

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
TEXT INFORMATION RECOGNITION TOOLS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-213М
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Системи штучного інтелекту

КУШНАРЕНКО М.П.

(ПРИЗВИЩЕ та ініціали)

Керівник ІЛЬЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КУШНАРЕНКА Миколи Павловича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів розпізнавання текстової інформації. Research and Software Implementation of Text Information Recognition Tools

керівник проєкту (роботи) к.т.н., доцент, ІЛЬЯШЕНКО Матвій Борисович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “07” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

‘

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ІЛЛЯШЕНКО М.Б., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання “09” вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки.		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис)

Микола КУШНАРЕНКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Матвій ІЛ'ЯШЕНКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
86 с., 3 табл., 30 рис., 3 дод., 19 джерел.

C++, OPENCV, VISUAL STUDIO, МОВА ПРОГРАМУВАННЯ,
ПРОГРАМНИЙ ЗАСТОСУНОК, РОЗПІЗНАВАННЯ ОБРАЗІВ, ТЕКСТОВА
ІНФОРМАЦІЯ.

Об'єкт дослідження – процес розпізнавання образів.

Предмет дослідження – програмні засоби для розпізнавання текстової інформації.

Мета роботи – дослідження та розробка програмного забезпечення для розпізнавання текстової інформації.

Матеріали, методи та технічні засоби: мова програмування C++, середовище розробки Visual Studio, бібліотека OpenCV.

Результати. Розроблено програмне забезпечення для розпізнавання текстової інформації за допомогою мови програмування C++ та середовища розробки Visual Studio й бібліотеки OpenCV.

Практична цінність роботи полягає у розробці програмного забезпечення, що забезпечує розпізнавання текстової інформації.

Висновки. Виконано проєктування програмного забезпечення для розпізнавання текстової інформації. Розроблено програмне забезпечення для розпізнавання текстової інформації. Здійснено тестування розробленого програмного забезпечення для розпізнавання текстової інформації.

Галузь використання – розпізнавання текстової інформації.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 86 pages, 3 tables, 30 figures, 3 appendixes, 19 sources.

C++, OPENCV, VISUAL STUDIO, PROGRAMMING LANGUAGE, SOFTWARE APPLICATION, PATTERN RECOGNITION, TEXT INFORMATION.

The object of research is the process of pattern recognition.

The subject of the research is software for recognizing textual information.

The purpose of this work is to research and develop the software for recognizing textual information.

Materials, methods and technical tools: C++ programming language, Visual Studio development environment, OpenCV library.

Results. The software tools for recognizing textual information using the C++ programming language, the Visual Studio development environment and the OpenCV library have been developed.

The practical value of the work lies in the development of software for recognizing textual information.

Conclusions. The software to support for recognizing textual information has been designed. The software for recognizing textual information has been developed. The developed software for recognizing textual information has been tested.

Scope of use – recognition of textual information.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для розпізнавання текстової інформації	11
1.2 Аналіз програмного забезпечення для розпізнавання текстової інформації	21
1.3 Висновки за розділом 1	32
2 Матеріали і методи	34
2.1 Вибір мови програмування	34
2.2 Вибір середовища розробки для створення програмного забезпечення для розпізнавання текстової інформації	36
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації	38
2.4 Висновки за розділом 2	40
3 Опис програми	41
3.1 Структура програмного забезпечення для розпізнавання текстової інформації	41
3.2 Функціонування програмного забезпечення для розпізнавання текстової інформації	45
3.3 Розробка бази даних програмного забезпечення для розпізнавання текстової інформації	46
3.4 Проєктування інтерфейсу програмного забезпечення для розпізнавання текстової інформації	47
3.5 Висновки за розділом 3	50
4 Експлуатація, тестування та експериментальне дослідження програми	52
4.1 Призначення й умови застосування програми	52
4.2 Характеристики програми для розпізнавання текстової інформації	53
4.3 Інструкція по експлуатації програми	54

4.3.1 Звернення до програми	54
4.3.2 Вхідні й вихідні дані	54
4.3.3 Повідомлення.....	54
4.4 Виконання програмного забезпечення для розпізнавання текстової інформації.....	54
4.5 Тестування програмного забезпечення для розпізнавання текстової інформації.....	57
4.6 Висновки за розділом 4	58
Висновки	59
Перелік джерел посилань	62
Додаток А Технічне завдання	64
Додаток Б Фрагмент тексту програми	68
Додаток В Слайди презентації	79

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- OCR – Optical Character Recognition;
- БД – база даних;
- ПЗ – програмне забезпечення;
- ПЗРТІ – програмне забезпечення для розпізнавання текстової інформації;
- РО – розпізнавання образів;
- ТІ – текстова інформація;
- ШІ – штучний інтелект.

ВСТУП

Програмне забезпечення для розпізнавання текстової інформації (ПЗРТІ) дозволяє автоматизувати рутинні завдання, пов'язані з обробкою паперових документів [1]-[6].

Програмні засоби для розпізнавання текстової інформації (ТІ) відіграють важливу роль у цифровій трансформації, перетворюючи паперові архіви в електронні бази даних. Це полегшує доступ до інформації, робить її пошуковою та доступною з будь-якого місця. На платформах електронної комерції ПЗРТІ допомагає автоматично обробляти інформацію про товари з фотографій, що значно прискорює створення онлайн каталогів і поліпшує користувацький досвід [1]-[6].

ПЗРТІ також використовується в мобільних застосунках для сканування документів або квитанцій. Наприклад, мобільні банківські застосунки дозволяють зчитувати дані з чеків, що спрощує процедуру їх обробки. Це також важливо для інклюзивності, наприклад, для людей з вадами зору, яким програмні засоби для розпізнавання ТІ допомагають перекладати друкований текст у формат для синтезаторів мови [1]-[6].

Це програмне забезпечення дозволяє автоматизувати обробку друкованих документів і рукописного тексту, що знижує витрати на ручну роботу та мінімізує людські помилки. Програмні засоби для розпізнавання ТІ застосовуються в бізнесі для автоматизації введення даних, обробки документів, рахунків, банківських виписок, анкет та інших паперових джерел інформації [1]-[6].

Програмне забезпечення для розпізнавання текстової інформації є важливим інструментом для автоматизації рутинних процесів, забезпечення доступності та збереження даних, а також для підвищення ефективності в різних галузях. З розвитком технологій, таких як штучний інтелект та машинне навчання, ПЗРТІ продовжує розширювати можливості обробки й аналізу тексту, що робить його ще більш актуальним у сучасному світі [1]-[6].

Проте деякі програми для розпізнавання текстової інформації можуть не забезпечувати прийнятну точність розпізнавання символів, що обумовлює потребу у витратах додаткового часу для корекції отриманих результатів. Іншим недоліком є те, що інтеграція та використання деяких ПЗРТІ у великих компаніях може бути дорого вартісною, вартість професійного програмного забезпечення та його налаштування під конкретні потреби компанії може бути значною, особливо якщо йдеться про великі обсяги документів. Також існують проблеми з розпізнаванням спеціальних символів, що робить використання ПЗРТІ менш ефективним у певних галузях, таких як наука, технології або інженерія, де використовуються подібні символи. При обробці багатомовних документів програмне забезпечення для розпізнавання ТІ може не завжди коректно перемикається між мовами, що призводить до помилок в обробці та зменшення якості отриманих результатів [1]-[6].

Тому актуальною є розробка програмного забезпечення для розпізнавання текстової інформації. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання дослідження та розробки програмного забезпечення для розпізнавання текстової інформації.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання текстової інформації;
- розробити модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації;
- здійснити проектування програмного забезпечення для розпізнавання текстової інформації;
- створити програмне забезпечення для розпізнавання текстової інформації;
- виконати тестування розробленого програмного забезпечення для розпізнавання текстової інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для розпізнавання текстової інформації

Розпізнавання текстової інформації є технологією, яка дозволяє автоматично перетворювати текст із зображень у формат, що може бути прочитаний та оброблений комп'ютером. Іноді цей процес називають також оптичним розпізнаванням символів (Optical character recognition, OCR). Розпізнавання текстової інформації обробляє дані, отримані зі сканованих документів, фотографій або файлів із зображеннями, перетворюючи текст з них у машино зчитуваний формат [1]-[6].

Ця технологія дозволяє цифрувати текст, перетворюючи символи на слова, а слова — на речення, що полегшує подальшу роботу з отриманими даними, зокрема їх редагування та пошук. Розпізнавання текстової інформації значно спрощує процес обробки даних, знімаючи необхідність у ручному введенні інформації [1]-[6].

Програмні системи OCR поєднують апаратне та програмне забезпечення для перетворення друкованих документів у цифровий формат. Апаратні засоби, наприклад, сканери або спеціальні пристрої, розпізнають текстові шаблони, після чого програмне забезпечення здійснює обробку тексту [1]-[8].

Сучасні OCR-системи часто використовують штучний інтелект (ШІ), зокрема для розпізнавання рукописного тексту. Технологія розпізнавання текстової інформації широко застосовується організаціями для оцифрування юридичних або історичних документів, що дозволяє зберігати, редагувати та шукати їх так, ніби вони були створені за допомогою комп'ютера [1]-[8].

Технології OCR можуть працювати через сканери для оцифрування фізичних документів або бути інтегрованим як окреме програмне забезпечення або функція в інших програмах [1]-[8].

При роботі програмних систем розпізнавання текстової інформації спочатку всі сторінки копіюються, а потім програмна система перетворює текст у зображення з двома кольорами, зазвичай чорним і білим. Відскановане зображення, або растрове зображення, аналізується з точки зору світлих і темних ділянок. Програма вважає темні області символами, які слід розпізнати, а світлі частини фоном. Потім виконується попередня обробка. Цифрове зображення очищається від зайвих елементів. На цьому етапі може відбуватись виправлення перекосів або спотворень, видалення рамок чи ліній, присутніх на оригіналі, а також визначення областей, де знаходиться текст. На етапі аналізу зображення темні області зображення обробляються для пошуку букв, цифр або символів. Зазвичай цей процес полягає в розпізнаванні символів або слів по одному за раз. Символи розпізнаються за допомогою одного з двох підходів: розпізнавання шаблонів або аналіз об'єктів [1]-[8].

Програмне забезпечення розпізнавання текстової інформації використовує зіставлення шаблонів для визначення символів на основі їхньої схожості з заздалегідь відомими зразками тексту або графічних елементів. Для коректної роботи програми OCR повинна мати зразки символів із різних мов і писемностей, таких як арабська, китайська, англійська, французька, німецька та інші [1]-[8].

Етап ідентифікації елементів включає аналіз програмою окремих символів, які ще не були розпізнані. Програма дотримується правил щодо форми та особливостей конкретних букв або цифр. Після того, як символ розпізнано, він перетворюється у відповідний формат, який використовується комп'ютером для подальшої обробки [1]-[8].

Повнофункціональна OCR-програма також аналізує структуру документа, розбиваючи його на різні елементи, як-от таблиці, діаграми чи карти. Далі сторінка розділяється на рядки, а ті – на окремі символи. Програма порівнює ці символи з наявними шаблонами в базі даних. Після аналізу можливих відповідностей, програма генерує відомий текст [1]-[8].

Отримані дані зберігаються у форматі, придатному для редагування, або як PDF-документ. Деякі OCR-системи зберігають як вихідне зображення, так і результати OCR для спрощеного порівняння та надійного зберігання документів [1]-[8].

Проаналізуємо основні види програмних засобів для розпізнавання текстової інформації. Прості OCR-системи виконують аналіз поступово, порівнюючи відскановані символи з уже збереженими шаблонами. Через велику кількість можливих шрифтів і мов, він має обмежене застосування для певних типів документів [1]-[8].

Оптичне розпізнавання міток використовується для розпізнавання відміток і символів, таких як значки чи підписи, логотипи і торгові марки. Подібно до простого OCR, вона використовує зіставлення зі збереженими зразками для їх розпізнавання [1]-[8].

Інтелектуальне розпізнавання символів використовує ШІ для підвищення ефективності. Завдяки машинному навчанню або глибинному навчанню, програмне забезпечення навчається розпізнавати текст подібно до того, як це робить людина. Нейронна мережа аналізує текст і виявляє складні елементи, як-от лінії, перетини та інші нестандартні форми [1]-[8].

