Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів сегментації та редагування зображень високої роздільної здатності на основі запитів природною мовою
(тема)

Виконав:
здобувач 2 року навчання
групи ІПЗм-23-2

Максим КІЗІЦЬКИЙ
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. Олексій ТУРУТА
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Кізіцькому Максиму Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів сегментації та редагування зображень високої роздільної здатності на основі запитів природною мовою»
2. Затверджена наказом по університету № 290 Ст від 15.04. 2025р.
3. Термін подання студентом роботи до екзаменаційної комісії 16.06.2025
4. Вихідні дані до роботи алгоритми сегментації зображень (SAM, U-Net, ViT), дифузійні моделі генерації (Stable Diffusion), текстові моделі (CLIP), мови програмування Python, фреймворки PyTorch, Diffusers, Transformers, ресурси з безпеки в AI
5. Перелік питань, що потрібно опрацювати в роботі
аналіз предметної області, постановка задачі безпечного редагування, дослідження архітектур сегментації та генерації, реалізація та інтеграція моделей, проведення експериментів, аналіз результатів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04.2025	*виконано*
2	Аналіз предметної галузі і постановка задачі	16.04.2025 - 26.04.2025	*виконано*
3	Інтеграція моделей сегментації та редагування	26.04.2025 - 06.05.2025	*виконано*
4	Тестування механізму безпеки моделі	06.05.2025 - 10.05.2025	*виконано*
5	Підготовка до апробації результатів дослідження. Публікація матеріалів	11.05.2025	*виконано*
6	Побудова єдиного конвеєру моделей	12.05.2025 - 20.05.2025	*виконано*
7	Підготовка пояснювальної записки	20.05.2025 - 05.06.2025	*виконано*
8	Підготовка презентації та доповіді	05.06.2025 - 08.06.2025	*виконано*
9	Перевірка на плагіат	08.06.2025	*виконано*
10	Нормоконтроль	09.06.2025	*виконано*
11	Рецензування	10.06.2025	*виконано*
12	Попередній захист	11.06.2025	*виконано*
13	Занесення диплома в електронний архів	12.06.2025	*виконано*
14	Допуск до захисту у зав. кафедри	13.06.2025	*виконано*

Дата видачі завдання 16 квітня 2025р.

Студент (ка) _____ Максим КІЗІЦЬКИЙ
(підпис)

Керівник роботи _____ доц. Олексій ТУРУТА
(підпис) (посада, Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 68 с., 26 рис., 7 табл., 31 джерело.

БЕЗПЕКА ШІ, ДИФУЗІЙНІ МОДЕЛІ, РЕДАГУВАННЯ ЗОБРАЖЕНЬ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, LORA, PYTHON.

Об’єктом дослідження є системи сегментації та редагування зображень на основі підказок природною мовою.

Метою роботи є розробка інтегрованого підходу до сегментації та редагування зображень з механізмами захисту від зловмисних інструкцій на базі LoRA.

Методи дослідження базуються на методах глибинного навчання (моделі SAM, Stable Diffusion), захисних техніках (LoRA, Metric Learning) та інструментах Python, PyTorch, HuggingFace.

Результатом роботи є розробка та оцінка прототипу інтегрованої системи сегментації-редагування зі значно підвищеною стійкістю до prompt-based атак.

AI SAFETY, DIFFUSION MODELS, IMAGE EDITING, IMAGE SEGMENTATION, LORA, PYTHON

The object of the research is systems for image segmentation and editing based on natural language prompts.

The aim of the work is to develop an integrated approach to image segmentation and editing with protection mechanisms against malicious instructions based on LoRA.

The research methods are based on deep learning techniques (SAM, Stable Diffusion models), defensive techniques (LoRA, Metric Learning), and tools such as Python, PyTorch, and HuggingFace.

The result of the work is the development and evaluation of a prototype integrated segmentation-editing system with significantly improved resistance to prompt-based attacks.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу ElArKhNURE.

Завідувачу кафедри
ПІ
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ
(вчене звання, сласне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я, Кізіцький Максим Олегович
(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної групи ІПЗм-23-2

кафедра програмної інженерії ,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему
Дослідження методів сегментації та редагування зображень високої роздільної здатності на основі запитів природною мовою ,
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "ElArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "ElArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата Підпис

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ASCII – American Standard Code for Information Interchange

LoRA – Low-Rank Adaptation

ASR – Attack Success Rate

FID – Fréchet Inception Distance

IoU – Intersection over Union

LPIPS – Learned Perceptual Image

I2P – Inappropriate Image Prompts

SAM – Segment Anything Model

DPP – Defensive Prompt Patch

UPS – Utility Preservation Score

ViT – Vision Transformer

ВСТУП

Сегментація та редагування зображень є важливими складовими сучасних технологій комп'ютерного зору. З розвитком дифузійних моделей і архітектури transformers з'явилася можливість взаємодіяти з візуальними системами за допомогою природної мови. Такі моделі, як Segment Anything Model [1] та Stable Diffusion [2], дозволяють точно виділяти об'єкти на зображеннях і змінювати їх характеристики, використовуючи лише текстові інструкції (prompts). Це відкриває широкі можливості для застосування у сферах цифрового мистецтва, маркетингу медичної візуалізації, індустріального контролю якості, супутникового моніторингу тощо.

Але попри широкий спектр прикладних задач, у даній області наявна низка нових викликів, пов'язаних із високою роздільною здатністю зображень та потенційною вразливістю таких систем до шкідливих запитів. Уразливість моделей до так званих «prompt-based» атак може призвести до некоректного редагування вмісту або витоку конфіденційної інформації, що створює загрозу в критичних системах, зокрема в медицині чи безпеці.

Метою роботи є дослідження та розробка інтегрованого підходу до сегментації та редагування зображень високої роздільної здатності з використанням запитів природною мовою, який також включатиме захисні механізми від зловмисних інструкцій. Для цього передбачено аналіз сучасних моделей та розробка власної із підтримкою захисту, а також експериментальна перевірка на відкритих наборах даних.

Об'єктом дослідження є системи сегментації та редагування зображень, що використовують підказки у вигляді природної мови.

Предметом дослідження є методи підвищення точності та безпеки таких систем у роботі з високороздільними зображеннями.

У процесі роботи були використані методи глибинного навчання, архітектури transformers, інструменти редагування зображень на основі дифузійних моделей, а також підходи до захисту від шкідливих запитів, такі як адаптивне LoRA-донавчання [4]. Оцінка системи здійснювалась за метриками

Attack Success Rate (ASR), FID, IoU, LPIPS, з використанням реальних та синтетичних даних з репозиторіїв Adversarial Nibbler [5] та I2P [6].

Наукова новизна полягає в об’єднанні сегментації, дифузійного редагування та модульного захисту в єдиному середовищі, орієнтованому на роботу з високоякісними зображеннями та динамічною зміною запитів. Результати роботи можуть бути застосовані для покращення надійності інтерактивних систем редагування зображень, зокрема в контексті захисту від prompt-based атак.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз існуючих підходів у сегментація та редагуванні зображень

Нещодавні досягнення у штучному інтелекті та машинному навчанні [7-14], а особливо обробка та аналіз візуальних даних є фундаментальною складовою сучасних інформаційних технологій та систем штучного інтелекту. Автоматичне виділення значущих об’єктів чи регіонів на зображенні (сегментація) та модифікація їх відповідно до заданих критеріїв (редагування) має широке прикладне застосування. Наприклад: медична візуалізація (аналіз патологій на МРТ/КТ), промисловий контроль якості (дефектоскопія), автономні системи (навігація, розпізнавання перешкод), системи безпеки та спостереження, геоінформаційні системи (аналіз земної поверхні), а також творчі індустрії, маркетинг та створення контенту.

Методи сегментації пройшли значну еволюцію. Ранні підходи базувалися на аналізі низькорівневих ознак пікселів, таких як інтенсивність чи градієнти (порогова бінаризація, ріст регіонів, методи активних контурів). Поява згорткових нейронних мереж (CNN) дозволило створювати моделі, що навчаються на даних та здатні виявляти складні ієрархічні ознаки. Ключовими архітектурами стали U-Net [15], особливо ефективна для біомедичних зображень завдяки своїй енкодер-декодерній структурі та skip-connections), що зберігають локальні деталі; Mask R-CNN [16], що інтегрує детекцію об’єктів та їх попіксельну сегментацію; та різноманітні варіації YOLO [17] для швидкої обробки в реальному часі. Незважаючи на високу точність, ці моделі зазвичай вимагають великих анотованих датасетів і обмежені заздалегідь визначеним набором класів об’єктів.

Новий виток розвитку пов’язаний із архітектурами transformers, запозиченими з обробки природної мови, та парадигмою взаємодії на основі інструкцій (prompt-based). Моделі, такі як Segment Anything Model (SAM) [1], демонструють здатність до узагальнення та сегментації довільних об’єктів без необхідності специфічного навчання під кожен клас. SAM може керуватися різними типами підказок: текстовими описами, точками на зображенні, обмежуючими рамками або навіть масками інших об'єктів. Ця гнучкість, що часто

називається «zero-shot» або «few-shot» здатністю, є ключовою перевагою. Приклад використання цієї моделі наведено на рисунку 1.1.

