

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ПОГОДЖЕНО

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Декан факультету (Директор ННІ)

Завідувач кафедри

інформаційних технологій

комп'ютерних наук

(назва факультету (ННІ))

(назва кафедри)

_____ Ігор БОЛБОТ

_____ Белла ГОЛУБ

(підпис)

(ім'я ПРИЗВИЩЕ)

(підпис)

(ім'я ПРИЗВИЩЕ)

“ ___ ” _____ 2025 р.

“ ___ ” _____ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Інтелектуальна система з обробки та перекладу рукописного тексту»

Спеціальність _____ 122 «Комп'ютерні науки»

(код і найменування)

Освітня програма _____ Інформаційні управляючі системи

(код і назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

кандидат технічних наук, доцент

(науковий ступінь та вчене звання)

(підпис)

Белла ГОЛУБ

(ім'я ПРИЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

кандидат технічних наук, доцент

(науковий ступінь та вчене звання)

(підпис)

Віктор КИРИЧЕНКО

(ім'я ПРИЗВИЩЕ)

Виконала

(підпис)

Анастасія ТРЕТЯК

(ім'я ПРИЗВИЩЕ здобувача)

КИЇВ-2025

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет (ННІ) Інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

доцент, к.т.н. Голуб Б. Л.
(науковий ступінь, вчене звання) (підпис) (ПІБ)
“ ” . 2025 року

З А В Д А Н Н Я

**ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Третяк Анастасії Романівни

(прізвище, ім'я, по батькові)

Спеціальність 122 “Комп'ютерні науки”

(код і назва)

Освітня програма “Інформаційні управляючі системи та технології”

(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи «Інтелектуальна система з обробки та перекладу рукописного тексту»

затверджена наказом ректора НУБіП України від “1” листопада 2024р. №1964 «С»

Термін подання завершеної роботи на кафедру 2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: публічний набір даних, сучасний англomовний датасет для розпізнавання слів, символів: IAM Handwriting Database”.

Перелік питань, що підлягають дослідженню:

1. Які методи машинного навчання так комп'ютерного зору забезпечують найвищу точність розпізнавання рукописного тексту.

2. Яким чином можна забезпечити автоматизований переклад розпізнаного рукописного тексту з використанням сучасних моделей машинного перекладу.

3. Які методи оптимізації дозволяють підвищити швидкодію та адаптивність інтелектуальної системи при роботі з рукописними текстами.

Перелік графічного матеріалу (за потреби) постер; презентація.

Дата видачі завдання “ ” 2025 р.

Керівник магістерської кваліфікаційної роботи Віктор КИРИЧЕНКО

(підпис)

(ім'я ПРІЗВИЩЕ)

Завдання прийняла до виконання Анастасія ТРЕТЯК

(підпис)

(ім'я ПРІЗВИЩЕ)

ЗМІСТ

<i>1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</i>	<i>13</i>
1.1 Предметна область	13
1.2 Постановка завдання	16
1.3 Проектування системи	19
<i>2 МЕТОДОЛОГІЯ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ</i>	<i>22</i>
2.1 Архітектура системи	22
2.3 Моделювання даних	30
2.4 Машинне навчання в системі обробки рукописного тексту	33
2.5 Основи машинного перекладу	36
<i>3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА МОДУЛІВ</i>	<i>39</i>
3.1 Механізми вилучення, обробки і передачі даних	39
3.2 Побудова матриці ймовірностей	44
3.3 Алгоритм One-Rule	47
3.4 Використання методу Наївного Байєса	49
3.5 Оцінка ефективності та результати навчання нейронної мережі	54
<i>4 ПРОГРАМНА РЕАЛІЗАЦІЯ</i>	<i>60</i>
4.1 Розробка системи	60
<i>ВИСНОВКИ</i>	<i>68</i>
<i>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</i>	<i>70</i>
<i>ДОДАТКИ</i>	<i>72</i>

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання магістерської кваліфікаційної роботи	Строк виконання етапу магістерської кваліфікаційної роботи	Примітка
1	Видача завдання	01.11.2024	Виконано
2	Аналіз предметної області	02.11.2024- 24.11.2024	Виконано
3	Проектування системи	26.11.-31.12.2024	Виконано
4	Розробка системи	01.02.-30.04.2025	Виконано
5	Аналіз результатів	01.05-16.07.2025	Виконано
6	Оформлення записки	16.07-12.11.2025	Виконано
7	Оформлення постеру	05.10-18.10.2025	Виконано
8	Написання тез до постеру	18.10-27.10.2025	Виконано
9	Постерна сесія	28.10-29.10.2025	Виконано
10	Перевірка на плагіат	14.10.2025	Виконано
11	Попередній захист	24.11-29.11.2025	Виконано
12	Захист роботи	05.12-13.12.2025	Виконано

Здобувач _____

Керівник роботи _____

РЕФЕРАТ

Основна частина пояснювальної записки займає сторінок 76 із них 66 сторінок основного тексту. Додатки мають обсяг в 4 сторінки. Робота містить 26 рисунків. Для виконання дослідження та підготовки матеріалів використано 25 джерел інформації.

Об'єкт дослідження - Процес автоматизованої обробки, розпізнавання та перекладу рукописного тексту засобами штучного інтелекту.

Предмет дослідження - Моделі, методи та алгоритми побудови інтелектуальної системи, яка здійснює розпізнавання рукописного тексту з використанням нейронних мереж і забезпечує його переклад засобами машинного перекладу.

Методи дослідження. У ході дослідження було застосовано низку сучасних методів, що дозволили досягти поставленої мети та отримати достовірні результати. Використовувалися різні підходи машинного навчання, серед яких ключову роль відіграли згорткові нейронні мережі, рекурентні нейронні мережі та моделі типу Transformer. Ці методи надали можливість ефективно розпізнавати структуру рукописного тексту, враховуючи контекст та стилістичні особливості письма. Також застосовувалися методи комп'ютерного зору для попередньої обробки зображень такі як- видалення шумів, сегментація, нормалізація яскравості та контрасту, що покращило якість вхідних даних і підвищило точність розпізнавання. Для оцінки результатів використовувалися математичне моделювання та статистичний аналіз, які допомогли проаналізувати рівень точності, частоту помилок і стабільність системи на різних наборах даних.

Мета і завдання дослідження. Полягає у створенні ефективної нейронної мережі для розпізнавання рукописного тексту, здатної точно і швидко перетворювати зображення письмових символів у цифровий

формат. Це включає розробку моделі, яка враховує контекстні та стилістичні особливості рукопису.

Завдання дослідження включають:

- Аналіз сучасних методів машинного навчання, зокрема згорткових та рекурентних нейронних мереж, а також моделей типу Transformer, для застосування у розпізнаванні тексту.
- Розробку методів попередньої обробки зображень (шумозаглушення, сегментація, нормалізація яскравості і контрасту) для підвищення якості вхідних даних.
- Створення і тренування архітектури нейронної мережі з використанням сучасних алгоритмів навчання.
- Проведення експериментального тестування прототипу системи на відкритих базах рукописних текстів (IAM Handwriting Database, MNIST) для оцінки її точності, стабільності та адаптивності до різних стилів почерку.
- Виконання кількісного аналізу результатів з допомогою методів математичного моделювання та статистичного аналізу для визначення рівня помилок і ефективності системи.

Наукова новизна полягає у розробці та впровадженні сучасного комплексного підходу до розпізнавання рукописного тексту на основі сучасних архітектур нейронних мереж, зокрема згорткових, рекурентних мереж і трансформерів. Запропоновано нові методи попередньої обробки зображень, що підвищують якість вхідних даних і відповідно точність розпізнавання. Вперше адаптовано ці технології для роботи з різноманітними стилями почерку, враховуючи контекст і стилістичні особливості письма.

Практичне значення розроблена система може бути інтегрована у різноманітні програмні комплекси, що забезпечують цифрову обробку

документів, автоматичне введення даних, навчальні платформи чи сервіси машинного перекладу. Її впровадження сприяє зменшенню часу обробки рукописних матеріалів, підвищенню точності передачі текстової інформації та зручності взаємодії користувача з інформаційними системами.

Створений програмний прототип може бути використаний у сфері освіти для оцифрування конспектів і нотаток, у науково-дослідній діяльності — для перекладу історичних рукописів, у бізнесі — для обробки заповнених вручну форм, анкет чи документів.

Апробація. Матеріали роботи були опубліковані в тези під назвою «Інтелектуальна система з обробки та перекладу рукописного тексту» в збірнику наукових праць «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ '2025'», 19 квітня 2025 року, НУБіП України, Київ. Режим доступу: <http://econference.nubip.edu.ua/index.php/taacsd/2025/paper/view/3627>

ABSTRACT

The main part of the explanatory note occupies 76 pages, 66 of which are the main text. The appendices are 4 pages long. The work contains 26 figures. Twenty-five sources of information were used to conduct the research and prepare the materials.

The object of the study is the process of automated processing, recognition and translation of handwritten text using artificial intelligence.

The subject of the research is models, methods and algorithms for building an intelligent system that recognises handwritten text using neural networks and translates it using machine translation.

Research methods. During the study, a number of modern methods were used to achieve the set goal and obtain reliable results. Various machine learning approaches were used, among which convolutional neural networks, recurrent neural networks, and Transformer-type models played a key role. These methods made it possible to effectively recognise the structure of handwritten text, taking into account the context and stylistic features of the writing. Computer vision methods were also used for pre-processing images, such as noise removal, segmentation, brightness and contrast normalisation, which improved the quality of the input data and increased recognition accuracy. Mathematical modelling and statistical analysis were used to evaluate the results, which helped to analyse the accuracy, error rate and stability of the system on different data sets.

Research objectives and tasks. The aim is to create an effective neural network for handwritten text recognition, capable of accurately and quickly converting images of written characters into digital format. This includes developing a model that takes into account the contextual and stylistic features of handwriting.

The objectives of the research include:

- Analysis of modern machine learning methods, in particular convolutional and recurrent neural networks, as well as Transformer-type models, for use in text recognition.
- Developing image pre-processing methods (noise suppression, segmentation, brightness and contrast normalisation) to improve the quality of input data.
- Creating and training a neural network architecture using modern learning algorithms.
- Conducting experimental testing of the prototype system on open handwritten text databases (IAM Handwriting Database, MNIST) to evaluate its accuracy, stability, and adaptability to different handwriting styles.
- Performing quantitative analysis of the results using mathematical modelling and statistical analysis methods to determine the error rate and effectiveness of the system.

The scientific novelty lies in the development and implementation of a modern comprehensive approach to handwritten text recognition based on modern neural network architectures, in particular convolutional, recurrent networks and transformers. New methods of image pre-processing are proposed, which improve the quality of input data and, accordingly, the accuracy of recognition. For the first time, these technologies have been adapted to work with a variety of handwriting styles, taking into account the context and stylistic features of writing.

The developed system can be integrated into various software complexes that provide digital document processing, automatic data entry, educational platforms, or machine translation services. Its implementation helps reduce the time required to process handwritten materials, increase the accuracy of text information transfer, and improve the user experience with information systems.

The software prototype can be used in education to digitise lecture notes and notes, in research to translate historical manuscripts, and in business to process handwritten forms, questionnaires, or documents.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

БД – База даних

ПЗ – Програмне забезпечення

СУБД – програмне забезпечення, що допомагає створювати керувати та організовувати дані в базах даних.

CNN – (Convolutional Neural Network) Згорткова нейронна мережа

API – Application Programming Interface

JSON – JavaScript Object Notation

API – Application Programming Interface

CER – Character Error Rate (Частота помилок символів)

WER – Word Error Rate (Частота помилок слів)

OpenCV – бібліотека функцій та алгоритмів з комп'ютерним зором.

HTR – розвиток оптичного розпізнавання тексту

ВСТУП

Сучасним програмним рішенням, що поєднує в собі технології розпізнавання символів(OCR), машинного навчання та нейронних мереж є інтелектуальна система перекладу та обробки рукописного тексту. Адже такі системи дозволяють комп'ютерам не лише розпізнавати графічне зображення рукописного тексту, а також здійснювати глибоку мовну обробку, аналіз контексту та смислової інтерпретацію текстів.

Завдяки системам розпізнавання тексту, обробляється велика кількість документів без ручного введення інформації, що значно пришвидшує погодження контрактів, вхідну реєстрацію та аналітику даних. Завдяки поєднанню алгоритмів початкової обробки зображень, визначення контурів, сегментів та використанню convolutional neural networks для класифікації окремих символів та слів, інтелектуальні платформи також забезпечують точність складних і неструктурованих рукописів. Актуальність таких систем зростає у зв'язку з автоматизацією великої обробки документів. Розробка і впровадження інтелектуальних систем перекладу та обробки рукописного тексту, відкривають нові можливості для цифрової трансформації документообігу, покращення доступності інформації, а також інтеграції рукописних текстів у мови глобальних цифрових сервісів.

Історично розвиток інтелектуальних систем розпізнавання рукописного тексту почався задовго до появи сучасних методів штучного інтелекту. Важливий прорив стався у 1989 році завдяки згортковим нейронним мережам, які успішно були застосовані для розпізнавання цифр на поштових індексах. З тих пір технології вдосконалювались і з'являлись нові алгоритми обробки зображень, сегментації тексту та ієрархічного аналізу, що дозволяє працювати з складними рукописними формами.

Сучасні інтелектуальні системи перекладу інтегрують обробку природної мови для розуміння і генерації змісту, це дозволяє підвищити точність перекладу, і адаптувати його до специфіки області застосування.

Окрім точності, такі системи значно скорочують час і витрати на ручний переклад з транскрипцією.

Загалом інтелектуальна система перекладу це важливий інструмент між сучасним і минулим мовними бар'єрами.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Предметна область

Область інтелектуальних систем для обробки та перекладу рукописного тексту охоплює методи і технології автоматичного розпізнавання тексту, який написаний від руки, та його перетворення у цифровий формат, з подальшою багатомовною інтерпретацією. В центрі цієї сфери лежить використання алгоритмів комп'ютерного зору, обробки природної мови та штучного інтелекту для вирішення завдань оцифрування рукописних документів, автоматичної корекції помилок, відновлення структури тексту, визначення семантики та машинного перекладу отриманих текстів.

Також сюди входить сканування рукописних матеріалів, попередня обробка зображення (сегментація, виділення символів), побудова та навчання моделі розпізнавання на основі різних нейронних мереж, аналізу контексту, нормалізації тексту, а також інтеграції з різними мовними технологіями перекладу. Особлива увага приділяється адаптації систем до варіацій почерку, багатомовності, специфіки історичних чи спеціалізованих матеріалів. Практична значущість предметної області полягає у прискоренні доступу до даних, автоматизації архівних, освітніх чи ділових процесів.

Основна мета інтелектуальної системи обробки та перекладу рукописного тексту полягає в створенні автоматизованого інструменту, здатного розпізнавати рукописний текст різних стилів та форматів, а також перетворювати його у структурований цифровий формат, ну і головне здійснювати якісний і контекстно коректний переклад між мовами. Це дозволяє ефективно знизити потребу у ручній праці, також підвищити точність і швидкість обробки інформації, також важливо забезпечити зручний доступ до даних у цифровому вигляді для подальшого аналізу.

Досягання цієї мети вимагає застосування методів обробки зображень машинного навчання та нейронних мереж. Система має розуміти контекст і граматику, забезпечуючи високу якість і точність кінцевого перекладу.

Основні завдання інтелектуальної системи обробки та перекладу рукописного тексту:

Автоматичне розпізнавання рукописних символів і слів із зображень, або сканованих документів.

Попередня обробка рукописного тексту, включно з покращенням якості зображення, сегментацією рядків і слів.

Аналіз та інтерпретація розпізнаного тексту з урахуванням мовного контексту, граматики і стилістики, для підвищення точності розпізнавання.

Автоматичний переклад рукописного тексту між різними мовами з використанням машинного навчання і моделей обробки природної мови.

Надання інтерфейсу для взаємодії користувача із системою, з уточненням результатів перекладу.

Головну роль у розпізнаванні рукописного тексту відіграють згорткові нейронні мережі(CNN), які ефективно виділяють ключові ознаки символів і слів із оцифрованих зображень. Часто CNN комбінують з нейронними мережами типу RNN(рекурентні нейронні мережі), що дозволяє системі враховувати послідовність символів і контекст усередині слів та речень. Для багатомовної обробки або роботизованого перекладу застосовують трансформер-архітектури і великі мовні моделі.

Попередня обробка тексту включає підвищення якості зображення, бінаризацію, сегментацію на рядки, слова та символи, що важливо для підготовки даних до розпізнавання. Обробка природної мови допомагає аналізувати структуру, виявляти граматичні і семантичні зв'язки, коригувати помилки розпізнавання та вдосконалювати переклад.

У сфері перекладу отриманих текстів використовуються сучасні неймережеві машинні перекладачі, що працюють на основі великих паралельних корпусів даних і здатні адаптувати критичні культурні та мовні особливості для контекстуального, і не лише прямого перекладу рукописної інформації. Додатково впроваджуються системи постредагування, автоматичного оцінювання якості та глибокого інтелектуального аналізу.

