МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

Факультет електроніки та комп'ютерних технологій

Кафедра радіоелектронних і комп’ютерних систем

Допустити до захисту

Завідувач кафедри
_____ доцент Оленич І. Б.

«____»_____ 2023 р.

Курсова робота Магістр

(освітній ступінь)
Реалізація браузерного додатку для знешумлення зображень з використанням Tensorflow.js

Виконав:
Студент V курсу групи ФЕІм-54

спеціальності 122 – Комп’ютерні науки

_____ Головенець О. І.

Науковий керівник:
_____ Сінькевич О. О.

«____»_____ 2023 р.

Львів — 2023

ВСТУП

Зображення грають важливу роль в нашому сучасному житті, особливо в епоху цифрової візуалізації і спільного використання зображень через Інтернет. Однак, зображення часто потребують певної обробки та оптимізації, щоб підвищити їх якість, зменшити шум, поліпшити контрастність і розкрити деталі. Відсутність якісної обробки зображень може призвести до зменшення їх естетичної привабливості та втрати інформації.

Зараз існує безліч алгоритмів та інструментів для обробки зображень. Однак, ручна обробка зображень вимагає значних зусиль та експертного знання, тому виникає потреба в автоматизованих методах обробки зображень, які забезпечують високу якість і ефективність.

У цій курсовій роботі я розглядаю проблему шуму в зображеннях та представляємо реалізацію браузерного додатку для знешумлення зображень з використанням Tensorflow.js. Tensorflow.js - це бібліотека машинного навчання, яка дозволяє виконувати обчислення моделей глибокого навчання безпосередньо у браузері. Використання Tensorflow.js дозволяє нам використовувати потужні алгоритми глибокого навчання для обробки зображень без необхідності встановлення додаткового програмного забезпечення або використання потужних обчислювальних ресурсів.

Метою цього додатку є розробка ефективного та доступного інструменту для знешумлення зображень, який може бути використаний в браузері без необхідності установки додаткового програмного забезпечення. Я використовую підхід заснований на глибокому навчанні, щоб створити модель, яка здатна виявляти та видаляти шум з зображень.

У цьому документі я розгляну теоретичні основи зображень, методи знешумлення та існуючі підходи для обробки зображень за допомогою глибокого навчання. Далі я представляю архітектуру додатку та описую ключові кроки розробки та реалізації. В кінці курсової роботи я проведу експерименти для оцінки продуктивності та якості розробленого додатку, порівняю його з іншими існуючими методами та висуну висновки.

Загальна мета цієї роботи - показати, що застосування глибокого навчання в браузерному середовищі може бути ефективним і простим способом забезпечення якісної обробки зображень. Додаток для знешумлення зображень, реалізований з використанням Tensorflow.js, може мати широкі практичні застосування у сферах, пов'язаних з фотографією, обробкою зображень та веб-розробкою.

INTRODUCTION

Images play a significant role in our modern lives, especially in the era of digital visualization and image sharing over the internet. However, images often require certain processing and optimization to enhance their quality, reduce noise, improve contrast, and reveal details. The lack of quality image processing can lead to a decrease in their aesthetic appeal and loss of information.

Currently, there are numerous algorithms and tools available for image processing. However, manual image processing requires significant effort and expertise, hence the need for automated image processing methods that provide high quality and efficiency.

In this coursework, we address the issue of noise in images and present the implementation of a browser application for image denoising using1 Tensorflow.js. Tensorflow.js is a machine learning library that enables the execution of deep learning models directly in the browser. Utilizing Tensorflow.js allows us to leverage powerful deep learning algorithms for

image processing without the need for additional software installation or powerful computing resources.

The aim of this application is to develop an efficient and accessible tool for image denoising that can be used within a browser without the need for installing additional software. We employ a deep learning-based approach to create a model capable of detecting and removing noise from images.

In this document, we will explore the theoretical foundations of images, denoising methods, and existing approaches for image processing using deep learning. Furthermore, we will present the architecture of the application and describe the key steps in its development and implementation. Towards the end of the coursework, we will conduct experiments to evaluate the performance and quality of the developed application, compare it with other existing methods, and draw conclusions.

