

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Розробка підсистеми дослідження методів підвищення якості
цифрових зображень»

Осокін Арсен Станіславович

(прізвище, ім'я та по батькові студента повністю)

Київ 2023 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ

НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Розробка підсистеми дослідження методів підвищення якості
цифрових зображень»

Виконав: студент 4-го курсу, групи КН-41

Спеціальності: 122 «Комп'ютерні науки

Спеціалізація: «Інформаційні
управляючі системи і технології»

(шифр і назва напрямку підготовки,
спеціальності)

_____ Осокін А.С.

(прізвище та ініціали)

Керівник к.т.н., доц. Горда О.В.

(прізвище та ініціали)

Рецензент к.т.н., доц. Єрукаєв А.В.

(прізвище та ініціали)

Київ 2023 р

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій
Кафедра: інформаційних технологій
Освітній рівень: «бакалавр» за ОП
Спеціальність: 122 «Комп'ютерні науки»
Спеціалізація: Інформаційні управляючі системи і технології.

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ
д.т.н., професор Цюцюра С.В.

„_17_” ___ листопада ___ 2022 року

**ЗАВДАННЯ
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Осокін Арсен Станіславович

1. Тема роботи: Розробка підсистеми дослідження методів підвищення якості цифрових зображень затверджена наказом ректора КНУБА № 1811/2 від « 17 » листопада 2022 р.
2. Керівник роботи: к.т.н, доцент Горда Олена Володимирівна, кафедри інформаційних технологій проектування і прикладної математики.
3. Строк подання студентом роботи до захисту: 12 червень 2023 р
4. Зміст пояснювальної записки за розділами:
 - Р.1. Аналіз предметної області
 - Р.2. Математичні методи обробки зображень, програмне проектування
 - Р.3. Практична реалізація
 - Р.4. Техніко-економічне обґрунтування розробки підсистеми
5. Інформаційні слайди:

- C.1. Розробка підсистеми дослідження методів підвищення якості цифрових зображень
- C.2. Основні проблеми
- C.3. Існуючі рішення
- C.4. Дерево функцій
- C.5. Математичні методи обробки зображень
- C.6. Побудова моделі
- C.7. Технічна платформа
- C.8. Програмна реалізація
- C.9. Приклади роботи
- C.10. Висновки

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Лютий 2022 р.
Р. 2. Розробка інформаційного забезпечення	Квітень 2022 р.
Р. 3. Розробка програмного забезпечення. Тестовий приклад програми	Квітень 2022 р.
Р. 4. Техніко-економічне обґрунтування розробки підсистеми	Квітень 2022 р.
Оформлення роботи	Травень 2022 р.
Направлення роботи на рецензування	Червень 2022 р.
Попередній захист роботи на кафедрі	Червень 2022 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	Дата	Підпис
Техніко-економічне обґрунтування розробки підсистеми	д.т.н. проф. Цюцюра С.В.		
Прийом програмного продукту	к.т.н. доц., Горда О.В.		

8. Дата видачі завдання: 17 листопада 2022 р.

Керівник

Горда О.В.

(підпис)

(прізвище та ініціали)

Бакалавр

Осокін А.С.

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Осокін А.С. «Розробка підсистеми дослідження методів підвищення якості цифрових зображень».

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», спеціалізація: «Інформаційні технології». – Київський національний університет будівництва та архітектури. – Київ, 2023.

Ця робота присвячена розробці підсистеми, яка спрямована на дослідження методів підвищення якості цифрових зображень. На основі проаналізованих методів була створена практична реалізація. Було проведено її тестування, та перевірка на працездатність.

Особливості архітектури програмного продукту передбачають можливість його модифікації для подальшого розширення функціоналу.

Ключові слова: методи, оцінка, якість, цифрові зображення.

ANNOTATION

Osokin A.S. "Development of a subsystem for researching methods for improving the quality of digital images".

Bachelor's thesis for a bachelor's degree in speciality: 122 "Computer Science", specialisation: "Information Technologies - Kyiv National University of Construction and Architecture - Kyiv, 2023.

This work is devoted to the development of a subsystem aimed at researching methods for improving the quality of digital images. Based on the analysed methods, a practical implementation was created. It was tested and checked for performance.

The features of the software product architecture provide for the possibility of its modification for further expansion of the functionality.

Keywords: methods, assessment, quality, digital images.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Аналіз проблеми	8
1.2 Визначення цілей	11
1.3 Особливості процесу формування цифрових зображень ...	12
1.4 Аналіз задач	14
1.5 Аналіз методів оцінки якості зображення	20
1.6 Аналіз факторів впливу та їх вплив обрання методів	22
1.7 Аналіз існуючих рішень	24
1.8 Постановка задачі	26
1.9 Дерево функцій	27
2 МАТЕМАТИЧНІ МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ, ПРОГРАМНЕ ПРОЕКТУВАННЯ	28
2.1 Концептуалізація схеми обробки	28
2.2 Моделі представлення зображень	29
2.3 Бікубічна інтерполяція	33
2.4 Фільтр Собеля	35
2.5 Фільтр Гаусса	37
2.6 Алгоритм Кенні	39
2.7 Стандартне відхилення	41
2.8 Фільтр Лапласа	42
2.9 Медіанний фільтр	43
2.10 Нерізкість маскуваня	45
2.11 Передбачення сірого світу	47
2.12 Ліміт розтягування	49
2.13 Побудова моделі	51
3. МАТЕМАТИЧНІ МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ, ПРОГРАМНЕ ПРОЕКТУВАННЯ	53
3.1 Вибір технічної платформи	53

3.1	Структура технічної платформи.....	54
3.2	Модуль Image Processing Toolbox	56
3.3	Створення модуля дослідження методів підвищення якості зображень 58	
3.4	Програмна реалізація модуля дослідження методів підвищення якості зображень.....	62
3.5	Контрольний приклад	66
4.	ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ПІДСИСТЕМИ	70
4.1	Вступ	70
4.2	Резюме	70
4.3	Проектований продукт або вид послуг	74
4.4	Оцінка ринку збуту.....	77
4.5	Конкуренція	80
4.6	Стратегія маркетингу	81
4.7	План виробництва	82
4.8	Організаційний план.....	83
4.9	Юридичний план	85
4.10	Оцінка ризику та страхування.....	87
4.11	Фінансовий план	89
4.12	Стратегія фінансування	91
4.13	Висновки	92
	ВИСНОВОК.....	94
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	95

ВСТУП

Впровадження технологій цифрового оброблення зображень є необхідною складовою сучасного світу, де візуальна інформація відіграє вирішальну роль. Зростання кількості цифрових зображень, які ми стикаємося щодня, вимагає постійного покращення якості цих зображень, щоб забезпечити насиченість, чіткість і достовірність отриманої інформації.

Розробка підсистеми дослідження методів підвищення якості цифрових зображень є актуальною задачею, що відкриває безліч можливостей для вдосконалення обробки зображень. Створюєма підсистема буде спрямована на вивчення і застосування передових методів та алгоритмів, які допоможуть покращити деталізацію, контрастність, розмір та інші характеристики цифрових зображень.

Застосування підсистеми дослідження методів підвищення якості цифрових зображень знаходить широке застосування в різних сферах, таких як медицина, наука, медіа, безпека, машинне бачення та багато інших. Цей розвиток приведе до значного покращення візуального сприйняття, що має велике значення для точних діагнозів, надійної ідентифікації, ефективного аналізу даних і багатьох інших задач.

У даній роботі ми маємо за мету розробити підсистему, яка дозволить вивчати імплементувати різні методи підвищення якості цифрових зображень, зосереджуючись на визначених аспектах, таких як шуми, розмиття, відновлення деталей, підвищення контрастності та роздільної здатності. Використовуючи нашу підсистему, дослідники і практики зможуть ефективно оцінювати різні методи та алгоритми, порівнювати їх результати та вдосконалювати існуючі підходи [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз проблеми

Методи обробки цифрових зображень мають широке застосування у багатьох галузях. Вони використовуються для покращення якості зображень, виявлення та аналізу об'єктів, розпізнавання образів, відновлення зображень, обробки медичних зображень, визначення якості продукції та багатьох інших сферах.

Наприклад, в медицині обробка цифрових зображень використовуються для діагностики та визначення лікувального курсу хвороб, а в промисловості - для контролю якості виробів та процесів виробництва. Крім того, методи обробки цифрових зображень знаходять застосування у сфері комп'ютерної візуалізації, розробки ігор, створення віртуальної реальності та інших сучасних технологій.

Також однією з галузей, де різні методи обробки цифрових зображень мають велике застосування, є фотограметрія. Ця наука вивчає методи та засоби отримання геометричної інформації про об'єкти зображень.

Застосування фотограметрії дуже широке і включає в себе такі галузі, як картографія, геодезія, містобудування, археологія, геологія, лісове господарство, агрономія та багато інших. Дана галузь є важливим інструментом для вивчення змін на земній поверхні, розробки планів міст, створення карт та планування місцевості для будівництва доріг та інфраструктури [2].

Одним з важливих її аспектів є стереообробка зображень. Вона дозволяє отримати глибинну інформацію про об'єкти на земній поверхні за допомогою аналізу пари фотографій з різних точок зору. Ця інформація може бути використана для створення тривимірних моделей об'єктів та їх поверхні, що дозволяє здійснювати більш точне планування та конструювання.

Також важливим аспектом цієї галузі є космічна фотограметрія, яка дозволяє отримувати зображення Землі з космосу. Це дає змогу вивчати глобальні процеси, такі як зміна клімату, динаміка річок, екологічні зміни та інші.

Обробка цифрових зображень має також велике значення в поліграфії, оскільки ця галузь вимагає високої якості друку та точності відтворення кольорів і зображень.

Один з основних методів обробки цифрових зображень в поліграфії - це колориметрія, тобто наука про вимірювання кольорів. Завдяки цьому методу, друкована продукція може мати точні кольорові характеристики, що дозволяє досягати високої якості друку. Крім того, методи обробки цифрових зображень використовуються для покращення різних елементів друкованої продукції, таких як зображення, текст, фон, рамки, логотипи тощо. Вони дозволяють зменшити шум, розмиття та інші артефакти, які можуть з'являтися під час процесу друку [3].

Крім того, методи обробки цифрових зображень знаходять застосування в поліграфії для розробки та створення оригінальних дизайнів друкованої продукції. Вони дозволяють створювати унікальні зображення, які відповідають конкретним вимогам клієнта або ринку, та допомагають зробити продукцію більш привабливою для споживачів.

Перевірка якості зображень також є важливим етапом обробки цифрових зображень. Для цього можна використовувати різноманітні методи, такі як вимірювання показників якості, аналіз гістограм, порівняння зразків, виявлення та корекція дефектів, відновлення пошкоджених зображень тощо.

У галузі фотограметрії перевірка якості зображень є важливим етапом при створенні топографічних карт, карт земних ресурсів та карт лісового покриву. Використовуючи методи обробки цифрових зображень, можна автоматично визначити точність та рівномірність освітлення, розпізнати та

видаляти шум, виявляти та коригувати геометричні спотворення, що дозволяє отримати якісні та точні дані для подальшої роботи.

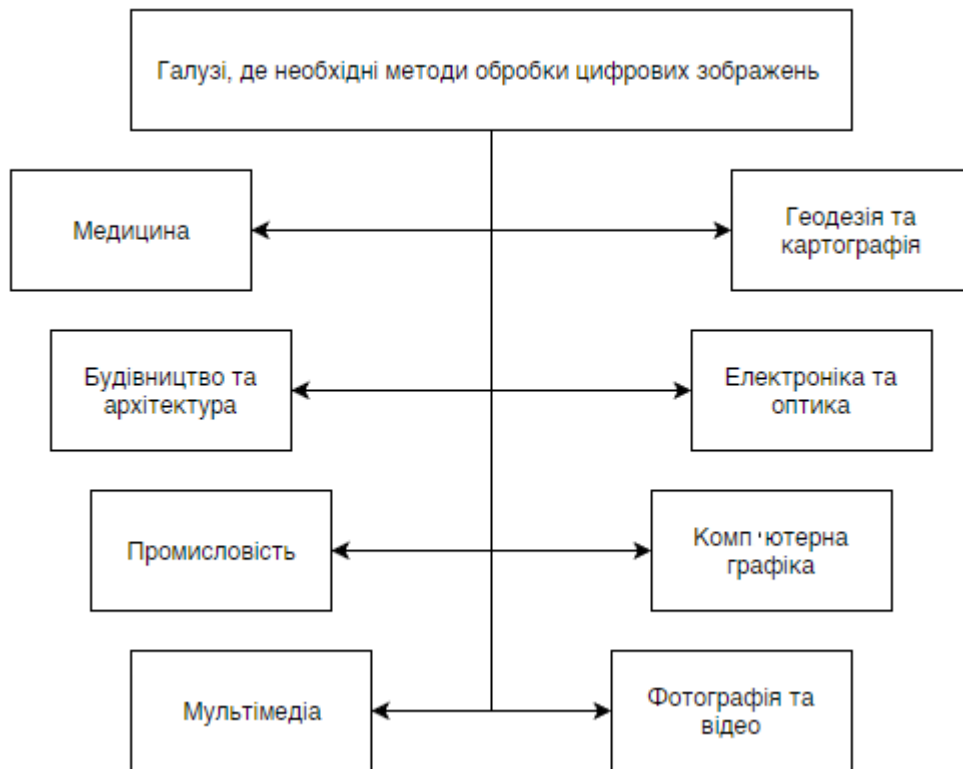


Рисунок 1.1 – Схема галузей застосування методів обробки цифрових зображень

У промисловості також велика увага приділяється перевірці якості зображень. Наприклад, в автомобілебудуванні перевірка якості зображень дозволяє виявляти дефекти на поверхнях кузовів автомобілів та інших деталей, що дозволяє уникнути появи проблем на етапі експлуатації.

В залежності від галузі застосування завдання обробки цифрових зображень суттєво відрізняються. Через це розробка єдиної універсальної методології робиться неможливою задачею. Тому методи, що будуть застосовані будуть залежати від завдання, тобто обробка зображень буде проводитися з оглядом на певну кінцеву мету галузі.

З огляду на вищесказане застосування цифрових зображень, вимоги до якості відрізняються різних організацій та установ, але наявність зображення з високою роздільною здатністю завжди буде в пріоритеті, бо набагато легше зменшити якість вихідного зображення, ніж його підвищити.

1.2 Визначення цілей

Цілі дослідження якості цифрових зображень для забезпечення якісного масштабування зображень наведено на рисунку 1.2.

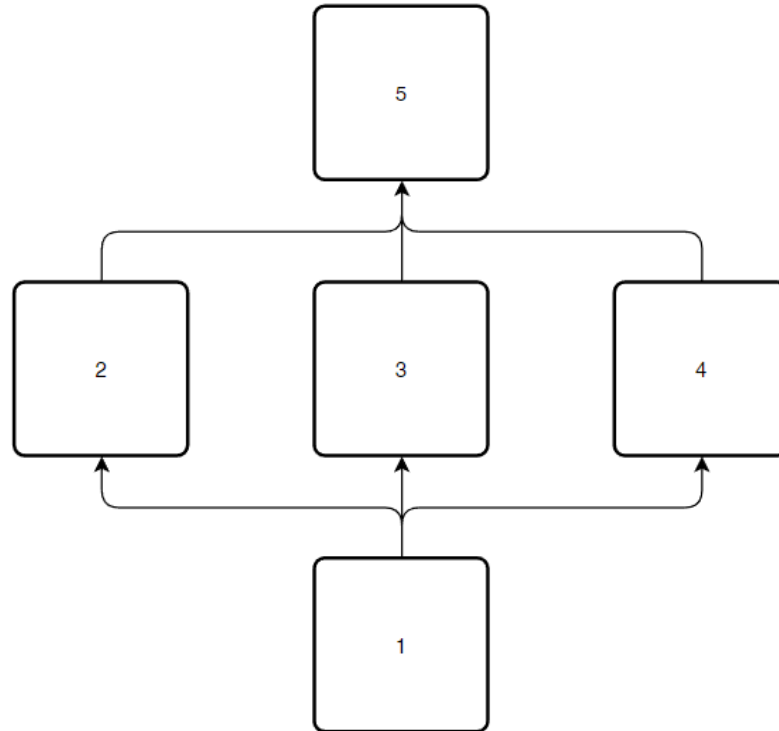


Рисунок 1.2 – Дерево цілей дослідження якості цифрових зображень

1. Оцінка якості зображення перед масштабуванням:

Вимірюються такі параметри, як роздільна здатність, контрастність, чіткість та інші, щоб встановити, наскільки висока якість масштабованого зображення порівняно з оригіналом.

2. Визначення оптимального методу масштабування зображення:

Досліджуються методи, такі як білінійне масштабування, методи на основі інтерполяції, методи на основі нейромереж та інші, щоб визначити найкращий метод для кожного типу зображення та вимог до якості.

3. Оцінка впливу розміру масштабування на якість зображення:

Досліджуються такі параметри, як роздільна здатність, контрастність та інші, щоб встановити оптимальний розмір масштабування для різних типів зображень та різних вимог до якості.

4. Визначення оптимального рівня компресії зображення:

Досліджуються такі параметри, як ступінь стиснення, роздільна здатність та інші, щоб встановити оптимальний рівень компресії для кожного типу зображення та вимог до якості.

5. Визначення кращих методів для відповідних типів зображень:

Це допоможе забезпечити високу якість масштабованих зображень та допоможе забезпечити взаємну сумісність між різними програмами та пристроями, що використовують масштабовані зображення.

Усі ці дослідження можуть бути проведені з використанням різних методів та технологій, таких як комп'ютерна томографія, цифрова мікроскопія, зображення високої роздільної здатності, тестування людської сприйнятливості та інші. Результати досліджень можуть бути використані для покращення якості масштабованих зображень та розробки нових технологій для забезпечення ще вищої якості зображень.

1.3 Особливості процесу формування цифрових зображень

Процес отримання зображення значною мірою залежить від умов отримання зображення, формування цифрових зображень складається з декількох етапів, що включають підготовку до отримання зображення, захоплення і обробку даних, що надходять від датчика зображення, а також збереження, передачу та відтворення отриманих зображень [4]. Стадії процесу формування цифрових зображень наведено на рисунку 1.3.

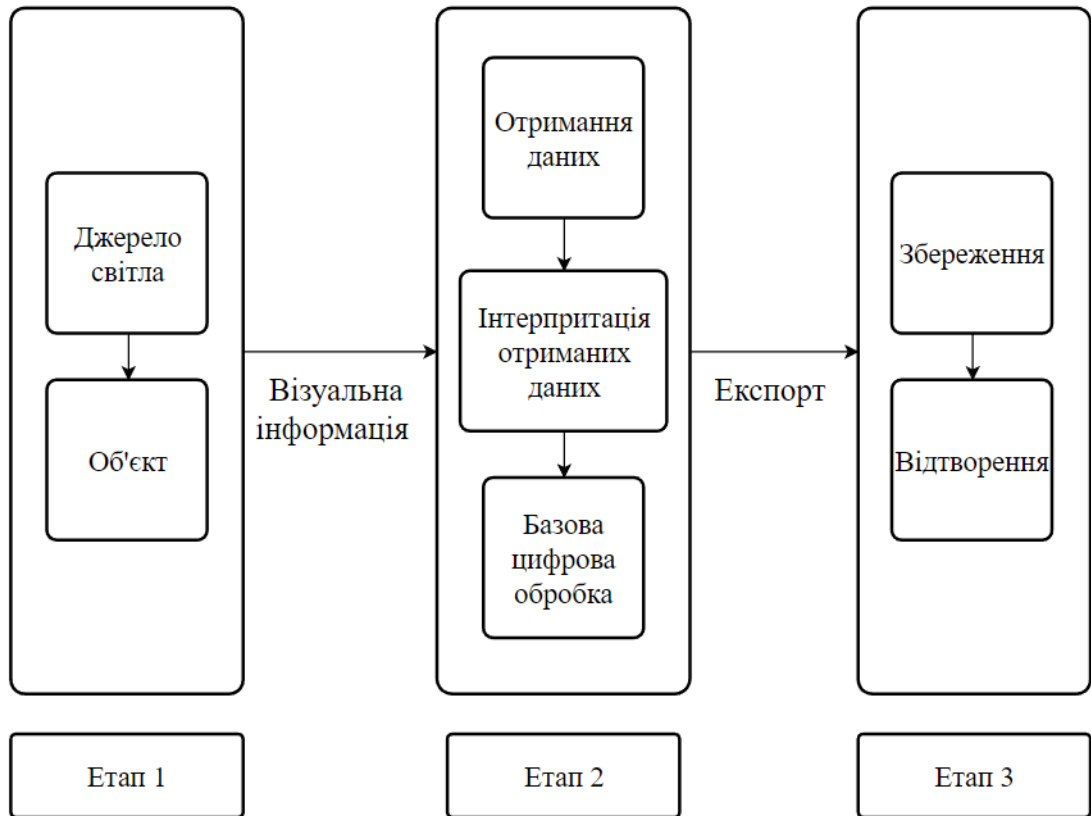


Рисунок 1.3 – Стадії процесу формування цифрових зображень

Першим етапом є підготовка до отримання зображення. Цей етап включає в себе вибір оптимальної області зйомки, визначення відстані до об'єкта та налаштування камери, щоб забезпечити оптимальні умови для отримання зображення.

Другий етап - захоплення та обробка даних, що надходять від датчика зображення. В цьому етапі інформація про зображення зчитується з датчика і перетворюється в цифрову форму. Цифрова обробка може включати в себе калібрування, корекцію шуму, компресію та інші процеси, що допомагають отримати якісне зображення.

Останнім етапом є збереження, передача та відтворення зображень. Цифрові зображення можуть бути збережені в різних форматах, таких як JPEG, PNG, TIFF тощо. Для передачі зображення можуть використовуватися різні комунікаційні технології, такі як Wi-Fi, Bluetooth, USB тощо.

Остаточний етап - відтворення зображення - відбувається на екрані відповідного пристрою та не є обмеженим єдиним ПК.

Умови отримання зображення можуть включати такі параметри, як освітленість, рівень шуму, відстань до об'єкта, кут зйомки, об'єктні характеристики та інші фактори. Врахування цих параметрів дозволяє отримати якісне зображення з оптимальною деталізацією та колірною гамою. Для отримання якісного зображення, можуть використовуватися різні техніки та методи. Наприклад, для зменшення шуму можна використовувати різні алгоритми, такі як фільтрація Гауса, медіанна фільтрація, фільтр Калмана тощо [5]. Для отримання оптимальної колірної гами можна використовувати кольорові профілі та калібрування кольору.

Також важливо враховувати особливості самого об'єкта зйомки та його характеристик, такі як розмір, форма, текстура тощо. Наприклад, при зйомці об'єктів з нерівною поверхнею, можуть виникати проблеми з відображенням деталей на зображенні. У таких випадках можуть використовуватися методи компенсації кута зйомки, які дозволяють вирішити цю проблему.

Крім того, важливо враховувати особливості середовища зйомки. Наприклад, при зйомці в умовах недостатнього освітлення може виникати шум на зображенні. У таких випадках можуть використовуватися методи підвищення якості зображення з низькою освітленістю, такі як обробка зображення з використанням алгоритмів, що зменшують шум та підсилюють контрастність.

Отже, вивчення та врахування параметрів умов отримання зображення, а також використання різноманітних технік та методів обробки даних, дозволяє отримати якісне та деталізоване цифрове зображення.

1.4 Аналіз задач

Задачі обробки цифрових зображень можуть бути дуже різноманітні і зазвичай вони пов'язані з вирішенням певних завдань.

Ще один тип задач, пов'язаних з обробкою цифрових зображень, - підвищення різкості зображення. Підвищення різкості зображення дозволяє поліпшити якість зображення, збільшити контрастність та чіткість деталей на зображенні. Для досягнення цієї мети можуть бути використані різні методи, такі як фільтрація, дифузія, еквалізація гістограм, розширення контрастності тощо [6]. Вибір підходящого методу залежить від бажаного результату та типу зображення.

