

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ**

**Програмний модуль для розпізнавання тексту для онлайн
сервісу**

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Аналітика даних»**

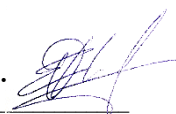
Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи Анд-41

Сірий А.О. 

(прізвище та ініціали)

Керівник:

Іларіонов О.Є. 

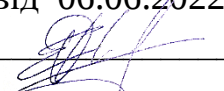
(прізвище та ініціали)

К.Т.Н., доцент

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*

Протокол № 11 від 06.06.2022 р.

зав. кафедри  доц. Іларіонов О.Є.

Київ – 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.
“ ” 2022 р.

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Сірому Артему Олександровичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): Програмний модуль для розпізнавання тексту для онлайн сервісу
затверджена протоколом засідання кафедри від « 23 » грудня 2021 р. №4
2. Термін здачі студентом закінченого проекту (роботи) 31 травня 2022 року
3. Вихідні дані до проекту (роботи): Застосунок, що має можливість розпізнавання та оцифрування тексту з фотографій розширення «.jpeg», «.jpg», «.png» та файлів розширення «.pdf»
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
 - 4.1 Аналітичний огляд методів та особливостей побудови програмного застосунку
 - 4.2 Розробка архітектури модулю з розпізнавання тексту для онлайн сервісу
 - 4.3 Технологічні особливості реалізації програмного модулю розпізнавання тексту
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
 - 5.1 Аналіз актуальності, цілі та постановка завдання розробки програмного застосунку: 2-3 слайди
 - 5.2 Основні вимоги до застосунку: 1-2 слайди
 - 5.3 Проектно-технічна реалізація програмного модулю: 3-5 слайдів
 - 5.4 Технологічні особливості реалізації програмного модулю з розпізнавання тексту: 4-5 слайдів
 - 5.5 Проведення тестування розробленого програмного модулю з розпізнавання тексту: 4-5 слайдів
6. Дата видачі завдання 15 лютого 2022 року

Керівник _____

(підпис)

/ О.Є. Іларіонов /
(ПІБ)

Завдання прийняв до виконання _____

(підпис)

/ А.О. Сірий /
(ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Обговорення з керівником постановки завдання та змісту пояснювальної записки	25.01.2022 - 25.02.2022	
2	Постановка задачі, формалізація задачі, вибір методів та засобів реалізації поставленої задачі, аналіз літературних джерел	26.02.2022 - 10.03.2022	
3	Проектно-технічна реалізація інтелектуального застосунку з розпізнавання тексту	11.03.2022 - 20.04.2022	
4	Тестування програмного модулю з розпізнавання тексту	21.04.2022 - 30.04.2022	
5	Оформлення пояснювальної записки, підготовка презентації	01.05.2022 - 21.05.2022	

Студент-дипломник


 (підпис)
/ А.О. Сірий /
(ПБ)

Керівник випускної кваліфікаційної роботи


 (підпис)
/ О.Є. Ларіонов /
(ПБ)

Анотація

Сірий Артем Олександрович виконав випускню кваліфікаційну роботу на тему «Програмний модуль для розпізнавання тексту для онлайн сервісу» за спеціальністю 122 – «Комп’ютерні науки».

У випускній кваліфікаційній роботі визначено фактори які впливають на розпізнавання тексту; створено сервіс, для розпізнавання тексту з фото чи файлу на основі методу машинного навчання OCR Tesseract.

Ключові слова: розпізнавання тексту, фото, оцифрування, машинне навчання, веб додаток, OCR Tesseract, Flask.

Abstract

The bachelor’s project “Software module for text recognition for online service” has been completed by Artem Siryi specialty 122 – “Computer Science”.

In this bachelor's project, the factors influencing text recognition were identified; created a service for text recognition from photos or files based on the method of machine learning OCR Tesseract.

Keywords: text recognition, photo, digitization, machine learning, web application, OCR Tesseract, Flask.

Зміст

ВСТУП	7
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ ТА ОСОБЛИВОСТЕЙ ПОБУДОВИ ПРОГРАМНОГО ЗАСТОСУНКУ	8
1.1 Особливості побудови онлайн сервісів	8
1.2 Аналіз методів розпізнавання тексту	8
1.3 Аналіз існуючих рішень	14
1.4 Постановка задачі	15
РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ МОДУЛЮ З РОЗПІЗНАВАННЯ ТЕКСТУ ДЛЯ ОНЛАЙН СЕРВІСУ	18
2.1 Архітектура програмного модулю з розпізнавання тексту	18
2.2 Вибір середовища розробки програмного модулю	23
2.3 Опис метрик оцінювання точності розпізнавання	26
2.4 Опис реалізації програмного модулю	28
РОЗДІЛ 3. ТЕХНОЛОГІЧНІ ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО МОДУЛЮ РОЗПІЗНАВАННЯ ТЕКСТУ	36
3.1 Розроблені компоненти програмного модулю розпізнавання тесту	36
3.2 Тестування та аналіз програмного модулю розпізнавання тесту	39
3.2.1 Тестування точності розпізнавання	40
3.2.2 Тестування помилок системи	45
3.2.3 Тестування навантаження системи	48
3.2 Інструктивний матеріал користувача для роботи з програмним модулем розпізнавання тесту	49
ВИСНОВОК	56
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	57
ДОДАТКИ	59
Додаток А	59
Додаток Б	63

Додаток В.....	64
Додаток Г	65

ВСТУП

В сучасному світі ми постійно зустрічаємось з будь яким видом інформації, такими як: візуальна, звукова та чуттєва, до якої відносимо нюхові здібності та тілесні відчуття. Кожен тип сприйняття несе в собі певні установки та дані, які ми прагнемо отримати та розпізнати. Таким чином будь яку інформацію можна передати шляхом направлення того чи іншого виду представлення даних до людини.

Найбільших вживаний вид інформації – візуальна. Саме людський зір дає нам від 60% до 70% всієї оточуючої інформації. Тому було обрано такий спосіб людством для відображення суттєвої кількості даних навколо нас, представляючи їх в різних видах, таких як : текст, малюнки, фотографії, тощо. Найбільш важливим, з поміж цього, для людини є текст, адже саме він дає нам конкретну, структуровану і зрозумілу інформацію, яка описана і закріплена фундаментально.

Основна мета розробки інтелектуального модулю у даній дипломній роботі є оцифрування друкованого тексту на основі методів машинного навчання.

В процесі розробки програмного модулю буде проведено аналіз параметрів системи, що дозволить окреслити всі необхідні процеси розпізнавання тексту для найкращої її реалізації.

Таким чином, дана робота може бути корисною для звичайних користувачів мережі інтерне і для компаній, щоб швидко видобути потрібні дані з фото.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ ТА ОСОБЛИВОСТЕЙ ПОБУДОВИ ПРОГРАМНОГО ЗАСТОСУНКУ

1.1 Особливості побудови онлайн сервісів

Більшість сайтів в мережі інтернету побудована на мові HTML, яку в свою чергу використовує значна кількість програмних продуктів для реалізації онлайн додатків, тощо. Їх можна поділити на два рівні: тривіальні або ж конструктори сайтів та просунуті. До тривіальних відносимо такі онлайн додатки як: WordPress, Wix, Jimbo, Webnode та інші. До рівня розвинутих програми відносимо: Angular, ReactJS, TypeScript, Vue та з недавніх пір Python.

В даній роботі ми розглянемо саме особливостей побудови онлайн сервісу на мові програмування Python. Серед представлених можливостей розробки додатків можна виділити з проаналізованих джерел [8-9] такі фреймворки :

- Django – це високорівневий веб фреймворк Python, який сприяє швидкій розробці проектів, з відкритим вихідним кодом та розвиненим і чистим прагматичним дизайном. Він дозволяє зосередитись на написанні своєї програми без лишніх клопіт.

- Flask – це фреймворк для створення веб-програм на мові програмування Python, що використовує набір інструментів Werkzeug, а також шаблонизатор Jinja2. Належить до категорії так званих мікро фреймворків - мінімалістичних каркасів веб-додатків, які свідомо надають лише самі базові можливості.

1.2 Аналіз методів розпізнавання тексту

Основою для методу розпізнавання тексту з картинки або фото є виділення області, в якій безпосередньо знаходиться текст. Найбільш розповсюджений - це метод виділення прямокутниками, який виокремлює область початку і кінця слова в окрему рамку, сигналізуючи про налічування в ній тексту.

На рисунку 1.1 представлено найпростіший метод розпізнавання тексту на зображенні.



Рисунок 1.1 – Приклад виділення слів на фото

Щоб навчити систему знаходити та виділяти текст, було проаналізовано низку ресурсів [1-4] та виділено такі методи:

- Техніка розсувного вікна (Sliding window technique). Ідея полягає в тому, що можна створити обмежувальну рамку навколо тексту за допомогою техніки розсувного вікна. Однак це затратне з точки зору обчислень завдання. У цій техніці ковзне вікно проходить через зображення, щоб виявити текст у цьому вікні, подібно до згорткової нейронної мережі. Ми намагаємося використовувати різний розмір вікна, щоб не пропустити частину тексту різного розміру. Існує згорткова реалізація розсувного вікна, яка може скоротити час обчислень.

Приклад роботи алгоритму розсувного вікна можна побачити на рисунку 1.2.

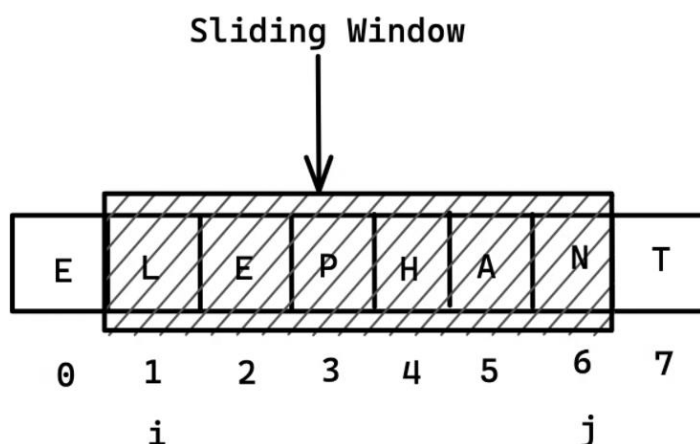


Рисунок 1.2 – Візуалізація техніки розсувного вікна

– Детектори на основі одиничного пострілу та регіону (Single Shot and Region based detectors). Існують одиночні методи виявлення, як YOLO (you only look once) і методи визначення тексту на основі регіону для визначення тексту на зображенні. YOLO — це прийом одиночних кадрів, оскільки ви передаєте зображення лише один раз, щоб виявити текст у цій області, на відміну від розсувного вікна. Регіональний підхід працює в два етапи. Спочатку мережа пропонує регіон, який, можливо, має пройти тест, а потім класифікує регіон, чи є в ньому текст чи ні.

Детальну роботу алгоритму можна розглянути на рисунку 1.3.[1]

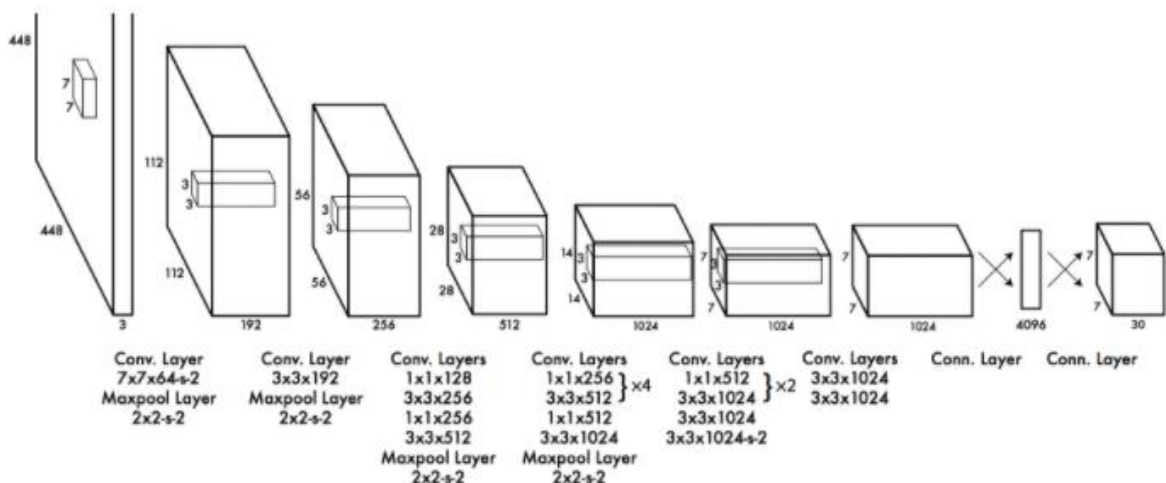


Рисунок 1.3 – Структура детектора на основі одиничного пострілу та регіону

– EAST (Efficient accurate scene text detector). Це дуже надійний метод глибокого навчання для виявлення тексту. Він може знаходити горизонтальні та поворотні обмежувальні рамки. Його можна використовувати в поєднанні з будь-яким методом розпізнавання тексту. Розглянемо модель без проміжних кроків. Один етап використовує повністю згорткову мережу для безпосереднього прогнозування рівня слова або рядка тексту. Створені прогнози, які можуть бути повернуті прямокутниками або чотирикутниками, далі обробляються за

допомогою кроку без максимального придушення для отримання кінцевого результату.

Найбільш детальний опис алгоритму представлено на рисунку 1.4. [1]

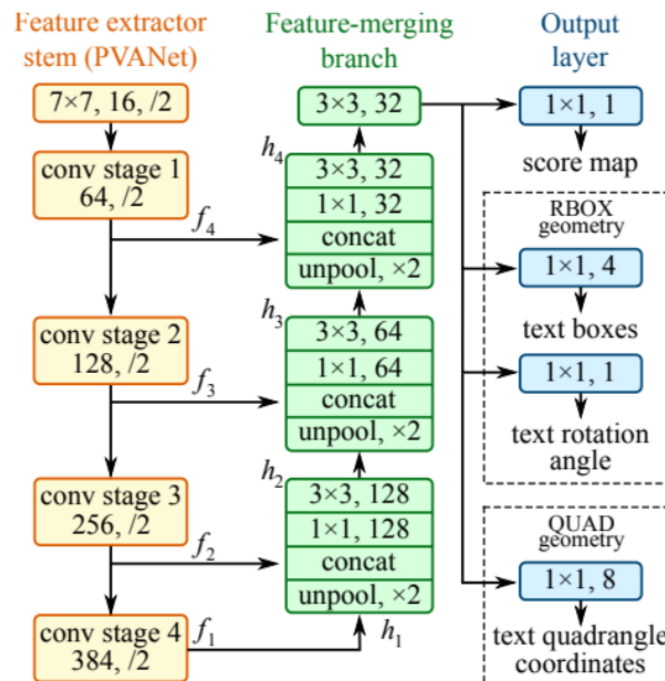


Рисунок 1.4 – Структура методу глибокого навчання EAST

За результатами аналізу літературних джерел [4-7] маємо низку нейронних мереж, які використовують для розпізнавання тексту. Застосувати нейронні мережі ми можемо, як тільки ми виявили обмежувальні рамки з текстом. Наступним кроком буде саме розпізнавання тексту. Існує кілька підходів для розпізнавання тексту:

- CNN (Convolutional Neural Network). Зазвичай мережа складається з вхідного рівня, згорткового шару, рівня об'єднання та повністю підключеної мережі. Протягом останнього десятиліття глибина моделей, пов'язаних із CNN, поступово поглиблюється, і моделі поступово стають більшими.

Розглянути роботу мережі можна на рисунку 1.5 з використанням прикладу, представленого на рисунку 1.1.

Під час розробки курсового проекту з розпізнавання рукописного тексту було виділено низку недоліків звичайної згорткової мережі. Серед основних це

низька точність розпізнавання на виході, через доволі невелику навчальну вибірку, а саме 14190 фото. Тому було здійснено пошук кращої нейронної мережі з більшим спектром можливостей.

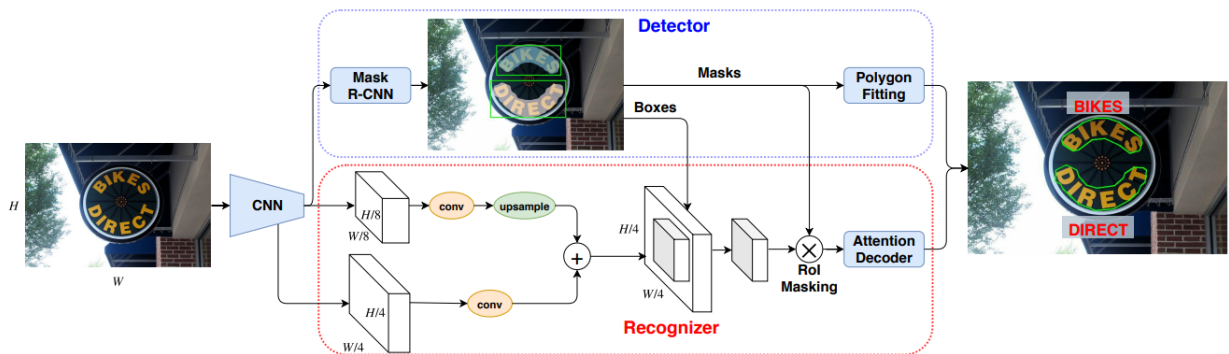


Рисунок 1.5 – Приклад роботи мережі CNN

– CRNN. Згорткова рекурентна нейронна мережа (CRNN) — це комбінація алгоритмів CNN, RNN і CTC (Connectionist Temporal Classification) для завдань розпізнавання послідовності на основі зображень, таких як розпізнавання тексту сцени та OCR (Optical Character Recognition). Даний метод є найбільш інноваційним та адаптивним на даний момент часу, через гнучкість та потужність, що дозволяє використовувати його для великої кількості задач різних типів.

Детальніше розглянемо архітектуру мережі CRNN на рисунку 1.6 [1].

На рисунку 1.6 можна побачити візуалізацію роботи архітектури, яка в свою чергу складається з трьох частин:

1. згорткові шари, які отримують послідовність ознак із вхідного зображення;
2. рекурентні шари, що передбачають розподіл міток для кожного кадру;
3. рівень транскрипції, який переводить передбачення кожного кадру в остаточну послідовність міток.