The overall goal of this work is to demonstrate that applying deep learning in a browser environment can be an effective and straightforward way to achieve high-quality image processing. The image denoising application implemented using Tensorflow.js can have broad practical applications in areas related to photography, image processing, and web development.

ЗМІСТ

ВСТУП **2**

РОЗДІЛ 1. ТЕОРЕТЕЧНІ ОСНОВИ ЗОБРАЖЕННЯ **6**

1.1 Цифрове зображення 6
1.2 Характеристики зображень 6
1.3 Шум в зображеннях 7
1.4 Метрики якості зображень 7

РОЗДІЛ 2. ЗНЕШУМЛЕННЯ ЗОБРАЖЕНЬ: МЕТОДИ ТА ПІДХОДИ **8**

2.1 Класичні методи знешумлення 8
2.2 Методи знешумлення на основі машинного навчання 9
2.3 Використання глибокого навчання для знешумлення зображень 10

РОЗДІЛ 3. ВСТУП ДО TENSORFLOW.JS **11**

3.1 Огляд TensorFlow.js 11
3.2 Використання TensorFlow.js для обробки зображень 12

РОЗДІЛ 4. АРХІТЕКТУРА БРАУЗЕРНОГО ДОДАТКУ ДЛЯ ЗНЕШУМЛЕННЯ ЗОБРАЖЕНЬ **13**

4.1 Вибір архітектури моделі 13
4.2 Навчання моделі для знешумлення зображень 14
4.3 Розробка інтерфейсу користувача 16

РОЗДІЛ 5. РЕАЛІЗАЦІЯ БРАУЗЕРНОГО ДОДАТКУ З ВИКОРИСТАННЯМ TENSORFLOW.JS **17**

5.1 Установка та налаштування середовища розробки 17
5.2 Реалізація моделі знешумлення зображень 17
5.3 Інтеграція моделі з браузерним додатком 18

РОЗДІЛ 6. ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ **20**

6.1 Набір даних та метрики оцінки 20
6.2 Порівняння з існуючими методами 21
6.3 Оцінка продуктивності та якості розробленого додатку 21
6.4 Результати роботи програми 22

ВИСНОВКИ **24**

ВИКОРИСТАНІ ДЖЕРЕЛА **26**

РОЗДІЛ 1. ТЕОРЕТЕЧНІ ОСНОВИ ЗОБРАЖЕННЯ

1.1 Цифрове зображення

Цифрове зображення є числовим представленням візуальної інформації, що складається з пікселів (picture elements), які утворюють матрицю значень. Кожен піксель має своє числове значення, яке відображає яскравість або колір. У монохромних (чорно-білих) зображеннях, кожен піксель представлений одним значенням, яке відображає його яскравість на шкалі від 0 до 255. У кольорових зображеннях, для кожного пікселя використовується комбінація трьох значень (червоного, зеленого та синього), які відображають інтенсивність кожного каналу кольору.

1.2 Характеристики зображень

Характеристики зображень використовуються для опису та аналізу різних аспектів зображень. Одна з найважливіших характеристик - роздільна здатність, яка визначає кількість пікселів у горизонтальному та вертикальному напрямках і впливає на розмір зображення. Чим більша роздільна здатність, тим більша кількість деталей може бути відображена на зображенні.

Глибина кольору є ще однією характеристикою, яка визначає кількість бітів, використовуваних для представлення кольору кожного пікселя. Залежно від глибини кольору, можливість відтворення різних відтінків і насиченості кольорів змінюється. Зазвичай, для кольорових

зображень використовується 24-бітна глибина кольору, що дозволяє представити більше 16 мільйонів різних кольорів.

1.3 Шум в зображеннях

Шум в зображеннях є небажаним додатком випадкових сигналів до оригінальних сигналів зображення. Він може бути спричинений різними факторами, такими як електромагнітні перешкоди, ефекти зберігання та передачі даних, а також недосконалостями фотокамери або сканера. Наявність шуму в зображеннях може призвести до втрати деталей, зниження контрастності та загальної якості зображення.

