

Чорнобривець Д. В., здобувач вищої освіти,
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0009-0006-3647-9318

Поперешняк С. В., кандидат фізико-математичних наук,
доцент, доцент кафедри інформатики та програмної інженерії
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0000-0002-0531-9809

ІНТЕРАКТИВНА СИСТЕМА СЕМАНТИЧНОГО РЕДАГУВАННЯ РАСТРОВОЇ ГРАФІКИ НА ОСНОВІ ІНТЕГРАЦІЇ МУЛЬТИМОДАЛЬНИХ ГЕНЕРАТИВНИХ АРІ

У роботі досліджується процес автоматизованого семантичного редагування растрових зображень із використанням методів штучного інтелекту. Актуальність дослідження зумовлена високою ресурсоемністю сучасних генеративних моделей, що потребують потужних графічних процесорів для виконання операцій *Inpainting*, а також обмеженою гнучкістю хмарних сервісів у частині точного просторового керування редагуванням. Це створює проблему доступності інтелектуальних інструментів редагування для користувачів із базовими обчислювальними ресурсами.

Метою роботи є підвищення доступності та ефективності процесів семантичного редагування зображень за рахунок розподілу обчислювального навантаження між клієнтською та серверною частинами системи. У роботі запропоновано підхід, що поєднує локальні інструменти формування просторових масок із використанням хмарного мультимодального АРІ (*Gemini 3 Flash Image*) для виконання генеративних перетворень.

У результаті дослідження спроектовано та реалізовано легковаговий десктопний застосунок із модульною клієнт-серверною архітектурою. Ключовими особливостями системи є використання асинхронної багатопотокової обробки мережеских запитів, що забезпечує відсутність блокування графічного інтерфейсу, а також застосування алгоритмів зворотного композитингу для коректного інтегрування згенерованих фрагментів у вихідне зображення. Реалізовано механізм формування бінарних масок у реальному часі на основі координат курсору, що дозволяє забезпечити високу точність виділення зон інтересу.

Отримані результати пояснюються ефективним перенесенням тензорних обчислень на хмарну інфраструктуру при збереженні локального контролю над процесом редагування. Експериментальні дослідження підтвердили можливість виконання складних графічних модифікацій на пристроях із низькою обчислювальною потужністю без втрати якості результату.

Практичне значення роботи полягає у можливості використання розробленої системи цифровими художниками, дизайнерами та дослідниками для швидкого прототипування та редагування зображень без необхідності використання спеціалізованого апаратного забезпечення.

Ключові слова: семантичне редагування, комп'ютерний зір, генеративний штучний інтелект, *Inpainting*, мультимодальні АРІ, графічний інтерфейс користувача, клієнт-серверна архітектура.

Chornobryvets D. V., Popereshnyak S. V. An interactive system for semantic editing of raster graphics based on the integration of multimodal generative APIs

This paper investigates the process of automated semantic editing of raster images using artificial intelligence methods. The relevance of the study is обусловлена high computational demands of modern generative models, which require powerful graphics processing units (GPUs) to perform inpainting operations, as well as the limited flexibility of cloud-based services in terms of precise spatial control over editing. This creates a significant barrier to the accessibility of intelligent image editing tools for users with limited computational resources.

The aim of the work is to improve the accessibility and efficiency of semantic image editing processes by distributing the computational workload between the client-side and server-side components of the system. The proposed approach combines local tools for spatial mask generation with the use of a cloud-based multimodal API (Gemini 3 Flash Image) to perform generative transformations.

As a result, a lightweight desktop application with a modular client-server architecture has been designed and implemented. The key features of the system include asynchronous multithreaded processing of network requests, which ensures a responsive graphical user interface, as well as the use of reverse compositing algorithms for seamless integration of generated fragments into the original image. A real-time binary mask generation mechanism based on cursor coordinates has been implemented, enabling high-precision selection of regions of interest.



© Д. В. Чорнобривець, С. В. Поперешняк, 2026

Стаття поширюється на умовах ліцензії відкритого доступу CC BY 4.0

The obtained results are explained by the effective offloading of tensor computations to cloud infrastructure while maintaining local control over the editing process. Experimental evaluation confirmed the feasibility of performing complex image transformations on low-performance devices without loss of output quality.

The practical significance of the work lies in the possibility of using the developed system by digital artists, designers, and researchers for rapid prototyping and image editing without the need for specialized hardware.

Key words: semantic editing, computer vision, generative artificial intelligence, inpainting, multimodal APIs, graphical user interface, client-server architecture

Постановка проблеми. Зростання обсягів цифрового контенту та вимог до швидкості його створення призвело до серйозних викликів у сфері комп'ютерної графіки та дизайну. Складність локального редагування зображень, витрати часу на ручну ретуш та високий поріг входження для опанування професійного інструментарію стали поширеною проблемою для цифрових художників та контент-мейкерів у всьому світі. Для вирішення цієї проблеми зростає попит на інноваційні технології, здатні автоматизувати процес внесення просторових та семантичних змін у візуальні матеріали. Одним із таких рішень є інтерактивні графічні редактори, засновані на інтеграції з мультимодальними генеративними моделями штучного інтелекту.

