманих результатів на етапі моделювання за допомогою механізме діограми послідовності показав, що запропонована архітектура програмного додатк та реалізований набір сценаріїв є повною та дозволяє досягти очікуваного результати за мінімальну кількість виконаних активностей. Як видно з діаграми в основу робти програмного додатку було покладено лінійний процес проходження обробки зображення. Це з однієї сторони дещо сповільнює процес обробки вхідних зображень, проте в постевленій задачі основним критерієм є все таки точність, а не швидкість роботи, тому такий підхід є допустими. Проте як показали результати тестування програми обробка зображень відбувається без помітних часових затримок, що дозволяє зробити висновок, що даний підхід може використовувати в системах реального часу.

Для оцінки правильності спроектованих класі та стурктур даних було додатково проведена оцінка проекту програмної ситеми за допомогою діаграми класів. Даний тип діаграм дозволяє оцінити внутрішню структуру окремих класів та інтерфейси взаємодії між ними в системі. Результати провденого моделювання наведено наведена на рисунку 3.5

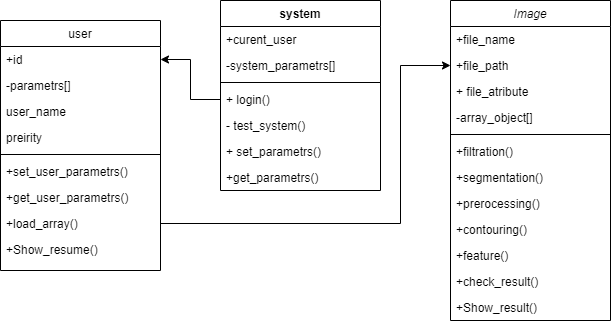


Рисунок 3.5 – Діаграма класів програмної системи

Як показав провдений аналіз, при проектування структури внутріхніх класів системи та розробки набуло їх параметрів та методів вся інформація є чітко структурована, класи не претинаються між собою, мають усі технічні моливості для взаємодії один з одним. Додатково під час аналізу було впровадежно клас “main”, що дозволило провести більш детальну організацію функція які вирішуються в центральному блоці програмної ситсеми.

Моделювання запропонованої архітектури програного додатку, структури внутрішніх класів та типів даних, сценаріїв роботи акремих груп акторів дозволило зробити висновки, що запропоновані рішення дозволяють реалізувати прогармну систем аналізу та опису цифровиз зображень у повній мірі. Невеликі недоліки які були виявлені на етапі моделювання були виправлені і на подальший процес реалізації програмної розробки не мають впливу.

Під час використання програмної системи аналізу та обробки інформації користувачі будуть активно взаємодіяти з нею. Тому для підвищення рівня зручності виокристання програмного додатку було спроектовано та реалізовано дизай графічного інтерфейсу користувача. Під час проектування інтерфейсу був врахований той факт, що основна інформація яка буде надаватись користувачеві для аналізу передається в вигляді цифрових зображень, тому основна частина робочої області програми повинна займати область для відображення зображень. Графічний інтерфейс користувача розробленої програмної системи наведено на рисунку 3.6.

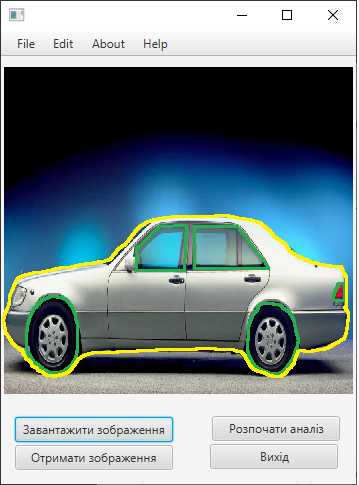


Рисунок 3.6 – Графічний інтерфейс користувача

Дизайн графічного інтерфейсу враховує особливості використання даного програмного продукту. При цьому на головному вікні відображено мінімально необхідний набір структурних елементів, що з однієї сторони не відволікає увагу користувача від аналізу цифрового зображення, а з іншого надає йому можливості для швидкого доступ до візуальних керуючих елементів інтерфейсу. Даний інтерфейс спроектовано в стилі Windows додатків тому щокристувачі можуть швидко опанувати принципи роботи з даною програмою. Окрім того користувачі можкть індивідуально налаштувати параметри візуального відображення в залежності від особистих потреб та вимог до зручнності роботи з програмними системами даного типу.

