

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації і інформаційних технологій

Кафедра інформаційних технологій

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТР**

на тему:

Розробка GAN моделі для підвищення якості зображень

Осокін Арсен Станіславович

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

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації і інформаційних технологій

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ

“ ___ ” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТР**

Розробка GAN моделі для підвищення якості зображень
(назва)

Виконав: Осокін Арсен Станіславович
(прізвище, ім'я та по батькові повністю)

122 «Комп'ютерні науки»
(спеціальність)

«Комп'ютерні науки»
(освітня програма)

Групи: КНМ-23

Керівник: Горда О.В.
(прізвище та ініціали)

Кандидат технічних наук, доцент
(вчене звання, науковий ступінь)

Ідентичність підтверджую

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «магістр» за ОП «Комп'ютерні науки»

Спеціальність: 122 «Комп'ютерні науки»

Спеціалізація: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

Гончаренко Т.А.

“ ___ ” _____ 2024 року

**ЗАВДАННЯ
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

Осокін Арсен Станіславович

1. Тема роботи: Розробка GAN моделі для підвищення якості зображень затверджена наказом ректора КНУБА № 2213/2 від «8» жовтня 2024 р.
2. Керівник роботи: к.т.н., доцент Горда Олена Володимирівна, кафедри інформаційних технологій.
3. Строк подання студентом роботи до захисту: « » листопада 2024 р.
4. Зміст пояснювальної записки за розділами:
 - P.1. Аналіз предметної області
 - P.2. Математичні методи обробки зображень та програмне проектування
 - P.3. Проектування програмної реалізації
 - P.4. Програмна реалізація та експериментальне дослідження системи
 - P.5. Техніко-економічне обґрунтування розробки GAN-моделі
5. Інформаційні слайди:
 - C.1. Розробка GAN моделі для підвищення якості зображень
 - C.2. Актуальність та мета
 - C.3. Об'єкт, предмет та методи дослідження
 - C.4. Дерево цілей
 - C.5. Аналіз існуючих методів
 - C.6. Що таке GAN ?
 - C.7. Онтологічний підхід до поняття зображення
 - C.8. Визначення критеріїв оцінки
 - C.9. Обрання набору даних для навчання

- С.10. Проектування моделі
 С.11. Детальніше про генератор та дискримінатор
 С.12. Оптимізація процесу навчання
 С.13. Технічні платформи
 С.14. Навчання моделі
 С.15. Технічні вимоги та обмеження
 С.16. Приклад роботи
 С.17. Висновки
 С.18. Дякую за увагу

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області	Вересень 2024 р.
Р. 2. Математичні методи обробки зображень та програмне проектування	Вересень 2024 р.
Р. 3. Проектування програмної реалізації	Жовтень 2024 р.
Р. 4. Програмна реалізація та експериментальне дослідження системи	Жовтень 2024 р.
Р. 5. Техніко-економічне обґрунтування розробки GAN-моделі	Листопад 2024 р.
Оформлення роботи та направлення роботи на рецензування	Листопад 2024 р.
Попередній захист роботи на кафедрі	Грудень 2024 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		Дата	Підпис
Розділ 1	к.т.н, доцент Горда О.В.		
Розділ 2	к.т.н, доцент Горда О.В.		
Розділ 3	к.т.н, доцент Горда О.В.		
Розділ 4	к.т.н, доцент Горда О.В.		
Розділ 5	к.т.н, доцент Горда О.В.		

8. Дата видачі завдання: «_» _____ 2024 р.

Зав. кафедри

(підпис)

Гончаренко Т.А.

(прізвище та ініціали)

Керівник

(підпис)

Горда О.В.

(прізвище та ініціали)

Здобувач

(підпис)

Осокін А.С.

(прізвище та ініціали)

РЕФЕРАТ

Обсяг роботи 156 сторінок, 51 ілюстрацій, 44 джерел посилань.

МОДЕЛЬ ПІДВИЩЕННЯ ЯКОСТІ ЗОБРАЖЕНЬ, ГЕНЕРАТИВНІ ЗМАГАЛЬНІ МЕРЕЖІ, ГЛИБОКЕ НАВЧАННЯ, REAL-ESRGAN, НЕЙРОННА МЕРЕЖА, СУПЕРРОЗДІЛЕННЯ, ОБРОБКА ЗОБРАЖЕНЬ.

Об'єктом роботи є процес підвищення якості цифрових зображень з використанням методів генеративних змагальних мереж.

Предметом роботи є методи програмний засіб та алгоритми підвищення якості зображень на основі генеративних змагальних мереж.

Метою роботи є розробка та реалізація моделі підвищення якості цифрових зображень з використанням архітектури Real-ESRGAN Compact для збільшення роздільної здатності та покращення візуальної якості зображень.

Методи розроблення: глибоке навчання, генеративні змагальні мережі, комп'ютерний зір, методи обробки цифрових зображень.

Інструменти розроблення: фреймворк PyTorch, мова програмування Python, Jupyter Notebook, Anaconda.

Результати роботи: розроблено та реалізовано систему на базі архітектури Real-ESRGAN Compact, яка забезпечує чотирикратне збільшення роздільної здатності зображень з одночасним покращенням їх якості. Система демонструє високу ефективність у видаленні шуму, корекції розмиття та відновленні дрібних деталей зображення.

Новизна результатів полягає у оптимізації архітектури Real-ESRGAN для ефективної роботи з обмеженими обчислювальними ресурсами при збереженні високої якості результатів.

Значимість роботи полягає у створенні ефективного інструменту для автоматизованого підвищення якості зображень, що має широкий спектр практичних застосувань.

ABSTRACT

The work consists of 156 pages, 51 illustrations, 44 references.

IMAGE QUALITY ENHANCEMENT MODEL, GENERATIVE ADVERSARIAL NETWORKS, DEEP LEARNING, REAL-ESRGAN, NEURAL NETWORK, SUPER-RESOLUTION, IMAGE PROCESSING.

The object of the work is the process of improving digital image quality using generative adversarial networks methods.

The subject of the work is software tools and algorithms for improving image quality based on generative adversarial networks.

The aim of the work is to develop and implement a model for improving digital image quality using the Real-ESRGAN Compact architecture to increase resolution and enhance visual image quality.

Development methods: deep learning, generative adversarial networks, computer vision, digital image processing methods.

Development tools: PyTorch framework, Python programming language, Jupyter Notebook, Anaconda.

Results of the work: developed and implemented a system based on the Real-ESRGAN Compact architecture, which provides a four-fold increase in image resolution while simultaneously improving their quality. The system demonstrates high efficiency in noise removal, blur correction, and fine detail restoration.

The novelty of the results lies in optimizing the Real-ESRGAN architecture for efficient operation with limited computational resources while maintaining high-quality results.

The significance of the work lies in creating an effective tool for automated image quality enhancement that has a wide range of practical applications.

ЗМІСТ

ЗМІСТ	6
ВСТУП.....	9
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Аналіз проблеми	10
1.2 Визначення цілей.....	15
1.3 Аналіз факторів впливу на якість зображень	17
1.4 Аналіз спотворень зображень	25
1.5 Аналіз методів оцінки якості зображень	32
1.6 Традиційні методи покращення якості зображень	36
1.7 Сучасні методи покращення якості зображень	44
1.8 Постановка задачі	49
Розділ 2. МАТЕМАТИЧНІ МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ ТА ПРОГРАМНЕ ПРОЕКТУВАННЯ.....	51
2.1 Застосування онтологічного підходу для вибору робочого класу зображень	51
2.2 Моделі представлення зображень.....	55
2.3 Вибір та обґрунтування методів оцінки якості зображень	59
2.4 Аналіз та вибір архітектури GAN для вирішення поставленої задачі.....	61
2.5 Математичний опис та проектування генератора.....	64
2.6 Математичний опис та проектування дискримінатора	68
2.7 Проектування активаційних функцій.....	71
2.8 Проектування функцій втрат.....	73
Розділ 3. ПРОЕКТУВАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	76

	7
3.1 Аналіз технічних вимог та обмежень системи	76
3.2 Розробка загальної архітектури програмного рішення	79
3.3 Проектування системи попередньої обробки та підготовки даних	82
3.4 Розробка алгоритму навчання моделі	85
3.5 Проектування системи оцінки якості результатів.....	89
3.6 Розробка стратегії оптимізації обчислювальних ресурсів	90
3.7 Формування методики тестування та оцінки результатів.....	91
Розділ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ	94
4.1 Налаштування робочого середовища	94
4.2 Імплементація системи обробки та завантаження даних	96
4.3 Реалізація архітектури генератора та дискримінатора	98
4.4 Розробка та оптимізація процесу навчання моделі.....	100
4.5 Імплементація системи оцінки якості та логування результатів.....	102
4.6 Проведення експериментів та аналіз ефективності моделі	105
4.7 Тестування системи на контрольних прикладах.....	116
Розділ 5. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ	123
5.1 Резюме проекту.....	123
5.2 Детальний опис програмного продукту та послуг.....	124
5.3 Аналіз ринку систем обробки зображень	125
5.4 Оцінка ринку збуту та потенційного попиту.....	128
5.5 Аналіз конкурентного середовища.....	130
5.6 Стратегія маркетингу	132

5.7 План розробки та впровадження системи.....	135
5.8 Організаційний план реалізації проекту	137
5.9 Юридичне забезпечення проекту.....	139
5.10 Фінансовий план проекту	142
5.11 Стратегія фінансування розробки.....	143
5.12 Оцінка ризиків проекту.....	147
5.13 Страхування та методи мінімізації ризиків	148
ВИСНОВОК.....	151
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	153
ДОДАТКИ.....	157

ВСТУП

Стрімкий розвиток цифрових технологій та зростаюча роль візуального контенту в сучасному світі створюють постійну потребу в ефективних методах підвищення якості зображень. Особливої актуальності набуває проблема покращення якості зображень при їх масштабуванні, що є критичним для багатьох сфер застосування – від медичної діагностики до систем комп'ютерного зору.

В останні роки значного прогресу в вирішенні цієї задачі досягнуто завдяки використанню генеративних змагальних мереж (GAN). Ці нейромережеві архітектури демонструють вражаючі результати у відновленні деталей та текстур при збільшенні роздільної здатності зображень. Особливо перспективним є напрямок розвитку компактних архітектур GAN, які зберігають високу якість результатів при оптимізованому використанні обчислювальних ресурсів.

Практична значимість роботи обумовлена широким спектром потенційних застосувань розробленої системи – від покращення якості фотографій для соціальних мереж до обробки медичних знімків та реставрації архівних матеріалів. Використання сучасних технологій глибокого навчання та оптимізованої архітектури дозволяє досягти високої якості результатів при збереженні практичної застосовності системи в реальних умовах використання.

В роботі представлено комплексний підхід до вирішення задачі, що включає теоретичне обґрунтування обраних методів, детальний опис архітектури системи, реалізацію програмного рішення та експериментальну валідацію результатів. Особлива увага приділяється питанням оптимізації та забезпечення стабільності роботи системи, що є критичним для її практичного застосування [1].

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз проблеми

У сучасному інформаційному суспільстві візуальна інформація відіграє ключову роль, пронизуючи практично всі сфери людської діяльності. Від медичної діагностики до космічних досліджень, від збереження культурної спадщини до забезпечення громадської безпеки – якість зображень стає визначальним фактором ефективності та точності в багатьох галузях. Проте, незважаючи на стрімкий технологічний прогрес, проблема підвищення якості зображень залишається актуальною та багатогранною, вимагаючи постійного вдосконалення існуючих методів та розробки нових підходів.

Актуальність проблеми підвищення якості зображень зумовлена низкою взаємопов'язаних факторів. По-перше, спостерігається експоненціальне зростання обсягів візуального контенту, що генерується щодня завдяки поширенню смартфонів, цифрових камер та інших пристроїв для створення зображень. Це призводить до підвищення вимог щодо ефективності алгоритмів обробки та покращення якості зображень, здатних працювати з великими обсягами даних у реальному часі [2].

По-друге, незважаючи на вдосконалення технологій створення зображень, існують об'єктивні обмеження, пов'язані з фізичними властивостями оптичних систем, умовами зйомки та характеристиками сенсорів. Ці обмеження призводять до виникнення різноманітних спотворень, таких як шуми, розмиття, недостатня роздільна здатність, що негативно впливають на якість зображень та їх інформативність.

По-третє, в багатьох професійних сферах зростають вимоги до якості зображень, що перевищують можливості сучасних технологій зйомки. Наприклад, у медичній діагностиці підвищення якості рентгенівських знімків, МРТ та КТ зображень може суттєво вплинути на точність діагнозу та ефективність лікування. В астрономії та

космічних дослідженнях покращення якості телескопічних зображень дозволяє виявляти нові небесні об'єкти та вивчати їх властивості з більшою точністю.

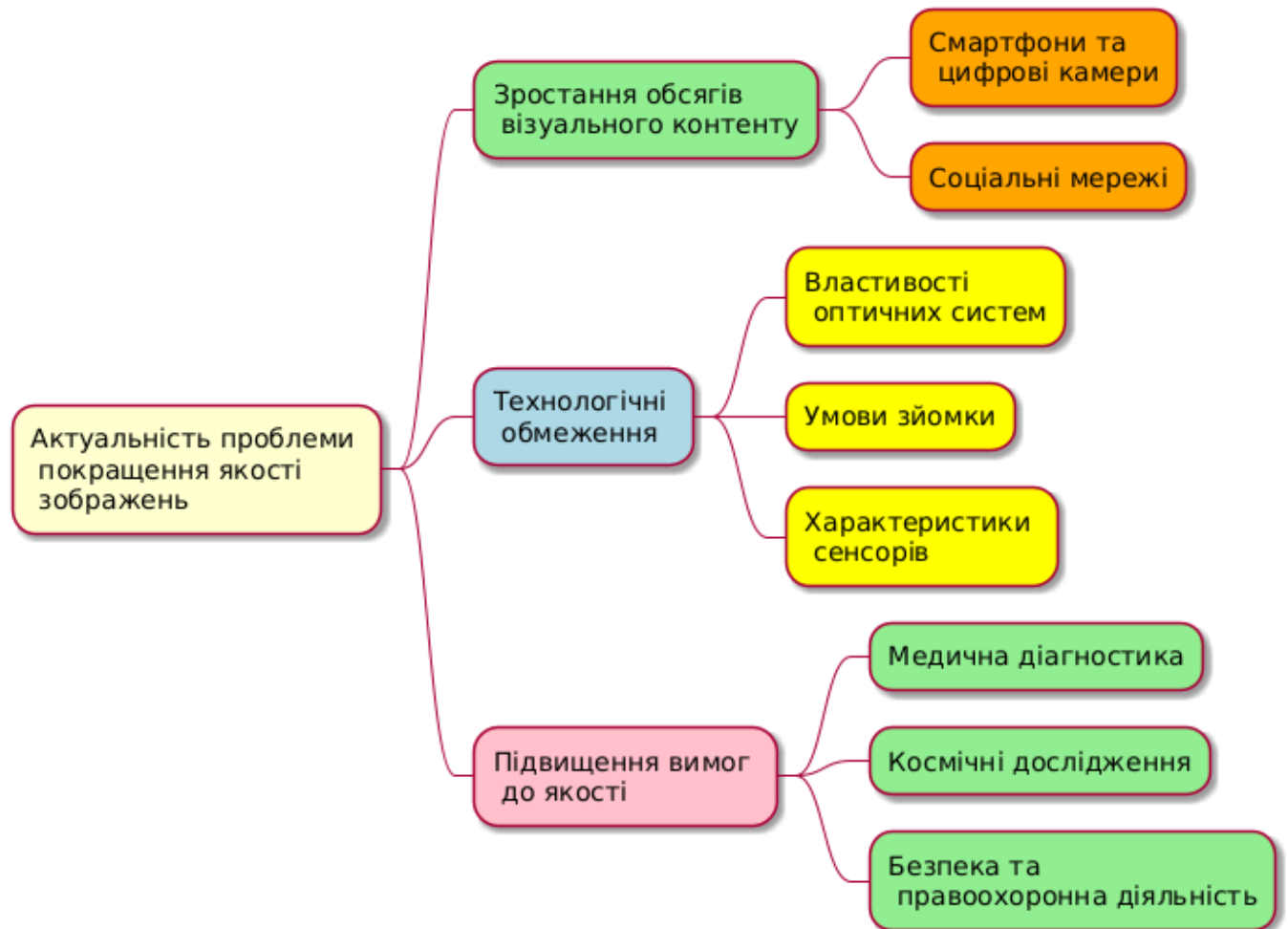


Рисунок 1.1 – Візуалізація факторів, що зумовлюють актуальність проблеми покращення якості зображень

Практична цінність вирішення проблеми підвищення якості зображень виявляється у широкому спектрі застосувань. У сфері безпеки та правоохоронної діяльності покращення якості відеозаписів та фотографій може стати ключовим фактором у розкритті злочинів та забезпеченні громадського порядку. У галузі дистанційного зондування Землі підвищення якості супутникових знімків дозволяє

більш точно відстежувати зміни клімату, прогнозувати природні катастрофи та оцінювати стан сільськогосподарських угідь.

У сфері культури та мистецтва методи підвищення якості зображень застосовуються для реставрації старих фотографій та картин, що сприяє збереженню культурної спадщини. В індустрії розваг та кіновиробництва ці технології використовуються для покращення якості архівних матеріалів та створення високоякісного візуального контенту.

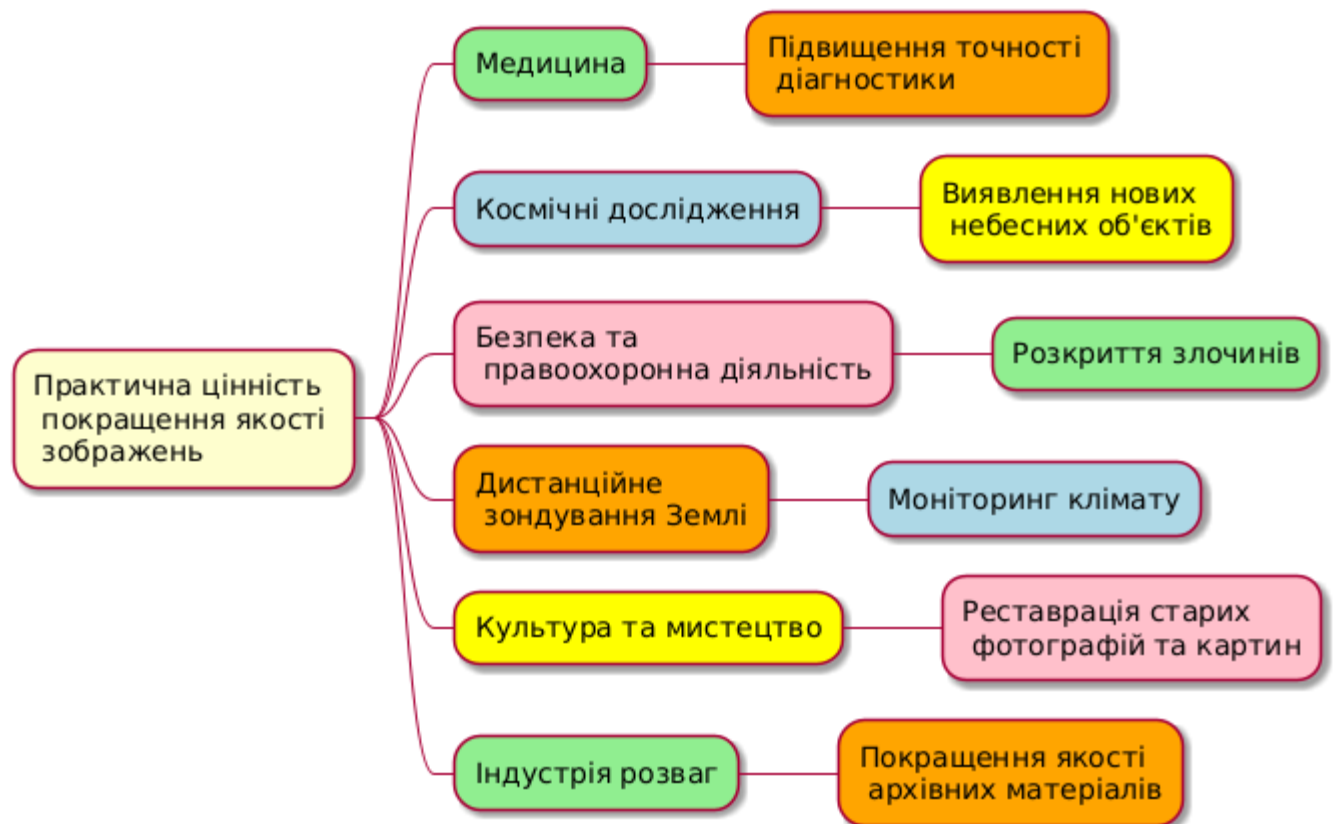


Рисунок 1.2 – Ілюстрація практичної цінності покращення якості зображень у різних сферах

Проблема підвищення якості зображень тісно пов'язана з іншими напрямками досліджень у галузі обробки інформації та комп'ютерних наук. Вона інтегрується з теорією обробки сигналів, зокрема в аспектах фільтрації шумів, відновлення сигналів

та компресії даних. У сфері комп'ютерного зору якість зображень є критичним фактором для ефективності алгоритмів розпізнавання об'єктів, сегментації зображень та трекінгу рухомих об'єктів.

Розвиток методів машинного навчання та штучного інтелекту відкриває нові перспективи для вирішення проблеми підвищення якості зображень. Зокрема, застосування глибоких нейронних мереж та генеративних змагальних мереж (GAN) дозволяє створювати алгоритми, здатні не лише покращувати якість існуючих зображень, але й генерувати деталі, відсутні в оригінальному зображенні, базуючись на статистичних моделях та великих наборах даних [3].



Рисунок 1.3 – Візуалізація зв'язків між проблемою покращення якості зображень та іншими задачами

Історичний розвиток методів підвищення якості зображень відображає еволюцію технологій обробки інформації в цілому. Початковий етап, що припадає на 1960-1970-ті роки, характеризувався розробкою базових алгоритмів фільтрації та відновлення зображень, заснованих на класичних методах обробки сигналів. У цей період були закладені основи частотного аналізу зображень з використанням перетворення Фур'є.

1980-1990-ті роки ознаменувалися розвитком більш складних методів, таких як вейвлет-аналіз та нелінійна фільтрація. Ці підходи дозволили більш ефективно боротися з шумами та артефактами в зображеннях, зберігаючи при цьому важливі деталі та контури. У цей же період почали розвиватися методи суперрозділення, спрямовані на підвищення роздільної здатності зображень.

Початок 2000-х років став поворотним моментом у розвитку методів підвищення якості зображень завдяки впровадженню технологій машинного навчання. Спочатку це були відносно прості моделі, такі як метод опорних векторів (SVM) та неглибокі нейронні мережі. Проте справжній прорив відбувся з розвитком глибокого навчання та появою згорткових нейронних мереж (CNN), які виявилися надзвичайно ефективними для вирішення різноманітних задач обробки зображень [4].

Сучасний етап, що розпочався приблизно з 2014 року, характеризується активним застосуванням генеративних моделей, зокрема генеративних змагальних мереж (GAN). Ці моделі дозволяють не лише покращувати якість існуючих зображень, але й синтезувати нові деталі, базуючись на статистичних закономірностях, виявлених у великих наборах даних. Це відкриває нові можливості для вирішення таких складних задач, як відновлення сильно пошкоджених зображень, збільшення роздільної здатності та навіть реконструкція тривимірних об'єктів з двовимірних зображень.

Таким чином, проблема підвищення якості зображень залишається актуальною та багатогранною, перебуваючи на перетині різних галузей науки та технологій. Її

вирішення вимагає комплексного підходу, що поєднує глибоке розуміння природи зображень, класичні методи обробки сигналів та сучасні досягнення у сфері штучного інтелекту. Подальший прогрес у цій галузі обіцяє не лише покращення якості візуальної інформації, але й відкриття нових можливостей для аналізу та інтерпретації зображень у різноманітних сферах людської діяльності.

1.2 Визначення цілей

Визначення чітких та структурованих цілей є ключовим етапом у процесі розробки. Цей етап дозволяє сформулювати чітке бачення кінцевого результату та визначити необхідні кроки для його досягнення.

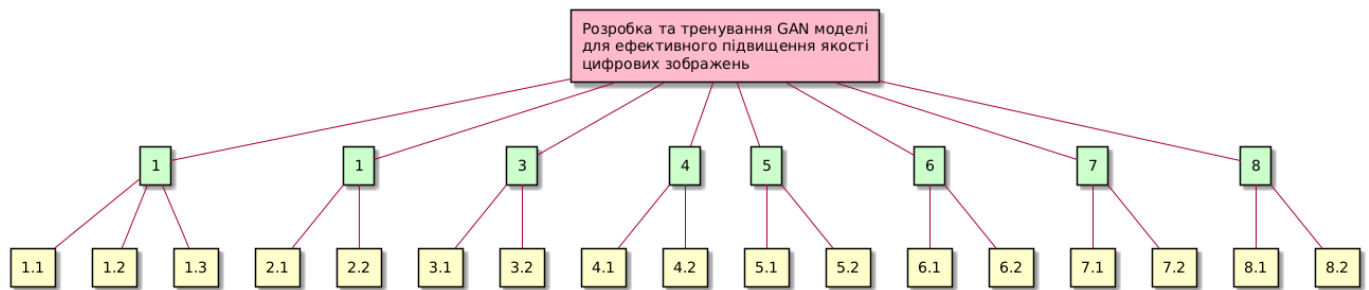


Рисунок 1.4 – Дерево цілей

Головною метою є розробка та тренування GAN моделі для ефективного підвищення якості цифрових зображень з урахуванням різноманітних факторів впливу та типів спотворень.

Для її досягнення необхідно виконання наступні підцілі:

1. Комплексний аналіз проблеми якості зображень:
 - 1.1. Дослідження факторів впливу на якість зображень;
 - 1.2. Аналіз типів спотворень зображень;
 - 1.3. Дослідження методів оцінки якості зображень.
2. Дослідження і аналіз існуючих методів підвищення якості цифрових зображень з використанням GAN:

- 2.1. Огляд літератури та останніх досліджень у галузі;
- 2.2. Вивчення різних архітектур GAN та їхніх переваг і недоліків у контексті підвищення якості зображень.
3. Вибір та адаптація архітектури GAN для задачі підвищення якості зображень:
 - 3.1. Аналіз та порівняння існуючих архітектур GAN для обробки зображень;
 - 3.2. Вибір базової архітектури та її модифікація відповідно до специфіки задачі та виявлених факторів впливу.
4. Підготовка даних та розробка стратегії тренування моделі:
 - 4.1. Збір та підготовка набору даних для тренування GAN, що враховує різні типи спотворень;
 - 4.2. Розробка стратегії тренування, включаючи вибір функцій втрат та оптимізаторів, з урахуванням виявлених факторів впливу.
5. Реалізація та тренування GAN моделі:
 - 5.1. Програмна реалізація обраної архітектури GAN;
 - 5.2. Проведення процесу тренування моделі з використанням підготовлених даних.
6. Експериментальне тестування та оцінка результатів:
 - 6.1. Застосування натренованої моделі до різних тестових зображень з різними типами спотворень;
 - 6.2. Оцінка якості отриманих результатів з використанням обраних методів оцінки якості зображень.
7. Оптимізація та вдосконалення розробленої моделі:
 - 7.1. Аналіз результатів та виявлення можливих напрямків покращення з урахуванням різних факторів впливу;
 - 7.2. Внесення змін до архітектури або стратегії тренування для підвищення ефективності моделі при роботі з різними типами спотворень.
8. Порівняльний аналіз розробленої моделі з існуючими рішеннями:

- 8.1. Вибір репрезентативних методів-конкурентів для порівняння;
- 8.2. Проведення порівняльних тестів та аналіз отриманих результатів з використанням обраних методів оцінки якості.

Подібна структура цілей інтегрує аналіз факторів впливу на якість зображень, дослідження спотворень зображень та вивчення методів оцінки якості зображень як важливі складові процесу розробки GAN моделі. Такий підхід забезпечує більш комплексне розуміння проблеми та дозволяє розробити модель, яка враховує різноманітні аспекти якості зображень.

Аналіз факторів впливу та типів спотворень допоможе створити більш репрезентативний набір даних для тренування та тестування моделі, що дозволить адаптувати архітектуру GAN для ефективної роботи з різними видами деградації якості зображень.

Включення дослідження методів оцінки якості зображень забезпечить вибір найбільш підходящих метрик для оцінки ефективності розробленої моделі. Це особливо важливо в контексті нашої роботи, де традиційні метрики якості зображень можуть не повністю відображати візуальне сприйняття генерованих зображень.

Таким чином, ця структура цілей забезпечує всебічний підхід до розробки нашої моделі для підвищення якості зображень, враховуючи всі ключові аспекти проблеми.

1.3 Аналіз факторів впливу на якість зображень

Аналіз факторів, що впливають на якість зображень, дозволяє не лише глибше зрозуміти природу проблеми, але й розробити більш цілеспрямовані та ефективні рішення, які враховують різноманітні аспекти деградації якості зображень.

Фактори, що впливають на якість зображень, можна розділити на три основні категорії: апаратні, програмні та зовнішні. Кожна з цих категорій містить ряд специфічних факторів, які по-різному впливають на кінцеву якість зображення.

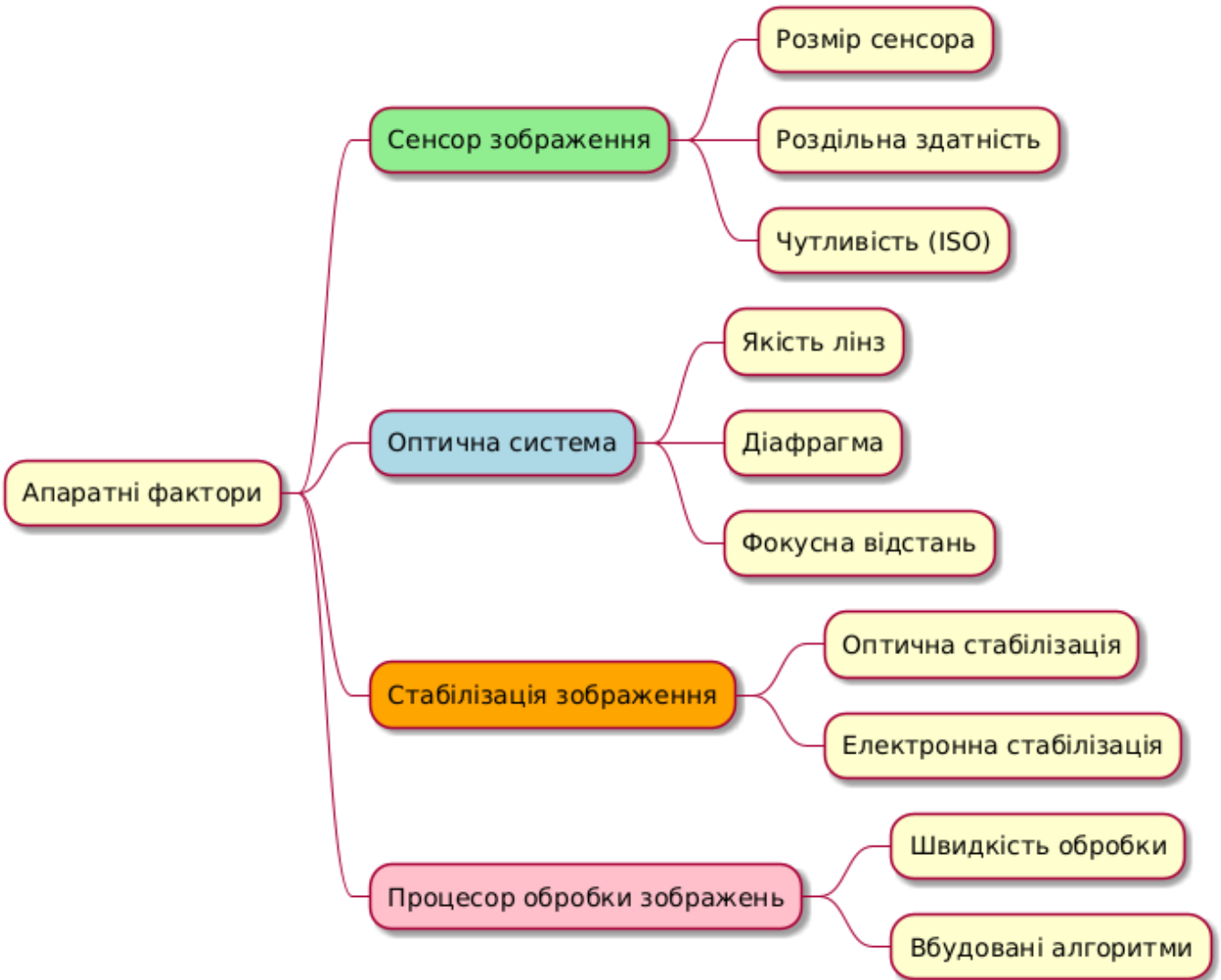


Рисунок 1.5 – Ілюстрація впливу апаратних факторів на якість зображення

Апаратні фактори пов'язані з фізичними компонентами системи створення та обробки зображень [5]. Ці фактори часто визначають базову якість зображення, яку можна отримати в ідеальних умовах.

1. Характеристики сенсора зображення:

- **Розмір сенсора:** Більший розмір сенсора зазвичай забезпечує кращу якість зображення, особливо в умовах низької освітленості.

- Роздільна здатність: Визначає кількість деталей, які можна захопити. Однак, надмірно висока роздільна здатність при малому розмірі сенсора може призвести до збільшення шуму.
- Чутливість: Вища чутливість дозволяє знімати при слабкому освітленні, але може призвести до збільшення шуму.

2. Якість оптичної системи:

- Якість лінз: Впливає на чіткість зображення, рівень хроматичних аберацій та спотворень.
- Діафрагма: Впливає на глибину різкості та кількість світла, що потрапляє на сенсор.
- Фокусна відстань: Визначає поле зору та перспективу зображення.

3. Стабілізація зображення:

- Оптична стабілізація: Зменшує вплив тремтіння камери, особливо при довгих витримках або великому збільшенні.
- Електронна стабілізація: Може допомогти зменшити розмиття, але іноді за рахунок якості зображення.

4. Процесор обробки зображень:

- Швидкість обробки: Впливає на можливість застосування складних алгоритмів обробки в реальному часі.
- Алгоритми обробки: Вбудовані алгоритми можуть покращувати або погіршувати якість зображення залежно від їх реалізації.

Програмні фактори включають алгоритми обробки зображень, які застосовуються безпосередньо в пристрої зйомки або при подальшій обробці. Ці фактори можуть значно змінити якість зображення, як покращуючи його, так і вносячи артефакти.



Рисунок 1.6 – Ілюстрація впливу програмних факторів на якість зображення

1. Алгоритми шумопоглинання:

- Просторові фільтри: Можуть ефективно зменшувати шум, але також можуть призвести до втрати дрібних деталей.
- Частотні фільтри: Дозволяють вибірково видаляти шум в певних частотних діапазонах.

- Адаптивні методи: Враховують локальні характеристики зображення для більш ефективного шумопоглинання.

2. Корекція кольору:

- Баланс білого: Коригує кольорову температуру зображення для більш природного відтворення кольорів.
- Насиченість: Регулювання насиченості може покращити візуальне сприйняття, але надмірне збільшення може призвести до неприродних кольорів.
- Гама-корекція: Впливає на загальну яскравість та контраст зображення.

3. Підвищення різкості:

- Нерізде маскування: Популярний метод підвищення різкості, але може створювати ореоли навколо контурів.
- Адаптивне підвищення різкості: Враховує локальні характеристики зображення для більш природного результату.

4. Компресія даних:

- Алгоритми стиснення з втратами: Зменшують розмір файлу, але можуть призвести до втрати якості та появи артефактів.
- Алгоритми стиснення без втрат: Зберігають всю інформацію, але забезпечують менший ступінь стиснення.

5. HDR (High Dynamic Range) обробка:

- Тональне відображення: Дозволяє відобразити широкий динамічний діапазон сцени на обмежений діапазон дисплея.
- Злиття експозицій: Комбінує кілька зображень з різною експозицією для створення HDR-зображення.

Зовнішні фактори охоплюють умови, в яких відбувається зйомка. Ці фактори часто найважче контролювати, і вони можуть суттєво впливати на якість кінцевого зображення.

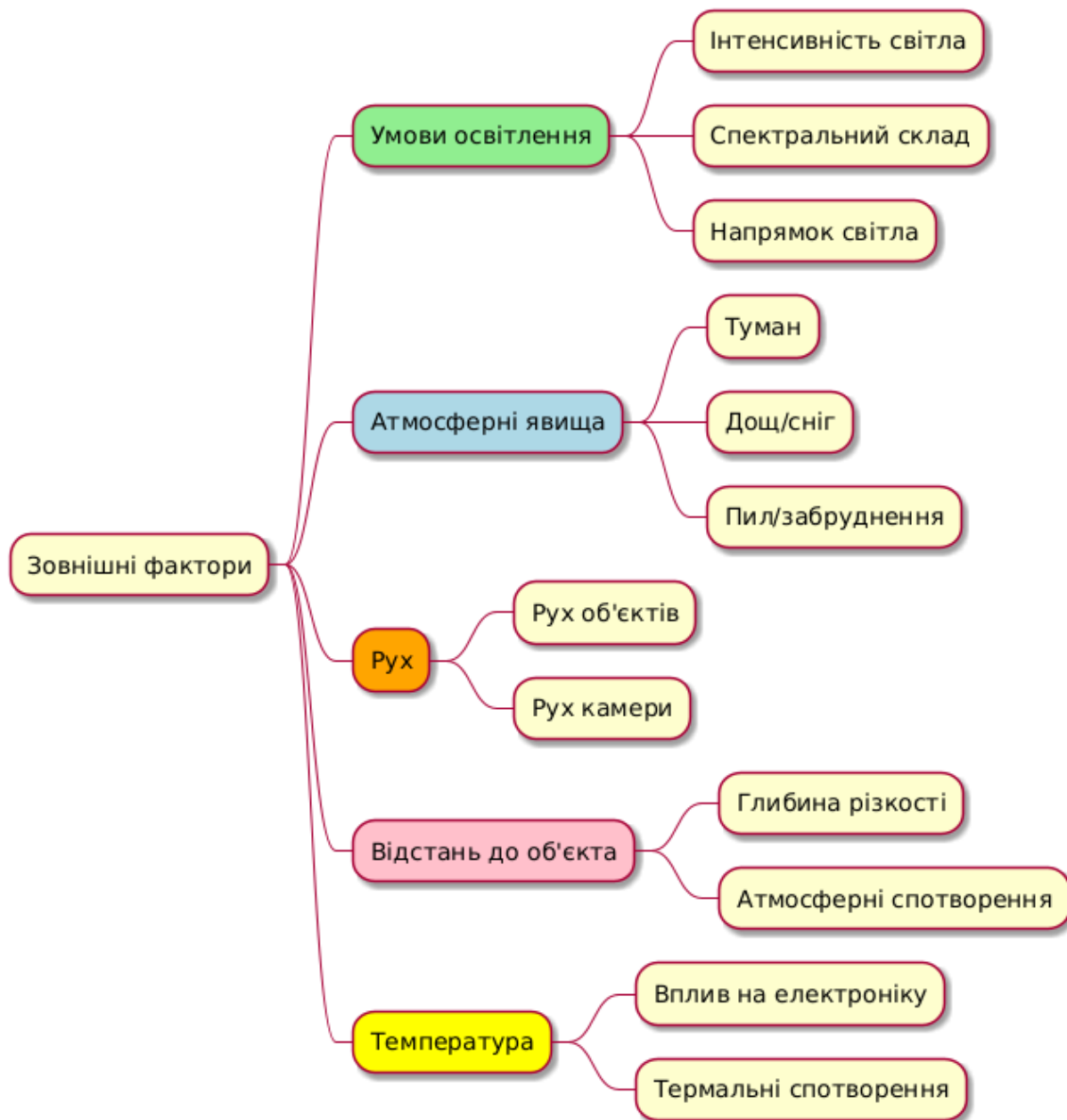


Рисунок 1.7 – Ілюстрація впливу зовнішніх факторів на якість зображення

1. Умови освітлення:

- **Інтенсивність світла:** Недостатнє освітлення призводить до підвищення рівня шуму та втрати деталей.
- **Спектральний склад світла:** Впливає на відтворення кольорів та може призвести до небажаних відтінків.

- Напрямок світла: Може створювати небажані тіні або відблиски.

2. Атмосферні явища:

- Туман: Зменшує контраст та чіткість зображення, особливо на великих відстанях.
- Дощ або сніг: Можуть створювати додаткові елементи на зображенні, які можуть бути як бажаними, так і небажаними.
- Пил або забруднення повітря: Можуть знижувати чіткість та контраст зображення.

3. Рух:

- Рух об'єктів: Може призвести до розмиття рухомих об'єктів на зображенні.
- Рух камери: Може призвести до загального розмиття зображення, особливо при довгих витримках.

4. Відстань до об'єкта:

- Вплив на глибину різкості: Близькі об'єкти можуть мати меншу глибину різкості, що може бути як бажаним, так і небажаним ефектом.
- Атмосферні спотворення: На великих відстанях атмосфера може значно впливати на якість зображення.

5. Температура:

- Вплив на роботу електроніки: Екстремальні температури можуть впливати на роботу сенсора та інших компонентів камери.
- Термальні спотворення: Висока температура може призвести до появи теплового шуму на зображенні.

Важливо відзначити, що ці фактори часто взаємодіють між собою, створюючи комплексний вплив на якість зображення. Наприклад, в умовах низької освітленості (зовнішній фактор) може знадобитися підвищення чутливості сенсора (апаратний

фактор), що, в свою чергу, призведе до збільшення рівня шуму, який потім буде оброблятися алгоритмом шумопоглинання (програмний фактор) [6].

Для систематизації цих факторів та їх впливу на якість зображень доцільно створити базову концептуальну структуру. Вона може служити основою для розробки GAN моделі, допомагаючи визначити ключові аспекти, які потрібно враховувати при проектуванні архітектури та стратегії тренування. Структура може включати не лише самі фактори, але й їх взаємозв'язки, типові наслідки їх впливу на зображення, а також можливі методи корекції. Це дозволить створити більш цілісне розуміння проблеми та розробити більш ефективні стратегії для її вирішення.

Важливо зазначити, що вплив цих факторів на якість зображення може варіюватися залежно від конкретного застосування та вимог користувача. Наприклад, для медичних зображень критично важливим є збереження дрібних деталей, тоді як для фотографій у соціальних мережах більш важливим може бути загальне візуальне сприйняття [7]. Тому при розробці GAN моделі важливо враховувати не лише технічні аспекти якості зображення, але і контекст їх використання.

Розуміння цих факторів та їх взаємодії є критично важливим для розробки, дозволяє створити більш репрезентативний набір даних для тренування, який охоплює різноманітні сценарії деградації якості зображень. Крім того, це знання може бути використане для адаптації архітектури GAN, наприклад, шляхом включення спеціалізованих шарів або втрат, які враховують специфічні типи деградації якості.

