

**ПИЛИПІВ Людмила Андріївна**

**Програмний модуль генерації візуального  
контенту для цифрової культурної спадщини /  
Software Module for Generating Visual Content for  
Digital Cultural Heritage**

спеціальність: 122 - Комп'ютерні науки  
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи  
КНз-41  
Л. А. Пилипів

---

Науковий керівник:  
д.т.н., доцент, Х. В.  
Ліп'яніна-Гончаренко

---

Кваліфікаційну роботу  
допущено до захисту:

" \_\_\_\_ " \_\_\_\_\_ 20\_\_ р.

Завідувач кафедри  
\_\_\_\_\_ **Н. В. Дзюбановська**

**Навчально-науковий інститут інноваційних освітніх технологій**  
Кафедра інформаційно-обчислювальних систем і управління  
Ступінь вищої освіти «бакалавр»  
спеціальність 122 – «Комп'ютерні науки»  
освітньо-професійна програма – «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

В.о. завідувача кафедри

Н.В. Дзюбановська

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

**ПИЛИПІВ Людмилі Андріївні**

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

**Програмний модуль генерації візуального контенту для цифрової  
культурної спадщини/ Software Module for Generating Visual Content for  
Digital Cultural Heritage**

керівник роботи д.т.н., доцент Х.В. Ліп'яніна-Гончаренко,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– дослідити сферу цифрової культурної спадщини та потенціал генеративних моделей;

– провести огляд методів генерації даних та обрати дифузійну парадигму;

– проаналізувати інструменти створення відео за визначеними критеріями ефективності;

– розробити архітектуру модуля генерації анімованого мультимедійного контенту;

– реалізувати конвеєр інференсу AnimateDiff з налаштуванням параметрів;

– створити систему взаємодії користувача в Jupyter з валідацією запитів;

– інтегрувати операції візуалізації, збереження та завантаження результатів;

– провести апробацію модуля на сценаріях цифрової культурної спадщини;

– оцінити керованість, відтворюваність та результативність роботи системи.

5. Перелік графічного матеріалу в роботі:

– схема архітектури програмного модуля;

– схема алгоритму дифузійної генерації відео;

– UML-діаграма станів процесу взаємодії користувача з інтерфейсом;

– логічна модель (ER-діаграма) зберігання даних та метаданих.

## 6. Консультанти розділів кваліфікаційної роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата      |                     |
|--------|-------------------------------------------|-------------------|---------------------|
|        |                                           | Завдання<br>видав | Завдання<br>прийняв |
|        |                                           |                   |                     |
|        |                                           |                   |                     |
|        |                                           |                   |                     |

## 7. Дата видачі завдання 01 грудня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи                                                                                      | Строк виконання етапів кваліфікаційної роботи | Примітка |
|-------|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|----------|
| 1     | Огляд літературних джерел відповідно до предметної області та складання плану роботи                                     | до 01.01. 2026 р.                             |          |
| 2     | Написання 1 розділу кваліфікаційної роботи                                                                               | до 01.02. 2026 р.                             |          |
| 3     | Написання 2 розділу кваліфікаційної роботи                                                                               | до 01.04.2026 р.                              |          |
| 4     | Написання 3 розділу кваліфікаційної роботи                                                                               | до 01.05. 2026 р.                             |          |
| 5     | Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником                        | до 15.05.2026 р.                              |          |
| 6     | Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів | до 25.05.2026 р.                              |          |
| 7     | Перевірка кваліфікаційної роботи на оригінальність тексту                                                                | до 30.05.2026 р.                              |          |
| 8     | Оформлення кваліфікаційної роботи та отримання допуску до захисту                                                        | до 10.06.2026 р.                              |          |
| 9     | Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії                                              | до 10.06. 2026 р.                             |          |

Студент \_\_\_\_\_ Л.А. Пилипів  
( підпис ) (прізвище та ініціали)

Керівник кваліфікаційної роботи \_\_\_\_\_ Х.В. Ліп'яніна-Гончаренко  
( підпис ) (прізвище та ініціали)

## АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль генерації візуального контенту для цифрової культурної спадщини» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом 65 сторінок і містить 18 ілюстрацій, 1 таблицю, 2 додатки та 31 використане джерело.

Метою роботи є розроблення та експериментальна перевірка програмного модуля керованої генерації короткого відеоконтенту (GIF) за текстовим описом сцени на основі дифузійного підходу AnimateDiff із забезпеченням відтворюваності результатів через фіксацію параметрів інференсу та можливість базового архівування згенерованих артефактів.

Методами розроблення обрано метод аналізу (для дослідження предметної області цифрової культурної спадщини та існуючих рішень генерації відео), метод порівняльного аналізу (для формування критеріїв оцінювання open-source і пропрієтарних інструментів за керованістю, відтворюваністю та ресурсними вимогами), методи моделювання та проєктування (для побудови архітектури модульного конвеєра інференсу), методи програмної реалізації та експериментального тестування (для перевірки функціональної коректності, керованості та відтворюваності генерації за фіксованих параметрів).

Внаслідок виконання роботи спроектовано архітектуру програмного модуля та реалізовано алгоритмічний конвеєр генерації анімованого контенту на основі AnimateDiff у середовищі Jupyter/Kaggle із параметризацією інференсу (prompt, negative prompt, кількість кадрів, кількість кроків, сила текстового умовлення, seed, розмір кадру) та механізмами збереження/архівування результатів у форматі GIF. Проведено тестування на сценаріях, релевантних цифровій культурній спадщині (архітектура, сакральний інтер'єр, музейний артефакт, реставраційна реконструкція, цифрова експозиція, аероперспектива), що підтвердило керованість параметрами інференсу та відтворюваність результатів за фіксованого seed.

Результати дослідження можуть бути використані в навчальних і науково-дослідницьких підрозділах, а також у цифрових проєктах установ культурної пам'яті для швидкого формування демонстраційних мультимедійних матеріалів та прототипування візуальних наративів із фіксацією параметрів генерації.

Ключові слова: ЦИФРОВА КУЛЬТУРНА СПАДЩИНА, ДИФУЗІЙНА МОДЕЛЬ, ANIMATEDIFF, TEXT-TO-VIDEO, ГЕНЕРАЦІЯ ВІДЕО, ВІДТВОРЮВАНІСТЬ, JUPYTER, ПРОГРАМНИЙ МОДУЛЬ.

## ANNOTATION

Qualification thesis on the topic “Software Module for Generating Visual Content for Digital Cultural Heritage ” for obtaining the Bachelor’s degree in Specialty 122 “Computer Science” (Educational Program “Computer Science”) has a volume of 65 pages and contains 18 figures, 1 table, 2 appendixs, and 31 references.

The purpose of the thesis is to develop and experimentally validate a software module for controlled short video (GIF) generation from a natural-language scene description based on the AnimateDiff diffusion approach, ensuring reproducibility by fixing inference parameters and providing basic archiving of generated artifacts.

The research methods include analysis (to examine the digital cultural heritage domain and existing video generation solutions), comparative analysis (to define evaluation criteria for open-source and proprietary tools in terms of controllability, reproducibility, and resource requirements), modeling and design methods (to build the architecture of a modular inference pipeline), and software implementation and experimental testing methods (to verify functional correctness, controllability, and reproducibility under fixed parameters).

As a result, the software module architecture was designed and an AnimateDiff-based animated content generation pipeline was implemented in the Jupyter/Kaggle environment with inference parameterization (prompt, negative prompt, number of frames, number of steps, guidance scale, seed, frame size) and mechanisms for saving/archiving outputs in GIF format. Testing on scenarios relevant to digital cultural heritage (architecture, sacred interior, museum artifact, restoration reconstruction, digital exhibition, aerial perspective) confirmed controllability via inference parameters and reproducibility under a fixed seed.

The obtained results can be used in educational and research units, as well as in digital projects of cultural memory institutions, to rapidly create demonstrative multimedia materials and prototype visual narratives with documented generation parameters.

Keywords: DIGITAL CULTURAL HERITAGE, DIFFUSION MODEL, ANIMATEDIFF, TEXT-TO-VIDEO, VIDEO GENERATION, REPRODUCIBILITY, JUPYTER, SOFTWARE MODULE.

## ЗМІСТ

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Вступ.....                                                                 | 9  |
| 1 Аналіз предметної області та існуючих рішень .....                       | 12 |
| 1.1 Цифрова культурна спадщина як об'єкт ІТ-підтримки .....                | 12 |
| 1.2 Генеративні моделі для створення візуального контенту.....             | 15 |
| 1.3 Огляд існуючих рішень.....                                             | 17 |
| 1.4 Постановка задачі.....                                                 | 19 |
| 2 Алгоритмічне та інформаційне забезпечення модуля .....                   | 23 |
| 2.1 Архітектура програмного модуля .....                                   | 23 |
| 2.2 Алгоритм генерації відео .....                                         | 26 |
| 2.3 Інформаційне забезпечення.....                                         | 29 |
| 3 Програмно-технологічна реалізація та тестування.....                     | 33 |
| 3.1 Програмно-технологічна реалізація модуля генерації відеоконтенту ..... | 33 |
| 3.2 Реалізація інтерфейсу користувача та сценарії взаємодії.....           | 36 |
| 3.3 Тестування та оцінювання результатів роботи модуля.....                | 39 |
| Висновки .....                                                             | 49 |
| Список використаних джерел .....                                           | 51 |
| Додаток А Програмний код .....                                             | 55 |
| Додаток Б Апробація отриманих результатів .....                            | 58 |



## ВСТУП

Цифрова трансформація інституцій культурної пам'яті (музеїв, архівів, бібліотек) зумовлює потребу у створенні та масштабуванні мультимедійного контенту, здатного підвищувати доступність, залученість аудиторій і якість освітньо-популяризаційних матеріалів. У домені цифрової культурної спадщини (digital cultural heritage) зростає роль 3D-репрезентацій, XR-візуалізації та семантичного опису об'єктів, однак практичним викликом залишається оперативне формування коротких відеофрагментів (анімованих сцен/наративів), придатних для демонстрації в онлайн-експозиціях, презентаціях і навчальних матеріалах. Традиційне виробництво відеоконтенту потребує значних часових і кадрових ресурсів, а також спеціалізованих навичок монтажу й анімації, що обмежує масштабування таких практик у типовому освітньому чи музейному середовищі.

Сучасний стан розв'язання задачі генерації візуального контенту визначається розвитком генеративних моделей, зокрема дифузійних підходів, які демонструють високу якість синтезу та керованість за рахунок умовлення текстом. Перехід від генерації статичних зображень (Text-to-Image) до генерації коротких відеопослідовностей (Text-to-Video) супроводжується додатковими вимогами до темпоральної узгодженості (stability/temporal consistency), керування рухом і відтворюваності експериментів у конкретному обчислювальному середовищі. У прикладних сценаріях культурної спадщини ці вимоги посилюються необхідністю прозорого документування параметрів генерації (prompt, seed, steps, guidance), а також можливістю повторного відтворення результатів і подальшого архівування отриманих цифрових артефактів.

Актуальність роботи зумовлена потребою у відтворюваних та керованих програмних рішеннях, які дозволяють автоматизовано генерувати короткі анімовані візуальні фрагменти за текстовим описом сцени та використовувати їх як мультимедійні компоненти цифрових проєктів культурної спадщини. З огляду

на інженерні обмеження типових навчальних/демонстраційних середовищ, доцільним є застосування open-source екосистеми дифузійних моделей та підходів, що підтримують локальний запуск на GPU і контроль параметрів інференсу. У цьому контексті технологія AnimateDiff є практично релевантною, оскільки розширює можливості дифузійних Text-to-Image моделей до генерації узгоджених послідовностей кадрів шляхом інтеграції motion-модуля.

Мета роботи — розробити та експериментально перевірити програмний модуль керованої генерації короткого відеоконтенту (GIF) за текстовим описом сцени на основі дифузійного підходу AnimateDiff, забезпечивши відтворюваність результатів через фіксацію параметрів інференсу та можливість базового архівування згенерованих артефактів.

Для досягнення мети поставлено такі завдання:

1. Проаналізувати предметну область цифрової культурної спадщини та обґрунтувати застосування генеративних моделей для створення мультимедійного контенту.
2. Дослідити підходи генеративного моделювання візуальних даних (GAN, VAE, diffusion) та обґрунтувати вибір дифузійної парадигми для синтезу відеопослідовностей.
3. Проаналізувати наявні інструменти та рішення генерації відео (open-source і пропріетарні) та сформулювати критерії порівняння (керованість, відтворюваність, ресурсні вимоги, формат результату).
4. Спроектувати архітектуру програмного модуля генерації анімованого контенту та реалізувати алгоритмічний конвеєр інференсу AnimateDiff з параметризацією.
5. Реалізувати інтерфейс взаємодії користувача в середовищі Jupyter із валідацією введення, відображенням результату та операціями збереження/завантаження GIF.
6. Провести тестування модуля на сценаріях, релевантних цифровій культурній спадщині, та виконати аналіз керованості й відтворюваності результатів за фіксованих параметрів.

Об'єкт дослідження — процес автоматизованої генерації візуального контенту на основі генеративних моделей у задачах цифрової культурної спадщини.

Предмет дослідження — методи та програмні засоби дифузійної генерації коротких відеопослідовностей за текстовим описом сцени (AnimateDiff), включно з параметризацією інференсу, механізмами відтворюваності та формуванням вихідних мультимедійних артефактів.

Методи дослідження: методи генеративного моделювання та дифузійного синтезу; методи керованої генерації (text conditioning, negative prompting, guidance scaling); методи програмної інженерії прототипів (проектування архітектури модулів, параметризація, обробка помилок); експериментальне тестування та аналіз результатів за контрольованими параметрами (seed, steps, num\_frames, роздільність).

Практичне значення роботи полягає у створенні відтворюваного програмного модуля, який забезпечує генерацію короткого анімованого контенту для демонстраційних і навчальних задач у сфері цифрової культурної спадщини, а також може бути використаний як основа для подальших удосконалень (експорт у mp4/webm, керування FPS, розширене журналювання метаданих, інтеграція з архівними контурами та системами керування цифровими колекціями).

