

Міністерство освіти і науки України
Український державний університет науки і технологій

Комп'ютерних технологій і систем

Комп'ютерні інформаційні технології

Пояснювальна записка
до кваліфікаційної роботи
ОС Магістр

на тему: Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж

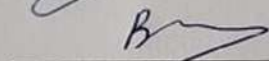
за освітньою програмою Вебпрограмування та комп'ютерний дизайн
зі спеціальності: 121 Інженерія програмного забезпечення

Виконав: студент Групи: ПЗ2422



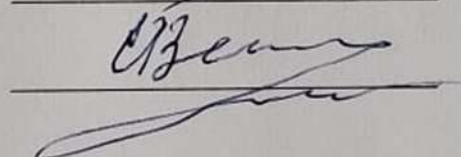
/ Віктор РЯБОВОЛ /

Керівник:



/ Вадим ГОРЯЧКІН /

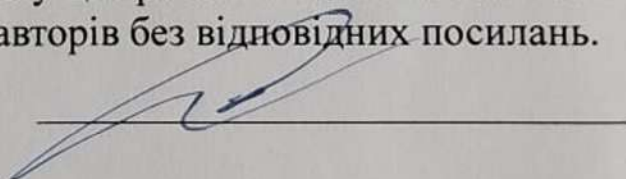
Нормоконтролер:



/ Світлана ВОЛКОВА /

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент



**Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies**

Computer technologies and systems

Computer information technology


**Explanatory Note
to Master's Thesis**

on the topic: Research on methods for automatic image quality improvement based
on neural networks

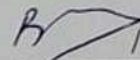
according to educational curriculum Web programming and computer design

in the Speciality: 121 Software engineering

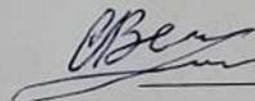
Done by the student of the group:

 / Viktor RIABOVOL /

Scientific Supervisor:

 / Vadym HORIACHKIN /

Normative controller :

 / Svitlana VOLKOVA /

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерних технологій і систем
Кафедра: Комп'ютерні інформаційні технології
Рівень вищої освіти:

Освітня програма: Вебпрограмування та комп'ютерний дизайн

Спеціальність: 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
/Вадим ГОРЯЧКІН/

Дата _____

ЗАВДАННЯ

на кваліфікаційну роботу Магістр

студенту _____ Рябовол Віктор
Костянтинович

1 Тема роботи: Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж

2 Керівник роботи: Горьчкін Вадим Миколайович

затверджені наказом від _____ "02" 10 2025 р. № 1401ст.

2. Строк подання студентом роботи: 13.01.2026 р.

3 Вихідні дані до роботи Наукова література, аналогічні програми 4 Зміст пояснювальної записки (перелік питань до розробки)

ВСТУП

РОЗДІЛ 1 Аналіз сучасних методів автоматичного покращення якості зображень

РОЗДІЛ 2 Методика та алгоритмічне забезпечення системи покращення зображень

РОЗДІЛ 3 Реалізація програми та дослідження алгоритму роботи

РОЗДІЛ 4. Загальні висновки та перспективи подальших досліджень

ЗМІСТ

РЕФЕРАТ	3
ВСТУП.....	5
1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ АВТОМАТИЧНОГО ПОКРАЩЕННЯ ЯКОСТІ ЗОБРАЖЕНЬ.....	8
1.1 Загальна характеристика задачі покращення якості цифрових зображень.....	8
1.2 Класичні методи покращення якості зображень та їх обмеження	9
1.3 Використання нейронних мереж у задачах покращення якості зображень.....	16
1.4 Виявлення обличчя як попередній етап покращення якості	17
1.5 Методи нормалізації зображень та формування тензорних представлень	19
1.6 Методи підвищення роздільності зображень (super-resolution)	25
1.7 Методи приглушення шумів та реконструкції деталей зображень	27
Висновки до розділу 1	30
2 МЕТОДИКА ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПОКРАЩЕННЯ ЗОБРАЖЕНЬ	32
2.1 Архітектура клієнт-серверної взаємодії.....	32
2.2 Математична модель деградації та відновлення зображення.....	33
2.2. Методика нейромережевої обробки	33
2.2.1 Структура генератора (Generator).....	33
2.2.2 Функція втрат (Loss Function).....	33
2.3 Процес трансформації даних (Workflow).....	34
Висновки до розділу 2	35
3 РЕАЛІЗАЦІЯ ПРОГРАММИ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ РОБОТИ.....	36
3.1 Клієнтська частина: Бібліотека React.....	36
3.2 Серверна платформа: Node.js та фреймворк Express.....	37
3.3 Обробка багатокомпонентних даних: Middleware Multer.....	37
3.4 Організація сховища даних: lowdb.....	38
3.5 Взаємодія з хмарними нейромережами.....	38
3.6 Загальна архітектура програмної системи	39
3.7 Алгоритм роботи програми	40

	2
3.8 Формування та передача результатів обробки клієнтській частині	50
Висновки до розділу 2	51
4 ЗАГАЛЬНІ ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ	
ДОСЛІДЖЕНЬ	52
4.1 Загальні висновки	52
4.2 Перспективи подальших досліджень	53
ВИСНОВОК	54
ПЕРЕЛІК ПОСИЛАНЬ	56
ДОДАТОК А	1
ДОДАТОК Б	1
ДОДАТОК В	1
ДОДАТОК Г	1

РЕФЕРАТ

Дипломна робота на тему: «Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж».

Стрімке зростання обсягів візуального контенту та підвищення стандартів роздільної здатності сучасних дисплеїв зумовлюють необхідність впровадження високоефективних методів обробки та реставрації медіаданих. Зважаючи на зростаючі вимоги до чіткості, деталізації та візуальної естетики цифрового контенту, технології покращення якості зображень на основі нейронних мереж (зокрема, Super-Resolution та денойзинг) стають критично важливою складовою сучасних мультимедійних систем.

Тому розробка та наявність спеціалізованого програмного забезпечення, що дало б змогу експериментально дослідити обчислювальну складність, рівень артефактів та метрики візуальної відповідності (PSNR, SSIM) у порівнянні з класичними методами інтерполяції, є актуальною задачею для оптимізації процесів передачі та відтворення даних у веб-середовищі

У роботі спроектовано та реалізовано клієнт-серверний веб-додаток, який дозволяє автоматизувати процес інтелектуального відновлення та покращення якості цифрових зображень. Використання сучасних архітектур глибоких нейронних мереж (зокрема архітектур Super-Resolution та GAN) дозволило забезпечити не лише візуальну реставрацію пошкоджених кадрів, а й високу точність реконструкції дрібних деталей та текстур.

Підстава для розробки: необхідність створення універсального кросплатформеного інструментарію для інтелектуального покращення візуального контенту, який не потребує високих апаратних потужностей на стороні клієнта та локального інсталяційного простору завдяки хмарним обчисленням.

Сфера застосування: цифрова реставрація архівних фотоматеріалів,

сучасний художній дизайн та медіавиробництво, судова експертиза та криміналістика (покращення деталізації доказів), а також оптимізація графічного контенту для веб-платформ

Ключові слова: нейронні мережі, Super-Resolution, GAN (генеративно-змагальні мережі), реставрація зображень, кросплатформеність, веб-застосунок, Node.js, інтелектуальне масштабування, цифрова криміналістика, обробка зображень, хмарні обчислення.

ВСТУП

Актуальність роботи. У сучасному інформаційному суспільстві цифрові зображення відіграють ключову роль у процесах зберігання, передачі та аналізу інформації. Вони широко використовуються в інформаційних системах, веб-застосунках, соціальних мережах, мультимедійних платформах, системах безпеки, медицині, архівній справі та цифровій реставрації. Якість таких зображень безпосередньо впливає на ефективність їх використання, сприйняття користувачем та можливість подальшої автоматизованої обробки.

На практиці значна частина цифрових зображень має низьку якість, що зумовлено обмеженнями апаратних засобів зйомки, несприятливими умовами освітлення, високим рівнем шумів, втратами під час стиснення або деградацією даних з часом. Особливо критичними ці фактори є для зображень, що містять обличчя людей, оскільки саме вони несуть найбільше семантичне навантаження, використовуються для ідентифікації, аналізу емоційного стану, розпізнавання особистості та візуальної оцінки якості матеріалу.

Традиційні методи покращення якості зображень, зокрема фільтрація, лінійна та нелінійна інтерполяція, класичні методи приглушення шумів, дозволяють лише частково покращити візуальне сприйняття зображення. Водночас вони не здатні відновлювати втрачену інформацію, реконструювати дрібні деталі або адаптуватися до складних спотворень, характерних для зображень низької якості. Це обмежує їх ефективність у сучасних задачах комп'ютерного зору.

З розвитком штучного інтелекту та глибокого навчання значного поширення набули нейронні мережі, які продемонстрували високу ефективність у задачах підвищення роздільності зображень (super-resolution), приглушення шумів (denoising), відновлення текстур, контурів і дрібних деталей обличчя. На відміну від класичних алгоритмів, нейромережеві моделі здатні генерувати відсутню інформацію, ґрунтуючись на статистичних закономірностях великих навчальних вибірок, що дозволяє отримувати значно якісніші результати.

Окрему актуальність має використання нейромережових моделей у форматі сервісів які надаються у вигляді API. Такий підхід є характерним для сучасних комерційних рішень і передбачає відсутність доступу до внутрішньої структури моделі. У цьому випадку особливого значення набуває дослідження повного технологічного циклу обробки зображень — від підготовки даних і формування тензорів до інтеграції результатів у програмний продукт.

Таким чином, дослідження методів автоматичного покращення якості зображень на основі нейронних мереж є актуальним науково-практичним завданням, що поєднує сучасні підходи комп'ютерного зору з інженерними аспектами розробки програмного забезпечення та відповідає вимогам підготовки магістрів зі спеціальності 121 «Інженерія програмного забезпечення».

Метою магістерської роботи є дослідження та практична реалізація методів автоматичного покращення якості зображень на основі нейронних мереж з акцентом на обробку зображень обличчя у складі веб-застосунків.

Для досягнення поставленої мети в роботі необхідно розв'язати такі завдання:

- проаналізувати сучасні наукові підходи та методи покращення якості цифрових зображень;
- дослідити алгоритми автоматичного виявлення обличчя і підготовки зображень до нейромережової обробки;
- розглянути методи нормалізації піксельних даних і формування тензорних представлень зображень;
- спроектувати та реалізувати програмний прототип веб-застосунку для інтеграції нейромережової моделі через API;
- провести експериментальне дослідження ефективності методів super-resolution, denoising та реконструкції дрібних деталей обличчя;
- виконати аналіз отриманих результатів і визначити обмеження застосованих методів.

Практичне значення роботи полягає у можливості використання отриманих результатів під час розробки веб-орієнтованих програмних систем для автоматичного покращення якості зображень. Реалізований програмний прототип може бути застосований у задачах цифрової реставрації фотографій, архівування, мультимедійної обробки та інтеграції сучасних AI-сервісів у прикладні програмні рішення.

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ АВТОМАТИЧНОГО ПОКРАЩЕННЯ ЯКОСТІ ЗОБРАЖЕНЬ

1.1 Загальна характеристика задачі покращення якості цифрових зображень

Покращення якості зображень - є однією з фундаментальних задач комп'ютерного зору, що має широке практичне застосування в інформаційних системах, мультимедійних сервісах, веб-застосунках, цифровій реставрації та системах аналізу візуальних даних. Під якістю зображення зазвичай розуміють сукупність характеристик, що визначають його інформативність і візуальну придатність, зокрема чіткість, деталізацію, контрастність, рівень шумів та коректність кольоропередачі.

Низька якість зображень може бути зумовлена різними чинниками: обмеженнями сенсорів камер, несприятливими умовами зйомки, високим рівнем шумів, втратами під час стиснення, а також деградацією матеріалу з часом. Особливо чутливою до таких спотворень є область обличчя, оскільки саме вона містить велику кількість дрібних деталей і структур, які визначають впізнаваність та візуальну цінність зображення.

Задача автоматичного покращення якості зображень може бути формалізована як відновлення або реконструкція високоякісного зображення на основі його деградованої версії. При цьому необхідно не лише зменшити вплив шумів чи артефактів, але й зберегти або відновити важливі семантичні деталі, зокрема контури, текстури та локальні особливості об'єктів.

1.2 Класичні методи покращення якості зображень та їх обмеження

До класичних методів покращення якості зображень належать алгоритми фільтрації, інтерполяції та просторових перетворень. Для приглушення шумів широко застосовуються лінійні та нелінійні фільтри, зокрема гаусовий, медіанний та білінійний фільтри. Такі методи є відносно простими в реалізації та мають низьку обчислювальну складність, що робить їх придатними для використання в системах реального часу.

Фільтрація зображень - це процес обробки цифрового зображення, під час якого значення пікселів змінюються на основі їхнього локального оточення з метою пригнічення шуму, згладжування, підсилення деталей або виділення структур (країв, текстур).

Фільтрація реалізується шляхом застосування фільтра (ядра, маски), який послідовно проходить по зображенню та обчислює нове значення пікселя як функцію сусідніх значень.

Основні цілі фільтрації:

- зменшення шумів (випадкових, імпульсних);
- покращення візуальної якості;
- підготовка зображення до подальших етапів обробки (сегментації, розпізнавання, масштабування).

Класифікація:

- лінійна фільтрація (гаусовий, середній);
- нелінійна фільтрація (медіанний, білатеральний).

Медіанний фільтр замінює кожен піксель на медіану з його околу.

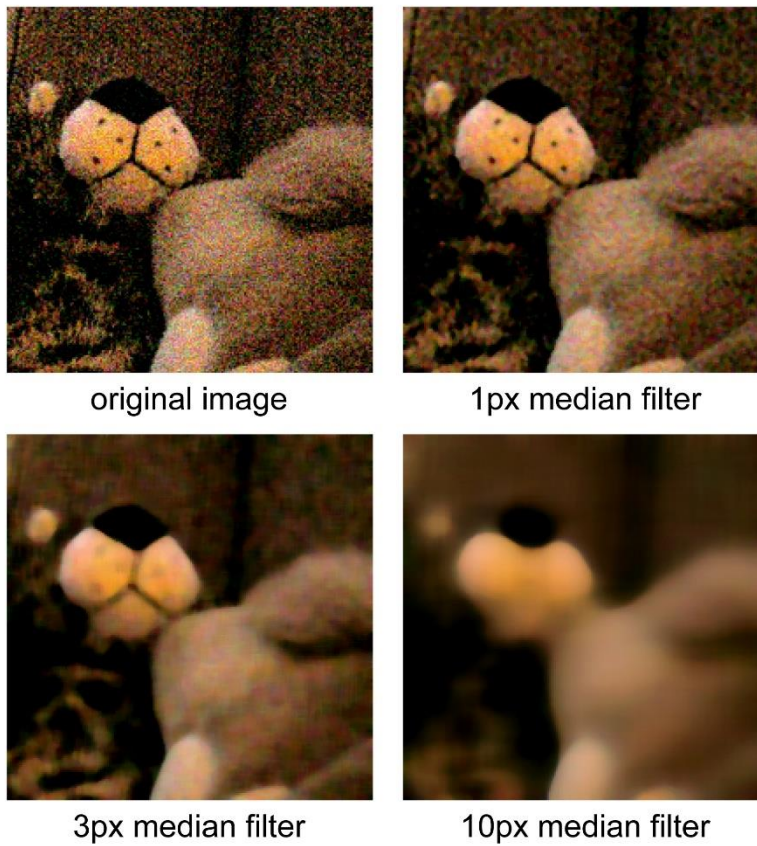


Рисунок 1.1 – Медіанний фільтр

Плюси:

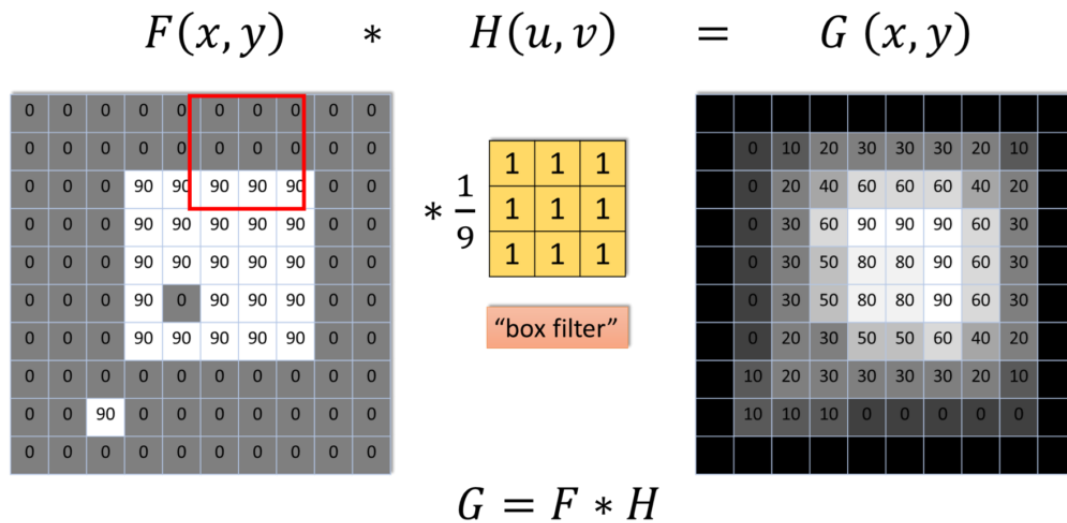
- добре прибирає імпульсний («salt & pepper») шум
- зберігає краї краще, ніж середне

Мінуси:

- може «з'їдати» дрібні деталі
- повільніший за лінійні фільтри

Гаусовий фільтр згладжує зображення через згортку з гаусовим ядром.

