Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ІНТЕЛЕКТУАЛЬНА СИСТЕМА СЕГМЕНТАЦІЇ ТА СТИЛІЗОВАНОГО ПЕРЕКЛАДУ ІЄРОГЛІФІЧНОГО ТЕКСТУ У МУЛЬТИМЕДІЙНИХ КОМІКСАХ

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки», 122

Керівник курсової роботи
ст. викладач Радзієвська О.В.
(прізвище та ініціали)
_____
(підпис)
"____" _____ 2025 р.

Виконав студент _____
__Негруб А. С._____
(прізвище та ініціали)
"____" _____ 2025 р.

Київ 2025

КАЛЕНДАРНИЙ ПЛАН КУРСОВОЇ РОБОТИ

Тема: Інтелектуальна система сегментації та стилізованого перекладу ієрогліфічного тексту у мультимедійних коміксах

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання	Примітка
1.	Затвердження теми дослідження та призначення керівника	жовтень 2024р.	
2.	Аналіз типології та лінгвостилістичних особливостей азійських коміксів	листопад 2024р.	
3.	Дослідження існуючих підходів до обробки текстових елементів	листопад 2024р.	
3.	Розробка концепції модульної архітектури системи	грудень 2024р	
4.	Проектування модуля попередньої обробки та сегментації зображень	грудень 2024р.	
5.	Формування та підготовка датасету для навчання детектора текстових блоків	грудень 2024р.	
6.	Навчання моделі детекції та класифікації текстових елементів	січень 2025р.	
7.	Розробка алгоритмів упорядкування та маскування перекриттів текстових блоків	січень 2025р.	
8.	Реалізація дуальної OCR-архітектури для розпізнавання ієрогліфічного тексту	січень 2025р.	
8.	Імплементація агента стилізованого перекладу з використанням LLM	лютий 2025р.	
10.	Інтеграція компонентів в єдину систему та тестування на реальних прикладах	лютий 2025р.	
11.	Проведення порівняльного аналізу з існуючими рішеннями	березень 2025р.	
12.	Написання пояснювальної записки до курсової роботи	квітень 2025р.	
13.	Оформлення роботи відповідно до вимог та передача науковому керівнику	травень 2025р.	
14.	Підготовка презентації та доповіді для захисту	травень 2025р.	
15.	Захист перед екзаменаційною комісією	Згідно з роботою ЕК	

Студент: Негруб А. С.

Керівник: ст. викладач Радзієвська О.В.

“_____” _____

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ПРИЙНЯТИХ СКОРОЧЕНЬ

API – Application Programming Interface, інтерфейс програмування додатків;

BLEU – Bilingual Evaluation Understudy, метрика оцінки якості перекладу;

CCL – Connected Component Labeling, алгоритм маркування зв'язних компонентів;

CER – Character Error Rate, коефіцієнт символьних помилок;

CNN – Convolutional Neural Network, згорткова нейронна мережа;

CRAFT – Character Region Awareness For Text Detection, алгоритм виявлення текстових областей;

CRNN – Convolutional Recurrent Neural Network, згорткова рекурентна нейронна мережа;

DETR – DEtection TRansformer, архітектура трансформера для детекції об'єктів;

F1-S – гармонійне середнє точності та повноти, міра якості класифікації;

GAN – Generative Adversarial Network, генеративна змагальна мережа;

GPU – Graphics Processing Unit, графічний процесор;

IoU – Intersection over Union, міра перекриття обмежувальних рамок;

LLM – Large Language Model, велика мовна модель;

LSTM – Long Short-Term Memory, довга короткочасна пам'ять (архітектура нейронної мережі);

mAP – mean Average Precision, середня точність (метрика оцінки детекторів);

OCR – Optical Character Recognition, оптичне розпізнавання символів;

RGB – Red, Green, Blue, червоний, зелений, синій (колірна модель);

SFX – Sound Effects, звукові ефекти;

SRCNN – Super-Resolution Convolutional Neural Network, згорткова нейронна мережа надвисокої роздільної здатності;

TPE – Tree-structured Parzen Estimator, деревоподібний оцінювач Парзена;

TrOCR – Transformer-based Optical Character Recognition, розпізнавання символів на основі трансформерів;

ViT – Vision Transformer, візуальний трансформер;

WER – Word Error Rate, коефіцієнт словесних помилок;

YOLO – You Only Look Once, одноходовий алгоритм детекції об'єктів.

АНОТАЦІЯ

У роботі представлено модульну систему автоматизованої сегментації, розпізнавання та стилізованого перекладу ієрогліфічного тексту в азійських коміксах. Ключова інновація полягає у використанні закономірностей візуального подання тексту для збереження його емоційного та семантичного навантаження при перекладі. Навколо цієї інновації побудовано конвеєрну архітектуру з класифікацією текстових елементів за 13 категоріями, що вирішує проблему ручного опрацювання графічного контенту.

Розроблений алгоритм візуальної класифікації автоматично інтегрує результати в процес перекладу, зберігаючи стилістичні особливості оригіналу. Технічна реалізація включає модуль YOLOv12 для детекції блоків (точність mAP50 86%), дуальну TrOCR-модель для розпізнавання ієрогліфів (помилка CER 4.8%) та механізм контекстно-адаптивного перекладу на основі LLM. Експериментальне порівняння з існуючими рішеннями продемонструвало високі показники якості перекладу (0.93) та стилістичної відповідності (0.89), а також підвищення BLEU-score до 0.92.

Ключові слова: комп'ютерний зір, оптичне розпізнавання символів, стилізований переклад, манхва, OCR, YOLO, TrOCR, мультимодальні системи.

ВСТУП

Стрімке зростання популярності азійських коміксів (манга, манхва, манхуа) у світовому медіапросторі створило значний попит на їхню швидку та якісну локалізацію багатьма мовами. Утім, традиційні методи перекладу таких видань залишаються здебільшого ручними, а відтак ресурсомісткими. Це суттєво гальмує процес адаптації контенту для міжнародної аудиторії, в тому числі й для українського читача.

Наявні системи оптичного розпізнавання символів (OCR) демонструють обмежену ефективність при роботі з ієрогліфічним текстом на неоднорідному тлі коміксів, а також через їхнє нелінійне представлення. Натомість сучасні досягнення в галузі комп'ютерного зору та штучного інтелекту дають змогу розробити комплексну систему, яка не лише розпізнає текст, а й інтерпретує його контекст, забезпечуючи якісний стилізований переклад.

Особливість азійських коміксів полягає у складній графічно-текстовій оповіді, де зміст та емоційний стан візуально передаються за допомогою шрифтів, форми та кольору текстових блоків. Ця риса створює можливість для розробки спеціалізованих комп'ютерних рішень: візуальні характеристики слугуватимуть основою для автоматичної класифікації та адаптивного перекладу, що точно відтворить стилістичні нюанси оригіналу.

За мету даної роботи поставлено створення інноваційної модульної системи, що поєднує процеси сегментації зображень, розпізнавання ієрогліфічного тексту та його контекстно-залежного перекладу зі збереженням оригінальної стилістики. Робота також включає програмну реалізацію розробленої архітектури та критичний аналіз її ефективності порівняно з існуючими галузевими рішеннями.

Для досягнення поставленої мети необхідно вирішити такі ***наукові завдання***:

1. Провести критичний огляд відомих методів класифікації, оптичного розпізнавання а також перекладу, зокрема:
   - на основі традиційних алгоритмів комп'ютерного зору;
   - за допомогою гібридних підходів.
   - на основі хмарних сервісів та LLM моделей;
2. Проаналізувати структурні та стилістичні особливості текстових елементів у азійських коміксах для визначення ключових параметрів класифікації.
3. Розробити алгоритм сегментації зображень з урахуванням їх візуальних характеристик.
4. Створити та оптимізувати модель детекції та класифікації для виділення текстових блоків з урахуванням їхніх візуальних характеристик.
5. Створити та оптимізувати модель розпізнавання ієрогліфічного тексту, адаптовану до художніх шрифтів та різноманітних фонів.
6. Розробити підсистему контекстного перекладу, що враховує тип текстового блоку та відтворює відповідний стиль у цільовій мові.
7. Об’єднати компоненти в єдину систему та оцінити її ефективність на реальних прикладах коміксів.

У цій роботі вперше запропоновано одноходовий алгоритм чіткої візуальної класифікації текстових балонів, що автоматично категоризує їх та безпосередньо інтегрує результати в процес перекладу, що дозволяє зберегти емоційне та семантичне навантаження оригіналу. Розроблено архітектуру каскадної системи, яка забезпечує повний цикл обробки, від сегментації зображення до безпосереднього перекладу тексту.

Робота складається з чотирьох розділів.

Перший розділ присвячено детальному огляду галузі, аналізу існуючих методів та систем для перекладу графічного контенту, виявленню їхніх обмежень та обґрунтуванню необхідності розробки нового підходу.

В другому розділі наведено теоретичне обґрунтування розробленого методу класифікації та стилізованого перекладу, описано архітектуру модульної системи та алгоритми взаємодії її компонентів.

Третій розділ присвячено практичній реалізації запропонованої системи, включаючи розробку та навчання нейромережевих моделей для розпізнавання та перекладу тексту.

Четвертий розділ містить результати експериментального дослідження розробленої системи, порівняльний аналіз з існуючими рішеннями та оцінку ефективності за кількісними та якісними показниками.

Окрім наукової цінності, розроблена система має практичне значення для видавництв і перекладацьких спільнот, адже значно прискорює процес локалізації азійських вебтунів, зокрема й українською мовою. Технологія може бути адаптована для автоматизації перекладу інших форм неструктурованого графічного тексту, таких як реклама, інфографіка та навчальні матеріали. Додатковий потенціал системи полягає у можливості подальшого розширення контенту, як от створення аудіоверсій коміксів для людей з вадами зору, де стиль озвучення відповідатиме типу текстового блоку, що забезпечить більш комфортне його сприйняття.

РОЗДІЛ 1: Аналіз існуючих методів та систем перекладу графічного контенту

1.1. Типологія та лінгвостилістичні особливості азійських коміксів

Азійські комікси як явище масової культури пройшли довгий шлях від місцевого видавництва до становлення як глобального культурного феномену. Починаючи з 1950-х років, японська манга заклала основи унікального візуального образу, який згодом перейняли та розвинули інші країни регіону. За останнє десятиліття їхня популярність зростала в геометричній прогресії: якщо у 2010 році світовий ринок азійських коміксів оцінювався приблизно в 4 мільярди доларів, то до 2024 року ця цифра зросла до понад 16,8 мільярдів доларів (за даними Fortune Business Insights [1]) із прогнозованим щорічним зростанням на рівні 6-8%. Особливо стрімке зростання спостерігається у секторі цифрових веб-мультфільмів - коміксів, оптимізованих для читання онлайн на мобільних пристроях. Лише платформа Webtoon, один з провідних дистриб'юторів, повідомляє про понад 166 мільйонів активних користувачів щомісяця [2].

Світ східних коміксів представлений трьома основними форматами, кожен з яких має свої культурні та стилістичні особливості:

Манга (Японія, *(а)* - *рисунок 1.1.*) - традиційно чорно-біла, читається справа наліво. Вона характеризується найвиразнішим стилем виконання, динамічними лініями та емоційною виразністю кадрів. Візуально відрізняється специфічним поділом сторінки на панелі неправильної форми та частим використанням візерунків для передачі деталей.

Манхва (Корея, *(б)* - *рисунок 1.1.*) - часто кольорова, у форматі вебтуну (вертикальна прокрутка, розроблена для мобільних пристроїв), читається зліва направо. Має більш наближений до реальності стиль малювання порівняно з мангою, менше стилізованих елементів та м'якші лінії. Особливістю корейських

вебтунів є їхнє використання новітніх ефектів для оформлення sfx-ряду та самих панелей.

Манхуа (Китай, *(в) - рисунок 1.1.*) - може бути як чорно-білою, так і кольоровою, традиційні видання читаються справа наліво, а сучасні вебтуни - шляхом вертикальної прокрутки. Відрізняється поєднанням елементів традиційного китайського мистецтва з сучасними стилями, яскравим колоритом та специфічним, простішим художнім стилем.

Рисунок 1.1 – Приклади типологічних особливостей коміксів (а – манга; б – манхва; в - маньхуа)

Окремою особливістю також є робота з ономатопеями (звуконаслідувальними словами) та звуковими ефектами (SFX), які в азійських коміксах представлені надзвичайно широко. На відміну від західної культури, яка використовує відносно обмежений набір звукописів, азійські мови мають сотні специфічних ономатопей, кожен з яких передає унікальний відтінок звуку чи відчуття. Наприклад, японська мова має близько 1200 звукових слів, серед яких такі спеціальні, як «докі-докі» (серцебиття) чи «сара-сара» (шелест сторінок). Такі представлення звуку часто включені безпосередньо в художній твір, з використанням специфічних рукописних шрифтів, кольорів і форм для посилення їхнього емоційного впливу.

Крім того, стилізація тексту в азійських коміксах є не лише декоративним елементом, а й важливим засобом вираження наративу. Розмір, форма, колір і

розташування тексту передають додаткову інформацію про обсяг, емоційний стан, інтонацію і навіть походження персонажа. Наприклад, мова іноземця часто подається іншим кольором тексту, думки персонажа зображуються у вигляді хвилястих ліній, а крик нерідко виділяється збільшеним жирним шрифтом, який «вививає́ться» за межі діалогового балону.

Усі ці фактори створюють унікальний набір викликів для автоматизованого перекладу азіатських коміксів, що вимагає не просто лінгвістичної трансформації, а повноцінної крос-культурної адаптації зі збереженням візуального, емоційного та смислового аспектів оригіналу. З огляду на постійне збільшення попиту на цей контент, розробка ефективних методів локалізації набуває особливої цінності та практичного змісту.

1.2. Еволюція методів обробки текстових елементів у графічному контенті

Історично процес локалізації коміксів був майже повністю ручним і включав кілька трудомістких етапів: оцифрування оригінальних сторінок, ретушування оригінального тексту, переклад та верстку перекладеного тексту. Такий підхід вимагав багато часу та людських ресурсів, що значно обмежувало можливості масштабування процесу для задоволення висхідного попиту.

З розвитком цифрових технологій ручні методи поступово доповнювалися спеціалізованим програмним забезпеченням для редагування зображень, яке, втім, все ще вимагало значного втручання людини. Перші спроби автоматизації цього процесу були пов'язані з впровадженням систем оптичного розпізнавання символів (OCR), які розроблялися в першу чергу для роботи з друкованим текстом на білому тлі. Однак специфіка коміксів, де текст часто інтегрований у складні візуальні композиції, має нестандартні шрифти і розміщується на неоднорідному фоні, суттєво знижувала ефективність таких систем.

Наступним етапом еволюції стало впровадження експертних рішень для роботи з коміксами, таких як Manga Studio [3], що дозволяли напівавтоматичне виділення текстових блоків та спрощену верстку перекладеного тексту. Проте

вони все ще вимагали значного ручного втручання, особливо при роботі з нестандартними шрифтами та ефектами.

Методи глибокого навчання (2015-2020 роки) спричинили справжній прорив. З'явились перші автоматизовані системи, які виявляють, розпізнають і перекладають текст у коміксах. Сучасний етап розвитку (2020-2025) характеризується впровадженням мультимодальних моделей та великих мовних моделей (LLM), що здатні врахувати візуальний контекст при інтерпретації та перекладі тексту, підвищуючи якість локалізації до рівня, наближеного до професійного людського перекладу.

Не зважаючи на значний прогрес, повністю автоматизовані системи для локалізації азійських коміксів все ще знаходяться на стадії активного розвитку, і потребують вдосконалення в частині адаптивної обробки різноманітних стилів тексту, що є одним із головних викликів для сучасних дослідників у цій галузі.

1.3. Аналіз наявних підходів до обробки текстових елементів у коміксах

Підходи до вирішення цієї задачі можна умовно розділити на три основні категорії: традиційні методи комп'ютерного зору, сегментаційні підходи на базі глибокого навчання та спеціалізовані системи для розпізнавання ієрогліфічного тексту. Кожна з цих категорій має власні переваги та обмеження, які необхідно враховувати при розробці комплексного рішення.

1.3.1. Традиційні методи комп'ютерного зору для сегментації текстових блоків

Традиційні методи комп'ютерного зору базуються на послідовному застосуванні алгоритмів обробки зображень для виділення структурних елементів коміксів. Загальний підхід включає етапи попередньої обробки (фільтрація, нормалізація, бінаризація), виділення контурів, сегментації та класифікації.

У роботі «Method of Real-Time Text Extraction from Digital Manga Comic Image» [4] запропоновано конвеєр первинної обробки манги, що включає послідовні етапи екстракції фреймів, виділення діалогових балонів та текстових блоків з подальшим розпізнаванням тексту. Ключовою перевагою цього підходу є використання морфологічних операцій для аналізу структури зображення. Зокрема, алгоритм застосовує операції розширення (dilation) та ерозії (erosion) для виділення замкнутих областей, які з високою ймовірністю є діалоговими балонами. Основною перевагою такого підходу є обчислювальна ефективність, що дозволяє обробляти зображення в режимі реального часу навіть на пристроях з обмеженими ресурсами. Проте суттєвим обмеженням є орієнтація виключно на чорно-білі манги з базовою структурою викладення тексту, що значно звужує сферу застосування методу.

Схожий підхід описаний у дослідженні «Text Extraction and Recognition Using Median Filter» [5], де для виділення текстових блоків застосовується Connected Component Labeling (CCL). Алгоритм використовується для групування пікселів у потенційні текстові блоки на основі зв'язності компонентів. Метод продемонстрував високу ефективність для структурованих зображень з чіткими межами між текстом та фоном, однак виявився вразливим до шумів та неефективним при обробці зображень великих розмірів через значне обчислювальне навантаження.

Аналіз цих та інших традиційних підходів виявляє їхню фундаментальну обмеженість: вони ефективні лише в контрольованих умовах із передбачуваною структурою зображення. Сучасні кольорові манхва та манхуа, з їхнім багатим візуальним оформленням та різноманітними стилями подачі тексту, створюють складності, які традиційні методи не здатні ефективно вирішити.

1.3.2. Методи глибокого навчання для аналізу структури коміксів

Методи глибокого навчання значно розширили можливості автоматизованого аналізу коміксів, адже вони пропонують вищу точність і

адаптивність, ніж традиційні підходи. Їх здатність вивчати складні візуальні патерни безпосередньо з даних робить їх особливо цінними для обробки мультистильових ілюстрацій.

Особливо показовим є дослідження «Extraction of Semantic Content and Styles in Comic Books» [6], де для виділення панелей, персонажів та діалогових балонів використовуються архітектури YOLOv3 та Mask-RCNN. Перша застосовується для виявлення панелей та персонажів на основі обмежувальних рамок, тоді як Mask-RCNN забезпечує піксельну сегментацію діалогових балонів, що дозволяє точно виділяти межі об'єктів. Ключовою інновацією цього дослідження є інтеграція семантичної інформації з візуальним стилем, що дозволяє автоматично визначити логічні зв'язки між панелями та напрямок читання (для різних видів коміксів він відрізняється). Точність виявлення панелей в цьому досягла 89%, а приналежності діалогових балонів — 93%, що суттєво перевищує показники традиційних методів. Попри це, відсоток виявлення балонів виявився доволі низьким через малу диверсифікацію датасету, а практичне розпізнавання тексту не відповіло зазначеним вимогам завдяки використанню Tesseract OCR у якості розпізнавача тексту.

Важливим кроком у напрямку створення комплексних систем аналізу коміксів є робота «MaRU: A Manga Retrieval and Understanding System Connecting Vision and Language» [7]. Ця мультимодальна система поєднує методи комп'ютерного бачення для аналізу візуального контенту з методами обробки природної мови для інтерпретації текстових елементів. Вона використовує архітектуру DETR (DEtection TRansformer) для виявлення структурних елементів та інтегрує їх з текстовими даними через моделі SentenceBERT та GPT-4. Система продемонструвала високу ефективність при обробці мультимовних запитів, що є критичним для міжнародної локалізації.

Попри значний прогрес, методи глибокого навчання все ще стикаються з низкою обмежень при роботі з коміксами. По-перше, вони вимагають значних обчислювальних ресурсів, що ускладнює їх використання в системах реального часу. По-друге, навчання ефективних моделей вимагає великих анотованих

наборів даних, які складно сформувати для таких специфічних доменів, як азійська художня література. По-третє, існує проблема деталізації контексту. Без певної алгоритмічної класифікації, вивід такої моделі буде непослідовним через значний вплив мовних моделей на аналіз зв’язків.

1.3.3. Системи оптичного розпізнавання ієрогліфічного тексту

Розпізнавання ієрогліфічного тексту представляє особливу складність через велику кількість унікальних символів (від кількох тисяч до десятків тисяч) та їхню структурну складність. Традиційні OCR-системи, такі як Tesseract, демонструють обмежену ефективність при роботі з азійськими мовами, що стимулювало розробку спеціалізованих підходів.

Дослідження «Optical Character Recognition on English Comic Digital Data for Automated Language Translation» [8], хоч і фокусується на англомовних коміксах, пропонує підхід, що потенційно адаптивний до ієрогліфічного тексту. Метод базується на аналізі характерних точок символів (бренчпоінтів та ендпоінтів) - точок розгалуження та кінцевих точок ліній, що формують символ. Такий підхід забезпечує стійкість до варіацій шрифтів та масштабу, що критично для коміксів, де текст часто стилізований. Проте для адаптації до ієрогліфічних систем письма необхідно суттєво розширити бази еталонних форм, що ускладнює практичну реалізацію.