Проблема розпізнавання тексту бере свій початок ще у середині ХХ століття, коли з розвитком обчислювальної техніки почали з'являтися перші системи оптичного розпізнавання символів. Ці технології спочатку були спрямовані на зчитування друкованого тексту для автоматизації введення даних у банки, поштові служби та бібліотеки. Однак розпізнавання рукописного тексту залишалось складним завданням через варіативність почерків, нахилу, товщини ліній та стилістичних особливостей письма.

А з появою методів машинного навчання у 1980-1990-х роках, і згодом з нейронними мережами точність систем OCR значно зросла. У 2000-х роках активно почали застосовуватись згорткові нейронні мережі (CNN) для виділення ознак символів, а пізніше рекурентні мережі RNN, що дозволили враховувати послідовність символів у словах.

Останнє десятиліття характеризується стрімким розвитком трансформер-архітектур (GPT, T5, BERT), що поєднують зорове сприйняття тексту з контекстним розумінням змісту. Це дозволило створювати інтелектуальні системи, здатні не лише розпізнавати, а й перекладати рукописний текст у реальному часі, зберігаючи контекст, граматику і стиль.

1.2 Постановка завдання

У сучасних умовах цифровізації дуже актуальним є завдання автоматизованого розпізнавання та перекладу рукописної інформації. Основною метою є розробка системи, яка дозволяє перетворювати рукописний текст з різних джерел у структурований електронний вигляд і здійснювати його автоматичний переклад на іншу мову, із збереженням змісту та точністю.

Для цього необхідно:

Провести вхідну обробку зображень з рукописним текстом, що включає корекцію шумів, підвищення чіткості, бінаризацію та сегментацію для виділення текстових областей;

Розробити та застосувати алгоритми машинного навчання та нейронних мереж для розпізнавання символів, слів, чи речень при значній варіативності почерку, або наявності дефектів;

Впровадити процеси корекції та постобробки розпізнаного тексту з використанням інструментів обробки природньої мови, забезпечивши нормалізацію, перевірку граматики, лексичну точність і підготовку до подальшого перекладу;

Здійснити автоматичний переклад розпізнаного тексту із застосуванням сучасних нейромережевих або статистичних моделей, орієнтованих на збереження сенсу та стилістики інформації;

Забезпечити можливість збереження отриманих результатів у структурованому вигляді для подальшого проведення пошуку, аналізу.

Об'єктом дослідження є процес автоматизованої обробки та перекладу рукописного тексту. Предметом дослідження є методи машинного навчання, нейронні мережі та алгоритми обробки природньої мови, що забезпечують розпізнавання та переклад рукописної інформації.

Практична значущість роботи обробки рукописного тексту – від введення до отримання перекладу. Такий підхід дозволить автоматизувати процеси оцифрування архівних матеріалів, спростити роботу з рукописними документами у різних галузях, а також підвищити ефективність комунікації.

Також важливим етапом у процесі розробки інтелектуальної системи розпізнавання та перекладу рукописного тексту є первинна обробка вхідних даних. Адже саме від цього етапу залежить точність подальшого розпізнавання та перекладу, система повинна працювати навіть із зображеннями низької якості, на яких можуть бути шуми, тіні або спотворення.

Первинна обробка зображення зазвичай включає кілька основних процедур. Насамперед здійснює фільтрацію шумів, щоб прибрати випадкові спотворення, що можуть заважати правильному виділенню символів. Для цього часто використовують гаусів фільтр, який згладжує зображення за допомогою ядра певного розміру. Його функцію можна описати рівнянням:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Де дельта означає ступінь розмиття, а (x, y) – координати пікселя.

Після цього виконується бінаризація зображення, тобто переведення його у двокольоровий формат – чорний і білий, це дозволяє чіткіше відокремити символи від фону.

Найпростіше це робиться за допомогою порогового перетворення:

$$B(x, y) = \begin{cases} 1, & \text{якщо } I(x, y) > T, \\ 0, & \text{інакше} \end{cases}$$

Де $I(x, y)$ – це інтенсивність пікселя, а T – це порогове значення, яке можна визначитися автоматично.

Далі проводиться сегментація, тобто розділення тексту, на окремі рядки, слова, символи. Це потрібно для того щоб нейронна мережа могла навчатися на менших, чітко виділених елементах. Для цього можуть застосовуватись методи пошуку зв'язних компонентів або морфологічні операції – ерозія і дилатація, які допомагають покращити видимість контурів.

Останнім етапом підготовки є нормалізація зображень – вирівнювання розміру, масштабу та орієнтації символів. Наприклад, координати пікселів можна перетворити до стандартного діапазону значень за формулою :

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad y' = \frac{y - y_{min}}{y_{max} - y_{min}}$$

Це робиться для того, щоб усі символи, незалежно від почерку чи розміру, подавались до моделі у єдиному форматі.

Для реалізації всіх цих етапів доцільно використовувати сучасні засоби програмування і машинного навчання. Основною мовою розробки є Python, оскільки вона має велику кількість бібліотек для обробки зображень і побудови нейронних мереж. Зокрема, для аналізу зображень застосовується OpenCV, для побудови моделей – TensorFlow, Keras або PyTorch, а для обробки тексту та перекладу – NLTK, spaCY або інші платформи які містять навчені багатомовні моделі.

Використання цих технологій дозволяє створити єдину систему, що виконує повний цикл обробки даних – від зображення рукописного тексту до готового перекладу. Таким чином, що первинна обробка зображень у поєднанні з алгоритмами машинного навчання створює основу для подальшого навчання моделі, підвищуючи точність розпізнавання, стійкість до варіацій почерку та якість перекладу.

1.3 Проектування системи

Проектування починається з визначення даних, на яких вона буде працювати. У випадку рукописного тексту це відкриті бази даних з рукописними символами і словами, наприклад Handwriting Database, де представлені різноманітні почерки; скановані документи чи нотатки, що оцифровані шляхом сканування або фотографування, а також спеціально створені штучні зображення, які допомагають збільшити різноманітність даних і зробити систему більш стійкою до різних стилів почерку.

Після збору даних важливо розуміти, що саме система повинна робити. Вона не тільки повинна розпізнавати символи і слова, а й коригувати помилки, перекладати текст на різні мови і зберігати у зручному форматі. Крім того потрібно враховувати швидкість обробки, точність.

Архітектура системи зазвичай складається з кількох взаємопов'язаних модулів. Перший – це модуль попередньої обробки зображень, що усуває шуми, підвищує чіткість, нормалізує розмір зображень.

Потім саме розпізнавання тексту, із застосуванням нейронних мереж (CNN) для виділення ознак і символів і RNN для послідовності і розуміння контексту слів у реченні. Далі модуль обробки природньої мови, який перевіряє граматику і готує до перекладу. Сам переклад здійснюється за допомогою спеціалізованих багатомовних моделей, що зберігають контекст і стиль тексту. І нарешті інтерфейс користувача, що дозволяє завантажувати зображення і отримувати переклад.

Щоб краще розуміти як взаємодіє користувач із системою використовують Use Case діаграму. На якій показано, що користувач може завантажувати рукописні зображення та отримувати оброблений текст, перевіряти його, перекладати і зберігати результати. Діаграма також відображає взаємодію різних модулів системи, що автоматично виконують свої завдання за лаштунками – наприклад, попередню обробку, розпізнавання або переклад. Use Case допомагає не загубитись і визначити що має робити кожен компонент.

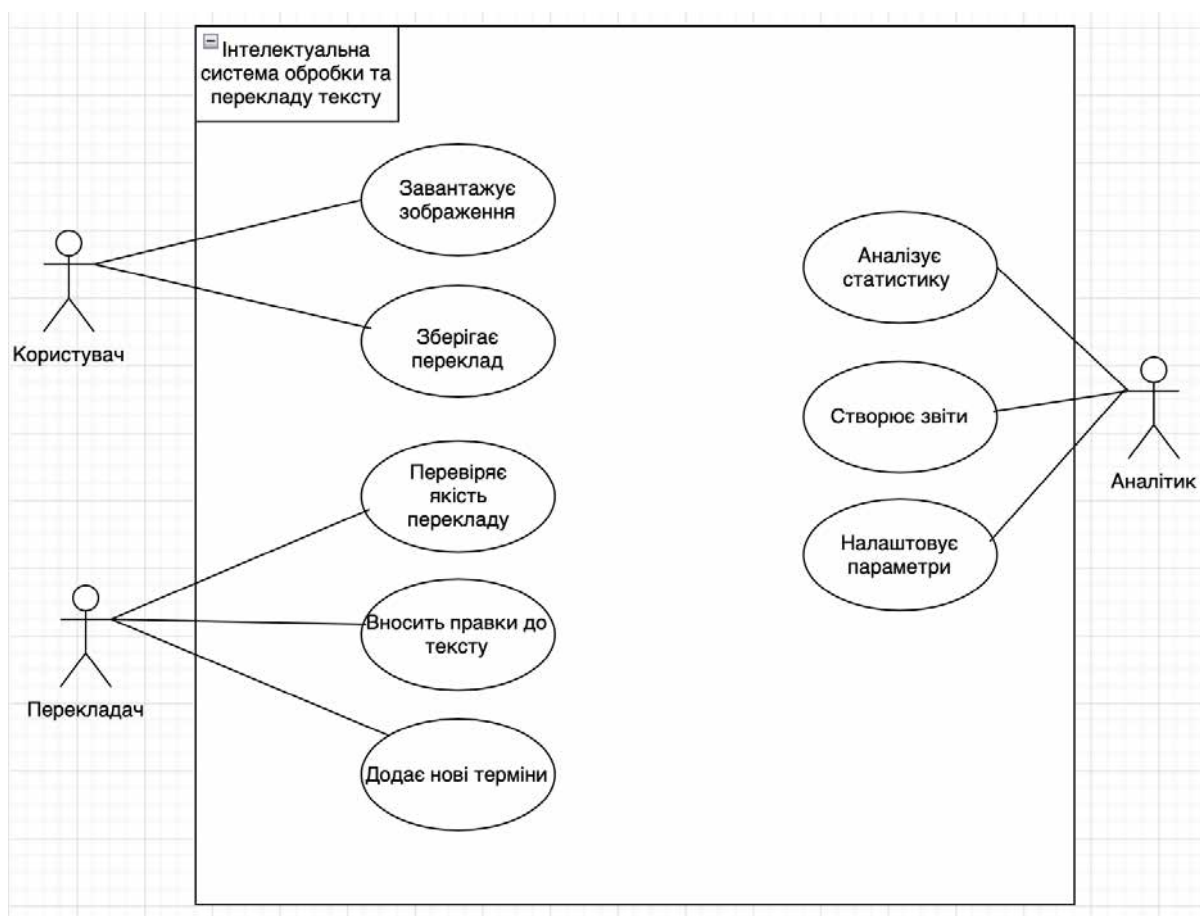


Рис. 1 Діаграма прецедентів (Use Case).

На представлений діаграмі прецедентів показано взаємодію основних користувачів із системою інтелектуальної обробки та перекладу рукописного тексту.

Діаграма допомагає уявити, які дії можуть виконувати різні категорії користувачів і як система підтримує ці процеси. У системі виділено три

основні ролі: Користувач, Перекладач та Аналітик. Кожен з них взаємодіє з певними функціями системи, які реалізовані у вигляді прецедентів.

Користувач - може завантажувати зображення рукописного тексту, зберігати отриманий переклад та перевіряти якість перекладу. Ці дії дозволяють забезпечити первинний контакт із системою та отримати результат обробки тексту.

Перекладач - виконує більш детальні операції: перевіряє якість перекладу, вносить необхідні правки до тексту і додає нові терміни, що дозволяє покращити точність системи та адаптувати її до специфічної термінології. Завдяки цьому модулю досягається висока якість обробки та перекладу рукописних текстів.

Аналітик - відповідає за статистичний і аналітичний супровід системи: він аналізує статистику обробки, створює звіти та налаштовує параметри системи. Це дозволяє відслідковувати ефективність роботи системи та приймати рішення щодо її оптимізації. Use case діаграма демонструє, що система інтегрує взаємодію різних користувачів із комплексними функціями обробки тексту, підтримує процес перевірки і корекції перекладу, а також забезпечує аналітичний контроль. Завдяки такій моделі можна чітко визначити функціональні обов'язки кожного користувача і спростити процес розробки та тестування системи.

2 МЕТОДОЛОГІЯ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

2.1 Архітектура системи

Архітектура системи побудована таким чином, щоб забезпечити повний цикл обробки від моменту завантаження зображення до отримання готового перекладеного тексту. Умовно її можна поділити на кілька взаємопов'язаних модулів, кожен з яких виконує свою частину роботи, але вони всі взаємодіють між собою через спільну логіку обробки даних.

Першим і дуже важливим етапом у роботі є первинна обробка даних, адже від її якості залежить подальша точність розпізнавання. Коли користувач завантажує зображення рукописного тексту, воно потрапляє до модуля попередньої обробки. Тут проводиться масштабування, вирівнювання, перетворення кольорового зображення в градації сірого, а також нормалізація контрасту. Такі дії дозволяють отримати максимально чистий вхідний сигнал, що є придатним для подальшого аналізу нейронною мережею. На цьому етапі можуть застосовуватись математичні операції типу згортки або фільтрації. Наприклад нормалізація інтенсивності пікселів часто використовуються за формулою:

$$I_{norm} = \frac{I - I_{min}}{I_{max} - I_{min}}$$

де I – інтенсивність пікселя, I_{min} і I_{max} – мінімальне та максимальне значення яскравості. Це дозволяє зменшити вплив різниці освітлення при скануванні або фотографуванні тексту.

Другим кроком є опрацювання шуму на зображенні, тобто усунення плям, подряпин або артефактів, що можуть з'явитись при зйомці або скануванні рукописних документів. Для цього використовуються як класичні фільтри так і сучасні методи, засновані на згорткових нейронних мережах, які вчать розрізняти текстові області від фонових шумів. Одним із поширених способів є застосування гаусівського згладжування, яке описується формулою :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Де дельта визначає ступінь розмиття. Така обробка допомагає зберегти контури символів, одночасно з прибиранням дефектів. У результаті вихідне зображення стає більш чітким, а алгоритм розпізнавання отримує якісний матеріал для роботи.

Важливою складовою проєктування є топологія системи, яка визначає, як саме взаємодіють між собою її компоненти.

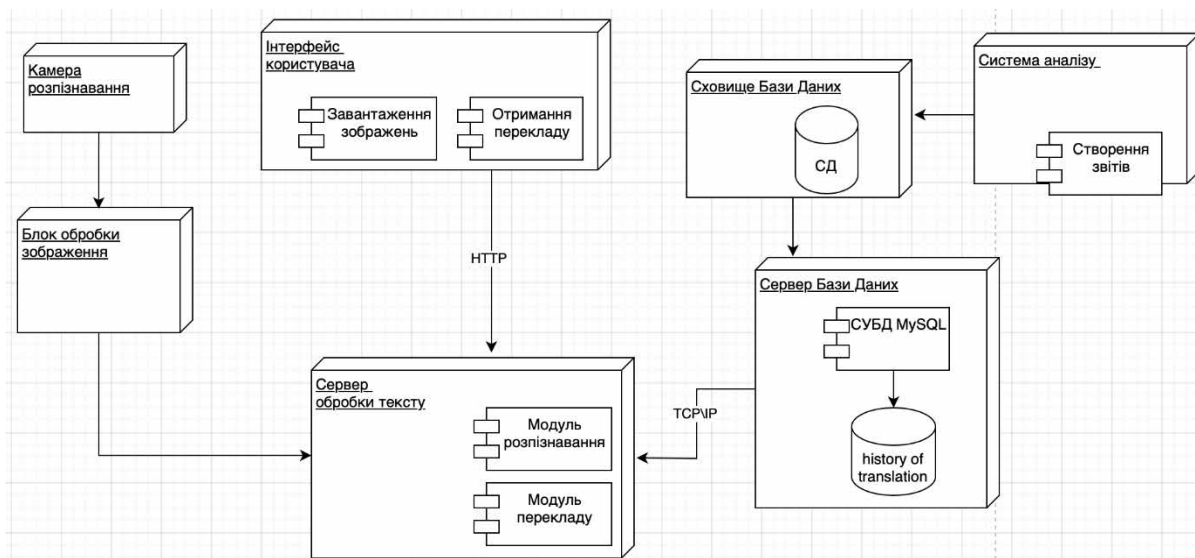


Рис. 2 Топологія системи.

На рисунку подано топологію інтелектуальної системи, яка реалізує повний цикл — від зчитування рукописного тексту до отримання його перекладу та аналітичної обробки результатів. Архітектура має багаторівневу структуру, що поєднує апаратні, програмні та серверні компоненти в єдину інтегровану систему.

Початковим елементом є камера розпізнавання, яка виконує захоплення зображення рукописного тексту. Отримані дані надходять до блоку обробки зображення, де проводиться попередня підготовка — фільтрація шумів, підвищення чіткості, вирівнювання контрасту та сегментація текстових областей. Цей етап забезпечує якісний вхідний матеріал для подальшого розпізнавання.