Системи семантичного редагування, що використовують алгоритми генерації зображень за текстовим запитом (Text-to-Image) та локальної модифікації (Inpainting), дозволяють автоматично змінювати вміст растрових полотен, синтезуючи нові об'єкти з урахуванням навколишнього контексту, освітлення та перспективи. Такі системи використовують візуальні маски для просторового обмеження генерації, після чого застосовуються хмарні неймережеві алгоритми для аналізу та обробки цих даних. Вони забезпечують безпрецедентну точність та реалістичність генерації, що дозволяє користувачам отримувати якісні візуальні результати в режимі майже реального часу.

У цій роботі представлено розробку автоматизованої системи семантичного редагування растрової графіки, заснованої на інтеграції мультимодальних генеративних API. Система використовує методи обробки зображень для створення локальних масок інтересу (ROI) та аналізу текстових промптів, щоб безшовно інтегрувати згенерований контент у вихідне полотно. Застосування таких систем дає змогу значно скоротити час на виконання складних графічних маніпуляцій, зменшити вимоги до обчислювальних потужностей клієнтських пристроїв та підвищити ефективність роботи дизайнерів.

Інтеграція таких рішень може суттєво покращити робочі процеси в креативних індустріях, зменшити витрати на виробництво візуального контенту та знизити бар'єр для початківців. Крім того, така система може стати основою для розвитку комплексних інтелектуальних графічних пайплайнів, де можуть бути реалізовані додаткові функції, такі як автоматичний перенос стилю, пакетна генерація варіацій об'єктів та інші інноваційні інструменти.

Розроблені архітектурні рішення інтеграції з хмарними API мають великий потенціал для подальшого розвитку та адаптації в умовах різних творчих середовищ. Вони можуть бути масштабовані для використання в різноманітних сферах, від веброзробки до створення ігрових асетів, що дозволяє зробити ці системи універсальними і доступними для широкого кола користувачів. Крім того, архітектура може бути адаптована для інтеграції в існуючі комерційні платформи.

Постановка проблеми полягає в тому, що в умовах постійного зростання попиту на швидке створення та модифікацію графіки, а також у зв'язку зі складністю ручних інструментів Inpainting та Outpainting у традиційних редакторах, спеціалісти стикаються з труднощами при внесенні локальних змін у зображення. Це призводить до зривів дедлайнів, зайвих витрат робочого часу, збільшення рівня стресу та зниження загальної продуктивності через рутинне маніпулювання пікселями. Традиційна система редагування, яка базується на ручному штампуванні та виділенні, є неефективною та часозатратною, що спричиняє додаткові незручності.

З огляду на це, виникає потреба в автоматизованих клієнт-серверних рішеннях для семантичного редагування, які дозволяють інтерактивно визначати зони модифікації та надавати системі текстові інструкції для виконання складної генеративної роботи. Мультимодальний штучний інтелект, який дозволяє автоматично синтезувати графіку відповідно до текстового та візуального контексту, стає ефективним інструментом для розв'язання цієї проблеми.

Використання таких технологій дозволяє автоматично замінювати, додавати або видаляти об'єкти на зображеннях, скорочуючи час на ручну ретуш і покращуючи загальну ефективність пайплайнів створення контенту.

Стаття також охоплює архітектурні аспекти застосування хмарних генеративних моделей для ефективного управління обчислювальними ресурсами, що має на меті розширення доступу до передових ШІ-інструментів для користувачів з базовим апаратним забезпеченням та зниження навантаження на локальні системи.

Метою дослідження є розробка програмної архітектури та функціонального прототипу інтерактивної системи семантичного редагування растрової графіки з використанням хмарних мультимодальних генеративних API. Це дасть можливість оптимізувати процес локальної модифікації зображень (Inpainting), зменшити вимоги до обчислювальної потужності клієнтських пристроїв та підвищити швидкість і зручність створення цифрового контенту

Для досягнення поставленої мети необхідно розглянути наступні завдання:

- Провести аналіз наявної проблеми семантичного редагування зображень та існуючі технологічні рішення на базі генеративного штучного інтелекту.
- Розробити архітектурну модель клієнт-серверної взаємодії для асинхронної обробки просторових масок (зон інтересу) та мережеских запитів без блокування графічного інтерфейсу.
- Розглянути основні етапи реалізації алгоритмів формування візуальних масок та алгоритмів зворотного безшовного композитингу згенерованих результатів із вихідним зображенням.
- Описати запропонований прототип програмної системи семантичного редагування та підтвердити його працездатність на практиці.

Ці задачі забезпечують структурований підхід до дослідження, розробки та оцінки інтерактивної системи редагування графіки, що спирається на сучасні хмарні мультимодальні генеративні моделі.

Аналіз останніх досліджень і публікацій. Сучасні дослідження у сфері обробки зображень, комп'ютерного зору та інтелектуальних систем демонструють активний розвиток як класичних, так і нейромережеских підходів до аналізу та модифікації візуальних даних.