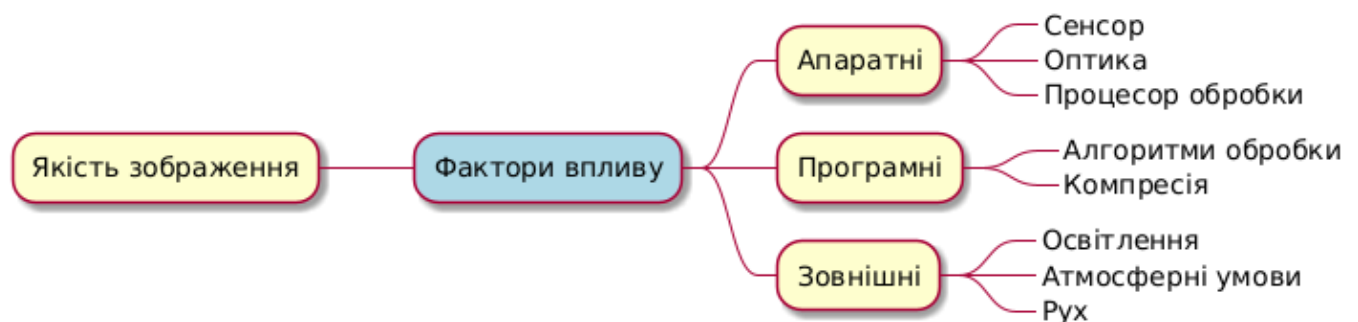


Рисунок 1.8 – Схема факторів впливу на якість зображення

Таким чином, ретельний аналіз факторів впливу на якість зображень дозволяє створити більш повне розуміння проблеми, розробити більш репрезентативні набори даних для тренування та тестування, а також адаптувати подальшу архітектуру моделі для ефективного вирішення різноманітних сценаріїв деградації якості зображень.

1.4 Аналіз спотворень зображень

Аналіз спотворень зображень є логічним продовженням дослідження факторів впливу на якість зображень. Цей аналіз фокусується на конкретних проявах зниження якості, які виникають внаслідок дії різноманітних факторів, розглянутих у попередньому розділі. Глибоке розуміння природи цих спотворень є фундаментальним для розробки ефективних методів підвищення якості зображень.

Спотворення зображень можна класифікувати за різними критеріями, але найбільш загальний поділ – це розділення на структурні та неструктурні спотворення [8].

Структурні спотворення.

1. Розмиття (blur): Розмиття є одним з найпоширеніших типів спотворень, яке суттєво впливає на чіткість та деталізацію зображення. Воно може виникати з різних причин:
 - Розмиття руху (motion blur): виникає при русі камери або об'єкта під час експозиції. Характеризується направленим "змазуванням" деталей у напрямку руху.
 - Розфокусування (defocus blur): результат неправильного фокусування камери. Проявляється як загальне зниження різкості по всьому зображенню або в певних його частинах.
 - Оптичне розмиття: пов'язане з недосконаlostями оптичної системи, такими як сферичні аберації. Розмиття безпосередньо пов'язане з апаратними факторами (якість оптики, наявність стабілізації) та

зовнішніми умовами (рух, освітлення). Воно особливо помітне при недостатньому освітленні, коли потрібна довша витримка.

2. Геометричні спотворення: Ці спотворення змінюють геометрію зображення і включають:

- Дисторсія: викривлення прямих ліній, особливо помітне по краях зображення. Може бути подушкоподібною або бочкоподібною.
- Хроматичні аберації: проявляються як кольорові контури навколо об'єктів, особливо по краях кадру.
- Астигматизм: нерівномірна фокусування в різних напрямках, що призводить до розмиття в певних областях зображення. Геометричні спотворення в основному пов'язані з апаратними факторами, зокрема з якістю та конструкцією оптичної системи.

3. Ефект аліасингу: Аліасинг виникає при недостатній частоті дискретизації зображення і проявляється у вигляді:

- "Сходинок" на похилих лініях та кривих.
- Муарового візерунку на текстурах з дрібними повторюваними елементами.
- Мерехтіння дрібних деталей при русі в відео.

4. Ефект аліасингу: Аліасинг виникає при недостатній частоті дискретизації зображення і проявляється у вигляді:

- "Сходинок" на похилих лініях та кривих.
- Муарового візерунку на текстурах з дрібними повторюваними елементами.
- Мерехтіння дрібних деталей при русі в відео.

Це спотворення пов'язане як з апаратними (роздільна здатність сенсора), так і з програмними факторами (алгоритми обробки зображень та відсутністю або неефективністю антиаліасингу).



Розмиття руху



Розфокусування



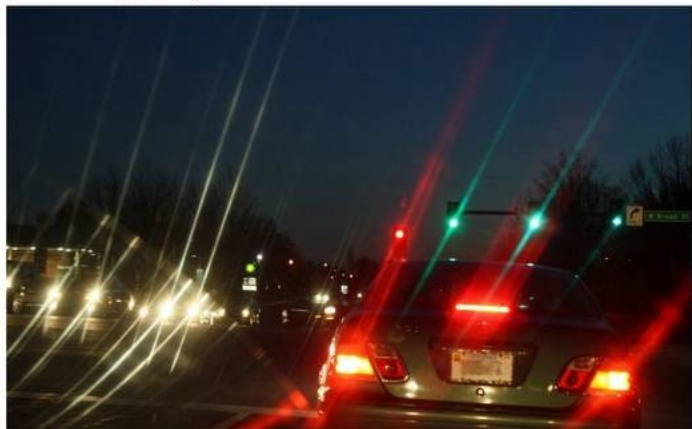
Оптичне Розмиття (Сферична Аберация)



Дисторсія



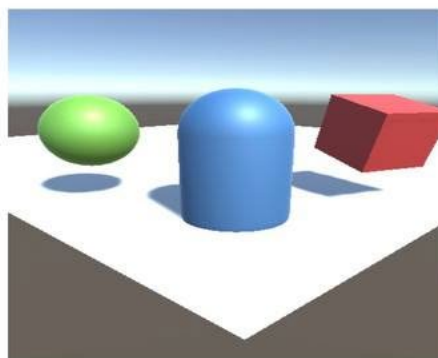
Хроматична Аберация



Астигматизм



Аліасинг (Муар)



Аліасинг (Сходи́нки)

Рисунок 1.9 – Ілюстрація різних типів структурних спотворень з прикладами

Неструктурні спотворення.

1. Шум: Шум є одним з найбільш поширених і складних для усунення спотворень.

Розрізняють кілька типів шуму:

- Гаусівський шум: має нормальний розподіл і впливає на всі пікселі зображення.
- Імпульсний шум (сіть і перець): проявляється як випадкові білі та чорні пікселі.
- Дробовий шум: пов'язаний з квантовою природою світла і особливо помітний при низькій освітленості.
- Шум квантування: виникає при аналого-цифровому перетворенні сигналу.

Шум є наслідком як апаратних (теплові шуми сенсора, електронні шуми в системі), так і зовнішніх факторів (недостатнє освітлення, високі значення ISO).

2. Артефакти компресії: Виникають при стисненні зображення з втратами і можуть проявлятися як:

- Блочність: помітні квадратні блоки, особливо в областях з плавними переходами кольору.
- Ореоли навколо контрастних границь.
- Втрата дрібних деталей та текстур.

3. Недостатній динамічний діапазон: Проявляється у вигляді втрати деталей у дуже світлих або дуже темних ділянках зображення:

- Пересвічення: повна втрата деталей у яскравих областях.
- Недостатня експозиція: втрата інформації в темних ділянках, які виглядають як суцільна чорнота.

Це спотворення пов'язане як з апаратними (обмеження динамічного діапазону сенсора), так і з зовнішніми факторами (складні умови освітлення з високим контрастом).

4. Кольорові спотворення: Включають різні проблеми з передачею кольору:

- Неправильний баланс білого: загальний кольоровий зсув всього зображення.
- Недостатня насиченість кольорів: тьмяність і невиразність кольорів.
- Кольорові шуми: особливо помітні в темних областях при високих значеннях ISO.



Гаусівський шум



Імпульсний Шум



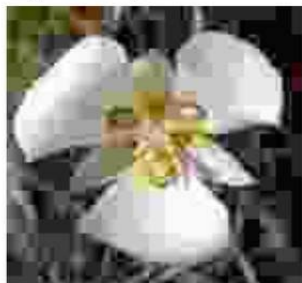
Пересвічення



Шум Квантування



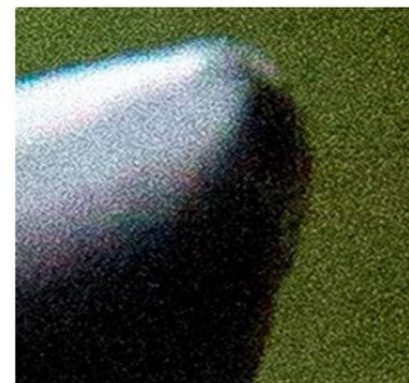
Недостатня Експозиція



Артефакти Компресії



Різні значення Балансу Білого



Кольорові Шуми

Рисунок 1.10 – Ілюстрація різних типів неструктурних спотворень з прикладами

Ці спотворення можуть бути викликані як апаратними (якість сенсора та системи обробки кольору), так і програмними (алгоритми корекції кольору) та зовнішніми факторами (спектральний склад освітлення) [9].

Роздільна здатність.

Окремо варто розглянути питання роздільної здатності зображень. Хоча недостатня роздільна здатність не є спотворенням у класичному розумінні, вона суттєво впливає на якість зображення і часто розглядається в контексті задач підвищення якості.

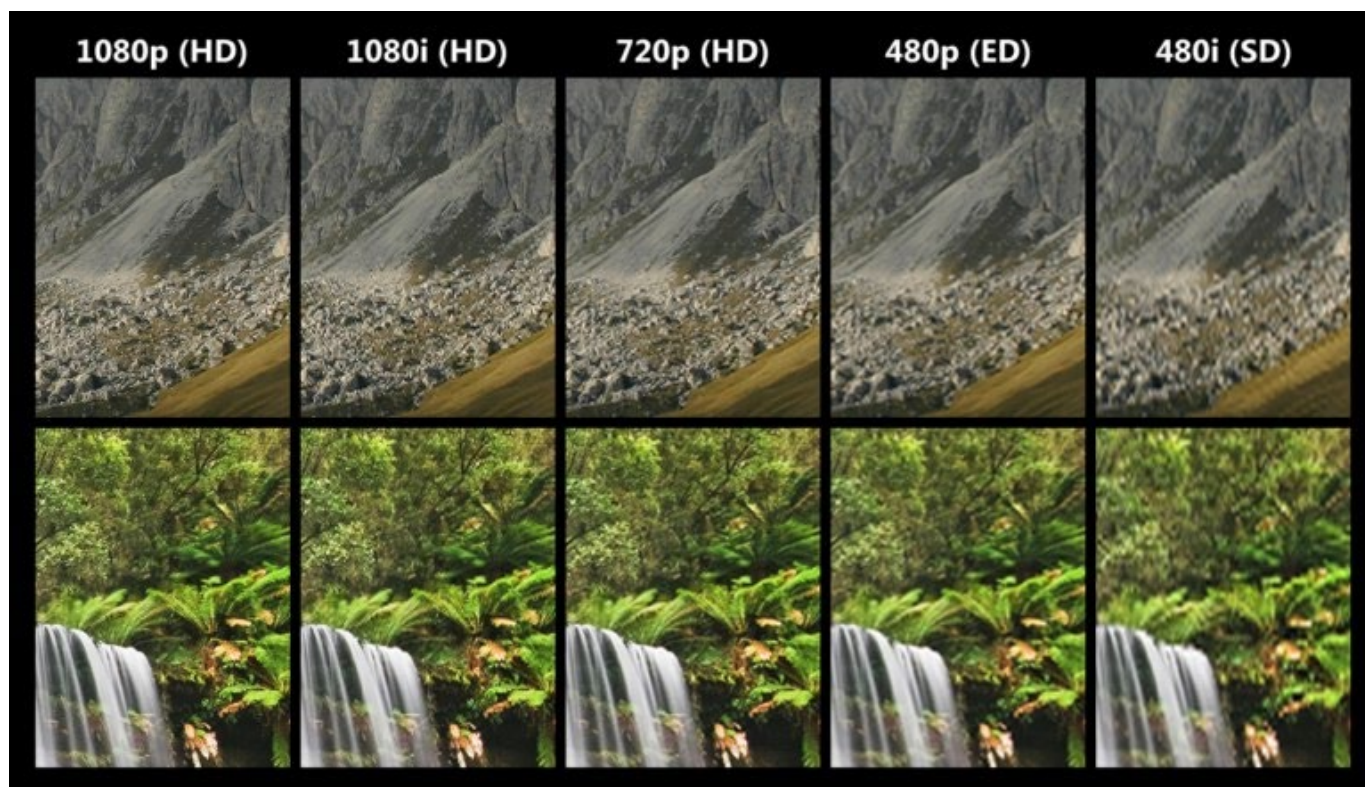


Рисунок 1.11 – Візуалізація впливу роздільної здатності на якість зображення з прикладами

Роздільна здатність визначається кількістю пікселів, які формують зображення, і безпосередньо впливає на рівень деталізації [10]. Недостатня роздільна здатність призводить до:

1. Втрати дрібних деталей.
2. Загальної "нечіткості" зображення.
3. Посилення ефекту аліасингу при масштабуванні.

Проблема низької роздільної здатності може бути наслідком як апаратних обмежень (характеристики сенсора камери), так і програмних факторів (зменшення розміру зображення для економії місця або прискорення передачі).

Важливо розуміти, що просте збільшення кількості пікселів не вирішує проблему, оскільки не додає нової інформації до зображення. Тому задача підвищення роздільної здатності (super-resolution) є однією з найскладніших у області обробки зображень і вимагає складних алгоритмів для "вгадування" відсутніх деталей на основі наявної інформації та статистичних моделей.

Взаємодія різних типів спотворень.

Варто зазначити, що різні типи спотворень часто виникають одночасно і можуть взаємодіяти між собою, створюючи комплексні ефекти. Наприклад:

1. Низька роздільна здатність може маскувати деякі види шуму, але при цьому посилювати ефект аліасингу.
2. Розмиття може зменшувати видимість шуму, але при цьому призводити до втрати важливих деталей.
3. Артефакти компресії можуть посилюватися при наявності шуму або при подальшій обробці зображення.
4. Кольорові спотворення можуть ускладнювати виявлення та корекцію інших типів спотворень.

Ця взаємодія різних типів спотворень створює додаткові складнощі при розробці алгоритмів підвищення якості зображень, оскільки вимагає комплексного підходу до вирішення проблеми.

Значення аналізу спотворень.

Розуміння природи та характеристик різних типів спотворень є критично важливим для розробки ефективних методів підвищення якості зображень. Воно дозволяє:

1. Створювати більш репрезентативні набори даних для тренування та тестування алгоритмів, які включають різноманітні комбінації спотворень.
2. Розробляти спеціалізовані алгоритми для боротьби з конкретними типами спотворень або їх комбінаціями.
3. Вдосконалювати методи оцінки якості зображень, враховуючи різні типи спотворень та їх вплив на візуальне сприйняття.
4. Покращувати алгоритми попередньої обробки зображень для мінімізації впливу спотворень на подальші етапи аналізу чи обробки.
5. Розробляти більш ефективні методи компресії зображень, які мінімізують візуально помітні спотворення.
6. Вдосконалювати апаратні компоненти систем створення та обробки зображень для мінімізації виникнення спотворень на етапі зйомки.

Таким чином, аналіз спотворень зображень є невід'ємною частиною загального дослідження проблеми якості зображень. Він тісно пов'язаний з аналізом факторів впливу і служить основою для розробки ефективних методів підвищення якості зображень, включаючи методи на основі глибокого навчання та генеративних моделей.

1.5 Аналіз методів оцінки якості зображень

Аналіз методів оцінки якості зображень є логічним продовженням дослідження факторів впливу та типів спотворень, розглянутих у попередніх розділах. Ефективна оцінка якості зображень є критично важливою для розробки та вдосконалення методів

підвищення якості, оскільки вона дозволяє об'єктивно вимірювати ефективність цих методів та порівнювати різні підходи.

Методи оцінки якості зображень можна розділити на дві основні категорії: суб'єктивні та об'єктивні. Кожна з цих категорій має свої переваги та недоліки, і вибір конкретного методу залежить від контексту застосування та специфіки задачі [11].

Суб'єктивні методи оцінки якості зображень базуються на сприйнятті людини. Вони вважаються найбільш достовірними, оскільки кінцевим споживачем візуального контенту зазвичай є людина. Однак ці методи мають ряд обмежень:

1. Часозатратність: проведення суб'єктивних тестів вимагає значного часу та ресурсів.

2. Варіативність: результати можуть відрізнятися залежно від групи оцінювачів, їх досвіду, умов проведення тесту тощо.

3. Неможливість автоматизації: суб'єктивні методи не можуть бути використані для автоматичної оцінки якості в реальному часі.

Існує кілька стандартизованих методик проведення суб'єктивної оцінки якості зображень:

1. Метод середньої оцінки (Mean Opinion Score, MOS): оцінювачі ранжують якість зображень за п'ятибальною шкалою.

2. Метод парного порівняння: оцінювачі порівнюють пари зображень і вибирають краще з кожної пари.

3. Метод одиночного стимулу: оцінювачі оцінюють кожне зображення окремо, без порівняння з іншими.

Ці методи дозволяють отримати загальну оцінку візуальної якості зображення, яка враховує всі аспекти сприйняття, включаючи різкість, контраст, кольоропередачу, наявність артефактів тощо.

Об'єктивні методи оцінки якості зображень, на відміну від суб'єктивних, базуються на математичних моделях і алгоритмах. Вони можуть бути автоматизовані

і не вимагають участі людини-оцінювача. Об'єктивні методи можна розділити на три категорії:

1. Методи з повним еталоном (Full-Reference, FR): порівнюють спотворене зображення з оригінальним, неспотвореним зображенням. Ці методи є найбільш точними, але вимагають наявності оригінального зображення, що не завжди можливо в реальних умовах.

2. Методи з частковим еталоном (Reduced-Reference, RR): використовують часткову інформацію про оригінальне зображення для оцінки якості спотвореного зображення.

3. Методи без еталону (No-Reference, NR): оцінюють якість зображення без будь-якої інформації про оригінальне зображення. Ці методи найбільш універсальні, але зазвичай менш точні.

Розглянемо деякі популярні об'єктивні метрики оцінки якості зображень:

1. Пікове відношення сигнал/шум (Peak Signal-to-Noise Ratio, PSNR): ця метрика базується на середньоквадратичній помилці між оригінальним та спотвореним зображенням. PSNR широко використовується завдяки простоті обчислення, але має обмеження, оскільки не завжди корелює з візуальним сприйняттям якості.

2. Структурна схожість (Structural Similarity Index, SSIM): ця метрика враховує структурну інформацію в зображенні і краще корелює з людським сприйняттям якості. SSIM порівнює локальні паттерни яскравості та контрасту між оригінальним та спотвореним зображеннями.

3. Візуальна інформаційна вірність (Visual Information Fidelity, VIF): ця метрика базується на теорії інформації і моделює процес сприйняття зображення людським зором. VIF оцінює кількість інформації, яка може бути вилучена з оригінального зображення в порівнянні зі спотвореним.

4. Багатомасштабна структурна схожість (Multi-Scale Structural Similarity Index, MS-SSIM): це розширення SSIM, яке враховує структурну інформацію на різних масштабах, що робить його більш стійким до змін роздільної здатності.

Говорячи про роздільну здатність, варто зазначити, що вона відіграє важливу роль в оцінці якості зображень. Традиційні метрики, такі як PSNR, можуть бути недостатньо інформативними при оцінці якості зображень з різною роздільною здатністю. Тому були розроблені спеціальні методи для оцінки якості в контексті зміни роздільної здатності:

1. Метрика якості зображення з інформацією про інформаційний вміст (Information Content Weighted PSNR, IW-PSNR): ця метрика враховує важливість різних областей зображення на основі їх інформаційного вмісту.

2. Метрика якості суперрозділення (Super-Resolution Image Quality Measure, SRIQ): спеціально розроблена для оцінки якості алгоритмів підвищення роздільної здатності.

Зв'язок з попереднім розділом про аналіз спотворень зображень є очевидним: методи оцінки якості зображень повинні враховувати різні типи спотворень, розглянуті раніше. Наприклад:

1. Для оцінки рівня розмиття були розроблені спеціальні метрики, такі як метрика якості різкості на основі кумулятивної ймовірності розмиття (Cumulative Probability of Blur Detection, CPBD).

2. Для оцінки рівня шуму використовуються такі метрики, як оцінка рівня шуму на основі блочного підходу (Block-based Noise Estimation).

3. Для оцінки артефактів компресії застосовуються специфічні метрики, такі як метрика блочності (Blocking Artifact Measure, BAM).

Важливо відзначити, що жодна окрема метрика не може повністю охопити всі аспекти якості зображення. Тому на практиці часто використовують комбінацію різних метрик для отримання більш повної оцінки.

Розробка методів оцінки якості зображень є активною областю досліджень, особливо в контексті нових технологій обробки зображень, таких як глибоке навчання. Традиційні метрики можуть бути недостатньо ефективними для оцінки якості зображень, отриманих за допомогою генеративних моделей, оскільки ці моделі можуть створювати візуально привабливі зображення, які значно відрізняються від оригіналу на піксельному рівні [12].

У зв'язку з цим, розробляються нові підходи до оцінки якості зображень, які враховують особливості сучасних методів обробки:

1. Перцептивні метрики на основі глибокого навчання: ці метрики використовують нейронні мережі, навчені на великих наборах даних, для оцінки візуальної якості зображень.

2. Метрики на основі генеративно-змагальних мереж (GAN): ці методи використовують дискримінатор GAN для оцінки реалістичності згенерованих зображень.

3. Комбіновані метрики: поєднують традиційні підходи з методами машинного навчання для більш точної оцінки якості.

Підсумовуючи, можна сказати, що вибір відповідних метрик оцінки якості залежить від конкретної задачі, типів спотворень, з якими доводиться мати справу, та загального контексту застосування. Розуміння сильних та слабких сторін різних методів оцінки дозволяє розробникам створювати більш ефективні та надійні системи обробки зображень.

1.6 Традиційні методи покращення якості зображень

Традиційні методи покращення якості зображень представляють собою фундаментальні підходи, які були розроблені та вдосконалені протягом десятиліть розвитку цифрової обробки зображень.

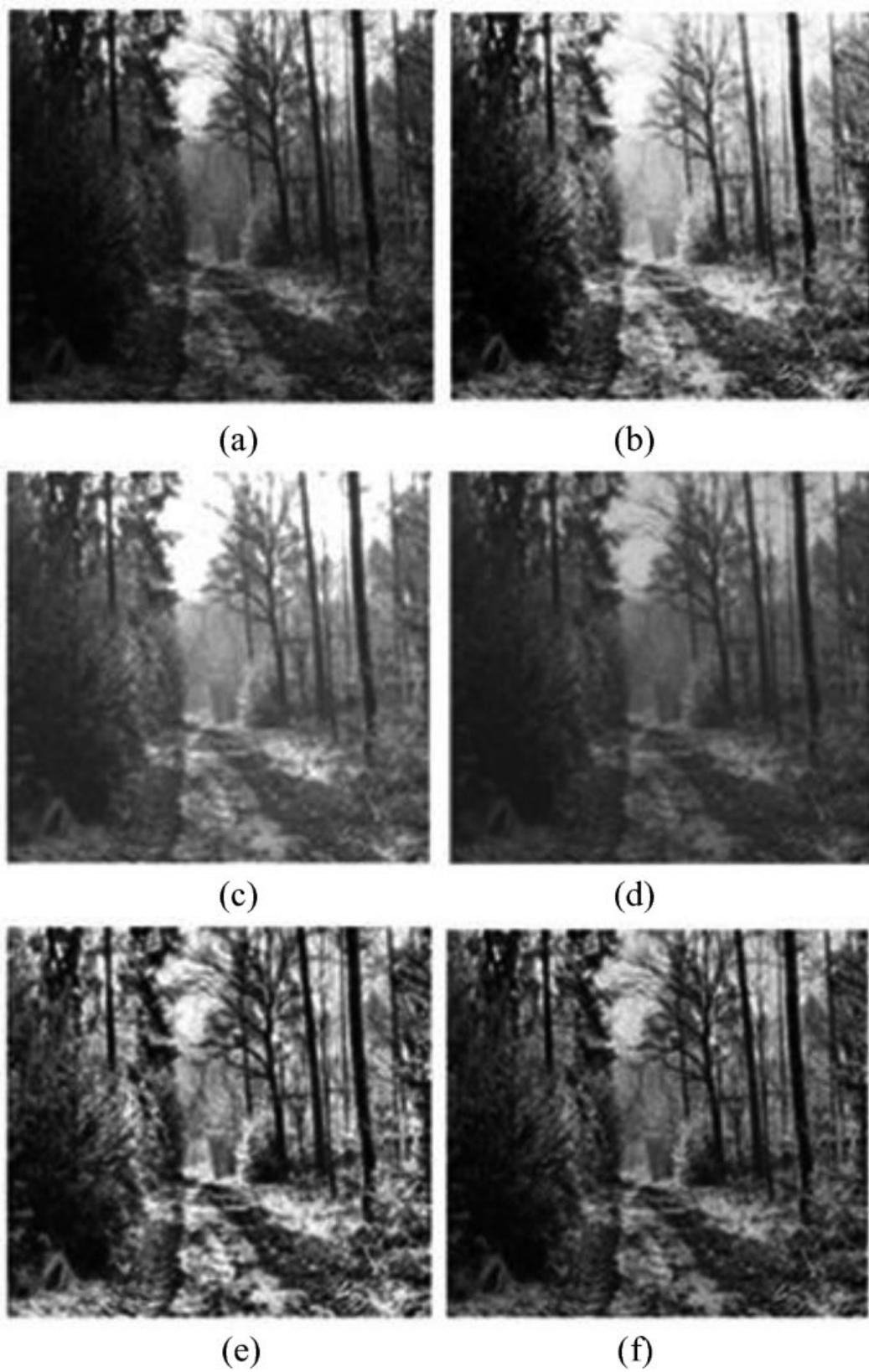


Рисунок 1.12 – Порівняння результатів різних методів корекції яскравості та контрасту

Ці методи спрямовані на вирішення конкретних проблем якості, які ми детально розглядали в попередніх розділах, зокрема в аналізі факторів впливу та типів спотворень. Важливість цих методів полягає в їх ефективності, зрозумілості та широкій застосовності в різних сферах обробки зображень.

Розглянемо основні категорії традиційних методів покращення якості зображень:

1. Методи корекції яскравості та контрасту

Ці методи спрямовані на покращення візуального сприйняття зображення шляхом оптимізації розподілу яскравості пікселів. Вони особливо ефективні для вирішення проблем з недостатнім динамічним діапазоном, які ми обговорювали в розділі про типи спотворень.

- Лінійне розтягування контрасту: Цей метод масштабує діапазон інтенсивностей пікселів на весь доступний діапазон (наприклад, 0-255 для 8-бітних зображень).
- Гістограмна еквалізація: Метод перерозподіляє значення інтенсивності пікселів для досягнення більш рівномірного розподілу. Це особливо корисно для зображень з низьким контрастом.
- Адаптивна гістограмна еквалізація: Вдосконалення попереднього методу, яке застосовує еквалізацію локально до різних частин зображення, що дозволяє краще зберегти деталі [13].

2. Методи зменшення шуму

Ці методи спрямовані на вирішення проблеми шуму, яку ми детально розглядали в розділі про типи спотворень. Вони особливо важливі для зображень, отриманих в умовах низької освітленості або з використанням високої чутливості сенсора.

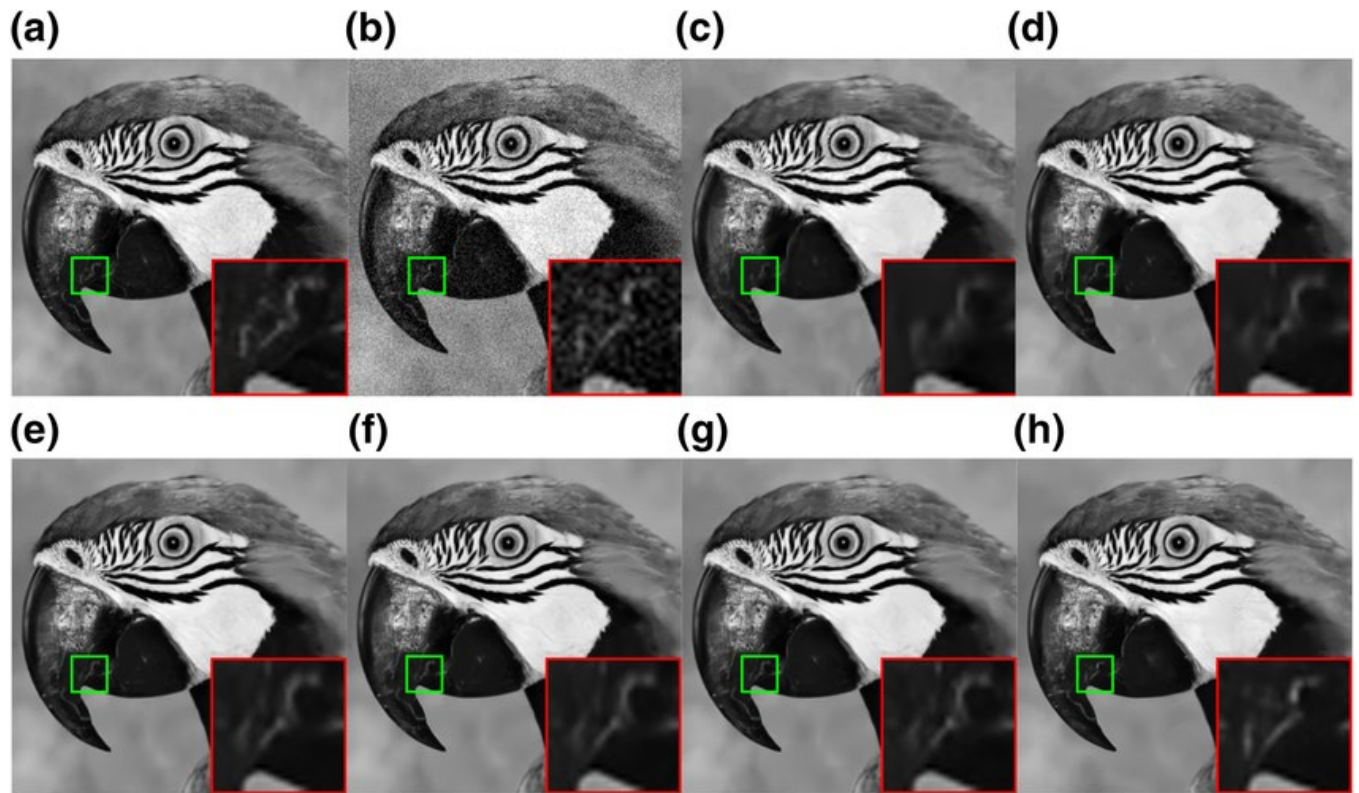


Рисунок 1.13 – Порівняння ефективності різних методів зменшення шуму

- Лінійна фільтрація: Включає такі методи, як фільтр Гауса або усереднюючий фільтр. Ці фільтри ефективні для зменшення гаусівського шуму, але можуть призводити до розмиття деталей.
- Медіанна фільтрація: Особливо ефективна для усунення імпульсного шуму ("сіль і перець"), зберігаючи при цьому краї об'єктів.
- Білатеральна фільтрація: Нелінійний метод, який враховує як просторову близькість, так і схожість інтенсивності пікселів, що дозволяє зменшувати шум, зберігаючи краї [14].
- Вейвлет-фільтрація: Використовує вейвлет-перетворення для розділення зображення на різні частотні компоненти і застосовує фільтрацію до кожного компонента окремо [15].

3. Методи підвищення різкості

Ці методи спрямовані на покращення чіткості деталей та країв об'єктів на зображенні. Вони особливо корисні для компенсації розмиття, яке ми обговорювали в розділі про типи спотворень.

- Нерізде маскування: Класичний метод, який підкреслює краї шляхом віднімання розмитої версії зображення від оригіналу.
- Фільтр підвищення різкості: Використовує ядро згортки для підкреслення країв та дрібних деталей [16].

4. Методи корекції кольору

Ці методи спрямовані на покращення кольоропередачі зображення, вирішуючи проблеми, пов'язані з кольоровими спотвореннями, які ми обговорювали раніше.

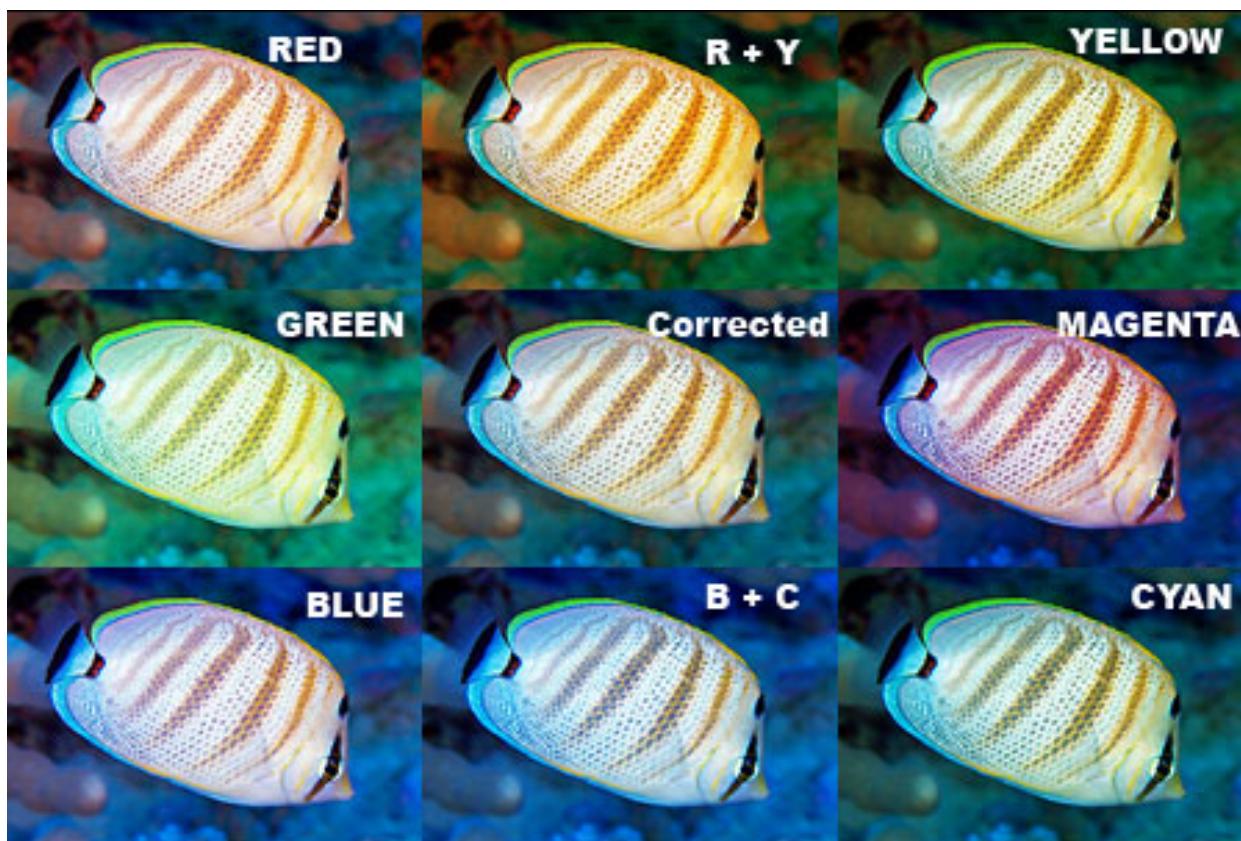


Рисунок 1.14 – Демонстрація ефектів різних методів корекції кольору

- Балансування білого: Коригує кольорову температуру зображення для досягнення нейтрального відтінку білого кольору.
- Корекція гама: Регулює співвідношення між числовими значеннями пікселів та їх фактичною яскравістю.
- Корекція насиченості: Дозволяє посилити або зменшити інтенсивність кольорів.

5. Методи збільшення роздільної здатності

Ці методи спрямовані на збільшення кількості пікселів у зображенні, що особливо важливо при необхідності збільшення розміру зображення або покращення його деталізації. Вони тісно пов'язані з проблемою недостатньої роздільної здатності, яку ми обговорювали в попередніх розділах.

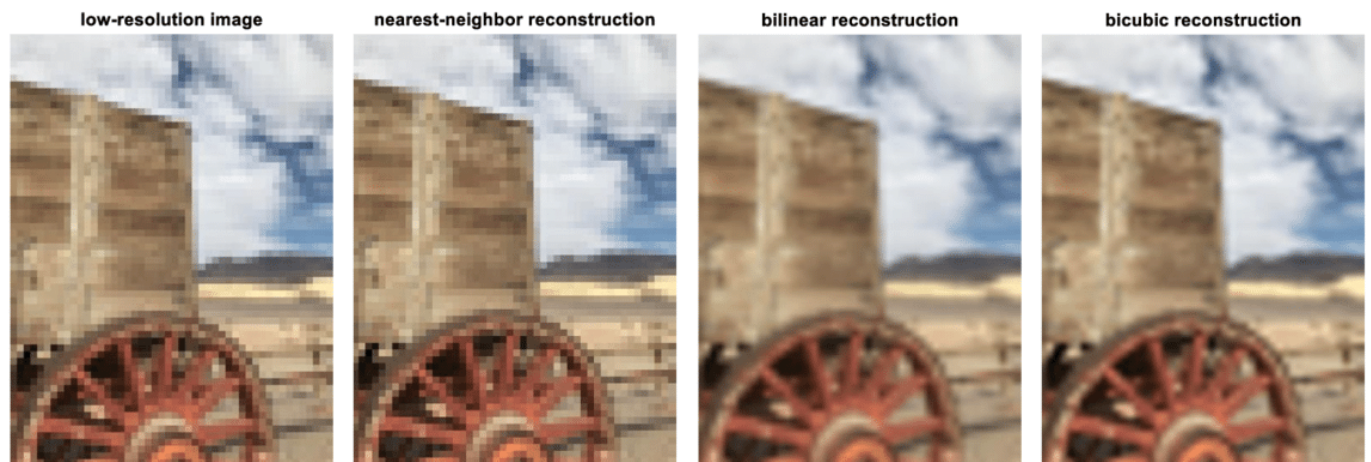


Рисунок 1.15 – Порівняння результатів різних традиційних методів збільшення роздільної здатності

- Метод найближчого сусіда: Найпростіший метод, який дублює значення найближчого пікселя. Швидкий, але може призводити до появи "сходинок" на краях об'єктів.

- Білінійна інтерполяція: Використовує зважене середнє чотирьох найближчих пікселів. Дає більш гладкий результат, ніж метод найближчого сусіда.
- Бікубічна інтерполяція: Використовує кубічні сплайни для інтерполяції значень пікселів. Забезпечує ще більш гладкий результат, але може призводити до невеликої втрати різкості.
- Ланцош інтерполяція: Використовує функцію Ланцоша для інтерполяції. Часто дає найкращий баланс між різкістю та гладкістю.

6. Методи видалення артефактів

Ці методи спрямовані на усунення специфічних артефактів, які можуть виникати в результаті стиснення зображень або інших процесів обробки [17].



Рисунок 1.15 – Приклад видалення артефактів за допомогою фільтра від Adobe

- Деблокінг фільтри: Призначені для зменшення блочних артефактів, що виникають при сильному JPEG-стисненні.

- Дерингінг фільтри: Усувають кільцеві артефакти, які можуть виникати навколо різких країв при використанні вейвлет-стиснення.

Важливо зазначити, що ці традиційні методи часто використовуються в комбінації для досягнення найкращих результатів. Наприклад, після зменшення шуму може бути застосоване підвищення різкості, а потім - корекція кольору.

Кожен з цих методів має свої переваги та недоліки, які необхідно враховувати при виборі конкретного підходу:

- Швидкість обробки: Деякі методи, такі як лінійна фільтрація або метод найближчого сусіда для збільшення роздільної здатності, дуже швидкі, але можуть давати менш якісні результати.
- Збереження деталей: Методи, які краще зберігають деталі (наприклад, білатеральна фільтрація), зазвичай вимагають більше обчислювальних ресурсів.
- Артефакти обробки: Деякі методи можуть вводити власні артефакти, наприклад, надмірне підвищення різкості може призвести до появи ореолів навколо країв.

Ефективність цих методів може бути оцінена за допомогою метрик якості зображень, які ми обговорювали в попередньому розділі. Наприклад, PSNR часто використовується для оцінки ефективності методів зменшення шуму, а SSIM може бути корисним для оцінки методів підвищення різкості.

Розуміння цих традиційних методів є критичним для розробки більш складних алгоритмів покращення якості зображень, включаючи методи на основі машинного навчання. Багато сучасних підходів, включаючи глибокі нейронні мережі, часто використовують ідеї з цих класичних методів або навіть включають їх як частину своєї архітектури.

1.7 Сучасні методи покращення якості зображень

Сучасні методи покращення якості зображень значно відрізняються від традиційних підходів, що ми розглянули раніше. Вони базуються на досягненнях у галузі машинного навчання, глибокого навчання та комп'ютерного зору, дозволяючи більш точно й ефективно обробляти зображення. Завдяки цим технологіям стає можливим не лише усунення основних дефектів зображень, як-от шум чи низька роздільна здатність, а й відновлення втрачених деталей і структур, а також корекція кольорів і підвищення чіткості.

Один із ключових елементів сучасних методів — здатність моделювати і передбачати складні процеси спотворення, враховуючи такі фактори, як розмитість через рух, освітлення, зміну контрасту і деталізації. За допомогою нейронних мереж та архітектур, таких як GAN, стало можливим навчати моделі на великих наборах даних, що дозволяє їм розпізнавати складні патерни [18].

Крім того, ці методи пропонують новий підхід до оцінки якості зображень, часто інтегруючи його безпосередньо в алгоритми покращення. Завдяки цьому система може оцінити результат не тільки на основі кількісних метрик, але й за візуальними критеріями, що враховують сприйняття людини. Це дозволяє досягти більш якісних результатів.

Розглянемо основні напрямки розвитку сучасних методів покращення якості зображень:

1. Методи на основі глибокого навчання

Глибоке навчання стало революційним підходом у галузі обробки зображень, дозволяючи створювати моделі, здатні вчитися на великих наборах даних та захоплювати складні залежності між вхідними та вихідними зображеннями.

- Згорткові нейронні мережі (CNN):

CNN стали основним інструментом для багатьох задач обробки зображень. Вони особливо ефективні для таких завдань, як зменшення шуму, підвищення різкості

та корекція кольору. Наприклад, мережа DnCNN показала вражаючі результати в задачі зменшення шуму, перевершивши традиційні методи, особливо при роботі з реальними шумними зображеннями.

- Автоенкодери:

Ці нейронні мережі ефективні для задач стиснення та відновлення зображень. Вони можуть бути використані для видалення шуму, заповнення відсутніх частин зображення та навіть для перетворення зображень з одного стилю в інший [19].

- Рекурентні нейронні мережі (RNN):

Хоча RNN зазвичай асоціюються з обробкою послідовностей, вони також знайшли застосування в обробці зображень, особливо для задач, які вимагають врахування контексту, таких як відновлення пошкоджених ділянок зображення [20].