Після первинної обробки інформація передається на сервер обробки тексту, який є центральним елементом системи. На ньому функціонують два основні модулі — модуль розпізнавання та модуль перекладу. Перший відповідає за ідентифікацію символів, слів і речень за допомогою нейронних мереж та алгоритмів машинного навчання, а другий — за автоматичний переклад розпізнаного тексту на цільову мову із збереженням змісту та стилістики. Взаємодія між користувачем та сервером здійснюється через HTTP-з'єднання, що дозволяє швидко передавати результати перекладу на клієнтську частину.

З боку користувача працює інтерфейс користувача, який забезпечує простоту взаємодії із системою: тут можна завантажити зображення, переглянути результати розпізнавання, отримати переклад і за потреби внести правки. Усі виконані дії фіксуються в сервері бази даних, де за допомогою системи управління базами даних MySQL формується історія перекладів. Передача даних між сервером обробки тексту та сервером бази даних здійснюється через протокол TCP/IP, що гарантує стабільність та безпеку обміну інформацією.

Збережені результати потрапляють до сховища бази даних, з якого дані можуть бути використані системою аналізу. Цей компонент відповідає за створення статистичних звітів, моніторинг точності перекладу, аналіз ефективності роботи модулів та формування рекомендацій щодо покращення системи. Таким чином, реалізується не лише цикл «розпізнавання–переклад», а й етап зворотного зв'язку, який дає змогу удосконалювати модель на основі накопичених результатів.

У цілому, представлена топологія демонструє багаторівневий принцип побудови інтелектуальної системи, де кожен компонент виконує чітко визначену роль, а їхня взаємодія забезпечує цілісний, автоматизований процес обробки, перекладу та аналізу рукописного тексту.

Такий підхід робить систему гнучкою, масштабованою та придатною для інтеграції в більші інформаційні середовища.

2.2 Загальні поняття з напрямку OLAP - технології

При дослідженні сучасних інтелектуальних системах, зокрема у системах обробки та перекладу текстів, OLAP-технології (Online Analytical Processing) відіграють ключову роль у забезпеченні аналітичного доступу до великих обсягів даних, що накопичуються під час функціонування системи. Основне призначення OLAP полягає у багатовимірному аналізі даних, який дозволяє ефективно виявляти закономірності, тенденції та залежності між різними аспектами роботи системи. Інтелектуальна система перекладу текстів постійно генерує значні обсяги інформації — це можуть бути статистичні дані про частоту використання слів, оцінки якості перекладу, час обробки запитів, активність користувачів, рівень збігів із базою перекладів чи результатами машинного навчання. Усі ці дані потребують структурованого зберігання та гнучкого інструменту для аналізу — саме це забезпечує OLAP.

OLAP-сховище дозволяє представити інформацію у вигляді багатовимірного куба даних, де кожен вимір відповідає певній характеристиці процесу перекладу. Наприклад:

Вимір тип тексту — технічний, юридичний, художній, науковий;

Вимір час — дата або період обробки;

Вимір алгоритм перекладу — статистичний, нейронний, гібридний;

Вимір якість перекладу — відсоток точності, кількість редагувань або оцінка користувача.

Завдяки такій структурі система може миттєво відповідати на складні аналітичні запити: наприклад, визначати, у яких мовних парах нейронна модель демонструє найвищу точність; як змінюється продуктивність

системи при перекладі технічних документів; або які помилки найчастіше трапляються при перекладі певних типів текстів. Використання OLAP у поєднанні з інтелектуальними алгоритмами дозволяє не лише зберігати дані, а й перетворювати їх на знання, що є основою для подальшого навчання моделей перекладу, удосконалення лінгвістичних правил та покращення користувацького досвіду.

Історично склалося так, що сьогодні термін "OLAP" має на увазі не тільки багатовимірний погляд на дані з боку кінцевого користувача, але і багатовимірне представлення даних в цільовій БД. Саме з цим пов'язано появу як самостійні терміни "Реляційний OLAP" (ROLAP) і "Багатовимірний OLAP" (MOLAP).

OLAP-сервіс є інструментом для аналізу великих об'ємів даних в режимі реального часу. Взаємодіючи з OLAP- системою, користувач зможе здійснювати гнучкий перегляд інформації, одержувати довільні зрізи даних і виконувати аналітичні операції деталізації, згортки, крізного розподілу, порівняння в часі одночасно по багатьох параметрах. Вся робота з OLAP- системою відбувається в термінах наочної області і дозволяє будувати статистично обґрунтовані моделі ділової ситуації.

Програмні засоби OLAP - це інструмент оперативного аналізу даних, що містяться в сховищі. Головною особливістю є те, що ці засоби орієнтовані на використання не фахівцем у області інформаційних технологій, не експертом-статистиком, а професіоналом в прикладній області управління - менеджером відділу, департаменту, управління, і, нарешті, директором. Засоби призначені для спілкування аналітика з проблемою, а не з комп'ютером.

На рис. 3 показаний елементарний OLAP-куб, що дозволяє проводити оцінки даних по трьох вимірюваннях.

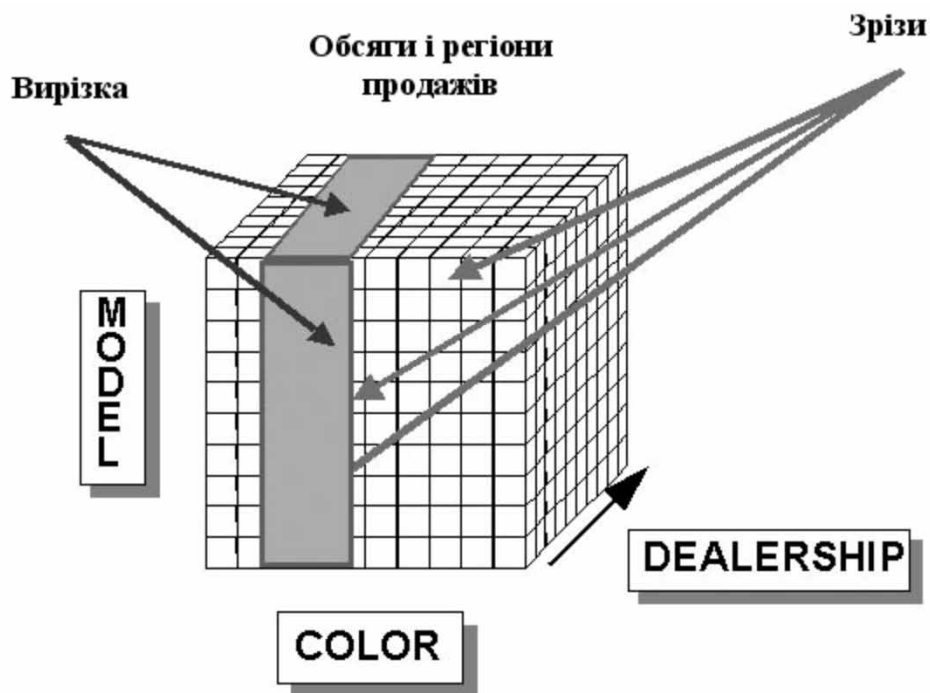


Рис. 3 OLAP-куб.

Багатовимірний OLAP-куб і система відповідних математичних алгоритмів статистичної обробки дозволяє аналізувати дані будь-якої складності на будь-яких тимчасових інтервалах.

Для ефективної реалізації OLAP-технологій та забезпечення швидкого доступу до великих обсягів інформації в інтелектуальній системі обробки та перекладу тексту необхідною складовою є сховище даних. Сховище даних — це централізована база, у якій зберігаються структуровані, узгоджені й очищені дані, отримані з різних джерел: текстових корпусів, баз перекладів, результатів машинного навчання, користувацьких запитів і журналів роботи системи. Основна мета сховища — забезпечити єдине джерело достовірних даних для аналітичної обробки, навчання моделей та прийняття управлінських рішень щодо вдосконалення системи.

Однорівнева архітектура

Однорівнева архітектура передбачає зберігання всіх даних в одному централізованому сховищі. Вона використовується у невеликих системах, де обсяг інформації незначний і немає складних аналітичних запитів. У

контексті перекладацької системи такий підхід може застосовуватися на початкових етапах — для зберігання невеликих обсягів текстів та статистики перекладу. Перевага — простота реалізації; недолік — обмежена масштабованість та швидкодія при зростанні кількості даних.

Дворівнева архітектура

Дворівнева архітектура передбачає наявність операційних баз даних (OLTP) і аналітичного сховища (OLAP). Дані з робочих систем обробки текстів спочатку накопичуються в операційних базах, а потім за допомогою процесів ETL (Extract, Transform, Load) витягуються, очищуються, перетворюються у потрібний формат і завантажуються до аналітичного сховища.

У системі перекладу це означає, що результати перекладу, оцінки користувачів і лінгвістичні статистики спочатку формуються в оперативному середовищі, а потім переносяться у сховище для подальшого аналітичного аналізу — наприклад, визначення ефективності моделей перекладу або якості перекладених корпусів.

3 Трирівнева архітектура

Трирівнева архітектура є найбільш розповсюдженою та оптимальною для складних інтелектуальних систем. Вона складається з трьох рівнів:

Рівень джерел даних — містить текстові бази, машинно-перекладні корпуси, користувацькі журнали, результати аналізу нейронних мереж.

Рівень інтеграції (сховище даних) — забезпечує об'єднання, очищення, стандартизацію та структурування даних у єдиному форматі.

Рівень презентації (OLAP та аналітика) — надає інструменти для аналітичного перегляду, візуалізації та прогнозування результатів роботи системи.

У такій архітектурі дані можуть бути розподілені за тематичними вітринами даних (Data Marts), наприклад:

вітрина для статистики машинного перекладу;

вітрина для оцінки якості перекладів;

вітрина для аналізу продуктивності користувачів чи моделей.

Трирівнева архітектура дозволяє розділити аналітичне навантаження від основних процесів перекладу, що підвищує продуктивність, забезпечує масштабованість і підтримує глибоку інтеграцію з модулями штучного інтелекту.

4 Хмарна архітектура сховищ

Сучасні інтелектуальні системи перекладу часто використовують хмарні сховища даних, які забезпечують гнучкість і можливість обробки великих текстових масивів у режимі реального часу. Хмарні рішення (наприклад, Google BigQuery, AWS Redshift, Microsoft Azure Synapse) дозволяють масштабувати ресурси під потреби системи, швидко оновлювати дані та підтримувати інтеграцію з сервісами машинного навчання. Для системи перекладу це означає, що результати аналізу, статистика помилок, навчальні корпуси та мовні моделі можуть зберігатися у хмарі й бути доступними для спільної аналітики або автоматичного вдосконалення моделей перекладу.

2.3 Моделювання даних

Для реалізації інтелектуальної системи обробки та перекладу рукописного тексту критично важливим є ефективне моделювання даних, яке дозволяє організувати збереження, обробку та аналіз інформації у структурованому вигляді.

Основою для будь-якої структури даних є відображення елементарної одиниці даних у вигляді такої трійки: об'єкт, властивість об'єкта, значення властивості. Сукупність взаємопов'язаних між собою елементарних одиниць даних може відображатися різноманітними способами, що

приводить до формування різних структур, а відтак – різних моделей даних. Моделі даних поділяються на два класи: сильно та слабо типізовані.

У сильно типізованих моделях усі дані мають належати до певної категорії, або типу. Якщо дані не підпадають під жодну з категорій, їх потрібно типізувати штучно. Деякі моделі будуються у такий спосіб, що категорії визначаються наперед і не можуть змінюватися динамічно. У цьому випадку модельований світ начебто вміщується в гальмівну сорочку. Наприклад, категорія «службовець» – строго фіксована, й усі її об'єкти повинні мати однакові властивості та структуру. Сильно типізовані моделі мають значні переваги, бо дають змогу побудувати абстракції властивостей даних і дослідити їх у термінах категорій. Більшість моделей, що використовуються в автоматизованих системах, зокрема й базах даних, належать до сильно типізованих.

Для слабо типізованих моделей належність даних до тієї чи іншої категорії не має жодного значення. Категорії використовуються настільки, наскільки це доцільно в кожному конкретному випадку. Окремі дані можуть існувати як незалежно, так і у зв'язку з іншими. Інформація про категорії (якщо вони використовуються) розглядається як додаткова.

Моделювання даних є одним із ключових етапів побудови сховища даних, оскільки саме на цьому етапі визначається логічна структура зберігання, взаємозв'язки між сутностями та принципи організації інформації для подальшого аналітичного опрацювання. Основна мета моделювання даних - забезпечити логічну, зрозумілу та масштабовану структуру бази даних, що підтримує всі функції системи: від завантаження та обробки зображень рукописного тексту до його розпізнавання, перевірки, перекладу та збереження результатів.

Таким чином, моделювання даних базується на логічному поділі процесів на сутності та визначенні зв'язків між ними. Ключові аспекти, які враховані:

Зв'язок користувача з його запитами (Users to TranslationRequests).

Прив'язка зображень до конкретних запитів (Images to TranslationRequests).

Відстеження алгоритмів розпізнавання для кожного запиту (RecognitionSystems to TranslationRequests).

Можливість аналітики та ведення історії помилок (Errors to TranslationRequests).

Взаємодія з перекладачами для підвищення якості результату (Translators to TranslationRequests).

На основі цього підходу побудована структура бази даних, що дозволяє ефективно зберігати, обробляти і аналізувати всі дані системи. Усі сутності та їх взаємозв'язки відображено на схемі сховища даних, по типу «Зірка» (рис. 4). Для створення та налаштування сховища даних був використаний Microsoft SQL Server та СУБД Microsoft SQL Server Manager.

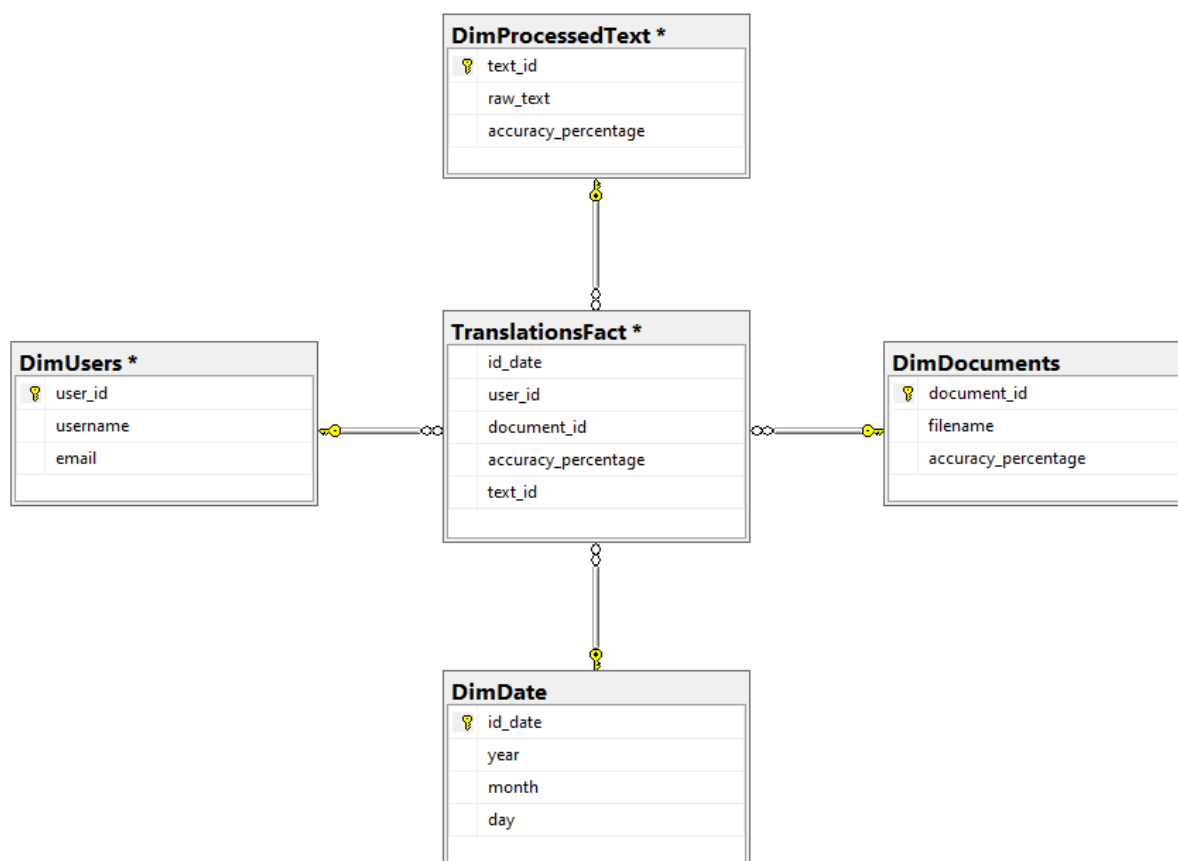


Рис.4 Структура сховища даних.