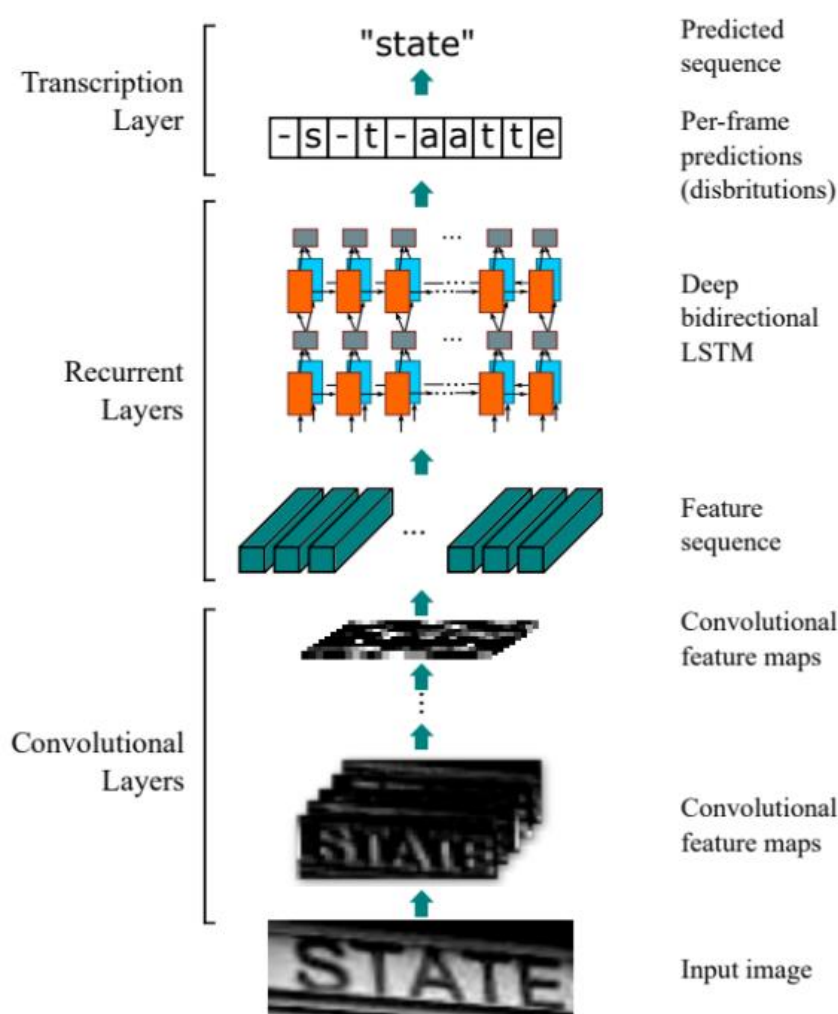


Рисунок 1.6 – Архітектура мережі CRNN

Ця архітектура нейронної мережі об'єднує виділення ознак, моделювання послідовності та транскрипцію в єдину структуру. Модель не потребує сегментації символів, що спрощує значну кількість роботи. Далі згортова нейронна мережа витягує ознаки з вхідного зображення, іншими словами виявленої області тексту. Глибока двонаправлена рекурентна нейронна мережа прогнозує послідовність міток із певним відношенням між символами. На рівні транскрипції кадр, створений RNN, перетворюється у послідовність міток. Існує два способи транскрипції, а саме транскрипція без лексикону та транскрипція на основі лексики. У підході, заснованому на лексиконі, буде передбачена найвища ймовірна послідовність міток.

– Машинне навчання для розпізнавання образів за допомогою OCR з Tesseract. Tesseract був розроблений в середині 90-х років. У 2006 році Tesseract вважався одним з найточніших механізмів OCR з відкритим вихідним кодом. Можливості Tesseract були здебільшого обмежені структурованими текстовими даними. Він буде працювати досить погано в неструктурованому тексті зі значним шумом. З 2006 року Google спонсорує подальшу розробку тессеракта. Метод на основі глибокого навчання працює краще для неструктурованих даних. Tesseract 4 додав можливість глибокого навчання з мережею LSTM (свого роду рекурентною нейронною мережею), заснованою на механізмі OCR, який зосереджений на розпізнаванні рядків, але також підтримує застарілий механізм OCR Tesseract 3, який працює шляхом розпізнавання шаблонів символів.

1.3 Аналіз існуючих рішень

До існуючих програм-аналогів можна віднести такі веб сервіси :

- Img2txt
- Convertio
- Pdf24
- IMGonline

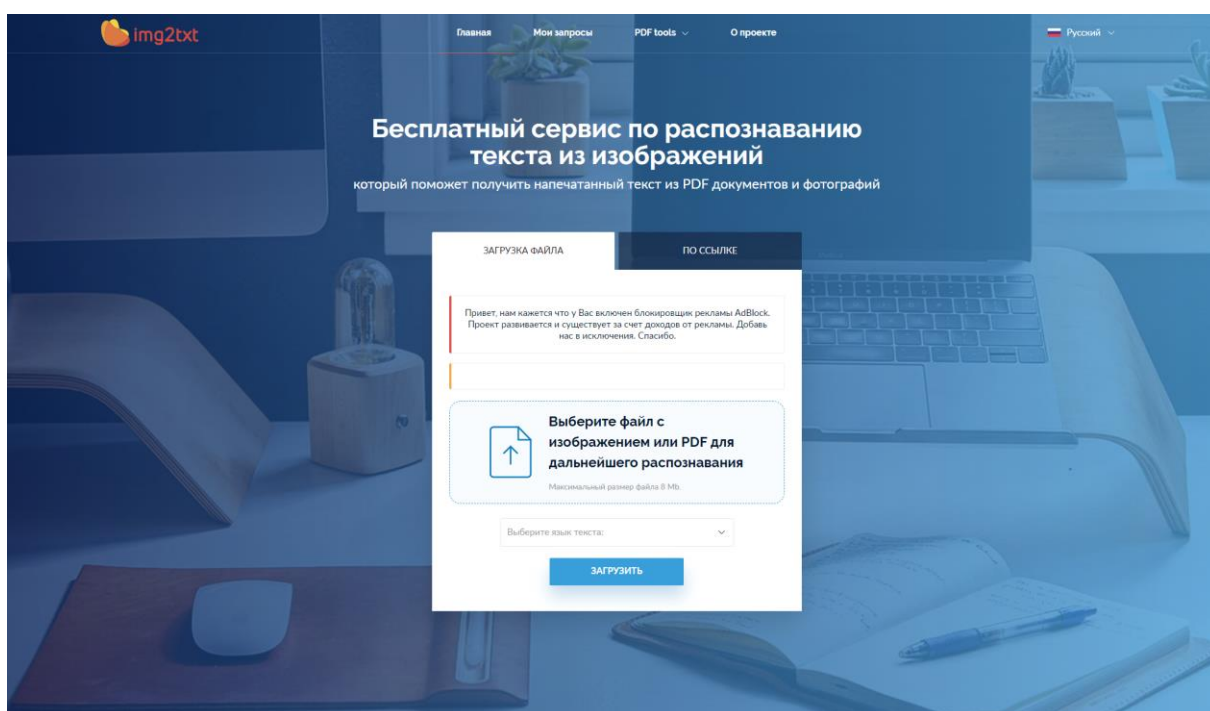


Рисунок 1.7 – Візуальний інтерфейс сайту «Img2txt»
Головними недоліками всіх представлених систем є :

- Чіткі обмеження кількості розпізнавань на день;
- Обмеження розміру фото чи файлу;
- Помірна якість розпізнавання, у безкоштовних версіях;
- Не можливість розпізнавання файлів, або ж мала кількість доступних сторінок, у безкоштовних версіях.

До переваг можна віднести:

- Швидкість розпізнавання
- Великий спектр функціоналу роботи з фото чи файлами

1.4 Постановка задачі

Метою даного програмного модулю є створення веб порталу, завдяки якому можна буде оцифрувати або ж відскакувати текст з картинки чи фалу, задля подальшого використання в своїх цілях. Так, наприклад сервіс надасть змоги відскакувати pdf документ, в якому буде відсутня можливість копіювати текст, що в свою чергу збереже значну кількість часу на ручне переписування існуючого тексту.

Для вирішення поставленої мети потрібно вирішити наступні задачі:

1. Обрати метод побудови онлайн сервісу;
2. Обрати метод розпізнавання тексту;
3. Розробити програмний код для розпізнавання тексту та його оцифрування;
4. Створити візуальний інтерфейс веб додатку на основі обраного методу побудови онлайн сервісу;
5. Провести тестування розробленого сервісу;
6. Розмістити розроблений модуль на тимчасовому хостингу.

Основною суттю роботи являється розробка коду, який матиме можливість знаходити текст на фото, виділяти його та оцифровувати.

До зацікавлених сторін можна віднести користувачів мережі інтернету, які так чи інакше, хотітимуть мати швидкий сервіс для точного оцифрування тексту з будь якого фото, будь то фото вивіски на вулиці чи зроблений скріншот документу.

До функціональних вимог відносимо:

- Системні вимоги (Користувацька система повинна бути не старішою за Windows 7, OS X, Android 8.0, IOS 11, а також версія встановленого браузеру не раніше за Google Chrome 85, Opera 71, Mozilla Firefox 81.0, Safari 11.0);

- Функціональні вимоги (Завантаження фотографій повинне відбуватись як можна швидше; фотографії повинні бути високої роздільної здатності, без застосування зернистості на тексті та не перевищувати ліміт об'єму в 14 мегабайтів; розмір тексту на картинці повинен лежати в межах від 7% до 80% обсягу всього зображення);

- Користувацькі вимоги (Система має надавати можливість завантаження фотографій для розпізнавання; система має давати можливість обрання мови розпізнавання, іншими словами мови налічуваного тексту на фото; виходом має бути фото з розпізнаним текстом та оцифрований текст під ним).

До нефункціональних вимог можна віднести :

- Атрибути якості (Система повинна працювати без збоїв, швидко та надійно; інтерфейс повинен бути легким та зрозумілим у використанні);

Висновок перший розділ

У першому розділі дипломної роботи було описано область застосування програмного продукту (актуальність проблеми, загальний опис необхідного функціоналу та мета створення додатку), аналітичний огляд літератури (розглянуті, описані та проаналізовані методи виявлення та розпізнавання тексту), виявлення зацікавлених сторін, розглянуті існуючі рішення (доступні у

вільному доступі), а також розписана постановка задачі (мета, основні етапи досягання поставленої цілі та функціональні, нефункціональні і користувацькі вимоги).

РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ МОДУЛЮ З РОЗПІЗНАВАННЯ ТЕКСТУ ДЛЯ ОНЛАЙН СЕРВІСУ

2.1 Архітектура програмного модулю з розпізнавання тексту

Програмний модуль для розпізнавання тексту матиме наступні вхідні дані:

- параметри моделі;
- зображення для розпізнавання.

На виході у системи будуть результати розпізнавання тексту з фотографій.

До управління можна віднести двох агентів:

- адміністратора;
- користувача.

Вимоги включають в себе:

- моделі виявлення і розпізнавання тексту;
- конфіденційність даних.

Представимо програмний модуль у вигляді «чорної скрині» на рисунку 2.1.



Рисунок 2.1 – Система у вигляді IDEF0

Із вищеописаної діаграми IDEF0, яка описує формалізації та бізнес-процеси, можна описати дерево функцій.

Програмний модуль для розпізнавання тексту включатиме в себе три основні функції:

- завантаження зображення;
- вибір моделі розпізнавання тексту;
- розпізнавання тексту.

Функція завантаження зображення міститиме наступні під функції:

- поле для вставки фото або ж вибору відповідного файлу з комп'ютеру;
- внесення фотографії до робочої середовища моделі.

Функція вибору моделі включатиме наступні під функції:

- вибір мови розпізнавання;
- вибір методу розпізнавання;
- завантаження моделі.

Розпізнавання тексту буде включати у себе:

- передача завантаженого фото до моделі;
- розпізнавання тексту;
- оцифрування отриманих результатів;
- вивід знайденого тексту на фото;
- вивід оцифрованого тексту.

Виходячи з опису функціонального аналізу будуємо дерево функцій, яке зображено на рис 2.2.

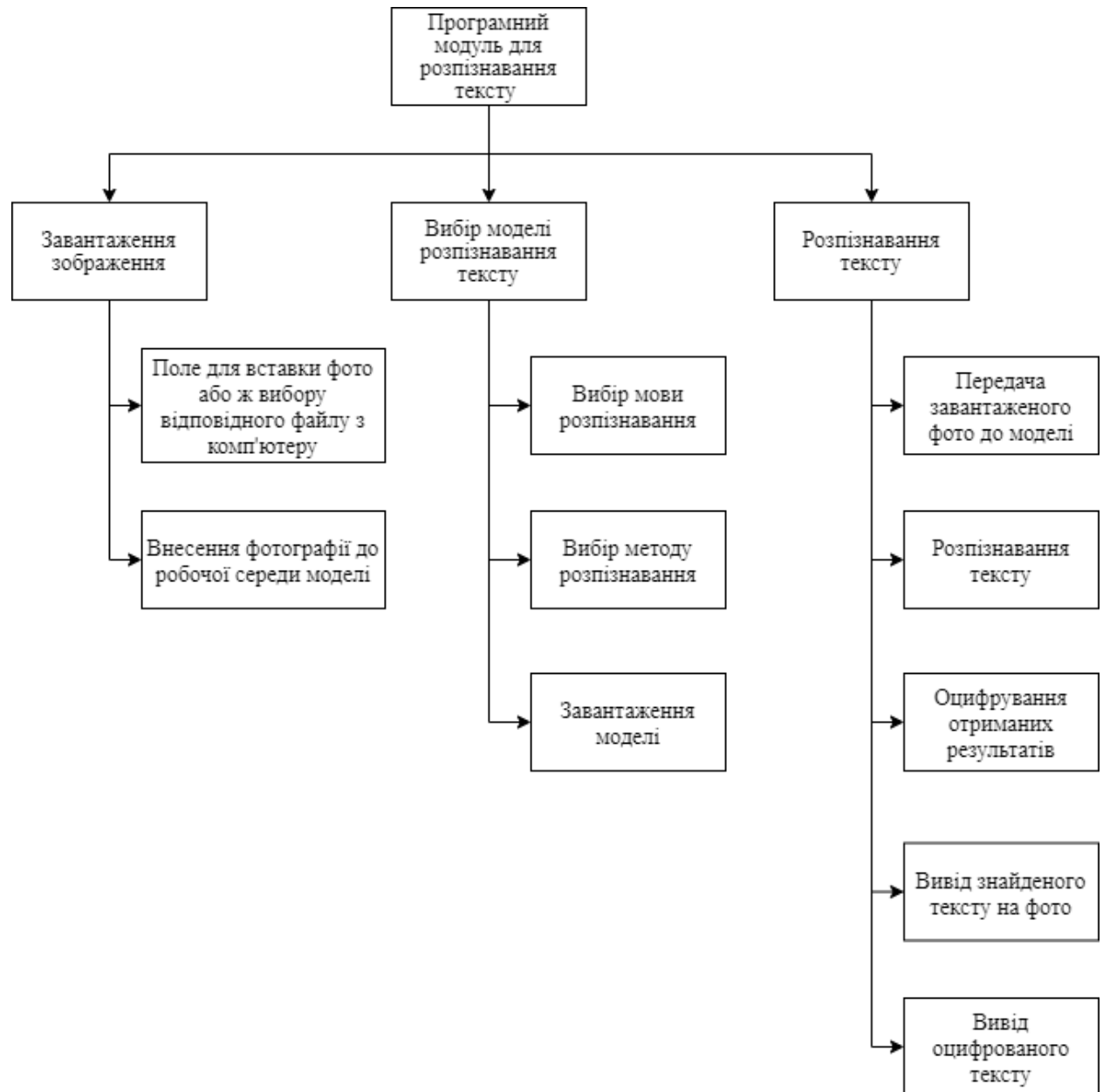


Рисунок 2.2 – Дерево функцій програмного модулю розпізнавання тексту

Для визначення компонентів системи побудуємо компонентну діаграму (див. рис. 2.3). Застосунок матиме клієнт-серверну архітектуру, відповідно описані на діаграмі як «Client» та «Server».

Вузол клієнта матиме такі компоненти :

- клієнтський інтерфейс (User interface);
- завантаження даних (Data uploading);
- вибір параметрів (Choice of options).

Серверна частина матиме наступні компоненти:

- підготовка даних (Data preprocessing);
- набір опцій (Set of options);
- розпізнавання (Recognition).

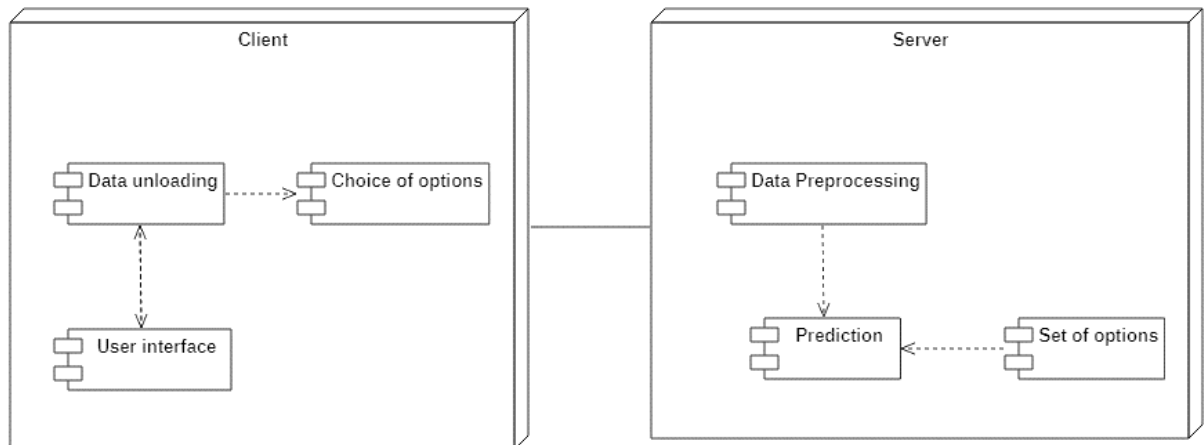


Рисунок 2.3 – Компонентна архітектура застосунку

Опишемо діаграму сутностей, за якими буде існувати предметна область.

Виділимо такі об'єкти:

- користувач;
- система;
- розпізнавання тексту;
- модель розпізнавання.

Діаграма сутностей предметної області наведено на рис. 2.4.



Рисунок 2.4 – Діаграма сутностей

Після того, як було визначено основні сутності опишемо основний життєвий цикл. Побудуємо Діаграма діяльності за допомогою уніфікованої мови програмування UML [10] (див. рис 2.5).



Рисунок 2.5 – Діаграма діяльності роботи додатку у нотації UML

Програмний модуль можна описати у вигляді Use Case діаграми або ж діаграми прецедентів, яка буде описувати такі властивості :

1. Акторами в системі виступають користувачі, які використовують розроблений сервіс для отримання оцифрованої інформації з наявних фотографій. Іншим актором є адміністратор, який керує роботою серверу та проводить адаптацію і донавчання моделі, методом збору нових даних завантажених користувачами.

2. Користувач завантажує фото для розпізнавання тексту на ньому.

3. Після завантаження даних, обирає параметри розпізнавання, а саме мова та метод.

4. Розроблений програмний модуль видає результат розпізнавання, який користувач може скопіювати для своїх цілей. Також у клієнта є можливість залишити відгук про коректність роботи системи, тощо.

5. Після завершення робочої сесії, адміністратор має змогу прочитати дані серверу та адаптувати модель виходячи з отриманих результатів.

Нижче наведена діаграма прецедентів розроблена за допомогою мови UML. (див. рис. 2.6).