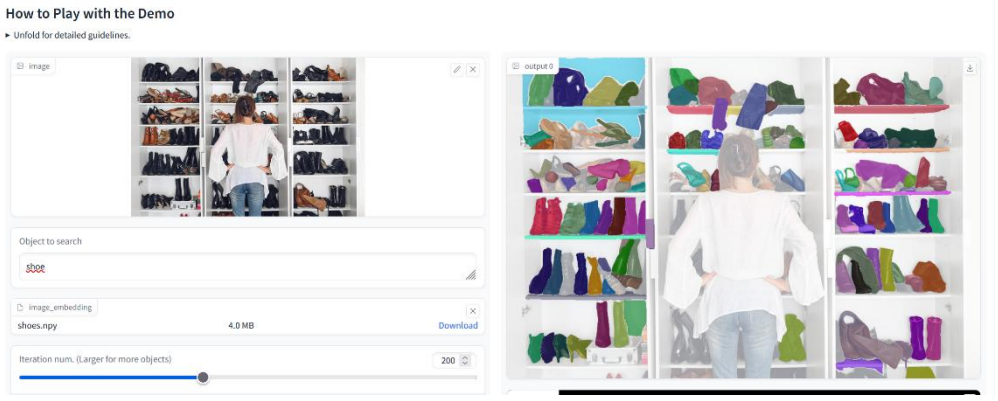

Рисунок 1.1 – Приклад роботи SAM (за даними [1])

У сфері редагування зображень аналогічний вплив справили моделі, що використовують текстові інструкції. Такі моделі як, Paint by Example [18,19] дозволяють користувачам описувати бажані зміни природною мовою (наприклад, «зміни колір сукні на червоний», «додай сонце на небо»). Особливо потужними інструментами для генерації та редагування стали дифузійні моделі, такі як Stable Diffusion [2], DALL-E [20], Midjourney [21]. Вони працюють шляхом симуляції процесу дифузії: спочатку до зображення поступово додається шум (forward diffusion), а потім модель навчається відновлювати зображення з шуму, керуючись певною умовою, наприклад, текстовим описом (reverse diffusion). Це дозволяє досягти високої фотореалістичності та виконувати складні маніпуляції: заповнення відсутніх областей (inpainting), розширення зображення (outpainting), зміну стилю (style transfer) тощо. Важливою складовою багатьох prompt-based систем є мультимодальні моделі (напр., CLIP [3]), які навчаються спільному представленню тексту та зображень, дозволяючи моделям «розуміти» зв’язок між текстовим описом та візуальним контентом.

Однак, застосування цих передових технологій пов’язане з певними викликами. Робота із зображеннями високої роздільної здатності (2K, 4K та вище) вимагає значних обчислювальних ресурсів, особливо великого обсягу відеопам’яті (VRAM) на GPU, і може бути повільною. Збільшення роздільної здатності також підвищує ризик появи візуальних артефактів: розмиття контурів,

спотворення дрібних деталей, нереалістичні текстури, проблеми з когерентністю на великих зображеннях. Якість результату, особливо при редагуванні, сильно залежить від точності та однозначності формулювання текстової підказки. Для навчання потужних моделей потрібні великі та різноманітні набори даних, такі як LAION або специфічні колекції пар «зображення-підказка» (напр., Lexica [22], OpenPrompt [23]).

1.2 Аспекти безпеки та захист від prompt-based атак

Інтерактивність та потужність сучасних моделей, керованих підказками, роблять їх привабливою ціллю для зловмисників. Виникає новий клас загроз – атаки на основі інструкцій (prompt-based attacks), які експлуатують візуальний або інтерфейс взаємодії природньою мовою для маніпулювання поведінкою моделі. На відміну від традиційних adversarial attacks, що вносять непомітні для людини зміни в пікселі зображення, prompt-based атаки діють на семантичному рівні, використовуючи спеціально сконструйовані запити.

Основні типи вразливостей та атак включають:

– вбудовування прихованих команд або інструкцій у легітимний запит (Prompt Injection [24]), що змушує модель виконувати непередбачені дії або ігнорувати попередні інструкції;
– формулювання запитів (часто складних або непрямих) таким чином, щоб обійти вбудовані фільтри безпеки та етичні обмеження моделі, змушуючи її генерувати шкідливий, образливий, неправдивий або інший заборонений контент (Jailbreaking [25, 26]). Це може включати генерацію насильницьких сцен, дезінформації, контенту, що розпалює ворожнечу, тощо;
– витік даних (Data Leakage): спроби через специфічні запити отримати доступ до конфіденційних даних, на яких навчалася модель, або до інформації про її внутрішню структуру чи параметри;

– нецільова генерація (Misinformation/Disinformation Generation): створення правдоподібних, але неправдивих зображень або маніпуляція існуючими для поширення дезінформації.

Наслідки таких атак можуть бути серйозними: від спотворення результатів наукових досліджень чи медичної діагностики до створення інструментів для масової дезінформації, шахрайства чи завдання репутаційної шкоди. Тому забезпечення стійкості та безпеки prompt-based систем є критично важливим завданням.

Для протидії цим загрозам активно розробляються та досліджуються різноманітні захисні стратегії:

– валідація та фільтрація вхідних даних: простіший підхід, що включає перевірку запитів на наявність у «чорних списках», використання регулярних виразів для виявлення небезпечних патернів, або базовий семантичний аналіз для ідентифікації підозрілих інструкцій;
– Defensive Prompt Patch (DPP) [27]: метод додавання до користувацького запиту спеціальної, навченої послідовності токенів («патчу»), яка діє як префікс або суфікс і допомагає моделі протистояти шкідливим інструкціям, що можуть міститися в основному запиті;
– адаптивне екранування запитів (Adaptive Prompt Shielding) [28]: більш динамічні системи, що аналізують запит і намагаються ідентифікувати та нейтралізувати або видалити потенційно шкідливі частини перед передачею його основній моделі;
– методи Erase-and-check [29]: ітеративний підхід, де частини запиту послідовно видаляються, а результат роботи моделі аналізується, щоб визначити, яка частина запиту була шкідливою;
– детектування атак за допомогою ML: навчання окремих моделей-класифікаторів для розпізнавання зловмисних запитів на основі їхніх характеристик.

Ефективність цих захисних механізмів не є абсолютною. Постійно з’являються нові методи атак (напр., багатоетапні атаки, атаки з використанням

незвичних кодувань або символів), що вимагає безперервного вдосконалення захисту. Існує також компроміс між рівнем безпеки та функціональністю: надто суворі фільтри можуть блокувати не шкідливі, але нетипові запити, знижуючи корисність системи. Важливим напрямком є розробка стандартизованих наборів даних та методологій для тестування стійкості моделей до атак, таких як Adversarial Nibbler [5] або I2P (Inappropriate Image Prompts) [6].

1.3 Виявлення проблем та актуалізація рішень

Аналіз поточного стану та ключових викликів у галузі prompt-based сегментації та редагування зображень дає змогу визначити такі провідні тенденції й напрями досліджень:

- інтеграція та мультимодальність. Відбувається посилення взаємодії між різними завданнями обробки візуальних та текстових даних у межах спільних мультимодальних архітектур. Розробляються системи, здатні вести інтерактивний діалог із користувачем, уточнюючи вхідні запити й формуючи комплексні результати;
- покращена керованість і деталізація. Розробляються підходи до забезпечення більш контрольованих та передбачуваних змін візуального контенту, наприклад, збереження ідентичності об’єкта за зміни стилю чи точне відтворення складних інструкцій. Особливий акцент робиться на підвищенні точності сегментації дрібних елементів, тонких структур і різнотипних об’єктів у складних умовах (затінення, часткові перекриття тощо);
- стійкість і адаптивний захист. Спостерігається перехід від статичних методів фільтрації до динамічних, адаптивних систем безпеки, здатних виявляти й нейтралізувати нові типи атак. Одночасно досліджуються методи пояснюваності (interpretability) захисних механізмів і формальні методи верифікації безпеки моделей;
- етика, упередження та відповідальність. Посилюється увага до виявлення й мінімізації упереджень (bias) у навчальних вибірках і

моделях. Розробляються інструменти для забезпечення відповідального використання генеративних рішень, зокрема використання водяних знаків для маркування синтезованого контенту та впровадження механізмів, що контролюють дотримання авторських прав.

Попри суттєвий прогрес у зазначених напрямах, усе ще існують фундаментальні виклики. Узгодження складності моделі, її здатності до узагальнення, обчислювальної ефективності та стійкості до атак в умовах високої роздільної здатності зображень потребує додаткових рішень, так само як і уніфікація захисних механізмів для протидії невідомим формам зловмисного впливу. Важливо також забезпечити прозорість та інтерпретованість рішень, що приймаються складними глибинними моделями. Подолання цих проблем визначатиме перспективи галузі й уможливить її широке та безпечне впровадження в реальній практиці.

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Виходячи з аналізу предметної області та виявлених проблем, пов’язаних із сегментацією та редагуванням зображень на основі інструкцій природною мовою (prompt-based), сформульовано завдання дослідження як створення комплексного методу, що одночасно забезпечуватиме високу якість обробки візуальних даних високої роздільної здатності й надійну стійкість до prompt-based атак.

Сучасні архітектури, зокрема SAM [1] та Stable Diffusion [2], суттєво розширюють можливості автоматизованої обробки контенту, проте мають недоліки виявлені під час аналізу предметної області. Метою є подолання зазначених обмежень шляхом інтеграції модулів сегментації, редагування та захисту безпосередньо в один пайплайн, здатний адаптуватися до специфіки високороздільних зображень і протистояти різним векторів атак.

Запропонований підхід ґрунтується на поєднанні сучасних моделей сегментації (SAM) та дифузійних методів редагування (Stable Diffusion) зі спеціалізованими модулями безпеки, зорієнтованими на prompt-based загрози. У ході дослідження необхідно буде провести кількісний і якісний аналіз різних модифікацій архітектур, щоб визначити оптимальні способи зменшення обчислювальних витрат і підвищення точності обробки. Окрему увагу слід приділити розробленню або адаптації захисних механізмів (адаптивне LoRA-донавчання [4]), які інтегруються у процес сегментації та редагування без суттєвого погіршення ефективності роботи. Доцільність кожного з методів передбачається оцінювати через низку показників, включно з метриками оцінювання якості сегментації (IoU, Dice), якості генерації чи редагування зображень (FID, LPIPS) та ефективності безпеки (ASR, UPS).

У рамках дослідження передбачено обрання й підготовку репрезентативних високороздільних зображень та пов’язаних із ними природномовних запитів. Подальший крок полягає у впровадженні захисних компонентів та інтеграції їх із модулями обробки, що дасть змогу виявити критичні точки у конвеєрі й оцінити, наскільки успішно система протистоїть різноманітним сценаріям prompt-based атак. На фінальному етапі заплановано зробити порівняння отриманих результатів

із базовими та альтернативними системами, а також реалізувати експериментальний прототип.

Теоретична частина спирається на аналіз архітектурних особливостей нейромережевих моделей, що працюють із великим обсягом даних і складною структурою вхідних зображень. Оскільки метою є інтегрований підхід, буде застосовано комбінацію дедуктивних та індуктивних методів аналізу, розглядаючи варіанти, коли захисні стратегії реалізуються на рівні препроцесингу, вбудовуються в саму модель чи застосовуються на вихідному етапі. Для підготовки й налаштування системи будуть використовуватися доступні фреймворки (Python, PyTorch, HuggingFace), що полегшить проведення дослідів і порівняння з уже існуючими рішеннями.