Розумне розпізнавання мовлення є вдосконаленням програмного забезпечення для інтелектуального розпізнавання символів, де штучний інтелект навчається розпізнавати цілі слова в межах одного зображення, що значно прискорює процес обробки [1]-[8].

До основних переваг використання технології розпізнавання текстової інформації належать: зниження витрат завдяки мінімізації або усуненню ручного введення даних, оптимізація робочих процесів за рахунок імпорту попередньо надрукованих форм і прискореного пошуку в базах даних, автоматизоване керування документами та обробка вмісту, скорочення витрат на зберігання паперових документів, централізоване та надійне зберігання даних, що захищає від пожеж, втрат або пошкоджень, підвищення доступності інформації для працівників і клієнтів, зокрема для людей з порушеннями зору,

збільшення продуктивності завдяки наданню співробітникам точних і своєчасних даних [1]-[8].

Найпоширенішим застосуванням програмних засобів розпізнавання текстової інформації є перетворення паперових документів у електронні файли. Після сканування документу програмні засоби розпізнавання текстової інформації можуть перетворити його на текстовий файл, який можна редагувати у різних програмах. Ця технологія використовується в різних галузях, включаючи освіту, фінанси, охорону здоров'я, логістику, страхування, обробку кредитних заяв, медичні записи, форми, гарантії та квитанції [1]-[8].

Оптичне розпізнавання символів часто працює у фоновому режимі, підтримуючи багато популярних програмних систем, які ми використовуємо щодня. Менш помітні варіанти застосування цієї технології включають автоматичне введення символів, допомогу для людей із вадами зору, а також індексацію документів, які можна шукати. Це можуть бути паспорти, номерні знаки, підписи, чеки, облігації, штампи та візитні картки, а також перевірка виданих табличок [1]-[8].

OCR значно спрощує обробку великих обсягів інформації, дозволяючи стискати паперові та графічні документи у файли, які можна обробляти. Щоб отримати корисні дані, документи спочатку потрібно обробити за допомогою програмних засобів розпізнавання текстової інформації, щоб видалити всі текстові шари [1]-[8].

Завдяки програмним засобам розпізнавання текстової інформації скановані документи можуть бути завантажені в центральну базу даних, де система зчитує інформацію, наприклад, з підписів, договорів та інших важливих документів. Замість того, щоб співробітники вручну вводили дані з численних файлів, організації можуть використовувати програмні засоби розпізнавання текстової інформації для автоматизації цього процесу, дозволяючи програмному забезпеченню виконувати пошук у базі даних. Програмні засоби розпізнавання текстової інформації витягують текст із

зображень і зберігають його у текстовому файлі, який можна експортувати в різні формати [1]-[8].

Оптичне розпізнавання символів є технологією, яка перетворює друкований текст на цифровий формат. Вона використовує автоматизовані процеси для конвертації сканованих документів у файли, які можна редагувати та поширювати. Наприклад, після сканування квитанції комп'ютер може зберегти її як зображення. Хоча таке зображення не дозволяє здійснювати пошук, редагування або підрахунок слів, технологія розпізнавання символів перетворює його на текстовий документ, де весь вміст можна обробляти в текстовому форматі. OCR може зчитувати дані з фотографій, файлів і сканованих документів, що спрощує роботу з контентом і знімає потребу в ручному наборі [1]-[8].

Незважаючи на цифровізацію, багато компаній досі використовують паперові документи, такі як рахунки-фактури, контракти чи юридичні документи. Обробка паперових матеріалів потребує більше фізичного простору та часу, тому безпаперові рішення стають усе популярнішими. Оскільки редагування зображень документів вручну займає багато часу, існують програмні засоби розпізнавання текстової інформації, які допомагають перетворювати зображення в текстові дані. Це дозволяє зберегти час і кошти, автоматизуючи процеси обробки даних та підвищуючи загальну ефективність [1]-[8].

OCR надає значно більше можливостей, ніж просто текстовий аналіз і пошук. Ця технологія може зробити документи доступнішими для людей із вадами зору. Процес розпізнавання OCR коригує текст і структуру, виправляючи помилки, що покращує точність інформації. Вбудовані синтезатори можуть озвучувати текст, допомагаючи сліпим користувачам отримувати доступ до інформації через друковані документи. Технологія також підтримує збільшення екрану або конвертацію тексту в шрифт Брайля, надаючи користувачам можливість сприймати текст залежно від їхніх індивідуальних потреб [1]-[8].

Програмне забезпечення розпізнавання текстової інформації працює, застосовуючи кілька етапів. Спочатку сканер зчитує текст і перетворює його на двійковий код, далі аналізується отримане зображення, визначаючи світлі зони як фон і темні як текст. Після цього йде попередня обробка, де технологія OCR покращує якість зображення. Це включає згладжування країв для усунення текстур і плям, а також виправлення проблем, що виникають під час сканування, для покращення якості документа. Програмне забезпечення також підтримує багатомовне розпізнавання тексту та визначає лінії й прямокутники на зображеннях [1]-[8].

Аналіз тексту виконується шляхом виділення характерних елементів і їхнього зіставлення з моделями. Це розділяє текст на такі компоненти, як лінії і перетини, що допомагає системі знайти найбільш відповідну форму. Наприклад, виділений символ порівнюється з наявними у базі даними. Розпізнавання шаблонів можливе, коли розмір і стиль шрифту збігаються з наявними зразками, що робить цей метод ефективним для стандартних надрукованих документів [1]-[8].

Після аналізу система перетворює розпізнані дані на текстовий файл. Деякі безкоштовні OCR-інструменти створюють документи з примітками, що містять скановані символи з обох сторін аркуша. Якщо OCR не розпізнає текст, необхідно перевірити якість сканування, рівень освітлення та відсутність подряпин на зображенні [1]-[8].

Однією з найбільш поширених сфер застосування OCR є перетворення друкованих документів у текстові формати. Інші важливі застосування включають забезпечення доступу до інформації для людей із вадами зору, автоматизацію баз даних та пошук у документах, таких як номерні знаки, страхові поліси, паспорти тощо [1]-[8].

Існує широкий вибір OCR-програм, які використовуються для повсякденних особистих і професійних цілей. Програма аналізує скановані документи та розділяє їх на частини, придатні для розпізнавання тексту. OCR-системи розпізнають окремі символи, а пакети розробки програмного

забезпечення дозволяють інтегрувати OCR у інші програми, наприклад, для керування документами чи архівування. OCR може бути налаштоване для роботи з різними форматами документів [1]-[8].

Використання програмних засобів обробки текстової інформації у бізнесі є важливим для документації та спільної роботи. З ростом компанії збільшується кількість завдань, що виконуються, і автоматизація процесів стає необхідною. Виключення ручного введення даних для деяких операцій дозволяє зосередитися на більш важливих бізнес-задачах. Завдяки автоматизації збору та зберігання даних компанії можуть підвищити ефективність і продуктивність, зменшивши витрати. Оцифровані дані стають централізованими та захищеними, а використання програмних засобів обробки текстової інформації дозволяє легко експортувати інформацію в електронні таблиці для зручного перегляду [1]-[8].

Оптичне розпізнавання символів може озвучувати текст, перетворюючи його на мовлення, що значно спрощує сприйняття інформації. Також можна змінювати кольори та розмір тексту для кращої візуалізації. Крім того, програмне забезпечення дозволяє виділяти ключові слова, фрази або абзаци для зручності запам'ятовування. До того ж, можна додавати цифрові позначки, які допомагають учням краще орієнтуватися в текстовому матеріалі [1]-[8].

У галузі охорони здоров'я програмні засоби обробки текстової інформації застосовують для управління документацією пацієнтів, зокрема результатами аналізів, медичними записами, лікуванням і страховими документами. Ця технологія спрощує процеси, зменшуючи обсяг ручної роботи в медичних установах. Програмні засоби обробки текстової інформації забезпечує легкість зберігання документів і постійну актуальність даних, зменшуючи час, необхідний для введення інформації у форми, і підвищуючи точність. Це допомагає мінімізувати ризик помилок під час введення текстових даних [1]-[8].

OCR також дозволяє отримувати інформацію з електронних медичних записів. Наприклад, історія хвороби пацієнта може зберігатися у медичній картці, і лікарі можуть швидко знайти необхідні дані за допомогою пошуку OCR. Це дозволяє медпрацівникам оперативно отримувати потрібну інформацію для прийняття рішень. Цю технологію можна також використовувати для перевірки рецептів та інших важливих документів, що допомагає зменшити кількість помилок і підвищити безпеку пацієнтів. OCR дозволяє створювати цифрові документи, заощаджуючи час і папір, а також спрощуючи зберігання [1]-[8].

Програмні засоби обробки текстової інформації у поєднанні з машинним навчанням значно вдосконалилися за останні десятиліття і продовжують розвиватися. Нові покоління OCR базуються на алгоритмах машинного навчання та штучного інтелекту, що забезпечує більш точне розпізнавання тексту, ніж традиційні програми. Незалежно від того, чи використовуються безкоштовні інструменти для роботи з файлами, чи сучасне OCR-програмне забезпечення, технологія стає все розумнішою і ефективнішою [1]-[8].

Системи оптичного розпізнавання символів є поєднанням апаратного та програмного забезпечення, що дозволяє перетворювати фізичні документи на формати, які можна редагувати. Апаратні засоби, такі як сканери або спеціальні клавіатури, використовуються для сканування або зчитування тексту, у той час як програмне забезпечення виконує рутинні функції. Додатково, технології штучного інтелекту можуть використовуватися для розпізнавання голосу або складних візерунків [1]-[8].

Процес OCR часто застосовується для оцифрування юридичних або історичних документів. Після оцифрування документу користувачі можуть редагувати, організовувати та шукати інформацію в ньому так, ніби він був створений в текстовому редакторі [1]-[8].

Першим етапом обробки текстової інформації є обробка фізичного документа за допомогою сканера. Після копіювання всіх сторінок програма

перетворює документ у двокольоровий або чорно-білий формат. Далі програма виявляє літери або цифри, спираючись на темні плями. Хоча технології OCR можуть варіюватися, зазвичай вони аналізують один символ, слово або літеру одночасно. Потім сигнали обробляються за допомогою спеціальних алгоритмів [1]-[8].

Програмне забезпечення OCR використовує зразки документів з різних джерел і форматів для порівняння та ідентифікації тексту у відсканованих файлах. Воно слідує правилам, що стосуються символів певних літер або цифр, для розпізнавання тексту [1]-[8].

Основні переваги технології OCR включають економію часу, зменшення кількості помилок, зниження зусиль і виконання завдань, які неможливі з фізичними копіями, такі як публікація файлів, індексація ключових слів, інтеграція з веббраузерами та пошук електронної пошти. Хоча зображення документів дозволяють зберігати їх у цифровому форматі, програмне забезпечення для розпізнавання текстової інформації надає додаткові можливості для організації та сканування цих файлів [1]-[8].

Оптичне розпізнавання символів або розпізнавання тексту є сучасною технологією, яка дозволяє перетворювати друковані або рукописні документи в цифрові формати за допомогою обробки зображень. OCR здатне ідентифікувати текст у зображеннях і перетворювати його на текст, що може читатися комп'ютерами. Процес зазвичай включає виявлення та розпізнавання текстових зображень, після чого вони перетворюються в текстовий формат. Однак комп'ютери можуть стикатися з труднощами в розпізнаванні символів, якщо вони написані різними шрифтами або стилями [1]-[8].

Технології розпізнавання символів, що використовують штучний інтелект, довели свою ефективність і здатні значно оптимізувати бізнес-процеси. Крім автоматизації різних бізнес-операцій, OCR також робить внесок у захист навколишнього середовища, зменшуючи кількість паперових копій документів і, відповідно, економлячи папір [1]-[8].

Оптичне розпізнавання символів являє собою процес, що дозволяє перетворювати текст із зображень у формат, який може бути прочитаний комп'ютером. Наприклад, якщо сканувати документ, такий як рахунок-фактура або квитанція, комп'ютер збереже його у вигляді зображення. Проте текст у такому файлі не може бути відредагований, знайдений через пошук або підрахований за допомогою текстових редакторів. За допомогою OCR ці зображення можна конвертувати в текстовий файл, який зберігає всі дані у вигляді тексту [1]-[8].