Підсумовуючи можна сказати, що для повноцінної роботи з системою користувач повинен виконати ряд обов’язкових та допоміжних дій. Дані кроки виконуються послідовно та є інтуєтивно зрозумілими. Для даної послідовності кроків в системі впроваджено елементи допомоги у вигляді підказок які можуть з’являтись під час запуску програмного додатку, а також може бути проілюстрований в процесі роботи програми у вигляді додаткових підказок. Послідовність необхідних дій відображено на рисунку 3.7:

Завантаження зображення

Попередня обробка

Підкреслення границь однорідних областей

Корекція результатів підкреслення границь

Виділення границь об’єктів

Опис отриманих реультатів

Рисунок 3.7 – Послідовність дій користувача при роботі порограмною системо аналізу та опису об’єктів на цифровому зображенні

З наведено вище схеми видно, що основними діями користувача є ввід вхідного зображення, проведення підкреслення границь однорідних областей та подальший опис границь складних об’єктів. Додаткові дії виконуються системою в автоматичному режимі і залучення користувача на даних етапах не є обов’язковою. Прете, контроль додаткових етапів роботи програми дозволить підвищити отримані кінцеві результати.

## 3.2 Програмні модулі додатку обробки та аналізу зображень

Для реалізації окремих функцій роботи програмної системи було використано цифрову бібліотеку OpenCV. OpenCV надає дві прості функції, за допомогою яких можна легко знаходити та підкреслювати контури. Це дві функції:

* findContours();
* drawContours().

OpenCV надає два різні алгоритми, які надають розробникам можливість виділити контури та провести опис їх характеристичних ознак. Ці два алгоритми:

* CHAIN\_APPROX\_SIMPLE;
* CHAIN\_APPROX\_NONE.

Контури можна визначити за допомогою наступних чотирьох кроків:

1. Перетворення зображення у формат градацій сірого. Зображення має бути прочитано та перетворено у формат сірого. Перетворення на відтінки сірого має вирішальне значення, оскільки воно готує зображення до наступного кроку. Зображення має бути перетворено на одноканальне зображення у градаціях сірого для визначення порогу, що необхідно для ефективної роботи техніки визначення контурів.

2. Виконати бінарне порогове перетворення. Віртуальна бінаризація необхідна для спрощення обчислювальноїскладності процесу пошуку порогових значень. У цьому випадку використовується двійкове порогове значення. Це перетворює зображення на чорно-біле, підсвічуючи точки інтересу, щоб полегшити роботу алгоритму визначення контурів. Порогове значення робить рамку об’єкта зображення абсолютно білою з однаковою інтенсивністю для всіх пікселів. За цими білими пікселями програма тепер може розрізняти краї об’єкта.

3. Підкреслення контурів. Для цього завдання використаємо функцію findContours(), щоб виявити контури зображення.

4. Нанесення контурів на вихідне зображення. Для візуалізації отриманих результатів використаємо функцію drawContours(), щоб накласти контури на оригінальне RGB-зображення після визначення контурів.

Зчитування зображення та перетворення його на градації сірого. Для використання функцій з бібліотеки OpenCV необхідно виконати імпорт цієї бібліотеки в проект. І для початку обробки зображення завантажимо його в систему.

import cv2

img = cv2.imread('InputImage.png')

Функція відображення.

def display(mainPhoto, s):

while(True):

cv2.imshow('MyImage', photo)

k = cv2.waitKey(1)

if k == ord('q'):

break

cv2.imwrite(s, mainPhoto)

cv2.destroyAllWindows()

cv2.waitKey(1)

Відображення оригінального зображення.

display(image, "'InputImage.jpeg")

Перетворіть зображення у формат градацій сірого.

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

Метод findContours(). Щоб підкреслити контури, використовувати метод findContours(), який входить в пакет OpenCV. Обов’язково потрібно виконати три обовязкові опції:

Формат зображення: двійкове зображення - вхід.