У роботах [1, 2] розглянуто класичні методи відновлення та редагування зображень (image inpainting), що базуються на інтерполяції градієнтних полів і рівнів яскравості. Дані підходи забезпечують високу точність відновлення локальних областей, однак мають обмеження при роботі зі складними текстурами та семантичними структурами, оскільки не враховують високорівневий контекст зображення.

Подальший розвиток отримали сучасні нейромережескі методи, зокрема дифузійні моделі, що розглядаються у роботі [3]. Вони дозволяють виконувати семантичне редагування зображень та генерацію відсутніх фрагментів із високою якістю. Проте такі підходи характеризуються значною обчислювальною складністю та потребують використання високопродуктивних обчислювальних ресурсів, що обмежує їх застосування на користувачьких пристроях із обмеженими ресурсами.

Паралельно розвиваються прикладні системи комп'ютерного зору, орієнтовані на обробку відео- та зображувальних даних у реальному часі. Зокрема, у роботах [4, 5, 6] досліджуються системи моніторингу зайнятості паркувальних місць із використанням комп'ютерного зору та технологій IoT. Дані рішення демонструють ефективність автоматизованого аналізу візуальної інформації, однак переважно орієнтовані на задачі детекції та класифікації об'єктів, а не на їх семантичну модифікацію.

У роботі [7] запропоновано підхід до обробки даних у системах «розумного міста», який базується на інтеграції різнорідних джерел даних та їх ефективній обробці. Це підкреслює важливість оптимізації обчислювальних процесів і структурування даних у складних інформаційних системах, що є релевантним і для задач обробки графічної інформації.

Дослідження [8] демонструє використання методів комп'ютерного зору для визначення доступності об'єктів на зображеннях, що підтверджує актуальність задач локалізації та аналізу областей інтересу (ROI). Проте в роботі не розглядаються питання інтерактивного редагування зображень або інтеграції таких методів із генеративними моделями.

З точки зору програмної реалізації, у роботі [9] розглядаються підходи до створення графічних інтерфейсів користувача, що забезпечують інтерактивну взаємодію із системами обробки даних. Однак питання ефективної інтеграції GUI із асинхронними обчислювальними процесами та віддаленими сервісами залишаються недостатньо дослідженими.

Таким чином, аналіз літературних джерел показує, що існуючі підходи можна умовно поділити на три групи:

- (1) класичні методи обробки зображень [1, 2],
- (2) сучасні нейромережескі моделі генерації та редагування [3],
- (3) прикладні системи комп'ютерного зору та обробки даних [4-9].

Незважаючи на значні досягнення у кожному з напрямів, залишається невирішеною комплексна задача поєднання:

- ефективних методів обробки зображень,
- генеративних можливостей сучасних моделей,
- ресурсно-ефективної архітектури виконання,
- та інтерактивної взаємодії з користувачем.

Причиною цього є складність інтеграції обчислювально інтенсивних генеративних моделей із легковаговими клієнтськими застосунками, що працюють у реальному часі. З огляду на це виникає потреба у розробці програмних рішень, які б забезпечували делегування обчислювально складних операцій у хмарне середовище, зберігаючи при цьому інтерактивність та ефективність локальної обробки даних.

Матеріали та методи досліджень. Об'єктом дослідження є процес автоматизованого семантичного редагування растрових зображень з використанням методів штучного інтелекту в умовах обмежених локальних обчислювальних ресурсів.

1. Архітектура програмної системи та технологічна реалізація

Основна гіпотеза дослідження полягає в тому, що делегування важких генеративних обчислень (зокрема операцій Inpainting) хмарним мультимодальним API з одночасним використанням локальних інструментів для створення просторових масок дозволить досягти високої якості редагування без втрати чуйності графічного інтерфейсу клієнтського застосунку.

Прийняті в роботі припущення: вважається, що клієнтський пристрій має стабільне ширококутне підключення до мережі Інтернет, швидкість якого є достатньою для передачі растрових масивів даних без критичних мережових затримок (таймаутів).

Прийняті в роботі спрощення: у межах створеного прототипу не розглядаються механізми шифрування трафіку на рівні застосунку, локального кешування історії генерацій та просунутої кольорокорекції, оскільки головний науковий фокус зосереджено безпосередньо на архітектурі взаємодії з ШІ та управлінні потоками.

Для проведення досліджень та створення програмного прототипу використовувалася мова програмування Python. Графічний інтерфейс користувача (GUI) було спроектовано за допомогою стандартної бібліотеки tkinter, що забезпечує кросплатформеність та низьке споживання оперативної пам'яті. Робота з растровою графікою реалізована на базі бібліотеки Pillow (PIL). З її допомогою виконувалися операції завантаження зображень, їх масштабування без втрати пропорцій (з використанням фільтра LANCZOS), генерація 8-бітних бінарних масок (режим «L») у реальному часі на основі координат курсора, а також альфа-композиція для візуального відображення зони редагування користувачу.