Однією з задач обробки цифрових зображень є видалення шуму. Шум може бути результатом різних чинників, таких як низька якість зображення, недосконалість оптичних систем або нестабільність джерела світла. Для видалення шуму можна застосовувати фільтри згладжування, такі як фільтр Гауса або медіанний фільтр.

Задача розмиття на цифровому зображенні полягає в зменшенні контрастності між сусідніми пікселями на зображенні. Цей процес може бути використаний для зменшення шуму, видалення деталей або створення ефекту розмиття. Розмиття можна виконати з використанням різних методів, таких як фільтри Гауса, медіанні фільтри, фільтри розмивання з рухомим середнім та інші.

Одним з найбільш поширених методів розмиття є фільтр Гауса, який використовується для згладжування зображення шляхом зменшення різниці між інтенсивністю пікселів в околі кожного пікселя. Цей фільтр є лінійним та ефективним, а його параметри можуть бути налаштовані в залежності від потреб користувача.

Медіанні фільтри, з іншого боку, використовуються для видалення шуму на зображенні, особливо коли шум має дискретний характер. Ці фільтри працюють шляхом заміни значення пікселя на медіану його сусідніх пікселів, що дозволяє ефективно видаляти шум, не впливаючи на деталі зображення.

Фільтри розмивання з рухомим середнім також використовуються для розмиття зображень, зокрема для зменшення контрастності та видалення шуму. Ці фільтри працюють шляхом заміни значення пікселя на середнє значення пікселів в його області.

Ще однією задачею є покращення кольорів зображення. Для цього можна застосовувати фільтри покращення контрасту, які збільшують різницю між світлими та темними областями зображення [7]. Також можна використовувати фільтри кольорової корекції, які змінюють баланс кольорів на зображенні.

Іншою задачею є видалення артефактів, таких як відблиски, відбитки пальців або ретушування. Для цього можна застосовувати різні методи, такі як використання масок або клонування [8].

Виправлення перспективи – також задача, яка може виникнути при обробці зображень. Вона полягає у виправленні спотворення перспективи, яке може виникнути при фотографуванні з висоти або під кутом. Цей процес може бути виконаний за допомогою перетворень гомографії або ще відомого як Афінна трансформація [9]. Також можна застосовувати методи сегментації, щоб виділити окремі об'єкти на зображенні та використовувати їх для подальшого аналізу.

Окрім цього, існують задачі, пов'язані з розпізнаванням об'єктів на зображенні, визначенням форм та розмірів об'єктів, сегментацією зображення на окремі частини та інші.

Обробка зображень може включати алгоритми, які використовують статистичні методи для визначення та аналізу об'єктів на зображенні, а також методи машинного навчання для класифікації зображень та відновлення втрачених даних.

Для масштабування цифрових зображень можуть бути використані різні методи, такі як інтерполяція, роздільна здатність, згортка, прямокутне згорання та інші. Вибір підходящого методу залежить від багатьох факторів,

таких як розмір оригінального зображення, бажаний розмір зображення після масштабування, тип зображення тощо.

Проблематика масштабування цифрових зображень полягає в тому, що під час збільшення зображення можуть виникати різні проблеми, пов'язані зі зниженням якості та деталізацією зображення.

Основні проблеми, що виникають під час масштабування цифрових зображень, включають наступні:

- Втрата роздільної здатності

При збільшенні розміру зображення, пікселі розтягуються, що призводить до зниження роздільної здатності та розмиття зображення.

- Втрата деталізації

При збільшенні зображення, низькі деталі можуть зникнути, тому що вони не можуть бути відтворені на більш великій масштабній зоні.

- Розриви в зображенні

При збільшенні розміру зображення можуть виникнути розриви в зображенні, особливо на контурах об'єктів, що призводить до їх неприродного виду.

- Артефакти компресії

Під час масштабування стиснутих зображень можуть виникати артефакти, такі як блоки, шум та інші аномалії.

- Пікселізація

Ефект пікселізації може виникати, коли розмір пікселів стає занадто великим, щоб забезпечити якісну деталізацію зображення.

Апаратна частина має значний вплив на отримання цифрових фотографій. Сучасні цифрові фотокамери мають складну систему обробки зображень, яка включає в себе оптичну систему, матрицю, процесор, обробку зображень та інші компоненти [10].

Схема апаратної частини типової сучасної фотокамери зображена на рисунку 1.4.

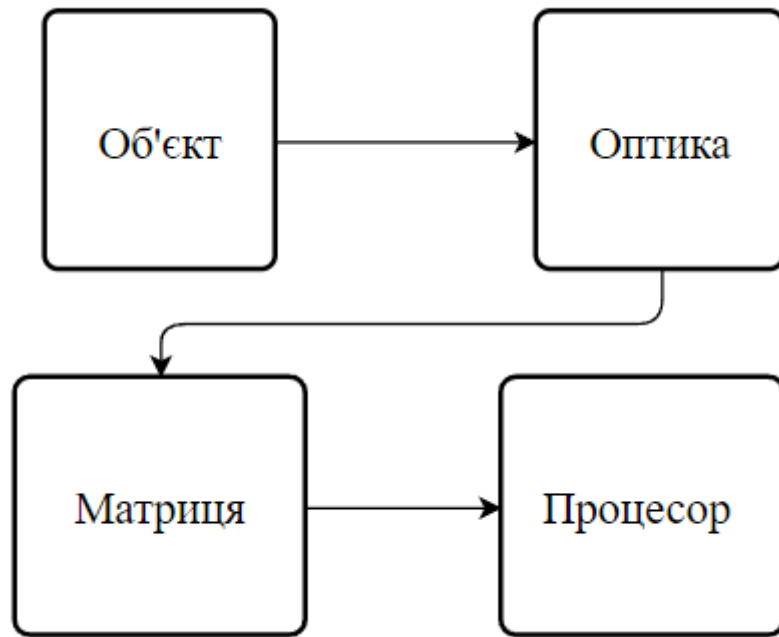


Рисунок 1.4 – Типова апаратна частина сучасних фотокамер

Оптична система відповідає за формування зображення на матриці. Якість оптичної системи визначає рівень деталізації, чіткість та якість зображення. Якість оптичної системи залежить від таких факторів, як кількість лінз, їх якість, величина діафрагми, фокусна відстань та інші параметри.

Матриця відповідає за реєстрацію зображення. Якість матриці визначається кількістю пікселів та їх розміром. Більша кількість пікселів забезпечує більшу деталізацію зображення, але може підвищити рівень шуму. Розмір пікселів впливає на роздільну здатність зображення та якість кольорової передачі.

Процесор та обробка зображень відповідають за обробку та збереження отриманого зображення. Якість процесора та алгоритмів обробки зображень визначає рівень шуму, деталізацію та кольорову передачу.

Різниця між алгоритмами базової обробки фото в камерах різних виробників може бути значною і включати в себе різноманітні функції і параметри, такі як кольорова корекція, затримка експозиції, розмиття

заднього плану та інші. Деякі приклади різниці між алгоритмами базової обробки фото у камер різних виробників:

- Canon

Камери Canon часто мають більш насичені кольори та менший рівень шумів на високих ISO порівняно з іншими виробниками. Для зменшення шумів високих ISO значень використовуються спеціальні алгоритми.

- Nikon

Камери Nikon зазвичай мають більш низький рівень контрастності порівняно з Canon, що дає змогу зберігати більше деталей в тінях та світлих ділянках зображення. Крім того, камери Nikon часто мають більш точну систему автофокусування.

- Sony

Камери Sony часто мають вбудований динамічний діапазон, який дає можливість зберігати деталі як в темних, так і в світлих частинах зображення. Також камери Sony зазвичай мають вбудовану систему стабілізації зображення.

- Fujifilm

Камери Fujifilm зазвичай мають більш насичені кольори та менший рівень шумів на високих ISO значеннях. Крім того, камери Fujifilm зазвичай мають високоякісні об'єктиви, що дає змогу отримувати чіткі та деталізовані зображення.

- Leica

Камери Leica зазвичай мають високоякісні об'єктиви та деталізовані зображення. Крім того, камери Leica зазвичай мають вбудовані елементи, що дозволяють коригувати перспективу та розмиття заднього плану в режимі зйомки. Крім того, камери Leica часто мають більш насичені кольори та деталізовані зображення, що дозволяє отримувати професійні фотографії.

Загалом, різниця між алгоритмами базової обробки фото у камер різних виробників може бути значною і залежати від конкретної моделі камери.

Але особлива увага останнім часом приділяється саме методам обробки зображень, що дозволяє отримати достовірну інформацію використовуючи навіть не найкраще обладнання.

Як приклад, Камери iPhone довго використовували камери на 12 мегапікселів. Однак, навіть з такою роздільною здатністю, камери iPhone дозволяли отримувати досить якісні фотографії завдяки методам обробки зображень, таким як Smart HDR та Deep Fusion.

Smart HDR використовується для зйомки фотографій з високою динамікою, тобто тих, де є яскраві і темні ділянки одночасно. Цей метод використовує різні кадри з різними експозиціями, які об'єднуються в один зображення з більшою динамікою.

Deep Fusion, зі свого боку, дозволяє отримувати більш деталізовані зображення завдяки обробці кадрів на рівні пікселів. Для цього камера робить декілька знімків різної експозиції та різної глибини різкості, які потім об'єднуються в одне зображення з більш високою якістю.

1.5 Аналіз методів оцінки якості зображення

Існує багато різних методів оцінки якості цифрових зображень. Деякі з них орієнтовані на суб'єктивну оцінку людиною, а інші - на об'єктивну оцінку з використанням математичних алгоритмів та параметрів.

1. Суб'єктивні методи оцінки якості зображень.

Дані методи оцінки вимагають участі людей у процесі оцінки якості зображення. Перевагою цього підходу є аналіз людьми зображення, із урахуванням його змісту та контексту. Але людський фактор призводить до суб'єктивності результатів. Оцінка якості зображення залежить від сприйняття індивідуальної особистості, її настрою, зору, звичок і т.д. Найбільш поширеним методом є Absolute Category Rating (ACR) [11], в якому учасникам показують кілька версій зображення, та вони повинні оцінити якість кожного з них за шкалою.

2. Об'єктивні методи оцінки якості зображень.

Ці методи ґрунтуються на математичних алгоритмах та метриках. Найбільш відомими з них є :

- Peak Signal to Noise Ratio (PSNR) - це одна з найбільш поширених метрик для оцінки якості зображень. PSNR обчислюється як відношення максимальної можливої потужності сигналу до середньоквадратичної помилки між оригінальним зображенням та його стиснутою версією [12].

- Structural Similarity Index (SSIM) - метрика, яка оцінює подібність між двома зображеннями, яка базується на оцінці схожості між локальними блоками зображення [12].

- Mean Squared Error (MSE) - це метрика, яка обчислює середньоквадратичну помилку між оригінальним зображенням та його стиснутою версією [13].

- Visual Information Fidelity (VIF) - метрика, яка враховує людський фактор оцінки якості зображення, враховуючи взаємодію між головними параметрами сприйняття зображень.

Хоча об'єктивні методи оцінки якості зображень мають свої переваги, вони також мають свої обмеження та проблеми.

Одна з найбільших проблем полягає в тому, що об'єктивні методи можуть не враховувати важливі аспекти зображення, які сприймаються людиною. Наприклад, метрики, які використовуються для вимірювання різкості, можуть недостатньо відображати сприйняття людиною цієї характеристики зображення.

Ще однією проблемою є те, що об'єктивні методи можуть бути недостатньо чутливими до певних артефактів зображень. Наприклад, певні метрики можуть бути менш чутливі до артефактів стиснення, які не помітні при перегляді зображення на екрані, але можуть стати помітні при друку зображення в великому форматі.

Важливим є відсутність універсальної характеристики: немає жодної об'єктивної міри, яка б відповідала для всіх типів зображень та за всіх умов. Кожен тип зображення може мати власні особливості, які важко врахувати.

Крім того, великий вплив має форматування зображення. Наприклад, якщо зображення стиснуте у форматі, який не підтримує певні функції, то результати можуть бути дуже неточними.

Немає одного "найкращого" методу аналізу якості цифрового зображення, оскільки різні методи мають свої переваги та недоліки і можуть бути застосовані в залежності від конкретних задач та обставин.

Якщо йдеться про оцінку якості зображень з метою порівняння різних кодеків або методів компресії, то можуть бути використані такі методи, як PSNR, SSIM, MS-SSIM, VIF та інші. Кожен з цих методів має свої переваги та недоліки і може бути більш або менш ефективним в різних умовах.

Якщо ж йдеться про оцінку якості зображень для конкретної задачі, наприклад, розпізнавання обличчя або медичну діагностику, то можуть бути використані спеціалізовані методи, які враховують специфіку цієї задачі.

Окрім того, неможливо оцінити та покращити якість зображення тільки за допомогою об'єктивних методів, оскільки вони не враховують суб'єктивних аспектів сприйняття зображення людьми. Тому для повної оцінки якості зображення можуть використовуватись як об'єктивні, так і суб'єктивні методи.

1.6 Аналіз факторів впливу та їх вплив обрання методів

Обрання методів обробки цифрових зображень залежить від багатьох факторів, які можна розглянути більш детально. Одним із таких факторів є характер самого зображення – його розмір, роздільна здатність, кількість кольорів, розподіл кольорів, формат та ін. Іншим фактором є тип завдання, для якого необхідно обробити зображення – це може бути розпізнавання

об'єктів, зменшення шуму, видалення артефактів, покращення якості зображення та інші.

Також на обрання методів обробки впливає технічне обладнання, що використовується для обробки зображень. Наприклад, деякі методи можуть бути виконані тільки на потужних серверах, під час коли інші можуть бути виконані на звичайних ПК або мобільних пристроях. Також важливо враховувати наявність певних програмних інструментів для обробки зображень та їх можливості.

Окрім цього, на обрання методів обробки цифрових зображень можуть впливати певні вимоги до часу виконання операцій з обробки зображень. Деякі методи можуть бути виконані дуже швидко, під час коли інші можуть займати деякий час для виконання операцій обробки. Також на час виконання можуть впливати такі фактори, як кількість оброблюваних зображень, їх розмір та складність завдання.

Нарешті, на вибір методів обробки зображень можуть впливати індивідуальні вимоги та налаштування користувачів. Це можуть бути особливості відображення кольорів, якості відображення, рівня деталізації та інші параметри, які можуть бути важливі для конкретного користувача.

Першим етапом обробки зображення є введення даних в цифровому форматі в комп'ютер. Це може бути зроблено шляхом сканування або фотографування зображення, яке потім імпортується до програми для подальшої обробки.

Далі, зображення можна піддати різним видам обробки, включаючи корекцію кольору, видалення шуму, ретушування, налаштування контрасту та яскравості, зміну розміру та форматування для різних пристроїв та платформ.

Важливо також враховувати якість монітору, на якому працює фахівець з обробки зображень. Якщо монітор погано налаштований або має низьку

роздільну здатність, це може призвести до неправильної обробки зображення та зниження якості результату.

Крім того, обробка зображень може бути часомісткою та вимагати багато часу на завантаження, обробку та збереження кінцевого зображення. Тому важливо ретельно планувати процес обробки та використовувати ефективні інструменти та алгоритми для максимальної продуктивності.

При роботі з цифровими зображеннями також важливо враховувати колірний простір монітора з яким ви працюєте, оскільки це може вплинути на кінцевий результат обробки. Наприклад, якщо ви працюєте з фотографією, яка має широкий діапазон кольорів, і ваш монітор не може відобразити всі ці кольори, то ви можете втратити деяку інформацію під час обробки. Крім того, якщо ви плануєте друкувати своє зображення, то важливо враховувати колірний простір монітора, оскільки кольори на друкарському пристрої можуть відрізнитись від кольорів на моніторі.

Узагальнюючи, можна сказати, що обрання методів обробки цифрових зображень залежить від багатьох факторів, таких як призначення зображення, його тип та характеристики, умови зйомки або створення зображення, а також вимоги користувача. Щоб обрати найкращий метод обробки зображення, потрібно враховувати всі ці фактори та знаходити оптимальний баланс між ними. Важливо також знати основні параметри цифрового зображення та знати, як вони впливають на кінцевий результат.

1.7 Аналіз існуючих рішень

На ринку присутні багато рішень для автоматичної корекції фотографій на різних пристроях та в програмах. Наприклад, опція "Auto" в редакторі фотографій на iPhone - це автоматична корекція зображення, яка забезпечує оптимальну якість та кольорову збалансованість для вашої фотографії. Коли ви відкриваєте фотографію в редакторі, натиснувши на "Edit", ви можете вибрати опцію "Auto".

Для цього Apple використовує багато пов'язаних алгоритмів. Один з них - це алгоритм автоматичного збільшення різкості. Він визначає контур об'єктів на фотографії та збільшує їх різкість, покращуючи деталі. Це допомагає зменшити розмиття та зробити зображення більш чітким.

Інший алгоритм, що задіяний в опції auto, - це автоматична настройка балансу білого. Він забезпечує правильну настройку кольорової гами зображення, що робить фотографії більш реалістичними та природними.

Також у процесі автоматичної корекції фотографій задіяний алгоритм, що дозволяє виправити експозицію. Він визначає, чи потрібна корекція яскравості, тону та насиченості зображення, і виконує відповідні налаштування.

Використовуються також алгоритми для виправлення кривизни та перспективи, для покращення колірної гами та контрастності. Загалом, комбінація цих алгоритмів допомагає досягти оптимальної якості зображення без необхідності вручну регулювати налаштування.

Окрім Adobe Photoshop одним з подібних рішень є програма Adobe Lightroom, яка має вбудовані інструменти автоматичної корекції фотографій, такі як автотонування, автовиправлення перспективи, автоекспозиція та інші. Для цього використовуються алгоритми, подібні до тих, що використовуються на iPhone, такі як аналіз гистограми, контроль якості, корекція кольору та налаштування контрастності. В останніх версіях Adobe Lightroom автокорекція вже базується на машинному навчанні та може виявляти та автоматично виправляти проблеми з експозицією, тоном, балансом білого, насиченістю та іншими параметрами зображення [14].

Іншим подібним рішенням є програма Skylum Luminar, яка також має вбудовані інструменти автоматичної корекції фотографій, що базуються на аналізі зображення та його підтримці. Вона використовує алгоритми, які автоматично пристосовують колір, контрастність та експозицію, щоб зробити фотографії більш привабливими та якісними [15].

Ще одним прикладом подібного рішення є Google Photos, який має вбудований алгоритм автоматичної корекції фотографій, що базується на машинному навчанні. Цей алгоритм використовує нейронні мережі для виявлення та корекції різних елементів фотографії, включаючи контраст, колір та експозицію.

Capture One має функцію автоматичної корекції експозиції та кольорової корекції, яку можна застосувати до кожної фотографії окремо або до груп фотографій [16].

DxO PhotoLab має вбудований модуль Smart Lighting, який використовує машинне навчання для автоматичної корекції експозиції та виправлення тіней та світлих ділянок на фотографії. Він також має модуль ClearView Plus, який автоматично виправляє туман та покращує деталі на фотографії [17].

Усі ці рішення базуються на подібних алгоритмах, які використовують аналіз зображення та його параметрів, для того, щоб автоматично налаштувати його на максимально привабливий та якісний для користувача результат.

1.8 Постановка задачі

В даній роботі проводиться аналіз та розробка підсистеми для дослідження методів підвищення якості цифрових зображень.

Метою даної роботи є ознайомлення з основними принципами автоматичного покращення зображень. Основні вимоги до підсистеми дослідження методів підвищення якості цифрових зображень такі:

- Можливість подальшого розширення та модифікації підсистеми.
- Використання сучасних технологій.
- Прозорий та зрозуміло написаний алгоритм роботи.

1.9 Дерево функцій

Головна мета роботи є вершиною дерева функцій, в якому гілки представляють функції або завдання, що потрібно виконати для досягнення головної мети та підцілей. Функціональна декомпозиція є методом розбиття системи на функціональні компоненти. Кожен елемент системи розглядається окремо, а також його роль у контексті системи.

На рисунку 1.5 зображено структуру дерева функцій.



Рисунок 1.5 – Дерево функцій

2 МАТЕМАТИЧНІ МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ, ПРОГРАМНЕ ПРОЕКТУВАННЯ

2.1 Концептуалізація схеми обробки

Створення загальної схеми обробки для підсистеми є важливим етапом у розробці [18].

Перш ніж розпочати розробку схеми обробки, необхідно провести аналіз основних етапів обробки зображень для виявлення та вимірювання характеристик якості. Це можуть бути етапи попередньої обробки, такі як підсилення контрасту, фільтрація шуму, збільшення різкості тощо, або етапи порівняння, які залучають стандарти якості, метрики та моделі оцінки. Важливо врахувати особливості задачі та потреби користувача, оскільки різні домени можуть вимагати різних підходів до обробки та оцінки якості.

Слід також розглянути можливість використання сучасних алгоритмів машинного навчання та глибинного навчання для автоматичного виявлення та оцінки якості зображень. Такі алгоритми можуть навчитися розрізняти якісні та некомпліантні зображення, а також прогнозувати метрики якості без необхідності використання ручно підготовлених даних.

У процесі концептуалізації схеми обробки варто враховувати можливість гнучкої настройки параметрів обробки для досягнення оптимальних результатів відповідно до потреб користувача. Крім того, слід передбачити механізми контролю та зворотного зв'язку для покращення ефективності системи та забезпечення постійного вдосконалення якості зображень.

У підсумку, концептуалізація схеми обробки для підсистеми дослідження якості цифрових зображень передбачає визначення основних етапів обробки, вибір підходів та алгоритмів, врахування особливостей задачі, а також розробку гнучких механізмів контролю та зворотного зв'язку.

Цей підхід дозволить створити ефективну систему, здатну виявляти та оцінювати якість цифрових зображень з високою точністю та надійністю.

На рисунку 2.1 представлено сформовану концептуальну схему обробки зображень для використання в нашій підсистемі.

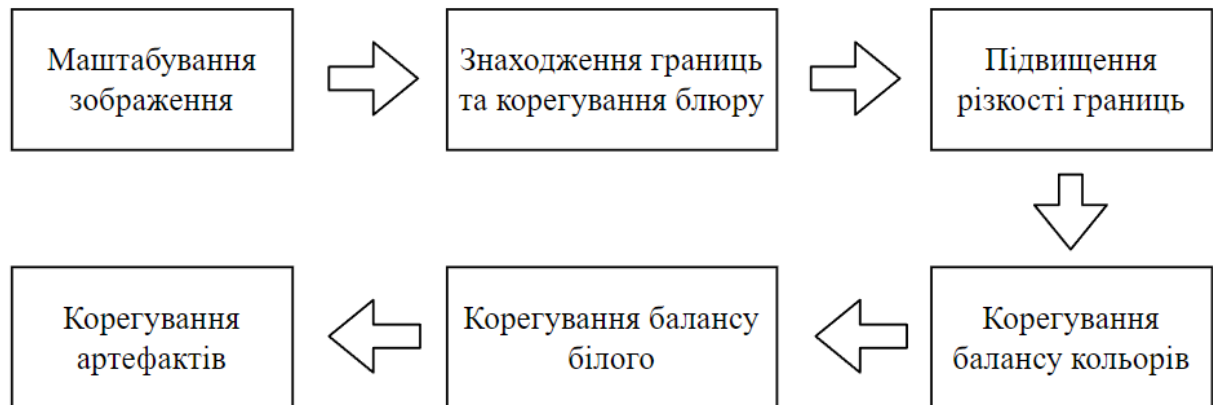


Рисунок 2.1 – Концептуальна схема

Сформувавши схему, ми повинні переходити до методів реалізації її компонентів. Але перед цим також важливо розглянути існуючі колірні моделі. Врахування колірних моделей в концептуалізації схеми обробки допоможе забезпечити більш повну та точну оцінку якості цифрових зображень, зокрема колірних аспектів.