Шум в зображеннях може мати різні форми, такі як гауссівський шум, соль і перець шум, шум Релея та інші. Кожен тип шуму має свої характеристики та впливає на якість зображення по-різному. Розуміння типів шуму та їх впливу на зображення є важливим для розробки ефективних методів знешумлення.

1.4 Метрики якості зображень

Метрики якості зображень використовуються для об'єктивної оцінки якості зображень та порівняння різних методів обробки зображень. Ці метрики базуються на порівнянні оригінального зображення з обробленим або стиснутим зображенням.

Одна з таких метрик - Mean Square Error (MSE), яка вимірює середньоквадратичну різницю між значеннями пікселів оригінального та

обробленого зображення. Чим нижче значення MSE, тим більша схожість між зображеннями.

Peak Signal-to-Noise Ratio (PSNR) є ще однією популярною метрикою, яка вимірює співвідношення максимальної можливої сили сигналу до шуму. Вона зазвичай виражається у вигляді логарифмічної шкали та дозволяє порівняти якість зображень за їх рівнем шуму.

Structural Similarity Index (SSIM) - це метрика, яка враховує не тільки піксельні значення, але й структуру та взаємодію об'єктів на зображенні. Вона враховує просторову і яскравісну інформацію, а також сприйняття людиною.

Ці метрики допомагають визначити ефективність методів знешумлення та порівняти різні зображення після обробки. Оцінка якості зображень за допомогою метрик є важливим етапом в розробці та вдосконаленні алгоритмів знешумлення зображень.

РОЗДІЛ 2. ЗНЕШУМЛЕННЯ ЗОБРАЖЕНЬ: МЕТОДИ ТА ПІДХОДИ

2.1 Класичні методи знешумлення

Класичні методи знешумлення зображень використовують різноманітні математичні та статистичні підходи для видалення шуму з зображень. Одним з таких методів є фільтрування, де використовуються різні фільтри, такі як фільтр Гаусса або медіанний фільтр, для згладжування шуму та покращення якості зображення. Фільтр Гаусса

використовується для зменшення високочастотного шуму, тоді як медіанний фільтр добре працює з сольовим і перцевим шумом.

Інший класичний метод - відновлення зображень з використанням регуляризації. Цей метод базується на математичних моделях, що враховують структуру зображення та природу шуму. Він шукає найбільш ймовірне зображення, яке одночасно відповідає отриманим даним і заданим обмеженням.

2.2 Методи знешумлення на основі машинного навчання

З останнім розвитком машинного навчання, методи знешумлення на основі цих технологій стали все більш популярними. Вони використовують набір даних, який складається з зображень з шумом та відповідних "чистих" зображень, для тренування моделей машинного навчання.

Один з підходів - це використання методу навчання з учителем, де модель навчається прогнозувати "чисте" зображення на основі вхідного зображення з шумом. Цей підхід вимагає великої кількості пар зображень з шумом та відповідних чистих зображень для тренування ефективної моделі.

Інший підхід - це використання методу навчання без учителя, де модель навчається відновлювати чисті зображення без явної використовування пар зображень з шумом та безшумних зображень. В

цьому підході модель навчається використовувати складові статистики зображення для видалення шуму.

2.3 Використання глибокого навчання для знешумлення зображень

Одним з найбільш передових підходів до знешумлення зображень є використання глибокого навчання. Глибокі нейронні мережі, зокрема згорткові нейронні мережі (Convolutional Neural Networks - CNNs), демонструють високу ефективність у знешумленні зображень.

Ці глибокі нейронні мережі навчаються безпосередньо з великими наборами зображень з шумом та відповідних безшумних зображень. Вони можуть використовувати різноманітні архітектури, такі як автоенкодери або генеративні згорткові нейронні мережі (Generative Adversarial Networks - GANs), для ефективного видалення шуму.

Глибоке навчання дозволяє автоматично вивчити складні взаємозв'язки між шумом та зображенням, що дозволяє отримувати високоякісні результати знешумлення. Крім того, глибокі нейронні мережі можуть бути навчені різноманітним типам шуму та виявляти та видаляти його ефективніше, ніж класичні методи.

