Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗОВАНОГО ПЕРЕКЛАДУ СТОРІНОК МАНГИ АНГЛІЙСЬКОЮ МОВОЮ З ВИКОРИСТАННЯМ OCR ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-22-2
Остапчук В. О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Яковлева О. В.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Остапчук Валерії Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення застосунку для автоматизованого перекладу сторінок манги англійською мовою за допомогою OCR та великих мовних моделей

затверджена наказом університету від 26 травня 2026 року № 435Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 20 червня 2026 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література з питань перекладу манги та розпізнавання японського тексту, матеріали конференцій по темі мультимодальних мовних моделей та їх використання у сфері перекладу, дані інтернет-мережі, документація фреймворків Django, React Native та бібліотеки Google Gemini API, OpenAI API, Gemini 2.5 Flash, Gemini 3 Pro, ChatGPT 4.0, спеціалізовані інструменти оптичного розпізнавання символів з відкритим кодом MangaOCR, бібліотека комп'ютерного зору з відкритим кодом Comic Text Detector, Gradio.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд готових рішень та підходів для перекладу манги та аналіз їх можливостей.
2. Розробка підходу для використання у застосунку.
3. Проєктування та розробка MVP-вебзастосунку на Gradio.
4. Проєктування та розробка мобільного застосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми перекладу манги, існуючі методи та рішення для задачі перекладу манги, постановка задачі, аналіз існуючих наборів даних для проведення експерементів та розробки підходу перекладу манги, сформований набір даних для експерементів, тестові експеременти використання різних мовних моделей та підходів для здійснення найбільш схожого з оригіналом перекладу з використанням контекстних підказок зображення, перевірка подібності отриманих перекладів до еталонного варіанту, проведення опитування для виявлення найбільш привабливого для потенційного користувача перекладу, висновок щодо обрання моделі, діаграми взаємодії компонентів застосунку, ілюстрації роботи застосунку, аналіз роботи застосунку.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	15.04.26-17.04.26	
3	Аналіз літератури з проблеми, що вирішується	18.04.26-26.04.26	
4	Аналіз існуючих методів розпізнавання	27.04.26-28.04.26	
5	Формування кроків вирішення проблеми	29.04.26-01.05.26	
6	Програмна реалізація	01.05.26-20.05.26	
7	Аналіз отриманих результатів	21.05.26-04.06.26	
8	Оформлення пояснювальної записки	05.06.26-09.06.26	
9	Перевірка на нормоконтроль	10.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ _____доц. Яковлева О. В._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 71 с., 5 табл., 28 рис., 29 джерел.

АВТОМАТИЗАЦІЯ, АНАЛІЗ ВІЗУАЛЬНОГО КОНТЕКСТУ, ГЕНЕРАЦІЯ, ВЕБЗАСТОСУНОК, КОНТЕКСТНИЙ ПЕРЕКЛАД, МОБІЛЬНИЙ ЗАСТОСУНОК, ПАЙТОН, РОЗПІЗНАВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, DJANGO, GEMINI 2.5 FLASH, REACT NATIVE, SQLITE.

Об’єктом роботи є питання автоматизованого перекладу тексту з багатопанельних зображень сторінок манги.

Метою роботи є розробка застосунку для автоматизованого перекладу сторінок манги англійською мовою з використанням OCR та великих мовних моделей.

Під час роботи використано: методи штучного інтелекту та аналізу даних, методи веб та мобільної розробки.

Практична цінність застосунку полягає в автоматизації перекладу манги, покращенні локалізації та спрощенні ведення особистої бібліотеки манги.

Розроблено застосунок, що забезпечує розпізнавання тексту зі сторінки манги, генерацію перекладу, ведення особистої бібліотеки манги та перегляд статистики.

Область застосування: прискорена локалізація манги, вивчення мови, поширення культури.

AUTOMATIZATION, ARTIFICIAL INTELLIGENCE, CONTECT-AWARE TRANSLATION, DJANGO, GEMINI 2.5 FLASH, GENERATION, MOBILE APPLICATION, PYTHON, REACT NATIVE, RECOGNITION, SGLITE, VISUAL CONTEXT ANALYSIS, WEB APPLICATION.

The objective of this work is the automated translation of text from multi-panel manga page images.

The goal of this work is to develop an application for context-aware manga translation based on Gemini 2.5 Flash.

The following were used during the work: artificial intelligence and data analysis methods, web and mobile development methods.

The practical value of the application lies in the automation of manga translation, improving localization, and simplifying the management of a personal manga library.

An application has been developed that ensures text recognition from manga pages, translation generation, maintaining a personal manga library, and viewing statistics. Scope of application: accelerated manga localization, language learning, cultural exchange .

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Канджі – ієрогліфічне письмо, складова частина японської писемності

Катакана – друга базова система японської писемності, використовується для займаних слів

Мультимодальна модель – модель штучного інтелекту, здатна обробляти кілька типів даних, зокрема текст і зображення

Промпт – текстова інструкція для моделі, яка визначає правила аналізу зображення та формат відповіді

Токен – мінімальна одиниця тексту, яку велика мовна модель використовує для аналізу та генерації

Хірагана – базова система японської писемності, використовується в першу чергу для запису звуків

Фурігана – фонетичні підказки в японській мові

ШІ – штучний інтелект

API – Application Programming Interface (інтерфейс програмного застосунку, використовується для обміну даними між клієнтом і сервером)

CORS – Cross-Origin Resource Sharing (механізм безпеки сучасних браузерів, який контролює доступ до веб-ресурсів на іншому домені)

Gradio – фреймворк для швидкого створення вебінтерфейсів для моделей машинного навчання

GPT – Generative Pre-trained Transformer (сімейство великих мовних моделей)

OCR – Optical Character Recognition (оптичне розпізнавання символів)

OpenCV – бібліотека функцій та алгоритмів комп’ютерного зору, обробки зображень і чисельних алгоритмів загального призначення

PIL – Pillow (бібліотека мови Python, яка використовується для роботи з растровою графікою)

Python – високорівнева мова програмування загального призначення

Microsoft Word Object Library – спеціальна бібліотека компонентів, яка дозволяє зовнішнім програмам та скриптам керувати Microsoft Word

MLLM – Multimodal Large Language Model (мультимодальна велика мовна модель)

MLLMs – Multimodal Large Language Models (мультимодальні великі мовні моделі)

MVP – Minimum Viable Product (мінімально життєздатний продукт)

NMT – Neural Machine Translation (нейронний машинний переклад)

URL – Uniform Resource Locator (уніфікована адреса ресурсу в мережі Інтернет)

ВСТУП

Комп’ютерна графіка розділяється за трьома основними напрямками: візуалізація, обробка зображень і розпізнавання образів.

Візуалізація – це створення зображення на основі якогось опису (моделі). Приміром, це може бути відображення графіку, схеми, імітація тривимірної віртуальної реальності в комп’ютерних іграх, у системах архітектурного проєктування.

Основне завдання розпізнавання образів – це одержання семантичного опису зображених об’єктів [1]. Мета розпізнавання може бути різна, як виділення окремих елементів на зображенні, так і класифікація зображення в цілому. У якомусь сенсі завдання розпізнавання є зворотним стосовно завдання візуалізації. Областями застосування є системи розпізнавання текстів [2, 3], створення тривимірних моделей людини по фотографіях, підрахунок людей у натовпі та закритому просторі [4, 5], розпізнавання обличь [6, 7], аналіз текстур [8] та інше.

Актуальність роботи полягає у розв’язанні практичної проблеми доступності манги для широкої аудиторії, що стрімко зростає в умовах глобальної популяризації японської поп-культури. Попри значний попит, переважна більшість манги залишається недоступною для читачів, які не володіють японською мовою. Офіційні локалізації виходять із суттєвою затримкою або взагалі не публікуються для певних ринків, а наявні інструменти автоматичного перекладу не враховують специфіки манґи як медіаформату.

Задача автоматизованого перекладу манги знаходиться на перетині кількох активно розвинутих напрямів сучасних досліджень, а саме комп’ютерного зору, оптичного розпізнавання символів та великих мовних моделей. Поява мультимодальних LLM, здатних одночасно аналізувати зображення та генерувати текст, відкрила принципово нові можливості для вирішення цієї задачі, однак порівняльний аналіз однокрокових та

двокрокових підходів на основі MLLM у контексті перекладу манги залишається малодослідженим.

Робота присвячена розробці застосунку для автоматизованого перекладу сторінок манги англійською мовою з використаням OCR та великих мовних моделей.

Результати роботи можуть бути використані для спрощення доступу до манги, що не має офіційної локалізації, прискорення процесу фанатського та професійного перекладу, а також як основа для подальшого розширення функціональності, зокрема підтримки пакетної обробки томів, інтеграції з платформами для читання манги або адаптації під інші мови цільового перекладу.

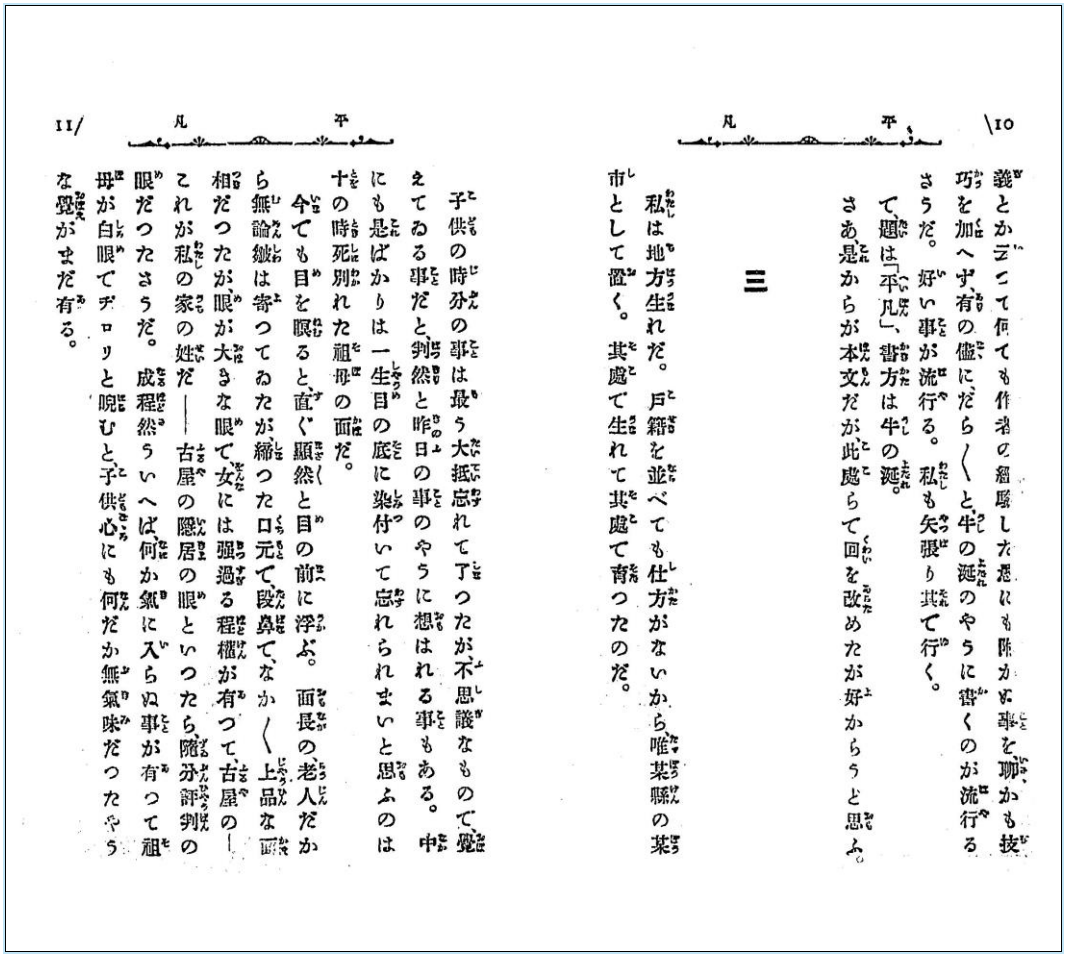
1 ОГЛЯД ПОТОЧНОГО СТАНУ ПИТАННЯ ПЕРЕКЛАДУ МАНГИ

1.1 Манга як окремий вид літератури

Сучасний глобальний культурний простір характеризується стрімкою експансією східноазійських медіапродуктів, серед яких провідне місце посідає манга, тобто японські комікси, що сформували унікальну візуально-літературну мову. На відміну від західних графічних новел, манга є не просто розважальним контентом, а складним пластом японської культури, що охоплює широку жанрову різноманітність та вікові категорії. Зростання попиту на цей вид літератури за межами Японії зумовлює гостру потребу в оперативній та якісній локалізації [9].

Специфіка манги як об’єкта перекладу та локалізації зумовлена глибоким синтезом зображення та тексту. Візуальна складова часто несе не менше семантичного навантаження, ніж пряма мова персонажів. Це створює унікальні виклики для розробників систем автоматизованого перекладу, оскільки розуміння контексту вимагає аналізу не лише текстових символів, а й графічних елементів сторінки, таких як виразів облич, поз персонажів, розташування текстових хмар (баблів) та навіть загальної атмосфери сцени, тож манга постає як мультимодальний текст, де лінгвістична та візуальна інформація є нерозривними частинами єдиного цілого.

Технічна складність роботи з оригінальними японськими виданнями посилюється особливостями самої японської мови та традиціями оформлення видань. Японська писемність використовує комбінацію трьох систем: канджі, хірагана та катакана, що значно ускладнює процеси комп’ютерного розпізнавання тексту (рис. 1.1). Крім того, у манзі часто використовується фуріґана, маленькі фонетичні підказки поруч із канджі, які є критично важливими для правильного розуміння змісту, але створюють додатковий «шум» для стандартних систем розпізнавання символів.



義とか云つて何でも作者の都合した悪れも隠かな事を聊かも技巧を加へず有の儘にだら〳〵と牛の涎のやうに書くのが流行るさうだ。好い事が流行る。私も矢張り其て行く。

て、題は「平凡」、書方は牛の涎。

さあ是からが本文だが此処らて回を改めたが好からうと思ふ。

三

私は地方生れだ。戸籍を並べても仕方がないから唯某縣の某市として置く。其處で生れて其處て育つたのだ。

子供の時分の事は最う大抵忘れて了つたが不思議なもので覚えてゐる事だと判然と昨日の事のやうに想はれる事もある。中にも是ばかりは一生目の底に染付いて忘れられまいと思ふのは十の時死別れた祖母の面だ。

今でも目を瞑ると直ぐ顯然と目の前に浮ぶ。面長の老人だから無論皺は寄つてゐたが締つた口元で段鼻で、なか〳〵上品な面相だつたが眼が大きな眼で女には強過る程權が有つて古屋の——これが私の家の姓だ——古屋の隠居の眼といつたら、隨分評判の眼だつたさうだ。成程然ういへば、何か氣に入らぬ事が有つて祖母が白眼てゐるりと睨むと子供心にも何だか無氣味だつたやうな覚がまだ有る。

Рисунок 1.1 – Приклад використання японських систем письма

Ще однією характерною рисою манги є специфічний напрямок читання справа наліво та зверху вниз, що радикально відрізняється від європейської системи та потребує адаптації логіки розпізнавання текстових блоків. Візуальне оформлення часто містить складні фони, на які накладається текст, використання різноманітних художніх шрифтів та рукописних написів, що інтегровані безпосередньо в малюнок (рис. 1.2). Таке розмаїття графічних форм робить мангу одним із найскладніших типів документів для автоматизованої обробки, де класичні методи сегментації та розпізнавання часто виявляються недостатньо ефективними, що й обумовлює актуальність пошуку нових підходів на основі сучасних інтелектуальних технологій.

Рисунок 1.2 – Приклад сторінки манги

Саме тому манга є показовим полігоном для випробування можливостей мультимодальних моделей штучного інтелекту.

1.2 Огляд готових рішень

Аналіз поточного ринку програмного забезпечення дозволяє виділити кілька ключових сервісів, кожен з яких використовує власні архітектурні підходи та має специфічні обмеження, що впливають на якість фінального продукту [10]. Огляд існуючих рішень наведено у таблиці 1.1.

Одним із найбільш відомих інструментів є проєкт Mokuro [11]. Це рішення базується на створенні інтерактивного середовища для читання, де текст розпізнається заздалегідь і стає доступним для перегляду у вебінтерфейсі. Основною перевагою Mokuro є його здатність ефективно працювати з великими бібліотеками та надавати користувачеві можливість миттєвого доступу до словникових значень слів.

Таблиця 1.1 – Огляд існуючих рішень

Назва	Функціональність	Обмеження	Технології	Потребує налаштувань	Тип
Google Translate	Переклад	Так	NMT	Ні	Сайт, мобільний застосунок
Mokuro	Переклад	Так	OCR, Yomitan	Так	Сайт
Cotrans	Переклад	Так	DeepL, GPT, Papago	Ні	Сайт
Balloons Translator	Робота з зображеннями	Так	Ні	Так	Десктоп-застусунок

Проте, з технічної точки зору, Mokuro не є повноцінним перекладачем у класичному розумінні, оскільки він фокусується переважно на етапі оптичного розпізнавання символів та наданні контекстних підказок, залишаючи процес інтерпретації змісту безпосередньо читачеві (рис. 1.3). Крім того, складність процесу попереднього налаштування та необхідність розгортання локального середовища обмежують коло потенційних користувачів лише технічно грамотною аудиторією.