Це дасть змогу створити більш комплексну та реалістичну систему дослідження якості цифрових зображень.

2.2 Моделі представлення зображень

Колірні моделі є системами, які використовуються для представлення кольорів у цифровій формі. Вони визначають спосіб, яким кольорова інформація кодується, зберігається та відтворюється на різних пристроях і застосунках.

Існує кілька основних колірних моделей, найпопулярнішими з яких є RGB (Red-Green-Blue), CMYK (Cyan-Magenta-Yellow-Key), HSL (Hue-Saturation-Lightness), HSV (Hue-Saturation-Value), LAB та Greyscale [19].

На рисунку 2.2 зображено наведені колірні моделі.

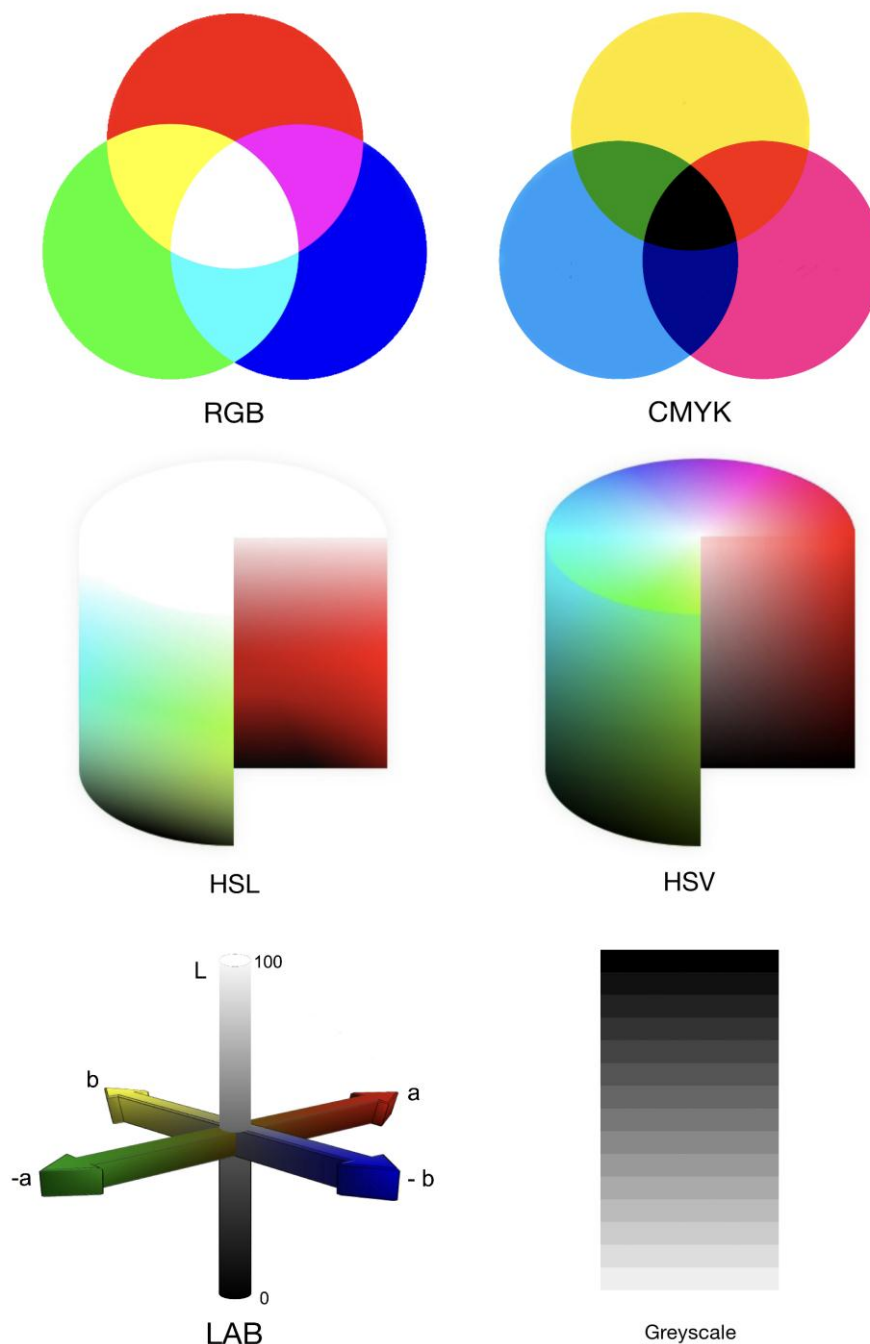


Рисунок 2.2 – Колірні моделі

Давайте розглянемо кожну наведену детальніше:

- Модель RGB (Red-Green-Blue)

Модель RGB базується на способі змішування червоного (Red), зеленого (Green) та синього (Blue) світла в різних пропорціях для створення багатьох кольорів. Кожен колір представлений трьома числовими

значеннями, які вказують на інтенсивність червоного, зеленого та синього компонентів. Значення зазвичай знаходяться в діапазоні від 0 до 255. Наприклад, (255, 0, 0) вказує на повний червоний колір, (0, 255, 0) - на повний зелений колір, а (0, 0, 255) - на повний синій колір.

- Модель CMYK (Cyan-Magenta-Yellow-Key)

Модель CMYK використовується в друкарстві та дозволяє отримати широкий спектр кольорів шляхом змішування чотирьох основних кольорів - бірюзового (Cyan), пурпурового (Magenta), жовтого (Yellow) та ключового (Key) (також відомого як чорний). Ця модель базується на відображенні кольорів шляхом використання фарб або барвників. Значення кожного компонента можуть бути від 0 до 100, де 0 означає відсутність кольору, а 100 - його повну насиченість.

- Модель HSL (Hue-Saturation-Lightness)

Модель HSL визначає колір за допомогою трьох компонентів: відтінку (Hue), насиченість (Saturation) та світлоти (Lightness). Відтінок вказує на конкретне положення кольору на колірному колесі та представляється значенням від 0 до 360 градусів. Насиченість визначає насиченість кольору та знаходиться в діапазоні від 0 до 100%. Світлота вказує на яскравість кольору та також представлена значенням від 0 до 100%.

- Модель HSV (Hue-Saturation-Value)

Модель HSV схожа на модель HSL, але замість світлоти використовується значення value (значення). Значення вказує на яскравість кольору та також знаходиться в діапазоні від 0 до 100%. Відтінок та насиченість визначаються так само, як і в моделі HSL.

- Модель LAB

Модель LAB є іншою кольоровою моделлю, яка використовується для опису кольорів. Вона базується на трьох компонентах: світлоті (L), зелено-червоному (a) та синьо-жовтому (b).

Компонент L визначає яскравість кольору, де значення L знаходиться в діапазоні від 0 до 100. Значення 0 відповідає для абсолютно чорного кольору, тоді як 100 відповідає для абсолютно білого кольору.

Компоненти a та b представляють два перпендикулярних напрямки в просторі кольорів. Компонент a відображає відстань від зеленого до червоного кольору, де від'ємне значення вказує на зелений колір, а позитивне - на червоний. Компонент b відображає відстань від синього до жовтого кольору, де від'ємне значення вказує на синій колір, а позитивне - на жовтий.

Дана модель забезпечує більш однорідне розподілення кольорів у порівнянні з моделлю RGB. Вона також здатна враховувати особливості сприйняття кольорів людиною, оскільки розділяє яскравість та колірність.

Для перетворення кольору з моделі RGB до моделі LAB, спочатку виконується перетворення з RGB до XYZ, а потім з XYZ до LAB. Це дозволяє отримати кольоровий простір, який незалежний від пристрою та більш придатний для обробки та аналізу кольору.

- Greyscale

Модель Greyscale (від англ. "відтінки сірого") використовується для представлення зображень у відтінках сірого кольору. У відтінках сірого відсутні окремі компоненти кольору, такі як червоний, зелений і синій. Замість цього, кожен піксель представлений лише одним значенням яскравості.

Значення яскравості для кожного пікселя зазвичай виражається у відсотках або у числовому діапазоні від 0 до 255, де 0 відповідає абсолютній чорності, а 255 - абсолютній білизні. Значення між цими крайніми точками представляють відтінки сірого від темного до світлого.

Наприклад, значення яскравості 0 відповідає чорному кольору, 127 - середньому сірому, а 255 - білому кольору. Всі інші проміжні значення представляють відтінки сірого між цими крайніми точками.

Модель Greyscale зазвичай використовується, коли відсутня необхідність передавати повноцінні кольори, і можна обійтися лише інформацією про яскравість. Вона широко використовується в чорно-білих фотографіях, монохромних зображеннях, друкованих матеріалах та багатьох інших випадках, де кольори не мають принципового значення.

Кожна з цих колірних моделей має свої переваги та застосування в різних галузях. Наприклад, модель RGB широко використовується в комп'ютерній графіці та електронних пристроях, таких як монітори та телевізори, тоді як модель CMYK популярна в друкарстві. Моделі HSL та HSV надають зручніші способи роботи з кольорами, особливо при виконанні операцій, таких як зміна відтінку, насиченості або яскравості.

Наведені вище колірні моделі є лише деякими прикладами. Існує також безліч інших моделей, таких як XYZ, YUV та інші, кожна з цих колірних моделей має свої переваги і використовується в різних ситуаціях. Вони дозволяють точно представляти кольори і працювати з ними у цифрових середовищах, допомагаючи дизайнерам, фотографам та іншим фахівцям працювати з кольорами ефективно і точно.

2.3 Бікубічна інтерполяція

При наявності зображення з малою роздільною здатністю для здійснення ефективної роботи над ним, потрібно збільшити його вихідні розміри.

Бікубічний алгоритм розглядає найближче оточення - 4x4 відомих пікселів, тобто загалом 16 пікселів. Оскільки вони знаходяться на різних відстанях від невідомого пікселя, ближчим пікселям надається більша вага в обчисленні. Даний алгоритм створює помітно чіткіші зображення, ніж попередні два методи, і, мабуть, є ідеальним поєднанням часу обробки та якості виведення. З цієї причини він є стандартом у багатьох програмах для

редагування зображень (включаючи Adobe Photoshop), драйверах принтерів та інтерполяції у камерах [20].

Замість малювання ліній між точками кубічна сплайн-інтерполяція малює гладкі криві. Ці гладкі криві є поліномами 3-го ступеня.

Щоб розрахувати чотири коефіцієнти a , b , c і d , застосовується наступна система рівнянь:

$$f(x) = ax^3 + bx^2 + cx + d$$

Умови для розрахунку коефіцієнтів:

Значення функції на перших двох точках:

$$f(x_0) = a(x_0)^3 + b(x_0)^2 + c(x_0) + d = f_0$$

$$f(x_1) = a(x_1)^3 + b(x_1)^2 + c(x_1) + d = f_1$$

Значення похідних функції у перших двох точках:

$$f'(x_0) = 3ax_0^2 + 2bx_0 + c = m_0$$

$$f'(x_1) = 3ax_1^2 + 2bx_1 + c = m_1$$

Значення функції на останніх двох точках:

$$f(x_2) = a(x_2)^3 + b(x_2)^2 + c(x_2) + d = f_2$$

$$f(x_3) = a(x_3)^3 + b(x_3)^2 + c(x_3) + d = f_3$$

Значення похідних функції у останніх двох точках:

$$f'(x_2) = 3ax_2^2 + 2bx_2 + c = m_2$$

$$f'(x_3) = 3ax_3^2 + 2bx_3 + c = m_3$$

Розв'язавши цю систему рівнянь, можна отримати значення коефіцієнтів a , b , c і d для бікубічної інтерполяції.

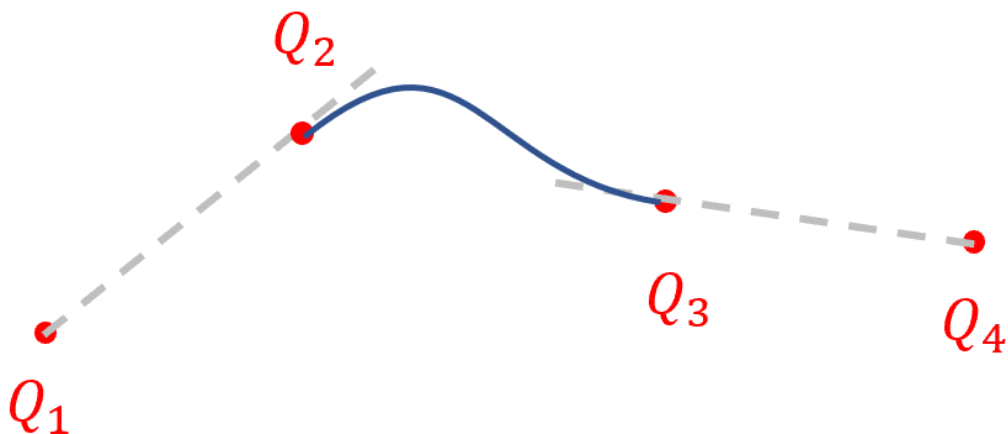


Рисунок 2.3 – Приклад бікубічної інтерполяції чотирьох точок.

Умови можна перевести у формули, обчисливши похідні та встановивши їх рівними градієнтам, а також шляхом оцінки кубічної формули у внутрішніх сусідніх точках. У 2D це включатиме 16 точок.

Ми виконаємо кубічну сплайн-інтерполяцію на 4 рядках точок, а потім виконаємо останню кубічну сплайн-інтерполяцію на нових 4 інтерпольованих точках, графічно це представлено на рисунку 2.4.

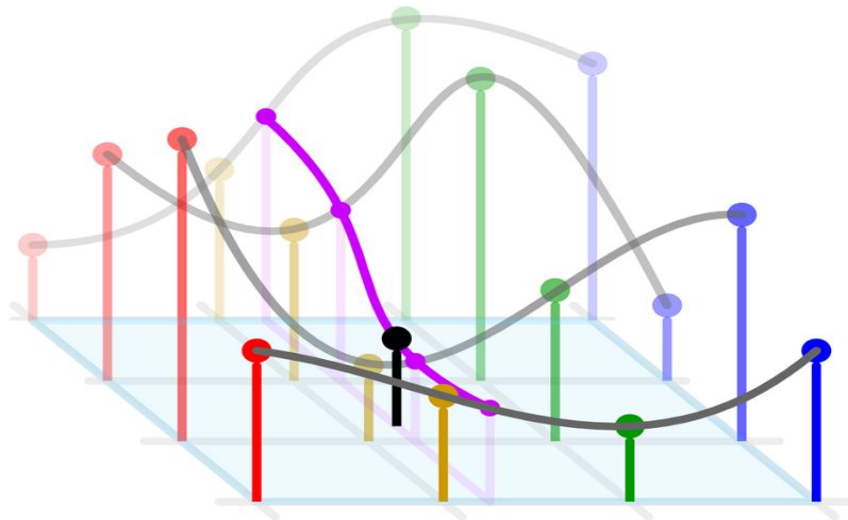


Рисунок 2.4 – Приклад бікубічної інтерполяції на площині

2.4 Фільтр Собеля

Даний фільтр - це вже класичний алгоритм обробки зображень, який застосовується для виявлення границь та зміни контрастності на зображенні. Він є одним з найбільш поширених алгоритмів обробки зображень і часто використовується в комп'ютерному зорі, виявленні границь та інших застосуваннях.

Алгоритм працює на основі апроксимації градієнту яскравості зображення. Він використовує два ядра або маски, які складаються з малих чисел, щоб обчислити похідні по горизонталі і вертикалі.

Ядро для обчислення горизонтальної похідної має вигляд:	-1	0	1
	-2	0	2
	-1	0	1
Ядро для обчислення вертикальної похідної має вигляд:	-1	-2	-1
	0	0	0
	1	2	1

Рисунок 2.5 – Типові матриці/ядра для обчислення похідних

Ці похідні репрезентують інтенсивність змін яскравості у напрямку, паралельному відповідній осі, приклад наведено на рисунку 2.5.

Для застосування його до зображення, кожен піксель зображення проходить фільтрацію з використанням цих двох ядер. Для кожного пікселя обчислюється градієнт яскравості, який є вектором з горизонтальною та вертикальною компонентами. Горизонтальна компонента градієнту представляє зміни яскравості в горизонтальному напрямку, а вертикальна компонента - в вертикальному напрямку [21].

Після обчислення градієнту для всіх пікселів зображення, використовуються різні методи для подальшої обробки результатів. Зазвичай використовується порогове значення, яке визначає, які зміни яскравості вважатимуться границями. Пікселі, які мають градієнт вище порогового значення, вважаються границями, інші пікселі - фоном.

Фільтр Собеля є ефективним і простим алгоритмом для виявлення границь на зображенні. Він дозволяє виділити контури об'єктів, що дозволяє більш точно застосовувати інші фільтри або алгоритми для поліпшення візуальної якості, що робить його корисним інструментом в багатьох областях, таких як обробка зображень, комп'ютерне зір, розпізнавання образів та інших.

2.5 Фільтр Гаусса

Фільтр Гаусса - це один з найпоширеніших фільтрів для розмиття зображень. Він базується на використанні гаусівської функції для створення розмиття згідно з розподілом Гаусса. Фільтр Гаусса використовується для зменшення шуму та згладжування різких переходів яскравості, що допомагає покращити якість зображення та підготувати його для подальшої обробки.

Процес застосування фільтра Гаусса до зображення полягає в наступних кроках:

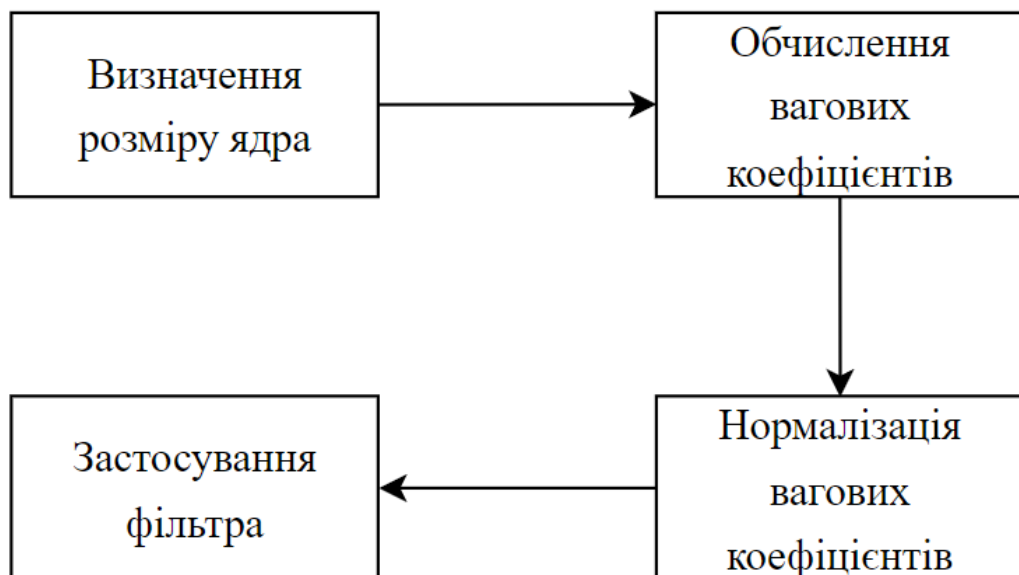


Рисунок 2.6 – Типові кроки для реалізації фільтра Гаусса

- **Визначення розміру ядра:**

Обирається розмір ядра (фільтру) Гаусса, який визначає ширину розмиття. Більший розмір ядра/матриці призводить до більшого розмиття зображення.

- **Обчислення вагових коефіцієнтів:**

Для створення фільтра Гаусса потрібно обчислити вагові коефіцієнти, які відповідають значенням гаусівської функції відповідно до відстані від

центрального пікселя. Ці коефіцієнти визначають, які значення яскравості мають внесок до кінцевого розмиття.

- Нормалізація вагових коефіцієнтів:

Після обчислення вагових коефіцієнтів їх необхідно нормалізувати, щоб забезпечити збереження суми всіх коефіцієнтів рівною одиниці. Це досягається поділом кожного коефіцієнта на суму всіх коефіцієнтів.

Для кожного пікселя зображення виконуються наступні кроки:

- Береться вікно пікселів, включаючи його сусідів, в розмірі ядра Гаусса.
- Множиться значення кожного пікселя вікна на відповідний ваговий коефіцієнт фільтра.
- Сумуються отримані значення пікселів та діляться на суму їх вагових коефіцієнтів.
- Призначається отримане значення центральному пікселю.
- Аналогічний процес повторюється для інших пікселів на зображенні.

Цей процес розмиття зображення за допомогою фільтра Гаусса дозволяє згладити шуми і видалити дрібні деталі, створюючи більш розмиті переходи яскравості та границі. Використання правильного розміру ядра і вагових коефіцієнтів важливо для досягнення бажаного ефекту розмиття [22].

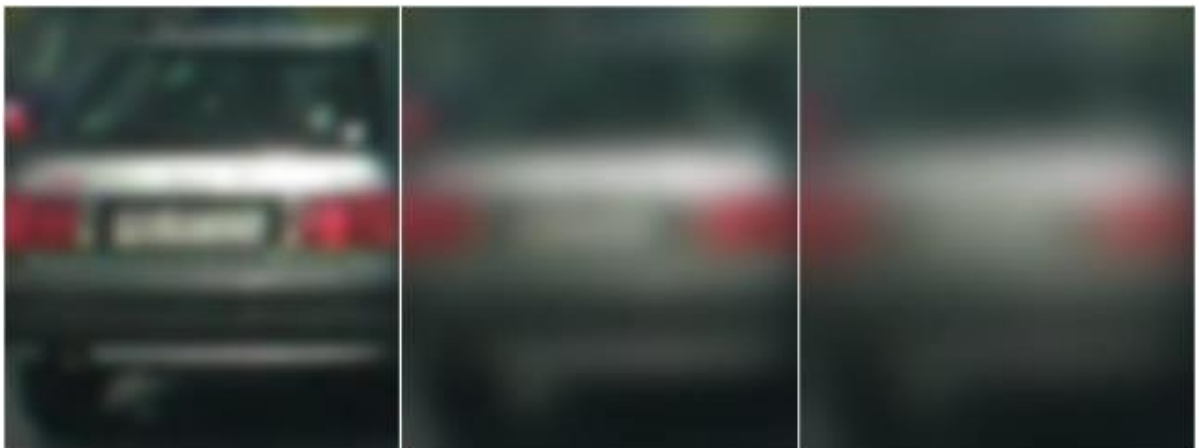


Рисунок 2.7 – Приклад використання різних вагових коефіцієнтів для розмиття

2.6 Алгоритм Кенні

Алгоритм Кенні (Canny edge detector) є одним з найбільш використовуваних алгоритмів для виявлення контурів на зображенні. Вперше введений Джоном Кенні в 1986 році, він знайшов широке застосування у комп'ютерному зору, обробці зображень та візуальних системах [23].

Основна мета алгоритму Кенні полягає в тому, щоб знайти та виділити границі об'єктів на зображенні, зводячи шум та деталі до мінімуму. Схема із базовим порядком дій в алгоритмі наведено в схемі 2.8.