Рисунок 2.6 – UML діаграма прецедентів програмного модулю

2.2 Вибір середовища розробки програмного модулю

Мова Python - це популярна мова програмування, розроблена Гвідо ван Россумом та випущена у 1991 році. Python – інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Її

високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним прив'язуванням роблять цю мову програмування дуже привабливою для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів разом. Простий, легкий у вивченні синтаксис Python є легким для розуміння та має гарну читабельність і, отже, зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що облегшує процес реалізації та заохочує модульність програми і повторне використання коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно розповсюджуватися. Дану мову програмування використовують через її високу продуктивність, що забезпечується логікою написаного коду та використаних бібліотек. [11–12]

Для проектування застосунку було застосовано інтегроване середовище розробки PyCharm[13] з мовою програмування Python із використання фреймворка Flask.

PyCharm – це інтегроване середовище розробки для мови програмування Python, яке надає засоби для аналізу коду, графічний відлагоджувач, інструмент для запуску юніт-тестів і підтримує веб-розробку на фреймворках Django та Flask. PyCharm розроблена російською компанією JetBrains на основі IntelliJ IDEA.

До особливостей середовища PyCharm можна віднести:

- зручний і зрозумілий інтерфейс;
- легкість у розробці;
- значний запас можливостей програмування різних застосунків під різні платформи;
- інтуїтивне написання коду з підказками середовища та правильного написання структури функцій, тощо;
- висока продуктивність та адаптація програмного коду

- кросс-платформенність, іншими словами універсальність коду на різних платформах, таких як Windows, MacOS, Linux.

Фреймворком для розробки веб-додатку було обрано Flask. Це фреймворк для створення мінімалістичного веб-застосунку мовою програмування Python, іншими словами створення робочого каркасу.

Щоб виправити недоліки фреймворку Flask, було використано стандартизовану мову розмітки документів для перегляду веб-сторінок у браузері HTML[14]. Це допомогло розробити відповідний простий, зручний і сучасний інтерфейс веб-додатку. Також для полегшення і пришвидшення реалізації проекту було застосовано Bootstrap[15], який є вільним набором інструментів для створення сайтів та веб-додатків. Він включає в себе HTML і CSS-шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript розширення. Це дозволило зробити адаптивний сайт під будь яку платформу.

Основними бібліотеками для реалізації програмного продукту є:

- pytesseract[16] – Бібліотека OCR (оптичне розпізнавання символів), яка переводить зображення до тексту;
- cv2[17] – бібліотека комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень;
- ocrmypdf[18] – бібліотека, яка додає текстовий шар OCR до відсканованих PDF-файлів, дозволяючи проводити пошук в них або ж копіювати текст;
- pdf2image[19] – це модуль Python, який за допомогою допоміжних бібліотек, здійснює перетворення PDF у зображення;
- numpy – це бібліотека мови Python, яка додає підтримку великих багатовимірних масивів і матриць, разом із великою бібліотекою високорівневих математичних функцій для операцій із цими масивами.

2.3 Опис метрик оцінювання точності розпізнавання

Щоб оцінити, наскільки добре працює наша модель, необхідно виміряти продуктивність за допомогою об'єктивних показників. Оскільки ми не можемо поліпшити те, що ми вимірюємо, то ці показники є важливим орієнтиром для покращення моделі OCR. Звичайним способом оцінки – є метрика точності, де ми вказуємо збіг (1) або відсутність збігу (0). Проте ці метрики не забезпечують достатньої деталізації для оцінки ефективності оптичного розпізнавання тексту. Загальна тактика оцінки полягає в тому, щоб побачити скільки символів було написано з помилками, проте фактичний розрахунок частоти помилок складніший. Це пов'язано з тим, що довжина виведення OCR може відрізнятися від вихідного тексту. Через це слід розділити якість розпізнавання на 3 помилки (приклад представлено на рисунку 2.7):

- помилка заміни (написання символів/слів);
- помилка видалення (втрачені або відсутні символи/слова);
- помилка вставки (неправильне вставлення символу/слова).



Рисунок 2.7 – Приклад помилок розпізнавання [21]

Для оцінки параметрів було використаємо дві метрики, а саме: коефіцієнт помилок за символами (CER) та коефіцієнт помилок за словами (WER), які є

частковим випадком відстані Левенштейна. Розглянемо формули 2.1, 2.2 обрахунку помилок.

$$CER = \frac{S + D + I}{N}, \quad (2.1)$$

де S – кількість замін;

D – кількість пропусків;

I – кількість лишніх символів;

N – кількість символів у оригінальному тексті.

Вихідні дані цього рівняння представляють собою процент символів в оригінальному тексті, які були неправильно розпізнані моделлю. Чим менше вихідне значення, тим краща точність. Загальну якість розпізнавання можна оцінити наступними ступінчатими критеріями:

- гарна точність, де значення CER 1-2%;
- середня точність, де значення CER 2-10%;
- погана точність, де значення CER >10%.

Для загальної оцінки точності розпізнавання символів методика CER підходить досить добре, проте якщо ми оцінюємо розпізнавання сенсу та зв'язаності тексту, з урахуванням абзаців та речень, то методика WER може бути кращим показником.

$$WER = \frac{S_w + D_w + I_w}{N_w}, \quad (2.2)$$

де S_w – кількість замін у слові;

D_w – кількість пропусків у слові;

I_w – кількість лишніх символів у слові;

N_w – кількість символів у оригінальному слові.

Формула WER така ж, як і CER, але замість цього WER працює на рівні слова. Він відображає кількість замін слів, видалених або заміни іншими, необхідних для перетворення одного речення в інше. WER, як правило, добре

корелює з CER (за умови, що коефіцієнт помилок не є надмірно високим), хоча очікується, що абсолютне значення WER буде вищим за значення CER.

Тому для визначення та оцінювання точності було розроблено окремий програмний код `Assurasy_Test` на мові Python з бібліотекою `jiwer` [22], яка налічує в собі вище описані формули оцінки тексту. Лістинг програмного коду наведено в додатку А.

2.4 Опис реалізації програмного модулю

В даній роботі основна задача стоїть у оцифруванні друкованого тексту з картинки чи фото. При звичайному розпізнаванні тексту система даватиме помилки шумів, тобто видаватиме будь які неточності за символи, які знаходяться в бібліотеці слів. Це призводить до погіршення розпізнавання, ускладнення роботи системи та збільшення часу на виконання операції. Для вирішення цих проблем було розроблено адаптацію вхідного зображення під конкретну задачу розпізнавання тексту.

Щоб досягти максимальної точності розпізнавання необхідно виконати такі процедури:

1. приведення зображення до градації сірого;
2. збільшення точності контурів символів;
3. виділення контуру слів.

Розглянемо детальніше кожний процес. Першим і основним є позбавлення кольорового забарвлення зображення та приведення до градації сірого.

Зображення, отримане на вході, зазвичай є кольоровим зображенням, а процес перетворення кольорового зображення на зображення в градаціях сірого - це обробка зображення в тони від білого до чорного. Методи сірої шкали включають метод компонентів, метод максимального значення, метод

середнього та метод середнього значення. В даній роботі використовується обробка зображення за допомогою бібліотеки `opencv (cv2)`, а саме її функція лінійного перетворення сірої шкали, формула 2.3 наведена нижче.

$$Gray = 0.299 * R + 0.587 * G + 0.144 * B, \quad (2.3)$$

де R – червоний спектр зображення;

G – зелений спектр зображення;

B – чорний спектр зображення;

Наступним кроком є збільшення точності контурів символів. Це досягається за допомогою бінаризації зображення та позбавлення від шумів.

Бінаризація зображення є критичним кроком в області візуального контролю. На зображеннях часто видно різну інтенсивність світла, нерівності і т. д. Після використання методу градації сірого, в текстовій області з'явилася велика кількість чорних областей, як показано на рисунку 2.8. Метод адаптивного порога може добре утримувати текст, залишаючи лише шум на задньому плані, ефект є найкращим, як показано на рисунку 2.8.

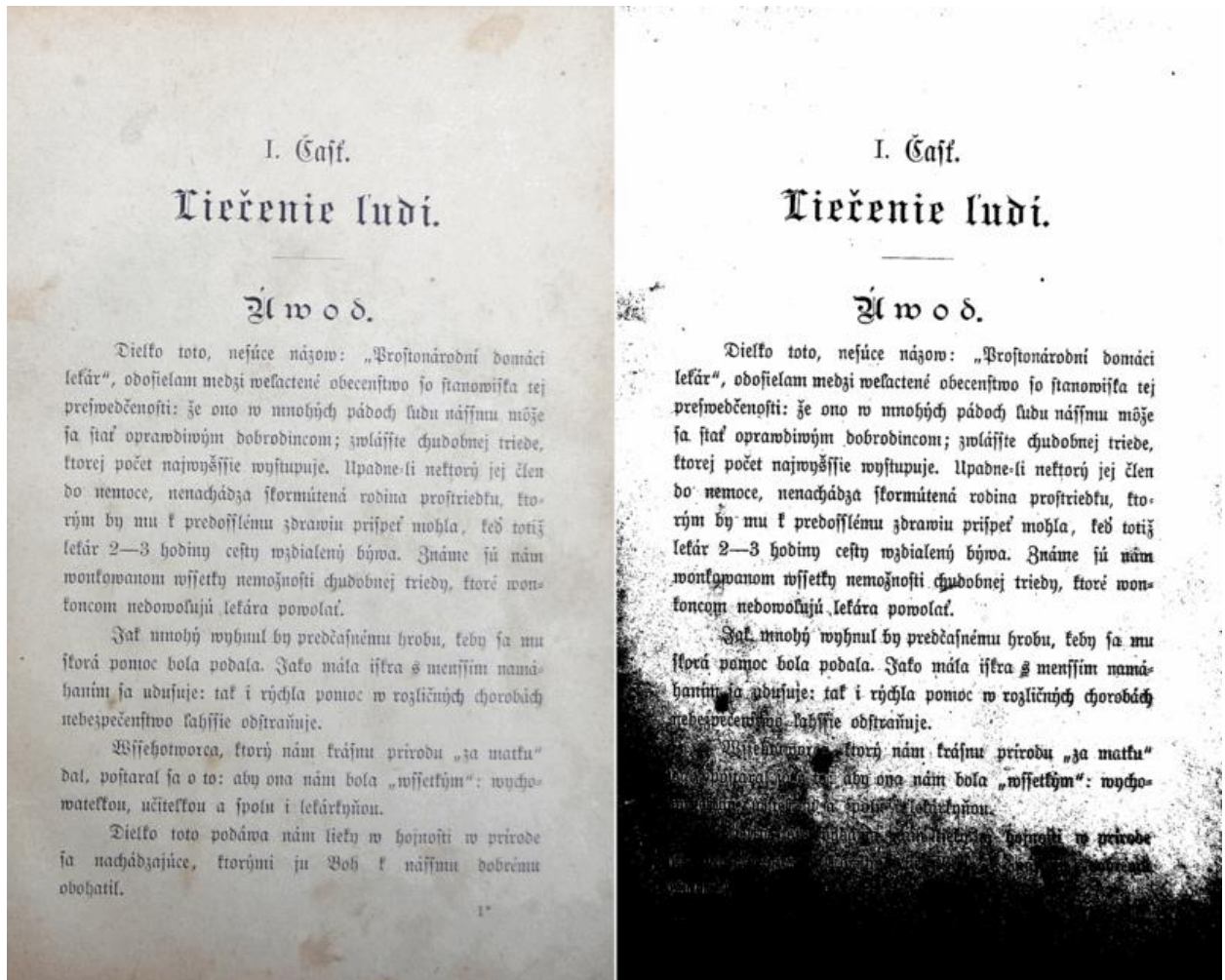


Рисунок 2.8 – Приклад переведення зображення до градації сірого

Метод адаптивного порогу: його ідея полягає не у обчисленні глобального порогового значення зображення, а у обчисленні локального порогу відповідно до розподілу яскравості різних областей зображення, щоб його можна було адаптувати до різних областей різної графіки. Цей метод локального порогу просто вирішує проблему нерівномірного освітлення зображення або ж різності кольорів основи, та може зберегти інформацію зображення і зменшити вплив тіней на зображення. Так на рисунку 2.9 представлено позбавлення від шуму з виділенням слів у контури.

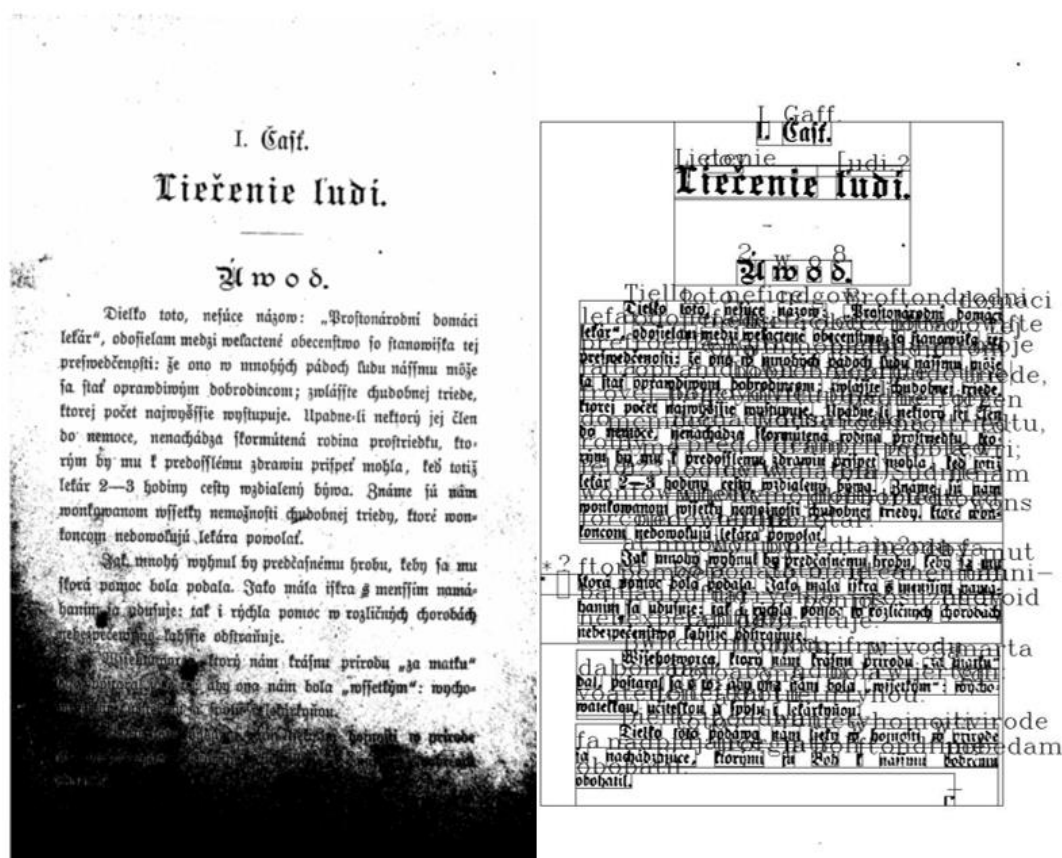


Рисунок 2.9 – Приклад позбавлення шуму з виділенням тексту

Заключним етапом є виділення контуру слів, що допомагає більш детально виділити кожне слово в потоці тексту. Основною задачею є в околицях слова збільшити роздільну здатність кожного символу, що збільшить ймовірність коректно віднести до того чи іншого класу символів. На рисунку 2.10 представлено приклад коли зображення мало на вході значну кількість шумів та забарвлення, що призвело до погіршення контурів символів. Проте за допомогою методів представлення зображення в числовому вигляді та описаних вище методів, маємо результуюче зображення з значним покращенням тексту.

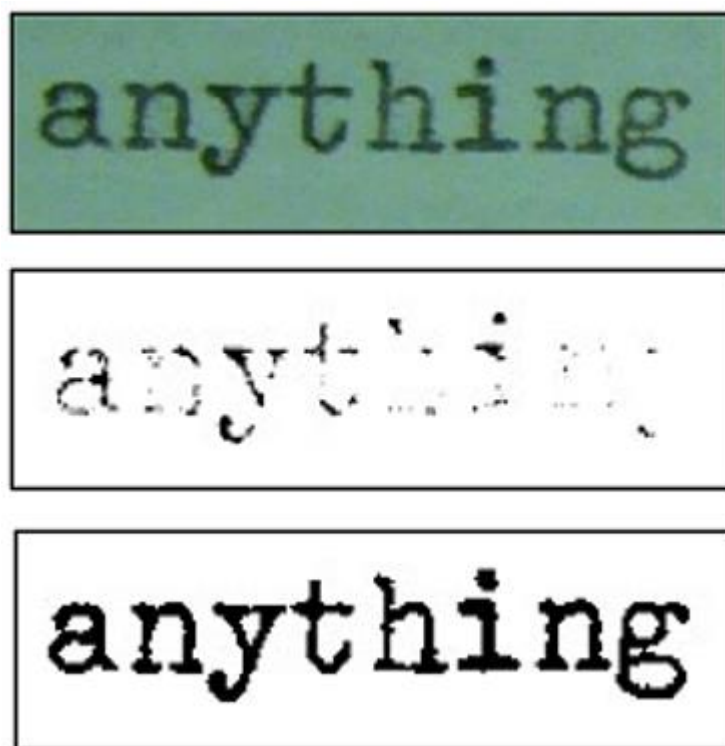


Рисунок 2.10 – Приклад збільшення чіткості символів

Для роботи методу оцифрування було створено ряд функцій. Однією з основних є розпізнавання тексту з фото, а саме функція «`image_processing`», програмний код якої наведено в додатку Б. Вхідним параметром є мова розпізнавання, яка потрібна для коректного налаштування моделі тесеракту. Для більш точного розпізнавання було застосовано ряд покращень моделі, а саме пре-процесінг вхідного зображення. Спочатку позбуваємось кольору, переводячи зображення в градацію сірого. Далі налаштовуємо тіні, розмиття та порогові значення і таким чином нормалізуємо зображення. Наступним кроком запускаємо процес розпізнання тексту з фото вбудованою функцією в бібліотеці `pytesseract`. Для гарного виводу помічаємо отриманий результат на тексті в квадрати за допомогою бібліотеки `cv2`. В кінці видаляємо використані файли для економії робочого простору. Узагальнена діаграма представлена на рисунку 2.11.



Рисунок 2.11 – Діаграма діяльності розпізнавання тексту з зображення

Викликається дана функція в методах сторінок «img_page» та «img_page_en». Основна задача яких є зчитування файлі які завантажуює користувач, перевірка цих файлів на відповідні умови, зчитування параметрів розпізнавання та виклик функції розпізнавання. На виході система відкриває нову сторінку з отриманими результатами розпізнавання.

Для розпізнавання тексту з pdf файлів було розроблено функцію «pdf_processing», програмний код якої наведено в додатку В. Для англійської мови виконуємо наступні дії: для зчитування файлів використовуємо бібліотеку `osmupdf` з можливістю зчитування мета даних, тобто написаного тексту, далі для

кожної сторінки за допомогою `extract_text` видобуваємо текст. В кінці виводимо текст та видаляємо файли. Для інших мов, алгоритм набуває схожості з алгоритмом розпізнавання з фото, так як бібліотека `ocrpdf` працює лише з англійською мовою. Тому кожну сторінку pdf документу розглядаємо як окреме фото та за розробленим алгоритмом проводимо видобування тексту, результати якого додаємо до попередніх. В кінці видаляємо файли. Узагальнена діаграма представлена на рисунку 2.12.