2. Генеративні змагальні мережі (GAN)

GAN відкрили нові можливості для покращення якості зображень, дозволяючи генерувати реалістичні деталі та текстури. Вони особливо ефективні для вирішення складних задач, таких як суперрозділення та відновлення сильно пошкоджених зображень.

- SRGAN (Super-Resolution GAN):

Ця архітектура спеціально розроблена для задачі суперрозділення. SRGAN здатна генерувати високоякісні зображення з високою роздільною здатністю на основі вхідних зображень низької роздільної здатності, відновлюючи дрібні деталі та текстури.

- CycleGAN:

Ця архітектура дозволяє здійснювати перетворення між різними доменами зображень без необхідності в парних навчальних даних. Це може бути корисно, наприклад, для перетворення денних зображень у нічні або для видалення певних типів спотворень.

3. Методи на основі глибокого навчання для збільшення роздільної здатності

Збільшення роздільної здатності залишається однією з найбільш актуальних задач у галузі покращення якості зображень. Сучасні методи на основі глибокого навчання значно перевершують традиційні підходи, які ми розглядали раніше.

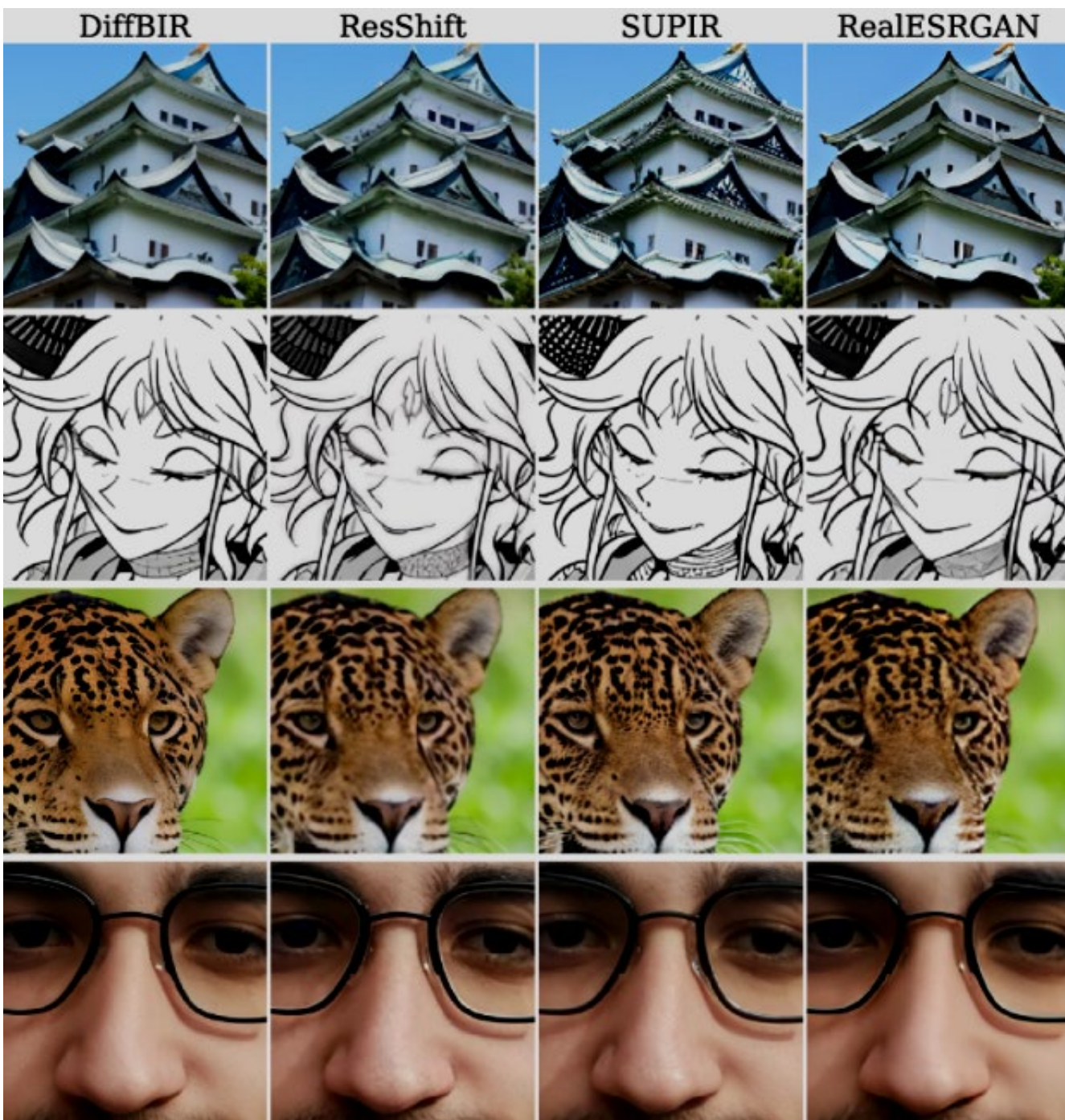


Рисунок 1.16 – Порівняння результатів різних моделей

- ESRGAN (Enhanced SRGAN):

Це вдосконалена версія SRGAN, яка використовує більш глибоку архітектуру та вдосконалену функцію втрат. ESRGAN здатна генерувати зображення з вищою роздільною здатністю та кращою якістю деталей порівняно з попередніми методами.

- RCAN (Residual Channel Attention Network):

Ця архітектура використовує механізм уваги для фокусування на найбільш інформативних частинах зображення при збільшенні роздільної здатності. Це дозволяє досягти кращих результатів, особливо при великих коефіцієнтах збільшення.

- Meta-SR:

Цей метод використовує мета-навчання для адаптації до різних коефіцієнтів збільшення роздільної здатності, дозволяючи використовувати одну модель для різних сценаріїв.

4. Методи на основі глибокого навчання для зменшення шуму

Зменшення шуму залишається критично важливою задачею, особливо для зображень, отриманих в умовах низької освітленості або з використанням високої чутливості сенсора.

- CBDNet (Convolutional Blind Denoising Network):

Ця мережа здатна ефективно видаляти шум з реальних фотографій, оцінюючи рівень та характеристики шуму безпосередньо з вхідного зображення.

- RIDNet (Real Image Denoising Network):

Ця архітектура спеціально розроблена для роботи з реальними шумними зображеннями, враховуючи складну природу реального шуму.

- Noise2Noise:

Цей інноваційний підхід дозволяє навчати моделі зменшення шуму без використання чистих зображень, що особливо корисно для сценаріїв, де отримання чистих еталонних зображень неможливе.

5. Методи покращення якості на основі перцептивних втрат

Ці методи використовують особливості людського сприйняття для оптимізації якості зображень, що тісно пов'язано з суб'єктивними методами оцінки якості, які ми обговорювали раніше.

- Perceptual Losses:

Цей підхід використовує попередньо навчені мережі (наприклад, VGG) для обчислення втрат на рівні ознак, що дозволяє оптимізувати моделі для генерації візуально привабливих результатів [21].

- Contextual Loss:

Цей метод дозволяє зберігати контекстуальну інформацію при перетворенні зображень, що особливо корисно для задач стилізації та перенесення стилю [22].

6. Мультизадачні підходи

Сучасні методи часто поєднують різні задачі покращення якості в єдиній моделі, що дозволяє досягти кращих результатів за рахунок використання взаємозв'язків між різними аспектами якості зображення.

- MWCNN (Multi-level Wavelet CNN):

Ця архітектура використовує вейвлет-перетворення в поєднанні з CNN для вирішення різних задач покращення якості, включаючи зменшення шуму, видалення артефактів JPEG та суперрозділення [23].

- Deep Image Prior:

Цей підхід використовує структуру нейронної мережі як апіорну інформацію про зображення, дозволяючи вирішувати різні задачі покращення якості без попереднього навчання на великих наборах даних [24].

Багато сучасних підходів інтегрують оцінку якості безпосередньо в процес навчання, використовуючи складні функції втрат, які враховують як низькорівневі характеристики зображення (наприклад, PSNR), так і високорівневі перцептивні аспекти.

Наприклад, моделі для зменшення шуму часто враховують особливості різних типів шуму (гаусівський, імпульсний, дробовий), а методи суперрозділення враховують специфіку спотворень, пов'язаних з низькою роздільною здатністю.

Порівняно з традиційними методами, сучасні підходи на основі глибокого навчання мають ряд переваг:

1. Здатність вчитися на великих наборах даних, що дозволяє краще узагальнювати та працювати з різноманітними типами зображень та спотворень.
2. Можливість захоплювати складні нелінійні залежності між вхідними та вихідними зображеннями.
3. Здатність адаптуватися до специфічних типів спотворень та особливостей конкретних наборів даних.
4. Можливість одночасного вирішення кількох задач покращення якості.

Однак, ці методи також мають певні обмеження, такі як необхідність у великих обсягах навчальних даних, значні обчислювальні вимоги та потенційна складність інтерпретації результатів.

Розвиток сучасних методів покращення якості зображень продовжується, і ми можемо очікувати подальших інновацій у цій галузі, особливо в напрямку більш ефективних та інтерпретованих моделей, а також методів, здатних працювати з більш обмеженими наборами даних в режимі реального часу.

1.8 Постановка задачі

На основі проведеного аналізу факторів впливу на якість зображень, типів спотворень, методів оцінки якості та огляду як традиційних, так і сучасних методів покращення якості зображень, ми можемо тепер чітко сформулювати постановку задачі для нашого дослідження.

Головною метою нашого дослідження є розробка та тренування моделі генеративної змагальної мережі (GAN) для ефективного підвищення якості цифрових

зображень з одночасним збільшенням їх роздільної здатності. Ця мета відображає сучасні тенденції в області обробки зображень, де задачі покращення якості та збільшення роздільної здатності часто розглядаються як взаємопов'язані проблеми.

Обмеження та вимоги:

1. Модель повинна ефективно працювати з різними типами зображень та спотворень, характерними для реальних сценаріїв застосування.
2. Результати повинні бути візуально привабливими та реалістичними, уникаючи типових артефактів GAN-моделей.
3. Модель повинна демонструвати баланс між якістю результатів та обчислювальною ефективністю.

2. МАТЕМАТИЧНІ МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ ТА ПРОГРАМНЕ ПРОЕКТУВАННЯ

2.1 Застосування онтологічного підходу для вибору робочого класу зображень

Для ефективного розв'язання задачі підвищення якості зображень необхідно провести формалізацію та структуризацію знань, отриманих у першому розділі. Для цього застосуємо онтологічний підхід, який дозволяє створити чітку концептуальну модель предметної області через визначення основних класів понять та їх взаємозв'язків.

На основі проведеного аналізу виділяємо чотири основні класи, які формують базис нашої предметної області:

- Зображення - клас, що описує характеристики цифрових зображень, включаючи їх базові властивості та параметри якості.
- Фактори впливу - клас, що охоплює всі чинники, які можуть впливати на якість зображення протягом його життєвого циклу.
- Спотворення - клас, що описує різні типи деградації якості зображень та їх характеристики.
- Методи оцінки якості - клас, що включає різні підходи та метрики для оцінювання якості зображень.

На рисунку 2.1 ми візуалізуємо комплексну структуру якості зображень, що базується на цих базових класах. Ця структура включає чотири основні гілки: характеристики зображень, фактори впливу, типи спотворень та методи оцінки якості.

Перша гілка – характеристики зображень – відображає ключові аспекти, що визначають візуальне сприйняття та технічну якість зображення:

- Різкість, що характеризується деталізацією та чіткістю контурів
- Шум, що проявляється через зернистість та кольорові спотворення

- Динамічний діапазон, який визначає контраст та деталізацію в тінях і світлих ділянках

- Кольоропередача, що включає точність кольорів та їх насиченість
- Артефакти, включаючи аберації, муар та ореоли
- Роздільна здатність, як просторова, так і часова для відео

Друга гілка – фактори впливу – систематизує причини виникнення різних проблем з якістю зображень:

- Апаратні фактори, пов'язані з фізичними компонентами системи
- Програмні фактори, що включають алгоритмічну обробку
- Зовнішні фактори, що відображають умови зйомки

Третя гілка – типи спотворень – класифікує можливі дефекти зображень:

- Структурні спотворення, включаючи різні види розмиття та геометричні спотворення
- Неструктурні спотворення, такі як шум різних типів та артефакти компресії

Четверта гілка – методи оцінки якості – відображає способи вимірювання та оцінки якості зображень:

- Суб'єктивні методи, засновані на людському сприйнятті
- Об'єктивні методи, що використовують математичні метрики
- Спеціалізовані методи для конкретних типів спотворень
- Методи на основі глибокого навчання

Візуалізація, зображена на рисунку 2.1 забезпечує системний підхід до вибору робочого класу зображень для нашого дослідження. Через аналіз взаємозв'язків між різними компонентами структури можна не лише визначити критичні характеристики зображень для конкретних застосувань, а й виявити найбільш поширені типи спотворень. Це дозволяє ефективніше оцінювати вплив окремих факторів на якість зображень, а також обирати відповідні методи покращення та оцінки якості.

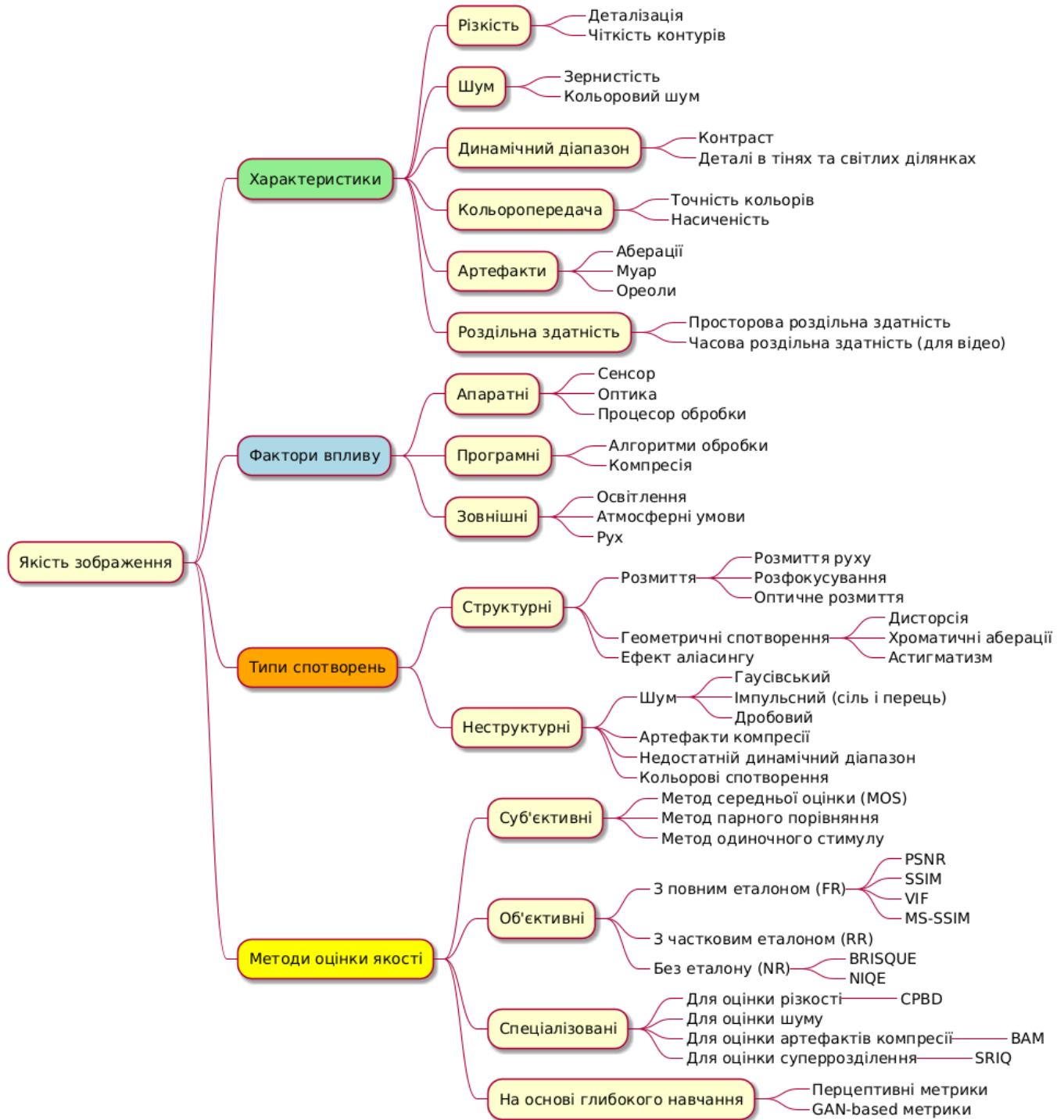


Рисунок 2.1 – Візуалізація онтології якості зображень

Далі після створення онтології класів ми можемо сформуванати більш чіткі вимоги до зображень, що будуть нами використовуватись при навчанні GAN моделі.

Для кращого розуміння та структуризації цих вимог було створено схему у вигляді mind map, яка візуалізує три основні гілки вимог до набору даних для тренування:

У центрі схеми знаходиться головний елемент "Вибір набору даних для тренування", який розгалужується на:

1. "Вимоги до якості" (зелена гілка)
2. "Вимоги до різноманітності" (синя гілка)
3. "Технічні вимоги" (рожева гілка)

Кожен з цих елементів має свої підкатегорії, що деталізують конкретні вимоги та характеристики. Така візуалізація допомагає краще структурувати та зрозуміти всі аспекти, які необхідно враховувати при підборі датасету для навчання GAN моделі.

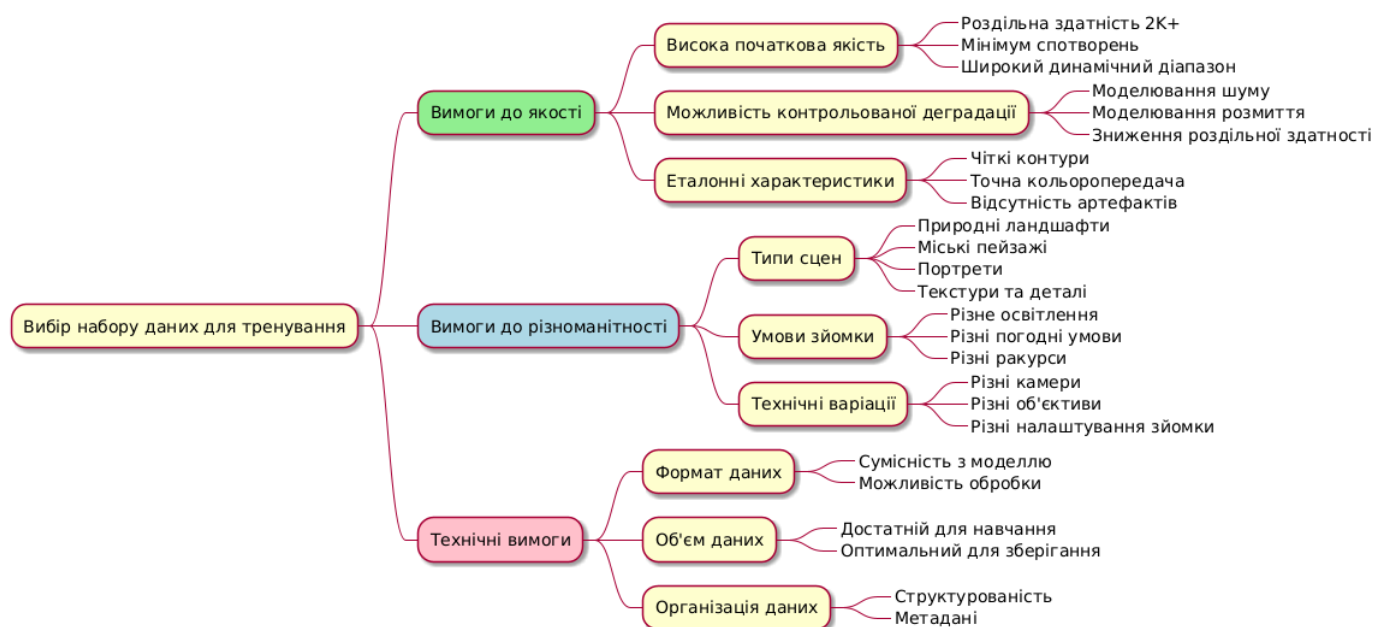


Рисунок 2.2 – Основні фактори для вибору даних

Керуючись створеними вимогами, був обраний набір даних DIVERse 2K resolution high quality images, який відповідає всім цим вимогам. Цей набір даних:

- Забезпечує високоякісні еталонні зображення з роздільною здатністю

- Містить різноманітні сцени, що дозволяє моделі вивчити широкий спектр візуальних паттернів
- Включає зображення з різними умовами освітлення та складністю текстур
- Дозволяє створювати синтетичні спотворення для тренування моделі
- Забезпечує можливість оцінки результатів за допомогою всіх типів метрик, представлених в онтології

Розроблена онтологія не тільки систематизує знання про якість зображень, але й забезпечує нам структурований підхід до вибору даних для тренування моделі, що є критично важливим для досягнення поставлених цілей дослідження.

2.2 Моделі представлення зображень

На основі проведеного в попередньому розділі онтологічного аналізу та вибору набору даних DIVerse 2K resolution high quality images, як основи для подальшого тренування моделі, необхідно детально проаналізувати характерні спотворення, які виникають у зображеннях високої роздільної здатності та потребують корекції.



Рисунок 2.3 – Схема взаємозв'язків між типами спотворень та їх джерелами для зображень високої роздільної здатності

Спираючись на розроблену онтологію, можемо визначити, що для зображень високої роздільної здатності характерні специфічні типи спотворень, які виникають як на етапі зйомки, так і під час подальшої обробки та зберігання. Розуміння природи цих спотворень критично важливе для розробки ефективної моделі покращення якості зображень.

Серед структурних спотворень для обраного класу зображень найбільш критичними є різні види розмиття. Розмиття руху особливо помітне при зйомці з рук у умовах недостатнього освітлення, коли потрібна довша витримка. Для зображень високої роздільної здатності це спотворення проявляється більш виразно через більшу кількість деталей, які можуть бути втрачені. Розфокусування також становить серйозну проблему, особливо при зйомці з малою глибиною різкості, що часто використовується для створення художнього ефекту. Оптичне розмиття, спричинене недосконалістю об'єктивів, може проявлятися по краях кадру навіть у високоякісній оптики.

Геометричні спотворення, такі як дисторсія та хроматичні аберації, також характерні для зображень високої роздільної здатності. Дисторсія проявляється у викривленні прямих ліній, особливо помітному на широкотних об'єктивах. При подушкоподібній дисторсії лінії вигинаються назовні від центру, а при бочкоподібній - навпаки, до центру. Ці спотворення можуть суттєво впливати на геометричну точність зображення та потребують математичної корекції. Хроматичні аберації виникають через різне заломлення світла різних довжин хвиль та проявляються у вигляді кольорових контурів навколо об'єктів, особливо на високонтрастних межах. Поздовжні хроматичні аберації призводять до різного фокусування різних кольорів, а поперечні - до появи кольорових облямівок на контрастних краях об'єктів. Ці артефакти особливо помітні при використанні недорогої оптики та на краях кадру.

Неструктурні спотворення також мають свої особливості для обраного класу зображень. Шум, як один з найпоширеніших типів спотворень, може бути різної

природи. Гаусівський шум часто виникає через електронні компоненти камери та особливо помітний при високих значеннях ISO. Його характерною особливістю є нормальний розподіл інтенсивності, що робить його відносно передбачуваним для алгоритмів шумопоглинання. Імпульсний шум (сілі і перець) може виникати через дефекти сенсора або помилки передачі даних. Він характеризується випадковою появою абсолютно білих або чорних пікселів, що суттєво погіршує візуальне сприйняття зображення. Дробовий шум, пов'язаний з квантовою природою світла, особливо проявляється в темних ділянках зображення. Його інтенсивність пропорційна квадратному кореню з інтенсивності сигналу, що робить його особливо помітним при тривалих експозиціях та високій чутливості сенсора. Крім того, при обробці RAW-файлів можуть виникати додаткові артефакти, пов'язані з демозаїкою та кольоровою корекцією [25].

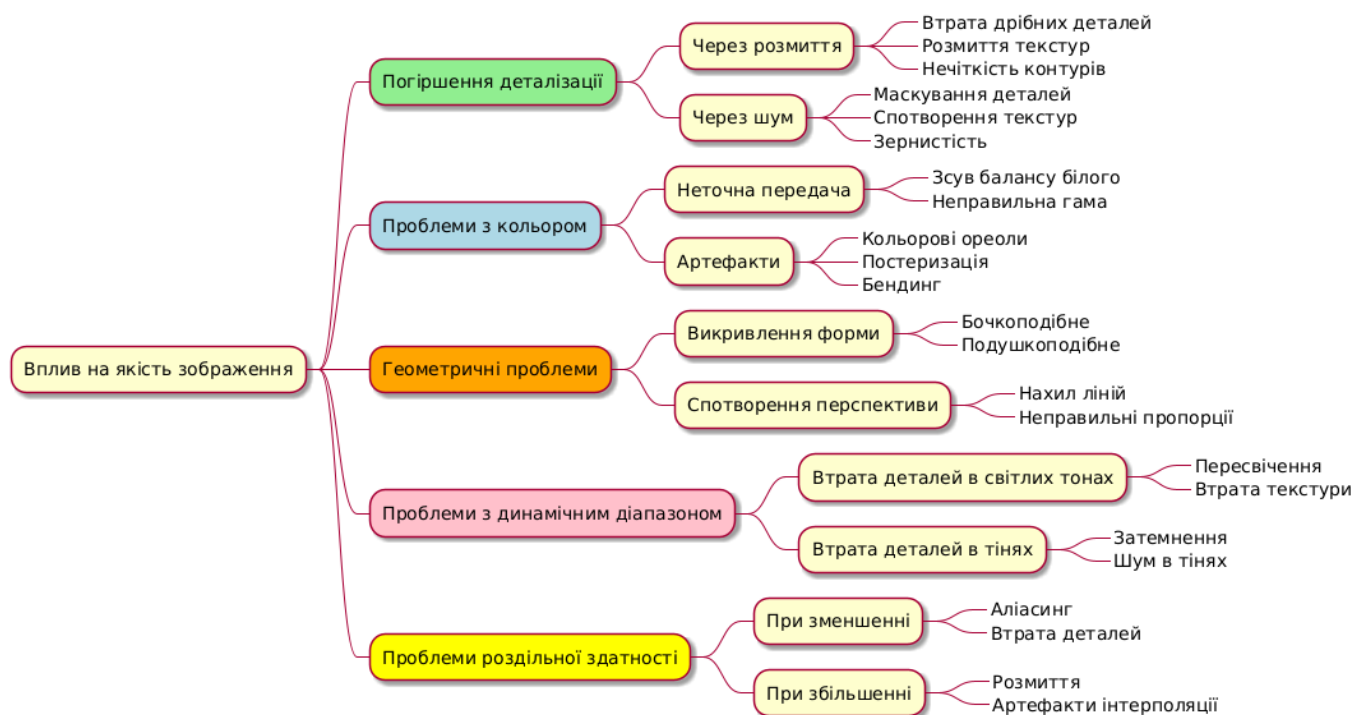


Рисунок 2.4 – Класифікація та взаємозв'язок спотворень за їх впливом на якість зображення

Артефакти компресії становлять особливу проблему для зображень високої роздільної здатності через великий обсяг даних. При стисненні можуть виникати блочні артефакти, особливо помітні на градієнтних переходах та в областях з дрібними деталями. Крім того, агресивна компресія може призводити до втрати тонких текстур та кольорових спотворень.

Проблеми з динамічним діапазоном також характерні для обраного класу зображень. Високодетальні сцени часто містять як дуже яскраві, так і темні ділянки, що може призводити до втрати деталей через обмеження динамічного діапазону сенсора. Це особливо помітно в умовах складного освітлення, наприклад, при зйомці проти світла або в приміщеннях з контрастним освітленням.

Кольорові спотворення можуть проявлятися у вигляді неправильного балансу білого, недостатньої або надмірної насиченості кольорів. Для зображень високої роздільної здатності точність кольоропередачі особливо важлива, оскільки навіть незначні відхилення можуть бути помітними через високу деталізацію.

Особливу увагу слід приділити спотворенням, які виникають при зміні роздільної здатності зображень. При зменшенні роздільної здатності можуть виникати проблеми з аліасингом, особливо на дрібних повторюваних структурах та похилих лініях. При збільшенні роздільної здатності виникає проблема відновлення втрачених деталей та текстур, яка потребує особливого підходу [26].

Розуміння взаємозв'язків між різними типами спотворень також важливе. Наприклад, наявність шуму може маскувати дрібні деталі, а спроби його усунення можуть призводити до додаткового розмиття. Компресія може посилювати існуючі спотворення та створювати нові артефакти. Тому при розробці моделі необхідно враховувати ці взаємодії для досягнення оптимального результату.

Аналіз характерних спотворень для обраного класу зображень безпосередньо впливає на подальше проектування архітектури генеративної моделі та вибір функцій

втратах. Це також визначає подальшу стратегію попередньої обробки зображень та вибір метрик для оцінки якості результатів.

2.3 Вибір та обґрунтування методів оцінки якості зображень

На основі проведеного в попередніх розділах онтологічного аналізу та визначення характерних спотворень для високороздільних зображень, необхідно обрати та обґрунтувати методи оцінки якості результатів роботи нашої моделі. Враховуючи специфіку задачі підвищення якості зображень з одночасним збільшенням роздільної здатності, особливо важливим є вибір методів, які дозволять комплексно оцінити як загальну візуальну якість результатів, так і точність відновлення дрібних деталей.

Серед широкого спектру методів оцінки якості зображень, розглянутих в онтології, для нашої задачі найбільш доцільним є використання комбінації двох взаємодоповнюючих метрик: PSNR (Peak Signal-to-Noise Ratio) та SSIM (Structural Similarity Index Measure). Цей вибір обґрунтований їх взаємодоповнюючими властивостями та широким визнанням у науковій спільноті [27].

PSNR є традиційною метрикою, яка оцінює якість зображення на основі середньоквадратичної помилки між оригінальним та обробленим зображеннями. Математично PSNR визначається через логарифмічне відношення максимально можливого значення сигналу до потужності шуму, що спотворює сигнал. У контексті цифрових зображень, PSNR обчислюється за формулою 2.1:

$$PSNR = 10 \times \log_{10} \left(\frac{(L-1)^2}{MSE} \right), \quad (2.1)$$

де L - динамічний діапазон значень пікселів, який визначається бітовою глибиною зображення: $L = 2^b$, де b - кількість біт на піксель. Для стандартних 8-бітних зображень $L = 256$ (значення пікселів від 0 до 255), для 16-бітних $L = 65536$.

MSE (Mean Squared Error) обчислюється по формулі 2.2 як:

$$MSE = \left(\frac{1}{(m \times n)} \right) \times \sum (I_1(i, j) - I_2(i, j))^2, \quad (2.2)$$

де m та n - розміри зображення у пікселях, $I_1(i, j)$ та $I_2(i, j)$ - значення пікселів у координатах (i, j) для порівнюваних зображень.

Для кольорових зображень MSE обчислюється по формулі 2.3 окремо для кожного каналу кольору (R, G, B), а потім усереднюється:

$$MSE = \frac{(MSE_R + MSE_G + MSE_B)}{3} \quad (2.3)$$

Основними перевагами PSNR для нашої задачі є:

- Простота обчислення та інтерпретації результатів
- Чутливість до піксельних відмінностей, що важливо при оцінці якості відновлення деталей
- Можливість порівняння з результатами інших досліджень завдяки широкому використанню метрики
- Ефективність при оцінці впливу шуму та розмиття

Проте PSNR має певні обмеження, зокрема слабку кореляцію з людським сприйняттям якості зображення. Саме тому необхідно доповнити її метрикою SSIM, яка краще відповідає особливостям людського зору.

SSIM є більш складною метрикою, яка враховує особливості людського сприйняття через оцінку структурної схожості між зображеннями. SSIM порівнює три компоненти: яскравість, контраст та структуру. Повна формула SSIM (2.4) має вигляд:

$$SSIM(x, y) = [l(x, y)]^\alpha \times [c(x, y)]^\beta \times [s(x, y)]^\mu \quad (2.4)$$

Компонент яскравості $l(x, y)$ обчислюється обчислюється за формулою 2.5 як:

$$l(x, y) = \frac{(2\mu_x \times \mu_y + C1)}{(\mu_x^2 + \mu_y^2 + C1)}, \quad (2.5)$$

де μ_x та μ_y - середні значення яскравості порівнюваних фрагментів зображень, $C1 = (k_1 L)^2$ - константа для забезпечення стабільності. Зазвичай $k_1 = 0.01$, таким чином для 8-бітних зображень $C1 = (0.01 \times 255)^2 = 6.5025$.

Компонент контрасту $c(x, y)$ обчислюється за формулою 2.6 як:

$$c(x, y) = \frac{(2\sigma_x \times \sigma_y + C2)}{(\sigma_x^2 + \sigma_y^2 + C2)}, \quad (2.6)$$

де σ_x та σ_y - стандартні відхилення значень пікселів, $C2 = (k_2 L)^2$ - друга константа стабілізації. Зазвичай $k_2 = 0.03$, таким чином для 8-бітних зображень $C2 = (0.03 \times 255)^2 = 58.5225$.

Компонент структури $s(x, y)$ обчислюється за формулою 2.7 як:

$$s(x, y) = \frac{(\sigma_{xy} + C3)}{(\sigma_x \times \sigma_y + C3)}, \quad (2.7)$$

де σ_{xy} - коваріація між порівнюваними фрагментами, $C3 = \frac{C2}{2}$ для стандартної реалізації.

Переваги SSIM для нашої задачі:

- Висока кореляція з людським сприйняттям якості зображення
- Чутливість до структурних змін у зображенні
- Ефективність при оцінці якості відновлення текстур та контурів
- Стійкість до невеликих геометричних трансформацій

Комбінація PSNR та SSIM дозволяє отримати комплексну оцінку якості результатів роботи нашої моделі. PSNR забезпечує об'єктивну оцінку точності відновлення на піксельному рівні, особливо важливу при збільшенні роздільної здатності, тоді як SSIM оцінює збереження структурної інформації та відповідність результатів людському сприйняттю.

2.4 Аналіз та вибір архітектури GAN для вирішення поставленої задачі

На основі проведеного аналізу характерних спотворень для високороздільних зображень та вибраних методів оцінки якості, розглянемо процес еволюції архітектур GAN для задачі суперрозділення та обґрунтуємо вибір Real-ESRGAN Compact (SRVGGNet) як оптимального рішення для нашої задачі.

Розвиток архітектур для задачі суперрозділення почався з базової SRGAN, яка встановила фундаментальні принципи застосування генеративних змагальних мереж у цій області. SRGAN використовувала генератор на основі послідовності залишкових блоків, де кожен блок містив згорткові шари з batch-нормалізацією та ReLU активацією. Дискримінатор базувався на архітектурі VGG з поступовим зменшенням просторової розмірності. Хоча ця архітектура продемонструвала потенціал GAN для генерації реалістичних деталей, вона мала суттєві обмеження при роботі з реальними спотвореннями зображень.



Рисунок 2.5 – Еволюція архітектури SRGAN

Наступним значним кроком стала ESRGAN, яка вдосконалила базову архітектуру впровадженням Residual-in-Residual Dense Block (RRDB). Ці блоки забезпечили кращий потік градієнтів завдяки щільним з'єднанням між згортковими шарами, що дозволило будувати більш глибокі та ефективні мережі. Важливим удосконаленням став відносний середній дискримінатор, який оцінював реалістичність зображення не абсолютно, а відносно інших зразків, що значно покращило стабільність процесу навчання [28].

Real-ESRGAN зробила наступний крок, розширивши можливості архітектури для роботи з реальними зображеннями через впровадження реалістичної моделі деградації та вдосконаленої стратегії навчання. Проте повна версія архітектури виявилася обчислювально складною для багатьох практичних застосувань.

Real-ESRGAN Compact (SRVGGNet) представляє оптимізовану версію архітектури, яка зберігає ефективність оригінальної моделі при значно меншій обчислювальній складності.

Особливу увагу в цій архітектурі приділено блоку масштабування, який використовує механізм піксельної перестановки замість традиційної інтерполяції. Цей підхід дозволяє мережі ефективніше навчатися відновлювати дрібні деталі та текстури при збільшенні роздільної здатності зображення. Фінальний згортковий шар генератора відповідає за формування вихідного RGB зображення з покращеною якістю та підвищеною роздільною здатністю.

Real-ESRGAN Compact демонструє відмінні результати в обробці різних типів спотворень завдяки своїй багаторівневій структурі. Архітектура ефективно справляється з розмиттям різних типів, шумом, артефактами компресії та проблемами з кольоропередачею. Важливою особливістю є здатність мережі адаптуватися до різних типів вхідних даних та генерувати природні, реалістичні результати без характерних для GAN артефактів.

Процес навчання архітектури стабілізований завдяки збалансованій структурі генератора та дискримінатора, а також ефективним залишковим з'єднанням. Це дозволяє досягати стабільних результатів навіть при тривалому навчанні на різноманітних наборах даних. Архітектура також добре інтегрується з обраними метриками оцінки якості – PSNR та SSIM, що дозволяє ефективно контролювати процес навчання та оцінювати якість результатів.

Таким чином, Real-ESRGAN Compact (SRVGGNet) представляє собою оптимальну архітектуру для нашої задачі, забезпечуючи необхідний баланс між якістю результатів та обчислювальною ефективністю. Її компоненти та механізми спеціально оптимізовані для роботи з реальними зображеннями високої роздільної здатності, що повністю відповідає поставленим задачам дослідження.

2.5 Математичний опис та проектування генератора

На основі попереднього аналізу та вибору архітектури Real-ESRGAN Compact (SRVGGNet) розглянемо детальний математичний опис та проектування генератора. Генератор призначений для ефективного підвищення роздільної здатності зображень з одночасним покращенням якості, враховуючи специфіку спотворень, визначених у попередніх розділах.

Математично генератор G представляє собою складну композицію функцій, що перетворює вхідне зображення низької роздільної здатності x у зображення високої роздільної здатності y [29]. Загальне перетворення описано формулою 2.8:

$$y = G(x) = f_n \circ f_{n-1} \circ \dots \circ f_2 \circ f_1(x), \quad (2.8)$$

де \circ позначає операцію композиції функцій, тобто послідовне застосування перетворень. Кожна функція f_i представляє окреме нелінійне перетворення, яке змінює вхідний тензор специфічним чином.

Розглянемо детально послідовність перетворень:

1. Початкове перетворення (вхідний згортковий шар) обчислюємо за формулою:

$$x_1 = f_1(x) = PReLU(Conv_1(x)) = PReLU(W_1 \times x + b_1), \quad (2.9)$$

де $x \in \mathbb{R}^{3 \times H \times W}$ - вхідне зображення;

$W_1 \in \mathbb{R}^{64 \times 3 \times 9 \times 9}$ - матриця вагів;

$b_1 \in \mathbb{R}^{64}$ - вектор зміщення;

операція згортки $x_1 \in \mathbb{R}^{64 \times H \times W}$ - результат першого перетворення.

2. Залишкові блоки ($k = 2$ до 17) обчислюємо за формулою 2.10:

$$x_k = f_k(x_{(k-1)}) = x_{(k-1)} + ResBlock_k(x_{(k-1)}), \quad (2.10)$$

де $ResBlock_k(x) = Conv_{k2}(PReLU(Conv_{k1}(x))) Conv_{k1}$.

$Conv_{k2} \in \mathbb{R}^{64 \times 64 \times 3 \times 3}$ - згорткові шари;

$x_k \in \mathbb{R}^{64 \times H \times W}$ - вихід k -го залишкового блоку.

3. Перший блок підвищення роздільної здатності обчислюємо за формулою 2.11:

$$x_{18} = f_{18}(x_{17}) = PReLU(PixelShuffle_1(Conv_{18}(x_{17}))), \quad (2.11)$$

де $Conv_{18} \in \mathbb{R}^{256 \times 64 \times 3 \times 3}$ - згортковий шар;

$PixelShuffle_1: \mathbb{R}^{256 \times H \times W} \rightarrow \mathbb{R}^{64 \times 2H \times 2W}$ - операція перегрупування пікселів;

$x_{18} \in \mathbb{R}^{64 \times 2H \times 2W}$ - проміжний результат з подвоєною роздільною здатністю.

4. Другий блок підвищення роздільної здатності обчислюємо за формулою 2.12:

$$x_{19} = f_{19}(x_{18}) = PReLU(PixelShuffle_2(Conv_{19}(x_{18}))), \quad (2.12)$$

де $Conv_{19} \in \mathbb{R}^{256 \times 64 \times 3 \times 3}$ - згортковий шар;

$PixelShuffle_2: \mathbb{R}^{256 \times 2H \times 2W} \rightarrow \mathbb{R}^{64 \times 4H \times 4W}$ - операція перегрупування пікселів;

$x_{19} \in \mathbb{R}^{64 \times 4H \times 4W}$ - результат з чотирикратним збільшенням роздільної здатності.

5. Фінальне перетворення обчислюємо за формулою 2.13:

$$y = f_{20}(x_{19}) = Conv_{20}(x_{19}) = W_{20} \times x_{19} + b_{20}, \quad (2.13)$$

де $W_{20} \in \mathbb{R}^{3 \times 64 \times 9 \times 9}$ - матриця вагів фінального шару;

$b_{20} \in \mathbb{R}^3$ - вектор зміщення;

$y \in \mathbb{R}^{3 \times 4H \times 4W}$ - фінальне зображення високої роздільної здатності.

Функція активації PReLU визначається за формулою 2.14:

$$PReLU(x)_i = \max(0, x_i) + \alpha_1 \times \min(0, x_i), \quad (2.14)$$

де a_i - навчальний параметр для i -го каналу.

Операція PixelShuffle визначається за формулою 2.15:

$$PS(x)[c, h, w] = x[c \times r^2 + r \times (w \bmod r) + (h \bmod r), \lfloor h/r \rfloor, \lfloor w/r \rfloor], \quad (2.15)$$

де $r = 2$ - коефіцієнт масштабування.

Для навчання генератора використовується комбінована функція втрат 2.16, яка складається з декількох компонентів:

$$L_{total} = L_{content} + \lambda_{adv} + L_{adversarial} + \lambda_{perc} + L_{perceptual}, \quad (2.16)$$

де λ_{adv} та λ_{perc} - коефіцієнти балансування різних типів втрат.

Контентна функція втрат $L_{content}$ (формула 2.17) визначається як середньоквадратична помилка між згенерованим зображенням та цільовим зображенням високої роздільної здатності:

$$L_{content} = \frac{1}{N} \sum_{i=1}^N \|G(x_i) - y_i\|^2, \quad (2.17)$$

де N - розмір батчу x_i - i -те вхідне зображення низької роздільної здатності y_i - відповідне цільове зображення високої роздільної здатності $\|\cdot\|^2$ - квадрат L2-норми

Змагальна функція втрат $L_{adversarial}$ визначається за формулою 2.18 через дискримінатор D :

$$L_{adversarial} = -\log(D(G(x))), \quad (2.18)$$

де $D(G(x))$ - оцінка дискримінатора для згенерованого зображення.

Перцептивна функція втрат $L_{perceptual}$ обчислюється з використанням попередньо навченої VGG мережі за формулою 2.19:

$$L_{perceptual} = \sum_l w_l \times \|\varphi_l(G(x)) - \varphi_l(y)\|^2, \quad (2.19)$$

де φ_l - виходи l -го шару VGG мережі w_l - ваговий коефіцієнт для l -го шару.