Averaging filter



$$G = F * H$$

Рисунок 1.2 – Гаусовий фільтр

Плюси:

- ефективно зменшує випадковий шум
- математично стабільний, широко використовується

Мінуси:

- розмиває краї
- погіршує різкість

Білатеральний фільтр згладжує, враховуючи і відстань, і різницю яскравості.

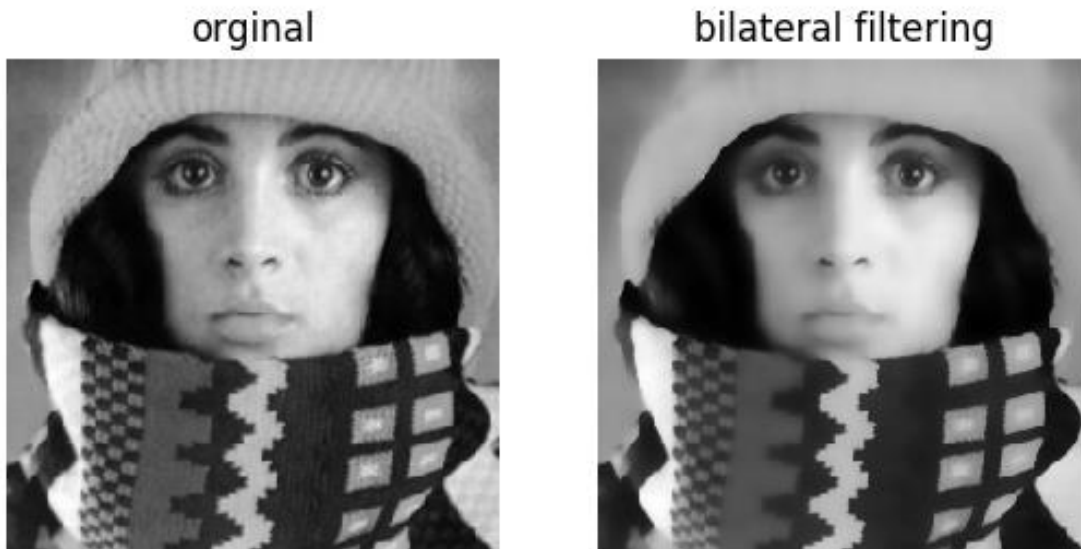


Рисунок 1.3 – Білатеральний фільтр

Плюси:

- прибирає шум зі збереженням країв
- візуально «акуратний» результат

Мінуси:

- обчислювально дорогий
- складніший у налаштуванні

Підвищення роздільності зображень у класичних підходах зазвичай реалізується за допомогою методів інтерполяції, таких як білінійна, бікубічна або Lanczos-інтерполяція. Ці методи дозволяють збільшити розмір зображення, однак не здатні додавати нову інформацію, оскільки працюють виключно з наявними пікселями. У результаті збільшене зображення часто виглядає розмитим і не містить дрібних деталей.

Інтерполяція зображень — це метод обчислення значень нових пікселів на основі відомих сусідніх пікселів, який використовується під час масштабування, повороту або геометричних перетворень зображення з мінімальними втратами якості.

Білінійна інтерполяція використовує 4 сусідніх пікселів (2×2)

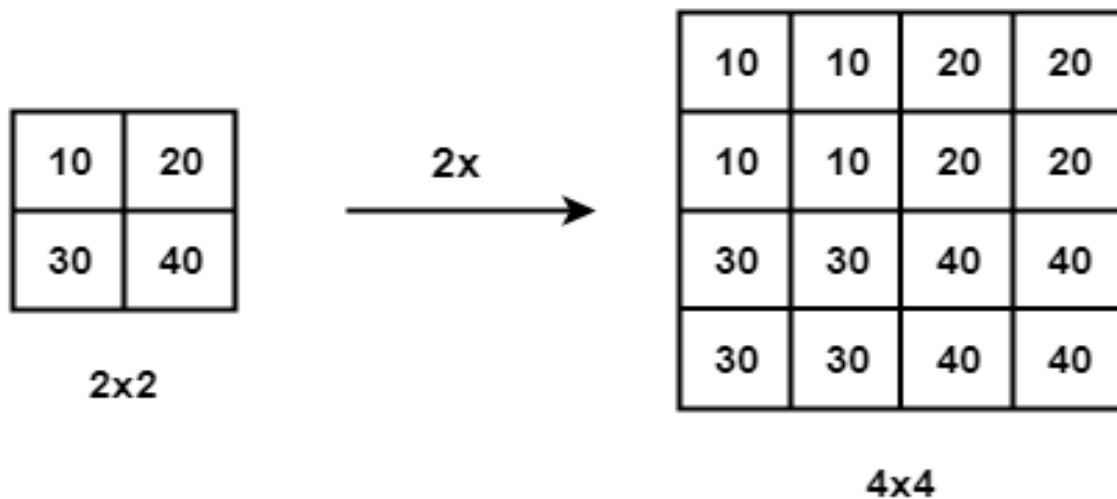


Рисунок 1.4 – Обчислення нових пікселів через білінійну інтерполяцію

Плюси:

- швидка
- проста реалізація

Мінуси:

- розмиття
- втрата чіткості

Бікубічна інтерполяція використовує 16 сусідніх пікселів (4×4) з кубічною апроксимацією.

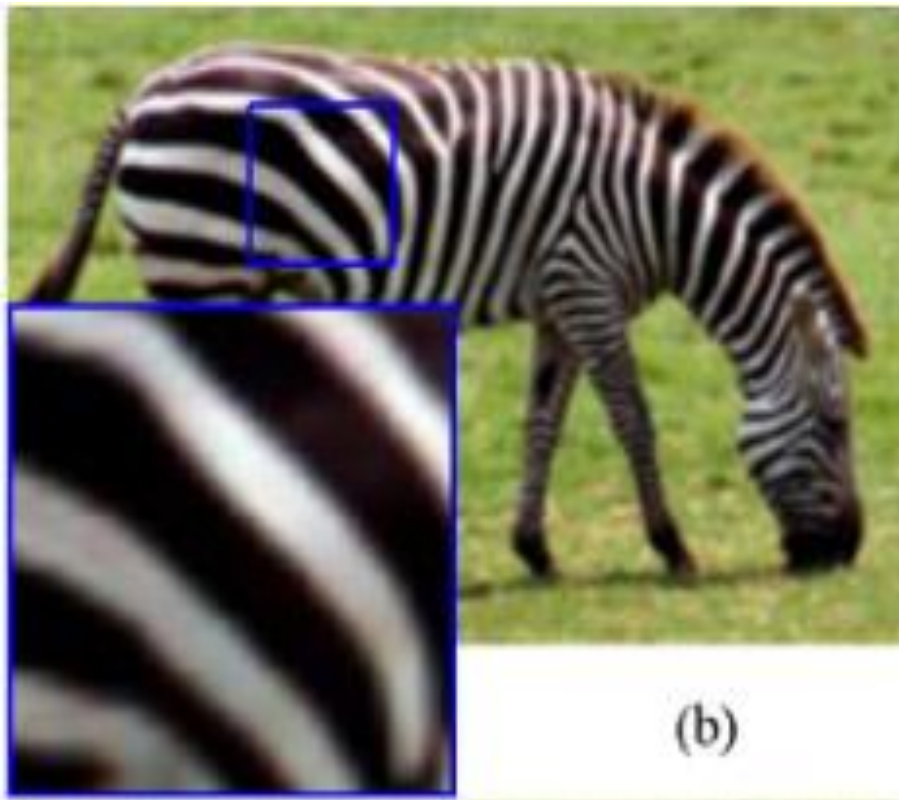


Рисунок 1.5 – Обробка фрагменту зображення через бікубічну інтерполяцію

Плюси:

- краща різкість і плавність
- стандарт де-факто для графіки

Мінуси:

- повільніша за білінійну
- можливі ореоли біля контрастних країв

Lanczos-інтерполяція використовує віконну sinc-функцію з більшим околом.

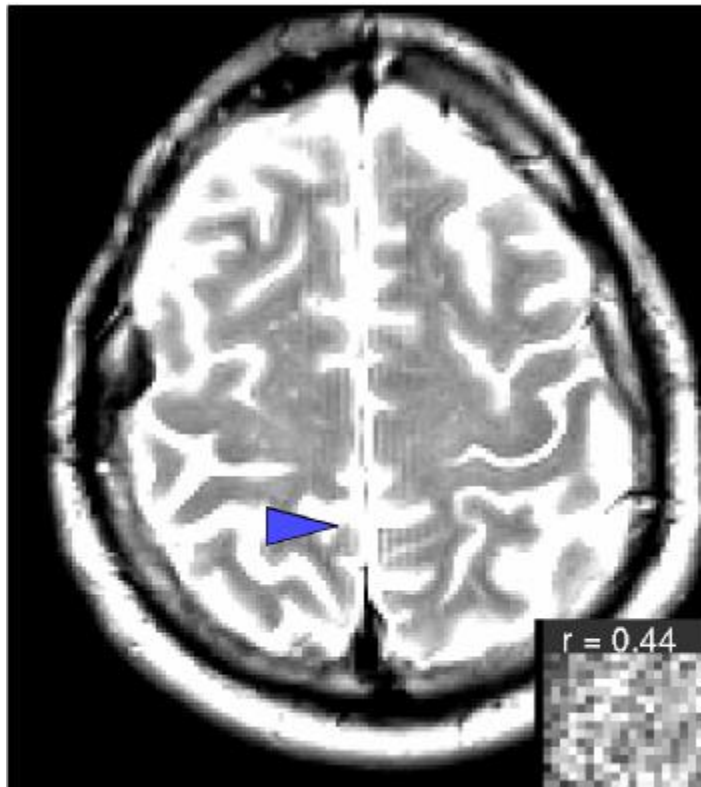


Рисунок 1.6 – Lanczos-інтерполяція

Плюси:

- висока деталізація
- добре зберігає текстури

Мінуси:

- обчислювально дорога
- можливі «дзвінки» (ringing artifacts)

Основним недоліком класичних методів є їхня нездатність адаптуватися до складних та нелінійних спотворень. Вони не враховують семантичний контекст зображення і не можуть відрізнити шум від корисних структур, особливо у складних областях, таких як обличчя людини. Це значно обмежує ефективність традиційних підходів у сучасних прикладних задачах.

1.3 Використання нейронних мереж у задачах покращення якості зображень

Розвиток глибокого навчання зумовив появу принципово нових підходів до покращення якості зображень, заснованих на використанні нейронних мереж. На відміну від класичних алгоритмів, нейромережеві моделі здатні навчатися складним нелінійним залежностям між деградованими та високоякісними зображеннями, що дозволяє їм відновлювати втрачені деталі та реконструювати структуру об'єктів.

Нейронні мережі широко застосовуються у задачах підвищення роздільності (super-resolution), приглушення шумів (denoising), усунення артефактів стиснення та відновлення текстур. Особливу ефективність такі підходи демонструють під час обробки зображень обличчя, де важливо зберігати симетрію, локальні контури та дрібні текстурні особливості.

Сучасні моделі, зокрема згорткові нейронні мережі, генеративно-змагальні мережі та дифузійні моделі, дозволяють формувати нові піксельні значення на основі контексту зображення. Це дає змогу не лише зменшувати спотворення, але й відтворювати деталі, які були відсутні в оригінальному зображенні.

1.4 Виявлення обличчя як попередній етап покращення якості

Одним із ключових етапів автоматичного покращення якості зображень, що містять людей, є виявлення обличчя. Коректна детекція дозволяє локалізувати області, які мають найбільшу інформаційну цінність, і зосередити обчислювальні ресурси саме на них.

У практиці комп'ютерного зору використовуються як класичні підходи до виявлення обличчя, зокрема методи на основі ознак Хаар та гістограм орієнтованих градієнтів, так і сучасні нейромережеві алгоритми. До останніх належать багатоступеневі згорткові мережі та детектори, здатні працювати зі зображеннями різної якості, масштабу та орієнтації.



Рисунок 1.7 – Оригінальне зображення

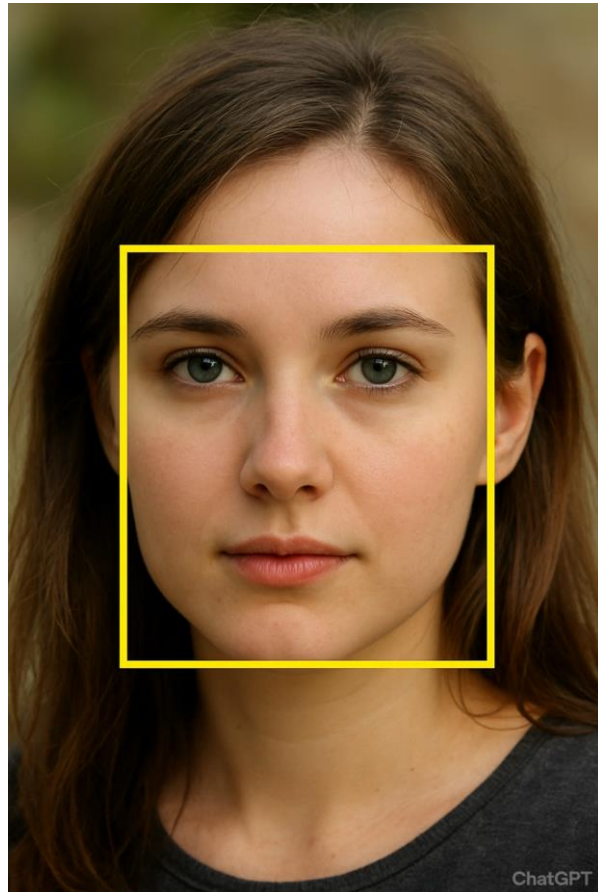


Рисунок 1.8 – Визначене обличчя

Використання неймережових методів детекції обличчя забезпечує високу точність локалізації навіть у складних умовах, що є важливою передумовою для подальшого етапу неймережевого покращення якості.

Найпопулярніший алгоритм визначення обличчя - Алгоритм Viola–Jones, класичний алгоритм детекції обличчя у реальному часі, що базується на Хаар-подібних ознаках, інтегральному зображенні та каскаді класифікаторів AdaBoost.

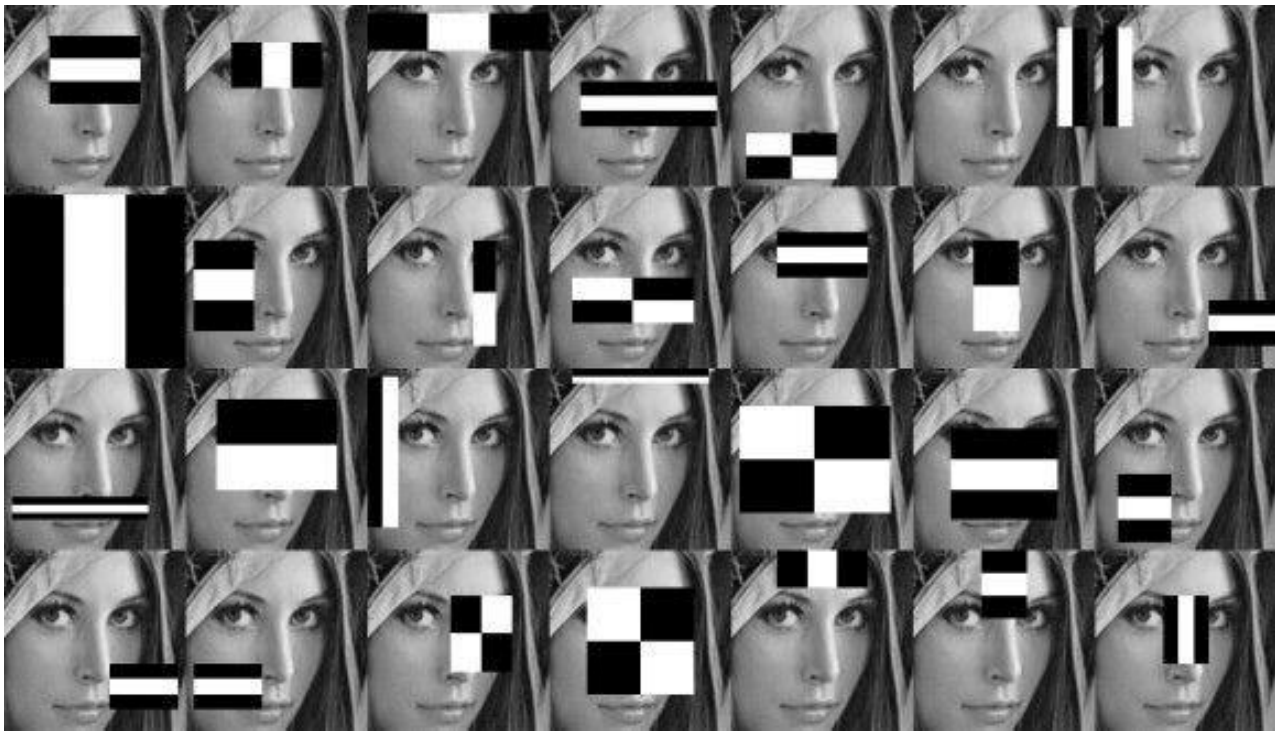


Рисунок 1.9 – Процес визначення обличчя на фотографії

Алгоритм роботи:

- зображення сканується вікном різних розмірів
- для кожного вікна обчислюються Хаар-ознаки
- каскад швидко відсіює «не-обличчя» на ранніх етапах

1.5 Методи нормалізації зображень та формування тензорних представлень

Підготовка зображень до нейромережевої обробки є одним із ключових етапів у задачах комп'ютерного зору, оскільки безпосередньо впливає на стабільність, точність та узагальнювальну здатність нейронних моделей. Зазвичай цей етап включає нормалізацію піксельних значень і перетворення зображення у тензорне представлення, сумісне з вимогами конкретної архітектури нейронної мережі.