Ця технологія має широке застосування в різних галузях. У мобільному банкінгу OCR дозволяє зчитувати дані з чеків і використовувати їх для депозитів. В медичній сфері вона використовується для керування інформацією про пацієнтів, що полегшує роботу медичних закладів. OCR також застосовується для сканування візитних карток, що дозволяє зберігати контакти в електронному вигляді безпосередньо в адресній книзі користувача. Технологія допомагає розпізнавати номерні знаки автомобілів, що використовується для підвищення безпеки та керування паркувальними місцями, а також автоматизує обробку пошти, дозволяючи поштовим службам швидко сортувати її за адресами та поштовими індексами [1]-[8].

OCR забезпечує низку переваг. Ця технологія оцифровує друковані або рукописні документи, перетворюючи їх у цифровий формат, з яким можна легко працювати. Ця технологія робить можливим пошук по тексту в документах, що дозволяє швидко знаходити потрібну інформацію. Автоматизація процесів за допомогою OCR допомагає компаніям зекономити час і кошти, які витрачалися на ручне введення даних і обробку документів. Крім того, OCR спрощує архівування паперових документів, зменшуючи потребу у фізичному зберіганні, і підтримує роботу з багатомовними документами, що робить її корисною для глобальних компаній [1]-[8].

Визначено, що розпізнавання текстової інформації є технологією, яка дозволяє автоматично перетворювати текст із зображень у формат, що може бути прочитаний та оброблений комп'ютером. Описано процес розпізнавання

текстової інформації, що передбачає обробку даних, отриманих зі сканованих документів, фотографій або файлів із зображеннями, перетворюючи текст з них у машино зчитуваний формат. Проаналізовано основні види програмних засобів для розпізнавання текстової інформації. Описано галузі практичного застосування розпізнавання текстової інформації.

1.2 Аналіз програмного забезпечення для розпізнавання текстової інформації

Проаналізуємо програмне забезпечення для розпізнавання текстової інформації [9]-[14].

Програмне забезпечення CamScanner (рис. 1.1) є програмою для сканування документів та їх оптичного розпізнавання, яка перетворює паперові документи на цифрові файли з можливістю їх зручного зберігання. Окрім сканування, програма дозволяє швидко конвертувати документи, а також зберігати різні файли у безпечному хмарному сховищі [9], [10], [12].

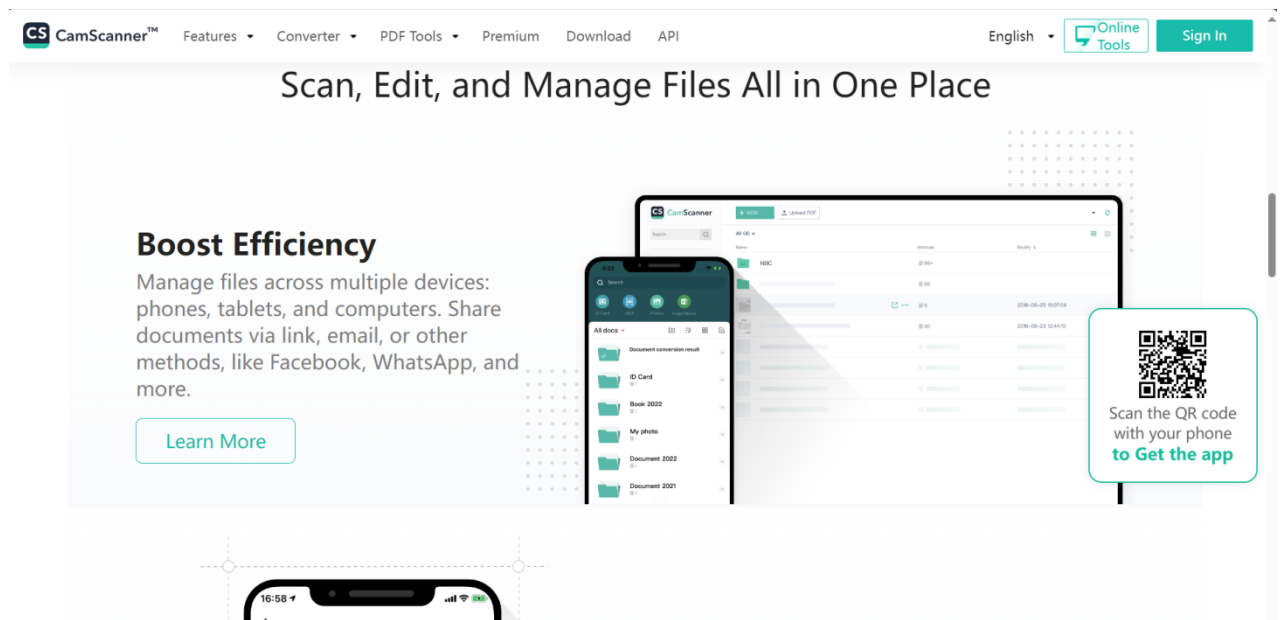


Рисунок 1.1 – Програмне забезпечення CamScanner [12]

Програмне забезпечення CamScanner є легкодоступним та інтуїтивно зрозумілим у користуванні. Ця програма з функцією OCR надає можливість сканувати, редагувати та обмінюватися файлами. Завдяки інтелектуальному обрізанню та автоматичному покращенню, документи стають більш читабельними, що підвищує якість тексту для кращого розпізнавання текстової інформації [9], [10], [12].

Серед основних можливостей CamScanner можна виділити підтримку кількох мов, включаючи англійську, французьку, німецьку, іспанську та японську. Також програма дозволяє перетворювати витягнутий текст у формати для редагування, як-от Microsoft Word, звичайний текст або розширений текстовий формат. Окрім цього, користувачі можуть конвертувати PDF у зображення, такі як JPEG або PNG, а також безпечно ділитися відсканованими документами через посилання або паролем захищені PDF-файли [9], [10], [12].

Програма пропонує підтримку клієнтів через електронну пошту та базу знань. CamScanner стане в нагоді для фахівців різних сфер, які можуть використовувати його для сканування візитних карток і створення цифрових контактів. Програма підтримує всі основні платформи, включаючи iOS, Android, macOS, Windows та вебверсії [9], [10], [12].

CamScanner здатний перетворювати зображення у файли Word, Excel або PowerPoint, а також забезпечує високу якість сканування. Серед інших функцій — конвертація PDF у редаговані формати, інтеграція електронних підписів та переклад на велику кількість мов [9], [10], [12].

Серед переваг CamScanner можна відзначити безпечне зберігання даних, просте резервне копіювання документів та зручний обмін ними на різних платформах [9], [10], [12].

Основними характеристиками програмного забезпечення CamScanner є такі [9], [10], [12]:

- засоби оптичного розпізнавання символів: програмне забезпечення CamScanner підтримує технологію OCR, що дозволяє перетворювати текст на

зображеннях у машино читабельний текст. Ця функція корисна для видобування тексту зі сканованих документів та його подальшого редагування або збереження [9], [10], [12];

- інтелектуальне обрізання та автоматичне покращення: програмне забезпечення CamScanner автоматично визначає межі документа на фото і виконує інтелектуальне обрізання, підвищуючи якість зображення. Це покращує читабельність сканованих документів [9], [10], [12];

- хмарне зберігання: програмне забезпечення CamScanner дозволяє користувачам зберігати документи у безпечному хмарному сховищі, забезпечуючи доступ до них з будь-якого пристрою. Також є функція автоматичного резервного копіювання [9], [10], [12];

- конвертація документів: програмне забезпечення CamScanner дозволяє конвертувати документи у різні текстові формати. Скановані файли можна конвертувати у різні формати зображень [9], [10], [12];

- підтримка великої кількості мов: CamScanner підтримує кілька мов для розпізнавання та витягнення тексту з документів, включаючи англійську, французьку, німецьку, іспанську, японську та інші [9], [10], [12];

- мобільність: CamScanner працює на різних платформах, включаючи iOS, Android, macOS, Windows і вебверсії. Це дозволяє користувачам сканувати документи з мобільних пристроїв у будь-який час та з будь-якого місця [9], [10], [12];

- функція електронного підпису: програмне забезпечення надає можливість додавання електронних підписів у документи за допомогою вбудованого інструменту для малювання [9], [10], [12];

- безпечний обмін документами: CamScanner забезпечує безпечний обмін файлами через посилання або створення паролем захищених PDF-файлів. Документи також можна надсилати через електронну пошту або месенджери прямо з програми [9], [10], [12];

– переклад на кілька мов: програмне забезпечення CamScanner підтримує переклад тексту на різні міжнародні мови, що робить її зручною для користувачів з різних країн [9], [10], [12];

– висока якість сканування: програмне забезпечення CamScanner дозволяє створювати високоякісні документи з опцією багаторазового сканування [9], [10], [12].

Серед недоліків програмного забезпечення CamScanner можна відзначити обмежений безкоштовний функціонал. Безкоштовна версія CamScanner має обмеження на кількість документів та обсяг хмарного сховища, для доступу до розширених функцій, таких як масова обробка файлів, захист паролем та розширені можливості редагування, необхідно оформити платну підписку. Для компаній, які працюють з великими масивами даних або документів, CamScanner може бути недостатньо потужним, оскільки його функціональність орієнтована на малі та середні підприємства. Хоча CamScanner дозволяє зберігати документи у хмарі, інтеграція з деякими сторонніми сервісами може бути обмеженою, що може ускладнювати обмін документами або роботу з ними через інші програми. [9], [10], [12].

Програмне забезпечення Rossum (рис. 1.2) представляє рішення для оптичного розпізнавання символів, засноване на штучному інтелекті, яке призначене для автоматизації збору даних з рахунків-фактур, значно скорочуючи потребу у ручному введенні інформації [9], [13]. Особливістю Rossum є те, що він не вимагає попередньо визначених шаблонів для роботи та може автоматично витягувати дані з будь-якого документа. Після обробки або сканування рахунку-фактури користувачі можуть редагувати або доповнювати інформацію та залишати свої коментарі. Програма імітує людський підхід до аналізу документів, швидко обробляючи дані, перевіряючи їх на наявність помилок, і передаючи їх для подальшого використання. Завдяки можливостям штучного інтелекту Rossum постійно навчається на кожному обробленому рахунку-фактурі, що дозволяє йому ставати точнішим з часом [9], [13].

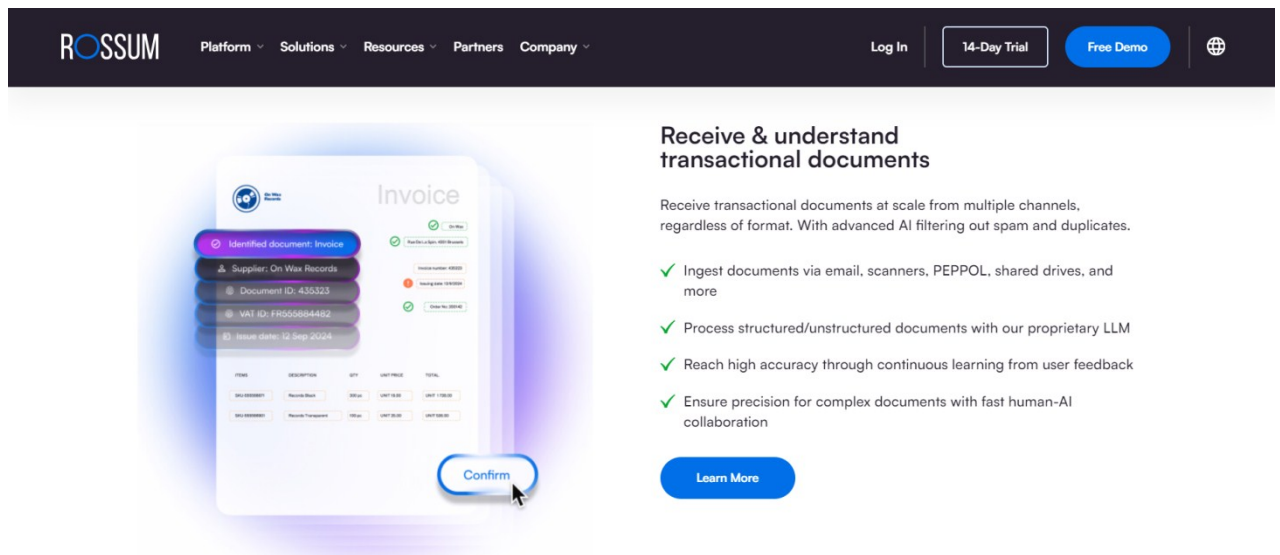


Рисунок 1.2 – Програмне забезпечення Rossum [13]

Програмне забезпечення Rossum використовує просторове OCR для кращого розуміння змісту, структури та шаблонів документів [9], [13]. Також Rossum автоматично запитує у користувачів перегляд і перевірку тих рахунків-фактур, щодо яких його система має сумніви в точності розпізнавання [9], [13].