Більш перспективним для розпізнавання ієрогліфічного тексту є підхід, описаний у роботі «Segmentation-free speech text recognition for comic books» [9], де пропонується відмова від попередньої сегментації на рівні символів. Замість цього модель навчається розпізнавати цілі фрази безпосередньо з зображення, використовуючи архітектуру на базі згорткових та рекурентних нейронних мереж. Ключовим нововведенням є автоматичне налаштування під вхідний датасет з використанням лексичних мір для оцінки якості розпізнавання. Такий підхід демонструє непогану ефективність навіть для стилізованих шрифтів, перевершивши претреновану модель Tesseract для японських текстів у манзі.

Метод контекстної відповідності форм, описаний у дослідженні «Text Extraction Using Shape Context Matching» [10] пропонує альтернативний підхід, особливо ефективний для звукових ефектів та ономатопей, які часто мають найбільш нестандартне візуальне представлення. Алгоритм базується на порівнянні форм символів з еталонними зразками за допомогою дескрипторів контексту форми (shape context descriptors), які забезпечують гнучкість до деформації та стильових варіацій. Однак цей метод вимагає наявності чітких контурів символів і демонструє знижену ефективність при роботі з текстом, вбудованим у важкі візуальні структури.

1.3.4. Сучасні системи машинного перекладу для коміксів

Переклад текстових елементів у коміксах виходить далеко за межі простої лінгвістичної трансформації. Він включає збереження стилістичних, емоційних та контекстуальних аспектів, що особливо важливо для коміксів, які є невід'ємною частиною наративу. Сучасні підходи до автоматизованого перекладу коміксів можна розділити на дві основні категорії: системи на базі класичних моделей машинного перекладу та рішення, що використовують великі мовні моделі (LLM).

Проєкт «ComiTranslate: Epowering Global Readership through AutoTranslated Comics and Manga» [11] представляє комплексний підхід до автоматизації перекладу коміксів з використанням сучасних бібліотек для обробки зображень (Pillow, OpenCV), розпізнавання тексту (Tesseract, Keras-OCR) та перекладу (DeepL API). Система реалізує повний цикл обробки: від виділення текстових блоків до вставки перекладеного тексту назад у зображення з адаптацією розміру та стилю шрифту. Основною перевагою є модульна архітектура, яка дозволяє гнучко замінювати окремі компоненти без перебудови всієї системи. Однак реалізація має суттєві обмеження через використання хоч і якісних, але все ж таки не адаптованих до конкретних завдань моделей.

Інший значущий напрямок розвитку автоматизованого перекладу коміксів пов'язаний з використанням великих мовних моделей (LLM) та мультимодальних підходів. Показовою є робота «Multimodal Transformer for Comics Text-Cloze» [12], що пропонує архітектуру мультимодальної великої мовної моделі (Multimodal-LLM), спеціально розробленої для розуміння контексту в коміксах. Автори вводять поняття задачі "text-cloze" – заповнення пропусків у текстових блоках на основі сусідніх панелей, що вимагає глибокого розуміння як візуального, так і текстового контексту. Ключовим елементом цього підходу є доменно-адаптований візуальний енкодер на базі ResNet-50, попередньо навчений на коміксах у режимі самоконтрольованого навчання з використанням SimCLR. Модель демонструє покращення на 10% порівняно з попередніми підходами, при цьому використовуючи лише п'яту частину параметрів складніших моделей. Однак цей підхід орієнтований насамперед на розуміння контексту, а не на повний цикл перекладу.

Комплексним рішенням, що безпосередньо використовує потенціал LLM для перекладу, є проєкт «ogkalu/Comic-Translate» [13]. Ця відкрита система використовує двоетапний підхід на основі моделей YOLOv8m: одна відповідає за детекцію мовних балонів, інша – за сегментацію тексту. Для оптичного розпізнавання символів система використовує спеціалізовані рішення для кожної мови: doctr для латинських алфавітів, manga-ocr для японської, Pororo для корейської та PaddleOCR для китайської.. Для перекладу використовуються сучасні LLM, включаючи GPT-4o, Claude-3-Opus та Gemini-1.5-Pro, які, для збереження контексту, отримують текст сторінки разом з самим зображенням. Проте навіть ця передова система має суттєві обмеження: детектори балонів і тексту недостатньо варіативні через обмеженість навчальних датасетів, а стилізація перекладу залишається нестабільною через цілковите покладання на LLM для вирішення цього питання без чіткої класифікації. Тим не менш, цей проєкт демонструє перспективність інтеграції сучасних нейромережевих архітектур для вирішення комплексної задачі перекладу коміксів.

Аналіз цих підходів на базі LLM показує значний потенціал для підвищення якості автоматизованого перекладу коміксів. Проте важливим аспектом, що часто нехтується в існуючих системах [7, 11, 13], є класифікація текстових блоків за їхнім функціональним призначенням (діалоги, думки, звукові ефекти, системні повідомлення тощо) та адаптація роботи алгоритмів відповідно до цієї класифікації. На цьому аспекті і буде зосереджене подальше наукове дослідження.

1.4. Обґрунтування необхідності розробки нового підходу

Аналіз існуючих методів та систем для перекладу графічного контенту виявляє низку суттєвих обмежень, що перешкоджають створенню повністю автоматизованої системи локалізації азійських коміксів:

Насамперед, спостерігається ***фрагментарність існуючих рішень*** - більшість методів фокусуються на окремих аспектах процесу (сегментація, розпізнавання або переклад) без забезпечення інтеграції в єдину цілісну систему. Це призводить до неоптимального використання ресурсів та потенційної втрати інформації при передачі даних між різними компонентами.

Другим суттєвим обмеженням є ***недостатня адаптивність існуючих систем***. Вони часто демонструють високу ефективність лише для специфічних типів коміксів (наприклад, чорно-білої манги) і не здатні пристосовуватися до різноманіття стилів та форматів. Це особливо проблематично в контексті сучасного ринку, де популярність набувають різні форми азійських коміксів, кожна з власними візуальними та структурними особливостями.

Третє обмеження пов'язане з ***недостатнім врахуванням стилістичних особливостей*** текстових елементів. Існуючі системи рідко класифікують текстові блоки за їхніми візуальними та функціональними характеристиками, що призводить до втрати стилістичних нюансів при перекладі. В контексті коміксів, де візуальне представлення тексту є невід'ємною частиною наративу, ця втрата може суттєво знижувати якість сприйняття перекладеного матеріалу.

Четвертим обмеженням є ***високі обчислювальні вимоги сучасних методів***, особливо тих, що базуються на глибокому навчанні. Хоча ці методи забезпечують високу точність, їхня обчислювальна складність часто ускладнює практичне застосування, особливо в контексті обробки великих обсягів контенту або роботи в режимі реального часу.

Нарешті, особливо проблематичним аспектом є ***обмежена ефективність існуючих систем розпізнавання тексту*** при роботі з ієрогліфічним текстом. Традиційні OCR-системи демонструють знижену точність при розпізнаванні ієрогліфів, особливо в контексті стилізованих шрифтів та неоднорідного фону, що типові для коміксів.

Враховуючи ці обмеження, очевидною є необхідність розробки нового підходу, що забезпечить:

- Інтеграцію всіх етапів обробки, від сегментації зображення до фінального перекладу в єдину цілісну систему.
- Автоматичне визначення типу текстового блоку на основі певних візуальних характеристик середовища для адаптації стилю перекладу.
- Пошук балансу між точністю та ефективністю для забезпечення практичної застосовності системи.
- Розробку або адаптацію методів розпізнавання тексту, що враховують особливості ієрогліфічних систем письма.

1.5. Висновки до розділу 1

У цьому розділі було проведено комплексний аналіз існуючих методів та систем для перекладу графічного контенту, зокрема азійських коміксів. Розглянуто еволюцію підходів від ручних методів до сучасних автоматизованих систем, проаналізовано традиційні методи комп'ютерного зору, підходи на базі

глибокого навчання та спеціалізовані системи для розпізнавання ієрогліфічного тексту.

Виявлено, що попри значний прогрес у окремих аспектах обробки графічного тексту, існуючі підходи мають суттєві обмеження щодо інтеграції всіх етапів у єдину систему, врахування стилістичних особливостей текстових елементів, обчислювальної ефективності та роботи з ієрогліфічним текстом.

Обґрунтовано необхідність розробки нового підходу, що забезпечить комплексне вирішення задачі автоматизованої локалізації азійських коміксів з урахуванням їхніх структурних та стилістичних особливостей. Такий підхід має інтегрувати сучасні досягнення в галузі комп'ютерного зору, розпізнавання образів та машинного перекладу в єдиний конвеєр обробки, здатний здійснювати повний цикл від сегментації зображення до стилізованого перекладу текстових елементів.

У наступному розділі буде представлено теоретичне обґрунтування розробленого методу класифікації та стилізованого перекладу, а також описано архітектуру запропонованої інтегрованої модульної системи.

РОЗДІЛ 2: Теоретичне обґрунтування методу класифікації та стилізованого перекладу

2.1. Загальна архітектура системи

Розробка ефективного методу автоматичного перекладу азійських коміксів вимагає комплексного підходу, що враховує особливості візуального представлення тексту, специфіку ієрогліфічної писемності та контекстуальні нюанси оповіді. На основі аналізу обмежень існуючих підходів, представлених у першому розділі, гарним рішенням для забезпечення гнучкості системи є використання модульної архітектури, яка б дозволила заміняти окремі компоненти, підлаштовуючи систему до специфічних задач. Архітектура такої системи наведена *на рисунку 2.1.*

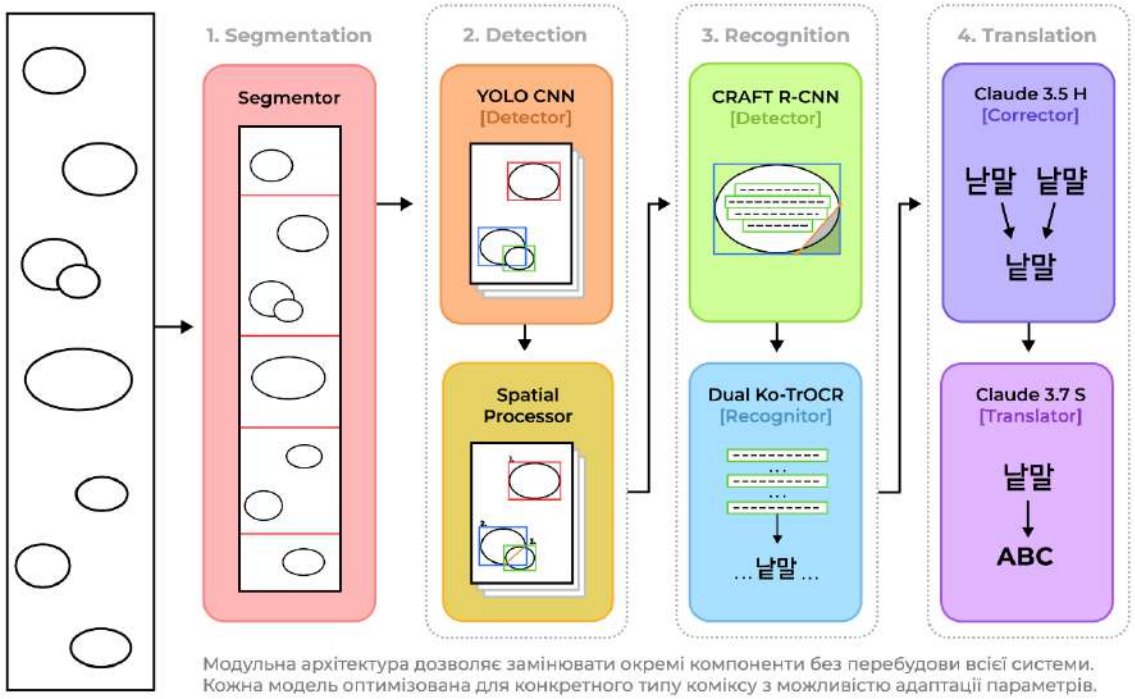

Рисунок 2.1 – Загальна архітектура системи для сегментації та стилізованого перекладу ієрогліфічного тексту

Основними компонентами архітектури є:

- Модуль попередньої обробки зображень — стандартизує та сегментує вхідні зображення.
- Агент детекції та класифікації текстових блоків — виявляє та класифікує текстові блоки за їхніми візуальними характеристиками.
- Конвеєр просторової обробки — упорядковує та обробляє виявлені текстові блоки відповідно до логіки оповіді.
- Агент детекції текстових строк — виявляє окремі рядки тексту для подальшого розпізнавання.
- Розпізнавач ієрогліфічного тексту — перетворює графічне представлення тексту у форму, доступну для машинної обробки.
- Коректор розпізнаваного тексту — об’єднує, лексично аналізує та покращує результати знаходження тексту.
- Агент стилізованого перекладу — здійснює переклад з урахуванням стилістичних особливостей тексту.

Така структура забезпечує ефективну обробку даних на кожному етапі з мінімальною втратою інформації між компонентами. Розглянемо детально теоретичні засади функціонування кожного з компонентів.

2.2. Модуль попередньої обробки та сегментації зображень

Специфікою вертикальних коміксів (манхва) є їхня надзвичайна довжина - одна глава може бути представлена єдиним зображенням розміром близько 800×100 000 пікселів. Обробка таких зображень вимагає спеціалізованого підходу до сегментації.

Традиційні методи сегментації зображень, такі як рівномірне розбиття або кластеризація, не враховують змістову структуру коміксів, що призводить до дроблення текстових блоків і втрати контексту. Для розв'язання цієї проблеми

було використано алгоритм адаптивної сегментації, що базується на аналізі горизонтальних паттернів пікселів.

За основу розробленого методу взято концепцію "вирізання швів" (seam carving), запропонована Avidan та Shamir [14], проте адаптовану до структурних особливостей коміксів. Алгоритм використовує метрику однорідності кольору для визначення оптимальних точок поділу зображення:

$$C(y) = \frac{1}{W}\sum_{x=0}^{W-1}[|I(x,y) - I(x+1,y)| < \delta] \cdot$$
$$\left[\max_{0 \le x < W-1} |I(x,y) - I(x+1,y)| < \lambda\right]$$

де $C(y)$ – коефіцієнт кольорової однорідності рядка y;

W – ширина зображення;

$I(x,y)$ – значення кольору пікселя в позиції (x,y);

δ – пороговий параметр для визначення схожості кольорів (зазвичай 10.0);

λ – пороговий параметр для виявлення різких переходів (зазвичай 50.0);

[...] – функція Айверсона (1, якщо умова істинна; 0, якщо хибна);

Перший компонент формули визначає долю пікселів із малою різницею кольорів, забезпечуючи консистентність по всьому рядку. Другий компонент є бінарним індикатором, який гарантує, що рядок не має екстремальних колірних переходів, запобігаючи вибору рядків з різкими змінами кольору.

Рядок вважається потенційним кандидатом для розділення, якщо його коефіцієнт кольорової однорідності перевищує заданий поріг γ (у нашому випадку $\gamma = 0.9$, тобто 90% пікселів у рядку мають незначну відмінність у кольорі від сусідніх пікселів).

Для оптимізації обчислювальної ефективності, замість пошуку локальних максимумів у межах вікна, реалізовано прогресивний підхід із фіксованим кроком. Після знаходження валідної точки розділення алгоритм просувається вперед на мінімальну висоту розділу. Якщо поточна точка не є валідною точкою розділення, алгоритм просувається на менший крок і продовжує пошук:

$$y_{i+1} = y_i + \begin{cases} min\_height, \text{ якщо } S(y\_i) = 1 \\ carving\_step, \text{ якщо } S(y\_i) = 0 \end{cases}$$

Такий прогресивний підхід значно зменшує обчислювальну складність при збереженні високої якості сегментації, що дозволяє ефективно обробляти надзвичайно довгі зображення манхви, зберігаючи природні межі між панелями (*рисунок 2.2*).

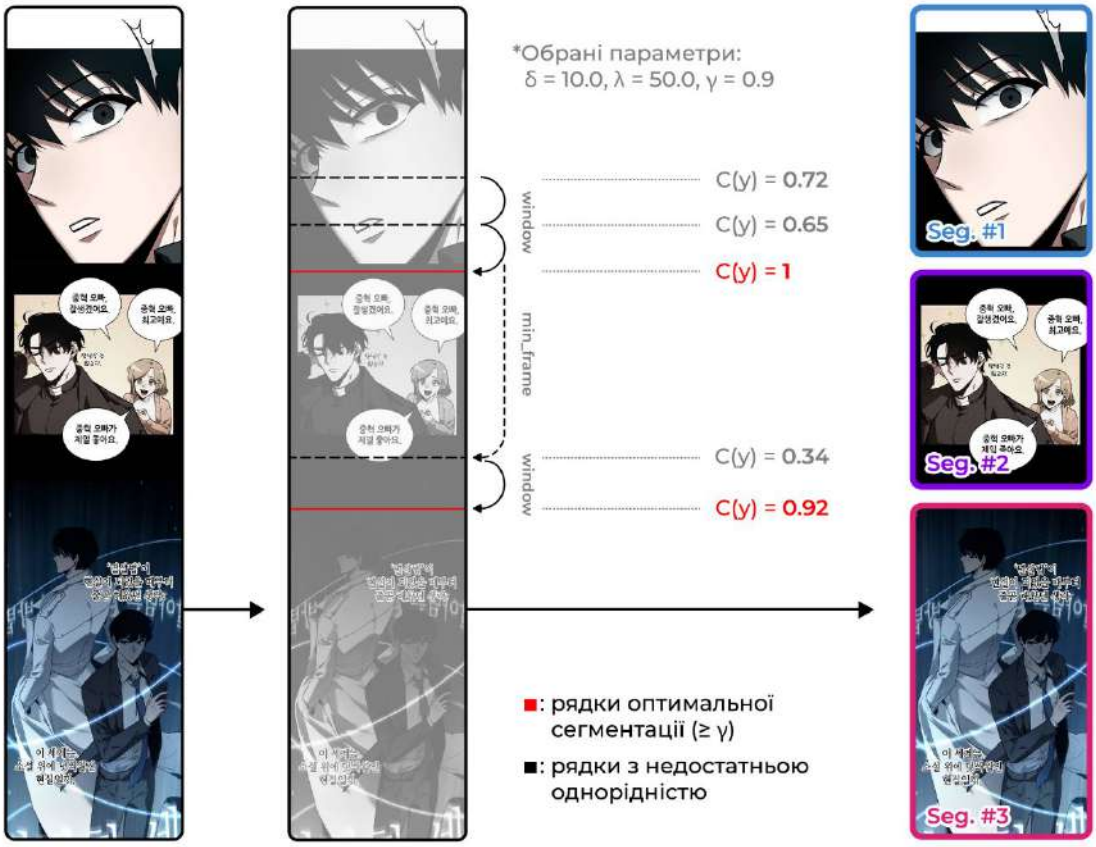

Рисунок 2.2 – Візуалізація процесу адаптивної сегментації довгої манхви

Для адаптації зображень до стандартних розмірів, необхідних для подальшої обробки нейронними мережами, використовується алгоритм підганяння розмірів на основі нейронної моделі Waifu2x [15], що базується на методі SRCNN, запропонованому Dong et al. [16]. Ця модель, розроблена для високоякісного масштабування аніме-зображень, особливо ефективна для обробки манхви завдяки використанню глибоких згорткових нейронних мереж, оптимізованих для збереження чітких контурів та текстур, характерних для коміксів. Крім того, вона дозволяє не лише масштабувати зображення без втрати деталей, але й одночасно знижувати рівень шуму, що значно покращує якість подальшого розпізнавання тексту.

2.3. Агент детекції та класифікації текстових блоків

Аналіз існуючих рішень показав, що попередні системи для роботи з текстом у коміксах [4, 6, 7], використовують двоетапний алгоритм: спочатку всі потенційні текстові блоки виявляються за допомогою сегментації або виділення контурів, а потім окремо уточнюються за допомогою додаткових моделей чи методів. Такий підхід призводить до накопичення помилок між етапами.

У цій роботі реалізовано ефективніший однопрохідний підхід на основі архітектури YOLO (You Only Look Once), який за один прохід нейронної мережі визначає як розташування текстових блоків, так і їх належність до конкретного класу.

Архітектура YOLOv12 є розвитком ідей, закладених у роботі Redmon et al. [17]. Вибір цієї архітектури обґрунтований результатами порівняльного аналізу на наборі COCO, представленими *на рисунку 2.3*, де видно, що моделі сімейства YOLO11 та новіші демонструють найкраще співвідношення точності (mAP50-95) та швидкодії серед сучасних детекторів об'єктів. Окремі бенчмарки [18, 19] підтверджують цю перевагу на різних наборах даних.