Взаємодія з хмарним генеративним штучним інтелектом здійснювалася через офіційний SDK для виклику мультимодального API моделі Gemini 3 Flash Image, а саме процедурного агенту **Nano Banana Pro**. Для забезпечення стабільності роботи застосунку застосовано методи асинхронного та багатопотокового програмування за допомогою стандартного модуля threading. Це дозволило ізолювати операції серіалізації графічних даних (перетворення зображення та маски у байтові масиви формату PNG) та очікування мережової відповіді у фонові процеси, відокремивши їх від головного циклу обробки подій інтерфейсу (Main Loop).

Експериментальне тестування розроблених алгоритмів та архітектури проводилося на базовій комп'ютерній станції без використання дискретного графічного прискорювача (GPU). Процес включав завантаження серії тестових зображень, ручне малювання просторових масок різної складності, відправку структурованих текстових промптів та оцінку коректності алгоритму зворотного накладання отриманих від API результатів на оригінальне полотно.

2. Математична модель семантичного редагування та обчислювальної взаємодії

Для формалізації процесу семантичного редагування растрових зображень та обґрунтування ефективності запропонованої архітектури доцільно представити відповідний обчислювальний процес у вигляді математичної моделі.

Нехай вхідне растрове зображення подано у вигляді дискретної функції:

$$I : \Omega \rightarrow \mathbb{R}^3,$$

де $\Omega \subset \mathbb{Z}^2$ – множина координат пікселів, а значення функції відповідають кольоровим каналам (RGB). Таким чином, кожен піксель зображення розглядається як елемент простору кольорів.

Для локалізації області редагування вводиться маска:

$$:\Omega \rightarrow \{0, 1\},$$

яка визначає просторову зону інтересу (ROI). Значення $M(x, y) = 1$ відповідає пікселям, що підлягають модифікації, тоді як $M(x, y) = 0$ – пікселям, що залишаються незмінними. Така формалізація дозволяє розділити обчислювальний процес на локальну та глобальну складові.

Процес семантичного редагування можна представити у вигляді оператора:

$$\mathcal{F}(I, M, T) = I',$$

де T – текстовий опис (промпт), що визначає характер модифікації, а I' – результуюче зображення. У цьому випадку оператор \mathcal{F} реалізується через взаємодію з хмарною генеративною моделлю.

З урахуванням архітектури клієнт-серверної взаємодії, процес генерації нових піксельних значень описується відображенням:

$$G : (I, M, T) \rightarrow I_{gen},$$

де I_{gen} – частково або повністю згенероване зображення, отримане в результаті обробки на стороні хмарного сервера. При цьому основне обчислювальне навантаження переноситься з локального пристрою на віддалені обчислювальні ресурси.

Після отримання результату виконується операція зворотного композивання, яка забезпечує інтеграцію згенерованого фрагмента у вихідне зображення:

$$I'(x, y) = \begin{cases} I_{gen}(x, y), & M(x, y) = 1 \\ I(x, y), & M(x, y) = 0 \end{cases}$$

Ця операція гарантує, що зміни застосовуються виключно в межах визначеної області, зберігаючи контекст та цілісність решти зображення.

З точки зору комп'ютерної інженерії важливим є аналіз часових характеристик обчислювального процесу. Загальний час обробки можна подати як суму окремих етапів:

$$T_{total} = T_{prep} + T_{net} + T_{gen} + T_{comp},$$

де T_{prep} – час формування маски та підготовки даних, T_{net} – затримки мережевої передачі, T_{gen} – час генерації у хмарному середовищі, а T_{comp} – час локального композитингу результату.

У запропонованій системі застосовано асинхронну багатопотокову обробку, що дозволяє виконувати окремі етапи паралельно. У цьому випадку ефективний час виконання визначається як:

$$T_{async} = \max(T_{UI}, T_{net} + T_{gen}),$$

де T_{UI} – час обробки подій графічного інтерфейсу. Такий підхід забезпечує відсутність блокування GUI та підвищує інтерактивність системи.

Для оцінки ефективності паралелізації введемо коефіцієнт прискорення:

$$E = \frac{T_{seq}}{T_{async}},$$

де T_{seq} – час послідовного виконання всіх етапів. Значення $E > 1$ свідчить про доцільність використання асинхронної моделі виконання.

Таким чином, запропонована математична модель дозволяє формалізувати процес семантичного редагування зображень, описати розподіл обчислювального навантаження між клієнтом і сервером та обґрунтувати ефективність використання паралельних обчислень у задачах обробки графічних даних.

Результати досліджень

1. Аналіз наявної проблеми та існуючих технологічних рішень

У результаті аналізу проблеми ресурсоемності семантичного редагування було визначено, що оптимальним шляхом її вирішення є використання моделі «тонкого клієнта». Замість розгортання важких дифузійних нейромереж у локальному середовищі комп'ютера, запропоновано підхід, за якого клієнтський застосунок відповідає лише за просторову розмітку (маскінг) та формування запиту, а безпосередня генерація пікселів (Inpainting) делегується хмарному мультимодальному API. Це нівелює залежність від апаратних характеристик робочої станції користувача.

