НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 303.732.4

До захисту допущено
Завідувач кафедри ММСА
_____Оксана ТИМОЩУК
«___» _____2024 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Системний аналіз фінансового ринку»
зі спеціальності 124 «Системний аналіз»
на тему: «Породжувальні моделі та методи глибокого навчання для формування зображень»

Виконала:
Студентка 2 курсу, групи КА-32мп
Ланько Анна Анатоліївна _____

Науковий керівник:
Професор каф. ММСА, д.т.н., доцент
Недашківська Надія Іванівна _____

Рецензент:
Доцент каф. СП, к.т.н., доцент
Булах Богдан Вікторович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань

Студентка: _____

Київ – 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти – другий (магістерський)
Спеціальність – 124 «Системний аналіз»
Освітньо-професійною програмою «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ
Завідувач кафедри ММСА
_____ Оксана ТИМОЩУК
«___» ___2024 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Ланько Анні Анатоліївні

1. Тема дисертації: «Породжувальні моделі та методи глибокого навчання для формування зображень», науковий керівник дисертації: Недашківська Надія Іванівна, професор каф. ММСА, д.т.н., затверджені наказом по університету від «07» листопада 2024 р. № 5001-с.
2. Строк подання студентом дисертації: 05.12.2024.
3. Об’єкт дослідження: високороздільні зображення, породжені методами глибокого навчання.
4. Предмет дослідження: архітектура та навчання породжувальних моделей та глибоких серед для збільшення роздільності.
5. Перелік завдань, які потрібно розробити:
   1) провести огляд літератури за обраною темою, на основі отриманих даних: обґрунтувати актуальність теми та обраного напрямку реалізації, визначити сучасні методи збільшення роздільності та методи оцінювання якості результатів;
   2) дослідити архітектуру відомих моделей збільшення роздільності зображень;
   3) здійснити постановку задачі для експерементальних досліджень та

обрати вихідні дані;

4) здійснити власну реалізацію моделей та застосувати технологію донавчання з тонким налаштуванням, знайти оптимальну за сукупністю показників модель збільшення роздільності;
5) представити програмний продукт, що генерує суперроздільні зображення;
6) запропонувати стартап за обраною темою, здійснити його маркетинговий аналіз;
7) оформити пояснювальну записку.

6. Перелік графічного (ілюстративного) матеріалу:
   1) пояснювальні ілюстрації до теоретичного матераілу: приклади сформованих описаними методами зображень та схеми алгоритмів (7 рисунків у розділі 1);
   2) схеми елемінтів архітектури описаних моделей та роз'яснбвальні ілюстрації (11 рисунків у розділі 2);
   3) результати досліджень: графіки оптимізації функцій втрат моделей, отримані суперроздільні зображення у порівнянні з вихідною парою (9 рисунків у розділі 3).
7. Орієнтовний перелік публікацій:
   1) Ланько А. А., Недашківська Н. І. Породжувальні моделі та методи глибокого навчання для задачі SISR // Збірник доповідей III науково-практичної конференції «Системні науки та інформатика», 25–29 листопада 2024 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024. 6 стор;
   2) Nedashkovskaya N., Lanko A. Quality assessment of models and deep learning methods for super-resolution image formation // System research and information technologies. – 2024. – №4. – 19 p.
8. Дата видачі завдання: 02.09.2024.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Огляд літератури за темою магістерської дисертації (актуальність, предметна область) та затвердження теми	02.09.2024 – 08.09.2024	Виконано
2	Отримання завдання на магістерську дисертацію: формулювання об’єкта, предмета та мети дослідження, новизни та практичної значущості результатів.	02.09.2024 – 08.09.2024	Виконано
3	Перший розділ: огляд задачі суперроздільності зображень, методів її розв’язання та оцінки результатів	09.09.2024 – 15.09.2024	Виконано
4	Другий розділ: опис архітектури провідних моделей суперроздільності та показників їхньої якості	16.09.2024 – 22.09.2024	Виконано
5	Третій розділ: власна реалізація моделей та застосування технології донавчання, визначення алгоритмів навчання та оцінювання з метою оптимізації обчислювальних ресурсів	23.09.2024 – 13.10.2024	Виконано
6	Представлення програмного продукту	14.10.2024 – 20.10.2024	Виконано
7	Четвертий розділ: розробка стартап-проекту за темою магістерської дисертації	21.10.2024 – 27.10.2024	Виконано
8	Допрацювання магістерської дисертації: доповнення теоретичного матераілу та результатів дослідження	28.10.2024 – 17.11.2024	Виконано
9	Висновки за результатами виконання завдання на магістерську дисертацію	18.11.2024 – 24.11.2024	Виконано
10	Оформлення пояснювальної записки	25.11.2024 – 01.12.2024	Виконано

Студентка _____ Анна ЛАНЬКО

Науковий керівник дисертації _____ Надія НЕДАШКІВСЬКА

РЕФЕРАТ

Магістерська дисертація: 100 с., 27 рис., 29 табл., 1 додаток, 41 джерело.

У даній роботі розглядаються породжувальні моделі та методи формування зображень на прикладі збільшення роздільності зображень. Проводяться експерименти з власної реалізації моделей SRGAN, VDSR, DRCN, SRCNN та використання технології їх донавчання, здійснюється аналіз результатів та визначається оптимальна модель на основі кількісних та перцепційних показників якості, технічних критеріїв та візуальної оцінки.

Об’єкт дослідження – високороздільні зображення, породжені методами глибокого навчання.

Предмет дослідження – архітектура та навчання породжувальних моделей та глибоких серед для збільшення роздільності.

Метою даної роботи є власна реалізація відомих моделей SISR з більш простою архітектурою та застосування технології донавчання з тонким налаштуванням для знаходження оптимальної за сукупністю показників моделі збільшення роздільності зображень.

Наукова новизна роботи полягає у визначенні алгоритмів навчання та об’єктивного оцінювання результатів для забезпечення гнучкості проведення експериментів, оптимізації використання обчислювальних ресурсів, вибору оптимальної моделі для різних вихідних даних та пріоритетних потреб задачі за сукупністю критеріїв.

Результатом роботи є програмний продукт для збільшення роздільності зображень, розроблений засобами мови програмування Python.

ГЛИБОКЕ НАВЧАННЯ, ПОРОДЖУВАЛЬНІ МОДЕЛІ, ЗБІЛЬШЕННЯ РОЗДІЛЬНОСТІ ЗОБРАЖЕНЬ, ПЕРЦЕПЦІЙНА ЯКІСТЬ, SRGAN, VDSR, DRCN, SRCNN.

ABSTRACT

Master's thesis: 100 p., 27 figures, 29 tables, 1 appendix, 41 references.

In this work, the generating models and methods of image formation are considered on the example of image resolution increase. Experiments are carried out on the own implementation of SRGAN, VDSR, DRCN, SRCNN models and the use of their training technology, the results are analyzed and the optimal model is determined on the basis of quantitative and perceptual quality indicators, technical criteria and visual assessment.

The object of research is high-resolution images generated by deep learning methods.

The subject of the study is the architecture and training of generating models and deep learning for resolution enhancement.

The purpose of this study is to implement the known SISR models with a simpler architecture and apply the fine-tuning training technology to find the optimal model for increasing image resolution in terms of a set of indicators.

The scientific novelty of the work lies in the definition of training algorithms and objective evaluation of results to ensure the flexibility of conducting experiments, optimizing the use of computing resources, selecting the optimal model for different input data and priority needs of the task according to a set of criteria.

The result of the work is a software product for increasing image resolution developed using the Python programming language.

DEEP LEARNING, GENERATIVE MODELS, IMAGE RESOLUTION ENHANCEMENT, PERCEPTUAL QUALITY, SRGAN, VDSR, DRCN, SRCNN.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

DRCN	–	Deeply-Recursive Convolutional Network
GPU	–	Graphics Processing Unit
HR	–	High-Resolution
MSE	–	Mean Squared Error
LPIPS	–	Learned Perceptual Image Patch Similarity
LR	–	Low-Resolution
PSNR	–	Peak Signal-to-Noise Ratio
ResNet	–	Residual Network
RNN	–	Recursive Neural Network
SISR	–	Single Image Super-Resolution
SR	–	Super-Resolution
SRGAN	–	Super-Resolution Generative Adversarial Network
SRCNN	–	Super-Resolution Convolutional Neural Network
SSIM	–	Structural Similarity Index Measure
VDSR	–	Very-Deep Super Resolution
НМ	–	Нейронна мережа
СП	–	Стартап-проект
ІШІ	–	Штучний інтелект

ВСТУП

З розвитком та розширенням доступності технологій зйомки все менше зображень є низькороздільними за своєю природою. Утім, задача суперроздільності є актуальною, оскільки зображення часто стискаються для передачі інформації на надвеликі відстані (наприклад, з супутників) або з метою економії ресурсів, коли у чергу безперервно надходять знімки (як на камерах спостереження). Також багато зображень втрачають свою якість при апаратних збоях знімального обладнаня та носіїв даних.

Розвиток технологій штучного інтелекту (ШІ) та глибокого навчання призвів до появи точних методів збільшення роздільності зображень, але оптимізація часу навчання та обчислювальної складності таких моделей залишаються об'єктом дослідження науковців. Особливо гостро ця проблема постає у задачі збільшення роздільності відео, де потрібно генерувати велику кількість суперроздільних кадрів.

Метою даної роботи є власна реалізація відомих моделей SISR з більш простою архітектурою та застосування технології донавчання з тонким налаштуванням для знаходження оптимальної за сукупністю показників моделі збільшення роздільності зображень.

Наукова новизна роботи полягає у визначенні алгоритмів навчання та об’єктивного оцінювання результатів для забезпечення гнучкості проведення експериментів, оптимізації використання обчислювальних ресурсів, вибору оптимальної моделі для різних вихідних даних та пріоритетних потреб задачі за сукупністю критеріїв: кількісних та перцепційних показників, часу навчання і складності архітектури моделей, процесу їх навчання.

Практична значущість даної роботи полягає у тому, що створений програмний продукт з навченими моделями можна легко використати для збільшення роздільності зображень у різних галузях та адаптувати його під

їхні специфічні потреби, а також інтегрувати в інші програмні продукти та системи.

Пояснювальна записка до цієї магістерської дисертації складається з вступу, чотирьох розділів, висновків, переліку джерел посилання та додатку А. У розділі 1 наводиться огляд літератури за темою роботи, у розділі 2 – опис архітектур моделей і показників оцінювання якості; у розділі 3 проводиться аналіз отриманих результатів, у розділі 4 запропоновано стартап-проект (СП) за темою роботи.

РОЗДІЛ 1 ОГЛЯД ЗАДАЧІ ЗБІЛЬШЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ЗОБРАЖЕНЬ

Даний розділ містить результати огляду літератури за предметною областю. Надається визначення процесу формування зображень та розглядаються різні задачі, що його реалізують. Наводяться вимоги до обчислювальних ресурсів. Описується задача суперроздільності зображення. Здійснюється огляд методів її розв’язання разом з методами оцінювання результатів.

1.1 Процес формування зображень

Формування зображень у контексті породжувальних моделей та методів глибокого навчання – це процес, за якого модель навчається створювати нове зображення на основі навчального набору даних. Він знаходить широке застосування у галузях, які потерпають від низької якості зображень, пов’язані з їх високотехнологічною обробкою або ж потребують великої бази графічних даних, яку неможливо зібрати з причин відсутності або гетерогенності даних, що зберігаються на відкритих ресурсах. До таких галузей відносяться різні види візуального мистецтва (наприклад, фотографія, поліграфія), медицина, геоінформатика тощо.

Процес формування зображень лежить в основі багатьох задач комп’ютерного зору, що розрізняються метою формування або вимогами до результату. Найпопулярнішою його формою є створення зображень, подібних до вихідних – тобто розширення набору даних, проте з точки зору складності ця задача є однією з найскладніших, оскільки передбачає створення зображень «з нуля».

До більш складних задач відносять наступні [1]:

1) суперроздільність зображення – збільшення чіткості зображень без втрати деталей та виникнення спотворень; зазначена точність деталізації досягається за рахунок того, що модель відновлює високочастотні деталі (текстури, краї) там, де вони дійсно втрачаються при масштабуванні;

2) ліквідація зашумленості – видалення з зображень зайвої інформації, що може з'являтись внаслідок невідповідних умов зйомки та апаратних збоїв; основною умовою є збереження моделлю важливих деталей та структур;

3) ефект широкого динамічного діапазону – створення зображень з високим діапазоном світлотіней на основі фотореалістичного злиття моделлю кількох зображень з різним налаштуванням експозиції;

4) кольорокорекція (у тому числі колоризація, рис. 1.1a) – адаптація кольорів зображення в умовах відсутності чіткого алгоритму обробки, який може реалізувати людина; задача моделі – «зрозуміти», яку саме кольорову гаму застосувати або які параметри та якою мірою виправити, щоб зображення стало фотореалістичним;

5) реставрація зображень – відновлення частин зображення, які були пошкоджені (корекція дефектів) або втрачені (заповнення пропущених ділянок, рис. 1.1b), зазвичай на старих фотографіях або зображеннях, які були створені в умовах апаратних збоїв; задача моделі полягає у якомога точнішому визначенні деталей, які притаманні повним зображенням зі схожими ознаками;

6) генерація відсутнього зображення з бажаною ознакою – модель навчається зливати набір ознак низки споріднених зображень з метою створення нового фотореалістичного екземпляру цього набору із заданою особливістю, наприклад, під іншим кутом (рис. 1.1e).

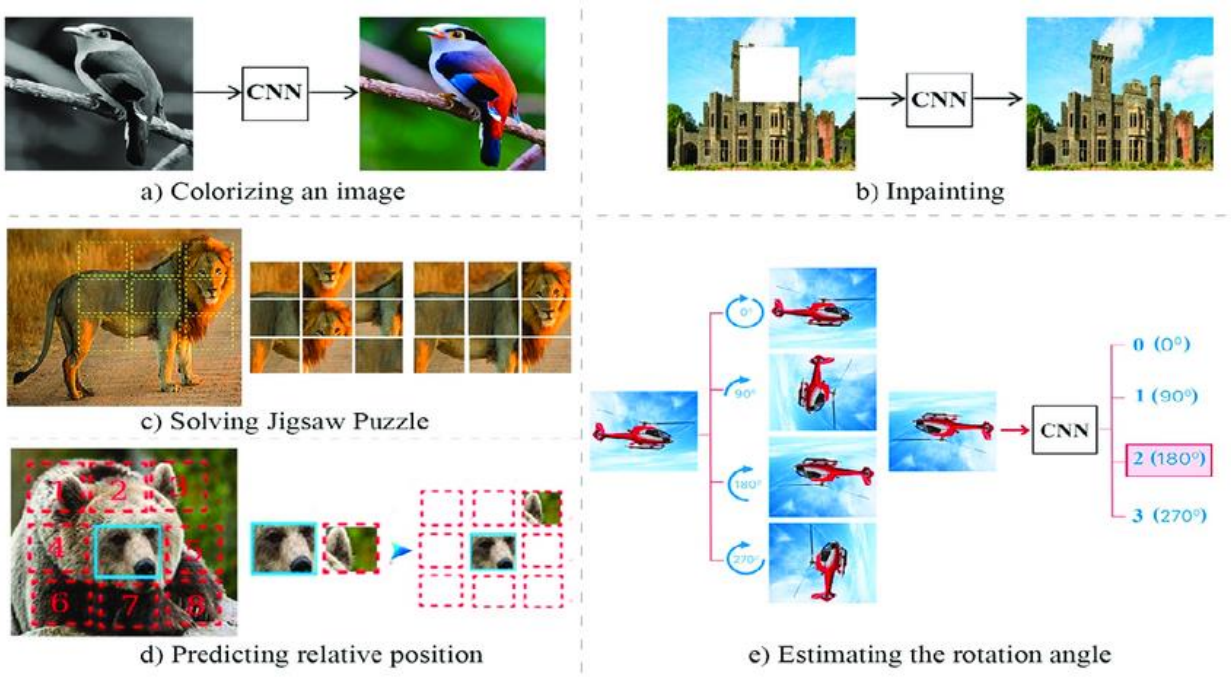

Рисунок 1.1 – Задачі формування зображень [2]

1.2 Вимоги до обчислювальних ресурсів

Через складність архітектури моделей, що працюють із зображеннями, та великі обсяги навчальних даних обчислювальні ресурси є критичним фактором для процесу формування зображень. Від них напряму залежить час навчання моделей, який для однакової задачі може варіюватися від кількох годин до кількох днів.

Графічні процесори (англ. Graphics Processing Units, GPUs) відіграють суттєву роль у прискоренні таких обчислень. GPU забезпечує паралельну обробку даних, необхідну для ефективного тренування глибоких нейронних мереж, та пропонує достатній обсяг графічної пам’яті. Вимоги до апаратного прискорювача в контексті генерації зображень варіюються в залежності від обсягу вихідних даних та складності конкретної задачі, для якої необхідно зберігати ваги, активації та проводити проміжні обчислення. Рекомендації щодо вибору GPU для формування зображень наведемо у таблиці 1.1 [3].

Таблиця 1.1 – Рекомендації щодо вибору GPU для різних задач формування зображень

GPU	**Задачі**	**Розмір батчу**	**VRAM**	**Переваги**
NVIDIA A100	Навчання масштабних породжувальних моделей	128-256	80 ГБ	Висока продуктивність для дуже глибоких моделей.
NVIDIA H100	Найвимогливіші генеративні задачі	256-512	80 ГБ	Найвища продуктивність для масштабних задач
RTX 6000 Ada	Навчання генеративних моделей	64-128	48 ГБ	Оптимальний для генеративних проектів середньої складності
RTX 4090	Навчання моделей суперроздільності	32-64	24 ГБ	Найкраще співвідношення ціна/якість
RTX 4060	Тонке налаштування невеликих моделей	8-16	8 ГБ	Доступний для локальних експериментів

1.3 Задача Single Image Super-Resolution

Задача SISR (англ. Single Image Super-Resolution), відома як задача суперроздільності зображення, передбачає відтворення високо деталізованої версії SR (англ. Super-Resolution) певного зображення низької роздільної здатності LR (англ. Low-Resolution), яке розглядається як об’єкт з «втраченими» або деградованими високочастотними деталями [4]. Відповідні високороздільні оригінали HR (англ. High-Resolution) цих зображень потрібні для створення LR-зображень шляхом зменшення їх розміру (при цьому

можуть додавати шум або застосовувати інші спотворення, як показано на рис. 1.2), вивчення закономірностей збільшення роздільності на парах «LR – HR» моделями ШІ при навчанні та оцінювання якості цих моделей шляхом порівняння SR та HR.

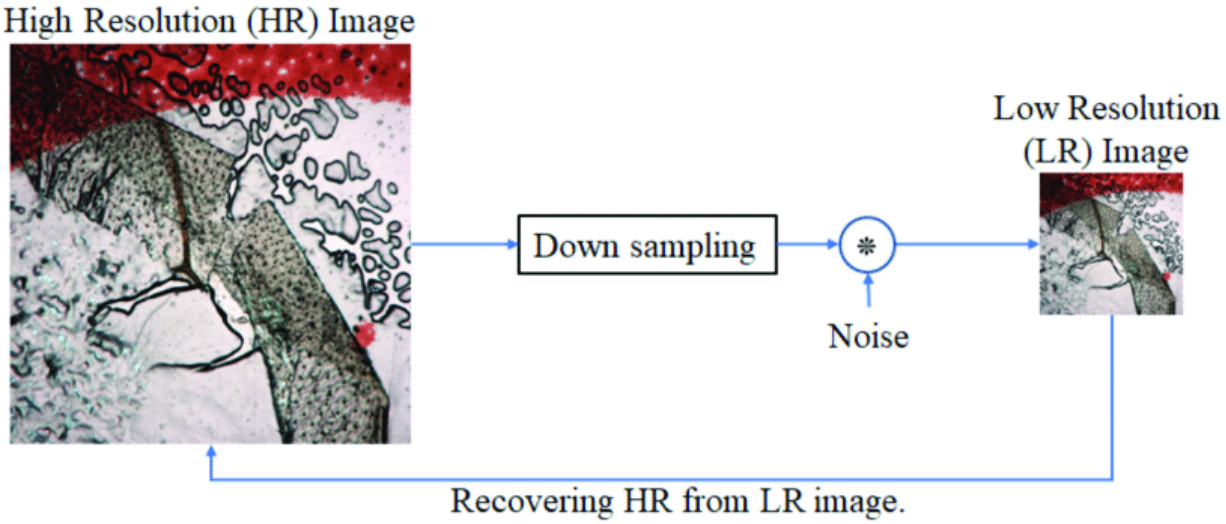

Рисунок 1.2 – Послідовність операцій у задачі SISR [4]

Існує два основних напрямки SISR:

1) відновлення оригінальної якості зображень, яка могла постраждати внаслідок передачі через канали зв'язку або інші причини стиснень, пов'язані з особливостями носіїв даних чи помилками в їхній роботі;

2) покращення якості зображень, які були отримані в умовах апаратних збоїв або використання застарілого знімального обладнання (старі фотографії, знімки з бюджетних версій супутників, камер відеоспостереження та медичного обладнання).

Історія SISR бере свій початок ще з часів виникнення потреби у формуванні зображень. Найперший метод збільшення роздільної здатності зображень – це масштабування, яке реалізовувалось за допомогою інтерполяційних технік, таких як білінійна та бікубічна інтерполяції, метод найближчих сусідів [6]. Результати масштабування відзначаються розмитими краями та значною кількістю втрачених або спотворених деталей (рис. 1.3).