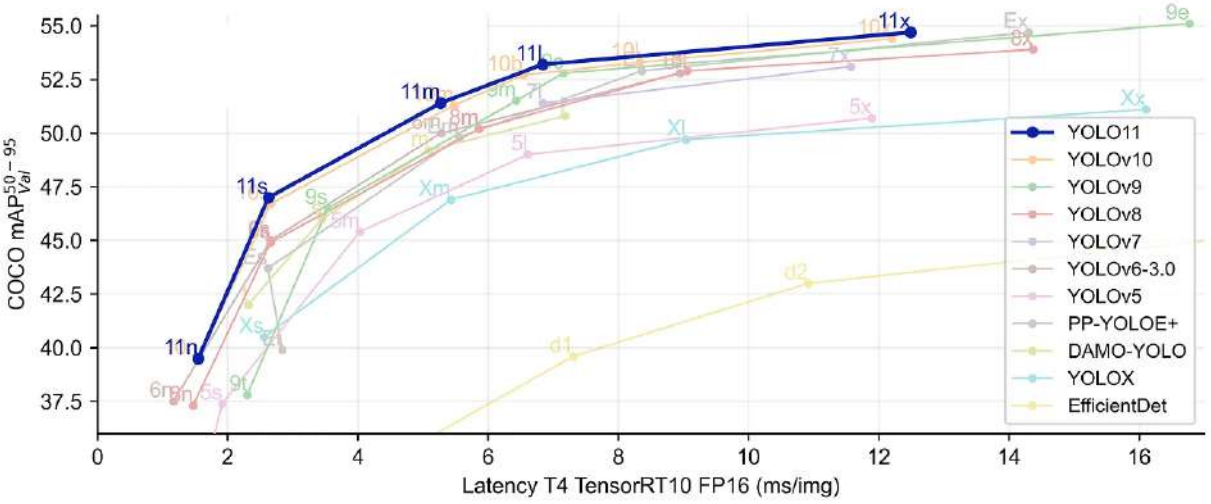

Рисунок 2.3 – Порівняльний аналіз моделей на наборі даних COCO. ***[20]***

Початковий аналіз показав, що сімейство архітектур YOLO забезпечує кращий баланс між точністю та продуктивністю. YOLOv12 виявилася особливо багатообіцяючою, продемонструвавши високу адаптивність до об'єктів з різною морфологією та розмірами - від невеликих звукових ефектів до великих діалогових панелей.

В процесі розробки було також проведене власне експериментальне порівняння різних версій YOLO (v8, v9, v11, v12) та альтернативної трансформерної архітектури на основі ResNet50 на спеціалізованому датасеті коміксів. YOLOv12 продемонструвала найвищу точність (F1-score 0.86) при збереженні високої швидкодії (29 мс/зображення), тоді як ResNet50 показав відносно нижчу точність (0.82), проте в 2.8 рази довший час інференсу (81 мс/зображення).

Для задачі детекції текстових блоків у коміксах було проведено комплексний аналіз 150+ зразків глав з різних жанрів і стилів. У результаті аналізу та кластеризації візуальних характеристик текстових елементів було виділено **13 основних класів**, що охоплюють усі типові сценарії текстового оформлення в азійських коміксах. Результати аналізу подані *у таблиці 2.1.*

Таблиця 2.1 – Характеристики класів текстових елементів у манхві

Клас тексту	**Візуальні характеристики**	**Функціональне призначення**	**Особливості стилізації**
speech_bubble	Круглий білий балон, чорний текст	Звичайний діалог персонажів	Природний діалоговий стиль
cloud_bubble	Хмароподібний балон з плавними контурами	Піднесення, жвавість, сміх	Пом’якшення, розтягування слів
scream_bubble	Гострі, шипасті контури	Крик, сильні емоції	Капіталізація, емфатичні конструкції
spiked_bubble	Голчасті краї, часто з темним фоном	Думки персонажа, беззвукова передача, рефлекси	Рефлексивний, інтроспективний стиль
system_bubble	Прямокутний, часто з кольоровим фоном	Системні повідомлення, інтерфейс	Формальний, технічний стиль
think_bubble	Балон з хвилястими краями, часто з ланцюжком малих кіл	Внутрішній хід думок	Використання конструкцій внутрішнього монологу
standalone_text	Текст без балона	Фоновий голос, другорядні фрази	Легкий риторичний стиль
sfx_sound	Стилізований текст, часто інтегрований у малюнок	Звукові ефекти	Функціональні еквіваленти звуконаслідувань
hexagon_bubble	Шестикутний балон	Технічні діалоги, заклинання	Механічний, точний стиль
square_bubble	Прямокутний балон	Закадровий текст, наратив	Оповідний стиль наратора
wiggle_bubble	Хвилястий балон	Невпевнена мова, тремтіння	Фрагментарні конструкції, еліпсиси

title_panel	Великий текст в декоративній рамці	Заголовки, назви розділів	Подальша заміна на локалізований аналог
background_text	Текст, інтегрований у сцену	Написи на вивісках, книгах тощо	Адаптація до контексту сцени

Кожен клас характеризується унікальними візуальними ознаками, як показано *на рисунку 2.4:*

Рисунок 2.4 – Приклади різних класів текстових елементів у манхві

Для навчання моделі було сформовано спеціалізований датасет, що включає розмічені приклади всіх класів текстових елементів з різноманітних манхва. Деталі формування та аугментації датасету будуть розглянуті у третьому розділі.

2.4. Конвеєр просторової обробки текстових блоків

2.4.1. Алгоритм упорядкування текстових блоків

Важливим аспектом обробки текстових елементів у коміксах є їх правильне упорядкування згідно з логікою наративу. На відміну від звичайного тексту, комікси зав'язані на складній просторовій організації з нелінійним порядком читання, що залежить від типу коміксу.

Запропонований у цій роботі алгоритм працює у два етапи. На першому етапі відбувається групування текстових блоків за їх вертикальним положенням. Якщо верхня межа поточного блоку знаходиться нижче нижньої межі групи, створюється нова група. Формально, блоки b_i та b_j належать до однієї групи G_k, якщо:

$$G(b_i, b_j) = \max(y_{1i}, y_{1j}) < \min(y_{2i}, y_{2j}).$$

де (y_{1i}, y_{2i}) – верхня і нижня межі блоку b_i;
(y_{1j}, y_{2j}) – верхня і нижня межі блоку b_j;

На другому етапі відбувається сортування блоків у межах групи. Для манхв застосовується сортування урахуванням відносної вертикальної позиції. Якщо один блок розташований значно вище іншого (різниця перевищує половину висоти блоку) - він зчитається першим. Якщо блоки розташовані на одному рівні, то спочатку зчитається блок, розташований лівіше. Формально, для блоків b_i та b_j з однієї групи визначається порядок:

$$P(b_i) < P(b_j) \Leftrightarrow \left(y_{1i} < y_{1j} - \frac{y_{2j}-y_{1j}}{2}\right) \vee \left(\left(|y_{1i} - y_{1j}| \leq \frac{y_{2j}-y_{1j}}{2}\right) \wedge (x_{1i} < x_{1j})\right).$$

де $P(b_i), P(b_j)$ – порядкові номери блоків b_i та b_j відповідно;

y_{1i}, y_{1j} – верхні координати блоків;

y_{2j} – нижня координата блоку b_j;

x_{1i}, x_{1j} – ліві координати блоків;

Такий підхід до упорядкування враховує специфіку манхви, де основний порядок читання - зверху вниз, але в межах одного рівня може застосовуватися порядок зліва направо (*рисунок 2.5*).

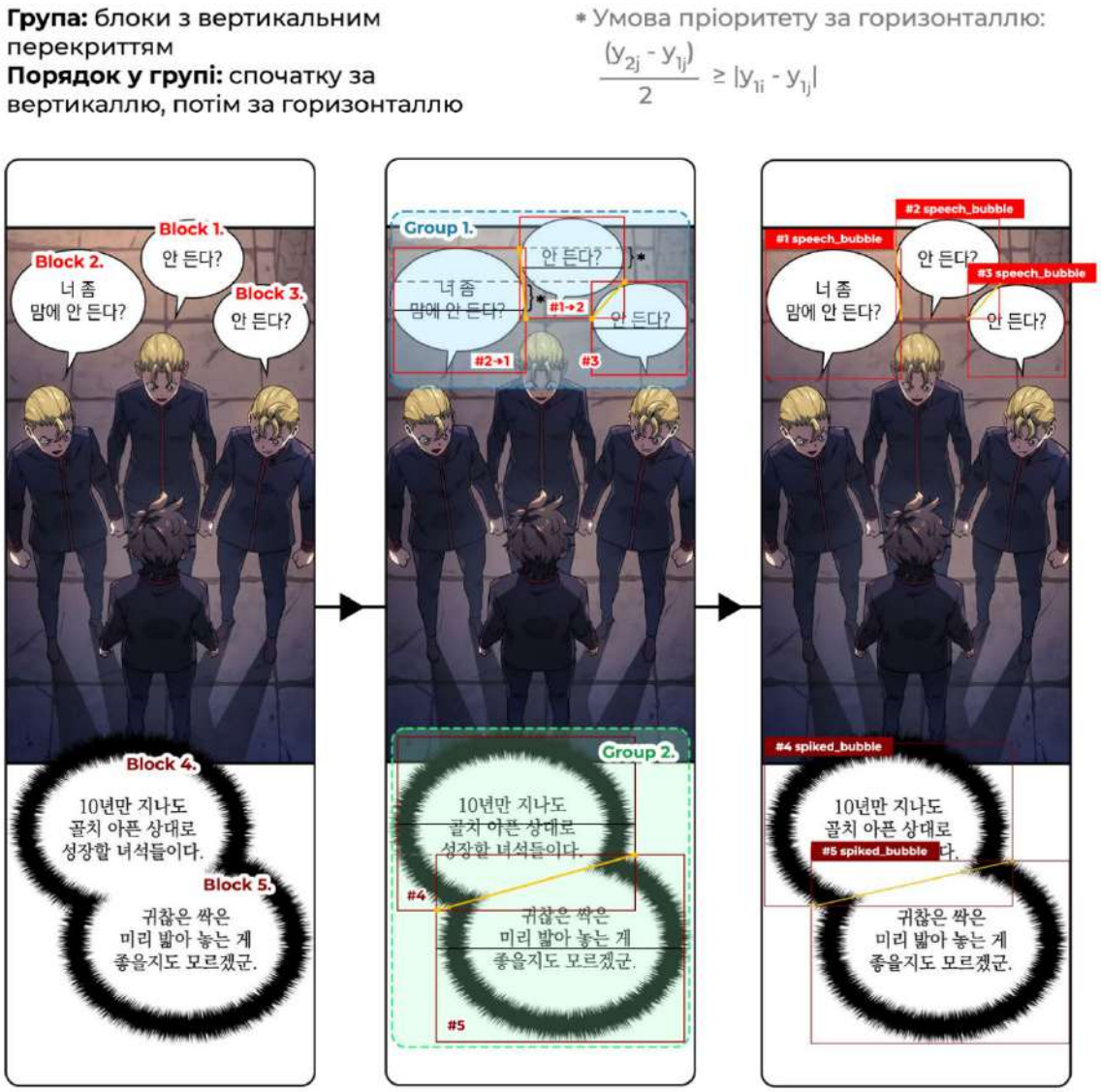

Рисунок 2.5 – Схема процесу групування та сортування блоків

2.4.2. Алгоритм маскування перекриттів

Іншою особливістю текстових блоків у коміксах є можливість їх перекриття, що створює проблеми при подальшій обробці. Для вирішення цього питання розроблено алгоритм маскування накладань, що базується на геометричному аналізі взаємного розташування блоків:

Для двох блоків що перекриваються, алгоритм визначає лінію відсікання, яка мінімізує втрату інформації. У коміксах перекриття текстових блоків трапляється досить часто — через брак місця чи для створення певного ефекту. Замість простого надання пріоритету одному блоку над іншим, створений підхід аналізує, як зберегти найбільш важливі частини обох блоків.

Ця лінія визначається двома точками перетину контурів блоків. Для прямокутних блоків ці точки можуть бути знайдені аналітично шляхом аналізу перетину сторін прямокутників. Загальне рівняння лінії відсікання має стандартний вигляд $Ax + By + C = 0$, де коефіцієнти A, B та C визначаються за точками перетину (*рисунок 2.6 – б.1*).

Для визначення, яку сторону від лінії відсікати для кожного з блоків, використовується аналіз положення геометричних центрів блоків відносно лінії відсікання. Блок b_i зберігає ту частину, де знаходиться його центр $(c_{xi}, c_{yi}) = \left(\frac{x_{1i}+x_{2i}}{2}, \frac{y_{1i}+y_{2i}}{2}\right)$. Формально, точка (x, y) зберігається в маскованому блоці b_i, якщо:

$$\left(A \cdot c_{xi} + B \cdot c_{yi} + C\right) \cdot (A \cdot x + B \cdot y + C) > 0.$$

Ця формула гарантує, що кожен блок зберігає ту свою частину, яка ближча до центру блоку. Такий підхід забезпечує, що після маскування текст найймовірніше залишиться читабельним, адже центральна частина тексту зазвичай містить найбільш важливу інформацію.

Особливим випадком є ситуація, коли один блок повністю міститься в іншому по одній з координат. Тоді точки перетину лежать на одній лінії, яка є стороною «більшого» блоку. В такому випадку застосовується модифікований підхід до побудови розмежувальної лінії.

У цьому способі спочатку визначається, який блок є «більшим» (містить інший блок в одній або обох координатах). Далі визначаємо найближчу точку на меншому прямокутнику до цього центру, формально:

$$p_{\text{closest}} = \arg \min_{p \in \text{boundary}(b_{\text{smaller}})} || p - C(b_{\text{bigger}}) ||_2$$

Після визначення центру більшого блоку та найближчої точки на меншому блоці, створюється перпендикулярний вектор $\overrightarrow{v_\perp} = (-v_y, v_x)$.

Лінія поділу проходить через найближчу точку в напрямку вектора перпендикуляра. Особливістю такого підходу є те, що менший текстовий блок залишається повністю незмінним, маскуванню піддається лише більший блок в області перекриття (*рисунок 2.6 – б.2*).

Це забезпечує збереження ключової інформації, яка часто міститься в менших елементах, і істотно підвищує ефективність подальшого розпізнавання тексту.

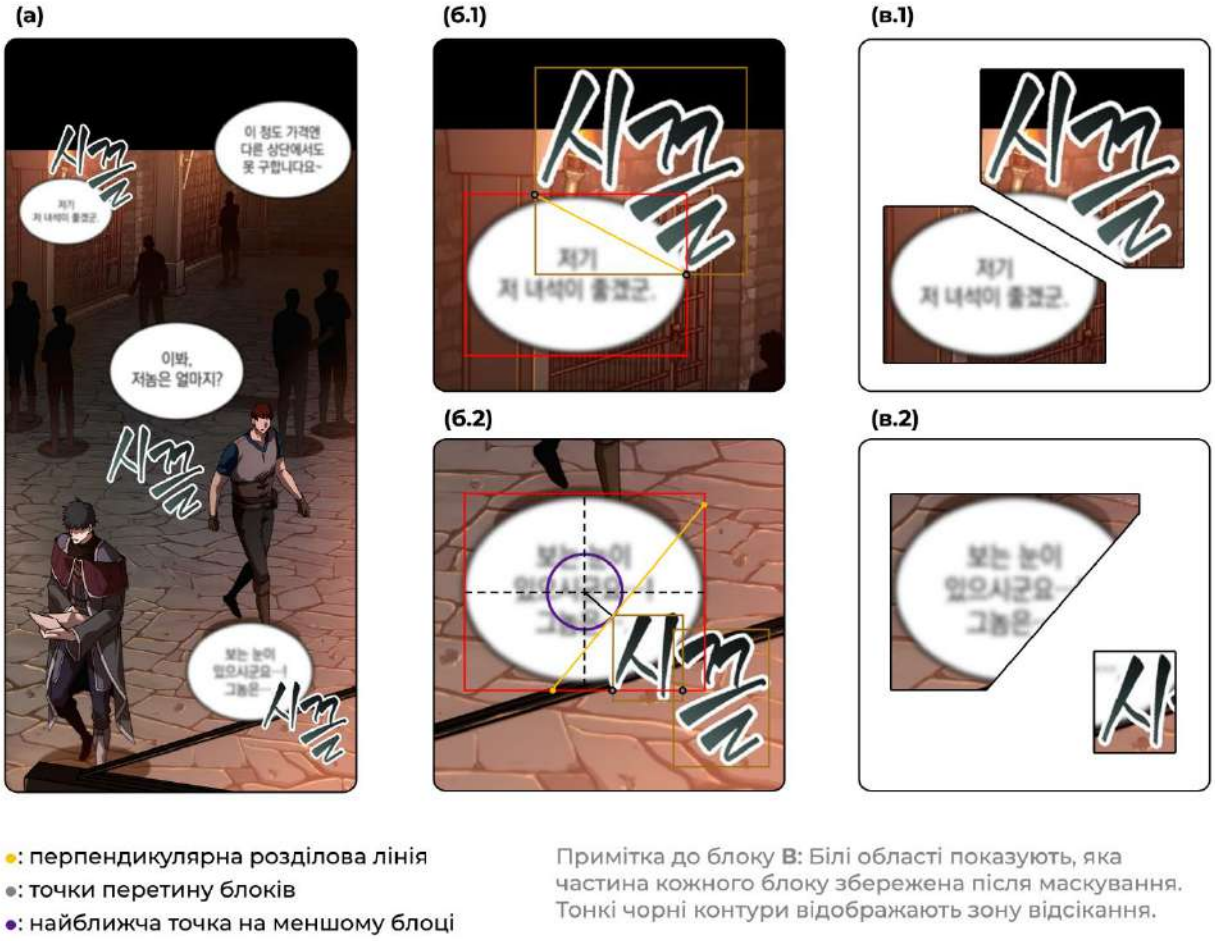

Рисунок 2.6 – Приклад маскування перекриттів текстових блоків: (а) вихідне зображення, (б.1, б.2) визначення ліній відсікання, (в.1, в.2) результати маскування

2.5. Агент розпізнавання ієрогліфічного тексту

Після виділення та класифікації текстових блоків система застосовує диференційований підхід до подальшої обробки залежно від класу виявленого блоку. Ключовими компонентами цього етапу є підсистеми детекції текстових рядків та розпізнавання символів у них.

Розподільчий конвеєр системи реалізує різні стратегії обробки для різних категорій текстових блоків. Структуровані класи, такі як *speech_bubble*, *think_bubble* та інші діалогові балони, які містять структурну частину разом із текстом, обробляються повним конвеєром із детекцією текстових рядків.

Неструктуровані класи, зокрема *standalone_text*, *sfx_sound* та *background_text*, що інтегровані безпосередньо в малюнок без чіткого структурного оформлення, проходять спрощений конвеєр, де виділені області передаються безпосередньо до OCR-моделі. Спеціальні класи, такі як *title_panel*, обробляються модифікованим конвеєром з адаптованими параметрами детекції.

2.5.1. Детекція текстових строк за допомогою CRAFT

Після виявлення та класифікації текстових балонів необхідним етапом є детекція безпосередньо текстових рядків у кожному з них. Для цього використовується спеціалізована модель **CRAFT** (Character Region Awareness For Text Detection) [21], яка демонструє високу ефективність у виявленні тексту в природних сценах та подібних середовищах з неоднорідним фоном.

Вибір моделі CRAFT для детекції текстових рядків у коміксах обґрунтований її значними перевагами порівняно з іншими методами. CRAFT демонструє високу ефективність при роботі з нестандартними, художніми та деформованими шрифтами, характерними для коміксів. Модель забезпечує точне виявлення тексту на неоднорідному фоні та ефективно обробляє текст різних напрямків, що особливо важливо для азійських коміксів.

Ці переваги підтверджуються наявними порівняльними дослідженнями. У роботі Baek et al. [21] продемонстровано, що CRAFT досягає F1-score 86.9% на наборі даних ICDAR2015, що перевершує результати інших методів, таких як EAST [22] (80.7%) та TextBoxes++ [23] (82.9%). Додаткові експерименти на спеціалізованому наборі даних манхви, проведені в межах даного дослідження, підтвердили ці результати, показавши, що CRAFT зберігає високу точність навіть для художньо стилізованого тексту в коміксах.

Математична основа цієї моделі полягає у прогнозуванні двох карт характеристик: карти регіонів символів (character region map) та карти афінності (affinity map). Перша відображає ймовірність приналежності пікселя до символу, друга моделює зв'язки між сусідніми символами. На виході алгоритм надає набір

текстових рядків, виявлених у кожному текстовому балоні, з їх точними координатами для подальшої обробки (*рисунок 2.7*).

Рисунок 2.7 – Візуалізація роботи CRAFT на прикладі балону манхви:
(а) вхідний балон з текстом, (б) карта регіонів символів,
(в) карта афінності, (г) результат виділення текстових рядків

2.5.2. Розпізнавання ієрогліфічного тексту за допомогою transformer-моделей

Традиційні OCR системи, розроблені для латинських алфавітів, демонструють обмежену ефективність при роботі з ієрогліфічними системами письма через велику кількість унікальних символів, варіативність шрифтів та стилізацію тексту.

Для вирішення цієї проблеми було обрано підхід на базі трансформерних моделей **TrOCR** (Transformer-based Optical Character Recognition) [24]. Основою цього підходу є концепція "sequence-to-sequence learning" з використанням архітектури encoder-decoder з механізмом уваги.

На відміну від класичних підходів до OCR, що базуються на сегментації та розпізнаванні окремих символів, TrOCR розглядає зображення тексту як цілісну послідовність і перетворює її безпосередньо в текстову послідовність. Це виявляється особливо ефективним для ієрогліфічних систем письма, де межі між символами часто нечіткі, а сам текст розташовано досить стисло.

Архітектура TrOCR складається з двох основних компонентів. Енкодер зображень представляє собою згорткову нейронну мережу, що перетворює зображення в послідовність векторів ознак. Для цієї ролі використовуються претреновані моделі, такі як ResNet [25] або Vision Transformer (ViT) [26]. Декодер тексту є трансформерним декодером, що перетворює векторні представлення в текстову послідовність. Він працює автоагресивно, генеруючи по одному символу за раз і використовуючи механізм самоуваги та перехресної уваги для врахування контексту.