Процес оптимізації генератора можна представити як формулу 2.20:

$$\theta_G = \operatorname{argmin}_{\theta_G} L_{total}(\theta_G), \quad (2.20)$$

де θ_G - параметри генератора.

Градiєнтне оновлення параметрiв вiдбувається за формулою 2.21:

$$\theta_G^{(t+1)} = \theta_G^t - \eta \nabla \theta_G L_{total}(\theta_G^t), \quad (2.21)$$

де θ_G - параметри генератора, η - коефiцiєнт швидкостi навчання $\nabla \theta_G$ - градiєнт по параметрам генератора.

Для стабiлiзацiї процесу навчання використовується адаптивний оптимiзатор Adam з параметрами [31]:

$\beta_1 = 0.9$ (коефiцiєнт для оцiнки першого моменту);

$\beta_2 = 0.999$ (коефiцiєнт для оцiнки другого моменту);

$\varepsilon = 10^{-8}$ (параметр числової стабiльностi).

Оновлення параметрiв при використаннi Adam зазначено у формулах 2.22 – 2.26:

$$m_t = \beta_1 \times m_{(t-1)} + (1 - \beta_1) \times \nabla \theta_G L_{total}(\theta_G^t) \quad (2.22)$$

$$v_t = \beta_2 \times v_{(t-1)} + (1 - \beta_2) \times \nabla \theta_G L_{total}(\theta_G^t)^2 \quad (2.23)$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.24)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.25)$$

$$\theta_G^{(t+1)} = \theta_G^t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \varepsilon}} \quad (2.26)$$

де m_t, v_t - оцiнки першого та другого моментiв градiєнта $\widehat{m}_t, \widehat{v}_t$ - скоригованi оцiнки моментiв.

Для покращення узагальнюючої здатностi моделi може також використовуватися технiка накопичення градiєнтiв [31], ми плануємо її використати, описавши її формулою 2.27:

$$L_{content} = \frac{1}{N} \sum_{i=1}^N \|G(x_i) - y_i\|^2, \quad (2.27)$$

де K - кiлькiсть крокiв накопичення градiєнтiв.

Така архітектура та схема навчання забезпечують:

1. Ефективне відновлення високочастотних деталей завдяки глибокій ієрархії ознак
2. Стабільність навчання через використання залишкових з'єднань та адаптивної оптимізації
3. Високу якість результатів завдяки комбінованій функції втрат
4. Практичну застосовність завдяки оптимізованій структурі та ефективному навчанню

2.6 Математичний опис та проектування дискримінатора

Дискримінатор відіграє критичну роль у навчанні генератора, забезпечуючи оцінку реалістичності згенерованих зображень [32].

Дискримінатор D представляє собою глибоку згорткову нейронну мережу, що відображає вхідне зображення у скалярну оцінку його реалістичності. Він виражається наступною формулою:

$$D: \mathbb{R}^{(3 \times H \times W)} \rightarrow \mathbb{R} \quad (2.28)$$

Математично дискримінатор можна представити як композицію функцій:

$$y = G(x) = f_n \circ f_{n-1} \circ \dots \circ f_2 \circ f_1(x) \quad (2.29)$$

Розглянемо детально послідовність перетворень:

1. Перший блок обробки виражено наступним чином:

$$x_1 = f_{1(x)} = \text{LeakyReLU}(\text{Conv}_1(x)), \quad (2.30)$$

де $x \in \mathbb{R}^{3 \times H \times W}$ - вхідне зображення;

$\text{Conv}_1: \mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^{64 \times H \times W}$ - згортковий шар ;

$\text{LeakyReLU}(x) = \max(0, x) + \alpha \times \min(0, x), \alpha = 0.2$.

2. Послідовність блоків з пониженням роздільної здатності ($k = 2$ до 6) виражена формулою 2.31:

$$x_k = f_{k(x_{\{k-1\}})} = \text{LeakyReLU}(\text{BatchNorm}_k(\text{Conv}_k(x_{k-1}))), \quad (2.31)$$

де $Conv_k$ зменшує просторову розмірність вдвічі та збільшує кількість каналів (формули 2.32 – 2.36):

$$k = 2: \mathbb{R}^{64 \times H \times W} \rightarrow \mathbb{R}^{64 \times \frac{H}{2} \times \frac{W}{2}} \quad (2.32)$$

$$k = 3: \mathbb{R}^{64 \times \frac{H}{2} \times \frac{W}{2}} \rightarrow \mathbb{R}^{128 \times \frac{H}{4} \times \frac{W}{4}} \quad (2.33)$$

$$k = 4: \mathbb{R}^{128 \times \frac{H}{4} \times \frac{W}{4}} \rightarrow \mathbb{R}^{128 \times \frac{H}{8} \times \frac{W}{8}} \quad (2.34)$$

$$k = 5: \mathbb{R}^{128 \times \frac{H}{8} \times \frac{W}{8}} \rightarrow \mathbb{R}^{256 \times \frac{H}{16} \times \frac{W}{16}} \quad (2.35)$$

$$k = 6: \mathbb{R}^{256 \times \frac{H}{16} \times \frac{W}{16}} \rightarrow \mathbb{R}^{256 \times \frac{H}{32} \times \frac{W}{32}} \quad (2.36)$$

3. BatchNorm визначається за формулою 2.37:

$$BN(x) = \gamma \times \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta, \quad (2.37)$$

де μ_B - середнє значення батчу σ_B^2 - дисперсія батчу γ, β - навчальні параметри $\varepsilon = 10^{-5}$ - параметр числової стабільності.

3. Глобальне усереднення визначається за формулою 2.38:

$$x_7 = f_7(x_6) = AdaptiveAvgPool(x_6), \quad (2.38)$$

де $AdaptiveAvgPool: \mathbb{R}^{256 \times \frac{H}{32} \times \frac{W}{32}} \rightarrow \mathbb{R}^{256 \times 1 \times 1}$.

4. Фінальне перетворення визначається за формулою 2.39:

$$y = f_8(x_7) = Conv_8(x_7), \quad (2.39)$$

де $Conv_8: \mathbb{R}^{256 \times 1 \times 1} \rightarrow \mathbb{R}^{1 \times 1 \times 1}$.

Функція втрат дискримінатора визначається за формулою 2.40:

$$L_D = L_{D_{real}} + L_{D_{fake}}, \quad (2.40)$$

де $L_{D_{real}} = -\mathbb{E}_y[\log(D(y))]$, а $L_{D_{fake}} = -\mathbb{E}_x[\log(1 - D(G(x)))]$;

де y - реальні зображення високої роздільної здатності;

$G(x)$ - згенеровані зображення.

Для стабілізації тренування використовується релятивістичний дискримінатор:

$$L_{Drel} = -\mathbb{E}_y[\log(\sigma(c(y) - \mathbb{E}_x[c(G(x))]))] - \mathbb{E}_x[\log(1 - \sigma(c(G(x)) - \mathbb{E}_y[c(y)]))], \quad (2.41)$$

де $C(\cdot)$ - виходи дискримінатора до сигмоїдної функції $\sigma(\cdot)$.

Процес оптимізації дискримінатора зображено формулою 2.42:

$$\theta_D = \operatorname{argmin}_{\theta_D} L_D(\theta_D) \quad (2.42)$$

Градiєнтне оновлення параметрів зображено формулою 2.43:

$$\theta_D^{t+1} = \theta_D^t - \eta \times \nabla_{\theta_D} L_D(\theta_D^t) \quad (2.43)$$

Використовується той самий оптимізатор Adam, що і для генератора, але зі відповідно зміненими індексами (формули 2.44 – 2.48):

$$m_t = \beta_1 \times m_{(t-1)} + (1 - \beta_1) \times \nabla_{\theta_D} L_{total}(\theta_D^t) \quad (2.44)$$

$$v_t = \beta_2 \times v_{(t-1)} + (1 - \beta_2) \times \nabla_{\theta_D} L_{total}(\theta_D^t)^2 \quad (2.45)$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.46)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.47)$$

$$\theta_D^{(t+1)} = \theta_D^t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \varepsilon}} \quad (2.48)$$

Для стабілізації навчання використовуються додаткові техніки:

1. Згладжування міток (label smoothing) виражаються такими формулами:

$$y_{real} = 0.9 + 0.1 \times \operatorname{random}(0,1) \quad (2.49)$$

$$y_{fake} = 0.1 \times \operatorname{random}(0,1) \quad (2.50)$$

2. Регуляризація градієнтів:

$$L_{GP} = \lambda_{GP} \times \mathbb{E}_x \left[\left(\|\nabla_{\widehat{x}D(\widehat{x})}\|_2 - 1 \right)^2 \right], \quad (2.51)$$

де $\widehat{x} = \varepsilon y + (1 - \varepsilon)G(x)$, $\varepsilon \sim U[0,1]$

Така архітектура дискримінатора забезпечує:

1. Ефективну оцінку реалістичності зображень завдяки поступовому зменшенню просторової розмірності

2. Стабільне навчання через використання BatchNorm та LeakyReLU
3. Глобальну оцінку зображення завдяки AdaptiveAvgPool
4. Баланс між якістю оцінки та обчислювальною ефективністю

2.7 Проектування активаційних функцій

Спираючись на математичний опис генератора та дискримінатора, наведений у попередніх розділах, проаналізуємо та обґрунтуємо вибір активаційних функцій для кожного компонента системи [33]. Активаційні функції відіграють критичну роль у здатності мережі вивчати нелінійні залежності та ефективно обробляти різні типи спотворень, що були нами визначені в розділі 2.2.

Для генератора основною активаційною функцією обрано Parametric ReLU (PReLU), яка математично визначається як формула 2.52:

$$PReLU(x)_i = \max(0, x_i) + \alpha_i \times \min(0, x_i) \quad (2.52)$$

де α_i - навчальний параметр для i -го каналу.

Градiєнт PReLU має вигляд:

$$\frac{\partial PReLU(x)_i}{\partial x_i} = \{ 1, \text{якщо } x_i > 0 \quad \alpha_i, \text{якщо } x_i \leq 0 \} \quad (2.53)$$

$$\frac{\partial PReLU(x)_i}{\partial x_i} = \{ 0, \text{якщо } x_i > 0 \quad x_i, \text{якщо } x_i \leq 0 \} \quad (2.54)$$

Перевага PReLU над стандартним ReLU полягає у можливості навчання оптимального нахилу для від'ємної частини функції. Це особливо важливо для генератора, оскільки дозволяє мережі краще адаптуватися до різних типів вхідних даних та спотворень. Математичне очікування виходу PReLU визначено наступною формулою :

$$\mathbb{E}[PReLU(x)_i] = \mathbb{E}[\max(0, x_i)] + \alpha_i \times \mathbb{E}[\min(0, x_i)] \quad (2.55)$$

Для дискримінатора використовується LeakyReLU з фіксованим коефіцієнтом нахилу:

$$LeakyReLU(x) = \max(0, x) + \beta \times \min(0, x) \quad (2.56)$$

де $\beta = 0.2$ - фіксований коефіцієнт нахилу.

де градієнт LeakyReLU:

$$\frac{\partial \text{LeakyReLU}(x)}{\partial x} = \{ 1, \text{якщо } x > 0 \beta, \text{якщо } x \leq 0 \} \quad (2.57)$$

Для генератора важливим аспектом є адаптивність до різних типів вхідних даних, що математично виражається через очікування градієнта по параметру нахилу:

$$\mathbb{E} \left[\frac{\partial L}{\partial \alpha_i} \right] = \mathbb{E} \left[\frac{\partial L}{\partial \text{PReLU}(x)_i} \times x_i \times \mathbb{I}(x_i \leq 0) \right] \quad (2.58)$$

Запобігання проблемі вмираючих нейронів досягається через властивість:

$$P(\text{PReLU}(x)_i = 0) < P(\text{ReLU}(x) = 0) \quad (2.59)$$

Збереження градієнтного потоку забезпечується через формулу 2.60:

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \text{PReLU}(x)_i} \times \{ 1 \text{ або } \alpha_i \} \quad (2.60)$$

Для дискримінатора стабільність навчання забезпечується через меншу варіацію градієнтів:

$$\text{var} \left(\frac{\partial \text{LeakyReLU}(x)}{\partial x} \right) < \text{var} \left(\frac{\partial \text{ReLU}(x)}{\partial x} \right) \quad (2.61)$$

Запобігання насиченню гарантується нижньою межею градієнта:

$$\left| \frac{\partial \text{LeakyReLU}(x)}{\partial x} \right| \geq \beta > 0 \quad (2.62)$$

Взаємодія активаційних функцій з батч-нормалізацією описується як :

$$\text{BN}(\text{PReLU}(x)) = \gamma \times \frac{\text{PReLU}(x) - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (2.63)$$

Взаємодія з залишковими з'єднаннями описується формулою 2.64:

$$\text{ResBlock}(x) = x + \text{PReLU} \left(\text{BN} \left(\text{Conv} \left(\text{PReLU} \left(\text{BN}(\text{Conv}(x)) \right) \right) \right) \right) \quad (2.64)$$

Вплив на функції втрат відображається через ланцюгове правило:

$$\frac{\partial L_{\text{content}}}{\partial x} = \frac{\partial L_{\text{content}}}{\partial G(x)} * \frac{\partial G(x)}{\partial \text{PReLU}(x)} * \frac{\partial \text{PReLU}(x)}{\partial x} \quad (2.65)$$

Статистичні властивості активацій для генератора виражаються через:

$$\mathbb{E}[\text{PReLU}(x)] = p \times \mu_+ + (1 - p) \times \alpha \times \mu_- \quad (2.66)$$

де p - ймовірність позитивної активації, μ_+ та μ_- - середні значення позитивних та негативних активацій відповідно.

Для дискримінатора відповідно формула 2.67:

$$\mathbb{E}[LeakyReLU(x)] = p \times \mu_+ + (1 - p) \times 0.2 \times \mu_- \quad (2.67)$$

Градiєнтний потiк через PReLU характеризується нормою:

$$\left\| \frac{\partial L}{\partial x} \right\| = \left\| \frac{\partial L}{\partial PReLU(x)} \right\| \times \sqrt{1^2 \times p + \alpha^2 \times (1 - p)} \quad (2.68)$$

Градiєнтний потiк через LeakyReLU:

$$\left\| \frac{\partial L}{\partial x} \right\| = \left\| \frac{\partial L}{\partial LeakyReLU(x)} \right\| \times \sqrt{1^2 \times p + 0.2^2 \times (1 - p)} \quad (2.69)$$

Обраний набір активаційних функцій забезпечує оптимальний баланс між здатністю мережі вивчати складні нелінійні залежності, стабільністю навчання, обчислювальною ефективністю та адаптивністю до різних типів вхідних даних та спотворень, що безпосередньо впливає на здатність мережі ефективно вирішувати поставлену задачу підвищення якості зображень.

2.8 Проектування функцій втрат

На основі проведеного аналізу архітектури та вибору активаційних функцій, розглянемо детальне проектування функцій втрат для нашої системи. Функції втрат відіграють ключову роль у навчанні моделі, забезпечуючи ефективне відновлення якості зображень та врахування всіх типів спотворень, визначених нами у розділі 2.2.

Загальна функція втрат для генератора представляє собою зважену суму трьох компонентів (формула 2.70):

$$L_G = \lambda_{content} \times L_{content} + \lambda_{adv} \times L_{adversarial} + \lambda_{perceptual} \times L_{perceptual} \quad (2.70)$$

де $\lambda_{content} = 1.0$, $\lambda_{adv} = 0.001$, $\lambda_{perceptual} = 0.1$ - коефіцієнти балансування різних компонентів втрат.

Контентна функція втрат $L_{content}$ базується на середньоквадратичній помилці (MSE) між згенерованим та цільовим зображеннями [34]:

$$L_{content} = \frac{1}{HWC} \times \sum_{h=1}^H \sum_{w=1}^W \sum_{c=1}^C (G(x)_{hwc} - y_{hwc})^2 \quad (2.71)$$

де H , W - просторові розміри зображення, C - кількість каналів, $G(x)$ - вихід генератора, y - цільове зображення високої роздільної здатності.

Змагальна функція втрат $L_{adversarial}$ використовує релятивістичний підхід:

$$L_{adversarial} = -\mathbb{E}_x[\log(\sigma(D(G(x)) - \mathbb{E}_{y[D(y)]})))] \quad (2.72)$$

де σ - сигмоїдна функція, D - дискримінатор, \mathbb{E}_x , \mathbb{E}_y - математичні очікування по вхідним та цільовим зображенням відповідно.

Для обчислення перцептивної функції втрат $L_{perceptual}$ використовуються проміжні виходи попередньо навченої мережі VGG19:

$$L_{perceptual} = \sum_i w_i \times \left\| \varphi_i(G(x)) - \varphi_i(y) \right\|^1 \quad (2.73)$$

де φ_i - виходи i -го шару VGG мережі, w_i - вагові коефіцієнти для різних шарів, $\| \cdot \|^1$ - L1 норма.

Функція втрат дискримінатора має симетричну форму:

$$L_D = -\mathbb{E}_y[\log(\sigma(D(y) - \mathbb{E}_{x[D(G(x))})))] - \mathbb{E}_x[\log(1 - \sigma(D(G(x)) - \mathbb{E}_{y[D(y)]})))] \quad (2.74)$$

Для стабілізації навчання використовується техніка регуляризації градієнтів:

$$L_{GP} = \lambda_{GP} \times \mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}D(\hat{x})}\|^2 - 1)^2] \quad (2.75)$$

де $\hat{x} = \varepsilon * y + (1 - \varepsilon) * G(x)$, $\varepsilon \sim U[0,1]$ $\lambda_{GP} = 10.0$ - коефіцієнт регуляризації

Повна функція втрат дискримінатора:

$$L_{D_{total}} = L_D + L_{GP} \quad (2.76)$$

Градієнти цих функцій втрат відносно параметрів моделей:

$$\frac{\partial L_G}{\partial \theta_G} = \frac{\partial L_{content}}{\partial \theta_G} + \lambda_{adv} \times \frac{\partial L_{adversarial}}{\partial \theta_G} + \lambda_{perceptual} \times \frac{\partial L_{perceptual}}{\partial \theta_G} \quad (2.77)$$

$$\frac{\partial L_{D_{total}}}{\partial \theta_D} = \frac{\partial L_D}{\partial \theta_D} + \lambda_{GP} \times \frac{\partial L_{GP}}{\partial \theta_D} \quad (2.78)$$

де θ_G , θ_D - параметри генератора та дискримінатора відповідно.

Оновлення параметрів відбувається за допомогою оптимізатора Adam:

$$\theta_G^{t+1} = \theta_G^t - \eta_G \times \frac{\hat{m}_G}{\sqrt{\hat{v}_G + \varepsilon}} \theta_D^{t+1} = \theta_D^t - \eta_D \times \frac{\hat{m}_D}{\sqrt{\hat{v}_D + \varepsilon}} \quad (2.79)$$

де $\eta_G = \eta_D = 0.0002$ - швидкості навчання, \hat{m}_G, \hat{m}_D - скориговані перші моменти градієнтів, \hat{v}_G, \hat{v}_D - скориговані другі моменти градієнтів.

Для стабілізації процесу навчання використовується накопичення градієнтів:

$$\nabla \theta_{accumulated} = \frac{1}{K} \times \sum_{k=1}^K \nabla \theta_k \quad (2.80)$$

де $K = 4$ - кількість кроків накопичення.

Зв'язок функцій втрат з метриками якості:

$$\text{PSNR} = -10 \times \log^{10}(L_{\text{content}}) \quad (2.81)$$

SSIM корелює з $L_{\text{perceptual}}$ через врахування структурної інформації.

Такий комплексний підхід до проектування функцій втрат забезпечує:

- Ефективне відновлення високочастотних деталей зображення
- Реалістичність згенерованих зображень
- Стабільність процесу навчання
- Збалансоване покращення як об'єктивних, так і суб'єктивних показників якості

В контексті нашої архітектури на основі Real-ESRGAN Compact (SRVGGNet), запропонована система функцій втрат забезпечує ефективне навчання моделі для вирішення поставленої задачі підвищення якості зображень з одночасним збільшенням роздільної здатності, враховуючи всі визначені вимоги та обмеження.

3. ПРОЕКТУВАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Аналіз технічних вимог та обмежень системи

На основі теоретичного проектування, виконаного в попередньому розділі, необхідно визначити технічні вимоги та обмеження для практичної реалізації системи підвищення якості зображень з використанням архітектури Real-ESRGAN Compact (SRVGGNet).

Апаратні вимоги для розробки та навчання моделі визначаються складністю обраної архітектури та розміром набору даних DIV2K. Система потребує графічного прискорювача з підтримкою CUDA для ефективного навчання нейронної мережі. Мінімальні вимоги до відеопам'яті складають 8 ГБ, що обумовлено розміром батчу та проміжними обчисленнями під час навчання. Оптимальна конфігурація включає відеокарту з 12-16 ГБ відеопам'яті для можливості збільшення розміру батчу та прискорення процесу навчання.

Програмне забезпечення повинно базуватися на фреймворку PyTorch версії не нижче 1.8.0, що забезпечує необхідну функціональність для реалізації глибоких нейронних мереж. Вимоги до системної пам'яті становлять не менше 32 ГБ для ефективного обробки та зберігання проміжних результатів. Операційна система повинна підтримувати актуальні версії CUDA toolkit та cuDNN для оптимальної роботи з GPU [35].

Обмеження на розмір вхідних зображень визначаються доступною відеопам'яттю та архітектурою мережі. При тренуванні оптимальний розмір вхідного зображення складає 64x64 пікселі для зображень низької роздільної здатності, що після обробки генератором збільшується до 256x256 пікселів. Це обмеження пов'язане з необхідністю зберігання проміжних активацій та градієнтів під час навчання.

Час обробки одного зображення при інференсі повинен не перевищувати 1 секунду на цільовому обладнанні, що вимагає оптимізації моделі та використання технік прискорення обчислень, таких як квантизація моделі та батчингу при обробці множини зображень.

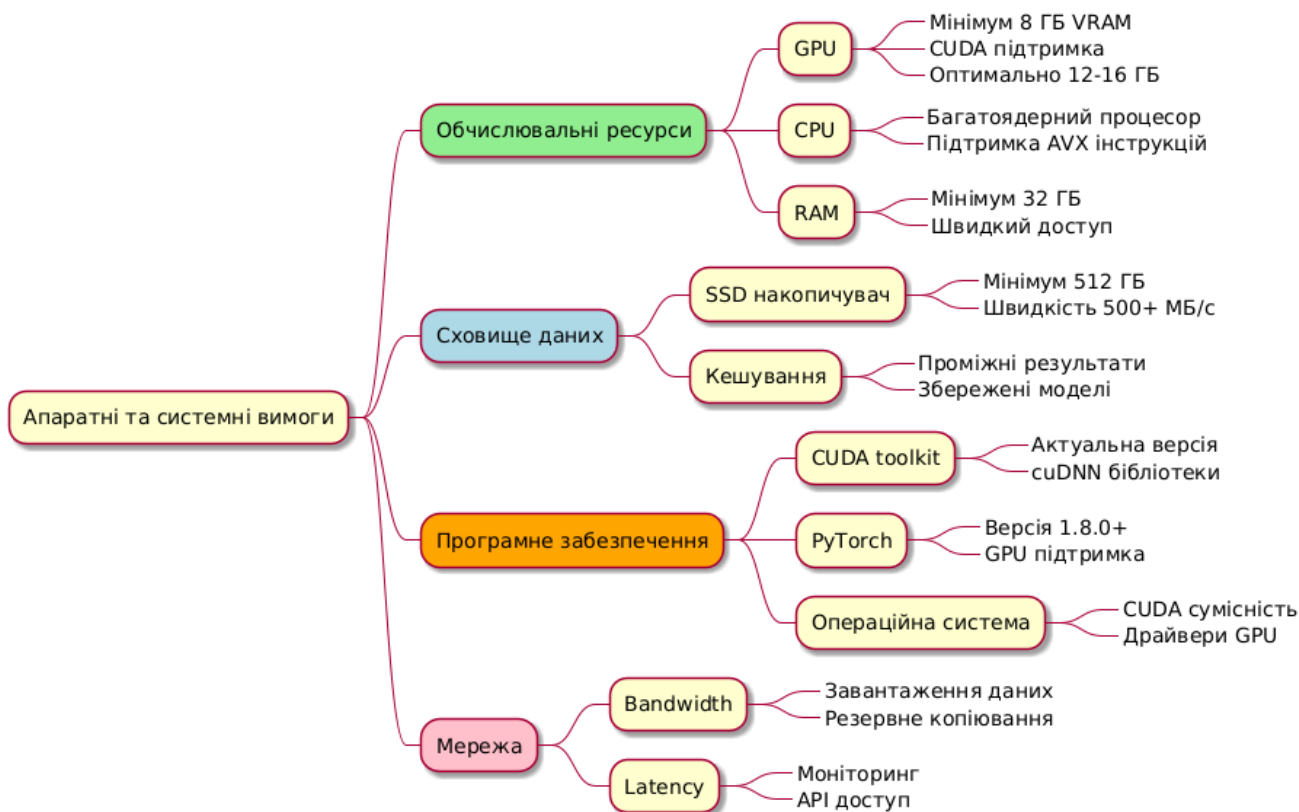


Рисунок 3.1 – Візуалізація апаратних та програмних вимог

Вимоги до системи зберігання даних включають необхідність швидкого доступу до навчального набору даних DIV2K. Рекомендується використання SSD накопичувача об'ємом не менше 512 ГБ для зберігання набору даних, проміжних результатів та збережених моделей. Швидкість читання/запису повинна бути не менше 500 МБ/с для мінімізації затримок при завантаженні даних.

Програмна реалізація повинна забезпечувати можливість моніторингу процесу навчання з використанням метрик PSNR та SSIM у реальному часі. Необхідно

передбачити механізми збереження проміжних результатів навчання та можливість відновлення процесу навчання після переривання.

Система повинна підтримувати масштабування для обробки різних розмірів зображень при інференсі, зберігаючи при цьому якість результатів. Це вимагає реалізації адаптивного механізму розбиття великих зображень на фрагменти та їх подальшого об'єднання.

Вимоги до точності обчислень визначають необхідність використання змішаної точності (mixed precision) при навчанні для оптимізації використання пам'яті та прискорення обчислень без значної втрати якості результатів. При інференсі можливе використання зниженої точності (FP16) для прискорення обробки [36].

Враховуючи специфіку задачі та вибрану архітектуру, система повинна забезпечувати можливість паралельної обробки даних з використанням багатопоточності для попередньої обробки зображень та оптимізації завантаження обчислювальних ресурсів.

Вимоги до інтерфейсу включають необхідність реалізації програмного API для інтеграції з іншими системами та можливість пакетної обробки зображень. Система повинна надавати інформативні повідомлення про помилки та детальну інформацію про процес обробки.

Безпека системи повинна забезпечуватися через механізми валідації вхідних даних, захист від переповнення буферів при обробці зображень великих розмірів та коректну обробку виняткових ситуацій.

Визначені технічні вимоги та обмеження формують основу для подальшої розробки програмного рішення та визначають напрямки оптимізації системи для досягнення поставлених цілей у підвищенні якості зображень.

3.2 Розробка загальної архітектури програмного рішення

На основі визначених у попередньому розділі технічних вимог та обмежень необхідно розробити загальну архітектуру програмного рішення для системи підвищення якості зображень. Архітектура повинна базуватися на використанні фреймворку PyTorch та забезпечувати ефективну імплементацію теоретичної моделі Real-ESRGAN Compact (SRVGGNet), розробленої в розділі 2.

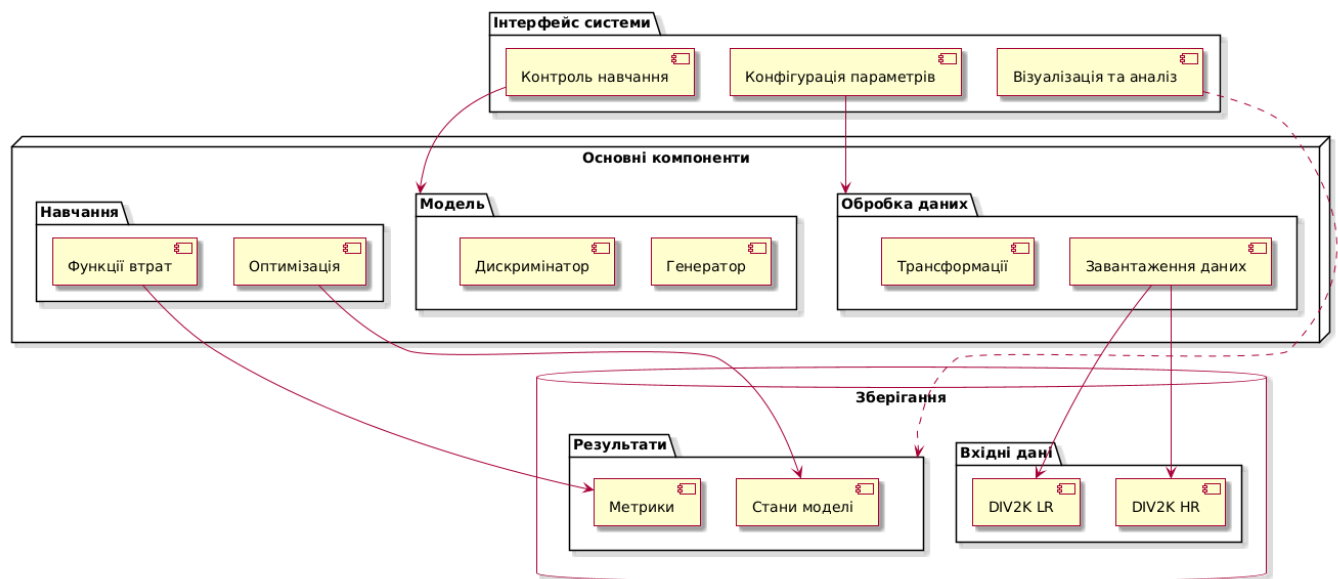


Рисунок 3.2 – Високорівнева архітектура системи підвищення якості зображень

В структурі програмного рішення необхідно реалізувати наступні основні модулі:

Модуль управління даними повинен забезпечувати роботу з набором даних DIV2K через спеціалізований клас, що наслідуватиме `torch.utils.data.Dataset`. В цьому модулі необхідно імплементувати функціональність для завантаження зображень високої роздільної здатності (HR) та відповідних їм зображень низької роздільної здатності (LR). Система попередньої обробки даних повинна включати операції нормалізації, приведення розмірів та аугментації зображень. Для оптимізації навчання

потрібно реалізувати механізм розділення даних на тренувальну та валідаційну вибірки з використанням `random_split` та забезпечити ефективне батчування даних через `DataLoader` з підтримкою багатопоточності.

Модуль нейронної мережі має складатися з двох основних компонентів. Генератор необхідно реалізувати як клас на основі `nn.Module` з використанням залишкових блоків (`ResidualBlock`) та механізму підвищення роздільної здатності через `PixelShuffle`. Архітектура генератора повинна підтримувати масштабування 4x та включати `BatchNormalization` для стабілізації навчання. Дискримінатор потрібно реалізувати як згорткову мережу з поетапним зменшенням просторової розмірності та використанням `LeakyReLU` для активації. Обидва компоненти повинні підтримувати режим змішаної точності для оптимізації використання пам'яті.

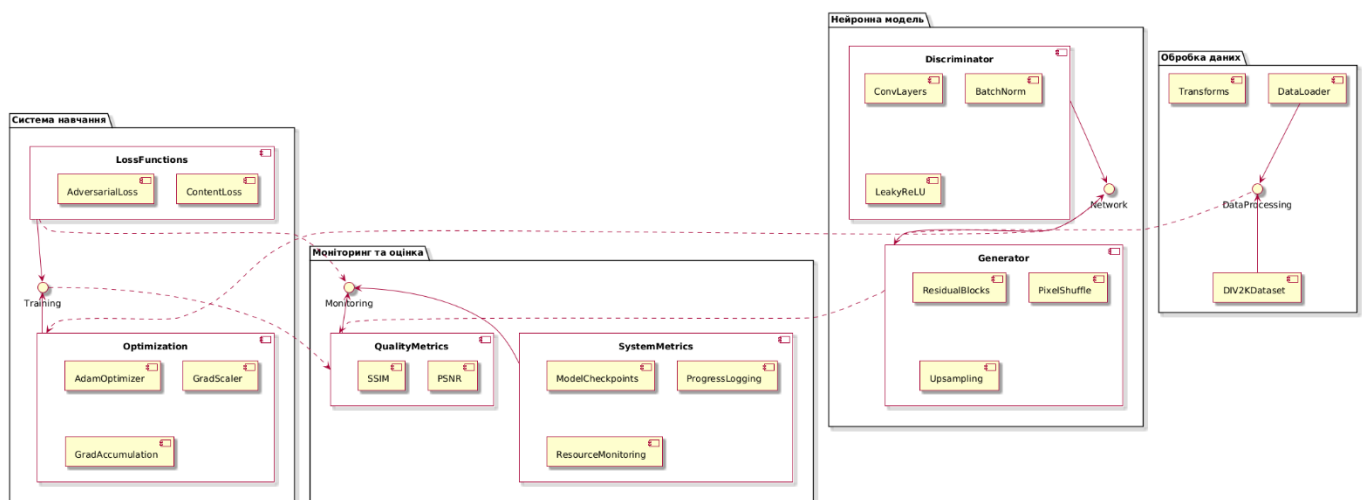


Рисунок 3.3 – Архітектура компонентів системи

Система оптимізації та навчання повинна використовувати оптимізатор Adam з налаштовуваними параметрами швидкості навчання та моментів. Необхідно імплементувати механізм автоматичного масштабування градієнтів через `GradScaler` для підтримки змішаної точності обчислень. Для стабілізації процесу навчання

потрібно реалізувати механізм накопичення градієнтів з конфігурованою кількістю кроків акумуляції.

У системі моніторингу та оцінки необхідно імплементувати функції для обчислення метрик PSNR та SSIM. Розрахунок SSIM повинен враховувати багатоканальну природу зображень та використовувати ковзне вікно для локальних обчислень. Система повинна підтримувати періодичне збереження стану моделі та генерацію прикладів результатів для візуальної оцінки прогресу навчання.

Модуль конфігурації має забезпечувати централізоване управління наступними параметрами:

- Розміри вхідних та вихідних зображень
- Розмір батчу та кількість епох навчання
- Параметри оптимізаторів (швидкість навчання, коефіцієнти бета)
- Ваги різних компонентів функції втрат
- Періодичність збереження та оцінки результатів

Необхідно реалізувати систему обробки помилок, яка забезпечить:

- Валідацію вхідних даних та параметрів
- Обробку виключних ситуацій при завантаженні та трансформації зображень
- Відновлення після збоїв під час навчання
- Логування помилок та діагностичної інформації

Для оптимізації використання обчислювальних ресурсів потрібно імплементувати:

- Автоматичне визначення доступності та налаштування GPU
- Механізми очищення кешу GPU та управління пам'яттю
- Підтримку змішаної точності обчислень (FP16/FP32)
- Оптимізацію завантаження даних через prefetch та pinned memory

Взаємодія між модулями повинна бути організована через чітко визначені інтерфейси з мінімальною зв'язністю компонентів. Необхідно передбачити

можливість заміни окремих компонентів без впливу на інші частини системи. Архітектура повинна підтримувати подальше розширення функціональності та адаптацію до різних сценаріїв використання.

Така архітектура програмного рішення забезпечить ефективну реалізацію теоретичної моделі та відповідатиме всім визначеним технічним вимогам та обмеженням системи, дозволяючи досягти поставлених цілей у підвищенні якості зображень.

3.3 Проектування системи попередньої обробки та підготовки даних

Спираючись на розроблену архітектуру програмного рішення та визначені технічні вимоги, необхідно спроектувати ефективну систему попередньої обробки та підготовки даних для навчання моделі підвищення якості зображень. Ця система повинна забезпечити оптимальну підготовку набору даних DIV2K для тренування нейронної мережі.

Центральним елементом системи попередньої обробки даних має стати спеціалізований клас `DIV2KDataset`, який успадковуватиме функціональність від базового класу `torch.utils.data.Dataset`. В рамках цього класу необхідно реалізувати всі базові методи для роботи з даними, включаючи `init`, `len` та `getitem`.

У конструкторі класу `DIV2KDataset` потрібно реалізувати ініціалізацію шляхів до директорій з зображеннями високої (HR) та низької (LR) роздільної здатності. При цьому необхідно передбачити валідацію існування вказаних директорій та перевірку відповідності структури файлів. Система повинна підтримувати коефіцієнт масштабування як параметр, що за замовчуванням дорівнює 4, відповідно до вимог архітектури `Real-ESRGAN Compact`.

Метод `len` має повертати загальну кількість пар зображень у наборі даних, що необхідно для коректної роботи `DataLoader` та планування епох навчання. Цей метод

повинен враховувати лише валідні пари зображень, для яких існують обидва файли (HR та LR версії).

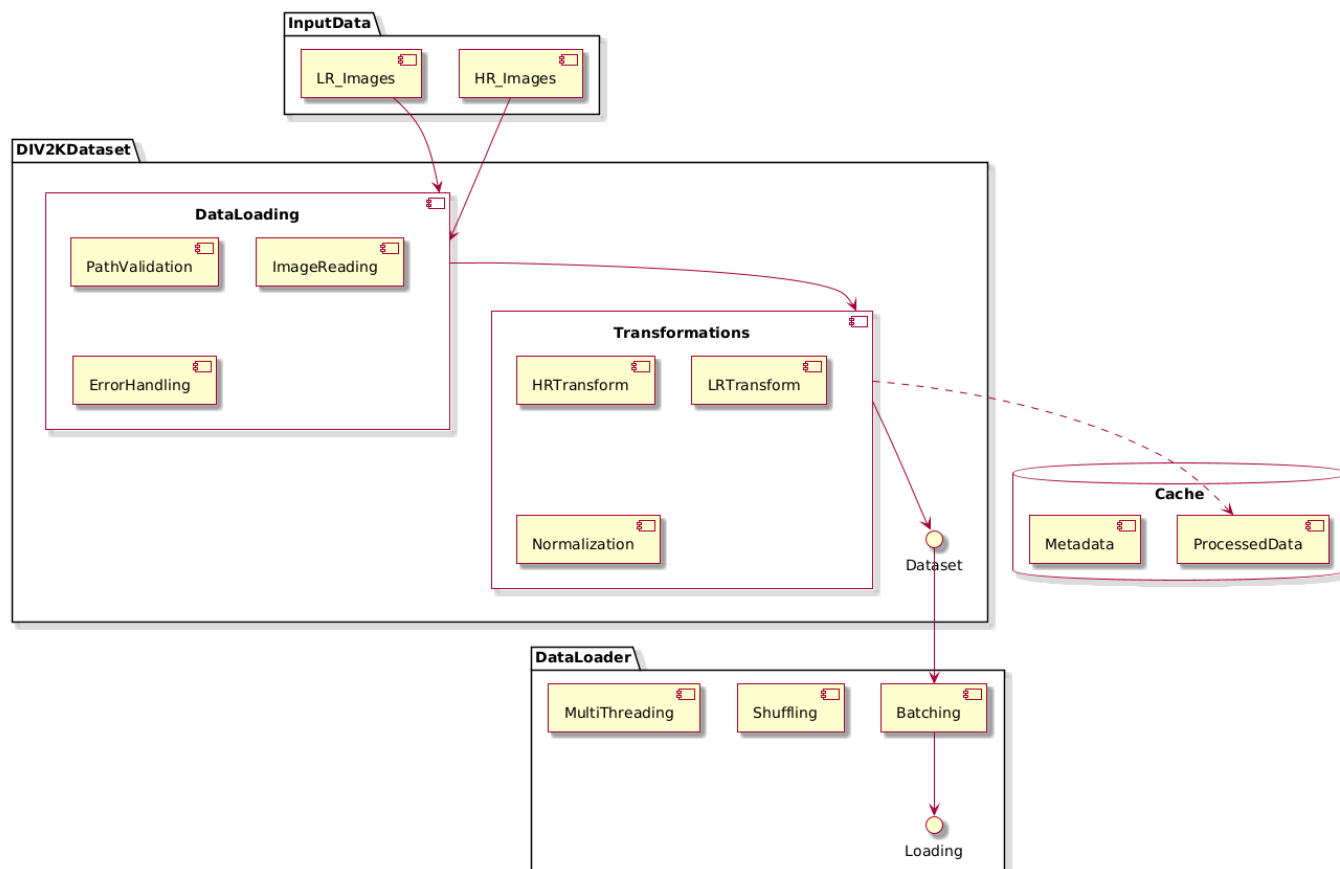


Рисунок 3.4 – Архітектура системи підготовки даних з відображенням потоків даних та трансформацій

В методі `getitem` необхідно реалізувати логіку завантаження та попередньої обробки окремих пар зображень. Цей метод повинен включати обробку помилок при читанні файлів та перетворення зображень. При завантаженні зображень потрібно використовувати бібліотеку `PIL` для забезпечення сумісності з `PyTorch` та ефективної обробки різних форматів зображень.

Система трансформацій зображень повинна включати наступні етапи обробки:

Для HR зображень необхідно реалізувати масштабування до цільового розміру 256x256 пікселів з використанням бікубічної інтерполяції для збереження якості.

Конвертацію кольорового простору в RGB необхідно виконувати для забезпечення уніфікованого формату вхідних даних. Нормалізацію значень пікселів потрібно проводити в діапазоні $[-1, 1]$ з використанням середніх значень $[0.5, 0.5, 0.5]$ та стандартних відхилень $[0.5, 0.5, 0.5]$.

Для LR зображень аналогічно потрібно виконати масштабування до розміру 64×64 пікселі, що відповідає коефіцієнту масштабування 4x. Конвертація кольорового простору та нормалізація мають відповідати обробці HR зображень для забезпечення консистентності даних.

Всі операції трансформації необхідно реалізувати з використанням `torchvision.transforms.Compose` для створення послідовності перетворень. При цьому важливо забезпечити детермінованість трансформацій для відповідних пар HR та LR зображень.

Для оптимізації процесу завантаження даних необхідно реалізувати механізм кешування трансформованих зображень у оперативній пам'яті. При цьому потрібно передбачити можливість очищення кешу при досягненні лімітів пам'яті. Також необхідно реалізувати підтримку багатопоточного завантаження даних через параметр `num_workers` у `DataLoader`.

Система повинна включати функціональність для розділення набору даних на тренувальну та валідаційну вибірки з використанням `random_split`. При цьому необхідно забезпечити збалансованість розділення та можливість відтворення результатів через фіксацію `seed` генератора випадкових чисел.

Для обробки помилок при завантаженні та трансформації зображень необхідно реалізувати спеціальну функцію `collate_fn`, яка буде фільтрувати невалідні елементи з батчу та забезпечувати стабільність процесу навчання при виникненні помилок обробки окремих зображень.

Система повинна підтримувати конфігурацію параметрів трансформації через зовнішній конфігураційний файл або словник параметрів, що дозволить гнучко налаштувати процес попередньої обробки даних без модифікації коду.

Для оптимізації використання пам'яті необхідно реалізувати підтримку `pin_memory` при роботі з GPU та префетчингу даних для зменшення латентності при завантаженні. Також потрібно передбачити можливість очищення неактуальних даних з пам'яті для запобігання її переповненню.