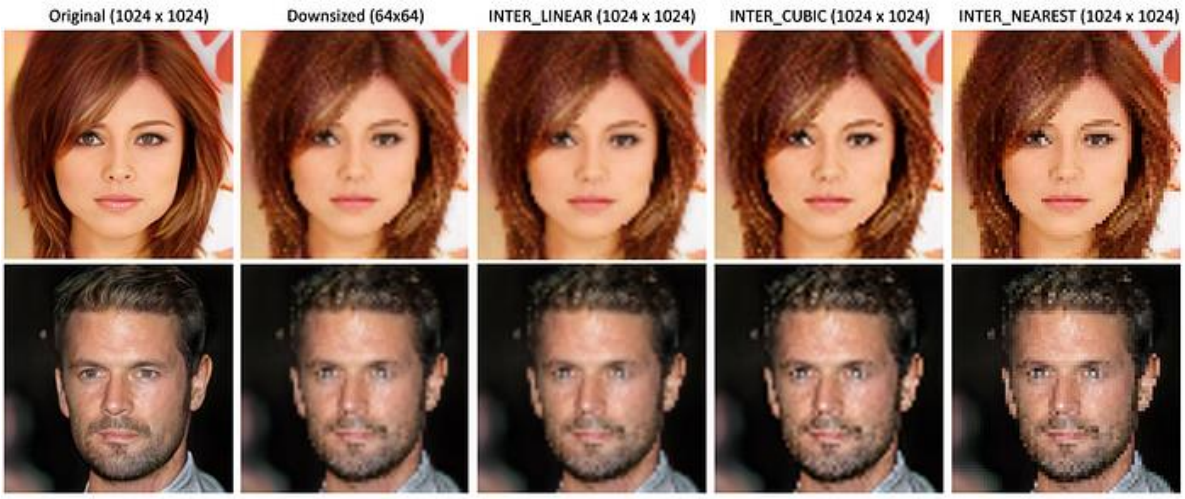

Рисунок 1.3 – Приклад втрати якості зображень при збільшенні роздільності інтерполяційними методами [6]

З розвитком машинного, зокрема глибокого, навчання з'явились більш досконалі методи розв'язання задачі SISR, що пропонують суттєво вищу деталізацію та реалістичність відтворення (рис. 1.4) [6]. На відміну від масштабування, вони вивчають розподіли зображень у наборі даних або глибокі залежності між LR- та HR-зображеннями. Такий підхід є критично важливим, коли LR-зображення формувалось з додаванням шуму та інших видів спотворення, для вирішення проблеми артефактів – небажаних або викривлених деталей.

Рисунок 1.4 – Порівняння якості збільшення роздільної здатності бікубічною інтерполяцією (b) та сучасними методами глибокого навчання (c-e) [7]

1.4 Огляд методів формування суперроздільних зображень

1.4.1 Відмінність між породжувальними моделями та методами глибокого навчання

Породжувальні моделі моделюють розподіл даних для формування нових зразків зі схожими статистичними властивостями. Основна мета породжувальних моделей – «зрозуміти» структуру даних та навчитися їх відтворювати. У випадку SISR модель вивчає, як зіставити текстури, деталі та структури між LR- і HR-зображеннями, з подальшим відтворенням реалістичних деталей, відсутніх на вихідному LR-зображенні [7].

Методи глибокого навчання фокусуються на вивченні складних залежностей у даних для задач класифікації, прогнозування чи регресії. Вони працюють на основі глибоких нейронних мереж, які оптимізують функцію втрат для досягнення конкретної мети. У випадку SISR ці вивчають глибокі залежності між парою зображень на рівні ознак і мапують LR на HR [7].

Представники методів глибокого навчання різняться архітектурою: застосуванням різних функцій активації, спеціальних шарів, блоків та їхніх послідовностей тощо. У той час породжувальні моделі використовують глибокі мережі і відразняються між собою цілими концепціями їх використання та взаємодії. Це може бути змаганням двох мереж, послідовністю операцій мереж з різними функціями тощо. Для SISR ці класи моделей також відрізняються тим, що породжувальні моделі генерують зображення з новими текстурами та деталями на основі вивчених розподілів, у той час як глибокі мережі вдосконалюють деталі на сформованому зображенні на основі витягнутих глибоких ознак.

Методи глибокого навчання є простішими у реалізації та швидшими у навчанні, що є суттєвими перевагами для великих наборів даних у порівнянні з породжувальними моделями. Вони мають хорошу здатність до узагальнення,

оскільки не генерують нові деталі і працюють з ознаками, а не розподілами – і тому дають стабільний та якісний результат. Попри цю перевагу, узагальнення також є суттєвим недоліком методів глибокого навчання, оскільки при цьому певні ознаки визнаються неважливими, що призводить до втрати деталей. Для SISR деталі є критично важливими, тому породжувальні моделі є ще більш популярними через свою здатність. До вже відомих недоліків – довгого часу навчання та складності архітектури – можна додати схильність до генерації артефактів («вигаданих» деталей), що призводить до надмірної деталізації та відсутності фотореалістичності. Отже, для задачі SISR, потрібно використовувати обидва підходи, оскільки неможливо заздалегідь оцінити, як поведуть себе моделі на конкретних даних, особливо після налаштування параметрів.

1.4.2 Породжувальні моделі

Задачу SISR можна розв’язати за допомогою таких породжувальних моделей, як генеративно-змагальні, дифузійні та потокові моделі, рідше – автокодувальників. Нижче наведемо коротку характеристику моделей кожного типу.

1. Генеративно-змагальні моделі (англ. Generative Adversarial Networks, GANs) складаються з двох мереж: генератора та дискримінатора. Генератор створює суперроздільні зображення з низькороздільних, у той час як дискримінатор намагається відрізнити сформовані SR-зображення від оригінальних високороздільних. Навчання моделі ґрунтується на змагальній взаємодії між цими мережами, оптимум досягається в умовах спадання функції втрат генератора та зростання втрат дискримінатора [4]. До відомих генеративно-змагальних мереж відносять (рис. 1.5):

1) SRGAN (англ. Super-Resolution GAN): найперша GAN-модель для SISR, використовує перцепційну втрату (perceptual loss) для підвищення візуальної якості (враховує реалістичність текстур) [9];
2) ESRGAN (англ. Enhanced SRGAN): покращення SRGAN, яке використовує вдосконалену архітектуру генератора, за рахунок чого процес навчання моделі є більш стабільним, а результат – більш реалістичним [10];
3) TSRGAN (англ. Texture SRGAN): покращення SRGAN, яке використовує текстурну втрату (texture loss) – більш складну модифікацію перцепційної [11].

До переваг GAN відносять високу реалістичність сформованих зображень за рахунок хорошого відтворення текстур, до недоліків – складність навчання та схильність до гіпердеталізації [4].

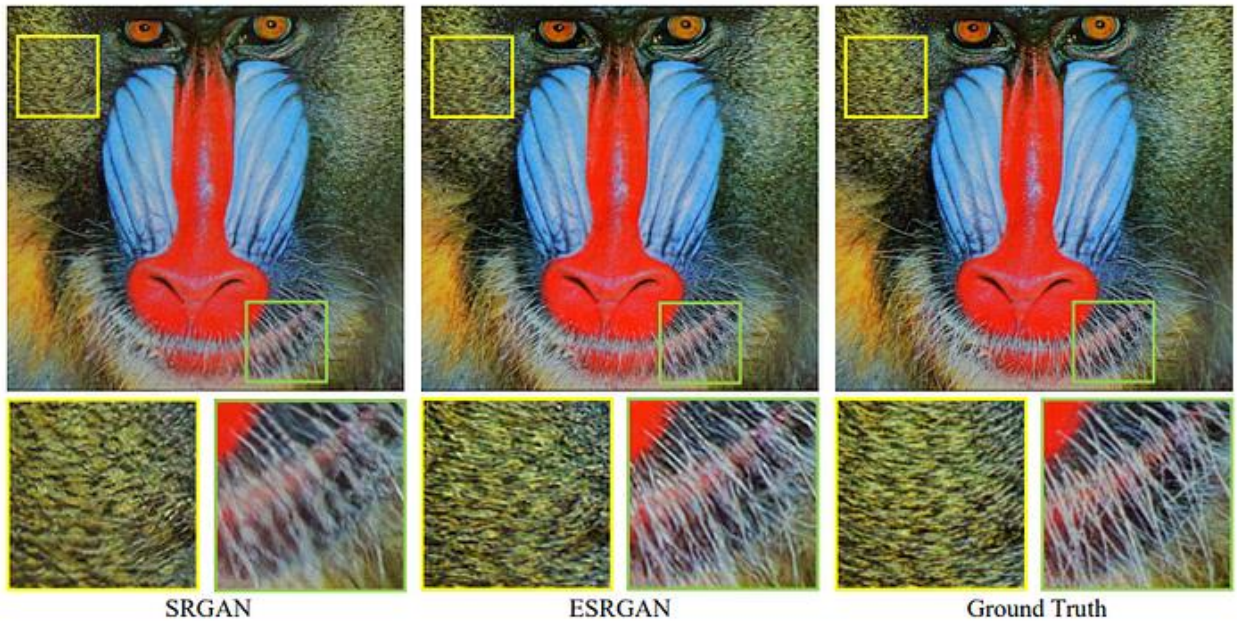

Рисунок 1.5 – Результати збільшення роздільності для деяких видів GAN [6]

2. **Дифузійні моделі** є новим типом генеративних моделей, принцип роботи яких полягає у поступовому внесенні зайвої інформації до зображення, яку модель вчиться видаляти для відновлення вихідного зображення (рис. 1.6) – а отже, логіка дифузійних моделей є протилежною до GAN. Процес додавання шуму до «чистого» зображення називається прямою дифузією

(англ. forward diffusion), процес перетворення зашумленого зображення назад у чисте – зворотною дифузією (англ. reverse diffusion) [12]. До відомих дифузійних моделей відносять:

1) SR3 (англ. Super-Resolution via Repeated Refinement): відновлює HR-зображення (результат – SR-зображення) шляхом послідовного очищення шуму [13];
2) StableDiffusion-based SISR: адаптація стабільних дифузійних моделей для задачі SISR [13].

До переваг дифузійних моделей відносять більшу стабільність та вищу якість результатів порівняно зі ншими типами породжувальних моделей, до недоліків – тривалий час навчання та високі вимоги до обчислювальних ресурсів, а також важкодоступність попередньо навчених зразків моделей [12].

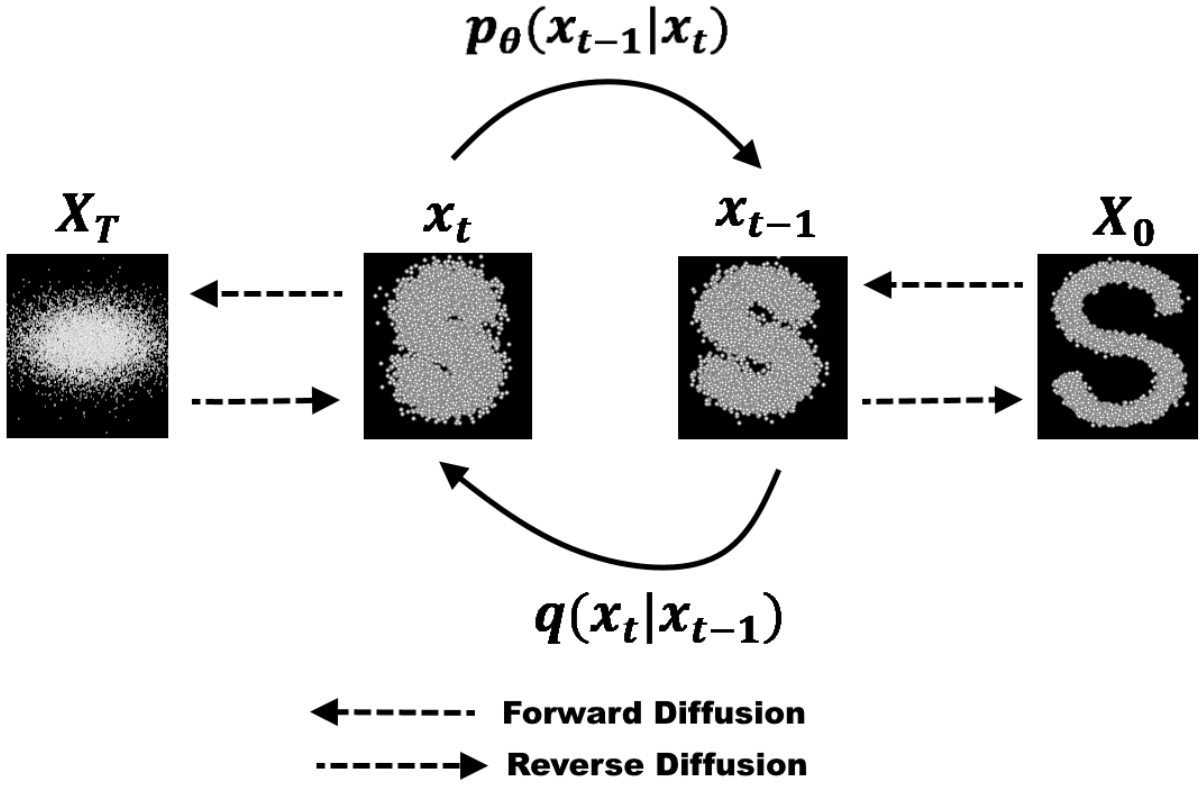

Рисунок 1.6 – Принцип роботи дифузійних моделей [12]

3. Потокові моделі (англ. flow-based models) працюють за принципом моделювання явного розподілу пікселів зображення через оборотні функції. Вони зберігають інформацію LR-зображення й додають

деталі для відновлення HR у вигляді SR [14]. До відомих потокових моделей відносять:

1) SRFlow (англ. Super-Resolution Flow): використовує нормалізуючі потоки (англ. normalizing flows), що забезпечують генерацію кількох можливих SR-зображень для одного LR-зображення з вибором найкращого [14].
2) RealSRFlow: орієнтована на дані реального світу, враховує різні спотворення LR-зображень [8].

До переваг потокових моделей відносять можливість створювати різні варіанти SR-зображень, до недоліків – складність моделі та великий час навчання [14].

Отже генеративно-змагальні моделі переважно забезпечують реалістичність, дифузійні – стабільність відновлення високороздільного зображення, а потокові – гнучкість результатів. Вибір підходу залежить від мети (наприклад, деталізація текстур або реконструкція з мінімальними спотвореннями) та можливостей обчислювальних ресурсів.

Слід зазначити, що для ще більшого підвищення реалістичності створюють **гібридні моделі** [8]. Це комбінації вищезазначених типів породжувальних моделей з **варіаційними автокодувальниками VAE** (англ. Variational Auto-Encoders) – нейронна мережа (НМ), що у випадку SISR стискають вхідні HR-зображення до меншого представлення у латентному просторі через енкодер і відновлюють їх через декодер, проте рідко використовуються незалежно [8]. Їх часто поєднують з потоковими моделями для кращого моделювання невизначеностей, з GAN – для можливості створення кількох високороздільних версій.

1.4.3 Методи глибокого навчання

Методи глибокого навчання для SISR використовують різноманітні архітектури глибоких мереж: згорткові, рекурсивні, залишкові мережі тощо. Нижче наведемо коротку характеристику мереж кожного типу [5].

1. Згорткові мережі (англ. Convolutional Neural Networks, CNNs) моделюють просторові залежності в зображеннях, використовуючи згорткові шари для вилучення ознак [5]. До відомих згорткових мереж для SISR відносять:

1) SRCNN (англ. Super-Resolution CNN): найперша модель цього типу для SISR, має дуже просту архітектуру з послідовності згорткових шарів, що поступово відновлюють зображення [15];
2) FSRCNN (англ. Fast SRCNN): покращення SRCNN з меншою кількістю параметрів і, відповідно, швидшим процесом навчання [16].

Перевагою CNN є простота реалізації, недоліком – обмежена здатність до моделювання складних текстур [5].

2. Рекурсивні мережі (англ. Recursive Neural Networks, RNNs) використовують рекурсивні обчислення (англ. recursive computations) з метою багаторазового збагачення ознак. Це дозволяє працювати з зображеннями великої роздільності, збільшуючи глибину, проте не збільшуючи при цьому кількість параметрів мережі [5]. Класичною рекурсивною мережею є DRCN (англ. Deeply-Recursive Convolutional Network), яка багаторазово використовує згортку одного і того ж шару для поступового покращення результату [17].

Переваги рекурсивних мереж – поєднання низької обчислювальної складності та здатності працювати з великими зображеннями. Недолік – схильність до проблеми затухання градієнта, через що більшість моделей поєднує у собі інші методи глибокого навчання, як це було показано для DRRN [5].

3. Залишкові мережі (англ. Residual Networks, ResNets) використовують залишкові зв’язки (англ. residual connections), щоб запобігти затуханню градієнта; завдяки своїй глибині здатні захоплювати багато інформації про текстури та деталі [5]. VDSR (англ. Very-Deep Super Resolution) є класичною залишковою мережею [18] з модифікацією EDSR (англ. Enhanced DSR) без нормалізації пакетів для покращення точності деталізації [19].

Перевагою залишкових мереж є висока якість сформованих зображень та легкий процес навчання завдяки залишковим зв’язкам, недоліком – високі вимоги до обчислювальних ресурсів.

4. Щільні мережі забезпечують максимальне повторне використання ознак шляхом з’єднання всіх шарів між собою для більш ефективного використання вилучених ознак [5]. Класичною щільною мережею є SRDenseNet [20]. До її переваг відносять ефективне використання інформації, до недоліків – високі вимоги до оперативної пам’яті для зберігання та обробки інформації про з’єднання [5].

З огляду на невиправданість тривалого часу навчання та залучення потужних обчислювальних ресурсів використовують **гібридні моделі** [8]:

1) DRRN (англ. Deep Recursive Residual Network): поєднує рекурсивну архітектуру з залишковими блоками для підвищення точності деталізації [21];

2) RCAN (англ. Residual Channel Attention Network): додає механізми уваги для вибору важливих каналів ознак у залишковій мережі [22];

3) RDN (англ. Residual Dense Network): поєднує залишкові та щільні блоки для більш ефективного вилучення ознак [23].

Також для SISR можна використовувати трансформери для глобального контекстного аналізу зображень, надаючи можливість відновлювати складні структури. Класичною моделлю є SwinIR (англ. Swin Transformer for Image Restoration) [24].

Вибір типу та архітектури глибоких мереж для SISR залежить від складності та обсягу навчальних даних, а також від того, як використовується

мережа: як самостійний метод (вимагає більш складної архітектури та залучення технік покращення результату), або ж у складі породжувальної моделі (тоді можна використати базову модель). Результати збільшення роздільності для деяких моделей глибокого навчання наведено на рисунку 1.7.

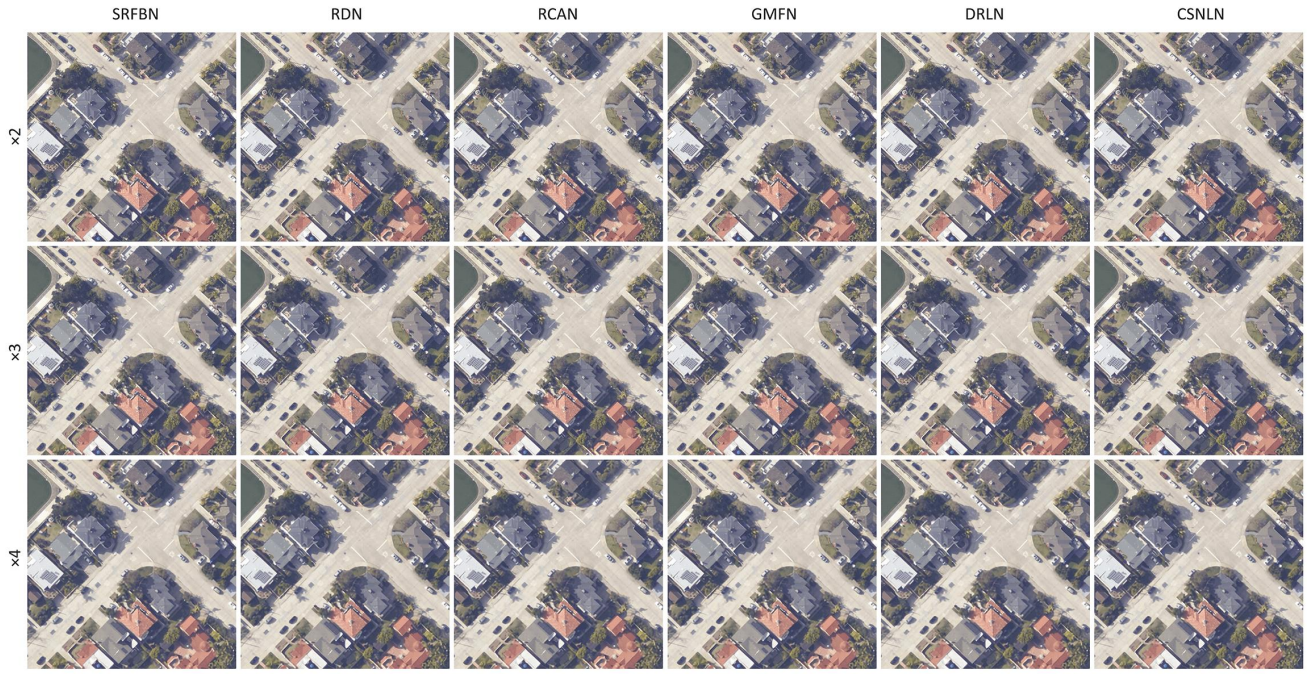

Рисунок 1.7 – Результати збільшення роздільності для деяких моделей глибокого навчання [25]

1.5 Оцінювання результатів формування суперроздільних зображень

Оцінка результатів розв’язання задачі SISR полягає у визначенні ступеня подібності згенерованих суперроздільних зображень до оригінальних високороздільних за різними критеріями. Цей процес поєднує використання математичних методів (кількісних та перцепційних показників) та візуальну оцінку людиною [26].

Кількісні показники оцінюють піксельну похибку між двома зображеннями. Це універсальні метрики, що легко обчислюються та інтегруються у програмні продукти, а також дозволяють отримати

об’єктивний результат для дуже різних зображень. Утім, піксельна точність може свідчити лише про загальну відповідність контурів та текстур, а не їхню візуальну реалістичність. А отже, можна отримати практично прийнятні показники для розмитих або, навпаки, надто контрастних зображень. Наявність локальних артефактів та спотворень також може бути суттєво применшена [26].

До кількісних показників оцінки результатів розв’язання задачі SISR відносять наступні:

1) MSE (англ. Mean Squared Error): розраховує середньоквадратичну помилку між пікселями SR- і HR-зображень (також розглядається квадратний корінь (Root) цього значення RMSE, що спрощує інтерпретацію відхилень) [26];

2) MAE (англ. Mean Absolute Error): розраховує середню абсолютну похибку між SR і HR, є менш чутливою до великих відхилень [26];

3) PSNR (англ. Peak Signal-to-Noise Ratio): вимірює відношення сигналу до шуму, оцінюючи точність піксельної реконструкції [27];

4) SSIM (англ. Structural Similarity Index Measure): оцінює структурну схожість, враховуючи яскравість, контраст і текстури [28];

5) CIEDE2000 [26]: вимірює різницю в кольорах між SR- і HR-зображеннями на основі колориметричних стандартів.