Режим: це режим пошуку контурів. Режими такі:

* RETR\_EXTERNAL отримує лише крайні зовнішні контури.
* RETR\_LIST отримує всі контури без встановлення будь-яких ієрархічних зв’язків.
* RETR\_CCOMP отримує всі контури та встановлює дворівневу ієрархію.
* RETR\_TREE отримує всі контури та створює повну ієрархію вкладених контурів.

Метод: це метод контурної апроксимації. Цей блог бачитиме як CHAIN\_APPROX\_NONE, так і CHAIN\_APPROX\_SIMPLE для збереження всіх контурів.

contours, hierarchy = cv2.findContours(image=thresh, mode=cv2.RETR\_TREE, method=cv2.CHAIN\_APPROX\_NONE)

Змінна contours зберігатиме всі контури бінарного зображення. Наступним кроком є промальовка отриманих контурів на вихідному зображенні; для цього використаємо функцію drawContours(), яка включена в бібліотеку OpenCV.

Метод drawContours(). Метод drawContours() надає функціональні можливості для малювання контурів, які генеруються функцією findContours() на вихідному зображенні. Ця функція приймає чотири обов'язкові аргументи. Нижче наведено чотири аргументи:

* зображення: це вхідне зображення RGB, на якому слід намалювати контур.
* contours: тут відображаються контури, повернуті методом findContours().
* contourIdx: у створених контурах перераховані піксельні координати точок контуру. Дана опція дозволяє обрати позицію індексу з цього списку. Це дозволоить вивести на екран точну точку контуру, яку необхідно вивести на екран. Якщо вказати від’ємне значення, буде намальовано всі точки контуру.
* колір: вказує на колір контуру, який необхідно відобразити контур.

Для прикладу наведемо програмний код для виводу контуру одноріжної області червоним кольором:

image\_copy = img.copy()

cv2.drawContours(image=image\_copy, contours=contours, contourIdx=-1, color=(0, 0, 255), thickness=2, lineType=cv2.LINE\_AA)

Даний функціональний набір дозволяє в повній мірі реалізувати допоміжні функції по попередній обробці цифрових зображень. Це дозволило зосередитись на реалізації запропонованих аллгоритмів та скоротити загальний час розробки програмного проекту.

## 3.3 Тестування та аналіз реалізованого програмного додатку

Для виконання етапу тестування реалізованого програмного продукту було підібрано робочу станцію з технічними параметрами середнього цінового діопазону. Дану конфігурацію було підібрано на основі тих вимог які ставляться до роботи самого програмного продукту:

* технічні параметри компютерів на яких буде використовуватись дана програмна система можуть бути мінімальними;
* вхідні зображення можуть бути сильно зашумленими та вимагати додаткової апаратних ресурсів для складної попередньої обробки;
* результати обробки цифрових зображенб повинні відображатись або в реальному часі або з мінімальними часовими затримками.

В результаті для проведення тестування було обрано робочу станцію з «офісними» параметрами, що є достатніми для повсякденного використання та дозволяє проектувати на відлагоджувати програмні системи для різних сфер використання. Може використовуватися як робоче місце дизайтера, секретаря, науковця тощо. Робоча станція має такі технічні характеристики які наведено на рисунку 3.8:

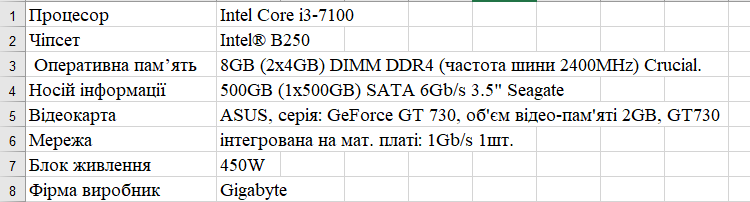


Рисунок 3.8 – Технічні характеристики пристрою відображення зображень

Для комфортної роботи з запропонованим програмним забезпеченням по обробці цифрових зображень важливим елементом апаратної системи є параметри монітора, тому вибір був зупинений на моніторах від фірмі dell. Дані монітори є професійними та їхні технічні параметри задовільняють умовам для вирішення поставлених задач. Технічні характеристики пристрою відображення зображень (DELL P2018H 19.45″ ) можна переглянути на рисуноку 3.9:

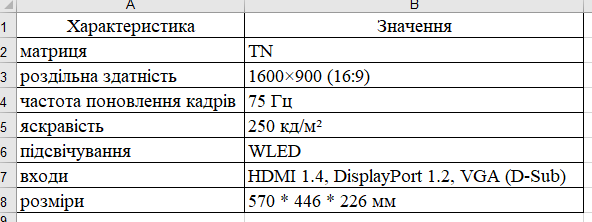


Рисунок 3.9 – Технічні характеристики пристрою відображення зображень

Аналіз технічних параметрі вибраної компютера в повній мірі задовільняють умовам технічного завдання та дозволяють в повній мірі оцінити функціональні можливості спроектованого програмного забезпечення та запропонованих алгоритмів. Тому, дану робчу станції можна використовувати для обробки тестових вибірок та отримання достовірних результатів обробки вхідних зображень.

Етап тестування вимагає створення тестових вибірок для оцінки функціональних можливосте програмної ситеми обробки та аналізу цифрових зображень. Для отримання повноцінних тестових наборів було виділено ряд критеріїв які обовязково повинні бути враховані при відборі тестових цифрових зображень. Серед таких параметрів слід зазначити:

* кількість об’єктів на зображенні;
* кількість однорідних областей на зображенні;
* рівень зашумленості зображення;
* рівень чіткості границь між однорідними областями;
* складність об’єктів.

Даний набір критеріїв є основним при групуванні цифрових зображень та дозволить оцінити розроблену програмну систему на вхідних зображеннях різного типу. В результаті для проведення тестування було сформовано такі групи зображень:

* група простих зображень – для даної групи цифрових зображень були відібрані зображення які містять в собі 1-3 об’єкти, об’єкти в загальному випадку мають просту структуру, містять невелику кількість внутрішніх областей. Границі між однорідними областями мають різкі перепади яскравості та можуть бути легко ідентифіковані. Зашумленість зображень є мінімальною та легко видаляється під час попередньої обробки зображення ;
* група звичайних зображень – в дану групу були віднесені тестові зображення які володіють наступними характеристиками. На зображенні модуть бути присутні до 10 об’єктів. Складні об’єкти складають з чітко виділених обнорідних областей які можуть бути розташовані як окремо так і дотикатись один до одного. Границі між окремими складними об’єктами є чітко видимими, рівень перепадів на границях областей є достатніми для однозначного підкреслення меж. Зашумленість зображення є значною, що вимагає проведення обовязкової попередньої обробки зображень.дана група зображень ілюструє зображення які можна отримати зі звичайних засобів фотофіксації та є основновними для проведення досліджень такого типу;
* група складних зображень. В дану групу зображень бутивіднесені зображення на який присутня велика кількість складних об’єктів, які можуть накладатись/дотикатись один до одного. Об’єкти складаються з великою кількості однорідних областей, а парепад яскравості на границі цих областей є невеликий. Границі складних об’єктів не є чітко видимі, а самі обєкти можуть бути подібними один на одного. Зображенням характерна зашумленість. Аналіз зображення потребує обовязкової попередньої обробки людиною. Дана група зображень є в більшій мірі штучними та була використана з метою оцінки можливостей робти програмного додатку зі складними зашумленими зображеннями низької якості.

В результаті проведеного тестування були отримані аналітичні дані. На основі отриманої інформації можна зробити висновок, що програмний додаток виконав усі поставлені завдання, а отримані в результаті обробки дані були на прогнозованому рівні з допустимими похибками. Очевидно, що зображення першої групи під час обробки показали найкращі результати, обєкти виділялись чітко, а контурна лінія підкреслювала межі складних об’єктів без відхилень, що доводили візуальні оцінки отриманих результатів. Зображення другої групи оброблядись за допустимий час, без значних відчутних затримок, при цьому основні часові затримки були на етапі попередньої обробки. Обєкти виділялиь однозначно правильно, проте у випадках коли границя міє обєктом та фоном зображення були візуально подібними, то точність відображення контурної лінії вже була з похибкою. Проте дана похибка була в межах допостимих значень.