2. Архітектурна модель клієнт-серверної взаємодії

Розроблено програмну архітектуру, яка забезпечує безперебійну роботу графічного інтерфейсу під час виконання тривалих мережевих запитів до генеративного штучного інтелекту. Вирішення проблеми «зависання» інтерфейсу досягнуто шляхом розпаралелювання процесів: головний потік (MainThread) обробляє виключно події графічної бібліотеки tkinter (рух миші, кліки, перемальовування полотна), тоді як формування байтових масивів та виклик хмарного API винесено в окремий фоновий потік (WorkerThread). Взаємодія між потоками реалізована через безпечні виклики `root.after()`, що дозволяє оновлювати UI лише після повного завантаження згенерованого результату в оперативну пам'ять.

3. Реалізація алгоритмів формування візуальних масок та композитингу

Реалізовано алгоритм інтерактивного формування зон інтересу (ROI) для генерації. Користувачке графічне полотно (Canvas) синхронізовано з тінювим об'єктом бінарної маски (зображення в режимі «L»). Під час руху курсору система динамічно вираховує координати відносно відступів (центрування масштабованого зображення) та малює білі зони (значення 255) на чорному тлі маски. Для забезпечення візуального контролю реалізовано алгоритм альфа-композитингу: чорні ділянки маски стають прозорими, а білі – заповнюються червоним напівпрозорим кольором (RGBA: 255, 0, 0, 120), який накладається поверх оригінального зображення (Image.composite). Після отримання відповіді від API згенероване зображення масштабується до актуальних розмірів вікна (за допомогою фільтра LANCZOS) і безшовно замінює попередній стан робочого простору (Рисунок 1).

4. Програмний прототип системи семантичного редагування

Створено функціональний прототип десктопного застосунку, який успішно виконує задачі семантичного редагування растрової графіки. Прототип інтегровано з хмарним API Gemini 3 Flash Image.

Робота розробленої системи відбувається за чітким послідовним алгоритмом. На першому етапі користувач завантажує вихідне зображення в робочу область програми. Графічний інтерфейс надає доступ до панелі інструментів для налаштування розміру пензля та поля введення текстових команд (Рисунок 2).

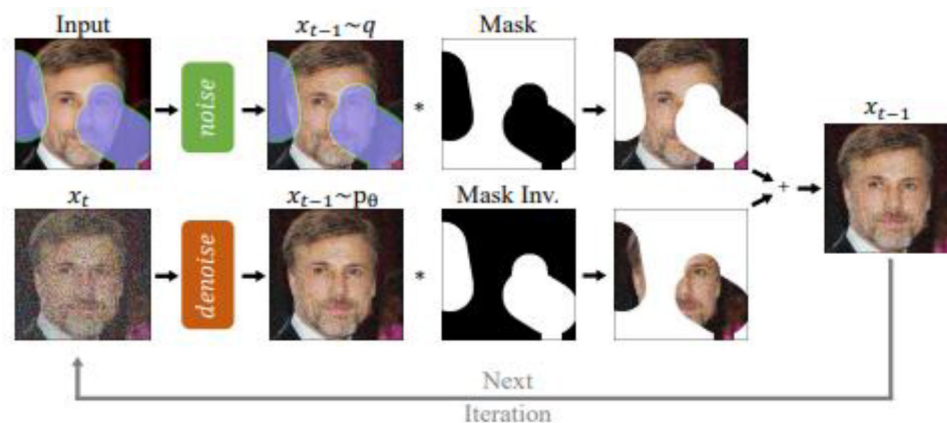


Рис. 1. Огляд нашого підходу

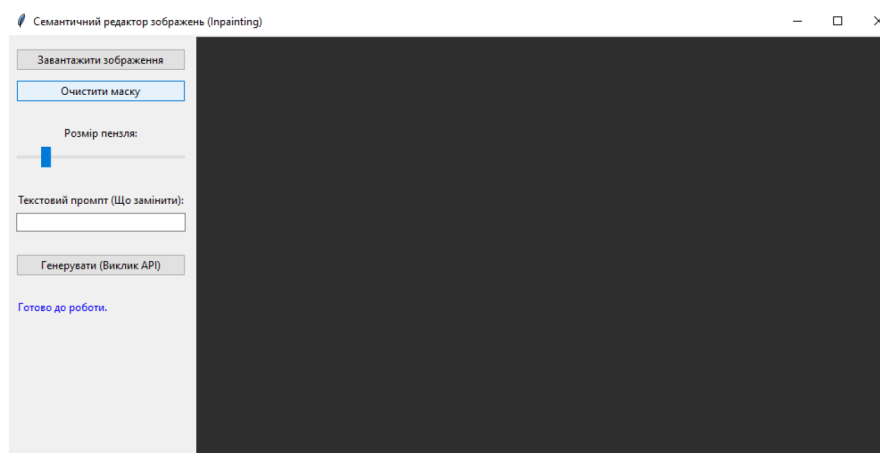


Рис. 2. Графічний інтерфейс розробленої системи

Наступним кроком є інтерактивне формування зони інтересу (ROI). За допомогою інструменту «Пензель» користувач виділяє область для модифікації. Система в реальному часі генерує бінарну маску та накладає її на оригінальне зображення у вигляді напівпрозорого шару (Рисунок 3), що дозволяє візуально контролювати межі генерації. Одночасно формується текстовий запит для хмарної моделі