До складу сховища входять такі таблиці вимірів:

- TranslationsFact — фактова таблиця, що акумулює дані про виконанні переклади. Містить ідентифікатори користувача, документа, тексту, дату виконання перекладу та відсоток точності;
- DimUsers — таблиця виміру користувачів, що зберігає інформацію про учасників процесу перекладу (ідентифікатор користувача, ім'я користувача, електронну пошту);

- DimDocuments — таблиця виміру документів, яка містить метадані про оброблені документи (ідентифікатор, назву файлу, середній показник точності);
- DimProcessedText — таблиця виміру текстів, де зберігається інформація про оброблений текст, його ідентифікатор, початковий вміст (*raw_text*) та точність обробки;
- DimDate — таблиця виміру дати, що забезпечує можливість аналізу даних у часовому розрізі (рік, місяць, день).

Завдяки такій структурі, дані зручно агрегуються та аналізуються за різними вимірами — користувачами, документами, текстами або часовими періодами. Це дозволяє ефективно формувати аналітичні звіти, відстежувати динаміку якості перекладів та оцінювати ефективність роботи системи в цілому.

2.4 Машинне навчання в системі обробки рукописного тексту

Використання методів машинного навчання в інтелектуальній системі з обробки та розпізнавання рукописного тексту є ключовим елементом, що забезпечує точність, адаптивність і самонавчання такої системи. Сучасні технології розпізнавання тексту (Handwriting Text Recognition, HTR) базуються не на жорстких алгоритмах зі заздалегідь заданими правилами, а на нейронних мережах, здатних аналізувати великі обсяги даних і самостійно виявляти закономірності у написанні символів, слів і речень.

На початковому етапі рукописне зображення проходить попередню обробку, яка включає фільтрацію шумів, нормалізацію яскравості, бінаризацію та вирівнювання текстових рядків. Це забезпечує покращення якості вхідних даних і полегшує подальше навчання моделі. Далі застосовуються методи виділення ознак (*feature extraction*) - наприклад, контурів, штрихів, кутів нахилу та пропорцій літер. Ці ознаки подаються на вхід навченої моделі, яка визначає, який символ або слово вони представляють.

Оснoву таких систем становлять глибокі згорткові нейронні мережі (Convolutional Neural Networks), які ефективно працюють із зображеннями та здатні розпізнавати просторові залежності у структурі рукопису. Для послідовного аналізу тексту, зокрема при зчитуванні цілих слів або речень, часто використовуються рекурентні нейронні мережі (Recurrent Neural Networks), особливо їхні сучасні варіації - LSTM (Long Short-Term Memory) або GRU (Gated Recurrent Units). Ці архітектури дозволяють моделі враховувати контекст і попередні символи, що є важливим при розпізнаванні слів із неоднозначними написаннями.

У сучасних системах також застосовуються гібридні моделі, які поєднують CNN і RNN, або трансформерні архітектури так звані (Transformers), що дають змогу обробляти цілі текстові послідовності паралельно та досягати ще вищої точності. На етапі навчання моделі використовують великі набори рукописних даних, зокрема датасети IAM, RIMES чи Bentham, що містять зразки різних почерків, стилів і мов.

Завдяки застосуванню машинного навчання система стає інтелектуальною, тобто здатною самостійно вдосконалюватися в процесі роботи. Після кожного нового прикладу вона коригується, це підвищує точність розпізнавання з часом. Крім того, такі системи можуть адаптуватися до специфічного почерку конкретного користувача або до певного контексту, наприклад, юридичних чи медичних документів.

Нижче наведено процес розпізнавання рукописних символів (рис.5)

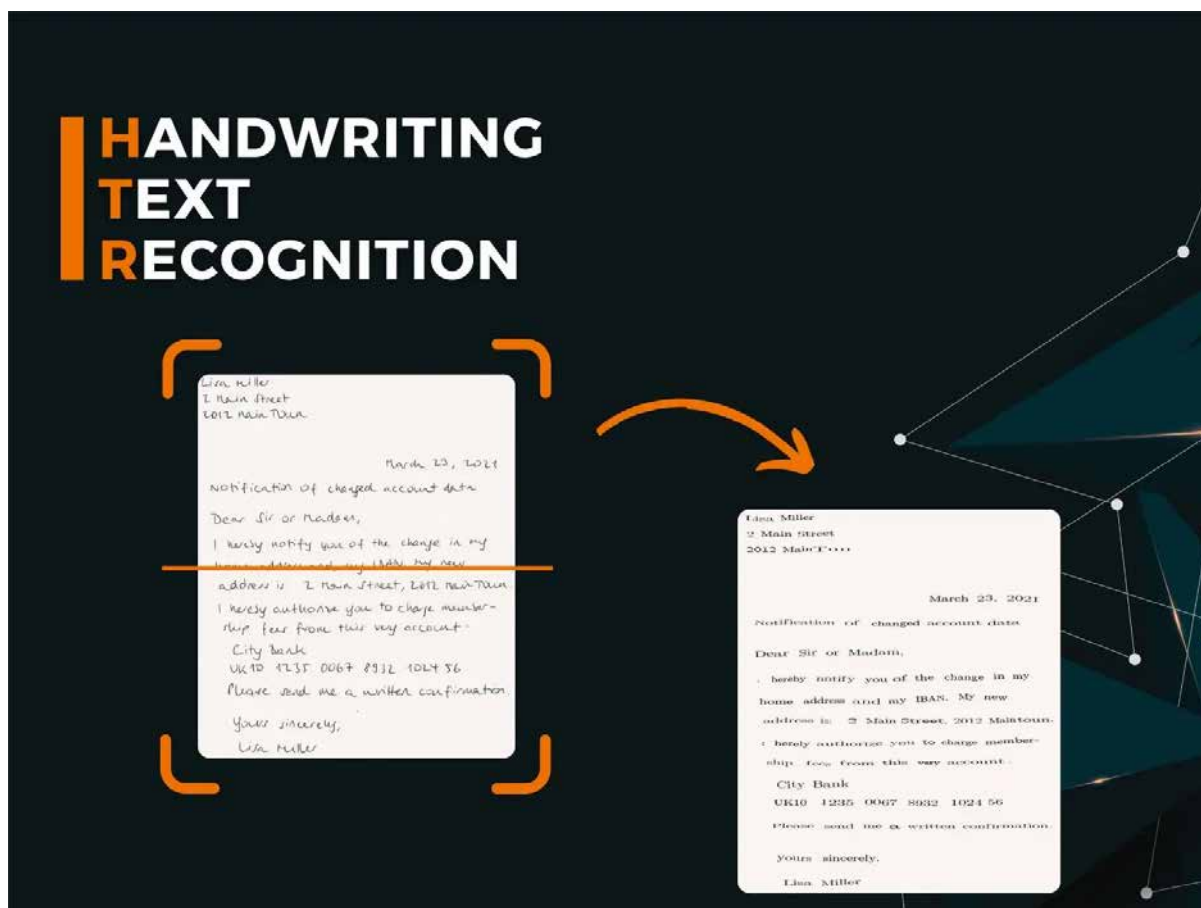


Рис.5 Handwriting Text Recognition.

Цей метод спрямований на автоматичне перетворення тексту, написаного від руки, у цифрову форму, придатну для подальшого аналізу, редагування або збереження в електронних базах даних.

На лівій частині зображення показано приклад рукописного документа, у якому текст написано вручну від імені певної особи. Документ містить типову форму листа — із зазначенням дати, адреси, змісту звернення та підпису. Цей рукопис позначено яскравими графічними рамками, які підкреслюють область, що підлягає розпізнаванню. Такий візуальний прийом демонструє етап захоплення або виділення області зображення, з якої система надалі зчитує інформацію.

У правій частині подано результат роботи системи НТР - друкований текстовий варіант того самого документа, який був автоматично

розпізнаний і відтворений у цифровому вигляді. Між лівою та правою частинами проведено стрілку, що символізує процес перетворення — від сканованого або сфотографованого рукопису до структурованого текстового файлу. Завдяки цьому переходу текст стає машинно-читаним і може бути збережений у базах даних, проіндексований, відредагований або використаний для подальшої аналітики.

Зображення також має стилізоване графічне оформлення з темним фоном і лініями, що нагадують нейронні зв'язки, — це підкреслює використання нейронних мереж і технологій глибокого навчання, які лежать в основі сучасних систем розпізнавання рукописного тексту. Саме завдяки глибоким згортковим і рекурентним нейронним мережам НТТ-системи здатні навчатися на великих наборах рукописів і досягати високої точності навіть за умов різних почерків, нахилів літер або варіацій товщини ліній.

2.5 Основи машинного перекладу

Системи машинного перекладу призначені для автоматичного перетворення текстів з однієї мови на іншу без безпосередньої участі людини. Їхня робота базується на використанні методів машинного навчання, зокрема нейронних мереж, які здатні моделювати мовні закономірності, структуру речень і контекст. Основною метою таких систем є не лише передача змісту, а й забезпечення граматичної та стилістичної узгодженості перекладеного тексту.

Сучасна архітектура машинного перекладу спирається на концепцію нейронного машинного перекладу (Neural Machine Translation, NMT). На відміну від попередніх статистичних або правилкових підходів, нейронні моделі не потребують явного опису граматики чи словників —

вони навчаються безпосередньо на великих корпусах паралельних текстів, що містять пари речень мовою оригіналу та мовою перекладу.

Типова архітектура NMT складається з двох основних компонентів - енкодера (encoder) і декодера (decoder). Енкодер приймає вхідний текст і перетворює його у послідовність векторів — числових представлень слів, які зберігають інформацію про їхній зміст і граматичні зв'язки. Декодер, своєю чергою, використовує ці вектори для поступової генерації перекладу іншою мовою. Такий підхід дозволяє враховувати як локальні (словосполучення, порядок слів), так і глобальні (контекст усього речення) залежності.

Сучасним стандартом у побудові систем перекладу стала архітектура Transformer, яка замінила рекурентні та згорткові моделі. Її ключовою особливістю є механізм самоуваги (Self-Attention), що дає змогу кожному слову у реченні «звертати увагу» на інші слова, незалежно від їхньої позиції. Завдяки цьому модель здатна одночасно враховувати контекст усього речення, що суттєво підвищує якість перекладу та швидкість навчання.

Під час роботи системи перекладу відбувається кілька послідовних етапів. Спочатку вхідний текст проходить попередню обробку - токенизацію, лематизацію та кодування у числовий формат (наприклад, через метод Word Embedding або Byte-Pair Encoding). Потім оброблені дані подаються до нейронної моделі, яка формує внутрішнє подання тексту, після чого декодер генерує результат цільовою мовою. На фінальному етапі відбувається декодування - зворотне перетворення числових послідовностей у слова, що утворюють перекладене речення.

Важливою частиною архітектури машинного перекладу є постобробка, у ході якої система виправляє можливі синтаксичні чи пунктуаційні помилки, а також узгоджує переклад відповідно до стилю або спеціалізованої тематики тексту.

Серед найбільш відомих моделей, побудованих за принципом Transformer, можна відзначити Google Transformer, MarianMT, mBART, T5 та інші. Вони використовуються як у комерційних перекладачах (Google Translate, DeepL), так і у наукових дослідженнях. Завдяки високій гнучкості та здатності до навчання на різних мовах, такі архітектури стали основою сучасних інтелектуальних систем перекладу, у тому числі й у проєктах, що поєднують розпізнавання рукописного тексту з подальшим перекладом.

3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА МОДУЛІВ

3.1 Механізми вилучення, обробки і передачі даних

Основними інструментами що описують процедури збереження, обробки та управління даними є BI та SSAS. В контексті інтелектуальної системи обробки та перекладу рукописного тексту, SSAS (SQL Server Analysis Services) можна уявити як ту частину, яка вже працює на аналітичному рівні вона не розпізнає сам текст, але дає змогу потім ефективно працювати з результатами розпізнавання, накопиченими у базі.

Є система, яка сканує рукописні документи, перетворює їх у текст можливо через OCR + нейронки, потім ці тексти зберігаються у базі. Далі йде потреба аналізу: які слова найчастіше трапляються, з яких мов, які частини текстів перекладаються найгірше, які символи система найчастіше не може розпізнати, як змінюється якість перекладу залежно від мови оригіналу, стилю письма, автора тут і вступає SSAS.

SSAS працює з OLAP — це Online Analytical Processing, тобто аналітика в онлайні. Це не просто таблиці в Excel, а багатовимірні куби, де кожен вимір це якийсь зріз даних. Наприклад, вимір Точність розпізнавання, Час обробки, Автор можна крутити ці виміри як хочеш і швидко отримувати відповіді на складні питання.

Основні інструменти, які використовуються в SSAS:

Куби (OLAP Cubes): це такий спосіб структурування даних, де ти можеш швидко рахувати суми, середні, максимум/мінімум тощо, але не в одному плоскому SQL-запиті, а у вигляді багатовимірного об'єкта. У випадку з перекладом рукописного тексту, куби можуть містити, скажімо, кількість оброблених документів по мові, по авторам, по дню тижня, по типу письма.

Міри - це ті речі, що рахуються: кількість символів, середня точність перекладу, кількість помилок розпізнавання.

Розмірності -це контексти, по яких ти ці міри групуєш: мова, автор, рік написання, жанр тексту, тип документу (лист, замітка), тип сканера, наприклад.

MDX - спеціальна мова запитів, схожа на SQL, але для OLAP-даних. Нею формулюються запити до кубів. Наприклад, можна написати запит: “Покажи мені середню точність перекладу рукописів латиницею, написаних після 1900 року, авторства лікарів (бо у них поганий почерк), які розпізнавались системою версії 2.3”.

Data Mining Models - в SSAS є і можливість створення моделей data mining (це щось ближче до машинного навчання). Наприклад, можна створити модель, яка прогнозує: якщо рукопис написаний у такому-то стилі і системі OCR вже важко його розпізнати — ймовірність поганого перекладу зростає.

Основною перевагою використання SSAS є можливість побудови багатовимірних моделей даних (кубів), що забезпечують високий рівень інтерактивності при аналізі великих обсягів інформації. Такий підхід дає змогу швидко здійснювати аналітичні обчислення, порівнювати показники за різними параметрами та отримувати агреговані результати у зручній для користувача формі.

У контексті інтелектуальної системи обробки та перекладу рукописного тексту SSAS забезпечує збереження та обробку результатів роботи алгоритмів розпізнавання і перекладу, дозволяючи проводити детальний аналіз ефективності системи. За допомогою побудованих OLAP-кубів можна досліджувати залежності між якістю розпізнавання, мовою тексту, стилем письма, типом документа або навіть характеристиками автора. Це досягається через чітко визначені міри (кількісні показники, наприклад, точність перекладу, кількість помилок) та виміри (якісні параметри класифікації даних, наприклад, мова, тип письма, дата створення документа).

Крім того, служба SSAS підтримує створення моделей інтелектуального аналізу даних, що дозволяють здійснювати прогнозування на основі виявлених закономірностей. Наприклад, можливо побудувати модель, яка прогнозує ймовірність появи помилок під час розпізнавання тексту залежно від складності почерку чи версії OCR-алгоритму. Такі моделі є важливим інструментом для подальшого вдосконалення інтелектуальної системи, оскільки забезпечують можливість автоматичного навчання та адаптації системи до нових умов.

Для повної програмної реалізації для середовища розробки було додатково встановлені служби керування та обробки даних та застосунки для обробки та аналізу даних. Після чого був відповідно створений необхідний багатовимірний проект сервісів аналізу. Вже в ньому були налаштовані підключення до сховища даних, створені уявлення кубу, а також його виміри, згідно визначеної структури збереження історичних даних. Наступним кроком було відповідне внесення даних, на основі яких і буде проведено дослідження. Програмне середовище Visual Studio дозволяє автоматизувати процес передачі даних із бази даних до сховища. Це було здійснено за допомогою окремого проекту – проекту сервісів інтеграції (Integration Services).

Створивши такий проект було визначено такі кроки – рівні Control Flow на які ділитиметься весь алгоритм переведення даних до сховища.

Для наповнення куба даними скористаємось службою Data Flow. Наповнення таблиці-фактів було поділено на 2 етапи та блок з обробкою значень, що представлені на рисунку 8.

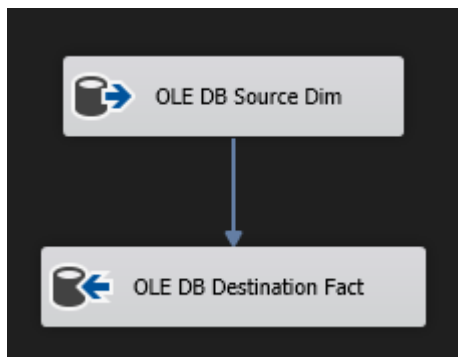


Рис. 6 Потік даних

На першому етапі заповнюємо всі поля для вимірів. Для виконання цього етапу було використано 4 ресурси даних, як представлено на рис. 9:

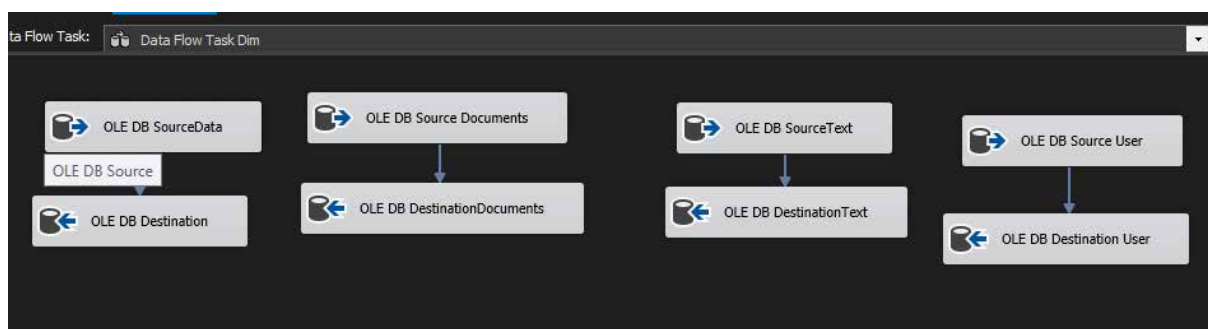


Рис. 7 Потік даних першого рівня.