Рисунок 1.3 – Прилад роботи застосунку Mokuro

Альтернативним підходом є використання хмарних сервісів, наприклад Cotrans [12]. Це комплексне вебрішення, що реалізує повний цикл обробки сторінки від детекції текстових баблів та видалення оригінального тексту до накладання перекладу з урахуванням вихідного форматування. Cotrans надає користувачеві вибір між різними моделями машинного перекладу, що дозволяє варіювати якість результату (рис. 1.4). Однак, централізована природа сервісу створює значне навантаження на серверну інфраструктуру, що призводить до виникнення тривалих черг на обробку запитів. Це робить інструмент незручним для оперативного читання великих обсягів контенту в режимі реального часу.

Рисунок 1.4 – Інтерфейс застосунку Cotrans

Окрему категорію становлять десктопні застосунки, такі як BalloonsTranslator. Ці програми пропонують більшу автономність та гнучкість у налаштуваннях, дозволяючи користувачеві самостійно інтегрувати різні

інтерфейси програмного застосунку (Application Programming Interface, API) перекладачів. Проте головною проблемою таких рішень залишається відсутність цілісності процесу. Часто користувач змушений вручну коригувати зони розпізнавання або стикатися з помилками сегментації тексту на складних художніх фонах. Більшість існуючих десктопних рішень не мають достатнього рівня інтелектуального аналізу контексту, що призводить до втрати змістової зв'язності між окремими репліками персонажів на сторінці [9].

Загальний аналіз готових рішень свідчить про те, що незважаючи на значний прогрес у сфері комп'ютерного зору та машинного перекладу, на ринку досі відсутній універсальний застосунок, який би поєднував у собі високу швидкість обробки, точність контекстуального перекладу та інтуїтивно зрозумілий користувацький інтерфейс. Існуючі інструменти або занадто складні в експлуатації, або мають значні часові затримки, що підкреслює необхідність розробки нового підходу, який би базувався на інтеграції сучасних MLLMs для оптимізації всього процесу локалізації.

1.3 Аналіз існуючих підходів до вирішення задачі

Процес автоматизованої локалізації манги є комплексною інженерною задачею, що вимагає інтеграції методів комп'ютерного зору та обробки природної мови. На сучасному етапі розвитку технологій задачу перекладу графічного контенту можна вирішити за допомогою трьох основних технологічних підходів, таких як використання класичних систем NMT, такі як Google Translate, застосування каскадних систем на основі OCR та впровадження MLLM. Кожен із цих підходів має власну специфіку обробки візуальної та текстової інформації, що безпосередньо впливає на точність передачі змісту. Наприклад, для обробки за допомогою OCR необхідна передня нормалізація, яку можна реалізувати на основі дескрипторів [13-18].

На рисунку 1.5 показано порівняння різних підходів перекладу на прикладі однієї зі сторінок манги.

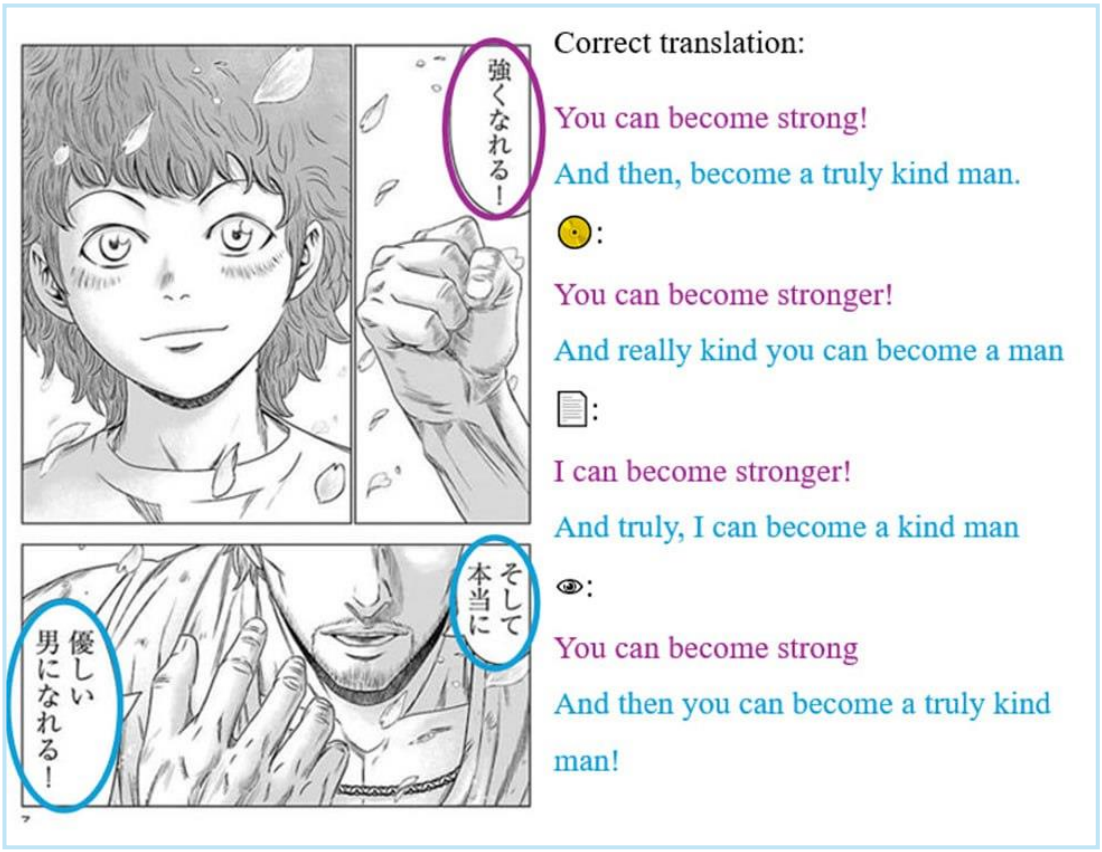


Рисунок 1.5 – Попередній аналіз підходів:
◎ – NMT; 📄 – OCR (text context); 👁 – LLM (visual context)

Аналіз візуального ряду та попередніх сцен дозволяє однозначно стверджувати, що репліки адресовані хлопчику, а не є монологом героя про самого себе. Проте системи, що не мають доступу до візуального контексту, часто помилково інтерпретують суб’єкта дії. Наприклад, підхід на основі OCR та NMT схильний генерувати фрази від першої особи, що повністю викривлює зміст поради. Це демонструє, що без розуміння того, хто саме зображений у кадрі та до кого він звертається, досягти адекватної локалізації неможливо.

Порівняльний аналіз та тестування базових конфігурацій систем NMT продемонстрували їх обмежену ефективність у роботі з мангою. Незважаючи на високу швидкість обробки «чистого» тексту, такі системи виявилися

найбільш вразливими до специфіки жанру. Основною проблемою стала повна відсутність врахування візуального контексту, що призводило до втрати змістової цілісності при перекладі гендерно-маркованої лексики, ономатопеї та діалогів, де суб'єкт дії визначається лише візуально. Через нездатність інтерпретувати графічні дані, традиційний машинний переклад показав найнижчі результати за критеріями адекватності та зв'язності, що обумовило необхідність переходу до більш складних, контекстуально-залежних архітектур.

Поява мультимодальних великих мовних моделей як класу систем штучного інтелекту ознаменувала перехід від вузькоспеціалізованих інструментів до універсальних моделей, здатних одночасно сприймати та інтерпретувати різні типи даних, такі як текст, зображення, а в окремих випадках і аудіо. На відміну від попередніх підходів, де задачі розпізнавання та розуміння вирішувалися окремими моделями з подальшим об'єднанням результатів, MLLM обробляють усі модальності в єдиному контексті, що дозволяє враховувати взаємозв'язок між візуальним і мовним змістом при формуванні відповіді. Саме ця властивість робить їх перспективним інструментом для задач, де текст невіддільний від зображення

Багато сервісів та застосунків базуються на розпізнаванні на основі MLLM, наприклад, мобільний застосунок Billka AI використовує MLLM для екстракції інформації з паперового чеку [19, 20].

Виходячи з цього, вибір оптимального підходу потребує балансування між точністю технічного розпізнавання символів та глибиною семантичної інтерпретації. Сучасні мультимодальні моделі демонструють здатність враховувати обидва аспекти одночасно, що робить їх природним вибором для вирішення цієї задачі. Подальший розвиток систем перекладу манги вбачається у синергії візуального сприйняття та глибокого лінгвістичного аналізу, що стає можливим завдяки архітектурі сучасних інтелектуальних моделей.

1.4 Бібліотеки та фреймворки

Технічна реалізація систем перекладу манги базується на використанні широкого стеку технологій, що охоплюють мови програмування високого рівня, спеціалізовані бібліотеки для комп'ютерного зору та інструменти взаємодії з хмарними API.

Python є універсальною мовою програмування, що вирізняється лаконічністю синтаксису, бо реалізація тієї самої логіки потребує значно меншого обсягу коду порівняно з більшістю альтернативних мов.

Важливою характеристикою Python є крос-платформеність. Написані програми виконуються на різних операційних системах без необхідності вносити будь-які зміни до вихідного коду. Окрему перевагу становить багата стандартна бібліотека, що з коробки надає інструменти для взаємодії з операційною системою, роботи з мережею та вебресурсами, підключення до баз даних, обробки різноманітних форматів даних, а також побудови графічного інтерфейсу користувача [21]. Для базової роботи із графічними файлами, їх нормалізації та попередньої обробки зазвичай використовуються такі бібліотеки, як OpenCV та Pillow. Вони дозволяють виконувати операції зміни розміру, корекції контрастності та фільтрації шумів, що є критично важливим для покращення якості подальшого розпізнавання тексту на складних фонах.

У межах двоетапного підходу особлива увага приділяється спеціалізованим OCR-фреймворкам, таким як MangaOCR, який оптимізований саме для детекції та розпізнавання японських ієрогліфів, фуріґани та вертикального тексту, специфічного для коміксів.

Важливим аспектом розробки є вибір архітектурного паттерну для створення інтерфейсу користувача. Для десктопних рішень часто застосовується Microsoft Word Object Library або мобільні фреймворки, проте для науково-дослідних MVP-проєктів найбільш доцільним є використання інструментів швидкого прототипування. Це дозволяє зосередитися на логіці

роботи алгоритмів перекладу, не витрачаючи надмірних ресурсів на розробку складного фронтенд-коду. Одним із найбільш прогресивних рішень у цьому контексті є бібліотека Gradio, яка забезпечує створення легких та функціональних веб-інтерфейсів безпосередньо мовою Python.

1.5 Переваги Gradio

Gradio – це спеціалізована Python-бібліотека з відкритим вихідним кодом, призначена для створення інтуїтивно зрозумілих інтерактивних інтерфейсів для моделей машинного навчання. Даний інструмент дозволяє стисло та ефективно проєктувати веб-орієнтовані графічні оболонки, що забезпечують пряму взаємодію користувача з інтелектуальними алгоритмами шляхом подання вхідних даних та отримання прогнозів або відповідей у режимі реального часу (рис. 1.6). Висока практична цінність Gradio обумовлена його універсальністю. Фреймворк є незамінним на етапах розробки, тестування та розгортання моделей, а також для створення функціональних прототипів та невеликих прикладних застосунків. [22]

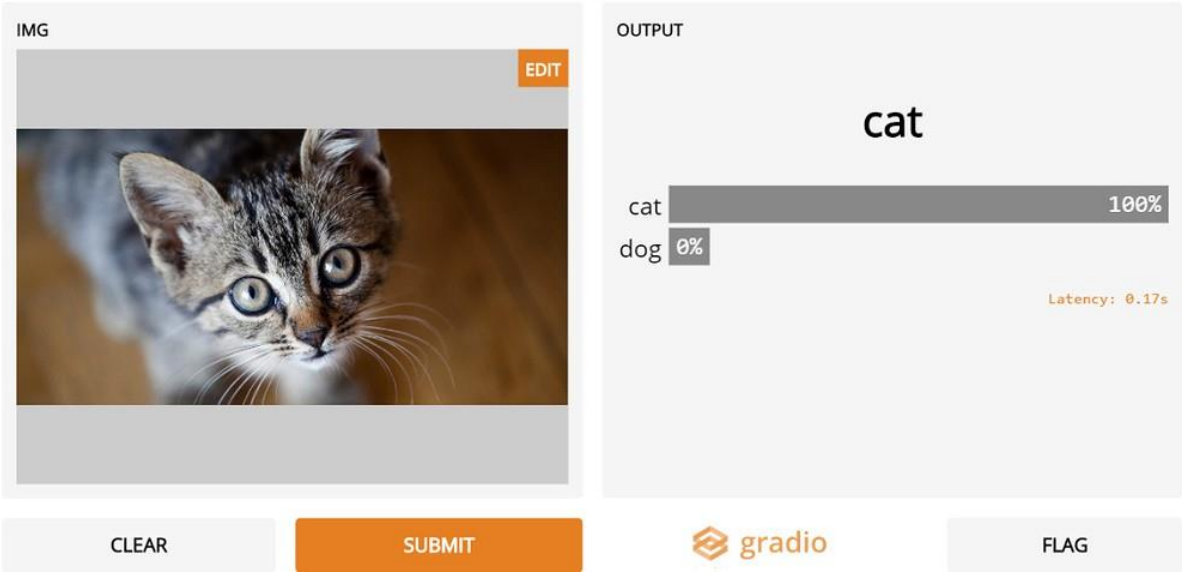


Рисунок 1.6 – Приклад застосунку на базі бібліотеки Gradio

Gradio пропонує набір інтерактивних компонентів, спеціально адаптованих для завдань штучного інтелекту, що дозволяє формувати ефективні інтерфейси без необхідності низькорівневої розробки кожного елемента. Ключові функціональні можливості бібліотеки:

– різноманітність вхідних форматів. Бібліотека підтримує роботу з текстом, зображеннями, аудіо, відео та файловими структурами. Це робить її оптимальним вибором для широкого спектра задач;

– робота з файловими вкладеннями. Фреймворк забезпечує стабільну роботу компонентів для завантаження документів та обробки файлів, що є важливим для систем контент-аналізу;

– гнучкість відображення результатів. Вихідні дані можуть бути представлені у вигляді тексту, статичних та динамічних зображень або інтерактивних діаграм. Це дозволяє кінцевому користувачеві легко інтерпретувати результати роботи моделі.

У контексті розробки систем для автоматизації перекладу манги Gradio є безальтернативним інструментом завдяки своїй простоті у використанні, гнучкості налаштувань та підтримці мультикористувацького доступу. Це дозволяє створювати доступні та функціональні застосунки, мінімізуючи витрати на розробку фронтенд-складової та зосереджуючись на якості роботи інтегрованих MLLMs.

1.6 Постановка задачі

Аналіз ринку перекладу манги показав настачу рішенньь, які б закрили усі потреби користувача. Автоматизований переклад манги є актуальним у контексті розвитку мультимодальних систем ШІ та ростом популярності манги у медіа-просторі. Тому ставиться завдання створити застосунок, що дозволить розпізнавати текст зі сторінки з урахуванням контексту та особливостей жанру із використанням сучасних генеративних моделей.

Об’єктом роботи є питання автоматизованого перекладу тексту з багатопанельних зображень сторінок манги.

Метою роботи є розробка застосунку для автоматизованого перекладу сторінок манги англійською мовою з використанням OCR та великих мовних моделей.

Для досягнення мети необхідно вирішити такі завдання:

– провести аналіз підходів класичного OCR та сучасних мультимодальних моделей для детектування та розпізнавання тексту на складних фонах;

– проаналізувати можливості сучасних MLLMs щодо здатності одночасної візуальної інтерпретації та контекстуального перекладу японського тексту;

– сформувати власний набір даних, що містить сторінки манги різних жанрів з автентичними шрифтами, фуріганою та складним візуальним оформленням;

– виконати порівняльний аналіз двоетапного підходу на основі OCR та MLLM і одноетапного на основі MLLM з використанням метрики відстані Левенштейна для оцінки якості розпізнавання та методу експертних оцінок для аналізу якості перекладу; обрати підхід, що найкраще вирішує задачу перекладу манги;

– провести експериментальний аналіз швидкодії обробки сторінок та економічної складової використання API різних моделей;

– розробити алгоритм перекладу манги на основі обранного підходу;

– спроєктувати та реалізувати MVP вебзастосунку на базі Gradio, який автоматизує процес перекладу завантаженої сторінки манги за обраною методикою;

– спроєктувати та реалізувати мобільний застосунку за допомогою React Native, який автоматизує процес перекладу завантаженої сторінки, підтримує збереження перекладу та повноцінне ведення особистої бібліотеки користувача.

2 РОЗРОБКА ПІДХОДУ ДЛЯ ПЕРЕКЛАДУ МАНГИ

2.1 План аналізу та критерії оцінювання

Розробка ефективного підходу до автоматизованого перекладу манги є комплексним завданням, що потребує систематичного планування та чіткого визначення критеріїв оцінювання результатів.