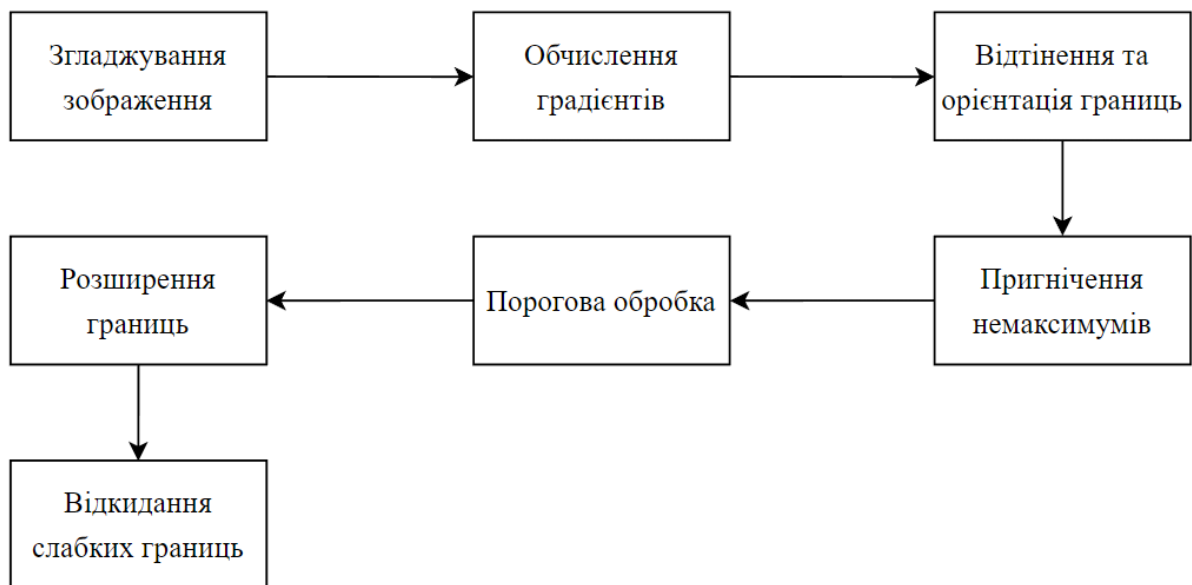


Рисунок 2.8 – Базовий порядок дій алгоритму Кенні

- Згладжування зображення:

Спочатку зображення піддається фільтрації для зменшення шуму. Зазвичай використовується гаусівський фільтр, який згладжує зображення шляхом розподілу інтенсивності пікселів у малому околі кожного пікселя.

- Обчислення градієнтів:

Після згладжування зображення використовується диференціальний оператор (наприклад, оператор Собеля) для обчислення градієнтів інтенсивності пікселів в горизонтальному та вертикальному напрямках. Це

дозволяє виявити місця, де інтенсивність різко змінюється, що може вказувати на границі об'єктів.

- Відтінення та орієнтація границь:

За допомогою обчислених градієнтів, кожен піксель отримує відтінення та орієнтацію границі. Відтінення показує силу границі, тоді як орієнтація вказує на напрямок границі.

- Пригнічення немаксимумів:

Цей етап пригнічує слабкі границі, які не є локальними максимумами інтенсивності вздовж напрямку градієнту. Ідея полягає в тому, щоб відфільтрувати пікселі, які не є частинами тонких границь, а залишити лише ті, які є потенційними краями.

- Порогова обробка:

Застосування порогового значення для відокремлення сильних і слабких границь. Пікселі з відтіненням, більшим за високий поріг, вважаються сильними границями, тоді як ті, які знаходяться між низьким і високим порогами, вважаються слабкими границями.

- Розширення границь:

Сильні границі розширюються, щоб включити сусідні слабкі границі. Це допомагає з'єднати перерви в границях та утворити повні контури.

- Відкидання слабких границь:

На цьому етапі відкидаються всі слабкі границі, які не з'єднані з сильними границями. Це допомагає зменшити шум та залишити тільки впевнені контури.

В результаті виконання всіх цих етапів алгоритм Кенні дає змогу виявити тонкі та точні границі об'єктів на зображенні. Його параметри, такі як порогові значення та розмір фільтра, можуть бути налаштовані в залежності від конкретних вимог додатку. Використовуючи алгоритм Кенні, можна досягти високої точності та надійності виявлення контурів на

зображеннях у різних застосуваннях, включаючи розпізнавання облич, виявлення об'єктів та стеження за рухом.

2.7 Стандартне відхилення

Стандартне відхилення, також відоме як стандартне відхилення вибірки або стандартне відхилення випадкової величини, є одним з основних показників розподілу даних. Воно вимірює розкид величини від середнього значення і надає уявлення про те, наскільки точним є середнє значення відносно окремих спостережень.

Стандартне відхилення використовується для кількох цілей. Воно дозволяє оцінити ризик або неоднорідність даних, порівнювати різні набори даних та виявляти відхилення від очікуваного значення. Чим вище значення стандартного відхилення, тим більше варіації (розкиду) в даних [24].

Для обчислення стандартного відхилення необхідно виконати наступні кроки:

- Обчислити середнє значення вибірки (суму всіх значень, поділену на кількість спостережень).
- Для кожного значення вибірки відняти середнє значення та піднести різницю до квадрату.
- Обчислити середнє арифметичне всіх квадратів різниць.
- Взяти квадратний корінь отриманого значення з попереднього кроку. Це і буде стандартним відхиленням.

Формульно стандартне відхилення може бути записане так:

$$\sigma = \sqrt{(\sum(x_i - \mu)^2 / N)}$$

де:

σ (грецька літера "сигма") - стандартне відхилення,

x_i - кожне окреме значення вибірки,

μ (грецька літера "мю") - середнє значення вибірки,

Σ - сумування всіх значень,

N - кількість спостережень у вибірці.

Стандартне відхилення вимірюється в тих же одиницях, що і вихідні дані. Воно дозволяє зрозуміти, як далеко варіюються дані від середнього значення і як вони розподілені навколо нього.

Це допомагає аналізувати дані, розуміти їхню варіабельність та допомагає в прийнятті рішень на основі статистичних даних.

2.8 Фільтр Лапласа

Також відомий як фільтр другої похідної або фільтр швидкості зміни, є одним із численних фільтрів, використовуваних в обробці зображень. Він використовується для виявлення різкості зображень, країв та границь.

На рисунку 2.9 зображено алгоритм роботи Фільтра Лапласа.

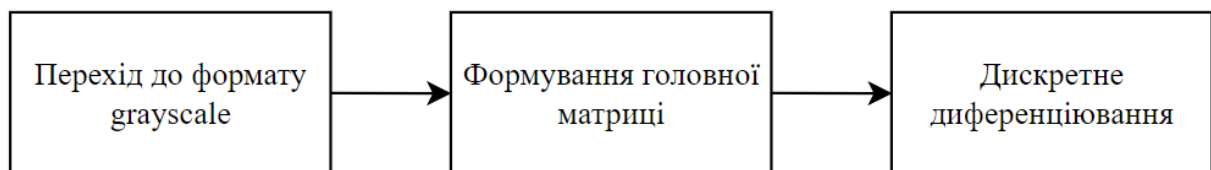


Рисунок 2.9 – Порядок застосування фільтра Лапласа

Перш за все, фільтр потребує, щоб вихідне зображення було у форматі відтінків сірого (grayscale), тому якщо вхідне зображення має кольоровий формат, спочатку його необхідно перетворити відповідним чином.

Даний фільтр базується на концепції дискретного диференціювання. Він використовує спеціальне ядро (матрицю), яке застосовується до кожного пікселя зображення. Це ядро має форму квадратної матриці з вагами, розташованими навколо центрального пікселя.

Він може мати кілька різних варіацій ядра, таких як ядро Лапласа або Якобіан. Однак, основна ідея полягає в тому, що ваги в ядрі зосереджені навколо центрального пікселя, і вони відображають зміни яскравості між центральним пікселем та його сусідами.

Застосовуючи ядро до кожного пікселя зображення, виконується операція згортки, де значення ядра помножені на значення пікселів у відповідних позиціях та сумуються. Це призводить до отримання нового значення для центрального пікселя, яке відображає суму змін яскравості в його оточенні.

Після застосування фільтра до всього зображення, отримується нове зображення, де кожен піксель відображає зміни яскравості та градієнти на вихідному зображенні.

Спектр застосування фільтра Лапласа:

- Виявлення країв:

Даний фільтр виявляє різкі зміни яскравості у зображенні, що допомагає виявити краї об'єктів.

- Підсилення деталей:

Застосування фільтра до зображення може допомогти виділити дрібні деталі та текстуру об'єктів.

- Виявлення артефактів:

Завдяки своїй здатності виявляти різкості зміни яскравості, фільтр Лапласа може використовуватися для виявлення артефактів у зображеннях.

Важливо відзначити, що даний фільтр може підсилити шум на зображенні. Тому часто використовуються додаткові методи для попередньої обробки та після обробки зображення, щоб зменшити цей ефект і покращити якість результату [25].

2.9 Медіанний фільтр

Медіанний фільтр - це один з найпоширеніших фільтрів, що використовуються для обробки зображень. Він використовується для зменшення шуму та згладжування зображень [26].

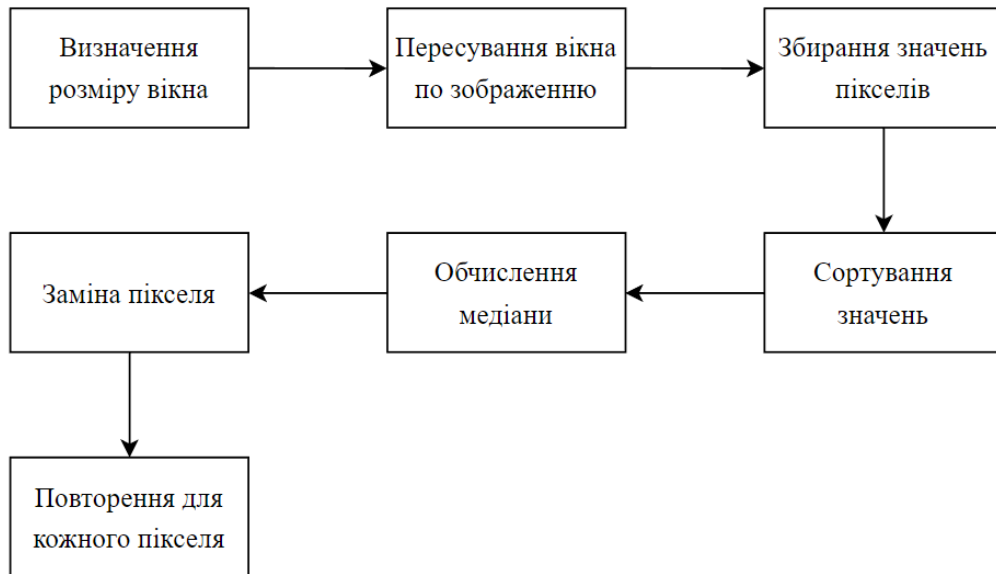


Рисунок 2.10 – Порядок застосування медіанного фільтру

Основна ідея медіанного фільтру полягає в тому, щоб замінити кожний піксель у зображенні на медіану значень пікселів у його околі.

На рисунку 2.10 розглядається алгоритм роботи медіанного фільтру.

- Визначення розміру вікна:

Спочатку вам потрібно визначити розмір вікна, яке буде використовуватися для обчислення медіани. Зазвичай використовують квадратне вікно, наприклад, 3×3 , 5×5 або 7×7 пікселів.

- Пересування вікна по зображенню:

Вікно розміром $N \times N$ пікселів пересувається по всьому зображенню. Для кожного пікселя вікно накриває його околиці.

- Збирання значень пікселів:

Зібравши всі значення пікселів, які перебувають під вікном, створюється список значень.

- Сортування значень:

Зібрані значення сортуються в порядку зростання або спадання.

- Обчислення медіани:

Медіана обчислюється як значення, що перебуває посередині відсортованого списку значень. Якщо вікно має непарний розмір, то медіана - це значення, що знаходиться саме посередині. У випадку парного розміру вікна медіану можна обчислити, взявши середнє арифметичне двох центральних значень.

- Заміна пікселя:

Значення центрального пікселя вікна замінюється обчисленою медіаною.

- Повторення для кожного пікселя:

Описані кроки повторюються для кожного пікселя в зображенні. Таким чином, кожен піксель замінюється медіаною значень пікселів у його околі.

Медіанний фільтр є досить ефективним для зменшення шуму на зображеннях, особливо тих, що мають випадковий або соль і перець шум. Він здатний згладжувати зображення, зберігаючи при цьому різкі краї та деталі.

Важливо зазначити, що медіанний фільтр може впливати на розмір зображення, оскільки він замінює кожен піксель на медіану значень пікселів у його околі. Тому результуюче зображення буде трохи меншим у порівнянні з вихідним.

Медіанний фільтр дуже широко використовується в обробці зображень, включаючи обробку медичних зображень, задачах пов'язаних з комп'ютерним зором, та багатьма іншими.

2.10 Нерізкість маскуваня

Нерізкість маскуваня (англ. unsharp masking) - це техніка обробки зображень, яка використовується для підвищення різкості та деталізації зображень. Цей метод зазвичай застосовується в графічних редакторах та фотообробних програмах.

Назва походить від того факту, що техніка використовує розмите, або «нерізде», негативне зображення для створення маски вихідного зображення.

Приклад процесу нерізкості маскування наведено на рисунку 2.11.

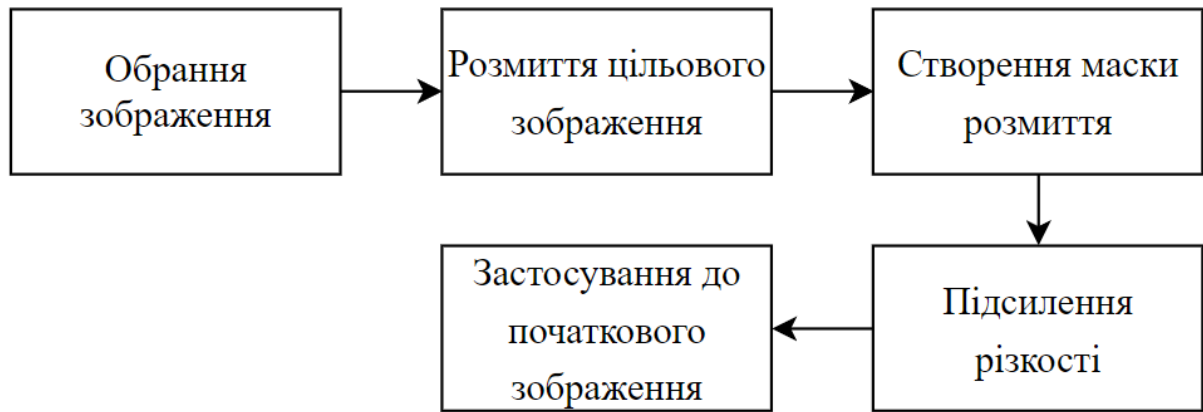


Рисунок 2.11 – Порядок застосування техніки нерізкості масштабування

- **Обрання зображення:**

Спочатку ми маємо початкове зображення, яке може містити розмиття або нечіткість. Це може бути через низьку якість об'єктиву фотокамери, рух під час зйомки або інші фактори, які спричиняють втрату різкості деталей.

- **Розмиття цільового зображення:**

Створюється розмитий еквівалент початкового зображення. Це може бути зроблено, наприклад, за допомогою гаусового фільтра, який використовується для видалення високочастотних складових зображення, залишаючи лише низькочастотні складові.

- **Створення маски розмиття:**

Створюється маска розмиття шляхом віднімання розмитого зображення від початкового зображення піксель за пікселем. Це дає нам інформацію про ті частини зображення, які були розмиті або змінені під час процесу розмиття.

- Підсилення різкості:

Маска розмиття розширюється або змінюється для підсилення різкості зображення. Це може бути зроблено, наприклад, шляхом збільшення контрасту або збільшення яскравості деталей в областях зображення, де присутній значний градієнт.

- Застосування до початкового зображення:

Змінена маска розмиття додається до початкового зображення для покращення різкості. Це може бути зроблено шляхом додавання піксельних значень зміненої маски розмиття до відповідних пікселів початкового зображення.

В результаті процесу нерізкості маскування отримується зображення, в якому різкість та деталізація підвищуються. Це досягається шляхом виділення контурів та ребер на зображенні та підсилення їх яскравості та контрастності. В результаті зображення стає більш чітким і деталізованим, що часто призводить до більш живого та привабливого візуального ефекту.

Важливо відзначити, що процес нерізкості маскування може бути налаштований залежно від вимог і вподобань користувача. Різні програми та редактори можуть надавати різні налаштування та параметри для контролю нерізкості маскування, такі як радіус розмиття, рівень підсилення різкості тощо.

У практичному застосуванні, нерізкість маскування може бути корисною для вдосконалення фотографій, особливо тих, які були розмите або мають низьку різкість деталей. Вона може бути також використана для покращення якості зображень перед їх друком або публікацією в Інтернеті.

2.11 Передбачення сірого світу

Передбачення сірого світу (англ. Gray World Assumption) - це припущення, що в освітленні зображення кольори розподіляються рівномірно, тобто середнє значення каналів кольору (червоного, зеленого та

синього) еквівалентне середньому значенню серого. Це припущення використовується в цифровій обробці зображень для автоматичного виправлення кольорового балансу та корекції кольорів.

Процес передбачення сірого світу може зазначено на рисунку 2.12.

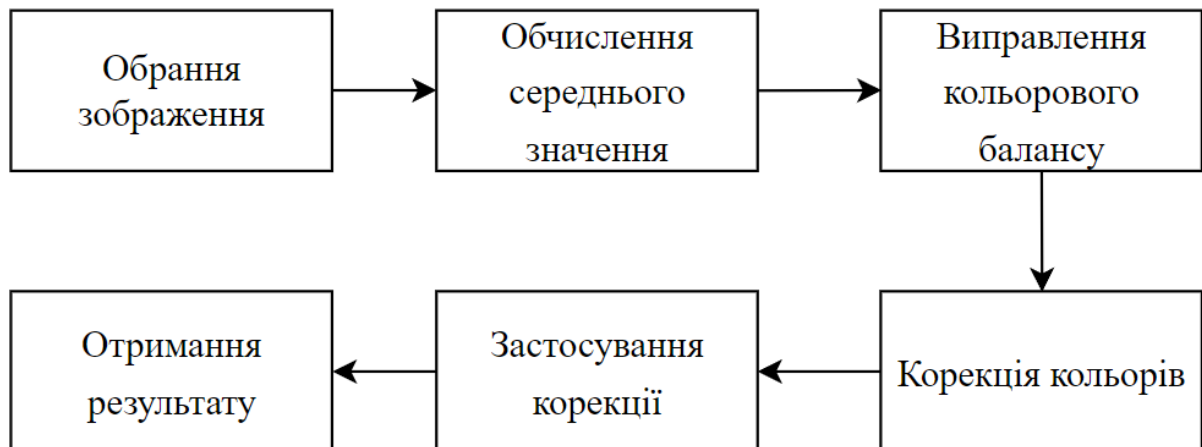


Рисунок 2.12 – Алгоритм реалізації передбачення сірого світу

- Обрання зображення:

Спочатку ми маємо цифрове зображення, яке складається з трьох каналів кольору: червоного (R), зеленого (G) та синього (B). Кожен канал представляє інтенсивність пікселів відповідного кольору.

- Обчислення середнього значення:

Обчислюється середнє значення кожного каналу кольору (R_{avg} , G_{avg} , B_{avg}), що відповідає середньому значенню інтенсивності пікселів у цьому каналі.

- Виправлення кольорового балансу:

За передбаченням сірого світу, ми припускаємо, що середнє значення каналів кольору має бути однаковим. Тому ми обчислюємо середнє значення серед усіх каналів ($avg = (R_{avg} + G_{avg} + B_{avg}) / 3$).

- Корекція кольорів:

Для корекції кольорового балансу ми використовуємо відношення між середнім значенням серед усіх каналів (avg) та кожним окремим значенням каналу кольору (R_avg, G_avg, B_avg). Коефіцієнти корекції обчислюються як коефіцієнти масштабування:

$$\text{red_gain} = \text{avg} / \text{R_avg}; \text{green_gain} = \text{avg} / \text{G_avg}; \text{blue_gain} = \text{avg} / \text{B_avg}$$

- Застосування корекції:

Кожне значення пікселя в кожному каналі кольору множиться на відповідний коефіцієнт корекції. Це здійснюється для всіх пікселів у зображенні.

- Отримання результату:

Після застосування корекції кольорів зображення отримує більш вірний кольоровий баланс, де кольори відтворюються більш точно.

Цей метод є одним з найпростіших підходів для автоматичної корекції кольорового балансу. Його основна перевага полягає в тому, що він не вимагає від користувача вручну задавати точні значення білого кольору або використовувати калібровку обладнання.

2.12 Ліміт розтягування

Ліміт розтягування (англ. stretchlim) - це метод обробки зображень, який використовується для автоматичного встановлення порогового значення для бінаризації зображення. Ліміт розтягування дозволяє вибрати оптимальний поріг, враховуючи гістограму яскравості зображення. Цей метод є ефективним для сегментації зображень та виділення об'єктів на зображенні. Алгоритм розглядаемого методу наведено на рисунку 2.13.

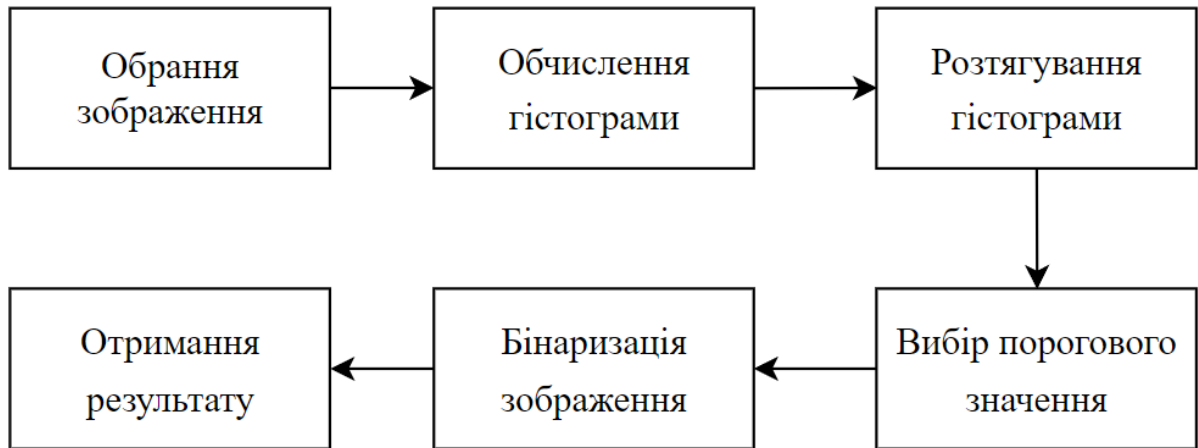


Рисунок 2.13 – Реалізація ліміту розтягування

- Обрання зображення:

Спочатку ми маємо чорно-біле або відтінкове зображення, на якому хочемо застосувати бінаризацію.

- Обчислення гистограми:

Обчислюється гистограма яскравості зображення. Гистограма представляє розподіл інтенсивності пікселів на зображенні. Вона показує, скільки пікселів мають певні значення яскравості.

- Розтягування гистограми:

Гистограма розтягується на всій шкалі яскравості. Це означає, що найтемніші пікселі стають ще темнішими, а найсвітліші пікселі стають ще світлішими. Розтягування гистограми допомагає розширити діапазон яскравості зображення та покращити контрастність.

- Вибір порогового значення:

Після розтягування гистограми виконується вибір оптимального порогового значення для бінаризації. Це зроблено на основі форми та характеристик гистограми. Зазвичай для ліміту розтягування використовуються відсотки значень яскравості, такі як нижні або верхні відсотки гистограми.

- Бінаризація зображення:

Зображення бінаризується, використовуючи вибране порогове значення. Усі пікселі, яскравість яких перевищує поріг, вважаються білими, а решта - чорними. Це призводить до розділення зображення на дві категорії: об'єкти та фон.

- Отримання результату:

В результаті застосування ліміту розтягування ми отримуємо бінаризоване зображення, де об'єкти виділені на фоні. Цей метод допомагає візуально розділити об'єкти на зображенні, спрощує подальшу обробку та аналіз, наприклад, в області комп'ютерного зору або обробки зображень.