Під час тестування прототип продемонстрував здатність коректно конвертувати локальну маску та вихідне зображення у байтові масиви (формат PNG) і надсилати їх разом із динамічно сформованим текстовим промптом. Промпт містить жорсткі системні інструкції, які обмежують дію нейромережі виключно виділеними зонами маски. Система успішно приймає багатокomпонентну відповідь від API, вилучає з неї растрові дані та виконує зворотний композитинг (Рисунок 4).

Як видно з отриманих результатів (Рисунок 3), генеративна модель модифікувала виключно виділену користувачем зону, зберігши ідентичність фонового контексту, освітлення та перспективи. Це доводить повну працездатність архітектури та ефективність запропонованого підходу для задач цифрового дизайну.

Обговорення результатів. Отримані результати успішного функціонування прототипу семантичного редактора можуть бути пояснені ефективним розділенням обчислювального навантаження між локальним клієнтом та хмарним сервером. Завдяки реалізації багатопотоковості (ізоляції мережевих запитів у `WorkerThread`), графічний інтерфейс залишається відгукливим навіть під час тривалого очікування відповіді від генеративної моделі.

На відміну від сучасних підходів до семантичного редагування на основі дифузійних моделей [3], де для виконання операцій Inpainting потрібні значні обчислювальні ресурси (зокрема GPU із великим обсягом відеопам'яті), запропоноване у роботі рішення дозволяє виконувати складні графічні модифікації на пристроях із обмеженими ресурсами. Це досягається за рахунок перенесення обчислювально складних етапів генерації у хмарне середовище.

Класичні підходи до відновлення зображень [1, 2], що базуються на інтерполяційних методах, не забезпечують достатньої якості при складних семантичних змінах, що також підтверджує доцільність використання генеративних моделей у запропонованій системі.