В роботі запропоновано дослідити два підходи, що ґрунтуються на методах, які, за результатами попереднього аналізу, мають найбільший потенціал (рис. 2.1). Перший етап аналізу було зосереджено на декомпозиції процесу перекладу. В межах двоетапного підходу окремо аналізується ефективність розпізнавання тексту за допомогою OCR порівняно з MLLM. Другий етап передбачає перевірку наявності переваги одноетапного підходу на основі MLLM над двоетапним.

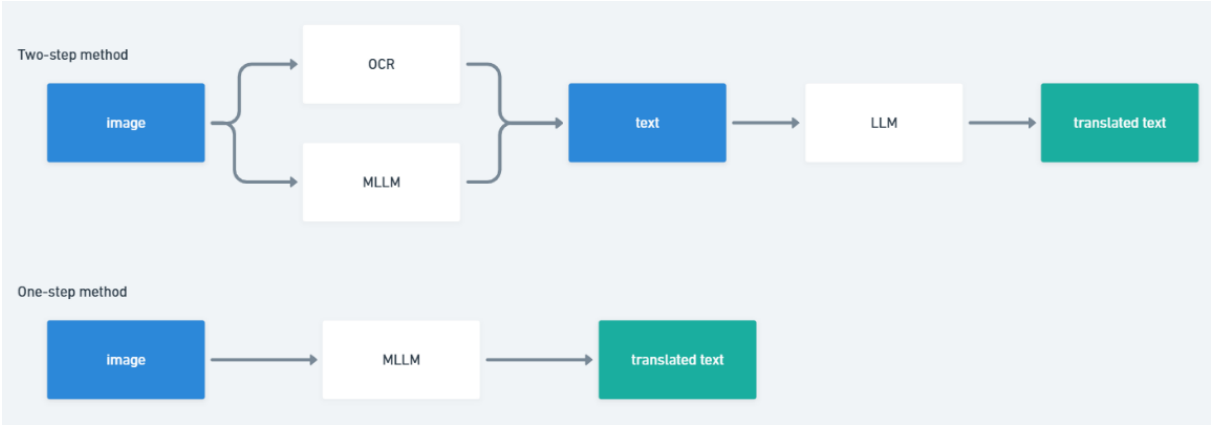


Рисунок 2.1 – Схема роботи підходів

Критерії оцінювання розробленої системи та обраних підходів поділяються на кількісні та якісні. До кількісних показників відносяться точність розпізнавання для двоетапного підходу, швидкість обробки та вартість.

Критерій якості є пріоритетним, оскільки кінцевою метою є забезпечення читача зрозумілим і точним перекладом. Якість розглядається у

таких двох вимірах, як якість розпізнавання тексту для двоетапного підходу та якість кінцевого перекладу для обох підходів.

Двоетапний або ж каскадний підхід, базується на послідовному виконанні операцій детекції, розпізнавання та перекладу. На першому етапі застосовуються алгоритми комп'ютерного зору, завданням яких є виділення текстових областей на складному фоні та перетворення графічного начерку ієрогліфів у машиночитаний текст. Другий етап передбачає передачу отриманого тексту до мовної моделі для виконання перекладу. В межах цього підходу виділено два варіанти розпізнавання, таких як класичне використання OCR-бібліотек, що фокусуються на геометричних параметрах символів, та використання інтелектуальних промпт-інженерних рішень, де мовна модель допомагає уточнити розпізнані символи. Головною проблемою залишається накопичення помилок, тому що будь-яка неточність на етапі OCR критично викривлює фінальний переклад.

Альтернативою є одноетапний підхід на основі MLLM. Ця методика передбачає пряму обробку зображення сторінки без проміжного виділення тексту в окремий файл. Модель одночасно «бачить» і графічну композицію, і текстовий зміст, що дозволяє їй аналізувати взаємозв'язок між репліками та діями персонажів. Такий підхід мінімізує втрату даних при сегментації та дозволяє зберігати правильну тональність перекладу завдяки розумінню візуального оточення. Використання цілісного аналізу сторінки дозволяє системі розрізняти основний текст та художньо оформлені звукові ефекти, що значно підвищує рівень занурення читача в адаптований контент.

Для оцінювання якості розпізнавання використовується відстань Левенштейна, метрика, що обчислює мінімальну кількість редакційних операцій, необхідних для перетворення одного рядка на інший. Ця метрика є стандартною у задачах оцінювання точності OCR та дозволяє кількісно порівняти результат розпізнавання з еталонним текстом. Нормалізоване значення подібності визначається за формулою:

$$sim(a,b) = 1 - \frac{lev(a,b)}{max\,(|a|,|b|)}, \quad (2.1)$$

де *lev(a, b)* – відстань Левенштейна між рядками *a* та *b*;

$|a|$ та $|b|$ – довжини рядків *a* та *b* відповідно.

Результат знаходиться у діапазоні від 0 до 1, де значення, близьке до 1, свідчить про високу схожість рядків. Якість кінцевого перекладу, натомість, не може бути повністю формалізована за допомогою автоматичних метрик, оскільки переклад є суб'єктивним завданням, що вимагає оцінки природності мови, точності передачі сенсу та відповідності стилю оригіналу. З огляду на це, для оцінки якості перекладу застосовується метод експертних оцінок.

Критерій швидкості обробки є важливим з точки зору практичного застосування у мобільному застосунку. Очікування перекладу однієї сторінки протягом тривалого часу суттєво знижує зручність використання продукту. Час обробки вимірюється у секундах для кожної моделі та кожного підходу і розраховується як середнє значення по всіх сторінках набору даних. Для двоетапного підходу враховується сумарний час виконання обох кроків.

Критерій вартості відображає економічну доцільність використання тієї чи іншої моделі у промисловому продукті. Сучасні MLLMs надаються, як правило, за моделлю оплати за використання API (pay-per-use), де вартість визначається кількістю оброблених токенів, тобто одиниць тексту чи зображення. Різні моделі мають суттєво відмінні тарифи, тому навіть невелика різниця у вартості одного запиту може призвести до значних витрат при масштабуванні на велику кількість користувачів. Вартість аналізується у перерахунку на одну сторінку манги, що дозволяє зіставляти моделі між собою в однакових умовах.

У висновку можна сказати, що план аналізу охоплює послідовні аналітичні та експериментальні етапи, кожен з яких описується у відповідному підрозділі. Спочатку формується набір даних, далі тестуються підходи до розпізнавання та перекладу, потім проводиться експертна оцінка, і на основі

зведеного аналізу за трьома критеріями, обирається найкращий з точки зору обраних показників підхід.

2.2 Огляд існуючих та формування власного набору даних

Наявність репрезентативного та якісно розміченого набору даних є необхідною умовою для проведення об’єктивного порівняльного аналізу методів перекладу манги. Набір даних визначає як можливості дослідження, так і його обмеження, оскільки від різноманітності та складності включених сторінок залежить, наскільки узагальненими будуть отримані результати.

Перед формуванням власного набору даних було проведено аналіз існуючого загальнодоступного манга-орієнтованого набру даних Manga109 з метою визначення можливості його використання для цілей даного аналізу. Manga109 є еталонним набором даних, що містить 109 томів манги загальним обсягом понад 21 000 сторінок, розробленим дослідниками Токійського університету та широко використовуваним у задачах комп’ютерного зору. Набір даних включає детальні анотації для таких задач, як детектування облич, фігур, рамок панелей, а також текстових блоків (рис. 2.2).

Однак при детальному аналізі структури та призначення Manga109 було виявлено суттєве обмеження. Набір даних зосереджується виключно на задачах аналізу зображення та розпізнавання тексту, але не охоплює питання перекладу. Анотації до текстових блоків містять координати розташування та транскрипцію оригінального японського тексту, однак жодного відповідного перекладу цільовою мовою не передбачено. Це унеможливлює використання набору даних для оцінки якості кінцевого перекладу, одного з ключових критеріїв аналізу.

У зв’язку з цим для проведення аналізу було прийнято рішення сформувати власний набір даних, що складається з пар зображень, а саме оригінальної сторінки манги японською мовою та відповідної сторінки з

офіційно опублікованого англійського перекладу. Такий підхід дозволяє використовувати офіційний переклад як еталон при оцінці якості автоматично отриманого перекладу.

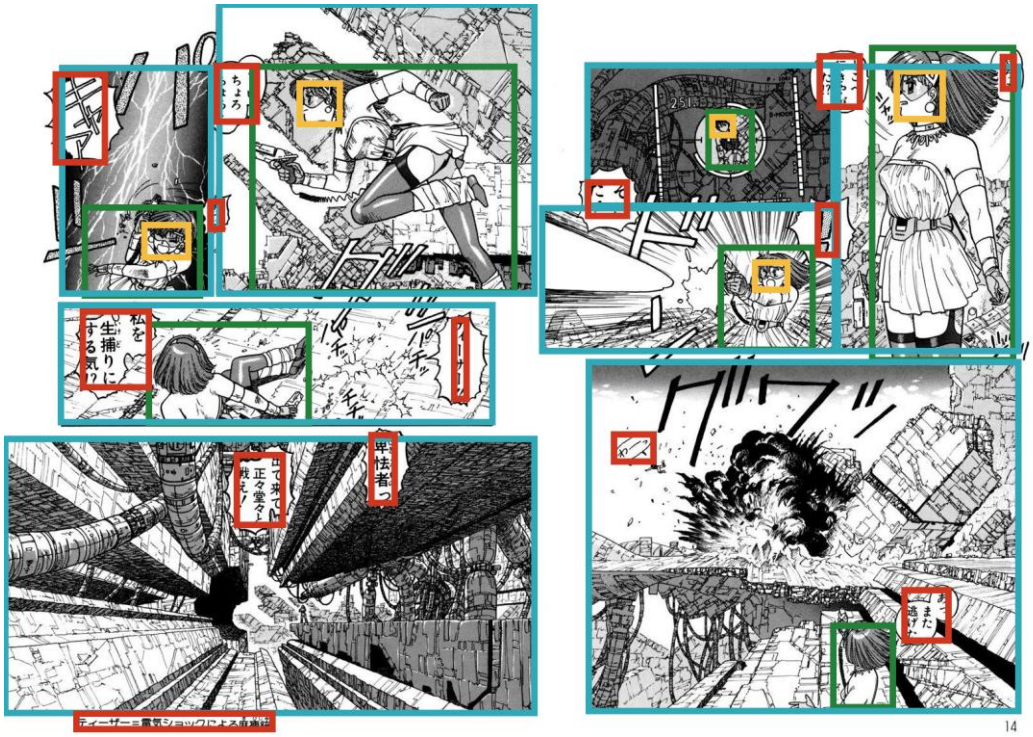


Рисунок 2.2 – Приклад сторінки з набору даних Manga109 з розмітками текстових блоків та панелей

При формуванні набору даних до включених матеріалів ставилися такі вимоги, як наявність офіційного перекладу від ліцензованого видавця для можливості порівняльного оцінювання, представленість різних типів текстового наповнення, діалогів, монологів, описових вставок, різноманітність жанрів манги для підвищення репрезентативності, різний рівень складності зображень, від простих чистих фонів до насичених деталізованих сцен з перекриттям тексту та фону, різноманітність шрифтів, наприклад наявність рукописних та стилізованих, а також сторінок із фуріґаною.

Набір даних включає 20 пар зображень, відібраних з різних серій манги. Обсяг у 20 сторінок є достатнім для проведення порівняльного аналізу на

початковому етапі експерементів, враховуючи, що кожна сторінка манги є структурно складним об’єктом, що містить у середньому від п’яти до п’ятнадцяти текстових блоків різного типу. При цьому важливою перевагою такого підходу є можливість проведення як автоматичного оцінювання якості розпізнавання на базі метрики Левенштейна, так і суб’єктивного оцінювання якості перекладу на основі залучення експертів. Приклад однієї з пар сформованого набору даних продемонстровано на рисунку 2.3.

Рисунок 2.3 – Приклад пари зображень зі сформованого набору даних

Вибір на користь англійської мови як цільової для перекладу обумовлений кількома чинниками. По-перше, переважна більшість офіційних перекладів манги, що здійснюються ліцензованими видавцями, такими як Viz Media, Kodansha USA, Yen Press, виконуються саме англійською, що забезпечує широку доступність еталонних матеріалів. По-друге, сучасні MLLMs демонструють найвищий рівень компетентності саме в англійській мові як найбільш представленій у навчальних даних. Водночас описаний підхід є мовно-незалежним, бо після підтвердження його ефективності для пари «японська англійська» він може бути адаптований для будь-якої іншої

цільової мови, включаючи українську, без необхідності перенавчання або окремого налаштування моделей.

Отже, сформований набір даних є цілеспрямованим інструментом експериментального аналізу, що поєднує вимоги до різноманітності матеріалу, наявності еталону для оцінювання та відповідності реальним умовам роботи майбутнього застосунку

2.3 Двоетапний підхід на основі OCR та LLM

Двоетапний підхід є найбільш прямолінійним рішенням задачі машинного перекладу і відтворює традиційний процес обробки документів. Спочатку текст екстрагується з зображення, а потім обробляється мовною моделлю. При роботі двоетапного підходу кожен компонент може бути замінений або вдосконалений незалежно від іншого. Водночас двоетапний підхід несе ризик накопичення помилок, тому що неточності на кроці розпізнавання безпосередньо впливають на якість перекладу.

2.3.1 Детектування та розпізнавання тексту манги за допомогою OCR

Перший крок двоетапного підходу передбачає автоматичне виявлення та розпізнавання японського тексту безпосередньо на зображенні сторінки манги. Ця задача суттєво відрізняється від стандартного OCR для друкованих документів через ряд специфічних характеристик манги як медіа формату.

По-перше, текст у манзі розміщується у хмаринках різних форм, накладається на деталізовані малюнки, може бути написаний вертикально або горизонтально, використовувати рукописні шрифти та включати фуріґану, тобто дрібні фонетичні підказки, розміщені поруч із канджі. По-друге, контраст між текстом та фоном може бути мінімальним на складних

ілюстрованих ділянках, а межі текстового блоку розмитими. По-третє, специфічна для манги японська мова включає розмовні скорочення, архаїзми, жаргон та звуконаслідування, що не входять до стандартних словників OCR-систем.

Методологія тестування OCR-підходу полягала у наступному. Для 20 сторінок набору даних за допомогою Comic Text Detector виконувалося виявлення та кропування текстових блоків, після чого MangaOCR застосовувалася до кожного блоку окремо для отримання текстового рядка японською мовою. Приклади робти MangaOCR на окремих блоках тексту продемонстровані на рисунку 2.4. Отримані рядки зіставлялися з відповідними фрагментами еталонного оригінального тексту, а якість розпізнавання оцінювалася за допомогою нормалізованої відстані Левенштейна.

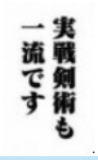
image	Manga OCR result
	素直にあやまるしか
	立川で見た〝穴〟の下の巨大な眼は：
	実戦剣術も一流です

Рисунок 2.4 – Приклад роботи MangaOCR: вхідне зображення тексту манги та результат розпізнавання тексту

Результати тестування показали, що середня нормалізована схожість розпізнаних текстів із оригіналом становила 0,7266. Це свідчить про те, що у середньому приблизно 27% символів у розпізнаному тексті потребують виправлення відносно еталонного рядка. Аналіз окремих помилок дозволив виявити декілька систематичних проблем. Першою з них є плутанина між візуально подібними канджі та хіраґаною у рукописних шрифтах, де модель допускає помилки в розпізнаванні ієрогліфів зі схожими графічними елементами. Другою проблемою є некоректна обробка вертикального тексту. Попри те, що MangaOCR декларує підтримку вертикального написання, у ряді складних випадків порядок читання збивається, що призводить до переставлених або пропущених рядків. Третьою є неповне виявлення блоків. Comic Text Detector не завжди коректно визначає межі всіх текстових областей, особливо на сторінках із насиченим малюнком або нестандартним розташуванням хмаринок. Також фуріґана викликає проблеми розпізнавання: дрібні фонетичні підказки поруч із канджі можуть сприйматися як частина основного тексту або, навпаки, ігноруватися.

Додатково можна відзначити практичні обмеження OCR-підходу. Підхід Comic Text Detector використаний разом з MangaOCR є двокомпонентним і потребує послідовного запуску двох різних моделей, що збільшує загальний час обробки та ускладнює інтеграцію. Крім того, якість OCR суттєво залежить від роздільної здатності та якості вхідного зображення. Стиснені або низькоякісні скани дають значно гірші результати розпізнавання.

Отримані результати дозволили сформулювати попередній висновок щодо доцільності використання класичного OCR-підходу як першого кроку у конвеєрі перекладу манги. При середній точності розпізнавання 72,66% кожен четвертий символ потенційно є помилковим, а помилкові символи у японській мові можуть кардинально змінювати сенс реплік, що є критичним для перекладу. Тому було вирішено перевірити, чи здатна MLLM виконати аналогічне завдання розпізнавання тексту якісніше, ніж спеціалізована OCR-система.