Така система попередньої обробки та підготовки даних забезпечить ефективне навчання моделі підвищення якості зображень, враховуючи всі визначені вимоги та обмеження. При цьому система залишається гнучкою для подальших модифікацій та оптимізацій.

3.4 Розробка алгоритму навчання моделі

На основі спроектованої архітектури системи та розробленої системи попередньої обробки даних, необхідно створити ефективний алгоритм навчання моделі. Алгоритм повинен забезпечувати стабільне навчання генеративної змагальної мережі з урахуванням всіх особливостей архітектури Real-ESRGAN Compact (SRVGGNet).

Процес навчання необхідно організувати на основі epoch-based підходу з загальною кількістю епох, що дорівнює 200. Кожна епоха повинна включати повний прохід по тренувальному набору даних з використанням розробленого `DataLoader` з розміром батчу 4, що обумовлено обмеженнями доступної відеопам'яті та необхідністю стабільного навчання.

В рамках кожної ітерації навчання необхідно реалізувати почергове оновлення параметрів дискримінатора та генератора. Дискримінатор повинен оновлюватися два рази на кожну ітерацію для забезпечення балансу в навчанні GAN. При цьому важливо

використовувати механізм накопичення градієнтів з параметром `accumulation_steps` рівним 4 для стабілізації процесу навчання.

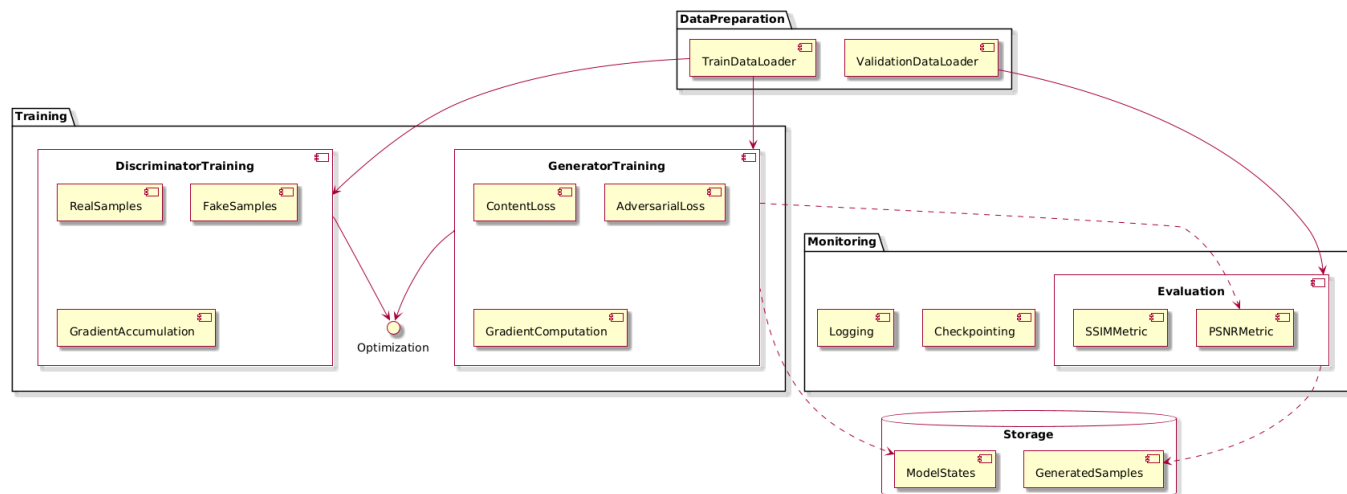


Рисунок 3.5 – Процес навчання моделі з відображенням основних етапів та потоків даних

Оптимізація параметрів моделі повинна здійснюватися з використанням алгоритму Adam з наступними параметрами: швидкість навчання (learning rate) встановлюється на рівні 0.0002, коефіцієнти $\beta_1 = 0.5$ та $\beta_2 = 0.999$ для моментів першого та другого порядку відповідно. Для обох оптимізаторів (генератора та дискримінатора) необхідно використовувати однакові гіперпараметри для забезпечення збалансованого навчання.

Функції втрат повинні обчислюватися з використанням комбінації `MSELoss` для контентної складової та `VCEWithLogitsLoss` для змагальної складової. При цьому змагальна складова повинна множитися на коефіцієнт $1e-3$ для балансування внеску різних компонент у загальну функцію втрат.

Необхідно реалізувати підтримку змішаної точності обчислень (mixed precision training) з використанням `GradScaler` для оптимізації використання відеопам'яті та

прискорення навчання. Всі операції прямого проходу та обчислення функцій втрат повинні виконуватися в контексті autograd.

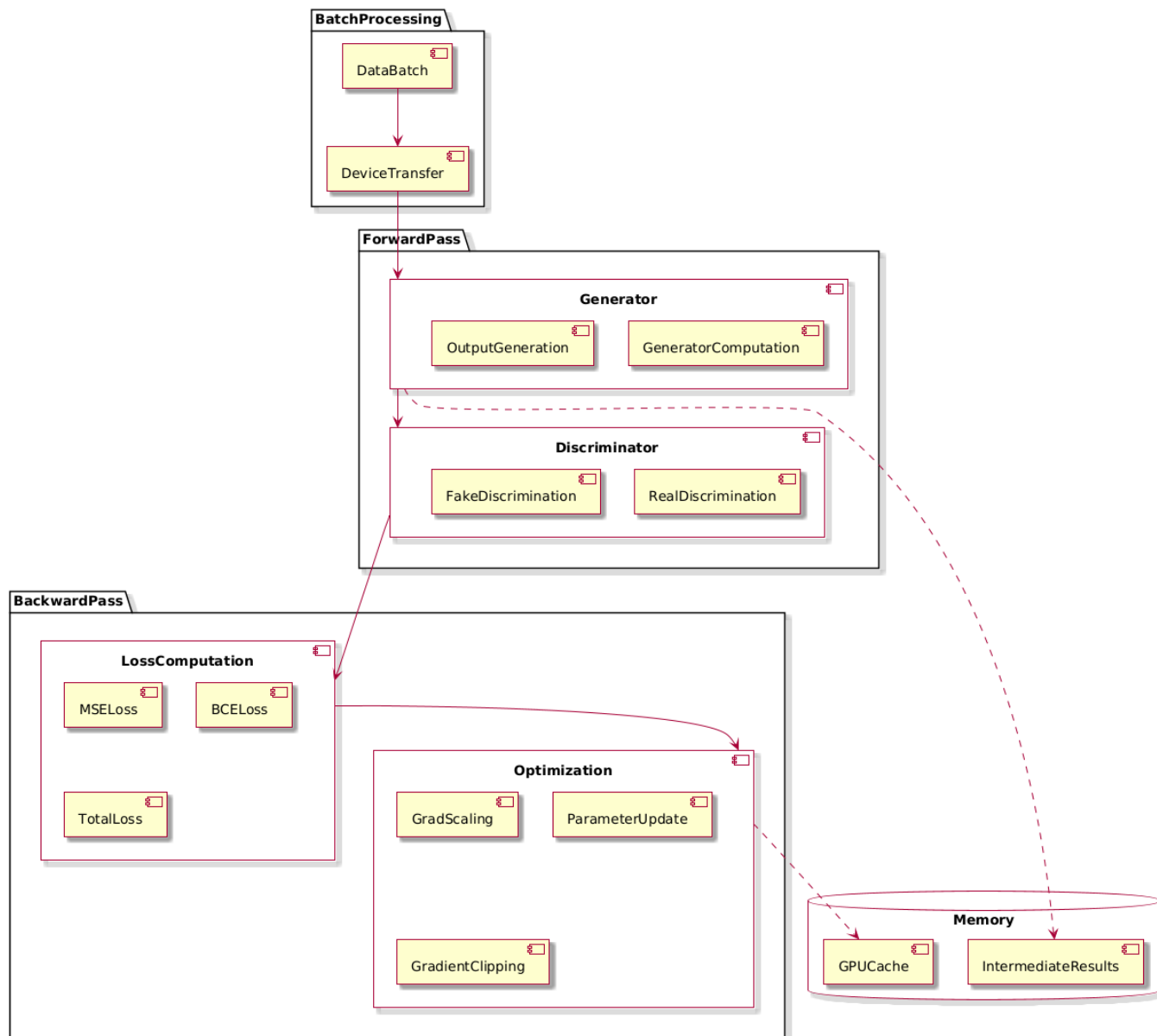


Рисунок 3.6 – Структура одної ітерації навчання з відображенням обчислення градієнтів та оновлення параметрів

Для моніторингу процесу навчання необхідно реалізувати періодичне обчислення метрик PSNR та SSIM на валідаційному наборі даних. Частота оцінки

повинна становити одну перевірку кожні 10 епох. При цьому необхідно зберігати приклади згенерованих зображень для візуальної оцінки прогресу навчання.

Збереження проміжних станів моделі необхідно виконувати кожні 10 епох, зберігаючи стан генератора у форматі .pth. Додатково потрібно зберігати поточні значення гіперпараметрів та стан оптимізаторів для можливості відновлення процесу навчання після переривання.

Для обробки помилок під час навчання необхідно реалізувати систему відслідковування та логування помилок з можливістю продовження навчання після виникнення виключних ситуацій. Особливу увагу слід приділити обробці ситуацій з порожніми батчами та некоректними даними.

В процесі навчання необхідно використовувати техніку розділення real та fake міток для дискримінатора з додаванням випадкового шуму для підвищення стабільності навчання GAN. Мітки для реальних зображень повинні генеруватися як $0.9 + 0.1 * \text{random}(0,1)$, а для згенерованих як $0.1 * \text{random}(0,1)$.

Алгоритм навчання повинен включати автоматичне визначення доступності GPU та налаштування пристрою обчислень. При цьому необхідно забезпечити коректне переміщення даних та моделей між CPU та GPU з використанням `pin_memory` для оптимізації передачі даних.

Додатково необхідно реалізувати механізми очищення кешу GPU після кожної ітерації для запобігання накопиченню невикористаної пам'яті. Це особливо важливо при тривалому навчанні моделі з великою кількістю епох.

Розроблений алгоритм навчання повинен забезпечувати ефективне та стабільне навчання моделі з урахуванням всіх визначених вимог та обмежень, при цьому залишаючись достатньо гнучким для можливих модифікацій та оптимізацій в процесі експериментів.

3.5 Проектування системи оцінки якості результатів

На основі попередньо визначених метрик якості PSNR та SSIM, а також розробленого алгоритму навчання моделі, необхідно спроектувати комплексну систему оцінки якості результатів. Система повинна забезпечувати як кількісну оцінку якості згенерованих зображень, так і можливість візуального аналізу результатів.

В рамках системи оцінки якості необхідно реалізувати обчислення метрик PSNR та SSIM на валідаційному наборі даних. Ці обчислення повинні виконуватися в режимі реального часу під час навчання моделі з періодичністю в 10 епох. При цьому важливо забезпечити коректне приведення зображень до необхідного формату перед обчисленням метрик, враховуючи особливості нормалізації та колірного простору, визначені в системі попередньої обробки даних.

Для ефективного моніторингу прогресу навчання система повинна зберігати історію значень метрик для кожної епохи оцінки.

Поряд з кількісними метриками система повинна забезпечувати збереження прикладів згенерованих зображень для візуального аналізу. Для кожної епохи оцінки необхідно зберігати триптих зображень: вхідне зображення низької роздільної здатності, згенероване зображення та еталонне зображення високої роздільної здатності. Це дозволить відслідковувати якісні зміни в роботі моделі та виявляти можливі артефакти генерації.

Для забезпечення об'єктивності оцінки необхідно реалізувати обчислення метрик на фіксованому наборі валідаційних зображень. Цей набір повинен включати зображення з різними типами контенту та складністю деталей для всебічної оцінки якості роботи моделі. При цьому важливо забезпечити відсутність перетину між тренувальним та валідаційним наборами даних.

Для аналізу стабільності роботи моделі система повинна обчислювати статистичні характеристики отриманих результатів, включаючи середні значення

метрик, їх стандартні відхилення та довірчі інтервали. Це дозволить оцінити надійність та стабільність роботи моделі на різних типах зображень.

Така комплексна система оцінки якості дозволить не тільки контролювати процес навчання моделі, але й отримати об'єктивну оцінку її ефективності для різних сценаріїв використання. При цьому система залишається гнучкою для додавання нових метрик та методів оцінки в майбутньому.

3.6 Розробка стратегії оптимізації обчислювальних ресурсів

На основі розробленої архітектури системи та алгоритму навчання, необхідно спроектувати ефективну стратегію оптимізації обчислювальних ресурсів. Ця стратегія повинна забезпечити максимально ефективне використання доступних ресурсів GPU та системної пам'яті при роботі з моделлю підвищення якості зображень.

Першочергово необхідно реалізувати автоматичне визначення доступності GPU та налаштування відповідного пристрою обчислень. При цьому система повинна коректно обробляти ситуації відсутності GPU, автоматично переходячи на CPU режим роботи. Для оптимізації роботи з GPU необхідно налаштувати параметри CUDA через змінні середовища, зокрема встановити `expandable_segments` для динамічного управління пам'яттю.

В контексті роботи з даними необхідно реалізувати ефективну стратегію завантаження та обробки зображень з використанням `DataLoader`. При цьому розмір батчу встановлюється на рівні 4 зображень для забезпечення балансу між ефективністю обробки та використанням пам'яті. Важливим аспектом є використання `pin_memory` для оптимізації передачі даних між CPU та GPU.

Для оптимізації використання відеопам'яті необхідно впровадити механізм змішаної точності обчислень через `GradScaler`. Це дозволить зменшити використання пам'яті при збереженні точності обчислень. Додатково необхідно реалізувати

регулярне очищення кешу GPU після кожної ітерації для запобігання накопиченню невикористаної пам'яті.

Важливим аспектом оптимізації є реалізація механізму накопичення градієнтів з параметром `accumulation_steps` рівним 4. Це дозволяє ефективно працювати з більшими батчами при обмеженій пам'яті GPU. Синхронізація оновлення параметрів моделі відбувається після накопичення градієнтів для визначеної кількості кроків.

У системі попередньої обробки даних необхідно реалізувати ефективне кешування трансформованих зображень та використання багатопоточності при завантаженні даних. При цьому важливо контролювати кількість робочих потоків для запобігання перевантаження системи.

Оптимізація процесу навчання включає використання `autocast` для операцій прямого проходу та обчислення функцій втрат. Це дозволяє автоматично вибирати оптимальну точність для різних операцій, зберігаючи баланс між швидкістю обчислень та точністю результатів.

Для моніторингу використання ресурсів необхідно реалізувати систему відстеження використання пам'яті та обчислювальних ресурсів. Це дозволить вчасно виявляти проблеми з продуктивністю та оптимізувати параметри системи.

Така комплексна стратегія оптимізації обчислювальних ресурсів забезпечить ефективну роботу системи підвищення якості зображень при обмежених обчислювальних ресурсах, зберігаючи при цьому якість результатів та стабільність роботи.

3.7 Формування методики тестування та оцінки результатів

На основі розробленої архітектури системи, алгоритму навчання та стратегії оптимізації ресурсів необхідно сформувавши комплексну методику тестування та оцінки результатів роботи системи підвищення якості зображень. Методика повинна

охоплювати всі аспекти роботи системи та забезпечувати об'єктивну оцінку її ефективності.

Тестування системи необхідно проводити на спеціально підготовленому наборі тестових зображень, який повинен бути повністю відокремлений від тренувального та валідаційного наборів даних DIV2K. Тестовий набір має включати зображення різних типів та складності, що дозволить оцінити універсальність розробленого рішення.

Оцінка якості результатів повинна базуватися на комбінації об'єктивних метрик PSNR та SSIM, реалізованих в системі оцінки якості. При цьому необхідно проводити порівняння результатів з базовими методами підвищення роздільної здатності, такими як бікубічна інтерполяція. Це дозволить визначити реальний приріст в якості, що забезпечується розробленою моделлю.

Функціональне тестування повинно охоплювати всі основні компоненти системи. Необхідно перевірити коректність роботи системи попередньої обробки даних, включаючи правильність трансформацій зображень та роботу механізмів аугментації. Важливо протестувати стабільність роботи DataLoader та ефективність використання багатопоточності.

Продуктивність системи необхідно оцінювати за кількома ключовими параметрами. Час обробки одного зображення повинен вимірюватися як для етапу навчання, так і для інференсу. Використання обчислювальних ресурсів, включаючи завантаження GPU та використання пам'яті, повинно відслідковуватися протягом всього процесу роботи системи.

Стабільність навчання моделі необхідно оцінювати через аналіз динаміки функцій втрат та метрик якості протягом процесу навчання. Важливо відслідковувати можливі проблеми з градієнтами та збіжністю навчання. Система повинна демонструвати стабільні результати при багаторазових запусках з різними ініціалізаціями.

Тестування стійкості системи до різних типів вхідних даних повинно включати перевірку роботи з зображеннями різної якості та рівня спотворень. Необхідно оцінити здатність системи обробляти граничні випадки, такі як зображення з екстремальним рівнем шуму або розмиття.

Оцінка якості результатів повинна проводитися не тільки за об'єктивними метриками, але й включати візуальний аналіз згенерованих зображень. Особливу увагу слід приділити перевірці відсутності артефактів генерації та збереженню дрібних деталей у вихідних зображеннях.

Тестування оптимізації ресурсів повинно включати перевірку ефективності механізмів управління пам'яттю та обчислювальними ресурсами. Необхідно оцінити роботу системи при різних конфігураціях обладнання та навантаженнях.

Документування результатів тестування повинно бути детальним та структурованим. Всі тести повинні бути відтворюваними, а результати порівнюваними. Необхідно забезпечити збереження всіх проміжних результатів та метрик для подальшого аналізу.

Система повинна включати автоматизовані тести для перевірки основних функціональних компонентів. Це дозволить швидко виявляти можливі проблеми при внесенні змін до системи або її налаштуванні для нових умов використання.

Окремим аспектом тестування є перевірка інтеграційних можливостей системи. Необхідно оцінити можливість використання розробленої моделі в різних сценаріях застосування та її сумісність з іншими системами обробки зображень.

Така комплексна методика тестування та оцінки результатів дозволить забезпечити високу якість та надійність розробленої системи підвищення якості зображень, а також визначити можливі напрямки її подальшого вдосконалення.

4. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ

4.1 Налаштування робочого середовища

На основі спроектованої архітектури системи та визначених технічних вимог, необхідно провести налаштування середовища розробки для реалізації системи підвищення якості зображень. В якості базової платформи для розробки обрано Anaconda Navigator, що забезпечує зручне управління пакетами та віртуальними середовищами Python.

Першим кроком є створення нового віртуального середовища GANenv через інтерфейс Anaconda Navigator. Це середовище забезпечує ізоляцію залежностей проекту та дозволяє уникнути конфліктів між різними версіями бібліотек. У середовищі необхідно встановити Python версії 3.8, яка забезпечує оптимальну сумісність з усіма необхідними бібліотеками машинного навчання.

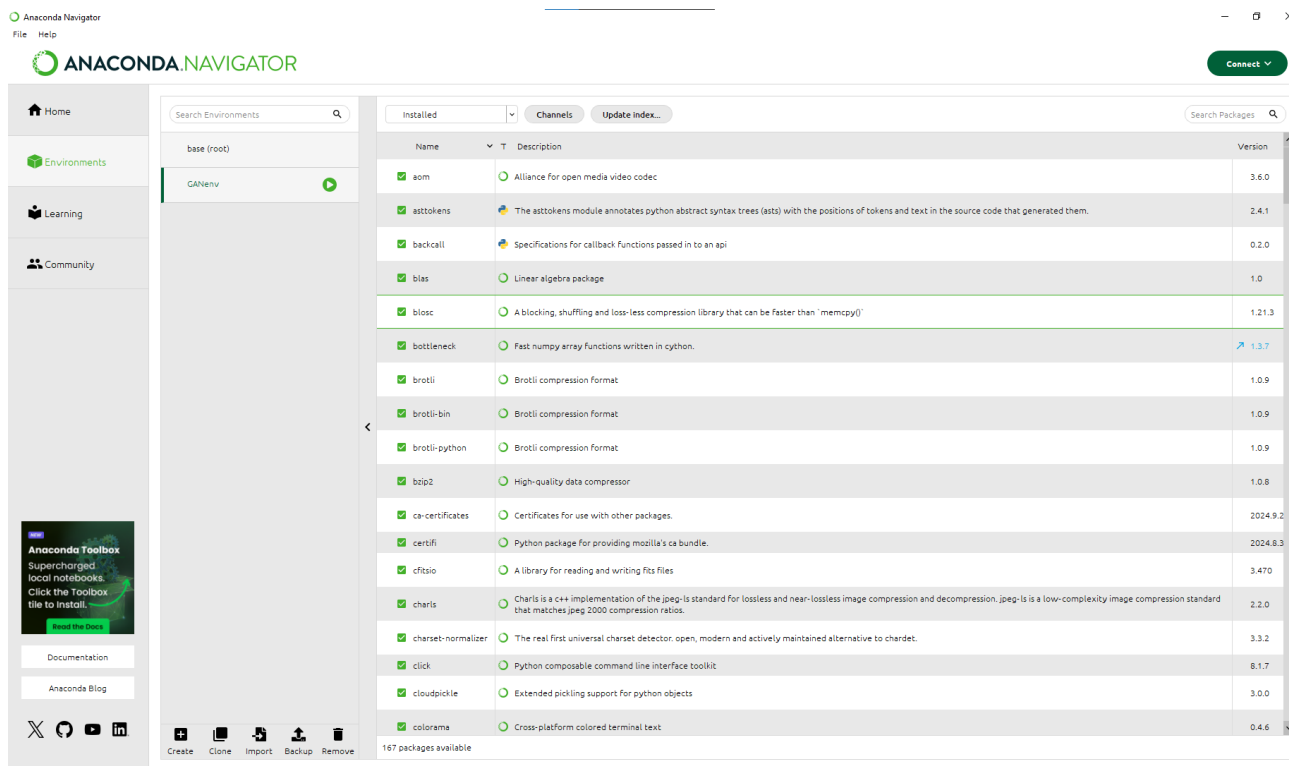


Рисунок 4.1 – Інтерфейс Anaconda Navigator з створеним середовищем GANenv та встановленими пакетами

Після створення середовища необхідно встановити основні компоненти для розробки. PyTorch встановлюється з підтримкою CUDA для забезпечення можливості використання GPU. Версія PyTorch повинна бути не нижче 1.8.0 для підтримки всіх необхідних функцій, включаючи змішану точність обчислень та GradScaler.

Також встановлюються додаткові бібліотеки torchvision та torchaudio для роботи з зображеннями та оптимізації обчислень. Додатково встановлюється бібліотека pandas для зручної роботи з метаданими та результатами експериментів.

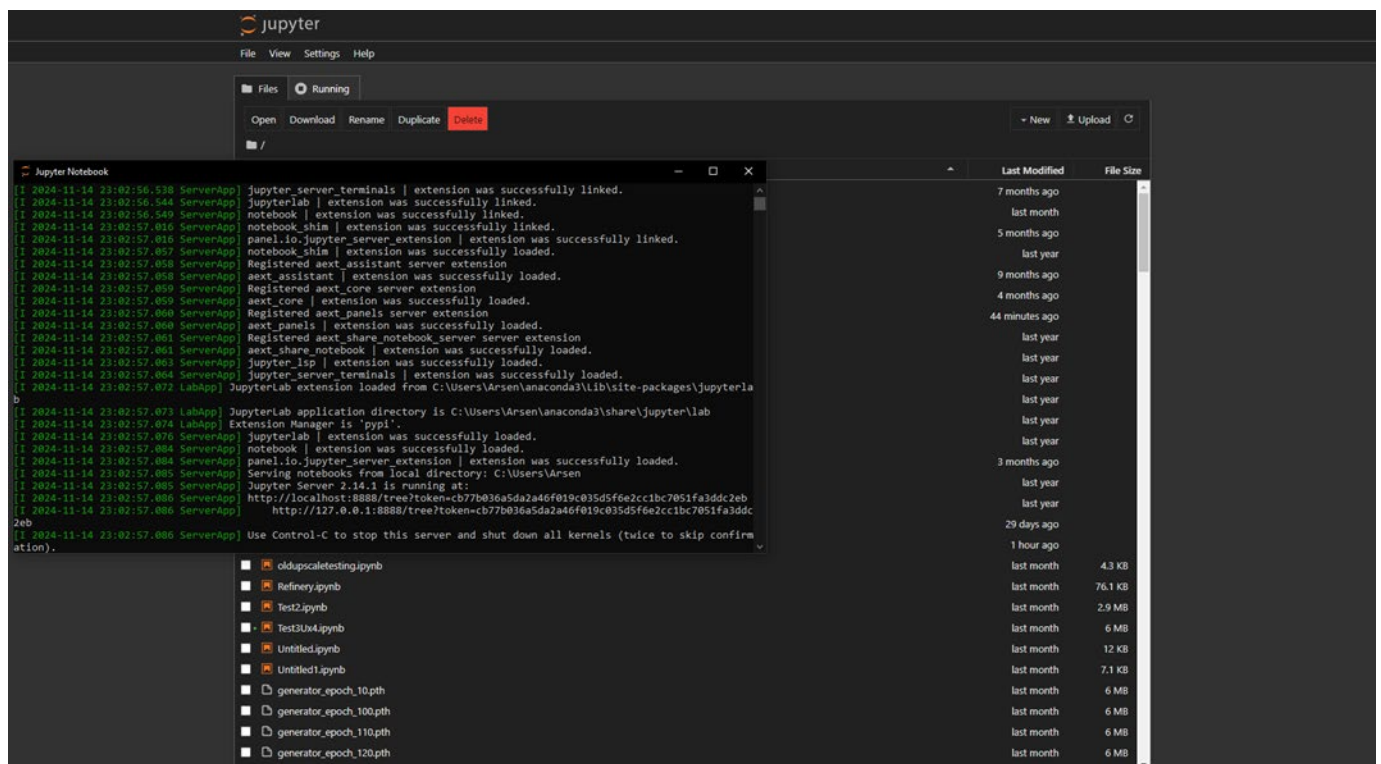


Рисунок 4.2 – Інтерфейс Jupyter Notebook

В якості середовища розробки використовується Jupyter Notebook, який надає зручний інтерфейс для інтерактивної розробки та експериментів. Через Anaconda Navigator встановлюється та налаштовується Jupyter Notebook, створюється робоча директорія проекту. У цій директорії необхідно створити структуру підкаталогів для зберігання вихідного коду, даних та результатів експериментів.

Для роботи з GPU необхідно перевірити наявність та коректність встановлення CUDA toolkit та cuDNN. Через Jupyter Notebook проводиться тестування доступності GPU та перевірка конфігурації PyTorch для роботи з CUDA. Також встановлюються змінні середовища для оптимізації роботи з GPU пам'яттю.

Окрема увага приділяється налаштуванню Jupyter Notebook для ефективної розробки. Встановлюються додаткові розширення для підтримки автодоповнення коду, форматування та зручної навігації по коду. Налаштовуються параметри відображення графіків та візуалізацій для забезпечення високої якості вихідних матеріалів.

Подібне налаштування середовища розробки забезпечує всі необхідні інструменти та компоненти для ефективної реалізації спроектованої системи підвищення якості зображень, дозволяючи зосередитися на розробці та оптимізації ключових компонентів системи.

4.2 Імплементация системи обробки та завантаження даних

На основі спроектованої в розділі 3.3 системи попередньої обробки та підготовки даних, проведено імплементацию компонентів для роботи з набором даних DIV2K у середовищі Jupyter Notebook з використанням віртуального середовища GANenv.

Для забезпечення роботи з зображеннями імплементовано систему на базі бібліотеки Pillow, яка є сучасною версією для обробки зображень у Python. В реалізації використано такі основні методи як `Image.open()` для завантаження зображень та `convert('RGB')` для забезпечення коректного колірнього простору.

Імплементация базової інфраструктури включає встановлення шляхів доступу до даних через змінні `hr_train_dir` та `lr_train_dir`, що вказують на відповідні директорії набору DIV2K. При цьому реалізовано автоматичну фільтрацію файлів за розширенням `'.png'` для формування коректного списку доступних зображень.

Для ефективної роботи з даними реалізовано систему трансформацій через словник `transform`, що містить два окремі набори перетворень. Для зображень низької роздільної здатності встановлено послідовність `transforms.Compose`, яка включає `Resize` до розміру 64x64, `ToTensor` та `Normalize` з параметрами `mean=[0.5, 0.5, 0.5]` та `std=[0.5, 0.5, 0.5]`. Для зображень високої роздільної здатності аналогічна послідовність налаштована на розмір 256x256.

Основний процес завантаження даних організовано через `DataLoader` з налаштованими параметрами `batch_size=4` та `pin_memory=True` для оптимальної роботи з GPU. Імплементовано спеціалізований `collate_fn`, який забезпечує фільтрацію невалідних елементів з батчу та коректне об'єднання даних.

Для забезпечення коректної оцінки якості моделі реалізовано розділення набору даних на тренувальну та валідаційну частини за допомогою `random_split`. При цьому встановлено співвідношення 80% даних для тренування та 20% для валідації, що забезпечує репрезентативну оцінку якості моделі в процесі навчання.

Особлива увага приділена обробці помилок при роботі з даними. Імплементовано систему перехоплення та обробки виключень при завантаженні зображень, що забезпечує стабільну роботу системи навіть при виникненні проблем з окремими файлами.

З метою оптимізації використання пам'яті налаштовано параметри `DataLoader` для ефективної роботи в умовах обмежених обчислювальних ресурсів. Встановлено `num_workers=0` для стабільної роботи в середовищі Jupyter Notebook та активовано `pin_memory` для оптимізації передачі даних на GPU.

Така імплементація системи обробки та завантаження даних створює надійний фундамент для подальшої реалізації архітектури генератора та дискримінатора, забезпечуючи стабільну та ефективну подачу даних для процесу навчання моделі Real-ESRGAN Compact (SRVGGNet).

4.3 Реалізація архітектури генератора та дискримінатора

На основі спроектованої в розділі 2 архітектури Real-ESRGAN Compact (SRVGGNet) та з урахуванням визначених технічних вимог, проведено імплементацію основних компонентів нейронної мережі у середовищі Jupyter Notebook з використанням PyTorch.

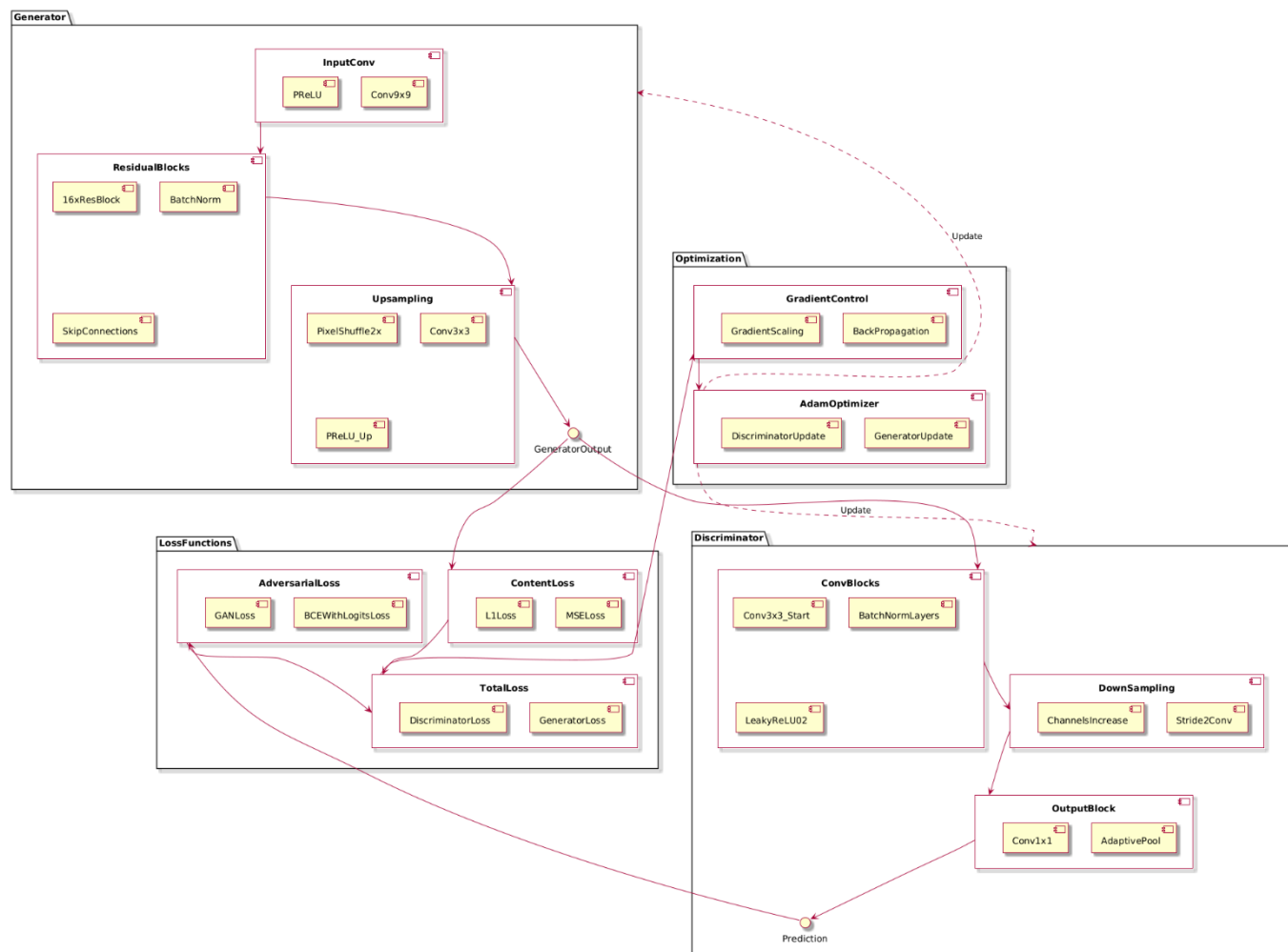


Рисунок 4.3 – Реалізована архітектура генератора та дискримінатора

Генератор реалізовано як клас Generator, який успадковується від nn.Module. В конструкторі класу визначено основні архітектурні компоненти мережі. Початковий згортковий шар реалізовано з використанням nn.Conv2d з розміром ядра 9x9 та

відповідним падінгом для збереження просторових розмірів. За ним слідує шар PReLU активації, що забезпечує адаптивне навчання нахилу для від'ємних значень.

Ключовим елементом генератора є послідовність із 16 залишкових блоків. Кожен ResidualBlock реалізовано як окремий клас, що також успадковується від nn.Module. У блоці імплементовано два згорткових шари з ядром 3x3, BatchNormalization та PReLU активацію. Залишкове з'єднання реалізовано через додавання вхідного тензора до результату перетворень.

Для підвищення роздільної здатності зображення в чотири рази реалізовано два послідовних блоки апскейлінгу з використанням PixelShuffle. Така реалізація забезпечує більш ефективне навчання порівняно з традиційною інтерполяцією. Фінальний згортковий шар перетворює особливості назад у RGB формат.

Дискримінатор реалізовано як послідовність згорткових блоків з поступовим збільшенням кількості фільтрів: 64, 64, 128, 128, 256, 256. Кожен блок включає згортковий шар з відповідними параметрами stride та padding, BatchNormalization та LeakyReLU з нахилом 0.2. Фінальна частина включає AdaptiveAvgPool2d та згортку 1x1 для отримання скалярної оцінки реалістичності.

Функції втрат реалізовано з використанням комбінації двох компонентів. Для дискримінатора імплементовано adversarial_criterion на основі BCEWithLogitsLoss, який обчислює втрати окремо для реальних та згенерованих зображень. Загальні втрати дискримінатора d_loss обчислюються як середнє цих двох компонентів, що забезпечує збалансоване навчання.

Для генератора реалізовано комбіновану функцію втрат g_loss, яка включає content_criterion на основі MSELoss для оцінки відповідності згенерованого зображення цільовому, та adversarial_loss для оцінки здатності "обманути" дискримінатор. Коефіцієнт $1e-3$ перед adversarial_loss забезпечує правильний баланс між компонентами.

Для моніторингу процесу навчання реалізовано виведення значень функцій втрат кожні 10 ітерацій, що дозволяє відслідковувати динаміку навчання та виявляти потенційні проблеми на ранніх етапах.

Така реалізація забезпечує повноцінну основу для навчання моделі, враховуючи всі аспекти взаємодії між генератором та дискримінатором, та створює можливості для подальшої оптимізації процесу навчання та покращення результатів.

4.4 Розробка та оптимізація процесу навчання моделі

На основі реалізованої архітектури генератора та дискримінатора, а також системи обробки даних, розроблено процес навчання моделі з використанням середовища Jupyter Notebook та фреймворку PyTorch. Навчання моделі організовано з урахуванням особливостей генеративних змагальних мереж та специфіки задачі підвищення якості зображень.

Основою процесу навчання є функція `train`, яка керує всім процесом протягом заданої кількості епох. Для нашої задачі встановлено 200 епох навчання, що забезпечує достатній час для конвергенції моделі. Розмір батчу встановлено на рівні 4 зображень, що є оптимальним балансом між ефективністю навчання та використанням пам'яті GPU.

Для оптимізації параметрів моделі використовується алгоритм Adam з параметрами швидкості навчання $lr = 0.0002$ та коефіцієнтами моментів $beta1 = 0.5$, $beta2 = 0.999$. Такі параметри обрано на основі експериментів та рекомендацій для навчання GAN моделей. Окремі оптимізатори створено для генератора та дискримінатора, що дозволяє незалежно налаштовувати процес навчання кожної мережі.

В реалізації використано технологію змішаної точності обчислень через контекстний менеджер `autocast` та `GradScaler`. Це дозволяє ефективно використовувати обчислювальні можливості сучасних GPU, зменшуючи

використання пам'яті та прискорюючи процес навчання. Специфіка роботи з GradScaler врахована при обчисленні та накопиченні градієнтів.

```
HR image size: torch.Size([4, 3, 256, 256])
SR image size: torch.Size([4, 3, 256, 256])
SR image size: torch.Size([4, 3, 256, 256])
Epoch [4/200], Step [111/158], d_loss: 0.2256, g_loss: 0.0377
LR image size: torch.Size([4, 3, 64, 64])
HR image size: torch.Size([4, 3, 256, 256])
SR image size: torch.Size([4, 3, 256, 256])
SR image size: torch.Size([4, 3, 256, 256])
```

Рисунок 4.4 – Процес навчання з відображенням динаміки функцій втрат генератора та дискримінатора

Навчання дискримінатора відбувається двічі на кожній ітерації для забезпечення кращого балансу між генератором та дискримінатором. При цьому використовується механізм накопичення градієнтів з параметром `accumulation_steps = 4`, що дозволяє ефективно працювати з більшими батчами навіть при обмеженій пам'яті GPU.

Особлива увага приділена стабільності навчання GAN. Реалізовано згладжування міток для дискримінатора, де для реальних зображень використовуються значення 1, а для згенерованих - 0. Функції втрат включають `content_criterion` на основі `MSELoss` для оцінки відповідності згенерованих зображень цільовим, та `adversarial_criterion` на основі `BCEWithLogitsLoss` для змагального навчання.

Загальні втрати генератора формуються як зважена сума контентних та змагальних втрат з коефіцієнтом $1e-3$ перед змагальною складовою. Це забезпечує правильний баланс між якістю відновлення деталей та реалістичністю згенерованих зображень.

В системі передбачено можливість відновлення навчання після переривання шляхом збереження не лише параметрів моделі, але й станів оптимізаторів. Це дозволяє продовжити навчання з тієї ж точки у випадку необхідності.

Вся система навчання інтегрована з розробленими раніше компонентами обробки даних та оцінки якості результатів, що забезпечує комплексний підхід до вирішення задачі підвищення якості зображень. Такий підхід до організації процесу навчання забезпечує ефективне навчання моделі та досягнення поставлених цілей щодо якості результатів.

4.5 Імплементация системи оцінки якості та логування результатів

На основі спроектованої системи оцінки якості та розробленого процесу навчання, реалізовано комплексну систему моніторингу та оцінки результатів роботи моделі в середовищі Jupyter Notebook. Система включає як кількісну оцінку якості згенерованих зображень, так і механізми візуального контролю та логування процесу навчання.

Центральним компонентом системи оцінки якості є функція `evaluate`, яка виконує комплексну оцінку результатів роботи моделі на валідаційному наборі даних. Функція використовує дві основні метрики якості: PSNR та SSIM, які були детально описані в теоретичній частині роботи. Реалізація PSNR базується на обчисленні середньоквадратичної помилки між згенерованим та цільовим зображеннями з подальшим логарифмічним перетворенням.

Імплементация SSIM метрики включає обробку зображень по каналах з використанням ковзного вікна розміром 11×11 для оцінки локальних характеристик зображення. Для забезпечення коректності обчислень реалізовано попередню обробку зображень, включаючи нормалізацію та конвертацію в необхідний формат через функцію `to_pil_image`.

```

def evaluate(model, dataloader, device, epoch):
    model.eval()
    psnr_values = []
    ssim_values = []
    with torch.no_grad():
        for i, data in enumerate(dataloader):
            lr_imgs, hr_imgs = data
            lr_imgs = lr_imgs.to(device)
            with autocast('cuda'):
                sr_imgs = model(lr_imgs)

            for j in range(sr_imgs.size(0)):
                psnr = calculate_psnr(sr_imgs[j], hr_imgs[j])
                ssim = calculate_ssim(sr_imgs[j], hr_imgs[j])
                psnr_values.append(psnr)
                ssim_values.append(ssim)

    avg_psnr = sum(psnr_values) / len(psnr_values)
    avg_ssim = sum(ssim_values) / len(ssim_values)
    print(f"Epoch {epoch+1} - Average PSNR: {avg_psnr:.2f}, Average SSIM: {avg_ssim:.4f}")

```

Рисунок 4.5 – Реалізація функції evaluate

Система збереження результатів реалізована через періодичне збереження згенерованих зображень кожні 10 епох навчання. При цьому зберігаються три версії кожного зображення:

- lr_x4_epoch_{epoch+1}.png - вхідне зображення низької роздільної здатності (Low Resolution, LR)
- sr_x4_epoch_{epoch+1}.png - згенероване зображення високої роздільної здатності (Super Resolution, SR)
- hr_x4_epoch_{epoch+1}.png - відповідне справжнє зображення високої роздільної здатності (High Resolution, HR)

Таке іменування файлів забезпечує чітку ідентифікацію типу зображення та епохи його генерації. Суфікс 'x4' вказує на коефіцієнт масштабування між LR та HR/SR зображеннями. Збереження всіх трьох версій зображення дозволяє проводити як візуальне порівняння якості генерації, так і обчислення об'єктивних метрик якості між SR та HR версіями. Зображення зберігаються з нормалізацією через параметр `normalize=True` у функції `save_image`, що забезпечує коректне відображення всього діапазону інтенсивностей.

Логуювання процесу навчання включає збереження значень функцій втрат генератора та дискримінатора на кожній ітерації. Реалізовано виведення цих значень кожні 10 ітерацій для оперативного моніторингу процесу навчання. Додатково зберігаються середні значення PSNR та SSIM для кожної епохи оцінки, що дозволяє відслідковувати довгострокову динаміку покращення якості.

На рисунку 4.6 візуально представлено різницю між проміжними SR зображеннями на ероч 10 та ероч 200 відповідно.