Перцепційні показники оцінюють структурну подібність двох зображень шляхом порівняння різних критеріїв, вивчених та оцінених попередньо навченими глибокими згортковими мережами. Вони є наближеними до візуального сприйняття людиною і здатні виявити артефакти, спотворення, розмитість. Утім, точність оцінювання напряму залежить від точності вивчення критеріїв мережею. Навчання такої мережі, яка буде ефективною на всьому наборі, зокрема на тестовій вибірці – окреме складне завдання, тому найчастіше використовують попередньо навчені мережі, які не завжди є універсальними [26].

До перцепційних показників оцінки результатів розв’язання задачі SISR відносять наступні:

1) LPIPS (англ. Learned Perceptual Image Patch Similarity): вимірює схожість зображень у просторі ознак, вилучених попередньо натренованою НМ [29];

2) NIQE (англ. Naturalness Image Quality Evaluator): оцінює природність зображення без необхідності референтного зображення, спираючись на відомі розподіли природних сцен, вилучені за допомогою НМ [30];

3) PIQE (англ. Perceptual Image Quality Evaluator): комбінує NIQE та результати суб’єктивних оцінок для кількісного представлення візуальної якості [31];

4) FID (англ. Fréchet Inception Distance): вимірює схожість між розподілами ознак оригінальних та сформованих зображень, вилучених інспекційною мережею [26].

5) BRISQUE (англ. Blind/Referenceless Image Spatial Quality Evaluator): оцінює якість зображення без референтного зображення, використовуючи статистики локальних ознак [32].

З огляду на вищевказані недоліки кількісних та перцепційних показників, до математичних методів оцінювання результатів SISR додають **візуальну оцінку людиною**. Для її об’єктивності до процесу залучають кількох експертів. Щоб отримати єдиний результат використовують статистичний показник MOS (Mean Opinion Score) – зважене середнє оцінок групи експертів за встановленою шкалою якості [26].

1.6 Висновки до розділу 1

У даному розділі було показано роль задач формування зображень у сучасному світі та обґрунтовано важливість, актуальність задачі SISR.

Також було здійснено порівняння ефективності породжувальних моделей та методів глибокого навчання як методів розв'язання цієї задачі, перелічено переваги та недоліки популярних моделей кожного класу, на основі чого можна обрати генеративно-змагальні мережі, згорткові, рекурсивні та залишкові глибокі мережі для подальшого розгляду.

На основі огляду методів оцінювання результатів було зроблено висновок щодо необхідності включення як кількісних, так і перцепційних показників, а також візуального оцінювання людиною.

РОЗДІЛ 2 МОДЕЛІ ЗБІЛЬШЕННЯ РОЗДІЛЬНОСТІ ТА ПОКАЗНИКИ ЇХНЬОЇ ЯКОСТІ

У даному розділі описується архітектура відомих сучасних моделей збільшення роздільності зображень, якою вона була представлена в оригінальних статтях авторів. Також наводяться показники оцінки якості цих моделей.

2.1 Модель SRCNN

Вперше модель SRCNN була представлена у роботі «Image Super-Resolution Using Deep Convolutional Networks» Ч. Донга та співавторів [15]. Основною метою роботи було продемонструвати здатність глибоких згорткових мереж ефективно реконструювати високоякісні зображення, зменшуючи час обробки порівняно з традиційними методами.

В основі архітектури SRCNN лежить послідовність згорткових шарів (рис 2.1), яка створює глибину даної мережі.

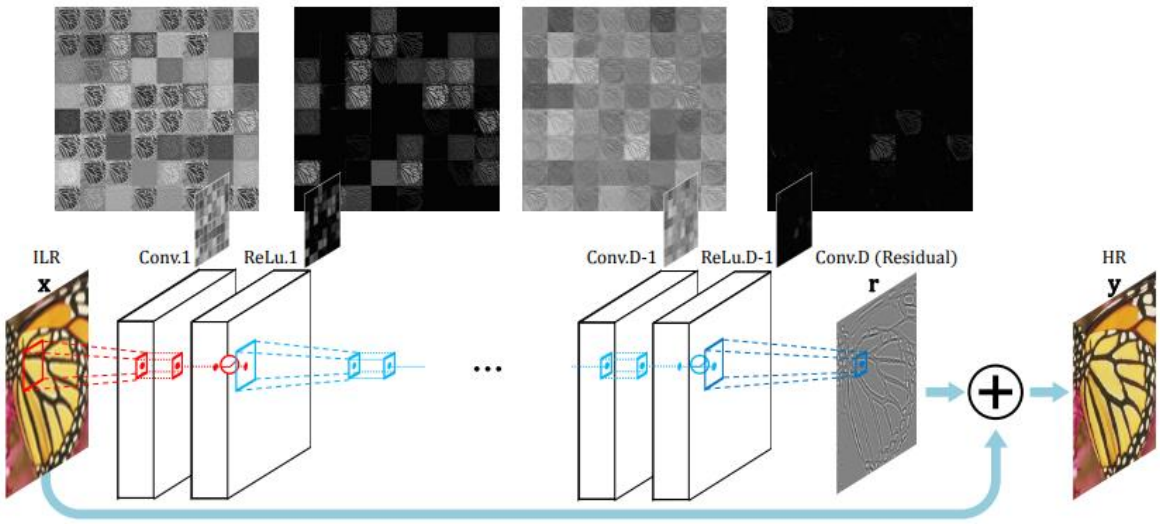

Рисунок 2.1 – Архітектура SRCNN [15]

На початку навчання вхідне зображення I_{LR} масштабується до цільового розміру шляхом бікубічної інтерполяції, що дає мережі змогу працювати безпосередньо в просторі високої роздільної здатності. Процес навчання моделі складається з трьох основних етапів: виділення ознак початковим шаром, нелінійне відображення та реконструкція зображення [15].

На **першому етапі** відбувається виділення базових структурних ознак зображення та їх представлення у багатовимірному просторі ознак через **операцію згортки**. Цей процес описується рівнянням [15]

$$F_1 = \sigma(W_1 * I_{LR} + b_1),$$

де W – ядро (або фільтр) згортки, b_1– зсув, а σ – функція активації (ReLU).

Ядро згортки є матрицею фіксованого розміру, яка послідовно накладається на всі ділянки вхідного зображення. Для кожної позиції фільтр обчислює суму добутків своїх значень та відповідних значень пікселів із області, на яку він накладається (рис. 2.2). Результатом є карта ознак — матриця, яка містить локальну інформацію про певний аспект зображення, наприклад, контури чи градієнти. Кожен фільтр відповідає за виділення однієї карти ознак, а набір усіх карт утворює багатоканальне представлення зображення. Це представлення передається до наступного шару для подальшої обробки (рис. 2.3).

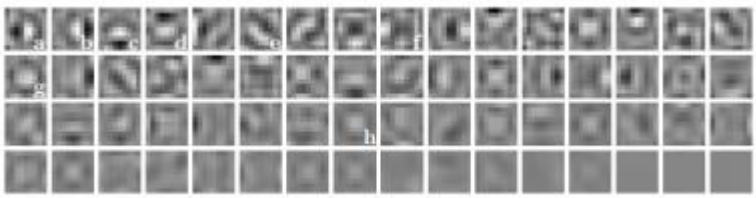

Рисунок 2.2 – Приклад навчання фільтрів першого шару моделі SRCNN [15]

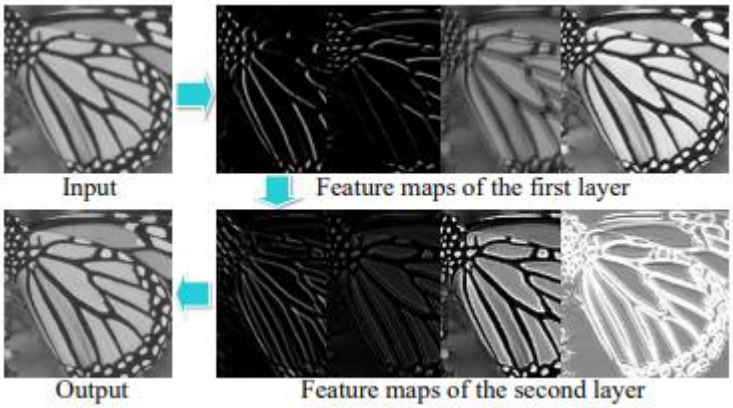

Рисунок 2.3 – Процес передачі карти ознак з одного згорткового шару в інший [15]

Роль функції активації ReLU на цьому етапі полягає в тому, щоб додати нелінійність, дозволяючи моделі вивчити більш складні залежності. ReLU обчислюється як $\sigma(x) = \max(0, x)$, що дозволяє ефективно пропускати лише додатні значення, усуваючи від'ємні.

Другий етап відповідає за нелінійне відображення отриманих ознак у більш складний простір з метою моделювання залежностей між локальними ознаками, враховуючи їх контекст [15]:

$$F_2 = \sigma(W_2 * F_1 + b_2),$$

де W_2 і b_2 – параметри другого шару.

На **третьому етапі** відновлюються кінцеві текстурні та детальні особливості – власне, реконструюється високороздільне зображення I_{SR} через лінійну комбінацію трансформованих ознак [15]:

$$I_{SR} = W_3 * F_2 + b_3,$$

де W_3 і b_3 – параметри третього шару.

Функцією втрат моделі є середньоквадратична помилка MSE, яка забезпечує оптимізацію на піксельному рівні, мінімізуючи різницю між між

пікселями згенерованих суперроздільних зображень I_{SR} та оригінальних високороздільних зображень I_{HR} [15]

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} \left\| I_{SR}^{(i)}(W) - I_{HR}^{(i)} \right\|^2, \quad (2.1)$$

де N – кількість зображень у тренувальному наборі, W – параметри моделі, $I_{SR}^{(i)}(W)$– згенероване зображення для i-го прикладу, а $I_{HR}^{(i)}$– референтне високороздільне зображення.

SRCNN перевершує традиційні методи збільшення роздільності, такі як бікубічна інтерполяція або методи на основі розрідженого представлення, забезпечуючи значно вищу якість відновлення текстур і деталей. Порівняно невелика кількість параметрів у SRCNN (приблизно 57 тисяч) дозволяє досягти реалістичності результату завдяки оптимальному поєднанню шарів [15]. Перевагу SRCNN надають в умовах економії ресурсів та прагненні використати просту універсальну архітектуру, проте для фотореалістичності результату слід використовувати більш складні методи глибокого навчання.

2.2 Модель VDSR

Модель була вперше представленна у роботі «Accurate Image Super-Resolution Using Very Deep Convolutional Networks» Дж. Кіма і співавторів [18]. В основі VDSR лежить підхід до збільшення роздільності зображень за допомогою дуже глибоких згорткових мереж. Основна мета роботи полягала в покращенні точності реконструкції зображень шляхом побудови значно глибшої архітектури порівняно з попередніми методами за рахунок використання залишкових з’єднань, а також розробці ефективної стратегії оптимізації для стабільного навчання настільки глибокої моделі [18].

Архітектура VDSR (рис. 2.4) базується на глибокій згортковій мережі, що складається з 20 згорткових шарів однакового розміру. Кожен шар виконує згорткову операцію з нелінійною функцією активації ReLU.

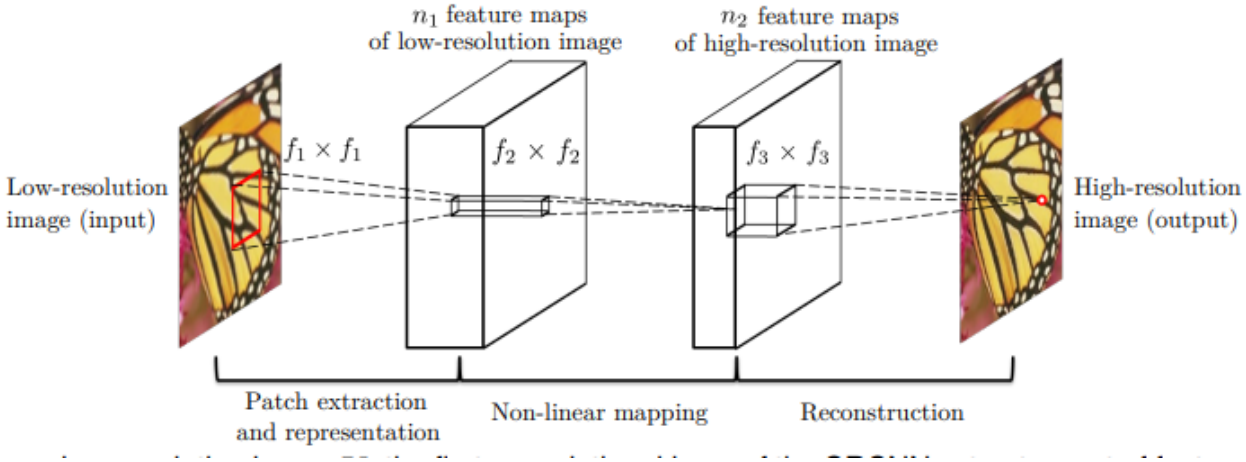

Рисунок 2.4 – Архітектура VDSR [18]

Входом мережі є низькороздільне зображення I_{LR}, яке попередньо масштабується до цільового розміру шляхом бікубічної інтерполяції. Мережа моделює залишкове зображення R, що визначається як різниця між I_{HR} і I_{LR} [18]:

$$R = I_{HR} - I_{LR}.$$

Цільове високороздільне зображення I_{SR} отримується шляхом додавання до низькороздільного зображення згенерованого залишкового зображення [18]:

$$I_{SR} = I_{LR} + R.$$

Вихід кожного згорткового шару визначається як [18]

$$F_l = \mathrm{ReLU}(W_l * F_{l-1} + b_l),$$

де W_l і b_l – ваги та зсуви l-го шару відповідно, а $F_0 = I_{LR}$ – початкове представлення.

У роботу мережі також інтегровано концепцію багатомасштабного навчання, коли для навчання однієї мережі використовувались вхідні зображення різного масштабу зменшення, що дозволяє не навчати окрему модель для кожного варіанту масштабування (рис. 2.5) [18].

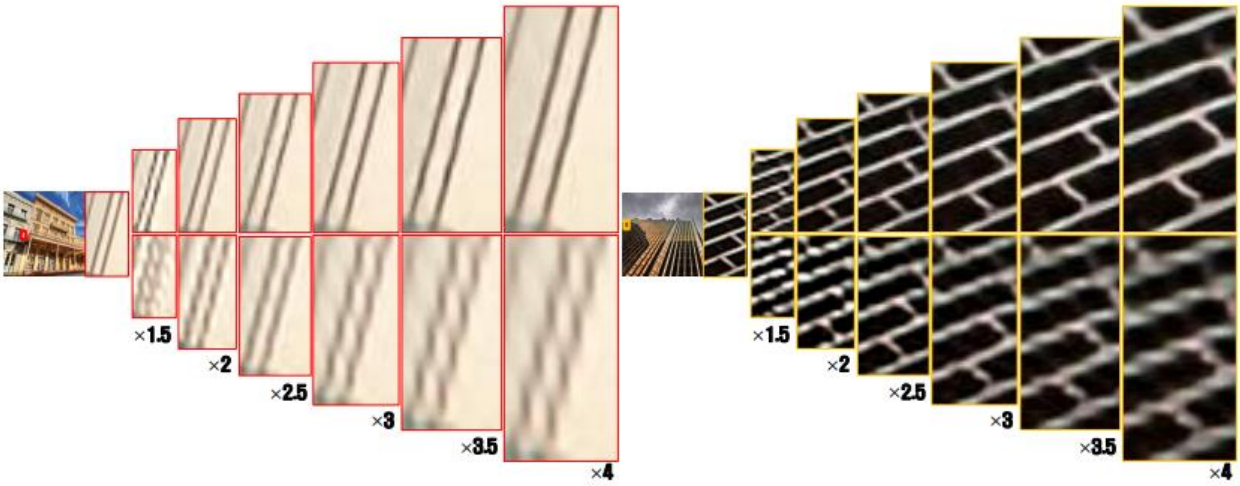

Рисунок 2.5 – Приклад різного масштабу збільшення зображень, виконаного однією і тією ж моделлю VDSR

Функція втрат моделі є класичною середньоквадратичною помилкою між відновленим зображенням I_{SR} та референтним зображенням I_{HR} [18]. Вона описується формулою (2.1).

Для подолання проблеми затухання градієнтів, що виникає через велику глибину мережі, використовується стратегічне нормалізоване навчання з великим значенням коефіцієнта швидкості навчання. Це сприяє швидкій збіжності навіть для дуже глибоких моделей [18].

Отримані за допомогою моделі VDSR суперроздільні зображення демонструють вищу схожість із зображеннями реального світу порівняно з результатами попередніх моделей. Це зумовлено здатністю VDSR моделювати складні залежності між пікселями, що є ключовим фактором для досягнення високої якості реконструкції. Залишкова архітектура дозволяє моделі зосередитися на дрібних деталях, які відіграють важливу роль у відтворенні

реалістичних текстур та контурів під час навчання. Використання багатомасштабного навчання позитивно впливає на точність відтворення цих деталей, забезпечуючи більш природну та деталізовану реконструкцію.

2.3 Модель DRCN

Модель вперше була запропонована у статті «Deeply-Recursive Convolutional Network for Image Super-Resolution» [17] творців VDSR (див. п. 2.2). У роботі представлено новаторський підхід до рекурсивної реконструкції високороздільних зображень глибокою згортковою мережею, яка багаторазово використовує один і той самий набір параметрів. Досягнення значної глибини без збільшення кількості параметрів сприяє ефективному навчанню та зменшенню обчислювальних ресурсів [17].

Архітектура **базової моделі DRCN** (рис. 2.6) побудована на основі двох основних модулів: рекурсивного згорткового ядра (рис. 2.7) та механізму багатоступеневої реконструкції [17].

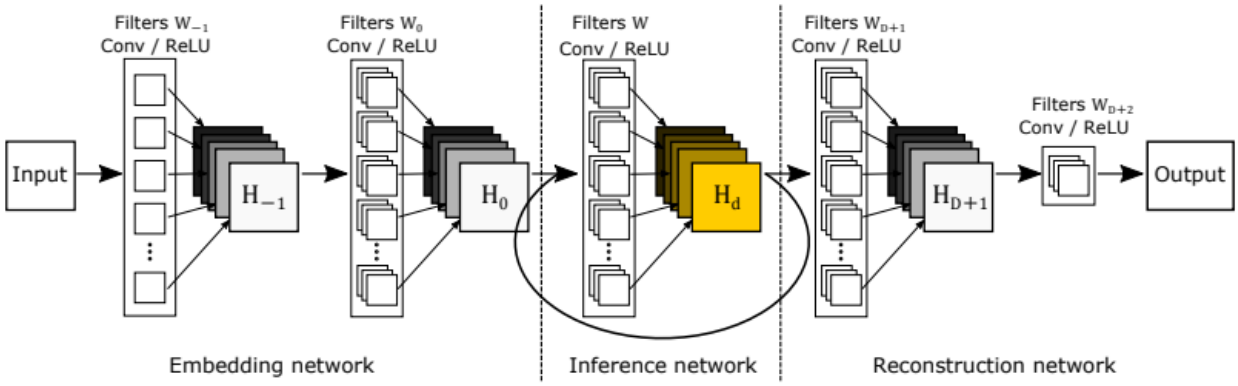

Рисунок 2.6 – Архітектура базової моделі DRCN [17]

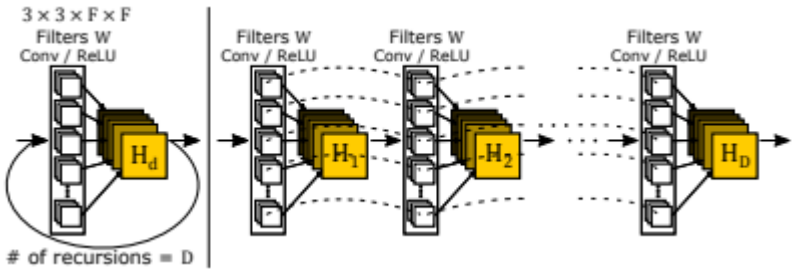

Рисунок 2.7 – Рекурсивний блок у розгорнутому вигляді [17]

Вхідне зображення I_{LR} передається через шар попередньої обробки, що виконує початкову лінійну згорткову трансформацію [17]:

$$F_0 = W_{pre} * I_{LR} + b_{pre},$$

де W_{pre} і b_{pre} – параметри (ядро та зсув) початкового згорткового шару, а $*$ – оператор згортки.

Потім отримане представлення F_0 передається в рекурсивний модуль, який виконує послідовність R рекурсивних згорткових операцій [17]

$$F_n = W_{rec} * F_{n-1} + b_{rec}, \qquad n = 1, \dots, R,$$

де W_{rec} і b_{rec} – параметри згортки, які спільно використовуються на всіх рівнях рекурсії.

Фінальне представлення F_R передається через механізм багатоступеневої реконструкції, який включає один або декілька додаткових згорткових шарів для отримання остаточного високороздільного зображення [17]

$$I_{SR} = W_{out} * F_R + b_{out},$$

де W_{out} і b_{out} – параметри вихідного згорткового шару.

Функція втрат моделі є класичною середньоквадратичною помилкою між відновленим зображенням I_{SR} та референтним зображенням I_{HR} [17]. Вона описується формулою (2.1).

Базова модель з простою архітектурою демонструє високу ефективність, але має обмеження, пов'язані зі затуханням градієнтів під час навчання глибоких рекурсій. Відсутність контролю проміжних прогнозів F_n ускладнює навчання та знижує стабільність моделі, проте відносно мала кількість параметрів залишається є ключовим аргументом для вибору цієї моделі при розв'язанні задачі SISR попри можливі переваги інших підходів.

Авторами DRCN також запропоновано архітектуру **вдосконаленої моделі DRCN** (рис. 2.8) з метою вирішення проблеми затухання градієнтів у глибоких мережах та вибору оптимальної кількості рекурсій [17]. Вона має дві суттєві переваги: рекурсивний контроль та використання .