2.3.2 Детектування та розпізнавання на основі LLM

У роботі було проведено аналіз альтернативного варіанту першого кроку, а саме використання мультимодальної великої мовної моделі у режимі розпізнавання тексту. Відмінність полягає у тому, що MLLM сприймає зображення цілком, а не обробляє окремі кроп-блоки, завдяки чому вона може використовувати загальний візуальний контекст для підвищення точності розпізнавання. Наприклад, зображуваний персонаж та його поза можуть допомогти моделі визначити, яке саме слово ужито у неоднозначному випадку.

Для аналізу цього підходу був розроблений спеціалізований промпт, з чітко визначеними завданнями для моделі, такими як виконати роль спеціалізованого OCR-рушія для манги та здійснити екстракцію японського тексту сторінки. Промпт містив такі ключові директиви, як необхідність дотримуватись стандартного порядку читання манги, виводити текст кожного окремого мовного блоку з нового рядка, ігнорувати звуконаслідування та звукові ефекти, не додавати жодних коментарів або пояснень до виводу. Текст промпту наведено у лістингу 2.1.

Лістинг 2.1 Промпт розпізнавання для двоетапного підходу:

[TASK] You are a specialized Manga OCR engine. Your goal is to extract all Japanese text from the provided image with 100% accuracy.

[RULES]

1. Reading Order: Follow the standard manga reading order (Right-to-Left, Top-to-Bottom).

2. Content: Capture all dialogue, narration, and handwritten notes.

3. Exclusion: Do NOT include or describe onomatopoeia/sound effects (SFX).

4. Place each speech bubble or text block on a new line.

5. Separate distinct character interactions with a blank line.

6. No Commentary: Output only the raw Japanese text found on the page. No "Here is the text" or introductory sentences.

[OUTPUT]

Provide only the Japanese text.

Тестування проводилося на тих самих 20 сторінках набору даних, що й для OCR-підходу, що дозволило забезпечити коректне порівняння. Для кожної сторінки MLLM отримувала зображення та вказаний промпт, а результат розпізнавання порівнювався з еталонним японським текстом за допомогою нормалізованої відстані Левенштейна. Середня нормалізована схожість, отримана за допомогою MLLM, становила 0,9021, що на 17,55 відсоткових пунктів вище, ніж результат OCR-підходу.

Аналіз переваг MLLM у задачі розпізнавання дозволяє виявити декілька ключових чинників. На відміну від OCR, що обробляє кожен блок ізольовано, MLLM бачить усю сторінку цілком і може використовувати контекст суміжних панелей для підвищення точності розпізнавання. Завдяки навчанню на масиві різноманітних текстових зображень, включаючи рукописний текст, MLLM демонструє вищу стійкість до нестандартних накреслень. Оскільки MLLM не потребує окремого кроку для виявлення текстових областей, вона не успадковує помилки детектора. Модель коректно відокремлює фонетичні підказки від основного тексту канджі, що є суттєвою перевагою для забезпечення чистоти виводу.

Разом з тим MLLM-підхід також має певні обмеження у задачі розпізнавання. Зокрема, спостерігалися поодинокі випадки галюцинацій коли модель генерувала правдоподібний, але відсутній на зображенні текст. Також у ряді випадків модель включала до виводу звукові ефекти всупереч інструкції, що свідчить про неповну відповідність результату завданню.

2.3.3 Висновки щодо підходів детектування та розпізнавання

За результатами порівняльного тестування обох підходів до першого кроку двоетапного методу класичного OCR, тобто Comic Text Detector у парі з MangaOCR, та MLLM у режимі розпізнавання було отримано зведені результати (табл. 2.1).

Таблиця 2.1 – Порівняння результатів розпізнавання тексту OCR та MLLM

Показник	OCR (Comic Text Detector, MangaOCR)	MLLM (розпізнавання)
Середня нормалізована схожість (відстань Левенштейна)	0,7266	0,9021
Обробка рукописних шрифтів (експертна оцінка)	Задовільна	Добра
Обробка фуріґани	Часткова	Добра
Необхідність детектування блоків	Так (окремий крок)	Ні
Наявність галюцинації	Відсутня	Незначна

Аналіз отриманих даних дозволяє зробити однозначний висновок, що MLLM-підхід до розпізнавання тексту манги є суттєво кращим за класичний OCR-підхід як за точністю, так і за простотою реалізації. Перевага у 17,55 відсоткових пункти за метрикою нормалізованої схожості є практично значущою, бо у задачах машинного перекладу якість вхідного тексту безпосередньо впливає на якість перекладу, і зменшення кількості помилок на першому кроці є необхідною умовою для отримання адекватного результату на другому.

Враховуючи це, подальший аналіз якості перекладу на основі

результатів класичного OCR було вирішено не проводити. Цей крок обґрунтованй тим, що при середній точності OCR у 72,66% значна частина перекладених реплік неминуче містила б артефакти помилкового розпізнавання, що унеможливлювало б коректну оцінку якості перекладу як такого.

Натомість порівняльний аналіз двоетапного та одноетапного підходів було сфокусовано на використанні MLLM як на кроці розпізнавання, так і на кроці перекладу, а різниця між підходами визначається виключно наявністю чи відсутністю проміжного текстового результату розпізнавання.

2.3.4 Двоетапний підхід до перекладу манги на основі LLM

Після встановлення того, що MLLM є оптимальним рішенням для першого кроку розпізнавання, увага дослідження зосередилася на другому кроці двоетапного підходу, а саме безпосередньому перекладі. У цьому варіанті модель отримує на вхід зображення оригінальної сторінки манги для розуміння візуального контексту та японський текст, витягнутий на попередньому кроці. Використання обох джерел, зображення та тексту одночасно, є особливістю, що відрізняє цей варіант від чистого текстового перекладу без зображення.

Для виконання перекладу на другому кроці було розроблено окремий промпт, орієнтований на задачу якісного та контекстно узгодженого перекладу японського тексту манги. У блоці завдання моделі ставиться роль досвідченого перекладача манги. У блоці вхідних даних зазначається, що модель отримує зображення для візуального контексту, такого як стать персонажів, тональність сцени, обстановка, та транскрипцію японського тексту сторінки. Настанови щодо перекладу вказують на необхідність використовувати зображення для визначення мовця та його емоційного стану, вживати природні англійські ідіоми та уникати дослівних перекладів,

зберігати характерні мовленнєві патерни персонажів, наприклад ввічливу мову. У виводі повертати лише текст перекладу, зберігати порядкову структуру вихідного тексту та не додавати приміток перекладача чи узагальнень. Повний текст промпту для другого кроку наведено у лістингу 2.2.

Лістинг 2.2 Промпт перекладу для двоетапного підходу:

[TASK] You are an expert Manga Translator. Translate the provided Japanese text into natural, localized English.

[INPUTS]

1. Image used for visual context (character gender, tone, and setting).

2. The Japanese transcript of the page provided below.

[TRANSLATION GUIDELINES]

1. Contextual Accuracy: Use the image to determine who is speaking and their emotional state.

2. Localization: Use natural English idioms. Avoid "stiff" or literal translations.

3. Tone: Maintain character-specific speech patterns (e.g., polite vs. rude).

4. Direct Speech: Wrap all dialogue in single apostrophes (‘...’).

[OUTPUT]

1. Return only the English translation.

2. Maintain the line-by-line structure of the source text.

3. No translator notes or summaries.

Для тестування двоетапного підходу до перекладу були обрані три MLLMs, що демонстрували найкращі результати у попередніх тестах, а саме Gemini 3 Pro, Gemini 2.5 Flash та ChatGPT 5.4. Вибір саме цих моделей обумовлений їх провідними позиціями у бенчмарках мультимодального розуміння та перекладу на момент проведення експеременів [23], а також доступністю через публічний API. Кожна з моделей отримувала зображення сторінки та японський текст, розпізнаний на попередньому кроці за

допомогою MLLM у режимі OCR, після чого генерувала англійський переклад.

Двоетапний підхід надає моделі перекладу явний текстовий ввід, що теоретично зменшує ризик пропустити частину тексту або неправильно розпізнати його безпосередньо під час перекладу (рис. 2.5). Водночас він обмежує модель тією якістю розпізнавання, яка була досягнута на першому кроці, і не дозволяє їй самостійно «переглянути» зображення при виникненні неоднозначностей.

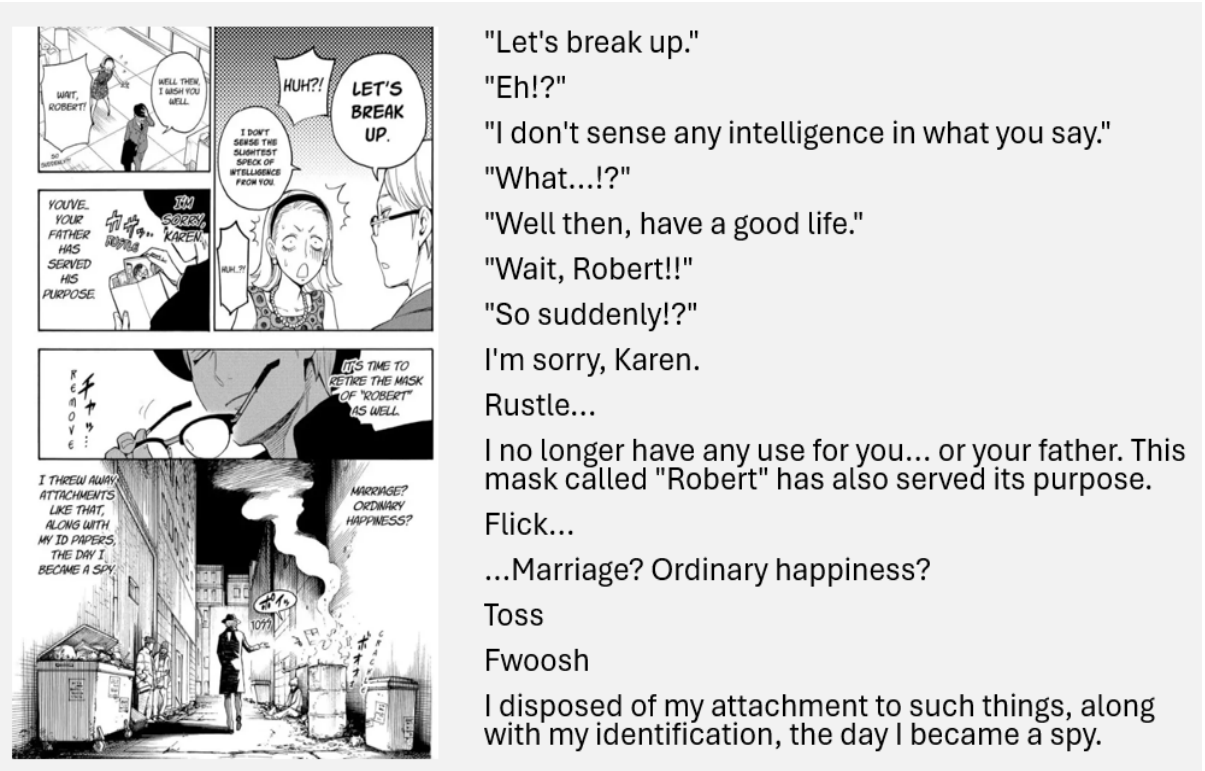


Рисунок 2.5 – Приклад результатів перекладу за допомогою Gemini 2.5 Pro

З точки зору технічних характеристик двоетапний підхід передбачає два окремих виклики API. Перший виклик для розпізнавання та другий для перекладу. Це суттєво впливає як на загальний час обробки сторінки, так і на вартість використання API, оскільки кожен виклик тарифікується окремо, що виникає супутні з цим проблеми.

2.4 Одноетапний підхід до перекладу манги на основі LLM

Одноетапний підхід є принципово відмінним від двоетапного, бо замість послідовного виконання розпізнавання та перекладу, мультимодальна модель отримує на вхід безпосередньо зображення сторінки манги та повертає готовий переклад без жодного проміжного текстового представлення. У такий спосіб модель вирішує задачу наскрізного «image-to-translation» перетворення, спираючись на власні здатності до одночасного розуміння візуального та мовного контенту.

Перевагою цього підходу є те, що MLLM не обмежена якістю попереднього розпізнавання, тому що у разі виникнення неоднозначності при читанні тексту модель може враховувати позу та вираз обличчя персонажа.Крім того, одноетапний підхід вимагає лише одного звернення до API, що безпосередньо впливає на швидкість обробки та вартість використання.

Для тестування одноетапного підходу було розроблено спеціалізований промпт, що визначає завдання моделі як наскрізний процес, що об’єднує розпізнавання, контекстний аналіз та переклад манги в єдиній інструкції. Текст промпту наведено у лістингу 2.3.

Лістинг 2.3 Промпт для однокроковог підходу:

[TASK] You are an expert AI Manga Translator. Your task is to perform an end-to-end process of OCR, context-aware analysis, and English translation on the provided manga page image.

[RULES]

1. Read the provided manga image as the sole source.

2. The primary reading order is from right to left, top to bottom. Follow this logic.

[TRANSLATION GUIDELINES]

1. Focus only on relevant text for a human-readable English version.

2. DO NOT include onomatopoeia (SFX like "BANG," "ZAP," "RUMBLE"). Ignore them entirely.

3. Keep translations precise to the scene and the characters' expressions.

4. If a word is a specific term from the manga universe (a name, an ability, a place), leave it untranslated, or include the original Japanese in parentheses if helpful for context.

[OUTPUT]

1. Generate the final output with a clean-text-only format.

2. Use single apostrophes ('...') to identify each distinct piece of dialogue or speech bubble.

3. Every speech bubble or text box should have its own line, preceded by a blank line for readability.

4. DO NOT provide any notes, commentary, or introduction of any kind. Deliver the translations in reading order.

5. Provide a direct translation of the page's text contents, without any headers or extra text.

Одноетапний підхід тестувався на тих самих трьох моделях, Gemini 3 Pro, Gemini 2.5 Flash та ChatGPT 5.4, та на тих же 20 сторінках набору даних, що й двоетапний підхід. Це забезпечило коректність порівняння між підходами в однакових умовах.

З технічної точки зору одноетапний підхід передбачає єдиний виклик API, що разом зі зображенням отримує зазначений промпт та повертає текст перекладу. Відсутність проміжного кроку розпізнавання безпосередньо впливає на часові характеристики обробки. Водночас одноетапний підхід вимагає від моделі одночасного виконання двох когнітивно складних завдань читання тексту та його перекладу, що теоретично може призводити до більшої кількості пропущених або переплутаних реплік порівняно з підходом, де кожен крок виконується окремо (рис. 2.6).

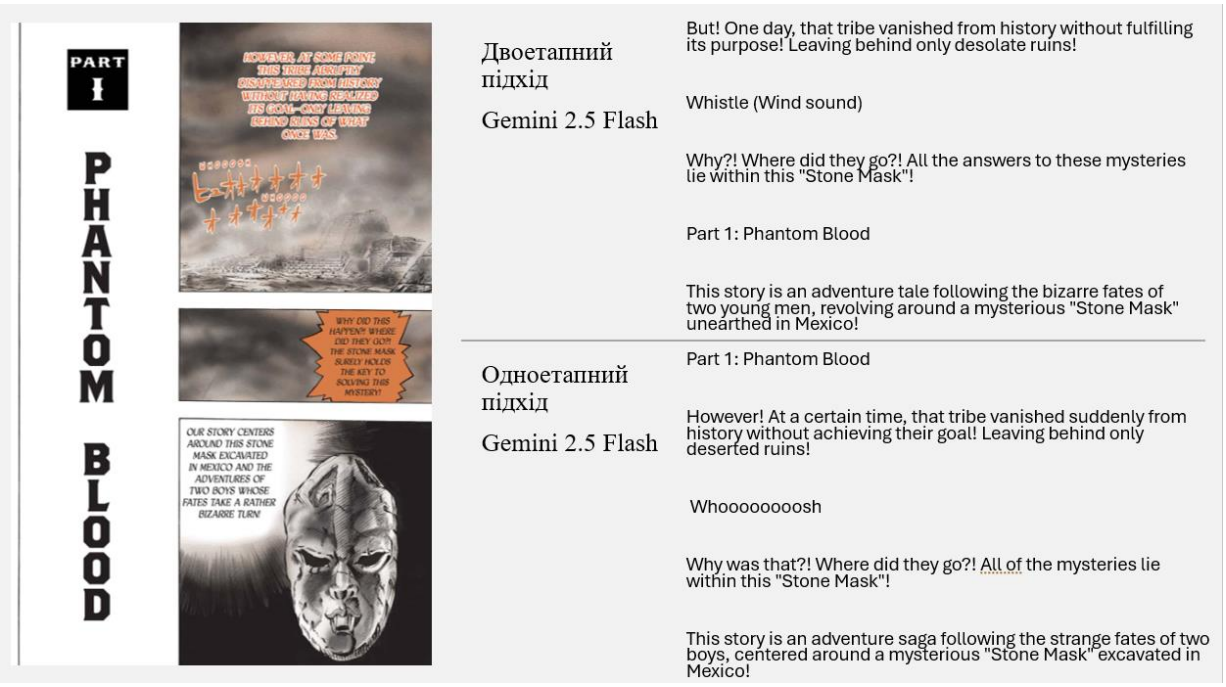


Рисунок 2.6 – Порівняння результатів двоетапного та одноетапного підходів

Аналіз якості перекладу обох підходів здійснюється за результатами експертного опитування.