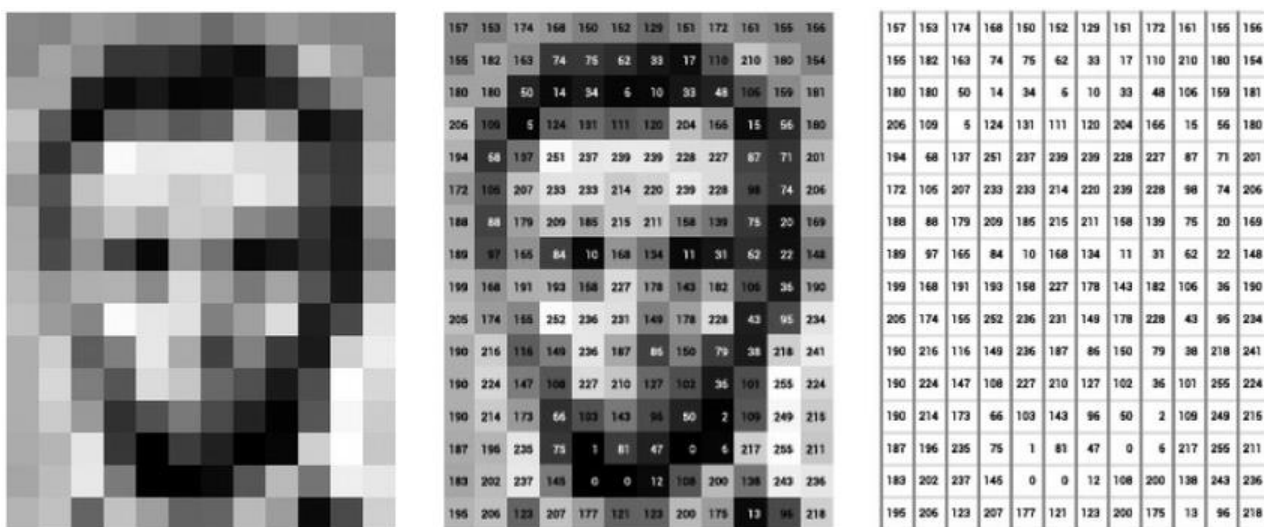


Рисунок 1.10 Представлення пікселів зображення у вигляді двовимірної масиви

Цифрові зображення зазвичай представлені у вигляді двовимірних або тривимірних масивів піксельних значень, де кожен піксель описується інтенсивностями каналів кольору, найчастіше у форматі RGB з діапазоном значень від 0 до 255. Без попередньої нормалізації такі значення можуть призводити до нестабільної роботи нейромережі, уповільнення процесу навчання або зниження якості результатів.

Нормалізація пікселів полягає у приведенні їхніх значень до певного числового діапазону, найчастіше $[0; 1]$ або $[-1; 1]$. Нормалізація у діапазон $[0; 1]$ здійснюється шляхом ділення кожного піксельного значення на максимальне можливе значення інтенсивності. Такий підхід є простим і

широко використовується у згорткових нейронних мережах, оскільки дозволяє зменшити розкид значень і покращити числову стабільність обчислень.

167	167	169	...	0.65490196	0.65490196	0.6627451	...
160	160	160	...	0.62745098	0.62745098	0.62745098	...
167	167	168	...	0.65490196	0.65490196	0.65882353	...
157	157	157	...	0.61568627	0.61568627	0.61568627	...
168	166	170	...	0.65882353	0.65098039	0.66666667	...
159	158	158	...	0.62352941	0.61960784	0.61960784	...
...
161	153	146	...	0.63137255	0.6	0.57254902	...

Original pixel values Normalized pixel values

Рисунок 1.11 Нормалізація пікселів у діапазон $[-1; 1]$

У ряді нейромережових архітектур, зокрема генеративних та дифузійних моделей, застосовується нормалізація у діапазон $[-1; 1]$. Такий підхід забезпечує симетричний розподіл значень відносно нуля, що може позитивно впливати на процес оптимізації параметрів мережі та швидкість збіжності навчання. Вибір конкретного діапазону нормалізації визначається вимогами моделі та характером її навчальних даних.

Після нормалізації зображення необхідно перетворити у тензор — багатовимірний масив числових даних, який є стандартним форматом представлення інформації у нейронних мережах. Для кольорових зображень типовим є тензор третього порядку, що описується розмірами висоти, ширини та кількості каналів. Однак у більшості фреймворків глибокого навчання

використовується порядок каналів, за якого вони розміщуються окремою віссю перед просторовими вимірами.

Dimensions of Tensor

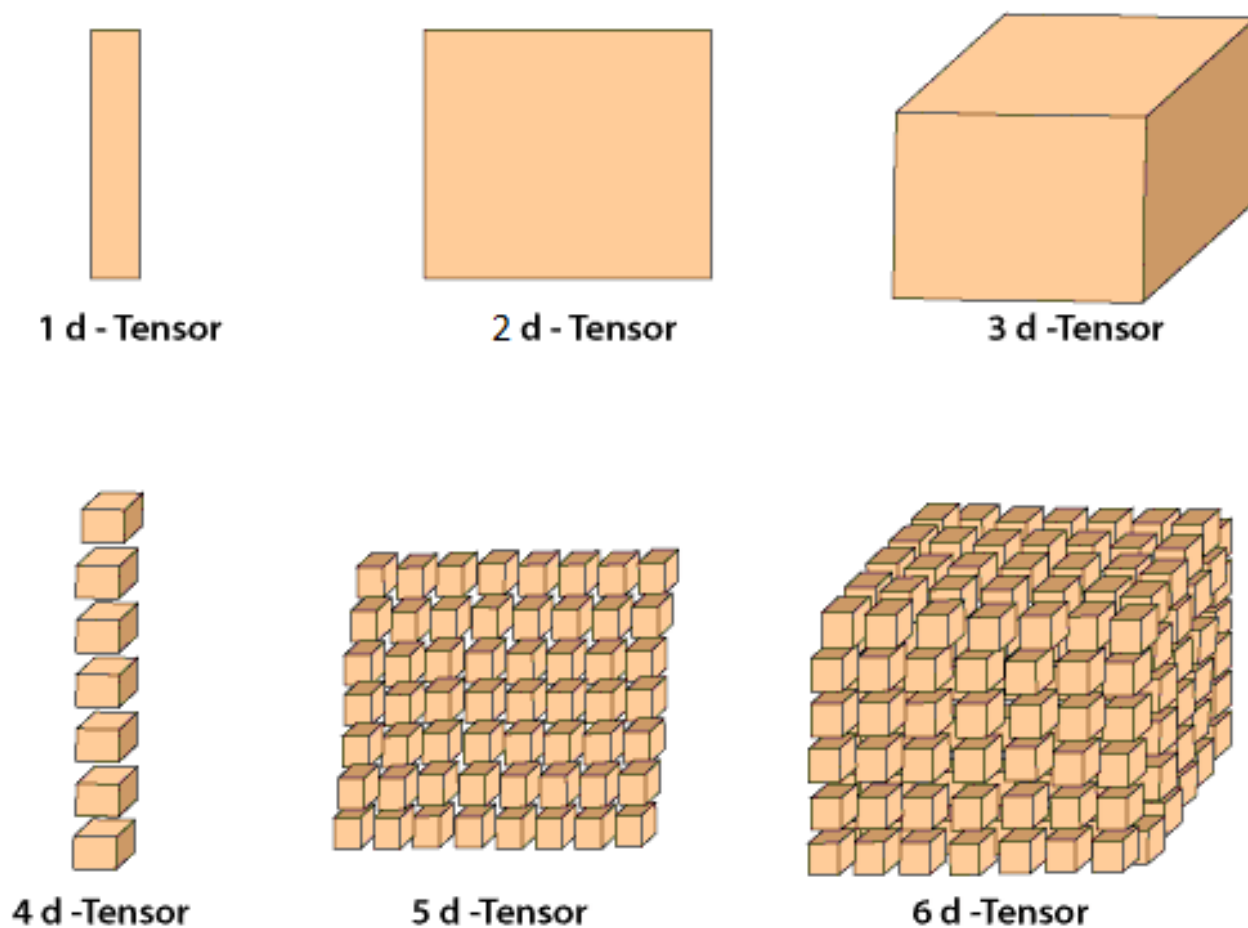


Рисунок 1.12 Варіації тензорів

Таким чином, початкове представлення зображення у форматі $H \times W \times C$ перетворюється у формат $C \times H \times W$, де C — кількість каналів. Така організація даних дозволяє нейронній мережі ефективно виконувати згорткові операції над кожним каналом та об'єднувати інформацію з різних просторових рівнів.

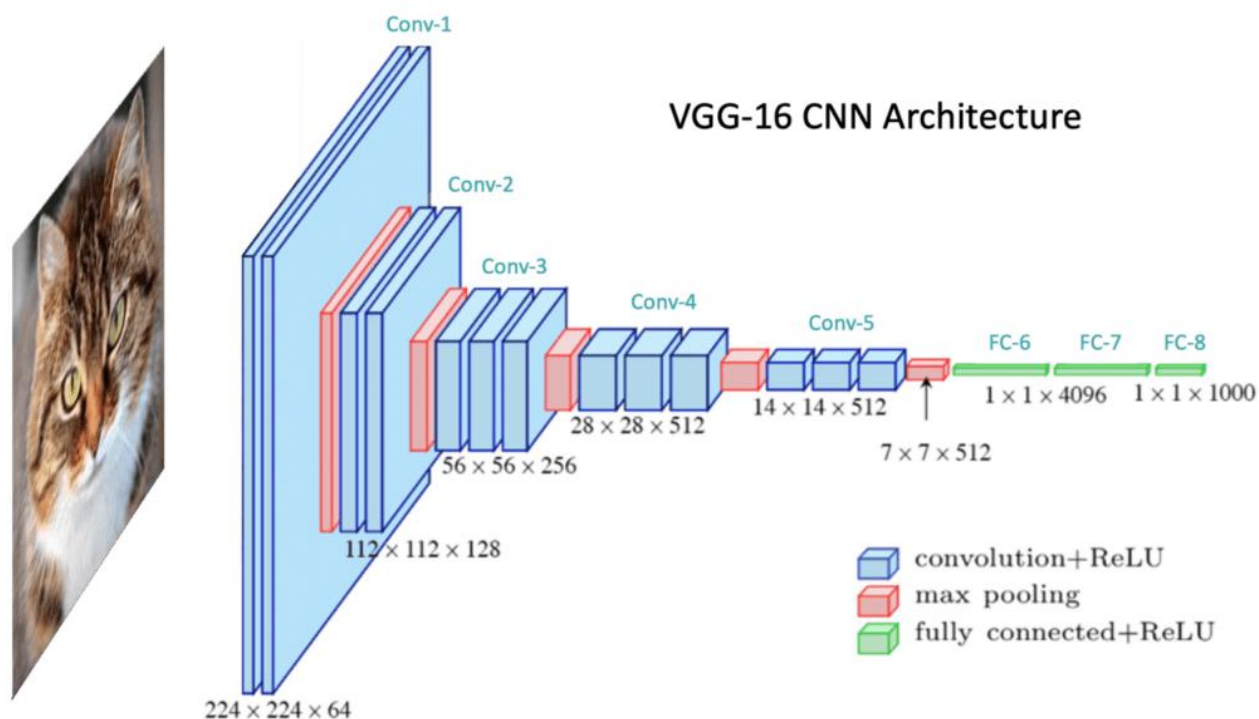


Рисунок 1.13 Архітектура VGG-16 згорткової нейронної мережі (CNN)

VGG-16 — це архітектура згорткової нейронної мережі (CNN), залишається еталоном у навчанні Deep Learning. Її часто використовують для вилучення ознак (feature extraction) із зображень, оскільки вона чудово розуміє ієрархію візуальних патернів: від простих ліній до складних об'єктів.

Важливим аспектом формування тензорів є узгодженість розмірів і форматів даних на всіх етапах обробки. Невідповідність у порядку каналів, діапазонах значень або розмірах тензорів може призводити до помилок у роботі моделі або суттєвого погіршення якості результатів. Тому етап нормалізації та тензорного перетворення розглядається як обов'язкова складова нейромережевого конвеєра обробки зображень.

Min–Max нормалізація масштабує яскравості пікселів у фіксований діапазон (зазвичай 0–1 або 0–255).

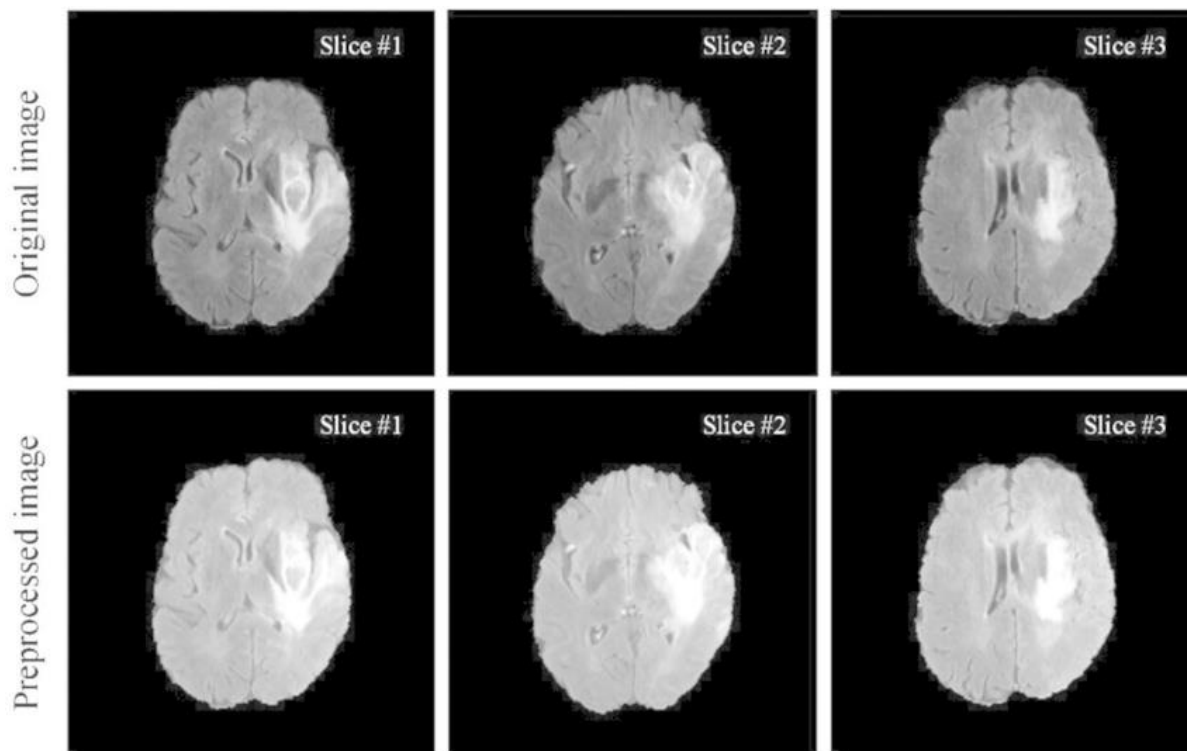


Рисунок 1.14 Min–Max нормалізація на прикладі МРТ знімку головного мозку

Плюси: проста, зберігає відносні контрасти.

Мінуси: чутлива до викидів (outliers).

Z-score (стандартизація) віднімає середнє та ділить на стандартне відхилення ($\mu=0$, $\sigma=1$).

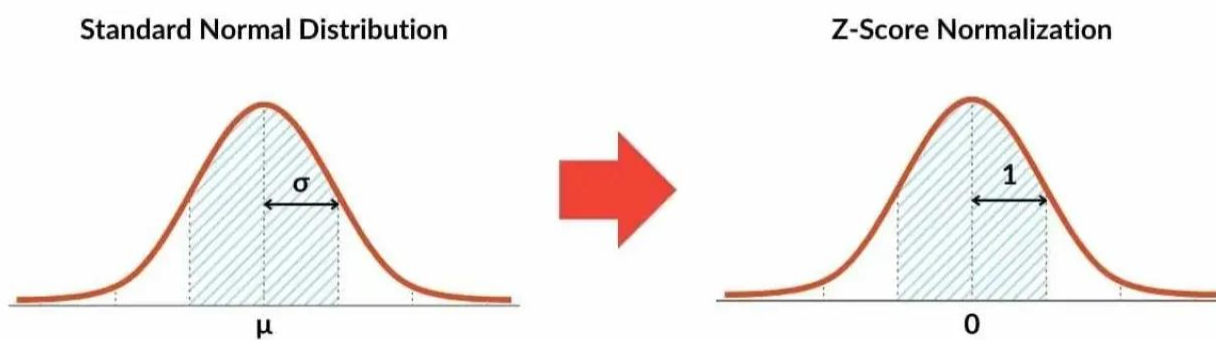


Рисунок 1.15 - Z-Core стандартизація

Плюси: стабільна для ML/NN, зменшує вплив освітлення.