1. **Рекурсивний контроль** (англ. Recursive Supervision) вирішує проблему затухання градієнтів. Основна ідея полягає в тому, щоб контролювати всі R рекурсій у мережі, забезпечуючи навчання на кожному етапі рекурсивного процесу [17]. На кожному n-му етапі рекурсії модель генерує проміжне передбачення F_n, яке порівнюється з референтним зображенням I_{HR} у процесі оптимізації функції втрат на основі MSE

$$L(F_n, I_{HR}) = \frac{1}{N}\sum_{i=1}^{N}\left\|F_{n,i} - I_{HR,i}\right\|^2.$$

Багатоступенева функція втрат, яка враховує всі проміжні результати $L(F_n, I_{HR})$, визначається як [17]

$$L_{DRCN} = \frac{1}{R}\sum_{n=1}^{R} L(F_n, I_{HR}).$$

Такий підхід гарантує, що всі рівні рекурсії роблять внесок у процес навчання, зменшуючи ймовірність деградації результатів по мірі заглиблення у модель. Вплив віддалених рівнів на загальну реконструкцію обмежується, тому вдосконалена DRCN досягає кращої узгодженості між проміжними рівнями та вихідним результатом.

2. **Пропуск з'єднань** (англ. Skip-Connection) використовується для покращення передачі інформації між вхідним зображенням I_{LR} та вихідним I_{SR}. I_{LR} напряму додається до модуля реконструкції, який використовується на кожному етапі рекурсії [17]:

$$F_n = W_{rec} * F_{n-1} + b_{rec} + I_{LR}.$$

де W_{rec} і b_{rec} – вже згадані при описі базової моделі параметри модуля реконструкції. Цей механізм дозволяє зберігати важливу інформацію про вхідний сигнал і сфокусувати мережу на відновленні деталей.

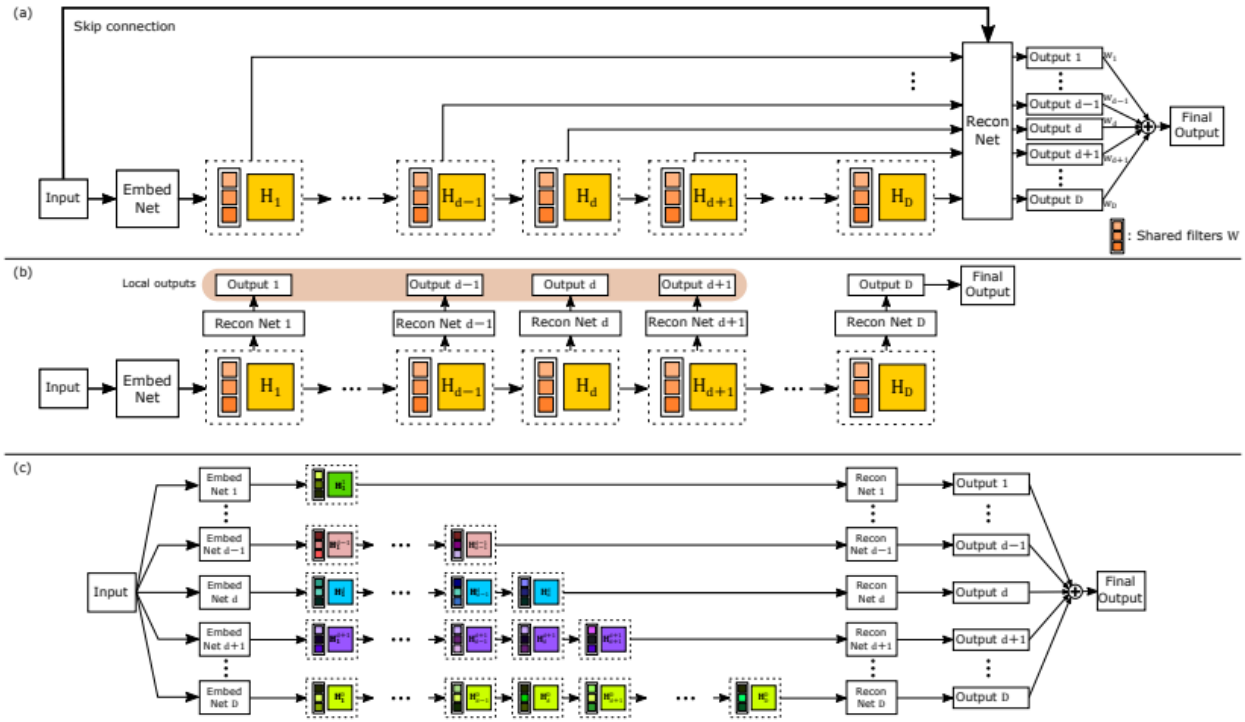

Рисунок 2.8 – Архітектура Advanced DRCN [17]

В подальшому під абревіатурою DRCN розглядатиметься саме **вдосконалена модель**. Порівняно з базовою моделлю, вона стабільно навчається та ефективно оптимізується навіть в умовах великої кількості рекурсій, оскільки її функція втрат враховує всі проміжні прогнози. Цій моделі також не властива надмірна увага до передачі сигналів, що дозволяє зберігати акцент на ключовій інформації про вхідне зображення. Адаптація до оптимальної кількості рекурсій меншує залежність моделі від фіксованої глибини [17].

З рисунку 2.9 видно, що DRCN є революційним підходом для деталізації складних структур (текстів, дрібних текстур) порівнянно з відомими на час представлення моделі підходами, що також підтверджується вищими значеннями показників PSNR та SSIM.

Рисунок 2.9 – Приклад якісної деталізації складних текстур DRCN [17]

2.4 Модель SRGAN

Модель SRGAN була вперше представлена у статті «Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network» К. Ледіга й співавторів [9]. Мета SRGAN – породження фотореалістичних високороздільних зображень – досягається шляхом врахування перцепційної подібності деталей та текстур через запропоновану авторами комбіновану функцію втрат.

Архітектура SRGAN (рис. 2.10) складається з багатошарового генератора, який фокусується на створенні високоякісного зображення, та дискримінатора, який навчається розрізняти реальні й згенеровані зображення.

Окрім нової функції втрат, що дозволяє враховувати важливі для людського сприйняття високорівневі ознаки, якість породжених зображень забезпечується також класичним для моделей класу GAN змаганням генератора з дискримінатором: генератор вдосконалює свою здатність породжувати фотореалістичні зображення за рахунок інтегрування в його функцію втрат змагальної компоненти, яка містить оцінку дискримінатором ймовірності того, наскільки реалістичними є згенеровані зображення. Як і для всіх GAN, SRGAN вимагає того, щоб обидві складові мережі були достатньо сильними та добре збалансованими – інакше не відбудеться ефекту змагання через пригнічення однієї мережі іншою, що у випадку слабкості генератора приведе до слабко збіжного процесу навчання з низькою ефективністю, у випадку дискримінатора – швидкої збіжності, що не відповідає дійсності.

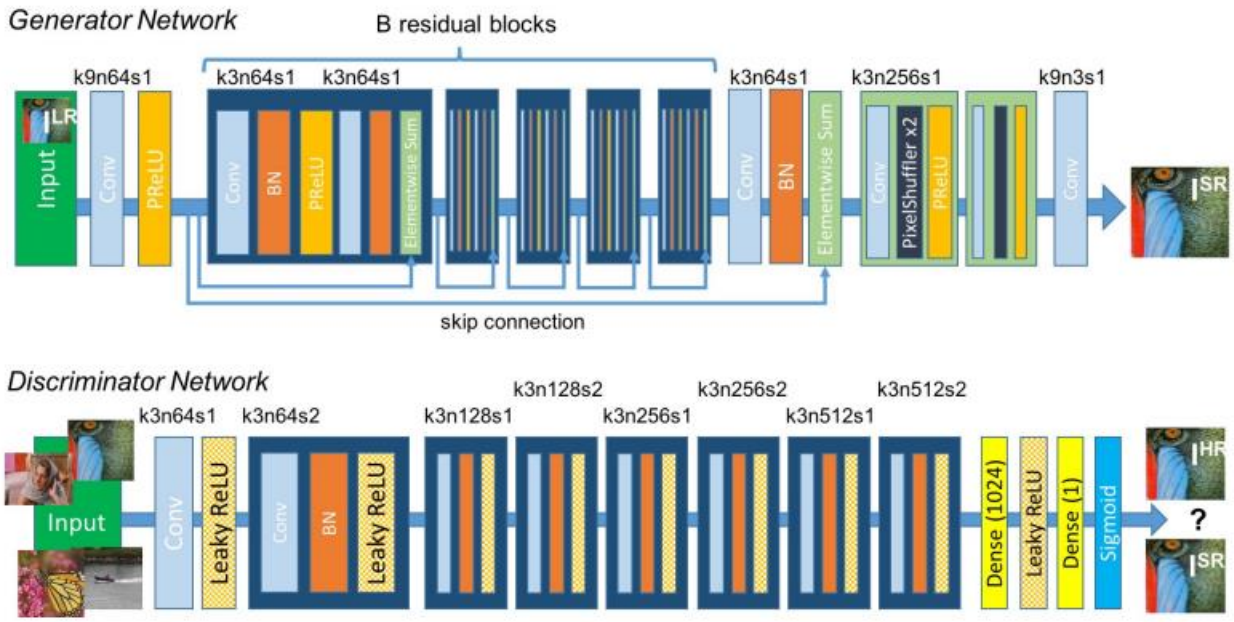

Рисунок 2.10 – Архітектура SRGAN [9]

Генератор SRGAN побудований на основі залишкових блоків з використанням архітектури, подібної до ResNet. Вхідне зображення I_{LR} проходить через початковий згортковий шар з нелінійною функцією активації ReLU [9]:

$$F_0 = ReLU(Conv(I_{LR})).$$

Далі знаходиться серія з B залишкових блоків [9], кожен i-ий з яких описується формулою

$$F_i = F_i - 1 + H(F_i - 1),$$

де H – нелінійна трансформація, що складається з двох згорткових шарів з нормалізацією і активацією ReLU між ними. Наприкінці всіх залишкових блоків міститься ще один згортковий шар, на виході результат додається до початкового вхідного представлення F_0 [9]:

$$F_{out} = F_0 + Conv(F_B).$$

Для отримання суперроздільного зображення виконується операція збільшення роздільності крізь субпіксельні шари (Pixel-Shuffle) [9]:

$$I_{SR} = SubPixel(Conv(F_{out})),$$

де SubPixel – модуль для просторового підвищення роздільної здатності.

Дискримінатор є згортковою мережею, що приймає на вхід випадкове зображення: справжнє I_{HR} або згенероване I_{SR} – і виконує послідовність згорткових операцій з активацією LeakyReLU (коефіцієнт нахилу 0.2) і зменшенням розміру карти через операцію пулінгу ($stride = 2$) [9]. На виході дискримінатор видає ймовірність того, що дане зображення є справжнім:

$$P = \sigma(FC)(F_{disc}),$$

де σ – сигмоїдна функція, FC – повнозв'язний шар.

Ключовою відмінністю SRGAN від інших генеративно-змагальних моделей для збільшення роздільності є комбінована функція втрат для генератора, яка поєднує перцепційну втрату змісту та змагальну компоненту.

Перцепційна втрата обчислюється як традиційна втрата змісту, проте не між пікселями зображення, а між його просторовими ознаками, витягнутими за допомогою попередньо навченої мережі [9]. Вона визначається як

$$L_{perceptual} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}\left\|\phi(I_{SR})_{x,y} - \phi(I_{HR})_{x,y}\right\|^2,$$

де ϕ – активації проміжних шарів попередньо навченої мережі, W і H – ширина та висота матриці відповідності.

Слід зазначити, що вибір та налаштування такої мережі є окремим важливим етапом навчання SRGAN, бо від її здатності до влучного виділення ознак напряму залежить якість оптимізації генератора. На рисунку 2.11 наведено приклад зображень, згенерованих різними моделями SRGAN в залежності від попередньо навчених мереж, що використовуватись для обчислення перцепційної втрати. Видно, що у випадку VGG22 деталізація є гіршою за варіант із застосуванням звичайної втрати змісту як середньоквадратична помилка (англ. Mean Squared Error, MSE) між пікселями, а генератор, оптимізований за допомогою VGG54 є ефективнішим за його аналог з ResNet.

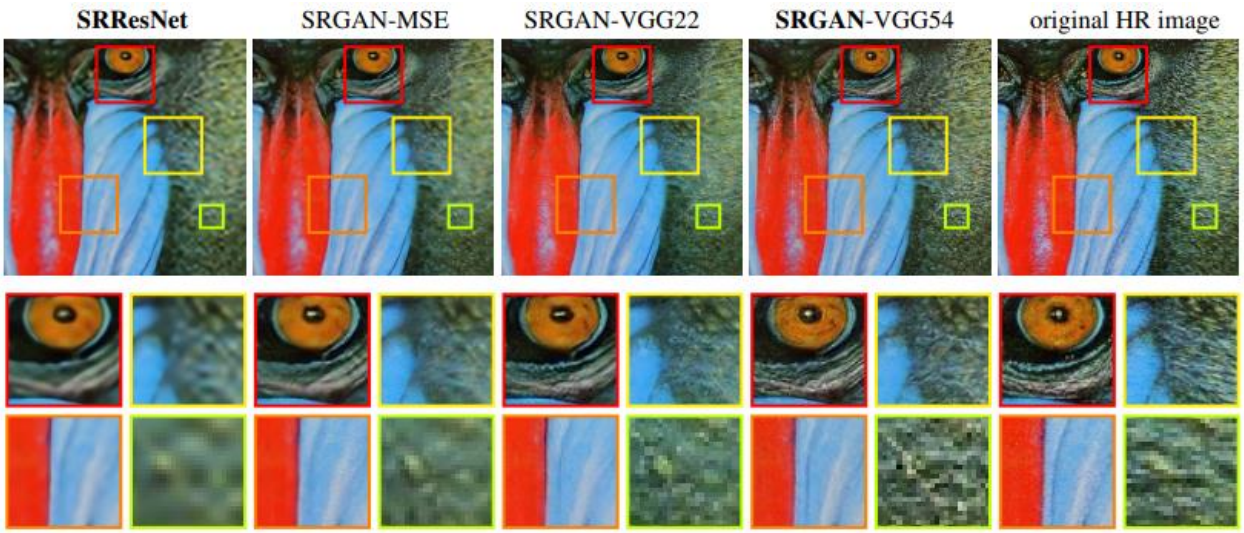

Рисунок 2.11 – Порівняння результатів збільшення роздільності моделями SRGAN з різними видами перцепційної втрати [9]

Змагальна втрата для генератора визначається як [9]

$$L_{GAN} = -\mathbb{E}_{I_{SR}\sim p_G}[\log D\,(I_{SR})],$$

де $\mathbb{E}$ – математичне сподівання (усереднення за всіма зразками набору), p_G – розподіл згенерованих генератором зображень, $D(I_{SR})$ – ймовірність того, що дискримінатор помилково класифікує згенероване зображення як справжнє.

Фінальна функція втрат для генератора SRGAN є сумою перцепційної та зваженої змагальної втрати [9]

$$L_G = L_{perceptual} + \lambda L_{GAN},$$

де λ – коефіцієнт, що збалансовує важливість змагальної втрати.

Бінарна крос-ентропія (англ. Binary Cross-Entropy, BCE) між двома зображеннями обчислюється як

$$BCE = -\frac{1}{H \cdot W} \sum_{x=1}^{H} \sum_{y=1}^{W} (I_{HR}(x,y) \cdot \log I_{SR}(x,y)) .$$

Функція втрат дискримінатора SRGAN базується на бінарній крос-ентропії [9]:

$$L_D = -\mathbb{E}_{I_{HR} \sim p_{data}}[\log D\,(I_{HR})] - \mathbb{E}_{I_{SR} \sim p_G}\big[\log\big(1 - D(I_{SR})\big)\big],$$

де p_{data} – розподіл справжніх зображень, $D(I_{HR})$ – ймовірність того, що дискримінатор правильно класифікує справжнє високороздільне зображення.

2.5 Показники якості

Для математичної оцінки якості збільшення роздільності зображень потрібно використовувати як піксельні, так і перцепційні показники (див. п. 1.5). З огляду на це для об’єктивності результатів оцінювання якості моделей оберемо наступні три показники.

1. **PSNR** є піксельною метрикою, яка вимірює відношення максимальної можливої потужності сигналу до потужності шуму, який впливає на якість реконструйованого зображення [27]. Порівняно з середньоквадратичною похибкою, яка є базовою метрикою для оцінки різниці між пікселями двох зображень, PSNR є більш зручною в інтерпретації, оскільки вимірюється в децибелах (dB), а також враховує максимальне значення інтенсивності пікселів, що дозволяє легко порівнювати якість різних зображень незалежно від їхніх масштабів або глибини кольору [26].

MSE між пікселями відновленого I_{SR} та референтного I_{HR} зображень обчислюється як [27]

$$MSE = \frac{1}{H \cdot W} \sum_{x=1}^{H} \sum_{y=1}^{W} \left(I_{HR}(x, y) - I_{SR}(x, y) \right)^2,$$

де H і W – висота і ширина зображення відповідно, x і y – порядковий номер пікселя за висотою та шириною. Цей запис є деталізацією виразу (2.1).

PSNR обчислюється на основі MSE за формулою [27]

$$PSNR = 10 \cdot \log_{10} \left(\frac{L^2}{MSE} \right), \quad (2.2)$$

де L – максимальне значення в динамічному діапазоні пікселів. Для зображень з 8-бітною глибиною на канал, якими є всі зразки з набору DIV2K (див. п. 3.2), $L = 255$.

Вищі значення вказують на меншу помилку реконструкції та кращу кількісну відповідність сигналів зображень [27].

2. **SSIM** – це перцепційний показник, що має кількісну природу. Він оцінює структурну подібність між I_{SR} та I_{HR} у локальних вікнах на основі статистичних понять, тож дозволяє оцінити не тільки яскравість, а й контрастність та загальну структуру зображень [28]. SSIM обчислюється як

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

де μ_x, μ_y – середні значення інтенсивності пікселів у I_{SR} та I_{HR}, σ_x^2, σ_y^2 – дисперсії інтенсивності, σ_{xy} – коваріація між зображеннями; $C_1 = (K_1\ L)^2$ і $C_2 = (K_2\ L)^2$ – малі сталі значення для стабілізації, де $K_1 \ll 1$ і $K_2 \ll 1$ зазвичай обирають як 0.01 та 0.03 відповідно, L є максимальним значенням пікселів, як у формулі (2.2).

На практиці значення SSIM варіюються між 0 до 1, де 1 вказують на повну структурну подібність, 0 відповідає за випадковість. Слід зазначити, що значення SSIM можуть бути гіршими за випадкові, тому можливі значення до -1, але такі випадки неможливі для SISR [28].

3. **LPIPS** є перцепційним показником, який вимірює відмінності між зображеннями на основі глибоких ознак, отриманих з попередньо навченої нейронної мережі. Основна ідея LPIPS полягає у спробі наблизитись до візуального сприйняття людиною на рівні деталей [29].

Для обчислення LPIPS спочатку витягуються ознаки з проміжних шарів мережі, зазвичай VGG або AlexNet. Нехай ϕ_l – активація цієї мережі на l-му шарі для зображення I. Тоді LPIPS обчислюється як [29]

$$LPIPS(I_{SR}, I_{HR}) = \sum_l w_l \cdot \frac{\|\phi_l(I_{SR}) - \phi_l(I_{HR})\|_2^2}{H_l \cdot W_l \cdot C_l},$$

де H_l, W_l, C_l – висота, ширина та кількість каналів відповідної карти ознак, а w_l – ваговий коефіцієнт, який коригує внесок різних шарів.

LPIPS повертає значення від 0 (ідеальна відповідність) до 1 (максимальна відмінність). Хоч LPIPS і є чутливим до текстур і деталей за рахунок врахування високорівневих семантичних особливостей, об’єктивність

отриманого значення напряму залежить від якості результатів навчання мережі розпізнавати ознаки зображень (див. п. 2.4).

2.6 Висновки до розділу 2

У даному розділі було детально розглянуто принцип роботи та архітектуру популярних моделей для розв'язання задачі SISR: SRGAN, VDSR, DRCN та SRCNN – у першому представленні розробниками. Було виділено переваги використання цих моделей над іншими з точки зору різних аспектів, таких як фотореалістичність результату SRGAN, швидша збіжність та реалістична реконструкція на основі VDSR, ефективність використання обчислювальних ресурсів завдяки меншій кількості параметрів DRCN за рахунок використання рекурсії, а також простоту й універсальність SRCNN.

Ці теоретичні дані буде використано для проведення практичних експериментів: власної реалізації перелічених моделей та застосування технології донавчання з метою визначення оптимальної моделі.

РОЗДІЛ 3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

У даному розділі наводиться постановка задачі для експерементальних досліджень та огляд набору даних, що використовується для навчання моделей та оцінювання результатів. Описуються проведені експерименти з різними типами породжувальних моделей та методів глибокого навчання, наводяться табличні та графічні результати досліджень та проводиться їх аналіз з метою визначення оптимальної за сукупністю факторів моделі для задачі SISR.

3.1 Постановка задачі

Позначимо вихідне зображення низької роздільної здатності як $I_{LR} \in$ $\in \mathbb{R}^{H \times W \times C}$, де H – висота, W – ширина, а C – кількість каналів зображення (наприклад, RGB). Якщо $I_{HR} \in \mathbb{R}^{H' \times W' \times C}$ – відповідне йому зображення високої роздільної здатності ($H' = 4H, \quad W' = 4W$), тоді задача суперроздільності зображення полягає у пошуку такого відображення

$$f: I_{LR} \to I_{HR}, \tag{3.1}$$

яке забезпечить максимально точне відновлення деталей зображення I_{HR} на основі інформації із зображення I_{LR}.

Слід зазначити, що відображення (3.1) є лише засобом формалізації задачі, оскільки воно може позначати різні процеси в залежності від методу збільшення роздільності, що описується. Наприклад, для GAN (3.1) позначає вивчений генератором та уточнений за допомогою дискримінатора розподіл деталей, для глибоких мереж – ознаки деталізації, виділені в процесі їх

навчання, який також має свої особливості в залежності від виду моделі (детальніше про відмінності між методами див. у п. 1.4).