Апробація результатів дослідження здійснена в межах X Міжнародної студентської конференції «Наука сьогодення: від досліджень до стратегічних рішень», яка відбулася в місті Одесі 27 лютого 2026 року.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

## 1.1. Цифрова культурна спадщина як об'єкт ІТ-підтримки

Цифрова культурна спадщина (digital cultural heritage) у сучасній ІТ-парадигмі розглядається як сукупність цифрових ресурсів, що репрезентують культурні цінності та знання (документальні свідчення, наукові описи, цифрові копії артефактів, колекції зображень, аудіовізуальні матеріали, 3D-моделі, веб-ресурси), які мають довготривалу суспільну, наукову й освітню значущість. Ключовою особливістю є те, що значна частина таких ресурсів або «народжується цифровою», або формується шляхом оцифрування матеріальних об'єктів, а їх збереження потребує не лише створення цифрової копії, але й підтримки контексту, автентичності та керованого доступу [1].

З позицій життєвого циклу даних цифрова спадщина є вразливою до втрат через стрімку моральну застарілість апаратного та програмного забезпечення, зміну форматів, залежність від конкретних платформ і сервісів, а також через організаційні ризики (нестача ресурсів, відсутність політик збереження, розпорошена відповідальність). ЮНЕСКО підкреслює, що цифрові матеріали часто мають «ефемерний» характер і потребують цілеспрямованого управління та підтримки, щоб бути збереженими для теперішніх і майбутніх поколінь [1]. Тому ІТ-підтримка цифрової культурної спадщини має проєктуватися як стійка система процесів, а не як одноразовий акт оцифрування.

Інституції культурної пам'яті (музеї, архіви, бібліотеки) в умовах цифрової трансформації переходять від традиційних практик інвентаризації до сервісної моделі, де цифровий контент стає основою комунікації з аудиторіями, підтримки досліджень та створення навчальних матеріалів. У політиці Європейської Комісії акцентується, що цифрові технології (дані, 3D, XR, AI) відкривають можливості «оживлення» культурних об'єктів і підвищення доступності матеріалів через онлайн-сервіси та віртуальні музеї [2]. Це, у свою чергу, формує вимоги до інтегрованості, стандартизації метаданих, масштабованості сховищ і забезпечення цифрового збереження на рівні інфраструктури.

Одним із центральних напрямів ІТ-підтримки є 3D-архівування, яке забезпечує документування геометрії та візуальних характеристик об'єктів спадщини для аналізу, консервації, моніторингу змін і реконструкції. 3D-подання розглядаються як еталонний цифровий «зліпок» стану об'єкта у визначений момент часу, що потребує відтворюваного конвеєра: захоплення даних → реконструкція → валідація точності → оптимізація → публікація/збереження. У цьому контексті суттєвими є не лише 3D-моделі, але й протоколи контролю якості та повнота метаданих, які фіксують умови та параметри створення цифрового ресурсу.

Для формування 3D-моделей у практиці цифрової спадщини найчастіше застосовують фотограмметрію та лазерне сканування як взаємодоповнювальні підходи. У науковому огляді F. Remondino показано, що розвиток сенсорів та методів 3D-вимірювань напряду впливає на якість цифрової документації та можливості подальшого використання моделей у задачах консервації й репрезентації [3]. Вибір технології визначається масштабом об'єкта, вимогами до точності, характеристиками поверхні, доступністю локації та допустимими витратами часу й ресурсів.

Окрему роль у популяризації та освіті відіграють VR/AR та ширше — XR-технології, що дозволяють відтворювати недоступні середовища, реконструювати втрачені фрагменти, демонструвати експонати у контексті та забезпечувати інтерактивний досвід. Європейська Комісія підкреслює, що саме поєднання 3D та XR здатне повертати культурні об'єкти «до життя» і розширювати доступ до матеріалів для різних категорій користувачів [2]. Для ІТ-підтримки це означає потребу в оптимізації 3D-контенту (рівні деталізації, текстуровання, продуктивність рендерингу), а також у розробленні інтерфейсів взаємодії, придатних для різних пристроїв.

Водночас цифрова культурна спадщина не зводиться до візуалізації: її цінність визначається також семантичним контекстом, атрибуцією, походженням (provenance), правами доступу й відтворюваністю інтерпретацій. Звідси впливає вимога до ІТ-підтримки як до поєднання сховищ, інструментів опису

(метаданих), пошуку та механізмів керування правами. Підхід ЮНЕСКО наголошує на необхідності балансу між відкритим доступом (особливо для матеріалів у публічному домені) та захистом чутливої інформації й правовласників [1].

На цьому тлі генеративні моделі стають технологічним підсилювачем цифрової спадщини, оскільки дозволяють масштабувати створення візуального контенту: від ілюстративних реконструкцій до відео-наративів, промо-матеріалів і навчальних роликів. Політичні документи ЄС прямо фіксують роль AI поряд із 3D та XR як чинника розвитку цифрового культурного середовища [2]. У практичних сценаріях музеїв та архівів генеративні моделі можуть підтримувати швидке прототипування візуальних гіпотез, створення мультимедійних пояснень та підвищення залученості аудиторії через сучасні формати подачі.

Разом із перевагами виникають і ризики застосування генеративного ШІ в культурно чутливих доменах: імовірність фактологічних помилок у реконструкціях, змішування стилів і епох, або створення «правдоподібних» але неавтентичних візуалізацій. Тому для програмних модулів генерації контенту принциповими стають механізми керуваності (параметри генерації, відтворюваність через seed), журналювання (prompt, налаштування моделі, версії залежностей), маркування синтетичного контенту та залучення експертної валідації перед публікацією результатів [1, 2]. Це забезпечує прозорість, знижує ризики хибної інтерпретації та підвищує довіру до цифрових продуктів у сфері спадщини.

Отже, цифрова культурна спадщина як об'єкт ІТ-підтримки вимагає системного підходу, де технології 3D-архівування, XR-візуалізації та генеративні моделі інтегруються в відтворюваний конвеєр: захоплення даних → обробка → збереження → доступ → повторне використання. У такій логіці розроблення програмного модуля генерації відеоконтенту на основі diffusion-підходів є прикладним інструментом, що поєднує вимоги довготривалого збереження й доступності цифрових ресурсів [1, 2] із сучасними методами 3D-документації та візуальної репрезентації [3].

## 1.2 Генеративні моделі для створення візуального контенту

Генеративні моделі є ключовим класом методів машинного навчання, орієнтованих на відновлення прихованого розподілу даних і синтез нових зразків, які є статистично подібними до навчальної вибірки. У контексті візуального контенту (зображення, відео) такі моделі виступають основою для автоматизованого створення ілюстрацій, реконструкцій, стилізацій та мультимодальних медіа, що є релевантним для цифрової культурної спадщини, де потрібна масштабованість, повторюваність і контрольованість процесу генерації [4–6].

Одним із найбільш впливових підходів у генеративному моделюванні стали генеративно-змагальні мережі (GAN), у яких генератор і дискримінатор навчаються в умовах мінімаксної оптимізації. Методологічно GAN забезпечують високу якість зразків завдяки навчанню через «змагальність», однак на практиці часто потребують чутливого налаштування гіперпараметрів та можуть демонструвати нестабільність навчання і схильність до mode collapse. Для задач синтезу візуального контенту це означає потенційну високу реалістичність, але складність забезпечення гарантованої відтворюваності та керованості результатів у прикладних системах [4, 7].

Іншим фундаментальним класом є варіаційні автоенкодери (VAE), що поєднують нейромережеве кодування з варіаційним висновуванням у латентному просторі. VAE формалізують компроміс між якістю реконструкції та регуляризацією латентного простору через максимізацію нижньої межі правдоподібності, що робить їх привабливими для задач, де потрібні інтерпретовані латентні фактори, контроль генерації та стабільність навчання. У домені культурної спадщини VAE-парадигма методологічно важлива як попередній крок до латентних генеративних схем, де генерування відбувається не в піксельному просторі, а у стисненому представленні [5, 8].

Сучасний state-of-the-art у синтезі зображень значною мірою асоціюється з дифузійними моделями (Diffusion Models), які моделюють процес поступового

зашумлення даних (forward diffusion) та навчаються відновлювати дані з шуму (reverse diffusion). Концептуально цей підхід надає сильні гарантії якості та різноманіття зразків, хоча інференс є послідовним і потенційно дорогим. Водночас саме дифузійні моделі відкрили шлях до високої керованості генерації через механізми умовлення (conditioning) і керування траєкторією денойзингу [6, 9].

Важливим практичним кроком до здешевлення дифузійних підходів стала концепція латентних дифузійних моделей, де генерація відбувається в латентному просторі попередньо навченого автоенкодера. Це суттєво знижує обчислювальні витрати порівняно з піксельним простором і водночас зберігає високу візуальну якість. Додатково, введення крос-уваги (cross-attention) для умовлення текстом зробило такі моделі технологічною базою для масового класу систем Text-to-Image, зокрема сімейства Stable Diffusion [9–10].

Для ефективного мультимодального умовлення у Text-to-Image критичним є узгодження текстових і візуальних репрезентацій. У цьому сенсі значущою стала методологія контрастивного навчання «текст–зображення», де модель навчається пов'язувати текстові описи з відповідними візуальними сценами. Такі підходи (зокрема CLIP) підсилюють семантичну релевантність генерації та формують базовий інтерфейс керування через природну мову, що є особливо цінним для культурно-історичних наративів та семантично багатих описів об'єктів спадщини [11].

Перенесення підходів Text-to-Image у домен Text-to-Video додає принципово новий вимір — часову узгодженість (temporal consistency) та моделювання руху. Наукові роботи в цьому напрямі демонструють, що пряме розширення 2D-архітектур до 3D-обробки відео швидко ускладнює обчислення, тому застосовуються каскадні схеми, просторово-часові модулі та окремі блоки суперроздільності. У таких системах цільовою стає не лише фотореалістичність, а й стабільність об'єктів у часі, керованість камерою та збереження причинно-наслідкової логіки сцени [12–15].



Підхід AnimateDiff є показовим прикладом практичної інтеграції motion-модуля в персоналізовані Text-to-Image моделі без необхідності специфічного тонкого налаштування під конкретну модель. Методично це реалізується як «plug-and-play» модуль, що інjektує апіорі руху у вже наявні генератори зображень, підсилюючи їх здатність створювати послідовності кадрів. Для прикладного програмного модуля (зокрема в культурній спадщині) це важливо тим, що дає можливість використання екосистеми існуючих T2I-моделей і персоналізацій, додаючи відеогенерацію як надбудову з контрольованими параметрами інференсу [16–18].

У межах кваліфікаційної роботи аналіз підходу AnimateDiff доцільно виконувати через призму інженерних критеріїв: (i) вимоги до ресурсів (VRAM, час інференсу), (ii) керованість (seed, steps, guidance scale, негативні промпти), (iii) якість темпоральної узгодженості та (iv) відтворюваність результатів у заданому середовищі (Kaggle/локальна GPU). Важливим також є питання інтеграції в архівний контур цифрової спадщини: збереження метаданих генерації (prompt, параметри, версії моделей), що забезпечує прозорість і можливість повторного відтворення відео-артефактів як частини цифрового фонду.

### 1.3 Огляд існуючих рішень

Практичний ландшафт генерації візуального контенту нині визначається поєднанням відкритих (open-source) екосистем і пропрієтарних хмарних сервісів. Для задач цифрової культурної спадщини ця дихотомія має принципове значення: open-source інструменти підвищують відтворюваність, контроль версій і можливість розгортання «on-premise», тоді як хмарні сервіси часто пропонують кращий UX, але обмежують прозорість моделі, даних і параметрів генерації [10, 19].

Stable Diffusion є базовою технологічною платформою для численних сценаріїв Text-to-Image і похідних застосувань (inpainting, image-to-image,

стилізація), оскільки реалізує латентну дифузію з доступними вагами і широкою інфраструктурною підтримкою. У контексті створення візуального контенту для спадщини це означає можливість формування ілюстраційних реконструкцій, стилізованих візуалізацій та концептів експозицій, а також технологічну сумісність з екосистемою персоналізацій (наприклад, LoRA/DreamBooth-подібні практики у широкому інструментарії спільноти) [9–10].

Разом із тим, Stable Diffusion є насамперед генератором статичних зображень; відеогенерація в такій екосистемі реалізується або через зовнішні «відео-надбудови», або через спеціалізовані модулі руху та інтерполяції. Саме тому рішення на кшталт AnimateDiff є технологічно логічним продовженням: вони зберігають сильні сторони T2I-моделей (візуальна якість і різноманіття) та додають часовий вимір через motion-модуль, що мінімізує потребу в повному перенавчанні під відео [16–18].

AnimateDiff як рішення доцільно розглядати у двох площинах: (i) як науково обґрунтований підхід (ICLR spotlight), де описано архітектуру та мотивацію, та (ii) як інженерну реалізацію, що підтримує інтеграцію з популярними моделями спільноти та дозволяє запуск інференсу на поширеному GPU-обладнанні. Для розроблення прикладного програмного модуля важливими є практичні характеристики: вимоги до VRAM, стабільність генерації, керованість параметрами (steps, seed тощо) і типова тривалість/роздільність відеофрагментів [16–18].

Runway Gen-2 репрезентує клас пропрієтарних веб-платформ, орієнтованих на кінцевого користувача, де Text-to-Video реалізовано як сервіс із мінімальним порогом входу. Публічні матеріали (офіційні демонстрації та незалежні огляди) підкреслюють, що Gen-2 забезпечує генерацію коротких відео з текстових описів і позиціонується як інструмент демократизації відеовиробництва, однак архітектурна прозорість, можливість локального розгортання і формальна відтворюваність для академічних експериментів є обмеженими [20–22].

Pika Labs (лінійка релізів, включно з Pika 1.0) також належить до сервісів, що конкурують у сегменті коротких Text-to-Video кліпів і надають користувачу швидкий цикл «prompt → результат». У відкритих описах релізів наголошується на розвитку якості руху, стилів і додаткових креативних функцій, однак — аналогічно до інших сервісів — зберігається залежність від зовнішньої платформи, непрозорість внутрішньої моделі та обмеження щодо повторюваності експериментів поза межами сервісу [23, 27].

Для академічного порівняння рішень у роботі доцільно застосувати уніфіковані критерії: (i) тип генерації (Text-to-Video / Image-to-Video), (ii) довжина ролика та FPS, (iii) контрольованість (керування seed, guidance scale, camera/motion), (iv) доступність і відтворюваність (open-source/closed, локальний запуск), (v) ресурсні вимоги (VRAM, час інференсу), (vi) обмеження ліцензування та можливість використання у некомерційних/освітніх сценаріях [12–15, 19]. Саме ці критерії мають бути відображені у порівняльній таблиці функціональності.