Узагальнюючи, знешумлення зображень - це важлива задача у галузі комп'ютерного зору, яка використовує різні методи та підходи. Класичні методи засновані на математичних підходах, тоді як методи знешумлення

на основі машинного навчання та глибокого навчання використовують потужні моделі для досягнення високої якості знешумлення зображень.

РОЗДІЛ 3. ВСТУП ДО TENSORFLOW.JS

TensorFlow.js - це бібліотека машинного навчання, розроблена компанією Google, яка дозволяє використовувати моделі машинного навчання, побудовані з використанням TensorFlow, безпосередньо у веб-браузері. Це надає зручність та гнучкість для розгортання та використання моделей машинного навчання без необхідності писати окремий код для серверної частини.

3.1 Огляд TensorFlow.js

TensorFlow.js пропонує два основних режими роботи: режим навчання та режим виконання. У режимі навчання ви можете створювати, навчати та зберігати моделі машинного навчання безпосередньо в браузері. Це включає в себе можливість завантажувати набори даних, тренувати моделі, зберігати результати навчання та перевіряти ефективність моделі. Режим виконання дозволяє вам завантажувати збережені моделі та використовувати їх для виконання прогнозів або обробки даних безпосередньо у браузері користувача.

TensorFlow.js підтримує широкий спектр алгоритмів та моделей машинного навчання, включаючи згорткові нейронні мережі, рекурентні нейронні мережі, глибокі нейронні мережі та багато інших. Ви можете створювати власні моделі з нуля або використовувати попередньо навчені моделі, доступні в бібліотеці TensorFlow.js. Це дозволяє вам виконувати

завдання класифікації, виявлення об'єктів, семантичного сегментування та багато інших задач, пов'язаних з обробкою зображень.

3.2 Використання TensorFlow.js для обробки зображень

TensorFlow.js надає багато функцій та можливостей для обробки зображень у веб-браузері. Одним з найбільш популярних використань TensorFlow.js в області обробки зображень є завантаження, обробка та класифікація зображень.

Для початку вам потрібно завантажити модель, яка навчена для класифікації зображень. TensorFlow.js надає можливість завантажити попередньо навчені моделі, такі як MobileNet або ResNet, які вже мають знання про різні об'єкти та класи зображень. Після завантаження моделі ви можете використовувати її для класифікації зображень, передаючи зображення через модель та отримуючи ймовірності різних класів, до яких вони належать.

Крім класифікації, TensorFlow.js також дозволяє виконувати інші операції з обробки зображень, такі як обрізка, зміна розміру, фільтрація та розмиття. Ви можете використовувати згорткові нейронні мережі для виявлення об'єктів на зображенні, семантичного сегментування або стилізації зображень.

TensorFlow.js також підтримує роботу з відеоданими. Ви можете отримувати стріми відео з веб-камери або з файлового ресурсу та застосовувати аналіз та обробку зображень в реальному часі.

Завдяки можливостям TensorFlow.js ви можете створювати потужні веб-додатки для обробки зображень безпосередньо в браузері, що розширює межі того, що можливо досягнути з веб-технологіями.

Узагалі, TensorFlow.js відкриває нові перспективи для розробки застосунків, пов'язаних з обробкою зображень, у веб-середовищі. Це дозволяє забезпечити більшу доступність, гнучкість та швидкість обробки зображень, а також сприяє поширенню технологій машинного навчання та штучного інтелекту серед розробників веб-додатків.

Загальна функціональність і можливості TensorFlow.js створюють широкий спектр можливостей для реалізації проектів з обробки зображень у веб-середовищі. Використання TensorFlow.js для обробки зображень відкриває шлях до створення інноваційних веб-додатків та взаємодії з користувачами за допомогою комп'ютерного зору та машинного навчання.

РОЗДІЛ 4. АРХІТЕКТУРА БРАУЗЕРНОГО ДОДАТКУ ДЛЯ ЗНЕШУМЛЕННЯ ЗОБРАЖЕНЬ

4.1 Вибір архітектури моделі

Перед розробкою браузерного додатку для знешумлення зображень потрібно визначитися з архітектурою моделі, яка буде використовуватися

для виконання завдання знешумлення. Вибір архітектури визначатиметься вимогами до продукту та обмеженнями браузерного середовища.