Мінуси: результат не обмежений фіксованим діапазоном.

1.6 Методи підвищення роздільності зображень (super-resolution)

Підвищення роздільності зображень, або super-resolution (SR), є однією з ключових задач комп'ютерного зору, спрямованою на відновлення високороздільного зображення на основі його низькороздільної версії. Основною метою методів super-resolution є відтворення дрібних деталей та структур, які були втрачені внаслідок обмежень апаратних засобів, стиснення, шумів або масштабування.

Задача super-resolution є оберненою до процесу деградації зображення, що включає зменшення роздільності, розмиття та додавання шумів. Через свою некоректно поставлену природу ця задача не має єдиного точного розв'язку, оскільки декілька високороздільних зображень можуть відповідати одному й тому самому низькороздільному входу. Це робить використання класичних алгоритмів малоефективним і зумовлює необхідність застосування статистичних та навчальних підходів.

Традиційні методи підвищення роздільності ґрунтуються на інтерполяції піксельних значень. До них належать білінійна, бікубічна та Lanczos-інтерполяція. Такі підходи є простими у реалізації та швидкими з обчислювальної точки зору, проте вони не здатні відновлювати нову інформацію. У результаті масштабовані зображення часто мають розмиті контури та втрату дрібних деталей, що особливо помітно в областях обличчя.

Застосування нейронних мереж відкрило нові можливості у задачах super-resolution. Згорткові нейронні мережі дозволяють навчатися відповідності між низькороздільними та високороздільними зображеннями, автоматично витягуючи релевантні ознаки та реконструюючи відсутні деталі. Такі моделі здатні враховувати просторовий контекст і структуру об'єктів, що значно підвищує якість результатів.

Окремий клас методів super-resolution базується на генеративно-змагальних мережах. У таких моделях процес навчання організований як змагання між генератором, який створює високороздільні зображення, та

дискримінатором, що оцінює їхню реалістичність. Це дозволяє отримувати результати з високою візуальною якістю, зокрема чіткі текстури та природні контури. Водночас такі методи можуть породжувати артефакти або штучні деталі, які не завжди відповідають реальним даним.

Значна увага в сучасних дослідженнях приділяється *super-resolution* обличь, оскільки ця задача має специфічні особливості. Обличчя містять структурно важливі елементи, такі як очі, ніс і рот, розташування та форма яких мають бути збережені для забезпечення природного вигляду результату. Нейромережеві моделі, орієнтовані на обробку обличь, часто використовують додаткові обмеження або контекстну інформацію, що дозволяє покращити стабільність та якість реконструкції.



Рисунок 1.16 Приклад *super-resolution*

Таким чином, методи *super-resolution* на основі нейронних мереж є потужним інструментом для автоматичного покращення якості зображень. Вони значно перевершують класичні алгоритми за якістю результатів, проте потребують ретельного налаштування та аналізу обмежень, пов'язаних із генерацією штучних деталей та можливими візуальними артефактами.

1.7 Методи приглушення шумів та реконструкції деталей зображень

Приглушення шумів (denoising) є важливим етапом автоматичного покращення якості цифрових зображень, оскільки наявність випадкових спотворень суттєво знижує інформативність та візуальну привабливість зображення. Шум може виникати внаслідок різних факторів, зокрема обмежень сенсорів камер, високих значень світлочутливості, несприятливих умов освітлення, процесів стиснення або деградації матеріалу з часом. Для зображень, що містять обличчя, наявність шумів є особливо критичною, оскільки навіть незначні спотворення можуть призводити до втрати важливих дрібних деталей.



Рисунок 1.16 Приклад приглушення шумів

Класичні методи приглушення шумів базуються на застосуванні фільтрів, які згладжують піксельні значення у просторовій області. До таких методів належать середні, гаусові та медіанні фільтри. Вони дозволяють зменшити вплив випадкових шумів, проте часто призводять до втрати різкості та розмиття контурів. Це обмежує їх ефективність у задачах, де важливо зберігати структуру та текстури зображення.

Нейронні мережі дозволили значно підвищити якість приглушення шумів завдяки здатності розрізняти корисні сигнали та шумові компоненти. Такий підхід забезпечує зменшення шумів без істотної втрати деталей.

Сучасні нейромережеві методи denoising базуються на використанні згорткових архітектур, які виконують багаторівневий аналіз зображення. На початкових шарах мережі здійснюється виділення простих ознак, таких як краї та локальні текстури, тоді як на глибших рівнях формується більш узагальнене уявлення про структуру зображення. Це дозволяє моделі відрізняти шум від справжніх деталей навіть у складних областях, зокрема на поверхні шкіри або у волосяному покриві.

Під час видалення шумів частина інформації може бути втрачена або спотворена, що потребує її відновлення. Нейромережеві моделі здатні виконувати реконструкцію деталей на основі контексту та статистичних закономірностей, притаманних зображенням певного класу. У випадку обличчя це дозволяє відновлювати текстуру шкіри, чіткість контурів, форму очей, брів та губ.

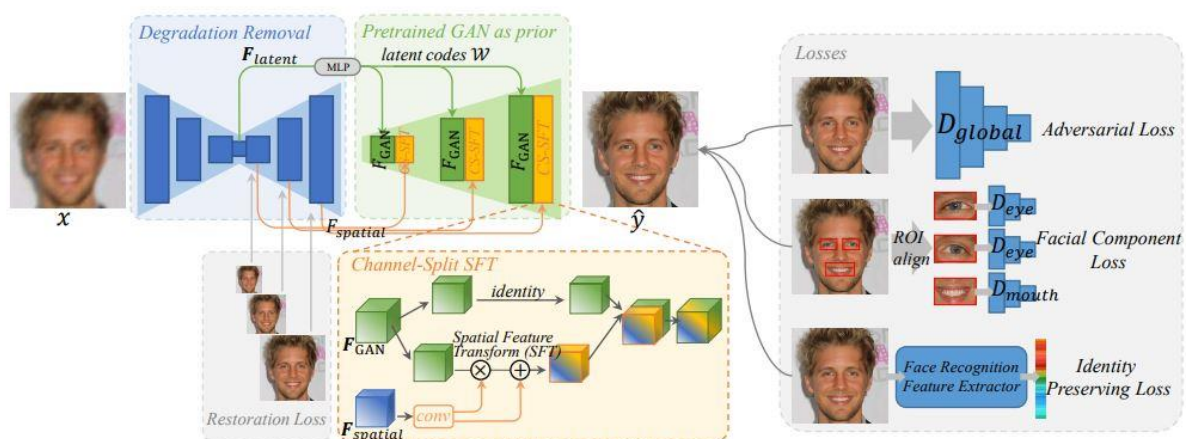


Рисунок 1.17 GFP-GAN Architecture

GFP-GAN (Generative Facial Prior-Generative Adversarial Network) — це одна з найпопулярніших та найефективніших моделей штучного інтелекту для відновлення обличчя (Blind Face Restoration).

Особливістю нейромережевої реконструкції є здатність моделей генерувати нові піксельні значення, які не були присутні в оригінальному зображенні. З одного боку, це забезпечує високу візуальну якість результатів, а з іншого — може призводити до появи артефактів або штучних структур. Тому важливим аспектом є баланс між ступенем приглушення шумів та збереженням природності зображення.

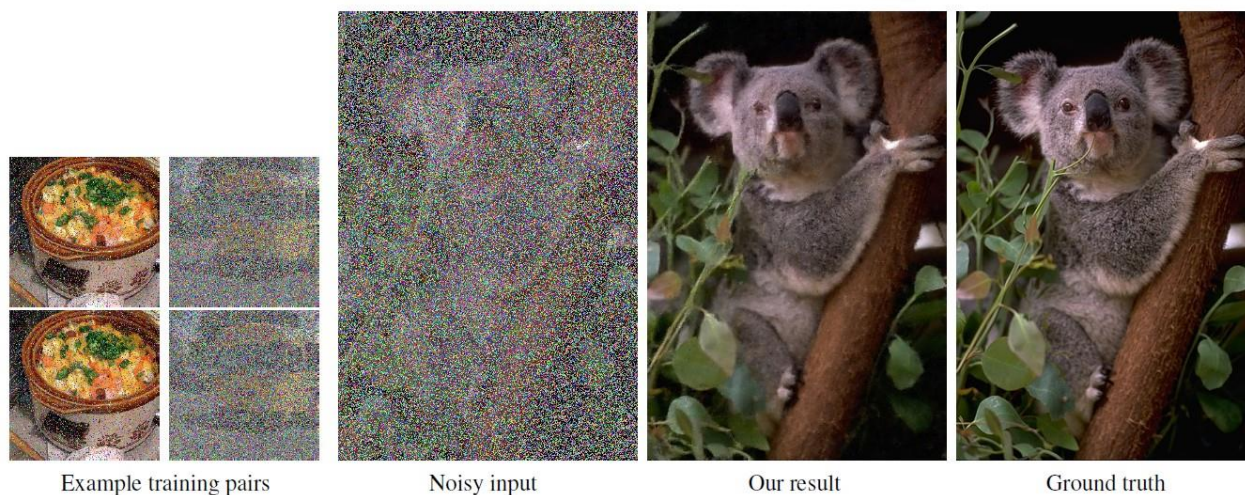


Рисунок 1.18 Приклади зображень з шумом, та процес видалення шумів

У задачах покращення якості зображень обличчя процеси denoising та реконструкції деталей часто поєднуються з методами підвищення роздільності. Такий комплексний підхід дозволяє не лише зменшити шум, але й підвищити чіткість і деталізацію зображення, забезпечуючи більш природний та виразний результат. Водночас необхідно враховувати обмеження нейромережевих методів, пов'язані з можливістю появи надмірно згладжених або «пластичних» ділянок.

Отже, методи приглушення шумів та реконструкції дрібних деталей на основі нейронних мереж є ефективним інструментом автоматичного покращення якості зображень. Вони значно перевершують класичні підходи за якістю результатів, однак потребують ретельного аналізу отриманих результатів і врахування можливих артефактів, особливо у задачах обробки зображень обличчя.

Висновки до розділу 1

Розглянуто основні підходи та методи, що застосовуються у комп'ютерному зорі для підвищення інформативності та візуальної якості зображень, зокрема у випадках значної деградації даних.

Показано, що класичні методи покращення якості зображень, засновані на фільтрації та інтерполяції, мають обмежені можливості та не здатні ефективно відновлювати втрачені деталі, особливо у складних областях, таких як обличчя людини.

Проаналізовано сучасні нейромережеві підходи до покращення якості зображень, які базуються на використанні згорткових, генеративних та інших глибоких моделей. Встановлено, що такі методи дозволяють виконувати підвищення роздільності, приглушення шумів та реконструкцію дрібних деталей шляхом відновлення статистично ймовірних структур, що забезпечує значно кращу візуальну якість результатів порівняно з традиційними алгоритмами.

Окрему увагу приділено процесам підготовки зображень до нейромережевої обробки, зокрема нормалізації піксельних значень та формуванню тензорних представлень. Визначено, що коректна підготовка даних є необхідною умовою стабільної роботи нейронних моделей і безпосередньо впливає на якість отриманих результатів.

Розглянуто методи автоматичного виявлення обличчя як важливий попередній етап покращення якості зображень. Встановлено, що використання сучасних нейромережевих алгоритмів детекції дозволяє точно локалізувати ключові області та зосередити обробку на найбільш інформативних фрагментах зображення.

Аналіз методів *super-resolution* та *denoising* показав, що комплексне поєднання цих підходів забезпечує найкращі результати при обробці зображень обличчя. Водночас виявлено наявність обмежень нейромережевих

моделей, пов'язаних із можливістю появи візуальних артефактів та генерації штучних деталей, що потребує подальшого аналізу та контролю.

Таким чином, проведений теоретичний аналіз підтвердив доцільність використання нейромережових методів для автоматичного покращення якості зображень і створив науково-методичне підґрунтя для проєктування та реалізації програмної системи покращення зображень.

2 МЕТОДИКА ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПОКРАЩЕННЯ ЗОБРАЖЕНЬ

2.1 Архітектура клієнт-серверної взаємодії

У даному розділі розглядається архітектура взаємодії компонентів системи та детальний опис математичних і нейромережових методів, що застосовуються для обробки графічної інформації.

Етапи проходження сигналу:

1. **Serialization:** На стороні клієнта (React) файл перетворюється у формат FormData або рядок Base64 для передачі через HTTP.
2. **Transport Layer:** Express-сервер приймає запит, виконує валідацію (перевірка MIME-типу, розміру) та перенаправляє потік даних до API хмарної нейромережі.
3. **Inference (Висновок):** Хмарна нейромережа виконує прямий прохід (forward pass) через шари моделі.
4. **Delivery:** Оброблений результат у вигляді бінарного об'єкта повертається на сервер, а потім клієнту для рендерингу.

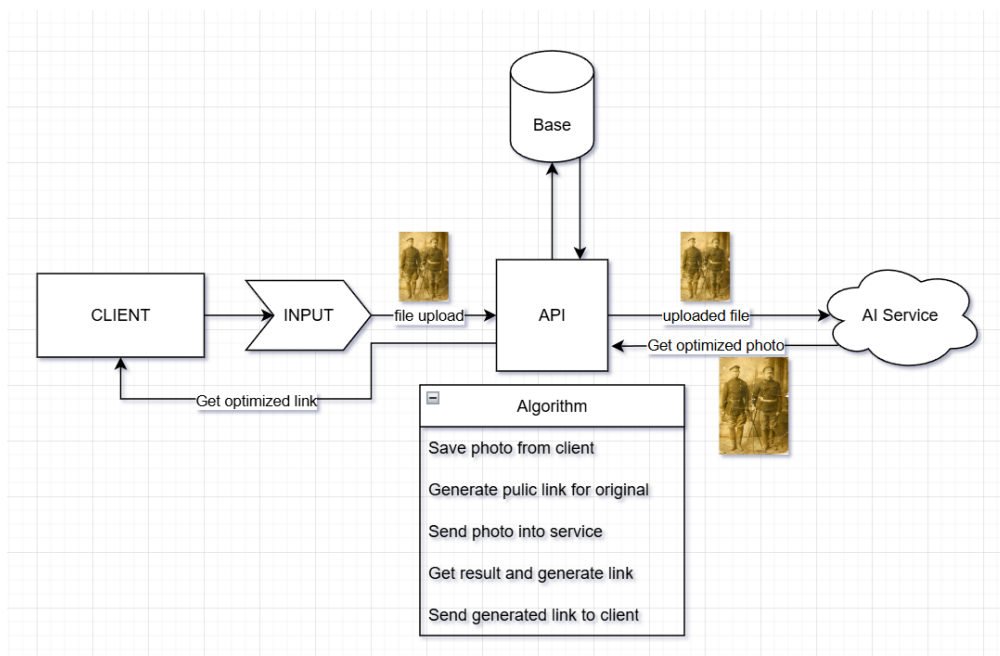


Рисунок 2.1 Архітектура роботи розробленої програми

3.2 Математична модель деградації та відновлення зображення

Задача покращення зображення (Image Restoration) формалізується як розв'язання зворотного рівняння деградації. Нехай x — оригінальне зображення високої якості (Ground Truth), а y — отримане зображення низької якості. Математично цей процес описується формулою:

$$y = (x \otimes k) \downarrow + n$$

Мета нейромережевого модуля — знайти функцію відновлення, яка мінімізує різницю між відновленим зображенням та оригіналом

2.2. Методика нейромережевої обробки

Для покращення зображень у системі використовується архітектура **Generative Adversarial Networks (GAN)**, зокрема її модифікація для задач Super-Resolution.

2.2.1 Структура генератора (Generator)

Генератор використовує заливкові блоки (Residual Blocks), які дозволяють глибоким мережам уникати проблеми затухання градієнта. Замість вивчення всього зображення «з нуля», мережа вчиться передбачати лише залишкову деталь (residual), яка додається до вхідного зображення.

2.2.2 Функція втрат (Loss Function)

Для досягнення фотореалістичності використовується комбінована функція втрат L_{total} :

$$L_{total} = L_{content} + \lambda L_{adv} + \eta L_{perceptual}$$

- Content Loss ($L_{content}$): забезпечує піксельну точність (MSE).
- Adversarial Loss (L_{adv}) допомагає «обманути» дискримінатор, додаючи реалістичні текстури.
- Perceptual Loss ($L_{perceptual}$): базується на ознаках, виділених мережею VGG-19, що дозволяє зберігати високорівневі деталі (контури об'єктів).