Узагальнюючи, open-source зв'язка Stable Diffusion + AnimateDiff є методично виправданою для побудови дослідницько-орієнтованого програмного модуля в університетському контексті, оскільки підтримує контроль параметрів, аудит експериментів і локальне розгортання. Натомість платформи Runway Gen-2 і Pika є корисними як еталонні «industry baseline» для порівняння UX та сприйнятної якості коротких кліпів, але потребують обережної інтерпретації результатів через закритість моделей і невизначеність щодо деталей навчання та інференсу [16–18, 20–23].

#### 1.4 Постановка задачі

Актуальність дослідження зумовлена комплексним поєднанням технологічних, організаційних та комунікаційних викликів, характерних для сфери цифрової культурної спадщини. По-перше, цифрова культурна спадщина як об'єкт ІТ-підтримки потребує не лише оцифрування, а й забезпечення

довготривалого збереження, автентичності, контексту (provenance) та керованого доступу до цифрових ресурсів. У практиці музеїв, архівів і бібліотек відбувається перехід до сервісної моделі, де цифровий контент стає інструментом комунікації з аудиторіями та підтримки освіти й досліджень. Водночас цифрові ресурси є вразливими до технологічної застарілості, фрагментації форматів і платформної залежності, тому особливої ваги набувають відтворювані конвеєри створення та збереження контенту, які можуть бути інтегровані в інфраструктуру цифрового архівування. На цьому тлі виникає прикладна потреба у швидкому масштабуванні мультимедійних матеріалів, зокрема коротких відео-сцен і анімованих фрагментів, які здатні “оживляти” 3D-об’єкти, підтримувати онлайн-експозиції та забезпечувати зрозумілі наративи для різних категорій користувачів.

По-друге, сучасний розвиток генеративного машинного навчання (GAN, VAE, diffusion) формує новий технологічний клас інструментів для синтезу візуального контенту, що є релевантним для цифрової спадщини саме через масштабованість, керованість та можливість параметризованого відтворення результатів. Якщо GAN-підходи можуть давати високу реалістичність, то вони часто характеризуються нестабільністю навчання та складністю гарантування повторюваності. VAE забезпечують більш структурований латентний простір, але зазвичай поступаються у фотореалістичності. Натомість дифузійні моделі стали практичним “стандартом якості” синтезу зображень і створили основу для переходу до Text-to-Video, де ключовою проблемою є темпоральна узгодженість та керування рухом. У контексті культурної спадщини це критично, оскільки відеоконтент має не лише бути візуально переконливим, а й відповідати змістовим вимогам: відтворювати стиль, епоху, атрибуцію та не створювати хибних “правдоподібних” реконструкцій. Тому актуальною стає розробка програмного модуля, який поєднує дифузійний підхід із механізмами контролю генерації (steps, guidance, negative prompt, seed) і прозорого фіксування метаданих, що підвищує довіру до результатів та їхню придатність для освітньо-музейних сценаріїв.

По-третє, аналіз існуючих рішень показує практичну дилему між відкритими та пропрієтарними екосистемами генерації відео. Хмарні сервіси (на кшталт комерційних Text-to-Video платформ) забезпечують низький поріг входу та швидке отримання результату, однак обмежують академічно важливі аспекти: прозорість архітектури, керування параметрами, локальне розгортання, контроль версій і відтворюваність експериментів поза межами платформи. Для кваліфікаційної роботи, де необхідні доказовість, повторюваність та можливість технічної верифікації, методично обґрунтованішими є open-source рішення, сумісні з локальним GPU-запуском і контрольованим інференсом. У цьому контексті зв'язка Stable Diffusion + AnimateDiff є актуальною як інженерно реалізований компроміс між якістю синтезу та можливістю відтворюваного експерименту: вона дозволяє використовувати існуючі T2I-моделі та надбудовувати відеогенерацію через motion-модуль без повного перенавчання під відео, що знижує бар'єр реалізації й робить підхід доступним для університетського середовища (Kaggle/локальна GPU).

Отже, актуальність роботи визначається необхідністю створення відтворюваного програмного модуля генерації короткого відеоконтенту для цифрової культурної спадщини, який: (i) підтримує інфраструктурну логіку збереження та повторного використання цифрових артефактів (підрозділ 1.1), (ii) спирається на сучасний state-of-the-art дифузійних моделей із керованістю та параметризацією генерації (підрозділ 1.2), та (iii) усуває обмеження закритих платформ за рахунок open-source реалізації з прозорими параметрами інференсу і можливістю аудиту результатів (підрозділ 1.3).

Мета роботи — розробити та експериментально перевірити програмний модуль керованої генерації короткого відеоконтенту (GIF) за текстовим описом сцени на основі дифузійного підходу AnimateDiff, забезпечивши відтворюваність результатів через фіксацію параметрів інференсу та можливість базового архівування згенерованих артефактів.

Для досягнення мети поставлено такі завдання:

1. Проаналізувати предметну область цифрової культурної спадщини та обґрунтувати застосування генеративних моделей для створення мультимедійного контенту.
2. Дослідити підходи генеративного моделювання візуальних даних (GAN, VAE, diffusion) та обґрунтувати вибір дифузійної парадигми для синтезу відеопослідовностей.
3. Проаналізувати наявні інструменти та рішення генерації відео (open-source і пропрієтарні) та сформувані критерії порівняння (керованість, відтворюваність, ресурсні вимоги, формат результату).
4. Спроекувати архітектуру програмного модуля генерації анімованого контенту та реалізувати алгоритмічний конвеєр інференсу AnimateDiff з параметризацією.
5. Реалізувати інтерфейс взаємодії користувача в середовищі Jupyter із валідацією введення, відображенням результату та операціями збереження/завантаження GIF.
6. Провести тестування модуля на сценаріях, релевантних цифровій культурній спадщині, та виконати аналіз керованості й відтворюваності результатів за фіксованих параметрів.

## 2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ

### 2.1 Архітектура програмного модуля

Архітектура розробленого програмного модуля побудована як параметризований конвеєр генерації анімованого візуального контенту за текстовим описом сцени з використанням дифузійного підходу AnimateDiff у середовищі Jupyter/Kaggle. Її проєктування орієнтоване на відтворюваність експериментів, керованість параметрами інференсу та прозорість етапів обробки даних — від введення текстового запиту до отримання та збереження GIF-результату. Загальна структурно-функціональна взаємодія компонентів, а також основні потоки даних і керування між ними подані на рисунку 2.1.

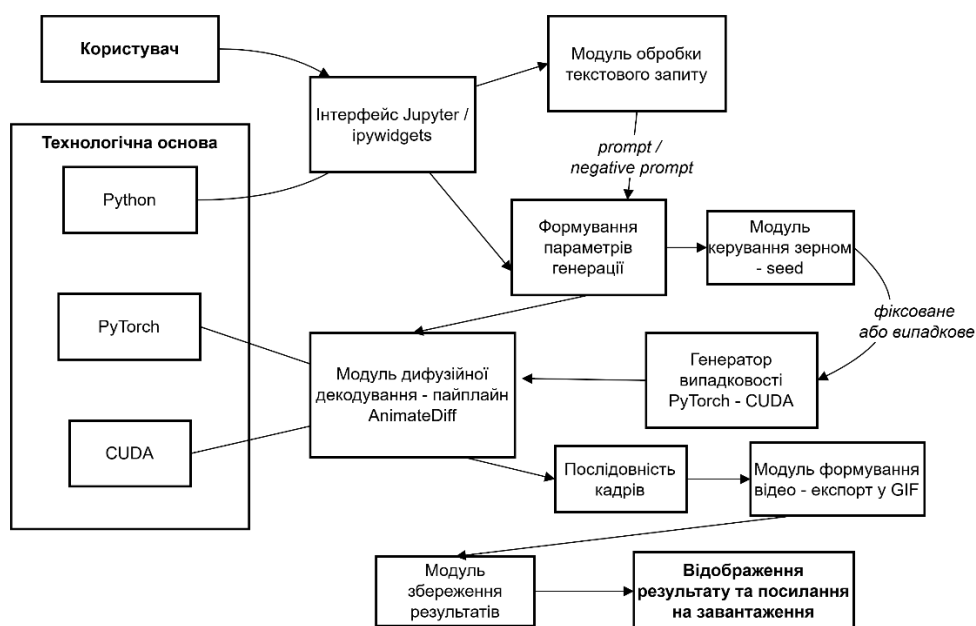


Рисунок 2.1 – Структурна схема архітектури програмного модуля

Першою ланкою конвеєра виступає модуль обробки текстового запиту, реалізований через українськомовний інтерфейс ipywidgets (поля введення “Опис сцени” та “Негативний опис”). На цьому етапі здійснюється початкова валідація вхідних даних: перевірка непорожнього опису, нормалізація рядкових значень, підготовка негативного опису як обмежувального контексту генерації, а

також узгодження числових параметрів (кількість кадрів, кількість кроків, сила впливу) з допустимими діапазонами. Додатково реалізовано опціональне встановлення розміру кадру (висота/ширина) з обробкою потенційно некоректних значень, що знижує ризики зупинки виконання через помилки користувача та підвищує надійність модулю на етапі підготовки аргументів.

Далі формується набір параметрів генерації, який визначає конфігурацію інференсу: `prompt`, `negative_prompt`, `num_frames`, `num_inference_steps`, `guidance_scale`, `seed`, а за потреби — `height` і `width`. В архітектурі це відповідає компоненту параметризації та зв'язування налаштувань, що забезпечує однозначність експерименту (ідентичні вхідні параметри → відтворюваний результат). Саме в цій точці конвеєра відбувається перехід від UI-рівня до обчислювального ядра пайплайна; відповідні залежності та напрямки передачі параметрів відображено на рисунку 2.1.

Критичною умовою наукової відтворюваності є модуль керування випадковістю, що реалізований шляхом ініціалізації `torch.Generator(device="cuda").manual_seed(seed)`. Фіксація `seed` забезпечує повторюваність траєкторії денойзингу та, як наслідок, стабільність відтворення результатів за незмінних параметрів. Для експериментальних сценаріїв передбачено режим “випадкове зерно”, який автоматизує пошук альтернативних варіантів генерації та водночас фіксує отримане значення `seed` у стані інтерфейсу, що дозволяє повернутися до конкретного результату або повторити його в подальших запусках.

Обчислювальне ядро системи становить модуль дифузійної декодування, реалізований як виклик пайплайна `pipe(**kwargs)` з параметрами умовлення текстом і конфігурацією інференсу. В межах *diffusion-based architecture* параметр `num_inference_steps` задає компроміс між якістю синтезу та обчислювальними витратами, тоді як `guidance_scale` регулює міру впливу текстового опису на процес відновлення сигналу з шуму. Для прикладних задач цифрової культурної спадщини такі параметри є інструментами керування семантичною



відповідністю сцени й стилістичною узгодженістю кадрів, що важливо для формування пояснювального або демонстраційного контенту.

Результатом виконання пайплайна є послідовність кадрів, яка надалі обробляється модулем формування відео. У поточній реалізації кадри витягуються з вихідної структури (`output.frames[0]`) та експортуються у GIF-формат через `export_to_gif`. Такий підхід забезпечує прикладну завершеність конвеєра: вихідний артефакт одразу є придатним до перегляду, демонстрації та поширення в освітніх/музейних сценаріях. Водночас архітектурно цей етап відокремлено від етапу дифузійної декодування, що створює передумови для подальшого розширення (експорт у `mp4`, керування FPS, постобробка, стабілізація).

Модуль збереження результатів охоплює контроль іменування файлів, створення копій з часовими мітками та формування посилань на завантаження. Реалізовані допоміжні функції `_ensure_gif_name` та `_timestamp_name` гарантують коректність імені вихідного файлу, а також підтримують зручний режим архівації результатів. Для мінімальної фіксації метаданих останнього експерименту використано структуру стану останній = `{"path": None, "seed": None}`, яка спрощує реалізацію операцій “зберегти копію” та “завантажити GIF” без повторного запуску генерації. Логіка взаємодії користувача з цими операціями, а також переходи між станами виконання (готово/генерація/успішно/помилка) наведені на рисунку 2.2.

Окремо слід підкреслити, що запропонована архітектура поєднує інтерактивність та інженерну дисципліну відтворюваних обчислень: інтерфейс забезпечує контроль параметрів, модуль керування випадковістю — повторюваність, а пайплайн дифузійної генерації — високу якість синтезу кадрів. Обґрунтування вибору технологічного стеку відповідає цим цілям: Python забезпечує швидке прототипування та інтеграцію з ML-екосистемою, PyTorch — гнучке керування інференсом і генераторами випадковості, CUDA — прискорення обчислень на GPU для дифузійного денойзингу та формування

багатокадрового виходу, а diffusion-based підхід — ефективне текстове умовлення та високу візуальну якість.