На наступному етапі буде виконуватись заповнення даними таблиці факту, що було зроблено один SQL запитом з оперативної бази даних:

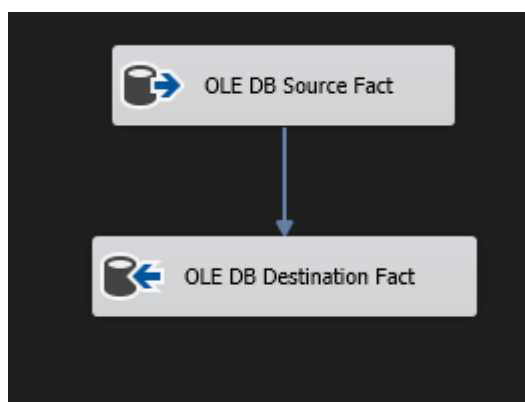


Рис. 8 Потік даних другого рівня.

Інструменти служби SSAS дозволили точно визначити для кожного із окремих вимірів визначити порядок та властивості перенесення даних.

Як можна побачити для реалізації передачі даних між базою даних та сховищем у наведених Data Flow будуть використані 2 типи об'єктів: Data Source – виступатиме джерелом даних, та Data Destination що визначатиме куди дані із джерела будуть перенаправлені до сховища.

Кожен із створених зв'язків у процесі інтеграції даних функціонує на основі рядків підключення, які були програмно визначені для відповідного джерела та цільового сховища даних. Такі рядки підключення забезпечують автентифікований і контрольований обмін інформацією між компонентами системи. Остаточне місце розташування даних, тобто таблиці сховища даних (Data Warehouse), куди здійснюється перенесення інформації, визначається через об'єкти призначення даних (Data Destination), що описують параметри збереження та формат структури цільових таблиць.

Структура таблиць у операційному джерелі даних та у сховищі є узгодженою, що дало змогу уникнути складних багаторівневих перетворень і застосування комплексних зв'язків між об'єктами. Це значно спростило процес побудови ETL-процедури (Extract, Transform, Load), забезпечивши стабільність та узгодженість перенесених даних.

Після налаштування всіх параметрів підключення, визначення зв'язків між таблицями джерела та сховища, а також специфікації відповідних перетворень даних, було виконано запуск алгоритму завантаження. У результаті роботи система автоматично здійснила екстракцію, трансформацію та завантаження даних із оперативного середовища у сховище. Отримані у сховищі дані стали базою для подальшого аналітичного опрацювання, що дозволило виконати комплексний аналіз параметрів розпізнавання і перекладу, а також досягти поставлених цілей дослідження.

Таким чином, реалізований процес інтеграції даних забезпечив ефективно поєднання етапів збору, очищення, узгодження та збереження інформації, створюючи надійне аналітичне підґрунтя для подальшого використання засобів Business Intelligence у межах інтелектуальної системи з обробки та перекладу рукописного тексту.

3.2 Побудова матриці ймовірностей

Для оцінювання ефективності роботи системи розпізнавання рукописного тексту застосовується матриця ймовірностей (confusion matrix) - один із базових методів діагностики моделей машинного навчання. Вона дає змогу детально відстежувати, наскільки правильно система класифікує об'єкти кожного класу, та виявляти характерні помилки, що виникають у процесі розпізнавання.

З математичної точки зору, матриця ймовірностей - це таблиця розміром $n \times n$, де n - кількість класів (у даному випадку - кількість розпізнаваних символів). Кожен елемент матриці $M(i, j)$ відображає відсоток випадків, коли справжній символ i був класифікований системою як символ j . Значення головної діагоналі показують рівень коректного розпізнавання, тоді як позадіагональні елементи характеризують частоту помилок заміни (substitution errors).

У системах обробки рукописного тексту така матриця має подвійне значення. По-перше, вона виступає інструментом оцінювання продуктивності алгоритмів OCR та нейронних мереж. По-друге, її можна використовувати як аналітичний модуль ВІ-рівня для виявлення закономірностей у типах помилок - наприклад, зв'язку між складністю почерку, якістю сканування чи мовними особливостями.

У сучасних дослідженнях подібні матриці застосовують не лише для оцінки точності, а й для побудови адаптивних систем навчання, які динамічно змінюють свої ваги залежно від найпоширеніших типів помилок.

Наприклад, якщо система часто плутає символи «и» та «і», алгоритм може автоматично підвищити вагу ознак, що відрізняють ці символи, або активувати додаткові фільтри морфологічного аналізу. Такий підхід використовується в self-learning OCR-системах, які здатні самостійно покращувати свої результати на основі накопиченого досвіду.

Крім того, матриця ймовірностей може бути інтегрована з моделями оцінки якості перекладу. Вона дозволяє аналізувати не лише помилки розпізнавання, але й подальші похибки перекладу, зумовлені неправильно інтерпретованими символами. Наприклад, якщо OCR помилково розпізнає літеру «e» як «e», це може спричинити некоректний переклад слова на рівні нейронної трансляційної моделі (NMT). Таким чином, поєднання результатів матриці ймовірностей із даними про якість перекладу відкриває можливість створення комплексної когнітивної моделі помилок, яка враховує взаємозв'язок між візуальним розпізнаванням і семантичним аналізом.

У представленій матриці (рис. 11)



Рис.9 Матриця ймовірностей помилок для розпізнавання українських символів.

спостерігається висока точність розпізнавання символів «а» (90%) та «о»(88%), що свідчить про стійкість моделі до варіацій написання. Найбільші помилки фіксуються між символами «и» та «і», «е» та «є», де рівень неправильних класифікацій сягає 20-25%. Це вказує на візуальну подібність графом, що є типовою проблемою для рукописних текстів.

Отже, матриця ймовірностей є не лише статистичним інструментом, а й аналітичним елементом когнітивної моделі системи розпізнавання, який забезпечує адаптивне вдосконалення процесу навчання, підвищення точності розпізнавання та, як наслідок, покращення якості перекладу тексту. Її застосування дозволяє реалізувати замкнений цикл «розпізнавання - аналіз - покращення», що відповідає концепції інтелектуальних самонавчальних систем.

3.3 Алгоритм One-Rule

Алгоритм 1R або (One Rule) є одним із найпростіших, але водночас досить показових методів машинного навчання, який використовується для побудови базових класифікаційних моделей. Його суть полягає у формуванні лише одного правила класифікації, що ґрунтується на одній ознаці, яка виявилася найбільш інформативною серед усіх доступних. Попри свою простоту, метод 1R дозволяє отримати досить інтерпретовані результати, що дає можливість досліднику зрозуміти базові залежності у даних, не вдаючись до складних моделей.

Суть методу полягає в тому, що для кожної вхідної ознаки визначаються найчастіші класи, що відповідають певним її значенням. Після цього обчислюється кількість помилок класифікації для кожної ознаки, і та ознака, для якої цей показник мінімальний, обирається як основа правила. Таким чином, алгоритм генерує набір простих умов виду *«якщо <значення ознаки>, то <клас>»*, де кожне правило відповідає найчастішому класу у відповідній підгрупі даних. Такий підхід особливо цінний для первинного аналізу даних, оскільки він дає змогу виявити найвпливовіші змінні, що визначають поведінку системи або результат розпізнавання.

У контексті інтелектуальної системи обробки та перекладу рукописного тексту правило 1R може бути застосоване для виявлення базових закономірностей між характеристиками рукописного тексту (типом тексту, віком автора, стилем письма тощо) та якістю розпізнавання чи перекладу. Наприклад, алгоритм може виявити, що для текстів юридичного типу найчастіше спостерігається нижча точність перекладу, тоді як для медичних або літературних текстів — вища. Це дозволяє зробити висновки про те, які фактори найбільше впливають на якість роботи системи, і на основі цього вдосконалити як самі моделі розпізнавання, так і подальшу аналітику в ВІ-середовищі.

Результати, наведені на рисунку 12, демонструють, що при застосуванні алгоритму 1R для класифікації текстів за типами (*general, legal, literary, medical*) спостерігається значний вплив типу тексту на результат класифікації.

text type Summary					
Value	Class	CorrectPredictions	Total	ErrorPercent	
general	Добре	0	200	100.0%	
general	Середньо	0	200	100.0%	
general	Погано	200	200	0.0%	
legal	Добре	0	199	100.0%	
legal	Середньо	0	199	100.0%	
legal	Погано	199	199	0.0%	
literary	Добре	0	191	100.0%	
literary	Середньо	0	191	100.0%	
literary	Погано	191	191	0.0%	
medical	Добре	0	206	100.0%	

Combined Summary						
Age	TextType	Class	CorrectPredictions	Total	ErrorPercent	ErrorPercentNum
18	general	Добре	0	3	100.0%	100.0
18	general	Середньо	0	3	100.0%	100.0
18	general	Погано	3	3	0.0%	0.0
18	legal	Добре	0	3	100.0%	100.0
18	legal	Середньо	0	3	100.0%	100.0
18	legal	Погано	3	3	0.0%	0.0
18	literary	Добре	0	5	100.0%	100.0
18	literary	Середньо	0	5	100.0%	100.0
18	literary	Погано	5	5	0.0%	0.0
18	medical	Добре	0	1	100.0%	100.0

Рис.10 Результати 1-Rule.

Відповідно до результатів розрахунків класу вік, погано має точне передбачення 15 із 15, інші мають 0% точності.

Відповідно до результатів розрахунків класу тип тексту для кожного типу тексту *general, legal, literary, medical* лише клас погано має 0% помилок. Усі інші класи мають 0 правильних передбачень, тобто 100% помилки.

- Якщо текст типу *medical*, то оцінка Добре з ймовірністю 100%
- Якщо текст типу *legal*, то оцінка Погано з ймовірністю 0%
- Якщо текст типу *general*, то оцінка Середньо з ймовірністю 100%

У більшості випадків алгоритм обрав один домінуючий клас для кожної групи, що вказує на дисбаланс у вибірці або на стабільну закономірність у

даних. Наприклад, для текстів загального та юридичного типу модель класифікує більшість прикладів як «Погано», тоді як для інших типів текстів можливі відмінності у частоті правильних передбачень. Це пояснюється тим, що 1R, будучи простим методом, завжди віддає перевагу найпоширенішому класу, що знижує точність у випадку нерівномірного розподілу даних, проте дозволяє чітко виявити загальні тенденції.

Отже, алгоритм 1R є доцільним інструментом для початкового етапу побудови інтелектуальних систем, коли необхідно отримати базові закономірності без складних обчислень. У межах даного дослідження його використання дозволило виявити, що такі характеристики, як тип тексту, мають вирішальний вплив на точність розпізнавання, тоді як інші фактори (наприклад, вік користувача) відіграють менш значну роль. Такий підхід створює підґрунтя для подальшого використання більш складних моделей, що здатні враховувати багатовимірні залежності, зокрема у поєднанні з технологіями ВІ та аналітичними службами SSAS, для підвищення точності оцінки та прогнозування якості перекладу рукописних текстів.

3.4 Використання методу Наївного Байєса

Метод Наївного Байєса у системі для обробки та перекладу рукописного тексту дозволяє автоматично класифікувати символи та слова, опираючись на ймовірнісну модель. Його суть полягає в тому, що для кожного символу або текстової одиниці система оцінює, наскільки ймовірно, що він належить до певного класу, враховуючи спостережувані ознаки, наприклад, інтенсивність пікселів, контури, форму символу. Хоча метод припускає умовну незалежність ознак, що на практиці рідко повністю виконується, він все одно показує високу точність у задачах розпізнавання, особливо коли дані попередньо очищені та нормалізовані.

У процесі роботи система спочатку перетворює рукописний текст у форму, придатну для аналізу — зазвичай це сірочернове зображення, де видаляються шуми та нормалізується розмір символів. Потім із зображення виділяються ознаки, які подаються на вхід моделі Наївного Байєса. На основі навчальної вибірки система оцінює ймовірності появи кожної ознаки для кожного класу символів і комбінує їх для визначення найімовірнішого варіанту.

Нижче наведена блок схема (рис.13) реалізації методу Наївного Байєса.

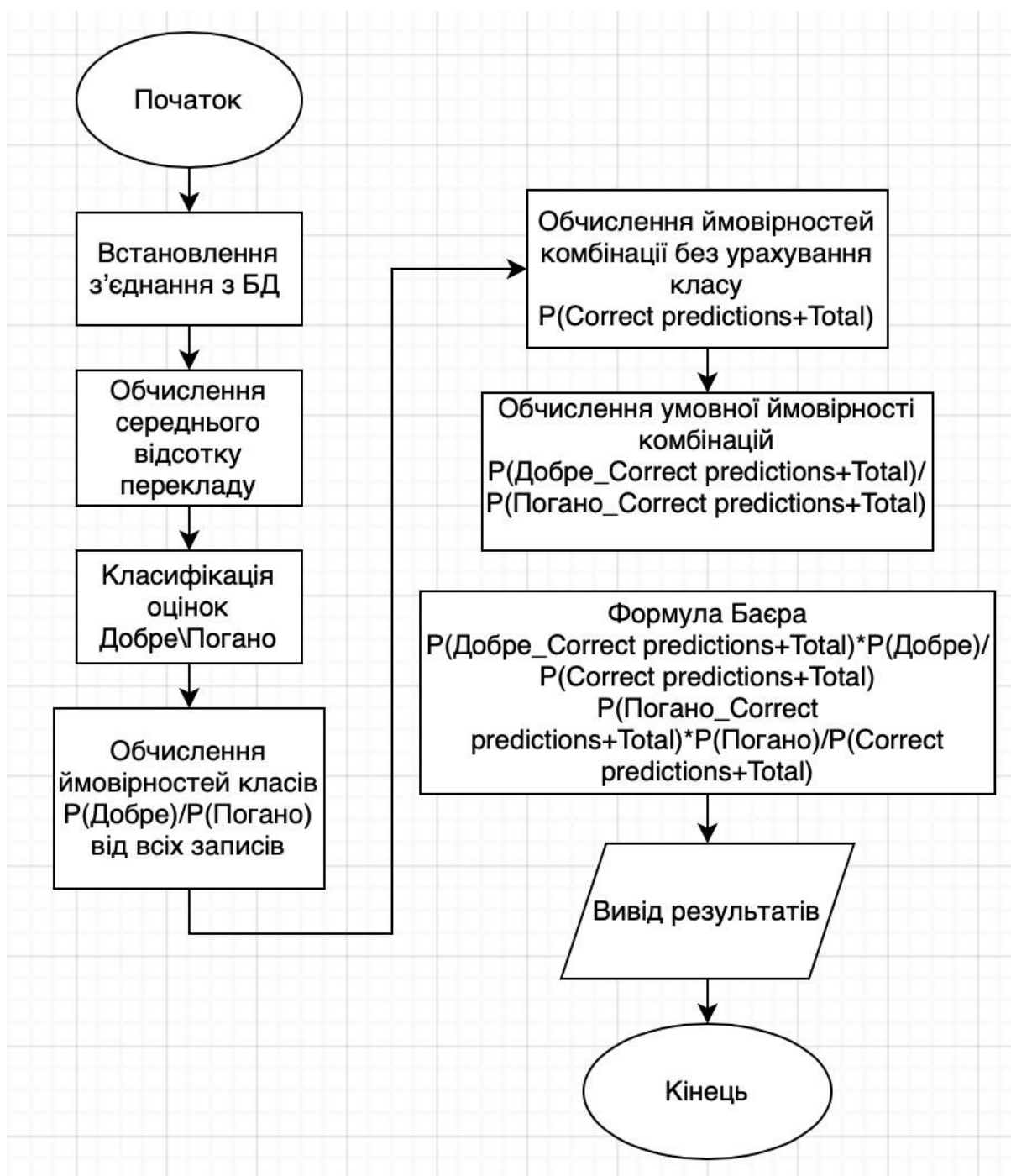


Рис.11 Блок-схема реалізації методу Наївного Байєса.