Важливо зазначити, що результат бінаризації може залежати від вибору порогового значення. В реалізаціях методу ліміту розтягування можуть бути використані додаткові параметри або евристики для автоматичного вибору оптимального порогу або його налаштування користувачем.

2.13 Побудова моделі

Наведені методи треба поєднати в порядок попередньої обробки зображення. Вихідний опис початкового інформаційного образу має бути максимально пристосований для зберігання, передачі й аналізу (прийняття рішення на основі цієї інформації).

Рисунок 2.14, зображає всі застосування описаних методів, буде корисним в подальшій роботі, бо надає візуальне уявлення про послідовність обробки зображень та методи для реалізації відповідних дій.

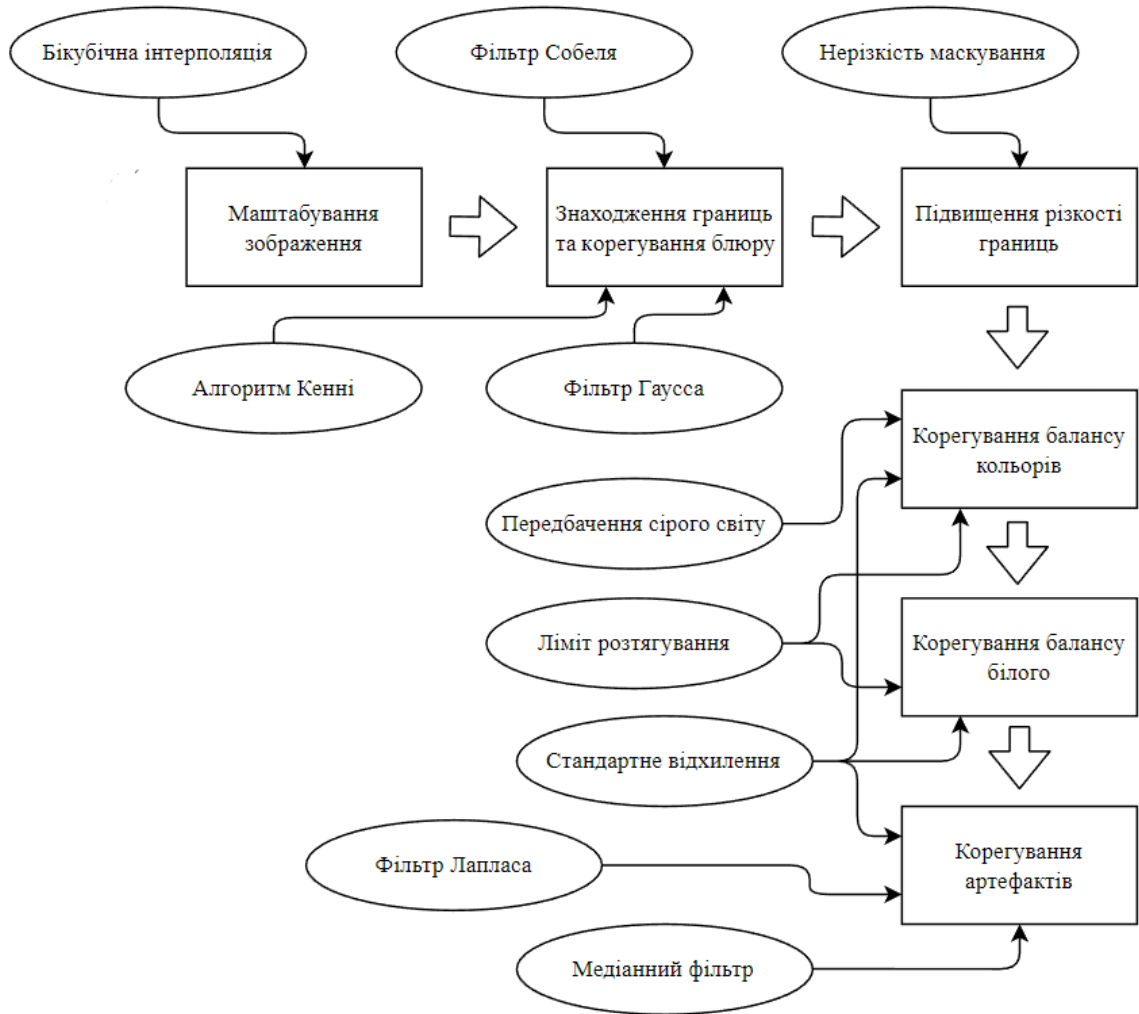


Рисунок 2.14 – Схема використання описаних методів

3. МАТЕМАТИЧНІ МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ, ПРОГРАМНЕ ПРОЕКТУВАННЯ

3.1 Вибір технічної платформи

Серед таких популярних програм, які ми можемо використати, таких як Adobe Photoshop, MATLAB і MathCAD, існує чимало відмінностей, які можуть вплинути на ваш вибір залежно від вашої специфічної потреби та експертного рівня.

Adobe Photoshop вважається однією з найпотужніших та найвідоміших програм для редагування та обробки фотографій. Вона має безліч інструментів, фільтрів та можливостей, що дозволяють вам виконувати навіть найскладніші завдання. Photoshop надає велику свободу творчості і варіативність, яка дозволяє користувачеві виконувати різні редагувальні процеси, від виправлення помилок до створення вражаючих графічних ефектів. Багато художників та професіоналів віддають перевагу Photoshop завдяки його зручному інтуїтивному інтерфейсу та можливості роботи з різними шарами, що дозволяє контролювати кожен аспект редагування.

MATLAB, з іншого боку, є мовою програмування та середовищем для числових обчислень та аналізу даних. Хоча вона не є прямою фотографічною програмою, MATLAB надає потужні інструменти для обробки та аналізу зображень. За допомогою MATLAB ви можете виконувати складні обчислення, застосовувати алгоритми машинного навчання та розробляти нові методи обробки зображень. MATLAB часто використовується в академічних дослідженнях та інженерних задачах, де потрібен глибокий аналіз даних та моделювання.

MathCAD - це інженерний математичний пакет, який також може бути використаний для роботи з фотографіями, але з фокусом на математичних обчисленнях і аналізі. Цей програмний продукт дозволяє користувачам виконувати складні математичні обчислення, побудову графіків та виконання

чисельних аналізів. Його інтуїтивний інтерфейс та можливість використовувати математичні формули дозволяють легко працювати з фотографіями та виконувати різні обчислювальні процеси.

Отже, вибір між Adobe Photoshop, MATLAB і MathCAD залежить від наших конкретних потреб. Якщо ви шукаєте потужну програму для редагування та обробки фотографій, Adobe Photoshop може бути найкращим варіантом. Якщо ви бажаєте виконувати складні аналізи даних та розробляти нові методи обробки зображень, то MATLAB буде корисним. Якщо ваші потреби включають в себе інженерний аналіз та обчислення, а також використання математичних формул, MathCAD може бути відмінним вибором.

З оглядом на нашу мету, платформою для реалізації проекту було обрано MATLAB. Даний застосунок є однією з найпопулярніших технічних платформ для створення технічних проектів у багатьох галузях, включаючи обробку зображень. Його багатофункціональність, потужність і дружній інтерфейс роблять його ідеальним інструментом для розробки модулів дослідження та підвищення якості отриманого зображення.

3.1 Структура технічної платформи

MATLAB (Matrix Laboratory) - це інтерактивне середовище програмування та обчислень, яке широко використовується у багатьох наукових, інженерних та математичних областях. Основною структурою MATLAB є мова програмування, вбудовані функції та робоче середовище.

Також дуже зручним є те, що MATLAB підтримує та активно використовує багато різних функціональних модулів, які розширюють функціонал і дозволяють виконувати більш складні обчислення та аналіз даних. Робота з модулями MATLAB має кілька важливих переваг, які потрібно розглянути перед початком роботи з ними.

По-перше, модулі MATLAB дозволяють легко розширювати основний функціонал програми. Вони надають зручний спосіб додавати нові алгоритми, методи і функції, які не входять в стандартний набір. Це дозволяє користувачам виконувати більш складні обчислення і аналізувати дані за допомогою спеціалізованих інструментів, які вони можуть самостійно розробляти або використовувати з бібліотек, розроблених іншими користувачами [28]. Для візуалізації цю структурну схему MATLAB наведено на рисунку 3.1.

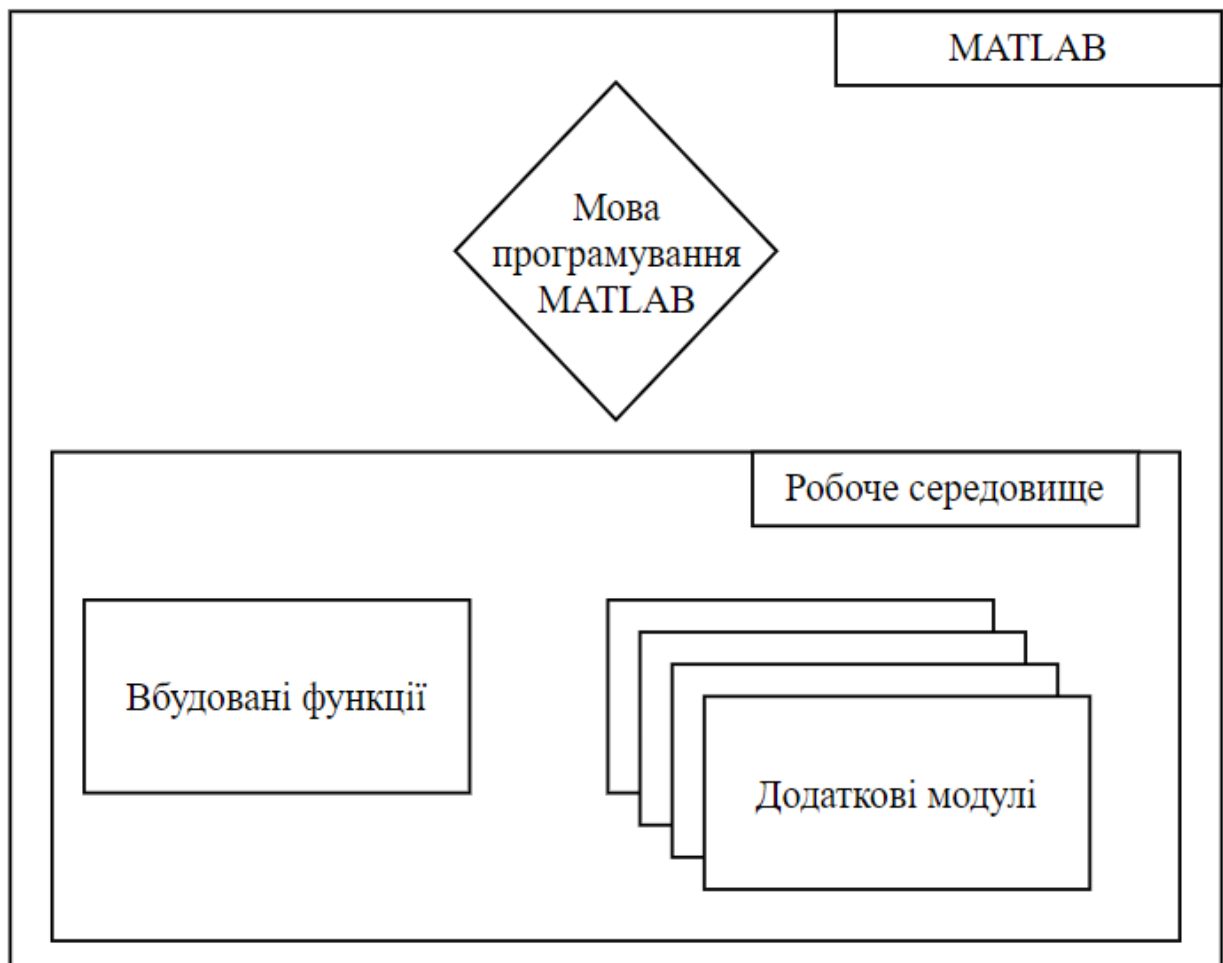


Рисунок 3.1 – Структура MATLAB

Модулі можуть бути організовані в ієрархічну структуру, що дозволяє організувати функції і ресурси в логічні групи. Це спрощує навігацію по коду і полегшує використання спеціалізованих функцій, забезпечуючи доступ до них відповідно до потреб користувача.

Функціональні модулі MATLAB забезпечують зручний механізм для поділу коду на відокремлені функціональні блоки. Це дозволяє розробникам створювати бібліотеки, які можуть використовуватися в різних проектах або надавати іншим користувачам для використання. Це сприяє повторному використанню коду, підвищує ефективність розробки та спрощує сумісність між різними проектами.

Модулі MATLAB працюють шляхом включення файлів з функціями та даними в основну робочу площину. Коли модуль імпортується, всі його функції стають доступними для використання в основному скрипті або в інших модулях. Це дозволяє організувати код в ієрархічну структуру, використовуючи простори імен для уникнення конфліктів імен і забезпечення чіткості роботи, що проводиться [29].

Узагалі, робота з модулями дозволяє розширити можливості самої MATLAB, зробити розробку більш ефективною і зручною, а також підвищити повторне використання коду і спрощення сумісності між проектами. Це допомагає користувачам виконувати складні завдання, відповідати на конкретні вимоги і розробляти розширені програми з високою гнучкістю і ефективністю.

3.2 Модуль Image Processing Toolbox

Одним з ключових компонентів для нашої роботи з MATLAB є Image Processing Toolbox. Цей модуль є потужним інструментарієм для роботи з зображеннями. Він надає різноманітні функції і алгоритми для обробки, аналізу та візуалізації зображень. Image Processing Toolbox може використовуватися для широкого спектра завдань, включаючи виявлення об'єктів, розпізнавання зразків, фільтрацію зображень, поліпшення якості, сегментацію та багато іншого [30].

Основні можливості Image Processing Toolbox :

- Завантаження та збереження зображень: Ви можете завантажувати зображення з файлів різних форматів, таких як JPEG, PNG, TIFF, BMP, і зберігати результати обробки у відповідному форматі.
- Відображення та візуалізація: Ви можете відображати зображення на екрані, масштабувати їх, робити збільшення та зменшення, відображати гістограми, робити 3D-подання та інше.
- Перетворення та фільтрація зображень: даний модуль містить широкий спектр фільтрів, які дозволяють вам виконувати операції розмиття, розкриття контрасту, підвищення гостроти, зменшення шуму та інші операції з покращення зображень.
- Сегментація зображень: Ви можете використовувати алгоритми сегментації для виділення окремих об'єктів або регіонів на зображенні.
- Виявлення об'єктів: Image Processing Toolbox надає функції для виявлення об'єктів на зображенні, включаючи виявлення країв, виявлення ліній, виявлення кіл та інших форм.
- Розпізнавання зразків: Ви можете використовувати алгоритми машинного навчання, такі як класифікатори, для розпізнавання зразків або об'єктів на зображенні.

Завдяки цим функціям ми можемо відносно легко реалізувати складні алгоритми обробки зображень та відображати результати у зручному для нас форматі.

Також однією з основних переваг розглядаємої програми є його синтаксис. Мова програмування MATLAB дуже проста для вивчення та розуміння, особливо для тих, хто вже має досвід у роботі з математичними або числовими алгоритмами. Вона дозволяє вам легко і швидко реалізовувати складні алгоритми обробки зображень з високою точністю.

Матрична обробка зображень є ще однією сильною стороною MATLAB. Оскільки зображення можна розглядати як матрицю пікселів,

програма надає нам потужні можливості для маніпулювання цими матрицями та виконання операцій над ними. Це дозволяє реалізувати швидкі та ефективні алгоритми обробки зображень.

Також важливо відзначити, що MATLAB має широку підтримку графічного відображення результатів обробки зображень. Ви можете легко відображати зображення, графіки, гістограми та інші візуальні представлення, що допомагають вам аналізувати та порівнювати результати вашої роботи. Крім того, програма має можливості для створення інтерактивних інтерфейсів користувача, що дозволяє вам створювати зручні інструменти для налаштування та візуалізації нашої обробки зображень.

3.3 Створення модуля дослідження методів підвищення якості зображень

У цьому розділі ми розглянемо проектування модуля, призначеного для дослідження різних методів підвищення якості зображень за допомогою MATLAB та Image Processing Toolbox.

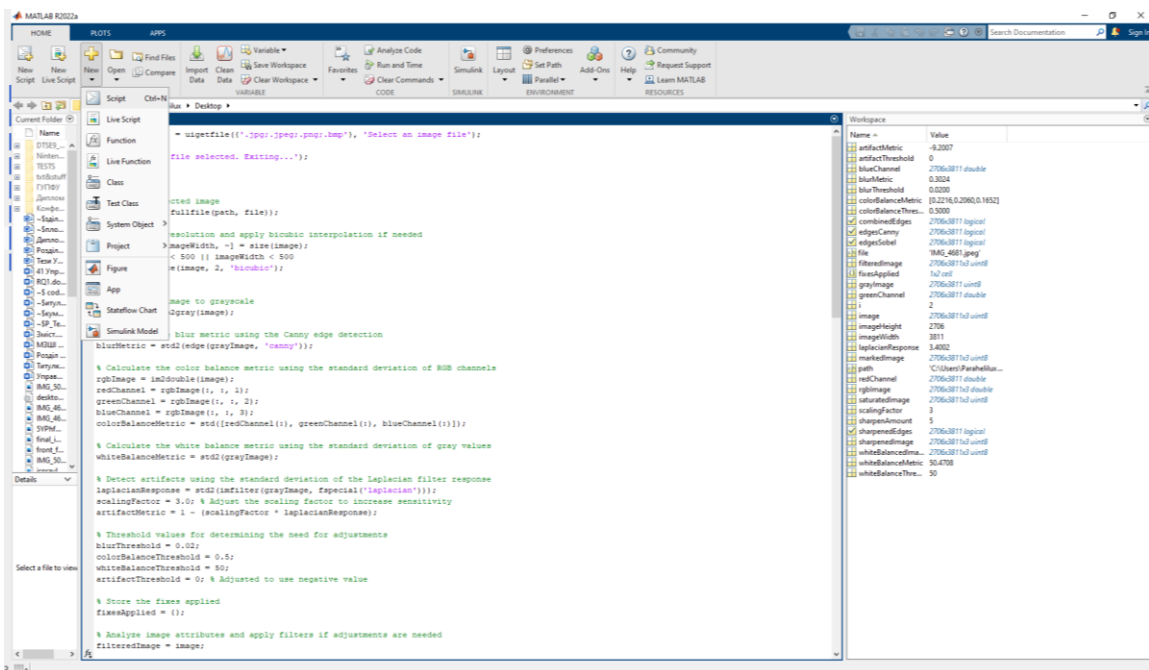


Рисунок 3.2 – Інтерфейс та робоча площина MATLAB

Розглянемо кожен етап проектування модуля детальніше:

У етапі реалізації запиту зображення ми реалізуємо функціональність, яка дозволяє користувачеві вибрати для подальшого аналізу та покращення. За допомогою відповідних функцій MATLAB та Image Processing Toolbox, це легко зробити.

Після отримання зображення необхідно аналізувати його параметри та виконувати відповідні операції для виділення його особливостей. Наприклад, можна виміряти роздільну здатність зображення, контрастність, наявність шуму, колірний профіль тощо. На основі результатів аналізу зображення ми можемо розробити функції покращення зображення, які сприятимуть підвищенню його якості. Наприклад, можна використовувати фільтри для зменшення шуму, методи підвищення контрастності, методи збільшення роздільної здатності, фільтри для видалення спотворень тощо. MATLAB та Image Processing Toolbox надають багато вбудованих функцій для цих цілей, таких як `imnoise` для додавання шуму, `imfilter` для застосування різних фільтрів, `imadjust` для налаштування контрастності та інші.

Після застосування функцій відповідних покращень до вихідного зображення, ми можемо проаналізувати результат та порівняти його з базовим зображенням. Можна використовувати різні метрики оцінки якості зображень, такі як піксельна помилка, структурний схожість, відношення сигнал-шум, для оцінки покращення. За допомогою MATLAB можна розробити алгоритми для порівняння та оцінки зображень, а також візуалізувати результати. На основі отриманих результатів можна зробити висновки про ефективність різних методів підвищення якості зображень та вибрати найкращий підхід для конкретних завдань.

Ці основні етапи проектування модуля дозволять нам реалізувати систему для дослідження та покращення якості зображень за допомогою MATLAB та Image Processing Toolbox.

Побудуємо план реалізації нашого модуля.



Рисунок 3.2 – Логічна схема програмного продукту

Етап 1: Реалізація запиту зображення

- Реалізація функціональності, що дозволяє користувачеві вибрати зображення для аналізу та покращення.
- Використання відповідних функцій MATLAB та Image Processing Toolbox для реалізації цієї функціональності.

Етап 2: Аналіз параметрів зображення

- Зчитування обраного зображення за допомогою функції `imread`.
- Отримання розмірів зображення за допомогою функції `imshow`.
- Аналіз інших параметрів зображення, таких як роздільна здатність, контрастність, наявність шуму та інші, за допомогою функцій обробки зображень.

Етап 3: Розробка функцій покращення зображення

- Використання результатів аналізу зображення для розробки функцій покращення зображення.
- Використання вбудованих функцій MATLAB та Image Processing Toolbox, таких як фільтри для зменшення шуму, методи підвищення контрастності, методи збільшення роздільної здатності та інші, для реалізації цих функцій покращення.

Етап 4: Аналіз та порівняння результатів

- Аналіз покращеного зображення та порівняння його з базовим зображенням.
- Використання різних метрик оцінки якості зображень, таких як піксельна помилка, структурна схожість, відношення сигнал-шум, для оцінки покращення.
- Розробка алгоритмів порівняння та оцінки зображень за допомогою функцій MATLAB.
- Візуалізація результатів аналізу та порівняння зображень.

Ці етапи дозволять реалізувати систему для дослідження та покращення фотографій. Кожен етап включає в себе використання

відповідних функцій та інструментів MATLAB та Image Processing Toolbox для досягнення поставлених цілей.

3.4 Програмна реалізація модуля дослідження методів підвищення якості зображень

Основна частина коду використовує наступні функції для реалізації різних операцій обробки зображення:

- `uigetfile`

Ця функція викликає діалогове вікно, що дозволяє користувачеві обрати зображення з певними розширеннями (наприклад, `.jpg`, `.jpeg`, `.png`, `.bmp`). Реалізація функції `uigetfile` зображено на рисунку 3.3.

```
>> [file, path] = uigetfile({' .jpg; .jpeg; .png; .bmp'}, 'Select an image file');
if file == 0
disp('No image file selected. Exiting...');
return;
end
```

Рисунок 3.2 – Функція `uigetfile`

- `imread`

Ця функція завантажує обране зображення з повним шляхом до файлу. Вона повертає матрицю зображення, яку можна подальше обробити.

- `imresize`

Ця функція застосовує бікубічну інтерполяцію для збільшення розміру зображення. Перший аргумент `image` є вихідним зображенням, а другий аргумент `2` вказує на те, що зображення потрібно збільшити в два рази по обох вимірах. Інтерполяція типу `'bicubic'` використовує гладку криву для визначення нових значень пікселів.

```
[imageHeight, imageWidth, ~] = size(image);
if imageHeight < 500 || imageWidth < 500
image = imresize(image, 2, 'bicubic');
end
```

Рисунок 3.4 – Функція `imresize`

- `rgb2gray`

Ця функція перетворює кольорове зображення в відтінки сірого шляхом відкидання інформації про кольори. Вона приймає вхідне кольорове зображення `image` і повертає зображення у відтінках сірого.

- `std2(edge(grayImage, 'canny'))`

Ця функція розраховує метрику розмитості зображення за допомогою оператора Кенні. Спочатку застосовується оператор Кенні до зображення в відтінках сірого за допомогою функції `edge(grayImage, 'canny')`, а потім розраховується стандартне відхилення (`std2`) отриманого результату. Ця метрика дозволяє оцінити рівень розмитості країв на зображенні.