В результаті проведених експерементів та на основі аналізу отриманих даних була сформована підсумкова таблиця де відображаються отримані результати у згрупованому вигляді. Отримані результати були згруповані на основі суб’єктивних оцінок тестера, проте є достатньо об’єктивними. Результати наведено в таблиці 3.1

Таблиця 3.1 – Узагальнена таблиця результатів тестування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Групи зображень/  Хар-ки зображення | Складність об’єкта | Наявність шумів | Подібність до фону | Подібність однорідних областей | Кількість об’єктів на зображенні |
| група простих зображень | Мінімальний вплив | Мінімальний вплив | Відчутний вплив | Відчутний вплив | Відчутний вплив |
| група звичайних зображень | Відчутний вплив | Відчутний вплив | Відчутний вплив | Максимальний вплив | Максимальний вплив |
| група складних зображень | Значний вплив | Відчутний вплив | Максимальний вплив | Максимальний вплив | Максимальний вплив |

Підсумковий аналіз отриманих результатів довів правильність обраних методів та алгоритмі обробки зображень, а такош зовнішніх цифрових бібліотек функцій. Усі поставлені перед програмною системою аналізу та опису зображень виконані в повній мірі. Програмна система опрацьовувала зображення за мінімальні проміжки часу, а отримані результати були в межах допустимих похибок.

## 3.4 Висновки до розділу

Розроблено структуру та проведено моделювання програмного додатку аналізу та опису обєктів на цифрових зображеннях, що надало можливість здійснити програмну реалізацію системи обробки цифрових зображень з функцією підкреслення краниць об’єктів.

Здійснено практичне тестування розробленої ситсеми підкреслення границь об’єктів цифрових зобраень на основі об’єктно-орієнтованого підходу, що доцільність запропонованої структури програмного додатку та алгоритму підклеслення границь.

# Висновки

На основі аналізу існуючих програмних додаткі для аналізу та опису сцен на цифрових зображеннях, а також відомих алгоритмів цифрової обробки зображень можна зробити такі висновки:

1. Проаналізовано основні завдання та сфери застосування механізмів цифрової обробки зображень, що дозволило провести класифікацію основних задач при проектуванні системи опрацювання та опису об’єктів на цифрових зображеннях.
2. Проаналізовано формати кодування та зберігання цифрових зображень, що дозволило обрати програмні структури даних підвищення швидкодії проходження контуром об’єкта.
3. Проведено дослідження програмних систем для аналізу та опису об’єктів на цифрових зображеннях, що дало можливість підкреслити основні архітектурні рішення які використовуються під час реалізації систем даного типу
4. Проведно аналітичний огляд алгоритмів виділення контурів однорідних областей на основі підкресленні парепадів яксравості, що дало можливість визначити множину детекронів границь для проектування алгоритму виділення контурів складних об’єктів.
5. Розроблено алгоритм підкреслення контурів на основі об’єктно-орієнтованого підходу, що дало можливість розробити та провести моделювання програмного додатку виділення та опису складних об’єктів на цифрових зображеннях.
6. Здійснено практичне тестування розробленої ситсеми підкреслення границь об’єктів цифрових зобраень на основі об’єктно-орієнтованого підходу, що доцільність запропонованої структури програмного додатку та алгоритму підклеслення границь