Ця хмарна платформа спрямована на автоматизацію процесу захоплення та обміну даними без помилок у форматуванні, спрощуючи робочі процеси бізнесу та усуваючи необхідність у зворотному листуванні. Серед її функцій можна виділити наступні: сканування та передача документів до будь-якої бізнес-системи, перевірка вмісту за допомогою ШІ для полегшення співпраці, відсів непотрібних або спам-документів, організація завдань за пріоритетністю, автоматичне коригування макетів документів у разі відхилень, вдосконалення роботи через кожен етап обробки людиною, прості налаштування робочих процесів без використання серверів, доступ до розширеної аналітики та даних користувача, а також докладні журнали подій та аудиту для кожного документа [9], [13].

Програмне забезпечення здатне працювати з будь-якими типами документів, включно з рахунками-фактурами, замовленнями на купівлю та коносаментами. Rossum також забезпечує інтеграцію з багатьма сторонніми

бізнес-системами, такими як Microsoft Dynamics, Netsuite та QuickBooks. Завдяки передовій технології штучного інтелекту, система здатна розпізнавати структури в текстах та зображеннях. Крім того, Rossum пропонує індивідуальні рішення для кожного клієнта та надає аналітичні звіти в реальному часі, що дозволяє компаніям та організаціям приймати більш обґрунтовані рішення [9], [13].

Система, побудована на надійній платформі штучного інтелекту, добре справляється з обробкою документів та контенту. Однак її використання іноді може вимагати певного рівня технічної підготовки. Перевагами Rossum є високоякісна підтримка клієнтів, автоматизоване супроводження процесів співпраці, точне OCR для різних типів документів. Водночас можливі збої системи під час обробки великих файлів, затримки з оновленнями служби та проблеми з автоматичним поділом документів [9], [13].

Програмне забезпечення Rossum має такі особливості [9], [13]:

- використання засобів штучного інтелекту: програмне забезпечення Rossum використовує передову технологію штучного інтелекту для розпізнавання та обробки структур документів, що допомагає швидко та точно отримувати дані з рахунків-фактур, замовлень на купівлю та інших бізнес-документів. Система стає розумнішою з кожним новим документом, покращуючи точність [9], [13];

- гнучкість і універсальність: програмне забезпечення Rossum здатне працювати з будь-якими форматами документів, незалежно від їх структури чи шаблонів, не вимагаючи заздалегідь визначених шаблонів. Це дозволяє автоматизувати обробку документів із різних джерел [9], [13];

- автоматизація захоплення даних: програмне забезпечення Rossum автоматично захоплює дані з документів та інтегрує їх у бізнес-системи без помилок форматування. Це спрощує документообіг і усуває необхідність ручного введення даних [9], [13];

- інтелектуальна перевірка: програмне забезпечення Rossum автоматично запитує перевірку документів у випадках, коли система має

сумніви щодо точності розпізнавання. Це дозволяє зменшити кількість помилок і підвищити якість обробки даних [9], [13];

- інтеграція з бізнес-системами: програмне забезпечення Rossum підтримує інтеграцію з популярними сторонніми бізнес-системами, що дозволяє автоматично передавати дані між системами [9], [13];

- хмарна платформа: програмне забезпечення Rossum працює у хмарі, що спрощує її використання та інтеграцію з різними системами. Вона не вимагає складних налаштувань або серверної інфраструктури [9], [13];

- покращення роботи через взаємодію з користувачем: кожен етап обробки документів дає змогу вдосконалювати систему завдяки зворотному зв'язку від користувачів, що дозволяє постійно покращувати результати розпізнавання [9], [13];

- аналітика та звітність: програмне забезпечення Rossum надає детальні звіти та аналітичні дані в реальному часі, що допомагає компаніям контролювати процеси обробки документів та приймати більш обґрунтовані рішення [9], [13].

Серед недоліків програмного забезпечення Rossum можна відзначити те, що для повного налаштування та ефективного використання Rossum можуть знадобитися певні технічні навички, що може стати складністю для компаній або окремих користувачів без спеціальної підготовки. Крім того, система може мати труднощі під час обробки дуже великих документів, що іноді призводить до збоїв або затримок у процесі обробки. Для деяких малих або середніх підприємств вартість програмного забезпечення може бути досить високою, особливо якщо необхідно обробляти великі обсяги документів [9], [13].

Програмне забезпечення DocuPhase (рис. 1.3) для автоматизації бізнес-процесів та обробки документів з підтримкою технології OCR, полегшує збір і розпізнавання даних. Після сканування та вилучення інформації програма автоматично зберігає й індексує документи, організовуючи їх в єдиній корпоративній системі управління контентом [11], [14].

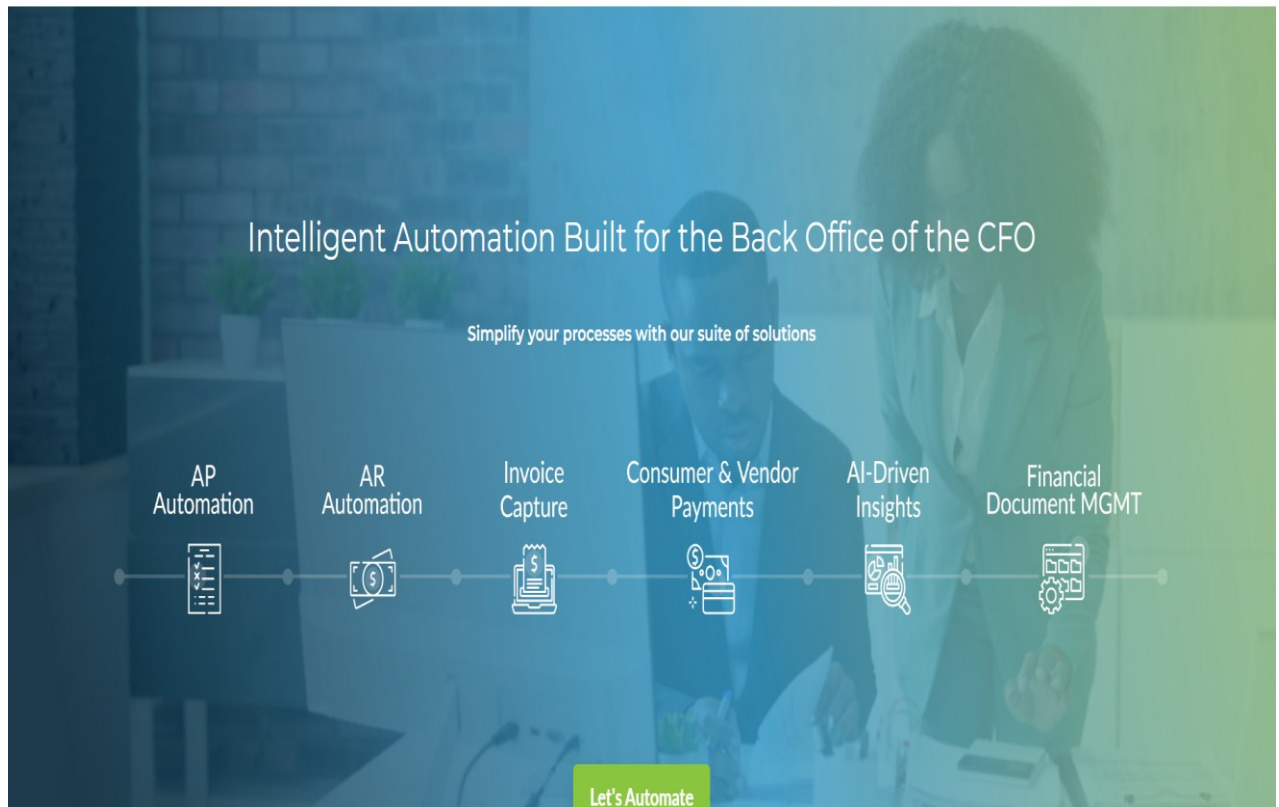


Рисунок 1.3 – Програмне забезпечення DocuPhase [14]

Програмне забезпечення DocuPhase також дозволяє витягувати конкретні дані з документів, наприклад, назви товарів, ціни, дати та іншу інформацію, у структурованому форматі [11], [14].

Програмне забезпечення DocuPhase дозволяє шукати конкретні ключові слова або фрази у сканованих документах, навіть якщо оригінал не був текстовим. Це скорочує час на пошук файлів вручну. Програмне забезпечення DocuPhase підвищує ефективність, дозволяючи віддаленим співробітникам сканувати й індексувати документи зі своїх мобільних пристроїв у будь-який час і будь-де через додаток для Android та iOS. Програма DocuPhase дозволяє імпортувати дані з різних джерел, таких як Outlook та Excel, і експортувати їх у формати CSV або PDF для подальшого аналізу [11], [14].

Ця система підходить для компаній, які потребують не тільки обробки документів за допомогою OCR, але й автоматизації процесів. DocuPhase також підтримує роботизовану автоматизацію процесів, що дозволяє автоматизувати задачі, такі як введення даних та синхронізація інформації між платформами, імітуючи дії користувача [11], [14].

Програмне забезпечення DocuPhase має такі особливості [11], [14]:

- автоматизація бізнес-процесів: платформа DocuPhase автоматизує різноманітні процеси, дозволяючи компаніям покращити ефективність і зменшити ручну роботу. Вона може автоматично обробляти завдання, такі як введення даних, синхронізація інформації, перевірка та контроль, імітуючи дії людини [11], [14];

- засоби оптичного розпізнавання символів: програмне забезпечення DocuPhase дозволяє автоматично сканувати документи, витягувати текст і дані, і зберігати їх у структурованому вигляді. Це полегшує подальший пошук та аналіз інформації [11], [14];

- повнотекстовий пошук: користувачі можуть легко шукати конкретні ключові слова або фрази у сканованих документах, навіть якщо оригінальний документ не є текстовим [11], [14];

- організація та управління контентом: після сканування документів, програма автоматично створює файли та індексує їх у єдиній корпоративній системі керування вмістом, що спрощує доступ до інформації [11], [14];

- витягування даних: DocuPhase дозволяє користувачам витягати певні елементи даних з документів, такі як назви продуктів, ціни, дати тощо, у структурованому форматі [11], [14];

- мобільна підтримка: платформа підтримує зйомку й індексацію документів за допомогою мобільних пристроїв. Спеціальний додаток дозволяє користувачам сканувати та працювати з документами на ходу через Android та iOS [11], [14];

- інтеграція з іншими системами: платформа легко інтегрується з іншими бізнес-системами, такими як Outlook, Excel, SAP, QuickBooks тощо.

Це дозволяє експортувати дані у формати CSV або PDF для подальшого аналізу [11], [14];

- роботизована автоматизація процесів: програмне забезпечення DocuPhase підтримує технологію автоматизації процесів, яка дозволяє автоматизувати задачі, такі як введення даних, перевірка та синхронізація, імітуючи дії оператора [11], [14];

- аналітика та звітність: програмне забезпечення DocuPhase надає розширені можливості аналізу даних із докладною аналітикою, що допомагає компаніям приймати більш обґрунтовані рішення на основі реальних показників [11], [14];

- гнучкість і налаштування: програмне забезпечення DocuPhase пропонує різноманітні налаштування, що дозволяє адаптувати її під конкретні бізнес-процеси та потреби організації [11], [14].

Серед недоліків програмного забезпечення DocuPhase можна відзначити складність налаштування та тривалий період навчання: з огляду на розширені можливості платформи, може знадобитися більше часу для навчання персоналу, щоб ефективно використовувати всі функції програмного забезпечення [11], [14]. Іншим недоліком є обмежені можливості інтеграції: хоча DocuPhase пропонує інтеграцію з багатьма популярними бізнес-системами, деякі користувачі можуть стикатися з труднощами під час інтеграції з менш відомими або спеціалізованими програмами. Крім того, для належної роботи програмного забезпечення DocuPhase та його постійного вдосконалення може знадобитися регулярна технічна підтримка, що інколи викликає незручності у випадку відсутності швидкої допомоги [11], [14]. Для невеликих підприємств функціонал програмного забезпечення DocuPhase може бути надмірним або не відповідати їхнім потребам, через що використання DocuPhase може виявитися надто складним або дорогим рішенням [11], [14].