- `std([redChannel(:), greenChannel(:), blueChannel(:)])`

Ця функція розраховує метрику балансу кольору за допомогою стандартного відхилення каналів RGB. Вона приймає вхідне кольорове зображення і розраховує стандартне відхилення (`std`) значень пікселів у кожному з трьох каналів кольору (червоному, зеленому, синьому). Ця метрика дозволяє оцінити рівень різниці між каналами кольору, що вказує на баланс кольору на зображенні. Реалізацію функції зображено на рисунку 3.5.

```
rgbImage = im2double(image);
redChannel = rgbImage(:, :, 1);
greenChannel = rgbImage(:, :, 2);
blueChannel = rgbImage(:, :, 3);
colorBalanceMetric = std([redChannel(:), greenChannel(:), blueChannel(:)]);
```

Рисунок 3.5 – Стандартний розподіл каналів RGB

- `std2(grayImage)`

Ця функція розраховує метрику балансу білого за допомогою стандартного відхилення значень сірого. Вона приймає вхідне зображення в відтінках сірого і розраховує стандартне відхилення (`std2`) значень пікселів. Ця метрика дозволяє оцінити рівень різниці між пікселями зображення, що вказує на баланс білого на зображенні.

- `std2(imfilter(grayImage, fspecial('laplacian')))`

Ця функція розраховує метрику артефактів за допомогою стандартного відхилення відповіді фільтра Лапласа. Спочатку застосовується фільтр Лапласа до зображення в відтінках сірого за допомогою функції `imfilter(grayImage, fspecial('laplacian'))`, а потім розраховується стандартне відхилення (`std2`) отриманого результату. Ця метрика дозволяє оцінити рівень артефактів на зображенні. Реалізацію цієї функції `imresize` зображено на рисунку 3.6.

```
laplacianResponse = std2(imfilter(grayImage, fspecial('laplacian')));
scalingFactor = 3.0; % Adjust the scaling factor to increase sensitivity
artifactMetric = 1 - (scalingFactor * laplacianResponse);
```

Рисунок 3.6 – Реалізація фільтра Лапласа

- `Medfilt`

Ці функції застосовують медіанний фільтр до кожного кольорового каналу (червоного, зеленого, синього) зображення для виправлення артефактів. Кожна функція приймає вхідне зображення `image` і повертає зображення з виправленими артефактами у відповідному каналі.

- `imsharpen`

Ця функція застосовує ефект нерізкості до зображення для розмиття або підвищення різкості. Вона приймає вхідне зображення `filteredImage` і параметри `'Amount'` і `'Radius'`, які контролюють ступінь розмиття або різкості зображення.

- `imadjust(filteredImage, stretchlim(filteredImage), [], 0.8)`

Ця функція виконує корекцію балансу кольору або балансу білого зображення. Вона приймає вхідне зображення `filteredImage`, межі розтягування кольору (`stretchlim(filteredImage)`) і параметр `0.8`, який контролює інтенсивність корекції.

- `imadjust(filteredImage, [], [], 1.5)`
- Ця функція підвищує насиченість кольору зображення. Вона приймає вхідне зображення `filteredImage` і параметр 1.5, який контролює рівень підвищення насиченості.
- `edge(rgb2gray(filteredImage), 'Canny')`, `edge(rgb2gray(filteredImage), 'Sobel')`

Ці функції виконують детекцію країв на зображенні за допомогою операторів Кенні та Собеля відповідно. Реалізація даної функції зображено на рисунку 3.7.

```
edgesCanny = edge(rgb2gray(filteredImage), 'Canny');
edgesSobel = edge(rgb2gray(filteredImage), 'Sobel');

% Combine the edge detection results
combinedEdges = edgesCanny | edgesSobel;
```

Рисунок 3.7 – Функції Canny та Sobel

- `filteredImage(combinedEdges) = 255`

Ця операція встановлює границі на зображенні для їх демонстрації. `combinedEdges` є маскою, що відповідає границям, і значення пікселів у відповідних позиціях `filteredImage` встановлюються на 255.

- `imsharpen`

Ця функція застосовує ефект нерізкості до виділених країв зображення за допомогою операції розмиття. `combinedEdges` є вихідним зображенням з виділеними краями, яке конвертується в тип `double`, а потім застосовується ефект нерізкості з параметрами 'Amount' і 'Radius'.

Це лише основні функції, які використовуються в коді. Інші функції та операції можуть бути додані та використані для керування вже існуючими змінними тощо. Програмний продукт збережено в форматі скрипта для MATLAB у форматі `.m` як `IAA.m`.

3.5 Контрольний приклад

В цьому розділі буде показано приклад роботи нашого програмного продукту.

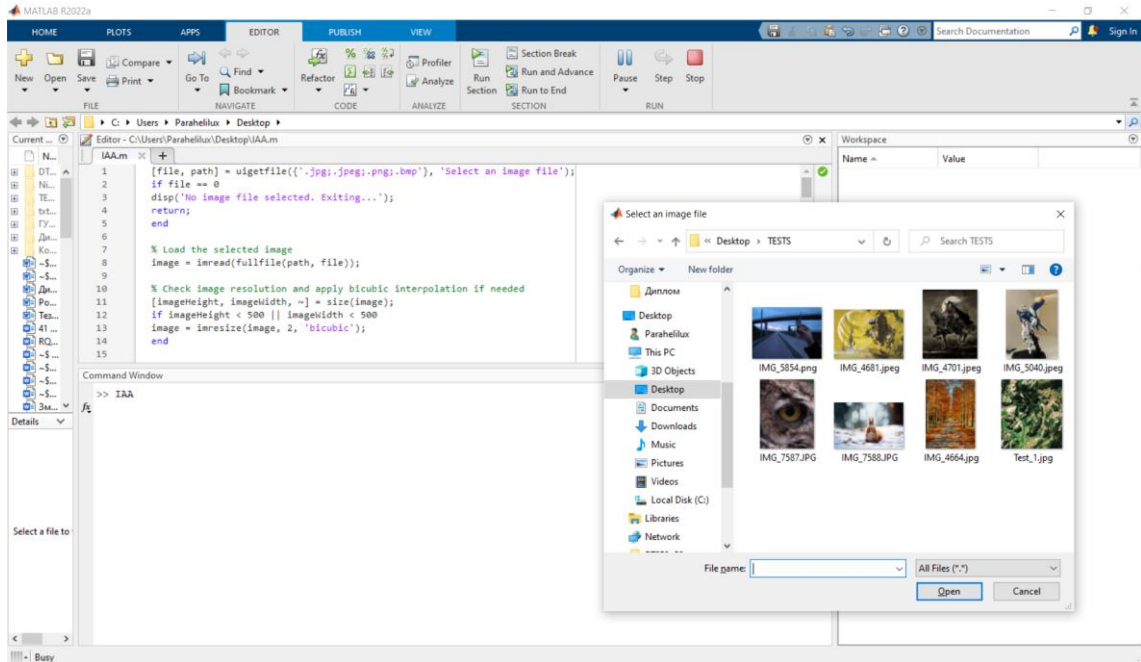


Рисунок 3.7 – Демонстрація 1

На рисунку 3.7 зображено виклик нашого скрипта IAA.m. Після виклику він надає можливість вручну обрати зображення для здійснення подальших дій.

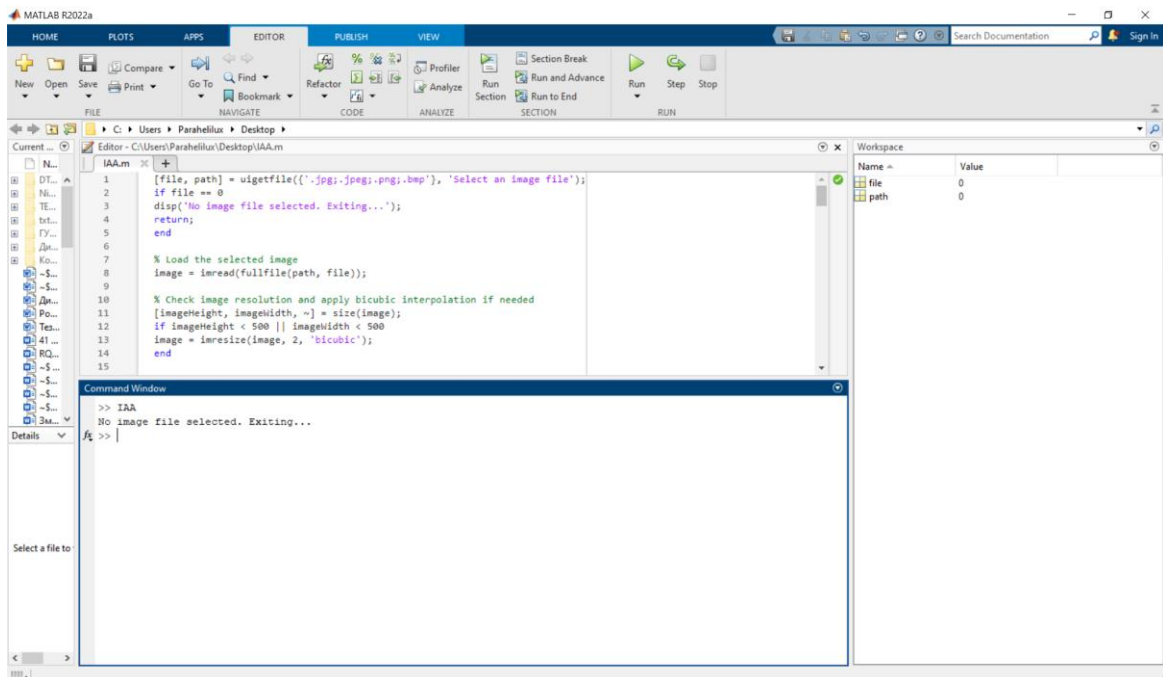


Рисунок 3.8 – Демонстрація 2

При відмові користувача обрати будь-яке зображення, наш скрипт говорить що зображення не обрано та припиняє роботу, цей сценарій зображено на рисунку 3.8.

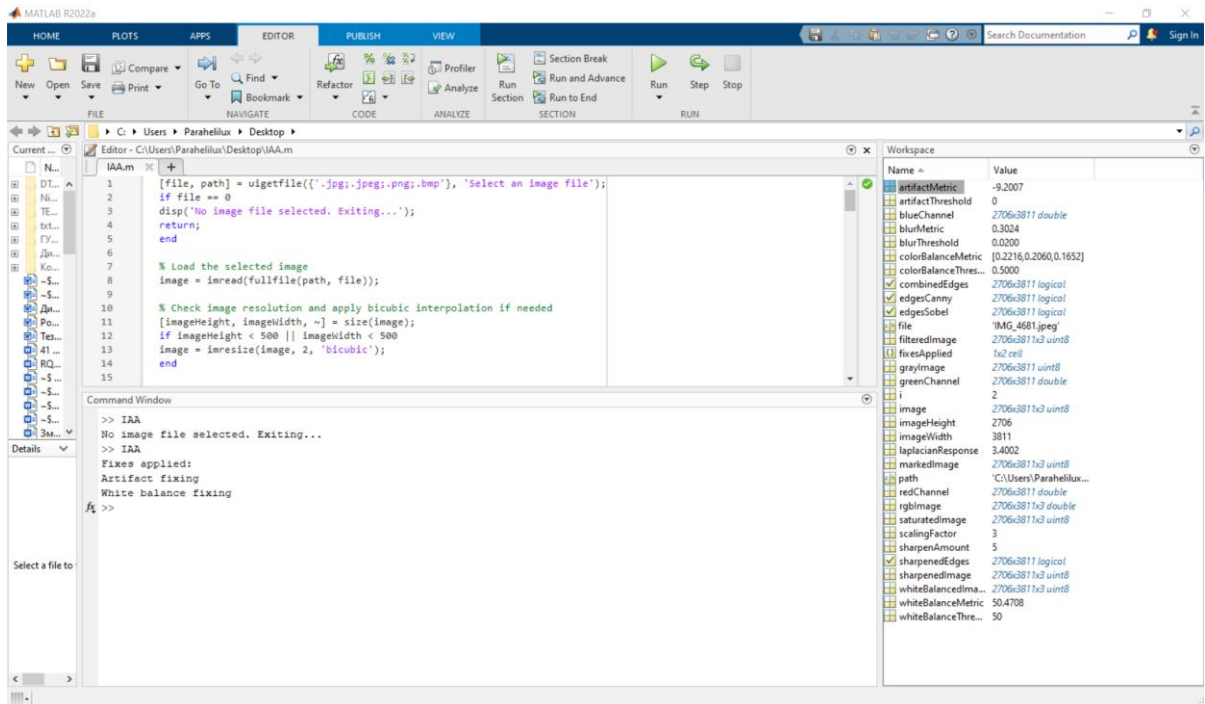


Рисунок 3.9 – Демонстрація 3

Якщо ж ми обрали зображення, то після обробки зображення, скрипт, як на рисунку 3.9, напише які саме параметри були скореговані та оновить наявні value, що доступні нам в нашій робочій площині. На тестовому зображенні були виправлені певні артефакти та скореговано баланс білого.

Поміж раніше згаданого, також буде створено нове вікно з чотирма зображеннями, його зображено на рисунку 3.10.

Перше зображення у цьому вікні буде оригінальним зображенням, яке ми обрали для обробки. Це зображення відобразить початковий стан, перед тим як будь-які фільтри чи алгоритми були застосовані.

Друге зображення буде представляти зображення, яке вже було оброблено за допомогою вибраних фільтрів чи обробки зображень. Це дасть вам можливість побачити, як саме впливають ці фільтри на оригінальне зображення та які зміни відбулися.

Третє зображення буде відображати виявлені границі вхідного зображення. Це може бути результатом використання алгоритмів, таких як детектори границь або оператори згладжування. Таке зображення покаже вам, де саме знаходяться границі об'єктів чи структур на зображенні.

Четверте зображення відобразить поєднання обробленого зображення, де були застосовані фільтри, та виявлені границі. Це надає нам візуальну демонстрацію того, як фільтри впливають на зображення та як вони взаємодіють з виявленими границями.

Це нове вікно з чотирма зображеннями буде допомагати вам більш наочно розуміти, які зміни відбуваються під час обробки зображень та використання алгоритмів щодо виявлення границь. Ми можемо порівняти різні етапи та побачити вплив фільтрів та алгоритмів на вхідне зображення.

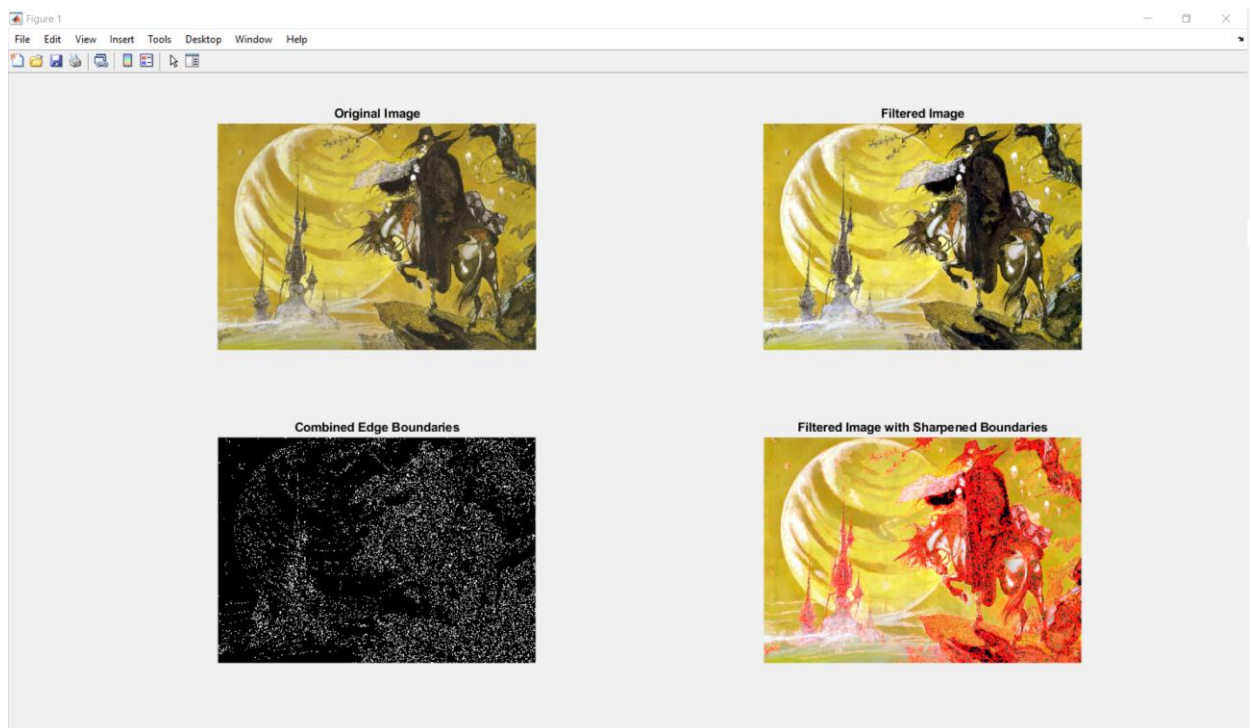


Рисунок 3.10 – Демонстрація 4

Для демонстрацій можливостей програмного продукту на рисунку 3.11 буде представлено результати роботи модуля з одним і тим же зображенням, але з різними заданими модифікаторами для обробки.



Оригінал



blurThreshold = 0.02;
colorBalanceThreshold = 0.5;
whiteBalanceThreshold = 50;
artifactThreshold = 0;



blurThreshold = 0.01;
colorBalanceThreshold = 0.7;
whiteBalanceThreshold = 60;
artifactThreshold = -10;



blurThreshold = 0.5;
colorBalanceThreshold = 2;
whiteBalanceThreshold = 100;
artifactThreshold = 80;



blurThreshold = 0.1;
colorBalanceThreshold = 0.1;
whiteBalanceThreshold = -200;
artifactThreshold = -200;

Рисунок 3.10 – Порівняння результатів

4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ПІДСИСТЕМИ

4.1 Вступ

Для розвитку компанії чи продукту потрібне чітке бачення розвитку потенціалу обраної ідеї. Проте, для того, щоб реалізувати свій потенціал та досягти успіху, нам необхідно виконати ряд завдань, запровадити стратегічні плани та дотримуватися конкретного плану дій [31].

Тому, було підготовлено цей бізнес-план, щоб продемонструвати наші плани та ідеї для розвитку компанії, а також показати, які кроки необхідно зробити, щоб досягти успіху.

4.2 Резюме

Наша компанія, як і будь-яка інша для успішного розвитку потребує певного набору послуг. Наша компанія бачить свій розвиток у своїй діяльності в одній із найдинамічніших сфер економіки кожної країни - сфері послуг.

Компанія «Арказац» планує надавати широкий спектр послуг, таких як:

- маркетинг власних програмних продуктів, товарів і послуг;
- консультативна діяльність і надання консультативних та інформаційних послуг;
- здійснення комплексних послуг, пов'язаних з організацією та оптимізацією існуючої it-інфраструктури;

Однак основною сферою нашої діяльності планується розробка , реалізація та сервіс програмного забезпечення, а також розробка і виробництво програмно-технічних засобів на базі нових інформаційних технологій. Дана галузь є важливою для українського та міжнародного ринків ще і тому, що більшість компаній відчувають нестачу в професійному програмному забезпеченні для вирішення локальних проблем. А такі фірми

як SAP, що має майже повну монополію на ринку програмного забезпечення для проведення інженерних робіт, виставляють непід'ємні для багатьох молодих компанії ціни на свої продукти. Це призводить до поширення піратських копій цих програм. Тому існує потреба в цьому виді програмного забезпечення. Компанія «Арказац» здатна надати необхідні програмні продукти для плідного функціонування різного роду компаній за доступними цінами.

Основним видом нашої діяльності є виробництво і реалізація програмного забезпечення. Ця галузь потребує високої кваліфікації персоналу, знання потреб ринку та якості виконання робіт. Проект буде реалізовуватися з використанням новітніх технологій. Для отримання необхідної інформації компанія «Арказац» володіє численними зв'язками з зарубіжними партнерами.

До початку активного розгортання програмного продукту компанія повинна була з'ясувати наступні питання:

- потреба на даний момент в програмному забезпеченні і розмір даної потреби;
- чи будуть клієнти застосовувати на виробництві цю продукцію, якщо вона буде цілком доступна і мати конкурентоспроможні ціни;
- відомі клієнту переваги використання доброякісної продукції для власної діяльності.

Результати проведеного нашою фірмою маркетинг дослідження підтвердили, що сучасна щорічна потреба ринку досить велика. Деякі з майбутніх споживачів даної продукції висловили пряме схвалення наших виробничих планів.

«Арказац» розробляє програми та програмно-технічні засоби. Ця продукція є необхідною як для фірм, так і для приватних користувачів. Програмне забезпечення включає різні програми в сфері фінансової і бухгалтерської звітності, планування виробництва, юридичного оформлення

та будь-які інші напрямки. Всі програми розроблені на основі потреб локального та міжнародного ринків з урахуванням глобального підходу. Інформаційна база оновлюється по мірі розвитку технологій і не відстає від технічного прогресу.

Серед особливостей нашої продукції хотілося б виділити наступне:

- високі стандарти якості;
- відповідність міжнародним стандартам;
- використання новітніх досягнень науки і техніки;
- доступність для молодих компаній.

Для розробки потрібно новітнє обладнання та технології. Для забезпечення матеріально-технічної бази компанія «Арказац» бере кредит у розмірі 250 тисяч гривень під 32% в monobank — продукті АТ «Універсал Банк», який виник у рамках співпраці з командою Fintech Vand. Це необхідно для покриття загальних операційних витрат, непередбачених витрат і різних одноразових початкових витрат. Даний банк було обрано після порівняння багатьох характеристик та через наявність прозорої взаємодії з підтримкою даного банку. Для реалізації проекту ми орендуємо офіс у районі метро «Оболонь». Там розташовується блок і офіс. У нас також укладені договори на поставку корпоративних ліцензій продуктів компанії «Microsoft» Штат наших співробітників містить дипломованих фахівців в даній області.

Проведене фінансове дослідження прогнозує мінімальні продажі програмних продуктів за наступний рік у приблизно 2,34 мільйони гривень.

а) Прогнозні об'єми продажу на найближчі роки:

Згідно з проведеним фінансовим дослідженням, мінімальні продажі програмних продуктів за наступний рік очікуються у приблизно 2,34 мільйони гривень. На наступний рік можна очікувати зростання продажів, якщо будуть успішно впроваджені нові технології та розширені ринки збуту.

б) Прибуток від продажів:

Прибуток від продажу може бути розрахований за формулою: $\text{прибуток} = \text{обсяг продажу} * \text{чистий прибуток від одиниці продукту}$. Якщо чистий прибуток від одиниці продукту складає 50%, то прибуток від продажу відпрацьованого проекту за наступний рік можна очікувати близько 1,17 мільйони гривень ($2,34 \text{ млн грн} * 50\%$).

в) Витрати на виробництво складатимуться з витрат на придбання необхідного обладнання, оплати праці, оренду офісу та інші операційні витрати. З урахуванням всіх цих витрат, припустимо, що загальні витрати на виробництво складатимуть 1,5 млн грн.