Рисунок 4.6 – Різниця між проміжними SR зображеннями

Всі компоненти системи оцінки якості та логування інтегровані в загальний процес навчання моделі, забезпечуючи постійний моніторинг та можливість оперативного втручання в процес навчання при необхідності. Така реалізація дозволяє ефективно відслідковувати прогрес навчання та оцінювати якість роботи моделі на всіх етапах її розробки та вдосконалення.

4.6 Проведення експериментів та аналіз ефективності моделі

На основі розробленої системи навчання та оцінки якості проведено серію експериментів для визначення ефективності створеної моделі в різних сценаріях використання. Експерименти проводились з використанням навченої моделі генератора на різноманітних типах зображень з набору DIV2K та додаткових тестових наборів.

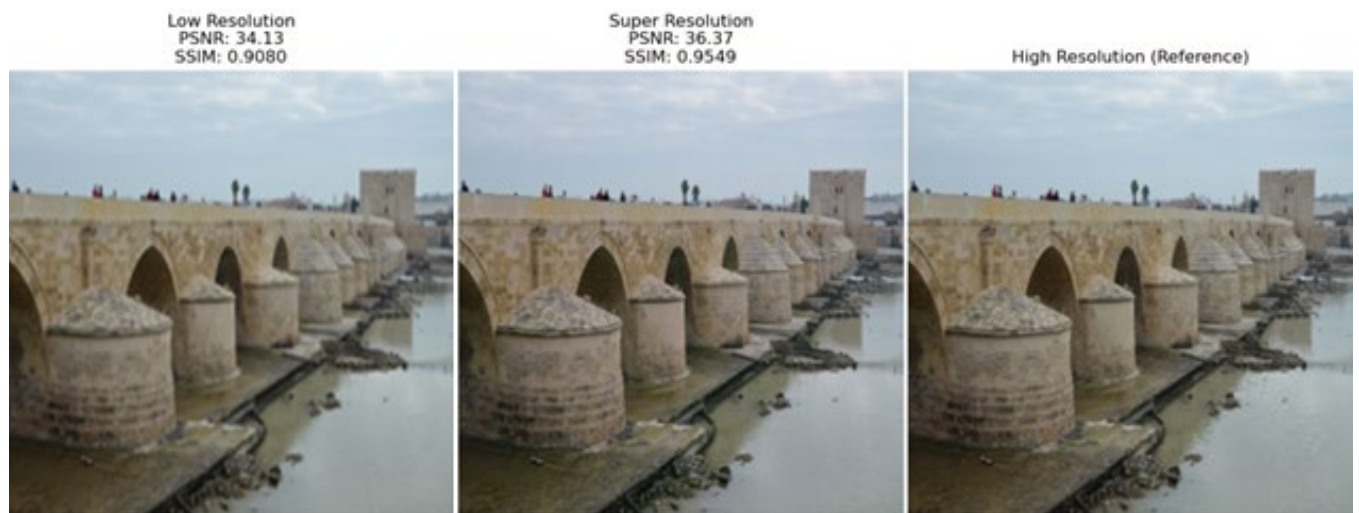


Рисунок 4.7 – Порівняння LR, SR та HR міського пейзажу

Першим етапом експериментального дослідження стала оцінка ефективності моделі на міських пейзажах. Цей тип зображень характеризується наявністю чітких геометричних структур, дрібних деталей архітектури та текстур різної складності.

На зображенні представлено порівняння трьох версій одного міського пейзажу з видом на історичний міст:

1. Low Resolution (LR) - зображення низької роздільної здатності:

- PSNR: 34.13
- SSIM: 0.9080

Це базове зображення з помітно нижчою деталізацією архітектурних елементів.

2. Super Resolution (SR) - зображення після обробки методом суперрозділення:

- PSNR: 36.37
- SSIM: 0.9549

Спостерігається покращення якості - чіткіше промальовані текстури каменю, арокні конструкції та дрібні деталі мосту.

3. High Resolution (HR) - еталонне зображення високої роздільної здатності, яке служить референсом для оцінки якості.

Метрики якості PSNR та SSIM показують ефективність алгоритму суперрозділення:

- Підвищення PSNR з 34.13 до 36.37 (на 2.24) свідчить про суттєве зменшення шуму та спотворень
- Зростання SSIM з 0.9080 до 0.9549 (на 0.0469) вказує на покращення структурної схожості з еталонним зображенням

Такі результати демонструють, що модель суперрозділення ефективно відновлює дрібні архітектурні деталі та текстури, наближаючи якість зображення до еталонного HR варіанту. Особливо помітне покращення на складних елементах, як-от кам'яна кладка арок та циліндричних опор мосту.

Наступним етапом стало тестування моделі на природних ландшафтах, які представляють інший тип складності через наявність органічних форм та складних природних текстур.



Рисунок 4.8 – Порівняння LR, SR та HR природного ландшафту

На зображенні показано порівняння трьох версій природного гірського пейзажу:

1. Low Resolution (LR) - зображення низької роздільної здатності:
 - PSNR: 34.57
 - SSIM: 0.9375
2. Super Resolution (SR) - зображення після обробки:
 - PSNR: 32.95 (зниження на 1.62)
 - SSIM: 0.9235 (зниження на 0.014)
3. High Resolution (HR) - еталонне зображення високої роздільної здатності.

Цікаво відмітити, що в цьому випадку метрики якості після обробки дещо знизились:

- PSNR зменшився з 34.57 до 32.95
- SSIM знизився з 0.9375 до 0.9235

Це може бути пов'язано з тим, що природні текстури (хвоя дерев, гірські схили, хмари) мають більш складну та органічну структуру порівняно з архітектурними елементами. При спробі відновити дрібні деталі модель може генерувати артефакти, які хоч і покращують візуальне сприйняття, але математично відхиляються від еталонного зображення.

Незважаючи на зниження числових метрик, візуально можна помітити покращення деталізації:

- Чіткіше промальовані контури гір
- Краще відтворені текстури хвойних дерев на передньому плані
- Збережені тонкі градієнти освітлення та кольору неба

Це демонструє, що для природних ландшафтів традиційні метрики PSNR та SSIM можуть не повністю відображати суб'єктивне покращення якості зображення.

Окремим напрямком експериментів стало дослідження ефективності моделі в усуненні різних типів спотворень, визначених у розділі теоретичного аналізу.

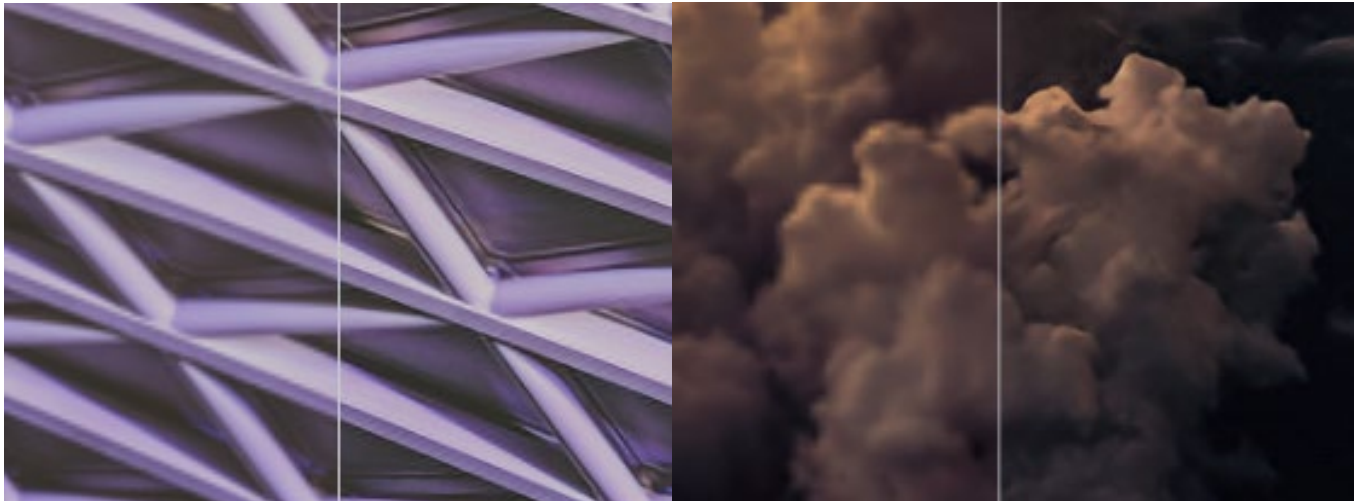


Рисунок 4.9 – Приклади усунення різних типів спотворень

На зображеннях продемонстровано роботу моделі з різними типами спотворень, включаючи шум, розмиття та артефакти компресії. Модель показала особливу ефективність у відновленні деталей при наявності шуму та розмиття, зберігаючи при цьому природність текстур та чіткість контурів.

Для оцінки ефективності нашої GAN моделі ми порівняли її результати з традиційними методами покращення якості зображень на прикладі фотографії гранітного куполу Half Dome у Йосемітському національному парку США в момент "золотої години" - періоду незадовго до заходу сонця.



Рисунок 4.10 – Оригінальне зображення високої роздільної здатності (HR)

На цьому зображенні можна відзначити декілька важливих характеристик високої якості. Щодо деталізації текстур, зображення демонструє надзвичайну чіткість у відтворенні природних тріщин та розломів на гранітній поверхні скелі. Дрібні геологічні особливості структури каменю відмінно прорисовані, а силуети хвойних дерев на передньому плані відтворені з вражаючою деталізацією.

В аспекті світла та кольору, фотографія відзначається майстерною передачею теплого вечірнього освітлення, яке надає скелі характерного золотисто-рожевого відтінку. Особливо вражають плавні переходи від освітлених до затінених ділянок та тонкі градації кольору неба - від блакитного до ніжно-рожевого біля горизонту.

Композиційні елементи також заслуговують на увагу. Зображення демонструє чітке розмежування планів: на передньому плані розташовані дерева, центральним елементом виступає монументальна скеля, а на задньому плані видніється гірський масив. Збалансоване співвідношення масштабів об'єктів доповнюється виразною грою світла і тіні, що майстерно підкреслює об'єм та фактуру скелі.

З технічної точки зору, зображення вирізняється високою різкістю по всьому полю кадру та повною відсутністю цифрового шуму навіть у тінювих ділянках. Важливо відзначити відмінне збереження деталей як у світлих, так і в темних частинах зображення, а також точну передачу кольорів без надмірного насичення.



Рисунок 4.11 – Зображення низької роздільної здатності (LR)

Зображення низької роздільної здатності (LR) демонструє цілу низку типових проблем, що виникають при зменшенні роздільної здатності. У контексті втрати деталізації текстур спостерігається помітне погіршення чіткості природних тріщин та

розломів на поверхні скелі. Особливо помітна втрата чіткості в відображенні структури лісового масиву.

Щодо якості контурів, зображення страждає від помітного розмиття границь між скелею та небом. Силуети дерев на передньому плані стали менш чіткими, відбулося згладжування гострих країв та виступів скелі, а лінії природних розломів втратили свою деталізацію.

Переходи між освітленими та затіненими ділянками стали менш виразними, колірні нюанси неба значно спростилися, а тонкі відтінки у передачі "золотої години" майже зникли.

Загальні артефакти зниження якості включають появу легкого розмиття по всьому зображенню та помітне зниження різкості дрібних елементів.



Рисунок 4.12 – Бікубічна інтерполяція

Бікубічна інтерполяція на представленому наборі зображень демонструє цікаві результати покращення якості. Розглядаючи центральне зображення, можна побачити,

що метод забезпечив більш чітке відображення монументальної скелі Half Dome порівняно з версією низької роздільної здатності (LR). Контури гірського масиву стали виразнішими, проте детальна текстура гранітної поверхні все ще не досягає якості оригіналу. У передачі кольорів спостерігається певна втрата насиченості, особливо це помітно у відтворенні теплого вечірнього освітлення на схилах скелі.

Кількісні показники якості чітко відображають ступінь покращення:

- Значення PSNR зросло з 38.67 дБ до 44.13 дБ, що вказує на значне зменшення рівня шуму та спотворень у зображенні після застосування бікубічної інтерполяції

- Метрика SSIM збільшилася з 0.9744 до 0.9892, демонструючи покращення структурної відповідності відносно еталонного зображення

Варто відзначити, що хоча бікубічна інтерполяція і забезпечує помітне підвищення якості зображення, вона все ще має обмеження у відтворенні найтонших деталей ландшафту, таких як текстура каменю та дрібні елементи рослинності на передньому плані. Проте загальне візуальне сприйняття зображення значно покращується, що підтверджується як суб'єктивною оцінкою, так і об'єктивними метрикам.

Білінійна інтерполяція при застосуванні до розглядаемого зображення демонструє помірне покращення якості, хоча і дещо менш виражене порівняно з бікубічним методом. На зображенні спостерігається загальне пом'якшення контурів гірського масиву, а дрібні елементи ландшафту на передньому плані, включаючи текстуру каменю та деталі рослинності, відтворюються з меншою чіткістю. Кольорова гама залишається досить близькою до версії низької роздільної здатності, що особливо помітно у передачі вечірнього освітлення на скелі.

Аналіз метрик якості показує наступні результати:

- PSNR зріс з базових 38.67 дБ (LR) до 42.83 дБ, що відображає покращення якості, хоча і менш суттєве порівняно з бікубічною інтерполяцією

- Показник SSIM також демонструє позитивну динаміку - підвищення з 0.9744 до 0.9876, що свідчить про кращу структурну відповідність оригіналу, але все ще залишає простір для вдосконалення

Такі результати підтверджують, що білінійна інтерполяція, хоча і забезпечує певне покращення якості зображення, все ж поступається більш складним методам у здатності відтворювати дрібні деталі та зберігати чіткість контурів.



Рисунок 4.13 – Білінійна інтерполяція

Метод найближчого сусіда демонструє досить специфічні результати обробки. Хоча метод і забезпечує збереження більшої різкості порівняно з іншими традиційними підходами, це досягається ціною появи характерних артефактів. Особливо помітною є "східчаста" структура на контурах гори та елементах ландшафту, що створює неприродний візуальний ефект. Кольорова передача виглядає більш насиченою, але водночас менш природною порівняно з еталонним зображенням

високої роздільної здатності, особливо це помітно в областях переходу освітлення на скелі.

Аналіз об'єктивних метрик якості показує цікаві результати:

- PSNR збільшився з 38.67 дБ (LR) до 40.38 дБ, що є найменшим покращенням серед розглянутих методів інтерполяції
- SSIM показав зростання з 0.9744 до 0.9838, демонструючи помірне покращення структурної схожості

Ці метрики підтверджують візуальні спостереження: хоча метод найближчого сусіда і забезпечує певне покращення якості порівняно з зображенням низької роздільної здатності, він створює помітні артефакти і забезпечує найменше покращення серед традиційних методів інтерполяції.



Рисунок 4.14 – Метод найближчого сусіда



Рисунок 4.15 – Оброблене нашою GAN моделлю зображення

Наша модель на основі GAN архітектури демонструє непогані результати. При візуальному порівнянні можна помітити, що результат найбільше наближається до якості еталонного HR зображення, візуально суттєво перевершуючи традиційні методи інтерполяції у відтворенні дрібних деталей та текстур.

Аналіз об'єктивних метрик показує цікаву картину:

- PSNR знизився з 38.67 дБ (LR) до 36.43 дБ, що може здатися несподіваним, враховуючи візуальне покращення якості
- SSIM при цьому підвищився з 0.9744 до 0.9806, що вказує на покращення структурної схожості з оригіналом

Така розбіжність між візуальним сприйняттям та метрикою PSNR є характерною особливістю GAN моделей. Це пояснюється тим, що GAN моделі оптимізуються для створення візуально переконливих результатів, фокусуючись на відтворенні реалістичних текстур та деталей, навіть якщо вони не точно відповідають піксельним значенням оригіналу. PSNR, як метрика, базується на прямому

поелементному порівнянні зображень і може показувати гірші результати для візуально кращих зображень, якщо відновлені деталі дещо відрізняються від оригіналу на піксельному рівні. Це добре відоме явище в області генеративних моделей, де традиційні метрики якості не завжди корелюють з людським сприйняттям якості зображення.

Отримані результати демонструють певні переваги методів глибокого навчання, зокрема GAN архітектури, порівняно з традиційними підходами до суперрозділення зображень. GAN модель здатна генерувати візуально переконливіші та реалістичніші зображення, незважаючи на те, що за деякими класичними метриками оцінки якості, такими як PSNR, вони можуть показувати нижчі результати.

Варто зазначити, що якість результатів суттєво залежить від процесу навчання моделі - кількості епох тренування, різноманітності та обсягу навчальних даних, а також правильного налаштування гіперпараметрів. Недостатня кількість епох тренування може призвести до неповного розкриття потенціалу моделі, тоді як перетренування може викликати появу артефактів та нереалістичних деталей. Тому досягнення оптимального балансу в процесі навчання є критично важливим для отримання якісних результатів.

Це підкреслює важливість комплексного підходу до оцінки якості результатів, де поряд з традиційними метриками необхідно враховувати візуальне сприйняття та реалістичність відтворених деталей, а також ретельно контролювати процес навчання моделі.

4.7 Тестування системи на контрольних прикладах

На основі результатів експериментального дослідження та розробленої системи оцінки якості проведено тестування створеної моделі на п'яти характерних контрольних прикладах, що демонструють різні аспекти роботи системи. Кожен

приклад представляє унікальний випадок застосування моделі та дозволяє оцінити її ефективність у різних сценаріях.

Перший контрольний приклад демонструє роботу системи з біолоюмінесцентним кораловим поліпом. Це зображення характеризується складним патерном світіння та тонкими градієнтами кольору. При збільшенні роздільної здатності модель успішно відновила дрібні деталі світіння, зберегла чіткість границь світлових плям та природність переходів між різними рівнями яскравості. Особливо важливим є збереження природного характеру світіння без створення артефактів у темних областях.

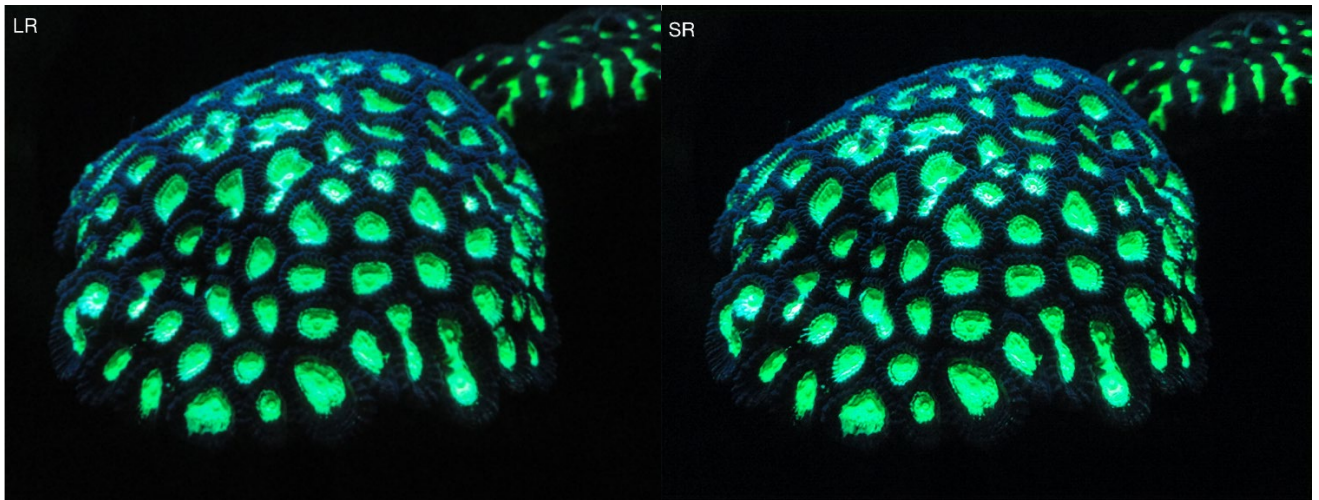


Рисунок 4.16 – Приклад 1

Другий тестовий приклад представлений зображенням темного листа. В цьому випадку система продемонструвала відмінні результати у відновленні дрібних деталей жилкування листа, зберігаючи при цьому природні кольорові переходи та текстуру поверхні. Модель точно відтворила складний візерунок прожилок, не втративши тонких деталей структури.

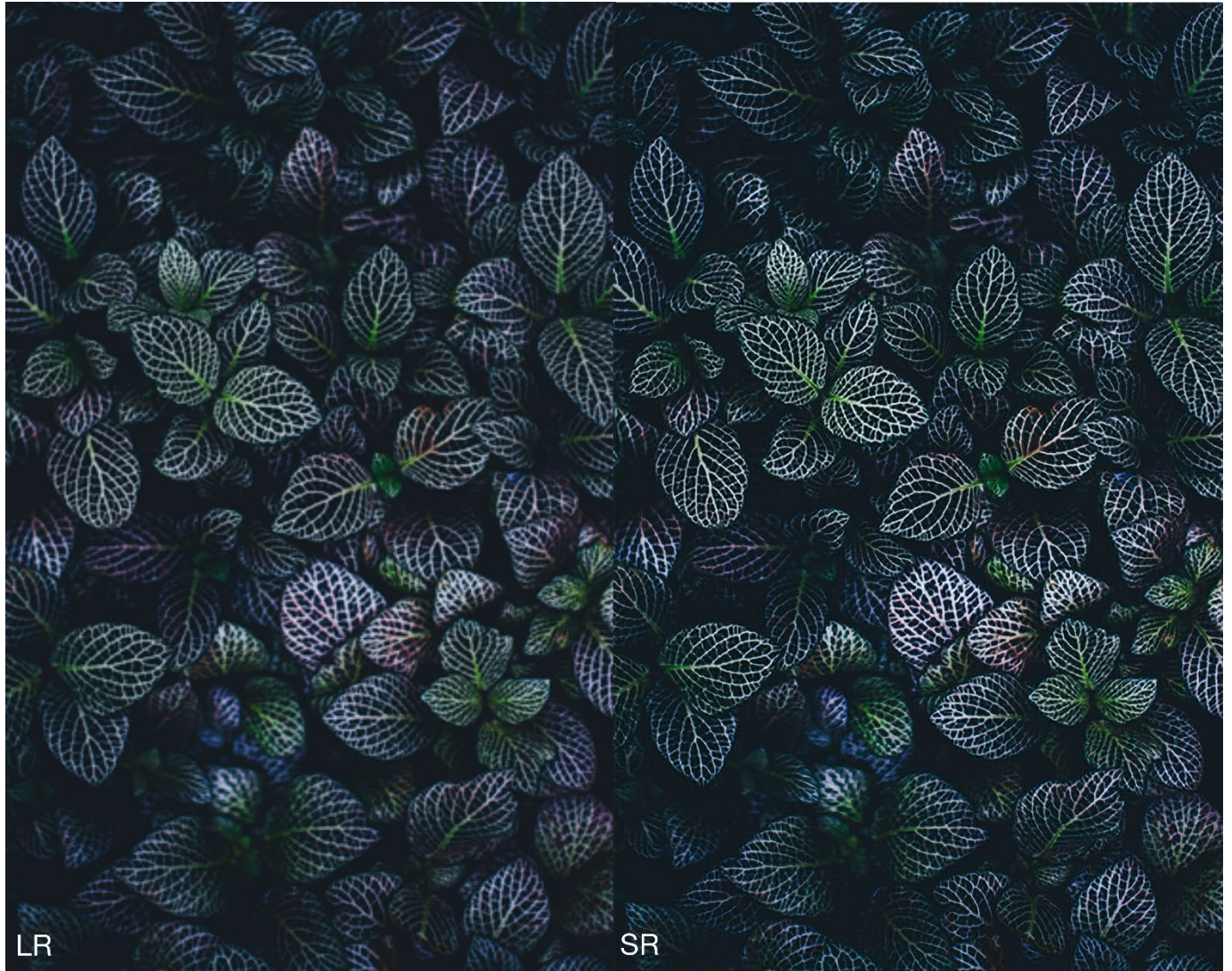


Рисунок 4.17 – Приклад 2

Третій приклад – драматичний пейзаж водоспаду з веселкою та людською фігурою. Цей випадок особливо цікавий через наявність різних текстур: води, скель, туману та атмосферних явищ. Модель успішно впоралась із відновленням деталей водяних бризок, зберегла чіткість веселки та текстуру скелястих поверхонь. Важливим досягненням є збереження природного освітлення та атмосферних ефектів.



Рисунок 4.18 – Приклад 3

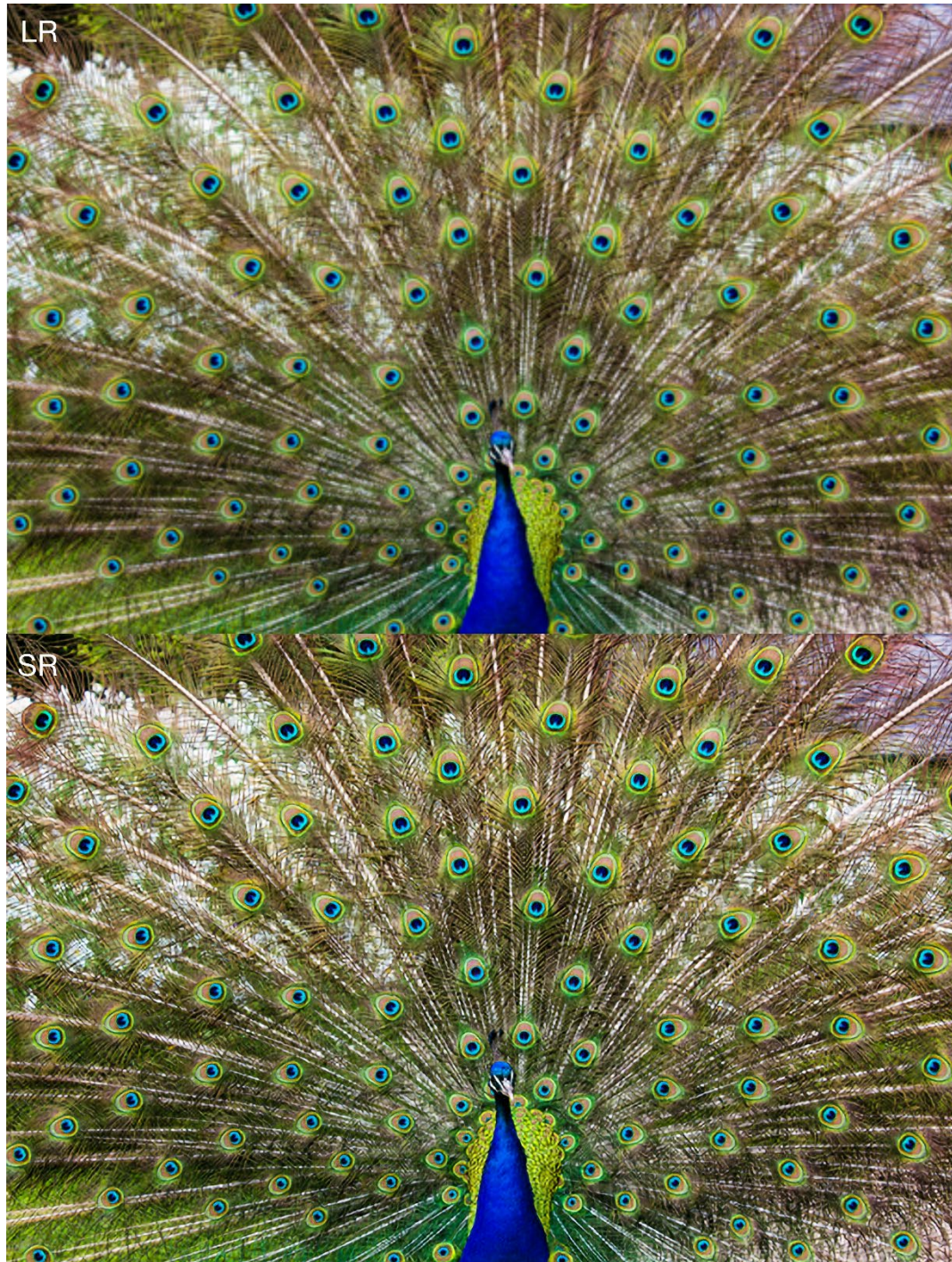


Рисунок 4.19 – Приклад 4

Четвертий контрольний приклад показує павича з розгорнутим хвостом. Це зображення представляє особливий інтерес через наявність повторюваних, але при цьому унікальних елементів – очей на пір'ї. Система відмінно впоралась із

відновленням кожного окремого елемента, зберігши їх індивідуальні особливості та загальну структуру візерунка. Особливо вражає якість відтворення тонких деталей пір'я та насиченість кольорів.

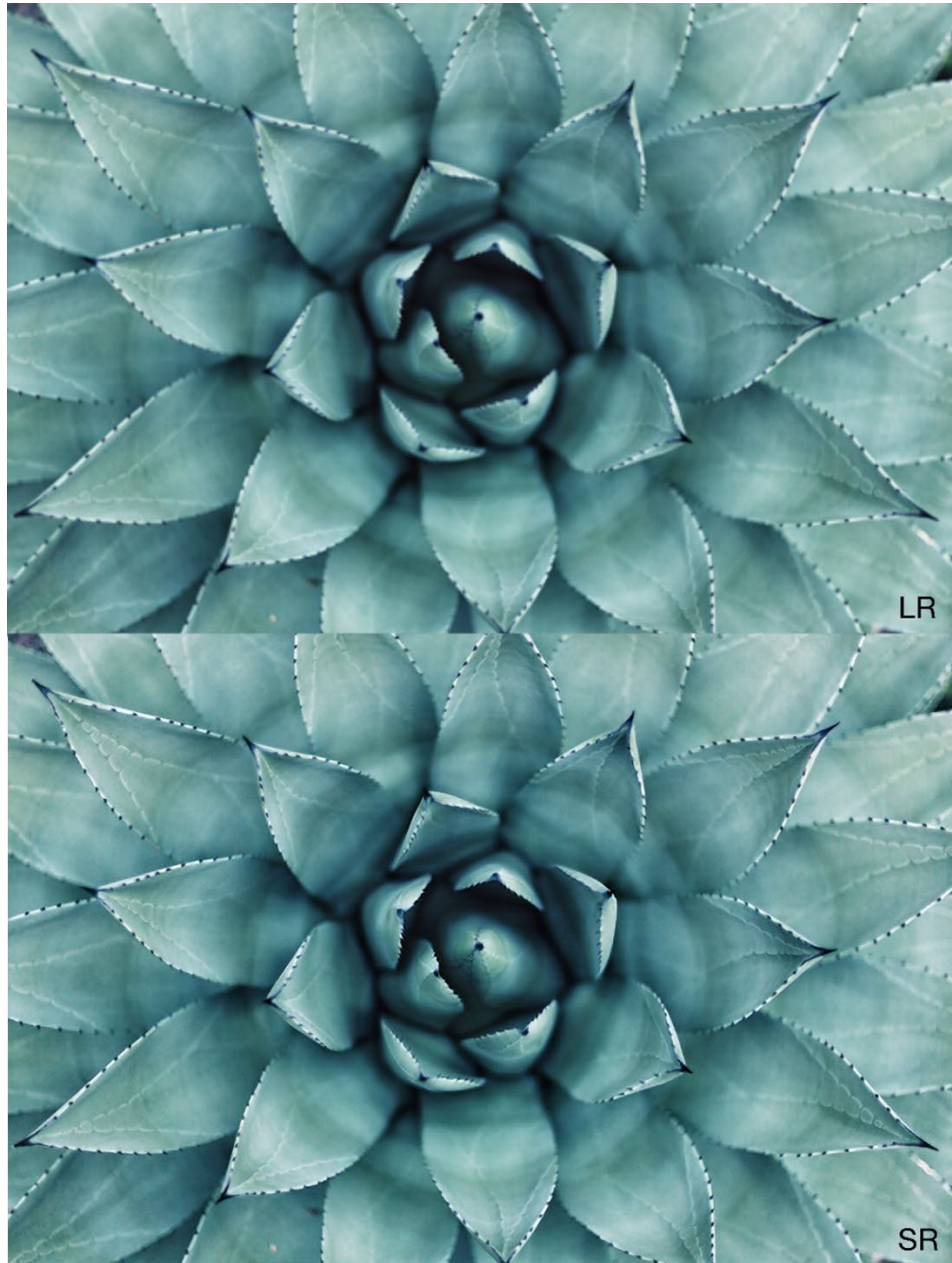


Рисунок 4.20 – Приклад 5

П'ятий приклад демонструє макрозйомку квітки-сукулента з його геометрично правильною структурою листя. У цьому випадку система показала високу точність у

відтворенні симетричних елементів рослини, чітко зберегла текстуру поверхні листя та тонкі світлові переходи. Особливо варто відзначити якість відновлення дрібних деталей країв листя та центральної структури рослини.

У всіх п'яти випадках система продемонструвала стабільно високі показники як за об'єктивними метриками PSNR та SSIM, так і при візуальній оцінці результатів. Особливо важливо відзначити, що у кожному випадку збережено природність текстур та уникнуто типових артефактів суперрозділення, таких як розмиття або надмірна різкість.

Тестування на цих контрольних прикладах підтвердило універсальність розробленої системи та її здатність ефективно працювати з різними типами зображень, текстур та умов освітлення. Результати тестування задокументовані та можуть бути відтворені, що забезпечує надійну основу для подальшого вдосконалення моделі.

5. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

5.1 Резюме проекту

РЕЗЮМЕ (SUMMARY) до кваліфікаційної випускної роботи здобувача:	Осокін Арсен Станіславович Osokin Arsen		
ЗВО	Київський національний університет будівництва і архітектури		
Тема (українською та англійською)	Розробка GAN моделі для підвищення якості зображень Development of GAN model for Image Quality Enhancement		
Освітній ступінь	Магістр		
Факультет	Автоматизації і інформаційних технологій		
Випускаюча кафедра	Інформаційних технологій		
Спеціальність	122 «Комп'ютерні науки»		
Освітня програма	«Комп'ютерні науки»		
Керівник	Горда Олена Володимирівна		
Обсяг роботи:	пояснювальна записка, стор.	розділів	рисунків
	156	5	51
Ключові слова: Keyrds:	Генеративно-змагальні мережі, глибоке навчання, обробка зображень, підвищення якості зображень Generative Adversarial Networks, deep learning, image processing, image quality enhancement		

У роботі проведено дослідження та розробку генеративно-змагальної нейронної мережі для підвищення якості зображень. Створено програмний продукт, який дозволяє покращувати якість зображень з низькою роздільною здатністю, використовуючи сучасні методи глибокого навчання та архітектуру GAN.

Здобувач: _____ / Арсен ОСОКІН /

Керівник: _____ / Олена ГОРДА /

“ ___ ” _____ 2024р.

5.2 Детальний опис програмного продукту та послуг

Розроблений програмний продукт представляє собою систему підвищення якості зображень з використанням технології генеративних змагальних мереж (GAN). Система базується на архітектурі Real-ESRGAN Compact (SRVGGNet) та призначена для обробки цифрових зображень з метою покращення їх якості та збільшення роздільної здатності.

Основною функціональністю системи є можливість підвищення роздільної здатності зображень у чотири рази з одночасним покращенням їх якості. Система ефективно працює з різними типами зображень, включаючи фотографії, графічні матеріали та скановані документи. Особливу увагу приділено обробці зображень з типовими проблемами якості, такими як шум, розмиття та артефакти компресії.

Технічно система реалізована на базі фреймворку PyTorch з використанням сучасних методів глибокого навчання. Архітектура системи включає оптимізований генератор на основі залишкових блоків та дискримінатор з ефективною структурою для оцінки якості генерації. Система підтримує роботу як на CPU, так і на GPU, забезпечуючи оптимальну продуктивність на доступному обладнанні.

Важливою особливістю системи є її здатність до автоматичної адаптації під різні типи вхідних даних. Реалізовано механізми попередньої обробки зображень, які забезпечують коректну роботу з файлами різних форматів та розмірів. Система автоматично визначає оптимальні параметри обробки на основі характеристик вхідного зображення.

Інтерфейс системи реалізований через Jupyter Notebook, що забезпечує зручність використання та можливість інтерактивної роботи з результатами. Користувачі можуть легко завантажувати зображення, налаштовувати параметри обробки та переглядати результати в режимі реального часу. Додатково реалізовано функціонал для пакетної обробки множини зображень.

Система включає вбудовані механізми оцінки якості результатів з використанням метрик PSNR та SSIM, що дозволяє об'єктивно оцінювати якість обробки зображень. Реалізовано функціонал для візуального порівняння оригінальних та оброблених зображень, що спрощує аналіз результатів.

Особлива увага приділена оптимізації використання обчислювальних ресурсів. Реалізовано механізми управління пам'яттю та обчислювальними ресурсами, що забезпечують стабільну роботу системи навіть при обробці великих зображень. Підтримується режим змішаної точності обчислень для оптимізації продуктивності на GPU.

В рамках системи реалізовано функціонал для збереження та завантаження попередньо навчених моделей, що дозволяє користувачам вибирати оптимальні параметри обробки для різних типів зображень. Система підтримує можливість донавчання моделі на специфічних наборах даних для покращення результатів у конкретних сценаріях використання.

Технічна документація включає детальний опис усіх компонентів системи, інструкції з встановлення та налаштування, а також приклади використання для різних сценаріїв. Додатково надаються рекомендації щодо оптимальних налаштувань системи для різних типів завдань та апаратних конфігурацій.

Система розроблена з урахуванням можливості подальшого розширення функціональності та інтеграції з іншими системами обробки зображень. Модульна архітектура дозволяє легко додавати нові функції та адаптувати систему під специфічні потреби користувачів.

5.3 Аналіз ринку систем обробки зображень

Ринок систем обробки зображень демонструє стабільне зростання та розвиток, що підтверджується даними провідних аналітичних агентств. За даними

MarketsAndMarkets Research, глобальний ринок систем обробки зображень у 2023 році досяг 20,7 мільярдів доларів, з яких значна частина припадає на рішення з використанням технологій штучного інтелекту. Аналіз динаміки за останні п'ять років показує стабільне зростання сегменту AI-систем обробки зображень, який збільшився з 3,2 млрд доларів у 2019 році до 12,3 млрд доларів у 2023 році, демонструючи середньорічний темп росту понад 40% [38].

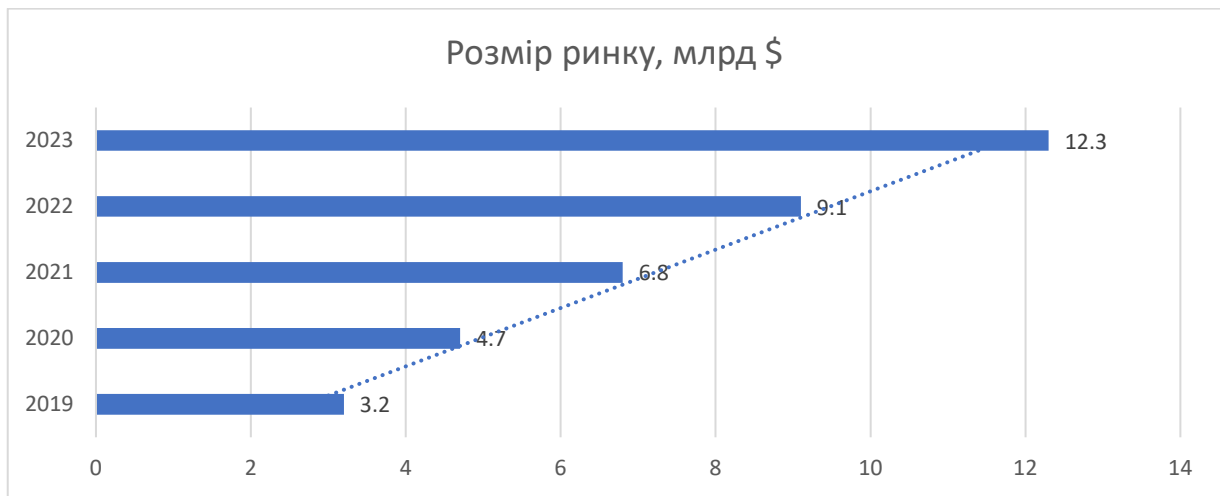


Рисунок 5.1 – Графік динаміки росту ринку в розглядаємій сфері

Структура світового ринку систем підвищення якості зображень відображає різноманітність потреб користувачів та галузей застосування. Найбільшу частку ринку займають професійні рішення, на які припадає 45% від загального обсягу, що становить 7,1 млрд доларів. Цей сегмент включає широкий спектр спеціалізованого програмного забезпечення для професійних фотографів, медичної візуалізації, архітектурного проектування та кіновиробництва. Корпоративний сегмент, який складає 30% ринку з обсягом 4,7 млрд доларів, орієнтований на потреби бізнесу в сфері електронної комерції, реклами та корпоративних медіа-систем. Сегмент користувацьких додатків, що займає 25% ринку з обсягом 3,9 млрд доларів, забезпечує потреби індивідуальних користувачів у базовій обробці зображень.

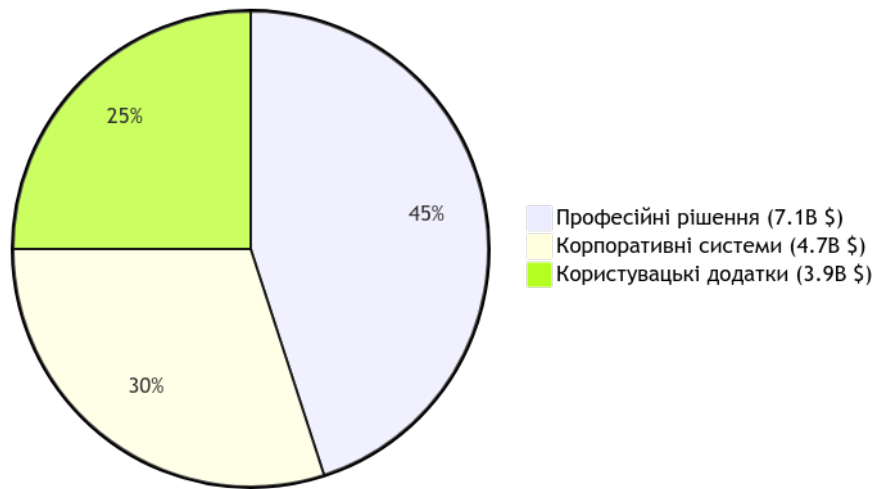


Рисунок 5.2 – Розподіл користувачів за сегментами

Географічний розподіл ринку демонструє домінування розвинених регіонів з високим рівнем технологічного розвитку. Північна Америка залишається найбільшим ринком з часткою 38% та обсягом 6,0 млрд доларів, де основним драйвером виступають Сполучені Штати. Європейський ринок, що складає 29% з обсягом 4,6 млрд доларів, демонструє стабільне зростання, особливо в країнах Західної Європи. Азіатсько-Тихоокеанський регіон, з часткою 24% та обсягом 3,8 млрд доларів, показує найвищі темпи росту, переважно завдяки розвитку ринків Китаю та Японії.

Важливим аспектом ринку є значна диференціація бюджетів компаній на системи обробки зображень. За даними IDC, великі корпорації інвестують від 250 тисяч до 1 мільйона доларів щорічно, тоді як бюджети середнього бізнесу коливаються в межах 50-250 тисяч доларів, а малого бізнесу - від 5 до 50 тисяч доларів. Така різниця обумовлена не лише масштабами діяльності, але й специфічними вимогами до функціональності та продуктивності систем [38].