Порівняльну характеристику програмного забезпечення для розпізнавання текстової інформації наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для розпізнавання текстової інформації

Критерій порівняння	CamScanner [12]	Rossum [13]	DocuPhase [14]
Підтримка великої кількості мов	+	+–	–
Використання засобів штучного інтелекту	+–	+	+–
Низька ціна підписки	–	+	+–
Ефективність роботи з великою кількістю документів	–	+	+–
Засоби інтеграції	+–	+	+–
Інтелектуальне обрізання та автоматичне покращення	+	+–	–

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для розпізнавання текстової інформації. Проте деякі програмні засоби для розпізнавання текстової інформації можуть не забезпечувати прийнятну точність розпізнавання символів, що обумовлює потребу у витратах додаткового часу для корекції отриманих результатів. Іншим недоліком є те, що інтеграція та використання деяких програм для розпізнавання ТІ може бути дорого вартісною, вартість професійного програмного забезпечення та його налаштування під конкретні потреби компанії може бути значною, особливо якщо йдеться про великі обсяги документів. Також існують проблеми з розпізнаванням спеціальних символів, що робить використання таких програм менш ефективним у певних галузях, таких як наука, технології або інженерія, де використовуються подібні символи. При обробці багатомовних документів програмне забезпечення для розпізнавання ТІ може не завжди коректно перемикається між мовами, що призводить до помилок в обробці та зменшення якості

отриманих результатів. Тому актуальною є розробка програмного забезпечення для розпізнавання текстової інформації.

При розробці програмного забезпечення для розпізнавання текстової інформації необхідно забезпечити такі функціональні вимоги:

- підтримка можливості авторизації користувачів;
- можливість завантаження графічного файлу, що містить текстову інформацію у вигляді зображення;
- можливість розпізнавання текстової інформації, поданої у вигляді зображення;
- підтримка можливості створення зображень символів для можливості їх подальшого розпізнавання;
- можливість виведення ймовірності віднесення символу, що розпізнається, до відповідних класів (символів);
- підтримка можливості автоматичного генерування символів за допомогою різних шрифтів;
- збереження результатів розпізнавання текстової інформації, поданої у вигляді зображення.

1.3 Висновки за розділом 1

Визначено, що розпізнавання текстової інформації є технологією, яка дозволяє автоматично перетворювати текст із зображень у формат, що може бути прочитаний та оброблений комп'ютером. Описано процес розпізнавання текстової інформації, що передбачає обробку даних, отриманих зі сканованих документів, фотографій або файлів із зображеннями, перетворюючи текст з них у машино зчитуваний формат. Проаналізовано основні види програмних засобів для розпізнавання текстової інформації. Описано галузі практичного застосування розпізнавання текстової інформації.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для розпізнавання текстової

інформації. Проте деякі програмні засоби для розпізнавання текстової інформації можуть не забезпечувати прийнятну точність розпізнавання символів, що обумовлює потребу у витратах додаткового часу для корекції отриманих результатів. Іншим недоліком є те, що інтеграція та використання деяких програм для розпізнавання ТІ може бути дорого вартісною, вартість професійного програмного забезпечення та його налаштування під конкретні потреби компанії може бути значною, особливо якщо йдеться про великі обсяги документів. Також існують проблеми з розпізнаванням спеціальних символів, що робить використання таких програм менш ефективним у певних галузях, таких як наука, технології або інженерія, де використовуються подібні символи. При обробці багатомовних документів програмне забезпечення для розпізнавання ТІ може не завжди коректно перемикається між мовами, що призводить до помилок в обробці та зменшення якості отриманих результатів. Тому актуальною є розробка програмного забезпечення для розпізнавання текстової інформації.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання текстової інформації.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для розпізнавання текстової інформації було використано мову програмування C++ [15], [16].

C++ – це мова програмування, яка підтримує різні парадигми розробки програмного забезпечення, зокрема, об'єктно-орієнтовану, загальну та процедурну. Мова базується на C, яку визначає стандарт ISO/IEC 14882:2003. При розробці C++ метою було зберегти сумісність із C, тому більшість програм на C можуть компілюватися за допомогою компілятора C++. Синтаксис C++ подібний до синтаксису C, а нові функції мови включають підтримку об'єктно-орієнтованого програмування, шаблонів і стандартних бібліотек [15], [16].

Основні нововведення C++ в порівнянні з C включають підтримку узагальнених програмних моделей, використання більш розширених бібліотек, а також різні оптимізації та покращення роботи з пам'яттю. C++ також сприяє кращому управлінню ресурсами і дозволяє створювати ефективні програми для різних сфер застосування [15], [16].

Об'єктно-орієнтоване програмування, що ефективно реалізується мовою C++, є парадигмою, яка розглядає програмування як роботу з об'єктами, що взаємодіють між собою. Об'єктно-орієнтоване програмування базується на трьох ключових концепціях: інкапсуляції, спадкуванні та поліморфізмі. Основною перевагою об'єктно-орієнтованого програмування є те, що воно дозволяє підвищити модульність коду, оскільки замість великої кількості функцій процедурних мов можна використовувати класи, кожен з яких має свої методи. Об'єктно-орієнтовані мови програмування набувають значного поширення завдяки тісному зв'язку між системним і прикладним рівнями [15], [16].

Варто відзначити, що для розробки програмного забезпечення для розпізнавання текстової інформації було використано бібліотеку OpenCV, яка

є потужною бібліотекою для різноманітних завдань у галузі комп'ютерного зору, обробки зображень і графіки. Вона надає інструменти для аналізу та обробки зображень, включно з розпізнаванням об'єктів (таких як людські обличчя, текст і інші зображення), відстеженням руху, перетвореннями зображень, а також дослідженнями в галузі механічного дизайну та міждисциплінарними знаннями, що використовуються у роботі з різними типами візуальних даних [17].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для розпізнавання текстової інформації наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для розпізнавання текстової інформації

Критерій порівняння мов програмування	Мова програмування		
	C++	Python	Java
Висока продуктивність	+	+–	+
Широкий набір бібліотек для обробки зображень	+	+	–
Можливість компіляції виконуваних файлів	+	–	+–
Зручність для створення ПЗ для розпізнавання текстової інформації	+	–	–
Ефективність використання ресурсів системи	+	+–	+–

Отже, для реалізації програмного забезпечення для розпізнавання текстової інформації обрано мову програмування C++, яка забезпечує високу продуктивність і ефективність використання ресурсів системи. Крім того, C++ має широкий набір бібліотек для обробки зображень і розпізнавання тексту,

таких як OpenCV і Tesseract, що спрощує інтеграцію готових рішень і прискорює розробку програмного забезпечення для розпізнавання текстової інформації. Ці бібліотеки вже містять ефективні алгоритми для обробки графічних даних, що дозволяє фокусуватися на побудові конкретного рішення, а не на розробці базових алгоритмів з нуля.

2.2 Вибір середовища розробки для створення програмного забезпечення для розпізнавання текстової інформації

В якості середовища розробки для створення програмного забезпечення для розпізнавання текстової інформації було обрано середовище Visual Studio [18], [19].

Microsoft Visual Studio — це потужне інтегроване середовище розробки від компанії Microsoft, яке включає в себе різноманітні інструменти для створення програмного забезпечення. Це середовище дозволяє розробляти різні типи програмних продуктів: від консольних програм і застосунків із графічним інтерфейсом до вебзастосунків і сайтів. Visual Studio підтримує розробку для багатьох платформ на основі Microsoft Windows, таких як Windows Mobile, Windows Phone, Windows CE, а також забезпечує засоби для створення програм під NET Framework, NET Compact Framework та Microsoft Silverlight [18], [19]. На даний момент Visual Studio стала багатоплатформеним рішенням, доступним для завантаження безкоштовно за умови реєстрації. Це середовище підтримує розробку не тільки для Windows, але й для iOS та Android, що робить його універсальним інструментом для створення мобільних застосунків [18], [19].

Visual Studio пропонує потужний і зручний інтерфейс для розробників, що дозволяє ефективно управляти складними проектами, зокрема проектами для розпізнавання тексту. Воно надає інтегровані інструменти для написання, тестування та налагодження коду, що прискорює процес розробки й допомагає швидко знаходити та виправляти помилки [18], [19].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання текстової інформації наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання текстової інформації

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	Dev-C++	Eclipse
Зручність використання інтегрованої системи контролю версій	+	–	–
Великий набір плагінів і розширень	+	–	+
Потужний і зручний інтерфейс для розробників	+	+–	+–
Підтримка розробки на мові C++	+	+	+–
Засоби інтеграції з іншими інструментами	+	+–	+–

Таким чином, для створення програмного забезпечення для розпізнавання текстової інформації обрано середовище розробки Visual Studio, що пропонує потужний і зручний інтерфейс для розробників, який дозволяє ефективно управляти складними проєктами, зокрема проєктами для розпізнавання тексту. Крім того, Visual Studio має вбудовану підтримку C++, що є важливою перевагою для розробки продуктивного програмного забезпечення для розпізнавання тексту. Ще однією перевагою Visual Studio є можливість використання інтегрованої системи контролю версій, яка дозволяє легко керувати змінами в коді та співпрацювати з іншими розробниками.

Visual Studio також забезпечує кросплатформну розробку, що дає змогу створювати рішення, які можуть працювати на різних операційних системах.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації

З метою можливості розробки програмного забезпечення для розпізнавання текстової інформації створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації враховуємо те, що користувачі програми мають можливості авторизації, завантаження графічного файлу, що містить текстову інформацію у вигляді зображення, розпізнавання текстової інформації, поданої у вигляді зображення, створення зображень символів для можливості їх подальшого розпізнавання, виведення ймовірності віднесення символу, що розпізнається, до відповідних класів (символів), можливості автоматичного генерування символів за допомогою різних шрифтів, а також збереження результатів розпізнавання текстової інформації, поданої у вигляді зображення.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для розпізнавання текстової інформації можна описати таким чином:

- користувач програмного забезпечення для розпізнавання текстової інформації авторизується;
- після авторизації користувача програмного забезпечення для розпізнавання текстової інформації йому відображається головна форма;
- користувач бачить головну форму програмного забезпечення для розпізнавання текстової інформації та обирає графічний файл з зображенням, яке містить текстову інформацію;

- користувач натискає відповідну кнопку та запускає процес розпізнавання текстової інформації, поданої у вигляді зображення;
- користувач отримує результат у вигляді тексту, що містить результати розпізнавання тексту на зображенні;
- користувач завершує роботу з програмним забезпеченням для розпізнавання текстової інформації.