2.5 Оцінка якості перекладу

Оскільки автоматичні метрики якості перекладу мають відомі обмеження при оцінці живої художньої мови та не здатні враховувати природність, комунікативну адекватність та відповідність стилю оригіналу, для оцінки якості перекладу у цьому дослідженні застосовується метод суб’єктивного експертного оцінювання.

Для збору даних було розроблено структуровану форму опитування. До участі залучено 10 експертів, які мають досвід читання манги та достатній рівень знання англійської мови для оцінки якості перекладу (рис. 2.7). Кожному експерту пропонувалося оцінити переклад 10 сторінок набору даних, представлених у шести варіантах, тобто два підходи для кожної з трьох моделей [9].

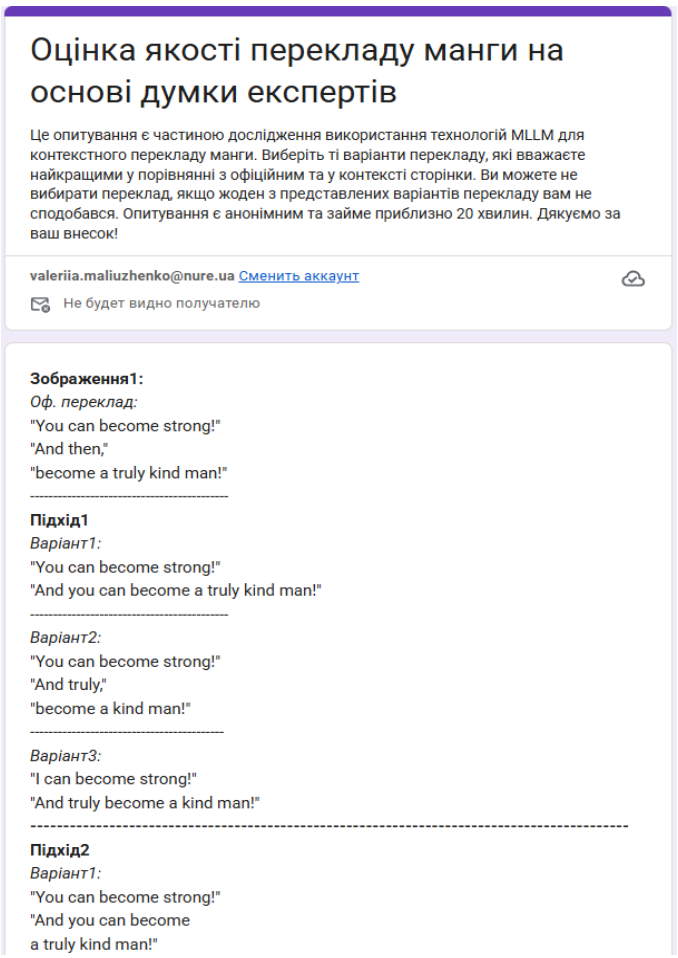


Рисунок 2.7 – Фрагмент форми опитування

Для кожного варіанту перекладу однієї сторінки експерт обирав найбільш якісний переклад із запропонованих шести для кожного підходу (рис. 2.8). Такий формат «вибору найкращого» дозволяє уникнути проблеми калібрування абсолютних оцінок між різними експертами та зосередитися на відносних перевагах підходів (рис. 2.9). Форма також містила фото оригінальної сторінки манги для забезпечення контексту.

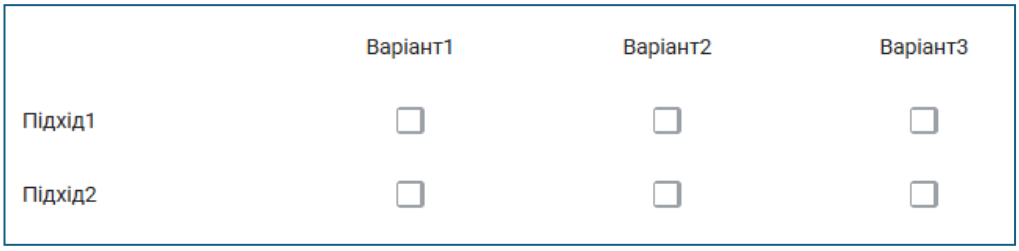


Рисунок 2.8 – Вигляд блоку відповідей створеної форми

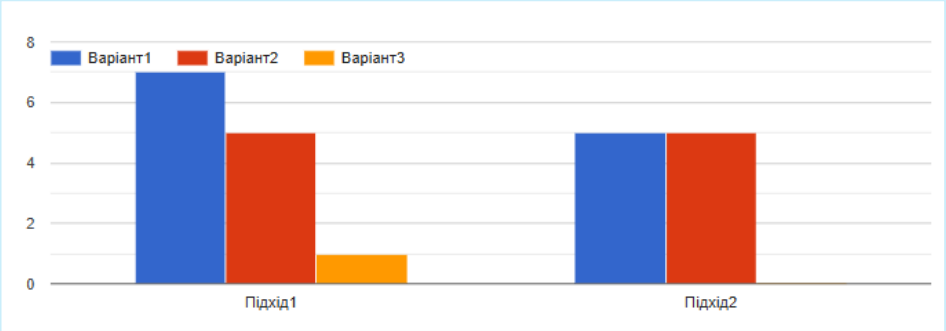


Рисунок 2.9 – Приклад статистики відповідей одного з питань

Узагальнені результати опитування наведено на рисунку 2.10.

	Варіанти відповіді					
	Однокроковий підхід			Двокроковий підхід		
№	Gemini 2.5 Flash	Gemini 3 Pro	Chat GPT 5.4	Gemini 2.5 Flash	Gemini 3 Pro	Chat GPT 5.4
1	7	5	1	5	4	0
2	6	5	0	6	3	2
3	8	5	0	3	6	0
4	9	2	0	3	8	0
5	9	3	2	5	3	0
6	6	4	0	6	5	1
7	10	3	0	2	7	0
8	5	7	0	7	2	1
9	8	6	0	2	6	0
10	8	2	0	3	8	1
	76	42	3	42	52	5

Рисунок 2.10 – Результати опитування експертів

Аналіз результатів дозволяє зробити декілька висновків. За загальною кількістю голосів лідером якості є модель Gemini 2.5 Flash, переклади цієї моделі найчастіше обирались експертами як найкращі. Разом з тим відрив від Gemini 3 Pro є відносно невеликим, тоді як ChatGPT 5.4 отримав помітно менше позитивних оцінок. Також порівняння підходів для кожної моделі демонструє відсутність явного домінування одного підходу над іншим, тобто для Gemini 3 Pro та ChatGPT 5.4 незначна перевага на боці двоетапного, у той час як для Gemini 2.5 Flash одноетапного. Це свідчить про те, що різниця у якості між підходами є несуттєвою для більшості сторінок, а відтак вибір між ними повинен спиратися переважно на технічні критерії швидкості та вартості.

2.6 Висновки щодо обрання підходу для перекладу

На основі результатів усіх проведених експериментів здійснено зведений порівняльний аналіз трьох MLLM за трьома критеріями відбору. Результати аналізу систематизовано на рисунку 2.11, зеленим та фіолетовим кольором виділено перші та другі результати відповідно.

			Середні часові витрати, с	Вартість, $/ млн.токенів	Оцінка експертів
Підходи	Одноетапний	Gemini 2.5 Flash	3,3	2,8	76
		Gemini 3 Pro	32,1	17,5	42
		Chat GPT 5.4	3,8	5,25	3
	Двоетапний	Gemini 2.5 Flash	4,6	5,6	42
		Gemini 3 Pro	30,7	35	52
		Chat GPT 5.4	21,7	10,5	5

Рисунок 2.11 – Порівняльний аналіз MLLM за критеріями відбору

Gemini 2.5 Flash демонструє найкращі показники за всіма обраними критеріями, тож з точки зору практичної доцільності для розробки застосунку було обрано саме цю модель.

При виборі підходу порівняння двоетапного та одноетапного методів показало відсутність суттєвої різниці у якості перекладу за оцінкою експертів. При цьому одноетапний підхід є швидшим та економічнішим, оскільки вимагає лише одного виклику API. Зокрема, для Gemini 2.5 Flash одноетапний підхід забезпечує час обробки 3,3 с проти 4,6 с для двоетапного, тож бачимо різницю у 28%. При масштабуванні ця різниця матиме значний економічний ефект.

На підставі проведеного аналізу для подальшої розробки застосунку було обрано одноетапний підхід на основі моделі Gemini 2.5 Flash. Цей вибір

забезпечує прийнятну якість перекладу, мінімальний час відповіді для кінцевого користувача та найнижчу вартість обробки одиниці контенту.

2.7 Розробка аглоритму

На основі результатів проведеного експериментального аналізу та обраного підходу було розроблено алгоритм функціонування системи. Алгоритм описує повний цикл обробки вхідного зображення від отримання запиту від користувача до повернення готового перекладу і є основою для технічної реалізації застосунку.

Даний алгоритм здійснюється за наступною схемою:

Крок 1. Отримання вхідного зображення від користувача.

Крок 2. Підготовка запиту до API.

Крок 3. Надсилання запиту до моделі та отримання текстового результату.

Крок 4. Повернення результату користувачу.

На першому етапі система отримує від користувача зображення сторінки манги. Вхідними даними може бути як окреме растрове зображення, так і сторінка, виділена з PDF-документу. У другому випадку система виконує попередню конвертацію сторінки PDF до растрового зображення з достатньою роздільною здатністю для коректного розпізнавання дрібного тексту.

На другому етапі виконується підготовка запиту до API. Зображення кодується у форматі base64 та разом зі стандартизованим промптом надсилається до API моделі Gemini 2.5 Flash. Промпт є фіксованим та не вимагає налаштування під конкретного користувача або серію манги.

На третьому етапі виконується запит та отримання відповіді. Модель обробляє зображення та повертає текстовий результат у вигляді перекладу реплік та текстових блоків сторінки у встановленому форматі.

На четвертому етапі система виконує постобробку відповіді, тобто валідацію формату виводу, видалення зайвих пробілів та порожніх рядків, перевірку на наявність коментарів моделі, що не є частиною перекладу. Після цього результат повертається користувачеві у вигляді структурованого тексту.

Описаний алгоритм є достатньо простим з точки зору архітектурної складності та не потребує складної інфраструктури для розгортання. Основне обчислювальне навантаження покладається на зовнішній API, що дозволяє реалізувати клієнтський застосунок без власних GPU-ресурсів.

3 РОЗРОБКА ВЕБ ТА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПЕРЕКЛАДУ МАНГИ

3.1 Розробка MVP вебзастосунку

3.1.1 Технічне завдання та мета MVP вебзастосунку

MVP (Minimal Viable Product) – це мінімально життєздатний продукт, тобто прототип із достатнім набором функцій для демонстрації та перевірки основної гіпотези без повноцінної реалізації всіх запланованих можливостей. MVP-вебзастосунок слугує інструментом для практичної перевірки гіпотези дослідження, а саме чи здатна обрана модель Gemini 2.5 Flash у рамках однокрокового підходу забезпечити достатню якість перекладу манги в умовах реального використання.

Необхідність розробки MVP обумовлена тим, що тестування на фіксованому наборі сторінок не відтворює повністю умов реального застосування. Технічне завдання на розробку MVP вебзастосунку передбачає реалізацію таких функціональних вимог, як можливість застосунку приймати на вхід зображення сторінки манги у форматах JPEG та PNG, так само як і PDF-файли, що містять одну або декілька сторінок. У разі завантаження PDF-документу застосунок повинен забезпечити можливість перекладу кожної сторінки окремо. Результат перекладу відображається користувачеві у текстовому вигляді безпосередньо в інтерфейсі застосунку. Зберігання даних не передбачається, бо застосунок не використовує базу даних та не зберігає жодних завантажених файлів або результатів перекладу після завершення сеансу.

До нефункціональних вимог відносяться простота та зрозумілість інтерфейсу, що не вимагає від користувача технічних знань, мінімальний час від завантаження файлу до отримання результату, можливість локального розгортання застосунку без необхідності хмарної інфраструктури. Оскільки MVP є дослідницьким прототипом, а не комерційним продуктом, він

запускається локально на сервері розробника та потребує запуску серверного процесу перед використанням.

3.1.2 Проєктування архітектури

Архітектура MVP вебзастосунку відповідає класичній клієнт-серверній моделі з тонким клієнтом. Усі обчислення та взаємодія з зовнішнім API виконуються на серверній стороні, тоді як клієнтська частина відповідає виключно за відображення інтерфейсу та передачу файлів.

Серверна частина реалізована на Python і включає три логічних компоненти. Модуль обробки вхідних даних відповідає за прийом завантаженого файлу, визначення його типу та підготовку до передачі в модуль перекладу. У разі отримання PDF-документу цей модуль виконує конвертацію кожної сторінки до растрового зображення з роздільною здатністю, достатньою для коректного розпізнавання тексту. Модуль перекладу формує запит до API Gemini 2.5 Flash, включаючи кодування зображення у форматі base64 та підстановку стандартизованого промпту, і повертає отриманий текстовий результат. Модуль інтерфейсу на базі Gradio визначає структуру вебсторінки, обробляє події завантаження файлів та відображає результати перекладу [22].

Взаємодія між компонентами є лінійною та однонаправленою, тобто запит від користувача проходить послідовно через модуль обробки вхідних даних, модуль перекладу та повертається до інтерфейсу. Відсутність бази даних та механізмів збереження стану спрощує архітектуру та виключає необхідність управління сеансами користувачів.

Застосунок побудований на основі набору сторонніх бібліотек, кожна з яких виконує чітко визначену роль у загальній архітектурі системи. Бібліотека gradio забезпечує побудову вебінтерфейсу користувача та надає декларативний підхід до створення інтерактивних компонентів без

необхідності написання коду на JavaScript. Бібліотека google-generativeai є офіційним Python SDK для доступу до API Gemini та забезпечує структуровану взаємодію із сервісами генеративного штучного інтелекту від Google [24]. Для конвертації сторінок PDF-документів у растрові зображення застосовується утиліта poppler, зокрема його компонент pdftoppm, що виконує растеризацію сторінок із заданою роздільною здатністю [25]. Бібліотека Pillow використовується для базової обробки отриманих зображень перед їх кодуванням та передачею до моделі. Усі залежності є загальнодоступними пакетами, що встановлюються через стандартний менеджер пакетів pip, що суттєво спрощує розгортання застосунку в різних середовищах.

Загальна архітектура вебзастосунку відображає принцип мінімальної складності, характерний для MVP-рішень. Взаємодія між компонентами системи організована як послідовний конвеєр обробки даних без збереження проміжного стану між запитами (рис. 3.1).

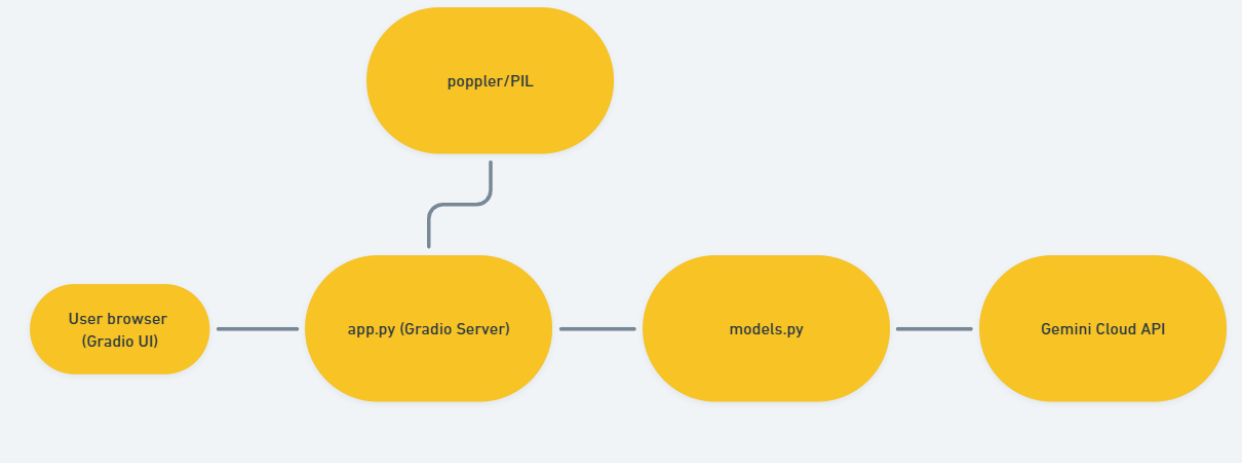


Рисунок 3.1 – Архітектура вебзастосунку

Кожен запит проходить через фіксовану послідовність етапів, таких як отримання PDF-файлу, його растеризація, формування мультимодального запиту до API Gemini та повернення перекладеного тексту у вебінтерфейс. Лінійний потік виконання унеможливлює виникнення побічних ефектів між окремими запитами та забезпечує детермінованість поведінки системи.