Прогнози розвитку ринку від Allied Market Research вказують на подальше зростання до 28,5 млрд доларів до 2027 року із середньорічним темпом росту 16,2%. Основними факторами зростання виступають розвиток технологій штучного

інтелекту, зростаючий попит на автоматизацію обробки візуального контенту та розширення сфер застосування систем комп'ютерного зору [39].

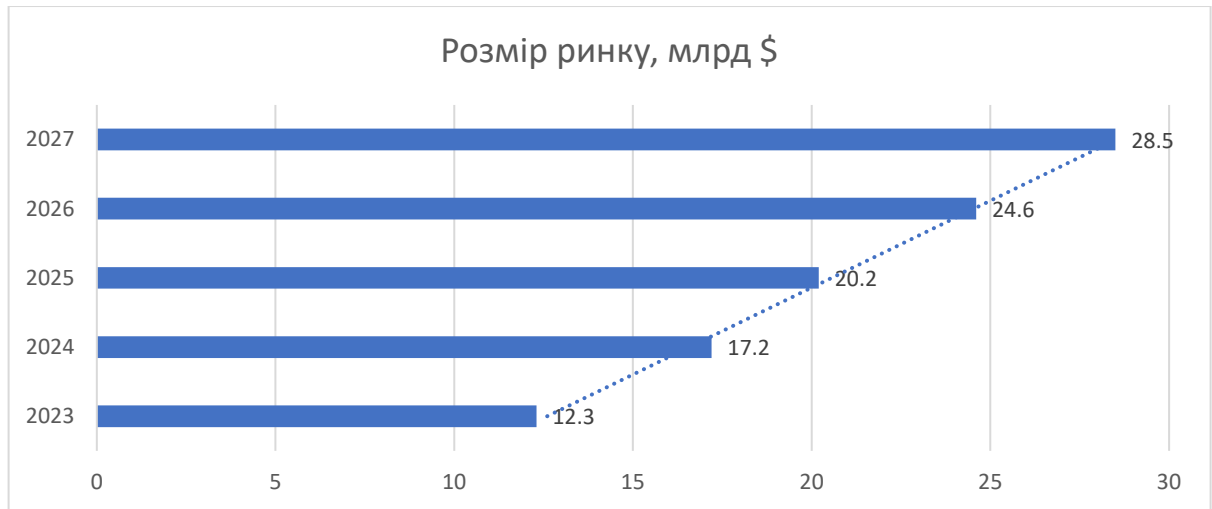


Рисунок 5.3 – Прогноз розвитку ринку на 2024-2027 роки

Аналіз тенденцій ринку вказує на зростаючий попит на спеціалізовані галузеві рішення, особливо в сферах медичної діагностики, промислового контролю якості та електронної комерції. Цей тренд підкреслює важливість розробки адаптивних систем, здатних враховувати специфічні вимоги різних галузей та сценаріїв використання.

Такий детальний аналіз ринку, підкріплений даними провідних аналітичних агентств, підтверджує значний потенціал для впровадження розробленої системи та визначає перспективні напрямки її розвитку відповідно до потреб різних сегментів користувачів та галузевих вимог.

5.4 Оцінка ринку збуту та потенційного попиту

На основі проведеного аналізу ринку систем обробки зображень можна здійснити оцінку потенційного ринку збуту та попиту на розроблену модель підвищення якості зображень на базі архітектури Real-ESRGAN Compact.

Загальний обсяг цільового ринку для систем покращення якості зображень оцінюється у 2.5 мільярди доларів США станом на 2024 рік, з прогнозованим

зростанням до 4 мільярдів доларів до 2027 року. Цей ріст підтримується активним розвитком електронної комерції, соціальних мереж та зростаючими потребами в якісному візуальному контенті.

Первинна сегментація ринку виділяє наступні ключові сегменти за професійним спрямуванням. Професійний сегмент, що включає фотографів, дизайнерів та спеціалістів з обробки зображень, оцінюється у 35% від загального ринку. Цей сегмент характеризується високою платоспроможністю та готовністю інвестувати в якісні інструменти для роботи з зображеннями.

Корпоративний сегмент, що охоплює онлайн-магазини, рекламні агентства та медіа-компанії, складає приблизно 40% ринку. Цей сегмент демонструє найвищі темпи зростання, що пов'язано з розвитком електронної комерції та цифрового маркетингу. Особливий інтерес представляють середні та малі підприємства, які потребують автоматизованих рішень для обробки зображень.

Науково-дослідний та освітній сегмент становить близько 15% ринку. Сюди входять медичні установи, дослідницькі інститути та навчальні заклади. Цей сегмент характеризується специфічними вимогами до якості обробки зображень та потребує спеціалізованих рішень.

Індивідуальний сегмент, що включає аматорів та приватних користувачів, оцінюється у 10% ринку. Хоча цей сегмент має найнижчу платоспроможність, він демонструє стабільне зростання завдяки популяризації соціальних мереж та зростаючому інтересу до якісного візуального контенту.

За географічним розподілом найбільший потенціал демонструють ринки Північної Америки (35% загального обсягу) та Європи (30%). Азійсько-Тихоокеанський регіон показує найвищі темпи зростання і складає 25% ринку. Решта світу становить 10% загального ринку [40].

Аналіз цінової сегментації показує, що найбільший попит спостерігається в середньому ціновому сегменті, де користувачі готові платити за якісні рішення, але

чутливі до співвідношення ціна/якість. Преміум-сегмент, орієнтований на професійних користувачів, демонструє стабільний попит незалежно від цінової політики.

Оцінка потенційного попиту базується на аналізі кількості активних користувачів систем обробки зображень та темпів зростання ринку. За консервативними оцінками, потенційна база користувачів для розробленої системи може скласти від 50 до 100 тисяч користувачів протягом першого року після виходу на ринок.

Важливим фактором при оцінці попиту є зростаюча тенденція до використання рішень на базі штучного інтелекту. Дослідження показують, що користувачі все більше віддають перевагу автоматизованим системам обробки зображень, які не вимагають глибоких технічних знань.

Аналіз сезонності попиту показує підвищену активність у періоди святкових розпродажів та на початку нових маркетингових кампаній. Це особливо характерно для корпоративного сегменту, де потреба в обробці зображень прямо пов'язана з маркетинговою активністю.

Оцінка потенціалу зростання ринку вказує на можливість збільшення частки ринку через розширення функціональності системи та адаптацію під специфічні потреби різних сегментів користувачів. Особливий потенціал має розробка галузевих рішень для медицини, архітектури та наукових досліджень.

5.5 Аналіз конкурентного середовища

На ринку систем підвищення якості зображень представлені різні програмні рішення, які можна розглядати як потенційних конкурентів нашої системи на базі Real-ESRGAN Compact. Проведений аналіз конкурентного середовища дозволяє виділити ключові особливості та переваги розробленого рішення.

Серед основних конкурентів можна відзначити Adobe Super Resolution, яка інтегрована в продукти Adobe та пропонує функціонал підвищення роздільної здатності зображень. Це рішення характеризується високою якістю результатів та зручною інтеграцією з екосистемою Adobe, проте має високу вартість та вимагає підписки на продукти Adobe [41].

Тораз Gigapixel AI представляє собою спеціалізоване рішення для збільшення роздільної здатності зображень. Продукт демонструє високу якість результатів та має зручний користувацький інтерфейс, але характеризується високою вартістю ліцензії та значними вимогами до обчислювальних ресурсів [42].

Let's Enhance пропонує онлайн-сервіс для покращення якості зображень з використанням штучного інтелекту. Перевагами є доступність через веб-інтерфейс та простота використання, проте сервіс має обмеження на розмір файлів та кількість оброблених зображень у безкоштовній версії [43].

На відкритому ринку також представлені різні реалізації ESRGAN та інших архітектур для підвищення якості зображень, проте більшість з них вимагає технічних знань для встановлення та налаштування, що обмежує їх доступність для широкого кола користувачів.

Розроблена нами система має ряд конкурентних переваг. По-перше, використання оптимізованої архітектури Real-ESRGAN Compact забезпечує високу якість результатів при менших вимогах до обчислювальних ресурсів порівняно з конкурентами. По-друге, реалізація через Jupyter Notebook надає гнучкість у налаштуванні та можливість інтеграції з різними робочими процесами.

Важливою перевагою є відкрита архітектура системи, що дозволяє користувачам адаптувати та розширювати функціонал під свої потреби. Це особливо важливо для наукового та дослідницького сегментів ринку, де потрібна можливість модифікації алгоритмів обробки.

Система також виділяється ефективним використанням обчислювальних ресурсів завдяки реалізації оптимізованих алгоритмів та підтримці змішаної точності обчислень. Це дозволяє досягати високої продуктивності навіть на середньому апаратному забезпеченні.

В порівнянні з конкурентами, наша система пропонує більш гнучку модель ліцензування та не вимагає постійної підписки. Це робить її більш доступною для малого та середнього бізнесу, а також для індивідуальних користувачів.

Додатковою перевагою є можливість пакетної обробки зображень та автоматизації процесів, що особливо важливо для корпоративних користувачів. Система також демонструє кращу продуктивність при обробці великих наборів даних порівняно з онлайн-сервісами.

Проведений аналіз також виявив потенційні напрямки вдосконалення системи. Зокрема, існує потреба в розробці більш дружнього користувацького інтерфейсу та покращенні документації для нетехнічних користувачів.

Важливо відзначити, що ринок систем обробки зображень динамічно розвивається, і конкуренти постійно вдосконалюють свої продукти. Це вимагає постійного моніторингу ринку та оперативного впровадження нових функцій та вдосконалень.

В цілому, розроблена система з деякими дороботками може зайняти певну нішу на ринку, пропонуючи оптимальне співвідношення якості результатів, гнучкості налаштувань та вимог до ресурсів.

5.6 Стратегія маркетингу

На основі проведеного аналізу ринку та конкурентного середовища буде розроблена комплексна маркетингова стратегія для просування системи підвищення якості зображень на базі Real-ESRGAN Compact.

Позиціонування продукту буде базуватися на трьох ключових перевагах: висока якість результатів обробки зображень, оптимізоване використання обчислювальних ресурсів та гнучкість налаштування під потреби користувача. Система буде позиціонуватися як професійне рішення для автоматизованого покращення якості зображень, доступне широкому колу користувачів.

Цінова політика буде побудована на основі моделі диференційованого ціноутворення з кількома рівнями ліцензій. Базова версія системи буде доступна за помірною ціною та включатиме основний функціонал для індивідуальних користувачів. Професійна версія запропонує розширені можливості налаштування та оптимізації для корпоративних клієнтів. Преміум-версія надасть повний доступ до всіх функцій та можливість кастомізації під специфічні потреби.

Для наукових установ та освітніх закладів буде передбачена спеціальна академічна ліцензія з суттєвими знижками. Додатково буде впроваджена програма тестового періоду, що дозволить потенційним клієнтам оцінити можливості системи перед придбанням.

Система буде просуватися через кілька ключових каналів. Основним каналом стане спеціалізований веб-сайт з детальним описом можливостей системи, прикладами результатів обробки та документацією. Буде створена галерея прикладів використання системи в різних сценаріях для демонстрації її ефективності.

Важливим напрямком просування стане співпраця з професійними спільнотами фотографів, дизайнерів та розробників. Будуть організовані онлайн-демонстрації можливостей системи та навчальні вебінари для різних цільових аудиторій.

Для залучення корпоративних клієнтів буде розроблена програма прямих продажів з можливістю індивідуального налаштування системи під потреби конкретного бізнесу. Буде передбачена технічна підтримка та консультації щодо інтеграції системи в існуючі робочі процеси.

Контент-маркетинг буде включати регулярні публікації технічних статей, порівняльних аналізів та кейс-стаді на профільних ресурсах. Буде створений блог з описом технічних особливостей системи та прикладами її практичного застосування.

В соціальних мережах буде проводитися активна кампанія з демонстрацією результатів роботи системи та поширенням корисного контенту для цільової аудиторії. Особлива увага буде приділена візуальним платформам, де можна ефективно продемонструвати можливості системи.

Партнерська програма передбачатиме співпрацю з розробниками суміжних продуктів та інтеграторами. Будуть розроблені API та документація для інтеграції системи з іншими рішеннями для обробки зображень.

Для підтримки користувачів буде створена система онлайн-навчання з відеоуроками та інтерактивними посібниками. Також буде запущено форум для обміну досвідом та обговорення особливостей використання системи.

Окрема увага буде приділена локалізації маркетингових матеріалів для різних регіональних ринків. Буде забезпечено переклад документації та інтерфейсу на основні мови цільових ринків.

Ефективність маркетингової стратегії буде оцінюватися за ключовими показниками: кількість активних користувачів, рівень конверсії, задоволеність клієнтів та обсяг продажів у різних сегментах ринку.

Стратегія просування буде регулярно оновлюватися на основі аналізу зворотного зв'язку від користувачів та змін ринкової ситуації. Це дозволить підтримувати конкурентоспроможність продукту та забезпечити його стабільне просування на ринку.

5.7 План розробки та впровадження системи

План розробки та впровадження системи підвищення якості зображень на базі Real-ESRGAN Compact буде структурований з особливим акцентом на створення власного набору даних для навчання моделі.

Першим критичним етапом, який триватиме три місяці, стане створення власного набору даних для тренування моделі. Буде організовано професійну фотозйомку різноманітних об'єктів та сцен з використанням високоякісного обладнання. Фотозйомка охопить широкий спектр умов освітлення, текстур та деталей. Всі зображення будуть створюватися з чітким дотриманням авторських прав та з можливістю комерційного використання. Паралельно буде розроблено систему автоматизованої генерації пар зображень різної роздільної здатності з контрольованими параметрами якості.

Етап обробки та підготовки набору даних займе два місяці. На цьому етапі буде створено систему каталогізації зображень, розроблено процедури контролю якості та валідації даних. Особлива увага буде приділена створенню метаданих для кожного зображення, що включатимуть інформацію про умови зйомки та технічні параметри. Набір даних буде розділено на тренувальну, валідаційну та тестову частини з урахуванням рівномірного розподілу різних типів зображень та умов зйомки.

Початковий етап розробки інфраструктури проекту триватиме півтора місяця. В цей період буде налаштовано середовище розробки в Anaconda, створено систему контролю версій та налаштовано процеси автоматизації розробки. Буде реалізовано систему управління наборами даних та їх версіями, що забезпечить можливість розширення набору даних у майбутньому.

Розробка та оптимізація архітектури моделі займе три місяці. Буде реалізовано оптимізовану версію генератора та дискримінатора, адаптовану під особливості створеного набору даних. Особлива увага буде приділена розробці механізмів

обробки різних типів зображень та умов зйомки, представлених у власному наборі даних. Буде проведено експерименти з різними модифікаціями архітектури для досягнення оптимальних результатів.

Етап тренування та валідації моделі триватиме два з половиною місяці. В цей час буде проведено повний цикл навчання моделі на створеному наборі даних. Процес навчання буде включати постійний моніторинг якості результатів та адаптацію параметрів під особливості власного набору даних. Буде реалізовано систему автоматичної оцінки якості результатів та збереження найкращих версій моделі.

Розробка користувацького інтерфейсу та документації займе два місяці. На цьому етапі буде створено зручний інтерфейс у форматі Jupyter Notebook, оптимізований під роботу з різними типами зображень. Буде підготовлено детальну документацію з прикладами використання системи, включаючи специфічні випадки та рекомендації щодо обробки різних типів зображень.

Етап тестування та оптимізації триватиме два з половиною місяці. В цей період буде проведено комплексне тестування системи на різноманітних типах зображень, включаючи випадки, не представлені в тренувальному наборі. Буде проведено оптимізацію продуктивності та стабільності роботи системи з урахуванням результатів тестування.

Підготовка до релізу займе півтора місяця. В цей час буде проведено фінальне тестування, підготовлено дистрибутиви системи та створено інструменти для автоматичного оновлення. Особлива увага буде приділена захисту інтелектуальної власності та ліцензуванню створеного програмного продукту.

Для реалізації проекту будуть потрібні: професійне фотографічне обладнання для створення набору даних, сервер для розробки та тестування з GPU NVIDIA (мінімум 8 ГБ відеопам'яті), робочі станції для команди розробників та доступ до хмарних обчислювальних ресурсів для масштабного тренування моделі.

Успішна реалізація цього плану дозволить створити оновлену модель підвищення якості зображень, що вже базується на власному наборі даних та враховує специфічні потреби цільової аудиторії та має можливість комерційного використання без обмежень, пов'язаних з ліцензуванням сторонніх даних.

5.8 Організаційний план реалізації проекту

Для успішної реалізації проекту з розробки системи підвищення якості зображень буде сформована професійна команда з чітким розподілом обов'язків та зон відповідальності.

Керівник проекту буде відповідати за загальну координацію роботи команди, управління ресурсами та комунікацію з зацікавленими сторонами. До його обов'язків входить стратегічне планування, контроль виконання етапів проекту та прийняття ключових рішень щодо розвитку продукту. Керівник проекту також забезпечуватиме взаємодію з потенційними інвесторами та партнерами.

Технічний керівник візьме на себе відповідальність за архітектуру системи та технічні рішення. У його компетенцію входить вибір технологій, оптимізація процесів розробки та забезпечення якості коду. Він координуватиме роботу команди розробників та прийматиме рішення щодо технічних аспектів реалізації проекту.

Для створення власного набору даних буде залучено професійного фотографа та асистента фотографа. Їхніми завданнями стануть організація та проведення фотозйомок, забезпечення різноманітності умов зйомки та типів об'єктів, а також первинна обробка отриманих зображень. Вони працюватимуть у тісній співпраці з командою розробки для забезпечення відповідності створюваного контенту технічним вимогам проекту.

Два спеціалісти з машинного навчання зосередяться на розробці та оптимізації архітектури нейронної мережі. Вони відповідатимуть за реалізацію генератора та дискримінатора, налаштування параметрів навчання та проведення експериментів з

різними модифікаціями архітектури. До їхніх обов'язків також входить аналіз результатів навчання та оптимізація продуктивності моделі.

Спеціаліст з обробки даних буде відповідати за створення системи підготовки та валідації даних. Його завданнями стануть розробка процедур попередньої обробки зображень, створення системи аугментації даних та забезпечення ефективного процесу завантаження даних для навчання моделі.

Python-розробник забезпечуватиме реалізацію базової інфраструктури проекту, розробку користувацького інтерфейсу та інтеграцію різних компонентів системи. Він також відповідатиме за створення інструментів автоматизації та системи контролю версій.

Тестувальник буде залучений для проведення комплексного тестування системи, включаючи функціональне тестування, оцінку продуктивності та перевірку якості результатів обробки зображень. Він також займатиметься документуванням виявлених проблем та верифікацією їх вирішення.

Технічний письменник візьме на себе створення документації, навчальних матеріалів та прикладів використання системи. Він забезпечуватиме якісний опис функціональності та створення зрозумілих інструкцій для користувачів різного рівня підготовки.

Для забезпечення ефективної комунікації в команді буде впроваджено систему регулярних зустрічей: щоденні стендапи для синхронізації роботи, щотижневі планування для обговорення прогресу та щомісячні ретроспективи для аналізу результатів та коригування планів.

Робота команди буде організована за гнучкою методологією розробки з двотижневими спринтами. Це дозволить забезпечити регулярні поставки результатів та можливість оперативного реагування на зміни вимог чи виявлені проблеми.

В процесі реалізації проекту буде забезпечено можливості для професійного розвитку членів команди через участь у профільних конференціях, проходження навчальних курсів та обмін досвідом всередині команди.

При необхідності для виконання специфічних завдань будуть залучатися зовнішні консультанти та експерти, зокрема для оптимізації процесів машинного навчання та вирішення складних технічних завдань.

Система мотивації команди включатиме як фінансову складову, так і можливості професійного зростання, участь у розподілі прибутку від проекту та гнучкий графік роботи.

Такий підхід до організації роботи команди забезпечить ефективну реалізацію проекту з дотриманням встановлених термінів та вимог до якості кінцевого продукту.

5.9 Юридичне забезпечення проекту

Юридичне забезпечення проекту з розробки системи підвищення якості зображень буде реалізовано відповідно до чинного законодавства України, зокрема Закону України "Про авторське право і суміжні права" та Закону України "Про охорону прав на винаходи і корисні моделі".

Першим етапом юридичного забезпечення стане реєстрація юридичної особи у формі товариства з обмеженою відповідальністю відповідно до Закону України "Про товариства з обмеженою та додатковою відповідальністю". Це забезпечить оптимальну форму ведення господарської діяльності та захисту інтересів учасників проекту.

Особлива увага буде приділена захисту інтелектуальної власності та правовій чистоті програмного коду. На першому етапі буде проведено детальний аудит використаних компонентів та бібліотек на предмет їх ліцензійної чистоти. Всі сторонні компоненти будуть перевірені на відповідність умовам ліцензування та можливість комерційного використання. У разі виявлення компонентів з

обмежувальними ліцензіями, буде проведено їх заміну або розробку власних альтернативних рішень.

Архітектура системи та програмний код будуть проаналізовані на предмет можливих порушень патентних прав та авторських прав третіх осіб. При необхідності буде проведено модифікацію технічних рішень для уникнення потенційних правових конфліктів. Після завершення аудиту та необхідних модифікацій буде проведено реєстрацію авторського права на програмний код та архітектуру системи відповідно до статті 11 Закону України "Про авторське право і суміжні права".

Додатково буде подано заявку на отримання патенту на корисну модель для захисту унікальних технічних рішень, використаних у системі. Перед подачею заявки буде проведено патентний пошук для перевірки новизни та патентоспроможності розроблених рішень.

Для захисту створеного набору даних буде оформлено авторські права на всі фотографічні твори згідно зі статтею 8 Закону України "Про авторське право і суміжні права". З фотографами та іншими авторами буде укладено договори про передачу майнових прав на створені ними об'єкти інтелектуальної власності.

У відповідності до Закону України "Про захист персональних даних" буде розроблено політику конфіденційності та механізми захисту персональних даних користувачів системи. Буде впроваджено процедури обробки та зберігання даних, що відповідають вимогам GDPR, для можливості виходу на європейський ринок.

Система ліцензування програмного продукту буде розроблена з урахуванням різних типів користувачів. Для кінцевих користувачів буде створено ліцензійну угоду, що регулюватиме умови використання системи та обмеження відповідальності розробника. Для корпоративних клієнтів будуть розроблені окремі ліцензійні умови з можливістю адаптації під специфічні потреби.

Особлива увага буде приділена правовому регулюванню використання технологій машинного навчання та штучного інтелекту відповідно до рекомендацій Міністерства цифрової трансформації України та міжнародних стандартів у цій галузі.

Для захисту комерційної таємниці буде розроблено систему внутрішньої документації та укладено угоди про нерозголошення з усіма співробітниками та партнерами проекту згідно зі статтею 505 Цивільного кодексу України.

Буде проведено реєстрацію торговельної марки відповідно до Закону України "Про охорону прав на знаки для товарів і послуг" для захисту бренду та маркетингових матеріалів проекту.

Для забезпечення правової підтримки проекту буде залучено юридичного консультанта, який забезпечить супровід всіх аспектів діяльності, включаючи підготовку договорів, консультації з питань інтелектуальної власності та вирішення можливих спорів.

У рамках співпраці з партнерами та клієнтами буде розроблено типові форми договорів, що враховують специфіку надання послуг у сфері програмного забезпечення та обробки даних відповідно до Цивільного кодексу України.

Для захисту інтересів інвесторів будуть розроблені відповідні юридичні механізми, включаючи угоди про конфіденційність та інвестиційні договори, що відповідають вимогам Закону України "Про інвестиційну діяльність".

Буде забезпечено відповідність діяльності проекту вимогам податкового законодавства України, зокрема щодо оподаткування доходів від реалізації програмного забезпечення та надання послуг.

Такий комплексний підхід до юридичного забезпечення проекту дозволить не тільки захистити права інтелектуальної власності та мінімізувати ризики порушення прав третіх осіб, але й забезпечити надійний захист інтересів всіх учасників проекту відповідно до вимог українського законодавства.

5.10 Фінансовий план проекту

Фінансовий план проекту з розробки системи підвищення якості зображень буде розроблено на період 24 місяців, охоплюючи етапи розробки, впровадження та початкової комерціалізації продукту.

Капітальні витрати проекту складатимуться з придбання серверного обладнання з GPU NVIDIA RTX 4090 вартістю 120,000 грн, фотографічного обладнання для створення набору даних за 180,000 грн, робочих станцій для команди розробників (5 комплектів) за 250,000 грн та ліцензій на програмне забезпечення вартістю 150,000 грн.

Щомісячні операційні витрати включатимуть заробітну плату команди розробників у розмірі 410,000 грн, оренду офісного приміщення за 35,000 грн, комунальні послуги - 8,000 грн, інтернет та зв'язок - 5,000 грн, хмарні обчислювальні ресурси - 25,000 грн, маркетингові витрати - 40,000 грн, юридичний супровід - 15,000 грн та адміністративні витрати - 20,000 грн.

Одноразові витрати на запуск складатимуться з реєстрації юридичної особи за 15,000 грн, патентування та реєстрації авторських прав - 45,000 грн, розробки брендингу та маркетингових матеріалів - 80,000 грн, створення набору даних (включаючи оплату фотографам) - 200,000 грн.

Модель монетизації базуватиметься на трьох типах ліцензій: базова ліцензія вартістю 2,500 грн/міс, професійна ліцензія - 6,000 грн/міс та корпоративна ліцензія від 15,000 грн/міс.

У перші 6 місяців після запуску очікуватиметься продаж 50-100 базових та 20-30 професійних ліцензій. Протягом 7-12 місяців передбачатиметься зростання на 150% для базових та 100% для професійних ліцензій. Починаючи з другого року, планується стабільне зростання на 50% щоквартально.

Розрахунок собівартості включатиме прямі витрати на обслуговування клієнта (15% від вартості ліцензії), витрати на технічну підтримку (10% від вартості ліцензії),

витрати на розвиток продукту (25% від доходу) та маркетингові витрати (20% від доходу).

Точка беззбитковості очікуватиметься через 14-16 місяців після запуску проекту при досягненні понад 300 активних базових ліцензій, більше 100 професійних ліцензій та понад 10 корпоративних клієнтів.

У перший рік роботи очікуватиметься збиток до 1.5 млн грн. На другий рік планується вихід на прибуток з очікуваною рентабельністю 25-30%, а на третій рік - зростання рентабельності до 40-45%.

Основні фінансові показники проекту передбачатимуть загальний обсяг необхідних інвестицій у розмірі 3.5 млн грн, очікуваний термін повернення інвестицій 24-28 місяців, плановану внутрішню норму доходності (IRR) 35% та чисту приведену вартість (NPV) при ставці дисконтування 20% - 2.8 млн грн.

Для забезпечення фінансової стабільності проекту буде створено резервний фонд у розмірі 20% від загального обсягу інвестицій, що дозволить компенсувати можливі коливання доходів та непередбачені витрати на початкових етапах реалізації проекту.

Система фінансового контролю включатиме щомісячний моніторинг ключових показників ефективності, квартальний перегляд фінансового плану та щорічний аудит фінансової діяльності, що дозволить оперативно реагувати на відхилення від планових показників та вносити необхідні корективи в фінансову стратегію проекту.

5.11 Стратегія фінансування розробки

Стратегія фінансування проекту з розробки системи підвищення якості зображень буде реалізована через комбінацію різних джерел фінансування для оптимального розподілу ризиків та забезпечення стабільного розвитку проекту.

Загальна потреба в інвестиціях складає 3.5 млн грн, які будуть залучені поетапно відповідно до стадій розвитку проекту. Структура фінансування передбачає наступний розподіл:

Власні кошти засновників становитимуть 25% від загальної суми (875,000 грн), які будуть внесені на початковому етапі проекту. Ці кошти будуть спрямовані на формування статутного капіталу, реєстрацію юридичної особи та початкові організаційні витрати.

Залучення приватних інвестицій планується в обсязі 40% від необхідної суми (1,400,000 грн). Ці кошти будуть використані для придбання обладнання, створення набору даних та фінансування початкового етапу розробки. В якості потенційних інвесторів розглядаються представники IT-індустрії та венчурні інвестори з досвідом у сфері машинного навчання та комп'ютерного зору.

Грантове фінансування планується отримати через участь у програмах підтримки інноваційних проектів, зокрема в рамках Українського фонду стартапів та міжнародних програм підтримки технологічних проектів. Очікуваний обсяг грантового фінансування складає 15% від загальної суми (525,000 грн).

Кредитні кошти будуть залучені в обсязі 20% (700,000 грн) через програми підтримки малого та середнього бізнесу з пільговими умовами кредитування. Кредитні ресурси планується використовувати для фінансування операційної діяльності та масштабування проекту.

Для оптимізації фінансових потоків буде впроваджено систему поетапного фінансування. На першому етапі (0-6 місяців) фінансування здійснюватиметься за рахунок власних коштів засновників та першого траншу приватних інвестицій, з фокусом на розробку базової інфраструктури та створення набору даних.

На другому етапі (7-12 місяців) буде залучено другий транш інвестицій та грантове фінансування, які спрямовуватимуться на розробку core-функціоналу та тестування.

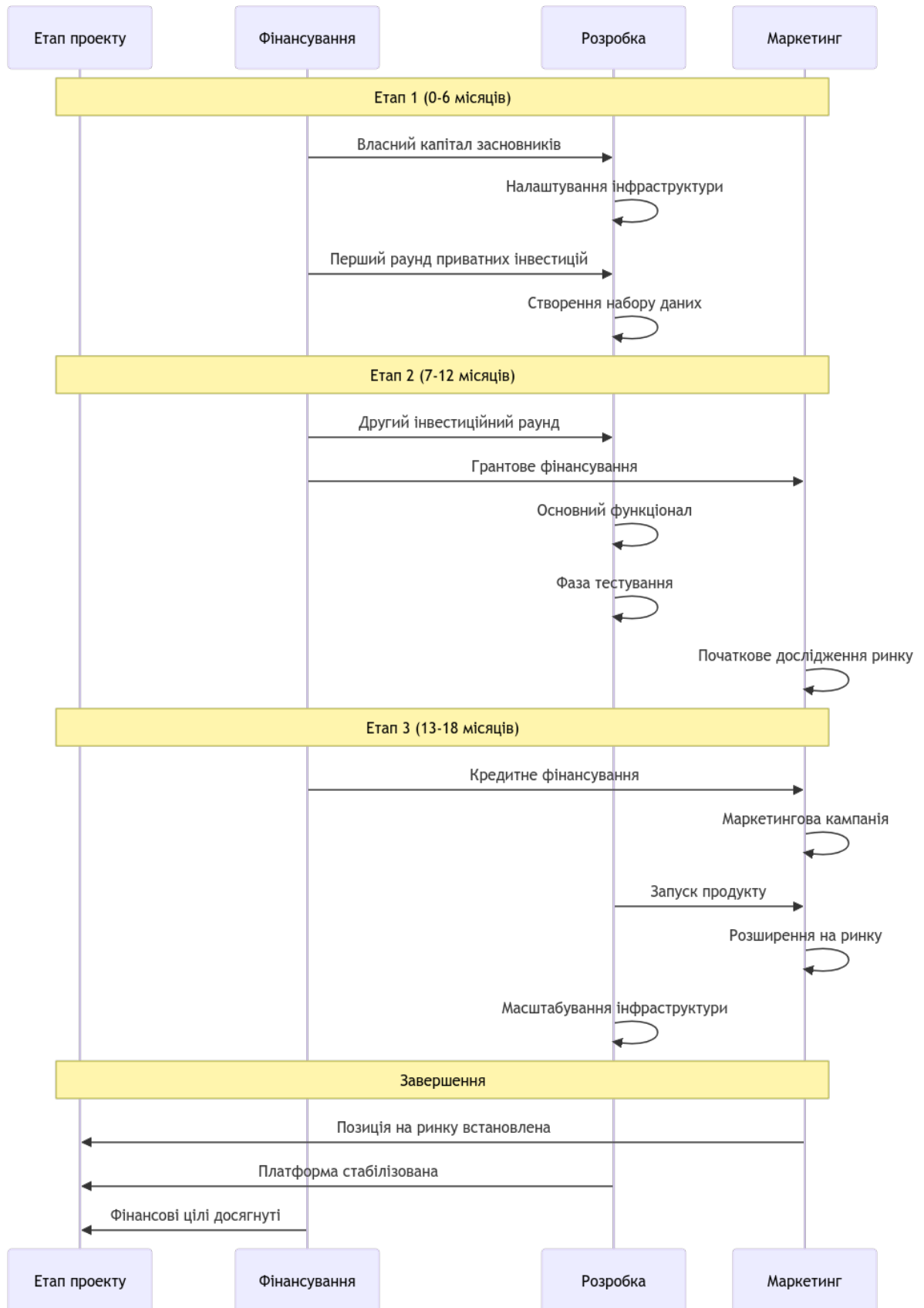


Рисунок 5.4 – Стратегічний план розвитку проекту

Третій етап (13-18 місяців) передбачатиме залучення кредитних коштів для фінансування маркетингової активності та масштабування.

Система фінансового контролю включатиме щомісячний моніторинг витрат, квартальний аналіз ефективності використання коштів та регулярну звітність перед інвесторами.

Особлива увага приділятиметься створенню фінансових резервів: операційного резерву (15% від залучених коштів), резерву на розвиток (10%) та антикризового резерву (5%).

План повернення інвестицій передбачатиме виплати інвесторам починаючи з 18-го місяця проекту, погашення кредиту рівними частинами протягом 24 місяців та реінвестування 30% прибутку в розвиток проекту.

Для мінімізації фінансових ризиків планується впровадження диверсифікації джерел фінансування, страхування ключових ризиків та хеджування валютних ризиків при міжнародних операціях.

Додатково розглядатимуться можливості залучення стратегічних партнерів з індустрії для спільного фінансування окремих напрямків розвитку проекту, що дозволить не лише отримати додаткове фінансування, але й забезпечити вихід на нові ринки.

У разі успішної реалізації початкових етапів проекту та досягнення планових показників, розглядатиметься можливість залучення венчурного фінансування для масштабування бізнесу та виходу на міжнародні ринки.

Така стратегія фінансування забезпечить необхідну гнучкість у використанні фінансових ресурсів та дозволить ефективно управляти розвитком проекту на всіх етапах його реалізації. При цьому диверсифікація джерел фінансування знизить залежність від окремих інвесторів та зменшить фінансові ризики проекту.

5.12 Оцінка ризиків проекту

В рамках реалізації проекту з розробки системи підвищення якості зображень проведено комплексний аналіз потенційних ризиків та розроблено стратегії їх мінімізації. Всі виявлені ризики можна класифікувати за кількома ключовими категоріями з відповідними стратегіями управління.

Технічні ризики представляють першу категорію потенційних загроз для проекту. Основним технічним ризиком є можлива неефективність розробленої архітектури нейронної мережі, що може призвести до затримки виходу продукту на ринок на 2-3 місяці та спричинити додаткові витрати на розробку до 200,000 грн. Для мінімізації цього ризику буде впроваджено поетапне тестування архітектури та створення декількох альтернативних варіантів реалізації. Також існує ризик недостатньої якості власного набору даних для навчання моделі, що може знизити ефективність системи на 15-20%. Цей ризик буде контролюватися через впровадження суворих критеріїв якості при створенні датасету та постійний моніторинг результатів навчання.

Ринкові ризики становлять другу важливу категорію. Поява сильних конкурентів з аналогічними рішеннями може призвести до зниження прогнозованих продажів на 25-30%. Для управління цим ризиком буде проводитися постійний моніторинг ринку та підтримка конкурентних переваг через розробку унікальних функцій. Ризик недостатнього попиту на преміум-версії продукту може спричинити зниження доходу на 20-25%, що вимагатиме впровадження гнучкої цінової політики та розробки додаткових функцій для преміум-сегменту.

Фінансові ризики включають можливість перевищення бюджету розробки через непередбачені технічні складності, що може збільшити витрати на 30-40%. Для управління цим ризиком буде створено фінансовий резерв та впроваджено поетапне фінансування розробки. Ризик затримки виходу на точку беззбитковості може

збільшити період окупності на 6-8 місяців, що вимагатиме створення додаткових джерел доходу та оптимізації операційних витрат.

Операційні ризики пов'язані з можливими затримками розробки через проблеми в команді, що може відтермінувати запуск проекту на 2-3 місяці. Цей ризик буде мінімізовано через впровадження чітких процедур управління проектом та системи мотивації команди. Також існує ризик проблем з інтеграцією системи у робочі процеси клієнтів, що може знизити конверсію продажів на 20-25%. Для управління цим ризиком буде створено детальну документацію та навчальні матеріали.

Юридичні ризики включають можливість порушення прав інтелектуальної власності третіх осіб, що може призвести до необхідності виплати штрафів та модифікації продукту. Цей ризик буде контролюватися через ретельну перевірку всіх компонентів системи на відповідність ліцензійним вимогам.

Для ефективного управління всіма виявленими ризиками буде впроваджено систему регулярного моніторингу та оцінки ризиків. Щоквартально буде проводитися перегляд та оновлення оцінки ризиків для забезпечення актуальності та ефективності заходів з управління ризиками.

У рамках системи управління ризиками буде створено резервний фонд для оперативного реагування на виникаючі загрози та впроваджено систему раннього попередження для своєчасного виявлення потенційних проблем. Це дозволить мінімізувати негативний вплив ризиків на успішну реалізацію проекту та забезпечити досягнення поставлених цілей у встановлені терміни та в межах запланованого бюджету.

5.13 Страхування та методи мінімізації ризиків

На основі проведеної оцінки ризиків проекту буде впроваджено комплексну систему страхування та методів мінімізації ризиків. Ця система охопить всі ключові аспекти проекту та забезпечить його стабільну реалізацію.

В рамках технічного страхування буде укладено договір страхування обладнання та програмного забезпечення. Страхове покриття включатиме захист від фізичного пошкодження серверного обладнання, втрати даних та кіберризиків. Додатково буде впроваджено систему резервного копіювання даних з використанням географічно розподілених серверів для мінімізації ризику втрати критично важливої інформації.

Для мінімізації ризиків, пов'язаних з розробкою програмного забезпечення, буде впроваджено систему контролю версій з автоматичним резервним копіюванням коду. Буде створено процедури регулярного тестування та валідації всіх компонентів системи. Важливим елементом стане впровадження системи моніторингу продуктивності та автоматичного виявлення потенційних проблем.

Фінансові ризики будуть мінімізовані через створення резервного фонду в розмірі 25% від загального бюджету проекту. Буде укладено договір страхування професійної відповідальності, який покриватиме можливі збитки, пов'язані з помилками в роботі системи. Додатково буде впроваджено систему фінансового моніторингу з щотижневим аналізом ключових показників.

У сфері захисту інтелектуальної власності буде проведено страхування патентних ризиків та ризиків порушення авторських прав. Це включатиме юридичний захист у випадку претензій третіх осіб та покриття можливих витрат на судові розгляди. Регулярні юридичні аудити допоможуть виявляти потенційні проблеми на ранніх стадіях.

Операційні ризики будуть мінімізовані через впровадження системи управління якістю та регулярні тренінги персоналу. Буде розроблено детальні інструкції та процедури для всіх критичних процесів. Система моніторингу продуктивності дозволить оперативно виявляти та усувати можливі проблеми в роботі системи.

Для захисту від ринкових ризиків буде створено систему постійного моніторингу ринку та конкурентного середовища. Регулярні опитування користувачів

та аналіз зворотного зв'язку дозволять вчасно адаптувати продукт до змін потреб ринку. Гнучка цінова політика забезпечить можливість швидкого реагування на зміни ринкової ситуації.

Особлива увага буде приділена страхуванню відповідальності перед клієнтами. Це включатиме покриття можливих збитків, пов'язаних з помилками в роботі системи або втратою даних. Буде розроблено чіткі процедури обробки претензій та компенсації можливих збитків.

Для мінімізації репутаційних ризиків буде впроваджено систему управління відносинами з клієнтами та активну комунікаційну стратегію. Регулярний моніторинг згадувань у медіа та соціальних мережах дозволить оперативно реагувати на можливі проблеми.

В рамках управління персоналом буде впроваджено систему мотивації та розвитку співробітників, що дозволить мінімізувати ризики втрати ключових спеціалістів. Буде забезпечено страхування здоров'я співробітників та створено комфортні умови праці.

Таким чином, комплексний підхід до страхування та мінімізації ризиків забезпечить стабільний розвиток проекту та його успішну реалізацію. Регулярний перегляд та оновлення стратегій управління ризиками дозволить підтримувати ефективність системи захисту на високому рівні.

Реалізація всіх запланованих заходів зі страхування та мінімізації ризиків завершить формування цілісної системи управління проектом, забезпечивши його надійну основу для подальшого розвитку та масштабування. Це дозволить не тільки захистити інвестиції та інтереси всіх учасників проекту, але й створити стійку базу для довгострокового успіху на ринку систем обробки зображень.

ВИСНОВОК

В результаті виконаної роботи було розроблено та реалізовано ефективну модель підвищення якості зображень на базі архітектури Real-ESRGAN Compact (SRVGGNet). Детальний аналіз проблематики в першому розділі дозволив виявити ключові аспекти, що впливають на якість зображень, та систематизувати їх у вигляді комплексної онтології. Це створило міцну теоретичну базу для подальшої розробки системи.

Застосування онтологічного підходу дозволило не тільки структурувати існуючі знання про фактори впливу на якість зображень та методи їх корекції, але й виявити нові взаємозв'язки між різними типами спотворень. Розроблена онтологія включає класифікацію спотворень за їх природою та впливом на зображення, що дозволило оптимально налаштувати архітектуру моделі для ефективної боротьби з кожним типом спотворень.

У другому розділі було проведено детальне теоретичне проектування системи, що базується на результатах онтологічного аналізу. Особлива увага була приділена розробці архітектури генератора та дискримінатора, де кожен компонент був оптимізований відповідно до виявлених взаємозв'язків між різними типами спотворень. Такий підхід дозволив створити систему, здатну ефективно працювати з комплексними випадками деградації якості зображень.

Використання онтологічного підходу також відкрило нові можливості для подальшого розвитку системи, оскільки створена онтологія може бути розширена для включення нових типів спотворень та методів їх корекції. Це дозволило оптимально налаштувати параметри моделі для певних сценаріїв використання.

Розділ проектування архітектури представив детальну розробку компонентів системи, включаючи оптимізовану структуру генератора та дискримінатора. Особливу увагу було приділено вибору активаційних функцій та проектуванню

системи функцій втрат, що забезпечило стабільність навчання та високу якість результатів.

Практична реалізація підтвердила ефективність запропонованого підходу, незважаючи на складність задачі суперрозділення зображень. Використання GAN архітектури на наборі даних DIV2K дозволило досягти обнадійливих результатів у відновленні дрібних деталей та текстур. Хоча обрані традиційні метрики показали не суто неймовірні результати, візуальне покращення якості та підвищення структурної схожості свідчать про потенціал методу. Подальша оптимізація архітектури та параметрів навчання може додатково покращити не тільки ці показники, що робить даний напрямок перспективним для майбутніх досліджень.

Розроблена модель демонструє універсальність у застосуванні, ефективно справляючись як з базовими задачами підвищення роздільної здатності, так і з комплексними випадками, що включають різні типи спотворень. Особливо вражаючі результати досягнуті у відновленні дрібних деталей та текстур при збереженні природності зображення.