Враховуючи сценарії роботи користувачів при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

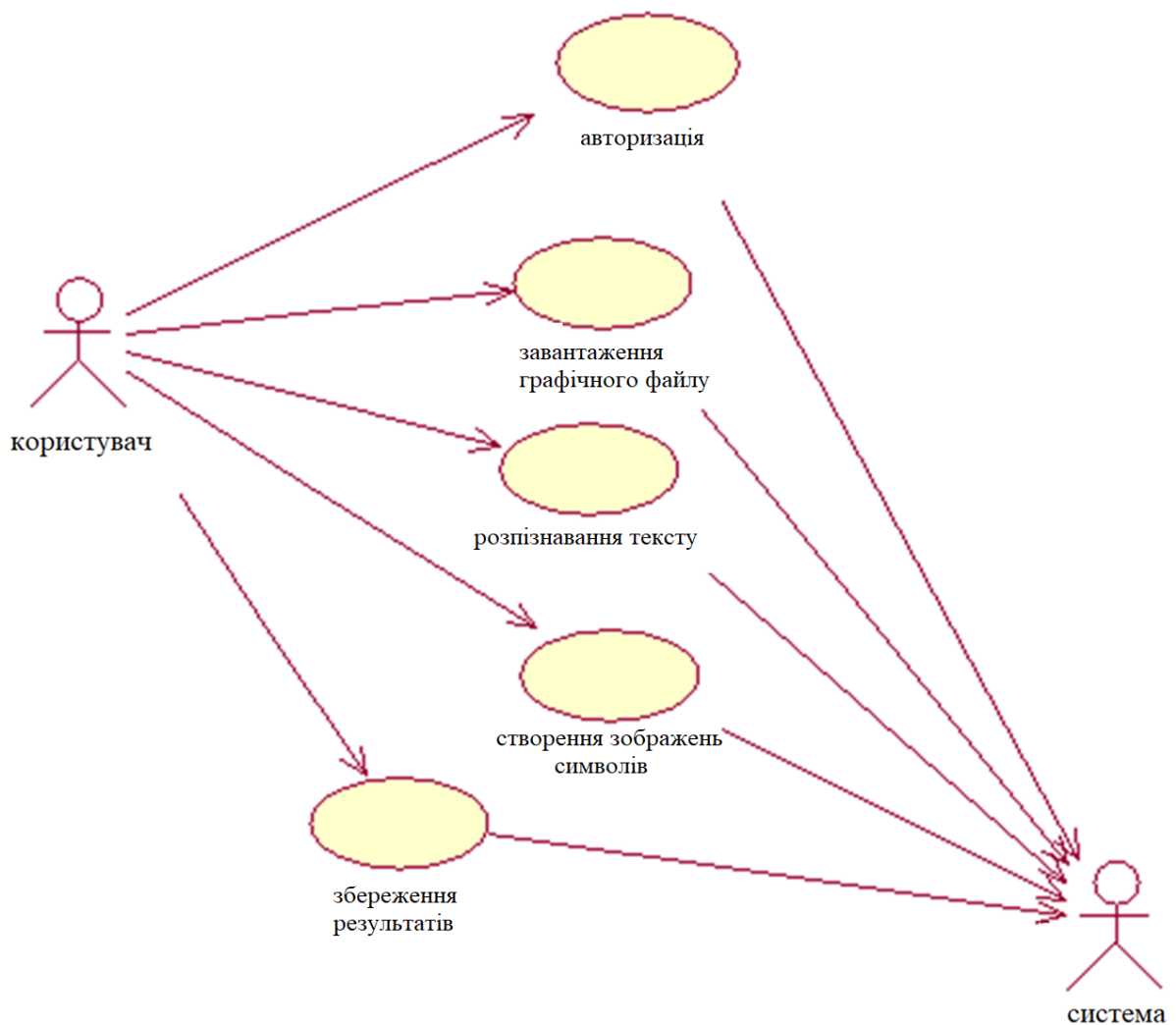


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для розпізнавання текстової інформації.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для розпізнавання текстової інформації обрано мову програмування C++, яка забезпечує високу продуктивність і ефективність використання ресурсів системи. Крім того, C++ має широкий набір бібліотек для обробки зображень і розпізнавання тексту, таких як OpenCV і Tesseract, що спрощує інтеграцію готових рішень і прискорює розробку програмного забезпечення для розпізнавання текстової інформації. Ці бібліотеки вже містять ефективні алгоритми для обробки графічних даних, що дозволяє фокусуватися на побудові конкретного рішення, а не на розробці базових алгоритмів з нуля.

Для створення програмного забезпечення для розпізнавання текстової інформації обрано середовище розробки Visual Studio, що пропонує потужний і зручний інтерфейс для розробників, який дозволяє ефективно управляти складними проєктами, зокрема проєктами для розпізнавання тексту. Крім того, Visual Studio має вбудовану підтримку C++, що є важливою перевагою для розробки продуктивного програмного забезпечення для розпізнавання тексту. Ще однією перевагою Visual Studio є можливість використання інтегрованої системи контролю версій, яка дозволяє легко керувати змінами в коді та співпрацювати з іншими розробниками.

Створено модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для розпізнавання текстової інформації.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для розпізнавання текстової інформації

Структуру програмного забезпечення для розпізнавання текстової інформації наведено на рис. 3.1.

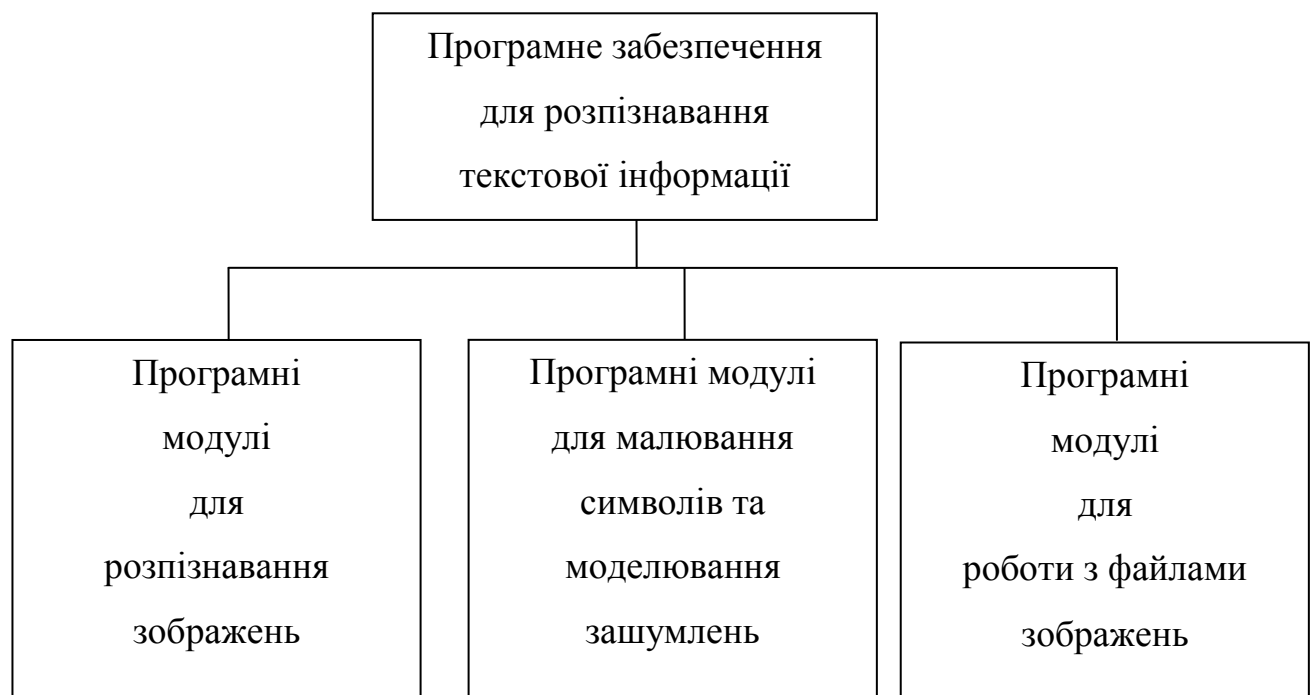


Рисунок 3.1 – Структура програмного забезпечення
для розпізнавання текстової інформації

Як видно з рис. 3.1, програмне забезпечення для розпізнавання текстової інформації складається з програмних модулів для розпізнавання зображень, модулів для малювання символів та моделювання зашумлень, а також модулів для роботи з файлами зображень.

Програмні модулі для розпізнавання зображень призначені для виконання ключових операцій, пов'язаних з обробкою та аналізом текстової інформації.

Діаграму класів модулів для розпізнавання зображень наведено на рис. 3.2.

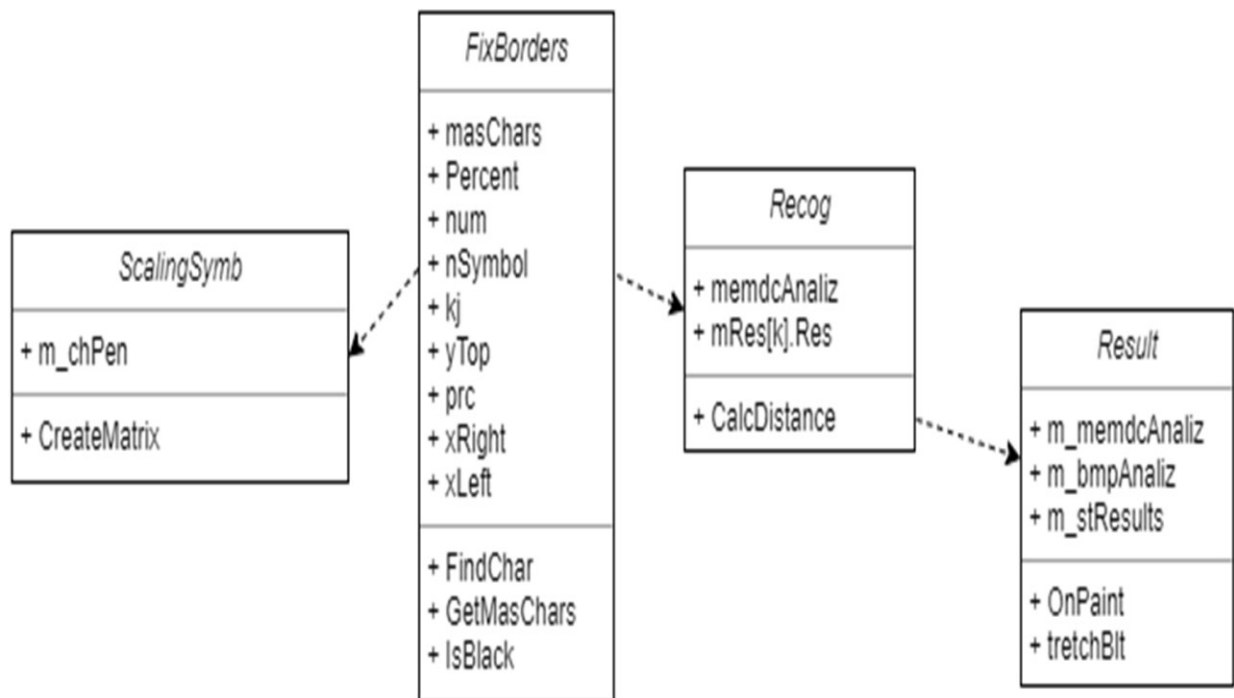


Рисунок 3.2 – Діаграма класів модулів для розпізнавання зображень

Структура модуля включає чотири основні компоненти, кожен з яких відповідає за конкретну функцію. Модуль визначення меж виконує задачу виявлення меж символів на зображенні, навіть якщо на ньому присутні шуми або спотворення. Цей етап дозволяє ізолювати символ для подальшої обробки. Після цього певний клас відповідає за нормалізацію розміру виділеного символу. Цей процес полягає в тому, щоб перетворити знайдений символ у квадратну форму, зберігаючи пропорції, що дозволяє спростити порівняння з еталонними зразками.

Основна логіка розпізнавання реалізована в класі, який використовує модифіковані методи регресійного аналізу для ідентифікації символів. Масштабований символ порівнюється з набором еталонних символів (наприклад, великими літерами українського алфавіту), що зберігаються в базі даних. Результати порівняння представляються у вигляді ймовірнісної

таблиці, яка показує ступінь схожості між розпізнаним символом і кожним еталоном. Таблиця сортується за спаданням ймовірності, що дозволяє швидко визначити найбільш імовірний варіант.

Інший клас модуля розпізнавання зображень забезпечує виведення розпізнаного символу та його ймовірностей у візуальному інтерфейсі. Вікно для аналізу, розташоване в інтерфейсі під кнопкою "Аналіз", показує масштабоване зображення символу, що розпізнається. Для зберігання обробленого символу в пам'яті створюється копія контексту вікна з білим фоном, куди додається бітове зображення, що дозволяє користувачеві переглянути результати розпізнавання в зручному форматі.

Модулі для малювання символів та моделювання зашумлень призначений для автоматичного створення або ручного промальовування символів, які потім використовуються в процесі розпізнавання. Вікно малювання, яке відображається у програмі, дає можливість користувачеві або системі зобразити символ, що буде оброблятися. Символ може бути створений вручну за допомогою миші або згенерований автоматично з використанням заздалегідь вибраного шрифту. Для збереження зображення символу в пам'яті створюється копія контексту пристрою, пов'язана з вікном малювання. Ця копія зберігає бітову картинку, заповнену білим кольором, яка відповідає розміру вікна малювання, що забезпечує коректну візуалізацію символу на екрані.

Якщо необхідно оновити відображення вікна, викликається відповідна функція, яка відповідає за повторне промальовування вікна. Бітове зображення зберігається в контексті пристрою і копіюється на екран, що дозволяє швидко відновити зображення символу після будь-яких змін або оновлень інтерфейсу. Основна відповідальність за ручне малювання символів покладена на клас, який активується під час натискання лівої кнопки миші. Цей клас контролює весь процес промальовування, дозволяючи користувачеві створювати символи безпосередньо у вікні за допомогою курсору миші.