Реалізацією відображення (3.1) є модель $f_\theta \in F$, де θ – параметри цієї моделі, F – множина всіх моделей SISR. Цільове суперроздільне зображення є виходом цієї моделі та результатом розв'язання задачі:

$$I_{SR} = f_\theta(I_{LR}).$$

Оптимізаційна задача для кожної моделі з F передбачає знаходження таких параметрів θ моделі, за яких значення функції втрат L, яка вимірює різницю між I_{HR} та I_{SR}, є мінімальним:

$$\theta^* = \arg\min_\theta L(I_{HR}, I_{SR}).$$

Нехай $F^* \subset F$ – підмножина моделей з оптимізованими параметрами. Якщо A – набір показників якості моделей з F^*, то A^* – набір направлених показників, елементи якої обчислюються за формулою

$$a_i^* = \begin{cases} a_i(I_{HR}, I_{SR}), \text{якщо } a_i \to min \\ -a_i(I_{HR}, I_{SR}), \text{якщо } a_i \to max' \end{cases}$$

Метою задачі SISR є знаходження такої моделі $f_\theta^* \in F^*$, для якої значення всіх показників з набору A^* є мінімальним:

$$f_\theta^* = \arg\min_{f_\theta} A^*.$$

3.2 Огляд набору даних

Набір даних DIV2K (англ. DIVerse 2K resolution dataset, рис. 3.1) [33] був представлений у рамках змагання NTIRE 2017 Challenge on Single Image Super-Resolution, проведеного на конференції CVPR Workshops 2017. Він був зібраний для підвищення ефективності розв’язання задачі SISR, оскільки вирішував присутні на той час проблеми недостатньої різноманітності сцен та кількості зображень у наявних наборах [34].

DIV2K містить 1000 пар зображень низької та високої роздільності розмірністю 2040 пікселів по більшій стороні. До навчальної вибірки належить 800 зразків, до тестової та валідаційної – по 100. Історично тестова вибірка призначалась для оцінювання моделей конкурсантами після навчання, валідаційна – організаторами для визначення переможців, тому містила лише LR-версії, для яких пропонувалось сформувати SR-зображення. Після оприлюднення HR-версій валідаційної вибірки для оцінювання власних моделей можна використовувати обидві вибірки [34].

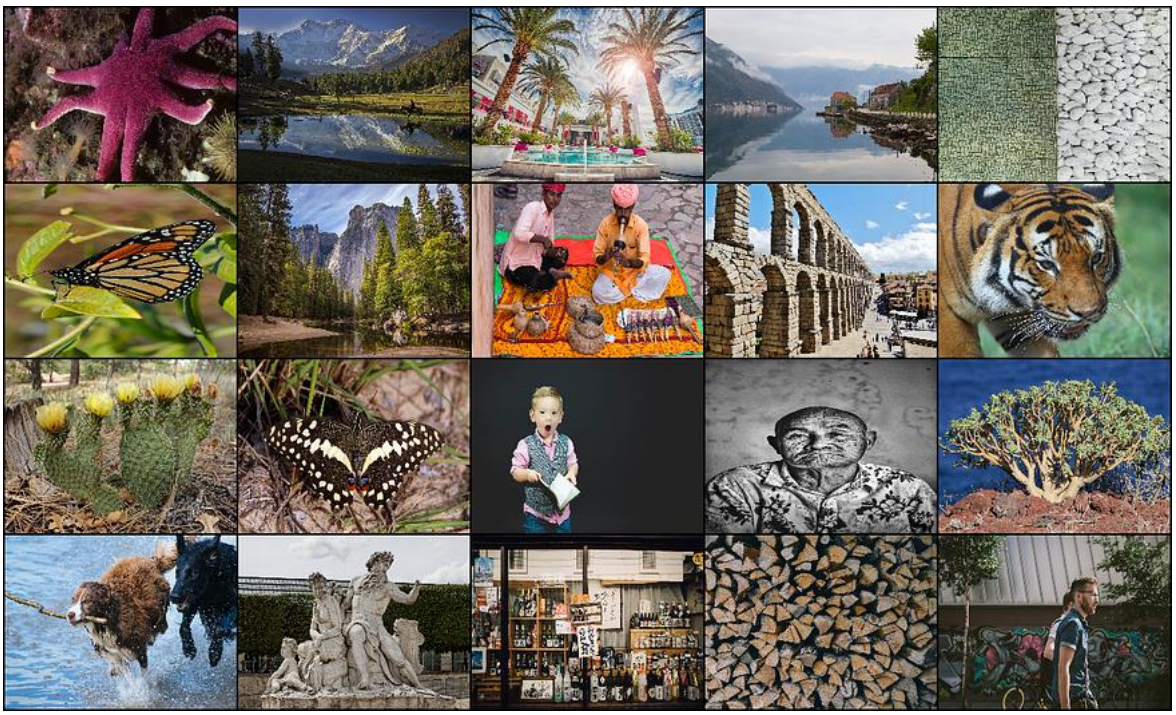

Рисунок 3.1 – Приклад зображень з набору DIV2K [35]

У класичному варіанті DIV2K низькороздільні зображення отримані з високороздільних за допомогою бікубічної інтерполяції. У різних треках змагань було введено більш складні методи деградації, які імітують умови реального світу. У змаганні NTIRE 2017 трек 2 включає зображення, зменшені за методом, що імітує непередбачувані спотворення. У NTIRE 2018 введено ще декілька видів спотворень [34]:

– м'які спотворення: LR-зображення страждають від розмиття руху, додавання дробового шуму та нерівномірного відображення пікселів;

– складні спотворення: м’які спотворення варіюються за інтенсивністю поміж зразків набору, що ускладнює задачу.

Повна версія набору містить різні масштаби зменшення зображень: x2, x3 та x4. Задача збільшення роздільності у меншу кількість разів є легшою через кращу деталізацію LR-зображення (рис. 3.2), проте цей же фактор зумовлює більшу ресурсовитратність процесу навчання.

Рисунок 3.2 – Погіршення якості зображення зі зменшенням роздільної здатності у 2 та 4 рази

У роботі використовується класичний для задачі SISR [35] варіант DIV2K з LR-зображеннями, зменшеними шляхом бікубічної інтерполяції у 4 рази. Всі зображення у наборі розділені на багато квадратних фрагментів меншого розміру з метою збільшення обсягу навчальних зразків та прискорення часу обчислень за рахунок зменшення задіяного об’єму оперативної пам’яті.

3.3 Власна реалізація моделей

Спочатку було здійснено власну реалізацію моделей SRGAN, VDSR та DCRN. Метою цього експерименту є оцінка якості суперроздільних зображень, сформованих моделями з менш складною архітектурою, ніж в оригінальних мереж, представлених дослідниками (див. п. 2.2–2.4 відповідно).

Розробка програмних рішень здійснювалась у середовищі Jupyter Notebook засобами мови програмування Python з використанням бібліотеки для побудови моделей PyTorch та TensorFlow, а також бібліотеки для візуалізації matplotlib. Навчання моделей проводилось на ПК з прискорювачем Nvidia GeForce RTX 4060.

Опис програмно реалізованих класів, що відповідають архітектурі вищезгаданих моделей або її складовим, наведено у таблиці 3.1.

Таблиця 3.1 – Опис реалізованих класів

Клас	**Опис**
ResidualBlock	Блок залишкового з'єднання, що додає вхідні дані до вихідних для підтримки стабільності градієнтів. Містить два згорткові шари 3×3, шар нормалізації (BatchNorm2d) після кожного згорткового та функцію активації PReLU після першого шару
RecursiveBlock	Блок з одним згортковим шаром 3×3 і активацією ReLU, що може рекурсивно повторюватися для збільшення глибии мережі
Generator_SRResNet	Генератор SRGAN з архітектурою SRResNet, складається з початкового згорткового шару 9×9, п’ять залишкових блоків (ResidualBlock), проміжного згорткового блоку 3×3, блоку підвищення роздільної здатності (два згорткові шари 3×3 з PixelShuffle), та фінального згорткового шару 9×9

Продовження таблиці 3.1

Клас	**Опис**
Discriminator_CNN	Дискримінатор для SRGAN, складається з восьми згорткових шарів 3×3 із збільшенням кількості каналів з нормалізацією (BatchNorm2d) і активацією LeakyReLU, одного шару адаптивного усереднення та двох фінальних повнозв'язних шарів. Розмір фільтру для всіх згорткових шарів – 3×3
VDSR	Складається з початкового згорткового шару, 18 згорткових шарів з активацією ReLU, та вихідного шару, який додає залишок до вхідного зображення. Розмір фільтру для всіх згорткових шарів – 3×3
DRCN	Складається з вхідного згорткового шару, рекурсивного блоку (RecursiveBlock), який повторюється 16 разів, і вихідного згорткового шару. Розмір фільтру для всіх згорткових шарів – 3×3

Запропонований **алгоритм навчання моделей** має наступні кроки.

1. Розбиття набору на навчальну та перевірочну вибірки (вже розподілені в наборі).

2. Ініціалізація ваг моделі (автоматична ініціалізація Хе або Глоро залежно від використаного класу бібліотеки PyTorch).

3. Навчання у задану кількість епох (для породжувальної моделі – 200 з розміром батчу 16, для методів глибокого навчання – 100 з розміром батчу 32) на навчальній вибірці з відслідковуванням значень функції втрат (для SRGAN: генератор – перцепційна втрата як сума змістової (MSE) та адверсаріальної (BCE) втрат, дискримінатор – BCE; для методів глибокого навчання – MSE).

4. Збереження ваг моделі на випадок переривання навчання чи ранньої зупинки (англ. early stopping).

5. Розрахунок часу навчання моделей.

6. Оцінка результатів на тестовій вибірці (для обчислення показника LPIPS використовуватимемо попередньо навчену мережу VGG19 [36], середнє значення показників моделі подамо для 10 випадкових зображень).

Оцінювання моделей проводиться на основі набору кількісних та перцепційних показників: PSNR, SSIM та LPIPS (див. п. 2.5) – на тестовій вибірці у наступні кроки:

1) обчислюється значення всіх показників для кожної пари HR- та SR-зображень;

2) розраховується загальний результат для моделі за кожною метрикою як середнє значення за випадковою підвибіркою.

З метою економії ресурсів запропоновано **адаптивний підхід** до застосування цього алгоритму, що передбачає обчислення показників на випадковій малій підвибірці. Слід перевірити, аби отриманий результат був послідовним на кількох таких підвибірках – інакше збільшуємо розмір підвибірок для досягнення цієї умови і лише у найгіршому випадку повертаємось до необхідності обчислень на повному тестовому наборі.

У процесі аналізу результатів також необхідно звернути увагу на значення метрик для бікубічної інтерполяції (звичайного масштабування): за сукупністю показників метрик модель не має бути гіршою за цей результат.

Критерії оцінювання за кожною з обраних показників наведено у таблиці 3.2. Шляхом візуального аналізу великої кількості породжених зображень та значень показників їхньої якості, а також результатів інших дослідників [14] отримано порогові значення для практично прийнятного та якісного результату, які наведено в останніх двох стовпчиках цієї таблиці.

Таблиця 3.2 – Критерії аналізу метрик для задачі SISR

Показник	**Діапазон значень**	**Практично прийнятний результат**	**Якісний результат**
PSNR↑	[0; +∞)	>20	>30
SSIM↑	[-1; 1]	>0.7	>0.9
LPIPS↓	[0; 1]	<0.3	<0.1

Результати навчання наведено на рисунках 3.3 – 3.5 та у таблиці 3.3.

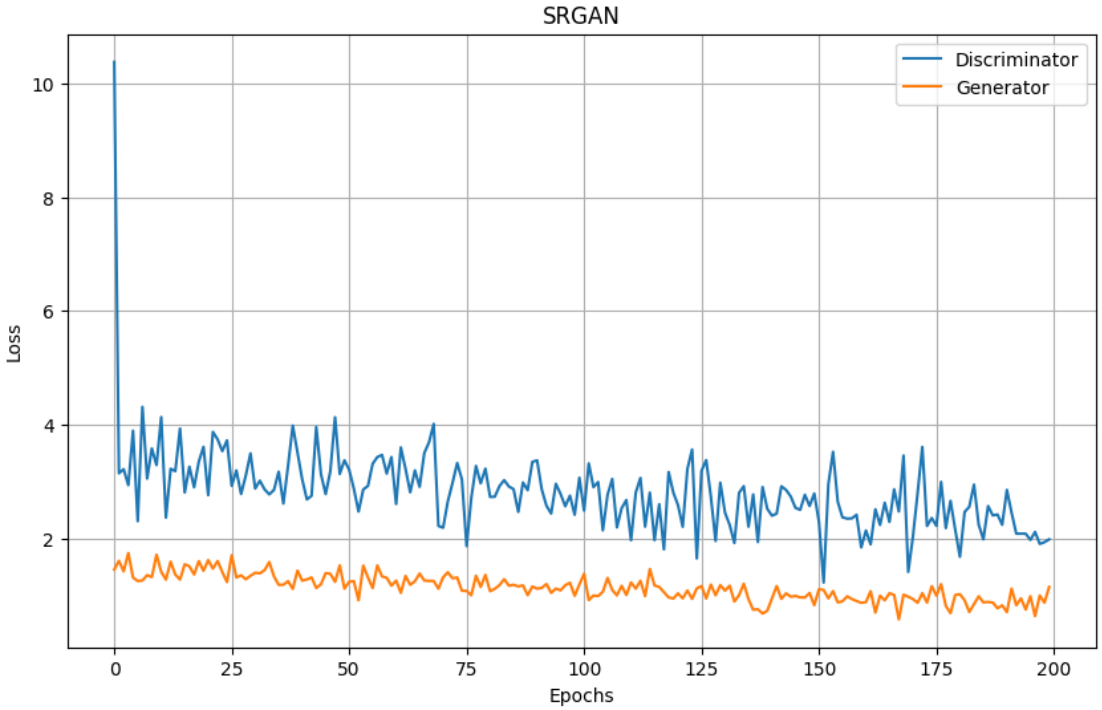

Рисунок 3.3 – Оптимізація функцій втрат генератора та дискримінатора SRGAN

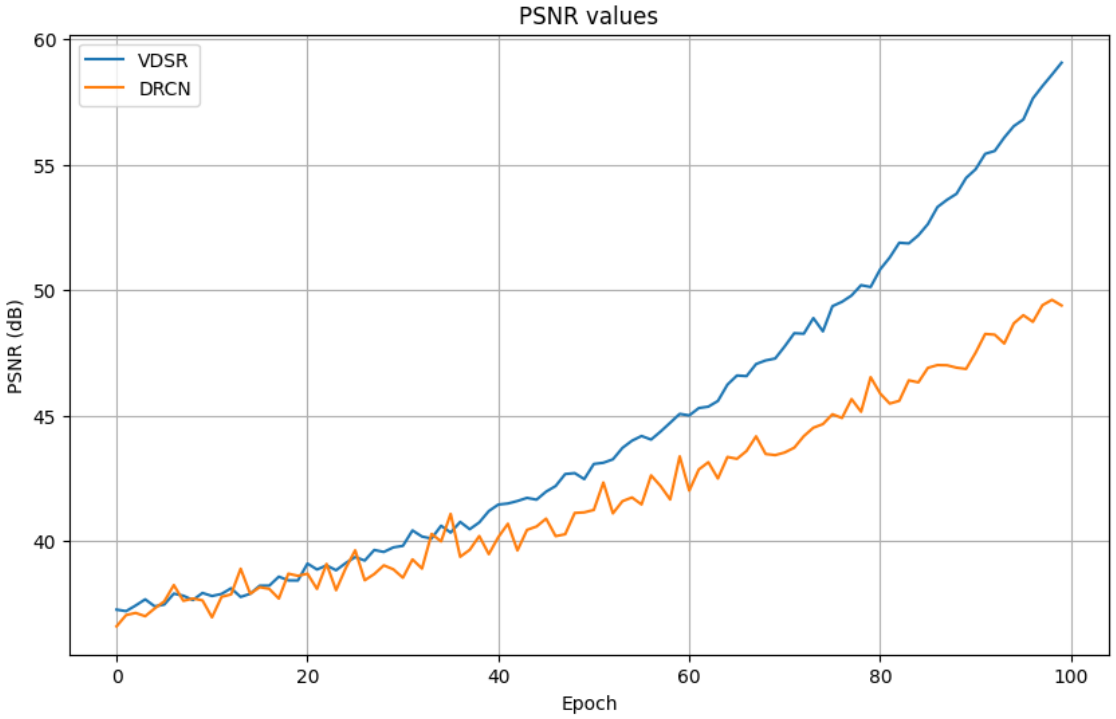

Рисунок 3.4 – Зміна значень PSNR у процесі навчання моделей VDSR та DRCN

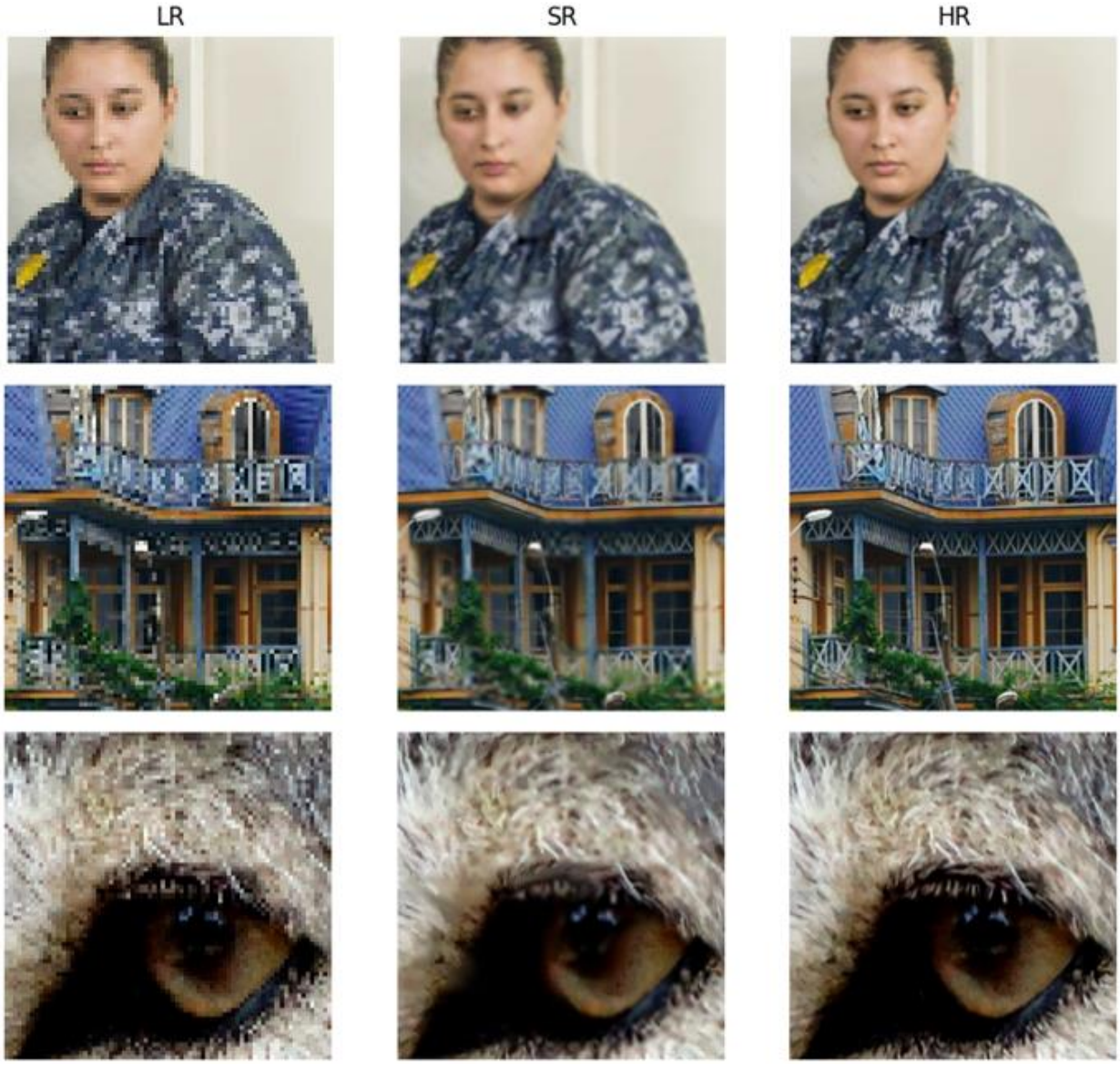

Рисунок 3.5 – Порівняння результатів для власної реалізації моделі VDSR

Таблиця 3.3 – Результати власної реалізації моделей збільшення роздільності для набору DIV2K x4

Моделі	**Показники**			**Час навчання (год.)**
	PSNR↑	SSIM↑	LPIPS↓	
Bicubic	25.80	0.74	0.46	-
SRGAN	24.50	0.71	0.33	32
VDSR	26.73	0.77	0.31	16
DRCN	26.41	0.76	0.37	25

Отримали практично прийнятний результат, який є набагато кращим за масштабування через бікубічну інтерполяцію. Найкраще всього спрацювала модель VDSR. Це свідчить, зокрема, про те, що архітектура DRCN може містити надлишкові рішення, а архітектура SRGAN може бути занадто простою для даної задачі.

У статті [14], що розглядає застосування моделі нормалізованих потоків для збільшення роздільної здатності зображень, наводяться результати, отримані авторами для різних моделей (табл. 3.4, рис. 3.6), які візьмемо за еталонні.