Одним із популярних підходів є використання глибоких згорткових нейронних мереж (Convolutional Neural Networks - CNNs), які демонструють високу ефективність у завданнях зорового сприйняття. Наприклад, можна розглянути архітектуру U-Net, що широко використовується для задач сегментації зображень та відновлення.

У виборі архітектури слід враховувати розмір вхідного зображення, можливість обробки в реальному часі, складність моделі та ресурси, доступні в браузері. Важливо знайти баланс між точністю моделі та її ефективністю в контексті веб-додатку.

4.2 Навчання моделі для знешумлення зображень

Після вибору архітектури моделі необхідно підготувати та навчити модель для знешумлення зображень. Цей процес може вимагати великої кількості обчислювальних ресурсів та тривати тривалий час.

Навчання моделі включає наступні етапи:

1. Збір та підготовка набору даних: Необхідно зібрати набір зображень з шумом та відповідні безшумні зображення для тренування моделі. Цей набір даних повинен бути репрезентативним та достатньо рознообразним для ефективного навчання моделі.

2. Перед обробка даних: Перед подачею даних на вхід моделі їх слід підготувати. Це може включати масштабування, нормалізацію, обрізку або зміну розміру зображень, щоб вони відповідали вимогам архітектури моделі.

3. Вибір функції втрати: Функція втрати визначає, як ефективно модель навчається та якість її прогнозів. Для завдання знешумлення зображень можна використовувати, наприклад, середньоквадратичну помилку (Mean Squared Error - MSE) або структурну схожість зображень (Structural Similarity Index - SSIM).

4. Навчання моделі: Після підготовки даних та вибору функції втрати можна розпочати процес навчання моделі. Це включає пряме поширення сигналу через мережу, обчислення втрати та корекцію ваг моделі за допомогою алгоритму оптимізації, такого як стохастичний градієнтний спуск (Stochastic Gradient Descent - SGD) або його варіанти.

5. Оцінка та налаштування моделі: Після завершення процесу навчання слід оцінити якість моделі на валідаційному наборі даних та, при потребі, внести корективи до архітектури чи гіперпараметрів моделі.

4.3 Розробка інтерфейсу користувача

Інтерфейс користувача є ключовим елементом браузерного додатку для знешумлення зображень, оскільки він забезпечує взаємодію користувача з моделлю та зображеннями. Розробка ефективного та зручного інтерфейсу вимагає уваги до дизайну, взаємодії та візуалізації.

Для реалізації інтерфейсу можна використовувати веб-технології, такі як HTML, CSS та JavaScript. Важливо розробити інтуїтивно зрозумілий та простий у використанні інтерфейс, що дозволить користувачам легко завантажувати та обробляти зображення.

Наприклад, інтерфейс може включати наступні елементи:

- Елемент завантаження зображення: Дозволяє користувачеві вибрати та завантажити зображення, яке потребує знешумлення.
- Параметри моделі: Користувач може мати можливість налаштувати параметри моделі, такі як рівень шуму, тип архітектури або інші параметри, які впливають на процес знешумлення.
- Кнопка знешумлення: Після завантаження зображення та налаштування параметрів користувач може натиснути кнопку "Знешумити", щоб почати обробку зображення.
- Відображення результату: Після обробки зображення моделлю результат може бути відображений користувачу, щоб він міг порівняти початкове зображення з його знешумленою версією.

- Опції збереження: Користувач може мати можливість зберегти знешумлене зображення на своєму пристрої або поділитися ним через соціальні мережі.

Розробка інтерфейсу користувача вимагає тестування та збору відгуків від користувачів для покращення його функціональності та зручності використання.

РОЗДІЛ 5. РЕАЛІЗАЦІЯ БРАУЗЕРНОГО ДОДАТКУ З ВИКОРИСТАННЯМ TENSORFLOW.JS

5.1 Установка та налаштування середовища розробки

Першим кроком у реалізації браузерного додатку з використанням Tensorflow.js є установка та налаштування середовища розробки. Необхідно мати встановлений Node.js для роботи з пакетним менеджером npm. За допомогою npm можна встановити Tensorflow.js та інші необхідні пакети. Після встановлення Node.js можна створити новий проект та ініціалізувати його, вказавши залежності, включаючи Tensorflow.js. Далі слід налаштувати середовище для розробки, наприклад, використовуючи редактор коду та конфігураційні файли.