Модуль також підтримує моделювання зашумлення для перевірки роботи системи розпізнавання в умовах перешкод. Це здійснюється за допомогою генератора псевдовипадкових чисел, який створює випадкові чорні точки на зображенні, імітуючи шуми. Коли користувач натискає кнопку Додати шум, бітова картинка, що містить символ, випадковим чином заповнюється такими точками, що дозволяє оцінити стійкість алгоритму розпізнавання до завад і спотворень.

Модуль роботи з файлами зображень призначений для завантаження й обробки зображень, що містять текст або символи, для подальшої їхньої обробки. Для відокремлення символів від фону та подальшого розпізнавання використовуються методи комп'ютерного зору. Першим етапом є завантаження зображення, після чого зображення переводиться у градації сірого. Для покращення якості подальшої обробки застосовується бінаризація, що дозволяє отримати чітке чорно-біле зображення. Після цього зображення трохи збільшується для покращення виділення контурів. Наступним кроком є знаходження контурів символів, після цього створюється ієрархічне дерево контурів, отримуються всі контури зображення, починаючи з загального контуру всієї картини, далі контури символів, і, врешті, внутрішні деталі контурів. Для подальшої роботи використовуються перевірки для фільтрації непотрібних елементів. Після того як контури символів виділені, кожен символ обробляється окремо. Спочатку створюється структура для зберігання символу в пам'яті: зображення символу зберігається у копії контексту пристрою, яка заповнюється білим кольором для подальшого промальовування. Бітова картинка зберігається в пам'яті, і якщо потрібно оновити відображення вікна, функція копіює збережене зображення символу на екран. Це забезпечує плавну візуалізацію оброблених символів під час роботи із файлами зображень, дозволяючи швидко оновлювати і відображати зміни на екрані.

3.2 Функціонування програмного забезпечення для розпізнавання текстової інформації

Функціонування програмного забезпечення для розпізнавання текстової інформації подамо за допомогою діаграми діяльності, зображеної на рис. 3.3.

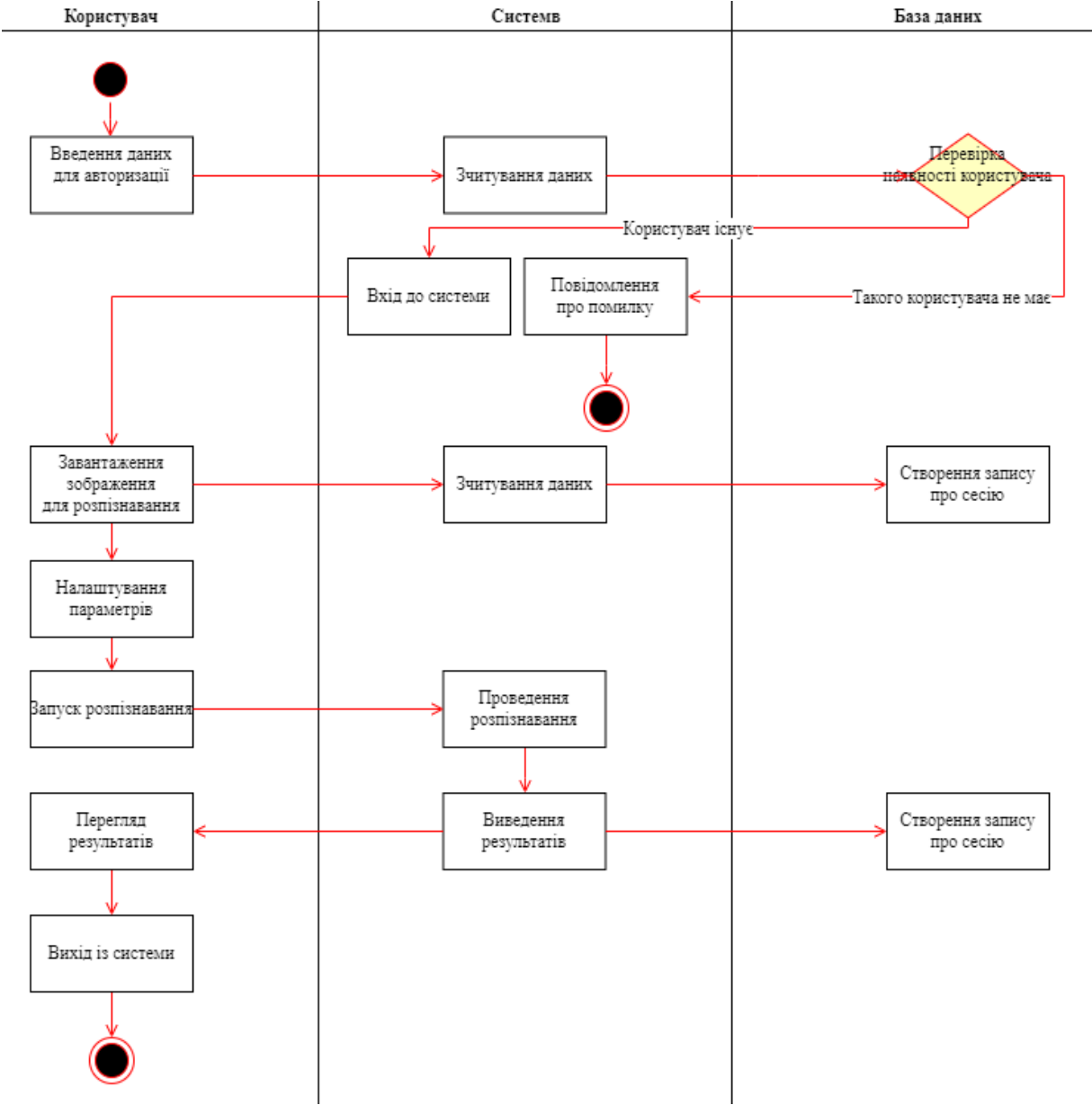


Рисунок 3.3 – Схема функціонування програмного забезпечення для розпізнавання текстової інформації у вигляді діаграми діяльності

Функціонування програмного забезпечення для розпізнавання текстової інформації починається з моменту запуску користувачем програми та введення даних для авторизації. Спочатку користувач надає вхідні дані у вигляді зображення або документа, що потребує обробки. Після цього користувач має можливість налаштувати параметри розпізнавання. Після визначення параметрів користувач ініціює процес розпізнавання. На цьому етапі програмне забезпечення починає обробку зображення: спочатку відбувається підготовка зображення шляхом виділення символів та їхньої бінаризації, а потім – безпосередній аналіз символів та порівняння їх з еталонами у базі даних. Програма використовує відповідні алгоритми для визначення символів, після чого проводиться підрахунок ймовірностей ідентифікації кожного символу.

Коли розпізнавання завершене, програмне забезпечення генерує результати, які виводяться на екран у зручному для перегляду форматі. Після перегляду та аналізу отриманих даних користувач може завершити роботу з програмою, зберігаючи результати у потрібному форматі.

3.3 Розробка бази даних програмного забезпечення для розпізнавання текстової інформації

База даних програмного забезпечення для розпізнавання текстової інформації призначена для зберігання даних про користувачів програми, інформації про зображення, що розпізнаються, а також додаткової інформації, що може бути використана у подальшій роботі програми.

Схему бази даних програмного забезпечення для розпізнавання текстової інформації наведено на рис. 3.4.

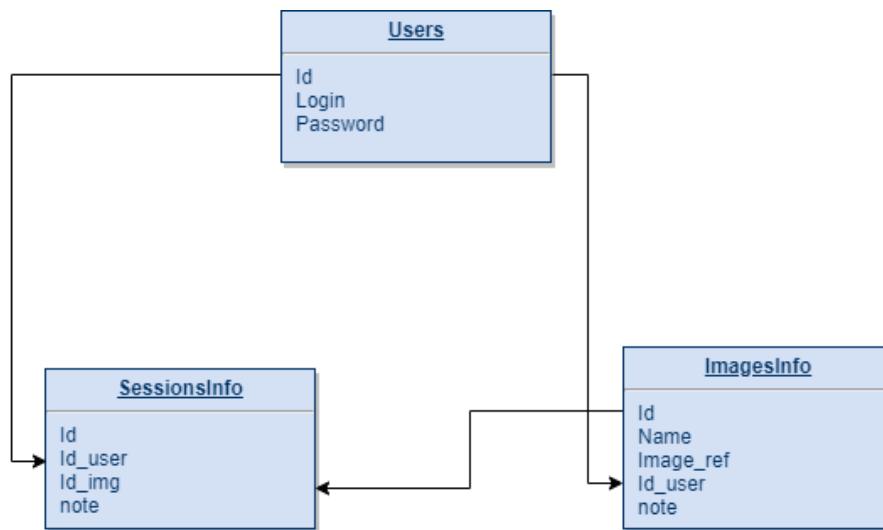


Рисунок 3.4 – Схема бази даних програмного забезпечення для розпізнавання текстової інформації

Схема бази даних програмного забезпечення для розпізнавання текстової інформації відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Основними сутностями бази даних обрано такі: користувачі (Users), інформація про зображення (ImagesInfo), інформація про програмні сесії (SessionsInfo).

3.4 Проектування інтерфейсу програмного забезпечення для розпізнавання текстової інформації

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для розпізнавання текстової інформації враховані функціональні вимоги до програми:

- підтримка можливості авторизації користувачів;
- можливість завантаження графічного файлу, що містить текстову інформацію у вигляді зображення;

- можливість розпізнавання текстової інформації, поданої у вигляді зображення;
- підтримка можливості створення зображень символів для можливості їх подальшого розпізнавання;
- можливість виведення ймовірності віднесення символу, що розпізнається, до відповідних класів (символів);
- підтримка можливості автоматичного генерування символів за допомогою різних шрифтів;
- збереження результатів розпізнавання текстової інформації, поданої у вигляді зображення.

Інтерфейс форми авторизації програмного забезпечення для розпізнавання текстової інформації наведено на рис. 3.5.

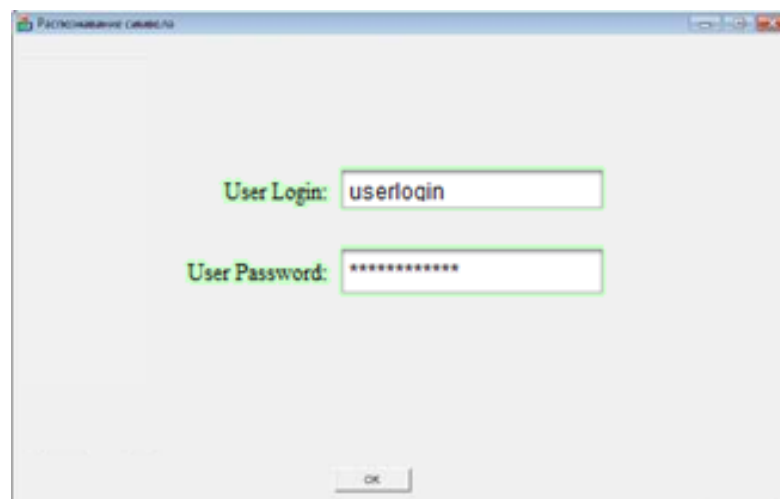


Рисунок 3.5 – Інтерфейс форми авторизації програмного забезпечення для розпізнавання текстової інформації

Інтерфейс головної форми програмного забезпечення для розпізнавання текстової інформації наведено на рис. 3.6.

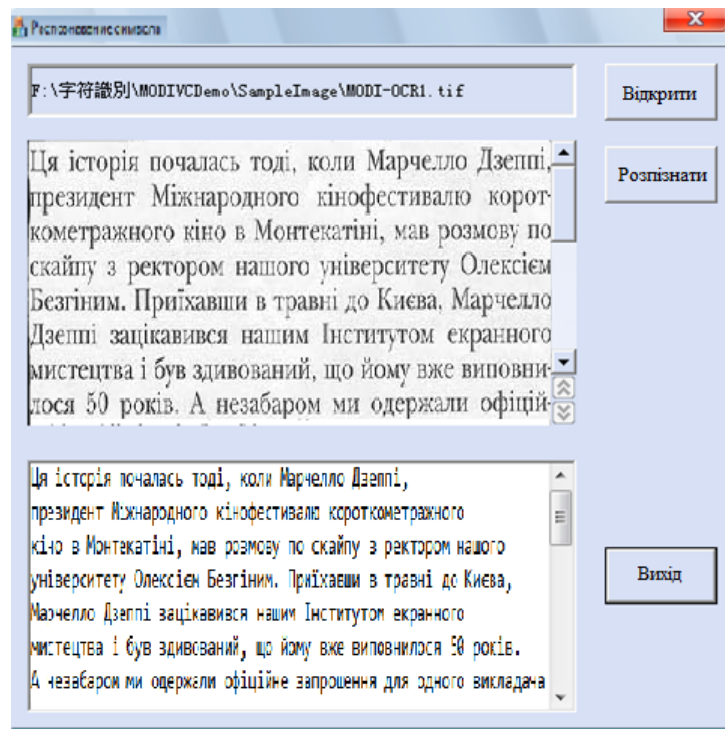


Рисунок 3.6 – Інтерфейс головної форми програмного забезпечення для розпізнавання текстової інформації

Інтерфейс форми малювання та генерування символів наведено на рис. 3.7.