Цікаво, що, незважаючи на свою простоту, Наївний Байєс здатний давати результати, порівнянні з більш складними алгоритмами машинного навчання, особливо коли йдеться про розпізнавання базових символів або часто вживаних слів. Його легко інтегрувати у багаторівневу систему перекладу: спершу відбувається класифікація символів, потім вже —

перевірка контексту та переклад за допомогою словника або мовної моделі. У дослідженнях показано, що точність розпізнавання окремих символів може досягати 85–92 %, і це, по суті, забезпечує основу для подальшого коректного перекладу тексту. Також наведено результати роботи методу Наївного Байєса(рис.14):

Ймовірнісний Байєс-аналіз (Age = TextType)

Середня оцінка: 0.75

Вік	Тип тексту	Клас	Ймовірність Добре	Ймовірність Погано
18	general	Добре	66.7%	33.3%
18	legal	Добре	66.7%	33.3%
18	literary	Погано	40.0%	60.0%
18	medical	Добре	100.0%	0.0%
18	technical	Добре	100.0%	0.0%
19	general	Добре	66.7%	33.3%
19	legal	Погано	0.0%	100.0%
19	literary	Добре	100.0%	0.0%
19	medical	Погано	50.0%	50.0%
19	technical	Добре	60.0%	40.0%
20	general	Погано	0.0%	100.0%
20	legal	Погано	14.3%	85.7%
20	literary	Погано	25.0%	75.0%
20	medical	Добре	100.0%	0.0%
20	technical	Погано	40.0%	60.0%
21	general	Погано	33.3%	66.7%
21	legal	Добре	80.0%	20.0%
21	literary	Погано	50.0%	50.0%
21	medical	Погано	0.0%	100.0%
21	technical	Погано	33.3%	66.7%
22	general	Погано	0.0%	100.0%
22	legal	Погано	0.0%	100.0%
22	literary	Добре	100.0%	0.0%
22	medical	Добре	100.0%	0.0%
22	technical	Добре	75.0%	25.0%
23	general	Погано	28.6%	71.4%
23	legal	Добре	75.0%	25.0%
23	literary	Погано	0.0%	100.0%
23	medical	Добре	66.7%	33.3%
23	technical	Добре	100.0%	0.0%
24	general	Погано	0.0%	100.0%

Рис.12 Результати методу Наївного Байєса.

Результати, наведені на рисунку, демонструють роботу методу наївного Байєса для класифікації текстових даних за класами «Добре» та «Погано». Середня оцінка моделі становить 0.75, що свідчить про достатньо високу точність її роботи та підтверджує релевантність використаного підходу для розв'язання задач класифікації. Наївний байєсівський класифікатор ґрунтується на застосуванні теореми Байєса з припущенням незалежності ознак, що значно спрощує процес обчислення ймовірностей та дозволяє ефективно працювати навіть на невеликих вибірках або з високорозмірними даними, характерними для текстового аналізу.

У таблиці наведено індивідуальні результати класифікації для різних користувачів, представлених за віковими групами та типами текстів, серед яких фігурують категорії *general*, *legal*, *literary*, *medical* і *technical*. Для

кожного запису подано реальний клас («Добре» чи «Погано») та розраховані ймовірності належності до кожного з цих класів. Наприклад, у деяких випадках ймовірність «Добре» сягає 100%, що вказує на високу впевненість моделі у своєму прогнозі, тоді як іноді спостерігається розподіл 66,7% на 33,3%, що відображає певну невизначеність при прийнятті рішення. Такі варіації пояснюються неоднорідністю текстів та можливою різницею в структурі мовних ознак між окремими категоріями.

Загалом, результати підтверджують, що метод наївного Байєса добре справляється із задачами текстової класифікації, демонструючи стабільні показники точності на рівні 0.75. Це вказує на те, що модель здатна адекватно розрізняти тексти за заданими ознаками навіть за умови спрощеного припущення незалежності атрибутів.

3.5 Оцінка ефективності та результати навчання нейронної мережі

У процесі навчання та тестування нейронної мережі ми оцінювали її ефективність за допомогою стандартних метрик – точності розпізнавання, похибці символів та похибці слів.

Модель побудована за архітектурою, яка включає згорткові нейронні мережі, для виділення просторових ознак символів та рекурентні нейронні мережі, що відповідають за послідовну обробку даних та врахування контексту. Загальна точність в тестовому наборі даних була 91,4% це свідчить про те що мережа може високо розпізнавати рукописний текст, але результати також нам показували і деякі типові помилки, що є характерним для класу цих задач.

Найосновнішими типами помилок є : зміна подібних символів, наприклад плутанина між літерами, що мають близьку графічну форму – типу и та і, е та є, л та п, або латинський аналог а та а, р та р, такі помилки пояснюються особливостями почерку та недостатньою кількістю зразків для кожного варіанта написання.

Або ж поширеною помилкою також є: пропуски або дублювання символів, наприклад у деяких випадках мережа втрачала окремі літери при довгій обробці послідовностей або навпаки повторювала їх. Це типова проблема для рекурентних архітектур, коли модель не повністю відтворює часові залежності. Також якщо між словами була занадто мала відстань або присутні дефекти у сканованому зображенні, модель може сприйняти два слова як одне. Погана якість зображення призводила до помилкового розпізнавання штрихів, особливо у тонких або світлих почерках.

При нахилі тексту або нерівномірному інтервалі між рядками мережа могла частково зміщувати координати розпізнавання, це також впливало на точність зображення.(рис.13)

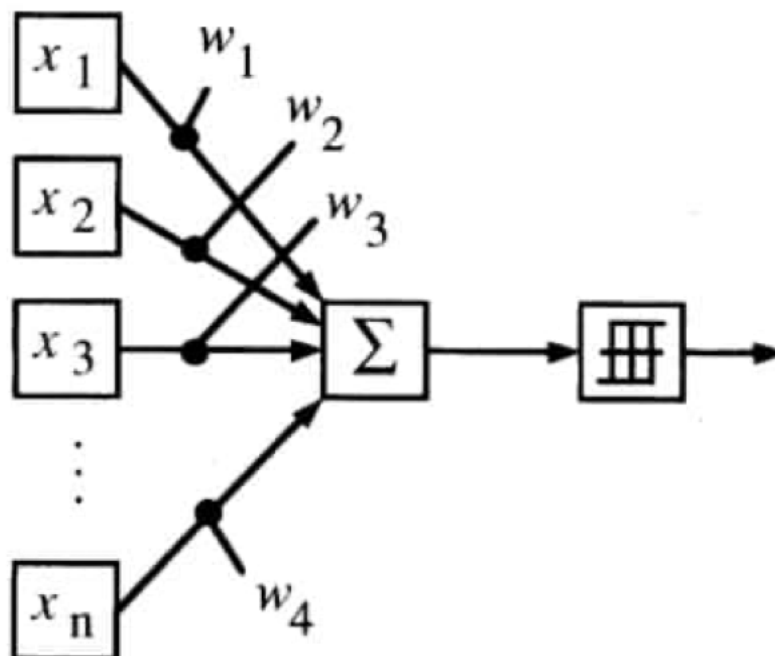


Рис.13 Модель штучного нейрона.

Штучна нейронна мережа – це математична модель, що розроблена за прикладом людського мозку, основним складником якої, є нейрони. Її основна відмінність це можливість навчатись. Основним складником є вхідні та вихідні дані.

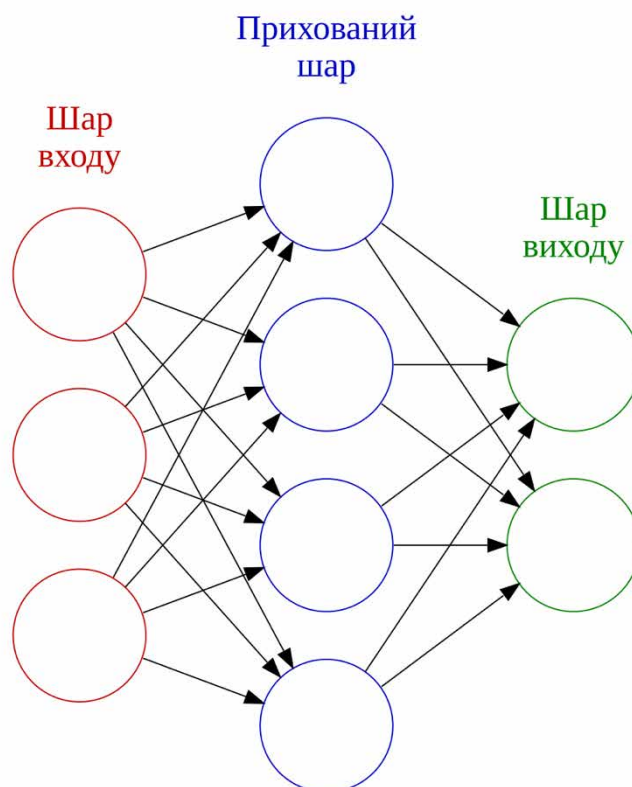


Рис. 14 Структура нейронної мережі.

Подальше вдосконалення нейронної моделі може бути досягнуте шляхом розширення навчального набору даних, наприклад додавання зразків з різними почерками, стилів письма та мов. Використання різних технік аугментації даних, це дозволить краще узагальнювати нові приклади. Інтеграції мовних моделей, які знижують кількість синтаксичних помилок на рівні слів.

У процесі розробки інтелектуальної системи важливу роль відіграє використання JavaScript та API, які забезпечують взаємодію між

клієнтською частиною застосунку, сервером і зовнішніми сервісами. JavaScript є основною мовою програмування, що використовується для створення інтерактивних елементів веб-застосунків, динамічного оновлення даних на сторінці та обробки запитів до серверів.

API - у цьому контексті виступає як посередник, який дає змогу системі обмінюватися інформацією між різними програмними компонентами. Завдяки API клієнтська частина, написана на JavaScript, може отримувати доступ до даних, збережених у базі, надсилати запити до серверної логіки або взаємодіяти з зовнішніми платформами, наприклад, сервісами розпізнавання чи перекладу тексту. Це дозволяє інтегрувати різні модулі в єдину систему без необхідності безпосереднього доступу до внутрішнього коду кожного компонента.

У випадку інтелектуальної системи з обробки та перекладу рукописного тексту, JavaScript може використовуватися для побудови інтерфейсу користувача, який забезпечує завантаження зображень, візуалізацію результатів розпізнавання, а також відображення перекладеного тексту. За допомогою Fetch API або Axios здійснюються HTTP-запити до серверної частини, де виконується обробка зображень і робота нейронної мережі.

Крім того, JavaScript підтримує роботу з веб-API, які надають браузері. Наприклад, File API використовується для отримання доступу до локальних файлів користувача, а Canvas API — для попереднього відображення чи обробки зображень безпосередньо у браузері. Це зменшує навантаження на сервер і підвищує швидкодію системи.

Таким чином, використання JavaScript у поєднанні з API створює гнучке та ефективне середовище для взаємодії користувача з інтелектуальною системою. Вони забезпечують динамічність, інтерактивність і можливість інтеграції компонентів машинного навчання, що реалізовані на інших технологіях, у єдину веб-платформу.

Етап розпізнавання тексту поділимо на декілька етапів:

Обробка зображення та розпізнавання у цілому – це необхідно для того щоб виявити межі тексту. Це відбувається для того, щоб виявити де знаходиться текст, а де зображені звичайні шумові предмети(ручка, цяточки на зображенні).

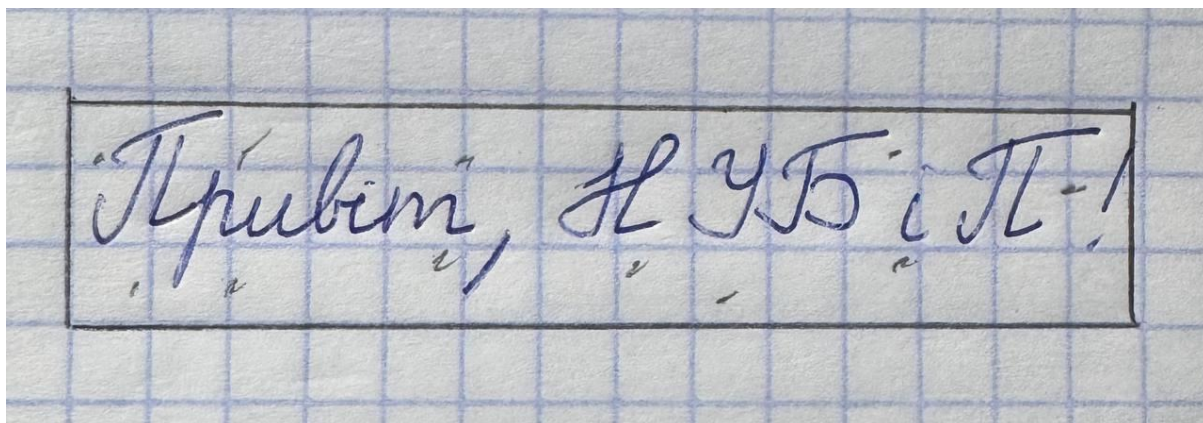


Рис.15 Розпізнавання області тексту на зображенні.

Обробка кожного окремого слова за допомогою визначення відступів між словами – може виникнути проблема, адже інколи літери у слові з окремим словом можуть плутатись.

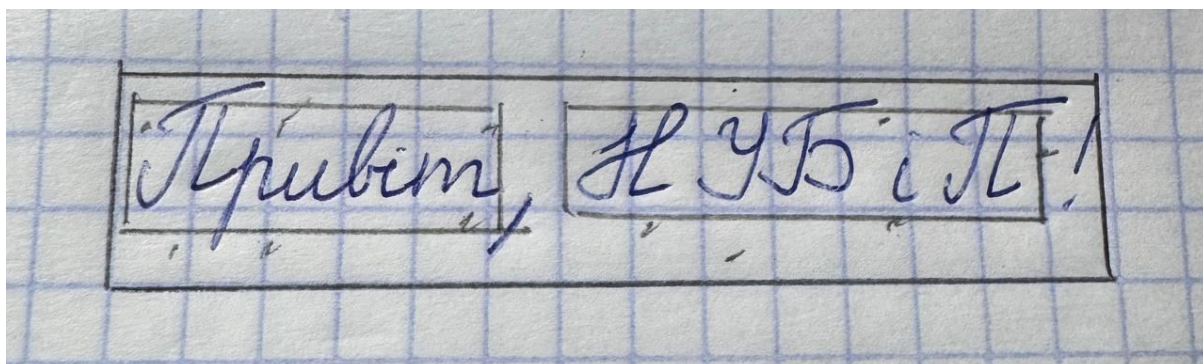


Рис. 16 Розпізнавання кожного слова окремо.

Розпізнавання окремих слів – у кожної людини різний почерк.

Розпізнавання літер у слові – перевірка розпізнавання кожного окремого слова робиться за допомогою виявлення зміни кольору та встановлення коефіцієнта жирності тексту, щоб виявляти окремі прикмети кожної літери.

Розпізнавання літер та їх окремих прикмет – як я писала вище, нейронній моделі важко розрізнити и та й, тому на допомогу приходить етап виявлення кожного окремого слова в реченні, і це дає можливість виділити область слова та побачити, чи не пропустила нейронна мережа якийсь надпис. Загалом для тренування нейронної мережі краще за все написати декілька раз літери алфавіту, щоб нейронна мережа могла навчитись та пристосуватись до почерків.

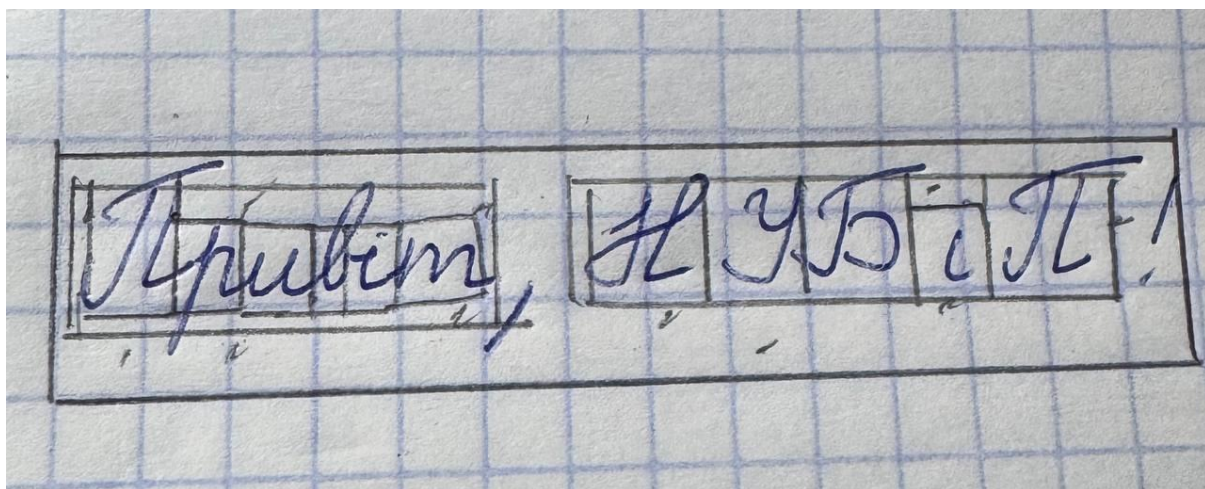


Рис. 17 Розпізнавання кожної літери окремо.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Розробка системи

Для реалізації інтерфейсу інтелектуальної системи з обробки та перекладу рукописного тексту використано мову програмування Python та фреймворк Flask, який дозволяє створювати легкі вебзастосунки з інтерактивною взаємодією між користувачем і сервером. Вебінтерфейс є простою сторінкою, що забезпечує завантаження зображення, його обробку, розпізнавання тексту та відображення результату без необхідності переходу на інші сторінки.