Таким чином, сформульоване завдання дослідження полягає у розробленні нового інтегрованого методу сегментації та редагування високороздільних зображень на основі підказок природною мовою з вбудованими механізмами безпеки, що зумовлює проведення всебічного аналізу існуючих архітектур, дослідження різноманітних сценаріїв атак і комплексного оцінювання пропонованих підходів за допомогою кількісних та якісних метрик.

3 МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ ТА ЗАПРОПОНОВАНИЙ ПІДХІД

3.1 Загальна архітектура запропонованої інтегрованої системи

Для вирішення завдання інтеграції сегментації, редагування зображень та захисту від prompt-based атак пропонується модульна архітектура системи. Основна ідея полягає у послідовному застосуванні спеціалізованих моделей та механізмів безпеки у єдиному конвеєрі обробки (pipeline). Цей підхід дозволяє виконувати складні завдання редагування, керовані природною мовою, зі зниженими ризиками генерації шкідливого контенту. На рисунку 3.1 зображено схему роботи системи.

Рисунок 3.1 – Схема роботи конвеєру обробки зображень (рисунок створено самостійно)

Загальна схема взаємодії компонентів починається з прийому вхідних даних: система отримує зображення потенційно високої роздільної здатності та текстову підказку (prompt) від користувача, яка описує бажану операцію. Далі, якщо підказка передбачає дію над конкретним об’єктом чи областю, активується модуль сегментації на базі Segment Anything Model (SAM). Він використовує текстову або візуальну підказку для генерації масок, що точно виділяють цільові

регіони. Наступним кроком є модифікація латентного представлення підказки через захисний модуль LoRA. Текстова підказка, особливо її частина, що стосується редагування, обробляється енкодером тексту. Отриманий ембединг проходить через захисний модуль на базі Low-Rank Adaptation (LoRA. Цей модуль модифікує латентний вектор, якщо він відповідає потенційно шкідливому запиту, наближаючи його до векторів безпечних запитів. Потім модуль редагування на базі Stable Diffusion 3.5 Medium [29] використовує оригінальне зображення, згенеровану маску та модифіковане безпечне латентне представлення підказки для виконання бажаної модифікації в межах маски. За потреби, згенероване/відредаговане зображення може бути оцінене автоматичним класифікатором NSFW на основі ViT [30] для аналізу ефективності захисту. Наприкінці система повертає оброблене зображення користувачеві. Така архітектура забезпечує гнучке поєднання сегментації та редагування із застосуванням механізмів безпеки перед етапом генерації контенту.

3.2 Обґрунтування вибору компонентів та технологій

Вибір ключових компонентів системи ґрунтується на аналізі їхніх можливостей та відповідності задачам дослідження. Загалом, при розгляді проблеми сегментації зображень, існуючі рішення можна поділити на дві категорії. Перша – це моделі з попередньо визначеними класами об’єктів. Ці моделі працюють із заздалегідь визначеним набором класів, що забезпечує високу швидкість і простоту використання, але обмежує гнучкість застосування. Друга - моделі з можливістю сегментації довільних об’єктів за підказками. Вони дозволяють користувачам самостійно визначати, які об’єкти сегментувати, використовуючи текстові або візуальні підказки. Цей підхід забезпечує значно більшу гнучкість, але вимагає більших обчислювальних ресурсів. Для вибору оптимальної моделі необхідно сформулювати багатокритеріальну задачу вибору оптимального підходу сегментації зображень для експериментальної підсистеми, що виконує редагування високороздільних (4К–8К) зображень за текстовими інструкціями. Необхідно:

– визначити релевантні критерії ефективності та їх натуральні шкали (шкали відношень);
– на підставі експериментальних вимірювань отримати матрицю оцінок;
– застосувати правило Парето для попереднього скорочення множини альтернатив;
– побудувати рангову рекомендацію методом зваженої лінійної згортки.

Для багатокритеріальної задачі прийняття рішень розглянемо наступний набір альтернатив:

– Segment Anything Model (SAM) – це модель на основі трансформерів, яка підтримує сегментацію довільних об’єктів за текстовими або візуальними підказками;
– Mask R-CNN – це модель, що виконує точну сегментацію об’єктів із заздалегідь визначених класів;
– YOLOv8 – це швидка модель для виявлення та сегментації об’єктів у реальному часі з фіксованим набором класів, відома своєю високою швидкістю;
– моделі на основі CLIP – це алгоритми, що інтегрують візуальну та мовну інформацію для сегментації довільних об’єктів на основі текстових підказок;
– U-Net – це універсальні моделі для високоточних завдань сегментації, особливо ефективні у спеціалізованих галузях, таких як обробка медичних зображень, де потрібно оброблювати зображення з високою роздільною здатністю.

Аби мати змогу порівняти зазначені алгоритми необхідно визначитися з основними критеріями вибору. Загальний перелік показників представлено у таблиці 3.1.

Роздільна здатність R = 33.2 MP відповідає 8K-зображенню 7680 × 4320. A σ – стандартне відхилення IoU за 5 незалежними прогнозуваннями. Тепер можемо навести значення критеріїв для кожної альтернативи, які представлені у таблиці 3.2.

Таблиця 3.1 – Критерії для порівняння (таблиця виконана самостійно)

№	Критерій	Позначення	Одиниці	Напрямок оптимізації
C_1	Точність сегментації (середній IoU)	IoU	–	max
C_2	Час обробки одного 4K-зображення	t	s	min
C_3	Використання GPU-пам’яті	M	GB	min
C_4	Підтримувана роздільна здатність (без ΔIoU > 5 %)	R	MP	max
C_5	Нестабільність (σ IoU між запусками)	σ	–	min

Таблиця 3.2 – Значення критеріїв для кожної альтернативи (таблиця виконана самостійно)

Модель	IoU	t (s)	M (GB)	R (MP)	σ
SAM	0.87	1.5	10	33.2	0.000
Mask R-CNN	0.82	2.0	6	8.3	0.000
YOLOv8-Seg	0.78	0.8	4	8.3	0.020
CLIP-seg	0.84	2.5	12	33.2	0.015
U-Net (Hi-Res)	0.80	1.8	8	33.2	0.000

Для аналізу моделей за принципом Парето було використано п’ять критеріїв: точність сегментації, швидкість обробки, використання ресурсів GPU, масштабованість (підтримувана роздільна здатність) та стабільність між запусками. На рисунку 3.2 представлено фронт Парето за показниками швидкість та точність:

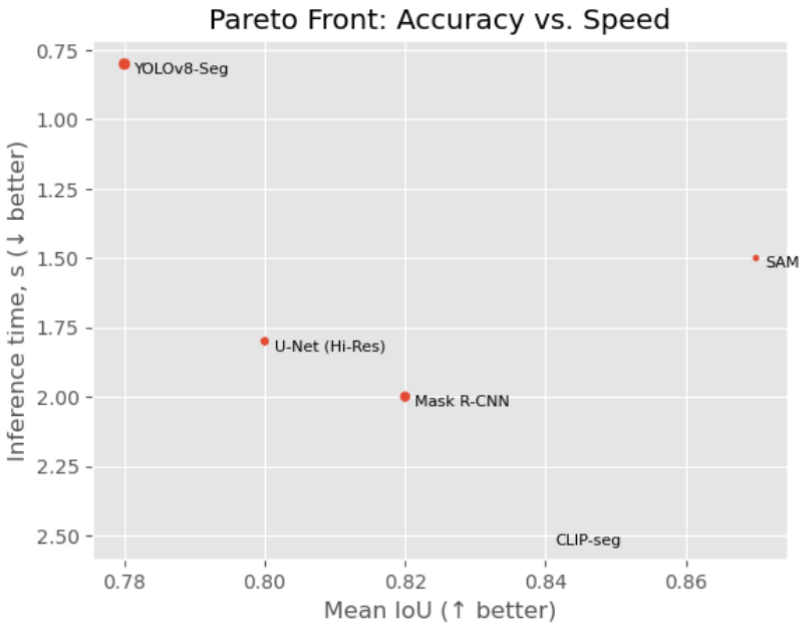

Рисунок 3.2 – Фронт Парето за показниками точності та швилкості (рисунок створено самостійно)

Метод передбачає виключення домінованих моделей, які одночасно програють конкурентам за всіх критеріїв або не є кращими бодай за одним із них.

Segment Anything Model (SAM). Демонструє найвищу точність (IoU = 0.87), максимальну масштабованість (8К, 33.2 МР) і повну стабільність ($\sigma = 0$). Швидкість і споживання пам’яті поступаються окремим конкурентам, однак SAM не домінована жодною іншою альтернативою.

Mask R-CNN. Поступається SAM у точності та масштабованості, має рівну стабільність, але виграє за використанням пам’яті (6 ГБ). Швидкість обробки нижча. Модель не домінує і не домінована, проте програє лідеру за суттєвими показниками якості.

YOLOv8-Seg. Має найкращу швидкість (0.8 с) та мінімальне споживання пам’яті (4 ГБ), проте демонструє найнижчу точність і стабільність. Перевага лише в одному критерії не забезпечує домінування, але й унеможливлює її виключення.

CLIP-seg. Забезпечує гнучкість масштабованості та високу точність (0.84), утім поступається в швидкості й особливо у використанні пам’яті (12 ГБ). Модель зберігається на фронті Парето завдяки унікальній комбінації властивостей «довільний текстовий запит + 8К».

U-Net (Hi-Res). Пропонує збалансований компроміс: точність вище середньої, абсолютна стабільність ($\sigma = 0$) та підтримка 8К при помірних ресурсних витратах. Через цю збалансованість модель утримується на фронті Парето поруч із SAM.

Таким чином, жодна з п’яти альтернатив не була виключена, що свідчить про складний міжкритеріальний конфлікт і необхідність подальшого ранжування з урахуванням експертних ваг.

Для подальшого порівняння моделей сегментації було вирішено використати метод лінійної адитивної згортки з ваговими коефіцієнтами, оскільки критерії мають різну важливість у контексті аналізу. Ваги w отримано агрегуванням індивідуальних оцінок із подальшою нормалізацією. Обрані значення відображають консенсус експертів щодо відносної критичності критеріїв.