Застосунок не використовує реляційну або нереляційну базу даних. Це рішення обумовлене цільовим призначенням прототипу. Оскільки завдання MVP полягає виключно у перевірці якості автоматичного перекладу, зберігання даних користувачів не є функціонально необхідни. Кожен запит на переклад є повністю незалежним від попередніх запитів, що відповідає принципу stateless-архітектури та спрощує горизонтальне масштабування у разі необхідності (рис. 3.2).

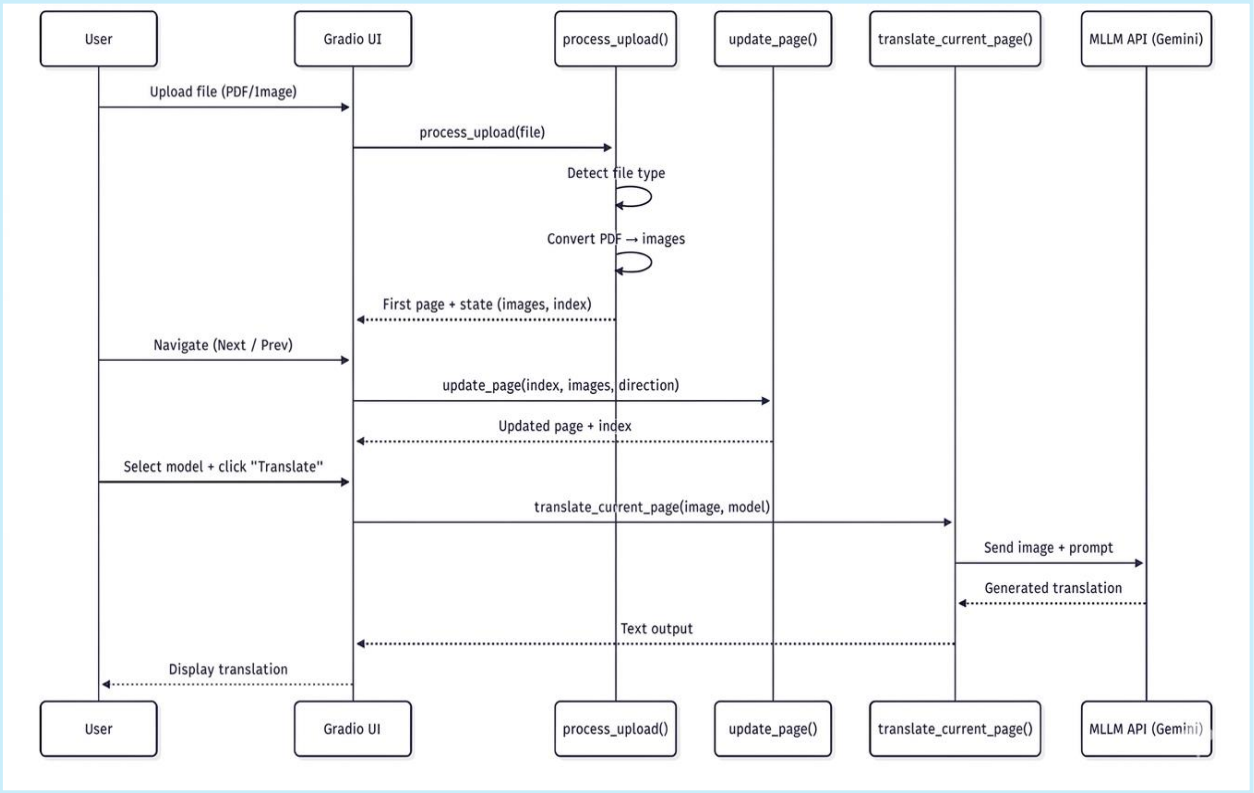


Рисунок 3.2 – UML діаграма послідовності процесу обробки запиту

Відсутність персистентного стану між запитами унаочнює архітектурне рішення щодо відмови від бази даних та підтверджує доцільність обраного підходу в контексті поставлених вимог до прототипу. Це дозволяє спростити архітектуру вебзастосунку. Оскільки кожна сесія обробки документів є ізольованою, система функціонує за принципом stateless-сервісу, що забезпечує високу швидкість обробки інформації.

3.1.3 Розробка дизайну вебзастосунку

Дизайн Gradio-застосунку визначається набором стандартних компонентів бібліотеки, розміщених у визначеній розробником структурі.

Інтерфейс складається з чотирьох основних блоків. Блок заголовку містить назву застосунку та короткий опис його призначення, що орієнтує користувача щодо функціональності без необхідності читати документацію. Блок області завантаження файлу є центральним елементом інтерфейсу та підтримує як клік для вибору файлу, так і перетягування файлу у зону завантаження. Блок приймає файли форматів JPEG, PNG та PDF і відображає прев'ю завантаженого зображення або першої сторінки PDF. Блок панелі керування для PDF-файлів з'являється динамічно при завантаженні PDF-документу та дозволяє користувачеві обрати конкретну сторінку для перекладу за допомогою числового поля введення або перейти до наступної/попередньої сторінки кнопками навігації. Блок області результату відображає отриманий переклад у текстовому полі з можливістю ручного копіювання вмісту. Кнопка запуску перекладу розміщена між блоком завантаження та областю результату і є єдиною активною дією, що вимагається від користувача (рис. 3.3).

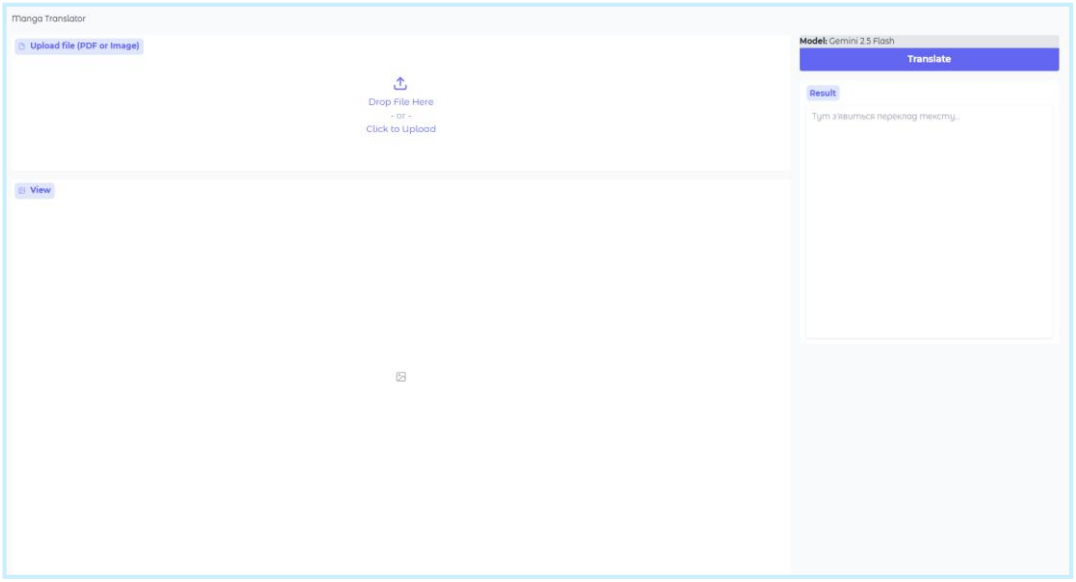


Рисунок 3.3 – Макет інтерфейсу вебзастосунку

Колірна схема та типографіка застосунку відповідають стандартній темі Gradio, що забезпечує нейтральний та сучасний вигляд без необхідності додаткового налаштування стилів. Адаптивна верстка Gradio автоматично коригує розташування блоків для різних розмірів екрану, що дозволяє використовувати застосунок як на десктопі, так і на мобільному пристрої через браузер.

3.1.4 Ілюстрація роботи вебзастосунку

Для демонстрації роботи MVP вебзастосунку наведено приклади обробки двох типів вхідних файлів, а саме окремого зображення сторінки манґи та PDF-документу з кількома сторінками.

У першому сценарії користувач завантажує зображення сторінки манґи у форматі JPEG. Після натискання кнопки перекладу застосунок кодує зображення та надсилає запит до API Gemini 2.5 Flash. В області результату відображається текст перекладу, де кожна репліка виокремлена одинарними лапками та розміщена на окремому рядку у порядку читання сторінки справа ліворуч, зверху донизу (рис. 3.4).

Рисунок 3.4 – Переклад окремого зображення сторінки манги

У другому сценарії користувач завантажує PDF-файл із томом манґи. Після завантаження активується панель навігації по сторінках, що відображає загальну кількість сторінок документу та дозволяє перейти до потрібної. Після вибору сторінки та натискання кнопки перекладу виконується конвертація відповідної сторінки PDF у растрове зображення та аналогічний до першого сценарію процес передачі до API. Час обробки PDF-сторінки є дещо більшим через додатковий крок конвертації (рис. 3.5).

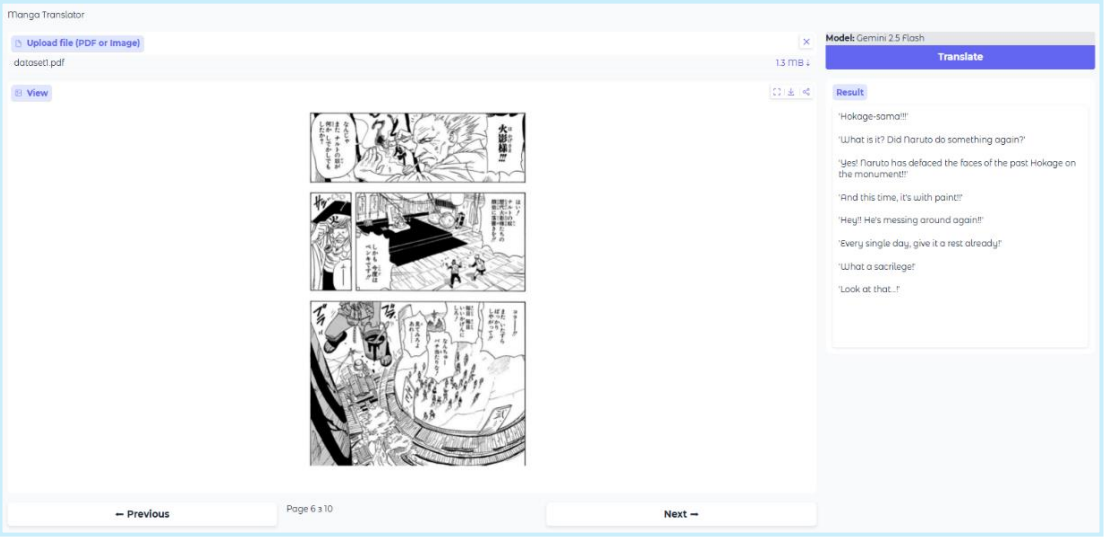


Рисунок 3.5 – Переклад сторінки з PDF-документу з активованою навігацією

Модель коректно визначає порядок читання реплік, зберігає характерні мовленнєві патерни персонажів та ігнорує звукові ефекти відповідно до інструкцій промпту.

3.1.5 Аналіз розробленого вебзастосунку

Розроблений MVP вебзастосунок на базі Gradio виконав свою основну функцію, тобто підтвердив гіпотезу дослідження на практиці. Однокроковий підхід до перекладу манґи на основі моделі Gemini 2.5 Flash демонструє прийнятну якість результату не лише у контрольованих умовах тестового

набору даних, а й при роботі з довільними сторінками різних серій та жанрів, завантаженими реальними користувачами. Це підтверджує узагальнюваність обраного підходу та його придатність як основи для повноцінного застосунку.

Водночас, практична експлуатація MVP виявила суттєвий недолік, що не був очевидним у ході тестування. Реальний час обробки сторінки виявився помітно більшим, ніж фіксований у тестових дослідженнях показник 3,3 секунди. Це пояснюється тим, що тестове середовище використовувало стабільне мережеве з'єднання та оптимальне навантаження на API, тоді як у реальних умовах на час відповіді впливають нестабільність мережі, черговість запитів до API та варіативність розміру і складності зображень. У ряді випадків час очікування перекладу сягав 10-15 секунд.

На фоні збільшеного часу очікування особливо гострим виявився інший недолік MVP, а саме відсутність можливості зберегти отриманий переклад. Оскільки застосунок не передбачає жодного механізму збереження результатів, користувачі були змушені вручну копіювати текст перекладу або повторно запускати переклад тієї самої сторінки після оновлення браузера. В умовах збільшеного часу очікування необхідність повторного перекладу становила значну незручність і знижувала практичну цінність застосунку. Саме цей недолік був визначений як пріоритетний для усунення при розробці повноцінного мобільного застосунку.

Підбиваючи підсумки, MVP виконав свою функцію підтвердження технічної та якісної спроможності обраного підходу. З іншого боку розроблений вебзастосунок визначив конкретні вимоги до наступної ітерації розробки. Ключовими вимогами, що випливають з аналізу MVP, є реалізація механізму збереження результатів перекладу, оптимізація взаємодії з API для зменшення впливу мережевої нестабільності на час відповіді та розробка нативного мобільного інтерфейсу замість вебінтерфейсу для забезпечення кращого користувацького досвіду.

3.2 Розробка мобільного застосунку

3.2.1 Опис функціональності

Мобільний застосунок є повноцінною реалізацією продукту на основі підходу, підтвердженого у ході розробки та тестування MVP. На відміну від вебпрототипу, що виконував лише функцію перекладу без збереження, мобільний застосунок охоплює повний цикл взаємодії користувача з мангою від організації особистої бібліотеки та читання до перекладу та ведення статистики. Ключовою відмінністю від MVP є наявність постійного локального сховища, що усуває головний недолік прототипу, тобто необхідність повторного перекладу при кожному запуску.

Функціональні вимоги до застосунку охоплюють п’ять основних модулів, такі як управління бібліотекою, читання та перегляд, переклад, статистика та система досягнень.

Модуль управління бібліотекою забезпечує організацію контенту на двох рівнях. Перший рівень предсавляє окремі файли манги у форматах JPEG, PNG та PDF, що завантажуються безпосередньо. Другий рівень представляє серії, що є контейнерами для групування пов’язаних файлів у вигляді глав. Кожна серія може містити довільну кількість глав, доданих у визначеному користувачем порядку. Для кожного елементу бібліотеки, як окремого файлу, так і серії, користувач може встановити статус, наприклад, «читаю», «прочитано», «заплановано», та числову оцінку. Бібліотека підтримує фільтрацію за статусом та оцінкою, а також сортування за датою додавання, назвою та оцінкою, що дозволяє ефективно орієнтуватися у великій колекції.

Модуль читання реалізований у вигляді переглядача, що відображає сторінки файлу або глави посторінково. Переглядач містить блок відображення поточної сторінки та блок перекладеного тексту під ним. Навігація між сторінками здійснюється кнопками «попередня» та «наступна сторінка», а також прямим переходом до обраної сторінки через введення її номера. Якщо файл є главою всередині серії та серія містить наступну главу,

на останній сторінці поточної глави з’являється можливість перейти до першої сторінки наступної глави без повернення до екрану серії. Для кожної сторінки користувач може залишити текстову нотатку, що зберігається локально та відображається при повторному відкритті тієї самої сторінки.

Модуль перекладу інтегрований як у рівень серії, так і у рівень окремої сторінки. На екрані серії кнопка перекладу глави запускає послідовний переклад усіх сторінок вибраної глави одним натисканням. Усередині переглядача кнопка перекладу поточної сторінки виконує запит до API та відображає результат у блоці перекладу під зображенням. Результати перекладу зберігаються локально, що виключає необхідність повторних запитів до API при повторному відкритті сторінки. Глави серії можуть бути позначені як прочитані або непрочитані безпосередньо з екрану серії.

Модуль статистики надає детальну інформацію про активність користувача у вигляді чотирьох діаграм, двох кругових та двох стовпчастих. Перша кругова діаграма відображає розподіл елементів бібліотеки за статусом у відсотках відносно загальної кількості. Друга вдображає розподіл за оцінками. Перша стовпчаста діаграма показує кількість виконаних перекладів по днях, друга показує кількість прочитаних сторінок по днях. Додатково відображається загальний відсоток перекладених сторінок відносно усіх завантажених. Скорочена версія статистики виводиться безпосередньо на головному екрані.

Модуль досягнень розміщений у нижній частині екрану статистики та містить набір нагород, що розблоковуються при досягненні певних порогових значень, таких як, наприклад, кількість прочитаних сторінок, кількість доданих манг, кількість виконаних перекладів. Система досягнень є мотиваційним елементом, що стимулює регулярне використання застосунку та допомагає користувачеві відслідковувати загальний прогрес власної бібліотеки.

3.2.2 Налаштування програмного середовища

Застосунок реалізовано за архітектурою клієнт-сервер, де клієнтська частина є мобільним застосунком на базі React Native, а серверна частина представлена REST API на базі Django. Вибір цих технологій зумовлений їх зрілістю, широкою екосистемою бібліотек та можливістю швидкого прототипування.

Серверна частина розроблена на мові Python з використанням вебфреймворку Django. Для організації REST API застосовано бібліотеку Django REST Framework, яка надає зручні засоби серіалізації даних, автентифікації та маршрутизації запитів. Обробка міжсайтових запитів, CORS, забезпечується пакетом django-cors-headers, що дозволяє мобільному клієнту взаємодіяти з API з будь-якого походження в режимі розробки. Дані фреймворки визнані стандартом мобільної розроби та використовуються у мобільних застосунках різних направленостей [26, 27]. Середовище виконання JavaScript забезпечується оптимізованим JavaScript-рушієм Hermes, розробленим спеціально для React Native. Hermes є рушієм за замовчуванням у сучасних версіях React Native та забезпечує швидший запуск застосунку та знижене споживання пам’яті порівняно з JavaScriptCore завдяки попередній компіляції JavaScript-коду у байткод.