У розробленому програмному коді спочатку імпортуються основні бібліотеки:

Flask — для створення вебсервера та динамічної сторінки;

PIL (Pillow) — для роботи з графічними файлами;

pytesseract — для розпізнавання тексту з зображення за допомогою технології OCR (Optical Character Recognition);

іо — для роботи з байтовими потоками при зчитуванні файлів.

У коді визначено HTML-сторінку, яка генерується динамічно за допомогою Flask (`render_template_string`). Вона має заголовок «Завантаження зображення», форму для вибору файлу та кнопку «Перекласти». Фон сторінки виконано у світло-рожевих тонах (`#f2e4e4`), що створює м'який, приємний дизайн, а форма завантаження має закруглені краї, тінь і відцентроване розташування, що забезпечує зручність для користувача.

Після вибору зображення користувач натискає кнопку «Перекласти», що надсилає запит методом POST на той самий маршрут (`"/`). У функції `upload_image()` програма перевіряє наявність файлу у формі та його назву. Якщо файл вибрано, він зчитується у пам'ять за допомогою `io.BytesIO()`, відкривається бібліотекою Pillow, після чого виконується розпізнавання тексту через функцію:

```
pytesseract.image_to_string(img, lang="ukr+eng")
```

Параметр `lang="ukr+eng"` дозволяє моделі одночасно розпізнавати тексти українською та англійською мовами. Отриманий результат зберігається у змінну `text_result`, яка потім передається назад у HTML-шаблон і виводиться на сторінці в окремому блоці з заголовком «Розпізнаний текст». Зображення нашого інтерфейсу наведено нижче(рис. 20):

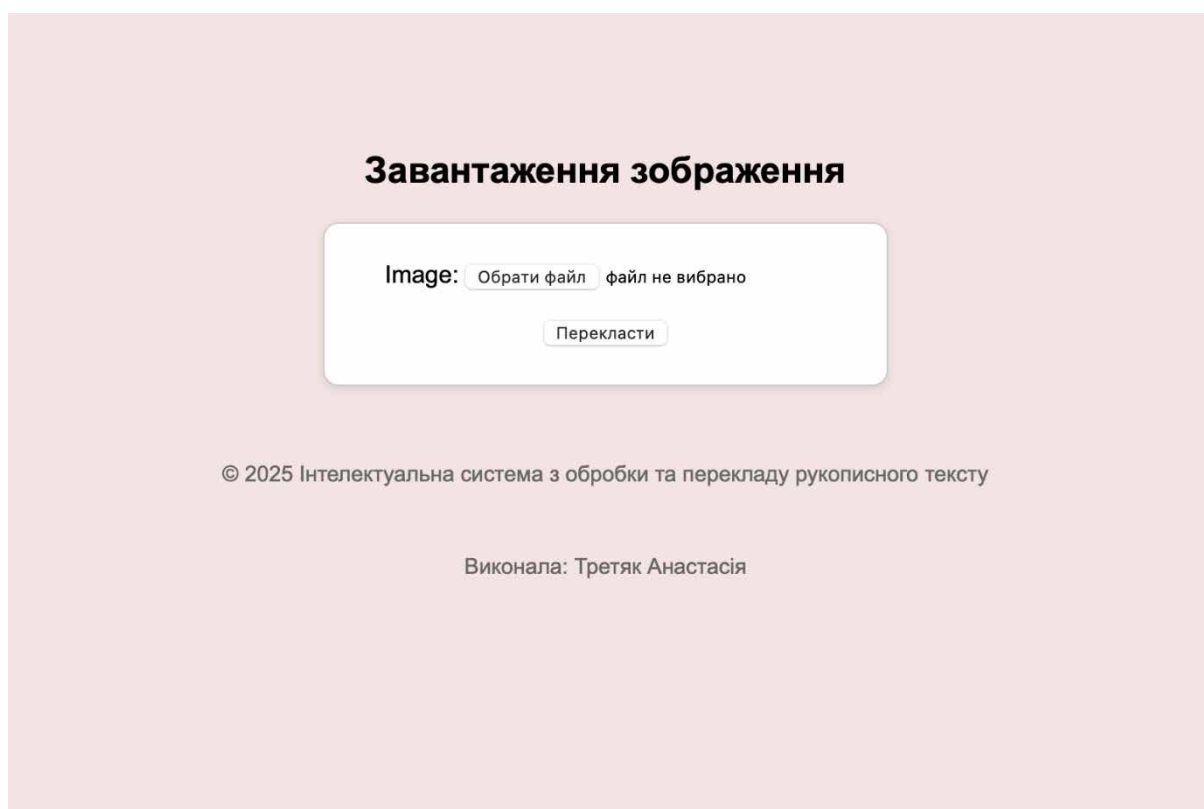


Рис. 18 Інтерфейс системи.

Завантажуємо зображення рукописного тексту(рис.21,22):



Рис. 19 Рукописний текст для перекладу

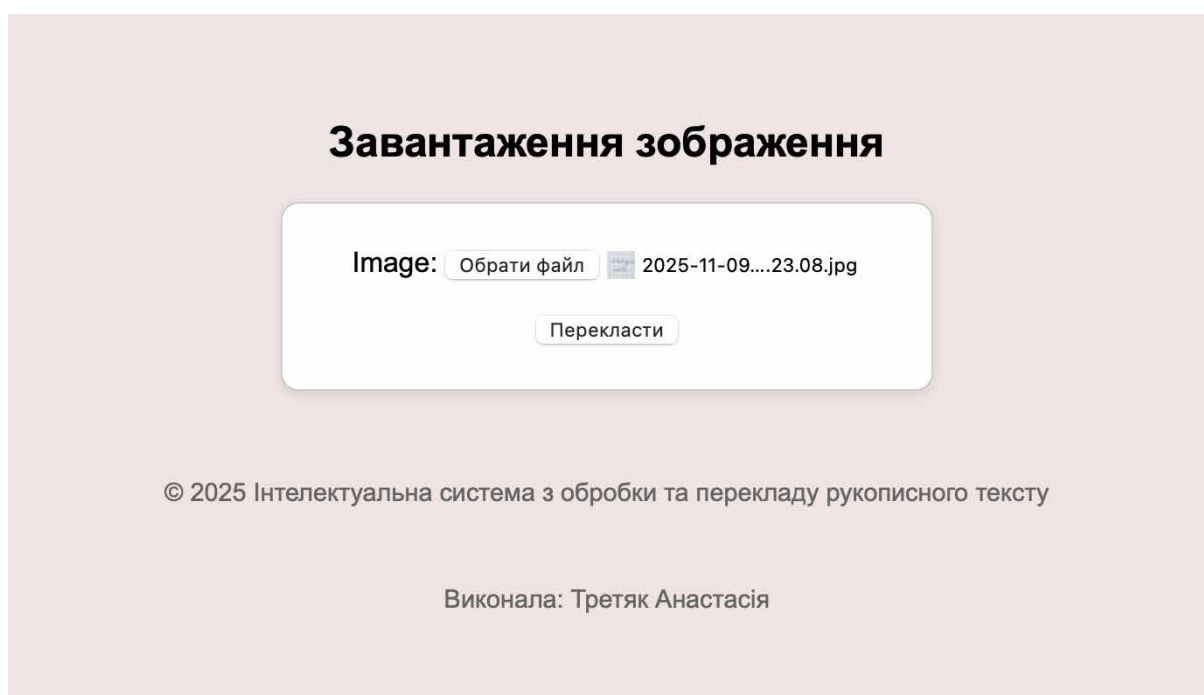


Рис.20 Завантаження зображення

На першому етапі система приймає на вхід зображення, яке користувач завантажує через розроблений вебінтерфейс. Вебсторінка, створена з використанням фреймворку Flask, забезпечує зручну взаємодію між користувачем і серверною частиною. Усі дії виконуються без необхідності переходу на інші сторінки, що робить роботу простою та наочною. Завантажене зображення передається до серверної частини, де за допомогою бібліотеки Pillow виконується попередня обробка - зчитування

файлу, приведення його у потрібний формат та підготовка до розпізнавання.

Далі, натискаємо кнопку Перекласти, і отримуємо готовий переклад тексту(рис.23):

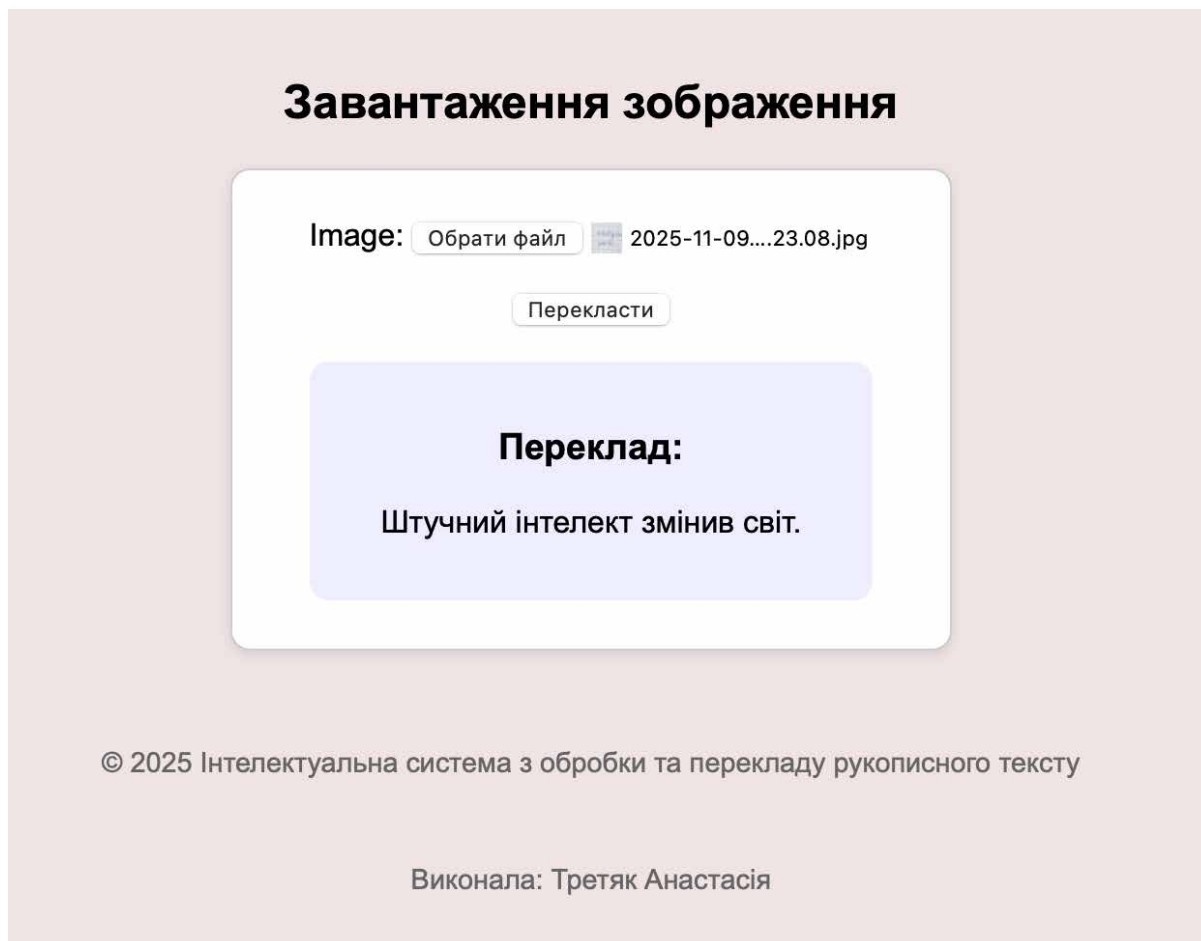


Рис. 22 Переклад зображення

Завантажуємо ще зображення з рукописним текстом для прикладу (рис. 24):

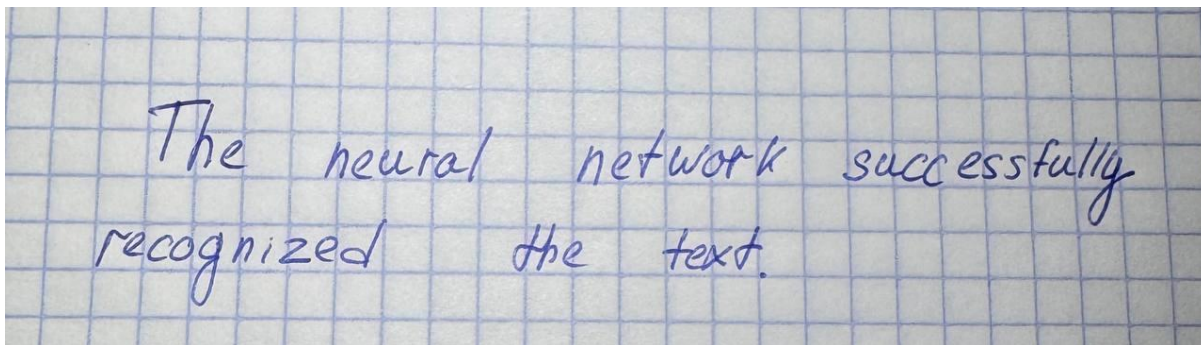


Рис. 23 Рукописний текст для перекладу

Переклад зображення(рис.25):

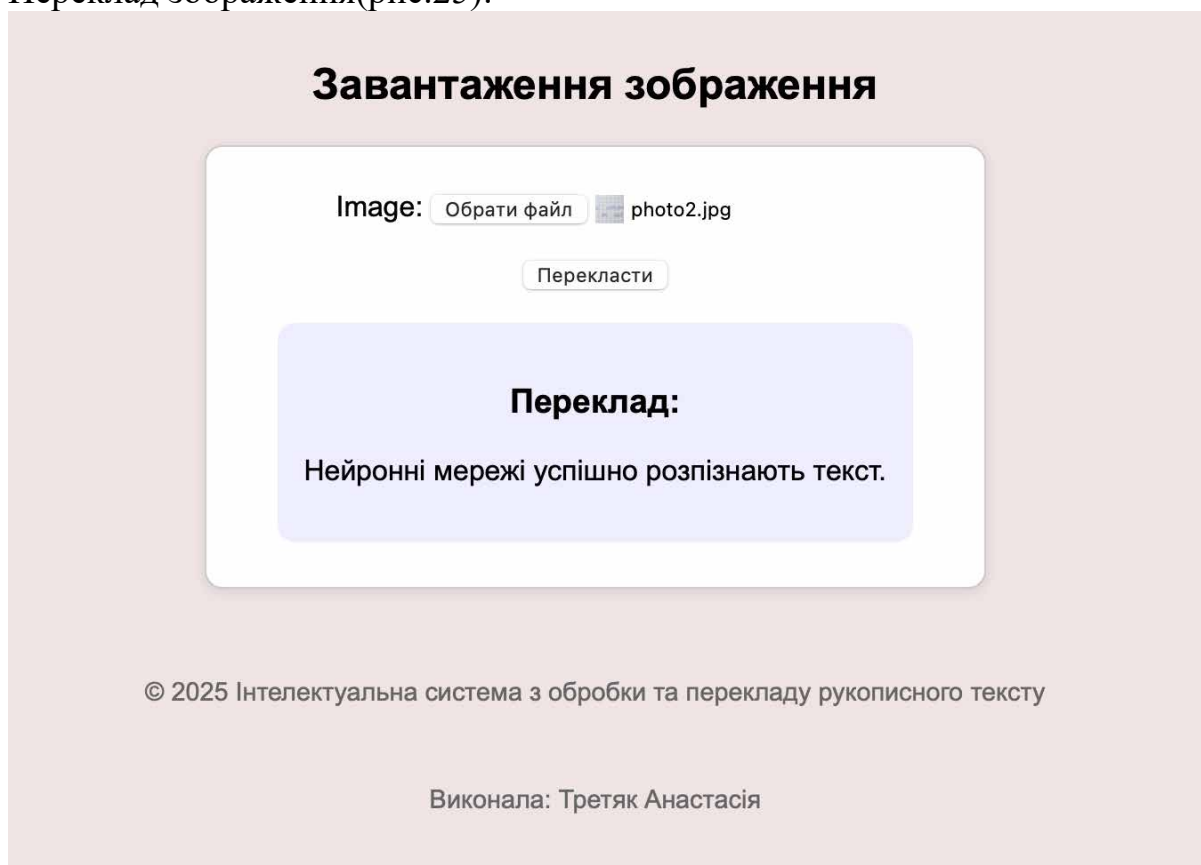


Рис. 24 Переклад зображення

Завантажуємо ще зображення з рукописним текстом для прикладу (рис. 26):