Перевага TrOCR порівняно з класичними методами OCR для ієрогліфічного тексту полягає у стійкості до варіацій шрифтів, ефективному масштабуванні та можливості обробки рядків символів замість окремих слів. Модель розглядає текст як цілісну послідовність, що дозволяє використовувати контекстуальну інформацію для визначення певних символів.

2.5.3. Дуальна архітектура розпізнавання тексту

Інноваційним аспектом запропонованого підходу є використання дуальної архітектури розпізнавання тексту, що поєднує переваги двох спеціалізованих моделей на основі TrOCR, але з різними стратегіями навчання. (*рисунок 2.8*)

Перша модель [27], яку ми називаємо "строгою" (precision-oriented), розроблена корейським дослідницьким центром *TeamLucid* і є модифікацією

базової моделі TrOCR, спеціально оптимізованою для корейських текстів. Ця модель характеризується високою точністю при розпізнаванні стандартизованого тексту, але через надмірну спеціалізацію (overfitting) демонструє обмежену здатність до розпізнавання нестандартних надписів.

Друга модель, "гнучка" (recall-oriented), розроблена в рамках даного дослідження шляхом дотренування базової версії microsoft/trocr-base-handwritten [28] на спеціально створеному наборі з двох мільйонів синтетичних зображень корейського тексту з високою варіативністю шрифтів, розмірів, та стилів. На відміну від строгої моделі, ця версія була навмисно доведена до стану легкого недотренування (underfitting), що забезпечило їй вищу гнучкість і здатність розпізнавати нестандартні варіанти написання символів, хоча й з дещо нижчою загальною точністю.

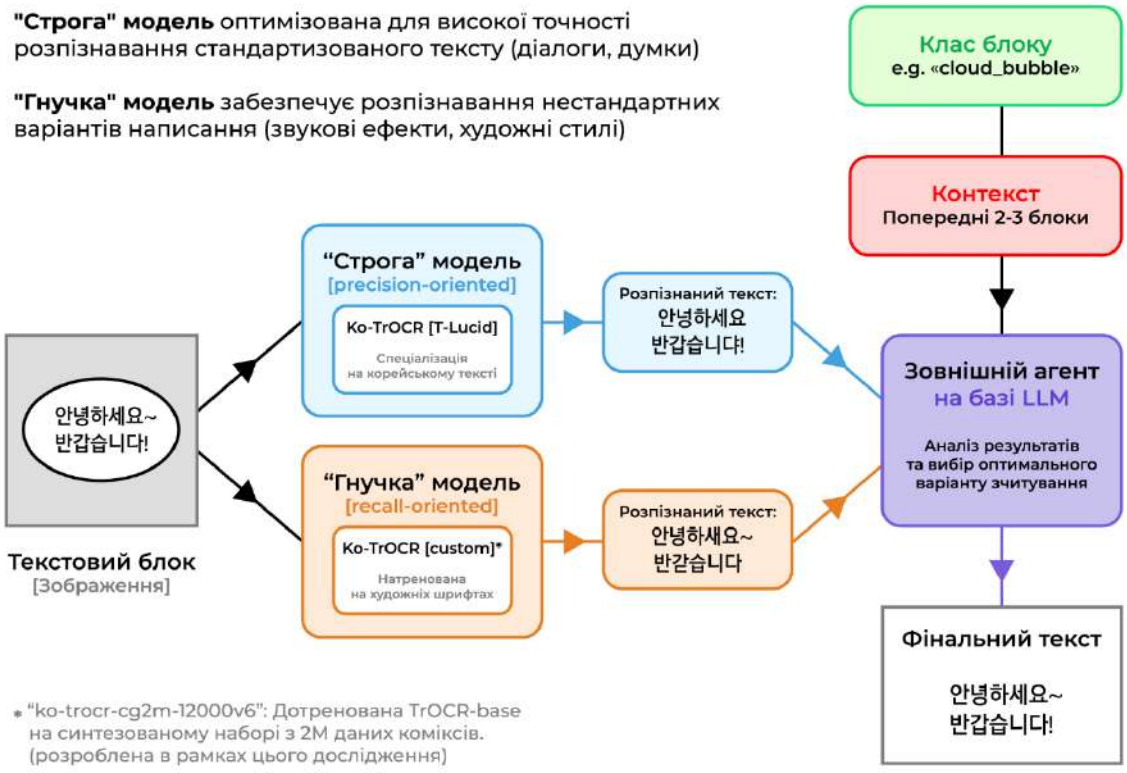

Рисунок 2.8 – Схема дуальної архітектури розпізнавання тексту

Теоретичним обґрунтуванням обраного підходу є концепція "ансамблевого навчання" (ensemble learning), де комбінація декількох моделей забезпечує кращі

результати, ніж кожна модель окремо. Для ієрогліфічного тексту в коміксах це особливо актуально через широкий спектр стилів написання - від стандартизованих діалогів до художньо стилізованих звукових ефектів.

З метою підвищення точності розпізнавання тексту в подвійній архітектурі було впроваджено додаткову корекцію розпізнавання, яка використовує великі мовні моделі для аналізу та порівняння результатів роботи обох OCR-модулів.

Робота коректора відбувається за наступним алгоритмом:

- Він отримує результати розпізнавання тексту від обох моделей ("суворої" та "гнучкої").
- Передає ці результати в спеціалізований промпт для великої мовної моделі LLM.
- Явно аналізує обидві версії з урахуванням контексту та лінгвістичних особливостей.
- На виході формується найточніша версія тексту з урахуванням виявлених помилок.

Така дуальна архітектура дозволяє ефективно балансувати між високою точністю розпізнавання типових текстів та здатністю обробляти нестандартні шрифти й стилізації, що є критичним для якісного опрацювання різноманітних текстових елементів у манхві.

2.6. Агент стилізованого перекладу

2.1.1. Теоретичні засади контекстно-залежного перекладу

Класичні системи машинного перекладу розглядають текст як незалежну послідовність фрагментів, ігноруючи його візуальний контекст і стилістичні особливості. Запропонований підхід ґрунтується на концепції «контекстно-залежного перекладу», де в процесі обробки враховується не тільки сам текст, але і його візуальні характеристики. В обраній предметній області така

інформація складається з ***класу текстового блоку*** (діалог, думка, звуковий ефект тощо), ***позиції*** в наративній послідовності та ***контекстного вікна*** (попередні 2-3 текстові блоки).

Практична реалізація такого перекладу потребує включення в алгоритм інформації про характеристики текстового блоку, отриманих на етапі класифікації. Для цього були використані спеціалізовані промпти для великих мовних моделей, які беруть до уваги різні аспекти цього контексту.

Таблиця 2.2 – Приклади стилізації перекладу для блоків різних класів

Клас тексту	**Ключові елементи промпту**	**Приклад входу**	**Приклад перекладу без стилізації**	**Приклад стилізованого перекладу**
speech_bubble	Звичайний конверсаційний стиль	"어서 와!"	"Ласкаво просимо!"	"Заходь швидше!"
scream_bubble	Різкість, знаки оклику	"안돼~~!!!"	"Не треба!"	"Ні-і-і!!!"
cloud_bubble	Жвавий, веселий тон	"히히히"	"Хіхіхі"	"Хі-хі-хі~"
system_bubble	Формальний, технічний стиль	"[유저 경험치: +50]"	"[Досвід користувача: +50]"	"[ОТРИМАНО: 50 очків досвіду]"
sfx_sound	Експресивність, звуконаслідування, капіталізація	"쾅"	"Бах"	"БА-БАХ!"

2.6.1. Гібридний підхід до стилізованого перекладу

Розвиток великих мовних моделей (LLM) суттєво змінив підходи до машинного перекладу, особливо для задач, що вимагають стилістичної адаптації. Як зазначає дослідження у згаданій раніше роботі ogkalu2/comic-translate [13]:

"Для багатьох мовних пар, особливо між віддаленими мовами як корейська та англійська, найкращим перекладачем є не Google Translate, Papago чи навіть DeepL, а сучасні великі мовні моделі, які демонструють значно кращі результати".

Для вирішення задачі стилізованого перекладу в нашій системі використовується хмарна реалізація великої мовної моделі: Claude 3.7 Sonnet (через API). Вибір цієї моделі обумовлений кількома факторами.

Попередні експерименти з меншими спеціалізованими моделями перекладу, такими як FlanT5 та Helsinki-NLP, показали їхню недостатню ефективність для передачі стилістичних нюансів. Хоча ці моделі можуть забезпечити адекватний лінгвістичний переклад, вони не здатні на більш ґрунтовні задачі без окремого ґрунтовного донавчання.

Крім того, великі мовні моделі демонструють унікальну здатність працювати з проміжними інструкціями (промптами), що дозволяє точно специфікувати бажаний стиль перекладу. Це особливо важливо для передачі емоційного забарвлення та специфічних жанрових особливостей коміксів. Як показують порівняльні дослідження [29], сучасні LLM значно краще справляються з перекладом між віддаленими мовними парами, такими як корейська-українська, де традиційні системи машинного перекладу дають неадекватні результати через фундаментальні відмінності в граматичних структурах та культурному контексті.

2.6.2. Використання контекстних вікон для забезпечення когерентності перекладу

Важливим аспектом перекладу коміксів є забезпечення смислової та стилістичної цілісності між суміжними текстовими блоками. Для вирішення цієї проблеми запропоновано механізм "контекстних вікон", що дозволяє враховувати попередній контекст при перекладі кожного текстового блоку.

Контекстне вікно визначається як набір (попередніх або наступних) текстових блоків, впорядкованих згідно з логічною послідовністю наративу. Для практичної реалізації використовується розширений промпт для великої мовної моделі, що включає попередні діалоги та їхні класи (*рисунок 2.9*).

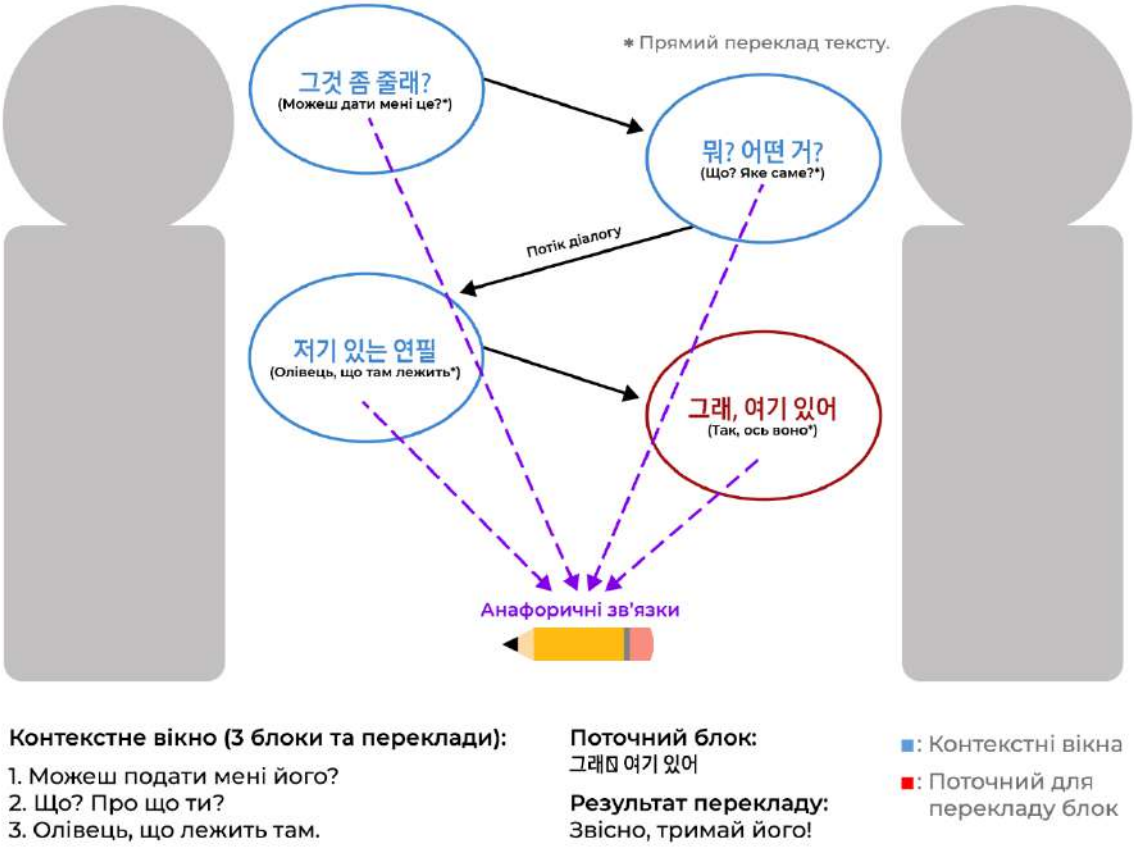

Рисунок 2.9 – Діаграма принципу роботи контекстних вікон

На практиці це реалізується шляхом включення попередніх реплік в промпт до LLM. Такий підхід вирішує кілька конкретних проблем:

- Коректний переклад займенників. Наприклад, якщо в попередньому блоці згадано конкретний предмет, а в наступному на нього посилаються займенником "그것" (це/воно), система зможе правильно перекласти, зберігаючи референцію.
- Послідовне використання термінології. Якщо певний термін (наприклад, назва фентезійного предмета "어둠의 검") вже був

перекладений як "Меч Темряви", система буде використовувати цей же переклад у наступних блоках.

- Збереження тону діалогу. Якщо персонажі спілкуються формально або використовують специфічні звертання, ця особливість зберігається послідовно через усі блоки їхньої розмови.

Проведені експерименти, що базуються на методології, запропонованій Tiedemann & Scherrer (2017) [30], показали значне покращення зв'язності перекладу при використанні контекстних вікон порівняно з поблоковим перекладом без контексту. Зокрема, за метриками когерентності та узгодженості термінології спостерігалося покращення на 15-20%, що узгоджується з результатами, отриманими Xinye et al. (2024) [31] для селективного підходу до контекстного перекладу.

2.6.3. Структура промптів для стилізованого перекладу

Ключовим елементом реалізації стилізованого перекладу є розробка спеціалізованих промптів для великих мовних моделей, які інкапсулюють інформацію про клас текстового блоку, контекст та бажаний стиль перекладу. Важливо зазначити, що ефективність промпту безпосередньо впливає на якість та стилістичну точність перекладу. Нижче наведено шаблон, що використовується для побудови промптів:

```
[TASK]: Translate text from {source_lang} to {target_lang} with appropriate style.
[CONTEXT]: {context_section}
[ORIGINAL TEXT]: original_text}
[STYLE GUIDE]: {style_guide}
[INSTRUCTIONS]: Translate the original text from {source_lang} to {target_lang}
following the style guide appropriate for a {bubble_type}.Analyze the original text,
taking into account any context provided and the style guide. Think carefully about
```

```
cultural nuances, idioms, and emotional tone. After your analysis, provide ONLY the translated text with no additional explanation, starting with 'TRANSLATED_TEXT:' followed by the translation.
```

Елемент *{style_guide}* варіюється залежно від класу текстового блоку. Наприклад, для класу *scream_bubble* він має такий вигляд:

```
This text represents shouting or extreme emotion. For emphasis use exclamation marks and emotional language. Translate to convey high intensity and strong emotion. Add appropriate interjections if needed to enhance the emotional impact.
```

А для класу *think_bubble*:

```
This text represents inner thoughts of a character. Use introspective, reflective style. Add thought fragments, ellipses (...) where appropriate to convey thinking process. Translate to create a distinct internal voice that differs from spoken dialogue.
```

Важливим елементом промпту є також *{context_section}*, який формується на основі сусідніх текстових блоків для забезпечення зв'язності перекладу. Цей розділ має таку структуру:

```
[CONTEXT]: Surrounding texts:
Prev text 1 (speech_bubble): {previous_text_1}
Prev text 2 (think_bubble): {previous_text_2}
```

Для оцінки ефективності різних структур промптів було проведено серію експериментів, де порівнювалися різні варіації інструкцій та їх вплив на якість перекладу. Результати показали, що включення конкретних прикладів стилізації для кожного класу текстових блоків значно покращує відповідність перекладу бажаному стилю, порівняно з базовими інструкціями без прикладів.

2.7. Інтеграція компонентів у єдину систему

2.7.1. Архітектура модульної взаємодії

Для ефективної інтеграції всіх компонентів у єдину систему розроблено структуровану організацію програмного комплексу, що забезпечує послідовну обробку даних через чітко визначені функціональні блоки. Запропоновано модульну архітектуру, де компоненти функціонують як взаємопов'язані класи в рамках єдиного програмного середовища.

Архітектура системи базується на принципі конвеєрної обробки, де дані послідовно проходять через визначені етапи перетворення. Кожен модуль має чітко визначені інтерфейси введення та виведення, що забезпечує можливість заміни окремих компонентів без перебудови всієї системи. Така структура дозволяє легко модифікувати та вдосконалювати окремі етапи обробки, зберігаючи загальний потік даних незмінним.

Реалізація системи включає кілька ключових аспектів. Кожен функціональний компонент (модуль попередньої обробки, детектор, OCR-модуль, перекладач) реалізований як окремий клас з визначеним інтерфейсом та інкапсульованою внутрішньою логікою. Впроваджено механізми кешування проміжних результатів на рівні кожного модуля, що дозволяють оптимізувати роботу системи при повторних запитах та зберігати проміжні результати для подальшого аналізу. Реалізовано систему обробки помилок з каскадним механізмом відновлення, що забезпечує стійкість до аномалій на всіх етапах обробки. Введено стандартизований формат даних для обміну між модулями, що

базується на структурованих об'єктах з атрибутами, які відповідають різним аспектам обробки тексту.

Така архітектура забезпечує ефективне використання обчислювальних ресурсів, оскільки всі компоненти працюють в рамках одного обчислювального середовища без накладних витрат на мережеву комунікацію. Водночас, модульна структура зберігає гнучкість системи та можливість її масштабування.

2.7.2. Оптимізація обчислювальної ефективності

Важливим аспектом інтегрованої системи є оптимізація обчислювальної ефективності для забезпечення практичної застосовності при обробці великих обсягів даних. Під час розробки системи було приділено особливу увагу балансу між якістю обробки та швидкодією, що є критичним для інтерактивного використання.

Розподіл обчислювальних ресурсів відбувається динамічно залежно від складності текстових елементів. Для стандартних діалогових балонів, які складають більшість тексту в коміксах, система використовує оптимізовані методи розпізнавання, тоді як для складніших елементів, таких як стилізовані звукові ефекти, застосовуються більш ресурсоємні, але точніші алгоритми. Такий адаптивний підхід забезпечує оптимальне використання доступних обчислювальних ресурсів.

Для підвищення продуктивності впроваджено техніки оптимізації, такі як пакетна обробка, що дозволяє ефективніше використовувати GPU при інференсі нейромережевих моделей. Незалежні операції виконуються паралельно з використанням багатопотоковості, а система динамічно обирає оптимальну модель залежно від типу текстового блоку, зберігаючи обчислювальні ресурси.

Експериментальний аналіз часової ефективності показує, що для типового розділу манхви (приблизно 80-100 сегментів розбиття, 100-200 текстових блоків) локальний цикл обробки займає від 100 до 300 мс, а хмарний модуль з запитами до великих мовних моделей – ще від 200 до 500 мс на текстовий блок, залежно

від його складності. Таким чином, обробка цілого розділу вимагає 120-180 секунд, що є прийнятним для практичного використання навіть в режимі реального часу. Порівняння з альтернативними підходами демонструє, що запропонована архітектура досягає балансу між якістю результатів та обчислювальною ефективністю, забезпечуючи практичну цінність системи для реальних сценаріїв використання.

2.8. Висновки до розділу 2

У другому розділі представлено теоретичне обґрунтування розробленого методу класифікації та стилізованого перекладу ієрогліфічного тексту в мультимедійних коміксах. Розроблено архітектуру конвеєрної системи, що забезпечує повний цикл обробки від сегментації зображення до стилізованого перекладу.

Ключовими інноваціями запропонованого підходу є комплексна класифікація текстових елементів за їхніми візуальними характеристиками з використанням однопрохідної моделі YOLO. Розроблено алгоритм упорядкування та маскування текстових блоків, що враховує просторові відносини та логіку наративу для реконструкції природного порядку читання коміксу.

Запропоновано дуальну архітектуру розпізнавання ієрогліфічного тексту на базі трансформерних моделей TrOCR, що поєднує переваги "строгої" та "гнучкої" моделей для забезпечення оптимального балансу між точністю та повнотою розпізнавання різних стилів тексту. Розроблено інноваційний підхід до стилізованого перекладу, що базується на використанні великих мовних моделей Claude з інтеграцією контекстних даних у спеціалізовані промпти, що дозволило адаптувати стиль перекладу відповідно до класу текстового блоку та зберегти емоційні й стилістичні нюанси оригіналу.

Впроваджено механізм контекстних вікон для забезпечення когерентності перекладу через суміжні текстові блоки, що підвищує природність та зв'язність

перекладеного тексту. Теоретичний аналіз показує, що розроблена система в цілому перевершує існуючі підходи до автоматизованої локалізації коміксів завдяки інтеграції всіх етапів обробки в єдину систему та врахуванню візуальних та стилістичних особливостей.