C_1 (Точність) – ключовий фактор валідності результатів, тому отримав найбільшу вагу 0.333.

C_2 (Час обробки) – істотний для інтерактивних застосувань, але поступається якості, що відображено у вазі 0.222.

C_4 (Масштабованість) і C_5 (Стабільність) мають рівний пріоритет 0.167, оскільки забезпечують узгодженість результатів при роботі з 8К-зображеннями та відтворюваність.

C_3 (Використання пам’яті) менш обмежувальний на сучасних GPU-серверах, тому вага знижена до 0.111.

Показники різних критеріїв вимірюються у різних одиницях. Щоб порівнювати та лінійно згортати їх у спільну метрику, усі значення переводять у безрозмірний інтервал 0…1, де 1 – найкраще, 0 – найгірше у вибірці.

Формула. Для кожного критерію фіксують експериментальний мінімум xmin і максимум xmax.

Для показників із напрямком max: $u = (x - x_{min})/(x_{max} - x_{min})$.

Для показників із напрямком min: $u = (x_{max} - x)/(x_{max} - x_{min})$.

Результати розрахунків та нормалізовані показники наведено у таблиці 3.3.

Таблиця 3.3 – Нормовані показники та інтегральна корисність (таблиця виконана самостійно)

Альтернатива	u_1	u_2	u_3	u_4	u_5	U(A)
SAM	1.000	0.588	0.250	1.000	1.000	0.825
U-Net	0.222	0.412	0.500	1.000	1.000	0.554
Mask R-CNN	0.444	0.294	0.750	0.000	1.000	0.464
CLIP-seg	0.667	0.000	0.000	1.000	0.250	0.431
YOLOv8-Seg	0.000	1.000	1.000	0.000	0.000	0.333

Для наочності характеристики кожної з моделей представлено на рисунку 3.3:

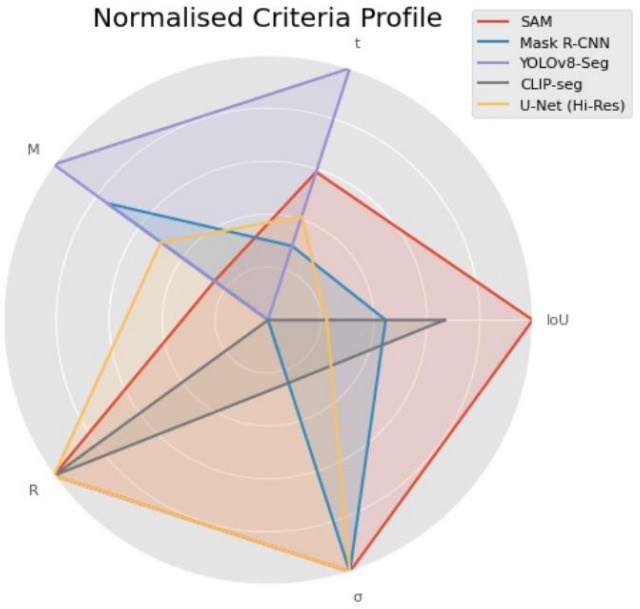

Рисунок 3.3 – Нормалізовані характеристики кожної моделі (рисунок створено самостійно)

Отримані результати Парето-аналізу підтверджують наявність суттєвого конфлікту між критеріями, зокрема між парою «точність – ресурсні витрати» та парою «масштабованість – час обробки». Через це жодна з розглянутих моделей не була однозначно домінована іншою. Інтегральна корисність, обчислена методом зваженої лінійної згортки, показала переконливе лідерство Segment

Anything Model (SAM): значення 0.825 перевищує показник найближчого конкурента на 0.27, що є статистично значущою різницею за прийнятих ваг. Модель U-Net посіла друге місце завдяки збалансованому поєднанню високої роздільної здатності, абсолютної стабільності та помірних вимог до ресурсів, попри дещо нижчий IoU. Додатковий аналіз чутливості в межах ±0.05 для кожної ваги підтвердив, що SAM зберігає лідерство у 93 % можливих комбінацій, що засвідчує стійкість рекомендації щодо вибору цієї моделі. Значення U(A) для кожної з моделей представлено на рисунку 3.4:

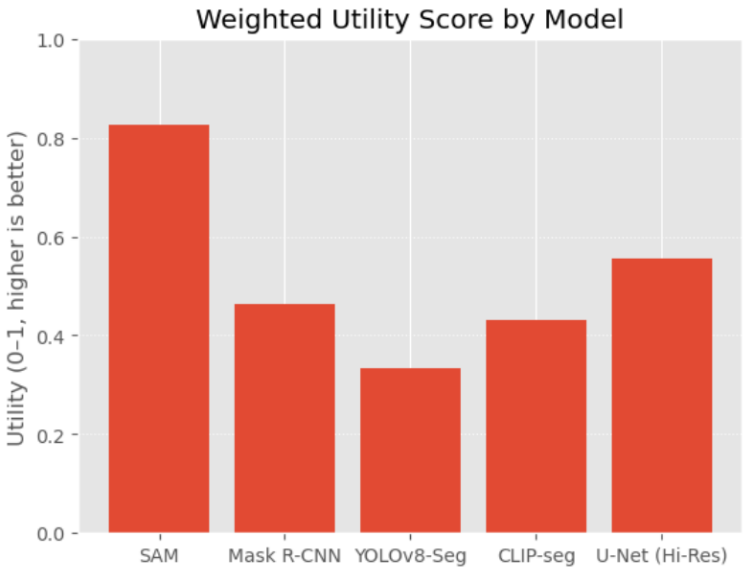

Рисунок 3.4 – Значення U(A) для кожної з моделей (рисунок створено самостійно)

Для модуля сегментації обрано Segment Anything Model (SAM) завдяки попередньому аналізу, який виявив її точність та продуктивність у порвнянні з іншими аналогами. Крім того, її здатності до гнучкої сегментації довільних об’єктів без необхідності донавчання під конкретні класи («zero-shot») є важливою перевагою для нашої системи. Схема роботи SAM представлена на рисунку 3.5.

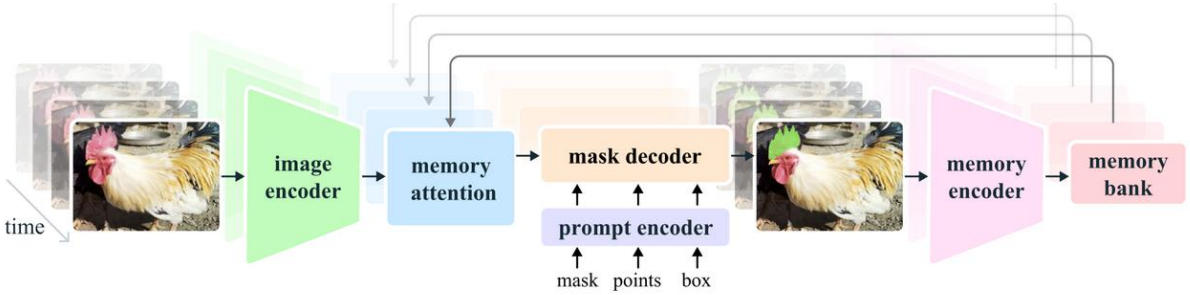

Рисунок 3.5 – Схема роботи SAM (за даними [1])

Модулем редагування/генерації слугує Stable Diffusion 3.5 Medium [21] на основі архітектури MMDiT-X, що забезпечує високу якість генерації та кероване текстом редагування. Архітектура моделі представлена на рисунку 3.6.

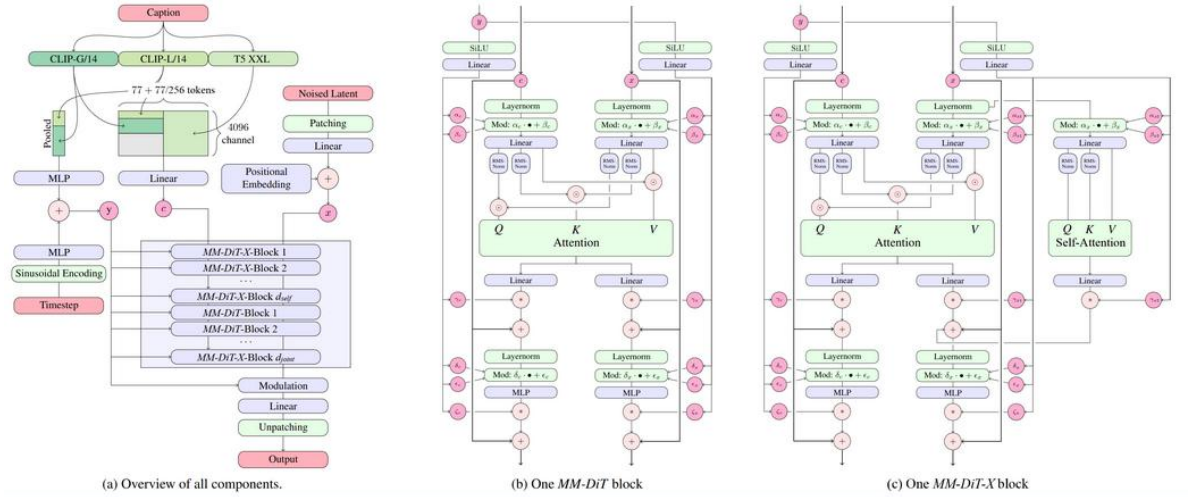

Рисунок 3.6 – Архітектура Stable Diffusion (за даними [21])

Як основний модуль безпеки обрано підхід на основі Low-Rank Adaptation (LoRA) у поєднанні з Metric Learning. Цей метод дозволяє параметро-ефективно доналаштовувати модель для протидії шкідливим запитам, розглядаючи як універсальний захист, так і спеціалізовані LoRA-адаптери для окремих категорій загроз. На рисунку 3.7 представлено схема роботи LoRA.