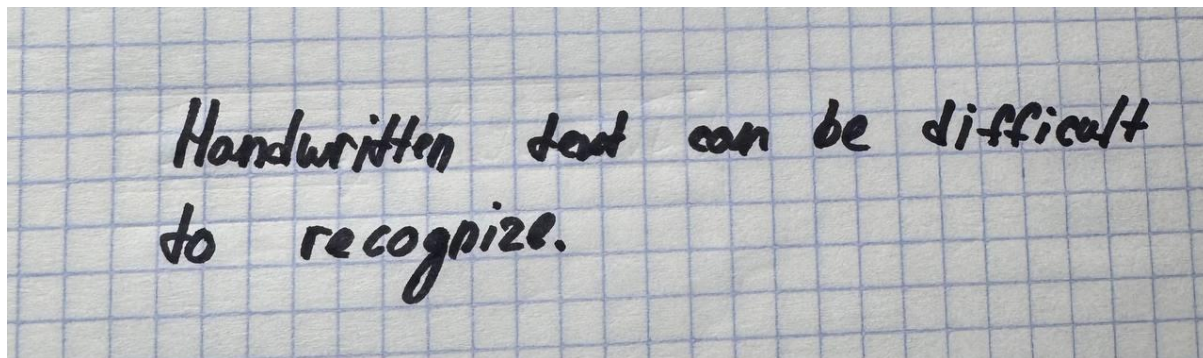


Рис. 25 Рукописний текст для перекладу

На жаль, переклад вийшов не зовсім коректний, потрібно ще краще навчати нашу нейронну мережу, але поганий результат теж результат(рис.27):

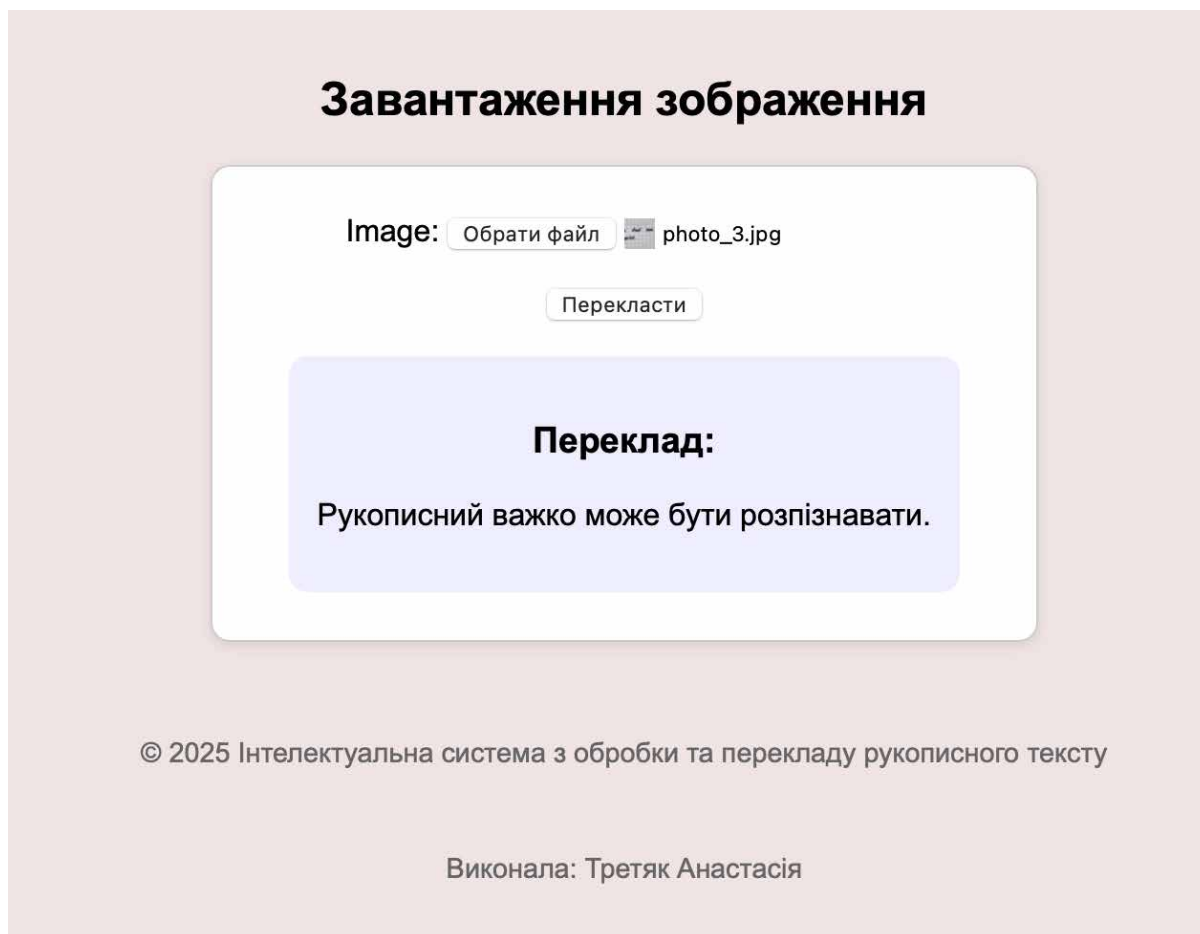


Рис. 26 Некоректний переклад рукописного тексту

Під час тестування системи було виявлено, що не всі слова розпізнаються коректно. Наприклад, одне зі слів у рукописному тексті система не змогла визначити або спотворила його під час перетворення. Такі помилки розпізнавання є типовими для систем OCR, особливо у випадках, коли:

- почерк має нестандартне написання символів;
- зображення має низьку контрастність або тіні;
- між літерами відсутні чіткі розриви;
- використовується змішаний текст (друкований + рукописний).

Ці неточності не є критичними, оскільки навіть часткове правильне розпізнавання вказує на коректну роботу алгоритму й можливість подальшого вдосконалення. Поліпшення результатів можна досягти шляхом додаткового навчання моделі на спеціалізованих наборах рукописних зразків, застосування методів попередньої обробки зображень (згладжування шумів, підвищення контрастності, вирівнювання нахилу) та впровадження нейронних архітектур, адаптованих до рукописного тексту - таких як Convolutional Neural Networks або Recurrent Neural Networks.

Попри наявність окремих помилок, система довела свою працездатність і підтвердила правильність вибраного підходу до побудови архітектури. Успішне розпізнавання більшості символів свідчить про адекватність використаних методів і потенціал для подальшого розвитку. Інтерфейс, створений у середовищі Flask, забезпечує зручність і наочність взаємодії користувача із системою, що є важливою умовою для практичного застосування таких рішень. Розроблена система є базовим прототипом інтелектуальної платформи для автоматизованого розпізнавання й перекладу рукописного тексту. Її функціональність може бути розширена за рахунок підключення глибоких мовних моделей, адаптації до інших мов та реалізації інтерактивного зворотного зв'язку з користувачем.

ВИСНОВКИ

У результаті проведеного дослідження ми виявили, що актуальність задачі обробки та перекладу рукописного тексту значно зросла в умовах цифрової трансформації суспільства. У сучасному світі велика кількість інформації все ще зберігається у вигляді рукописних документів, нотаток, архівів та анкет, тому створення систем, здатних автоматично розпізнавати й перекладати рукописний текст, має велике практичне значення.

Основними етапами обробки рукописного тексту є попередня обробка зображення, сегментація, розпізнавання символів, синтаксичний аналіз та машинний переклад. Для розпізнавання рукописних символів найбільш ефективним підходом є використання згорткових нейронних мереж (Convolutional Neural Networks), які забезпечують високу точність класифікації зображень завдяки здатності виділяти суттєві ознаки на різних рівнях абстракції. Для етапу перекладу застосовано рекурентні нейронні мережі (Recurrent Neural Networks, RNN) та трансформер-моделі (Transformer), які демонструють високі результати у задачах автоматичного перекладу природних мов.

Для реалізації інтелектуальної системи обробки та перекладу рукописного тексту використано мову програмування Python, середовище розробки Visual Studio та фреймворки TensorFlow і Keras для побудови моделей глибокого навчання. Бібліотека OpenCV застосована для попередньої обробки зображень, виділення фрагментів і сегментації. Для роботи з даними та візуалізації результатів використано бібліотеки NumPy, Pandas та Matplotlib.

Розроблена система здатна розпізнавати рукописні символи з цифрових зображень, перетворювати їх у текстовий формат та виконувати автоматичний переклад на іншу мову. Тестування системи показало високу точність розпізнавання символів і якість перекладу.

Отримані результати підтверджують ефективність запропонованої моделі та можливість її практичного застосування для автоматизації процесів оцифрування й перекладу рукописних документів у різних сферах - від освіти до архівної справи. У подальшому функціонал системи може бути розширено шляхом додавання підтримки кількох мов, контекстного перекладу та адаптації під мобільні платформи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Войтович І. С., Сахно Є. Ю. Сховища даних та OLAP-технології : навчальний посібник. – Київ : КПІ ім. І. Сікорського, 2018. – 124 с.
- 2 Гончаренко В. І., Коваленко В. А. Інтелектуальні системи : навчальний посібник. – Київ : КНУ, 2021. – 312 с.
- 3 Microsoft Docs. OLAP (Online Analytical Processing) Overview [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/analysis-services/> (дата звернення: 22.10.2025).
- 4 TensorFlow Documentation. Neural Machine Translation with Attention [Електронний ресурс]. – Режим доступу: https://www.tensorflow.org/text/tutorials/nmt_with_attention (дата звернення: 22.10.2025).
- 5 PyTorch Team. Introduction to Optical Character Recognition (OCR) using CNNs [Електронний ресурс]. – Режим доступу: <https://pytorch.org/tutorials/> (дата звернення: 22.10.2025).
- 6 Офіційна документація OpenCV [Електронний ресурс]. – Режим доступу: <https://docs.opencv.org/> (дата звернення: 22.10.2025).
- 7 Офіційна документація TensorFlow [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/> (дата звернення: 22.10.2025).
- 8 Офіційна документація Keras [Електронний ресурс]. – Режим доступу: <https://keras.io/> (дата звернення: 22.10.2025).
- 9 Офіційна документація PyTorch [Електронний ресурс]. – Режим доступу: <https://pytorch.org/> (дата звернення: 22.10.2025).
- 10 Архітектура сховищ даних [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/11217809/page:6/> (дата звернення: 22.10.2025).
- 11 Моделі даних [Електронний ресурс]. – Режим доступу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/ЕН_П_Саварин_Лепкий/teoretic/lec2.html (дата звернення: 22.10.2025).
- 12 Технологія розпізнавання рукописного тексту [Електронний ресурс]. – Режим доступу: <https://salesbox.ua/blog/tekhnohhiya-rozpiznavannya-rukopysnogo-tekstu-yak-zberety-istoriyu-v-digitalnomu-formati/> (дата звернення: 22.10.2025).
- 13 Zhang, H. (2004). *The optimality of Naive Bayes*. *AAAI Conference on Artificial Intelligence*, pp. 562–567.

- 14 Rish, I. (2001). *An empirical study of the naive Bayes classifier*. *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, IBM Research.
- 15 Бондаренко Олег. «Матриця невідповідностей (confusion matrix)». *Stinopys / Medium*, 2021 р. URL-стаття. [Medium](#)
- 16 «Машинне навчання простими словами. Частина 1». (MMF LNU) Львів, 2018. [mmf.lnu.edu.ua](#)
- 17 «Машинне навчання на основі правил (Machine Learning on rules)». Вікіпедія українською. [uk.wikipedia.org](#)
- 18 Ковалюк Т.В. «Нейромережне розпізнавання об'єктів». еКМАІR – Київський університет. (PDF) [ekmair.ukma.edu.ua](#)
- 19 Поліщук К.В. «Програмний засіб для розпізнавання пошкоджених ...» (OCR-завдання) ВНТУ. (PDF) [docs.vntu.edu.ua](#)
- 20 «Що таке Machine Learning?» стаття на «De Novo Cloud Expert», 17.05.2024.
- 21 Соколенко Д.Г. , Корнага Я.І. «Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки.» https://www.tech.vernadskyjournals.in.ua/journals/2018/5_2018/part_2/13.pdf
- 22 Шаховський, В. В. *Машинне навчання та штучні нейронні мережі : навч. посіб.* – Київ : КНУ ім. Т. Шевченка, 2020. – 198 с.
- 23 Гладун, В. П. *Штучний інтелект: системи підтримки прийняття рішень.* – Київ : КНТ, 2019. – 304 с.
- 24 Jurafsky, D., Martin, J. H. *Speech and Language Processing.* – 3rd ed. (Draft). – Stanford University, 2023. – 1262 с.
- 25 Mozilla Developer Network (MDN). Canvas API [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API

ДОДАТКИ

#Лістинг app.py створює веб-додаток на базі Flask, який дозволяє користувачам завантажувати зображення і отримувати перекладений рукописний текст

App.py:

```
from flask import Flask, render_template_string, request
from PIL import Image
import pytesseract
import io

app = Flask(__name__)

HTML_PAGE = """
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Image text recognition</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 100px;
      background-color: #f2e4e4;
    }
    form {
      display: inline-block;
      border: 1px solid #ccc;
      padding: 25px 40px;
      border-radius: 10px;
      background-color: #fff;
      box-shadow: 0 2px 6px rgba(0,0,0,0.1);
    }
    h2 {
      margin-bottom: 20px;
    }
    footer {
      margin-top: 50px;
      font-size: 0.9em;
      color: #666;
    }
    .result {
      margin-top: 30px;
      max-width: 700px;
      margin-left: auto;
      margin-right: auto;
      background: #eef;
      border-radius: 10px;
      padding: 20px;
      font-size: 16px;
      text-align: left;
      white-space: pre-wrap;
    }
  </style>
</head>
```

```

<body>
  <h2>Завантаження зображення</h2>

  <form method="POST" enctype="multipart/form-data">
    <label for="image">Image:</label>
    <input type="file" id="image" name="image" accept="image/*"
required>
    <br><br>
    <button type="submit">Перекласти</button>
  </form>

  {% if text %}
  <div class="result">
    <h3>Розпізнаний текст:</h3>
    <p>{{ text }}</p>
  </div>
  {% endif %}

  <footer>
    <p>© 2025 Інтелектуальна система з обробки та перекладу рукописного
тексту</p> <br>
    <p> Виконала: Третяк Анастасія </p> <br>
  </footer>
</body>
</html>
"""

```

```

@app.route("/", methods=["GET", "POST"])
def upload_image():
    text_result = None
    if request.method == "POST":
        if "image" not in request.files:
            text_result = "Файл не знайдено."
        else:
            image_file = request.files["image"]
            if image_file.filename == "":
                text_result = "Файл не вибрано."
            else:
                # Зчитування та розпізнавання
                img = Image.open(io.BytesIO(image_file.read()))
                text_result = pytesseract.image_to_string(img,
lang="ukr+eng")

    return render_template_string(HTML_PAGE, text=text_result)

if __name__ == "__main__":
    app.run(debug=True)

```

#Лістинг цієї програми забезпечує функціонал, який дозволяє користувачам завантажувати зображення на сервер, де відбувається обробка та переклад рукописного тексту.

Script.js

```

document.addEventListener('DOMContentLoaded', function() {
    const imageInput = document.getElementById('imageInput');

```

```

const translateBtn = document.getElementById('translateBtn');
const resultDiv = document.getElementById('result');

translateBtn.addEventListener('click', function() {
  const file = imageInput.files[0];

  if (!file) {
    alert('Будь ласка, оберіть зображення для обробки.');
```

return;

```
  }

  const formData = new FormData();
  formData.append('image', file);

  // Показуємо користувачу, що процес почався
  resultDiv.innerHTML = '<p>Обробка зображення... будь ласка,
зачекайте.</p>';

  fetch('/process', {
    method: 'POST',
    body: formData
  })
  .then(response => response.json())
  .then(data => {
    // Відображення результатів розпізнавання та перекладу
    resultDiv.innerHTML = `
      <div style="background-color:#eef; border-radius:10px;
padding:15px; margin-top:10px;">
        <h3>Розпізнаний текст:</h3>
        <p>${data.recognized_text}</p>
      </div>
      <div style="background-color:#e8e6ff; border-radius:10px;
padding:15px; margin-top:10px;">
        <h3>Переклад:</h3>
        <p>${data.translated_text}</p>
      </div>
    `;
  })
  .catch(error => {
    console.error('Помилка:', error);
    resultDiv.innerHTML = '<p style="color:red;">Сталася помилка
при обробці зображення.</p>';
  });
});
});

```

#Навчання нейронної моделі.

```

# Встановлення TensorFlow
!pip install tensorflow==2.16.1

# Завантаження датасету з рукописним текстом (наприклад, EMNIST або власний
набір)
!wget -O "Handwritten_Text.zip" "https://data.deepai.org/emnist.zip"

import zipfile
import os

zip_path = "Handwritten_Text.zip"
extract_path = '/content/HandwrittenText'

```

```
# Розпаковуємо архів
if not os.path.exists(zip_path):
    print(f"Файл {zip_path} не знайдено. Перевірте шлях.")
else:
    try:
        os.makedirs(extract_path, exist_ok=True)
        with zipfile.ZipFile(zip_path, 'r') as zip_ref:
            zip_ref.extractall(extract_path)
        print(f"Файли успішно розпаковано в {extract_path}")
    except zipfile.BadZipFile:
        print(f"Помилка: {zip_path} – це не ZIP архів.")
```