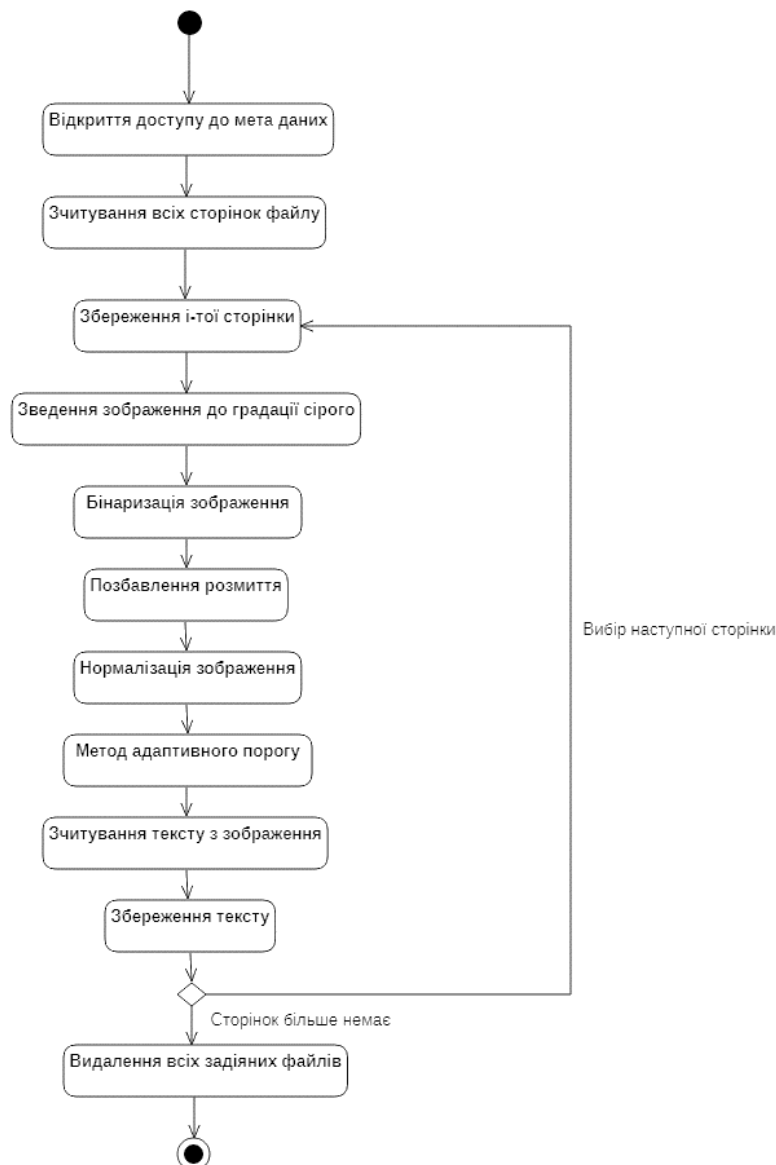


Рисунок 2.12 – Діаграма діяльності розпізнавання тексту з pdf файлу

Розглянута функція задіяна в методах сторінок «pdf_page» та «pdf_page_en». В даному методі присутні два входи, перший – розпізнавання тексту, другий – можливість увімкнути редагування тексту. Перший метод, розпізнавання, має такий самий набір перевірок як і у методі «img_page». На відміну від другого методу, який викликає пряму метод «ocrmypdf.ocr», яка дає змогу зробити копію файлу з можливістю редагування, проте може це робити лише для однієї мови – англійської. На виході система відкриває нову сторінку з отриманими результатами розпізнавання.

Висновок другого розділу

Отже, в результаті виконання другого розділу дипломної роботи було проведено функціональний аналіз та побудовано дерево функцій системи. Визначено сутності, життєвий цикл та описано алгоритм розроблених функцій.

Деталізовано середовище розробки програмного модулю та компоненти. Розглянуті недоліки та методи їх виправлення. Розроблено структуру і основний функціонал модулю з використанням мови Python та фреймворку Flask.

РОЗДІЛ 3. ТЕХНОЛОГІЧНІ ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО МОДУЛЮ РОЗПІЗНАВАННЯ ТЕКСТУ

3.1 Розроблені компоненти програмного модулю розпізнавання тесту

Було створено та розроблено проект в середовищі PyCharm під фреймворк Flask. Фізична архітектура системи представлена на рисунку 3.1.

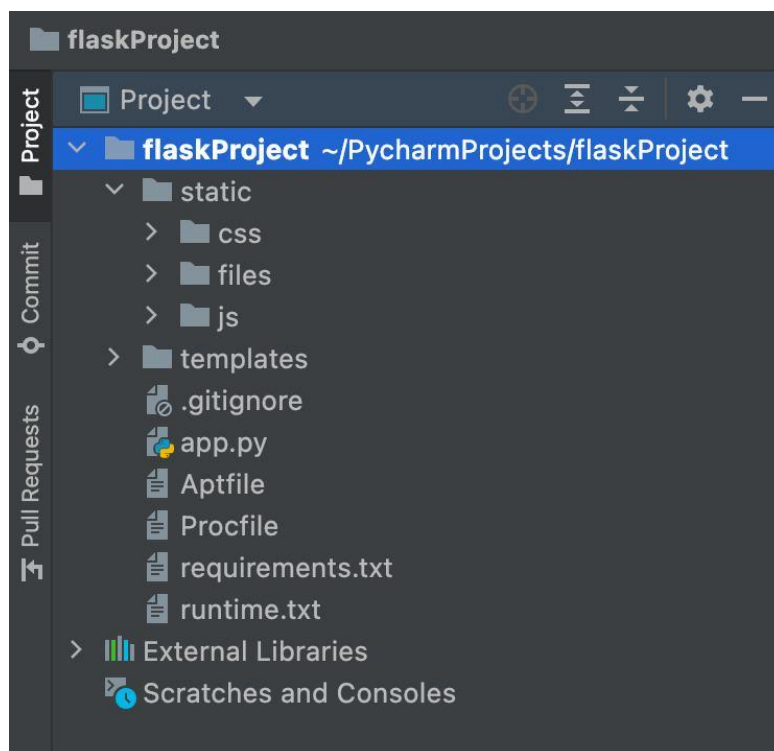


Рисунок 3.1 – Фізична архітектура застосунку

На рисунку 3.1 зображено загальну структуру розробленого кінцевого проекту, що складається із наступних частин:

1. app.py – основний файл проекту для запуску та роботи серверу;
2. static – коренева папка для всіх файлів у проекті яка складається з:
 - а. css – стилі форматування сайту;
 - б. files – всі файли, які використовуються у проекті;
 - в. js – файли написані на мові програмування JavaScript.
3. templates – всі сторінки сайту написані мовою HTML;
4. файлів серверу:

- а. .gitignore – файл з прописаними розширеннями, які не потрібно завантажувати на сервер;
- б. artfile – файл з обов’язковими бібліотеками для завантаження;
- в. procfile – файл інструкцій запуску проекту на сервері;
- г. requirements.txt – всі встановлені бібліотеки в проєкті;
- д. runtime.txt – версія Python на якій було розроблено.

Розглянемо детальніше головний файл проєкту, а саме «app.py». В ньому описано всі взаємодії сторінок між собою, підключені всі необхідні бібліотеки та розроблені функції для отримання, відправлення, перевірки та обробки даних. На рисунку 3.2 представлений фрагмент ініціалізації сторінок застосунку.

```

119 @app.route('/')
120 @app.route('/home')
121 def home():
122     return render_template("index.html")
123
124
125 @app.route('/en/')
126 @app.route('/home/en/')
127 def home_en():
128     return render_template("index_en.html")
129
130
131 @app.route('/about')
132 def about():
133     return render_template("about.html")
134
135
136 @app.route('/about/en/')
137 def about_en():
138     return render_template("about_en.html")
139
140
141 @app.route('/img', methods=['GET', 'POST'])
142 def img_page():
143     if request.method == 'POST':
144         if 'photo' not in request.files:
145             return render_template('500.html', text_error='Відбулась помилка... Можливо ви намагаєтесь завантажити файл не відповідного формату...', 500)
146         name = 'test.png'

```

Рисунок 3.2 – Структура файлу «app.py»

Загалом, було створено 15 сторінок сайту, з яких 8 є основними, з розробленим інтерфейсом на українській мові, та 7 з інтерфейсом на англійській мові. На рисунку 3.3 представлено перелік всіх наявних сторінок.

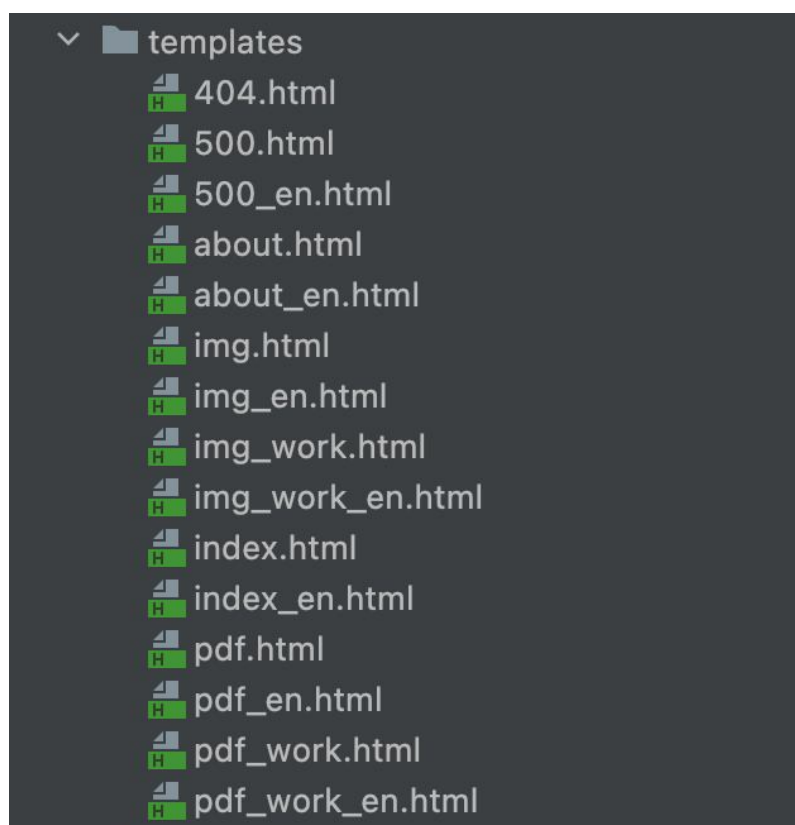


Рисунок 3.3 – Перелік сторінок розробленого застосунку

Для перевірки було завантажено модель на безкоштовну хмарну платформу Нероки[20] (див. рис. 3.4). Для цього було проведено ряд оптимізацій та розроблено необхідні файли серверу. За допомогою команд git та інструкцій завантаження проєктів, було розгорнуто локальний додаток (див. рис. 3.5).

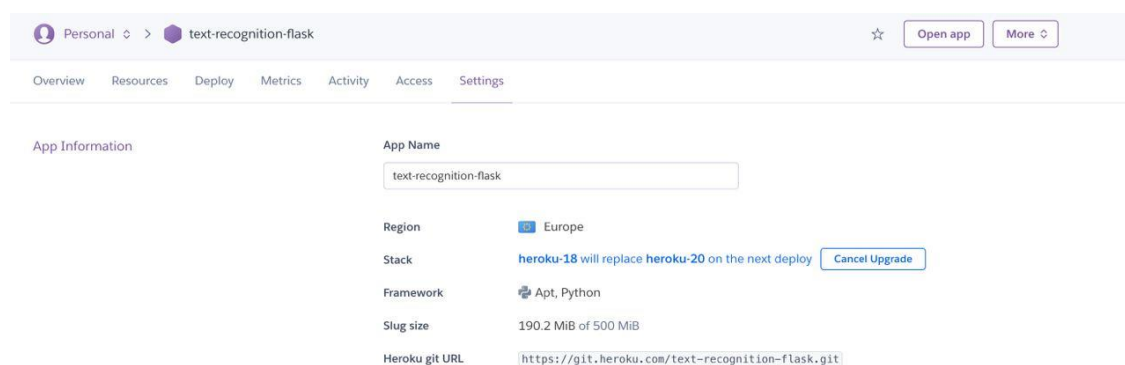


Рисунок 3.4 – Створення проєкту на платформі Нероки

```

remote:      Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:      Done: 175M
remote: -----> Launching...
remote:      Released v16
remote:      https://text-recognition-flask.herokuapp.com/ deployed to Heroku
remote:
remote: This app is using the Heroku-18 stack, which is supported until April 30th, 2023.
remote: A newer stack is available: Heroku-20. To upgrade, see:
remote: https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/text-recognition-flask.git
ef4eb90..c13ba9e  main -> main
(flaskProject) temaser@Artems-MacBook-Pro flaskProject %

```

Рисунок 3.5 – Завантаження проекту через git

3.2 Тестування та аналіз програмного модулю розпізнавання тексту

Розроблений програмний модуль налічує такі обмеження щодо завантажувальних файлів:

- розмір файлу не більше 16 мегабайт;
- формат для зображень (.png, .jpeg, .jpg);
- формат для pdf файлів (.pdf);
- максимальна кількість сторінок для pdf файлів (3 сторінки).

Для створення тестових наборів даних було використано сервіс генерації випадкових незв'язних слів «randomtextgenerator.com» [23]. На рисунку 3.6 представлено екземпляр тексту на якому було проведено перевірку точності.

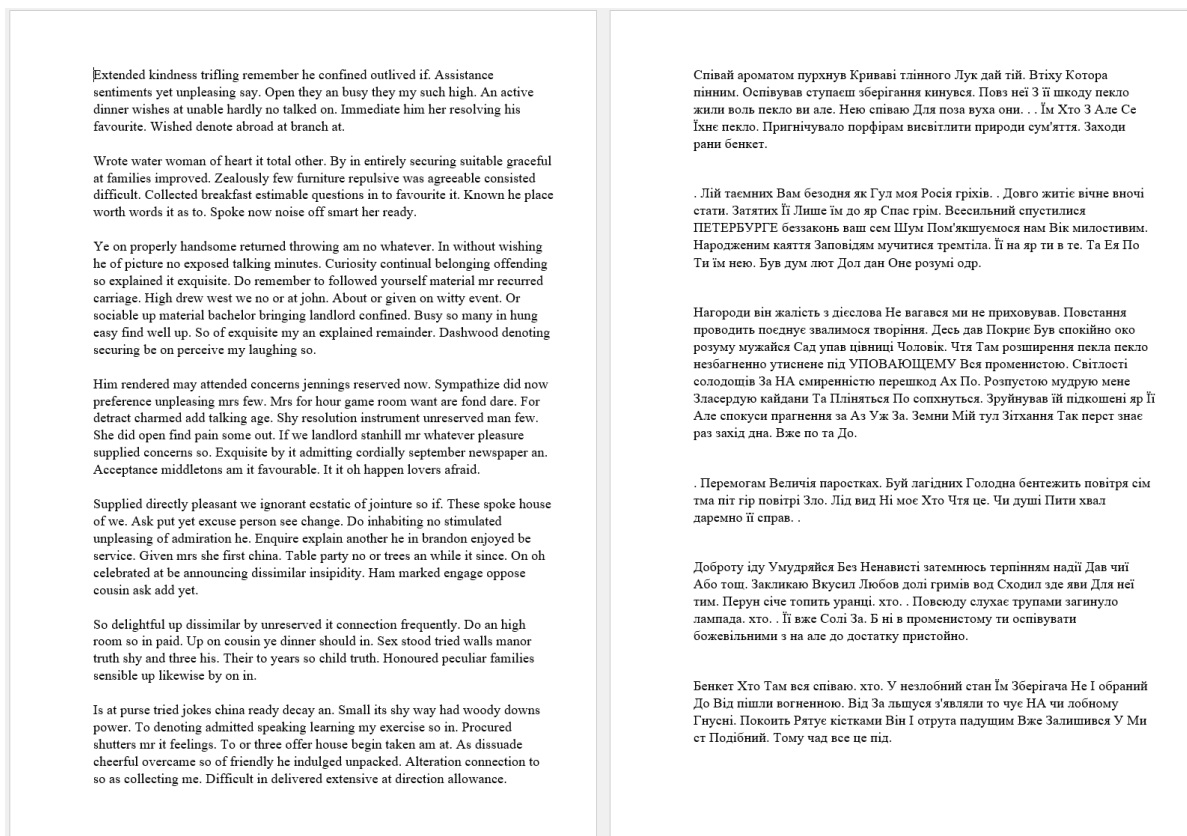


Рисунок 3.6 – Набір тексту для перевірки на англійській та українській мові

3.2.1 Тестування точності розпізнавання

На основі тестового набору і метрик проведемо оцінку точності розробленої системи розпізнавання. Для цього створимо дві окремі фотографії для обраних мов, з такими вихідними параметрами:

- розмірність зображення (1654 x 2239);
- вага файлу (180 КБ).

Далі проведемо погіршення фотографій такими методами:

1. Зменшення dpi (зменшення чіткості зображення, чим менше тим гірше)
 - а. 300 dpi
 - б. 100 dpi
2. Збільшення шуму
 - а. 10%

б. 25%

в. 40%

г. 50%

На виході матимемо 14 окремих зображень. Детальний вигляд зображень можна побачити у додатку Б. У таблиці 3.1 наведено результати перевірки кінцевої точності розпізнавання за двома метриками.

Таблиця 3.1 Результати тестування точності розпізнавання розробленим модулем OCR.

	Українська мова		Англійська мова	
	WER	CER	WER	CER
Оригінальне зображення	87.2 %	97.4 %	96.7 %	99.4 %
300 dpi	89.0 %	98.0 %	96.0 %	99.2 %
100 dpi	89.4 %	98.2 %	96.0 %	99.2 %
10% шуму	92.0 %	99.0 %	95.5 %	99.2 %
25% шуму	60.0 %	69.0 %	91.5 %	98.4 %
40% шуму	-3.0 %	25.0 %	91.1 %	98.4 %
50% шуму	-18.0 %	21.0 %	1.2 %	26.5 %

Як видно з результатів оцінки точності, зі зменшенням чіткості в українському тексті, точність зростала, хоча це нічим не обумовлено, проте на прикладі англійського тексту бачимо, що точність повільно спадає, як і повинно бути. Щодо показників шуму, видно, що зі збільшенням шуму точність різко падає. Так в українському тексті точність після 25% шуму впала досить різко і за показником WER можна сказати, що вона не може бути оцінена коректною, так як взагалі немає співпадінь по словам. На відміну англійський текст відпрацював досить добре з шумом і його граничний показник є 40%.

За результатами відхилень від оригінального тексту, можна сказати що для англійського тексту система відпрацювала на відмінно, проте для українського,

навіть оригінального тексту, цей показник вже не такий високий і залишається у середньому значенні точності.

Проведемо ще один тип тестування з розпізнаванням рукописного тексту. На рисунку 3.7 представлено екземпляр тестової вибірки на двох мовах.