г) Валовий прибуток буде складати різницю між загальними витратами на виробництво та доходами від продажу продукту. Отже, валовий прибуток складатиме 0,84 млн грн. Рівень прибутковості вкладень в майбутню справу можна розрахувати, використовуючи формулу ROI (Return on Investment):

- $\text{ROI} = (\text{валовий прибуток} / \text{загальні вкладення в бізнес}) * 100\%$

Припустимо, що загальні вкладення в бізнес складають 1,75 млн грн. Тоді рівень прибутковості вкладень в майбутню справу буде:

- $\text{ROI} = (840\ 000 / 1\ 750\ 000) * 100\% = 48\%$

д) Термін повернення банківського кредиту можна розрахувати, використовуючи формулу:

- $\text{Термін повернення} = (\text{загальні вкладення в бізнес} - \text{кредитний депозит}) / \text{чистий прибуток}$

Припустимо, що компанія взяла кредит у розмірі 250 тисяч гривень під 32% річних. Тоді загальні вкладення в бізнес складають:

- $\text{Загальні вкладення} = 1,75 \text{ млн грн} + 250 \text{ тис. грн} = 2 \text{ млн грн}$
- Чистий прибуток складає 840 тис. грн, тоді:
- $\text{Термін повернення} = (2\ 000\ 000 - 250\ 000) / 840\ 000 = 2,14 \text{ років}$

Отже, припускаючи, що прогнознi об'єми продажу будуть досягнуті, проект буде здатний повернути кредит у термін менше, ніж 3 роки, що є досить вигідним для компаній-інвесторів.

4.3 Проектований продукт або вид послуг

Програмний продукт, який ми пропонуємо, є модулем для дослідження якості зображень. Основною потребою, яку ми задовольняємо, є забезпечення високої якості зображень для наших користувачів.

Особливості нашого продукту включають автоматизований процес дослідження якості зображень, що забезпечує високу точність та швидкість обробки даних. Також наш продукт має можливість розпізнавати різні види дефектів зображень та давати рекомендації щодо їх виправлення.

На даний момент у нас немає патентів або авторських свідоцтв на продукт проекту, але ми розглядаємо можливість їх отримання у майбутньому.

Ми маємо наочне зображення нашого продукту, яке може бути представлено у вигляді скріншотів з програмного інтерфейсу, відеоматеріали та версію-прототип продукту.

Попередня оцінка реалізації ціни продукту проекту та витрат на його виробництво залежить від ряду факторів, таких як рівень складності, масштаби проекту та кількість робочих годин, потрібних для його розробки.

а) Продукт проекту повинен задовольнити потребу в якісному дослідженні зображень, що має важливе значення для багатьох галузей, таких як медицина, наука, промисловість та інші.

б) Особливості продукту проекту полягають у високій точності та швидкості обробки зображень, наявності різних алгоритмів покращення якості зображень, можливості збереження та відновлення оброблених даних. Також продукт проекту має інтуїтивно зрозумілий інтерфейс, що робить його доступним для широкої аудиторії.

в) На даний момент патентів або авторських свідоцтв на продукт проекту немає.

г) Існують наочні зображення продукту проекту у вигляді скріншотів та ілюстрацій.

д) Попередня оцінка витрат на виробництво продукту проекту є помірною, оскільки вона включає в себе витрати на розробку, тестування та виробництво обладнання, а також витрати на зарплати розробників та рекламну кампанію.

е) Очікувана величина прибутку, який приносить продукт проекту:

Очікуваний прибуток від продукту проекту залежить від багатьох факторів, таких як ціна продажу, обсяг продажів, вартість виробництва та інші витрати на розробку та рекламу. Проте, при належному плануванні і управлінні, очікуваний прибуток може бути значним.

ж) Характеристика якісних показників продукту проекту, його переваги:

Модуль для дослідження якості зображень повинен мати наступні характеристики:

- Висока точність вимірювання якості зображень;
- Широкий діапазон підтримуваних форматів зображень;
- Простий та зрозумілий інтерфейс користувача;
- Підтримка збільшення та зменшення розміру зображень;
- Можливість збереження результатів дослідження якості зображень у різних форматах.

Переваги продукту :

- Висока точність вимірювання якості зображень, що дозволяє користувачам отримувати точну інформацію про якість своїх зображень;

- Широкий діапазон підтримуваних форматів зображень, що дозволяє користувачам аналізувати зображення, створені в різних програмах;
- Простий та зрозумілий інтерфейс користувача, що дозволяє легко використовувати програмний продукт;
- Можливість збереження результатів дослідження якості зображень у різних форматах, що дозволяє легко обмінюватись результатами дослідження між різними користувачами.

з) Організація сервісу продукту проекту полягає у забезпеченні користувачів продукту якісним технічним та інформаційним супроводом. Для модуля для дослідження якості зображень це може включати наступне:

- Технічна підтримка

Надання інформації користувачам щодо встановлення, налагодження та використання модуля.

- Оновлення програмного забезпечення

Забезпечення регулярних оновлень програмного забезпечення з метою поліпшення функціональності та виправлення помилок.

- Користувацькі матеріали

Надання посібників користувача, які допоможуть у розумінні функціональності модуля та його можливостей.

- Рішення проблем

Вирішення проблем, що виникають під час використання модуля, та надання технічної підтримки у вирішенні цих проблем.

- Забезпечення безпеки:

Забезпечення безпеки даних та конфіденційності користувачів шляхом захисту їх персональної інформації від несанкціонованого доступу.

- Консультації та тренінги

Проведення консультацій та тренінгів для користувачів щодо використання модуля, якщо потрібно.

- **Гарантійне обслуговування**

Надання гарантійного обслуговування користувачам, якщо модуль виявиться непридатним до використання з моменту його придбання.

4.4 Оцінка ринку збуту

Продукт, який ми досліджуємо - модуль для дослідження якості зображень. Для проведення оцінки ринку збуту необхідно визначити умови постачання, виробництва і збуту даного продукту.

Умови постачання включають у себе джерела ресурсів, необхідних для виробництва продукту, а також технологічний процес виготовлення. В даному випадку, для створення модуля для дослідження якості зображень необхідні знання з програмування та технічний експертизу в області обробки зображень. Також варто враховувати наявність ресурсів для виготовлення продукту, таких як програмне забезпечення та обладнання.

Умови виробництва передбачають можливості виробництва продукту в необхідній кількості та якості. В даному випадку, необхідно мати достатньо кваліфікованого персоналу, щоб забезпечити якісне виготовлення продукту, а також наявність необхідних ресурсів та обладнання для виробництва продукту.

Умови збуту включають у себе канали розповсюдження продукту та способи його просування на ринку. В даному випадку, модуль для дослідження якості зображень можна розповсюджувати через різні канали, такі як інтернет-магазини, компанії, що спеціалізуються на продажу програмного забезпечення, або за допомогою прямих продажів. Способи просування продукту можуть включати в себе рекламні кампанії, SEO-оптимізацію та участь у відповідних фахових заходах.

Для оцінки потенціалу можливих конкурентів необхідно зібрати інформацію про інші продукти, які займаються дослідженням якості зображень. Дослідження конкурентів можна провести шляхом аналізу їхніх

веб-сайтів, прес-релізів, рекламних матеріалів, відео-оглядів, а також за допомогою інших джерел інформації. Важливо визначити їхні переваги та недоліки в порівнянні з вашим продуктом і знайти можливість вирішити проблеми, що виникають у ваших конкурентів.

При оцінці якості продукту необхідно враховувати, які функції він надає, наскільки вони ефективні, якість зображень, що отримуються з його використанням, а також інші характеристики, які можуть впливати на його ефективність.

Попередні ціни на продукт проекту можна дослідити шляхом аналізу цін конкурентів на аналогічні продукти та порівняти їх з ціною на ваш продукт. Важливо також визначити оптимальну ціну на продукт, яка буде привабливою для споживачів і не забезпечить занадто високу маржинальну вигоду для підприємства.

Умови продажу продукту проекту також важливо враховувати. Наприклад, якщо ви плануєте продавати продукт онлайн, необхідно проаналізувати ринок електронної комерції і з'ясувати, які онлайн-магазини займаються продажем схожих продуктів, які умови пропонують своїм клієнтам, які способи оплати і доставки є використовуваними тощо.

- Щодо джерел отримання інформації

Для аналізу даних про потенційних клієнтів можна використовувати різні методи, такі як опитування, фокус-групи, онлайн-анкетування, аналіз соціальних мереж та інші. Під час аналізу слід звернути увагу на такі показники, як демографічні характеристики потенційних клієнтів, їх потреби та бажання, готовність до купівлі та орієнтовний бюджет.

- Власні дослідження

Це означає проведення власних досліджень з метою вивчення ринку збуту продукту. Це може включати проведення опитувань клієнтів, аналіз конкурентів, вивчення відгуків про подібні продукти тощо. Це може бути

досить часомістким та витратним процесом, але може дати цінну інформацію для розробки стратегії збуту.

- Місцеві торгові палати та асоціації підприємців

Місцеві торгові палати та асоціації підприємців можуть дати цінну інформацію про ринок збуту на місцевому рівні. Вони можуть надати інформацію про попит на продукт, конкурентність, ціни та умови збуту, а також можуть допомогти з організацією презентації продукту на виставках та інших заходах.

- Аналіз даних

Це процес аналізу даних, який дозволяє визначити потенційний попит на продукт проекту в найближчій та подальшій перспективі, а також визначити зразкову реалізаційну ціну продукту проекту в умовах конкуренції. Для цього можна використовувати інформацію, зібрану власними дослідженнями та інформацію, отриману від місцевих торгових палат та асоціацій підприємців.

- Реалізація заходів

Після аналізу даних можна розробити стратегію збуту, яка включатиме в себе план реалізації продукту на ринку, встановлення цін, визначення каналів збуту та план маркетингу.

Після зібрання і аналізу даних про ринок збуту програмного продукту, необхідно розробити стратегію збуту та реалізувати заходи, що дозволять використати ці дані на користь підприємства. Основні заходи, які можна реалізувати, включають:

- Встановлення оптимальної ціни продукту.

На основі зібраних даних про ціни конкурентів і зразкової реалізаційної ціни, можна визначити оптимальну ціну продукту, що дозволить забезпечити конкурентоспроможність і прибутковість продукту.

- Розробка маркетингової стратегії.

Необхідно розробити маркетингову стратегію збуту, яка враховує особливості цільової аудиторії, конкурентів та маркетингові інструменти, що дозволять досягти успіху на ринку.

- Вивчення можливих способів реклами.

Необхідно вивчити можливі способи реклами продукту, які дозволять залучити увагу цільової аудиторії, збільшити продажі та підвищити свідомість бренду.

- Створення партнерських відносин.

Можна створити партнерські відносини з іншими компаніями, які працюють у сфері інформаційних технологій, і взаємодіяти з ними щодо просування продукту на ринку.

- Використання соціальних мереж.

Можна використовувати соціальні мережі для реклами продукту та залучення уваги цільової аудиторії.

4.5 Конкуренція

На ринку практично немає компаній, які серйозно займаються такого роду діяльністю. Фірми, як правило, намагаються виграти на низьких витратах, купуючи програмне забезпечення низької якості. Тому їх програми можуть не тільки поліпшити, але й погіршити роботу клієнтів. Таким чином, компанія «Арказець» вигідно відрізняється від усіх фірм, що працюють на ринку. Можна виділити основні чинники нашої конкурентоспроможності:

- якість;
- рівень цін;
- дизайн;
- винятковість товару;
- підтримка програмного продукту та регулярні безпекові оновлення;
- репутація компанії.

4.6 Стратегія маркетингу

Ми усвідомлюємо, що продажу не прийдуть самі собою і їх треба завоювати. Великі споживачі побажають самі провести випробування рентабельності наших програмних продуктів. Ці випробування можуть зайняти від одного до шести місяців залежно від закупівельної політики відповідних клієнтів. Основна мета першого року виробничої діяльності полягає в тому, щоб зарекомендувати себе надійним постачальником. Зрештою, ми очікуємо, що станемо основним постачальником для більшості, якщо не для всіх, основних споживачів на цьому ринку.

Інтерес покупця до товару ми розраховуємо залучити за рахунок:

- прямих зв'язків з основними покупцями;
- реклами через комп'ютерні мережі;
- вищої якості нашої продукції, дружньої атмосфери звернення, відкритості;
- особисті контакти наших співробітників.

Центр ваги приходиться на збільшення оптових продажів. Основними споживачами нашої продукції є:

- великі корпорації у виробничій, банківській та інших сферах;
- середні і дрібні фірми;
- окремі приватні особи;
- будь-які клієнти, які бажають придбати нашу продукцію. За останні кілька років частка ринку, що належить програмно-технічних засобів, постійно збільшувалася. Проте в нашій країні виробництво ще не в змозі охопити весь споживчий ринок.

Попередні тестові прогони наших програмних продуктів були найважливішою частиною нашої підготовчої роботи і допомогли встановленню досить широких зв'язків з основними покупцями в регіоні.

Ми зможемо обслуговувати як великих, так і дрібних клієнтів. Складання програм являє собою велику проблему для багатьох користувачів,

і наша поставка дасть нам порівняльну перевагу в обслуговуванні дрібних, але безсумнівно прибуткових клієнтів.

Свою продукцію ми плануємо поширювати через власний сайт та через платформи партнерів.

У широкої реклами наш товар не потребує, але передбачається опублікувати обмежені в функціоналі версії в якості рекламного матеріалу. Також необхідно дати оголошення через мережі і факси деяким фірмам. Цього буде достатньо для залучення великої кількості клієнтів. Ринок збуту в майбутньому передбачається розширити і вивести за межі України та відкрити офіси в інших країнах.

З існуючих трьох основних цілей ціноутворення, з яких може вибрати фірма (засновані на збуті, на прибутках, на існуючому стані), для нашої компанії переважно цілі, засновані на існуючому положенні, так як нас влаштовує обсяг реалізації, збільшення відносної прибутку не виправдано еластичністю попиту на товар. Але ми зацікавлені в стабільності і збереження сприятливого клімату для своєї діяльності.

Стратегія ціноутворення орієнтована на мінімізацію впливу таких зовнішніх сил, як конкуренти й учасники каналів збуту.

З всіляких методів ми віддали перевагу встановлення ціни на основі відчутною цінності товару, тобто основний чинник ціноутворення - не витрати, а купівельна спроможність клієнта. Це підкреслюється наявністю різних типів ліцензій для різних кінцевих користувачів.

4.7 План виробництва

План розробки не передбачає виробничої кооперації з іншими компаніями чи підприємствами. Розробка продукту буде проводитись власними силами компанії.

Лімітація об'ємів виробництва або поставок неможлива, і буде обмежена тільки можливостями реалізації компаній-ресцелерів.

Для розробки модуля для дослідження якості зображень необхідне відповідне програмне забезпечення, а також обладнання для тестування якості зображень. Обладнання можна придбати у спеціалізованих магазинах, або використовувати вже наявне в компанії.

При розробці програмного продукту можуть виникати проблеми з оптимізацією продукту, забезпеченням сумісності з різними ОС та забезпеченням безпеки даних. Для запобігання цим проблемам, планується проводити ретельний аналіз продукту та використовувати найсучасніші технології та підходи.

Процеси контролю якості будуть проводитись на кожній стадії розробки продукту, від аналізу вимог до випуску готового продукту на ринок. Контроль якості буде проводитись за допомогою автоматизованих інструментів та вручну, за необхідності. Для контролю якості будуть використовуватись стандарти, такі як ISO 9001.

Оцінка можливих витрат виробництва буде проводитись на кожній стадії розробки продукту. Витрати будуть оцінюватись з урахуванням витрат на розробку, тестування, маркетинг та продаж продукту.

4.8 Організаційний план

Організацію нової справи буде організовувати засновник компанії, який візьме на себе роль генерального директора. Для успішної роботи потрібен наступний персонал:

- Керівник проекту з досвідом розробки програмного забезпечення і профілем, пов'язаним з обробкою зображень.
- Розробник програмного забезпечення з глибокими знаннями в області обробки зображень і досвідом розробки алгоритмів для роботи з ними.
- Тестувальник програмного забезпечення з досвідом використання різних технік тестування для забезпечення якості продукту.
- Менеджер з маркетингу, який буде відповідати за розробку маркетингової стратегії та впровадження маркетингових кампаній для продажу продукту.

- Менеджер з продажів, який буде відповідати за залучення нових клієнтів і підтримання стосунків з існуючими.

Усі фахівці повинні мати відповідну освіту і досвід роботи відповідно до своїх посад. Оплата праці буде залежати від кваліфікації та досвіду, але буде конкурентоспроможною на ринку.

Працівники будуть прийматися на постійну роботу. Проте, компанія може використовувати послуги зовнішніх експертів, якщо потрібно додаткової кваліфікованої робочої сили.

Для найму фахівців компанія буде розглядати кандидатів з різних джерел, таких як рекламні оголошення, рекомендації інших фахівців та рекрутингові агенства.

Організаційна структура підприємства, що буде займатись розробкою програмного продукту, буде складатись з наступних служб і посад:

- Керівництво проекту:

Project Manager (Керівник проекту)

Technical Lead (Технічний лідер)

Business Analyst (Бізнес-аналітик)

- Розробка:

Software Developer (Розробник програмного забезпечення)

UI/UX Designer (Дизайнер інтерфейсу та користувацьких досвідів)

- Тестування:

Quality Assurance (Фахівець з контролю якості)

Маркетинг та продаж:

Marketing Specialist (Маркетолог)

Sales Manager (Менеджер з продажів)

Кожна служба буде відповідальна за свою частину робіт з розробки та продажу програмного продукту, а керівництво проекту буде здійснювати координацію і контроль за роботою всіх служб.

Кваліфікаційні вимоги до фахівців:

- Керівник проекту

Досвід управління розробкою програмного забезпечення, знання методології Agile, висока комунікативність та лідерські якості.

- Технічний лідер

Досвід розробки програмного забезпечення, знання основних технологій та архітектур, високий рівень аналітичних навичок.

- Бізнес-аналітик

Досвід роботи з бізнес-процесами та вимогами до програмного забезпечення, знання методології Scrum, високий рівень аналітичних навичок.

- Розробник програмного забезпечення

Досвід розробки програмного забезпечення, знання основних технологій та архітектур, високий рівень програмування.

- Дизайнер інтерфейсу

Досвід розробки дизайну інтерфейсу користувача, знання теорії дизайну та користувацьких досвідів, вміння працювати з графічними програмами та розробляти відповідні макети, а також тестувати і вдосконалювати інтерфейси з урахуванням потреб і поведінки користувачів.

4.9 Юридичний план

Юридичний план для приватної компанії, яка займається розробкою програмного продукту, повинен включати наступні пункти:

- Вибір правової форми підприємства.

Оскільки розробка програмного продукту є високотехнологічним процесом, рекомендується створити приватну компанію з обмеженою відповідальністю (LLC). Це дозволить зменшити ризик для власників інтелектуальної власності та уникнути особистої відповідальності за борги компанії.

- Зареєструвати компанію відповідно до законодавства.

Потрібно зареєструвати компанію в органах державної реєстрації підприємств, подати заявки на отримання податкового номеру та рахунків у банку.

- Визначити права інтелектуальної власності.

Для захисту програмного продукту необхідно зареєструвати авторські права на програмне забезпечення, домовитися про нерозголошення конфіденційної інформації зі співробітниками та партнерами, а також укласти договір з користувачами, який визначає правила використання програмного продукту.

- Розробити договори з партнерами та клієнтами.

Компанія повинна мати договори з партнерами, які надають послуги з тестування та зберігання даних, а також договори з клієнтами, які регулюють використання програмного продукту та правила оплати послуг.

- Відстежувати зміни у законодавстві.

Компанія повинна відстежувати зміни у законодавстві щодо захисту інтелектуальної власності, оподаткування та правил діяльності компаній в ІТ-сфері.

Організаційний план більш відповідає на запитання щодо організації процесу розробки програмного продукту. Юридичний план, натомість, має на меті розглянути питання, пов'язані з юридичними аспектами діяльності компанії, яка розробляє програмний продукт.

Отже, для приватної компанії, яка розробляє програмний продукт, до юридичного плану можна включити наступні питання:

- Реєстрація компанії

Залежно від країни, в якій компанія зареєстрована, можуть вимагатися різні юридичні форми і процедури для реєстрації. Варто врахувати такі фактори, як оподаткування, відповідальність акціонерів та засновників, капіталізація, наявність певних ліцензій або дозволів.

- Захист прав інтелектуальної власності

Програмний продукт може бути захищений патентом, авторським правом, товарним знаком або іншими формами інтелектуальної власності. Компанія повинна забезпечити свою інтелектуальну власність та захистити її від порушень інших осіб.

- Укладення контрактів

У зв'язку з розробкою програмного продукту можуть виникати різні юридичні питання, такі як укладання угод з клієнтами, партнерами, підрядниками, виконавцями, консультантами тощо. Компанія повинна укладати контракти, які відповідають її потребам та захищають її інтереси.

- Регулювання діяльності

Компанія повинна дотримуватися законодавства, яке стосується її діяльності.

4.10 Оцінка ризику та страхування

Оцінка ризиків і страхування є важливою складовою будь-якого бізнесу. При підготовці оцінки ризиків для приватної компанії необхідно в першу чергу визначити можливі ризики, з якими компанія може зіткнутися, а потім розглянути заходи щодо їх зменшення або уникнення.

Перша частина: Оцінка ризиків

- Пожежі та інші природні катастрофи.

Компанії необхідно розробити плани евакуації та пожежної безпеки для своїх співробітників та майна, а також забезпечитися страховими полісами на випадок пожеж або інших природних катастроф.

- Конфлікти та соціальні ризики.

Компанії можуть зіткнутися зі страйками, протестами або іншими формами соціальних ризиків. Для зменшення цих ризиків, компанії можуть проводити ефективну комунікацію зі своїми працівниками та зацікавленими сторонами, а також створювати плани управління кризовими ситуаціями.

- Зміни в податковому регулюванні.

Компанії повинні бути готові до змін в законодавстві та податковому регулюванні та мати плани для відповідного відстеження та забезпечення відповідності.

- Коливання валютних курсів.

Компанії, що займаються експортом та імпортом, повинні мати стратегію управління валютними ризиками та страховими полісами для захисту від потенційних збитків.

- Конкуренція та технологічні ризики.

Компанія повинні бути готові до конкуренції та швидких змін у технологічній сфері, яка може впливати на їхній бізнес. Для зменшення цих ризиків, компанії можуть проводити аналіз конкурентів та ринкових тенденцій, брати участь у процесах інновацій та створювати нові продукти та послуги.

Друга частина: Заходи щодо зменшення ризиків

- Проведення регулярних аудитів та оцінок ризиків.

Компанії повинні відстежувати та оцінювати ризики регулярно, щоб вчасно виявляти потенційні загрози та приймати заходи для їх зменшення.

- Розроблення планів управління кризовими ситуаціями.

Компанії повинні мати плани, щоб ефективно управляти кризовими ситуаціями, такими як природні катастрофи, соціальні конфлікти та інші.

- Забезпечення страхування.

Компанії повинні мати страхові поліси для захисту від потенційних ризиків, таких як пожежі, природні катастрофи та інші.

- Розроблення стратегії управління валютними ризиками.

Компанії, що займаються експортом та імпортом, повинні мати стратегію управління валютними ризиками, такими як захист від змін валютних курсів та інші.

- Підтримка комунікації зі зацікавленими сторонами.

Компанії повинні підтримувати ефективну комунікацію зі своїми працівниками та зацікавленими сторонами, щоб вчасно виявляти потенційні ризики та приймати заходи для їх зменшення.

Отже, для забезпечення успішності бізнесу важливо не тільки виявити потенційні ризики, а й приймати дієві заходи для зменшення цих ризиків. Проведення регулярних аудитів та оцінок ризиків, розроблення планів управління кризовими ситуаціями, забезпечення страхування та розроблення стратегії управління валютними ризиками - це всі важливі кроки, які допоможуть компаніям уникнути потенційних ризиків та збільшити свою успішність. Крім того, ефективна комунікація зі своїми працівниками та зацікавленими сторонами також може допомогти компанії вчасно виявляти та управляти ризиками, що може збільшити конкурентоспроможність на ринку.

4.11 Фінансовий план

Для складання фінансового плану для спеціалізованого програмного продукту (модуля для дослідження якості зображень) потрібно врахувати такі пункти:

- Розробка продукту

Це може включати в себе витрати на програмістів, дизайнерів, тестувальників і т.д. Вартість розробки може залежати від обсягу і складності функцій, які необхідно вбудувати в продукт.