Таблиця 3.4 – Результати моделей збільшення роздільності зі статті [14] для набору DIV2K x4

Моделі	**Показники**						
	PSNR↑	SSIM↑	LPIPS↓	LR-PSNR↑	NIQE↓	BRISQUE↓	PIQUE↓
Bicubic	26.70	0.77	0.409	38.70	5.20	53.8	86.6
EDSR	28.98	0.83	0.270	54.89	4.46	43.3	77.5
RRDB	29.44	0.84	0.253	49.20	5.08	52.4	86.7
RankSRGAN	26.55	0.75	0.128	42.33	2.45	17.2	20.1
ESRGAN	26.22	0.75	0.124	39.03	2.61	22.7	26.2
SRFlow $\tau = 0.9$	27.09	0.76	0.120	49.96	3.57	17.8	18.6

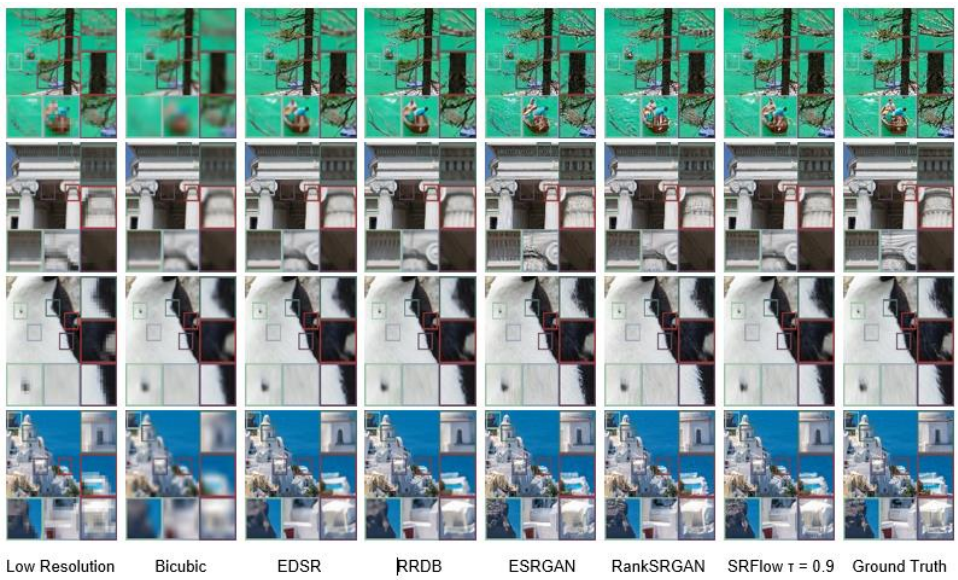

Рисунок 3.6 – Результати моделей зі статті [14] для набору DIV2K x4

Результати показників за власними моделями є дещо гіршим. Враховуючи час їх навчання та розмитість отриманих зображень, можна зробити висновок, що моделі з простою архітектурою доцільно використовувати виключно для прибирання явних пікселів на низькороздільному зображенні.

3.4 Використання технології попереднього навчання моделей

Далі проведемо експеримент з донавчання попередньо навчених (англ. pre-trained) моделей SRGAN, VDSR та SRCNN (див. п. 2.1) на 20 епохах.

Мета цього експерименту – покращити результат генерації та перевірити, чи вдасться отримати результат, кращий ніж у статті [14], і чи матиме місце перенавчання. Щоб не обчислювати показники метрик для всіх зображень навчального набору, фактором перенавчання вважатимемо різницю у якості між згенерованими зразками навчальної та тестової вибірки.

Запропонований **алгоритм навчання** у випадку використання попередньо навчених моделей має наступні кроки.

1. Ретельний підбір попередньо навченої моделі, яка обов’язково має бути націлена на ту ж саму задачу і бажано навчена на великому універсальному наборі даних.
2. Завантаження ваг для обраної моделі, зі значеннями яких продовжуватиметься навчанння.
3. Визначення кількості епох, на яку треба донавчити модель.
4. Тонке налаштування (англ. fine-tuning) моделі: заморозка початкових шарів та додавання нових, які працюють з високорівневими ознаками (залишкові блоки, згортки з малими ядрами, шари нормалізації, Upsampling або PixelShuffle), використання низького коефіцієнта швидкості навчання для забезпечення його стабільності, комбінування основної втрати з

перцепційною (LPIPS) для зосередження на візуальній якості згенерованих зображень.

5. Розрахунок часу навчання моделей.

6. Застосування ранньої зупинки у випадку виникнення ознак перенавчання моделі за показниками метрик.

Ваги попередньо навчених моделей SRGAN (перцепційна втрата на основі VGG19), VDSR, SRCNN взяті з джерел [36], [37] та [38]. Результати експерименту наведено у таблиці 3.5 та на рисунках 3.7 – 3.8.

Таблиця 3.5 – Результати донавчених моделей збільшення роздільності для набору DIV2K x4

Моделі	**Показники**			**Час навчання (хв.)**
	PSNR↑	SSIM↑	LPIPS↓	
Bicubic	25.80	0.74	0.46	-
SRGAN	26.9	0.79	0.16	27
VDSR	28.9	0.84	0.1	11
SRCNN	27.5	0.81	0.12	2

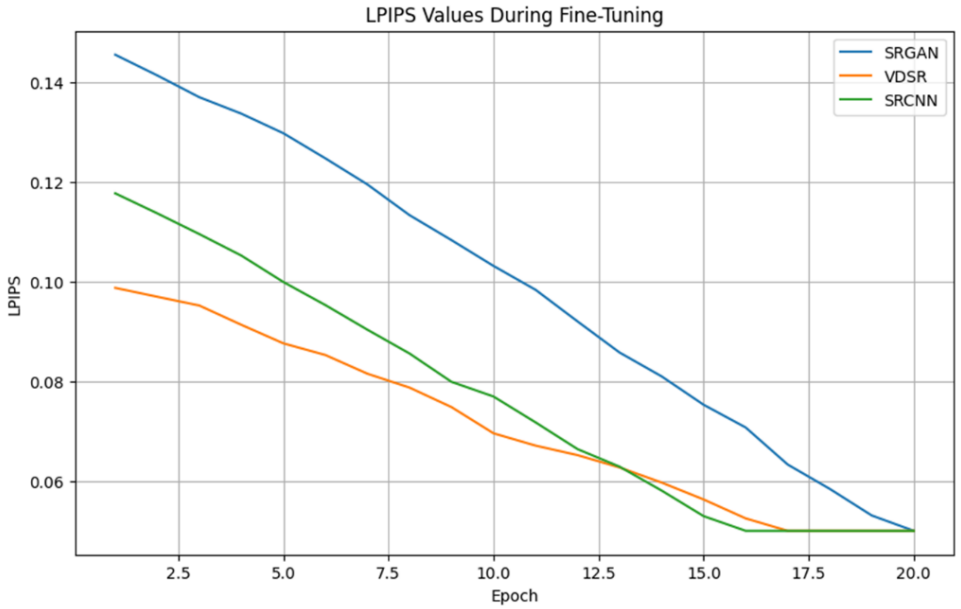

Рисунок 3.7 – Зміна показника LPIPS у процесі донавчання попередньо навчених моделей SRGAN, VDSR та SRCNN

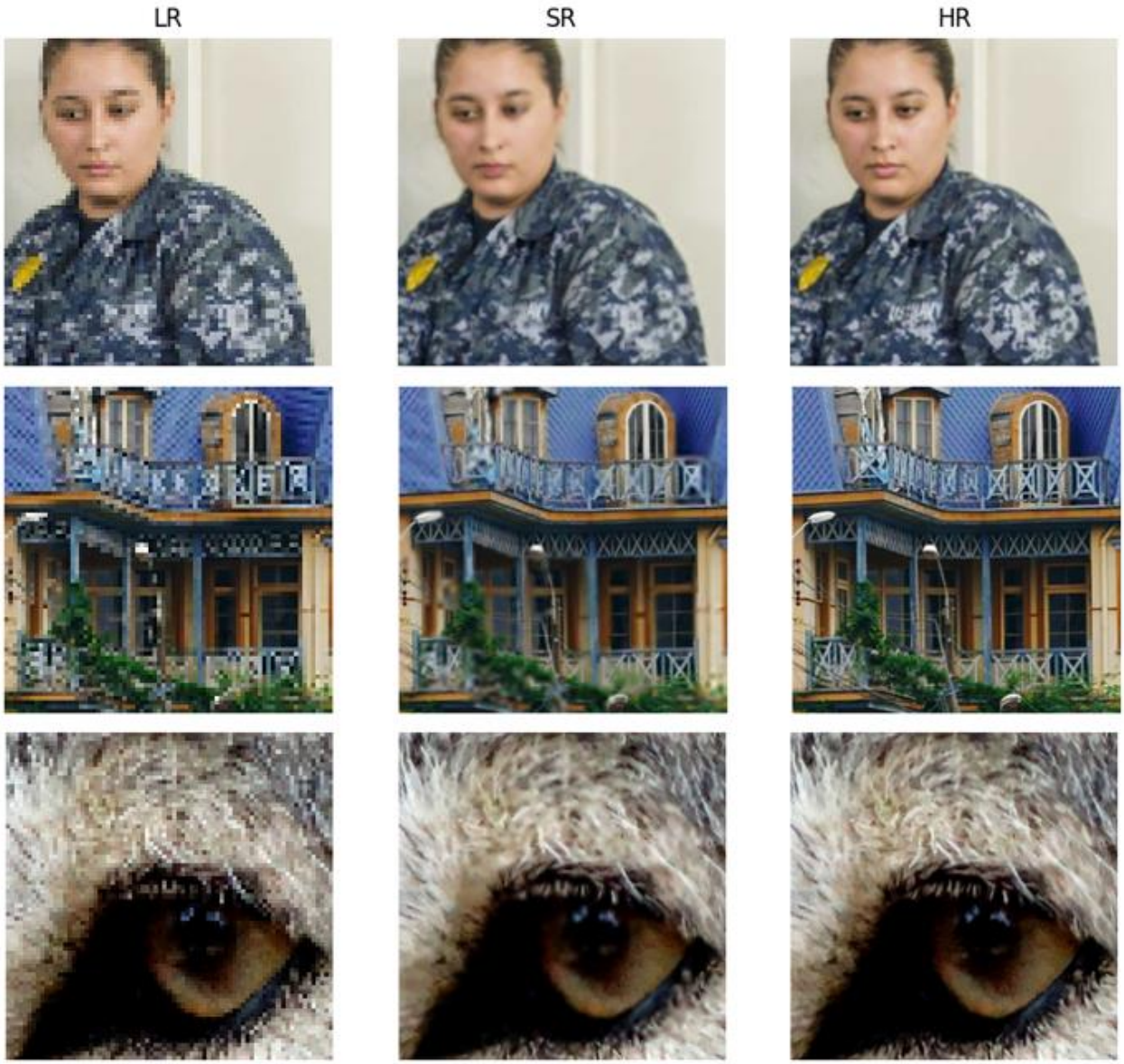

Рисунок 3.8 – Порівняння LR-, SR- та HR-зображень для донавчання попередньо навченої моделі VDSR

Результати навчання знаходяться на рівні результатів, отриманих у статті [14], перенавчання не виявлено. Візуально збільшення роздільності є успішним (рис. 3.8). Найкращий результат досі показує модель VDSR, що свідчить про перевагу використання залишкових з’єднань та загалом глибоких мереж (тобто покращення деталей, а не генерації нових) для даного набору та обраної кількості епох.

Також доцільно перевірити візуальну якість результату для моделі VDSR на зображенні, що не походить з набору DIV2K. Для цього візьмемо

власне зображення, яке відрізняється форматом та типом сцени від тих, що були представлені у наборі DIV2K (рис. 3.6). Результат є подібним до зображень на рисунку 3.9, що свідчить про універсальність моделі.

Рисунок 3.9 – Застосування донавченої моделіVDSR на власному зображенні

3.5 Висновки до розділу 3

У даному розділі було розв'язано задачу SISR і отримано суперроздільні зображення для тестових зразків набору даних DIV2K та власного зображення.

У ході практичних експериментів було випробовано обрані до розгляду моделі SRGAN, VDSR, DRCN та SRCNN як з власною реалізацією, так і з донавчанням. Найкращою серед розглянутих є модель VDSR для обох експериментів.

Запропоновані алгоритми навчання моделей дозволили зекономити обчислювальні ресурси та зменшити час виконання задачі, запропонований алгоритм оцінювання моделей дозволив зменшити час обчисленння показників якості і отримати при цьому об’єктивний результат.

Загалом дослідження показали: перевагу глибоких мереж для реалізованих архітектур даного рівня складності, необхідність використання

попередньо навчених моделей з тонким налаштуванням для кращої деталізації зображень, а також ефективність залишкових з'єднань при вивченні глибокими мережами ознак низько- та високороздільних зображень. загалом показано.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Основною причиною того, що ШІ використовується в певних технічних та природничих галузях не так активно, як в інших, є дефіцит або невідповідна якість специфічних навчальних даних для моделей. Особливо гостро ця проблема постає в задачах комп'ютерного зору через обмежену кількість потрібних високороздільних зображень. Для вирішення цієї проблеми запропонуємо систему, що зберігає мільйони високороздільних зображень, наданих користувачами, під спеціальними тегами і формує з них унікальний набір даних за запитом користувача на основі глибоких фільтрів або ШІ-промпту. На відміну від продуктів конкурентів, цей СП збирає поодинокі знімки, що підвищує ймовірність сформувати базу з важкодоступних зображень, а також є максимально персоналізованим.

У даному розділі розглядається перший етап процесу розробки та ринкового впровадження запропонованого СП – маркетинговий аналіз, що має на меті визначити принципові можливості ринкового впровадження СП та можливих напрямів реалізації цього впровадження. Він складається з наступних кроків [39]:

1) опис ідеї проекту;
2) технологічний аудит цієї ідеї;
3) аналіз ринкових можливостей запуску СП;
4) розроблення ринкової стратегії проекту;
5) розроблення маркетингової програми СП.

Подальші етапи: організація СП (календарний план, планування обсягів виробництва та розрахунок витрат), фінансово-економічний аналіз (основні показники, обсяг інвестиційних витрат, та показники інвестиційної привабливості) з оцінкою ризиків та розробка заходів з комерціалізації (масштабування) – у даному розділі не охоплюватимуться.

4.1 Опис ідеї проекту

Опис ідеї проекту передбачає визначення її змісту, напрямків застосування та вигоди для користувачів, а також порівняння техніко-економічних характеристик з показниками конкурентів (табл. 4.1) [39]. Це потрібно для того, аби на початку реалізації СП відповісти на головне питання, що вирішує подальшу долю проекту: чи є ідея актуальною та конкурентоспроможною?

Таблиця 4.1 – Опис ідеї стартапу

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**
Система для комплектації унікальних наборів даних зі сховища тегованих високороздільних зображень за допомогою глибоких фільтрів та нейронних мереж	Формування максимально повного набору даних під конкретну задачу машинного навчання	Вища точність навченої моделі машинного навчання за рахунок мінімізації шумів та відхилень
	Підбір споріднених зображень за заданими ознаками	Швидке отримання тематичного набору зображень, які можна використати як графічний матеріал для власних проектів, презентацій тощо

Додатково до опису ідеї наведемо інформаційну картку розглянутого стартапу у таблиці 4.2:

Таблиця 4.2 – Інформаційна карта стартап-проекту

Назва проекту	PuzzleData
Автори проекту	Ланько Анна Анатоліївна
Коротка анотація	Система для комплектації унікальних наборів даних під індивідуальні потреби користувача
Термін реалізації проекту	12 місяців

Продовження таблиці 4.2

Необхідні ресурси	Технічні ресурси (сучасні комп'ютери з встановленим ПЗ для розробки, хмарне сховище даних), приміщення для роботи з електропостачанням та доступом до Інтернету, людські ресурси (2 розробники, 1 тестувальник), фінансові ресурси на термін реалізації для щомісячної виплати заробітної та орендної плати, комунальних послуг, а також підписки на хмарне сховище
Опис проблеми, яку вирішує проект	Вирішує проблему змістової незбалансованості навчальних даних, коли у наборі присутній надлишок знімків, що зображують популярні елементи, та відсутні специфічні або важкодоступні об’єктів
Головні цілі та завдання проекту	Головна ціль проекту – забезпечити користувачів можливістю отримувати повні та якісні набори даних для більш результативного вирішення задач комп’ютерного зору, завдання проекту: створити масштабне сховище тегованих високороздільних зображень, веб-ресурс для формування наборів даних з цих зображень за допомогою глибоких фільтрів та НМ
Очікувані результати	Набори даних, зібрані під конкретну задачу, підвищать точність навчених на них моделей машинного навчання

Ідея є актуальною, тому що вирішує суттєву проблему з нестачею якісних даних і має безпосередню цінність для користувачів у вигляді поліпшення якості їхніх моделей та високим рівнем персоналізації.

У таблиці 4.3 проведемо порівняння техніко-економічних характеристик ідеї проекту з показниками конкурентів.

Таблиця 4.3 – Визначення сильних, слабких та нейтральних характеристик ідеї стартап-проекту

№ п/п	Техніко-економічні х-ки ідеї	(Потенційні) товари/концепції конкурентів				W	N	S
		Власний проект	Kaggle	Hugging Face	Roboflow			
1	Масштаб-ність	10M зображень для комплек-тації персоналі-зованих наборів даних	390K різнотип-них наборів даних для задач глибокого навчання	230K різнотип-них наборів даних для задач глибокого навчання	40 наборів зображень для задач комп'ютер-ного зору		+	
2	Ціна	Платна підписка	Безкоштов-но	Безкоштов-но	Безкоштов-но	+		
3	Персоналі-зація	Глибока система фільтрів, фільтрація за ШІ-промптом	Фільтри за базовими категорія-ми	Глибока система фільтрів	Фільтри відсутні			+

Кожна з трьох перелічених характеристик ідеї проекту може бути слабкою (W), нейтральною (N) або сильною (S) його стороною. Оскільки маємо по одній сильній та нейтральній, ідею проекту можна вважати конкурентоспроможною.

4.2 Технологічний аудит ідеї продукту

Технологічний аудит має на меті з’ясувати, наскільки доступними є технології для реалізації ідеї СП (табл. 4.4) [39].

Таблиця 4.4 – Технологічна здійсненність ідеї проекту

№ п/п	**Ідея проекту**	**Технології її реалізації**	**Наявність технологій**	**Доступність технологій**
1	Система для комплектації унікальних наборів даних зі сховища тегованих високороздільних зображень за допомогою глибоких фільтрів та нейронних мереж	Python	Наявна	Доступна
2		R	Наявна	Доступна
3		Java	Наявна, необхідні допрацювання для навчання НМ	Доступна
4		JavaScript (Node.js)	Наявна, необхідні допрацювання для навчання НМ	Доступна
Обрана технологія реалізації ідеї проекту: Python				

Маємо високий рівень технологічної здійсненності проекту за рахунок того, що його можна реалізувати засобами всіх популярних мов програмування, навіть якщо для окремих мов представлено не так багато бібліотек, що дозволяють навчати НМ.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Аналіз ринкових можливостей запуску СП проводиться з метою виявлення факторів конкурентоспроможності та можливих альтернатив ринкового впровадження на основі виділених сильних та слабких сторін,

можливостей та загроз [39]. У таблиці 4.5 наведемо характеристики потенційного ринку СП.

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристики
1	Кількість головних гравців, од	Приблизно 3
2	Загальний обсяг продаж, грн/ум.од	15000 і вище
3	Динаміка ринку (якісна оцінка)	Зростає (оскільки зростає попит на використання ШІ)
4	Наявність обмежень для входу (вказати характер обмежень)	Авторське право (дозвіл на зберігання захищених ним зображень у базі системи)
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	80%

У таблиці 4.6 дамо характеристику потенційних клієнтів, а саме їхні вимоги, потреби та відмінності у поведінці.

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	**Цільова аудиторія (сегменти ринку)**	**Відмінності у поведінці потенційних цільових груп клієнтів**	**Вимоги споживачів до товару**
Відсутність у готових наборах навчальних даних зображень з галузей вузького профілю, специфічних ракурсів тощо, що погіршує якість навчання моделей	Науковці, що вирішують задачі формування зображень в умовах важкодоступності даних	Для цієї групи клієнтів не є ключовими істотні переваги продукту, пов’язані зі зручністю використання, швидкодією та високим рівнем персоналізації	Наявність у зібраному наборі даних важкодоступних специфічних ображень, яких бракує у готових відкритих наборах

Продовження таблиці 4.6

Потреба, що формує ринок	**Цільова аудиторія (сегменти ринку)**	**Відмінності у поведінці потенційних цільових груп клієнтів**	**Вимоги споживачів до товару**
Мало наборів даних є одночасно повними, якісними та легкими у доступі, що зумовлює потребу спеціального пошуку, доповнення та обробки наявних даних	Компанії різного масштабу, що потребують отримання якісних даних для реалізації своєї діяльності	Компанії можуть орієнтуватись виключно на зручність експлуатації продукту та якість результату, не використовуючи таку перевагу, як наявність у базі даних рідкісних зображень	Зручність використання, швидкість отримання даних, їхня персоналізація, повнота та якість

Далі проведемо аналіз ринкового середовища шляхом визначення факторів, що перешкоджають (табл. 4.7) та сприяють (табл. 4.8) ринковому впровадженню проекту.

Таблиця 4.7 – Фактори загроз

№ п/п	**Фактор**	**Зміст загрози**	**Можлива реакція компанії**
1	Конкуренція з великими компаніями	Якщо великі компанії реалізують схожу технологію на базі своїх популярних платформ, даний проект ризикує програти конкуренцію	Розробка вдалої маркетингової стратегії та ведення привабливої цінової політики, забезпечення відмінної якості продукту та його підтримки
2	Технологічний прогрес	Технологічний прогрес призведе до появи технологій, які зроблять проект неактуальним	Постійний моніторинг інновацій на ринку ШІ, інвестування у впровадження новітніх технологій у проект

Продовження таблиці 4.7

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
3	Пріоритети споживачів	Велика частка цільової аудиторії буде надалі користуватись готовими наборами даних, бо вони є безкоштовними	Маркетингова стратегія продукту має включати рекламу зручності застосування та досягнення принципово нового рівня якості моделей завдяки даному продукту
4	Зміни законодавства	Введення регуляторних обмежень щодо використання ШІ, що зумовить зменшення попиту на генерацію зображень	Почати співпрацю з великими компаніями, які мають високий рівень сертифікації та ліцензії на діяльність у даній галузі

Таблиця 4.8 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зайняття власної ніші на ринку	Зайняття окремої ніші на ринку роботи з даними за рахунок мінімізації обробки зображень у створених за допомогою продукту компанії наборах даних завдяки їхній структурі та уніфікованості	Побудова маркетингової кампанії навколо цієї переваги продукту, розміщення таргетингової реклами на порталах та ресурсах, пов’язаних з обробкою даних
2	Масштабування на ринку	Розширення на ринку з можливістю виходу на нові сектори та ринки за рахунок швидкого та ефективного впровадження нових технологій	Аналіз змін на поточному ринку та потенційних нових для адаптації послуг компанії та її маркетингової стратегії до останніх тенденцій, приваблива цінова політика для більшого притоку клієнтів

Продовження таблиці 4.8

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3	Технологічний прогрес	Впровадження нових технології на ринку ШІ, які покращать швидкість та якість подібних послуг	Інвестиції у дослідження інновацій з метою їхнього впровадження у продукт, підвищення кваліфікації персоналу для успішного використання новітніх технологій
4	Партнерство	Співпраця з іншими компаніями для спільного розвитку галузі та/або успішної комерціалізації власного продукту	Постійна робота над іміджем компанії, дослідження діяльності гравців ринку та їхніх продуктів з метою своєчасного виявлення потенційного партнера та укладання угоди з ним

Далі проведемо аналіз пропозиції, визначивши риси конкуренції на ринку у таблиці 4.9.