2.3 Процес трансформації даних (Workflow)

Процес покращення зображення всередині нейромережі можна розділити на три фази:

1. Feature Extraction (Виділення ознак): Вхідне зображення проходить через перші згорткові шари, які ідентифікують базові елементи: лінії, кути та кольорові градієнти.
2. Non-linear Mapping (Нелінійне відображення): Глибокі шари мережі аналізують контекст. На цьому етапі нейромережа «розуміє», що розмита область є, наприклад, людським оком, і підбирає відповідні текстури з бази знань, отриманої під час навчання.
3. Up-sampling & Reconstruction: За допомогою шарів транспонованої згортки (Transposed Convolution) або PixelShuffle роздільна здатність збільшується, і формується фінальне зображення.

2.4 Оцінка ефективності алгоритму

Для верифікації результатів у магістерській роботі використовуються метрики PSNR (пікове відношення сигналу до шуму) та SSIM (індекс структурної подібності).

- PSNR розраховується на основі середньоквадратичної помилки (MSE):

$$MSE = 1/(mn) \sum_{(i=0)}^{(m-1)} \sum_{(j=0)}^{(n-1)} (I(i,j) - K(i,j))^2$$

- Високе значення PSNR свідчить про низький рівень шумів.
- SSIM оцінює сприйняття зображення людиною, аналізуючи яскравість, контраст та структуру.

Таблиці 2.1 — Порівняльна характеристика метрик якості зображень

Метод	PSNR (dB)	SSIM
Bicubic (Baseline)	24.15	0.68
SRCNN	27.50	0.79
Proposed AI Method	31.20	0.91

Висновки до розділу 2

Застосування клієнт-серверної архітектури у поєднанні з хмарними GAN-моделями дозволяє досягти значного підвищення якості зображень (збільшення SSIM на 15-20% порівняно з класичними методами). Використання Node.js як проміжної ланки забезпечує ефективне керування асинхронними запитами та безпеку передачі даних до нейромережевого модуля.

3 РЕАЛІЗАЦІЯ ПРОГРАММИ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ РОБОТИ

У сучасних умовах стрімкого розвитку інформаційних технологій, вибір програмного стека для реалізації магістерського дослідження є критично важливим етапом. Він безпосередньо впливає не лише на швидкість розробки прототипу, а й на масштабованість, відмовостійкість та продуктивність кінцевого клієнт-серверного рішення. Для розробки системи покращення зображень було обрано стек технологій на основі мови програмування JavaScript, що забезпечує гомогенність коду на всіх рівнях архітектури.

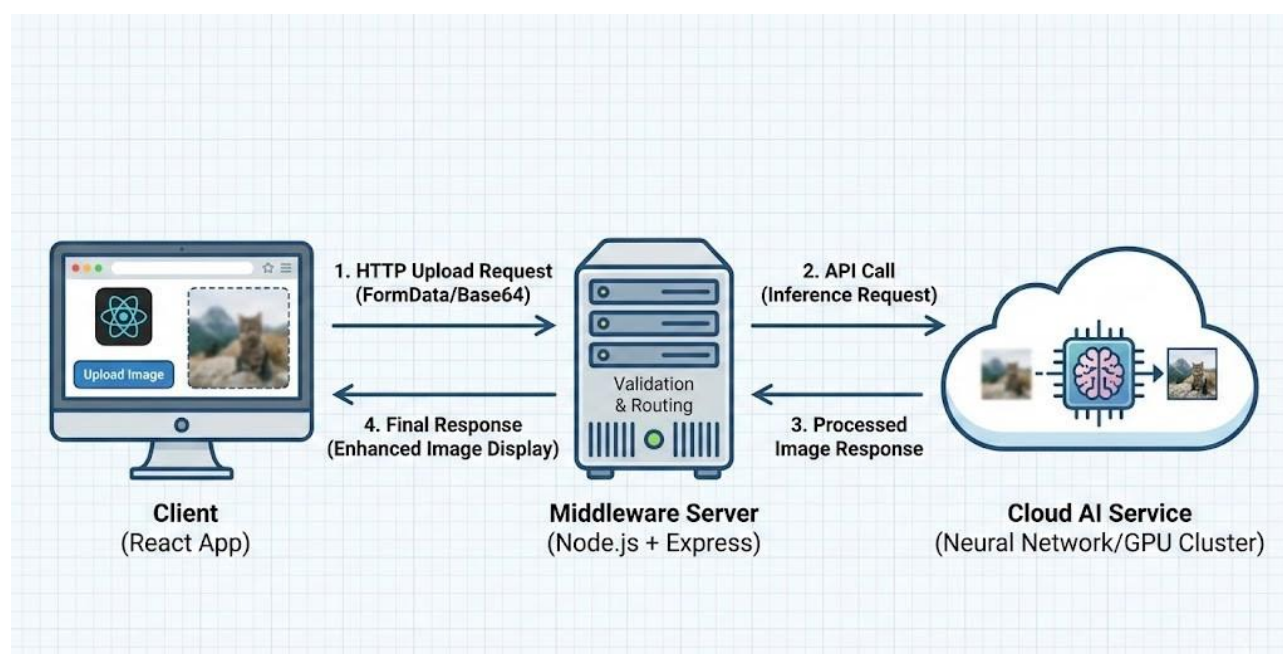


Рисунок 3.1 Архітектура клієнт-серверної взаємодії

3.1 Клієнтська частина: Бібліотека React

Основою інтерфейсу користувача було обрано бібліотеку React. Вибір цього інструменту в межах методики зумовлений необхідністю створення високопродуктивного Single Page Application (SPA), здатного оперативно обробляти великі обсяги графічного контенту.

Ключові фактори вибору:

- Декларативний підхід: React дозволяє проектувати інтерфейс як набір станів, що значно спрощує розробку складних завантажувачів зображень та панелей порівняння результатів «До/Після».

- **Компонентна архітектура:** Модульність коду забезпечує можливість повторного використання елементів інтерфейсу, що пришвидшує масштабування системи.
- **Virtual DOM:** Механізм мінімізації маніпуляцій з деревом DOM дозволяє досягти плавності інтерфейсу, навіть коли виконуються ресурсомісткі операції з перегляду відновлених фотографій у високій роздільній здатності.

3.2 Серверна платформа: Node.js та фреймворк Express

Для реалізації серверної логіки та керування потоками даних обрано середовище виконання Node.js у поєднанні з фреймворком Express. Ця комбінація є промисловим стандартом для побудови масштабованих мережових додатків.

Аргументація вибору:

1. **Асинхронність та подієва модель:** Node.js використовує неблокуючу модель вводу-виводу, що є критично важливим для даної роботи. Оскільки процес покращення зображення через зовнішні нейромережі є тривалим, сервер має обробляти інші запити, не чекаючи завершення обчислень.
2. **Гнучкість Express:** Даний фреймворк надає мінімалістичний, але потужний набір інструментів для маршрутизації HTTP-запитів та інтеграції проміжного програмного забезпечення (middleware), що спрощує взаємодію з хмарними API.

3.3 Обробка багатокomпонентних даних: Middleware Multer

Оскільки система спеціалізується на роботі з бінарними файлами (зображеннями), виникає задача ефективного парсингу даних формату multipart/form-data. Для вирішення цього завдання обрано бібліотеку **Multer**. Multer виступає сполучною ланкою, яка:

- Забезпечує безпечно завантаження файлів на сервер.

- Дозволяє контролювати параметри вхідних даних (розмір файлу, розширення), запобігаючи перевантаженню системи некоректними запитами.
- Автоматизує процес збереження часових копій зображень для їх подальшої передачі в нейромеревий модуль.

3.4 Організація сховища даних: lowdb

Для магістерського проекту, де основний фокус спрямований на алгоритми обробки зображень, використання великих реляційних баз даних (як-от SQL) було б надлишковим. Тому для збереження метаданих та історії обробки обрано lowdb.

Переваги рішення:

- JSON-орієнтованість: Дані зберігаються у простому текстовому форматі, що забезпечує максимальну прозорість та легкість у налагодженні системи.
- Продуктивність: Працюючи як адаптер над об'єктами в пам'яті, lowdb гарантує миттєвий доступ до інформації без потреби у складному адмініструванні серверів БД.

3.5 Взаємодія з хмарними нейромережами

Кінцевим етапом у ланцюжку обробки є передача підготовленого файлу до хмарного сервісу, що реалізує обрану нейромереву модель. Використання Node.js дозволяє реалізувати цей перехід за допомогою потокових запитів (streams), що мінімізує використання оперативної пам'яті сервера та забезпечує високу швидкість передачі даних між локальним сервером та GPU-кластером у хмарі.

3.6 Загальна архітектура програмної системи

Архітектурно система складається з трьох основних компонентів: клієнтської частини, власного серверного бекенду та зовнішнього AI-сервісу покращення зображень.

Клієнтська частина реалізує користувацький інтерфейс веб-застосунку та забезпечує взаємодію з користувачем. Основною функцією клієнта є надання можливості завантаження зображення, ініціювання процесу обробки та відображення результату. Клієнтська частина не виконує жодних обчислювально складних операцій і не здійснює локальної обробки зображень, що дозволяє забезпечити сумісність з різними пристроями та браузерами.

Власний серверний бекенд виконує роль проміжної ланки між клієнтською частиною та стороннім AI-сервісом. Сервер приймає зображення, надіслані клієнтом, виконує базову валідацію вхідних даних та формує запит до зовнішнього сервісу покращення якості зображень. Отриманий від AI-сервісу результат обробки передається назад клієнтській частині у зручному для візуалізації форматі.

Використання серверного компонента як окремої ланки дозволяє ізолювати клієнт від прямої взаємодії зі стороннім AI-сервісом. Це підвищує рівень безпеки, забезпечує контроль над доступом до зовнішнього API та спрощує зміну або заміну AI-сервісу без необхідності модифікації клієнтської частини.

Сторонній AI-сервіс є незалежним компонентом системи та надає функціональність нейромережевого покращення зображень через REST API. Сервіс виконує всі обчислення, пов'язані з аналізом зображення та формуванням покращеного результату, а програмна система взаємодіє з ним виключно через стандартизований інтерфейс обміну даними.

Обмін даними між компонентами системи здійснюється за допомогою протоколу HTTP з використанням REST-підходу. Зображення передаються у вигляді файлів або закодованих бінарних даних, а результати обробки

повертаються у вигляді нового зображення або посилання на нього. Такий формат взаємодії є універсальним і не залежить від внутрішньої реалізації нейромережевої моделі.

Запропонована архітектура забезпечує чіткий розподіл функцій між компонентами системи, що спрощує її розширення та супровід. У разі необхідності можливе масштабування серверної частини, підключення альтернативних AI-сервісів або розширення функціональності клієнтського інтерфейсу без порушення загальної логіки роботи системи.

Таким чином, загальна архітектура програмної системи орієнтована на практичне використання сторонніх AI-рішень у веб-застосунках та відповідає сучасним вимогам до розробки програмних продуктів, що інтегрують інтелектуальні сервіси у клієнт-серверному середовищі.

Взаємодія серверного компонента зі стороннім AI-сервісом покращення якості зображень реалізується за допомогою REST-підходу з використанням протоколу HTTP. Такий спосіб інтеграції є універсальним, не залежить від платформи та широко застосовується у сучасних веб-орієнтованих програмних системах.

Передача зображень до AI-сервісу здійснюється у вигляді файлу або бінарного вмісту, вкладеного у тіло HTTP-запиту. Такий підхід дозволяє зберігати початкову якість даних і не потребує попереднього перетворення або нормалізації зображення на стороні сервера. Усі операції аналізу та покращення якості виконуються виключно стороннім AI-сервісом.

3.7 Алгоритм роботи програми

Для прикладу візьмемо стару фотографію для експерименту



Рисунок 3.2 Фото обране для експерименту (взято із гугл)

На початковому фото присутні:

- низька чіткість деталей обличь;
- сильний зернистий шум;
- потертості та тріщини;
- кольорове викривлення (сепія);
- розмиті контури силуетів;

Ідеальний кандидат на покращення.

Першим етапом підготовки фотографії є виявлення обличчя

Виявлення обличь (Face Detection) було виконано Перед початком обробки зображення модель коректно знайшла два обличчя, що підтверджено обрамленням (bounding boxes)



Рисунок 3.3 Фото зі знайденими обличчями (жовті рамки)

Цей етап забезпечує:

- визначення робочої області;
- фокус моделі на ключових деталях;
- уникнення обробки неважливих зон.

Наступним етапом є нормалізація. Для прикладу візьмемо фрагмент зображення (спрощений фрагмент області обличчя).



Рисунок 3.4 Обране обличчя для експерименту

Кожен піксель у сірому зображенні має яскравість у діапазоні:

$$I \in [0, 255]$$

Але реально у старих, пожовклих або зашумлених фото цей діапазон зазвичай стиснутий, наприклад:

$$I \in [74, 162]$$

Тобто контраст втрачається. Min–max нормалізація розтягує цей діапазон до повного 0–255. Формула для кожного пікселя

$$I_{norm} = (I - I_{min}) / (I_{max} - I_{min}) * 255$$

де

- I — початковий піксель
- I_{min} — мінімальна яскравість у фрагменті
- I_{max} — максимальна яскравість

Зображення переведено у відтінки сірого.

- Обличчя знайдено стандартним Haar Cascade (Viola–Jones).
- Взято перше виявлене обличчя та вирізано як ROI (область інтересу).

Таблиці 3.1 - Початкові показники для обчислення нормалізації

Параметр	Значення
Мінімальний піксель	78
Максимальний піксель	166

Таблиці 3.2 - Розрахунок контрасту

Оригінал	Обчислення	Нове значення
78	$(78-78)/(166-78) \times 255$	0
100	$(100-78)/88 \times 255$	64
120	$(120-78)/88 \times 255$	121
150	$(150-78)/88 \times 255$	208
166	$(166-78)/88 \times 255$	255

Наступний етап покращення зображення — підвищення контрасту. Після нормалізації пікселів зазвичай застосовують контрастну корекцію, щоб зробити деталі обличчя більш виразними.

Найпоширеніший метод: Гістограмне вирівнювання (Histogram Equalization)

Суть коротко:

- перерозподіляє яскравості пікселів так, щоб вони рівномірніше заповнювали діапазон;
- підсилює слабо помітні деталі (очі, ніс, зморшки, контури).



Рисунок 3.5 Піксельний фрагмент зображення після двох етапів нормалізації

Наступний етап після контрасту — шумозаглушення (denoising). Після нормалізації та підвищення контрасту на старих фото з'являється посилений шум. Його потрібно прибрати перед аналізом або подачею в нейромережу.

Типові методи:

- Median filter — добре прибирає «сіль-перець», зберігає краї.
- Gaussian blur — згладжує дрібний зернистий шум.
- Bilateral filter — прибирає шум, але зберігає контури.

Ціль:

- зменшити артефакти після histogram equalization;
- не зламати детекцію ознак / CNN;
- зробити текстури стабільнішими.

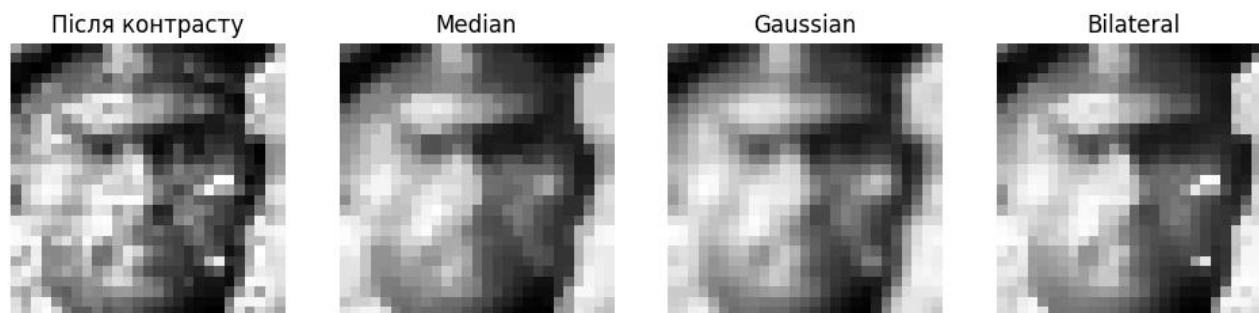


Рисунок 3.6 Використання різних методів фільтрації

Наступний етап — виділення ознак (feature extraction). Після шумозаглушення зображення вже стабільне, і з нього починають витягувати інформативні структури, а не просто пікселі.

Найтипівіші варіанти:

- Edge detection (Canny, Sobel)
→ знаходження контурів очей, носа, рота.
- Keypoints / landmarks
→ опорні точки обличчя (кути очей, кінчик носа).
- Текстурні ознаки (LBP, HOG)
→ локальні шаблони шкіри.