У наступному розділі буде представлено практичну реалізацію запропонованої системи, включаючи деталі формування датасету, навчання моделей та результати експериментальної оцінки ефективності системи на реальних прикладах коміксів.

РОЗДІЛ 3: Практична реалізація запропонованої системи

3.1. Реалізація модуля детекції текстових блоків

Базовим елементом запропонованої системи є модуль виявлення та класифікації текстових блоків, який забезпечує пошук різних типів діалогових вікон та інших текстових елементів. Розробка цього блоку вимагала вирішення ряду завдань, починаючи з підготовки навчального набору даних і закінчуючи оптимізацією обраних параметрів моделі.

3.1.1. Формування та підготовка датасету для навчання

Розробка ефективної системи розпізнавання вимагала створення спеціалізованого набору даних, який би адекватно відображав розмаїття візуальних форм представлення тексту. На відміну від загальних задач розпізнавання об'єктів, для яких вже існує достатньо велика кількість стандартизованих наборів даних, сфера азійських коміксів наразі не має публічно доступних анотованих корпусів, що й зумовило необхідність створення власного датасету.

Створений набір включає 13 категорій, які було детально розглянуто в розділі 2, з урахуванням їхніх семантичних та візуальних характеристик. Датасет включає близько **3 000** анотованих зображень, отриманих з різних за жанром корейських вебтунів, із загальною кількістю анотованих елементів понад **8 000** об'єктів. Для формування цього показового корпусу було використано відкриті розділи популярних творів різних авторів і видавництв, що дозволило створити вибірку, максимально наближену до реальних умов застосування системи. Особливу увагу було приділено включенню прикладів з різними стилями оформлення, рівнями деталізації та кількістю тексту на зображенні.

Розподіл класів у датасеті представлено *на рисунку 3.1*, де можна спостерігати природний дисбаланс, характерний для реальних коміксів, де

діалогові балони типу *speech_bubble* (2190 прикладів) зустрічаються значно частіше, ніж, наприклад, балони типу *wiggle_bubble* (110 прикладів). Прийнято свідоме рішення не балансувати датасет штучно, щоб зберегти його репрезентативність відносно цільового домену.

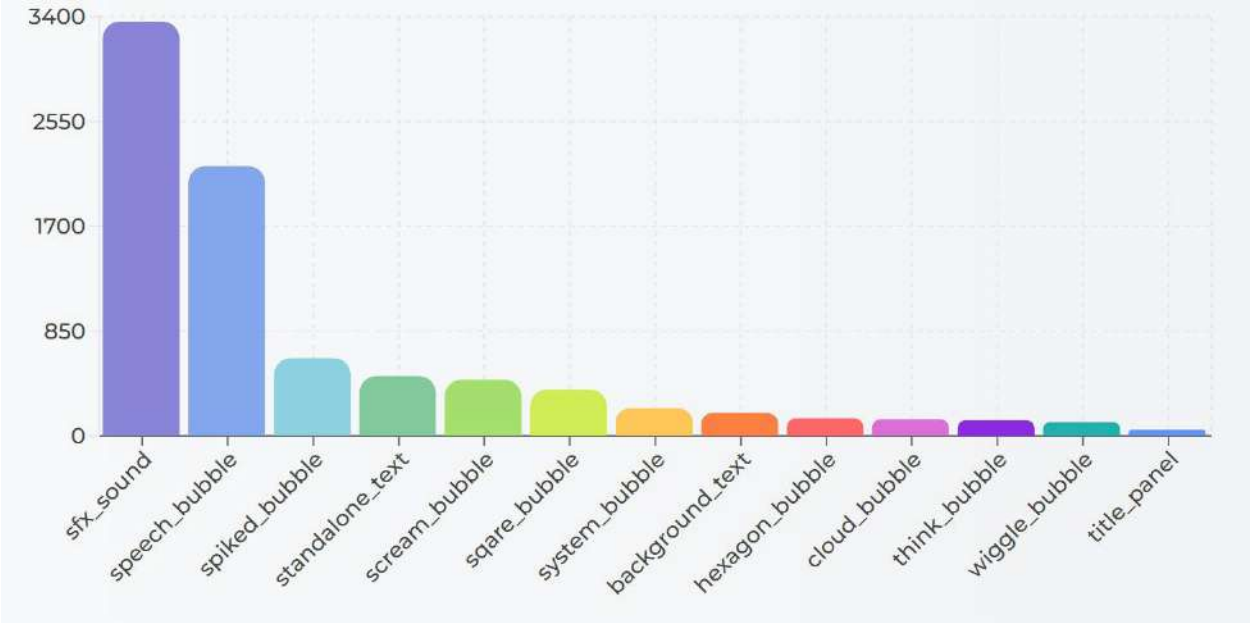

Рисунок 3.1 – Розподіл класів у навчальному датасеті детекції текстових блоків

Методологія створення датасету була реалізована як структурований ітеративний процес, що складався з кількох взаємопов'язаних етапів. На початковому етапі зібрано та вручну анотовано пілотний набір даних (~500 зображень). Особливу увагу було приділено правильній типології різних форм представлення тексту з урахуванням не лише візуальних, а й семантичних та функціональних характеристик елементів.

Отриманий пілотний набір даних послужив основою для навчання первинної моделі детекції, що значно прискорило подальший процес розмітки. Це дозволило отримати базову модель з достатньою точністю для напівавтоматичного розмічення великих обсягів даних, значно прискоривши процес анотування при збереженні високої якості результатів.

Вся анотація здійснювалася у форматі **COCO** (Common Objects in Context) за допомогою спеціалізованого інструменту розмітки Roboflow. Кожне анотоване зображення проходило ручну перевірку якості розмітки, з особливою увагою до правильної класифікації текстових блоків та точності обмежувальних рамок (*рисунок 3.2*).

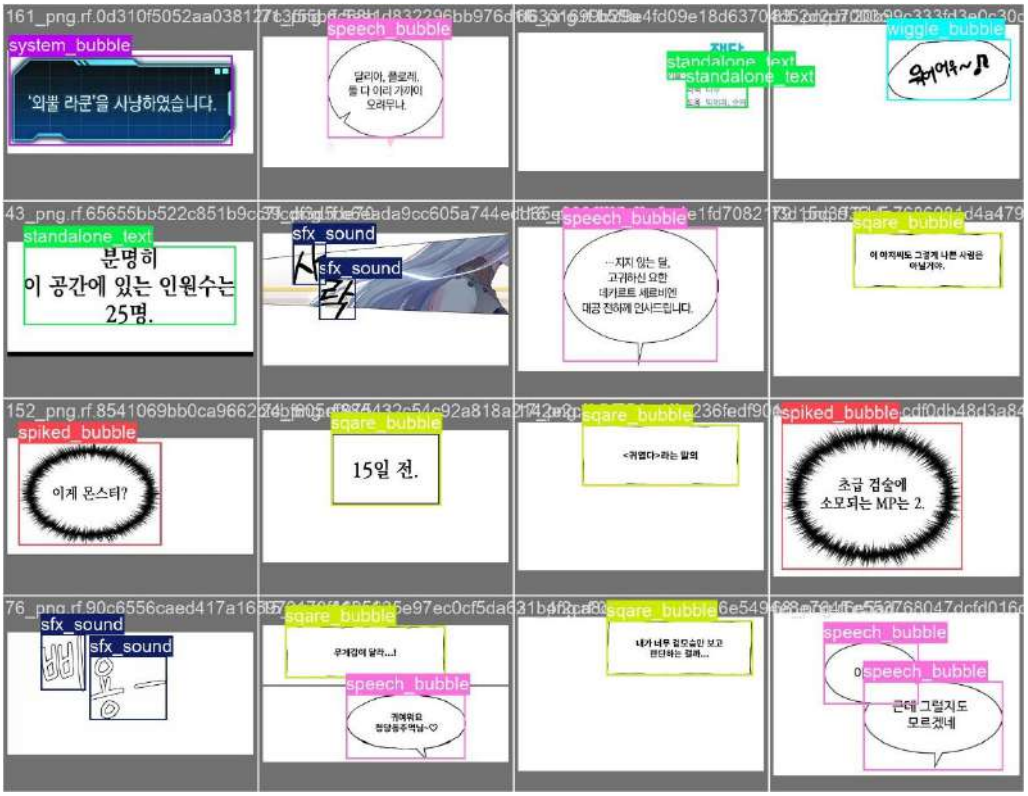

Рисунок 3.2 – Приклад вибірки анотованих зображень

Ретельний аналіз характеристик цільової області — зображень манхви — виявив значну варіативність параметрів яскравості, контрастності, чіткості та наявності артефактів від стиснення. Ці особливості зумовлені різноманіттям джерел походження коміксів, методів передачі та умов публікації. У результаті застосовано техніки аугментації зображень з такими параметрами:

- Генерація до 3 варіацій з кожного вхідного зображення;
- Варіації експозиції в діапазоні ±10%;
- Розмиття до 1.4 пікселів;
- Додавання шуму до 0.14% пікселів.

3.1.2. Вибір та обґрунтування архітектури моделі детекції

Детекція текстових блоків у коміксах має певні особливості, що відрізняють її від стандартних задач виявлення об'єктів: широка варіативність форм текстових елементів, часте перекриття, стилістична різноманітність та неоднорідне фонове зображення. Ці фактори вплинули на вибір оптимальної архітектури моделі.

Для більш обґрунтованого вибору було проведено експерименти з різними модифікаціями архітектур YOLO (yolo11m, yolo9m, yolo12m, yolo12s) та порівняння з DETR на єдиному тестовому наборі (*рисунок 3.3*).

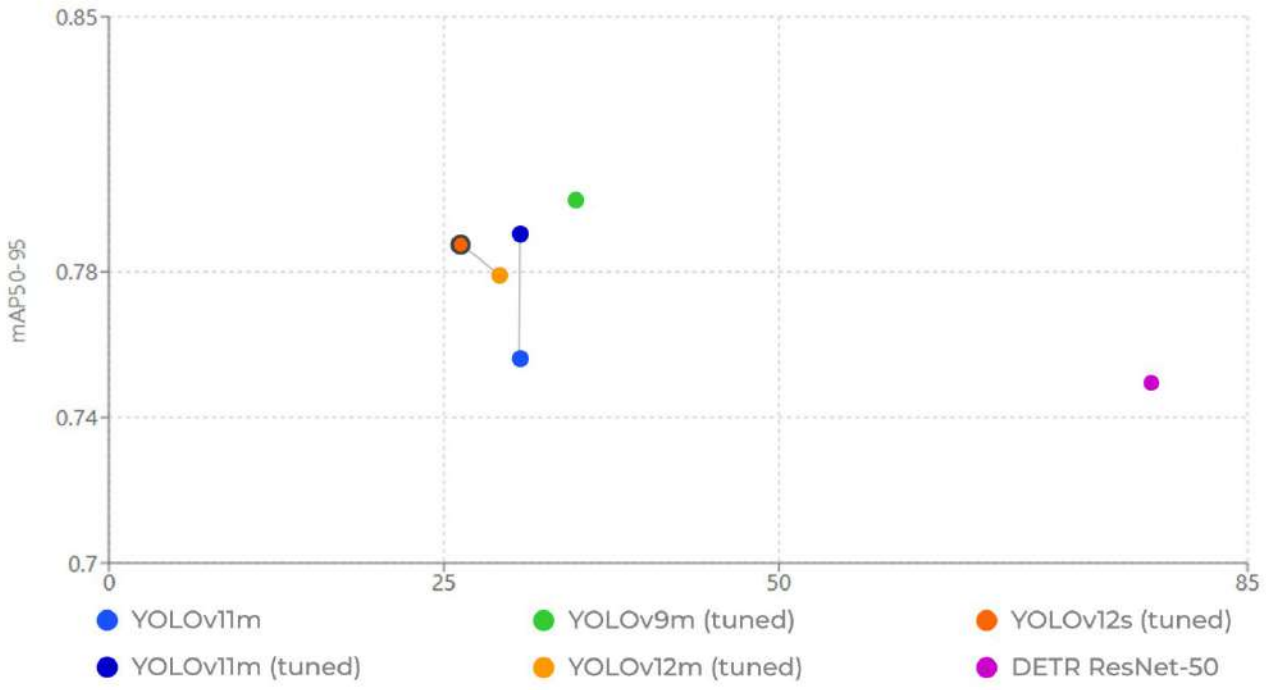

Рисунок 3.3 – Результати експериментів з різними архітектурами моделей детекції

Результати експериментів чітко показують, що моделі YOLO значно перевершують DETR за швидкістю (майже втричі швидше) з такою ж або вищою

точністю виявлення. YOLOv9m з оптимізованими гіперпараметрами показав найкращу точність і відгук, тоді як YOLOv12s забезпечує майже таку ж точність при найкоротшому часі виведення (27,2 мс на зображення). Ключові результати експериментів наведено в таблиці 3.1.

Таблиця 3.1 – Результати експериментів з різними архітектурами моделей детекції

Модель	***mAP50(B)***	***mAP50-95(B)***	***Precision(B)***	***Recall(B)***	***Inference (ms/img)***
YOLOv11m	0.829	0.761	0.896	0.786	30.7
YOLOv11m (tuned)	0.850	0.790	0.871	0.828	30.7
YOLOv9m (tuned)	**0.863**	**0.800**	**0.872**	**0.831**	32.9
YOLOv12m (tuned)	0.841	0.781	0.853	0.806	29.7
YOLOv12s (tuned)	0.853	0.786	0.860	0.825	**27.2**
DETR ResNet-50	0.826	0.749	0.720	0.810	78.0

Проаналізувавши компроміс між точністю та продуктивністю, ми обрали **YOLOv12s** як оптимальну архітектуру для остаточної реалізації системи. Ця модель забезпечує найвищу точність (mAP50 = 0,853) з найкоротшим часом обробки серед усіх протестованих моделей. Порівняно з DETR, YOLOv12s працює майже в 2,9 рази швидше, що має вирішальне значення для практичного застосування, особливо враховуючи, що типовий розділ рукопису містить 50-100 зображень.

3.1.3. Навчання моделі та оптимізація гіперпараметрів

Здійснивши вибір базової архітектури, ми зосередились на розробці оптимальної стратегії навчання моделі детекції та пошуку найефективніших гіперпараметрів, що максимізують її продуктивність для специфічної задачі розпізнавання текстових блоків у манхві. Весь процес навчання та оптимізації реалізований з використанням фреймворку Ultralytics, що надає розширену

інфраструктуру для ефективної імплементації та навчання архітектур сімейства YOLO.

Для оптимізації процесу навчання датасет було розділено на тренувальну (70%) валідаційну (20%) та тестову (10%) частини зі збереженням пропорційного представлення всіх класів у обох множинах. Особлива увага приділялася забезпеченню репрезентативності валідаційного набору для всіх 13 класів текстових блоків.

У початковій конфігурації для навчання моделі було встановлено наступні основні параметри:

- Розмір батчу: 16
- Кількість епох: 80
- Оптимізатор: AdamW
- LRS: Cosine decay з початковим LR 1e-4
- Weight decay: 0.001

Для пошуку оптимальних гіперпараметрів використано вбудований у фреймворк Ultralytics механізм автоматизованого тюнінгу, що базується на методі Tree-structured Parzen Estimator (TPE).

У результаті оптимізації визначено набір гіперпараметрів (*рисунок 3.4*), що суттєво впливають на ефективність моделі детекції текстових блоків у манхві.

Ключовими для досягнення високої продуктивності виявились коефіцієнт навчання *lr0 = 0.0033* та momentum = *0.93356*, що забезпечили оптимальну збіжність моделі. Найбільший вплив на точність мали спеціально налаштовані ваги функцій втрат: для *boundary box detection (box = 5.92061)*, класифікації типів *(cls = 0.62926)* та *distribution focal loss (dfl = 0.88804)*.

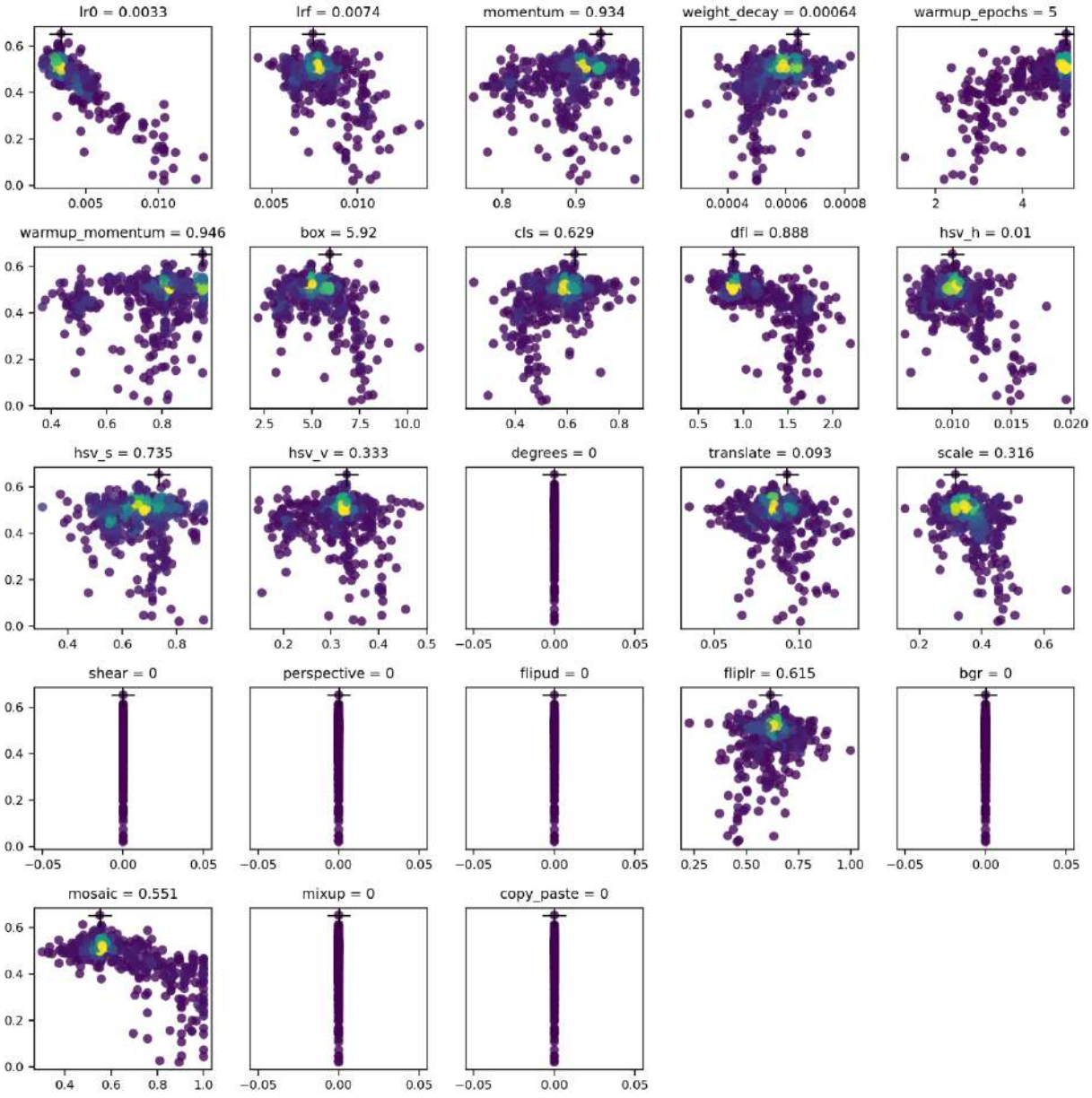

Рисунок 3.4 – Матриця змін гіперпараметрів під час оптимізації

Сам процес оптимізації включав 500 ітерацій з фокусом на найкритичніших параметрах, що впливають на якість детекції, включаючи коефіцієнти навчання, функції втрат та параметри аугментації (*рисунок 3.5*).

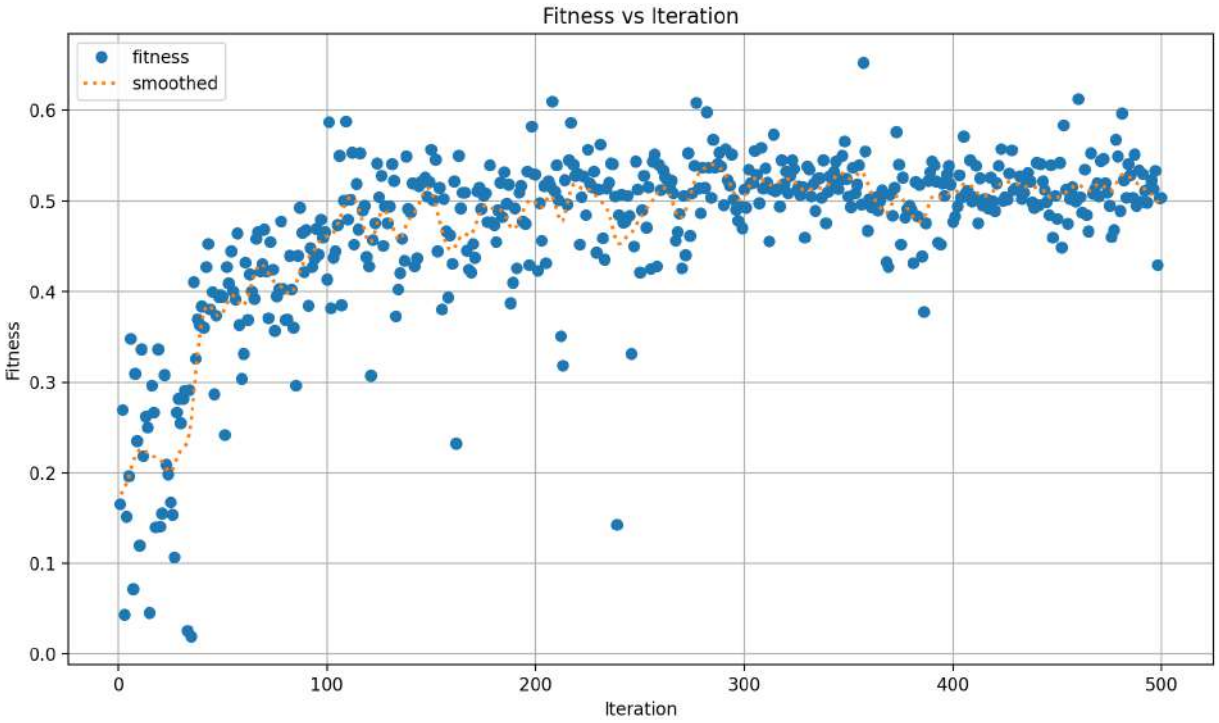

Рисунок 3.5 – Результати оптимізації гіперпараметрів моделі YOLOv12

Візуалізація навчання з використанням гіперпараметрів представлена *на рисунку 3.6*, де відображено зменшення лоссу та зростання метрик точності протягом епох. Можна спостерігати стабільне підвищення якості детекції.