- Маркетинг та реклама

Це може включати в себе витрати на інтернет-рекламу, виставки, конференції та інші методи реклами продукту. Ці витрати можуть залежати від того, як добре ви хочете просунути свій продукт.

- Операційні витрати

Це може включати в себе витрати на оренду офісу, зарплати працівників, обслуговування програмного забезпечення та інше.

- Продажі

Це може включати в себе витрати на продажі продукту, такі як комісії, податки та інші.

Прогноз обсягів реалізації можна зробити наступним чином:

Перший рік:

Місячний обсяг продажу: 5000 у.о.

Загальний обсяг продажу за рік: 60 000 у.о.

Другий рік:

Квартальний обсяг продажу:

1 квартал: 15 000 у.о.

2 квартал: 16 000 у.о.

3 квартал: 16 500 у.о.

4 квартал: 17 000 у.о.

Загальний обсяг продажу за рік: 64 500 у.о.

Третій рік:

Загальний обсяг продажу за 12 місяців: 70 000 у.о.

Прогнозований баланс грошових витрат і надходжень для спеціалізованого програмного продукту (модуль для дослідження якості зображень) з місячним розбиттям на перший рік наведено в таблиці нижче.

Місяць	Продажі	Витрати на розробку	Витрати на маркетинг	Витрати на утримання
Січень	150 000	75 000	15 000	10 000
Лютий	180 000	75 000	15 000	10 000
Березень	200 000	75 000	15 000	10 000
Квітень	220 000	75 000	15 000	10 000

Травень	240 000	75 000	15 000	10 000
Червень	260 000	75 000	15 000	10 000
Липень	280 000	75 000	15 000	10 000
Серпень	300 000	75 000	15 000	10 000
Вересень	320 000	75 000	15 000	10 000
Жовтень	340 000	75 000	15 000	10 000
Листопад	360 000	75 000	15 000	10 000
Грудень	380 000	75 000	15 000	10 000

Прогнозовані витрати на розробку програмного продукту складають 75 000 гривень на місяць, витрати на маркетинг - 15 000 гривень на місяць, а витрати на утримання - 10 000 гривень на місяць. Прогнозовані продажі на перший рік складають 2,34 мільйони гривень.

Прогноз на другий рік буде поданий в таблиці з розбиттям на квартали:

Квартал	Продажі	Витрати на розробку	Витрати на маркетинг	Витрати на утримання
1	700,000	70,000	20,000	12,000
2	900,000	50,000	25,000	15,000
3	1,200,000	50,000	30,000	18,000
4	1,500,000	40,000	35,000	20,000

4.12 Стратегія фінансування

Для реалізації проекту компанія «Арказац» використовує різні джерела фінансування. Зокрема, компанія бере кредит у розмірі 250 тисяч гривень під 32% від monobank — продукті АТ «Універсал Банк». Цей кредит призначений для покриття загальних операційних витрат, непередбачених витрат і різних одноразових початкових витрат.

Крім того, компанія вирішила орендувати офіс у районі метро «Оболонь», де будуть розташовані блок і офіс. Орендна плата за цей офіс також буде покрита з загальних операційних витрат компанії.

Також компанія уклала договори на поставку корпоративних ліцензій продуктів компанії «Microsoft», що дозволить скоротити витрати на програмне забезпечення.

Згідно з проведеним фінансовим дослідженням, прогноуються мінімальні продажі програмних продуктів за наступний рік у приблизно 2,34 мільйони гривень. Для забезпечення подальшого фінансування проекту, компанія може розглядати можливості отримання додаткових кредитів, пошук нових інвесторів, а також збільшення продажів за рахунок рекламних кампаній та розвитку нових продуктів. Компанія може також здійснювати оптимізацію витрат, щоб забезпечити ефективне використання коштів та збільшення прибутковості проекту.

Якщо розглядати інвестування з боку потенційних інвесторів, то вони, напевно, будуть зацікавлені в можливості отримати частку від компанії та забезпечити її додатковими фінансовими ресурсами для подальшого розвитку. Потенційні інвестори будуть аналізувати не лише поточні фінансові показники компанії, але й її потенціал, ринкову потребу в її продукції, конкурентоспроможність на ринку та інші фактори.

Зважаючи на те, що компанія "Арказец" є розробником програмного забезпечення, можна очікувати, що потенційні інвестори будуть зацікавлені в інноваційних рішеннях, які вона може запропонувати на ринку. Також можна очікувати, що компанія буде звертати увагу на тенденції в галузі та буде здійснювати дослідження, щоб бути готовою до викликів та змін на ринку.

4.13 Висновки

Детальний бізнес-план для створення спеціалізованого програмного продукту - модуля для дослідження якості зображень - було розроблено з

урахуванням різноманітних факторів, що впливають на його успішну реалізацію. План передбачає реалізацію проекту в чотири етапи, починаючи з дослідження ринку та аналізу конкурентів до створення та просування готового продукту на ринку.

План фінансування передбачає залучення коштів на ранніх етапах створення продукту від інвесторів та партнерів, а також отримання кредиту для забезпечення матеріально-технічної бази та покриття загальних витрат. Очікувана рентабельність проекту досить висока, з орієнтовними продажами на рівні 2,34 мільйонів гривень за перший рік.

Оцінюючи перспективи проекту, можна зробити висновок, що його успішна реалізація залежить від ряду факторів, таких як якість розробки програмного продукту, його відповідність ринковим потребам та конкурентному середовищу, ефективність маркетингових стратегій та здатність до швидкого відгуку на зміни на ринку. Однак з урахуванням наявності кваліфікованих спеціалістів у компанії та підтримки з боку інвесторів та партнерів, можна очікувати успішного запуску продукту на ринку та досягнення бажаних фінансових показників.

ВИСНОВОК

Було виконано аналіз для розробки підсистеми дослідження методів підвищення якості цифрових зображень, було виявлено проблеми, які мають бути вирішені під час проектування та реалізації рішення. На основі цього аналізу були визначені основні цілі проекту.

Побудовано дерево основних цілей, що відображає ієрархію цілей. Це дерево допомогло уявити загальну структуру проекту та визначити пріоритети.

Наступним кроком було здійснено пошук та аналіз наявних методів у цій галузі. Цей аналіз допоміг зрозуміти, які підходи та методи вже існують, які переваги та обмеження вони мають. Це надало базу для подальшого розроблення нових та вдосконалення наявних методів.

Поставлені основні задачі, пов'язані з розробкою підсистеми. Для цього було побудовано дерево функцій, яке описує виконання цих задач за та їх залежності одна від одної. Це дерево допомагає систематизувати функціональні вимоги та зрозуміти, які компоненти системи відповідають за кожну задачу.

Було проведено проектування програмного забезпечення. Описана технічна платформа та інші інструменти для реалізації проекту.

На практиці була реалізована серверна частина системи з необхідним функціоналом для дослідження якості цифрових зображень.

Був проведений тест усіх можливостей програмного забезпечення, щоб переконатись у його працездатності та відповідності поставленим вимогам. Виявлені помилки та недоліки були виправлені під час розробки, а продукт готовий для подальшого використання.

Усі ці кроки дозволили успішно розробити підсистему дослідження методів підвищення якості цифрових зображень, яка може бути використана для проведення досліджень та покращення якості зображень у різних сферах застосування.

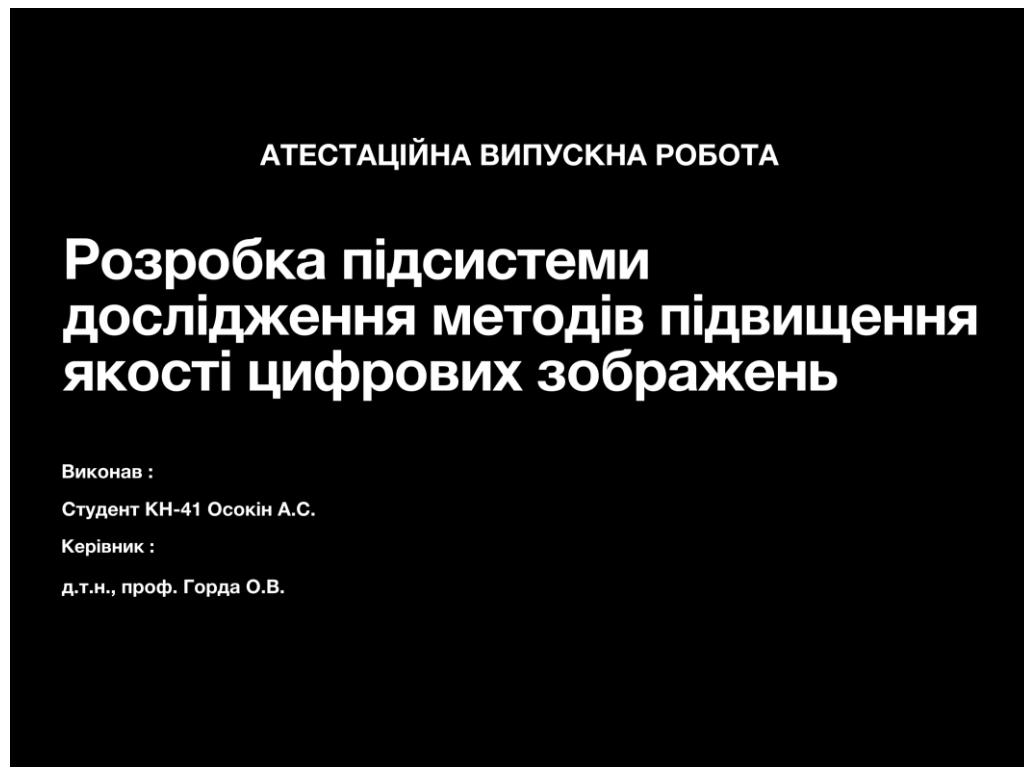
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Цюцюра С. В. «Методичні вказівки до виконання дипломної роботи» – // КНУБА – 2018 р.
2. Edward M. Mikhail, James S. Bethel, J. Chris McGlone "Introduction to Photogrammetry: Principles, Tools, and Applications" - видавництво John Wiley & Sons, Inc., 2015.
3. Giorgianni, Edward J., Madden, Thomas E. "Digital Color Management: Encoding Solutions" - видавництво Addison-Wesley Professional, 2007.
4. Pratt, W. K. "Digital Image Processing" - видавництво John Wiley & Sons, Inc., 2007.
5. Jain A.K. "Fundamentals of Digital Image Processing" - видавництво Prentice Hall, 1989.
6. Mark S. Nixon та Alberto S. Aguado "Feature Extraction and Image Processing for Computer Vision" - видавництво Academic Press, 2020.
7. Russ J.C. "The Image Processing Handbook (7th ed.)" - видавництво CRC Press, 2011.
8. Saleem Mariam та Altaf Qasim "Development & Comparison of Blind Cloning in Digital Image Forensics" - видавництво Lambert Academic Publishing, 2015.
9. Steven J. Janke "Mathematical Structures for Computer Graphics" - видавництво John Wiley & Sons, Inc., 2014.
10. Digital camera [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Digital_camera
11. Absolute Category Rating [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Absolute_Category_Rating

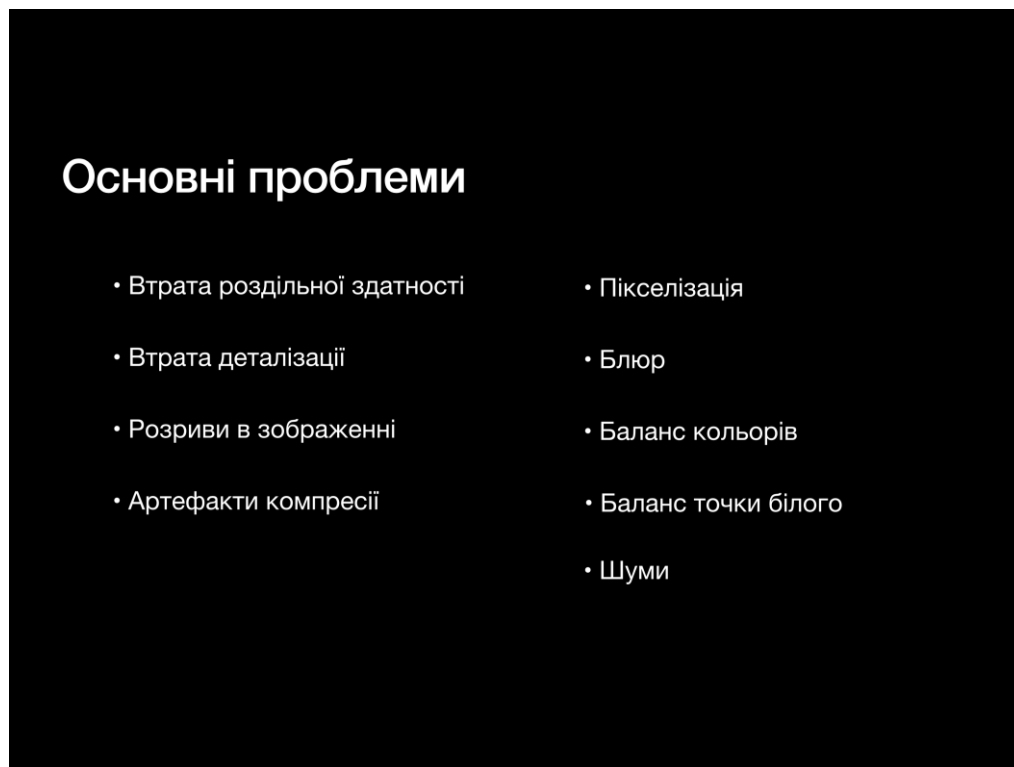
12. Alain Horé, Djemel Ziou "Image Quality Metrics: PSNR vs. SSIM" – видавництво IEEE, 2010.
13. Giuseppe Bonaccorso "Mastering Machine Learning Algorithms" – видавництво Packt Publishing, 2018.
14. Adobe Photoshop Lightroom [Електронний ресурс] // Матеріал з сайту компанії-виробника. – 2023. – Режим доступу до ресурсу: <https://www.adobe.com/ua/products/photoshop-lightroom.html>
15. Skylum Luminar [Електронний ресурс] // Матеріал з сайту компанії-виробника. – 2023. – Режим доступу до ресурсу: <https://skylum.com/luminar>
16. Capture One [Електронний ресурс] // Матеріал з сайту компанії-виробника. – 2023. – Режим доступу до ресурсу: <https://www.captureone.com/en>
17. DxO PhotoLab [Електронний ресурс] // Матеріал з сайту компанії-виробника. – 2023. – Режим доступу до ресурсу: <https://www.dxo.com/dxo-photolab/>
18. Горда О. В. – "Методичні вказівки до виконання курсової роботи з дисципліни «Моделювання систем»"// КНУБА – 2018р.
19. Color model [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Color_model
20. Bicubic interpolation [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Bicubic_interpolation
21. Sobel operator [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Sobel_operator

22. Gaussian filter [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Gaussian_filter
23. Canny edge detector [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Canny_edge_detector
24. Standard deviation [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Standard_deviation
25. Discrete Laplace operator [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Discrete_Laplace_operator
26. Median filter [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Median_filter
27. Unsharp masking [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Unsharp_masking
28. Gonzalez R.C., Woods R.E., & Eddins S.L. "Digital Image Processing Using MATLAB" – видавництво Pearson Education, 2018.
29. Burger W., & Burge M. "Digital Image Processing: An Algorithmic Approach with MATLAB." – видавництво Springer, 2009.
30. Image Processing Toolbox [Електронний ресурс] // Документація з сайту компанії-розробника. – 2023. – Режим доступу до ресурсу: <https://www.mathworks.com/help/images/>
31. Цюцюра С.В, Криворучко О.В., Цюцюра М.І. «Системи управління інвестиційними проектами» – // КНУБА – 2013 р.

Презентація Атестаційної випускної роботи



Слайд 1 – Розробка підсистеми дослідження методів підвищення якості цифрових зображень



Слайд 2 – Основні проблеми

Існуючі рішення



Слайд 3 – Існуючі рішення

Дерево функцій



Слайд 4 – Дерево функцій

Математичні методи обробки зображень

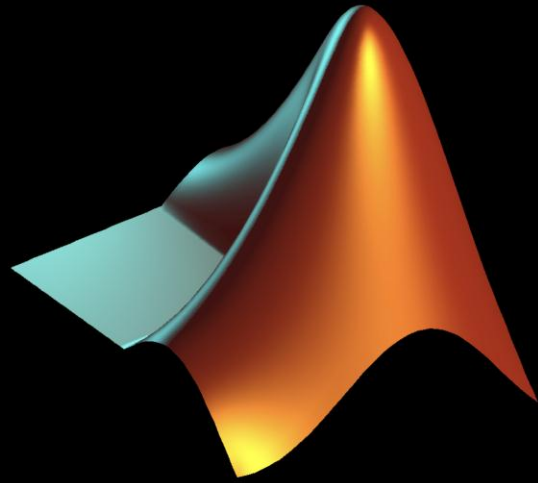
- Бікубічна інтерполяція
- Фільтр Лапласа
- Фільтр Собеля
- Медіанний фільтр
- Фільтр Гаусса
- Нерізкість маскуванню
- Алгоритм Кенні
- Передбачення сірого світу
- Стандартне відхилення
- Ліміт розтягування

Слайд 5 – Математичні методи обробки зображень



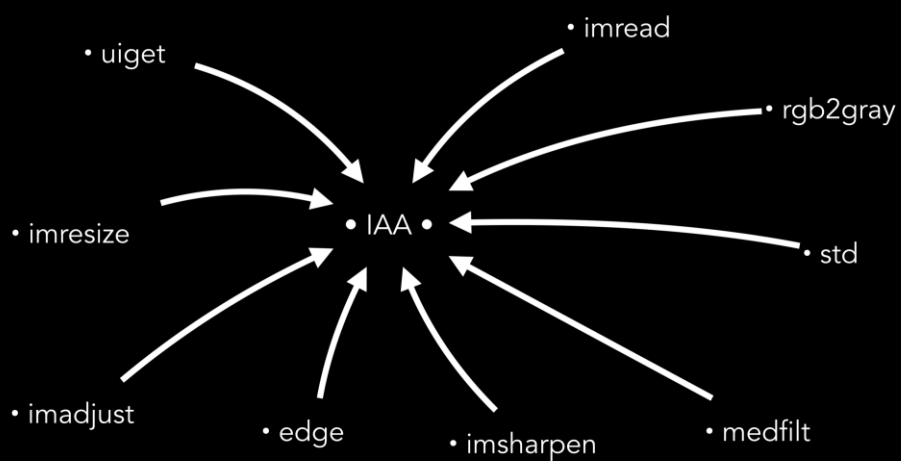
Слайд 6 – Побудова моделі

Технічна платформа



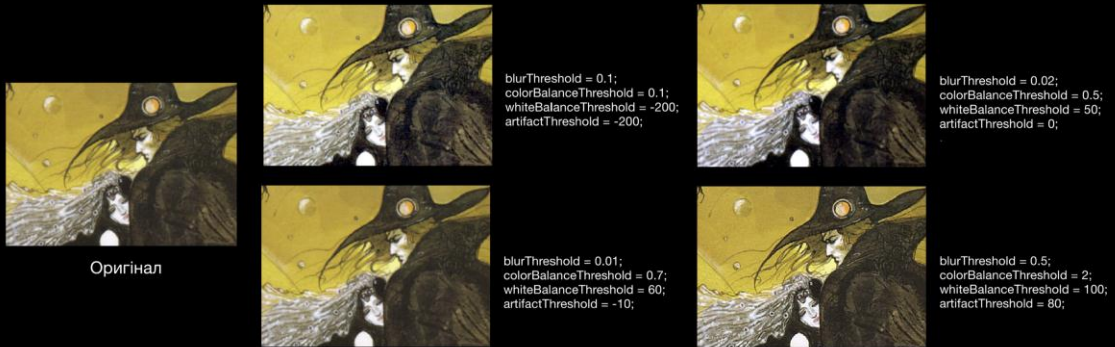
Слайд 7 – Технічна платформа

Програмна реалізація



Слайд 8 – Програмна реалізація

Приклади роботи



Слайд 9 – Приклади роботи

Висновки

Слайд 10 – Висновки

Лістинг коду

```

[file, path] = uigetfile({' .jpg;.jpeg;.png;.bmp'}, 'Select an image file');
if file == 0
disp('No image file selected. Exiting...');
return;
end
image = imread(fullfile(path, file));
[imageHeight, imageWidth, ~] = size(image);
if imageHeight < 500 || imageWidth < 500
image = imresize(image, 2, 'bicubic');
end
grayImage = rgb2gray(image);
blurMetric = std2(edge(grayImage, 'canny'));
rgbImage = im2double(image);
redChannel = rgbImage(:, :, 1);
greenChannel = rgbImage(:, :, 2);
blueChannel = rgbImage(:, :, 3);
colorBalanceMetric = std([redChannel(:), greenChannel(:), blueChannel(:)]);
whiteBalanceMetric = std2(grayImage);
laplacianResponse = std2(imfilter(grayImage, fspecial('laplacian')));
scalingFactor = 3.0; % Adjust the scaling factor to increase sensitivity
artifactMetric = 1 - (scalingFactor * laplacianResponse);
blurThreshold = 0.02;
colorBalanceThreshold = 0.5;
whiteBalanceThreshold = 50;
artifactThreshold = 0;
fixesApplied = {};
filteredImage = image;
if artifactMetric < artifactThreshold
filteredImage = image;
filteredImage(:, :, 1) = medfilt2(image(:, :, 1));
filteredImage(:, :, 2) = medfilt2(image(:, :, 2));
filteredImage(:, :, 3) = medfilt2(image(:, :, 3));
fixesApplied{end + 1} = 'Artifact fixing';
sharpenAmount = 5.0; % Adjust the sharpening amount as desired
else
sharpenAmount = 2.5; % Default sharpening amount
end

if blurMetric < blurThreshold
unblurredImage = imsharpen(filteredImage, 'Amount', sharpenAmount, 'Radius',
1);
filteredImage = unblurredImage;
fixesApplied{end + 1} = 'Unblur fixing';
end

```

```

if whiteBalanceMetric > whiteBalanceThreshold
whiteBalancedImage = imadjust(filteredImage, stretchlim(filteredImage), [], 0.8);
filteredImage = whiteBalancedImage;
fixesApplied{end + 1} = 'White balance fixing';
end

if colorBalanceMetric > colorBalanceThreshold
colorBalancedImage = imadjust(filteredImage, stretchlim(filteredImage), [], 0.8);
filteredImage = colorBalancedImage;
fixesApplied{end + 1} = 'Color balancing';
end

saturatedImage = imadjust(filteredImage, [], [], 1.5);
edgesCanny = edge(rgb2gray(filteredImage), 'Canny');
edgesSobel = edge(rgb2gray(filteredImage), 'Sobel');
combinedEdges = edgesCanny | edgesSobel;
markedImage = filteredImage;
markedImage(combinedEdges) = 255; % Set boundary pixels to white
sharpenedEdges = imsharpen(double(combinedEdges), 'Amount', sharpenAmount,
'Radius', 1);
sharpenedEdges = logical(sharpenedEdges);
sharpenedImage = filteredImage;
sharpenedImage(sharpenedEdges) = 255;
figure;
subplot(2, 2, 1);
imshow(image);
title('Original Image');
subplot(2, 2, 2);
imshow(filteredImage);
title('Filtered Image');
subplot(2, 2, 3);
imshow(combinedEdges);
title('Combined Edge Boundaries');
subplot(2, 2, 4);
imshow(sharpenedImage);
title('Filtered Image with Sharpened Boundaries');
if ~isempty(fixesApplied)
disp('Fixes applied:');
for i = 1:numel(fixesApplied)
disp(fixesApplied{i});
end
else
disp('No fixes were applied.');
```