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**У чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1. Тип конкуренції: олігополія	Обмежена кількість великих компаній на ринку концентрують в своїх руках майже всю пропозицію послуги	Необхідно зробити акцент на впровадження інноваційних рішень у галузі та персоналізацію продукту
2. За рівнем конкурентної боротьби: міжнародний	Конкуренція націлена на глобальний ринок (клієнти зі всього світу)	Розробка багатомовного інтерфейсу, адаптація маркетингової каипанії до культурних особливостей різних регіонів світу

Продовження таблиці 4.9

Особливості конкурентного середовища	**У чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
3. За галузевою ознакою: внутрішньогалузева	Конкуренція між компаніями, що	Встановлення привабливої цінової політики для великого
4. Конкуренція за видами товарів: товарно-видова	Конкуренція ведеться зі схожими продуктами різних компаній	Постійна підтримка якості та робота над іміджем компанії, приваблива цінова політика
5. За характером конкурентних переваг: нецінова	Конкуренція заснована на якості та повноті функцій продукту	Інвестувати у покращення якості наявних технологій та впровадження нових, більш ефективних
6. За інтенсивністю: марочна	Конкуренція між компаніями з високим рівенм впізнаваності на ринку	Розвиток корпоративної ідентичності та іміджу компанії з метою підвищення обізнаності про бренд серед клієнтів та потенційних партнерів

У таблиці 4.10 проведемо більш глибокий аналіз конкуренції в галузі за моделлю 5 сил М. Портера.

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Прямі конкуренти у галузі**	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товарозамінники**
	Великі платформи інструментів для МН, що також пропонують набори даних	Якість надання послуг та виконання вимог споживачів, ціна	Ціна на хмарні ресурси	Чутливість до цін, вимоги щодо якості та персоналізації	Якість, ціна та масштаб бренду

Продовження таблиці 4.10

Висновки	**Прямі конкуренти у галузі**	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товарозамінники**
	Конкуренція існує виключно за рахунок масштабу бренду конкурентів та надання ними безкоштовних послуг	Потенційними конкурентами є схожі стартапи, що можуть вийти на ринок за ~12 міс.)	Темпи розвитку та масштабування ринку залежать від цін на хмарні обчислювальні ресурси та сховища	Вимоги клієнтів зумовлюють масштабування та швидкі темпи розвитку ринку	Товарозамінники відсутні

На основі аналізу конкуренції (табл. 4.9 – 4.10) з урахуванням характеристик ідеї проекту (табл. 4.3), вимог споживачів (табл. 4.6) та факторів маркетингового серидовища (табл. 4.7 – 4.8) визначимо та обґрунтовуємо перелік факторів конкурентоспроможності у таблиці 4.11.

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

№ п/п	**Фактор конкурентоспроможності**	**Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)**
1	Персоналізація	Багатоетапний та багатокритеріальний підбір даних підвищує ефективність системи
2	Ціна	Доступна ціна робить проект вигідним товарозамінником схожих продуктів конкурентів
3	Масштабність	Кількість надаваних послуг, інструментів та доступних даних збільшує кількість напрямків застовування продукту
4	Універсальність	Універсальність продукту суттєво полегшує процес його застосування порівняно з іншими
5	Сила бренду	Імідж та впізнаваність компанії є вирішальним фаткором вибору клієнта при однакових умовах

У таблиці 4.12 проведемо аналіз сильних та слабких сторін СП. Бали від 1 до 20 відображають важливість кожного з факторів конкурентоспроможності. Бали від -3 до +3 вказують на рівень відставання чи переваги проекту в порівнянні з конкурентами за кожним фактором.

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін

№ п/п	**Фактор конкуренто-спроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів у порівнянні з запропонованою системою**						
			-3	**-2**	**-1**	**0**	**+1**	**+2**	**+3**
1	Персоналізація	20		+					
2	Ціна	19						+	
3	Масштабність	17	+						
4	Універсальність	16			+				
5	Сила бренду	12						+	

Підіб'ємо підсумки ринкового аналізу можливостей впровадження СП у вигляді матриці SWOT-аналізу (табл. 4.7 – 4.8, 4.11 – 4.12).

Таблиця 4.13 – SWOT-аналіз стартап-проекту

Сильні сторони (S)	**Слабкі сторони (W)**
1. Високий рівень персоналізації 2. Масштабність інструментів та ресурсів 3. Універсальність та уніфікація даних	1. Надання виключно платних послуг 2. Відсутність сильного бренду
Можливості (O)	**Загрози (T)**
1. Зайняття окремої ніші на ринку завдяки високої якості та унікальності послуг вузької спеціалізації 2. Вихід на нові ринки завдяки впровадженню інших технологій та інстурментів 3. Поява технологій, впровадження яких виведе проект у ринкові лідери 4. Укладання угод з великими компаніями для успішної комерціалізації бренду	1. Конкуренція з великими компаніями 2. Поява технологій, які зроблять ідею проекту неактуальною 3. Пріритетність цінового аспекту над можливостями проекту серед користувачів 4. Суворе законодавче регулювання діяльності у галузі ШІ

Для успішного впровадження СП маємо максимізувати переваги, використовувати можливості, підсилювати слабкі сторони та оминати загрози. Оскільки на практиці всі ці умови неможливо виконати в повному обсязі, маємо визначитись з альтернативами ринкового впровадження СП (табл. 4.14) та обрати найкращу.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	**Альтернатива (орієнтовний комплекс заходів) ринкової поведінки**	**Ймовірність отримання ресурсів**	**Строки реалізації**
1	Вийти на ринок з з недоукомплектованою базою зображень	Висока	9 місяців
2	Надання безкоштовної версії з обмеженим функціоналом з метою регулярного поповнення бази зображень активними користувачами	Висока	9 місяців
3	Вихід на ринок у колаборації з великим брендом на ринку	Низька	12 місяців
4	Надання послуг через API для інших популярних систем	Середня	6 місяців

Альтернатива 1, попри найбільшу доступність, може суттєво зашкодити репутації та масштабуванню СП на ринку, оскільки критично залежить від грамотності маркетингової кампанії та загальної толерантності користувачів до продуктів, що знаходяться у процесі реалізації. Альтернативи 3 – 4 можуть становити загрозу для розвитку бренду, 4 – також комерціалізації продукту. Саме тому вигідніше вийти на ринок з безкоштовною версією обмеженої функціональності, яка буде заповнювати прогалини та підвищуватиме його популярність.

4.4 Розроблення ринкової стратегії проекту

Ринкова стратегія будується для того, аби успішно вийти на ринок та масштабуватись попри наявні недоліки та перешкоди за рахунок правильного визначення цільової аудиторії, ефективної конкурентної поведінки та позиціонування [39]. У таблиці 4.15 визначимо цільові групи потенційних клієнтів.

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

№ п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит у межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Персональні користувачі	Висока	90%	Висока	Висока
2	Великі компанії, що займаються ШІ	Висока	60%	Середня	Середня
3	Стартапи у галузі ШІ	Висока	75%	Висока	Середня
4	Наукові установи	Середня	20%	Низька	Середня
Які цільові групи обрано: 1, 2, 3					

Далі необхідно визначити базову стратегію розвитку в цільових сегментах ринку для обраної альтернативи (табл. 4.16). Стратегія охоплення ринку при цьому залежить від кількості цільових груп.

Таблиця 4.16 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку**
2 (проект з безкоштовною версією)	Стратегія масового маркетингу	Масштабування, обсяг ресурсів та найвищий рівень персоналізації	**Стратегія диференціації** (важлива відмітна властивість – персоналізація)

У таблиці 4.17 оберемо базову стратегію конкурентної поведінки на основі того, як компанія планує використати досвід та досягнення конкурентів.

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першо-прохідником» на ринку?	**Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?**	**Чи буде компанія копіювати основні характеристики товару-конкурента, і які?**	**Стратегія конкурентної поведінки**
Ні (має відмітну властивість порівняно з продуктами конкурентів)	Так	Так (запропонувати власну платформу для хмарних обчислень, каталог ШІ моделей)	**Стратегія виклику лідера** (оскільки компанія прагне охопити всі сегменти і впроваджувати нові технології)

Враховуючи результати таблиць 4.16 та 4.17, проведемо визначення стратегії позиціювання (таблиця 4.18), тобто визначимо ринкові позиції, що будуть слугувати ідентифікатором проекту для клієнтів.

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	**Базова стратегія розвитку**	**Ключові конкуренто-спроможні позиції власногопроекту**	**Вибір асоціацій, які мають сформувати комплексну позицію власного проекту**
Великий вибір та високий рівень якості зображень, персоналізації процесу комплектації наборів даних	Стратегія диференціації	Великий вибір зображень, у тому числі важкодоступних чи специфічних, персоналізована комплектація наборів даних, універсальність даних в наборі	1. Система, що має найширший вибір зображень 2. Система, орієнтована на персоналізоване вирішення задачі 3. Універсальна истема зі структурованими, уніфікованими даними

Отже, компанія обирає стратегію масового маркетингу для охоплення ринку, стратегію диференціації для базового розвитку та виклик лідера як стратегію конкурентної поведінки. Ідентифікаторами проекту компанії для клієнтів є широкий вибір зображень, високий рівень персоналізації та універсальність.

4.5 Розроблення маркетингової програми стартап-проекту

Маркетингова програма розробляється з метою успішної комерціалізації створеного продукту. Вона включає встановлення привабливих цін, впровадження вигідної системи збуту та ефективної

концепції маркетингових комунікацій, враховуючи ключові переваги та рівні моделі товару [39]. У таблиці 4.19 визначимо ключові переваги потенційного товару.

Таблиця 4.19 – Визначення ключових переваг концепції потенційного товару

№ п/п	**Потреба**	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами**
1	Відповідність набору даних вимогам конкретної задачі	Використовується не готовий набір даних на певну тематику, а спеціально укомплектований з бази зображень набір під потреби користувача	Глибокі фільтри для максимального точно підбору зображень під вимоги користувача, можливість комплектації за ШІ-промптом
2	Широкий вибір зображень	База зображень системи поповнюється шляхом внеску користувачів, тому в ній наявні максимально різноманітні зображення	Набори даних конкурентів часто є неповними, зазвичай не містять рідкісних зображень специфічної тематики
3	Універсальність та однорідність даних	Зображення підбираються з єдиної бази та проходять обробку з метою структуризації та уніфікації	Зображення в наборах даних конкурентів потребують додаткової обробки та структуризації

Далі опишемо три рівні моделі товару – товар за задумом, товар у реальному виконанні та товар із підкріпленням (таблиця 4.20) для забезпечення максимальної цінності та конкурентної переваги продукту.

Таблиця 4.20 – Опис трьох рівнів моделі товару

Рівні товару	**Сутність та складові**		
І. Товар за задумом	Система комплектації наборів даних з високим рівнем персоналізації, а також великим вибором зображень, їхньою універсальністю та уніфікованістю		
ІІ Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Персоналізація 2. Універсальність 3. Масштабність 4. Зручний веб-ресурс	-	-
	Якість: відповідає вимогам щодо швидкості та ефективності, типу та формату зображень у базі та стандартам розробки веб-ресурсу		
	Пакування: веб-ресурс		
	Марка: компанія «Data Innovations», товар PuzzleData		
ІІІ Товар із підкріпленням	До продажу: технічна підтримка на пробний період, обмежена кількість запитів до системи		
	Після продажу: постійна технічна підтримка, доступ до останніх оновлень та програма виплат користувачам, що роблять регулярний внесок у базу зображень		
За рахунок чого потенційний товар буде захищено від копіювання: інтелектуальні патенти та авторські права			

Знаючи потреби та очікування споживачів, можемо визначити приблизні межі встановлення цін у таблиці 4.21.

Таблиця 4.21 – Визначення меж встановлення ціни

Рівень цін на товари-замінники, грн.	**Рівень цін на товари-аналоги, грн.**	**Рівень доходів цільової групи споживачів, грн.**	**Верхня та нижня межі встановлення ціни на товар/послугу, грн.**
-	-	Від 30000 грн./міс.	300-800 грн./міс.

Попередній аналіз також дозволяє сформувати систему збуту СП у таблиці 4.22.

Таблиця 4.22 – Формування системи збуту

№ п/п	**Специфіка закупівельної поведінки цільових клієнтів**	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна системазбуту**
1	Потреба в індивідувальному підході	Гнучкість та адаптивність до індивідуальних потреб клієнтів	0	Продажі через офіційний веб-ресурс компанії
2	Попит на доступність та організованість процесу купівлі	Реалізація зручного та інформативного ознайомчого ресурсу, платіжної системи	0	Продажі через офіційний веб-ресурс компанії
3	Попит на використання новітніх технологій	Використовувати високотехнологічні засоби маркетингу	0	Продажі через офіційний веб-ресурс компанії
4	Гарантування якості та підтримка	Наявність зворотного зв’язку та системи підтримки	0	Продажі через офіційний веб-ресурс компанії

Зазначимо, що для подібних продуктів оптимальною системою збуту завжди є офіційний веб-ресурс, оскільки таким чином забезпечується централізованість продажів та дотримання авторського права, а також збільшується сила бренду. Утім, для цього необхідно якомога точніше визначитись з рекламною стратегією та каналами її реалізації для ефективного досягнення всіх груп цільових клієнтів.

Для цього визначається стратегія маркетингових комунікацій, представлена у таблиці 4.23.

Таблиця 4.23 – Концепція маркетингових комунікацій

№ п/п	**Специфіка поведінки цільових клієнтів**	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
1	Пошук якісного та ефективного рішення	Веб-сайти, соціальні мережі, спеціалізовані веб-ресурси	Пріоритет на якості та ефективності	Подолати відсутність інформації та можливі сумніви стосовно якості та ефективності продукту	Чітко донести та продемонструвати ключові переваги та вигідні особливості продукту за допомогою клієнтських відгуків, експертних статей, фото- та відеоматеріалів
2	Пошук інноваційних технологій та рішень		Використання інноваційних технологій при розробці продукту	Акцентувати увагу на останніх технологічних трендах у функціоналі продукту	
3	Пошук найдешевшого продукту серед аналогів		Приваблива цінова політика програми лояльності	Підтвердити вигідність продукту серед пропозицій на ринку	

Продовження таблиці 4.23

№ п/п	**Специфіка поведінки цільових клієнтів**	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціону-вання**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
4	Бажання отримати персоналі-зова-ний досвід	Веб-сайти, соціальні мережі, спеціалізовані веб-ресурси	Адаптація під вимоги клієнта, система підтримки	Продемонстру-вати кейси роботи з потребами конкретних користувачів, якість підтримки	Чітко донести та продемонстру-вати ключові переваги та вигідні особ-ливості продук-ту за допомогою клієнтських відгуків, екс-пертних статей, фото- та відео-матеріалів

4.6 Висновки до розділу 4

У даному розділі в якості СП було запропоновано систему для комплектації наборів даних за параметрами користувача. Ця ідея є актуальною, оскільки вирішує проблему відсутності навчальних даних специфічної тематики, здійсненною та має важливу відмітну властивість, оскільки жоден з аналогічних ресурсів не формує набір з постійно оновлюваної бази даних зображень.

Ринкова комерціалізація цього проекту є можливою завдяки стрімкому масштабуванню ринку ШІ та наявності великого попиту на швидке

та ефективне розв'язання задач комп'ютерного зору у різних галузях – а також тому, що на ринку немає товарів-замінників для даного СП.

Конкурентоспроможність проекту є високою через високу якість, а також важливі відмітні властивості, такі як персоналізована комплектація наборів, універсальність та структурованість даних у них, а також постійно оновлювана велика база зображень. З огляду на низький бар'єр входження, малу кількість конкурентів та орієнтованість на всі потенційні групи клієнтів проект має широкі перспективи впровадження.

Для ринкової реалізації проекту обрано альтернативу. яка пропонує вийти на ринок з безкоштовною версією обмеженої функціональності для підвищення інтересу та впізнаваності серед потенційних клієнтів, а також регулярного поповнення бази зображень. Подальша імплементація цього продукту є доцільною за умови постійної апаптації його технологій до новітніх тенденцій та розробки нових функцій для масштабування на ринку.

ВИСНОВКИ

У даній роботі було проведено огляд відомих методів збільшення роздільності та архітектури популярних моделей, а також методів та вимог до оцінювання результатів у задачах формування зображень, проведено експерименти з власної реалізації та донавчання моделей SRGAN, VDSR та DRCN, проведено аналіз результатів за значеннями кількісних показників PSNR та SSIM, перцепційного показника LPIPS, часу навчання та візуальної оцінки породжених суперроздільних зображень. Результатом виконання роботи є програмний продукт для збільшення роздільності зображень на базі оптимальної (за вищевказаною сукупністю критеріїв) моделі.

Визначною особливістю даної роботи є оптимізація використання обчислювальних ресурсів, оскільки підхід до вибору оптимальної моделі відрізняється в залежності від потреб та потужності, доступності наявного апаратного прискорювача. Запропоновано алгоритми навчання для власної реалізації моделей з менш складною архітектурою та попередньо навчених, а також алгоритм об’єктивної оцінки їхньої якості за сукупністю критеріїв. Для прибирання пікселів можна використовувати прості моделі з ранньою зупинкою, для кращої деталізації потрібно збільшувати кількість епох навчання або застосовувати технологію донавчання.

Найкращою серед розглянутих для обох експериментів виявилась модель VDSR. Цей результат не є збігом, оскільки VDSR має швидку збіжність, яка виявилась ключовою перевагою в умовах даних обчислювальних потужностей та невеликої кількості епох навчання. Загалом у роботі було показано перевагу використання методів глибокого навчання над породжувальними моделями та перевагу використання залишкових з'єднань при вивченні деталей зображення глибокими мережами.

До напрямків подальшої роботи над дослідженням можна віднести реалізацію нових моделей та оптимізацію обчислень, перспективою

практичного впровадження розробленого програмного продукту є його інтеграція у веб-додаток для підвищення роздільної здатності зображень або представлення у вигляді API.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Brownlee J. 18 Impressive Applications of Generative Adversarial Networks (GANs). Machine Learning Mastery. URL: https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/.

2. Albelwi S. Survey on Self-Supervised Learning: Auxiliary Pretext Tasks and Contrastive Learning Methods in Imaging. Entropy. 2022. Vol. 24, no. 4, 551. URL: https://doi.org/10.3390/e24040551.

3. Ausare T. Ultimate Guide to Selecting a GPU for Deep Learning. Latest AI, ML & GPU Updates. NeevCloud. URL: https://blog.neevcloud.com/ultimate-guide-to-selecting-a-gpu-for-deep-learning.

4. M. S. Alam et al: Super-Resolution Enhancement Method Based on Generative Adversarial Network for Integral Imaging Microscopy. Sensors. 2021. Vol. 21, no. 6, 2164. URL: https://doi.org/10.3390/s21062164.

5. J. Li et al: A Systematic Survey of Deep Learning-based Single-Image Super-Resolution. ACM Computing Surveys. 2024. URL: https://doi.org/10.1145/3659100.

6. Image Super Resolution: A Comparison between Interpolation & Deep Learning-based Techniques to Improve Clarity of Low-Resolution Images. Medium. URL: https://medium.com/htx-s-s-coe/image-super-resolution-a-comparison-between-interpolation-deep-learning-based-techniques-to-25e7531ab207.

7. Cheikh Sidiya A., Li X. Style-Based Unsupervised Learning for Real-World Face Image Super-Resolution. Recent Advances in Image Restoration with Applications to Real World Problems. 2020. URL: https://doi.org/10.5772/intechopen.92320.

8. S. Bond-Taylor et al: Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive

Models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021. Issue 11,01. P. 7327 – 7347. URL: https://doi.org/10.1109/tpami.2021.3116668.

9. C. Ledig et al: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 21–26 July 2017. URL: https://doi.org/10.1109/cvpr.2017.19.

10. X. Wang et al: ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. Lecture Notes in Computer Science. Cham, 2019. P. 63–79. URL: https://doi.org/10.1007/978-3-030-11021-5_5.

11. C. Fang et al: TSRGAN: Real-world text image super-resolution based on adversarial learning and triplet attention. Neurocomputing. 2021. Vol. 455. P. 88–96. URL: https://doi.org/10.1016/j.neucom.2021.05.060.

12. J. Rafid Siddiqui. Diffusion Models Made Easy. Medium. URL: https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da.

13. Soufi O., Aarab Z., Belouadha F.-Z. Benchmark of deep learning models for single image super-resolution (SISR). 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 3–4 March 2022. URL: https://doi.org/10.1109/iraset52964.2022.9738274.

14. A. Lugmayr et al: SRFlow: Learning the Super-Resolution Space with Normalizing Flow. Computer Vision – ECCV 2020. Cham, 2020. P. 715–732. URL: https://doi.org/10.1007/978-3-030-58558-7_42.

15. C. Dong et al: Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016. Vol. 38, Issue 2. P. 295–307. DOI: 10.1109/TPAMI.2015.2439281.

16. Dong C., Loy C. C., Tang X. Accelerating the Super-Resolution Convolutional Neural Network. Computer Vision – ECCV 2016. Cham, 2016. P. 391–407. URL: https://doi.org/10.1007/978-3-319-46475-6_25.

17. Kim J., Lee J. K., Lee K. M. Deeply-Recursive Convolutional Network for Image Super-Resolution. 2016 IEEE Conference on Computer Vision and

Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. URL: https://doi.org/10.1109/cvpr.2016.181.

18. Kim J., Lee J. K., Lee K. M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. 2016. URL: https://doi.org/10.1109/cvpr.2016.182.

19. Lim B. et al: Enhanced Deep Residual Networks for Single Image Super-Resolution. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): NTIRE 2017 Workshop. URL: https://doi.org/10.48550/arXiv.1707.02921.

20. T. Tong et al: Image Super-Resolution Using Dense Skip Connections. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 22–29 October 2017. URL: https://doi.org/10.1109/iccv.2017.514.

21. Po-ChenLin. Image Super-Resolution via Deep Level Residual Network : thesis. 2018. URL: http://ndltd.ncl.edu.tw/handle/3uvfyp.

22. Y. Zhang et al: Image Super-Resolution Using Very Deep Residual Channel Attention Networks. Computer Vision – ECCV 2018. Cham, 2018. P. 294–310. URL: https://doi.org/10.1007/978-3-030-01234-2_18.