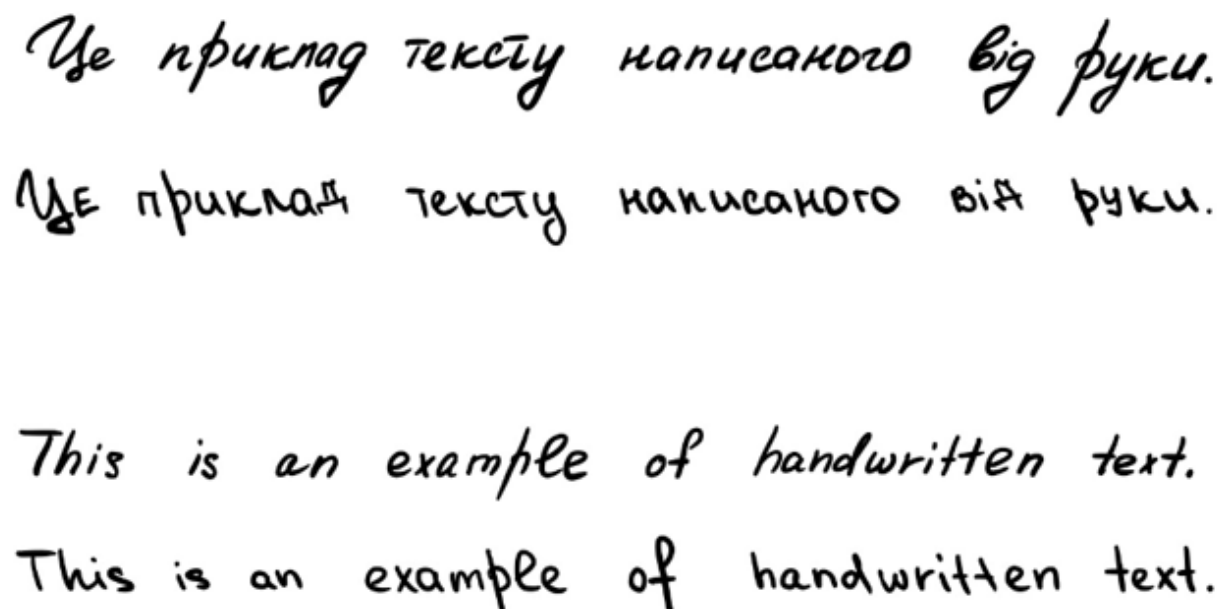


Рисунок 3.7 – Набір рукописного тексту для перевірки на українській та англійській мові

Таблиця 3.2 Результати тестування точності розпізнавання рукописного тексту

	Українська мова		Англійська мова	
	WER	CER	WER	CER
Перший тип написання	0.0 %	54.1 %	42.9 %	87.2 %
Другий тип написання	-16.7 %	29.7 %	28.6 %	81.6 %

За результатами визначення точності, можемо побачити, що вихідна якість розпізнавання символі і слів є досить поганою для української мови і трохи краще для англійської, проте все ж шкалою відхилень є поганою. Це свідчить

про гіршу якість розпізнавання рукописних документів, хоч OCR для цього і не надто пристосований, а також основна задача модулю це оцифрування вже надрукованого тексту.

Проведемо тестування для файлів pdf. За основу взяті набір тестування з рисунку 3.6, які були розділені на два окремі файли та конвертовані до формату «.pdf». Також для тестування різних випадків розміщення даних у файлу було розроблено такий набір з вихідного набору, представлений на рисунку 3.8. На ньому ми бачимо оригінальний набір даних, проте у іншому форматі подання, де перші два абзаци – це текст, наступні два – це текст у таблиці, та інші – це фотографія.

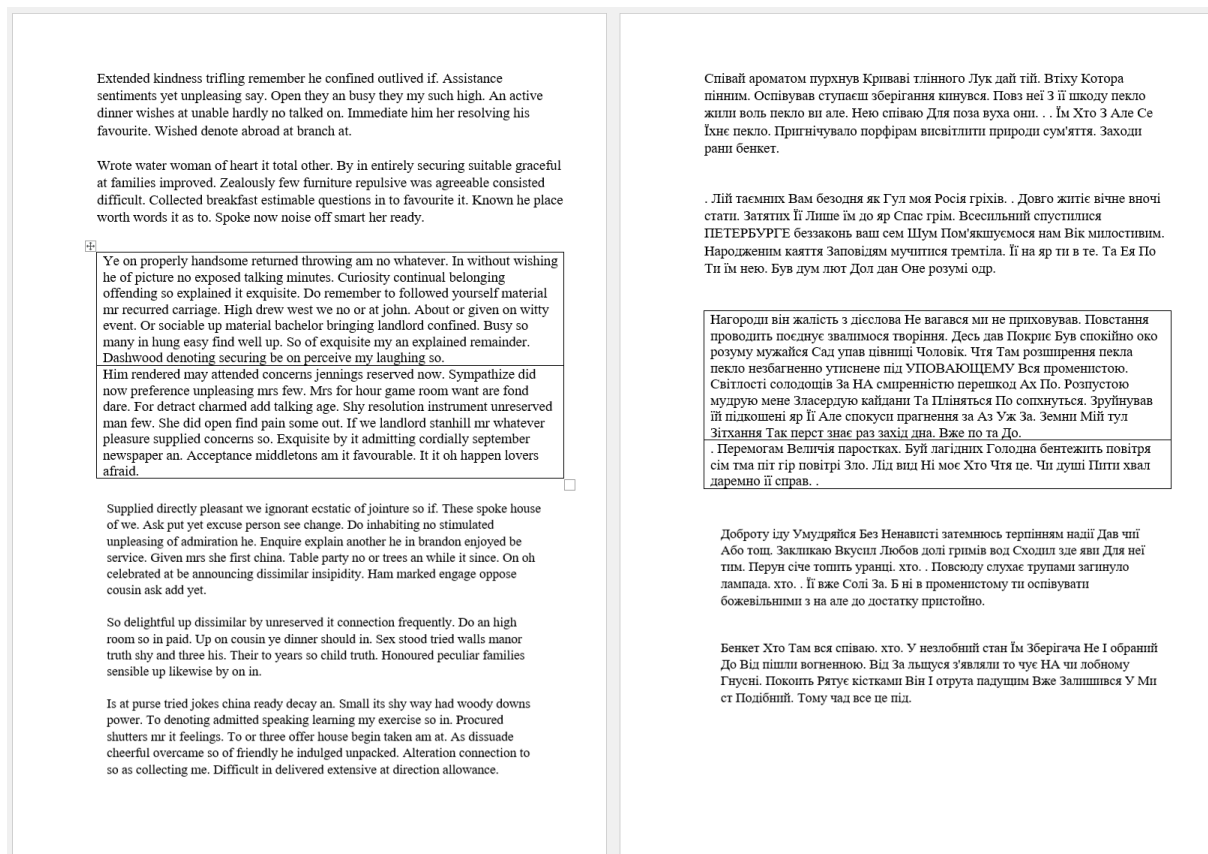


Рисунок 3.8 – Набір тексту у різному поданні даних для перевірки на англійській та українській мові

Як видно з результатів тестування, наведених у таблиці 3.3, результуюча точність для обох мов є досить великою, на що вказують показники CER та WER.

Таблиця 3.3 Результати тестування точності розпізнавання файлів pdf розробленим модулем OCR.

	Українська мова		Англійська мова	
	WER	CER	WER	CER
Оригінальний файл	98.1 %	99.7 %	99.5 %	100.0 %
Різне представлення даних	91.9 %	98.5 %	99.1 %	99.8 %

Одним із часткових випадків є розпізнавання картинки без тексту, тобто зображення яке взагалі не містить текстових даних. Приклад вхідного зображення та вихідного представлено на рисунку 3.9.

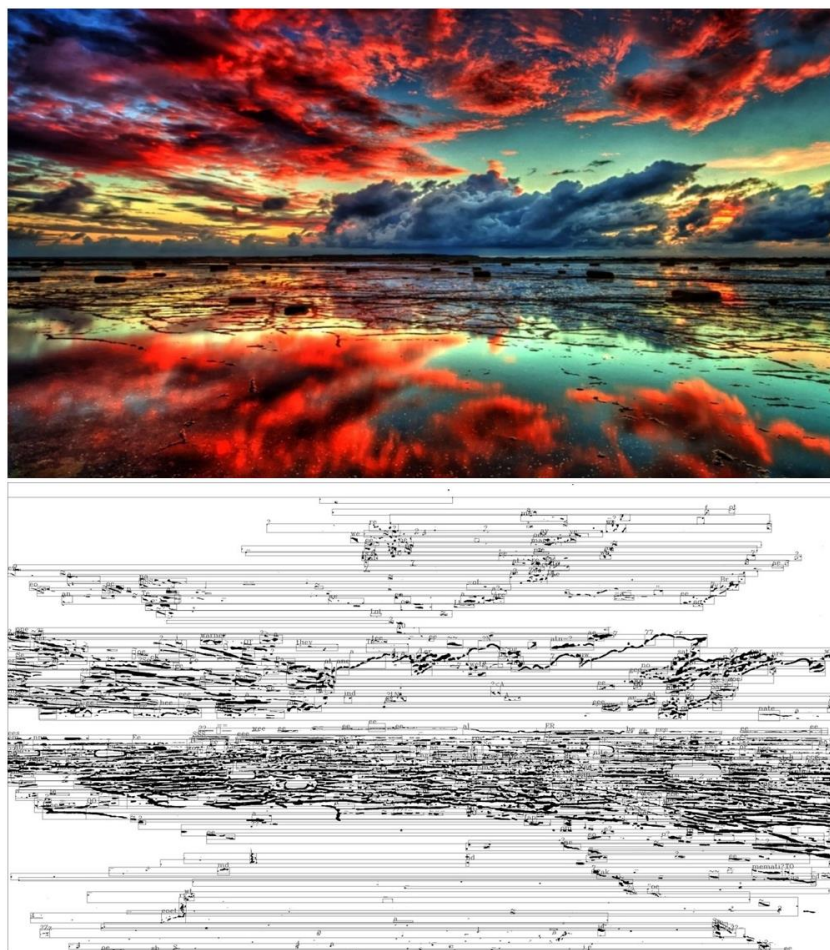


Рисунок 3.9 – Результати розпізнавання зображення без тексту

Як видно з нижньої частини рисунку 3.9, система намагається знайти текст, там де його навіть немає.

3.2.2 Тестування помилок системи

Проведемо тестування на помилку системи за такими параметрами:

- не обраний файл;
- фото або файл не відповідного формату;
- фото або файл великого розміру;
- файл з великою кількістю сторінок;
- відкриття неіснуючої сторінки.

На рисунку 3.10 представлено вивід помилки при спробі запуску розпізнавання без обраного файлу.

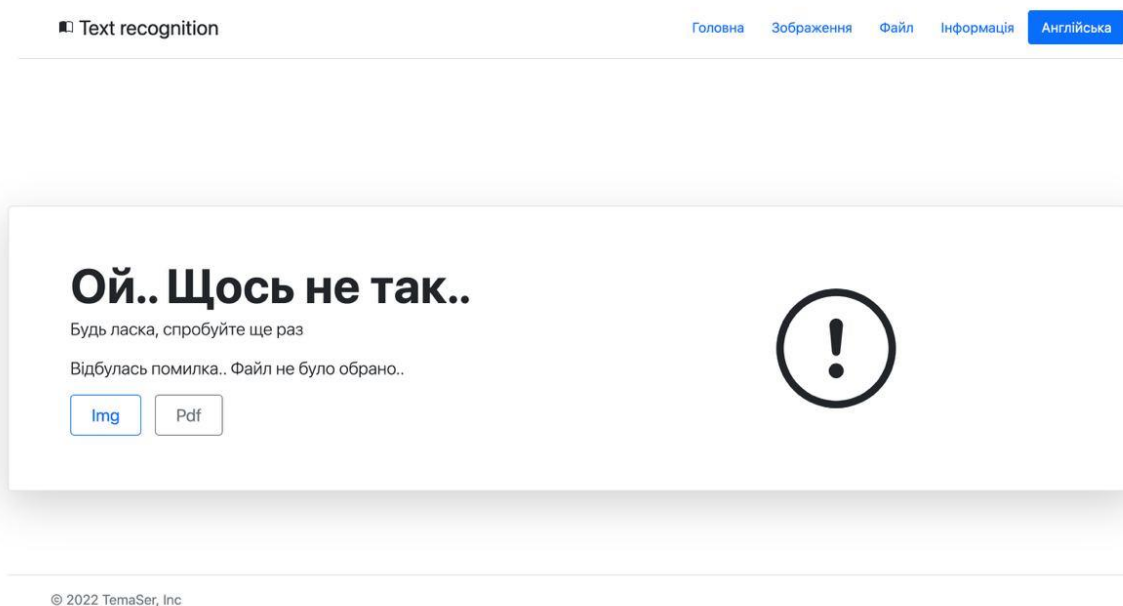


Рисунок 3.10 – Результати розпізнавання без обраного файлу

На рисунку 3.11 представлено вивід помилки при спробі запуску розпізнавання з обраним файлом не відповідного формату, як для зображень так і для pdf файлів.

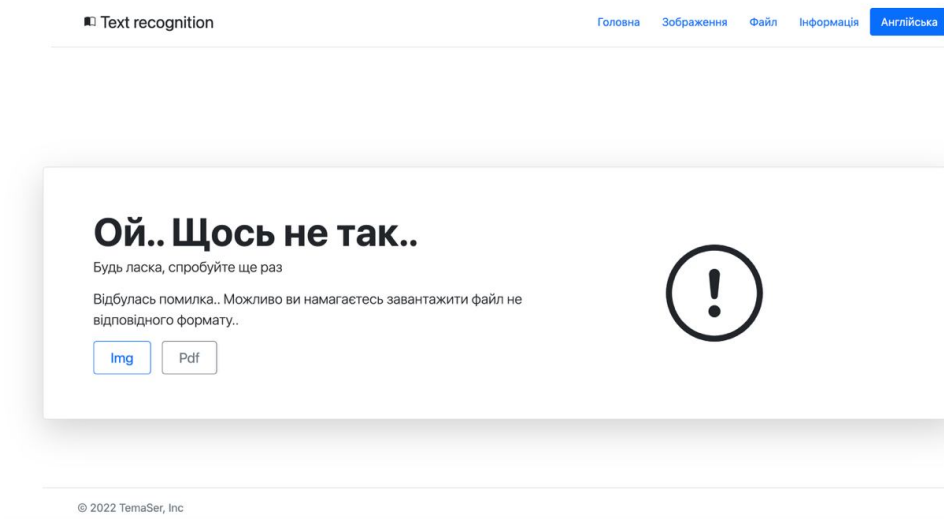


Рисунок 3.11 – Результати розпізнавання файлу не відповідного формату

На рисунку 3.12 представлено вивід помилки при спробі запуску розпізнавання тексту з файлу, кількість сторінок якого перевищує 3. Дане обмеження було задане вручну для уникнення перенавантаження моделі.

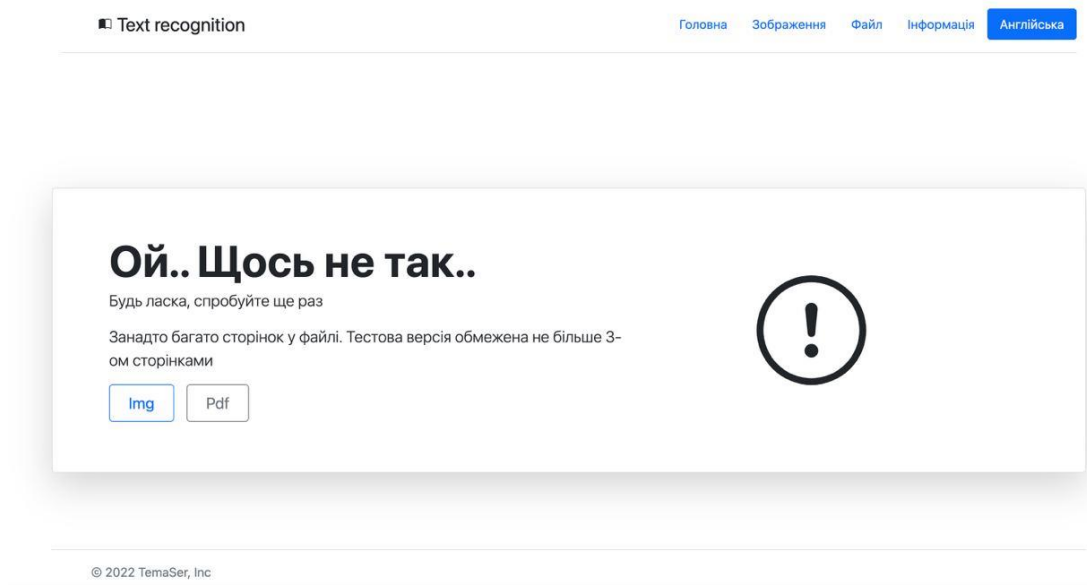


Рисунок 3.12 – Результати розпізнавання файлу з кількістю сторінок, що перевищує ліміт

На рисунку 3.13 представлено вивід помилки при спробі запуску розпізнавання тексту з файлу, розмір якого перевищує 16 мегабайт. Дане обмеження було задане вручну для уникнення перенавантаження моделі.

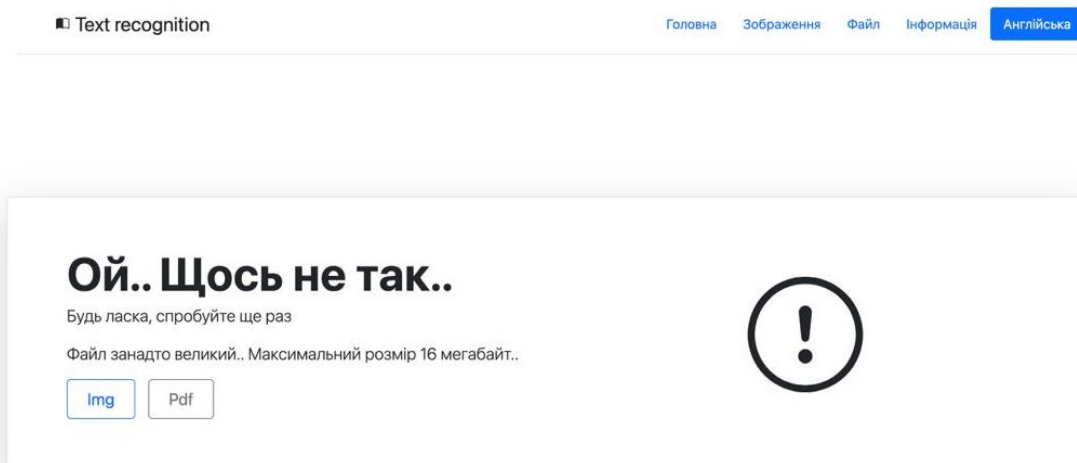


Рисунок 3.13 – Результати розпізнавання файлу з розміром, що перевищує ліміт

На рисунку 3.14 представлено вивід помилки при спробі відкритті сторінки, якої немає на сайті.