Обробка завантажених PDF-файлів здійснюється бібліотекою pdf2image, яка конвертує сторінки документа у растрові зображення формату JPEG. Бібліотека залежить від системного пакету Poppler, який виконує безпосередній розбір формату PDF [27]. Подальша обробка зображень, а саме стиснення, конвертація колірних моделей та кодування у формат Base64 для передачі через API, реалізована засобами бібліотеки Pillow версії 10. Завантаження змінних середовища з файлу конфігурації здійснюється пакетом python-dotenv.

Взаємодія з моделлю штучного інтелекту для перекладу сторінок манґи реалізована через офіційний Python SDK бібліотеки google-generativeai.

Для підтримки паралельного пакетного перекладу без блокування основного потоку виконання використовується вбудований модуль threading стандартної бібліотеки Python.

У якості системи управління базами даних на етапі розробки обрано SQLite, яка не потребує окремого серверного процесу і зберігає всі дані в єдиному файлі на диску. Детальні залежності серверної частини мобільного застосунку представлені і таблиці 3.1.

Таблиця 3.1 – Залежності серверної частини

Бібліотека	Версія	Призначення
Django	4.2.16	Основний вебфреймворк
djangorestframework	3.15	REST API
django-cors-headers	4.x	Підтримка CORS
Pillow	10.x	Обробка зображень
Pdf2image	1.17.x	Конвертація PDF
google-generativeai	0.8	Переклад через Gemini API
python-dotenv	1.x	Управління конфігурацією

Мобільний клієнт розроблено на платформі Expo SDK з використанням фреймворку React Native. Expo забезпечує уніфіковане середовище збірки для платформ iOS та Android, а також надає набір готових нативних модулів. Маршрутизація між екранами реалізована засобами бібліотеки Expo Router, яка забезпечує файлову систему маршрутів аналогічно до Next.js. Додатково для окремих екранів застосовується React Navigation зі стеком навігації, що дозволяє управляти навігацією програмно.

Вибір файлів для завантаження здійснюється через expo-document-picker, який надає нативний системний діалог вибору документів. Анімації інтерфейсу, зокрема висувна панель перекладу з підтримкою жесту перетягування, реалізовані засобами вбудованого модуля Animated з React Native у поєднанні з PanResponder для обробки жестів. Для побудови кругових

діаграм на екрані статистики використовується бібліотека react-native-svg, яка надає повноцінну підтримку векторної графіки SVG у середовищі React Native через JavaScript-інтерфейс до нативних SVG-рушіїв платформи. Детальні залежності клієнтської частини мобільного застосунку представлені і таблиці 3.2.

Таблиця 3.2 – Залежності клієнтської частини мобільного застосунку

Бібліотека	Версія	Призначення
Expo SDK	52	Платформа мобільної розробки
React Native	0.76.x	UI-фреймворк
Expo Pouter	6.x	Файлова маршрутизація
react-navigation/stack	6.x	Програмна навігація
expo-document-picker	12.x	Вибір файлів
react-native-svg	15.x	SVG-графіка для діаграм
react-native-gesture-handler	2.x	Обробка жестів

Глобальний стан фонових задач перекладу зберігається у власному модулі, реалізованому без зовнішніх бібліотек управління станом. Модуль реалізує паттерн спостерігача з набором слухачів, які отримують сповіщення про зміну стану задачі незалежно від того, який екран наразі активний.

3.2.3 Проєктування архітектури та бази даних

Архітектура розробленного мобільного застосунку базується на клієнт-серверній моделі з чітким розділенням зон відповідальності між мобільним клієнтом, бекенд-сервером, реляційною базою даних та зовнішніми сервісами штучного інтелекту.

Клієнтську частину системи представлено мобільним застосунком,

розробленим на базі React Native із використанням інструментарію Expo. Навігація та маршрутизація всередині додатка реалізовані за допомогою Expo Router. Користувацький інтерфейс охоплює кілька ключових екранів. Екран бібліотеки, екран серій, інтерактивну панель читання та перекладу, а також екран аналітики для відображення статистичних графіків і досягнень. Взаємодія з сервером здійснюється через модуль api.js за допомогою HTTP-запитів, а локальний стан тривалих фонових завдань адмініструється через модуль store.js.

Серверна частина функціонує на базі Django REST Framework, виступаючи в ролі центрального вузла обробки бізнес-логіки. Взаємодія між мобільним клієнтом і сервером відбувається через REST API за допомогою обміну JSON-даними, а також через HTTP-тунель для специфічних потоків даних (рис. 3.6).

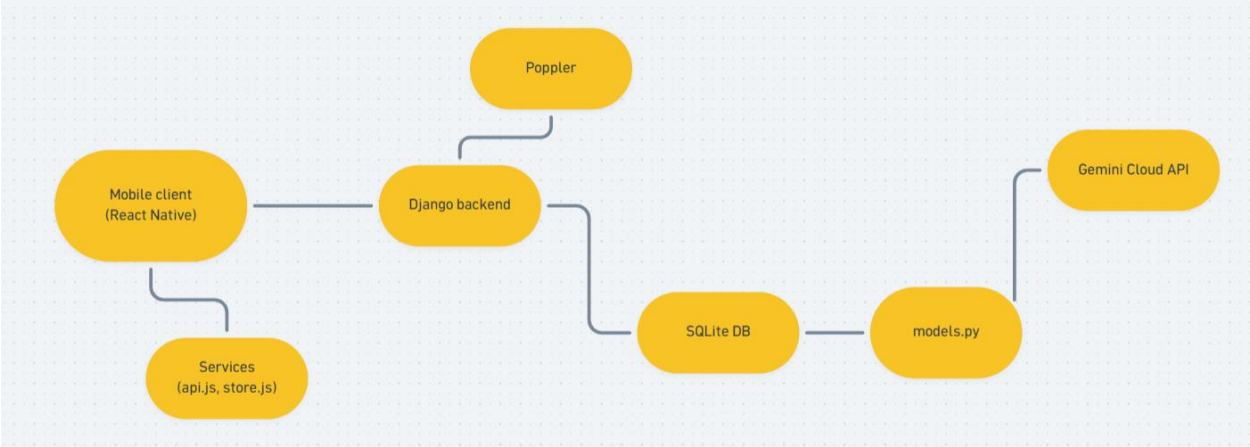


Рисунок 3.6 – Архітектура мобільного застосунку

Для збереження інформації застосовується реляційна база даних SQLite, яка є оптимальним рішенням для поточного етапу розгортання додатка. Схема бази даних містить взаємопов’язані сутності для збереження інформації про мангу та серії, окремі сторінки та нотатки користувачів, а також таблиці для фіксації прогресу читання та історії активності. Повний список таблиць бази даних представлено на рисунку 3.7.

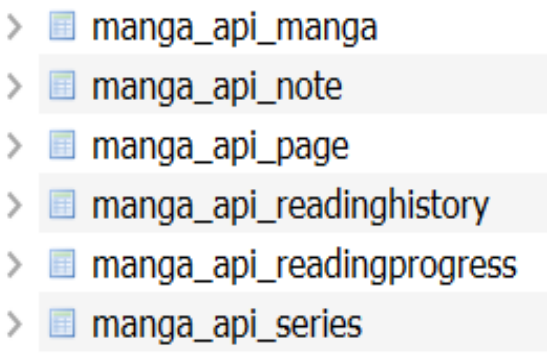


Рисунок 3.7 – Таблиці бази даних застосунку

База даних містить шість основних сутностей, зв’язки між якими забезпечують цілісність даних на рівні зовнішніх ключів з каскадним видаленням. Схема спроєктована таким чином, щоб одиничний твір та розділ серії були представлені однією моделлю Manga, де наявність зовнішнього ключа до Series визначає тип запису.

3.2.4 Розробка дизайну мобільного застосунку

Візуальний дизайн користувацького інтерфейсу застосунку було розроблено у графічному редакторі Figma. Ця платформа є хмарним інструментом для проєктування інтерфейсів, який поєднує в собі потужний векторний редактор та засоби для створення інтерактивних прототипів [28]. Головними перевагами Figma є можливість роботи в режимі реального часу, кросплатформенність, наявність зручних інструментів авторозмітки (для адаптивного дизайну та гнучка система компонентів, що дозволяє швидко вносити зміни та масштабувати проєкт без втрати цілісності.

Створений дизайн-макет відображає комплексну структуру мобільного застосунку, виконану в сучасній темній колірній палітрі, що мінімізує навантаження на зір під час тривалого читання. Центральним вузлом навігації є головний екран бібліотеки, від якого відгалужуються всі ключові користувацькі сценарії, а саме екрани перегляду статистики з круговими та

стовпчастими діаграмами, інтерфейси додавання і редагування контенту, та деталізовані сторінки розділів і самого рідера (рис. 3.8).

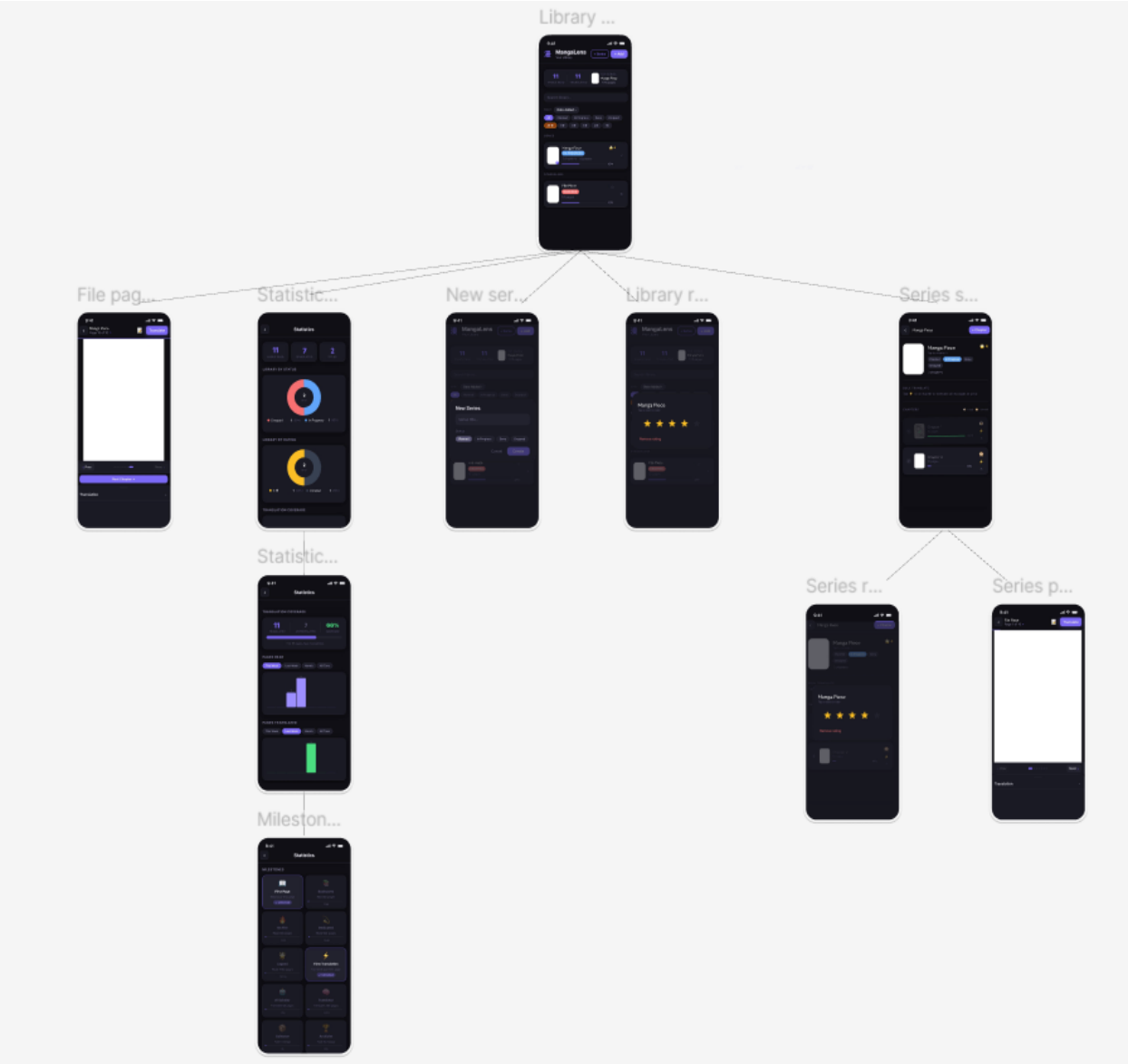


Рисунок 3.8 – Повна розгортка дизайну застосунку

Завдяки логічному компонуванню елементів керування, картковій структурі списків та чіткій візуальній ієрархії, створений інтерфейс забезпечує інтуїтивно зрозумілу взаємодію із функціоналом перекладу та аналітики читання для користувача.

3.2.5 Ілюстрація роботи мобільного застосунку

На рисунку 3.9 представлено головний екран бібліотеки застосунку у межах одного користувача, який представляє список з серій та одиночних файлів. У верхній частині екрану є блок з поточною статистикою бібліотеки, пошукова строка та інструменти для фільтрування та сортування елементів бібліотеки. Кнопка «+ Series» дозволяє додати до бібліотеки користувача пусту серію; кнопка «+ ADD» дозволяє завантадити окремий файл до бібліотеки.

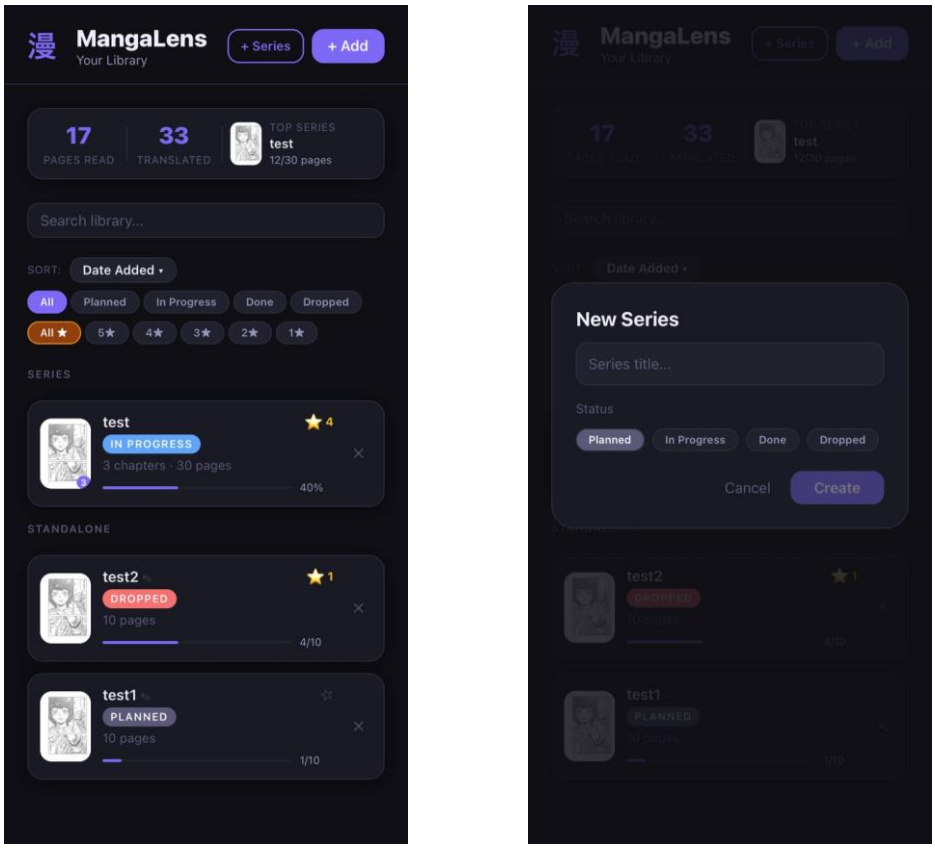


Рисунок 3.9 – Головний кран бібліотеки та інтерфейс додавання серії

При переході до сторінки статистики бачимо діаграми наповнення бібліоткеи за статусом та оцінкою, відсоток перекладених сторінок від усіх завантажених до бібліотеки, та діаграми активності користувача у певні дні. Внизу сторінки є блок з досягеннями користувача за весь час користування застосунком (рис. 3.10).