23. Musunuri Y. R., Kwon O.-S. Deep Residual Dense Network for Single Image Super-Resolution. Electronics. 2021. Vol. 10(5), 555. URL: https://doi.org/10.3390/electronics10050555.

24. SwinIR: Image Restoration Using Swin Transformer / J. Liang et al: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021. URL: https://doi.org/10.1109/iccvw54120.2021.00210.

25. Farahmand M. CNN-Based Single-Image Super-Resolution: A Comparative Study. Medium. URL: https://farahmand-m.medium.com/cnn-based-single-image-super-resolution-6ffcd39ec993.

26. Q. Jiang et al: Single Image Super-Resolution Quality Assessment: A Real-World Dataset, Subjective Studies, and an Objective Metric. IEEE

Transactions on Image Processing. 2022. Vol. 31. P. 2279–2294. URL: https://doi.org/10.1109/tip.2022.3154588.

27. Fardo, F. A., Conforto, V. H., de Oliveira, F. C., & Rodrigues, P. S. (2016). A Formal Evaluation of PSNR as Quality Measurement Parameter for Image Segmentation Algorithms. URL: https://doi.org/10.48550/arXiv.1605.07116.

28. Z. Wang et al: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing. 2004. Vol. 13, Issue 4. P. 600–612. URL: https://doi.org/10.1109/tip.2003.819861.

29. R. Zhang et al: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, 18–23 June 2018. URL: https://doi.org/10.1109/cvpr.2018.00068.

30. Mittal, A., Soundararajan, R., Bovik, A.C. Making a “completely blind” image quality analyzer. IEEE Signal Process. Lett., 2013, 20(3), P. 209–212.

31. Blind image quality evaluation using perception based features / N. Venkatanath et al: 2015 Twenty First National Conference on Communications (NCC), Mumbai, India, 27 February – 1 March 2015. P. 1–6. URL: https://doi.org/10.1109/ncc.2015.7084843.

32. Mittal, A., Moorthy, A., Bovik, A.: Referenceless image spatial quality evaluation engine. In: 45th Asilomar Conference on Signals, Systems and Computers, 2011, vol. 38, pp. 53–54.

33. DIV2k Dataset. Kaggle: Your Machine Learning and Data Science Community. URL: https://www.kaggle.com/datasets/kai0430/div2k-dataset.

34. Agustsson E., Timofte R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017. URL: https://doi.org/10.1109/cvprw.2017.150.

35. OpenMMLab. Awesome Datasets for Super-Resolution: Introduction and Pre-processing. Medium. URL: https://openmmlab.medium.com/awesome-datasets-for-super-resolution-introduction-and-pre-processing-55f8501f8b18.

36. GitHub - tensorlayer/SRGAN: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. GitHub. URL: https://github.com/tensorlayer/SRGAN.

37. GitHub - twtygqyy/pytorch-vdsr: VDSR (CVPR2016) pytorch implementation. GitHub. URL: https://github.com/twtygqyy/pytorch-vdsr.

38. GitHub - Lornatang/SRCNN-PyTorch: Pytorch framework can easily implement srcnn algorithm with excellent performance. GitHub. URL: https://github.com/Lornatang/SRCNN-PyTorch.

39. Розроблення стартап-проекту: методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей. Уклад. О. А. Гавриш, С. О. Солнцев, В. В. Дергачова, О. В. Зозульов [та ін.]. Київ: НТУУ «КПІ», 2016. – 28 с. URL: https://ela.kpi.ua/items/53c140af-585e-4e63-b6e2-e43460684209.

40. Ланько А. А., Недашківська Н. І. Породжувальні моделі та методи глибокого навчання для задачі SISR. Системні науки та інформатика: збірник доповідей III науково-практичної конференції «Системні науки та інформатика», 25–29 листопада 2024 року, Київ. Київ: НН ІПСА КПІ ім. Ігоря Сікорського, 2024. 6 стор.

41. Nedashkovskaya N., Lanko A. Quality assessment of models and deep learning methods for super-resolution image formation // System research and information technologies. – 2024. – №4. – 19 p.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import os
import time
import random
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader
from torchvision import transforms
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
from skimage.metrics import structural_similarity as ssim
import lpips

# === Metrics and Utilities ===
def calculate_psnr(img1, img2):
    mse = torch.mean((img1 - img2) ** 2)
    if mse == 0:
        return float('inf')
    return 20 * torch.log10(1.0 / torch.sqrt(mse))

def calculate_ssim(img1, img2):
    img1_np = img1.squeeze().permute(1, 2, 0).cpu().numpy()
    img2_np = img2.squeeze().permute(1, 2, 0).cpu().numpy()
    return ssim(img1_np, img2_np, multichannel=True, data_range=img1_np.max()
- img1_np.min())

loss_fn_vgg = lpips.LPIPS(net='vgg').to('cuda' if torch.cuda.is_available()
else 'cpu')
def calculate_lpips(img1, img2):
    img1 = 2 * img1 - 1  # Normalize to [-1, 1]
    img2 = 2 * img2 - 1
    return loss_fn_vgg(img1, img2).mean().item()

# === Dataset ===
class DIV2KDataset(torch.utils.data.Dataset):
    def __init__(self, root_dir, scale=4, mode='train'):
        self.root_dir = root_dir
        self.scale = scale
        self.mode = mode
        if self.mode == 'train':
            self.hr_dir = os.path.join(root_dir, 'DIV2K_train_HR')
            self.lr_dir = os.path.join(root_dir,
f'DIV2K_train_LR_bicubic/X{scale}')
        elif self.mode == 'valid':
            self.hr_dir = os.path.join(root_dir, 'DIV2K_valid_HR')
            self.lr_dir = os.path.join(root_dir,
f'DIV2K_valid_LR_bicubic/X{scale}')
        else:
            self.hr_dir = os.path.join(root_dir, 'DIV2K_test_HR')
```

```
            self.lr_dir = os.path.join(root_dir,
f'DIV2K_test_LR_bicubic/X{scale}')

        self.image_filenames = [f for f in os.listdir(self.hr_dir) if
f.endswith('.png')]
        self.transform_hr = transforms.ToTensor()
        self.transform_lr = transforms.ToTensor()

    def __len__(self):
        return len(self.image_filenames)

    def __getitem__(self, idx):
        hr_image_path = os.path.join(self.hr_dir, self.image_filenames[idx])
        lr_image_path = os.path.join(self.lr_dir,
self.image_filenames[idx].replace('.png', f'x{self.scale}.png'))

        hr_image = Image.open(hr_image_path).convert('RGB')
        lr_image = Image.open(lr_image_path).convert('RGB')

        hr_image = self.transform_hr(hr_image)
        lr_image = self.transform_lr(lr_image)
        return lr_image, hr_image

# === Models ===
class ResidualBlock(nn.Module):
    def __init__(self, channels):
        super(ResidualBlock, self).__init__()
        self.block = nn.Sequential(
            nn.Conv2d(channels, channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(channels),
            nn.PReLU(),
            nn.Conv2d(channels, channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(channels)
        )

    def forward(self, x):
        return x + self.block(x)

class Generator_SRResNet(nn.Module):
    def __init__(self, num_residual_blocks=16):
        super(Generator_SRResNet, self).__init__()
        self.initial = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=9, padding=4),
            nn.PReLU()
        )
        self.residual_blocks = nn.Sequential(
            *[ResidualBlock(64) for _ in range(num_residual_blocks)]
        )
        self.conv_block = nn.Sequential(
            nn.Conv2d(64, 64, kernel_size=3, padding=1),
            nn.BatchNorm2d(64)
        )
        self.upsampling = nn.Sequential(
            nn.Conv2d(64, 256, kernel_size=3, padding=1),
            nn.PixelShuffle(2),
```

```
            nn.PReLU(),
            nn.Conv2d(64, 256, kernel_size=3, padding=1),
            nn.PixelShuffle(2),
            nn.PReLU()
        )
        self.final = nn.Conv2d(64, 3, kernel_size=9, padding=4)

    def forward(self, x):
        initial = self.initial(x)
        residual = self.residual_blocks(initial)
        conv_block = self.conv_block(residual)
        out = initial + conv_block
        out = self.upsampling(out)
        out = self.final(out)
        return out

class VDSR(nn.Module):
    def __init__(self, num_channels=3):
        super(VDSR, self).__init__()
        layers = [nn.Conv2d(num_channels, 64, kernel_size=3, padding=1),
nn.ReLU(inplace=True)]
        for _ in range(18):
            layers.append(nn.Conv2d(64, 64, kernel_size=3, padding=1))
            layers.append(nn.ReLU(inplace=True))
        layers.append(nn.Conv2d(64, num_channels, kernel_size=3, padding=1))
        self.layers = nn.Sequential(*layers)

    def forward(self, x):
        residual = x
        out = self.layers(x)
        return out + residual

class DRCN(nn.Module):
    def __init__(self, num_channels=3, num_features=64, num_recursions=16):
        super(DRCN, self).__init__()
        self.num_recursions = num_recursions
        self.fe = nn.Conv2d(num_channels, num_features, kernel_size=3,
padding=1)
        self.recursive = nn.Conv2d(num_features, num_features, kernel_size=3,
padding=1)
        self.reconstruction = nn.Conv2d(num_features, num_channels,
kernel_size=3, padding=1)
        self.relu = nn.ReLU(inplace=True)

    def forward(self, x):
        residual = x
        out = self.relu(self.fe(x))
        outputs = []
        for _ in range(self.num_recursions):
            out = self.relu(self.recursive(out))
            outputs.append(self.reconstruction(out))
        out = sum(outputs) / self.num_recursions
        return out + residual

class Discriminator(nn.Module):
```

```
    def __init__(self):
        super(Discriminator, self).__init__()
        self.model = nn.Sequential(
            nn.Conv2d(3, 64, 3, 1, 1),
            nn.LeakyReLU(0.2, inplace=True),

            nn.Conv2d(64, 64, 3, 2, 1),
            nn.BatchNorm2d(64),
            nn.LeakyReLU(0.2, inplace=True),

            nn.Conv2d(64,128,3,1,1),
            nn.BatchNorm2d(128),
            nn.LeakyReLU(0.2,inplace=True),

            nn.Conv2d(128,128,3,2,1),
            nn.BatchNorm2d(128),
            nn.LeakyReLU(0.2,inplace=True),

            nn.Conv2d(128,256,3,1,1),
            nn.BatchNorm2d(256),
            nn.LeakyReLU(0.2,inplace=True),

            nn.Conv2d(256,256,3,2,1),
            nn.BatchNorm2d(256),
            nn.LeakyReLU(0.2,inplace=True),

            nn.AdaptiveAvgPool2d((6,6)),
        )
        self.fc = nn.Sequential(
            nn.Linear(256*6*6,1024),
            nn.LeakyReLU(0.2,inplace=True),
            nn.Linear(1024,1),
            nn.Sigmoid()
        )
    def forward(self,x):
        out = self.model(x)
        out = out.view(out.size(0),-1)
        out = self.fc(out)
        return out

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

root_dir = 'C:/datasets/DIV2K'
scale_factor = 4

train_dataset = DIV2KDataset(root_dir, scale=scale_factor, mode='train')
valid_dataset = DIV2KDataset(root_dir, scale=scale_factor, mode='valid')
test_dataset = DIV2KDataset(root_dir, scale=scale_factor, mode='test')

batch_size = 16
epochs = 20

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
valid_loader = DataLoader(valid_dataset, batch_size=1, shuffle=False)
test_loader = DataLoader(test_dataset, batch_size=1, shuffle=False)
```

```
generator = Generator_SRResNet().to(device)
discriminator = Discriminator().to(device)
vdsr = VDSR().to(device)
drcn = DRCN().to(device)

optimizer_G = optim.Adam(generator.parameters(), lr=1e-4)
optimizer_D = optim.Adam(discriminator.parameters(), lr=1e-4)
optimizer_VDSR = optim.Adam(vdsr.parameters(), lr=1e-4)
optimizer_DRCN = optim.Adam(drcn.parameters(), lr=1e-4)

criterion = nn.MSELoss()
adversarial_criterion = nn.BCELoss()

srgan_g_losses = []
srgan_d_losses = []
vdsr_losses = []
drcn_losses = []

vdsr_psnr_epochs = []
drcn_psnr_epochs = []

print("Starting Training...")
for epoch in range(epochs):
    epoch_g_loss = 0
    epoch_d_loss = 0
    epoch_vdsr_loss = 0
    epoch_drcn_loss = 0

    generator.train()
    discriminator.train()
    vdsr.train()
    drcn.train()

    for i, (lr_imgs, hr_imgs) in enumerate(train_loader):
        lr_imgs, hr_imgs = lr_imgs.to(device), hr_imgs.to(device)

        # Train Discriminator (SRGAN)
        optimizer_D.zero_grad()
        sr_imgs_srgan = generator(lr_imgs)
        real_validity = discriminator(hr_imgs)
        fake_validity = discriminator(sr_imgs_srgan.detach())
        real_label = torch.ones_like(real_validity, device=device)
        fake_label = torch.zeros_like(fake_validity, device=device)
        d_loss_real = adversarial_criterion(real_validity, real_label)
        d_loss_fake = adversarial_criterion(fake_validity, fake_label)
        d_loss = (d_loss_real + d_loss_fake) / 2
        d_loss.backward()
        optimizer_D.step()

        # Train Generator (SRGAN)
        optimizer_G.zero_grad()
        fake_validity = discriminator(sr_imgs_srgan)
        g_adv_loss = adversarial_criterion(fake_validity, real_label)
        g_content_loss = criterion(sr_imgs_srgan, hr_imgs)
```

```
        g_loss = g_content_loss + 1e-3 * g_adv_loss
        g_loss.backward()
        optimizer_G.step()

        epoch_g_loss += g_loss.item()
        epoch_d_loss += d_loss.item()

        # Train VDSR
        optimizer_VDSR.zero_grad()
        sr_imgs_vdsr = vdsr(lr_imgs)
        v_loss = criterion(sr_imgs_vdsr, hr_imgs)
        v_loss.backward()
        optimizer_VDSR.step()
        epoch_vdsr_loss += v_loss.item()

        # Train DRCN
        optimizer_DRCN.zero_grad()
        sr_imgs_drcn = drcn(lr_imgs)
        drcn_loss_val = criterion(sr_imgs_drcn, hr_imgs)
        drcn_loss_val.backward()
        optimizer_DRCN.step()
        epoch_drcn_loss += drcn_loss_val.item()

    # Average losses
    avg_g_loss = epoch_g_loss / len(train_loader)
    avg_d_loss = epoch_d_loss / len(train_loader)
    avg_vdsr_loss = epoch_vdsr_loss / len(train_loader)
    avg_drcn_loss = epoch_drcn_loss / len(train_loader)

    srgan_g_losses.append(avg_g_loss)
    srgan_d_losses.append(avg_d_loss)
    vdsr_losses.append(avg_vdsr_loss)
    drcn_losses.append(avg_drcn_loss)

    # Validation PSNR for VDSR and DRCN
    generator.eval()
    vdsr.eval()
    drcn.eval()

    vdsr_psnrs = []
    drcn_psnrs = []
    with torch.no_grad():
        for lr_imgs, hr_imgs in valid_loader:
            lr_imgs, hr_imgs = lr_imgs.to(device), hr_imgs.to(device)
            # VDSR PSNR
            sr_vdsr_val = torch.clamp(vdsr(lr_imgs), 0, 1)
            psnr_v = calculate_psnr(sr_vdsr_val, hr_imgs)
            vdsr_psnrs.append(psnr_v.item())

            # DRCN PSNR
            sr_drcn_val = torch.clamp(drcn(lr_imgs), 0, 1)
            psnr_d = calculate_psnr(sr_drcn_val, hr_imgs)
            drcn_psnrs.append(psnr_d.item())

    avg_vdsr_psnr = sum(vdsr_psnrs) / len(vdsr_psnrs)
```

```
    avg_drcn_psnr = sum(drcn_psnrs) / len(drcn_psnrs)

    vdsr_psnr_epochs.append(avg_vdsr_psnr)
    drcn_psnr_epochs.append(avg_drcn_psnr)

    print(f"Epoch [{epoch+1}/{epochs}]")
    print(f"SRGAN G Loss: {avg_g_loss:.4f}, D Loss: {avg_d_loss:.4f}")
    print(f"VDSR Loss: {avg_vdsr_loss:.4f}, DRCN Loss: {avg_drcn_loss:.4f}")
    print(f"VDSR PSNR: {avg_vdsr_psnr:.4f} dB, DRCN PSNR: {avg_drcn_psnr:.4f}
dB")

# Plot SRGAN losses
plt.figure()
plt.plot(srgan_d_losses, label='Discriminator')
plt.plot(srgan_g_losses, label='Generator')
plt.title('SRGAN')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.savefig('srgan_loss_plot.png')
plt.close()

# Plot PSNR for VDSR and DRCN
plt.figure()
plt.plot(vdsr_psnr_epochs, label='VDSR')
plt.plot(drcn_psnr_epochs, label='DRCN')
plt.title('PSNR Values')
plt.xlabel('Epoch')
plt.ylabel('PSNR (dB)')
plt.legend()
plt.savefig('psnr_values_plot.png')
plt.close()

# Plot MSE Loss for VDSR and DRCN
plt.figure()
plt.plot(vdsr_losses, label='VDSR')
plt.plot(drcn_losses, label='DRCN')
plt.title('MSE Loss over Epochs')
plt.xlabel('Epoch')
plt.ylabel('MSE Loss')
plt.legend()
plt.savefig('mse_loss_plot.png')
plt.close()

# Save models
model_dir = 'C:/models/SISR'
os.makedirs(model_dir, exist_ok=True)
torch.save(generator.state_dict(), os.path.join(model_dir,
'srgan_generator.pth'))
torch.save(discriminator.state_dict(), os.path.join(model_dir,
'srgan_discriminator.pth'))
torch.save(vdsr.state_dict(), os.path.join(model_dir, 'vdsr.pth'))
torch.save(drcn.state_dict(), os.path.join(model_dir, 'drcn.pth'))

# Evaluate on 2 subsets of 10 random test images
```

```
test_indices = list(range(len(test_dataset)))
random_subset_1 = random.sample(test_indices, 10)
random_subset_2 = random.sample(test_indices, 10)

def evaluate_subset(indices, generator, vdsr, drcn):
    gen_metrics = {'psnr':[], 'ssim':[], 'lpips':[]}
    vdsr_metrics = {'psnr':[], 'ssim':[], 'lpips':[]}
    drcn_metrics = {'psnr':[], 'ssim':[], 'lpips':[]}

    generator.eval()
    vdsr.eval()
    drcn.eval()

    with torch.no_grad():
        for idx in indices:
            lr_img, hr_img = test_dataset[idx]
            lr_img = lr_img.unsqueeze(0).to(device)
            hr_img = hr_img.unsqueeze(0).to(device)

            sr_gen = torch.clamp(generator(lr_img), 0, 1)
            sr_vdsr = torch.clamp(vdsr(lr_img), 0, 1)
            sr_drcn = torch.clamp(drcn(lr_img), 0, 1)

            # SRGAN metrics
            gen_metrics['psnr'].append(calculate_psnr(sr_gen, hr_img).item())
            gen_metrics['ssim'].append(calculate_ssim(sr_gen, hr_img))
            gen_metrics['lpips'].append(calculate_lpips(sr_gen, hr_img))

            # VDSR metrics
            vdsr_metrics['psnr'].append(calculate_psnr(sr_vdsr,
hr_img).item())
            vdsr_metrics['ssim'].append(calculate_ssim(sr_vdsr, hr_img))
            vdsr_metrics['lpips'].append(calculate_lpips(sr_vdsr, hr_img))

            # DRCN metrics
            drcn_metrics['psnr'].append(calculate_psnr(sr_drcn,
hr_img).item())
            drcn_metrics['ssim'].append(calculate_ssim(sr_drcn, hr_img))
            drcn_metrics['lpips'].append(calculate_lpips(sr_drcn, hr_img))

    # Average metrics
    def avg_metrics(m):
        return {k: sum(v)/len(v) for k, v in m.items()}

    return avg_metrics(gen_metrics), avg_metrics(vdsr_metrics),
avg_metrics(drcn_metrics)

subset1_gen, subset1_vdsr, subset1_drcn = evaluate_subset(random_subset_1,
generator, vdsr, drcn)
subset2_gen, subset2_vdsr, subset2_drcn = evaluate_subset(random_subset_2,
generator, vdsr, drcn)

print("\nSubset 1 Metrics:")
print("SRGAN:", subset1_gen)
print("VDSR :", subset1_vdsr)
```

```
print("DRCN :", subset1_drcn)

print("\nSubset 2 Metrics:")
print("SRGAN:", subset2_gen)
print("VDSR :", subset2_vdsr)
print("DRCN :", subset2_drcn)

# Plotting LR, SR (VDSR), and HR for 3 random images
three_random_indices = random.sample(test_indices, 3)

vdsr.eval()

fig, axes = plt.subplots(3, 3, figsize=(12, 12))  # 3 rows (images), 3 cols
(LR, SR, HR)
fig.suptitle("Comparison of LR, SR, and HR")

for i, idx in enumerate(three_random_indices):
    lr_img, hr_img = test_dataset[idx]
    lr_img = lr_img.unsqueeze(0).to(device)
    hr_img = hr_img.unsqueeze(0).to(device)
    with torch.no_grad():
        sr_img = torch.clamp(vdsr(lr_img), 0, 1)

    # Convert tensors to numpy for plotting
    lr_np = lr_img.squeeze().permute(1, 2, 0).cpu().numpy()
    sr_np = sr_img.squeeze().permute(1, 2, 0).cpu().numpy()
    hr_np = hr_img.squeeze().permute(1, 2, 0).cpu().numpy()

    lr_np = np.clip(lr_np, 0, 1)
    sr_np = np.clip(sr_np, 0, 1)
    hr_np = np.clip(hr_np, 0, 1)

    axes[i, 0].imshow(lr_np)
    axes[i, 0].set_title("LR")
    axes[i, 0].axis('off')

    axes[i, 1].imshow(sr_np)
    axes[i, 1].set_title("SR")  # Not mentioning the model name
    axes[i, 1].axis('off')

    axes[i, 2].imshow(hr_np)
    axes[i, 2].set_title("HR")
    axes[i, 2].axis('off')

plt.tight_layout()
plt.savefig("lr_sr_hr_comparison.png")
plt.close()
```