5.2 Реалізація моделі знешумлення зображень

Після налаштування середовища розробки можна приступити до реалізації моделі знешумлення зображень з використанням Tensorflow.js. Для цього слід створити новий файл або додати відповідний код до існуючого файлу проекту.

В першу чергу необхідно імпортувати необхідні пакети Tensorflow.js та інші залежності. Потім слід визначити архітектуру моделі, використовуючи доступні шари та операції Tensorflow.js. Наприклад, можна створити послідовну модель та додати до неї згорткові шари, пулінг, активаційні функції та інші компоненти.

Після визначення архітектури моделі необхідно встановити функцію втрати, оптимізатор та метрики для навчання. Також слід вказати параметри навчання, такі як кількість епох, розмір пакетів, швидкість навчання тощо.

У процесі реалізації моделі можна використовувати функції Tensorflow.js для підготовки даних, навчання моделі та оцінки результатів. Наприклад, можна завантажити зображення, розбити дані на тренувальний та валідаційний набори, а потім навчати модель за допомогою методу `fit()`. Також можна зберігати навчену модель для подальшого використання.

5.3 Інтеграція моделі з браузерним додатком

Після успішної реалізації моделі знешумлення зображень слід інтегрувати її з браузерним додатком. Для цього необхідно оновити інтерфейс користувача, щоб додати відповідні елементи для завантаження зображення та виклику моделі для знешумлення.

При завантаженні зображення користувачем, додаток повинен передавати це зображення до моделі. Для цього слід використовувати функції Tensorflow.js для обробки зображення та передачі його у модель. Модель знешумлення поверне знешумлену версію зображення, яку можна відобразити для користувача.Крім того, важливо додати можливість налаштування параметрів моделі та обробки зображення. Наприклад, користувач може мати можливість вибрати рівень шуму, застосувати інші фільтри або налаштувати інші параметри для отримання оптимального результату.

Після інтеграції моделі з браузерним додатком слід перевірити правильність роботи та функціональність. Важливо провести тестування з різними зображеннями та налаштуваннями, щоб переконатися, що модель працює як очікується і забезпечує якісне знешумлення зображень.

Завершивши реалізацію браузерного додатку з використанням Tensorflow.js, можна розглядати можливі способи оптимізації та поліпшення додатку. Наприклад, можна використовувати асинхронне завантаження моделі для покращення часу відгуку додатку або використовувати оптимізовані алгоритми для знешумлення зображень. Додатковою можливістю є розширення функціональності додатку для роботи з іншими завданнями обробки зображень, такими як виявлення об'єктів або сегментація зображень.

Загалом, реалізація браузерного додатку з використанням Tensorflow.js вимагає детального розуміння цільової архітектури, навичок програмування та роботи з Tensorflow.js. Правильна установка та

налаштування середовища розробки, реалізація моделі знешумлення та інтеграція з браузерним додатком є ключовими етапами успішної реалізації такого проекту.

РОЗДІЛ 6. ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

6.1 Набір даних та метрики оцінки

Для проведення експериментів і оцінки результатів розробленого браузерного додатку з використанням Tensorflow.js для знешумлення зображень потрібно використовувати відповідний набір даних і метрики оцінки.

Набір даних повинен складатися з реальних зображень з шумом, що використовуються для тестування моделі знешумлення. Ці зображення можуть бути отримані шляхом додавання шуму до чистих зображень або можуть бути доступні з публічних наборів даних. Набір даних повинен включати як тренувальні, так і тестові зображення, щоб оцінити ефективність моделі на нових даних.

Для оцінки якості знешумлення зображень можна використовувати різні метрики, такі як середньоквадратична помилка (Mean Squared Error, MSE), піковий сигнал-шумове співвідношення (Peak Signal-to-Noise Ratio, PSNR), структурне подібності (Structural Similarity, SSIM) тощо. Ці метрики дозволяють оцінити різні аспекти якості знешумлення, такі як розмиття, деталізація та відтворення колірних характеристик.