Проведений економічний аналіз виявив значний комерційний потенціал розробки, особливо в контексті зростаючого попиту на якісну обробку зображень у різних галузях. Запропонована модель має перспективи впровадження як в професійних системах обробки зображень, так і в споживацьких додатках.

Подальший розвиток проекту планується зосередити на розширенні функціональності моделі для специфічних галузевих застосувань, оптимізації продуктивності та інтеграції з існуючими системами обробки зображень. Розроблена онтологія та методологія налаштування моделі створюють міцну основу для подібних вдосконалень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Цюцюра С. В. «Методичні вказівки до виконання дипломної роботи» – // КНУБА – 2018 р.
2. Ericsson Mobility Report: глобальний аналіз мобільного трафіку та технологій / Ericsson. – 2021. – URL: <https://www.ericsson.com/en/reports-and-papers/mobility-report>
3. Бугаєва Л. М. «Досягнення та перспективи розвитку технологій штучного інтелекту та їх впровадження в навчальному процесі» - збірник наукових статей Дев'ятої міжнародної науково-практичної конференції - // КПІ ім. Ігоря Сікорського – 2021 р.
4. Alom M. Z., Taha T. M., Yakopcic C., Westberg S., Sidike P., Nasrin M. S., Asari V. K. "A state-of-the-art survey on deep learning theory and architectures" - Electronics, 2019.
5. Holst G. C., Lomheim T. S. "CMOS/CCD sensors and camera systems" - JCD Publishing, 2011.
6. Liu X., Tanaka M., Okutomi M. "Noise Level Estimation Using Weak Textured Patches of a Single Noisy Image" - IEEE, 2014.
7. Russ J. C. "The image processing handbook" - CRC press, 2016.
8. Wang Z., Bovik A. C. "Modern image quality assessment" - Synthesis Lectures, 2006.
9. Lukac R., Plataniotis K. N. "Color image processing: methods and applications" - CRC press, 2018.
10. Gonzalez R. C., Woods R. E. "Digital image processing" - Pearson, 2018.
11. Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P. "Image quality assessment: from error visibility to structural similarity" - IEEE, 2004.
12. Zhang R., Isola P., Efros A. A., Shechtman E., Wang O. "The unreasonable effectiveness of deep features as a perceptual metric" - IEEE, 2018.

13. Lee J. S., Kim Y. H., Park R. H. "Adaptive histogram equalization with varying clip limits for brightness and contrast enhancement" - IEEE, 2012.
14. Tomasi C., Manduchi R. "Bilateral filtering for gray and color images" - IEEE, 1998.
15. Starck J. L., Fadili J., Murtagh F. "The undecimated wavelet decomposition and its reconstruction" - IEEE, 2007.
16. Polesel A., Ramponi G., Mathews V. J. "Image enhancement via adaptive unsharp masking" - IEEE, 2000.
17. Foi A., Katkovnik V., Egiazarian K. "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images" - IEEE, 2007.
18. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. "Generative adversarial nets" - NIPS, 2014.
19. Kingma D. P., Welling M. "An introduction to variational autoencoders" - Foundations and Trends in Machine Learning, 2019.
20. Graves A., Schmidhuber J. "Offline handwriting recognition with multidimensional recurrent neural networks" - NIPS, 2008.
21. Johnson J., Alahi A., Fei-Fei L. "Perceptual losses for real-time style transfer and super-resolution" - Springer, 2016.
22. Mechrez R., Talmi I., Zelnik-Manor L. "The contextual loss for image transformation with non-aligned data" - ECCV, 2018.
23. Liu P., Zhang H., Zhang K., Lin L., Zuo W. "Multi-level wavelet-CNN for image restoration" - IEEE, 2019.
24. Ulyanov D., Vedaldi A., Lempitsky V. "Deep image prior" - IEEE, 2018.
25. Wang X., Zhang Y. "Analysis and reduction of color moiré patterns in digital photography" - IEEE, 2016.
26. Park S. C., Park M. K., Kang M. G. "Super-resolution image reconstruction: a technical overview" - IEEE, 2003.

27. Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P. "Image quality assessment: from error visibility to structural similarity" - IEEE, 2004.
28. Wang X., Yu K., Wu S., Gu J., Liu Y., Dong C., Change Loy C. "ESRGAN: Enhanced super-resolution generative adversarial networks" - ECCV, 2018.
29. Ledig C., Theis L., Huszár F., Caballero J., Cunningham A., Acosta A., Aitken A., Tejani A., Totz J., Wang Z., Shi W. "Photo-realistic single image super-resolution using a generative adversarial network" - IEEE, 2017.
30. Kingma D. P., Ba J. "Adam: A method for stochastic optimization" - arXiv, 2014.
31. Bottou L. "Large-scale machine learning with stochastic gradient descent" - Physica-Verlag, 2010.
32. Krichen M. "Generative Adversarial Networks" - IEEE, 2023.
33. Nwankpa C., Ijomah W., Gachagan A., Marshall S. "Activation functions: Comparison of trends in practice and research for deep learning" - arXiv, 2018.
34. Zhao H., Gallo O., Frosio I., Kautz J. "Loss functions for image restoration with neural networks" - IEEE, 2016.
35. Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Kopf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S. "PyTorch: An imperative style, high-performance deep learning library" - NIPS, 2019.
36. Micikevicius P., Narang S., Alben J., Diamos G., Elsen E., Garcia D., Ginsburg B., Houston M., Kuchaiev O., Venkatesh G., Wu H. "Mixed precision training" - ICLR, 2018.
37. Ledig C., Theis L., Huszár F., Caballero J., Cunningham A., Acosta A., Aitken A., Tejani A., Totz J., Wang Z., Shi W. "Photo-realistic single image super-resolution using a generative adversarial network" - IEEE, 2017.

38. Image Recognition Market Size [Электронный ресурс] // MarketsandMarkets Research. – 2023. – URL: <https://www.marketsandmarkets.com/Market-Reports/image-recognition-market-222404611.html>
39. Computer Vision Market Growth [Электронный ресурс] // IDC Report. – 2023. – URL: <https://www.idc.com/getdoc.jsp?containerId=prUS52371124>
40. Computer Vision Market Analysis [Электронный ресурс] // Allied Market Research. – 2023. – URL: <https://www.alliedmarketresearch.com/computer-vision-market-A12701>
41. Software Market Report [Электронный ресурс] // Grand View Research. – 2023. – URL: <https://www.grandviewresearch.com/industry-analysis/software-market-report>
42. Adobe Super Resolution [Электронный ресурс] // Adobe. – 2023. – URL: <https://www.adobe.com/products/photoshop-lightroom/super-resolution.html>
43. Topaz Gigapixel AI [Электронный ресурс] // Topaz Labs. – 2023. – URL: <https://www.topazlabs.com/gigapixel>
44. Let's Enhance [Электронный ресурс] // Let's Enhance. – 2023. – URL: <https://letsenhance.io>

ЛІСТИНГ КОДУ

```
import torch
import torch.nn as nn
import torchvision
import numpy as np
import matplotlib.pyplot as plt
import os
from torchvision import transforms
from torch.utils.data import DataLoader, random_split
from PIL import Image
from torchvision.utils import save_image
import cProfile
import pstats
from torch.amp import autocast, GradScaler
from torchvision.transforms.functional import to_pil_image

# GPU memory optimization settings
os.environ['PYTORCH_CUDA_ALLOC_CONF'] = 'expandable_segments:True'
torch.cuda.empty_cache()

print(f'PyTorch version: {torch.__version__}')
print(f'CUDA available: {torch.cuda.is_available()}')
print(f'CUDA version: {torch.version.cuda}')

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f'Using device: {device}')
```

```
# Hyperparameters
latent_dim = 100
batch_size = 4
lr_image_size = 64 # Low-resolution image size
hr_image_size = 256 # High-resolution image size (4x upscale)
channels = 3
epochs = 200
lr = 0.0002
beta1 = 0.5
accumulation_steps = 4

class DIV2KDataset(torch.utils.data.Dataset):
    def __init__(self, hr_dir, lr_dir, scale=4, transform=None):
        self.hr_dir = hr_dir
        self.lr_dir = lr_dir
        self.scale = scale
        self.transform = transform
        self.image_files = [f for f in os.listdir(hr_dir) if f.endswith('.png')]

    def __len__(self):
        return len(self.image_files)

    def __getitem__(self, idx):
        try:
            hr_path = os.path.join(self.hr_dir, self.image_files[idx])
```

```

lr_path = os.path.join(self.lr_dir, self.image_files[idx].replace('.png',
'x4.png'))

```

```

if not os.path.exists(hr_path) or not os.path.exists(lr_path):
    print(f'File not found: {hr_path} or {lr_path}')
    return None

```

```

hr_img = Image.open(hr_path).convert('RGB')
lr_img = Image.open(lr_path).convert('RGB')

```

```

if self.transform:
    hr_img = self.transform['hr'](hr_img)
    lr_img = self.transform['lr'](lr_img)

```

```

return lr_img, hr_img

```

```

except Exception as e:
    print(f'Error processing image at index {idx}: {str(e)}')
    return None

```

```

transform = {
    'lr': transforms.Compose([
        transforms.Resize((lr_image_size, lr_image_size)),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])
    ]),
    'hr': transforms.Compose([
        transforms.Resize((hr_image_size, hr_image_size)),

```

```

        transforms.ToTensor(),
        transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])
    ])
}

# Updated data paths
hr_train_dir = 'D:/GAN_Project/DIV2K_train_HR'
lr_train_dir = 'D:/GAN_Project/DIV2K_train_LR_bicubic/X4'

dataset = DIV2KDataset(hr_train_dir, lr_train_dir, transform=transform)

train_size = int(0.8 * len(dataset))
val_size = len(dataset) - train_size
train_dataset, val_dataset = random_split(dataset, [train_size, val_size])

def collate_fn(batch):
    batch = list(filter(lambda x: x is not None, batch))
    return torch.utils.data.data_loader.default_collate(batch)

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True,
num_workers=0, pin_memory=True, collate_fn=collate_fn)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False,
num_workers=0, pin_memory=True, collate_fn=collate_fn)

class Generator(nn.Module):
    def __init__(self, num_residual_blocks=16, upscale_factor=4):
        super(Generator, self).__init__()

```

```
self.conv1 = nn.Conv2d(3, 64, kernel_size=9, padding=4)
self.prelu = nn.PReLU()

res_blocks = []
for _ in range(num_residual_blocks):
    res_blocks.append(ResidualBlock(64))
self.res_blocks = nn.Sequential(*res_blocks)

self.conv2 = nn.Conv2d(64, 64, kernel_size=3, padding=1)
self.bn = nn.BatchNorm2d(64)

# Upscaling layers
upsample = []
for _ in range(2): # Two upscale blocks for 4x upscaling
    upsample.extend([
        nn.Conv2d(64, 256, kernel_size=3, padding=1),
        nn.PixelShuffle(2),
        nn.PReLU()
    ])
self.upsample = nn.Sequential(*upsample)

self.conv3 = nn.Conv2d(64, 3, kernel_size=9, padding=4)

def forward(self, x):
    out1 = self.prelu(self.conv1(x))
    out = self.res_blocks(out1)
```

```

out2 = self.bn(self.conv2(out))
out = out1 + out2
out = self.upsample(out)
return self.conv3(out)

```

```
class ResidualBlock(nn.Module):
```

```

    def __init__(self, channels):
        super(ResidualBlock, self).__init__()
        self.conv1 = nn.Conv2d(channels, channels, kernel_size=3, padding=1)
        self.bn1 = nn.BatchNorm2d(channels)
        self.prelu = nn.PReLU()
        self.conv2 = nn.Conv2d(channels, channels, kernel_size=3, padding=1)
        self.bn2 = nn.BatchNorm2d(channels)

```

```

    def forward(self, x):
        residual = x
        out = self.prelu(self.bn1(self.conv1(x)))
        out = self.bn2(self.conv2(out))
        out += residual
        return out

```

```
class Discriminator(nn.Module):
```

```

    def __init__(self):
        super(Discriminator, self).__init__()
        self.net = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=3, padding=1),
            nn.LeakyReLU(0.2),

```

```
nn.Conv2d(64, 64, kernel_size=3, stride=2, padding=1),  
nn.BatchNorm2d(64),  
nn.LeakyReLU(0.2),
```

```
nn.Conv2d(64, 128, kernel_size=3, padding=1),  
nn.BatchNorm2d(128),  
nn.LeakyReLU(0.2),
```

```
nn.Conv2d(128, 128, kernel_size=3, stride=2, padding=1),  
nn.BatchNorm2d(128),  
nn.LeakyReLU(0.2),
```

```
nn.Conv2d(128, 256, kernel_size=3, padding=1),  
nn.BatchNorm2d(256),  
nn.LeakyReLU(0.2),
```

```
nn.Conv2d(256, 256, kernel_size=3, stride=2, padding=1),  
nn.BatchNorm2d(256),  
nn.LeakyReLU(0.2),
```

```
nn.AdaptiveAvgPool2d(1),  
nn.Conv2d(256, 1, kernel_size=1)
```

```
)
```

```
def forward(self, x):  
    return self.net(x).view(-1)
```

```
netG = Generator().to(device)
netD = Discriminator().to(device)

optimizerG = torch.optim.Adam(netG.parameters(), lr=lr, betas=(beta1, 0.999))
optimizerD = torch.optim.Adam(netD.parameters(), lr=lr, betas=(beta1, 0.999))

content_criterion = nn.MSELoss()
adversarial_criterion = nn.BCEWithLogitsLoss()

scaler = GradScaler('cuda')

def calculate_psnr(img1, img2):
    img1 = np.array(to_pil_image(img1.cpu().squeeze(0)))
    img2 = np.array(to_pil_image(img2.cpu().squeeze(0)))
    mse = np.mean((img1 - img2) ** 2)
    if mse == 0:
        return float('inf')
    max_pixel = 255.0
    psnr = 20 * np.log10(max_pixel / np.sqrt(mse))
    return psnr

def calculate_ssim(img1, img2):
    img1 = np.array(to_pil_image(img1.cpu().squeeze(0)))
    img2 = np.array(to_pil_image(img2.cpu().squeeze(0)))
    C1 = (0.01 * 255)**2
    C2 = (0.03 * 255)**2
```

```

img1 = img1.astype(np.float64)
img2 = img2.astype(np.float64)
kernel = np.ones((11, 11)) / 121

def ssim_channel(i1, i2):
    mu1 = np.convolve(i1, kernel, mode='valid')
    mu2 = np.convolve(i2, kernel, mode='valid')
    mu1_sq = mu1**2
    mu2_sq = mu2**2
    mu1_mu2 = mu1 * mu2
    sigma1_sq = np.convolve(i1**2, kernel, mode='valid') - mu1_sq
    sigma2_sq = np.convolve(i2**2, kernel, mode='valid') - mu2_sq
    sigma12 = np.convolve(i1 * i2, kernel, mode='valid') - mu1_mu2
    cs = (2 * sigma12 + C2) / (sigma1_sq + sigma2_sq + C2)
    ssim = (2 * mu1_mu2 + C1) * cs / (mu1_sq + mu2_sq + C1)
    return np.mean(ssim), np.mean(cs)

ssims = []
css = []
for i in range(3): # For RGB channels
    ssim, cs = ssim_channel(img1[:, :, i], img2[:, :, i])
    ssims.append(ssim)
    css.append(cs)

return np.mean(ssims)

def evaluate(model, dataloader, device, epoch):

```

```
model.eval()
psnr_values = []
ssim_values = []
with torch.no_grad():
    for i, data in enumerate(dataloader):
        try:
            if len(data) == 0:
                continue

            lr_imgs, hr_imgs = data
            lr_imgs = lr_imgs.to(device)
            with autocast('cuda'):
                sr_imgs = model(lr_imgs)

            for j in range(sr_imgs.size(0)):
                psnr = calculate_psnr(sr_imgs[j], hr_imgs[j])
                ssim = calculate_ssim(sr_imgs[j], hr_imgs[j])
                psnr_values.append(psnr)
                ssim_values.append(ssim)

            if i == 0:
                # Save results
                save_image(sr_imgs, f'sr_x4_epoch_{epoch+1}.png', normalize=True)
                save_image(hr_imgs, f'hr_x4_epoch_{epoch+1}.png', normalize=True)
                save_image(lr_imgs, f'lr_x4_epoch_{epoch+1}.png', normalize=True)
        except Exception as e:
            print(f'Error during evaluation: {str(e)}")
            continue
```

```
avg_psnr = sum(psnr_values) / len(psnr_values)
avg_ssim = sum(ssim_values) / len(ssim_values)
print(f'Epoch {epoch+1} - Average PSNR: {avg_psnr:.2f}, Average SSIM:
{avg_ssim:.4f}')
```

```
model.train()
```

```
def train(epochs):
```

```
    for epoch in range(epochs):
```

```
        for i, data in enumerate(train_loader):
```

```
            try:
```

```
                if len(data) == 0:
```

```
                    print(f'Skipping empty batch {i}')
```

```
                    continue
```

```
                lr_imgs, hr_imgs = data
```

```
                batch_size = lr_imgs.size(0)
```

```
                real_label = torch.ones(batch_size).to(device)
```

```
                fake_label = torch.zeros(batch_size).to(device)
```

```
                lr_imgs = lr_imgs.to(device)
```

```
                hr_imgs = hr_imgs.to(device)
```

```
                # Check sizes
```

```
                print(f'LR image size: {lr_imgs.shape}')
```

```
                print(f'HR image size: {hr_imgs.shape}')
```

```
# Train discriminator
for _ in range(2): # Train discriminator multiple times
    netD.zero_grad()
    with autocast('cuda'):
        real_output = netD(hr_imgs)
        d_loss_real = adversarial_criterion(real_output, real_label)

        sr_imgs = netG(lr_imgs)
        # Check SR image size
        print(f"SR image size: {sr_imgs.shape}")
        fake_output = netD(sr_imgs.detach())
        d_loss_fake = adversarial_criterion(fake_output, fake_label)

    d_loss = (d_loss_real + d_loss_fake) / 2

    scaler.scale(d_loss).backward()
    if (i + 1) % accumulation_steps == 0:
        scaler.step(optimizerD)
        scaler.update()

# Train generator
netG.zero_grad()
with autocast('cuda'):
    sr_imgs = netG(lr_imgs)
    fake_output = netD(sr_imgs)
```

```

content_loss = content_criterion(sr_imgs, hr_imgs)
adversarial_loss = adversarial_criterion(fake_output, real_label)

g_loss = content_loss + 1e-3 * adversarial_loss

scaler.scale(g_loss).backward()
if (i + 1) % accumulation_steps == 0:
    scaler.step(optimizerG)
    scaler.update()

# Print progress
if i % 10 == 0:
    print(f'Epoch [{epoch+1}/{epochs}], Step [{i+1}/{len(train_loader)}], '
          f'd_loss: {d_loss.item():.4f}, g_loss: {g_loss.item():.4f}')
except Exception as e:
    print(f'Error processing batch {i}: {str(e)}')
    continue

# Save model and generate examples
if (epoch + 1) % 10 == 0:
    torch.save(netG.state_dict(), f'generator_x4_epoch_{epoch+1}.pth')
    evaluate(netG, val_loader, device, epoch)

# Start training
train(epochs)

```

Презентація Кваліфікаційної випускної роботи



Кваліфікаційна випускна робота

РОЗРОБКА GAN МОДЕЛІ ДЛЯ ПІДВИЩЕННЯ ЯКОСТІ ЗОБРАЖЕНЬ

ВИКОНАВ Студент КНм-23 Осокін А.С. КЕРІВНИК к.т.н., доц. Горда О.В.

Слайд 1 – Розробка GAN моделі для підвищення якості зображень

АКТУАЛЬНІСТЬ ТА МЕТА

- На сьогоднішній день проблема підвищення якості зображень є критично важливою для багатьох професійних сфер застосування. В умовах постійно зростаючих вимог до якості візуального контенту та поширення пристроїв з екранами високої роздільної здатності, завдання ефективного масштабування та покращення якості зображень набуває особливої актуальності.
- Методи глибокого навчання, зокрема генеративні змагальні мережі (**GAN**), відкривають нові можливості для вирішення цієї задачі, демонструючи значний потенціал у відновленні дрібних деталей та текстур при збільшенні роздільної здатності зображень.
- Метою роботи є розробка та тренування власної **GAN** моделі для підвищення якості зображень з одночасним збільшенням їх роздільної здатності. Особлива увага приділяється оптимізації архітектури моделі для забезпечення ефективного відновлення деталей при збереженні реалістичності результатів.

2

Слайд 2 – Актуальність та мета

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТОДИ ДОСЛІДЖЕННЯ

- Виходячи з поставленої мети, об'єктом дослідження визначено процес підвищення якості цифрових зображень з використанням методів машинного навчання. Методи глибокого навчання, зокрема генеративні змагальні мережі (**GAN**), відкривають нові можливості для вирішення цієї задачі, демонструючи значний потенціал у відновленні дрібних деталей та текстур при збільшенні роздільної здатності зображень.
- В якості предмету дослідження обрано методи та алгоритми на основі генеративних змагальних мереж, як найбільш перспективного напрямку вирішення поставленої задачі.
- Методологічною основою дослідження виступає комбінація онтологічного підходу до аналізу предметної області та експериментального методу при розробці системи. Онтологічний підхід дозволив створити формальну специфікацію концептуалізації в області обробки зображень, що стало основою для подальшого проектування системи. Експериментальна складова включала серію досліджень з різними параметрами моделі для визначення оптимальної конфігурації.
- Додатково в роботі були використані методи системного аналізу для декомпозиції проблеми та визначення критеріїв якості, а також статистичні методи для оцінки результатів експериментів. Це забезпечило комплексний підхід до вирішення поставленої задачі.

3

Слайд 3 – Об'єкт, предмет та методи дослідження

ДЕРЕВО ЦІЛЕЙ



- Базуючись на визначених об'єкті, предметі та методах дослідження, а також враховуючи актуальність проблеми, було сформовано детальне дерево цілей проекту.
- Кожен етап має основну ціль, яка безумовно підпорядкована головній меті, а також підцілі - невеликі завдання, виконання яких разом приводить до досягнення результату. Ієрархічний спуск відображає розкладання цілей на прості завдання, які їх складають.

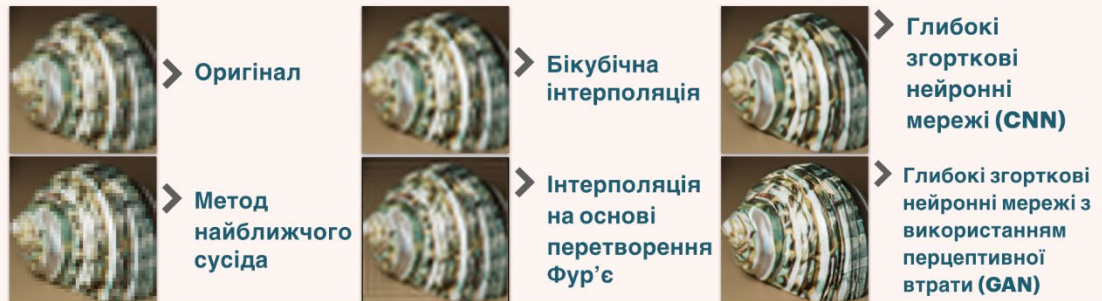
4

Слайд 4 – Дерево цілей

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ

➤ Проведений аналіз показав, що традиційні методи, такі як бікубічна інтерполяція та різні види фільтрації, мають суттєві обмеження при відновленні дрібних деталей зображення. Сучасні підходи на основі нейронних мереж демонструють значно кращі результати, особливо методи на базі **GAN**.

➤ Нижче наведено приклади прикладів їхньої роботи :

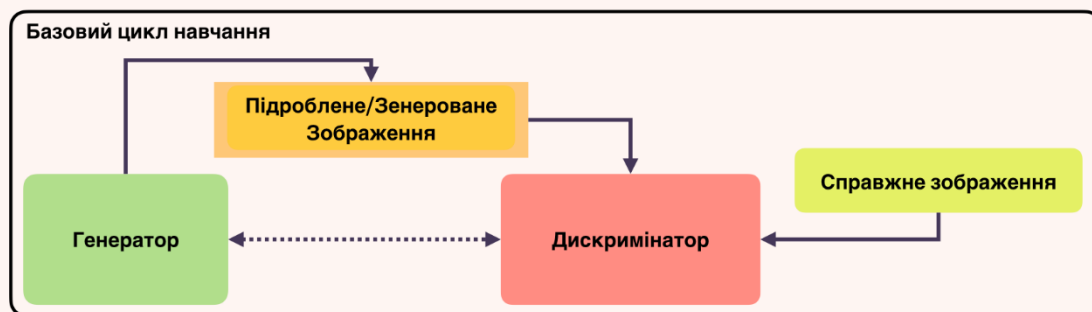


5

Слайд 5 – Аналіз існуючих методів

ЩО ТАКЕ GAN ?

➤ Генеративно-змагальна мережа (**GAN**) - це архітектура, де дві нейромережі змагаються між собою. Генератор створює підроблені зображення. Дискримінатор намагається відрізнити підроблені зображення від справжніх. У процесі такого змагання обидві мережі постійно вдосконалюється, навчаючись створювати все більш реалістичні зображення.

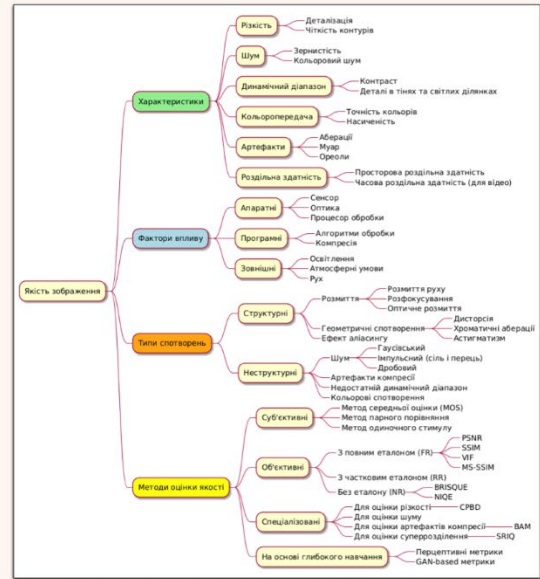


6

Слайд 6 – Що таке GAN ?

ОНТОЛОГІЧНИЙ ПІДХІД ДО ПОНЯТТЯ ЗОБРАЖЕННЯ

➤ Для систематизації знань про якість зображень та факторів впливу на неї було розроблено комплексну онтологію. Вона включає чотири ключові компоненти: характеристики зображень, фактори впливу, типи спотворень та методи оцінки якості. Такий структурований підхід дозволив чітко визначити всі аспекти, які необхідно врахувати при розробці системи. Створена онтологія стала фундаментом для подальшого проектування моделі та вибору методів оцінки.

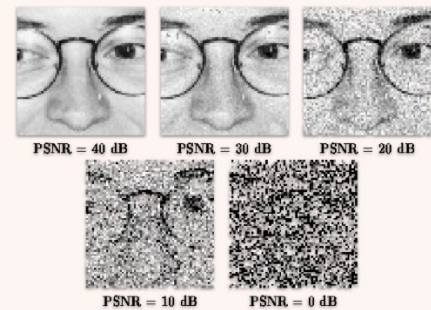
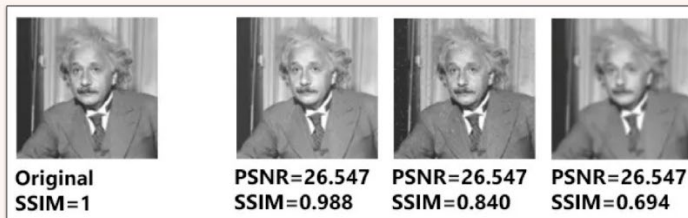


7

Слайд 7 – Онтологічний підхід до поняття зображення

ВИЗНАЧЕННЯ КРИТЕРІЇВ ОЦІНКИ

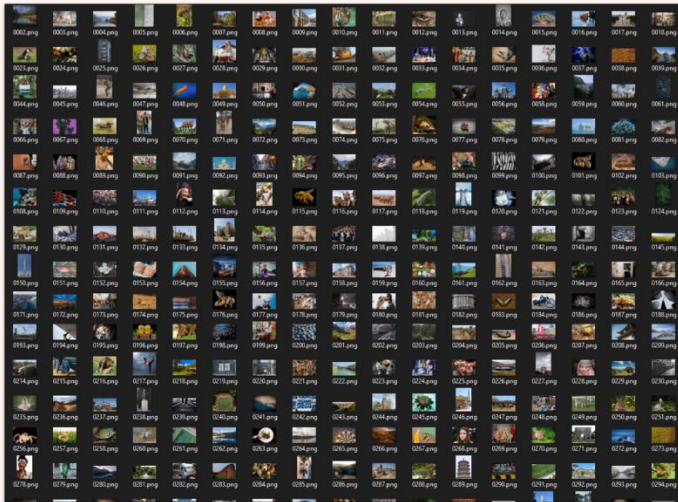
➤ На основі розробленої онтології визначено ключові критерії оцінки якості роботи системи. Основними метриками є **PSNR (Peak Signal-to-Noise Ratio)** та **SSIM (Structural Similarity Index)**, які дозволяють об'єктивно оцінювати якість результатів. Крім того, важливим критерієм є візуальне сприйняття, оскільки воно визначає реальну якість зображення з точки зору користувача.



8

Слайд 8 – Визначення критеріїв оцінки

ОБРАННЯ НАБОРУ ДАНИХ ДЛЯ НАВЧАННЯ



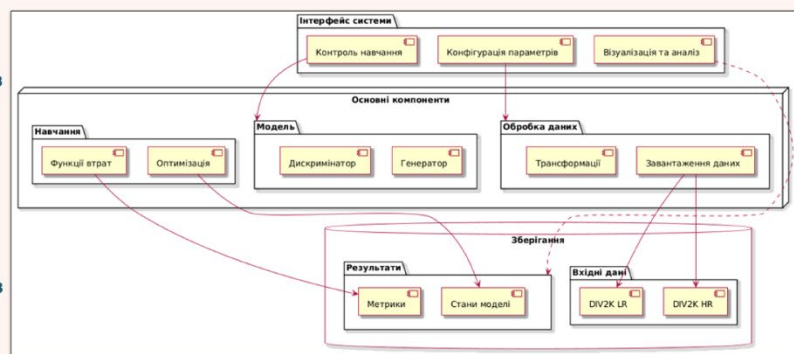
► На основі визначених цілей та критеріїв було обрано набір даних **DIV2K**, який містить високоякісні зображення для тренування моделі. Цей набір даних включає **800** зображень високої роздільної здатності разом з їх версіями нижчої якості. Важливою перевагою **DIV2K** є різноманітність контенту та типів зображень, що дозволяє моделі навчитися працювати з різними сценаріями. Для навчання дані було розділено на тренувальну та валідаційну вибірки у співвідношенні **80/20**.

9

Слайд 9 – Обрання набору даних для навчання

► На основі проведеного аналізу існуючих рішень та результатів онтологічного підходу, для нашої системи було обрано та адаптовано архітектуру **Real-ESRGAN Compact (SRVGGNet)**. Вибір цієї архітектури обумовлений її оптимальним балансом між якістю результатів та вимогами до обчислювальних ресурсів. В процесі адаптації особлива увага була приділена оптимізації компонентів для роботи в умовах обмежених ресурсів, зберігаючи при цьому високу якість генерації. Імплементація моделі виконана з використанням фреймворку **PyTorch**, що забезпечило необхідну гнучкість у налаштуванні та оптимізації архітектури.

ПРОЕКТУВАННЯ МОДЕЛІ

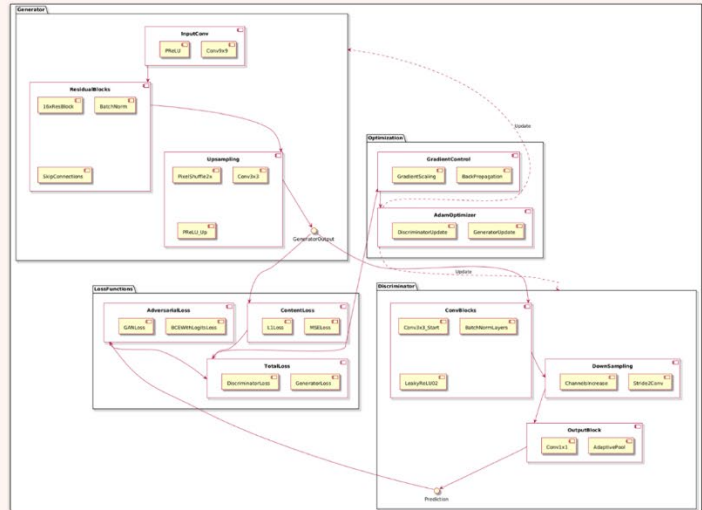


10

Слайд 10 – Проектування моделі

ДЕТАЛЬНІШЕ ПРО ГЕНЕРАТОР ТА ДИСКРИМІНАТОР

- Генератор побудовано на основі глибокої залишкової мережі з 16 резидуальними блоками та механізмом підвищення роздільної здатності через **PixelShuffle**.
- Дискримінатор реалізовано як згорткову мережу з поступовим зменшенням просторової розмірності та використанням **BatchNorm** для стабілізації навчання. Обидва компоненти оптимізовані для ефективної взаємодії та стабільного навчання.



11

Слайд 11 – Детальніше про генератор та дискримінатор

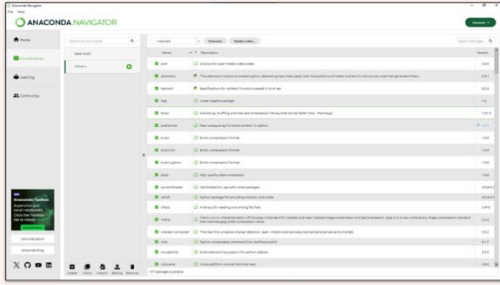
ОПТИМІЗАЦІЯ ПРОЦЕСУ НАВЧАННЯ

- Налаштування взаємодії між генератором та дискримінатором є важливим і складним завданням, оскільки обидві моделі мають навчатися гармонійно, зберігаючи баланс. Для цього використовується комбінована функція втрат, що складається з двох компонентів. Контентна складова забезпечує структурну схожість результатів, обчислюючи середньоквадратичну помилку (**MSE**) між згенерованим та цільовим зображенням. Змагальна складова стимулює генерацію реалістичних зображень, використовуючи бінарну крос-ентропію (**BCE**) між результатами дискримінатора. Загальна функція втрат формується як зважена сума цих компонентів, дозволяючи досягти оптимального балансу.
- Для забезпечення ефективного навчання генератора та дискримінатора використовується адаптивний оптимізатор **Adam**. Він враховує моменти градієнтів, обчислюючи середнє значення (перший момент) та дисперсію (другий момент), що дозволяє адаптувати швидкість навчання для кожного параметра окремо. Це забезпечує стабільне навчання навіть у випадках, коли градієнти мають різний масштаб, що є критично важливим для уникнення домінування однієї з моделей над іншою.
- У поєднанні ці підходи дозволяють створити систему, яка навчається узгоджено: генератор удосконалює якість створюваних зображень, водночас дискримінатор підвищує свою здатність відрізнити згенеровані зображення від реальних.

12

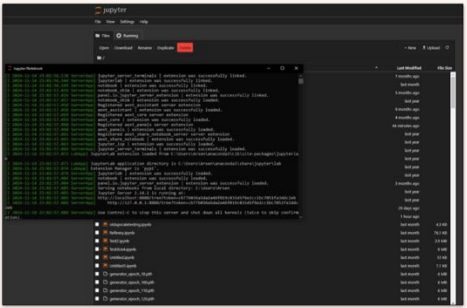
Слайд 12 – Оптимізація процесу навчання

ТЕХНІЧНІ ПЛАТФОРМИ



➤ Для реалізації системи обрано комплекс сучасних технологічних інструментів, що забезпечують ефективну розробку та впровадження AI-рішень. В якості основного середовища розробки використовується **Jupyter Notebook**, який надає інтерактивний інтерфейс для розробки та тестування компонентів системи.

- Управління залежностями та віртуальним середовищем здійснюється через **Anaconda Navigator**, де створено спеціалізоване середовище **GANenv** з усіма необхідними бібліотеками.
- Основою системи є фреймворк **PyTorch**, який забезпечує гнучкі можливості для розробки та навчання нейронних мереж. Для оптимізації обчислень використовується технологія **CUDA**, що дозволяє ефективно задіяти потужності графічного процесора. Система розроблена з урахуванням можливості масштабування та підтримки різних апаратних конфігурацій, від локальних робочих станцій до хмарних середовищ розробки.

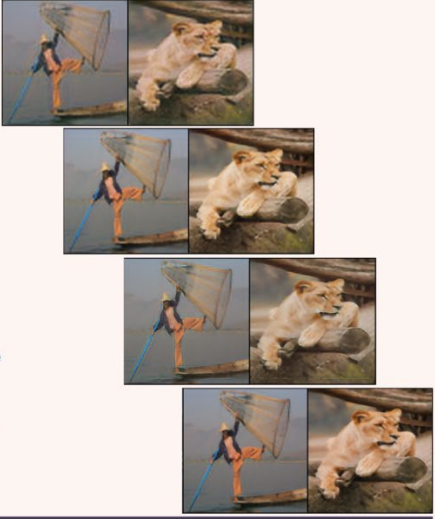


13

Слайд 13 – Технічні платформи

НАВЧАННЯ МОДЕЛІ

- В ході реалізації проекту було проведено навчання моделі протягом **200 епох** на обраному наборі даних. Для оптимізації використовувався алгоритм **Adam** з початковою швидкістю навчання **0.0002** та коефіцієнтами **beta1=0.5**, **beta2=0.999**. Стабільність навчання забезпечувалась використанням механізму накопичення градієнтів з параметром **accumulation_steps=4**.
- Для ефективного використання обчислювальних ресурсів було застосовано технологію змішаної точності обчислень (**mixed precision training**) з використанням **GradScaler**. Це дозволило оптимізувати використання відеопам'яті при збереженні точності обчислень. Розмір батчу встановлено на рівні **4** зображень, що забезпечило оптимальний баланс між швидкістю навчання та використанням пам'яті.
- Кожні **10 епох** проводилась оцінка якості на валідаційному наборі з обчисленням метрик **PSNR** та **SSIM**, а також збереження проміжних результатів для візуального аналізу. Процес навчання демонстрував стабільне покращення як об'єктивних метрик якості, так і візуальних результатів.
- Реалізована система логування дозволила відстежувати динаміку навчання та вчасно виявляти можливі проблеми. Додатково було реалізовано механізм збереження стану моделі, що забезпечило можливість відновлення навчання після можливих перерв або збоїв.



14

Слайд 14 – Навчання моделі

ТЕХНІЧНІ ВИМОГИ ТА ОБМЕЖЕННЯ

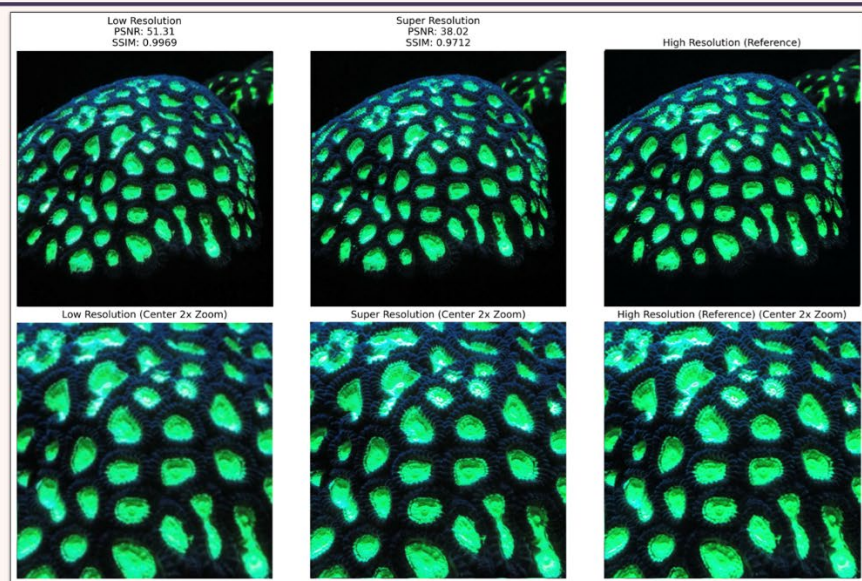
- При розробці та тестуванні системи було проведено серію експериментів на різних апаратних конфігураціях, де ключовим фактором виявився обсяг доступної відеопам'яті (**VRAM**). Експерименти показали, що обсяг **VRAM** безпосередньо впливає на розмір батчу та швидкість навчання: при **8 ГБ VRAM** система стабільно працює з батчем розміром **4** зображення, тоді як **16 ГБ** дозволяють збільшити батч до **12-16** зображень, прискорюючи навчання в **2.5-3** рази.
- Тестування на різних **GPU** підтвердило, що саме обмеження **VRAM**, а не обчислювальна потужність, є головним фактором, що впливає на продуктивність при роботі з високороздільними зображеннями. Оптимальна конфігурація включає **GPU з 12-16 ГБ VRAM, 32 ГБ** оперативної пам'яті та швидкий **SSD** накопичувач.
- Експерименти з різними об'ємами системної пам'яті та типами накопичувачів показали їх меншу, але все ж важливу роль у загальній продуктивності. При такій конфігурації система стабільно обробляє зображення до **2K** роздільної здатності, забезпечуючи оптимальний баланс між якістю результатів та швидкістю обробки.

15

Слайд 15 – Технічні вимоги та обмеження

ПРИКЛАД РОБОТИ

- Кількісна оцінка демонструє високу ефективність системи: для вхідного зображення **PSNR** складає **51.31** та **SSIM** **0.9969**, тоді як для обробленого зображення отримано **PSNR 38.02** та **SSIM 0.9712**. Незважаючи на дещо нижчі числові показники (очікуємо) у обробленого зображення, візуальний аналіз збільшених фрагментів показує, що система успішно відновлює дрібні деталі та текстури, зберігаючи природний вигляд зображення. Особливо помітно покращення чіткості контурів флуоресцентних структур та відновлення тонких градієнтів яскравості.



16

Слайд 16 – Приклад роботи

ВИСНОВКИ

- У цій роботі було створено власну генеративну змагальну модель. Аналіз актуальності теми підтвердив, що якість зображень є важливим аспектом для різних сфер застосування, таких як медицина, мультимедіа, машинне зорове сприйняття та інші.
- Реалізована модель на базі **Real-ESRGAN Compact (SRVGGNet)** продемонструвала високу ефективність у відновленні дрібних деталей та текстур, особливо в складних випадках з різними типами спотворень. Практична реалізація підтвердила стабільність навчання та візуальне покращення якості зображень.
- Застосування онтологічного підходу дозволило структурувати знання про спотворення та методи їх корекції, виявивши нові взаємозв'язки між ними. Це стало основою для оптимізації архітектури генератора та дискримінатора, що значно підвищило ефективність моделі.
- Подальша оптимізація параметрів навчання та архітектури моделі може покращити як традиційні метрики оцінки, так і візуальну якість результатів. Розширення онтології сприятиме адаптації моделі до нових типів спотворень та специфічних сценаріїв використання.
- Загалом, розробка була успішною та має значний потенціал для подальшого розвитку та масштабування.

17

Слайд 17 – Висновки

ДЯКУЮ ЗА УВАГУ



Слайд 18 – Дякую за увагу