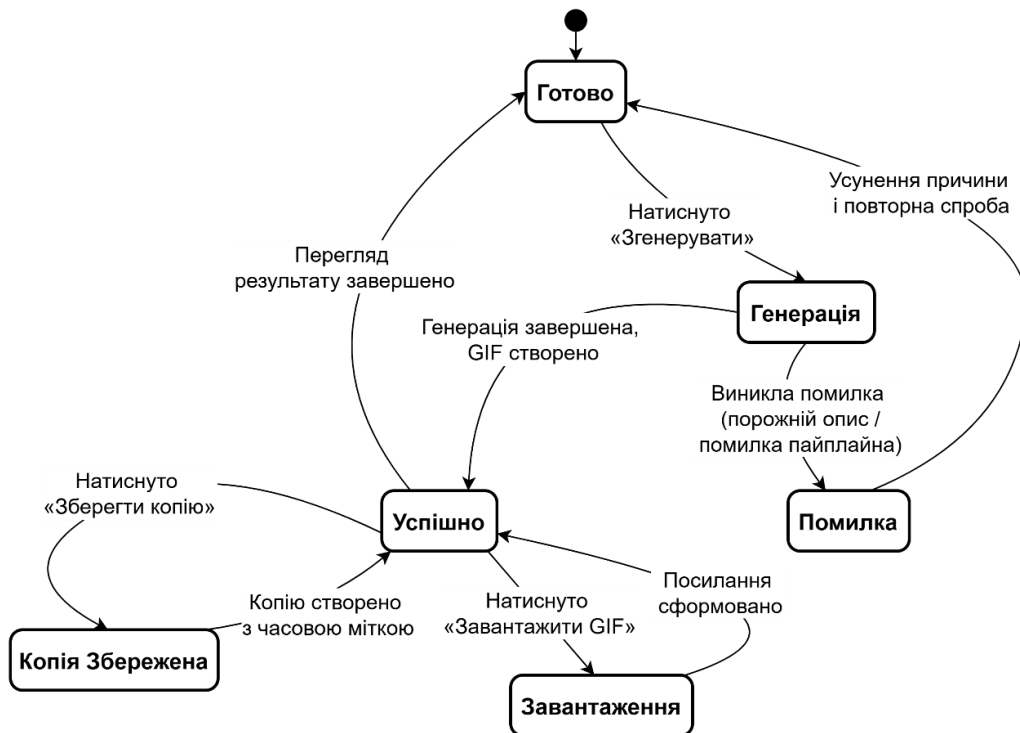


Рисунок 2.2 – Сценарій роботи користувача та стани виконання

У сукупності, як показано на рисунках 2.1–2.2, архітектура забезпечує логічно завершений і масштабовний каркас програмного модуля, придатний як для демонстраційних задач, так і для подальшого розвитку до повноцінної системи журналювання та архівування експериментів.

## 2.2 Алгоритм генерації відео

Алгоритм генерації відеоконтенту в розробленому програмному модулі базується на дифузійному підході до моделювання розподілу даних, що поєднує поступове зашумлення сигналу (forward diffusion) із навченою процедурою його поетапного відновлення (reverse diffusion). У контексті реалізації AnimateDiff дифузійний механізм працює в латентному просторі попередньо навченого автоенкодера, що зменшує обчислювальну складність порівняно з генерацією

безпосередньо у піксельному просторі та забезпечує баланс між якістю та швидкістю інференсу.

Формально прямий дифузійний процес описується як стохастичне перетворення даних, у межах якого на кожному кроці до латентного представлення додається гаусів шум. Узагальнена форма одного кроку може бути подана рівнянням:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \varepsilon, \quad (2.1)$$

де  $x_{t-1}$  — латентне представлення на попередньому кроці,  $\alpha_t$  — коефіцієнт масштабування сигналу, а  $\varepsilon$  — випадкова величина з нормального розподілу. У прямому процесі (forward diffusion) сигнал поступово втрачає структуру, наближаючись до чистого шуму. Водночас навчена модель апроксимує обернений процес (reverse diffusion), який дозволяє поетапно відновлювати структурований сигнал із шумового стану.

У практичній реалізації нашого модуля обернений процес ініціюється з латентного шуму, що генерується через фіксований або випадковий seed. Це забезпечує керованість та відтворюваність результатів. Параметр `num_inference_steps` визначає кількість ітерацій денойзингу, а отже — точність відновлення структури сцени. Зі збільшенням кількості кроків зростає деталізація кадрів, проте збільшується і час інференсу, що є важливим з погляду оптимізації обчислювальних ресурсів GPU.

Ключовим механізмом керування генерацією виступає умовлення (conditioning) через текстовий енкодер CLIP. Текстовий опис сцени (prompt) проходить токенизацію та перетворюється на векторне представлення у спільному латентному просторі “текст–зображення”. Під час кожного кроку reverse diffusion модель враховує це текстове представлення, коригуючи траєкторію денойзингу таким чином, щоб згенерований латентний код відповідав семантичному змісту запиту. Параметр `guidance_scale` визначає силу впливу текстового умовлення на процес відновлення.

У випадку відеогенерації алгоритм розширюється просторово-часовим виміром. Після формування послідовності латентних представлень для кожного

кадру здійснюється їх декодування у візуальний простір. AnimateDiff інтегрує модуль руху, який забезпечує узгодженість між кадрами та зменшує артефакти часової нестабільності. Таким чином, модель генерує не незалежні зображення, а узгоджену часову послідовність, що інтерпретується як відеоряд.

З алгоритмічної точки зору повний цикл генерації можна подати як послідовність операцій: введення текстового опису, токенизація, ініціалізація латентного шуму, ітеративний процес денойзингу з текстовим умовленням, декодування кадрів та об'єднання їх у фінальний відеофайл. Узагальнену блок-схему алгоритму генерації відео наведено на рисунку 2.3, де відображено логічні переходи між етапами обробки.

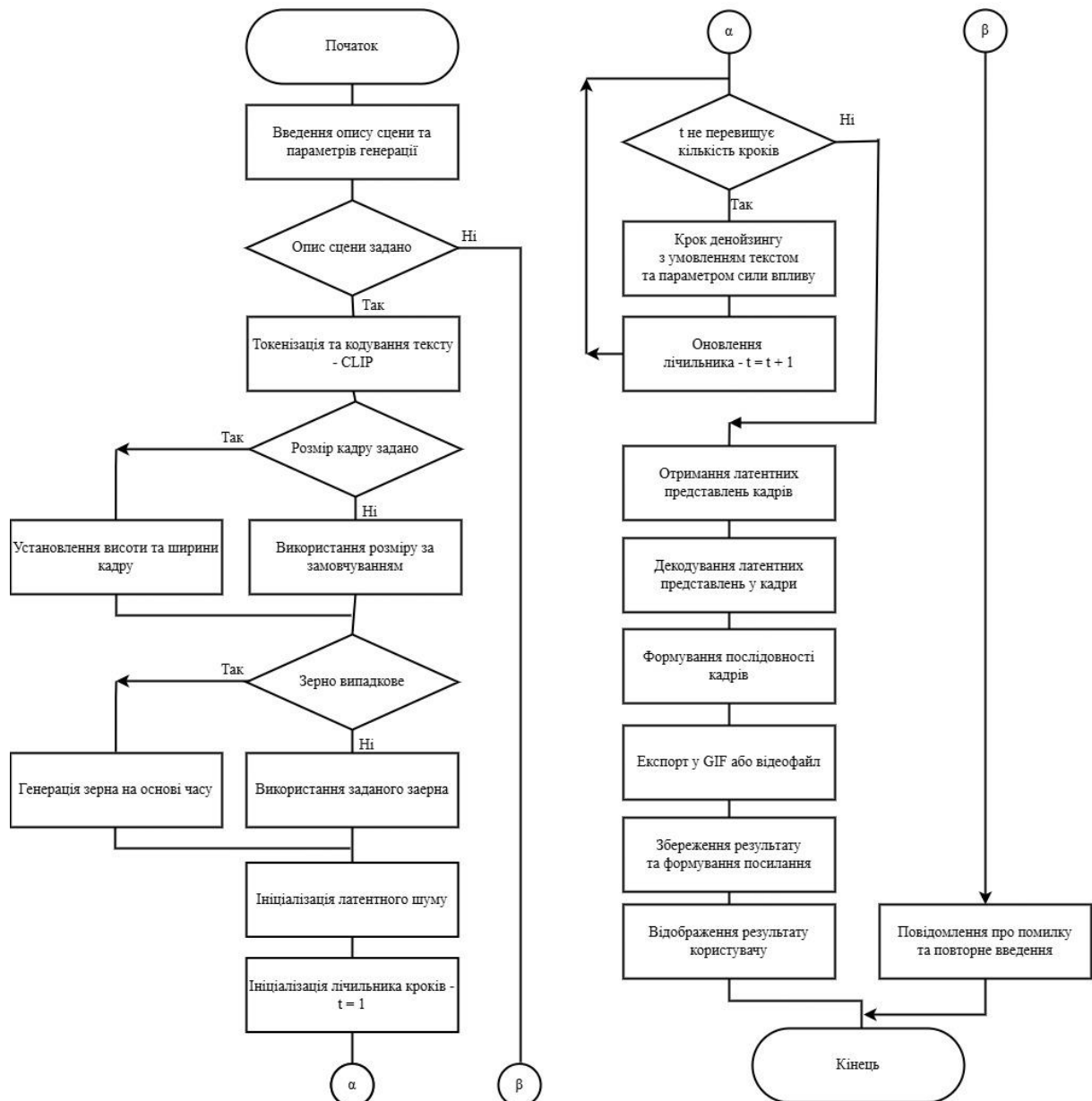


Рисунок 2.3 – Схема алгоритму генерації відео

Застосований алгоритм є масштабовним: зміна кількості кадрів (`num_frames`), кроків інференсу або сили умовлення дозволяє адаптувати якість і тривалість відео до конкретних задач, зокрема створення демонстраційних матеріалів для цифрової культурної спадщини. Водночас архітектура передбачає можливість подальшого розширення — інтеграцію суперроздільності, стабілізації руху або експорту в різні формати (`mp4`, `webm`).

Таким чином, дифузійний алгоритм у розробленому програмному модулі реалізує керовану генерацію відеоконтенту на основі природномовного опису, поєднуючи математично обґрунтовану модель поступового відновлення сигналу з інженерними механізмами керування параметрами та ресурсами обчислювального середовища.

## 2.3 Інформаційне забезпечення

Інформаційне забезпечення програмного модуля генерації відеоконтенту визначає структуру вхідних і вихідних даних, а також логіку їх зберігання та подальшого використання. У контексті розробленої системи воно спрямоване на забезпечення відтворюваності експериментів, простежуваності параметрів генерації та можливості повторного використання результатів у цифрових проєктах культурної спадщини. Формалізація інформаційної моделі дозволяє мінімізувати втрати метаданих, забезпечити контроль версій і створити основу для подальшого розширення до повноцінної системи архівування.

Структура вхідних даних формується на основі параметризованого текстового запиту та конфігурації інференсу. Ключовими елементами є: `prompt` (текстовий опис сцени), `seed` (початкове значення генератора випадковості), `steps` (кількість кроків інференсу), `guidance_scale` (сила текстового умовлення). Додатково можуть передаватися параметри розміру кадру та кількість кадрів. Така структура забезпечує однозначну ідентифікацію експерименту: при

фіксованих значеннях цих параметрів алгоритм здатен відтворити ідентичний результат, що є принциповим для наукової достовірності.

З точки зору типізації даних, `prompt` є текстовим полем довільної довжини, `seed` та `steps` — цілими числами, а `guidance_scale` — дійсним числом із плаваючою комою. Формалізація типів дозволяє здійснювати валідацію на етапі введення та зменшує ймовірність некоректної конфігурації пайплайна. Узагальнену структуру вхідних параметрів генерації наведено на рисунку 2.4, де показано логічну модель пакета параметрів інференсу.

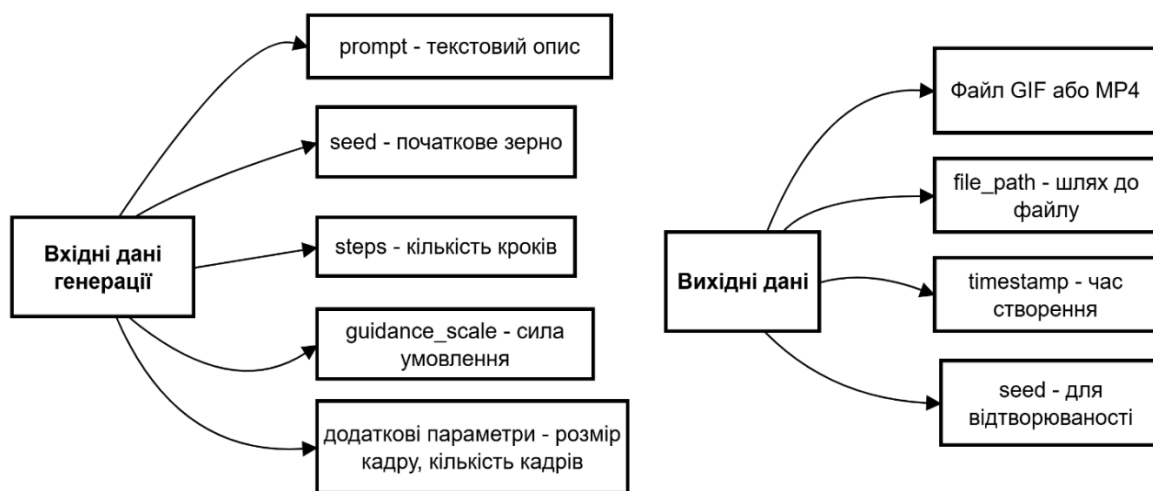


Рисунок 2.4 – Структура вхідних та вихідних даних генерації

Вихідні дані модулю є результатом декодування латентних представлень у послідовність кадрів та їх подальшого об'єднання у відеофайл. У поточній реалізації підтримується експорт у формат GIF, проте архітектура дозволяє розширення до mp4 або інших контейнерів. Вихідний файл супроводжується мінімальним набором метаданих: шлях до файлу (`file_path`), час створення (`timestamp`), а також значення `seed`, що забезпечує повторюваність генерації.

З інженерної точки зору, збереження лише зображень або відеофайлів є недостатнім для забезпечення відтворюваності. Тому інформаційна модель повинна містити параметри генерації як окремий структурований об'єкт. У межах розробленого рішення доцільно використовувати агреговане поле `generation_parameters`, яке зберігає JSON-подібну структуру з усіма параметрами

інференсу (steps, guidance\_scale, num\_frames, height, width тощо). Це дозволяє відновити повний контекст генерації без необхідності ручного документування.

Проектування моделі зберігання передбачає створення логічної сутності «Генерація», що пов'язує користувача з результатом виконання алгоритму. Основними атрибутами такої сутності є: user\_id, prompt, generation\_parameters, file\_path, timestamp. Такий підхід забезпечує нормалізацію даних і дозволяє реалізувати механізм журналювання експериментів, що є важливим для дослідницьких і освітніх застосувань.

Загальна ER-модель зберігання інформації подана на рисунку 2.5, де показано зв'язок між користувачем і результатами генерації. Запропонована структура легко масштабується до реляційної або документно-орієнтованої бази даних і може бути інтегрована з системами керування цифровими архівами.