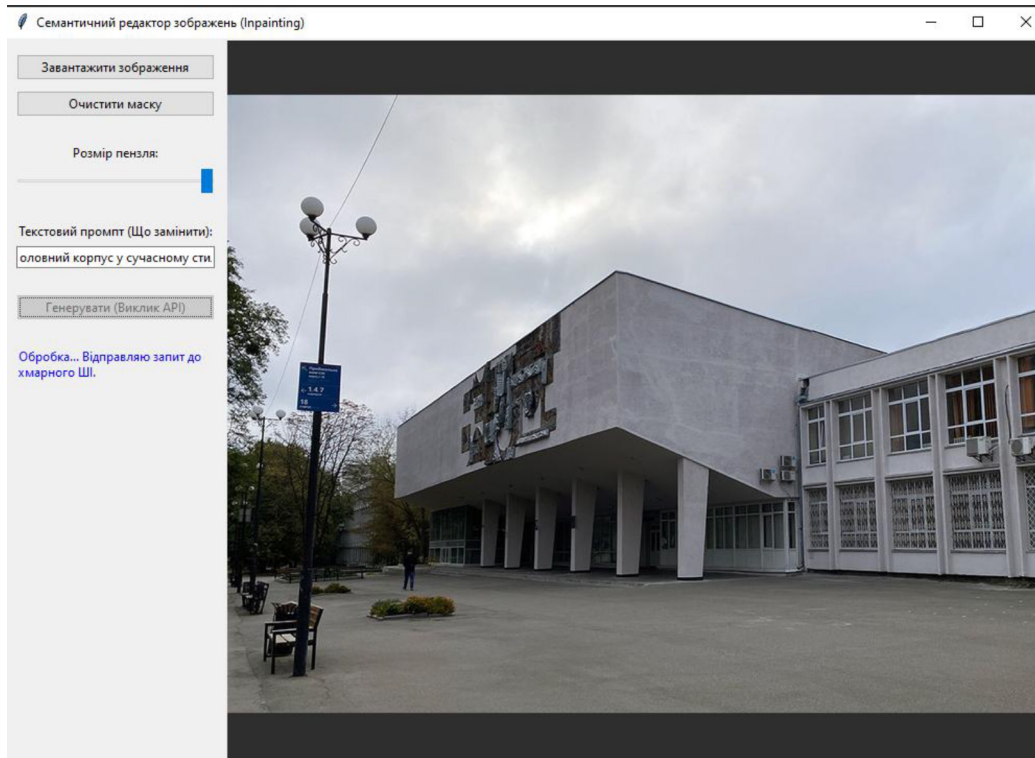


Рис. 3. Процес виділення ROI (червоний шар) та формування семантичного промпту

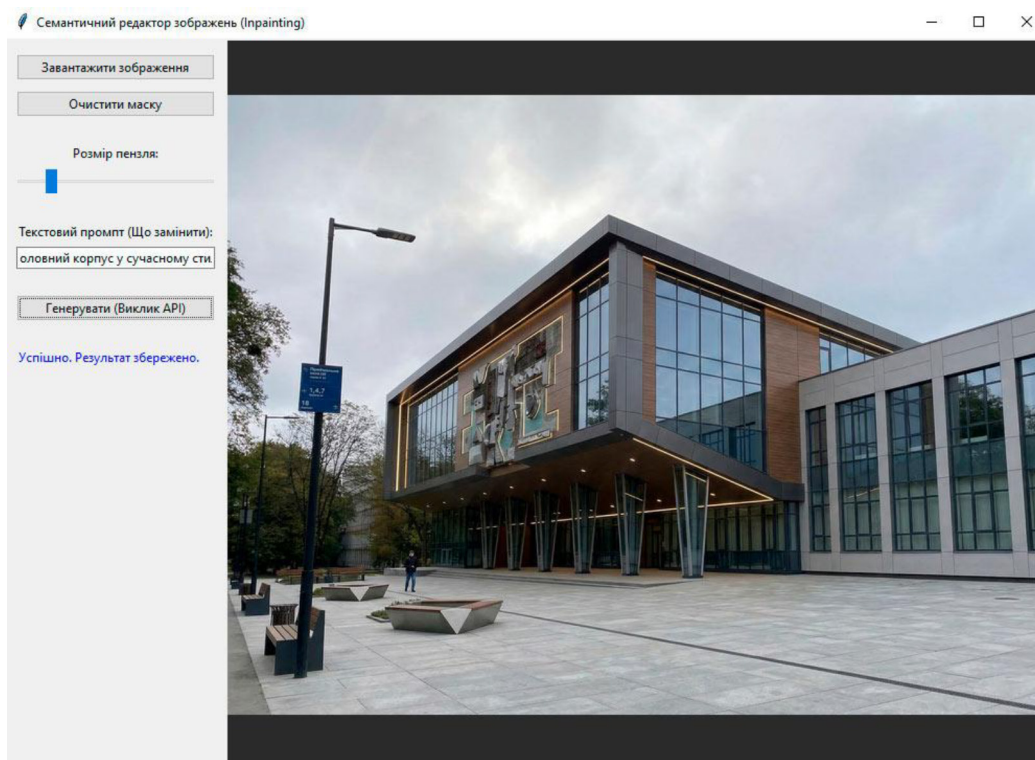


Рис. 4. Результат локальної модифікації зображення (Inpainting) на основі згенерованих хмарним API даних

Водночас, на відміну від традиційних систем обробки зображень із графічним інтерфейсом [9], розроблена система забезпечує інтеграцію інтерактивного інтерфейсу з асинхронною обробкою та віддаленими обчислювальними сервісами. Це дозволяє здійснювати прецизійне редагування за допомогою локально сформованих масок без втрати відгуку інтерфейсу.

Отримані результати узгоджуються із сучасними тенденціями розвитку систем комп'ютерного зору та обробки даних [4, 5, 6], де ефективність досягається за рахунок розподілу обчислень та використання зовнішніх обчислювальних ресурсів. Крім того, підходи до організації обробки даних у складних системах [7, 8] підтверджують доцільність використання модульної та масштабованої архітектури.

Отримані програмні рішення повністю закривають проблемну частину, яка визначена в роботі: вони ліквідують бар'єр високих апаратних вимог і водночас надають користувачу зручній інструментарій для просторового виділення зон інтересу.

Проте запропоноване дослідження має певні обмеження, які необхідно враховувати при застосуванні на практиці. Головним обмеженням є залежність системи від якості інтернет-з'єднання. При низькій пропускній здатності мережі передача растрових даних може спричинити значні затримки. Крім того, робота системи обмежується квотами та лімітами обраного API.

Основним недоліком поточного прототипу є відсутність механізмів локального кешування та функцій «скасування» (Undo/Redo), що знижує зручність використання. Також час очікування результату є більшим порівняно з локальною генерацією на високопродуктивному обладнанні.

Подальший розвиток дослідження доцільно спрямувати на впровадження пакетної обробки (Batch Processing) та розширення підтримки різних генеративних моделей, що дозволить підвищити гнучкість і відмовостійкість системи.

Висновки. У роботі проведено аналіз сучасних підходів до семантичного редагування зображень, який показав, що існуючі рішення мають суттєві обмеження: локальні системи характеризуються високими вимогами до обчислювальних ресурсів, тоді як хмарні сервіси не забезпечують достатньої гнучкості інтерактивного управління процесом редагування. Це обумовило необхідність розроблення легковагового клієнтського застосунку з ефективним розподілом обчислювального навантаження.

У результаті дослідження спроектовано та реалізовано модульну клієнт-серверну архітектуру, яка забезпечує асинхронну взаємодію з хмарними сервісами обробки зображень. Завдяки використанню багатопотокової моделі виконання досягнуто усунення блокування графічного інтерфейсу та забезпечено стабільну роботу системи в умовах тривалих мережевих операцій.