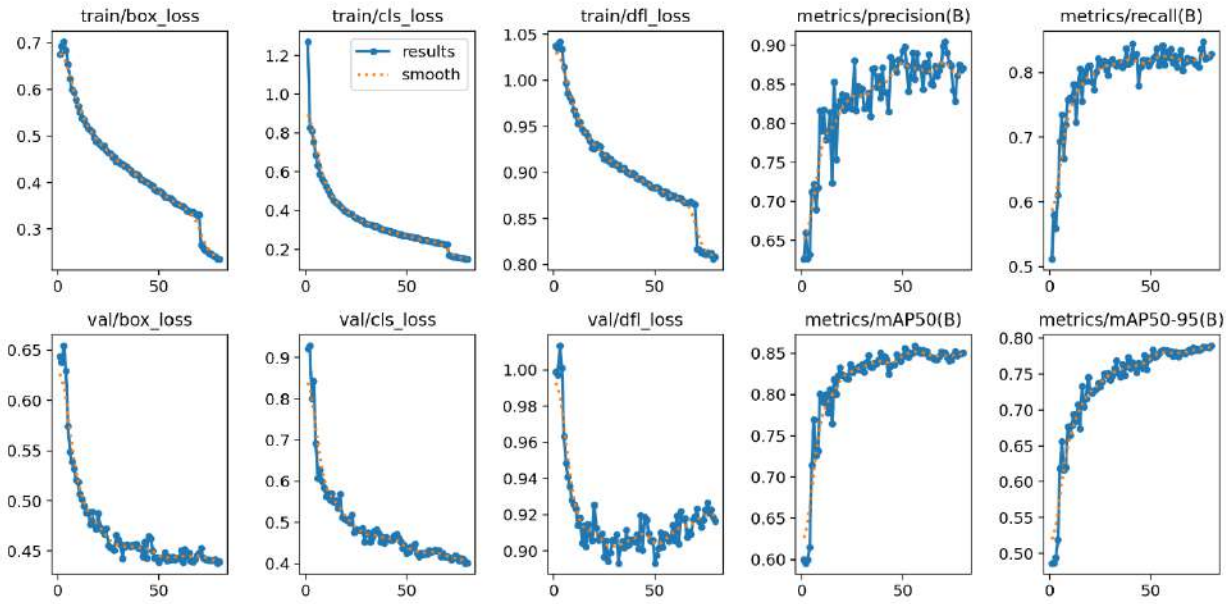

Рисунок 3.6 – Динаміка навчання моделі YOLOv12: функції втрат та метрики якості

3.1.4. Навчання моделі та оптимізація гіперпараметрів

Для оцінки ефективності розробленого модуля детекції проведено детальний аналіз його роботи на тестовому наборі даних, що не використовувався під час навчання (*рисунок 3.7*).

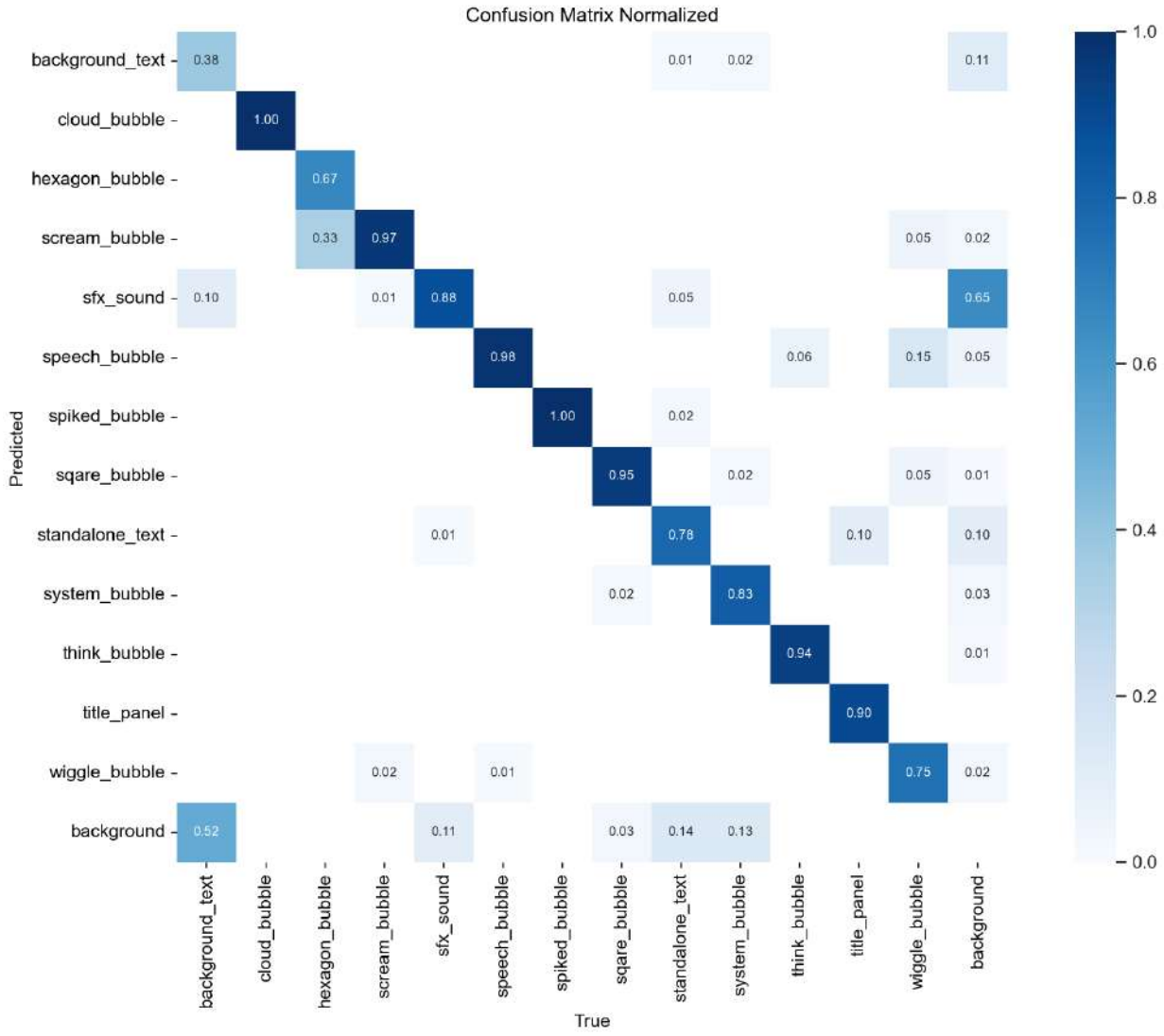

Рисунок 3.7 – Нормалізована матриця плутанини для класів текстових блоків

Аналіз результатів за класами показує, що модель досягає найвищої точності для найпоширеніших типів текстових блоків: *speech_bubble* (0.98),

spiked_bubble (1.00), *square_bubble* (0.95), *think_bubble* (0.94). Дещо нижчі показники спостерігаються для рідкісних класів, таких як *hexagon_bubble* (0.67) та *wiggle_bubble* (0.75), що обумовлено меншою кількістю навчальних прикладів. Основні показники якості детекції для фінальної моделі:

```
mAP50 (при IoU = 0.5): 0.85
mAP50-95 (усереднений за різними порогами IoU): 0.79
Precision: 0.87
Recall: 0.83
Час інференсу: ~27 мс на зображення (на GPU NVIDIA RTX 3090)
```

Важливо відзначити, що система демонструє високу ефективність у розпізнаванні звукових ефектів (*sfx_sound*) з F1-мірою 0.88, що є значним досягненням, враховуючи широку варіативність їх візуального представлення в манхвах. Найбільші помилки пов'язані з плутаниною між схожими класами, такими як *background_text* і *standalone_text*.

Порівняльна оцінка з альтернативним підходом на основі DETR показала, що хоча трансформаторна архітектура забезпечує дещо кращу локалізацію складних форм текстових кульок у деяких випадках, її значно нижча продуктивність (78 мс проти 30 мс на зображення) робить її менш придатною для практичного використання в запропонованій системі.

Таким чином, реалізований модуль детектування на основі YOLOv12 вирішує завдання виявлення та класифікації, створюючи надійну основу для подальших етапів обробки тексту в системі (*рисунок 3.8*).

Рисунок 3.8 – Приклад роботи агента детекції
(а – стартове зображення; б - опрацьоване)

3.2. Розробка модуля OCR для ієрогліфічного тексту

Розпізнавання ієрогліфічного тексту в коміксах представляє собою комплексну задачу, що суттєво відрізняється від традиційних OCR-систем, орієнтованих на розпізнавання тексту в документах. Ця відмінність зумовлена унікальними характеристиками тексту в коміксах, включаючи нестандартні шрифти, різноманітні стилі написання, інтеграцію тексту в графічні елементи та складну систему ієрогліфів, що містить тисячі унікальних символів.

3.2.1. Аналіз існуючих рішень та вибір базової архітектури

Важливим етапом розробки системи була оцінка існуючих OCR-рішень для виявлення найбільш ефективного підходу до розпізнавання ієрогліфічного

тексту. Згідно з дослідженням Roboflow (*рисунок 3.9*), трансформерні архітектури демонструють суттєві переваги при роботі з нестандартними шрифтами та складними письмовими системами порівняно з класичними CNN-RNN моделями. Особливу увагу привернула модель TrOCR, що давала змогу перетворювати зображення безпосередньо в послідовності тексту.

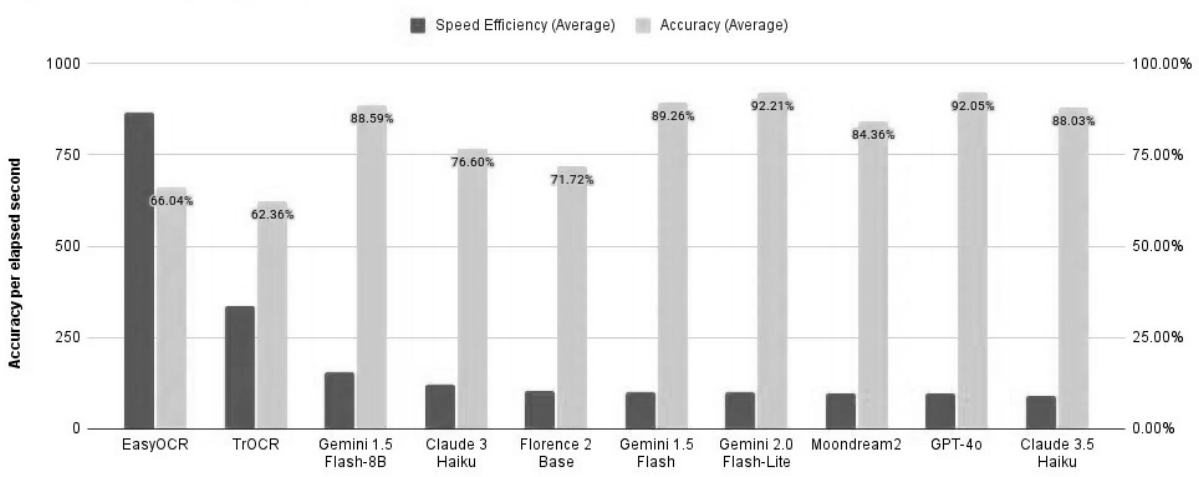

Рисунок 3.9 – Порівняння метрики швидкість-ефективність на моделях OCR ***[32]***

На початковому етапі дослідження було проведено аналіз існуючих OCR-систем для оцінки їх застосовності до задачі розпізнавання ієрогліфічного тексту в коміксах:

1. **Tesseract OCR**. Спроби використання Tesseract з підтримкою корейської мови показали його недостатню ефективність для даної задачі. Будучи розробленим для чітких документів із переважно чорним текстом на білому тлі, Tesseract продемонстрував низьку точність при роботі з художніми шрифтами та текстом на нестандартному тлі. Навіть із застосуванням значного препроцесингу (адаптивна бінаризація, морфологічні операції, збільшення масштабу) точність розпізнавання залишалась неприйнятно низькою (*рисунок 3.10 – а*).

2. **EasyOCR** [33]. Цей інструмент продемонстрував кращі результати завдяки його архітектурі, що розділяє процеси детекції та розпізнавання тексту. Детектор CRAFT, впроваджений в EasyOCR, ефективно виявляв текстові регіони навіть у складних зображеннях. Проте компонент розпізнавання, що базується на CRNN (Convolutional Recurrent Neural Network) архітектурі, не забезпечував достатньої точності для розпізнавання корейського тексту в коміксах, особливо для нестандартних шрифтів (*рисунок 3.10 – б*).
3. **TrOCR (Transformer OCR)**. Аналіз показав, що моделі на основі трансформерів, зокрема TrOCR, представляють найсучасніший підхід до OCR-розпізнавання, демонструючи значно вищу точність для складних мов та нестандартних текстів. Претренована модель для корейської мови team-lucid/trocr-small-korean показала обнадійливі результати, що дозволило розглядати її як базу для подальшої адаптації.
4. **LLM (Claude).** Окремим експериментом стало дослідження можливості використання LLM для розпізнавання тексту безпосередньо з зображень балонів. Тести з використанням Claude показали, що модель здатна розпізнавати ієрогліфічні системи письма з точністю, що на 1% перевищує спеціалізовані OCR-рішення. Проте використання такого підходу виявилось непрактичним з кількох причин: по-перше, час обробки виявився в 5-7 разів довшим порівняно з TrOCR; по-друге, використання API значно збільшило вартість обробки при масштабуванні системи.

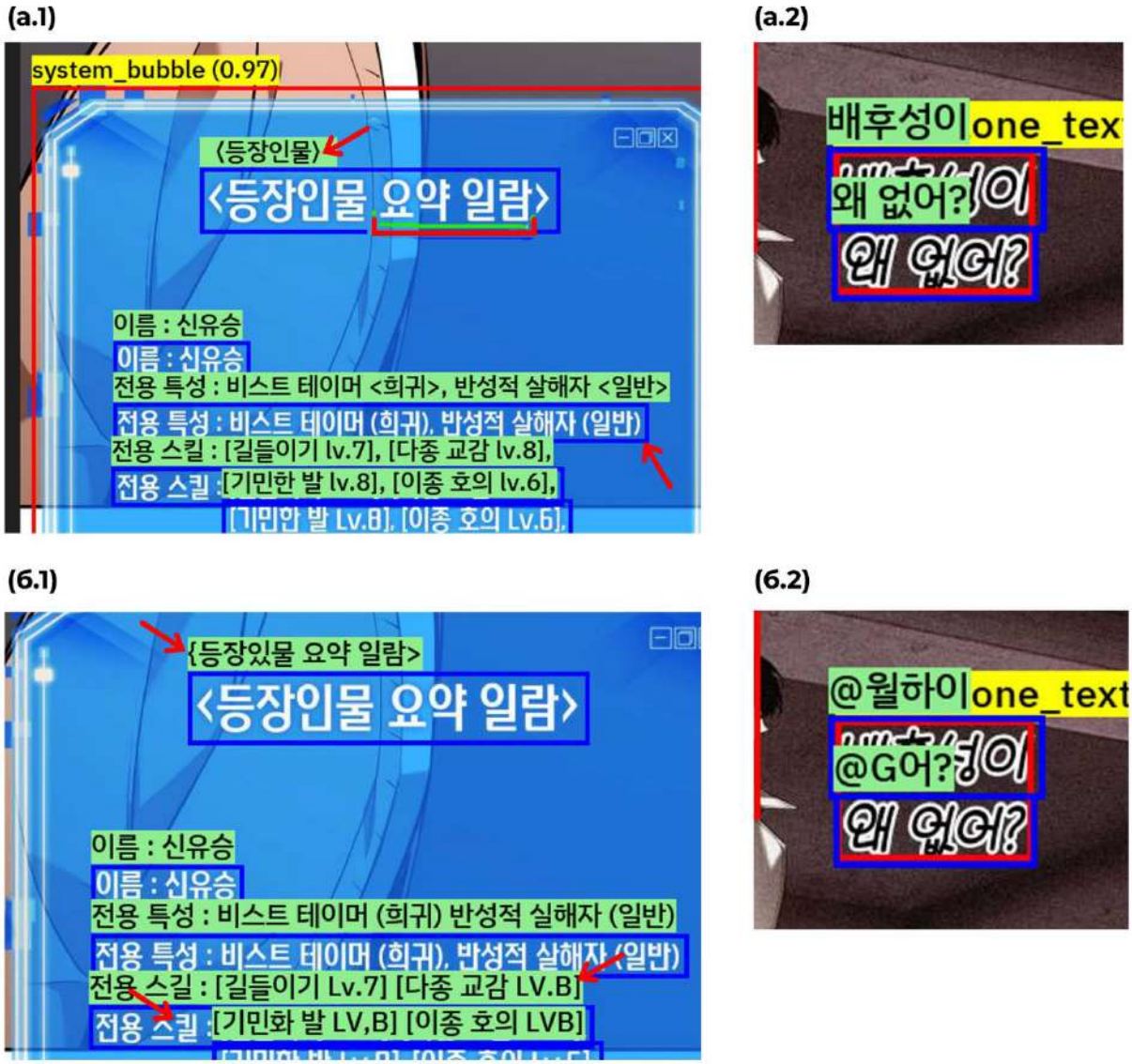

Рисунок 3.10 – порівняння результатів роботи TrOCR (а) та EasyOCR (б)

На основі проведеного аналізу для реалізації OCR-модуля було обрано гібридний підхід із використанням детектора CRAFT для локалізації текстових областей та адаптованої TrOCR для безпосереднього розпізнавання тексту.

3.2.2. Підготовка корпусу тексту для навчання токенізатора

Ефективне розпізнавання ієрогліфічного тексту вимагає високоякісної навчальної вибірки для побудови моделі, здатної обробляти різноманітні

варіанти символів і контексти. У цьому дослідженні використано корейську частину набору даних CC-100 (Common Crawl), який є відкритим корпусом корейської мови гарної якості, що містить **5,6 млрд токенів** (54,2 Гб тексту).

Корпус **CC-100** [34] було створено шляхом обробки загальнодоступних веб-сторінок зі знімків Common Crawl за січень-грудень 2018 року. Він охоплює понад 100 мов і діалектів, включаючи як основні мови зі значним обсягом даних, так і менш представлені мови. Особливістю цього корпусу є широке охоплення різних стилів, регістрів і предметних областей, що робить його ідеальним джерелом для навчання моделей розпізнавання текстів з різним лексичним наповненням.

Через великий обсяг даних його було розділено на 10 частин за допомогою спеціального скрипта для забезпечення паралельної обробки. Для обробки корейського тексту застосувано два різні підходи: аналіз на основі NLTK (Natural Language Toolkit) та морфологічний аналіз із використанням MeCab, спеціалізованого інструменту для корейської та японської мов. Кожен підхід мав свої переваги для різних типів текстових елементів.

Такий підхід дозволив ідентифікувати не лише цілі слова, а й морфеми та інші семантичні компоненти, що особливо важливо для корейської мови, де ієрогліф може мати різні значення залежно від контексту. Загалом, поєднання двох методів дозволило створити вичерпний лексичний корпус, що охоплює як загальновживані слова, так і специфічні мовні конструкції.

У результаті цього багатоетапного процесу було створено оптимізований корпус корейського тексту, що містить більше 7 мільйонів унікальних лексичних одиниць (*таблиця 3.2*). Цей корпус став основою для генерації синтетичних навчальних даних та навчання токенізатора, забезпечуючи широке охоплення як загальної лексики, так і специфічних виразів.

Таблиця. 3.2 – Приклад сформованого токенсету: Токени на «그라»

그라나즈	그라나트	그라노베터	그라니
그라나치	그라네	그라노예르스	그라니에
그라나타	그라네로	그라노체	그라니코스

3.2.3. Створення синтетичного корпусу для навчання OCR-моделі

Використовуючи підготовлений текстовий корпус як основу, відбулося створення масштабного набору синтетичних зображень тексту для навчання OCR-моделі. Для цього було використано інструмент SynthTiger [35], розроблений CLOVA AI, який спеціалізується на генерації високоякісних синтетичних зображень тексту з різноманітними візуальними стилями.