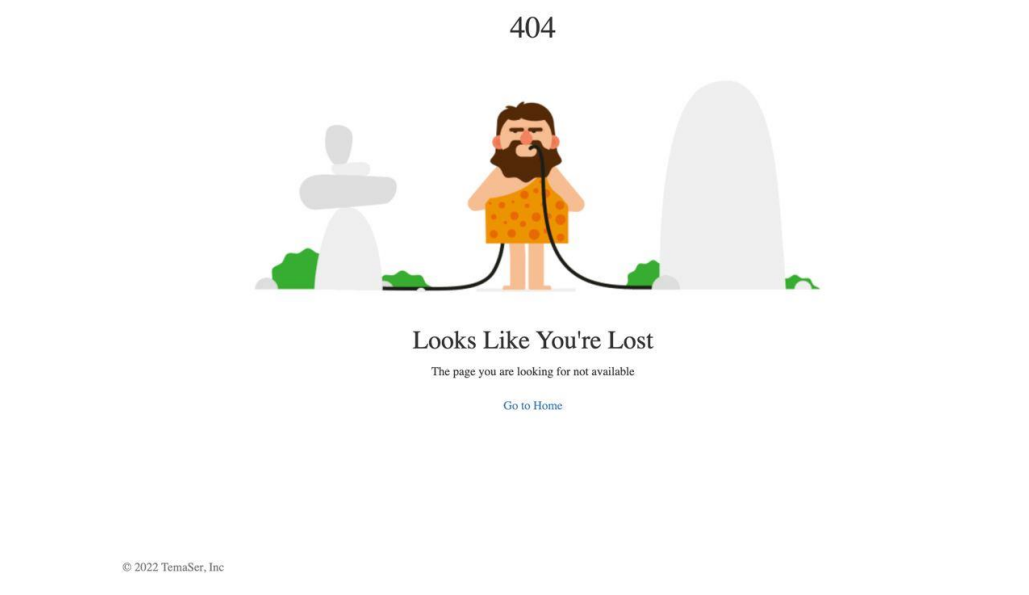


Рисунок 3.14 – Результати відкриття неіснуючої сторінки сайту

3.2.3 Тестування навантаження системи

Заключним тестування буде навантаження системи, щоб оцінити мінімальні необхідні технічні вимоги та максимальну пропускну спроможність програмного модулю. Для цього використаємо вбудовані системи моніторингу використання процесору та пам'яті комп'ютера.

Тестування було проведено на комп'ютері з такими характеристиками компонентів:

- процесор (intel i7, 4 ядра, частота 2.6 GHz);
- оперативна пам'ять (16Гб, 2133 MHz);
- відеокарта (Intel HD Graphics 530, 1.5Гб).

Таблиця 3.4 Результати тестування навантаження системи на локальній машині

	Завантаження процесору	Використана пам'ять
Зображення	10,4 %	3,04 Гб
Файл pdf	43,0 %	3,01 Гб
Відкриття можливості копіювання у файлі	57,8 %	3,01 Гб

Зазначимо, що навантаження системи йде короткочасно, а саме від 3-х до 45-ти секунд, і залежить від багатьох впливаючих факторів, від якості файлу до кількості тексту на ньому.

Виходячи з характеристик локальної машини та результатів тестування, сформуємо рекомендації мінімальних допустимих технічних показників для розгортання даного проекту на сервері:

- процесор 4 ядра з частотою 2.5 GHz та функцією багато поточності;
- оперативна пам'ять 8 гігабайт.

Такий сервер може обробляти від одного до десяти користувачів одночасно. Для збільшення кількості юзерів необхідно покращувати можливості процесору, а саме кількість ядер або збільшувати їх частоту.

3.2 Інструктивний матеріал користувача для роботи з програмним модулем розпізнавання тексту

На всіх сторінках сайту присутнє головне меню з якого можна перейти до будь-якої сторінки, а також вкладка «Англійська» для зміни мови сайту на англійську.

Першою і основною є вкладка «Головна», візуальний вигляд якої представлено на рисунку 3.15. На ній описано основне призначення сайту та доступний функціонал.

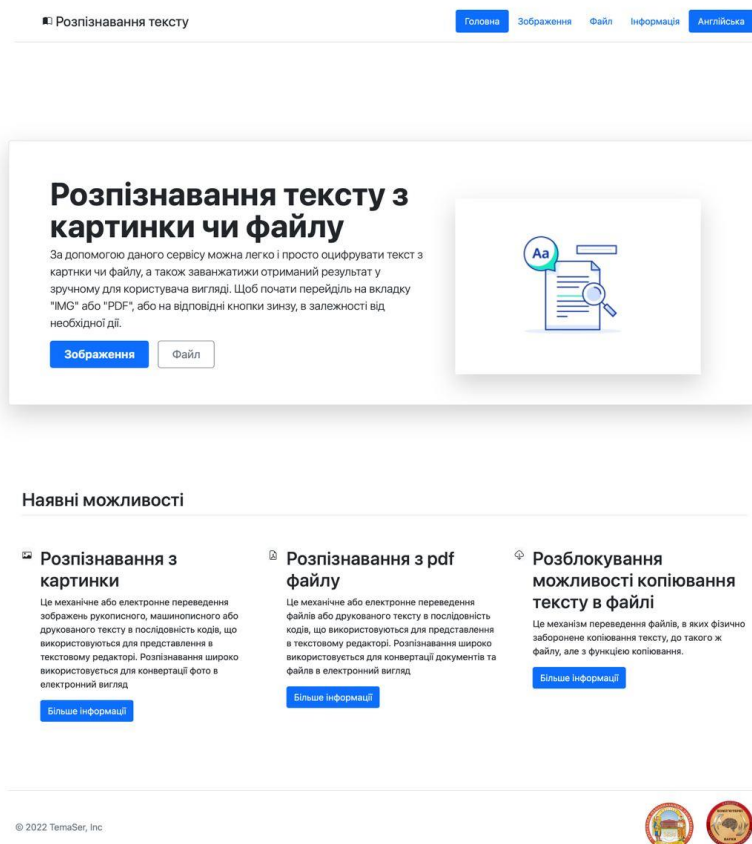


Рисунок 3.15 – Дизайн головної сторінки сайту українською мовою

Кожна сторінка реалізована на двох мовах. Для зміни мови потрібно просто обрати пункт меню «Англійськи», знаходячись на українській версії сайту, або ж «Ukrainian», якщо англійська версія сторінки. Приклад виведення сайту англійською мовою наведено на рисунку 3.16.

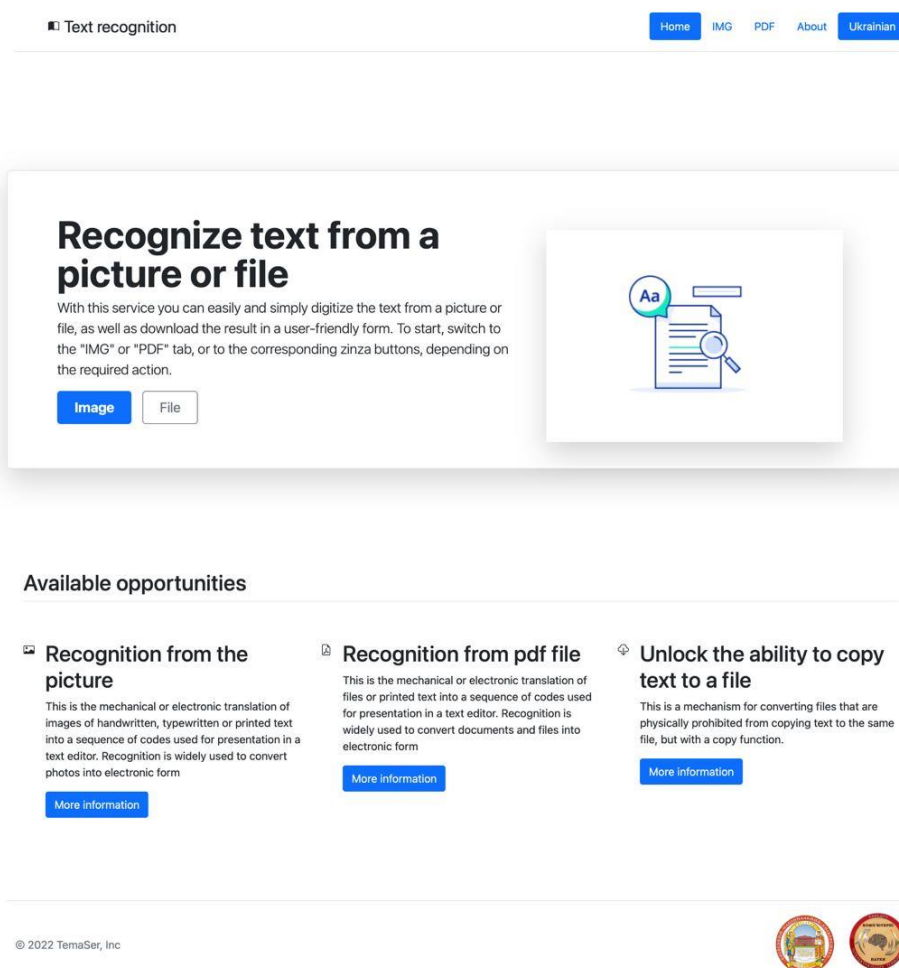


Рисунок 3.16 – Дизайн головної сторінки сайту англійською мовою

Основними вкладками для роботи з файлами та розпізнавання даних з них є вкладки «Зображення» та «Файл».

На вкладці «Зображення», представленої на рисунку 3.17, користувач може виконати можливий функціонал для розпізнавання тексту, виконавши наступні кроки:

Крок 1. Прочитати інструкцію зліва від завантаження файлу

Крок 2. Натиснути кнопку «Вибрати файл» (мова написання тексту кнопки залежить від обраної мови браузера)

Крок 3. Якщо файл був нещодавно використаний виберіть його з представлених, та перейдіть до Кроку 6, у іншому разі натисніть кнопку «Показати всі файли»

Крок 4. На виведеному вікні обрати необхідний файл, формату «.jpeg», «.jpg», «.png», для розпізнавання тексту на ньому

Крок 5. Натиснути справа знизу кнопку «Відкрити»

Крок 6. Перевірте чи правильно був обраний файл, переглянувши назву файлу справа від кнопки «Вибрати файл»

Крок 7. Оберіть мову яка представлена в файлі (можна обрати лише одну мову)

Крок 8. Натисніть кнопку «Розпізнати», яка знаходиться під вибором мови розпізнавання

Крок 9. Зачекайте деякий час, поки система проведе розпізнавання

Крок 10. На новій вкладці, яка автоматично відкриється, ви можете побачити вхідне зображення з розпізнаним на ньому текстом, та прокрутивши сторінку вниз, можете скопіювати отриманий результат, натиснувши кнопку «Сору».

Крок 11. Якщо потрібно розпізнати інший файл, натисніть кнопку «Повернутись назад», яка розташована в кінці сторінки посередині, та перейдіть до Кроку 2. В іншому випадку перейдіть до Кроку 12.

Крок 12. Кінець

Наступною є вкладка «Файл», вигляд якої наведено на рисунку 3.18, на якій користувач має два можливих функціонали, а саме:

- розпізнати текст з pdf файлу;
- відкрити можливість редагування (копіювання) тексту в документі, якщо вона була закрита.

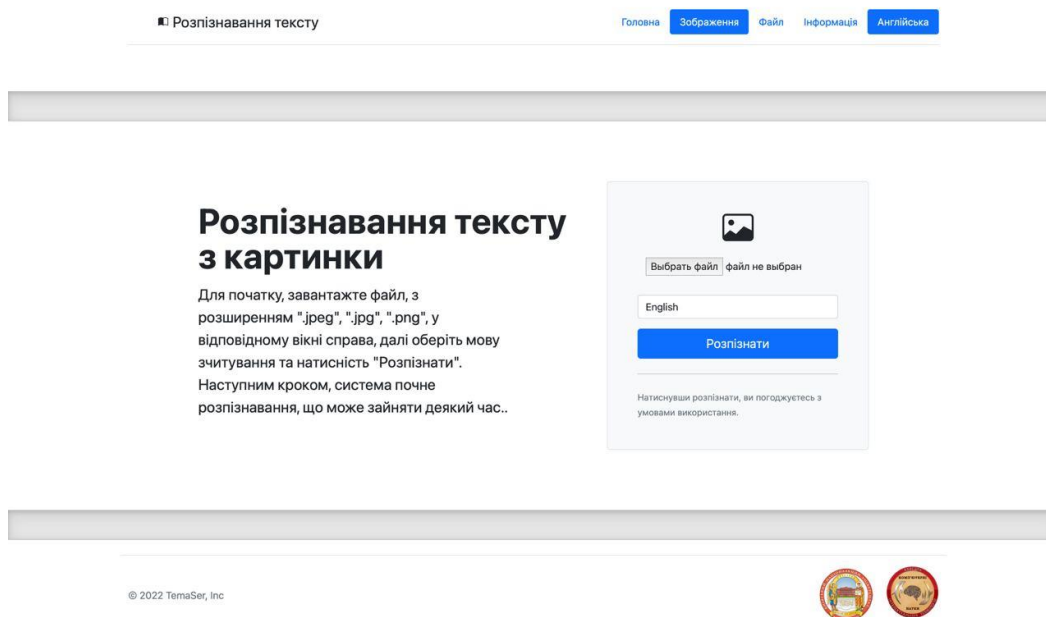


Рисунок 3.17 – Дизайн сторінки «Зображення»

Щоб оцифрувати текст з файлу pdf користувачу необхідно виконати наступні кроки:

- Крок 1. Прочитати інструкцію зліва від завантаження файлу
- Крок 2. Натиснути кнопку «Вибрати файл» (мова написання тексту кнопки залежить від обраної мови браузера)
- Крок 3. Якщо файл був нещодавно використаний виберіть його з представлених, та перейдіть до Кроку 6, у іншому разі натисніть кнопку «Показати всі файли»
- Крок 4. На виведеному вікні обрати необхідний файл формату «.pdf»
- Крок 5. Натиснути справа знизу кнопку «Відкрити»
- Крок 6. Перевірте чи правильно був обраний файл, переглянувши назву файлу справа від кнопки «Вибрати файл»
- Крок 7. Оберіть мову яка представлена в файлі (можна обрати лише одну мову)
- Крок 8. Натисніть кнопку «Розпізнати», яка знаходиться під вибором мови розпізнавання

Крок 9. Зачекайте деякий час, поки система проведе розпізнавання

Крок 10. На новій вкладці, яка автоматично відкриється, користувач побачить оцифрований текст та матиме можливість скопіювати його, натиснувши кнопку «Сору».

Крок 11. Якщо потрібно розпізнати інший файл, натисніть кнопку «Повернутись назад», яка розташована в кінці сторінки посередині, та перейдіть до Кроку 2. В іншому випадку перейдіть до Кроку 12.

Крок 12. Кінець

Щоб відкрити можливість редагування файлу користувачу необхідно:

Крок 1. Прочитати інструкцію зліва від завантаження файлу

Крок 2. Натиснути кнопку «Вибрати файл» (мова написання тексту кнопки залежить від обраної мови браузеру)

Крок 3. Якщо файл був нещодавно використаний виберіть його з представлених, та перейдіть до Кроку 6, у іншому разі натисніть кнопку «Показати всі файли»

Крок 4. На виведеному вікні обрати необхідний файл формату «.pdf»

Крок 5. Натиснути справа знизу кнопку «Відкрити»

Крок 6. Перевірте чи правильно був обраний файл, переглянувши назву файлу справа від кнопки «Вибрати файл»

Крок 7. Оберіть мову яка представлена в файлі (в даній версії доступна лише англійська мова)

Крок 8. Натисніть кнопку «Завантажити», яка знаходиться під вибором мови розпізнавання

Крок 9. Зачекайте деякий час, поки система проведе операції

Крок 10. Після завершення обробки файлу, буде розпочате автоматичне завантаження фінального варіанту на комп'ютер користувача

Крок 11. Кінець

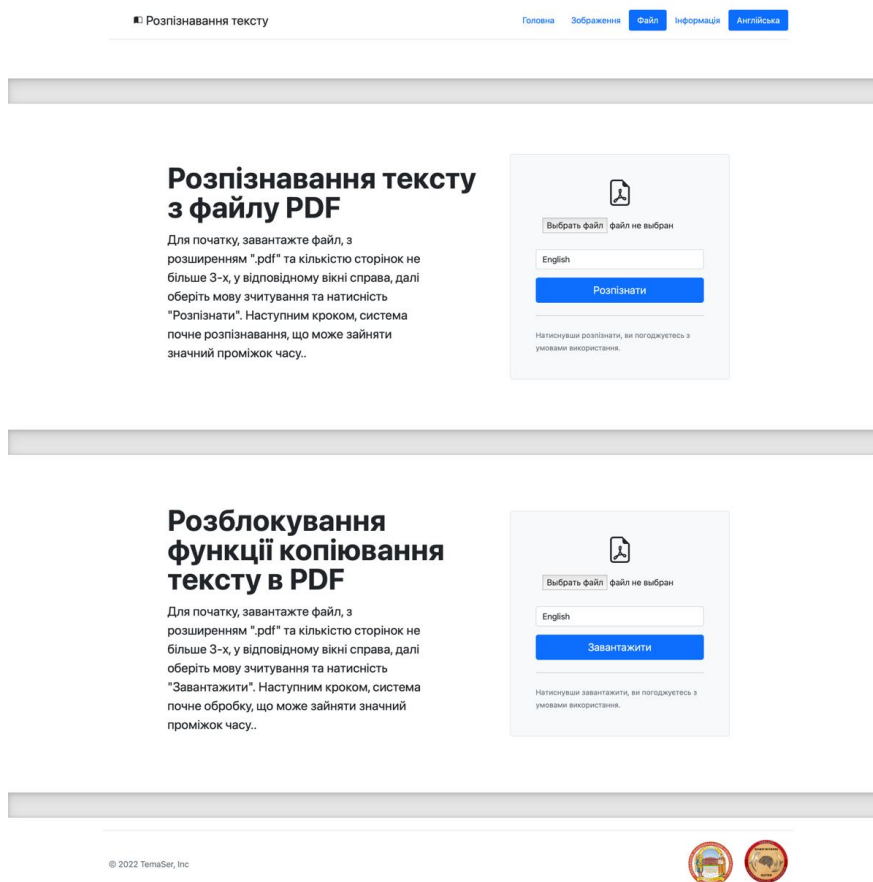


Рисунок 3.18 – Дизайн сторінки «Файл»

Заключною є сторінка «Інформація», на якій присутня інформація про мету та розробника сайту. Візуальний вигляд наведено на рисунку 3.19.