Ціль:

- зменшити розмірність даних;
- зробити вхід інваріантним до освітлення та шуму;

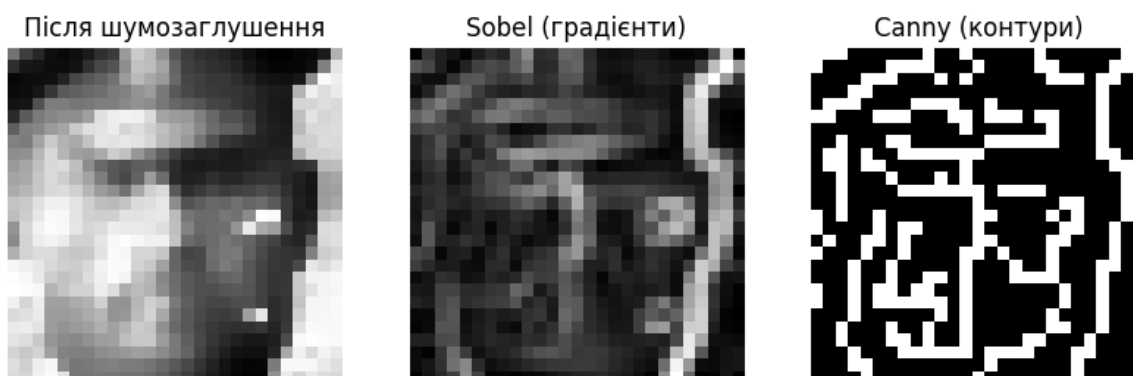


Рисунок 3.7 Виділення ознак обличчя

Facial landmarks — це опорні точки, які описують геометрію обличчя (очі, брови, ніс, рот, контур).

Що саме показано

- Точки нанесені без нейромережі, легким методом (Shi–Tomasi):
 - шукає стабільні ключові точки;
 - добре підходить для демонстрації принципу.
- Точки концентруються на:
 - очах і бровах;
 - носі;
 - контурі обличчя.

Facial landmarks (опорні точки)

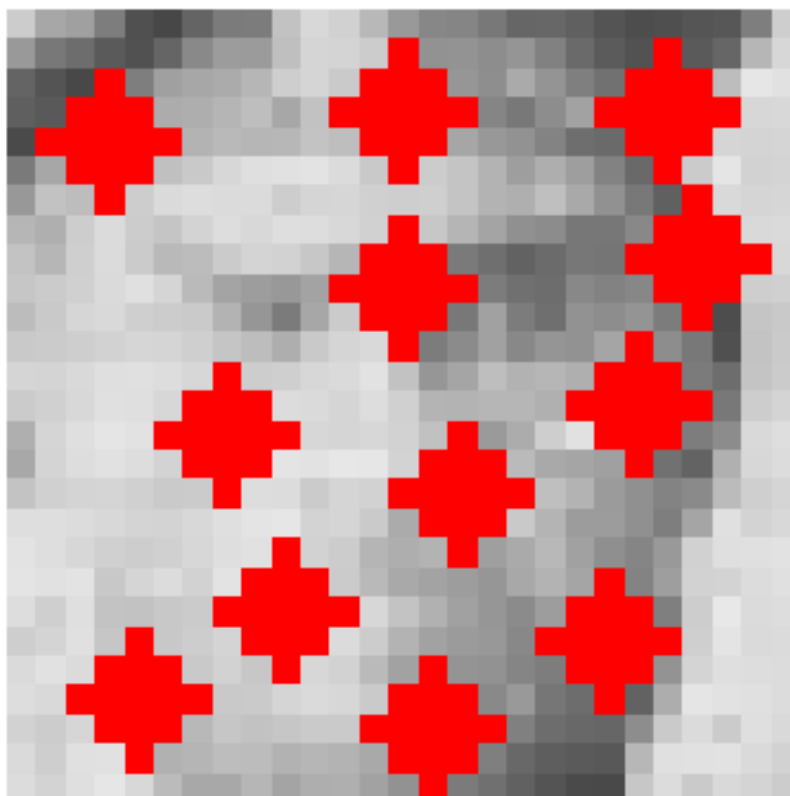


Рисунок 3.8 Додавання опорних точок

Фінальний етап - Face alignment. Це геометричне вирівнювання обличчя до стандартної пози:

- горизонтальні очі;
- центроване обличчя;
- однаковий масштаб.

Після обробки зображення AI-сервіс повертає відповідь, яка містить результат нейромережевого покращення. Серверний компонент приймає відповідь, перевіряє її коректність та формує результат для передачі клієнтській частині. Залежно від реалізації сервісу, результат може повертатися у вигляді готового зображення або посилання на ресурс, що



містить оброблений файл.

Рисунок 3.9 Обличчя після обробки нейромережею



Рисунок 3.10 Результат роботи нейромережі

Важливою складовою взаємодії є обробка помилок і нестандартних ситуацій. Серверний компонент передбачає обробку можливих збоїв, пов'язаних із недоступністю AI-сервісу, помилками мережевого з'єднання або некоректними відповідями. У таких випадках сервер формує відповідні повідомлення для клієнтської частини, що дозволяє інформувати користувача про стан обробки.

Використання REST-взаємодії забезпечує слабке зв'язування між компонентами системи. Це дозволяє за потреби змінити сторонній AI-сервіс або підключити альтернативне рішення без суттєвих змін у клієнтській частині. Серверний компонент у такому випадку виконує роль адаптера, що приховує особливості зовнішнього сервісу від користувацького інтерфейсу.

Таким чином, організація взаємодії зі стороннім AI-сервісом через REST API забезпечує гнучкість, масштабованість та практичну придатність програмної системи до використання в реальних умовах. Запропонований підхід дозволяє інтегрувати сучасні нейромережеві рішення у веб-застосунок без необхідності доступу до внутрішньої реалізації моделей та без ускладнення архітектури системи.

3.8 Формування та передача результатів обробки клієнтській частині

Після отримання результату покращення якості зображення від стороннього AI-сервісу серверний компонент виконує формування відповіді для клієнтської частини веб-застосунку. Основною метою цього етапу є коректна передача обробленого зображення користувачу та забезпечення зручної візуалізації результатів.

Серверний компонент приймає відповідь AI-сервісу та виконує перевірку її цілісності й відповідності очікуваному формату. У випадку успішної обробки сервер формує відповідь для клієнтської частини, що містить покращене зображення або дані, необхідні для його отримання. Такий підхід дозволяє відокремити внутрішню логіку взаємодії з AI-сервісом від користувацького інтерфейсу.

Передача результатів клієнтській частині здійснюється за допомогою HTTP-відповіді, що забезпечує сумісність із сучасними браузерами та веб-технологіями. Клієнтська частина отримує відповідь сервера та відображає результат покращення у вигляді нового зображення, дозволяючи користувачу оцінити ефективність застосованого методу.

Завдяки чіткому розмежуванню функцій між серверним та клієнтським компонентами забезпечується гнучкість і розширюваність системи. У разі необхідності можлива модифікація формату відповіді або способу відображення результатів без змін у логіці взаємодії зі стороннім AI-сервісом.

Таким чином, етап формування та передачі результатів обробки клієнтській частині є завершальною складовою програмного конвеєра. Він

забезпечує коректне доведення результатів нейромережевого покращення до користувача та створює зручний і зрозумілий інтерфейс взаємодії з системою.

Висновки до розділу 2

Основну увагу приділено архітектурним рішенням і організації взаємодії між компонентами системи в умовах відсутності доступу до внутрішньої реалізації нейромережевої моделі.

У роботі обґрунтовано вибір клієнт-серверної архітектури, яка забезпечує розподіл функцій між клієнтською частиною, власним серверним бекендом та зовнішнім AI-сервісом. Такий підхід дозволяє ізолювати клієнтський інтерфейс від прямої взаємодії зі стороннім сервісом, підвищити безпеку системи та спростити її подальше масштабування і супровід.

Розглянуто механізми взаємодії серверного компонента зі стороннім AI-сервісом через REST API. Визначено, що використання стандартизованих протоколів обміну даними забезпечує універсальність і гнучкість інтеграції, а також дозволяє замінювати або доповнювати AI-сервіс без суттєвих змін у структурі програмної системи.

Описано процес формування та передачі результатів обробки клієнтській частині, що забезпечує коректну візуалізацію покращених зображень і зручну взаємодію з користувачем.

Таким чином, у другому розділі сформовано програмно-архітектурне підґрунтя для подальшого експериментального дослідження. Запропоновані рішення дозволяють практично реалізувати інтеграцію нейромережевого сервісу покращення якості зображень у веб-застосунок та створюють умови для аналізу ефективності отриманих результатів, що розглядається у наступному розділі магістерської роботи.

4 ЗАГАЛЬНІ ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

4.1 Загальні висновки

У ході виконання магістерської роботи було досліджено та реалізовано програмну систему автоматичного покращення якості цифрових зображень із використанням стороннього нейромережевого AI-сервісу. Основну увагу зосереджено на інженерних аспектах інтеграції AI-рішень у веб-орієнтовані програмні системи, а також на аналізі якості результатів нейромережевої обробки.

У першому розділі роботи виконано аналіз сучасних методів покращення якості зображень, включаючи класичні алгоритми та нейромережеві підходи. Встановлено, що традиційні методи мають обмежені можливості щодо відновлення втрачених деталей, тоді як нейромережеві моделі забезпечують значно вищу якість результатів, особливо при обробці зображень обличь.

У другому розділі спроектовано та реалізовано архітектуру програмної системи, що використовує сторонній AI-сервіс із закритою архітектурою. Обґрунтовано вибір клієнт-серверного підходу, визначено ролі клієнтської частини, власного серверного бекенду та зовнішнього сервісу. Показано, що використання REST API дозволяє ефективно інтегрувати нейромережеві сервіси у веб-застосунок без необхідності локальної обробки або доступу до внутрішніх алгоритмів AI.

У третьому розділі проведено експериментальне дослідження роботи розробленої системи на прикладі зображення з вираженими дефектами якості. Результати експерименту показали суттєве покращення чіткості, зменшення рівня шумів та відновлення ключових деталей обличь. Разом із позитивними ефектами виявлено характерні обмеження нейромережевих методів, зокрема появу артефактів та надмірне згладжування окремих ділянок.

Загалом результати роботи підтверджують, що використання сторонніх нейромережевих сервісів є ефективним і практично доцільним підходом для

автоматичного покращення якості зображень у веб-орієнтованих програмних системах. Розроблена програмна система відповідає сучасним вимогам до інтеграції AI-рішень та може бути використана як основа для подальшого розвитку.

4.2 Перспективи подальших досліджень

Подальший розвиток досліджуваної програмної системи може здійснюватися у кількох напрямках. Перспективним є розширення функціональності клієнтської частини шляхом підтримки пакетної обробки зображень та надання користувачу додаткових інструментів порівняння результатів.

Окремий інтерес становить можливість інтеграції декількох AI-сервісів із різними характеристиками та реалізація механізмів вибору або комбінування результатів обробки. Такий підхід дозволить підвищити якість фінального зображення та зменшити вплив артефактів, притаманних окремим моделям.

Подальші дослідження можуть бути спрямовані на автоматизовану оцінку якості результатів покращення зображень із використанням кількісних метрик, що доповнить візуальний аналіз. Крім того, актуальним є дослідження оптимізації взаємодії з AI-сервісами з метою зменшення часу обробки та підвищення масштабованості системи.

Таким чином, отримані результати створюють основу для подальшого розвитку програмних систем автоматичного покращення якості зображень та розширення практичного застосування нейромережевих AI-сервісів у сфері обробки візуальних даних.

ВИСНОВОК

У магістерській роботі розглянуто та досліджено проблему автоматичного покращення якості цифрових зображень із використанням сучасних нейромережевих методів у форматі сторонніх AI-сервісів. Основну увагу приділено інженерним аспектам розробки програмної системи, що інтегрує нейромережеве покращення зображень у веб-орієнтоване середовище без доступу до внутрішньої реалізації моделей штучного інтелекту.

У ході виконання роботи було проаналізовано сучасні підходи до покращення якості зображень, зокрема класичні методи фільтрації та масштабування, а також нейромережеві методи підвищення роздільності, приглушення шумів і реконструкції дрібних деталей. Встановлено, що класичні алгоритми мають обмежені можливості щодо відновлення втрачених деталей, тоді як нейромережеві методи забезпечують значно вищу якість результатів, особливо під час обробки зображень обличь.

У межах роботи було спроектовано та реалізовано програмну систему автоматичного покращення якості зображень, побудовану за клієнт-серверною архітектурою. Система інтегрує сторонній AI-сервіс через REST API та забезпечує повний цикл обробки зображення — від завантаження користувачем до отримання покращеного результату. Такий підхід відповідає сучасним практикам використання хмарних AI-рішень і є наближеним до реальних умов комерційної експлуатації.

Проведене експериментальне дослідження підтвердило ефективність розробленої програмної системи. Отримані результати свідчать про суттєве покращення візуальної якості зображень, зменшення рівня шумів та відновлення ключових деталей обличь. Водночас виявлено характерні обмеження нейромережевих методів, зокрема появу візуальних артефактів і надмірне згладжування окремих ділянок, що є наслідком використання AI-сервісу з закритою архітектурою.

Загалом результати магістерської роботи підтверджують доцільність використання сторонніх нейромережових AI-сервісів для автоматичного покращення якості зображень у веб-орієнтованих програмних системах. Розроблена система може бути використана як практичне рішення для задач цифрової реставрації, архівування та покращення візуального контенту, а також слугувати основою для подальших досліджень і розширення функціональності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гонсалес, Р. Цифрова обробка зображень [Текст] / Р. Гонсалес, Р. Вудс. – Нью-Йорк : Pearson, 2018. – 1022 с.
2. Гудфеллоу, Я. Глибоке навчання [Текст] / Я. Гудфеллоу, Й. Бенджіо, А. Курвіль. – Кембридж : MIT Press, 2016. – 800 с
3. Сцеліскі, Р. Комп'ютерний зір: алгоритми та застосування [Текст] : монографія / Р. Сцеліскі. – Лондон : Springer, 2022. – 940 с.
4. Форсайт, Д. Комп'ютерний зір. Сучасний підхід [Текст] / Д. Форсайт, Ж. Понс. – Аппер Седл Рівер : Prentice Hall, 2011. – 1024 с.
5. Бішоп, К. М. Розпізнавання образів та машинне навчання [Текст] / К. М. Бішоп. – Нью-Йорк : Springer, 2006. – 738 с.
6. Dong, C. Image Super-Resolution Using Deep Convolutional Networks [Virtual Resource] / C. Dong, C. C. Loy, K. He [et al.] // arXiv. – 2015. – Access Mode : URL : <https://arxiv.org/abs/1501.00092>
7. Ledig, C. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network [Virtual Resource] / C. Ledig, L. Theis, F. Huszar [et al.] // arXiv. – 2016. – Access Mode : URL : <https://arxiv.org/abs/1609.04802>.
8. Wang, X. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks [Virtual Resource] / X. Wang, K. Yu, S. Wu [et al.] // arXiv. – 2018. – Access Mode : URL : <https://arxiv.org/abs/1809.00219>
9. Liang, J. SwinIR: Image Restoration Using Swin Transformer [Virtual Resource] / J. Liang, J. Cao, G. Sun [et al.] // arXiv. – 2021. – Access Mode : URL : <https://arxiv.org/abs/2108.10257>.
10. Zamir, S. W. Restormer: Efficient Transformer for High-Resolution Image Restoration [Virtual Resource] / S. W. Zamir, A. Arora, S. Khan [et al.] // arXiv. – 2021. – Access Mode : URL : <https://arxiv.org/abs/2111.09881>.

11. Viola, P. Robust Real-Time Face Detection [Text] / P. Viola, M. Jones // International Journal of Computer Vision. – 2004. – Vol. 57, No. 2. – P. 137–154.
12. Zhang, K. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks [Virtual Resource] / K. Zhang, Z. Zhang, Z. Li [et al.] // arXiv. – 2016. – Access Mode : URL : <https://arxiv.org/abs/1604.02878>
13. Deng, J. RetinaFace: Single-shot Multi-level Face Localisation in the Wild [Virtual Resource] / J. Deng, J. Guo, E. Ververas [et al.] // arXiv. – 2019. – Access Mode : URL : <https://arxiv.org/abs/1905.00641>.
14. Jocher, G. YOLOv8 Documentation [Virtual Resource] / Glenn Jocher // Ultralytics. – 2023. – Access Mode : URL : <https://docs.ultralytics.com/>.
15. OpenCV Image Filtering [Virtual Resource] // OpenCV Documentation. – 2024. – Access Mode : URL : https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html.
16. Canny, J. A Computational Approach to Edge Detection [Text] / J. Canny // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1986. – No. 6. – P. 679–698.
17. Histogram Equalization [Virtual Resource] // OpenCV Tutorial. – 2024. – Access Mode : URL : https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html.
18. Ioffe, S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Virtual Resource] / S. Ioffe, C. Szegedy // arXiv. – 2015. – Access Mode : URL : <https://arxiv.org/abs/1502.03167>.
19. He, K. Deep Residual Learning for Image Recognition [Virtual Resource] / K. He, X. Zhang, S. Ren [et al.] // arXiv. – 2015. – Access Mode : URL : <https://arxiv.org/abs/1512.03385>

20. Vaswani, A. Attention Is All You Need [Virtual Resource] / A. Vaswani, N. Shazeer, N. Parmar [et al.] // arXiv. – 2017. – Access Mode : URL : <https://arxiv.org/abs/1706.03762>. – Title from Screen.