Розроблено графічний інтерфейс користувача на основі фреймворку tkinter, у якому реалізовано геометричний алгоритм формування бінарних масок у реальному часі. Це дозволило забезпечити точне виділення областей інтересу без використання сторонніх інструментів і підвищити зручність взаємодії користувача із системою.

Інтеграція з хмарним мультимодальним API та реалізація алгоритму зворотного композитингу забезпечили коректне вбудовування згенерованих фрагментів у вихідне зображення. Експериментальні результати підтвердили, що запропонований підхід дозволяє здійснювати локальні зміни зображення без порушення цілісності контексту, що свідчить про ефективність використання маскінгу у поєднанні з генеративними моделями.

Отримані результати мають практичну цінність для розробки інтелектуальних систем редагування зображень і можуть бути використані при створенні легковагових графічних редакторів, інтегрованих із хмарними обчислювальними сервісами. Подальші дослідження доцільно спрямувати на підвищення автономності системи, зменшення залежності від мережевих умов та розширення функціональних можливостей за рахунок підтримки різних генеративних моделей.

Список використаних джерел:

1. Coloma Ballester, Bertalmio M., Caselles V., Sapiro G., Verdera J. Filling-in by joint interpolation of vector fields and gray levels // *IEEE Transactions on Image Processing*. 2001. Vol. 10, No. 8. P. 1200–1211.
2. Bertalmio M., Sapiro G., Caselles V., Ballester C. Image inpainting // *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. 2000. P. 417–424.
3. Lugmayr A., Danelljan M., Romero A., Yu F., Timofte R., Van Gool L. RePaint: inpainting using denoising diffusion probabilistic models // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022. P. 11461–11471.
4. Dixit M., Srimathi C., Doss R., Loke S., Saleemdurai M. A. Smart parking with computer vision and IoT technology // *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. Milan, Italy, 2020. P. 170–174. DOI: 10.1109/TSP49548.2020.9163467
5. Giampaoli L. E., Hessel F. Parking space occupancy monitoring system using computer vision and IoT // *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. New Orleans, LA, USA, 2021. P. 7–12. DOI: 10.1109/WF-IoT51360.2021.9595935
6. Kuzela M., Fryza T., Zeleny O. Using computer vision and machine learning for efficient parking management: a case study // *2024 13th Mediterranean Conference on Embedded Computing (MECO)*. Budva, Montenegro, 2024. P. 1–4. DOI: 10.1109/MECO62516.2024.10577808

7. Popereshnyak S., Yurchuk I. Car parking data processing technique for smart parking system as part of smart city // *Advances in Intelligent Systems and Computing*. 2021. Vol. 1246. DOI: <https://doi.org/10.1007/978-3-030-54215-3>

8. Поперешняк С. В., Чорнобривець Д. В. Підхід до виявлення доступності паркувальних місць на основі комп'ютерного зору // *Системи та технології*. 2025. № 69(1). С. 83–91. DOI: <https://doi.org/10.32782/2521-6643-2025-1-69.10>

9. Moore A. Python GUI programming with Tkinter: develop responsive and powerful GUI applications with Tkinter and Python 3. 2nd ed. Birmingham: Packt Publishing, 2018. 368 p.

References:

1. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., & Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8), 1200–1211.

2. Bertalmio, M., Sapiro, G., Caselles, V., & Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 417–424).

3. Dixit, M., Srimathi, C., Doss, R., Loke, S., & Saleemdurai, M. A. (2020). Smart parking with computer vision and IoT technology. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)* (pp. 170–174). <https://doi.org/10.1109/TSP49548.2020.9163467>

4. Giampaoli, L. E., & Hessel, F. (2021). Parking space occupancy monitoring system using computer vision and IoT. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)* (pp. 7–12). <https://doi.org/10.1109/WF-IoT51360.2021.9595935>

5. Kuzela, M., Fryza, T., & Zeleny, O. (2024). Using computer vision and machine learning for efficient parking management: A case study. In *2024 13th Mediterranean Conference on Embedded Computing (MECO)* (pp. 1–4). <https://doi.org/10.1109/MECO62516.2024.10577808>

6. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11461–11471).

7. Moore, A. (2018). *Python GUI programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter and Python 3* (2nd ed.). Packt Publishing.

8. Popereshnyak, S., Yurchuk, I. (2021). Car parking data processing technique for smart parking system as part of smart city. *Advances in Intelligent Systems and Computing*, 1246. <https://doi.org/10.1007/978-3-030-54215-3>

9. Popereshnyak, S. V., Chornobryvets, D. V. (2025). Pidkhyd do vyyavlennya dostupnosti parkuval'nykh mist' na osnovi komp'yuternoho zoru [Approach to detecting parking space availability based on computer vision]. *Systemy ta tekhnolohiyi* [Systems and Technologies], 69(1), 83–91. <https://doi.org/10.32782/2521-6643-2025-1-69.10>

Дата першого надходження статті до видання: 24.03.2026

Дата прийняття статті до друку після рецензування: 20.04.2026

Дата публікації (оприлюднення) статті: 30.05.2026