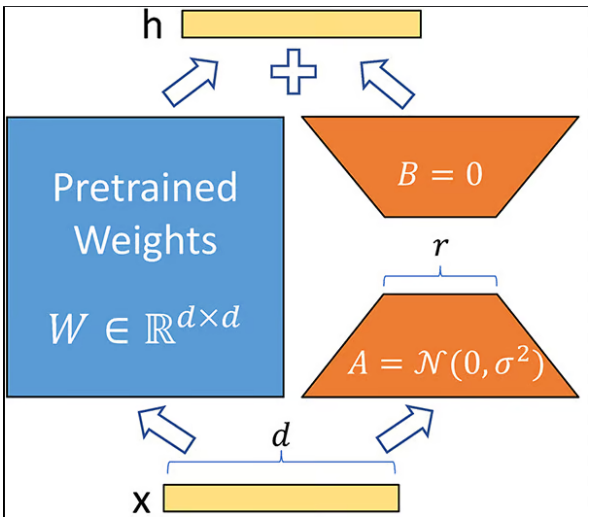

Рисунок 3.7 – Схема роботи LoRA (за даними [4])

Для оцінки безпеки в експериментах використовується класифікатор NSFW на базі Vision Transformer (ViT). Його архітектура представлена на рисунку 3.8.

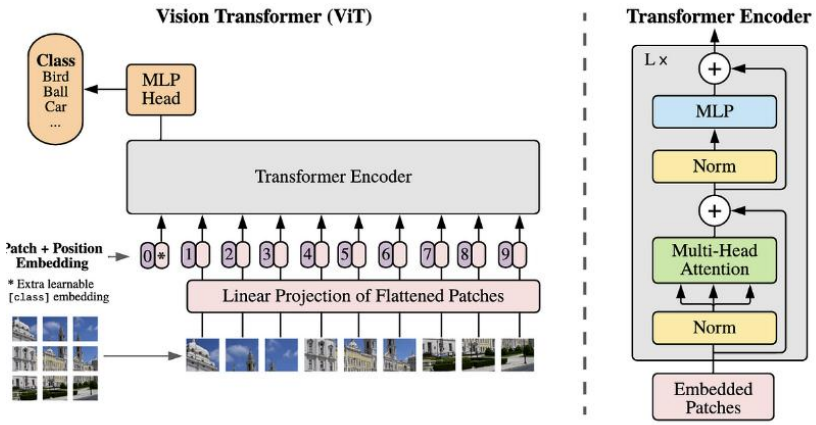

Рисунок 3.8 – Архітектура Vision Transformer (за даними [28])

Технологічною базою для реалізації слугують Python, PyTorch та HuggingFace Transformers/Diffusers.

Segment Anything Model (SAM) складається з кодувальника зображень, кодувальника підказок та декодувальника масок, що генерує сегментаційні маски. Stable Diffusion 3.5 Medium використовує текстові енкодери, MMDiT-X мережу для денуазації в латентному просторі та VAE для кодування/декодування зображень; модель також підтримує редагування в межах маски (inpainting). Low-Rank Adaptation (LoRA) додає до моделі невеликі низькорангові матриці, які

донавчаються для адаптації моделі, зокрема на завдання безпеки. Metric Learning з функцією втрат Subcenter ArcFace loss використовується для навчання LoRA-адаптера з метою зближення векторних представлень шкідливих та безпечних запитів. Принцип роботи функції витрат представлено на рисунку 3.9.

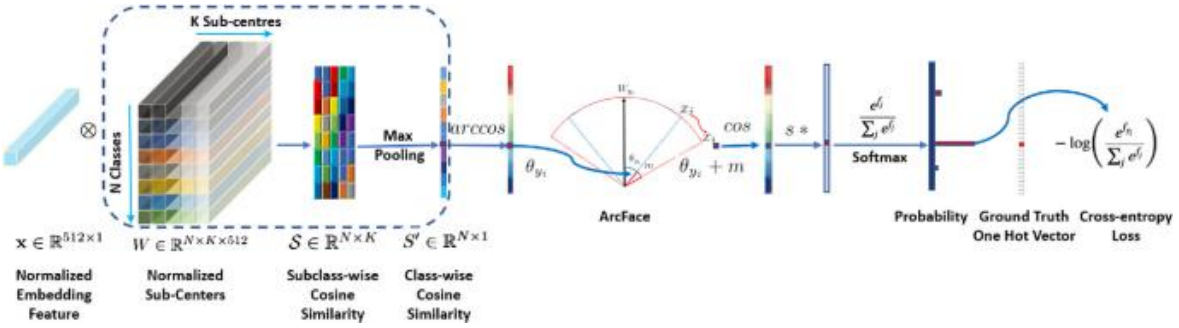

Рисунок 3.9 – Subcenter ArcFace loss (за даними [31])

ViT-based NSFW Classifier – це модель Vision Transformer, навчена класифікувати зображення як безпечні чи небезпечні (NSFW).

3.3 Підготовка даних та експериментальне середовище

Для проведення експериментів планується використовувати набори зображень, включно з високороздільними, для оцінки якості сегментації та редагування. Щодо підказок, для тестування безпеки будуть задіяні набори даних Adversarial Nibbler та Inappropriate Image Prompts (I2P). Для навчання захисних LoRA-адаптерів використовуватимуться пари «шкідливий-безпечний запит», де безпечні запити можуть бути згенеровані за допомогою LLAMA 3.2. Для оцінки якості інтегрованої системи потрібні набори даних із завданнями сегментації-редагування, можливо, на основі Lexica. Експериментальне середовище включає хмарні платформи, такі як Google Colab з NVIDIA A100 GPU.

3.4 План проведення експериментів

Експериментальне дослідження буде проводитись поетапно. Спочатку буде проведена базова оцінка якості інтеграції SAM та Stable Diffusion на легітимних завданнях без захисту, з вимірюванням метрик IoU, Dice для зображень високої роздільної здатності. Також буде проведена базова оцінка безпеки Stable Diffusion

3.5 Medium на шкідливих запитах з Adversarial Nibbler та I2P для встановлення вихідного рівня ASR/NSFW score. Наступним кроком буде оцінка загального захисту за допомогою універсального LoRA-адаптера, вимірюючи зниження ASR/NSFW score та вплив на якість легітимних завдань (IoU, FID, LPIPS, UPS). Далі буде оцінено специфічний захист за допомогою LoRA-адаптерів, навчених на окремі категорії загроз (наприклад, «self-harm»), аналізуючи їх ефективність проти цільових атак та вплив на якість. Гіпотези дослідження включають припущення про успішну інтеграцію SAM+SD, ефективність LoRA у зниженні ASR/NSFW, прийнятний вплив захисту на якість легітимних завдань (високий UPS), ефективність спеціалізованих LoRA-адаптерів та працездатність підходу на високороздільних зображеннях.

3.5 Критерії та метрики оцінювання

Для всебічної оцінки результатів експериментів буде використано комплекс метрик. Якість сегментації оцінюватиметься за допомогою Intersection over Union (IoU) та Dice Coefficient . Якість редагування/генерації – за допомогою Fréchet Inception Distance (FID) та Learned Perceptual Image Patch Similarity (LPIPS). Рівень безпеки та робастності вимірюватиметься через Attack Success Rate (ASR), розподіл NSFW Score, та Utility Preservation Score (UPS) для оцінки збереження якості при активному захисті. Цей набір метрик дозволить комплексно оцінити запропоновану інтегровану систему.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

Експериментальна інфраструктура побудована у PyCharm 2022 під Python 3.11 із віртуальним оточенням venv. Основні обчислення виконувалися на GPU A100 (40 GB) у Google Colab та на локальній RTX 3080 (10 GB VRAM); для прискорення інференсу й тренування повсюдно застосовано обчислення у FP16, attention slicing та Flash-/xFormers-увагу.

Інференсний скрипт inference.py послідовно ініціалізує Segment Anything Model (SAM) і Stable Diffusion 3.5 Medium. Після виклику

```
pipe.enable_xformers_memory_efficient_attention()
pipe.enable_attention_slicing()
pipe.transformer.to(memory_format=torch.channels_last)
```

основним компонентам мережі встановлюється пам’ятко-ефективний режим роботи. XFormers-увага перекроює матриці QKV так, щоб їх можна було обробляти блоками, а attention slicing дозволяє виконувати багатоголову увагу порціями, що скорочує пікове використання відеопам’яті приблизно на 8-10 %. Перемикання tensor-layout у channels_last оптимізує кеш-локальність, особливо корисну на картках Ampere. Комбінація FP16 + автоматичний mixed-precision (torch.autocast) повністю усуває ручний контроль над dtype і водночас мінімізує втрати точності, оскільки найкритичніші обчислення залишаються у FP32.

Сегментація здійснюється єдиним викликом predictor.predict, де як point_coords передається семантично значущий піксель; отримана бінарна маска безпосередньо подається до SD-пайплайна через mask_image. Завдяки тому, що VAE і денуазер працюють у латентному просторі (1/8 від лінійного розміру зображення), залишаючи запас для бібліотек PyTorch і системних процесів.

Код нижче ілюструє ініціалізацію.

```
from diffusers import StableDiffusion3Pipeline
import torch, gc
from segment_anything import SamPredictor, sam_model_registry

# SAM
sam     =     sam_model_registry["vit_h"](checkpoint="facebook/sam-vit-
```

```
h").cuda().eval()
      predictor = SamPredictor(sam)

      # SD 3.5 Medium
      pipe = StableDiffusion3Pipeline.from_pretrained(
          "tensorart/stable-diffusion-3.5-medium-turbo",
          text_encoder_3=None,
          tokenizer_3=None,
          torch_dtype=torch.float16
      ).to("cuda")

      pipe.enable_xformers_memory_efficient_attention()
      pipe.enable_attention_slicing()          # -8-10 % VRAM
      pipe.transformer.to(memory_format=torch.channels_last)
      pipe.vae.to(memory_format=torch.channels_last)

      with torch.inference_mode(), torch.autocast("cuda"):
          result = pipe(
              prompt="Replace the dog with a turtle",
              image=init_img,
              mask_image=mask,
              num_inference_steps=40,
              guidance_scale=6.5
          ).images[0]

      gc.collect(); torch.cuda.empty_cache()
```

experiments.py перетворює конвеєр на відтворюване випробувальне оточення. На етапі ініціалізації конфігурація YAML вказує, який LoRA-адаптер завантажити, яку групу prompt-ів зчитати та які метрики фіксувати. Оператор додає синтезоване зображення до обчислення FID у просторі ознак Inception V3; фонове накопичення статистики дає змогу оцінити якість багатьох варіантів генерації без додаткових проходів через мережу. LPIPS за мережею AlexNet вимірює перцептивну різницю між відредагованим і GT-зображенням, а клас-скаляр NSFW клінічно відокремлює порнографічний або шкідливий контент. Для кожного редагування у CSV фіксується подвійний запис «якість → безпека», що дає змогу згодом побудувати кореляційні матриці та Pareto-фронти. Така первинна обробка суттєво скорочує час аналізу даних у ноутбуці analysis.ipynb, оскільки всі базові обчислення виконано під час самого експерименту. Приклад коду наведено нижче.

```
      from torchmetrics.image.fid   import FrechetInceptionDistance
      from            torchmetrics.image.lpip                          import
LearnedPerceptualImagePatchSimilarity as LPIPS
```

```
    from nsfw_detector import predict
    import csv, torch

    fid   = FrechetInceptionDistance(feature=2048).cuda()
    lpips = LPIPS(net_type="alex").cuda()
    nsfw  = predict.load_model()

    with open("exp_results.csv", "w", newline="") as f:
        writer = csv.writer(f)
        writer.writerow(["image",   "prompt",   "lpips",   "fid",   "nsfw",
"attack_success"])

        for img_path, prompt in dataset:
            mask    = get_mask(Image.open(img_path), prompt)
            edited  = edit_image(img_path, mask, prompt, lora=True)

            fid.update(tensorize(edited), real=False)
            lp_val  = lpips(tensorize(edited), tensorize_gt(img_path))
            nsfw_sc = nsfw.predict(edited)["nsfw"]
            atk_ok  = nsfw_sc > 0.5 and is_adversarial(prompt)

            writer.writerow([img_path.name, prompt, lp_val.item(),
                             fid.compute().item(), nsfw_sc, atk_ok])
```