# Список використаної літератури

1. Пастернак В.М. Аналіз алгоритмів підкреслення контурів зображень. Збірник тез VІ Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі»., Тернопіль, 10 листопада 2022 р. с. 28.
2. Пастернак В.М. Алгоритм підкреслення контурів на основі обєктно-орієнтованого підходу. Збірник тез VІ Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі»., Тернопіль, 10 листопада 2022 р. с. 28.
3. Bradski G. Learning OpenCV - Computer Vision with the OpenCV Library, 2018. 580 с.
4. Canny, J. F. Finding edges and lines in images, Master's thesis, MIT. AI Lab. 1983. TR-720.
5. Canny, J. F “A computational approach to edge detection”, IEEE Transaction on Pattern Analysis and Machine Intelligence, 8, 1986. 679-714.
6. Courtney. P & N. A. Thacker “Performance Characterization in Computer Vision: The Role of Statistics in Testing and Design”, Chapter in: “Imaging and Vision Systems: Theory, Assessment and Applications”, Jacques Blanc-Talon and Dan Popescu (Eds.), NOVA Science Books 2001.
7. Hanzi Wang. Robust Statistics for Computer Vision: Model Fitting, Image Segmentation and Visual Motion Analysis, Ph.D thesis, Monash University, Australia. 2001.
8. Huber, P.J. Robust Statistics, Wiley New York.1999.
9. Kirsch, R. “Computer determination of the constituent structure of biological images”, Computers and Biomedical Research, 4, 1971. 315–328.
10. Lakshmi,S. V.Sankaranarayanan. A Study of edge detection techniques for segmentation computing approaches. Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications, 2010. 35-41.
11. Lee, K.. M, Meer, P. Robust Adaptive Segmentation of Range Images, IEEE Trans. Pattern Analysis and Machine Intelligence, 20(2), 2010. 200-205.
12. Marr, D & E. Hildreth. Theory of edge detection. Proc. Royal Society of London, B, 207, 1997. 187–217.
13. Marr, D Vision, Freeman Publishers. 1976.
14. Sujatha P, Sudha KK. Performance analysis of dierent edge detection techniques for image segmentation. Indian Journal of Science and Technology. 2015 Jul; 8(14):1–6.
15. Loganayagi T, Kashwan KR. A robust edge preserving bilaterallter for ultrasound kidney image. Indian Journal of Science and Technology. 2015 Sep; 8(23):1–10.
16. Naraghi MG, Koohi M. Satellite image edge detection based on morphology models fusion. Indian Journal of Science and Technology. 2012 Jul; 5(7):2997–3000.
17. Mukunthan R, Sairam N. Delaunay edge detection using modifed star formation in two dimensional data. Indian Journal of Science and Technology. 2014 Apr; 7(4):426–9.
18. Anjana N, Priestley JJ, Nandhini V, Elamaran V. Color image enhancement using edge based histogram equalization. Indian Journal of Science and Technology. 2015 Nov. 8(29):1–6.
19. Junxi S, Gu D, Chen Y, Zhang S. A multiscale edge detection algorithm based on wavelet domain vector hidden Markov tree model. Pattern Recognition. 2004 Jul; 37(7):1315–24.
20. Jahangir M, Nayak DR. An ecient edge detection technique by two dimensional rectangular cellular automata. International Conference on Information Communication and Embedded Systems (ICICES); 2014 Feb. p. 1–4.
21. Tian Q, Yan Y, Lu G. An autoadaptive edge-detection algorithm for flame and fire image processing. IEEE Transactions on Instrumentation and Measurement. 2012 May; 61(5):1486–93.
22. Melin P, Gonzalez CI, Castro JR, Mendoza O, Castillo O. Edge-detection method for image processing based on generalized type-2 fuzzy logic. IEEE Transactions on Fuzzy Systems. 2014 Dec; 22(6):1515–25.
23. Mohsen S, Fathy M, Mahmoudi MT. A classified and comparative study of edge detection algorithms. Proceedings of International Conference on Information Technology: Coding and Computing; 2002 Apr. p. 117–20.
24. Maini R, Aggarwal H. Study and comparison of various image edge detection techniques. IJIP. 2009 Mar; 3(1):1–12.
25. Marr, P & Doron Mintz, D. Robust Regression for Computer Vision: A Review. International Journal of Computer Vision, 6(1), 1998. 59-70.
26. Orlando, J, Tobias., Rui Seara. Image Segmentation by Histogram Thresholding Using Fuzzy Sets. IEEE Transactions on Image Processing, Vol.11, No.12, 2015. 1457-1465.