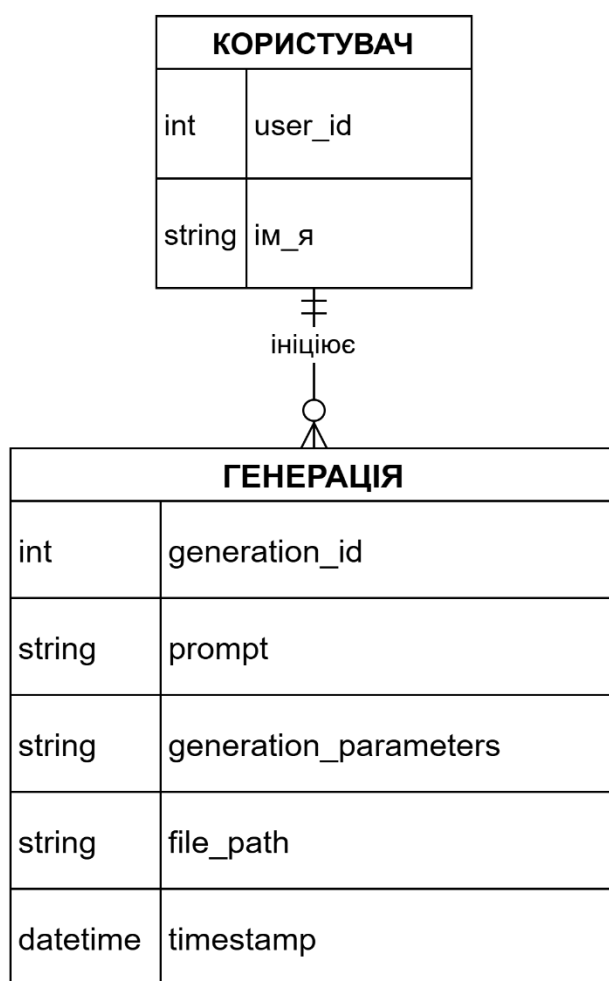


Рисунок 2.5 – Логічна модель зберігання результатів генерації

Таким чином, інформаційне забезпечення програмного модуля не обмежується передачею параметрів у пайплайн генерації, а формує системну основу для відтворюваності, керованості та архівування результатів. Це є принциповим у контексті цифрової культурної спадщини, де важливо не лише створити візуальний контент, а й забезпечити його документованість і можливість повторного використання.



## 3 ПРОГРАМНО-ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

### 3.1 Програмно-технологічна реалізація модуля генерації відеоконтенту

Реалізацію програмного модуля здійснено у середовищі Kaggle/Jupyter, що забезпечує інтерактивність розроблення, доступ до GPU-ресурсів та відтворюваність експериментів. Ключові залежності системи встановлюються безпосередньо у ноутбучі, що дозволяє зафіксувати конкретні версії бібліотек та уникнути конфліктів сумісності.

Зокрема, у роботі використано бібліотеку `diffusers==0.23.1`, що містить реалізацію пайплайна `AnimateDiff`, а також `ipywidgets>=8.0.0` для побудови інтерактивного інтерфейсу користувача. Фрагмент коду встановлення залежностей наведено на рисунку 3.1.

```
%%capture
!pip install diffusers==0.23.1

%%capture
!pip -q install -U "ipywidgets>=8.0.0" "widgetsnbextension>=4.0.0"
```

Рисунок 3.1 – Встановлення ключових залежностей у середовищі виконання

Після встановлення залежностей виконується ініціалізація обчислювального середовища. Насамперед перевіряється доступність GPU-прискорення через `torch.cuda.is_available()`, що дозволяє автоматично обрати пристрій виконання (`cuda` або `cpu`). Такий підхід підвищує універсальність програмного модуля та дозволяє адаптувати його до різних апаратних конфігурацій.

Далі завантажується `MotionAdapter`, який відповідає за формування узгодженого руху між кадрами. Базова модель `SG161222/Realistic_Vision_V5.1_noVAE` використовується як джерело латентного простору для генерації реалістичних зображень. Конфігурація

scheduler (DDIMScheduler) визначає часову дискретизацію зворотного дифузійного процесу.

Крім того, інтегрується LoRA-модуль zoom-out із пониженою вагою 0.35, що дозволяє контролювати динаміку сцени без суттєвого посилення артефактів. Оптимізація пам'яті реалізується через використання torch.float16 та активацію pipe.enable\_vae\_slicing().

Повний фрагмент ініціалізації та конфігурації пайплайна наведено на рисунку 3.2.

```
import torch
from diffusers import MotionAdapter, AnimateDiffPipeline, DDIMScheduler

print("CUDA доступна:", torch.cuda.is_available())
device = "cuda" if torch.cuda.is_available() else "cpu"

adapter = MotionAdapter.from_pretrained("guoyww/animatediff-motion-adapter-v1-5-2")

model_id = "SG161222/Realistic_Vision_V5.1_noVAE"
pipe = AnimateDiffPipeline.from_pretrained(
    model_id,
    motion_adapter=adapter,
    torch_dtype=torch.float16 if device == "cuda" else torch.float32
)

scheduler = DDIMScheduler.from_pretrained(
    model_id,
    subfolder="scheduler",
    clip_sample=False,
    timestep_spacing="linspace",
    steps_offset=1
)
pipe.scheduler = scheduler

pipe.load_lora_weights("guoyww/animatediff-motion-lora-zoom-out", adapter_name="zoom-out")
pipe.set_adapters(["zoom-out"], adapter_weights=[0.35])

pipe.to(device)
pipe.enable_vae_slicing()
```

Рисунок 3.2 – Ініціалізація та конфігурація пайплайна AnimateDiff

Центральним компонентом програмного модуля є функція generate\_animation(), яка інкапсулює весь процес інференсу: формування аргументів виклику пайплайна, ініціалізацію генератора випадковості, запуск дифузійного декодування та експорт результату у формат GIF.

Для забезпечення відтворюваності експериментів використовується фіксація значення seed через torch.Generator(...).manual\_seed(seed). Це дозволяє

повторно отримати ідентичний результат за незмінних параметрів генерації. Основна логіка функції наведена на рисунку 3.3.

Оскільки різні версії пайплайнів можуть відрізнятися підтримкою параметрів `height` та `width`, у реалізації передбачено механізм автоматичної перевірки сигнатури виклику. Це підвищує сумісність модуля та зменшує ризик помилок виконання. Відповідний фрагмент подано на рисунку 3.4.

```
def generate_animation(prompt, negative_prompt, num_frames,
                      num_inference_steps, guidance_scale,
                      seed, out_name):

    gen = torch.Generator(device="cuda").manual_seed(int(seed))

    output = pipe(
        prompt=prompt,
        negative_prompt=negative_prompt,
        num_frames=num_frames,
        guidance_scale=guidance_scale,
        num_inference_steps=num_inference_steps,
        generator=gen,
    )

    frames = output.frames[0]
    export_to_gif(frames, out_name)
    return out_name
```

Рисунок 3.3 – Реалізація функції `generate_animation()`

```
import inspect

sig = inspect.signature(pipe.__call__)

if height is not None and "height" in sig.parameters:
    kwargs["height"] = int(height)

if width is not None and "width" in sig.parameters:
    kwargs["width"] = int(width)
```

Рисунок 3.4 – Автоперевірка підтримки параметрів `height/width`

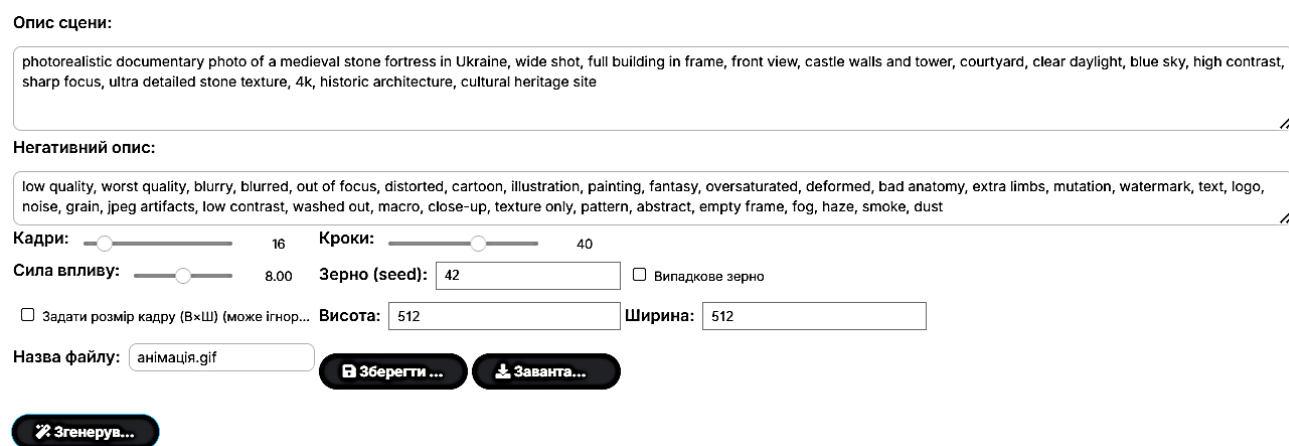
Отже, реалізовано повний програмно-технологічний контур модуля генерації відеоконтенту на основі AnimateDiff у середовищі Kaggle/Jupyter: зафіксовано критичні залежності та забезпечено їх відтворювану інсталяцію, виконано коректну ініціалізацію пайплайна (MotionAdapter, базова модель, DDIMScheduler) і налаштовано керовану інтеграцію LoRA з регулюванням ваги для стабілізації якості. Ядро генерації інкапсульовано у функції `generate_animation()`, де формалізовано параметри інференсу, забезпечено відтворюваність результатів через фіксацію `seed` та реалізовано сумісність із різними конфігураціями пайплайна шляхом автоматичної перевірки підтримки `height/width`. Сукупно ці рішення формують модульну, ресурсно оптимізовану та відтворювану реалізацію, що створює основу для подальшого розгортання інтерфейсу користувача й проведення системного тестування в підрозділі 3.2–3.3.

### 3.2 Реалізація інтерфейсу користувача та сценарії взаємодії

Інтерфейс користувача програмного модуля реалізовано засобами бібліотеки `ipywidgets`, що забезпечує інтерактивну взаємодію в середовищі Jupyter/Kaggle без необхідності розгортання окремого вебзастосунку. Обраний підхід дозволяє інтегрувати параметризацію генеративної моделі безпосередньо в дослідницький робочий процес, забезпечуючи контроль над ключовими змінними інференсу та оперативну візуалізацію результатів.

Інтерфейс побудовано з використанням українськомовних підписів полів та елементів керування, що відповідає вимогам локалізації програмного продукту. Логіка компоновання передбачає групування пов'язаних параметрів (кадри, кроки, сила впливу, зерно, розмір кадру) у горизонтальні контейнери (HBox), що підвищує ергономічність сприйняття. Загальна структура інтерфейсу наведена на рисунку 3.5.

Центральним елементом інтерфейсу є текстове поле «Опис сцени», яке дозволяє вводити семантичний запит (prompt) до дифузійної моделі. Додатково передбачено поле «Негативний опис», яке використовується для обмеження небажаних характеристик (артефакти, розмиття, шум, логотипи тощо). Такий механізм відповідає сучасній практиці prompt engineering і підвищує керованість результатів генерації.



Опис сцени:

photorealistic documentary photo of a medieval stone fortress in Ukraine, wide shot, full building in frame, front view, castle walls and tower, courtyard, clear daylight, blue sky, high contrast, sharp focus, ultra detailed stone texture, 4k, historic architecture, cultural heritage site

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out, macro, close-up, texture only, pattern, abstract, empty frame, fog, haze, smoke, dust

Кадри: 16 Кроки: 40

Сила впливу: 8.00 Зерно (seed): 42 ☐ Випадкове зерно

☐ Задати розмір кадру (ВхШ) (може ігнор... Висота: 512 Ширина: 512

Назва файлу: анімація.gif

Рисунок 3.5 – Інтерфейс користувача модуля генерації анімації в середовищі Jupyter

Параметри інференсу реалізовано через інтерактивні повзунки:

- Кадри (num\_frames) — визначає тривалість анімації;
- Кроки (num\_inference\_steps) — кількість ітерацій зворотного дифузійного процесу;
- Сила впливу (guidance\_scale) — інтенсивність текстового умовлення.

Для забезпечення відтворюваності передбачено поле «Зерно (seed)» та прапорець «Випадкове зерно», що дозволяє або зафіксувати генератор випадковості, або автоматично створити нове значення на основі часової мітки.

Додатково інтерфейс містить опційний блок встановлення розміру кадру (висота/ширина), який активується через відповідний прапорець. Поле «Назва файлу» дозволяє керувати іменем згенерованого GIF-файлу, що є важливим для організації архіву результатів.

Фрагмент створення основних елементів UI наведено на рисунку 3.6.

```
опис_w = widgets.Textarea(
    description="Опис сцени:",
    layout=widgets.Layout(width="100%", height="120px"),
)

негатив_w = widgets.Textarea(
    description="Негативний опис:",
    layout=widgets.Layout(width="100%", height="90px"),
)

кадри_w = widgets.IntSlider(value=16, min=8, max=64, step=1, description="Кадри:")
кроки_w = widgets.IntSlider(value=35, min=10, max=60, step=1, description="Кроки:")
вплив_w = widgets.FloatSlider(value=6.5, min=1.0, max=15.0, step=0.5, description="Сила впливу:")

зерно_w = widgets.IntText(value=42, description="Зерно (seed):")
випадкове_зерно_w = widgets.Checkbox(value=False, description="Випадкове зерно")
```

Рисунок 3.6 – Створення ключових елементів інтерфейсу ipywidgets

Механізм взаємодії з користувачем реалізовано через систему обробників подій (on\_click), що забезпечують реакцію на натискання кнопок керування. Основним сценарієм є виклик функції on\_generate\_clicked(), яка:

1. Очищає область виведення;
2. Перевіряє коректність введених даних;
3. Ініціює генерацію через generate\_animation();
4. Оновлює статус виконання;
5. Відображає результат у вигляді GIF.

Обробка виняткових ситуацій (наприклад, порожній prompt або помилки інференсу) реалізована через конструкцію try/except, що запобігає аварійному завершенню виконання ноутбука та дозволяє вивести інформативне повідомлення користувачу.

Додатково передбачено два допоміжні сценарії:

— on\_save\_cory\_clicked() — створення копії файлу з часовою міткою для архівування;

— on\_download\_clicked() — формування посилання на завантаження результату.