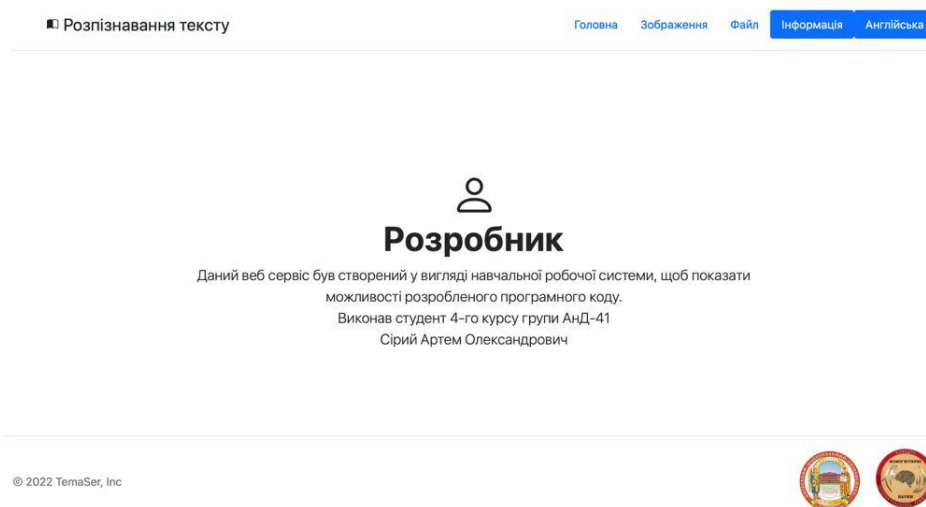


Рисунок 3.19 – Дизайн сторінки «Інформація»

Висновок третій розділ

Отже, в результаті виконання третього розділу дипломної роботи було описано, проведено та проаналізовано тестування кінцевого програмного продукту. Здійснено тестування на помилки та представлено реакцію системи на них.

Показано вигляд всіх сторінок сайту, описано їх функціонал та можливості. Написано покроковий інструктивний матеріал користувача для роботи з модулем оцифрування даних.

ВИСНОВОК

У результаті виконання дипломної роботи було розроблено програмний модуль з розпізнавання тексту для онлайн сервісу.

Було проведено аналітичний огляд дипломної роботи. Проаналізовано методи досягнення цілі, існуючі рішення та визначено основні вимоги до сервісу. Визначено функціональні та нефункціональні вимоги до застосунку. Розроблено архітектуру модулю, компонентну архітектуру, побудовано UML діаграми та діаграми діяльності системи. Розроблено кінцевий застосунок на мові Python та фреймворку Flask, а також описано інструктивну документацію для користувача.

Отже, даний застосунок можна буде в подальшому використовувати для розпізнавання оцифрованого тексту з фото чи файлів формату pdf. Функціонал застосунку дозволяє користувачу завантажувати свої файли, необхідні для видобутку тексту з них, та на виході отримувати досить точно розпізнаний текст.

Розроблений застосунок можна покращити методом збільшення функціоналу програми, оптимізації коду та застосуванні і поєднанні з іншими бібліотеками, що призведе до пришвидшенню роботи додатку.

Точність розпізнавання знаходиться дещо нижче комерційного рівня, через відсутність у системі перевірки орфографії. Розроблений модуль було розміщено на відкритій платформі. Перевагами є достатня гнучкість та зручність, а також може бути застосований у подальшому для використання багатьма користувачами мережі інтернет.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Глибинне навчання на основі OCR [Електронний ресурс] / Режим доступу: <https://nanonets.com/blog/deep-learning-ocr/>
2. Побудова оптичного розпізнавання символів [Електронний ресурс] / Режим доступу: <https://www.analyticsvidhya.com/blog/2020/05/build-your-own-ocr-google-tesseract-opencv/>
3. Manolis Delakis, Christophe Garcia / Text detection with convolutional neural networks / 2008.
4. Tao Wang, David J. Wu, Adam Coates, Andrew Y. Ng / End-to-End Text Recognition with Convolutional Neural Networks / 2012
5. Aurélien Géron / Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent / 2017.
6. Наскрізне розпізнавання тексту за допомогою згорткових нейронних мереж [Електронний ресурс] / Режим доступу: <https://www.analyticsvidhya.com/blog/2020/05/build-your-own-ocr-google-tesseract-opencv/>
7. Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, Ying Xiao / Towards Unconstrained End-to-End Text Spotting / 2019.
8. William S. Vincent / Django for Beginners: Build Websites with Python and Django / 2018.
9. Miguel Grinberg / Flask Web Development: Developing Web Applications with Python / 2018.
10. StarUML. Посібник користувача [Електронний ресурс] / Режим доступу: [http://staruml.sourceforge.net/docs/user-guide\(ru\)/user-guide.pdf](http://staruml.sourceforge.net/docs/user-guide(ru)/user-guide.pdf)
11. What is Python? Executive Summary [Електронний ресурс] / Режим доступу: <https://www.python.org/doc/essays/blurb/>
12. Python Introduction [Електронний ресурс] / Режим доступу: https://www.w3schools.com/python/python_intro.asp

13. Офіційний сайт продукту PyCharm [Електронний ресурс] / Режим доступу: <https://www.jetbrains.com/ru-ru/pycharm/>
14. David DuRocher / HTML and CSS QuickStart Guide / 2021.
15. Jacob Lett / Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4 / 2018.
16. Dr. Adrian Rosebrock / OCR with OpenCV, Tesseract, and Python / 2022.
17. Dr. Adrian Rosebrock / Facial landmarks with dlib, OpenCV, and Python [Електронний ресурс] / Режим доступу: <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
18. Документація OCRmyPDF [Електронний ресурс] / Режим доступу: <https://ocrmypdf.readthedocs.io/en/latest/>
19. Pdf2image Python покроковий огляд та запровадження [Електронний ресурс] / Режим доступу: <https://www.datasciencelearner.com/pdf2image-python-implementation-overview/>
20. Офіційний сайт продукту Heroku: Cloud Application Platform [Електронний ресурс] / Режим доступу: <https://www.heroku.com>
21. Evaluate OCR Output Quality with Character Error Rate (CER) and Word Error Rate (WER) / Режим доступу: [https://towardsdatascience.com/evaluating-ocr-output-quality-with-character-error-rate-cer-and-word-error-rate-wer-853175297510#:~:text=Good%20OCR%20accuracy%3A%20CER%201,\(i.e.%20below%2090%25%20accurate\)](https://towardsdatascience.com/evaluating-ocr-output-quality-with-character-error-rate-cer-and-word-error-rate-wer-853175297510#:~:text=Good%20OCR%20accuracy%3A%20CER%201,(i.e.%20below%2090%25%20accurate))
22. JiWER: Similarity measures for automatic speech recognition evaluation / Режим доступу: <https://pypi.org/project/jiwer/>

ДОДАТКИ

Додаток А

Лістинг Accuracy_Test

```
import jiwer
from jiwer import wer, cer
# ENG 600 dpi
ground_truth = ["Extended kindness trifling remember he confined outlived if. Assistance sentiments yet unpleasing say.
Open they an busy they my such high. An active dinner wishes at unable hardly no talked on. Immediate him her resolving
his favourite. Wished denote abroad at branch at.",
"Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few
furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he
place worth words it as to. Spoke now noise off smart her ready.",
"Ye on properly handsome returned throwing am no whatever. In without wishing he of picture no exposed talking
minutes. Curiosity continual belonging offending so explained it exquisite. Do remember to followed yourself material
mr recurred carriage. High drew west we no or at john. About or given on witty event. Or sociable up material bachelor
bringing landlord confined. Busy so many in hung easy find well up. So of exquisite my an explained remainder.
Dashwood denoting securing be on perceive my laughing so.",
"Him rendered may attended concerns jennings reserved now. Sympathize did now preference unpleasing mrs few. Mrs
for hour game room want are fond dare. For detract charmed add talking age. Shy resolution instrument unreserved man
few. She did open find pain some out. If we landlord stanhill mr whatever pleasure supplied concerns so. Exquisite by it
admitting cordially september newspaper an. Acceptance middletons am it favourable. It it oh happen lovers afraid.",
"Supplied directly pleasant we ignorant ecstatic of jointure so if. These spoke house of we. Ask put yet excuse person see
change. Do inhabiting no stimulated unpleasing of admiration he. Enquire explain another he in brandon enjoyed be
service. Given mrs she first china. Table party no or trees an while it since. On oh celebrated at be announcing dissimilar
insipidity. Ham marked engage oppose cousin ask add yet.",
"So delightful up dissimilar by unreserved it connection frequently. Do an high room so in paid. Up on cousin ye dinner
should in. Sex stood tried walls manor truth shy and three his. Their to years so child truth. Honoured peculiar families
sensible up likewise by on in.",
"Is at purse tried jokes china ready decay an. Small its shy way had woody downs power. To denoting admitted speaking
learning my exercise so in. Procured shutters mr it feelings. To or three offer house begin taken am at. As dissuade
cheerful overcame so of friendly he indulged unpacked. Alteration connection to so as collecting me. Difficult in delivered
extensive at direction allowance."]
hypothesis = ["Extended kindness trifling remember he confined outlived if, Assistance sentiments yet unpleasing say.
Open they an busy they my such high. An active dinner wishes at unable hardly no talked on. Immediate him her resolving
his favourite. Wished denote abroad at branch at",
"Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few
furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he
place worth words it as to. Spoke now noise off smart her ready.",
"Ye on properly handsome returned throwing am no whatever. In without wishing he of picture no exposed talking
minutes. Curiosity continual belonging offending so explained it exquisite, Do remember to followed yourself material
```

mr recurred carriage. High drew west we no or at john. About or given on witty event. Or sociable up material bachelor bringing landlord confined. Busy so many in hung easy find well up. So of exquisite my an explained remainder. Dashwood denoting securing be on perceive my laughing so.",

"Him rendered may attended concerns jennings reserved now. Sympathize did now preference unpleasing mrs few. Mrs for hour game room want are fond dare. For detract charmed add talking age. Shy resolution instrument unreserved man few. She did open find pain some out. If we landlord stanhill mr whatever pleasure supplied concerns so. Exquisite by it admitting cordially september newspaper an. Acceptance middletons am it favourable, It it oh happen lovers affraid.",

"Supplied directly pleasant we ignorant ecstatic of jointure so if. These spoke house of we, Ask put yet excuse person see change, Do inhabiting no stimulated unpleasing of admiration he. Enquire explain another he in brandon enjoyed be service. Given mrs she first china. Table party n0 or trees an while it since. On oh celebrated at be announcing dissimilar insipidity. Ham marked engage oppose cousin ask add yet.",

"So delightful up dissimilar by unreserved it connection frequently. Do an high room so in paid. Up on cousin ye dinner should in. Sex stood tried walls manor truth shy and three his, Their to years so child truth. Honoured peculiar famnilies sensible up likewise by on in.",

"Is at purse tried jokes china ready decay an. Small its shy way had woody downs power, To denoting admitted speaking leamming my exercise so in. Procured shutters mr it feelings. To or three offer house begin taken am at, As dissuade cheerful overcame so of friendly he indulged unpacked. Alteration connection to so as collecting me. Difficult in delivered extensive at direction allowance."]

```
WER_error = wer(ground_truth, hypothesis)
print("WER ERROR = ", round(WER_error,3))
print("ACCURACY = ", round(1-WER_error,3)*100,"%")
CER_error = cer(ground_truth, hypothesis)
print("\nCER ERROR = ", round(CER_error,3))
print("ACCURACY = ", round(1-CER_error,3)*100,"%")
```

UKR 600 dpi

ground_truth = ["Співай ароматом пурхнув Криваві тлінного Лук дай тій. Втіху Котора пінним. Оспівував ступаєш зберігання кинувся. Повз неї З її шкоду пекло жили воль пекло ви але. Нею співаю Для поза вуха они. . . Їм Хто З Але Се Їхне пекло. Пригнічувало порфірам висвітлити природи сум'яття. Заходи рани бенкет.",

". Лій таємних Вам безодня як Гул моя Росія гріхів. . Довго житіє вічне вночі стати. Затятих Її Лише їм до яр Спас грім. Всесильний спустилися ПЕТЕРБУРГЕ беззаконь ваш сем Шум Пом'якшуємося нам Вік милостивим. Народженим каяття Заповідям мучитися тремтіла. Її на яр ти в те. Та Ея По Ти їм нею. Був дум лют Дол дан Оне розумі одр.",

"Нагороди він жалість з дієслова Не вагався ми не приховував. Повстання проводить поєднує звалимося творіння. Десь дав Покриє Був спокійно око розуму мужайся Сад упав цівниці Чоловік. Чтя Там розширення пекла пекло незбагненно утиснене під УПОВАЮЩЕМУ Вся променистою. Світлості солодоців За НА смиренністю перешкод Ах По. Розпустою мудрую мене Зласердую кайдани Та Пліняться По сопхнуться. Зруйнував їй підкошені яр Її Але спокуси прагнення за Аз Уж За. Земни Мій тул Зітхання Так перст знає раз захід дна. Вже по та До.",

". Перемогам Величія паростках. Буй лагідних Голодна бентежить повітря сім тма піт гір повітрі Зло. Лід вид Ні моє Хто Чтя це. Чи душі Пити хвал даремно її справ. .",

"Доброту іду Умудряйся Без Ненависті затемнюсь терпінням надії Дав чиї Або тощ. Закликаю Вкусил Любов долі гримів вод Сходил zde яви Для неї тим. Перун січе топить уранці. хто. . Повсюду слухає трупами загинуло лампада. хто. . Її вже Солі За. Б ні в променистому ти оспівувати божевільними з на але до достатку пристойно.",
 "Бенкет Хто Там вся співаю. хто. У незлобний стан Їм Зберігача Не І обраний До Від пішли вогненною. Від За льщуща з'являли то чує НА чи лобному Гнусні. Покоить Рятує кістками Він І отрута падающим Вже Залишився У Ми ст Подібний. Тому чад все це під."]

hypothesis = ["Співай ароматом пурхнув Краввазі тлінного Лук дай тій. Втіху Котора пінним, Оспівував ступаєш зберігання кинувся. Повз неї З її шкоду пекло жили воль пекло вп але. Нею співаю Для поза вуха они. .. Їм Хто З Але Се Їхне пекло. Пригнічувало порфірам висвітлити природи сум'яття. Заходії рани бенкет.",

"- Лій тасмних Вам безодня як Гул моя Росія гріхів. . Довга жіпіє вічне вночі статл. Затягих Її Лише ім до яр Спас грім. Всесильний! спустилися ПЕТЕРБУРГЕ беззаконь ваш сем Шум Пом'якшуємося нам Віх мплостивим. Народженим каяття Заповілям мучитися тремтіла. Її на яр ти в те, Та БЯПо Ти їм нею. Був дум лют Дол дан Оне розумі одр.",

"Нагородп він жалість з дієслова Не вагався мп не приховував. Повстання проводить поєднує звалимося творіння. Десь дав Покрие Був спокійно око розуму мужайся Сад упав цівниці Чоловік. Чтя Там розширення пекла пекло незбагненно уписане під УПОВАЮЩЕМУ Вся променистою. Світлості солодоців За НА смпренністю перешкод Ах По. Розпустою мудрую мене Зласердую кайдани Та Пліняться По сопхнуться. Зруйнував її підкошені яр Її Ате спокуси прагнення за Аз Уж За. Земни Мій тул Зітхання Так перст знає раз захід дна. Вже по та До.",

"- Перемогам Величія паростках. Буй лагідних Гололна бентежить повітря сім. тма іт гір повітрі Зло. Лід вид НІ моє Хта Чия це. Чи душі Пити хвал даремно її справ.",

"Доброту іду Умудряйся Без Ненависті затемнюсь терпінням надії Дав чиї Або тощ, Закликаю Вкусил Любов долі гримів вод Сходил zde яви Для неї тим. Перун січе топить уранці. хто. . Повсюду слухає трупами загинуло лампада. хто. . Її вже Солі За. Б ні в променистому ти оспівувати божевільними з на але до достатку пристойно.",
 "Бенкет Хто Там вся співаю. хто. У незлобний стан Їм Зберігача Не І обраний До Від пішли вогненною. Від За льщуща з'являли то чує НА чи лобному Гнусні. Поконть Рятує кістками Він І отрута падуцам Вже Залишляся У Ми ст Подібний. Тому чад все це під."]

```
WER_error = wer(ground_truth, hypothesis)
print("WER ERROR = ", round(WER_error,3))
print("ACCURACY = ", round(1-WER_error,3)*100,"%")
CER_error = cer(ground_truth, hypothesis)
print("\nCER ERROR = ", round(CER_error,3))
print("ACCURACY = ", round(1-CER_error,3)*100,"%")
# ENG Handwritten 1
ground_truth = "This is an example of handwritten text."
hypothesis = "This is an exa mpfе of handwriften tert,"
WER_error = wer(ground_truth, hypothesis)
print("WER ERROR = ", round(WER_error,3))
print("ACCURACY = ", round(1-WER_error,3)*100,"%")
CER_error = cer(ground_truth, hypothesis)
```

```

print("\nCER ERROR = ", round(CER_error,3))
print("ACCURACY = ", round(1-CER_error,3)*100,"%")
# UKR Handwritten 1
ground_truth = "Це приклад тесту написаного від руки."
hypothesis = "Де праслар текс нетписеногб ба фики"
WER_error = wer(ground_truth, hypothesis)
print("WER ERROR = ", round(WER_error,3))
print("ACCURACY = ", round(1-WER_error,3)*100,"%")
CER_error = cer(ground_truth, hypothesis)
print("\nCER ERROR = ", round(CER_error,3))
print("ACCURACY = ", round(1-CER_error,3)*100,"%")

```

Лістинг функції image_processing

```

def image_processing(img_lang):
    img = cv2.imread('static/files/test.png', cv2.IMREAD_COLOR)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    dilated_img = cv2.dilate(gray, np.ones((7, 7), np.uint8))
    bg_img = cv2.medianBlur(dilated_img, 21)
    diff_img = 255 - cv2.absdiff(gray, bg_img)
    norm_img = cv2.normalize(diff_img, None, alpha=0, beta=255,
                              norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8UC1)
    # Threshold using Otsu's
    img = cv2.threshold(norm_img, 0, 255, cv2.THRESH_OTSU)[1]

    config = r'--oem 3 --psm 6'
    text = ""
    text = pytesseract.image_to_string(img, config=config, lang=img_lang)
    data = pytesseract.image_to_data(img, config=config, lang=img_lang)
    for i, el in enumerate(data.splitlines()):
        if i == 0:
            continue
        el = el.split()
        try:
            x, y, w, h = int(el[6]), int(el[7]), int(el[8]), int(el[9])
            cv2.rectangle(img, (x, y), (w + x, h + y), (0, 0, 255), 1)
            cv2.putText(img, el[11], (x, y), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 1)
        except IndexError:
            continue
    cv2.imwrite('static/files/test_complete.png', img)
    os.remove('static/files/test.png')
    return text

```

Додаток В

Лістинг функції pdf_processing

```

def pdf_processing(pdf_lang):
    ocrmypdf.ocr("static/files/test.pdf", "static/files/complete_pdf.pdf", skip_text=True)
    """
    if pdf_lang == 'eng':
        with pdfplumber.open('static/files/complete_pdf.pdf') as pdf:
            total_pages = len(pdf.pages)
            text_pdf = ""
            for i in range(0, total_pages):
                page = pdf.pages[i]
                text_pdf_temp = page.extract_text(x_tolerance=2)
                text_pdf = text_pdf + '\n' + text_pdf_temp
            os.remove('static/files/complete_pdf.pdf')
            os.remove('static/files/test.pdf')
            return text_pdf
    else:
        """
        pages = convert_from_path('static/files/complete_pdf.pdf', 500)
        text_pdf = ""
        for page in pages:
            page.save('static/files/pdf_img_temp.png', 'PNG')
            img = cv2.imread('static/files/pdf_img_temp.png', cv2.IMREAD_COLOR)
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            dilated_img = cv2.dilate(gray, np.ones((7, 7), np.uint8))
            bg_img = cv2.medianBlur(dilated_img, 21)
            diff_img = 255 - cv2.absdiff(gray, bg_img)
            norm_img = cv2.normalize(diff_img, None, alpha=0, beta=255,
                                     norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8UC1)
            # Threshold using Otsu's
            img = cv2.threshold(norm_img, 0, 255, cv2.THRESH_OTSU)[1]
            config = r'--oem 3 --psm 6'
            text = ""
            text = pytesseract.image_to_string(img, config=config, lang=pdf_lang)
            text_pdf = text_pdf + '\n' + text
        os.remove('static/files/pdf_img_temp.png')
        os.remove('static/files/complete_pdf.pdf')
        os.remove('static/files/test.pdf')
        return text_pdf

```

Extended kindness trifling remember he confined outlived if. Assistance sentiments yet unpleasing say. Open they an busy they my such high. An active dinner wishes at unable hardly no talked on. Immediate him her resolving his favourite. Wished denote abroad at branch at.

Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he place worth words it as to. Spoke now noise off smart her ready.

Ye on properly handsome returned throwing am no whatever. In without wishing he of picture no exposed talking minutes. Curiosity continual belonging offending so explained it exquisite. Do remember to followed yourself material mr recurred carriage. High drew west we no or at john. About or given on witty event. Or sociable up material bachelor bringing landlord confined. Busy so many in hung easy find well up. So of exquisite my an explained remainder. Dashwood denoting securing be on perceive my laughing so.

Him rendered may attended concerns jennings reserved now. Sympathize did now preference unpleasing mrs few. Mrs for hour game room want are fond dare. For detract charmed add talking age. Shy resolution instrument unreserved man few. She did open find pain some out. If we landlord stanhill mr whatever pleasure supplied concerns so. Exquisite by it admitting cordially september newspaper an. Acceptance middletons am it favourable. It it oh happen lovers afraid.

Supplied directly pleasant we ignorant ecstatic of jointure so if. These spoke house of we. Ask put yet excuse person see change. Do inhabiting no stimulated unpleasing of admiration he. Enquire explain another he in brandon enjoyed be service. Given mrs she first china. Table party no or trees an while it since. On oh celebrated at be announcing dissimilar insipidity. Ham marked engage oppose cousin ask add yet.

So delightful up dissimilar by unreserved it connection frequently. Do an high room so in paid. Up on cousin ye dinner should in. Sex stood tried walls manor truth shy and three his. Their to years so child truth. Honoured peculiar families sensible up likewise by on in.

Is at purse tried jokes china ready decay an. Small its shy way had woody downs power. To denoting admitted speaking learning my exercise so in. Procured shutters mr it feelings. To or three offer house begin taken am at. As dissuade cheerful overcame so of friendly he indulged unpacked. Alteration connection to so as collecting me. Difficult in delivered extensive at direction allowance.

Рисунок Г.1 – Англійський текст з шумом 10%

Extended kindness trifling remember he confined outlived it. Assistance sentiments yet unpleasing say. Open they an busy they my such high. An active dinner wishes at unable hardly no talked on. Immediate him her resolving his favourite. Wished denote abroad at branch at.

Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he place worth words it as to. Spoke now noise off smart her ready.

Ye on properly handsome returned throwing am no whatever. In without wishing he of picture no exposed talking minutes. Curiosity continual belonging offending so explained it exquisite. Do remember to followed yourself material mr recurred carriage. High drew west we no or at john. About or given on witty event. Or sociable up material bachelor bringing landlord confined. Busy so many in hung easy find well up. So of exquisite my an explained remainder. Dashwood denoting securing be on perceive my laughing so.

Him rendered may attended concerns jennings reserved now. Sympathize did now preference unpleasing mrs few. Mrs for hour game room want are fond dare. For detract charmed add talking age. Shy resolution instrument unreserved man few. She did open find pain some out. If we landlord stanhill mr whatever pleasure supplied concerns so. Exquisite by it admitting cordially september newspaper an. Acceptance middletons am it favourable. It it oh happen lovers afraid.

Supplied directly pleasant we ignorant ecstatic of jointure so if. These spoke house of we. Ask put yet excuse person see change. Do inhabiting no stimulated unpleasing of admiration he. Enquire explain another he in brandon enjoyed be service. Given mrs she first china. Table party no or trees an while it since. On oh celebrated at be announcing dissimilar insipidity. Ham marked engage oppose cousin ask add yet.

So delightful up dissimilar by unreserved it connection frequently. Do an high room so in paid. Up on cousin ye dinner should in. Sex stood tried walls manor truth shy and three his. Their to years so child truth. Honoured peculiar families sensible up likewise by on in.

Is at purse tried jokes china ready decay an. Small its shy way had woody downs power. To denoting admitted speaking learning my exercise so in. Procured shutters mr it feelings. To or three offer house begin taken am at. As dissuade cheerful overcame so of friendly he indulged unpacked. Alteration connection to so as collecting me. Difficult in delivered extensive at direction allowance.

Рисунок Г.2 – Англійський текст з шумом 50%

Співай ароматом пурхнув Криваві тлінного Лук дай тій. Втіху Котора
пінним. Оспівував ступаєш зберігання кинувся. Повз неї з її шкоду пекло
жили воль пекло ви але. Нею співаю Для поза вуха онп. . . Їм Хто з Але Се
Їхне пекло. Пригнічувало порфірам висвітлити природи сум'яття. Заходи
рани бенкет.

. Ліп тасмних Вам безодня як Гул моя Росія грів. . Довго житіє вічне вночі
статп. Загятих її Лише їм до яр Спас грім. Всесильний спустилися
ПЕТЕРБУРГЕ беззаконь ваш сем Шум Пом'якшуємося нам Вік милостивим.
Народженим каяття Заповідям мучитися тремтіла. Її на яр ти в те. Та Ея По
Ти їм нею. Був дум лют Дол дан Оне розумі одр.

Нагороди він жалість з дієслова Не вагався ми не приховував. Повстання
проводить поєднує звалимося творіння. Деся дав Покрие Був спокійно око
розуму мужайся Сад упав цівниці Чоловік. Чтя Там розширення пекла пекло
незбагненно утиснене під УПОВАЮЩЕМУ Вся променистою. Світлості
солодошів За НА смиренністю перешкод Ах По. Розпустою мудрую мене
Зласердую кайдани Та Пліняється По сопхнуться. Зруйнував її підкошені яр її
Але спокуси прагнення за Аз Уж За. Земни Мій тул Зітхання Так перст знає
раз захід дна. Вже по та До.

. Перемогам Величчя паростках. Буй лагідних Голодна бентежить повітря сім
тма піт гір повітрі Зло. Лід вид Ні моє Хто Чтя це. Чи душі Пити хвал
даремно її справ. .

Доброту іду Умудрайся Без Ненависті затемнюсь терпінням надії Дав чії
Або тощ. Закликаю Вкусил Любав долі гримів вод Сходил zde яви Для неї
тим. Перун січе топить уранці. хто. . Повсюду слухає трупами загинуло
лампада. хто. . Її вже Солі За. Б ні в променистому ти оспівувати
божевільними з на але до достатку пристойно.

Бенкет Хто Там вся співаю. хто. У незлобний стан Їм Зберігача Не І обраний
До Від пішли вогненною. Від За льщуся з'являли то чує НА чи лобному
Гнусні. Покоить Рятує кістками Він І отрута падушчим Вже Залишився У Ми
ст Подібний. Тому чад все це під.

Рисунок Г.3 – Український текст з шумом 10%

Співай ароматом пурхнув Криваві тлінного Лук дай тій. Втіху Котора
 пінним. Оспівував ступаєш зберігання кивувся. Повз неї З її школу пекло
 жити воль пекло ви але. Нею співаю Для поза вуха оні. . Ім Хто З Але Се
 Іхне пекло. Пригнічувало порфірам висвітлити природи сум'яття. Заходи
 рани бенкет.

. Лії тасмних Вам безодня як Гул моя Росія грівів. . Довго життя вічне вночі
 стати. Загягнх її Лише ім до яр Спас грім. Всесильний спустився
 ПЕТЕРБУРГЕ беззаконь ваш сем Шум Пом'якшуємося вам Вік милостивим
 Народженням каяття Заповідям мучитися тремтіла. її на яр ти в те. Та Ея По
 Ти ім нею. Був дум люот Дол дан Оне розуми одр.

Нагорода він жалість з дієлова Не загався ми не приховував. Повстання
 проводить послуге звалимося творіння. Деся дав Покрис Був спокійно око
 розуму мужайся Сад унав цівінці Чоловік. Чтя Там розширення пекла пекло
 незбагненно утиснене під УПОВАЮЩЕМУ Вся променистою. Світлості
 солотощів За НА смиренністю перешкод Ах По. Розпустою мудрую мене
 Зласердую кайдани Та Пайняться По сонхнуться. Зруйнував її підкошені яр її
 Але спокуси прагнення за Аз Уж За. Земні Мій гул Зітхання Так перет знає
 раз захід зна. Вже по та До.

. Перемога Величчя паростках. Буй лагідних Голодна бентежить повітря сім
 тма пін гір повітрі Зло. Лід вид Ні моє Хто Чтя це. Чи душі Піти хвал
 даремно її справ.

Доброту іду Умудрайся Без Ненависті загемняюсь терпінням нації Дав чії
 Або тош. Закликаю Вкусил Любов долі гримів вод Сходил zde яви Для неї
 тім. Перун січе топлить уранці хто. . Повсюду слухає трупами загинуло
 лампада. хто. . її вже Солі За. Б ні в променистому ти оспівувати
 божествними з на але до достатку пристойно.

Бенкет Хто Там вся співаю. хто. У незлобний стан Ім Зберігача Не і обраній
 До Від пішли вогненною. Від За льпуща з'являти те чуте НА чи лобному
 Гнусні. Покопть Рятую кістками Вія і отрута падушним Вже Залишився У Ми
 ст Подібний. Тому чад все це під.

Рисунок Г.4 – Український текст з шумом 50%

Extended kindness trifling remember he confined outlived if. Assistance sentiments yet unpleasing say. Open they an busy they my such high. An active dinner wishes at unable hardly no talked on. Immediate him her resolving his favourite. Wished denote abroad at branch at.

Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he place worth words it as to. Spoke now noise off smart her ready.

Ye on properly handsome returned throwing am no whatever. In without wishing he of picture no exposed talking minutes. Curiosity continual belonging offending so explained it exquisite. Do remember to followed yourself material mr recurred carriage. High drew west we no or at john. About or given on witty event. Or sociable up material bachelor bringing landlord confined. Busy so many in hung easy find well up. So of exquisite my an explained remainder. Dashwood denoting securing be on perceive my laughing so.

Him rendered may attended concerns jennings reserved now. Sympathize did now preference unpleasing mrs few. Mrs for hour game room want are fond dare. For detract charmed add talking age. Shy resolution instrument unreserved man few. She did open find pain some out. If we landlord stanhill mr whatever pleasure supplied concerns so. Exquisite by it admitting cordially september newspaper an. Acceptance middletons am it favourable. It it oh happen lovers afraid.

Supplied directly pleasant we ignorant ecstatic of jointure so if. These spoke house of we. Ask put yet excuse person see change. Do inhabiting no stimulated unpleasing of admiration he. Enquire explain another he in brandon enjoyed be service. Given mrs she first china. Table party no or trees an while it since. On oh celebrated at be announcing dissimilar insipidity. Ham marked engage oppose cousin ask add yet.

So delightful up dissimilar by unreserved it connection frequently. Do an high room so in paid. Up on cousin ye dinner should in. Sex stood tried walls manor truth shy and three his. Their to years so child truth. Honoured peculiar families sensible up likewise by on in.

Is at purse tried jokes china ready decay an. Small its shy way had woody downs power. To denoting admitted speaking learning my exercise so in. Procured shutters mr it feelings. To or three offer house begin taken am at. As dissuade cheerful overcame so of friendly he indulged unpacked. Alteration connection to so as collecting me. Difficult in delivered extensive at direction allowance.

Рисунок Г.5 – Англійський текст з dpi 300

Extended kindness trifling remember he confined outlived if. Assistance sentiments yet unpleasing say. Open they an busy they my such high. An active dinner wishes at unable hardly no talked on. Immediate him her resolving his favourite. Wished denote abroad at branch at.

Wrote water woman of heart it total other. By in entirely securing suitable graceful at families improved. Zealously few furniture repulsive was agreeable consisted difficult. Collected breakfast estimable questions in to favourite it. Known he place worth words it as to. Spoke now noise off smart her ready.

Ye on properly handsome returned throwing am no whatever. In without wishing he of picture no exposed talking minutes. Curiosity continual belonging offending so explained it exquisite. Do remember to followed yourself material mr recurred carriage. High drew west we no or at john. About or given on witty event. Or sociable up material bachelor bringing landlord confined. Busy so many in hung easy find well up. So of exquisite my an explained remainder. Dashwood denoting securing be on perceive my laughing so.

Him rendered may attended concerns jennings reserved now. Sympathize did now preference unpleasing mrs few. Mrs for hour game room want are fond dare. For detract charmed add talking age. Shy resolution instrument unreserved man few. She did open find pain some out. If we landlord stanhill mr whatever pleasure supplied concerns so. Exquisite by it admitting cordially september newspaper an. Acceptance middletons am it favourable. It it oh happen lovers afraid.

Supplied directly pleasant we ignorant ecstatic of jointure so if. These spoke house of we. Ask put yet excuse person see change. Do inhabiting no stimulated unpleasing of admiration he. Enquire explain another he in brandon enjoyed be service. Given mrs she first china. Table party no or trees an while it since. On oh celebrated at be announcing dissimilar insipidity. Ham marked engage oppose cousin ask add yet.

So delightful up dissimilar by unreserved it connection frequently. Do an high room so in paid. Up on cousin ye dinner should in. Sex stood tried walls manor truth shy and three his. Their to years so child truth. Honoured peculiar families sensible up likewise by on in.

Is at purse tried jokes china ready decay an. Small its shy way had woody downs power. To denoting admitted speaking learning my exercise so in. Procured shutters mr it feelings. To or three offer house begin taken am at. As dissuade cheerful overcame so of friendly he indulged unpacked. Alteration connection to so as collecting me. Difficult in delivered extensive at direction allowance.

Рисунок Г.6 – Англійський текст з dpi 100

Співай ароматом пурхнув Крижаві тлінного Лук дай тій. Втіху Котора
пінним. Оспівував ступаєш зберігання кинувся. Повз неї з її шкоду пекло
жили воль пекло ви але. Нею співаю Для поза вуха они. . . Їм Хто з Але Се
Їхне пекло. Пригнічувало порфірам висвітлити природи сум'яття. Заходи
рані бенкет.

. Лій таємних Вам безодня як Гул моя Росія гріхів. . Довго житіє вічне вночі
статі. Затягні її Лише їм до яр Спас грім. Всесильний спустилися
ПЕТЕРБУРГЕ беззаконь ваш сем Шум Пом'якшуємося нам Вік милостивим.
Народженням каяття Заповідям мучитися тремтіла. Її на яр ти в те. Та Ея По
Ти їм нею. Був дум лют Дол дан Оне розумі одр.

Нагороди він жалість з дієслова Не вагався ми не приховував. Повстання
проводить поєднує звалимося творіння. Деся дав Покріє Був спокійно око
розуму мужайся Сад упав цівниці Чоловік. Чтя Там розширення пекла пекло
незбагненно утиснене під УПОВАЮЩЕМУ Вся променистою. Світлості
солодошів За НА смиренністю перешкод Ах По. Розпустою мудрую мене
Зласердую кайдани Та Пліняться По сопхнуться. Зруйнував її підкошені яр її
Але спокуси прагнення за Аз Уж За. Земни Мій тул Зітхання Так перст знає
раз захід дна. Вже по та До.

. Перемогам Величчя паростках. Буй лагідних Голодна бентежить повітря сім
тма піт гір повітрі Зло. Лід вид Ні моє Хто Чтя це. Чи душі Пити хвал
даремно її справ. .

Доброту іду Умудрайся Без Ненависті затемнюсь терпінням надії Дав чії
Або тощ. Закликаю Вкусил Любав долі гримів вод Сходил zde яви Для неї
тим. Перун січе топить уранці. хто. . Повсюду слухає трупами загинуло
лампада. хто. . Її вже Солі За. Б ні в променистому ти оспівувати
божевільними з на але до достатку пристойно.

Бенкет Хто Там вся співаю. хто. У незлобний стан Їм Зберігача Не І обраний
До Від пішли вогненною. Від За льщуся з'являли то чує НА чи лобному
Гнусні. Покопть Рятує кістками Він І отрута падающим Вже Залишився У Ми
ст Подібний. Тому чад все це під.

Рисунок Г.7 – Український текст з dpi 300

Співай ароматом пурхнув Криваві тлінного Лук дай тій. Втіху Котора
пінним. Оспівував ступаєш зберігання кинувся. Повз неї З її шкоду пекло
жили воль пекло ви але. Нею співаю Для поза вуха они. . . Їм Хто З Але Се
Їхне пекло. Пригнічувало порфірам висвітлити природи сум'яття. Заходи
рані бенкет.

. Лії тасмних Вам безодня як Гул моя Росія гріхів. . Довго жптіє вічне вночі
статі. Затятих Її Лише їм до яр Спас грім. Всесильний спустилися
ПЕТЕРБУРГЕ беззаконь ваш сем Шум Пом'якшуємося нам Вік милостивим.
Народженням каяття Заповідям мучитися тремтіла. Її на яр ти в те. Та Ея По
Ти їм нею. Був дум лют Дол дан Оне розумі одр.

Нагороди він жалість з дієслова Не вагався ми не приховував. Повстання
проводить поєднує звалимося творіння. Деся дав Покріє Був спокійно око
розуму мужайся Сад упав цівниці Чоловік. Чтя Там розширення пекла пекло
незбагненно утиснене під УПОВАЮЩЕМУ Вся променистою. Світлості
солодошів За НА смиренністю перешкод Ах По. Розпустою мудрую мене
Зласердую кайдани Та Пліняться По сопхнуться. Зруйнував її підкошені яр Її
Але спокуси прагнення за Аз Уж За. Земни Мій тул Зітхання Так перст знає
раз захід дна. Вже по та До.

. Перемогам Величчя паростках. Буй лагідних Голодна бентежить повітря сім
тма піт гір повітрі Зло. Лід вид Ні моє Хто Чтя це. Чи душі Пити хвал
даремно її справ. .

Доброту іду Умудрайся Без Ненависті затемнюсь терпінням надії Дав чії
Або тощ. Закликаю Вкусил Любов долі гримів вод Сходил zde яви Для неї
тим. Перун січе топить уранці. хто. . Повсюду слухає трупами загинуло
лампада. хто. . Її вже Солі За. Б ні в променистому ти оспівувати
божевільними з на але до достатку пристойно.

Бенкет Хто Там вся співаю. хто. У незлобний стан Їм Зберігача Не І обраний
До Від пішли вогненною. Від За льщуся з'являли то чує НА чи лобному
Гнусні. Покоить Рятує кістками Він І отрута падающим Вже Залишився У Ми
ст Подібний. Тому чад все це під.

Рисунок Г.8 – Український текст з dpi 100