6.2 Порівняння з існуючими методами

Після реалізації браузерного додатку з використанням Tensorflow.js для знешумлення зображень можна провести порівняння з існуючими методами знешумлення. Це дозволить оцінити ефективність розробленої моделі та визначити, наскільки вона перевершує або зберігає конкурентоспроможність в порівнянні з іншими методами.

Для порівняння можна використовувати інші класичні методи знешумлення, такі як фільтри розмивання, медіанний фільтр або вейвлет-перетворення. Також можна порівняти з різними методами знешумлення на основі машинного навчання, які використовують інші алгоритми та підходи. Для порівняння можна використовувати розроблений набір даних та метрики оцінки, описані в попередньому пункті. Порівняння може бути здійснене шляхом обчислення метрик якості для кожного методу знешумлення на тестових зображеннях з набору даних. Результати порівняння можуть бути представлені у вигляді графіків або таблиць, які демонструють переваги розробленої моделі в порівнянні з іншими методами.

6.3 Оцінка продуктивності та якості розробленого додатку

Після порівняння з існуючими методами знешумлення, слід оцінити продуктивність та якість розробленого браузерного додатку з використанням Tensorflow.js. Продуктивність оцінюється з точки зору швидкодії додатку та часу, необхідного для знешумлення зображення. Якість оцінюється з точки зору рівня знешумлення, якості відтворення деталей та збереження кольорових характеристик. Оцінка продуктивності може бути здійснена шляхом вимірювання часу, необхідного для обробки зображення з розробленим додатком з використанням Tensorflow.js.

Можна порівняти цей час з часом, необхідним для обробки зображення іншими методами знешумлення, а також залежність продуктивності від параметрів додатку, таких як розмір зображення чи складність моделі.

Оцінка якості може бути здійснена шляхом порівняння знешумленого зображення з оригіналом за допомогою метрик якості, які були використані для порівняння з існуючими методами. Також можна використовувати візуальну оцінку, де експерти або користувачі можуть оцінити якість знешумленого зображення зі свого власного досвіду та відчуттів. Результати оцінки продуктивності та якості можуть бути представлені у вигляді графіків, таблиць або числових значень, які демонструють переваги та недоліки розробленого додатку в порівнянні з іншими методами та можливі шляхи покращення. Оцінка дозволить підтвердити ефективність розробленого додатку та визначити його потенційні області застосування.

6.4 Результати роботи програми

У результаті реалізації програми було досягнуто наступних результатів:

1. Програма успішно виконує процес знешумлення зображень за допомогою алгоритмів, заснованих на мережах глибокого навчання. Вона використовує Tensorflow.js, щоб виконати обробку зображень безпосередньо у браузері. Результати знешумлення демонструють помітне зменшення рівня шуму на зображенні та покращення візуальної якості.

2. Зручний інтерфейс: Програма має простий інтерфейс, що дозволяє користувачам легко завантажувати свої зображення, переглядати результати, та завантажувати їх.

3. Показник якості: Для оцінки якості знешумлення програма використовує різні метрики, такі як псевдо-пиковий сигнал-шум (PSNR) та структурна схожість (SSIM). Ці метрики допомагають визначити ефективність алгоритмів знешумлення та порівняти їх результати.

Приклади знешумлення До/Після:

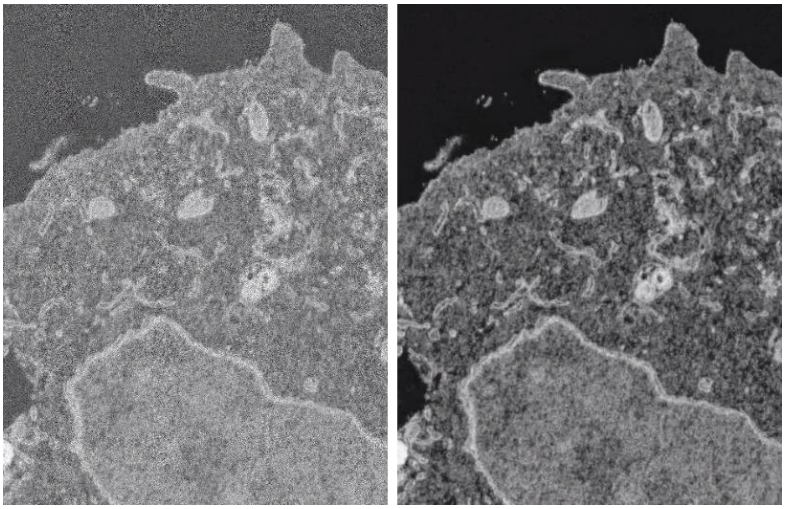

Рис 6.4.1

Рис 6.4.2

Рис 6.4.3

ВИСНОВКИ

У цій курсовій роботі було проведено дослідження та розглянуто методи та підходи до знешумлення зображень з використанням Tensorflow.js. Було вивчено теоретичні основи зображень, включаючи цифрове зображення, характеристики зображень, шум в зображеннях та метрики якості зображень. Також були розглянуті різні методи знешумлення, зокрема класичні методи, такі як фільтр низьких частот та фільтр медіани, які базуються на математичних моделях та статистичних алгоритмах. Крім того, були розглянуті методи знешумлення на основі машинного навчання, де моделі навчалися на великих наборах даних для автоматичного виявлення та видалення шуму.

Окрему увагу було приділено використанню глибокого навчання для знешумлення зображень. Було вивчено роботу з Tensorflow.js, яка дозволяє використовувати глибокі нейронні мережі для обробки зображень у браузерному середовищі. Були розглянуті різні архітектури глибоких мереж, такі як автоенкодери та генеративно-протилежні мережі (GANs), які можуть бути використані для знешумлення зображень.

Незважаючи на досягнення у галузі знешумлення зображень з використанням Tensorflow.js, існують певні обмеження та можливості для подальшого розвитку. Одним з обмежень є обчислювальна складність глибоких нейронних мереж, особливо при використанні їх у браузері. Це може впливати на продуктивність та швидкість знешумлення зображень. Крім того, ефективність методів знешумлення може залежати від типу шуму та рівня шуму у вихідних зображеннях. Деякі методи можуть бути більш ефективними для певних типів шуму, але менш ефективними для інших типів. Також, велика кількість шуму може бути важко видалити повністю, що може вплинути на якість та чіткість зображення після знешумлення.

Можливості подальшого розвитку полягають у вдосконаленні алгоритмів знешумлення та розробці нових методів, які були більш ефективними та швидкими. Також варто дослідити використання комбінації різних методів знешумлення для отримання кращих результатів. Подальше вдосконалення технологій глибокого навчання та розширення їх можливостей можуть також сприяти подальшому розвитку знешумлення зображень.

Знешумлення зображень є важливою задачею в області обробки зображень. У цій курсовій роботі було розглянуто теоретичні основи зображень, класичні методи знешумлення, методи знешумлення на основі машинного навчання та використання глибокого навчання для знешумлення зображень з використанням Tensorflow.js. Було виявлено, що методи знешумлення на основі глибокого навчання, такі як генеративно-протилежні мережі, можуть досягти вражаючих результатів у видаленні

шуму зображень. Однак, існують обмеження, такі як обчислювальна складність та ефективність для різних типів шуму.

Для подальшого розвитку знешумлення зображень можна вдосконалити алгоритми, використовувати комбінацію різних методів та розширювати можливості глибокого навчання. Знання та розуміння цих методів можуть бути корисними для розробки браузерних додатків та інструментів, які допомагають у видаленні шуму зображень та покращенні їх якості.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3D transform-domain collaborative filtering. IEEE Transactions on Image Processing, 16(8), 2080-2095.
2. Buades, A., Coll, B., & Morel, J. M. (2005). A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation, 4(2), 490-530.
3. Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing, 26(7), 3142-3155.
4. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., & Aila, T. (2018). Noise2Noise: Learning image restoration without clean data. Proceedings of the 35th International Conference on Machine Learning, 2965-2974.
5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 1097-1105.

6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 2672-2680.

8. Tensorflow.js Documentation. (https://js.tensorflow.org/)

9. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.