Візуалізація згенерованого результату здійснюється через об'єкт `IPyImage`, а доступ до файлу — через `FileLink`. Фрагмент логіки обробки події генерації наведено на рисунку 3.7.

```
def on_generate_clicked(_):
    try:
        статус_w.value = "<b>Статус:</b> генерую..."
        path = generate_animation(
            prompt=опис_w.value,
            negative_prompt=негатив_w.value,
            num_frames=кадри_w.value,
            num_inference_steps=кроки_w.value,
            guidance_scale=вплив_w.value,
            seed=зерно_w.value,
            out_name=назва_файлу_w.value
        )
        display(IPyImage(open(path, "rb").read()))
        display(FileLink(path))
        статус_w.value = "<b>Статус:</b> успішно ✓"
    except Exception as e:
        статус_w.value = "<b>Статус:</b> помилка ✗"
        display(str(e))
```

Рисунок 3.7 – Обробник події генерації та відображення результату

Таким чином, реалізований інтерфейс користувача забезпечує повний цикл взаємодії з модулем генерації відеоконтенту: від параметризації сцени та конфігурації інференсу до візуалізації та збереження результатів. Поєднання локалізованого UI, керованості параметрами та обробки помилок формує зручний інструмент для експериментального створення візуального контенту в задачах цифрової культурної спадщини.

### 3.3 Тестування та оцінювання результатів роботи модуля

Тестування програмного модуля генерації відеоконтенту на основі `AnimateDiff` виконувалося з метою підтвердження функціональної коректності, стабільності роботи конвеєра, керованості результату параметрами інференсу та відтворюваності (*reproducibility*) за фіксованих початкових умов. Функціональна коректність розуміється як успішне проходження повного циклу: введення

текстового запиту та параметрів → виконання інференсу → формування послідовності кадрів → експорт у GIF → відображення результату та формування посилання на завантаження.

Експерименти проведено в уніфікованих умовах, де базові параметри формували контрольоване середовище порівняння між сценаріями. Зокрема, у більшості тестів застосовано `num_frames = 16`, розмір кадру  $512 \times 512$ , а кількість кроків інференсу варіювалася в діапазоні 35–40. Для перевірки відтворюваності застосовано фіксовані значення `seed` (42, 123, 77, 256, 99, 314). Важливо, що в реалізації передбачено стабілізацію значень `guidance_scale` (для зменшення артефактів), тому у вихідних повідомленнях фіксується фактична сила впливу, яка в більшості тестів становить 6.50, а в окремих — 7.00.

Для емпіричної перевірки працездатності модуля сформовано 6 сценаріїв, що репрезентують типові задачі цифрової культурної спадщини: (1) архітектурний екстер'єр, (2) сакральний інтер'єр, (3) музейний артефакт, (4) реставраційна реконструкція фрагмента, (5) цифрова музейна експозиція, (6) аероперспектива (дрон). Усі сценарії виконувалися через єдиний інтерфейс із параметризацією `prompt/negative prompt` та керуванням кроками, кадрами, `seed` і (за потреби) розміром кадру.

У частині контролю якості ключову роль відіграло використання базового негативного опису (орієнтованого на усунення розмиття, шуму, артефактів стиснення, небажаних написів/водяних знаків), а також спеціалізованих розширень `negative prompt`:

- `anti-texture / anti-macro` — для уникнення залипання моделі на фактурі та «порожніх» текстурних площинах у сценах архітектури/експозиції;

- `anti-tilt-shift / anti-miniature` — для сцен аерозйомки, щоб зменшити ефект «іграшковості» та діорамічності.

Окремо керованість забезпечувалася режимом LoRA (`zoom-out`) зі зниженою вагою (режим “`low`”), що дозволяло стабілізувати динаміку без суттєвого посилення шумів. Отримані результати для кожного сценарію подано



на рисунках 3.8–3.13, причому в підписах зафіксовано ключові кількісні параметри (кадри, кроки, seed, фактична сила впливу).

Опис сцени:

photorealistic documentary photo of a medieval stone fortress in Ukraine, wide shot, full building in frame, front view, castle walls and tower, courtyard, clear daylight, blue sky, high contrast, sharp focus, ultra detailed stone texture, 4k, historic architecture, cultural heritage site

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out, macro, close-up, texture only, pattern, abstract, empty frame, fog, haze, smoke, dust

Кадри:



16

Кроки:



40

Сила впливу:



8.00

Зерно (seed):

42

☐ Випадкове зерно

☐ Задати розмір кадру (В×Ш) (може ігнор...

Висота:

512

Ширина:

512

Назва файлу: анімація.gif

Зберегти ...

Заванта...

Готово. Файл: анімація.gif | Зерно: 42 | Кроки (факт): 40 | Сила (факт): 6.50



Завантаження:

Рисунок 3.8 – Результат для сценарію “Архітектура (екстер’єр)” (16 кадрів; 40 кроків; seed = 42; сила впливу (факт) = 6.50)

Опис сцени:

photorealistic documentary interior photo of an 18th century Ukrainian wooden church, iconostasis, authentic icon painting, candlelight, realistic shadows, warm tones, sharp focus, ultra detailed, museum quality, cinematic lighting

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out

Кадри:  16 Кроки:  38  
Сила впливу:  7.50 Зерно (seed):  ☐ Випадкове зерно  
☐ Задати розмір кадру (В×Ш) (може ігнор... Висота:  Ширина:   
Назва файлу:

Готово. Файл: анімація.gif | Зерно: 123 | Кроки (факт): 38 | Сила (факт): 6.50



Завантаження:

Натисніть для завантаження: [анімація.gif](#)

 Завантажити файл

Рисунок 3.9 – Результат для сценарію “Сакральний інтер’єр” (16 кадрів;  
38 кроків; seed = 123; сила впливу (факт) = 6.50)

Опис сцени:

museum catalog photo of an archaeological ceramic artifact (Trypillian style), neutral background, studio soft lighting, high detail ornament, macro photography, sharp focus, ultra detailed, realistic material

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out

Кадри:  12

Кроки:  35

Сила впливу:  7.00

Зерно (seed):  ☐ Випадкове зерно

☐ Задати розмір кадру (В×Ш) (може ігнор...

Висота:

Ширина:

Назва файлу:

Зберегти ...

Заванта...

Згенерув...

Статус: успішно

Готово. Файл: анімація.gif | Зерно: 77 | Кроки (факт): 35 | Сила (факт): 7.00



Завантаження:

Натисніть для завантаження: [анімація.gif](#)

Завантажити файл

Рисунок 3.10 – Результат для сценарію “Музейний артефакт” (12 кадрів;  
35 кроків; seed = 77; сила впливу (факт) = 7.00)

Опис сцени:

photorealistic restoration reconstruction of a damaged fresco fragment, authentic pigments, delicate plaster texture, conservative restoration look, museum documentation style, natural soft light, sharp focus, ultra detailed

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out

Кадри:  16

Кроки:  38

Сила впливу:  7.50

Зерно (seed):

☐ Випадкове зерно

☐ Задати розмір кадру (В×Ш) (може ігнор...

Висота:

Ширина:

Назва файлу:

 Зберегти ...

 Заванта...

 Згенерув...

Статус: генерую...

Генерація... це може зайняти деякий час.

Статус: успішно 

Готово. Файл: анімація.gif | Зерно: 256 | Кроки (факт): 38 | Сила (факт): 6.50



Завантаження:

Натисніть для завантаження: [анімація.gif](#)

 Завантажити файл

Рисунок 3.11 – Результат для сценарію “Реставраційна реконструкція” (16 кадрів; 38 кроків; seed = 256; сила впливу (факт) = 6.50)

Опис сцени:

modern digital museum exhibition hall, interactive panels, realistic 3D visualization of a historical fortress, clean minimal design, soft white lighting, realistic rendering, sharp focus, ultra detailed, 4k

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out

Кадри:



16

Кроки:



35

Сила впливу:



7.20

Зерно (seed):

99

☐ Випадкове зерно

☐ Задати розмір кадру (В×Ш) (може ігнор...

Висота:

512

Ширина:

512

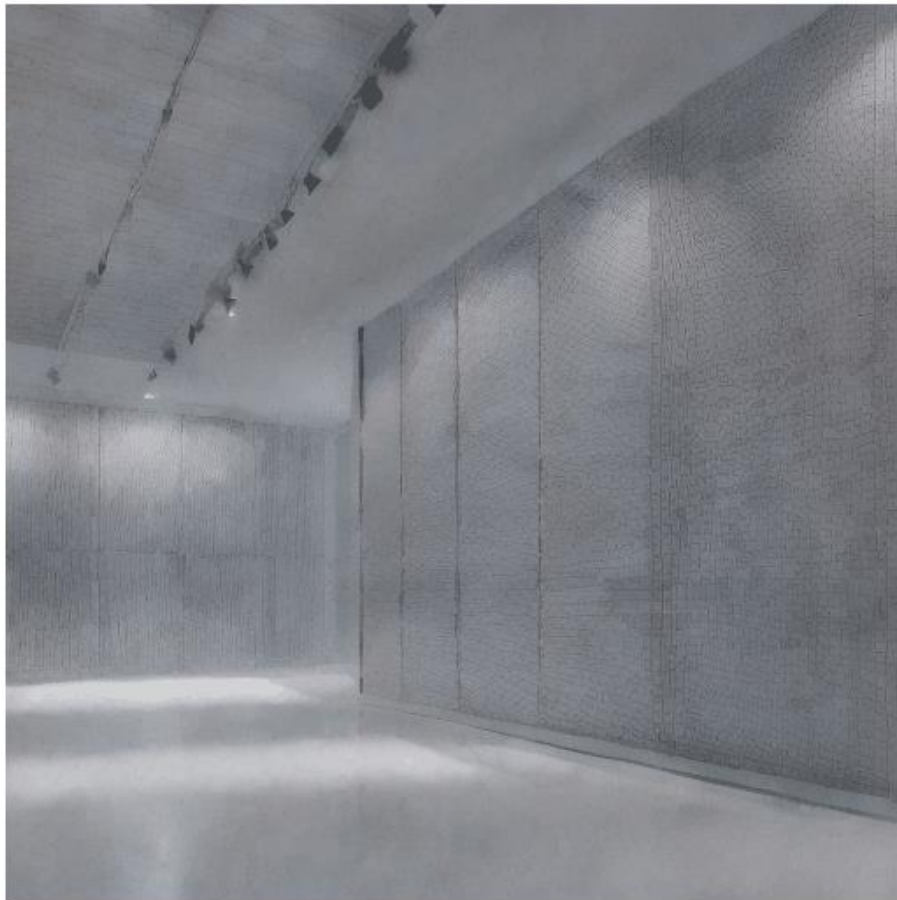
Назва файлу:

Зберегти ...

Заванта...

Згенерув...

Готово. Файл: анімація.gif | Зерно: 99 | Кроки (факт): 35 | Сила (факт): 6.50



Завантаження:

Натисніть для завантаження: [анімація.gif](#)

Завантажити файл

Рисунок 3.12 – Результат для сценарію “Цифрова музейна експозиція” (16 кадрів; 35 кроків; seed = 99; сила впливу (факт) = 6.50)



Опис сцени:

photorealistic drone aerial video of a historic fortress in Ukraine, wide establishing shot, full fortress visible, landscape context, sunset golden hour, natural colors, smooth camera motion, documentary style, sharp focus, ultra detailed, 4k

Негативний опис:

low quality, worst quality, blurry, blurred, out of focus, distorted, cartoon, illustration, painting, fantasy, oversaturated, deformed, bad anatomy, extra limbs, mutation, watermark, text, logo, noise, grain, jpeg artifacts, low contrast, washed out, macro, close-up, texture only, pattern, abstract, empty frame, fog, haze, smoke, dust, miniature, tilt-shift, toy, model, diorama

Кадри:  16 Кроки:  40  
Сила впливу:  7.80 Зерно (seed):  ☐ Випадкове зерно  
☐ Задати розмір кадру (В×Ш) (може ігнор... Висота:  Ширина:   
Назва файлу:

Готово. Файл: анімація.gif | Зерно: 314 | Кроки (факт): 40 | Сила (факт): 6.50



Завантаження:

Натисніть для завантаження: анімація.gif

Завантажити файл

Рисунок 3.13 – Результат для сценарію “Аероперспектива (дрон)” (16 кадрів; 40 кроків; seed = 314; сила впливу (факт) = 6.50)

Узагальнені параметри шести тестових сценаріїв (кількість кадрів, фактична кількість кроків інференсу, значення seed, фактична сила впливу та формат вихідного артефакту) наведено в таблиці 3.1.

Експериментальна частина була організована як контрольована варіація параметрів інференсу в межах, що є практично доцільними для генерації

коротких відеопослідовностей у ресурсно обмеженому середовищі. Основні варійовані параметри включали:

- кількість кроків інференсу *num\_inference\_steps*: 35, 38, 40;
- кількість кадрів *num\_frames*: переважно 16, а для предметної сцени артефакту — 12;
- сила текстового умовлення *guidance\_scale*: у вихідних результатах зафіксовано фактичні значення 6.50 або 7.00;
- *seed* як інструмент контролю відтворюваності: 42, 123, 77, 256, 99, 314;
- просторова роздільна здатність кадру: 512×512 (як стандартна конфігурація тестів).

Таблиця 3.1 – Параметри тестових сценаріїв та вихідні артефакти генерації

| № | Тестовий сценарій (цифрова культурна спадщина) | Кількість кадрів, <i>num_frames</i> | Кроки інференсу, <i>steps</i> (факт) | Зерно, <i>seed</i> | Сила впливу, <i>guidance</i> (факт) | Розмір кадру |
|---|------------------------------------------------|-------------------------------------|--------------------------------------|--------------------|-------------------------------------|--------------|
| 1 | Архітектура (екстер'єр)                        | 16                                  | 40                                   | 42                 | 6.50                                | 512×512      |
| 2 | Сакральний інтер'єр                            | 16                                  | 38                                   | 123                | 6.50                                | 512×512      |
| 3 | Музейний артефакт (кераміка)                   | 12                                  | 35                                   | 77                 | 7.00                                | 512×512      |
| 4 | Реставраційна реконструкція (фрагмент)         | 16                                  | 38                                   | 256                | 6.50                                | 512×512      |
| 5 | Цифрова музейна експозиція                     | 16                                  | 35                                   | 99                 | 6.50                                | 512×512      |
| 6 | Аероперспектива (дрон)                         | 16                                  | 40                                   | 314                | 6.50                                | 512×512      |

Критерії оцінювання включали дві взаємодоповнювальні групи. Технічні критерії охоплювали: (i) стабільність завершення інференсу з формуванням вихідного файлу, (ii) керованість виходу через зміну кроків/кадрів/силу впливу, (iii) відтворюваність за фіксованого *seed*. Якісні критерії експертної оцінки були зорієнтовані на задачі цифрової культурної спадщини: (i) семантична

відповідність текстовому опису (об'єкт/сцена), (ii) візуальна правдоподібність і деталізація, (iii) відсутність критичних артефактів (розмиття, шум, небажані написи), (iv) придатність для презентаційного/музейного контенту.