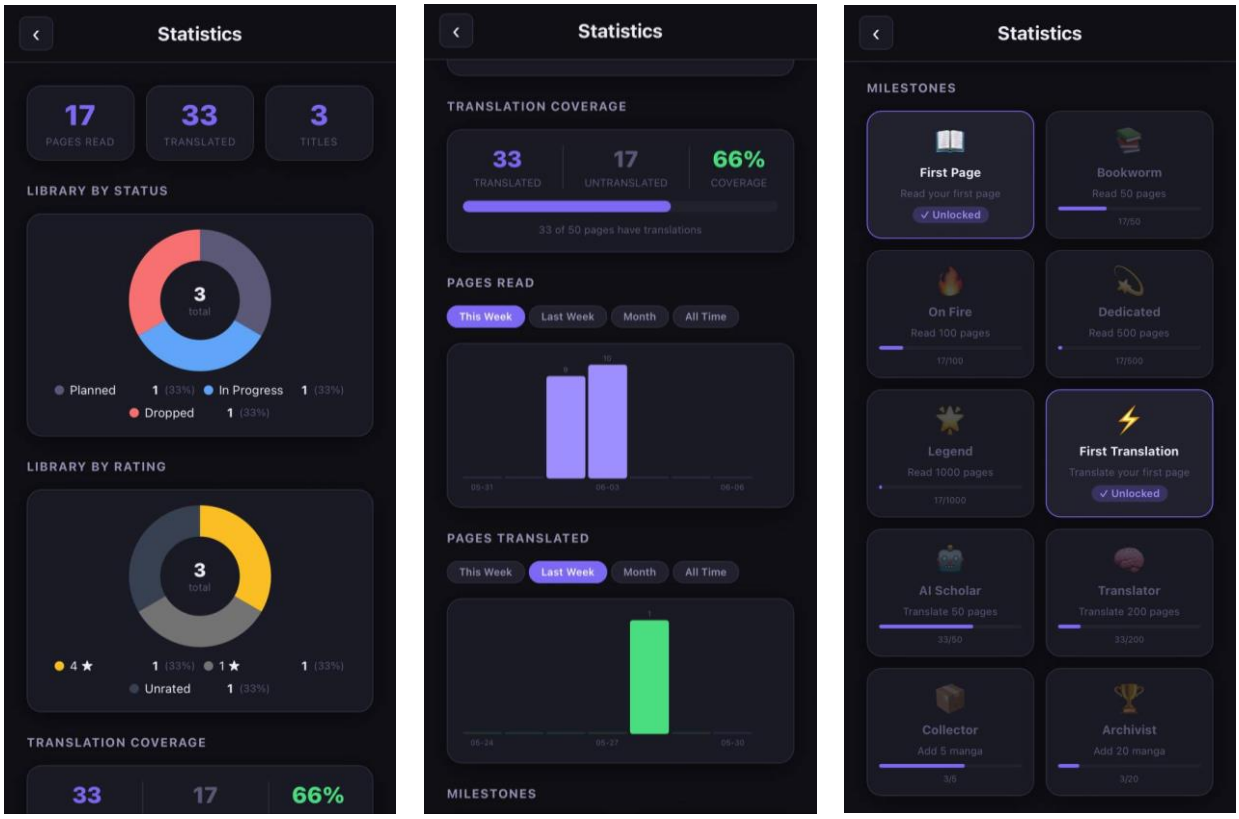


Рисунок 3.10 – Сторінка статистики

При натисканні на серію у бібліотеці, користувач переходить на сторінку відповідної серії, де має змогу змінити маркер серії, перекласти одразу всі сторінки однієї глави та відмітити главу як прочитану без необхідності переглядати всі сторінки. Всередині глави інтерфейс поділено на дві основні блоки, а саме блок сторінки та блок перегляду. Натиснувши на сторінку користувач має змогу написати нотаток який буде доступним до перегляду і надалі, між блоками розташовані кноки керування, які дозволяють перегортати сторінку. При знаходженні читача на останній сторінці глави, за умови наявності наступної глави, існує функціонал для переходу на наступну главу без необхідності виходу до екрану серії. При натисканні на номер сторінки кокистувач має змогу перейти до довільно введеної сторінки (рис. 3.11).

Користувач має змогу поставити оцінку серії або окремому файлу завантаженому до бібліотеки та виконувати подальшу фільтрацію на базі оцінок.

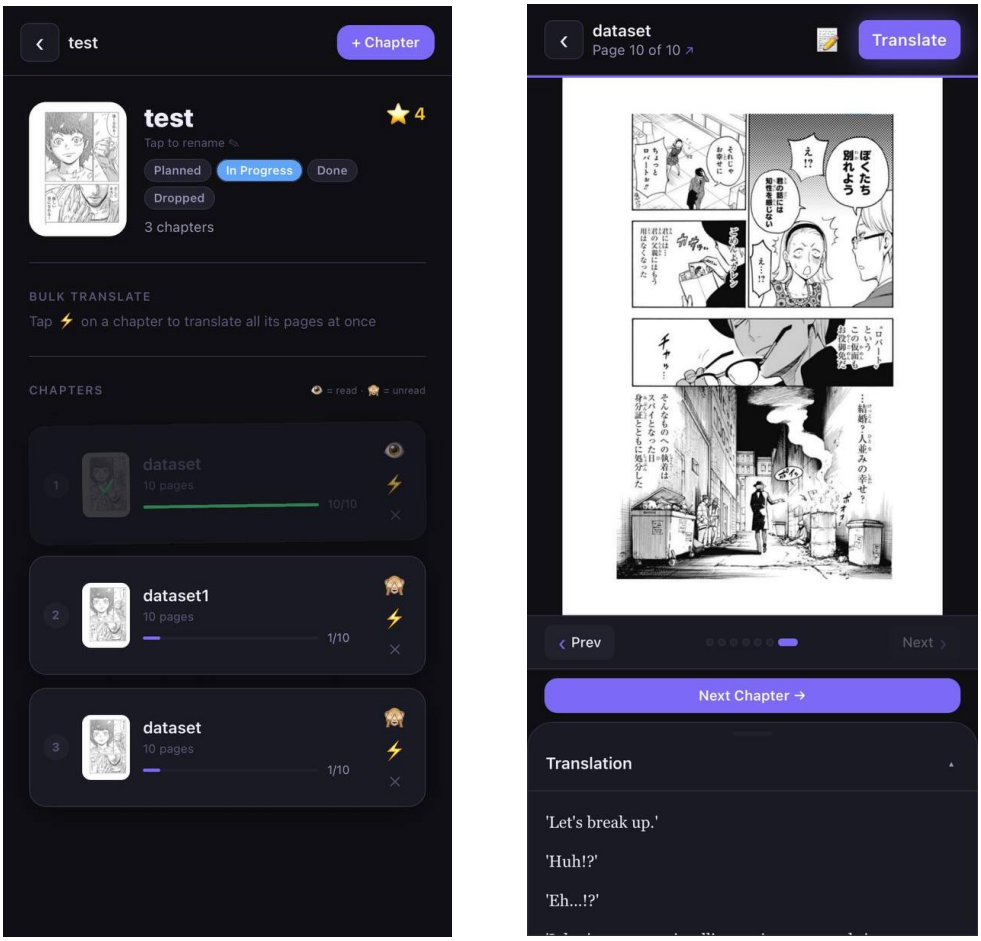


Рисунок 3.11 – Сторінка серії та глави всередені серії

3.2.6 Аналіз роботи мобільного застосунку

Для проведення аналізу роботи застосунку була створена форма опитування тестових користувачів, яким пропонувалось оцінити технічну роботу застосунку, загальні враження від використання та їх досвід з мангою, як з жанром літератури.

Результати показали загальну вдоволеність респондентів, однак оцінки швидкості свідчать про падіння швидкості під час реального використання через мобільний застосунок навідміну регульованого тестування API напряму. Детальні результати опитування представлені у таблиці 3.3.

Таблиця 3.3 – Результати опитування учасників бета-тестування мобільного застосунку

Критерій опитування	Результати опитування
Відсоток зацікавлених у манзі респондентів, %	54%
Найбільш популярний формат читання серед респондентів	Електронне перекладене видання
Відсоток респондентів, які користувалися схожими застосунками до опитування, %	47%
Оцінка перекладу від 1 до 5	4,65
Оцінка швидкодії роботи застосунку від 1 до 5	3,75
Оцінка зручності застосунку від 1 до 5	4,86
Оцінка закриття потреб користувча від 1 до 5	4,67
Оцінка загальних вражень під час використання застосунку від 1 до 5	4,81
Відсоток відмов від подальшого використання застосунку, %	5%

Подальший розвиток проєкту має бути спрямований на оптимізацію внутрішніх алгоритмів обробки потоків даних на серверній стороні з метою підвищення загальної чуйності інтерфейсу. Модернізація механізмів фонового виконання завдань та вдосконалення процедур передачі медіафайлів дозволять нівелювати наявні затримки. У підсумку, високі оцінки загального досвіду взаємодії створюють підґрунтя для подальшого розвитку системи, підтверджуючи доцільність обраного технологічного стеку та перспективність розробки в межах поставлених вимог.

Загалом, результати апробації підтверджують правильність вектору розробки та демонструють високий потенціал застосунку на ринку програмних рішень для автоматизації локалізації графічного контенту манги. Успішна інтеграція сучасних технологій штучного інтелекту дозволила створити продукт, який вирішує інженерні завдання та формує якісний досвід для кінцевого користувача.

ВИСНОВКИ

У рамках кваліфікаційної роботи було створено застосунок, що дозволяє розпізнавати текст зі сторінки з урахуванням контексту та особливостей жанру із використанням сучасних генеративних моделей.Основною метою роботи була розробка застосунку для автоматизованого перекладу сторінок манги англійською мовою з використанням OCR та великих мовних моделей, при цьому вирішено такі завдання:

– проведено аналіз підходів класичного OCR та сучасних мультимодальних моделей для детектування та розпізнавання тексту на складних фонах;

– проаналізовано можливості сучасних MLLMs щодо здатності одночасної візуальної інтерпретації та контекстуального перекладу японського тексту;

– сформовано власний набір даних, що містить сторінки манги різних жанрів з автентичними шрифтами, фуріганою та складним візуальним оформленням;

– виконано порівняльний аналіз двоетапного підходу на основі OCR та MLLM і одноетапного на основі MLLM з використанням метрики відстані Левенштейна для оцінки якості розпізнавання та методу експертних оцінок для аналізу якості перекладу; обрано підхід, що найкраще вирішує задачу перекладу манги;

– проведено експериментальний аналіз швидкодії обробки сторінок та економічної складової використання API різних моделей;

– розроблено алгоритм перекладу манги на основі обранного підходу;

– спроєктувано та реалізувано MVP вебзастосунку на базі Gradio, який автоматизує процес перекладу завантаженої сторінки манги за обраною методикою;

– спроєктувано та реалізувано мобільний застосунку за допомогою React Native, який автоматизує процес перекладу завантаженої сторінки, підтримує

збереження перекладу та повноцінне ведення особистої бібліотеки користувача.

Проведений аналіз двох підходів, а саме двоетапного підходу на основі OCR та MLLM і одноетапного на основі MLLM, показав, що одноетапний підхід дозволяє отримати кращу якість та швидкодію при роботі застосуку. Під час експерементального аналізу моделей Gemini 2.5 Flash, Gemini 3 Pro та ChatGPT 5.4, модель Gemini 2.5 Flash показала себе як найкраща для вирішення задачі перекладу. Вона продемонструвала найкращу якість, конкурентну швидкодію та оптимальну ціну. З огляду на це, у розробленому застосунку було використано саме її.

Розробка MVP застосунку надала можливість перевірити на практиці, що представлений у роботі підход до перекладу манги задовільняє користувацькі очікування у якості та швидкодії. На основі проведеного аналізу роботи MVP застосунку було розроблено технічне завдання та спроєктовано мобільний застосунок, який крім перекладу надає користувачеві функціональні можливості для повноцінного ведення особистої бібліотеки, зберігання перекладу та перегляду статистики

Результати кваліфікаційної роботи можуть бути корисними у сфері автоматизованої локалізації графічних видань, а також при подальшому створенні інтелектуальних систем перекладу контенту, де текст є невіддільним від зображення.У подальшому є сенс допрацювати систему, додавши до неї технологію пам’яті програми для покращення результатів розуміння контексту з аналізом декількох сторінок одночасно.

Результати роботи апробовано у вигляді 3 тез доповідей: під час Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті» [9], під час VI-ї Міжнародної науково-практичної конференції «Науковий вектор різноманітних сфер: реальність та майбутні тренди» [10], під час X Міжнародної науково-практичної конференції «Theoretical and Empirical Scientific Research: Concept and Trends» (Оксфорд, Велика Британія) [29].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гороховатський В., Творошенко І. (2026) Продуктивні моделі аналізу даних у методах розпізнавання зображень: монографія. Харків: ХНУРЕ, 153 с.

2. Yakovleva O., Matúšová S., Liubchenko V., Maksimov H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 21, no. 1, pp. 140-163.

3. Ковтуненко, А. Р., Яковлева, О. В., Любченко, В. А., & Янголенко, О. В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників. Вісник Національного технічного університету ХПІ (3). 24-31.

4. Cherednichenko, O., Yakovleva, O., & Hrechyshkin, D. (2025). *A data-centric approach to crowd counting: Synthetic dataset creation and evaluation*. In S. Vladov, Y. Bodyanskiy, V. Lytvyn, I. Aizenberg, & L. Chyrun (Eds.), *Proceedings of the International Workshop on Applied Intelligent Security Systems in Law Enforcement (AISSLE-2025)* CEUR Workshop Proceedings. pp. 22–34.

5. Yakovleva O., Matúšová S., Tvoroshenko I., Isaiev Y. (2024). Visitor counting based on video stream analysis from surveillance cameras. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 20, no. 1, pp. 67–87.

6. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. CEUR Workshop Proceedings Vol. 3403. pp. 69-86.

7. Yakovleva, O., Kovač, M., Ardasov, V. & Yeremenko, I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities. Public Administration and Regional Development, 19(1), pp. 158–183.

8. Yakovleva O., Nebeský L, Liakhov P. (2023). Research methods of texture image analysis to solve the texture search problem. Proceedings of the IV International Scientific and Practical Conference «The world of modern technologies and inventions». Vienna, Austria. 2023. pp. 252-261.

9. Остапчук В.О., Яковлева О.В. (2026) Порівняльний аналіз OCR та LLM підходів для перекладу тексту з зображення сторінки манґи. *30-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті». Збірник матеріалів форуму, 22–24 квітня 2026 року, Харків, Україна*, Т. 7, С. 127-129.

10. Yakovleva, O., & Ostapchuk, V. (2026). Comparative analysis of the quality and problem areas of current manga translation tools. In *Grail of Science* (pp. 1095–1097).

11. Mokuro. Mokuro.app. URL: https://reader.mokuro.app/ (дата звернення 13.02.2026).

12. Cotrans. Touhou.ai. URL: https://cotrans.touhou.ai/ (дата звернення 21.02.2026).

13. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2026) Feature space quantization and application of metrics in structural methods of image recognition, *IEEE Access*, vol. 14, pp. 17812-17824.

14. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.

15. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.

16. Gorokhovatskyi, O., & Yakovleva , O. (2024). medoids as a packing of ORB image descriptors. *Advanced Information Systems*, 8(2), 5–11.

17. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

18. Yakovleva, O., & Nikolaieva, K. (2020). Research Of Descriptor Based Image Normalization And Comparative Analysis Of SURF, SIFT, BRISK, ORB, KAZE, AKAZE Descriptors. *Advanced Information Systems*, 4(4), 89-101.

19. Billka AI Intelligent Receipt Scanning & Expense Tracking. Billka AI. URL: https://billka.sytoss.com/ (дата звернення 02.05.2026).

20. Billka AI Application For Bill Splitting And Expense Tracking, Simplified By Intelligence. SYTOSS. Software Development Company. URL: https://www.sytoss.com/billka-ai (дата звернення 02.05.2026).

21. Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model*, *17*(1), 57-61.

22. Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*.

23. Chatbot Arena - a Hugging Face Space by lmarena-ai. Hugging Face – The AI community building the future. URL: https://huggingface.co/spaces/lmarena-ai/chatbot-arena (дата звернення 26.02.2026).

24. *Gemini API Docs and Reference*. Google for Developers. URL: https://ai.google.dev/gemini-api/docs (дата звернення: 13.05.2026).

25. *Poppler*.Freedesktop Poppler. URL: https://poppler.freedesktop.org/ (дата звернення 04.04.2026).

26. Соколова А.М., Яковлева О.В. (2026) Проєктування мобільного Android-застосунку для розпізнавання та пояснення складу харчових продуктів з фокусом на українського користувача. 30-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті». Збірник матеріалів форуму, 22–24 квітня 2026 року, Харків, Україна, Т. 7, С. 156-158.

27. Datsok Y., Yakovleva O. (2026) Automated identification of medicinal products in mobile information systems based on multimodal models. Інформаційні технології: наука, техніка, технологія, освіта, здоров'я. Тези доповідей XXXIV міжнародної науково-практичної конференції MicroCAD-2026 (13–16 травня 2026 року). Харків: НТУ «ХПІ», С. 1692.

28. Huang, T. (2024). Fead: Figma-enhanced app design framework for improving UI/UX in educational app development. *arXiv preprint arXiv:2412.06793*.

29. Yakovleva O., Ostapchuk V., Datsok Y. (2026) Research on the effectiveness of multimodal large language models in the contextual manga translation *Theoretical and empirical scientific research: concept and trends: proceedings of the X international scientific and practical conference, May 8, 2026, Oxford, United Kingdom*, pp. 121-125.