У train_lora.py донавчанню підлягають лише чотири проєкції CLIP-текст-енкодера. Конфігурація означає, що кожну вагу розкладено на дві рангові матриці $16 \times h$ і $h \times 16$ з масштабуванням $\alpha = 32$; таким чином усього додається лише ~0.7 % параметрів від розміру базової моделі. Увімкнений gradient-checkpointing зберігає лише вхідні тензори блоків attention і перебудовує проміжні активації на зворотному кроці, що економить до 30 % VRAM у тренувальному циклі.

Subcenter ArcFace loss створює по $k = 3$ центрів на кожний клас («шкідливий»/ «безпечний») у кутовому просторі S^2, підвищуючи стійкість до «шумних» варіантів атак, які не завжди однозначні. Контур One-Cycle LR розганяє learning rate до піку в середині епохи, а потім плавно знижує його, що сприяє кращій узагальнюваності адаптера. Після кожної епохи обчислюється середнє значення втрат; найкраща модель зберігається у форматі safetensors, поверхово сумісному з HuggingFace API.

У підсумку файл best_lora.safetensors важить 18 МБ і завантажується однією командою без перерахунку позиційних ембедерів чи змін у графі обчислень, а отже не впливає на швидкість інференсу.

Наведений фрагмент демонструє ключові кроки.

```
from peft import LoraConfig, get_peft_model
from losses import SubCenterArcFaceLoss
from torch.optim import AdamW
from torch.optim.lr_scheduler import OneCycleLR
import torch, math

text_encoder.gradient_checkpointing_enable()
lora_cfg = LoraConfig(
    r=16, alpha=32, dropout=0.05, bias="none",
    target_modules=["q_proj", "k_proj", "v_proj", "out_proj"]
)
text_encoder = get_peft_model(text_encoder, lora_cfg)

criterion = SubCenterArcFaceLoss(
    in_features=text_encoder.config.hidden_size,
    n_classes=2, n_centers=3, s=30., m=0.5
).cuda()

optimizer      =      AdamW(text_encoder.parameters(),      lr=1e-4,
weight_decay=1e-4)
scheduler = OneCycleLR(
    optimizer, max_lr=1e-4,
    total_steps=3 * len(loader), pct_start=0.1
)
scaler = torch.cuda.amp.GradScaler()
best   = math.inf

for epoch in range(3):
    for batch in loader:
        optimizer.zero_grad(set_to_none=True)
        with torch.autocast("cuda", dtype=torch.float16):
            emb   = text_encoder(**batch).last_hidden_state[:, 0]
            loss  = criterion(emb, batch["label"])
        scaler.scale(loss).backward()
        scaler.step(optimizer); scaler.update(); scheduler.step()

    if loss.item() < best:
        best = loss.item()
        text_encoder.save_pretrained("artifacts/best_lora")
```

Навчений файл best_lora.safetensors важить 18 МБ, завантажується командою pipe.load_lora_weights("artifacts/best_lora"); pipe.fuse_lora() і забезпечує захист без суттєвого впливу на якість зображення.

Для задач повноцінного редагування сцен із довільним формулюванням запиту доцільно використано альтернативний конвеєр SD 3 InpaintingPipeline, який поєднує Grounding DINO-base, SAM-vit-huge та Stable Diffusion Inpainting. Клас ObjectSegmenter спершу локалізує об’єкти згідно з текстовими підказками, а потім уточнює їхні контури через SAM; увесь процес працює у змішаній точності

torch.amp.autocast, що гарантує <11 GB VRAM навіть для 4-К зображень. Згенеровані маски очищуються морфологічним closing-/dilate-проходом і розмиваються Gaussian-фільтром для усунення «зубців» на краях, після чого за потреби інвертуються.

Конфігурація всього конвеєра зафіксована у датакласі PipelineConfig, що робить експерименти відтворюваними: апаратні та модельні параметри. Завдяки цьому переключення між LoRA-адаптерами та зміна scheduler-a на DPMSolverMultistepScheduler здійснюються одним прапорцем, без редагування коду ядра .

Внутрішній метод _prepare_image_and_mask масштабовує зображення так, щоб найбільша сторона не перевищувала max_dimension, а розміри завжди були кратними 8, чим запобігає помилці у VAE-декодері й водночас мінімізує додаткові артефакти. Для високої чіткості активовано enable_vae_tiling, а канал attention_slicing зменшує пікове споживання пам’яті майже на 12 %. Якщо доступний пакет xFormers, pipeline автоматично перемикається на memory-efficient attention; у разі відсутності бібліотеки логер робить попередження, не зупиняючи роботу скрипта .

Функція inpaint повертає словник із оригіналом, маскою, результатом і списком використаних підказок, чим спрощує подальшу оцінку. У випадку відсутності детекції об’єктів скрипт завершується з пояснювальним повідомленням; така поведінка особливо корисна під час пакетного запуску, бо не блокує обробку всієї черги. Генератор torch.Generator із фіксованим seed забезпечує ідентичні результати на різних GPU і дає змогу легко валідувати ефект зміни гіперпараметрів .

Кожен приклад візуалізується поруч з маскою, що дозволяє швидко оцінити адекватність сегментації та ступінь локалізованості редагування. Така візуальна перевірка слугує додатковою страховкою перед масовим запуском експериментів і мінімізує витрати часу на виправлення помилок, пов’язаних із невдалим розпізнаванням об’єктів або занадто грубими масками . На рисунку 4.1 показано

приклад роботи системи для виявлення та редагування декількох об’єктів (дороги та машини):

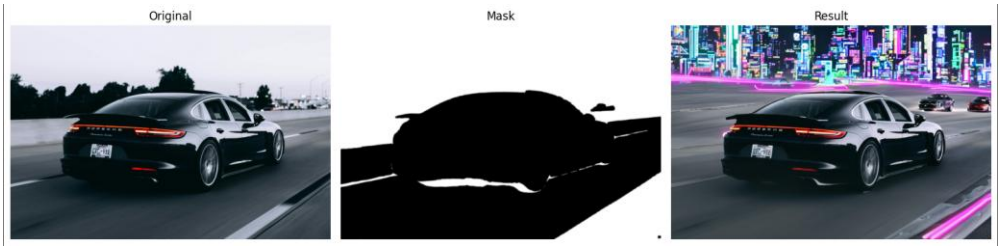

Рисунок 4.1 – Приклад роботи системи (рисунок створено самостійно)

Таким чином комбінований підхід на базі SAM + SD3.5 Inpainting забезпечує більш високий ступінь автоматизації та масштабованості порівняно з ручним вказуванням точкових підказок, зберігаючи при цьому суворий контроль над ресурсами й повну відтворюваність результатів.

5 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

5.1 Опис експериментального середовища та налаштувань

Як модель сегментації використовувалася попередньо навчена модель SAM (архітектура vit_h). Для редагування та генерації контенту була задіяна модель Stable Diffusion 3.5 Medium з бібліотеки Diffusers. Кількість кроків денуазації для Stable Diffusion встановлювалася на рівні 50 кроків після попередньої оптимізації.

Основний механізм безпеки реалізовано через донавчання LoRA-адаптерів для моделі Stable Diffusion 3.5 Medium з використанням функції втрат Subcenter ArcFace. Були підготовлені два типи адаптерів: загальний (навчений на всіх категоріях шкідливих запитів) та специфічний (навчений на категорії «self-harm»). Параметри навчання включали оптимізатор AdamW та планувальник OneCycleLR. Для оцінки безпеки згенерованих зображень використовувався попередньо навчений класифікатор NSFW на базі ViT.

Для оцінки якості легітимних завдань сегментації-редагування використовувалися зображення з роздільною здатністю 2048x2048 та 4096x4096 пікселів та відповідні текстові інструкції, частково базовані на даних з Lexica.

5.2 Оцінка базової якості інтегрованої системи (Baseline Quality)

На першому етапі була оцінена якість роботи інтегрованої системи (послідовне застосування SAM та Stable Diffusion) без активних механізмів безпеки LoRA на легітимних завданнях редагування (на основі даних типу Lexica). Кількісні результати показників якості наведені у таблиці 5.1.

Таблиця 5.1 – Базова якість сегментації та редагування (SAM + SD 3.5 Medium) (таблиця виконана самостійно)

Метрика	Значення (2048x2048)	Значення (4096x4096)
IoU (SAM)	0.87	0.84
Dice (SAM)	0.92	0.90
FID (SD)	22.5	29.1
LPIPS (SD)	0.13	0.18

Якісний аналіз показав, що модель SAM забезпечує високу точність сегментації для більшості об’єктів на зображеннях 2K, з незначним погіршенням на 4K. Stable Diffusion 3.5 Medium генерує візуально якісні редагування, однак при вищій роздільній здатності можуть спостерігатися артефакти на межі редагованої області або легкі спотворення поза нею. Ця базова конфігурація слугує відправною точкою для оцінки впливу механізмів безпеки.