Отримані результати (див. рисунки 3.8–3.13) демонструють, що модуль забезпечує керовану генерацію в умовах 35–40 кроків та 12–16 кадрів на фіксованій роздільній здатності 512×512, а також підтримує відтворюваність через використання заданих seed. Практичний ефект спеціалізованих negative prompt проявляється у зменшенні небажаної «текстуризації» та зниженні ймовірності артефактів у сценах, чутливих до перспективи та масштабу (архітектура, експозиційні простори, аерозйомка).



## ВИСНОВКИ

У кваліфікаційній роботі досягнуто поставленої мети — розроблено та експериментально перевірено програмний модуль керованої генерації короткого відеоконтенту (GIF) за текстовим описом сцени на основі дифузійного підходу AnimateDiff із забезпеченням відтворюваності результатів через фіксацію параметрів інференсу та реалізацію базового механізму архівування згенерованих артефактів.

За результатами виконання поставлених завдань сформульовано такі висновки.

1. У процесі аналізу предметної області цифрової культурної спадщини встановлено, що сучасні інституції культурної пам'яті потребують не лише оцифрування об'єктів, а й масштабованого створення мультимедійного контенту для онлайн-експозицій, освітніх матеріалів і популяризаційних ресурсів. Обґрунтовано доцільність використання генеративних моделей як інструменту автоматизованого синтезу візуального контенту, що дозволяє зменшити часові витрати на створення коротких відеофрагментів та підвищити гнучкість формування цифрових наративів.

2. У межах дослідження підходів генеративного моделювання (GAN, VAE, diffusion) встановлено, що дифузійна парадигма є найбільш придатною для задач синтезу відеопослідовностей у прикладному освітньо-дослідницькому контексті. Порівняльний аналіз показав, що GAN характеризуються складністю стабільного навчання, VAE — компромісом між регуляризацією та якістю зображень, тоді як дифузійні моделі забезпечують високу візуальну якість, керованість через текстове умовлення та параметризацію процесу інференсу. Це стало підставою для вибору підходу AnimateDiff як розширення дифузійних Text-to-Image моделей до формування узгоджених відеокадрів.

3. У результаті аналізу наявних інструментів генерації відео (open-source та пропрієтарних) сформовано критерії порівняння, що включають керованість параметрами (seed, steps, guidance), відтворюваність експериментів, ресурсні

вимоги (GPU/VRAM), формат вихідного результату та можливість локального розгортання. Обґрунтовано переваги open-source екосистеми (Stable Diffusion + AnimateDiff) для академічного середовища завдяки прозорості налаштувань, контролю параметрів та можливості відтворюваного запуску в середовищі Jupyter/Kaggle.

4. Спроектовано архітектуру програмного модуля як параметризований конвеєр генерації, що охоплює обробку текстового запиту, формування набору параметрів інференсу, ініціалізацію генератора випадковості з фіксацією seed, виклик дифузійного пайплайна та експорт результату у формат GIF. Реалізовано алгоритмічний конвеєр AnimateDiff із підтримкою ключових параметрів (prompt, negative\_prompt, num\_frames, num\_inference\_steps, guidance\_scale, seed, height, width), що забезпечує керованість і відтворюваність результатів.

5. Розроблено інтерфейс взаємодії користувача засобами ipywidgets у середовищі Jupyter, який підтримує введення текстового опису сцени, налаштування параметрів інференсу, валідацію даних, відображення згенерованого результату та виконання операцій збереження і завантаження GIF-файлу. Реалізовано механізми обробки помилок та інформування користувача про стан виконання, що підвищує надійність і зручність використання модуля.

6. Проведено експериментальне тестування модуля на шести сценаріях, релевантних цифровій культурній спадщині (архітектура, сакральний інтер'єр, музейний артефакт, реставраційна реконструкція, цифрова експозиція, аероперспектива). Тестування здійснювалося за контрольованих параметрів (num\_frames = 12–16, num\_inference\_steps = 35–40, роздільна здатність 512×512, фіксовані значення seed). Підтверджено стабільність завершення інференсу, коректність формування GIF-файлів та відтворюваність результатів за незмінних параметрів. Аналіз показав, що варіювання кількості кроків інференсу та сили умовлення дозволяє керувати балансом між деталізацією кадрів і обчислювальними витратами, а використання фіксованого seed забезпечує повторюваність експериментів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. UNESCO. Charter on the Preservation of the Digital Heritage : adopted 15 October 2003, Paris. UNESCO, 2003. URL: <https://www.unesco.org/en/legal-affairs/charter-preservation-digital-heritage> (дата звернення: 27.02.2026).
2. European Commission. Digital cultural heritage. Shaping Europe's digital future. European Commission, DG CNECT. Оновлено 11 October 2024. URL: <https://digital-strategy.ec.europa.eu/en/policies/cultural-heritage> (дата звернення: 27.02.2026).
3. Remondino F. Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. *Remote Sensing*. 2011. Vol. 3, No. 6. P. 1104–1138. DOI: 10.3390/rs3061104.
4. Goodfellow I. J., Pouget-Abadie J., Mirza M. et al. Generative Adversarial Nets. *arXiv*. 2014. URL: <https://arxiv.org/abs/1406.2661> (дата звернення: 27.02.2026).
5. Kingma D. P., Welling M. Auto-Encoding Variational Bayes. *arXiv*. 2013. DOI: 10.48550/arXiv.1312.6114. URL: <https://arxiv.org/abs/1312.6114> (дата звернення: 27.02.2026).
6. Ho J., Jain A., Abbeel P. Denoising Diffusion Probabilistic Models. *NeurIPS*. 2020. URL: <https://neurips.cc/virtual/2020/poster/17168> (дата звернення: 27.02.2026).
7. Just Tech (SSRC). Generative Adversarial Nets: Citation record. 2014. URL: <https://just-tech.ssrc.org/citation/generative-adversarial-nets/> (дата звернення: 27.02.2026).
8. Harvard ADS. Auto-Encoding Variational Bayes: abstract record. 2013. URL: <https://ui.adsabs.harvard.edu/abs/2013arXiv1312.6114K/abstract> (дата звернення: 27.02.2026).
9. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High-Resolution Image Synthesis with Latent Diffusion Models. *CVPR*. 2022. URL: <https://arxiv.org/abs/2112.10752> (дата звернення: 27.02.2026).

10. Stable Diffusion Documentation. Welcome to Stable Diffusion (overview). 2022. URL: <https://stablediffusion.gitbook.io/overview> (дата звернення: 27.02.2026).
11. Radford A., Kim J. W., Hallacy C. et al. Learning Transferable Visual Models From Natural Language Supervision (CLIP). *arXiv*. 2021. URL: <https://arxiv.org/abs/2103.00020> (дата звернення: 27.02.2026).
12. Ho J., Salimans T., Gritsenko A. et al. Video Diffusion Models. *arXiv*. 2022. DOI: 10.48550/arXiv.2204.03458. URL: <https://arxiv.org/abs/2204.03458> (дата звернення: 27.02.2026).
13. Ho J., Chan W., Saharia C. et al. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv*. 2022. URL: <https://arxiv.org/abs/2210.02303> (дата звернення: 27.02.2026).
14. Singer U., Polyak A., Hayes T. et al. Make-A-Video: Text-to-Video Generation without Text-Video Data. *arXiv*. 2022. DOI: 10.48550/arXiv.2209.14792. URL: <https://arxiv.org/abs/2209.14792> (дата звернення: 27.02.2026).
15. Phenaki: Variable Length Video Generation From Open Domain Textual Description. *arXiv*. 2022. URL: <https://arxiv.org/abs/2210.02399> (дата звернення: 27.02.2026).
16. Guo Y., Yang C., Rao A. et al. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. *arXiv*. 2023. DOI: 10.48550/arXiv.2307.04725. URL: <https://arxiv.org/abs/2307.04725> (дата звернення: 27.02.2026).
17. AnimateDiff Project Page (ICLR'24 Spotlight). AnimateDiff official page. URL: <https://animatediff.github.io/> (дата звернення: 27.02.2026).
18. Guoyww. AnimateDiff (official implementation) — GitHub repository. URL: <https://github.com/guoyww/AnimateDiff> (дата звернення: 27.02.2026).
19. Springer Nature. Text-to-video generators: a comprehensive survey. *Journal of Big Data*. 2025. URL: <https://link.springer.com/article/10.1186/s40537-025-01314-3> (дата звернення: 27.02.2026).

20. Runway. Introducing Gen-2: Text to Video (official video). YouTube. 2023. URL: <https://www.youtube.com/watch?v=trXPfpV5iRQ> (дата звернення: 27.02.2026).
21. Schneider J. Runway Gen-2 is the First Publicly Available Text-to-Video Generator. *PetaPixel*. 2023. URL: <https://petapixel.com/2023/03/20/runway-gen-2-is-the-first-publicly-available-text-to-video-generator/> (дата звернення: 27.02.2026).
22. Rosic A. Runway AI Gen-2: Text To Video AI Is Real Now. *Dataconomy*. 2023. URL: <https://dataconomy.com/2023/03/21/what-is-runway-ai-gen-2-text-to-video-ai/> (дата звернення: 27.02.2026).
23. Pika AI by Pika Labs — AI Video Generator (release overview). URL: <https://pikaais.com/> (дата звернення: 27.02.2026).
24. Hong W., Ding M., Zheng W. et al. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. *arXiv*. 2022. URL: <https://arxiv.org/abs/2205.15868> (дата звернення: 27.02.2026).
25. Axios. Meta announces AI-based tool for generating video from text (Make-a-Video). 2022. URL: <https://www.axios.com/2022/09/29/meta-ai-video-generation-tool> (дата звернення: 27.02.2026).
26. The Guardian. Sora: OpenAI launches tool that instantly creates video from text. 2024. URL: <https://www.theguardian.com/technology/2024/feb/15/openai-sora-ai-model-video> (дата звернення: 27.02.2026).
27. Pika Labs (канал). AI Text-to-Video generated with Pika 1.0 (video). YouTube. 2023. URL: <https://www.youtube.com/watch?v=LaWQISyopMY> (дата звернення: 27.02.2026).
28. Stability AI. Stable Video Diffusion (product page). URL: <https://stability.ai/stable-video> (дата звернення: 27.02.2026).
29. The Verge. Runway says its new text-to-video AI generator has ‘unprecedented’ accuracy (news). 2025. URL: <https://www.theverge.com/news/834905/runway-ai-text-video-generator-launch> (дата звернення: 27.02.2026).

30. Kaggle Notebook. Text-to-video with AnimateDiff (Aisuko / community notebook). Kaggle. URL: <https://www.kaggle.com/code/aisuko/text-to-video-with-animatediff> (дата звернення: 27.02.2026).

31. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Коваль В. С., Ліп'яніна-Гончаренко Х. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

32. Мелень В., Патраманська А., Пилипів Л., Ліп'яніна-Гончаренко Х.. Комплексне застосування методів штучного інтелекту для розпізнавання та генерації об'єктів культурної спадщини. *У матеріалах X Міжнародної мультидисциплінарної студентської наукової конференції «Наука сьогодення: від досліджень до стратегічних рішень»*. С. 90–92.

Додаток А  
Програмний код

```
#
=====
# Призначення: генерація GIF-анімації для візуального контенту
# цифрової культурної спадщини.
# Вимоги: GPU (CUDA), встановлені diffusers/transformers/accelerate/ipywidgets.
#
=====

import time, inspect, torch
import ipywidgets as w
from IPython.display import display, clear_output, Image as IPyImage, FileLink
from diffusers import MotionAdapter, AnimateDiffPipeline, DDIMScheduler
from diffusers.utils import export_to_gif

# ----- 1) Ініціалізація пайплайна -----
device = "cuda" if torch.cuda.is_available() else "cpu"

adapter = MotionAdapter.from_pretrained("guoyww/animatediff-motion-adapter-v1-5-2")
model_id = "SG161222/Realistic_Vision_V5.1_noVAE"

pipe = AnimateDiffPipeline.from_pretrained(
    model_id, motion_adapter=adapter,
    torch_dtype=torch.float16 if device == "cuda" else torch.float32
)
pipe.scheduler = DDIMScheduler.from_pretrained(
    model_id, subfolder="scheduler",
    clip_sample=False, timestep_spacing="linspace", steps_offset=1
)
pipe.to(device)
if hasattr(pipe, "vae") and hasattr(pipe.vae, "enable_slicing"):
    pipe.vae.enable_slicing()

# ----- 2) Тестові сценарії (6) -----
NEG_BASE = ("low quality, worst quality, blurry, distorted, cartoon, fantasy,
oversaturated, "
            "deformed, watermark, text, logo, noise, grain, jpeg artifacts, low contrast,
washed out")
NEG_NO_TEX = "macro, close-up, texture only, pattern, abstract, empty frame, fog,
haze, smoke, dust"
NEG_DRONE = "miniature, tilt-shift, toy, model, diorama"
```

```

# ----- 3) Генерація GIF -----
def run_anim(prompt, neg, frames, steps, guidance, seed,
out_name="animation.gif"):
    gen = torch.Generator().manual_seed(int(seed)) # CPU generator (стабільно в
різних runtime)
    kwargs = dict(prompt=prompt, negative_prompt=neg, num_frames=int(frames),
        num_inference_steps=max(30, int(steps)),
        guidance_scale=float(guidance), generator=gen)
    # height/width додаємо тільки якщо підтримується
    try:
        sig = inspect.signature(pipe.__call__)
        if "height" in sig.parameters and "width" in sig.parameters:
            pass
    except Exception:
        pass
    out = pipe(**kwargs).frames[0]
    export_to_gif(out, out_name)
    return out_name

# ----- 4) UI (українською) -----
dd = w.Dropdown(options=list(TESTS.keys()), description="Тест:")
btn = w.Button(description="Згенерувати", button_style="success")
name_w = w.Text(value="анімація.gif", description="Файл:")
out = w.Output()

def on_click(_):
    t = TESTS[dd.value]
    fn = name_w.value.strip() or "анімація.gif"
    if not fn.lower().endswith(".gif"): fn += ".gif"
    with out:
        clear_output(True)
        display(w.HTML("<i>Генерація...</i>"))
    path = run_anim(t["p"], t["n"], t["f"], t["s"], t["g"], t["seed"], fn)
    with out:
        clear_output(True)
        display(w.HTML(f"<b>Готово:</b> <code>{path}</code>"))
        display(IPyImage(open(path, "rb").read()))
        display(FileLink(path, result_html_prefix="Завантажити: "))

btn.on_click(on_click)

```



```
display(w.VBox([w.HTML("<b>Генерація GIF (AnimateDiff) — цифрова  
культурна спадщина</b>"),  
w.HBox([dd, name_w, btn]), out]))
```

Додаток Б  
Апробація отриманих результатів

МАТЕРІАЛИ X МІЖНАРОДНОЇ  
СТУДЕНТСЬКОЇ НАУКОВОЇ  
**КОНФЕРЕНЦІЇ**

НАУКА СЬОГОДЕННЯ:  
ВІД ДОСЛІДЖЕНЬ ДО  
СТРАТЕГІЧНИХ РІШЕНЬ



М. ОДЕСА, УКРАЇНА

**27 ЛЮТОГО  
2026 РІК**



МАТЕРІАЛИ X МІЖНАРОДНОЇ  
СТУДЕНТСЬКОЇ НАУКОВОЇ  
**КОНФЕРЕНЦІЇ**

.....