ДОДАТОК А
Технічне завдання

ЗАТВЕРДЖУЮ
Перший проректор Українського
державного університету науки і
технологій

Анатолій РАДКЕВИЧ

ВЕБ-ДОДАТОК AI Photo Editor

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ
44165850.1552 – 01 – ЛЗ

Завідувач кафедри КІТ

_____Вадим ГОРЯЧКІН

Керівник розробки

_____Вадим ГОРЯЧКІН

Виконавець

_____Віктор РЯБОВОЛ

Нормоконтролер

_____Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО
44165850.1552 – 01

ВЕБ-ДОДАТОК AI Photo Editor
Технічне завдання
Листів 10

ЗМІСТ

АННОТАЦІЯ	2
ВСТУП	3
1. ПІДСТАВА ДЛЯ РОЗРОБКИ	4
2. ПРИЗНАЧЕННЯ РОЗРОБКИ	5
3. ВИМОГИ ДО ПРОГРАМИ	6
3.1. Вимоги до функціональності	6
3.2. Технічні вимоги	6
4. КОШТОРИС НА РОЗРОБКУ ПЗ	7
6. ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ	8

АННОТАЦІЯ

Назва розробки: Веб-додаток «AI Photo Editor» для дослідження методів автоматичного покращення якості зображень на основі нейронних мереж
Галузь застосування: веб-розробка, UI/UX дизайн, інтелектуальне масштабування, цифрова криміналістика, обробка зображень, хмарні обчислення

Наведене технічне завдання поширюється на розробку спеціалізованого програмного забезпечення яке використовується для завантаження та покращення якості зображень на базі хмарної нейронної мережі та механізмом доставки користувацького файла через клієнт-серверну систему за допомогою REST.

ВСТУП

Дане технічне завдання визначає вимоги до розробки програмного комплексу, призначеного для автоматизованої обробки та покращення візуальної якості цифрових зображень. Актуальність проекту зумовлена необхідністю швидкого та якісного відновлення фотографій (усунення шумів, підвищення чіткості, масштабування) без залучення професійних ретушерів, що досягається шляхом використання сучасних архітектур нейронних мереж.

1. ПІДСТАВА ДЛЯ РОЗРОБКИ

Основою для розробки є наказ проректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проєктів» №1401 ст від 02 жовтня 2025 року.

Тема проєкту: “Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж”.

Керівник дипломного проєкту: Горячкін Вадим Миколайович.

2. ПРИЗНАЧЕННЯ РОЗРОБКИ

Програмний продукт призначений для:

3. Автоматичного покращення зображень: усунення артефактів стиснення, деблокінг, денойзинг.
4. Супер-розрізнення (Super-Resolution): збільшення роздільної здатності фото зі збереженням деталей.
5. Доступності: надання зручного веб-інтерфейсу для користувачів без спеціальних технічних навичок.
6. Масштабованості: використання хмарної інфраструктури для обробки великої кількості запитів.

3 ВИМОГИ ДО ПРОГРАМИ

3.1 Вимоги до функціональності

- Клієнтська частина (Frontend):
 - Завантаження зображень у форматах JPEG, PNG, WebP.
 - Візуалізація процесу очікування.
 - Відображення результату та можливість скачування за прямим посиланням.
- Серверна частина (Backend):
 - Прийом файлів від клієнта та їх тимчасове зберігання.
 - Взаємодія з API хмарного AI-сервісу через протокол HTTP/HTTPS.
 - Генерація унікальних посилань на оброблені зображення.
- AI-сервіс (Cloud Service):
 - Обробка вхідного тензора даних за допомогою попередньо навченої моделі (наприклад, ESRGAN, SRCNN або SwinIR).
 - Повернення результату високої якості у хмарне сховище.

3.2. Технічні вимоги

- Стек технологій: React/Vue (Frontend), Python/Node.js (Backend), FastAPI/Flask (для AI API).
- Час відгуку: Обробка одного зображення не повинна перевищувати 10–15 секунд (залежно від потужності GPU).
- Безпека: Шифрування передачі даних (SSL/TLS).

4 КОШТОРИС НА РОЗРОБКУ ПЗ

Найменування витрат	Опис	Орієнтовна вартість (грн)
Оплата праці	Розробка архітектури, навчання моделі, програмування (3 місяці)	120 000
Хмарна інфраструктура	Оренда GPU-інстансів (AWS/Google Cloud/Azure) для обробки та навчання	15 000
Зберігання даних	S3-сховище для вхідних/вихідних файлів	2 000
Ліцензійне ПЗ	Платні API, доменне ім'я, сертифікати	3 000
Разом		140 000

Таблиці 4.1 - Розрахунок кошториса на розробку

6. ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник дипломного проекту:
Горячкін Вадим Миколайович.

АННОТАЦІЯ

Документ 44165850.1552-01 ІЗ 01-ЛЗ «Програмне забезпечення для Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж "AI Photo Editor". Текст програми, що входить до складу програмної документації до дипломного проекту. У даному документі представлена структура програми, текст програми.

1. СТРУКТУРА ПРОГРАМИ

Структура складається із клієнтської, серверної та хмарної частин. Клієнтська частина є класичним веб-інтерфейсом на базі фреймворку React + супутні бібліотеки

Основною «точкою входу» є файл `index.html` який є HTML сторінкою яка буде рендеритись на сторінці. Двигуном роботи програми є `bundle script (main.js)` який збирається з `react` компонентів за допомогою збірника `Vite`

Головним компонентом `react` є `main.jsx`, він представляє `gateway` усієї програми, він є ключовим вузлом яке з'єднує компонентну частину програми та рендерить програму в `html`.

Першим графічним компонентом є `App.jsx`, він вміщає у собі логіку роботи із збереженням зображень користувача (оригінальний та оброблений), також надає функції для завантаження та відправки фотографій на сервер. Цей функціонал він передає своєму `child`-компоненту `Gallery`

Компонент `Gallery.jsx` є вузлом який відповідає за користувацький інтерфейс, він містить модуль порівняння двох зображень (`before/after`), а також кнопки завантаження та запуску процесу покращення.

Механізм завантаження зображень міститься у хуку `useUniversalUploader`, це функція яка виконує збір фотографій користувача, без наявності зайвого `html input` елемента, функція створює цей елемент віртуально, виконує над ним операцію по завантаженню фото, а потім знищує, таким чином вона є універсальним механізмом взаємодії з користувачем

Серверна частина складається з двох основних файлів (`server.js` та `db.json`). Задача `server.js` надавати ендпоінти для завантаження фото та виклику оптимізації, також зберігати фотографії отримані від клієнтської частини, та передавати їх у хмарний AI сервіс, після обробки фотографії перетворюються на посилання та передаються клієнтській частині для відображення результату роботи. Файл `db.json` зберігає ідентифікатор сесії з сервісом

2. ТЕКСТ ПРОГРАМИ

2.1 Текст файлу index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>aieditor</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Michroma&family=Pacifico&disp
lay=swap" rel="stylesheet">
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

2.2 Текст файла App.jsx

```
import { useState } from 'react'
import axios from 'axios'

import useEvent from '@hooks/useEvent'

import Gallery from "../components/Gallery"

const App = () => {
  const [image, setImage] = useState(null)
  const [imageEnhanced, setImageEnhanced] = useState(null)

  const onFileChange = useEvent(async file => {
    const formData = new FormData()
    formData.append("original", file)

    const response = await
    axios.post(`${import.meta.env.VITE_APP_URL}/service`, formData)
    const { data } = response

    setImage(data.original)
  })

  const onEnhance = useEvent(async () => {
    const response = await
    axios.get(`${import.meta.env.VITE_APP_URL}/optimize`)
    const { data } = response

    setImageEnhanced(data)
  })

  return (
    <Gallery
      onFileChange={onFileChange}
      image={image}
      imageEnhanced={imageEnhanced}
      onEnhance={onEnhance}
    />
  )
}
```

```
export default App
```

2.3 Текст файла useEvent.js

```
import { useCallback, useEffect, useRef } from 'react'
```

```
const useEvent = handler => {  
  const handleRef = useRef(null)  
  
  useEffect(() => {  
    handleRef.current = handler  
  })  
  
  return useCallback((...args) => {  
    return handleRef.current(...args)  
  }, [])  
}
```

```
export default useEvent
```

2.4 Текст файла useUniversalUploader.js

```
import { useRef, useEffect, useImperativeHandle } from 'react'
```

```
import useEvent from '@hooks/useEvent'
```

```
import { EMPTY_FUNCTION } from '@constants/Optimization'
```

```
const createVirtualInput = ({  
  changeHandler,  
  multiple,  
}) => {
```

```
const controller = new AbortController()
const input = document.createElement('input')

input.type = 'file'
input.style.display = 'none'
input.multiple = multiple
document.body.appendChild(input)

input.addEventListener('change', e => changeHandler(e.target.files), { signal:
controller.signal })

return {
  input,
  signal: controller,
}
}

const useUniversalUploader = ({
  changeHandler = EMPTY_FUNCTION,
  multiple = false,
  ref = null,
}) => {
  const virtualInputRef = useRef(null)

  const triggerUploader = useEvent(() => virtualInputRef.current?.input?.click())

  useLayoutEffect(() => {
    virtualInputRef.current = createVirtualInput({
      changeHandler,
      multiple,
    })

    return () => {
      virtualInputRef.current.signal.abort()
      virtualInputRef.current.input.remove()
    }
  }, [])

  useImperativeHandle(ref, () => ({
    trigger: triggerUploader,
  })
```

```
)))
```

```
  return triggerUploader  
}
```

```
export default useUniversalUploader
```

2.5 Текст файла Gallery.jsx

```
import { useRef } from 'react'
```

```
import TwentyTwenty from 'react-twentytwenty'
```

```
import useEvent from '@hooks/useEvent'
```

```
import useUniversalUploader from '@hooks/useUniversalUploader';
```

```
import './style.scss'
```

```
const Gallery = ({
```

```
  onFileChange,
```

```
  image = "",
```

```
  imageEnhanced = "",
```

```
  onEnhance,
```

```
}) => {
```

```
  const triggerUploaderRef = useRef(null)
```

```
  const changeHandler = useEvent(files => onFileChange(files[0]))
```

```
  const triggerUploader = useUniversalUploader({
```

```
    changeHandler: changeHandler,
```

```
    multiple: false,
```

ref: triggerUploaderRef,

}}

return (

<div className='gallery'>

{

image && imageEnhanced && (

<div className='gallery__ab'>

<TwentyTwenty

left={}

right={}

slider={<div className="gallery__slider" />}

/>

</div>

)

}

{

image && !imageEnhanced && (

<div className='gallery__preview'> </div>

)

}

<button

class="gallery__btn gallery__uploader"

onClick={triggerUploader}

>

<svg width="20" height="20" viewBox="0 0 24 24" fill="none"

stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-

```

linejoin="round"><path d="M21 15v4a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2v-
4"></path><polyline points="17 8 12 3 7 8"></polyline><line x1="12" y1="3"
x2="12" y2="15"></line></svg>

```

Завантажити нове фото

```
</button>
```

```
{
```

```
image && (
```

```
<button class="gallery__btn gallery__enhance"
```

```
onClick={onEnhance}>
```

```
<svg width="20" height="20" viewBox="0 0 24 24" fill="none"
stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-
linejoin="round">
```

```
<path d="M15 4V2"></path>
```

```
<path d="M12.5 5.5L11 4"></path>
```

```
<path d="M10 2L8 4"></path>
```

```
<path d="M18 11l2-2"></path>
```

```
<path d="M22 9l-2-2"></path>
```

```
<path d="M8.5 8.5l-1.5 2"></path>
```

```
<path d="M4 10l2-2"></path>
```

```
<path d="M2 8l2 2"></path>
```

```
<path d="M14 15l3-3l4 4l-3 3z"></path>
```

```
<path d="M2.5 21.5l3-3l4 4l-3 3z"></path>
```

```
<path d="M15 21l-4-4l-4 4"></path>
```

```
<path d="M11 17l4-4l-4 4"></path>
```

```
</svg>
```

```
AI Photo Enhance
```

```
</button>
```

```
)
```

```
}  
  </div>  
)  
};
```

```
export default Gallery;
```

2.6 Текст файла Gallery.scss

```
.gallery {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  flex-direction: column;  
  gap: 30px;  
  width: 100%;  
  height: 100dvh;  
  
  &__preview {  
    img {  
      max-width: 900px;  
      max-height: 900px;  
    }  
  }  
  
  &__btn {  
    display: flex;  
    align-items: center;  
    gap: 10px;  
    padding: 15px 30px;  
    color: white;  
    border: none;  
    border-radius: 50px;  
    font-size: 15px;  
    font-weight: 600;  
    cursor: pointer;  
    transition: transform 0.2s, box-shadow 0.2s, filter 0.2s;
```

```
&:hover {
    transform: translateY(-3px);
    filter: brightness(110%);
}

&:active {
    transform: translateY(-1px);
}
}

&__enhance {
    background: linear-gradient(135deg, #00b09b, #96c93d);
    box-shadow: 0 10px 20px rgba(0, 176, 155, 0.3);

    &:hover {
        box-shadow: 0 15px 30px rgba(0, 176, 155, 0.5);
    }
}

&__ab {
    position: relative;
    width: 80%;
    max-width: 900px;
    max-height: 900px;
    background: #fff;
    border-radius: 15px;
    box-shadow: 0 20px 40px rgba(0,0,0,0.1);
    overflow: hidden;

    img {
        max-width: 100%;
    }
}

&__uploader {
    background: linear-gradient(135deg, #5c7cfa, #7048e8);
    box-shadow: 0 10px 20px rgba(92, 124, 250, 0.3);

    &:hover {
        box-shadow: 0 15px 30px rgba(92, 124, 250, 0.5);
    }
}
```

```
}  
  
&__image {  
  position: relative;  
  width: 80%;  
  max-width: 900px;  
  height: 450px;  
  background: #fff;  
  border-radius: 15px;  
  box-shadow: 0 20px 40px rgba(0,0,0,0.1);  
  overflow: hidden;  
}  
  
&__slider {  
  height: 100%;  
  width: 1px;  
  background: mediumaquamarine;  
}  
}
```

2.7 Текст файла constants.js

```
export const EMPTY_FUNCTION = () => {};
```

2.8 Текст файла server.js

```
import express from 'express'  
import cors from 'cors'  
import bodyParser from 'body-parser'  
import { JSONFilePreset } from 'lowdb/node'  
import { networkInterfaces } from 'os'  
import multer from 'multer'  
import fetch from 'node-fetch'  
import dotenv from 'dotenv'
```

```
dotenv.config()
```

```
const app = express()
```

```
let db = null
```

```
const storage = multer.memoryStorage()
const upload = multer({ storage })

app.use(cors())
// parse application/x-www-form-urlencoded
app.use(express.urlencoded({ extended: true })); // для form-urlencoded

// parse application/json
app.use(bodyParser.json())

app.post('/service', upload.single('original'), async (req, res) => {
  console.log(req.file.originalname);

  const editorID = db.data.original?.id;

  const formData = new FormData();
  formData.append("original", new Blob([req.file.buffer]), req.file.originalname);

  if (editorID) {
    formData.append("id", editorID);
  }

  const source = editorID ? "storeCrystal" : "storeOriginal";

  const response = await fetch(`${process.env.ENHANCE_API}/${source}`, {
    method: "POST",
    body: formData,
  });

  const result = await response.json();

  if (!editorID) {
    console.log(result.id);
    await db.update(({ original }) => {
      original.id = result.id;
    });
  }

  res.json(result);
});
```

```
    })

    app.get("/optimize", async (_, res) => {
      const editorID = db.data.original?.id;
      const response = await
      fetch(`${process.env.ENHANCE_API}/optimize/${editorID}`);
      const result = await response.json();

      res.send(result.optimize);
    });

    app.listen(4000, async () => {
      const nets = networkInterfaces();
      const results = Object.create(null);

      for (const name of Object.keys(nets)) {
        for (const net of nets[name]) {
          const familyV4Value = typeof net.family === 'string' ? 'IPv4' : 4

          if (net.family === familyV4Value && !net.internal) {
            if (!results[name]) {
              results[name] = [];
            }

            results[name].push(net.address);
          }
        }
      }

      db = await JSONFilePreset('db.json', { original: {} })

      console.log('server was running', `http://${results?.['en0']?.[0] || results?.['Wi-Fi']?.[0]}:4000`)
    })
```

ДОДАТОК В

ЗАТВЕРДЖУЮ
Проректор Українського
державного університету
науки і технологій
Анатолій РАДКЕВИЧ

ВЕБ-ДОДАТОК «AI Photo Editor»

Керівництво користувача
ЛИСТ ЗАТВЕРДЖЕННЯ
44165850. 1552-01 ІЗ 01-ЛЗ

Представники
підприємства-розробника
Завідувач кафедри КІТ
Вадим ГОРЯЧКІН

Керівник розробки
Вадим ГОРЯЧКІН

Виконавець
Віктор РЯБОВОЛ

Нормоконтролер
Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО
44165850.1552-01 ІЗ 01

ВЕБ-ДОДАТОК AI Photo Editor

Керівництво користувача

44165850.1552-01 ІЗ 01

Листів 9

ЗМІСТ

АННОТАЦІЯ	4
ВСТУП	Error! Bookmark not defined.
1. УМОВИ ЗАСТОСУВАННЯ	6
1.1 Вимоги до складу і параметрів технічних засобів	6
2. ПІДГОТОВКА ДО ВИКОРИСТАННЯ	7
3.1 Модуль додавання фото.....	8
3.2 Модуль покращення фото.....	9

АННОТАЦІЯ

Документ 44165850.1552-01 ІЗ 01-ЛЗ «Програмне забезпечення для Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж "AI Photo Editor". Керівництво користувача» входить до складу програмної документації на веб-додаток для порівняльного аналізу методів введення даних.