27. Punam Thakare. A Study of Image Segmentation and Edge Detection Techniques. International Journal on Computer Science and Engineering, Vol 3, No.2, 2018. р.899-904.
28. Rafael C. Gonzalez, Richard E. Woods. Steven L. Eddins. Digital Image Processing Using MATLAB, Pearson Education Ptd. Ltd, Singapore. International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 6, Dec 2011
29. Ramadevi, Y Segmentation and object recognition using edge detection techniques, International Journal of Computer Science and Information Technology, Vol 2, No.6, 2010. 153-161.
30. Roberts, L Machine Perception of 3-D Solids, Optical and Electro-optical Information Processing, MIT Press 1981.
31. Robinson. G. Edge detection by compass gradient masks. Computer graphics and image processing, 6, 2004. р.492-501.
32. Rousseeuw, P. J., Leroy, A Robust Regression and outlier detection, John Wiley & Sons, New York. 2011.
33. Senthilkumaran. N., R. Rajesh Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches. International Journal of Recent Trends in Engineering, Vol. 1, No. 2, 2017. 250-254.
34. Sowmya. B., Sheelarani. B. Colour Image Segmentation Using Soft Computing Techniques, International Journal of Soft Computing Applications, Issue 4, 2019. 69-80.
35. Umesh Sehgal Edge detection techniques in digital image processing using Fuzzy Logic, International Journal of Research in IT and Management, Vol.1, Issue 3, 2018. 61-66.
36. Yu, X, Bui, T.D. Robust Estimation for Range Image Segmentation and Reconstruction. IEEE trans. Pattern Analysis and Machine Intelligence, 16 (5), 530-538.
37. Eltoft T. Modeling the amplitude statistics of ultrasonic images. IEEE Transaction on Medical Imaging. 2006 Feb; 25(2):229–40.
38. Tomasi C, Manduchi R. Bilateral filter for gray and color images. Proceedings of 6th International Conference on Computer Vision; Bombay. 1998 Jan 4–7. p. 839–46.
39. Bhonsle D, Chandra V, Sinha GR. Medical Image Denoising Using Bilateral Filter. I J Image, Graphics and Signal/processing. 2012 Jul; 4(6):36–43.
40. Balocco S, Gatta C, Pujol O, Mauri J, Radeva P. SRBF: Speckle Reducing Bilateral Filtering. Ultrasound in Medicine and Biology. 2010 Aug; 36(8):1353–63.
41. Tang J, Guo S, Sun Q, Deng Y, Zhou D. Speckle reducing bilateral filter for cattle follicle segmentation. International Conference on Bioinformatics and Computational Biology. Las Vegas, NV, USA. 2010 Nov; 11(2):S9.
42. Zhang M, Gunturk BK. Multiresolution bilateral filtering for image denoising. IEEE Transaction on Image Processing. 2008 Dec; 17(12):2324–33.
43. Raj VNP, Venkateswarlu T. Ultrasound medial image denoising using hybrid bilateral filtering. International Journal of Computer Applications. 2012 Oct; 56(14): 44–51.
44. Wenxuan SHI, Jie LI, Minyuan WU. An image denoising method based on multiscale wavelet thresholding and bilateral filtering. Wuhan University Journal of Natural Science. 2010 Apr; 15(2):148–52.
45. Kao WC, Chen YJ. Multistage noise filtering and edge detection for color image enhancement. IEEE Transaction on Consumer Electronics. 2005 Nov; 51(4):1346–51.
46. Han JW, Kim JH, Cheon SH, Kim JO, Ko SJ. A novel image interpolation method using the bilateral filter. IEEE Transaction on Consumer Electronics. 2010 Feb; 56(1):175–81.
47. Loganayagi T. De-noising methods for ultrasound renal images – A comparative study. Int J of Appl Engg Research. 2015; 10(4):8741–56.
48. Anitha U, Malarkkan S. A novel approach for despeckling of sonar image. Indian Journal of Science and Technology. 2015 May; 8(S9):252–9.
49. Shreyamsha Kumar BK. Image denoising based on nonlocal mean filter and its method noise. Signal, Image and Video Processing. 2013 Nov; 7(6):1211–27.
50. Березький О.М., Дубчак Л.О., Мельник Г.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 - Комп’ютерна інженерія. Магістерська програма - Комп’ютерна інженерія". Тернопіль: ЗУНУ, 2022. 32 с.
51. Гураль І.В., Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» Тернопіль: ТНЕУ, 2019. 33 с.