**НАУКА СЬОГОДЕННЯ:  
ВІД ДОСЛІДЖЕНЬ ДО  
СТРАТЕГІЧНИХ РІШЕНЬ**

.....

м. Одеса, Україна  
27 лютого 2026 рік

Вінниця, Україна  
«UKRLOGOS Group»  
2026

**УДК 082:001  
Н 44**



Голова оргкомітету: Корзняк І.О.

Верстка: Білаус Т.В.

Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 7 від 26.02.2026 року.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 468 від 10.06.2025).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Н 44

.....  
**Наука сьогодення: від досліджень до стратегічних рішень:**  
матеріали X Міжнародної студентської наукової конференції,  
м. Одеса, 27 лютого, 2026 рік / ГО «Молодіжна наукова ліга». —  
Вінниця: ТОВ «УКРЛОГОС Груп», 2026. — 194 с.

ISBN 978-617-8582-23-4

DOI 10.62732/liga-inter-27.02.2026

Викладено матеріали учасників X Міжнародної мультидисциплінарної студентської наукової конференції «Наука сьогодення: від досліджень до стратегічних рішень», яка відбулася 27 лютого 2026 року у місті Одеса, Україна.

**УДК 082:001**

© Колектив учасників конференції, 2026

© ГО «Молодіжна наукова ліга», 2026

© ТОВ «УКРЛОГОС Груп», 2026

ISBN 978-617-8582-23-4

## СЕКЦІЯ 14. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

|                                                                                                                               |    |
|-------------------------------------------------------------------------------------------------------------------------------|----|
| ЕКСПЕРТНІ ЗНАННЯ І ТОЧНІСТЬ ПРОГНОЗУВАННЯ: ПОРІВНЯННЯ PROPHET<br>ТА LSTM                                                      |    |
| Виговський Б.А.                                                                                                               | 86 |
| КОМПЛЕКСНЕ ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ<br>РОЗПІЗНАВАННЯ ТА ГЕНЕРАЦІЇ ОБ'ЄКТІВ ЦИФРОВОЇ КУЛЬТУРНОЇ<br>СПАДЩИНИ |    |
| Мелень В., Патраманська А., Пилипів Л., Науковий керівник: Лип'яніна-Гончаренко Х.                                            | 90 |
| ПОРІВНЯЛЬНИЙ АНАЛІЗ ХМАРНИХ РІШЕНЬ ДЛЯ ВЕБ-СЕРВІСУ З ОРЕНДИ ТА<br>КУПІВЛІ НЕРУХОМОСТІ                                         |    |
| Скульба С.О., Науковий керівник: Цвіркун О.А.                                                                                 | 93 |

## СЕКЦІЯ 15. ТРАНСПОРТ ТА ТРАНСПОРТНІ ТЕХНОЛОГІЇ

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| ВПРОВАДЖЕННЯ МЕТАНОЛУ ЯК НИЗЬКОВУГЛЕЦЕВОГО ПАЛИВА —<br>ЕФЕКТИВНИЙ ШЛЯХ ДО ДЕКАРБОНІЗАЦІЇ |    |
| Гуменюк А.В., Науковий керівник: Левківська А.Л.                                         | 95 |

## СЕКЦІЯ 16. ФІЛОЛОГІЯ ТА ЖУРНАЛІСТИКА

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| ПОЛІТИЧНА МЕТАФОРА В СОЦІОЛІНГВІСТИЧНОМУ АСПЕКТІ (НА<br>МАТЕРІАЛІ ЗМП)  |    |
| Чутро В.С., Науковий керівник: Конопелькіна О.О.                        | 97 |
| ПРО ВАЖЛИВІСТЬ СИСТЕМАТИЗАЦІЇ ТЕРМІНОСИСТЕМИ БЕЗПЕКОВИХ<br>УГОД УКРАЇНИ |    |
| Муранова В.В., Науковий керівник: Коцюк Л.М.                            | 99 |

## СЕКЦІЯ 17. ПЕДАГОГІКА ТА ОСВІТА

|                                                                                                                               |     |
|-------------------------------------------------------------------------------------------------------------------------------|-----|
| CHALLENGES OF ONLINE EDUCATION IN GEORGIA                                                                                     |     |
| Mchedlishvili N., Sadagashvili M., Scientific supervisor: Khorguashvili T.                                                    | 102 |
| АСПЕКТИ РЕАЛІЗАЦІЇ ІНТЕГРОВАНОГО НАВЧАННЯ В ПОЧАТКОВІЙ ШКОЛІ                                                                  |     |
| Цехмістер Б.В., Науковий керівник: Поберецька В.В.                                                                            | 105 |
| ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ НА УРОКАХ ІНФОРМАТИКИ В<br>ПОЧАТКОВІЙ ШКОЛІ                                                   |     |
| Даденко М.С., Науковий керівник: Рибалко О.О.                                                                                 | 108 |
| МЕТОД ПРОБЛЕМНОГО НАВЧАННЯ ЯК ЗАСІБ АКТИВІЗАЦІЇ ПІЗНАВАЛЬНОЇ<br>ДІЯЛЬНОСТІ УЧНІВ У ПРОЦЕСІ ВИВЧЕННЯ НІМЕЦЬКОЇ МОВИ            |     |
| Каменська Н.В.                                                                                                                | 110 |
| ОСОБЛИВОСТІ ЛОГОПЕДИЧНОЇ РОБОТИ З ДІТЬМИ З КОМПЛЕКСНИМИ<br>ПОРУШЕННЯМИ ПСИХОФІЗИЧНОГО РОЗВИТКУ В УМОВАХ ІНКЛЮЗИВНОЇ<br>ОСВІТИ |     |
| Комисарник Є.О., Науковий керівник: Савінова Н.В.                                                                             | 113 |

УДК 082:001  
Н 44



Голова оргкомітету: Корзюк І.О.

Варстка: Білоус Т.В.

Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 7 від 26.02.2026 року.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 468 від 10.06.2025).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Н 44

Наука сьогодення: від досліджень до стратегічних рішень: матеріали X Міжнародної студентської наукової конференції, м. Одеса, 27 лютого, 2026 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2026. — 194 с.

ISBN 978-617-8582-23-4

DOI 10.62732/liga-inter-27.02.2026

Викладено матеріали учасників X Міжнародної мультидисциплінарної студентської наукової конференції «Наука сьогодення: від досліджень до стратегічних рішень», яка відбулася 27 лютого 2026 року у місті Одеса, Україна.

УДК 082:001

© Колектив учасників конференції, 2026

© ГО «Молодіжна наукова ліга», 2026

© ТОВ «УКРЛОГОС Груп», 2026

ISBN 978-617-8582-23-4



**Віталій Мелень**, здобувач вищої освіти факультету комп'ютерних інформаційних технологій

*Західноукраїнський національний університет, Україна*

**Анастасія Патраманська**, здобувач вищої освіти факультету комп'ютерних інформаційних технологій

*Західноукраїнський національний університет, Україна*

**Людмила Пилипів**, здобувач вищої освіти факультету комп'ютерних інформаційних технологій

*Західноукраїнський національний університет, Україна*

**Науковий керівник: Христина Ліп'яніна-Гончаренко**, д-р.техн.наук, доцент, доцент кафедри інформаційно-обчислювальних систем і управління

*Західноукраїнський національний університет, Україна*

## КОМПЛЕКСНЕ ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗПІЗНАВАННЯ ТА ГЕНЕРАЦІЇ ОБ'ЄКТІВ ЦИФРОВОЇ КУЛЬТУРНОЇ СПАДЩИНИ

В епоху глобальної цифровізації суспільства питання збереження, дослідження та популяризації культурної спадщини набуває особливої актуальності. Сучасні музеї, галереї та архіви стикаються з проблемою обробки величезних масивів візуальних даних, які потребують каталогізації, детального атрибутування та реставрації у цифровому форматі. Вирішення цих завдань традиційними методами вимагає значних людських та часових ресурсів. Відтак, впровадження методів штучного інтелекту, зокрема згорткових нейронних мереж (ЗНМ) та генеративних моделей, є перспективним напрямком автоматизації цих процесів [1].

Метою даного дослідження є розробка концептуальної моделі комплексного програмного забезпечення, що об'єднує модулі розпізнавання, класифікації та генерації візуального контенту для ефективного управління об'єктами цифрової культурної спадщини.

Запропонована система складається з трьох ключових взаємопов'язаних підсистем, кожна з яких вирішує специфічний клас завдань.

Перша підсистема відповідає за багатоміткову класифікацію атрибутів мистецьких об'єктів. На відміну від традиційної однокласової класифікації, твори мистецтва мають складну семантику і вимагають одночасного визначення кількох атрибутів: епохи, стилю, жанру, автора та техніки виконання. Для вирішення цього завдання застосовано методи перенесення навчання (transfer learning) на базі глибоких згорткових нейронних мереж (наприклад, архітектур ResNet або EfficientNet). Використання попередньо натренованих моделей дозволяє суттєво зменшити час навчання та підвищити точність класифікації навіть на обмежених наборах даних [2].

Оцінка ефективності модуля багатоміткової класифікації здійснюється за допомогою метрики точності (Ассигасу), що розраховується за формулою (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

де: *TP* – істинно позитивні результати; *TN* – істинно негативні результати; *FP* – хибно позитивні результати; *FN* – хибно негативні результати.

Другою важливою складовою комплексу є програмний модуль розпізнавання архітектурних стилів. Архітектурні пам'ятки є невід'ємною частиною культурного надбання. Модуль призначений для автоматичної ідентифікації стилістичної приналежності будівель (готика, бароко, модерн, конструктивізм тощо) за їх фотографіями. Архітектура цього модуля також базується на комп'ютерному зорі, проте його алгоритми оптимізовані для виявлення специфічних просторових патернів, геометричних форм та елементів декору, притаманних різним епохам.

Третя підсистема – програмний модуль генерації візуального контенту. Якщо перші два модулі виконують аналітичну функцію, то даний модуль розширює можливості взаємодії з цифровою спадщиною через синтез нових зображень. За допомогою генеративно-змагальних мереж (GAN) та дифузійних моделей стає можливим відтворення втрачених фрагментів картин, колоризація історичних чорно-білих фотографій будівель, а також генерація ілюстративних матеріалів для віртуальних екскурсій та освітніх платформ [3].

Функціональний розподіл та взаємозв'язок розроблених модулів представлено у таблиці (табл. 1).

Таблиця 1

Функціональні можливості модулів системи

| Назва модуля               | Базові технології                        | Основне призначення                                        | Сфера застосування                         |
|----------------------------|------------------------------------------|------------------------------------------------------------|--------------------------------------------|
| Аналіз мистецьких об'єктів | ZHM, Transfer Learning, Multi-label      | Визначення автора, стилю, епохи, техніки                   | Цифрові архіви, онлайн-каталоги музеїв     |
| Розпізнавання архітектури  | Комп'ютерний зір, виділення ознак        | Класифікація архітектурних стилів будівель                 | Інтерактивні туристичні карти, реставрація |
| Генерація контенту         | GAN, дифузійні моделі (Stable Diffusion) | Відновлення пошкоджених артефактів, створення візуалізацій | Освітні платформи, VR/AR виставки          |

Комплексна архітектура розробленої системи дозволяє передавати дані між модулями. Наприклад, результати класифікації архітектурного стилю пошкодженої будівлі (модуль 2) можуть слугувати вхідними параметрами (умовами) для модуля генерації (модуль 3), що дозволить алгоритму максимально автентично реконструювати її первісний вигляд.

**Висновки.** Розроблена концепція інтегрованого програмного комплексу демонструє високий потенціал застосування методів машинного навчання у гуманітарній сфері. Об'єднання інструментів багатоміткової класифікації мистецьких творів, розпізнавання архітектурних стилів та генерації візуального контенту створює потужну екосистему для цифрової культурної спадщини.



Застосування перенесення навчання у згорткових нейронних мережах забезпечує високу точність каталогізації, а генеративні технології відкривають нові шляхи для реставрації та візуалізації втрачених пам'яток. Подальші дослідження будуть спрямовані на програмну реалізацію запропонованої архітектури та її тестування на реальних наборах музейних даних.

**Список використаних джерел:**

1. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя : ЗНТУ, 2018. 341 с.
2. Kornilova O., & Mykhalchuk M. Application of Transfer Learning Methods for Image Classification in Digital Humanities. *Journal of Information Technologies*, 2021. Vol. 15(2). P. 45–52.
3. Goodfellow I., Pouget-Abadie J., Mirza M., et al. Generative Adversarial Networks. *Communications of the ACM*, 2020. Vol. 63(11). P. 139–144.