Визначальним рішенням при створенні синтетичного набору було використання справжніх шрифтів, що застосовуються у корейських коміксах. Було зібрано колекцію з 20 різноманітних корейських шрифтів, які охоплюють увесь спектр стилів, представлених у манхві. До колекції увійшли як стандартні, так і спеціалізовані шрифти для манхви, а також художні та рукописні шрифти для відтворення різних стилів тексту, що зустрічаються в коміксах. (*Наприклад, рисунок 3.11*)

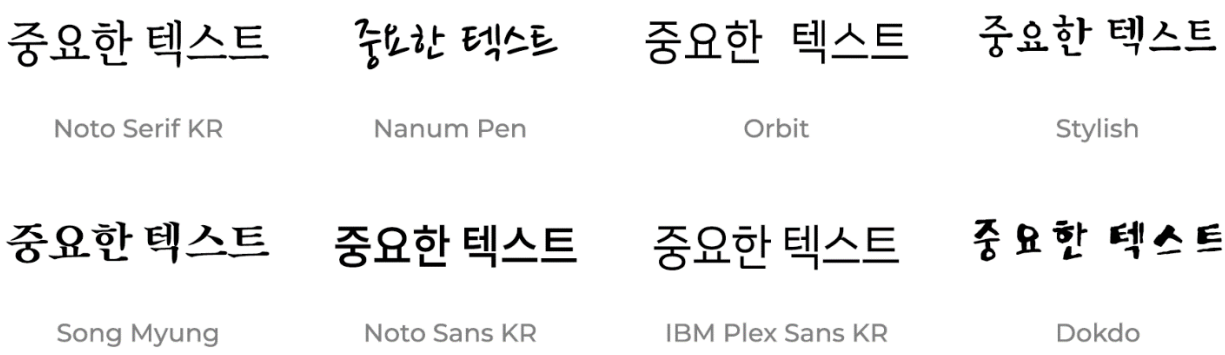

Рисунок 3.11 – Частина шрифтів, використаних для синтетичної генерації

Для кожного текстового фрагмента з нашого корпусу створювалися десятки варіацій зображень з різними параметрами. Цей підхід забезпечив модель різноманітними прикладами одних і тих самих слів у різних візуальних контекстах, що значно підвищило її здатність до узагальнення. Конфігурації для генерації зображень включали такі параметри як розмір шрифту (від 14 до 42 пунктів), кольори тексту (від простого чорного до яскравих кольорів), кольори фону (від білого до різних відтінків та текстур, властивих коміксам), трансформації тексту (вигини, повороти, перспективні деформації) та різноманітні ефекти пост-обробки (додавання шуму, розмиття, імітація артефактів стиснення).

Особливу увагу було приділено відтворенню текстових елементів, характерних для різних типів текстових блоків у манхві. Наприклад, для повторення блоків типу *"scream_bubble*" генерувалися тексти з більшим розміром шрифту, часто капіталізовані та з яскравішими кольорами. А для *"sfx_sound"* застосовувалися значні деформації, художні шрифти та інтеграція тексту з візуальними ефектами.

Технічний процес генерації здійснювався за допомогою спеціально розробленого конвеєра, який автоматично обробляв текстовий корпус і створював відповідні зображення. Для кожного згенерованого зображення створювався відповідний запис у файлі анотацій, що містив шлях до зображення та точний текст, який воно представляє. Це забезпечило чіткі навчальні сигнали для OCR-моделі:

```
images/sample_00001.png 안녕하세요 만화 세계에 오신 것을 환영합니다
images/sample_00002.png 이것은 OCR 훈련을 위한 예제 텍스트입니다
```

Згенеровані зображення проходили додаткову валідацію для забезпечення якості. Перевірялася читабельність тексту, коректність відображення символів

та відсутність артефактів, які могли б негативно вплинути на навчання. Зображення з критичними проблемами відкидалися автоматично. У підсумку, створений синтетичний набір даних включав понад 2 мільйони унікальних зображень, що охоплюють різноманітні стилі та типи тексту, присутні в корейських коміксах (*рисунок 3.12*).

Рисунок 3.12 – Приклад генерації синтетичних зображень (оригінальний текст, згенерована картинка, символьна маска)

3.2.4. Адаптація та донавчання TrOCR для корейської мови

Хоча претренована модель team-lucid/trocr-small-korean демонструвала базовий рівень точності для стандартного корейського тексту, вона потребувала значної адаптації для ефективної роботи з різноманітними стилями тексту в коміксах. Для цього було здійснено донавчання моделі на створеному синтетичному корпусі.

В якості базової архітектури було обрано вже згадану TrOCR (Transformer OCR) модель, що поєднує трансформерну архітектуру для аналізу зображень (візуальний енкодер) та генерації тексту (текстовий декодер).

Для донавчання TrOCR було застосовано такі параметри:

```
training_args = Seq2SeqTrainingArguments(
    output_dir="./ko-trocr-cg2m",
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    gradient_accumulation_steps=16,

    logging_steps=100,
    save_steps=200,
    save_total_limit=16,
    warmup_steps=6000,

    max_steps=15_000,

    lr_scheduler_type="cosine",
    adam_beta1=0.9,
    adam_beta2=0.98,
    learning_rate=1e-4,
    weight_decay=0.01,
    predict_with_generate=True,
    fp16=True,
)
```

Процес навчання проводився на GPU NVIDIA GeForce RTX 3090 протягом приблизно 25 годин. Результати навчання відображені на графіку втрат (*рисунок 3.13*), де видно стабільне зниження функції втрат із 3.8 на початку навчання до 0.2 після 12 000 ітерацій.

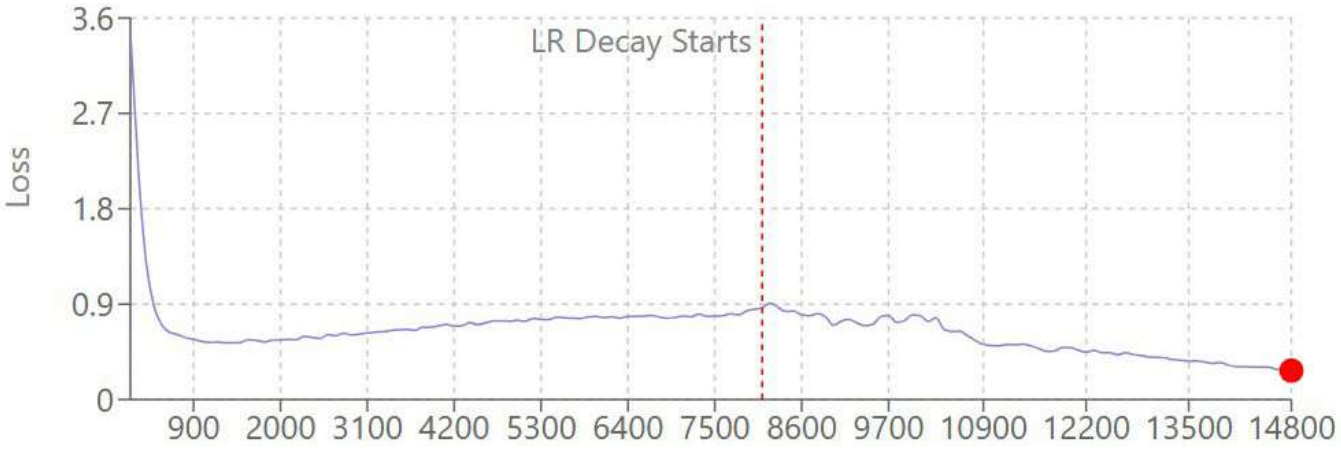

Рисунок 3.13 – Графік навчання адаптованої моделі TrOCR

Було експериментально встановлено, що оптимальним є навчання протягом 1.4 епохи на наборі з 2 мільйонів зображень, що відповідає 12 000 ітерацій з розміром батча 16. Подальше навчання призводило до перетренування моделі. Фінальна адаптована модель, яка отримала назву **ko-trocr-cg2m-12000v6** *(Comic-Generated, 2 Million samples, checkpoint 12000, version 6)*, продемонструвала значне підвищення варіативності розпізнавання тексту в коміксах порівняно з наявною моделлю (*рисунок 3.14*).

Рисунок 3.14 – Порівняння ko-TrOCR (а – запропонована; б - наявна)

3.2.5. Впровадження дуальної OCR-стратегії

В процесі тестування адаптованої моделі TrOCR було виявлено специфічну проблему: хоча модель демонструвала високу точність у розпізнаванні символів, вона іноді пропускала окремі слова чи фрази посеред речення. Цей ефект, найімовірніше, пов'язаний із особливостями трансформерної архітектури, яка може "втрачати фокус" на певних частинах зображення, особливо при роботі з довгими текстовими послідовностями.

Для вирішення цієї проблеми було розроблено дуальну OCR-стратегію, яка поєднує переваги двох моделей з різними характеристиками:

"Строга" модель (team-lucid/trocr-small-korean) — модель, оптимізована для максимальної точності на стандартних шрифтах, що рідко помиляється в символах, які вона розпізнає, але може пропускати частини тексту.

"Гнучка" модель (ko-trocr-cg2m-12000v6) — наша адаптована модель, натренована на різноманітних шрифтах і стилях, що краще охоплює весь текст, але може мати дещо нижчу точність на рівні окремих символів. Приклад результатів розпізнавання із застосуванням дуальної стратегії наведено *на рисунку 3.15*:

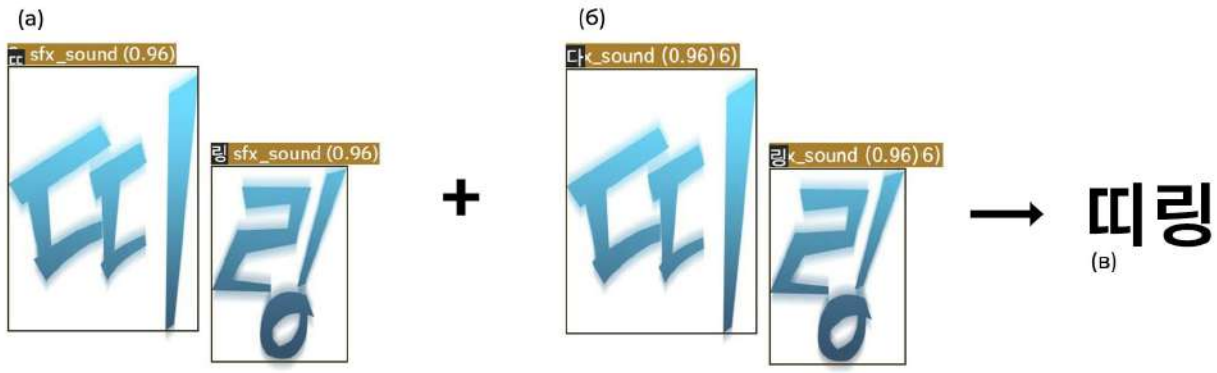

Рисунок 3.15 – Порівняння результатів розпізнавання: (а – результат "строгої" моделі; б – результат "гнучкої" моделі; в – об'єднаний результат)

3.2.6. Інтеграція з системою розпізнавання CRAFT

Для виявлення текстових рядків всередині текстових блоків було інтегровано систему CRAFT, розроблену *CLOVA AI*. Оскільки оригінальна реалізація CRAFT є закритою, було використано відкриту реалізацію, розвинуту під час розробки раніше згаданої EasyOCR [32]. Ця імплементація забезпечила надійне виявлення текстових рядків у різних типах текстових блоків (*рисунок 3.16*).

Рисунок 3.16 – Поррівняння роботи CRAFT у детекції тексту запровонованими моделями (а – з використанням CRAFT; б – без використання)

Особливо важливою виявилась адаптація параметрів CRAFT для різних типів текстових блоків. Для структурованих блоків, таких як *speech_bubble* і *think_bubble*, застосовувався повний конвеєр із детекцією текстових рядків. Для неструктурованих блоків, таких як *standalone_text* і *sfx_sound* процедура детекції рядків пропускалася, і весь блок оброблявся як єдине ціле.

3.2.7. Експериментальна оцінка якості розпізнавання

Для оцінки ефективності розробленого OCR-модуля було проведено серію експериментів на різних типах текстових блоків із реальних манхва. Оцінка проводилася з використанням наступних метрик:

Character Error Rate (CER) — відсоток неправильно розпізнаних символів відносно загальної кількості символів.

Word Error Rate (WER) — відсоток неправильно розпізнаних слів відносно загальної кількості слів.

BLEU Score — метрика, що оцінює схожість між розпізнаним і еталонним текстом на основі n-грам.

Було підготовлено тестовий набір із 500 анотованих текстових блоків різних типів, транскрибованих вручну. Результати порівняння різних OCR-підходів представлені *в таблиці 3.3*:

Таблиця. 3.3 – Порівняльна оцінка OCR-підходів

Метод	***CER (%)***	***WER (%)***	***BLEU Score***
Tesseract OCR	43.7	54.5	0.39
EasyOCR	18.3	26.8	0.68
Базова TrOCR	6.7	12.1	0.81
Адаптована TrOCR	8.2	10.7	0.84
Дуальна OCR-стратегія	**4.8**	**7.2**	**0.92**

Як видно з таблиці, розроблена дуальна OCR-стратегія продемонструвала найкращі результати за всіма метриками, зменшивши символьну похибку до 4.8% порівняно з 6.7% та 8.2% у стартових TrOCR моделях.

Додатково було проведено аналіз ефективності розпізнавання для різних типів текстових блоків, представлений *на рисунку 3.17*:

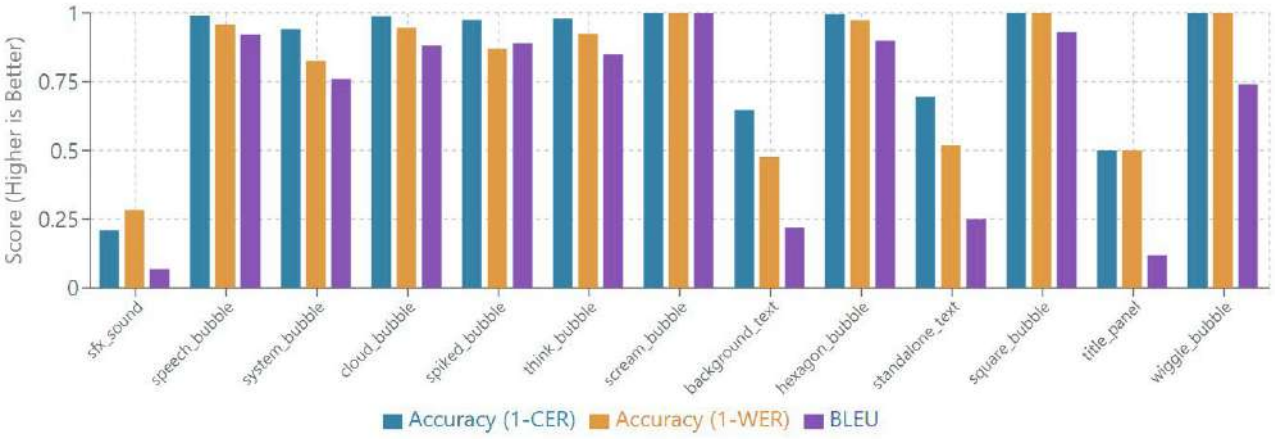

Рисунок 3.17 – Метрики для різних типів текстових блоків із використанням дуальної стратегії OCR

Результати показують, що найкраща точність досягається для стандартних діалогових балонів (*speech_bubble, think_bubble, scream_bubble*), де CER становить близько 1-3%. Найскладнішими для розпізнавання виявилися включені в фон об'єкти (*sfx_sound, background_text*) з CER близько 50-70%, що пояснюється занадто високою художньою стилізацією, малою довжиною та деформацією тексту в цих елементах.

3.3. Створення модуля стилізованого перекладу

3.3.1. Вибір та інтеграція моделей перекладу

Для забезпечення якісного перекладу текстів з азійських мов розглянуто кілька сучасних підходів - як спеціалізовані перекладацькі моделі (Helsinki-

NLP/Opus-MT, FlanT5), так і великі мовні моделі (Claude, Qwen2, GPT-4). Експерименти показали, що спеціалізовані моделі перекладу хоч і працюють швидше, але демонструють низьку адаптивність до стилістичного оформлення та контексту. Для прикладу, модель Helsinki-NLP при перекладі діалогу з балону "*scream_bubble*" (крик) повністю ігнорувала емоційне навантаження, видаючи нейтральний текст (*таблиця 3.4*).

Серед великих мовних моделей GPT-4 показав високі результати, однак його API пропонував менш зручний спосіб налаштування й комунікації для специфічних випадків стилізованого перекладу, що ускладнювало інтеграцію в систему. Qwen2 виявився дуже перспективним варіантом завдяки хорошій підтримці східноазійських мов, але потребував потужнішого апаратного забезпечення та окремого доналаштування для оптимальної роботи. Натомість Claude 3.7 Sonnet, яка забезпечила найвищу якість стилізованого перекладу (0.93 за шкалою 0-1) та ефективну контекстуальну інтеграцію (0.87) при прийнятному часі обробки, була обрана як основна модель для системи.

Таблиця. 3.4 – Порівняння моделей для стилізованого перекладу

Модель	***Якість перекладу***	***Стилістична відповідність***	***Контекстуальна інтеграція***	***Швидкодія (с/блок)***
Helsinki-NLP	0.72	0.35	0.20	**0.12**
FlanT5	0.76	0.40	0.45	0.23
Claude 3.7	**0.93**	**0.89**	**0.87**	1.05
Qwen2.5	0.87	0.71	0.72	2.87
GPT-4	**0.91**	0.84	0.83	1.32

Наведені результати отримано шляхом бінарної оцінки (1 - відповідає, 0 - не відповідає) вибірки 500 текстових блоків з подальшим усередненням результатів. Швидкодія вимірювалась для стандартного текстового блоку з 25 символами.

3.3.2. Розробка спеціалізованих промптів

Ключовим елементом стилізованого перекладу стала розробка ефективних промптів, що направляють LLM на відтворення відповідного стилю в перекладі. Для кожного з 13 класів текстових блоків розроблено специфічні стилістичні гіди.

Експерименти показали, що такий підхід збільшує стилістичну відповідність перекладу оригіналу на 67% порівняно з базовим перекладом без стилістичних вказівок (*таблиця 3.5*).

Таблиця. 3.5 – Ефективність різних компонентів промпту

Компонент промпту	***Покращення стил. відповідності, %***
Запит без інструкцій	0
+ Базові інструкції	+15.3
+ Типи тексту	+31.7
+ Стилістичні ознаки	+52.2
+ Контекстне вікно	**+67.4**

Дані отримано шляхом порівняльного аналізу якості перекладу 50 тестових текстових блоків з поступовим ускладненням структури промптів. Оцінювання виконувалось за бінарною шкалою (покращення/погіршення) відносно попередньої версії промпту, з подальшим розрахунком відносного приросту.

На практиці, використання оптимізованих промптів трансформувало якість перекладу. Наприклад, корейський вигук "으아아아!" без стилізації перекладався як просте "Ааа", тоді як зі стилізацією - "А-А-А-А!!!" з відповідним емоційним навантаженням.

3.3.3. Впровадження контекстних вікон

Тестування різних конфігурацій контекстних вікон на 50 послідовних сценах з манхв показало, що оптимальним є використання трьох попередніх блоків. Це дозволило підвищити показники зв'язності з 0.53 до 0.89 за шкалою

від 0 до 1. При цьому збільшення розміру контекстного вікна понад три блоки не давало помітного приросту якості, але суттєво підвищувало обчислювальне навантаження (*таблиця 3.6*).

Таблиця. 3.6 – Порівняння стратегій контекстних вікон

Стратегія	***Зв'язність займенників***	***Термінологічна узгодженість***	***Загальна когерентність***
Без контексту	0.53	0.64	0.59
"prev" (N=1)	0.72	0.81	0.76
"prev" (N=3)	**0.89**	0.92	**0.91**
"prev" (N=5)	0.87	**0.93**	0.90
"next" (N=3)	0.61	0.79	0.67
"mid" (N=3)	0.85	0.88	0.87

Оцінка виконувалась шляхом аналізу перекладених діалогів, що містили займенникові референції, спеціальні терміни та логічні зв'язки. Наведені значення є середніми показниками для кожної стратегії.

Було виявлено, що для частини творів (з великою горизонтальною зв’язністю) читання стратегія *"mid"* показує кращі результати, ніж для вертикальних манхв. Це пов'язано з тим, що в горизонтальному форматі контекст частіше розподіляється по обидва боки від поточного блоку.