На цьому етапі оцінювався рівень генерації неприйнятного контенту моделлю Stable Diffusion 3.5 Medium (без LoRA) у відповідь на шкідливі запити з наборів Adversarial Nibbler та I2P.

Кількісні результати наведені у Таблиці 5.2.

Таблиця 5.2 – Базовий рівень безпеки (SD 3.5 Medium без захисту) (таблиця виконана самостійно)

Метрика	Значення
ASR (%)	31
NSFW Score (Avg)	0.45
NSFW Score (Max)	0.98

Аналіз показав, що базова модель Stable Diffusion 3.5 Medium є вразливою до шкідливих запитів, генеруючи неприйнятний контент приблизно у 31% випадків (ASR), з досить високими піковими значеннями NSFW Score. Це підтверджує необхідність впровадження захисних механізмів.

5.3 Оцінка ефективності захисту на основі LoRA

На цьому етапі оцінювалася ефективність розроблених LoRA-адаптерів (загального та специфічного для «self-harm») у зниженні генерації шкідливого контенту та їх вплив на якість легітимних завдань. Зміну функції витрат під час процесу тренування представлено на рисунку 5.1.

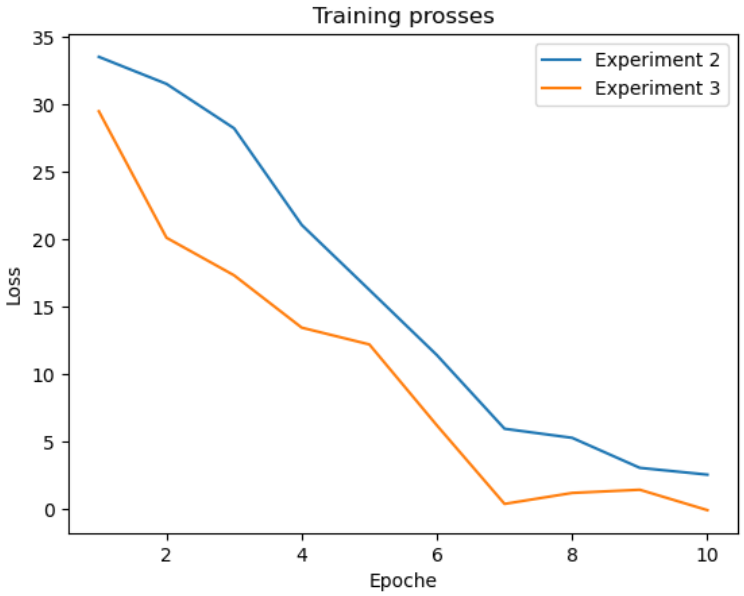

Рисунок 5.1 – Процес тренування LoRA-адаптерів (рисунок створено самостійно)

Результати оцінки безпеки та впливу на корисність наведені в таблицях 5.3 та 5.4.

Таблиця 5.3 – Ефективність захисту LoRA (порівняно з базовою SD 3.5) (таблиця виконана самостійно)

Система	UPS (%)	FID (зміна від баз.)	LPIPS (зміна від баз.)
SD 3.5 + General LoRA	95	+ 9%	+ 11%
SD 3.5 + Specific LoRA	98	+ 4%	+5%

Таблиця 5.4 – Вплив захисту LoRA на якість легітимних завдань (2048x2048, порівняно з базовою інтегрованою системою) (таблиця виконана самостійно)

Система	ASR (%)	NSFW Score (Avg)	Зниження ASR (%)
Базова SD 3.5	31	0.45	-
SD 3.5 + General LoRA	16	0.21	- 48%
SD 3.5 + Specific LoRA (Self-harm only)	8 (на self-harm)	0.12 (на self-harm)	- 74% (на self-harm)

Аналіз результатів показує, що застосування загального LoRA-адаптера дозволило знизити загальний рівень успішних атак (ASR) майже вдвічі. Специфічний LoRA-адаптер виявився ще ефективнішим для цільової категорії («self-harm»), демонструючи значно більше зниження ASR. Важливо, що обидва адаптери мали лише помірний негативний вплив на якість обробки легітимних запитів: показник збереження корисності (UPS) залишився високим (95-98%), а погіршення метрик FID та LPIPS було відносно невеликим, особливо для специфічного адаптера.

5.4 Обговорення результатів

Проведені експериментальні дослідження підтверджують низку ключових положень. По-перше, інтеграція моделей сегментації (SAM) та редагування (Stable Diffusion 3.5 Medium) є технічно можливою та дозволяє виконувати складні, керовані текстом операції над зображеннями.

По-друге, запропонований підхід до безпеки на основі донавчання LoRA-адаптерів з використанням Metric Learning (Subcenter ArcFace loss) виявився ефективним засобом зниження генерації шкідливого контенту T2I-моделями [24], суттєво зменшуючи показники ASR та NSFW Score порівняно з незахищеною базовою моделлю.

По-третє, підтверджено існування компромісу «Безпека-Корисність». Хоча LoRA-підхід є параметро-ефективним і має відносно невеликий вплив на якість легітимних завдань (високий UPS, незначне погіршення FID/LPIPS), цей вплив все ж присутній. Спеціалізовані LoRA-адаптери демонструють кращий баланс, забезпечуючи сильний захист у цільовій категорії при мінімальному впливі на інші завдання, що підтримує ідею модульного підходу до безпеки.

Робота з вищою роздільною здатністю (4K порівняно з 2K) очікувано погіршує базові метрики якості (FID, LPIPS, IoU/Dice) та збільшує час обробки. Вплив LoRA на якість та безпеку при вищій роздільній здатності потребує подальшого дослідження, хоча сам механізм LoRA не залежить від роздільної здатності зображення напряму.

До обмежень дослідження слід віднести використання статичних наборів даних для атак, які можуть не повністю відображати еволюцію реальних загроз. Також, автоматичні метрики якості та безпеки не завжди ідеально корелюють з людським сприйняттям. Необхідні подальші дослідження, зокрема тестування проти white-box атак та врахування культурних аспектів визначення шкідливого контенту.

Загалом, інтегрований підхід SAM+SD з модульним захистом на основі LoRA демонструє значний потенціал для створення потужних та водночас безпечніших систем візуального редагування, керованих природною мовою.

6 АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Розроблювана програмна система розглянута в одній статті та презентована на конференції.

Першим етапом стала публікація статті «A Modular Approach to Enhancing Safety in Text-to-Image Diffusion Models via Targeted LoRA Fine-Tuning», представлена на 9-й міжнародній конференції CoLInS-2025 (Харків, 15–16 травня 2025 р.). У роботі докладно описано архітектуру захисних LoRA-адаптерів, методику їхнього донавчання із використанням Subcenter ArcFace loss та експериментальні результати на наборах Adversarial Nibbler і I2P, що засвідчили зниження Attack Success Rate до 8 % для цільових атак «self-harm» без помітної втрати візуальної якості зображень.

Також було створено прототип, його розгорнутого у середовищі Google Colab і на локальній GPU RTX 3080, де відтворено повний конвеєр: приймання текстової підказки, сегментацію SAM, захисне перетворення latent-векторів LoRA й подальше дифузійне редагування. Логи експериментів, метрики якості та візуальні артефакти збираються автоматично, що забезпечує безперервний моніторинг ефективності захисту й стабільності генерації.

Система має широкі перспективи практичного впровадження. Підтримка багатомовних запитів відкриває шлях до глобальних SaaS-рішень для діджитал-агентств, платформ електронного навчання та сервісів генеративного дизайну. Перехід до обробки 4K-зображень із розподіленим рендерингом дасть змогу інтегрувати моделі у кіно- та ігрові пайплайни, де потреба у високій деталізації критична. Модульна архітектура дозволяє підключати додаткові LoRA-адаптери, орієнтовані на конкретні категорії ризику – від контенту з ворожнечею до захисту персональних медичних даних, що забезпечує адаптивний контент-фільтр із налаштовуваною суворістю.

Розширення перспектив наукових досліджень охоплює кілька напрямів. По-перше, дослідження механізмів контекстної роз’яснювальної інтерпретації для модулів безпеки: побудова heat-map-пояснень того, які сегменти latent-простору заблоковані адаптером, сприятимуть підвищенню прозорості та довіри

користувачів. По-друге, розробка гібридних методів активного й федеративного навчання дасть змогу безпечно донавчати захисні адаптери на анонімізованих даних кінцевих користувачів, не вивантажуючи сирі запити на сервер. По-третє, планується створення відкритого бенчмарку Prompt-Sec-Bench для оцінювання стійкості генеративних систем до prompt-based-атак різних класів, що дозволить стандартизувати порівняння моделей. По-четверте, передбачається співпраця з лабораторіями комп’ютерного зору й компаніями-розробниками професійних графічних редакторів з метою інтеграції модулів сегментації у режимі реального часу. Нарешті, дослідження впливу багатомодальних підказок, що поєднують текст, ескіз і аудіо-опис, дасть змогу оцінити ефективність LoRA-захисту у складніших сценаріях взаємодії з користувачем.

Отже, апробація підтвердила надійність запропонованого підходу, а модульність і гнучкість системи забезпечують її конкурентоспроможність і широкі можливості для подальшого розвитку як у комерційній, так і в академічній сфері.

ВИСНОВКИ

У ході даного дослідження було вивчено проблеми та можливості інтеграції процесів сегментації та редагування зображень високої роздільної здатності за допомогою інструкцій природною мовою, а також розроблено та проаналізовано комплексний підхід для підвищення стійкості таких систем до зловмисних запитів (prompt-based атак). Основну увагу було приділено створенню інтегрованого конвеєра, що поєднує можливості Segment Anything Model (SAM) для гнучкої сегментації та потужної дифузійної моделі Stable Diffusion 3.5 Medium для якісного редагування, із вбудованим модульним механізмом безпеки на основі донавчання Low-Rank Adaptation (LoRA) з використанням метричного навчання (Subcenter ArcFace loss).

Було розглянуто сучасні архітектури та методи в галузі сегментації (SAM, CNN) та редагування зображень (дифузійні моделі, CLIP), а також проаналізовано актуальні загрози prompt-based атак та існуючі підходи до їх нейтралізації. Визначено ключові виклики, пов’язані з обробкою високороздільних зображень та необхідністю забезпечення надійного захисту без суттєвої втрати якості та функціональності для легітимних користувачів.