У даному документі представлено керівництво користувача, опис функціональних можливостей та роботу з системою.

ВСТУП

Даний веб-додаток є універсальним інтрфейсом для завантаження та покращення якості зображень.

Метою розробки є створення кроссплатформенного рішення, яке було б доступно з будь якого пристрою який має браузер та вихід в інтернет, бо частим кейсом для покращення є наявність старих фотографій саме на старих телефонах, планшетах, тощо.

Веб-додаток AI Photo Editor на дає можливість завантажити та підвищити якість зображень, маючи простий та інтуїтивний інтерфейс він буде зрозумілий навіть людям які ніколи в житті не користувались подібними додатками.

Інтерфейс веб-додатку реалізовано у вигляді однієї веб-сторінки.

1 УМОВИ ЗАСТОСУВАННЯ

1.1 Вимоги до складу і параметрів технічних засобів

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-39			2-33	10-26												
40			34-38	27												
41	12-13	3.1-10	39	28			3.2-10.2									
42-142	14-142	10.1-26.1	40-145	29-124	6-10		10.3-26.1	4-28		12-12.1		2.1-4.4.4				2.5
143	143	26.2	146	125	11	143	26.2	29	all	80	15.5	143	146	14.9	13.52	3.1
144-146		26.3-TP	147-149				26.3									

Рисунок 1.1 Таблиця сумісності версій браузерів в різних ОС (Червоний – не підтримується додатком, Жовтий – підтримується з умовами, Зелений – повна підтримка)

Основними вимогами до технічного засобу є:

- Наявність браузера з підтримкою Javascript, список версій приведений на рисунку 1.1
- Стабільний доступ до мережі інтернет

2 ПІДГОТОВКА ДО ВИКОРИСТАННЯ

Носій даних з програмою містить керівництво користувача та текстовий файл формату txt з назвою *instruction.txt*, у якому наведено URL-адресу веб-додатку AI Photo Editor. Перед початком роботи користувачу рекомендується ознайомитися з даним керівництвом користувача.

Для початку роботи з веб-додатком необхідно відкрити підтримуваний веб-браузер та ввести URL-адресу застосунку в адресному або пошуковому рядку браузера. Також можливе копіювання URL-адреси з файлу *instruction.txt* з подальшим вставленням її в адресний рядок браузера.

3 ОПИС ЕКРАНІВ ТА ОПЕРАЦІЙ

3.1 Модуль додавання фото

Призначення: Модуль відображає інтерфейс для завантаження та відображення користувацького фото.

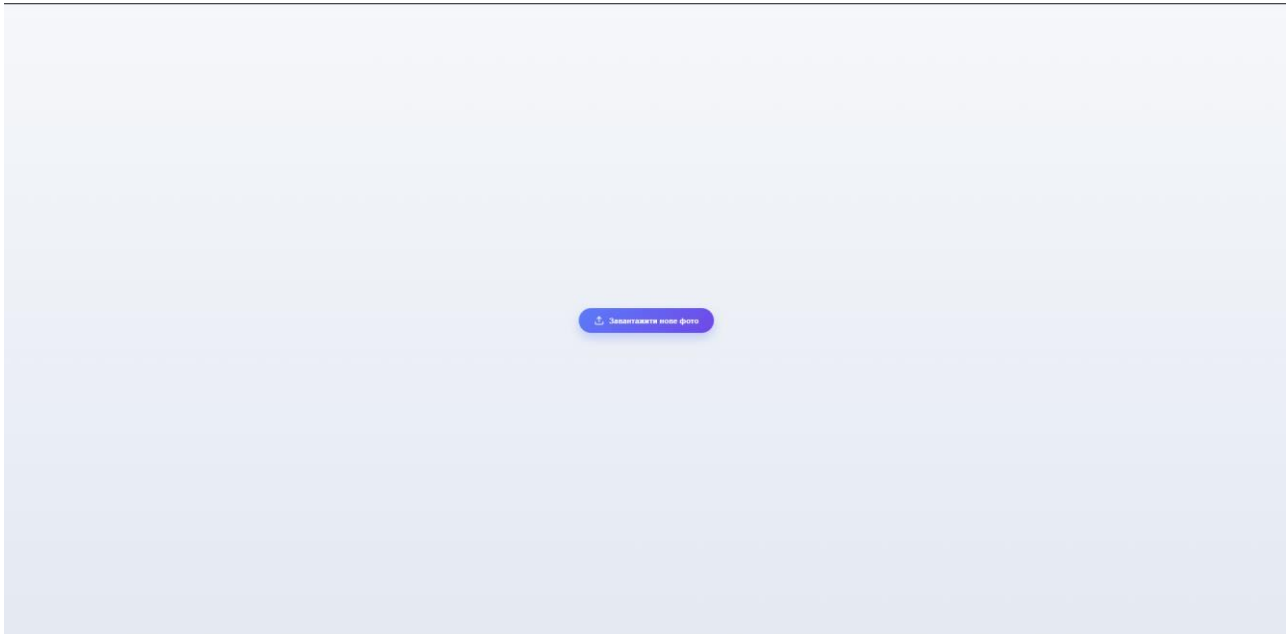


Рисунок 3.1 Скріншот додатка в початковому стані без завантаженого фото

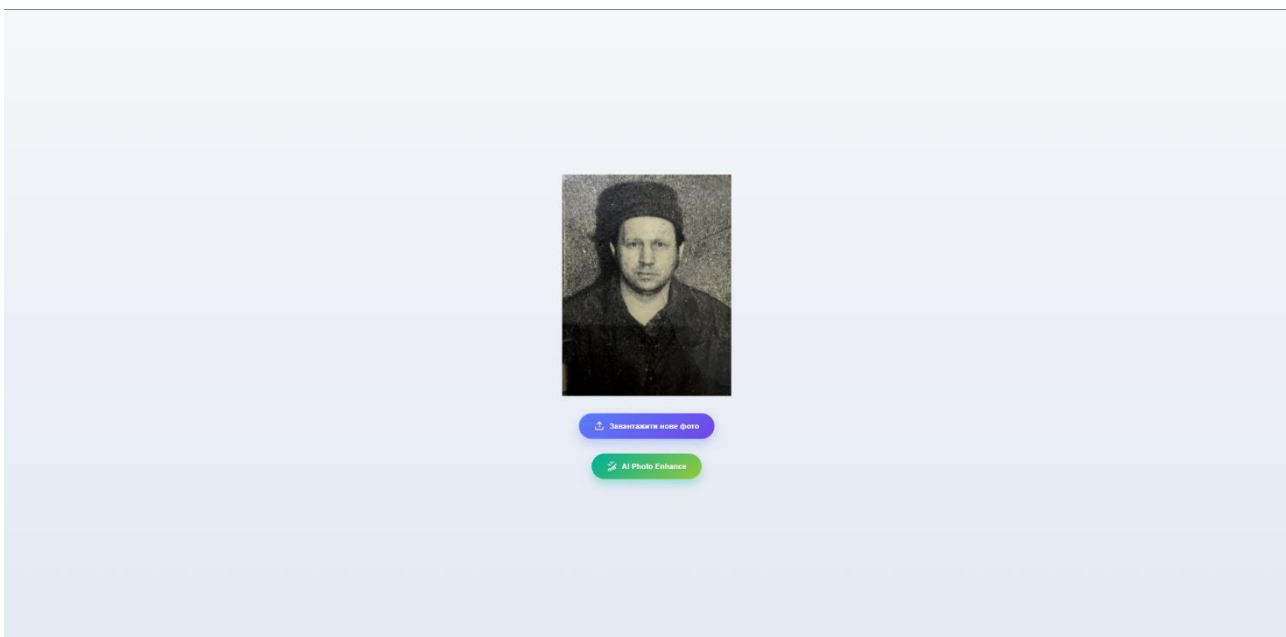


Рисунок 3.2 Скріншот додатка із завантаженим фото

Користувач (К) натискає на кнопку «завантажити нове фото», з'являється вікно вибору фотографій, К обирає потрібну фотографію, фотографія відображається в блоці фотографії та під кнопкою завантаження з'являється кнопка «AI Photo Enhance»

3.2 Модуль покращення фото

Призначення: Модуль, який виконує відправку завантаженого фото користувача на сервер для обробки та повертає оброблене фото у вигляді посилання

К натискає на кнопку «AI Photo Enhance», фотографія користувача передається в обробку та по завершенню записується в блок перегляду.

З'являється можливість порівняти обидві фотографії шляхом перетягування «ділителя»

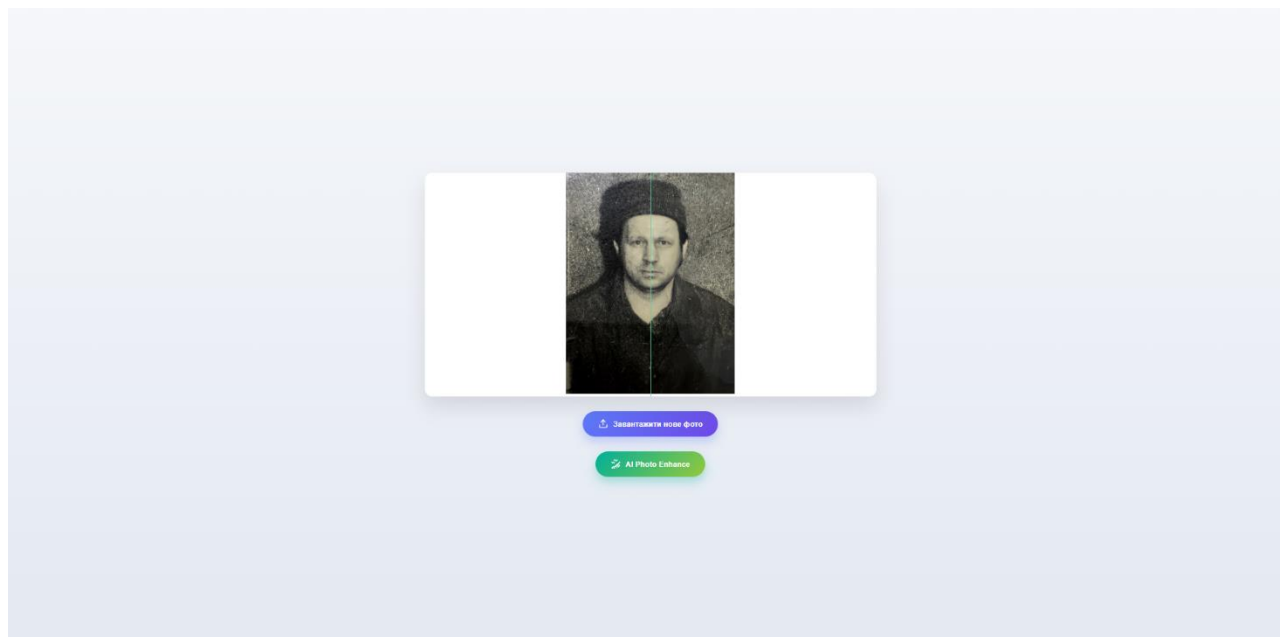


Рисунок 3.3 Скріншот додатка після обробки фотографії

ДОДАТОК Г



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
НАУКИ І ТЕХНОЛОГІЙ

ABSTRACTS
OF THE XIX INTERNATIONAL CONFERENCE
«MODERN INFORMATION AND COMMUNICATION
TECHNOLOGIES ON A TRANSPORT, IN INDUSTRY AND
EDUCATION»
18-19, December, 2025

СУЧАСНІ ІНФОРМАЦІЙНІ ТА
КОМУНІКАЦІЙНІ
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ,
В ПРОМИСЛОВОСТІ І ОСВІТІ

ПРИСВЯЧЕНО ПАМ'ЯТІ ПРОФЕСОРА ІГОРЯ ЖУКОВИЦЬКОГО

ТЕЗИ

XIX МІЖНАРОДНОЇ
НАУКОВО-
ПРАКТИЧНОЇ
КОНФЕРЕНЦІЇ
18-19 ГРУДНЯ 2025

ДНІПРО
2025

Чисельне моделювання нестационарних процесів геоміграції та теплопереноса	75
Біляев М. М., Козачина В. В., Матусевич М. О., Савченко Д. В., Український державний університет науки і технологій, Україна	
Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж	76
Рябовол В. К., Горячкін В. М., Український державний університет науки та технологій, Україна	
Дослідження нейромережевого розпізнавання образів за їх частинами із застосуванням онтологій.....	77
Жуковець О. О., Шинкаренко В. І., Український державний університет науки і технологій, Україна	
Використання штучних нейронних мереж для діагностування систем залізничної автоматики.....	78
Корда Б. О., Гаврилюк В. І., Український державний університет науки і технологій, Україна	
Супутниковий моніторинг поверхневих вод в умовах трансформованих ландшафтів.....	79
Кавац О.О., Кавац Ю.В., Фененко Т.М., Бровко О.Ю., Український державний університет науки і технологій, Україна	
Дослідження продуктивності роботи веб-додатків з різними форматами зображень	80
Посмітюха М.О., Гришечкіна Т.С., Український державний університет науки і технологій, Україна	
Огляд поширених архітектур для розпізнавання мовлення у реальному часі	81
Кравченко Т.О., Єгоров О.Й., Український державний університет науки і технологій, Україна	
Використання великих мовних моделей для обробки та структуризації великих масивів тексту	82
Борщенко В.О., Єгоров О.Й., Український державний університет науки і технологій, Україна	
Оптимізація швидкодії корпоративних веб-систем через впровадження гібридних сховищ та Read-Models	83
Гармаш О. О., Горячкін В. М., Український державний університет науки і технологій, Україна	
Analytical design of transport systems	84
Kravets V. V., Kapitsa M.I., Hryshechka T. S., Ukrainian State University of Science and Technologies, Ukraine Kravets T. V., Waghausel, Germany	
Визначення оптимального маршруту в комп'ютерній мережі залізничного транспорту за алгоритмом бактеріальної оптимізації.....	85
Пахомова В.М., Ланевич В.В., Український державний університет науки і технологій, Україна	

Дослідження методів автоматичного покращення якості зображень на основі нейронних мереж

Рябовол В. К., Горячкін В. М., Український державний університет науки та технологій, Україна

Автоматичне покращення якості зображень є однією з фундаментальних та найбільш затребуваних задач комп'ютерного зору. В умовах експоненційного зростання обсягів візуальної інформації, яка щоденно генерується та зберігається, виникає гостра потреба в розробці ефективних, швидких та високоточних алгоритмів її обробки. Ця технологія має критичне значення у таких галузях, як медична діагностика (підвищення чіткості знімків), системи безпеки та відеоспостереження, мультимедіа, цифрова реставрація історичних архівів, криміналістика.

Традиційні методи покращення якості, такі як фільтри (медіанні, гаусові) та алгоритми інтерполяції (білінійне, бікубічне масштабування), мають суттєві обмеження. Вони працюють лише з уже наявними пікселями та не здатні відновлювати інформацію, втрачену внаслідок знімання або стиснення, що призводить до розмиття, втрати текстур та деталей. На противагу цьому, сучасні підходи, засновані на використанні нейронних мереж, демонструють здатність не тільки до приглушення шумів та підвищення роздільної здатності, але й до генерації нової візуальної інформації та реконструкції складних текстур і дрібних деталей на основі статистичних знань, отриманих під час навчання.

Метою даної роботи є дослідження, систематизація та порівняльний аналіз ключових нейромережових методів, що застосовуються для автоматичного покращення якості зображень, а також оцінка їхнього потенціалу та практичних викликів при інтеграції у реальні системи.

Для реалізації завдань покращення якості зображень застосовуються такі поширені методи, як denoising (приглушення шумів), super-resolution (надвисока роздільна здатність), inpainting (заповнення відсутніх областей) та відновлення контурів.

Для реалізації цих методів використовуються три основні класи нейромережових архітектур: згорткові нейронні мережі (CNN), генеративно-змагальні мережі (GAN) та дифузійні моделі

Процес обробки зображень нейронними мережами є багатетапним. Він включає: підготовку та нормалізацію навчальних і тестових даних; формування тензорів для ефективної роботи на GPU; безпосередньо нейромережову обробку (інференс); та постобробку отриманого результату з подальшою інтеграцією в кінцеве зображення.

Попри значні переваги, використання нейронних мереж у цій галузі пов'язане з низкою суттєвих обмежень та проблем, таких як недостатність обчислювальних ресурсів, ризики появи штучних артефактів і спотворення реальних даних через перенавчання або неякісний інференс, залежність якості результату від репрезентативності та обсягу навчальних вибірок.

Подальший розвиток нейромережових методів автоматичного покращення якості зображень спрямований на вирішення цих проблем. Ключові напрямки включають підвищення точності та об'єктивних метрик якості (PSNR, SSIM), мінімізацію обчислювальних витрат для використання на мобільних і периферійних пристроях, підвищення стійкості моделей до різних типів шумів та створення універсальних, архітектурно-адаптованих моделей, придатних для обробки зображень різних типів без перенавчання. Впровадження цих рішень має забезпечити нові можливості для застосування у системах відеоспостереження та цифровій реставрації.