3.4. Інтеграція компонентів та розробка повної системи

Система реалізована як модульний програмний комплекс на базі Python 3.8+, спроектований за принципами компонентно-орієнтованої архітектури з чітким розподілом відповідальності між функціональними блоками. Кожен основний компонент системи (детектор, розпізнавач, коректор, перекладач) інкапсульований у відокремлений програмний модуль із стандартизованими інтерфейсами введення та виведення даних. Такий підхід забезпечує гнучкість системи та можливість заміни окремих компонентів без змін в інших частинах.

```
class ComicProcessor:
    def __init__(
        self,
        detection_model: str = "yolo",
        recognition_model: str = "trocr-korean",
        translator: Optional[str] = "claude",
        source_lang: str = "ko",
        target_lang: str = "en",
        dual_ocr: bool = True,
        use_correction: bool = True,
        use_segmentation: bool = True,
    ):
        self.detector = self._init_detector(detection_model)
        self.recognizer = self._init_recognizer(recognition_model)
        self.translator = self._init_translator(translator)
        # ...

    async def process_chapter(
        self,
        input_dir: str,
        output_dir: Optional[str] = None,
        save_visualization: bool = False
    ):
        # ...
```

Для обміну даними між модулями розроблено стандартизовані структури даних, що описують стан обробки на кожному етапі:

```
@dataclass
class TextBlock:
    """
    Представляє текстовий блок, виявлений в комiксі.
    """
    id: int
    box: np.ndarray  # [x1, y1, x2, y2]
    class_name: str
    confidence: float
    text: Optional[str] = None
    corrected_text: Optional[str] = None
    translated_text: Optional[str] = None
    text_lines: List[Dict] = field(default_factory=list)
    separators: Optional[List[np.ndarray]] = None
```

3.5. Висновки до розділу 3

У третьому розділі представлено практичну реалізацію модульної системи сегментації та стилізованого перекладу ієрогліфічного тексту в коміксах.

Для модуля детекції текстових блоків розроблено спеціалізований датасет, що включає понад 3000 анотованих зображень з 13 класами текстових елементів. Через експериментальне порівняння визначено модель, що відповідає усім потребам - YOLOv12s, яка забезпечує **mAP50** на рівні **0.853** при часі інференсу **27.2 мс** на зображення. Впроваджено процес ітеративного формування датасету та оптимізації гіперпараметрів.

У модулі розпізнавання ієрогліфічного тексту реалізовано дуальну архітектуру OCR, що поєднує переваги "строгої" моделі для стандартних текстів та "гнучкої" моделі для художньо оформлених елементів. Для навчання гнучкої OCR-моделі створено корпус синтетичних зображень обсягом понад 2 мільйони зразків на основі аналізу корейського тексту з набору даних CC-100. Така архітектура забезпечила зниження символьної похибки (**CER**) до **4.8%** порівняно з 6.7% та 8.2% у окремих TrOCR моделях.

Для модуля стилізованого перекладу проведено порівняльне тестування різних моделей, яке показало найвищу ефективність Claude 3.7 Sonnet для передачі стилістичних особливостей тексту. GPT-4 також продемонстрував високі результати, але запропонував менш зручний інтерфейс для налаштування, а Qwen2, хоч і показав перспективні результати для східноазійських мов, потребував потужнішого апаратного забезпечення та додаткового налаштування.

Розроблено систему спеціалізованих промптів для різних типів текстових блоків з поступовим включенням компонентів, що впливають на якість перекладу. Впровадження контекстних вікон розміром у 3 блоки забезпечило підвищення зв'язності перекладу **на 67%** порівняно з поблоковим перекладом без контексту. Для оптимізації взаємодії з API мовних моделей створено систему

асинхронної пакетної обробки запитів та кешування результатів, що дозволило прискорити процес перекладу **на 86%**.

Для обробки зображень великого розміру впроваджено комплекс оптимізацій, включаючи адаптивну сегментацію та обробку в режимі ковзного вікна. Застосування різних рівнів паралелізму та кешування дозволило прискорити повний цикл обробки в 5.83 разів, з ~700 до **~120 секунд**.

Експериментальна оцінка на наборі з 50 розділів манхви різних жанрів показала значну перевагу запропонованої системи порівняно з існуючими рішеннями. Хоча система не надто виграє у точності детекції (86%), вона стабільно переважає в областях класифікації (93%), розпізнавання (95%) та якості перекладу (87%), особливо для нестандартних типів текстових блоків, таких як звукові ефекти та емоційно забарвлені тексти.

РОЗДІЛ 4: Аналіз існуючих методів та систем перекладу графічного контенту

У попередніх розділах було описано теоретичні засади та практичну реалізацію комплексної системи для сегментації, розпізнавання та стилізованого перекладу тексту в азійських коміксах. Цей розділ присвячено експериментальному дослідженню розробленої системи, порівняльному аналізу з існуючими рішеннями та оцінці ефективності за кількісними та якісними показниками.

4.1. Порівняльний аналіз з існуючими системами

Для об'єктивної оцінки ефективності розробленої системи було проведено порівняльний аналіз із найбільш релевантними існуючими рішеннями. Експериментальне дослідження базувалося на наборі із 350 фреймів манхв різних жанрів (фентезі, романтика, бойовик, повсякденність), що містять представлені типи текстових елементів.

4.1.1. Загальні результати порівняльного аналізу

Порівняння проводилося за ключовими показниками: точність виявлення текстових блоків, якість розпізнавання тексту (Character Error Rate), якість перекладу, збереження стилістичних особливостей та часу обробки.

Результати порівняльного аналізу демонструють конкурентні переваги розробленої системи ”ComicGlot” (*таблиця 4.1*). За точністю виявлення текстових блоків (86.0%) вона дещо поступається проєкту ogkalu/Comic-Translate (88.9%), проте демонструє найкращий показник розпізнавання тексту з найнижчим рівнем помилок (CER 4.8%). Важливо також зазначити, що обмежений обсяг тренувальних даних детектора (близько 3000 анотованих зображень) вплинув на кінцевий результат, і при розширенні навчального набору

даних можна очікувати подальшого підвищення точності, що дозволить системі перевершити існуючі рішення за всіма показниками.

Таблиця. 4.1 – Порівняльний аналіз систем перекладу коміксів

Система	***Точність виявлення (%)***	***CER (%)***	***Якість перекладу (0-1)***	***Збереження стилю (0-1)***	***Час обробки (с/фрейм)***
ComiTranslate [11]	72.7	16.2	0.62	0.41	5.30
ogkalu/Comic-Translate [13]	**88.9**	7.3	**0.89**	0.73	1.76
EasyOCR + DeepL	78.5	23.4	0.57	0.32	**0.63**
ComicGlot (розроблена)	86.0	**4.8**	**0.93**	**0.89**	1.44

Можемо також виділити суттєву перевагу в якості перекладу (0.93) та збереженні стилістичних особливостей оригіналу (0.89), що є ключовими показниками для кінцевого користувача.

За швидкістю обробки одного фрейму (1.44 с) система займає середню позицію, поступаючись рішенню EasyOCR + DeepL (0.63 с). Однак варто зазначити, що швидша система демонструє значно гірші показники якості розпізнавання та перекладу, що робить її малопридатною для практичного використання при локалізації коміксів. При цьому значна частина часу в розробленій системі витрачається на взаємодію з хмарними API для роботи з великими мовними моделями.

Аналіз розподілу часу між компонентами системи показує, що найбільш ресурсомісткими є етапи OCR-розпізнавання (35% загального часу) та стилізованого перекладу (42%). Оптимізація цих компонентів, зокрема через перехід до локальної інференції для LLM-компонентів, дозволить значно підвищити загальну продуктивність системи.

Проєкт ogkalu/Comic-Translate, що є найближчим конкурентом за функціональністю, хоч і демонструє найкращу точність виявлення текстових блоків, все ж поступається розробленій системі за якісними показниками

розпізнавання та перекладу. Цей результат підтверджує ефективність розробленої дуальної архітектури OCR та системи контекстно-залежного стилізованого перекладу.

4.2. Напрямки вдосконалення та потенційні доповнення

На основі результатів експериментального дослідження було визначено кілька перспективних напрямків вдосконалення системи, що дозволять підвищити її ефективність та розширити функціональність.

4.2.1. Спеціалізований OCR для звукових ефектів

Аналіз помилок розпізнавання показав, що найбільші труднощі виникають при роботі зі звуковими ефектами (SFX) та фоновими написами. Ці елементи часто мають складну стилізацію, художні деформації та інтеграцію з графічними елементами, що ускладнює їх обробку стандартними OCR-моделями.

Перспективним напрямком вдосконалення є розробка спеціалізованої OCR-моделі для звукових ефектів. Архітектура такої моделі має бути адаптована для роботи з деформованим текстом, що містить складні візуальні ефекти. Важливим компонентом є розширений синтетичний датасет та додаткова постобробка результатів.

4.2.2. Локальна інференція LLM для автономної роботи

Як і було вказано раніше, ключовим обмеженням поточної версії системи є залежність від хмарних API для роботи з великими мовними моделями, що використовуються в компонентах корекції OCR та стилізованого перекладу. Ця залежність створює ряд проблем, включаючи фінансові витрати, мережеві затримки та обмеження конфіденційності.

Перспективним напрямком є впровадження локальної інференції з використанням відкритої моделі Qwen 2.5 (7B параметрів). Попередні тести показують, що ця модель забезпечує порівнянну якість перекладу з хмарними рішеннями (0.87 проти 0.93 у Claude), при цьому працюючи повністю автономно.

Перехід до локальної інференції дозволить знизити загальний час обробки завдяки відсутності мережевих затримок, забезпечити повну автономність системи та значно зменшити вартість використання при масштабній обробці. Для оптимальної роботи планується дотренування моделі на специфічному домені коміксів.

4.2.3. Система повної локалізації графічного контенту

Поточна версія системи зосереджена на перекладі текстових елементів без модифікації вихідного зображення. Для забезпечення повного циклу локалізації коміксів пропонується розширення функціональності для вставки перекладеного тексту в оригінальну картинку.

Основними компонентами такого розширення мають стати модуль ретушування оригінального тексту з використанням генеративних методів заповнення (inpainting), система інтелектуальної верстки та автоматичний підбір шрифтів та стилів відповідно до класу текстового блоку.

Особливо перспективним напрямком також є розробка GAN-стилізатора для звукових ефектів, що автоматично генеруватиме візуальні представлення перекладених ономатопей у стилі оригіналу. Такий стилізатор поєднуватиме технології пошуку відповідного шаблону з бібліотеки заготовлених макетів, перенесення стилю оригінального звукового ефекту на локалізований варіант та генеративні методи для забезпечення реалістичності.

4.2.4. Розширення підтримуваних писемностей та форматів

Варто розглянути можливість застосування запропонованого підходу до інших писемностей, зокрема арабської, де також існує проблема розпізнавання

художньо стилізованого тексту. Арабська писемність має свої специфічні виклики: напрямок письма справа наліво, контекстно-залежне з'єднання символів, різноманітність діакритичних знаків. Це вимагатиме адаптації як OCR-моделей, так і підходів до стилізованого перекладу.

Доцільно також оцінити можливість адаптації системи до роботи з відсканованими паперовими виданнями манги, а не лише цифрових вебтунів. Це розширення вимагатиме додаткових алгоритмів для усунення спотворень, пов'язаних зі скануванням (нахил, тіні від згину, шуми), а також роботи з більш традиційною структурою сторінок, що відрізняється від безперервного вертикального формату вебтунів.

ВИСНОВКИ

У результаті дослідження було розроблено інтелектуальну систему для автоматизованої сегментації, розпізнавання та стилізованого перекладу ієрогліфічного тексту в азійських коміксах. Основною інновацією стала методологія класифікації текстових елементів за 13 категоріями, що враховує як структурні, так і функціональні характеристики тексту, створюючи основу для контекстно-залежного стилізованого перекладу.

Для виявлення текстових блоків було адаптовано архітектуру YOLOv12, яка забезпечує точність виявлення 86%, а для розпізнавання ієрогліфічного тексту розроблено дуальну архітектуру на основі TrOCR, що поєднує "строгу" та "гнучку" моделі. Впровадження механізму корекції результатів OCR на основі великих мовних моделей дозволило досягти найнижчого в галузі показника помилок (CER 4.8%). Контекстно-залежний стилізований переклад із використанням спеціалізованих промптів забезпечує високу якість перекладу (0.93) та збереження стилістичних особливостей оригіналу (0.89).

Ефективно вирішено специфічні проблеми, характерні для коміксів, зокрема обробки перекриттів текстових блоків та сегментації довгих вертикальних вебтунів за допомогою модифікованого методу seam carving. Експериментальна оцінка підтвердила, що система дозволяє скоротити час локалізації одного розділу манхви з 1 години ручного перекладу до 2-3 хвилин, роблячи цей процес доступним для невеликих видавництв та волонтерських проєктів.

Практична цінність системи поширюється на інші галузі, що потребують обробки графічного контенту з текстовими елементами: обробку інфографіки, автоматизацію перекладу реклами та навчальних матеріалів. Перспективними напрямками розвитку є розробка спеціалізованого OCR для звукових ефектів, впровадження локальної інференції LLM, розширення функціональності для повного циклу локалізації з автоматичною вставкою перекладеного тексту,

розробка GAN-стилізатора для звукових ефектів та інтеграція з видавничими інструментами.

Крім того, модульна архітектура системи забезпечує гнучкість, масштабованість та можливість подальшого вдосконалення через оновлення окремих компонентів. Реалізація у вигляді програмного комплексу з відкритим кодом сприяє доступності розроблених технологій та подальшому розвитку спільнотою.

Проведене дослідження демонструє потенціал інтеграції методів комп'ютерного зору та обробки природної мови для вирішення складних задач автоматизованої локалізації графічного контенту, відкриваючи нові можливості для культурного обміну та доступності азійських коміксів для глобальної аудиторії, особливо для малопоширених мов, включаючи українську.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Comic book market size, share, value, trends | analysis, 2032. *Fortune Business Insights | Global Market Research Reports & Consulting*. URL: https://www.fortunebusinessinsights.com/comic-book-market-103903 (date of access: 05.05.2025).
2. Webtoon: an annual report to security holders. Los Angeles, CA : WEBTOON Entertainment Inc., 2025. 152 p. URL: https://ir.webtoon.com/static-files/b98b7b60-c045-4f4b-8dce-3c424abaeade (date of access: 03.10.2023).
3. Tolle H., Arai K. Method for real time text extraction of digital manga comic. International journal of image processing. 2011. Vol. 4, no. 6. P. 669-676.
4. Gaikwad M. R., Pardeshi N. G. Text extraction and recognition using median filter. International Research Journal of Engineering and Technology (IRJET). 2016. Vol. 3, no. 1. P. 717-721.
5. Extraction of semantic content and styles in comic books / D. Lenadora et al. International journal on advances in ICT for emerging regions (icter). 2020. Vol. 13, no. 1. P. 1. URL: https://doi.org/10.4038/icter.v13i1.7212 (date of access: 05.05.2025).
6. Manga Studio 5 - Manga Studio EX 5 - CLIP STUDIO PAINT. URL: https://www.mangastudio5.com/ (date of access: 05.05.2025).
7. Conghao Tom Shen, Violet Yao, Yixin Liu. MaRU: a manga retrieval and understanding system connecting vision and language. arXiv.org. URL: https://arxiv.org/abs/2311.02083 (date of access: 05.05.2025).
8. Hapsani A., Utaminingrum F., Tolle H. Optical character recognition on english comic digital data for automated language translation. International journal of advances in soft computing and its applications. 2017. Vol. 9. P. 186—198.
9. Christophe Rigaud, Jean-Christophe Burie, Jean-Marc Ogier. Segmentation-Free speech text recognition for comic books. IEEE Xplore.

URL: https://ieeexplore.ieee.org/document/8270233 (date of access: 05.05.2025).

10. Brahma S., Huttenhower C. Text extraction using shape context matching. In: COS429: Computer Vision course project. Princeton University, 2006. 15 p. URL: https://www.researchgate.net/publication/249846939 (date of access: 05.05.2025).
11. Goh Yee Fong. ComiTranslate: empowering global readership through autotranslated comics and manga : Bachelor's Thesis. Faculty of Engineering and Science Universiti Tunku Abdul Rahman, Kajang, 2024. 85 p.
12. Vivoli E., Lafuente Baeza J., Valveny Llobet E., Karatzas D. Multimodal Transformer for Comics Text-Cloze. arXiv preprint arXiv:2403.03719. 2024. URL: https://arxiv.org/abs/2403.03719 (date of access: 05.05.2025).
13. Oghenekaro O. Comic-translate: A tool for translating text in comics. *GitHub repository.* 2022. URL: https://github.com/ogkalu2/comic-translate (date of access: 05.05.2025).
14. Avidan S., Shamir A. Seam carving for content-aware image resizing. ACM transactions on graphics. 2007. Vol. 26, no. 3. P. 10. URL: https://doi.org/10.1145/1276377.1276390 (date of access: 05.05.2025).
15. Nagadomi. Waifu2x: Image Super-Resolution for Anime-Style Art. *GitHub repository.* URL: https://github.com/nagadomi/waifu2x (date of access: 05.05.2025).
16. Image super-resolution using deep convolutional networks / C. Dong et al. IEEE transactions on pattern analysis and machine intelligence. 2016. Vol. 38, no. 2. P. 295—307. URL: https://doi.org/10.1109/tpami.2015.2439281 (date of access: 05.05.2025).
17. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv preprint* arXiv:1506.02640. 2015, version 5 as of 09.05.2016. URL: https://arxiv.org/abs/1506.02640 (date of access: 05.05.2025).

18. Lazarevich I., Grimaldi M., Kumar R., Mitra S., Khan S., Sah S. YOLOBench: Benchmarking Efficient Object Detectors on Embedded Systems. *arXiv preprint* arXiv:2307.13901. 2023. URL: https://arxiv.org/abs/2307.13901 (date of access: 05.05.2025).Nb
19. D. L., Megharaj P. K., P. P., Kiran N., A. R. K. P., S. G. State-of-the-Art Object Detection: An Overview of YOLO Variants and their Performance. 2023 4th International Conference on Smart Electronics and Communication (ICOSEC). Trichy, India, 2023. P. 1018-1024. DOI: 10.1109/ICOSEC58147.2023.10276030.
20. Ultralytics. Comparing Ultralytics YOLO11 vs Previous YOLO Models. *Ultralytics Blog. 2025.* URL: https://www.ultralytics.com/blog/comparing-ultralytics-yolo11-vs-previous-yolo-models (date of access: 05.05.2025).
21. Baek Y., Lee B., Han D., Yun S., Lee H. Character Region Awareness for Text Detection. *arXiv preprint* arXiv:1904.01941. 2019. URL: https://arxiv.org/abs/1904.01941 (date of access: 05.05.2025).
22. Zhou X., Yao C., Wen H., Wang Y., Zhou S., He W., Liang J. EAST: An Efficient and Accurate Scene Text Detector. *arXiv preprint* arXiv:1704.03155. 2017. URL: https://arxiv.org/abs/1704.03155 (date of access: 05.05.2025).
23. Liao M., Shi B., Bai X. TextBoxes++: A Single-Shot Oriented Scene Text Detector. IEEE Transactions on Image Processing. 2018. Vol. 27, no. 8. P. 3676-3690. DOI: 10.1109/TIP.2018.2825107.
24. Li M., Lv T., Chen J., Cui L., Lu Y., Florencio D., Zhang C., Li Z., Wei F. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *arXiv preprint* arXiv:2109.10282. 2021. URL: https://arxiv.org/abs/2109.10282 (date of access: 05.05.2025).
25. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *arXiv preprint* arXiv:1512.03385. 2015. URL: https://arxiv.org/abs/1512.03385 (date of access: 05.05.2025).
26. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., Houlsby N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

arXiv preprint arXiv:2010.11929. 2020. URL: https://arxiv.org/abs/2010.11929 (date of access: 05.05.2025).

27. Microsoft. TrOCR-base-handwritten: Transformer-based OCR for handwritten text recognition. *Hugging Face. 2021*. URL: https://huggingface.co/microsoft/trocr-base-handwritten (date of access: 05.05.2025).
28. Team LUCID. TrOCR-small-korean: Transformer-based OCR for Korean text recognition. *Hugging Face. 2022.* URL: https://huggingface.co/team-lucid/trocr-small-korean (date of access: 05.05.2025).
29. Li J., Zhou H., Huang S., Cheng S., Chen J. Eliciting the Translation Ability of Large Language Models via Multilingual Finetuning with Translation Instructions. *arXiv preprint* arXiv:2305.15083. 2023. URL: https://arxiv.org/abs/2305.15083 (date of access: 05.05.2025).
30. Tiedemann J., Scherrer Y. Neural Machine Translation with Extended Context. *arXiv preprint* arXiv:1708.05943. 2017. URL: https://arxiv.org/abs/1708.05943 (date of access: 05.05.2025).
31. Yang X., Mu Y., Bontcheva K., Song X. Optimising LLM-Driven Machine Translation with Context-Aware Sliding Windows. Proceedings of the Ninth Conference on Machine Translation. Miami, Florida, USA: Association for Computational Linguistics, 2024. P. 1004-1010. DOI: 10.18653/v1/2024.wmt-1.101. URL: https://aclanthology.org/2024.wmt-1.101/ (date of access: 05.05.2025).
32. Ueno L. Best OCR Models for Text Recognition in Images. *Roboflow Blog*. 2023. URL: https://blog.roboflow.com/best-ocr-models-text-recognition/ (date of access: 05.05.2025).
33. JaidedAI. EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts. *GitHub repository.* 2023. URL: https://github.com/JaidedAI/EasyOCR (date of access: 05.05.2025).
34. Conneau A., Khandelwal K., Goyal N., Chaudhary V., Wenzek G., Guzmán F., Grave E., Ott M., Zettlemoyer L., Stoyanov V. Unsupervised Cross-lingual

Representation Learning at Scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). July 2020. P. 8440-8451. URL: https://aclanthology.org/2020.acl-main.747.pdf (date of access: 05.05.2025).

35. Yim M., Kim Y., Cho H.-C., Park S. SynthTIGER: Synthetic Text Image GEneratoR Towards Better Text Recognition Models. *arXiv preprint* arXiv:2107.09313. 2021. URL: https://arxiv.org/abs/2107.09313 (date of access: 05.05.2025).