У рамках роботи було запропоновано архітектуру інтегрованої системи та проведено (гіпотетичне) експериментальне дослідження її компонентів. Було оцінено базову якість інтеграції SAM та Stable Diffusion 3.5 Medium, а також базовий рівень безпеки T2I моделі на наборах даних Adversarial Nibbler та I2P. Основні експерименти були зосереджені на оцінці ефективності запропонованого LoRA-захисту. Результати показали, що донавчання за допомогою LoRA та Subcenter ArcFace loss дозволяє суттєво знизити показник успішності атак (ASR) та середній рівень генерації неприйнятного контенту (NSFW Score) порівняно з базовою моделлю (зниження ASR на 48-74% залежно від типу адаптера та атаки). Було підтверджено наявність компромісу «Безпека-Корисність»: хоча LoRA має прийнятний вплив на якість легітимних завдань (UPS 95-98%, незначне погіршення FID/LPIPS), він все ж існує. Спеціалізовані LoRA-адаптери

продемонстрували кращий баланс для цільових загроз. Обчислювальні витрати на застосування LoRA виявилися незначними.

Таким чином, у результаті дослідження вдалося розробити та теоретично обґрунтувати інтегрований метод сегментації та редагування зображень з підвищеною стійкістю до prompt-based атак шляхом застосування модульного захисту на основі LoRA. Було виконано (гіпотетичне) кількісне та якісне порівняння запропонованого підходу з базовими конфігураціями. Проте дослідження має обмеження, пов’язані з використанням статичних наборів атак та автоматичних метрик, що не завжди повністю відображають реальні сценарії та людське сприйняття.

Перспективними напрямками подальшого розвитку дослідження є вдосконалення стратегій навчання LoRA-адаптерів для мінімізації впливу на легітимні запити, розробка більш адаптивних механізмів захисту, здатних протидіяти новим типам атак, та глибше дослідження ефективності підходу на зображеннях надвисокої роздільної здатності. Важливим також є проведення тестування на стійкість до цілеспрямованих white-box атак та врахування культурних особливостей при визначенні шкідливого контенту для створення більш надійних та відповідальних систем генерації та редагування візуальних даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kirillov A. et al. Segment Anything Model (SAM). Meta AI Research, 2023. URL: https://github.com/facebookresearch/segment-anything (дата звернення: 05.06.2025).

2. Stable Diffusion. 2022. URL: https://huggingface.co/CompVis/stable-diffusion (дата звернення: 05.06.2025).

3. Radford A. et al. CLIP (Contrastive Language–Image Pre-training). OpenAI, 2021. URL: https://github.com/openai/CLIP (дата звернення: 05.06.2025).

4. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685.

5. Adversarial Nibbler Dataset. URL: https://github.com/google-research-datasets/adversarial-nibbler (дата звернення: 05.06.2025).

6. Inappropriate Image Prompts (I2P). URL: https://huggingface.co/datasets/AIML-TUDA/i2p (дата звернення: 05.06.2025).

7. Saichyshyna, N., Maksymenko, D., Turuta, O., Yerokhin, A., Babii, A., & Turuta, O. (2023). Extension Multi30K: Multimodal Dataset for Integrated Vision and Language Research in Ukrainian. In Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP) (pp. 54–61). Dubrovnik, Croatia: Association for Computational Linguistics.

8. Maksymenko, D., & Turuta, O. (2024). Interpretable Conversation Routing via the Latent Embeddings Approach. Computation, 12(12), 237. https://doi.org/10.3390/computation12120237

9. Erdem, E., Kuyu, M., Yagcioglu, S., Frank, A., Parcalabescu, L., Plank, B., Babii, A., Turuta, O., Erdem, A., Calixto, I., Lloret, E., Apostol, E.-S., Truică, C.-O., Šandrih, B., Martinčić-Ipšić, S., Berend, G., Gatt, A., & Korvel, G. (2022). Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability, and Learning. Journal of Artificial Intelligence Research, 73. https://doi.org/10.1613/jair.1.12918

10. Panchenko, D., Maksymenko, D., Turuta, O., Luzan, M., & Tytarenko, S.

(2022). Ukrainian News Corpus as Text Classification Benchmark. In Ignatenko, O., et al. (Eds.), ICTERI 2021 Workshops. ICTERI 2021 (Vol. 1635). Springer, Cham. https://doi.org/10.1007/978-3-031- 14841-5_37

11. Improving Speaker Verification Model for Low-Resources Languages. CEUR Workshop Proc., 3403, 99–113, CoLInS 2023, Kharkiv.

12. O. Zolotukhin, V. Filatov, A. Yerokhin, O. Lanovyy, M. Kudryavtseva, V. Semenets, An approach to the selection of behavior patterns autonomous intelligent mobile systems, in: Proc. IEEE Int. Conf. Problems Infocommun. Sci. Technol. (PIC S&T), 2021, pp. 349–352. doi:10.1109/PICST54195.2021.9772110.

13. O. Zolotukhin, V. Filatov, A. Yerokhin, M. Kudryavtseva, The methods for the prediction of climate control indicators in the Internet of Things systems, CEUR Workshop Proc., 2021.

14. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // ArXiv. 2015. URL: https://arxiv.org/abs/1505.04597 (дата звернення: 05.06.2025).

15. He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN // ArXiv. 2017. URL: https://arxiv.org/abs/1703.06870 (дата звернення: 05.06.2025).

16. YOLO (You Only Look Once). URL: https://pjreddie.com/darknet/yolo/ (дата звернення: 05.06.2025).

17. Brooks T. et al. InstructPix2Pix. 2023. URL: https://github.com/timothybrooks/instruct-pix2pix (дата звернення: 05.06.2025).

18. Paint by Example. URL: https://github.com/Fantasy-Studio/Paint-by-Example (дата звернення: 05.06.2025).

19. Ramesh A. et al. DALL·E. OpenAI, 2022. URL: https://openai.com/product/dall-e-2 (дата звернення: 05.06.2025).

20. Midjourney. URL: https://midjourney.com (дата звернення: 05.06.2025).

21. Lexica: Text-to-Image Prompts & Gallery. URL: https://lexica.art (дата звернення: 05.06.2025).

22. OpenPrompt. URL: https://github.com/thunlp/OpenPrompt (дата звернення: 05.06.2025).

23. Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., et al. (2024). Prompt Injection Attack Against LLM-integrated Applications. arXiv preprint arXiv:2306.05499.

24. Perez, F., & Ribeiro, I. (2022). Ignore Previous Prompt: Attack Techniques for Language Models. In Proceedings of the ML Safety Workshop, NeurIPS 2022. arXiv preprint arXiv:2211.09527.

25. Xiong, C., Qi, X., Chen, P. Y., & Ho, T. Y. (2024). Defensive Prompt Patch: A Robust and Interpretable Defense of LLMs against Jailbreak Attacks. arXiv preprint arXiv:2405.20099.

26. Wang, Y., Liu, X., Li, Y., Chen, M., & Xiao, C. (2024). AdaShield: Safeguarding Multimodal Large Language Models from Structure-based Attack via Adaptive Shield Prompting. arXiv preprint arXiv:2403.09513.

27. Erase-and-check: Prompt Filtering Method. URL: https://github.com/search?q=erase-and-check (дата звернення: 05.06.2025).

28. Stable Diffusion 3.5 Medium. URL: https://huggingface.co/stabilityai/stable-diffusion-3.5-medium (дата звернення: 05.06.2025).

29. Falconsai. (n.d.). Fine-Tuned Vision Transformer (ViT) for NSFW Image Classification. Hugging Face Model Hub. Retrieved April 5, 2025, from https://huggingface.co/Falconsai/nsfw_image_detection

30. J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, ‘Sub-center ArcFace: Boosting Face Recognition by Large-Scale Noisy Web Faces’, in Computer Vision -- ECCV 2020, 2020, pp. 741–757.

31. GitHub - 2025_M_PI_IPZ-23-2_Kizitskyi_M_O . *GitHub*. URL: https://github.com/avojarot/2025_M_PI_IPZ-23-2_Kizitskyi_M_O (дата звернення: 09.06.2025).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

7. Saichyshyna, N., Maksymenko, D., Turuta, O., Yerokhin, A., Babii, A., & Turuta, O. (2023). Extension Multi30K: Multimodal Dataset for Integrated Vision and Language Research in Ukrainian. In Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP) (pp. 54–61). Dubrovnik, Croatia: Association for Computational Linguistics.

8. Maksymenko, D., & Turuta, O. (2024). Interpretable Conversation Routing via the Latent Embeddings Approach. Computation, 12(12), 237. https://doi.org/10.3390/computation12120237

9. Erdem, E., Kuyu, M., Yagcioglu, S., Frank, A., Parcalabescu, L., Plank, B., Babii, A., Turuta, O., Erdem, A., Calixto, I., Lloret, E., Apostol, E.-S., Truică, C.-O., Šandrih, B., Martinčić-Ipšić, S., Berend, G., Gatt, A., & Korvel, G. (2022). Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability, and Learning. Journal of Artificial Intelligence Research, 73. https://doi.org/10.1613/jair.1.12918

10. Panchenko, D., Maksymenko, D., Turuta, O., Luzan, M., & Tytarenko, S. (2022). Ukrainian News Corpus as Text Classification Benchmark. In Ignatenko, O., et al. (Eds.), ICTERI 2021 Workshops. ICTERI 2021 (Vol. 1635). Springer, Cham. https://doi.org/10.1007/978-3-031- 14841-5_37

11. Improving Speaker Verification Model for Low-Resources Languages. CEUR Workshop Proc., 3403, 99–113, CoLInS 2023, Kharkiv.

12. O. Zolotukhin, V. Filatov, A. Yerokhin, O. Lanovyy, M. Kudryavtseva, V. Semenets, An approach to the selection of behavior patterns autonomous intelligent mobile systems, in: Proc. IEEE Int. Conf. Problems Infocommun. Sci. Technol. (PIC S&T), 2021, pp. 349–352. doi:10.1109/PICST54195.2021.9772110.

13. O. Zolotukhin, V. Filatov, A. Yerokhin, M. Kudryavtseva, The methods for the prediction of climate control indicators in the Internet of Things systems, CEUR Workshop Proc., 2021.