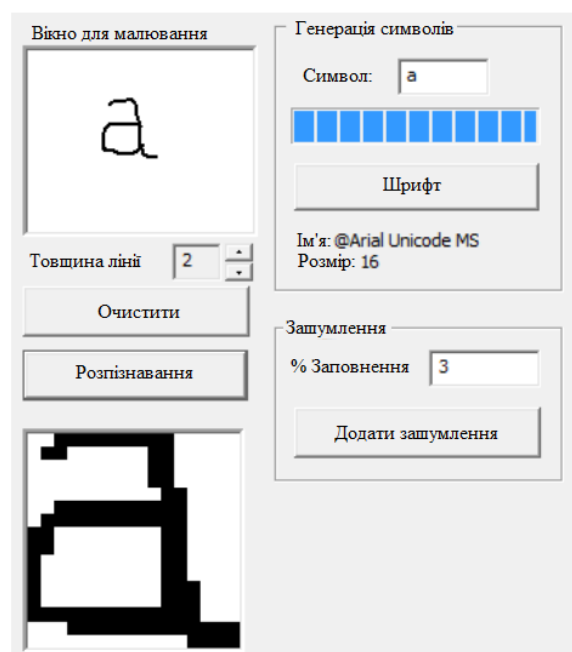


Рисунок 3.7 – Форма малювання та генерування символів

Форма результату розпізнавання символів наведена на рис. 3.8.

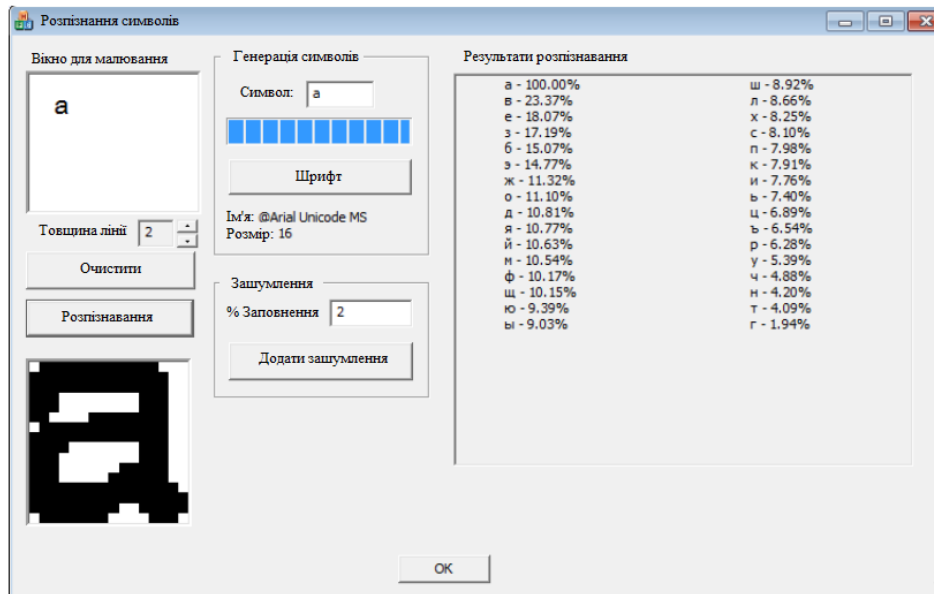


Рисунок 3.8 – Форма результату розпізнавання символів

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для розпізнавання текстової інформації.

Запропоновано структуру програмного забезпечення для розпізнавання текстової інформації. Визначено, що програмне забезпечення для розпізнавання текстової інформації складається з програмних модулів для розпізнавання зображень, модулів для малювання символів та моделювання зашумлень, а також модулів для роботи з файлами зображень.

Описано функціонування програмного забезпечення для розпізнавання текстової інформації.

Розроблено базу даних програмного забезпечення для розпізнавання текстової інформації, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази даних обрано такі: користувачі (Users), інформація про зображення (ImagesInfo), інформація про програмні сесії (SessionsInfo).

Описано особливості реалізації програмного забезпечення для розпізнавання текстової інформації. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу розпізнавання текстової інформації. Програма написана на мові C++, яка забезпечує високу продуктивність і ефективність використання ресурсів системи, має широкий набір бібліотек для обробки зображень і розпізнавання тексту, що спрощує інтеграцію готових рішень і прискорює розробку програмного забезпечення для розпізнавання текстової інформації. В якості середовища розробки для створення програмного забезпечення для розпізнавання текстової інформації обрано Visual Studio, що пропонує потужний і зручний інтерфейс для розробників, який дозволяє ефективно управляти складними проєктами.

Програмне забезпечення для розпізнавання текстової інформації забезпечує виконання таких функцій:

- підтримка можливості авторизації користувачів;
- можливість завантаження графічного файлу, що містить текстову інформацію у вигляді зображення;
- можливість розпізнавання текстової інформації, поданої у вигляді зображення;
- підтримка можливості створення зображень символів для можливості їх подальшого розпізнавання;
- можливість виведення ймовірності віднесення символу, що розпізнається, до відповідних класів (символів);
- підтримка можливості автоматичного генерування символів за допомогою різних шрифтів;
- збереження результатів розпізнавання текстової інформації, поданої у вигляді зображення.

Для експлуатації програмного забезпечення для розпізнавання текстової інформації необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 4 Гб і більше;
- 1–2 ГБ для встановлення програми та допоміжного програмного забезпечення;
- багатоядерний процесор з тактовою частотою 3,0 ГГц або вище;
- встановлена операційна система сімейства Windows.

Функціональні характеристики розробленого програмного забезпечення для розпізнавання текстової інформації наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для розпізнавання текстової інформації наведено у додатку Б.

4.2 Характеристики програми для розпізнавання текстової інформації

Програмне забезпечення для розпізнавання текстової інформації має модульну структуру. Програмне забезпечення для розпізнавання текстової інформації підтримує роботу з різними форматами графічних файлів, зокрема, JPEG, PNG, TIFF та PDF, що дозволяє обробляти зображення різних типів і розмірів. Програма забезпечує високу точність розпізнавання, особливо для текстів з чіткою роздільною здатністю, що важливо для забезпечення коректної обробки та мінімізації помилок.

Крім того, програмне забезпечення для розпізнавання текстової інформації здатне працювати з багатомовними документами, включаючи підтримку різних алфавітів і спеціальних символів. Важливим аспектом є швидкість обробки, що дозволяє відносно швидко обробляти різні обсяги тексту або зображень. Також вона повинна підтримувати можливість збереження розпізнаного тексту у відповідний текстовий файл.

Програмне забезпечення для розпізнавання текстової інформації написано на мові C++ з використанням інтегрованого середовища розробки Visual Studio.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для розпізнавання текстової інформації необхідно у каталозі з програмою запустити виконуваний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для розпізнавання текстової інформації є інформація про користувача, файл із зображенням у вигляді тексту.

Вихідними даними програмного забезпечення для розпізнавання текстової інформації є тексти, що являють собою результати розпізнавання вхідних зображень.

4.3.3 Повідомлення

Користувач програмного забезпечення для розпізнавання текстової інформації може отримати такі повідомлення: повідомлення про успішне розпізнавання тексту, повідомлення про відсутність тексту на зображенні та ін.

4.4 Виконання програмного забезпечення для розпізнавання текстової інформації

Після запуску програмного забезпечення для розпізнавання текстової інформації користувачу відображається форма авторизації (рис. 4.1), де необхідно ввести ім'я та пароль користувача.

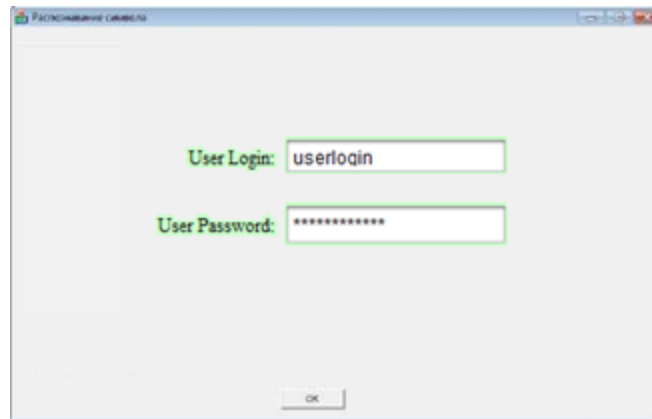


Рисунок 4.1 – Інтерфейс форми авторизації програмного забезпечення для розпізнавання текстової інформації

Після авторизації відображається головна форма (рис. 4.2), де користувач може обрати графічний файл із зображенням для подальшого розпізнавання тексту (для цього необхідно натиснути кнопку Відкрити). Зображення з файлу буде показано у відповідному полі головної форми. При натисканні на кнопку Розпізнати у нижньому блоці відображається розпізнаний текст.

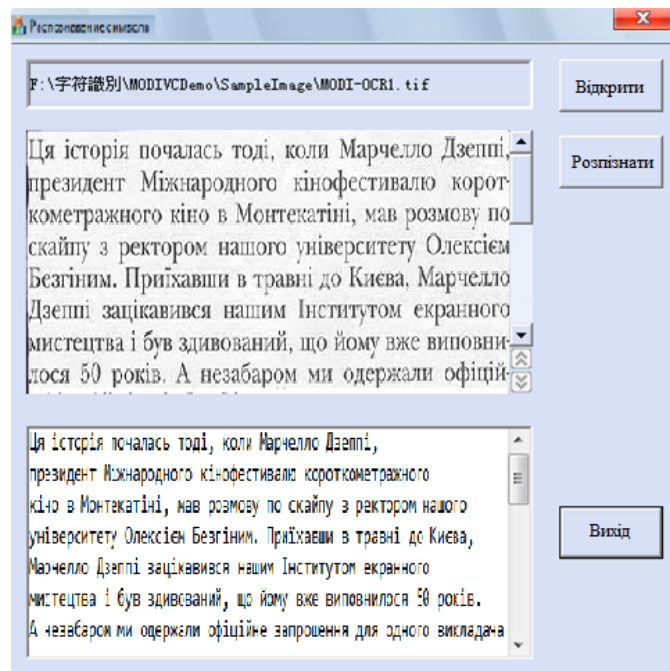


Рисунок 4.2 – Головна форма програмного забезпечення для розпізнавання текстової інформації

Програмне забезпечення для розпізнавання текстової інформації надає можливість працювати у режимі генерування символів. У відповідній формі (рис. 4.3) користувач може намалювати символ самостійно, або згенерувати його за допомогою обраного стандартного шрифту. Для генерування символу за допомогою обраного шрифту користувач вводить потрібний символ у полі Символ, натискає кнопку Шрифт, та у вікні, що з'являється, обирає потрібний шрифт.

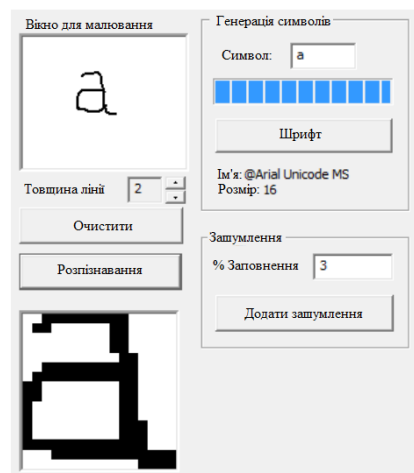


Рисунок 4.3 – Форма малювання та генерування символів

На рис. 4.4 наведено приклад генерування символу за допомогою обраного шрифту.

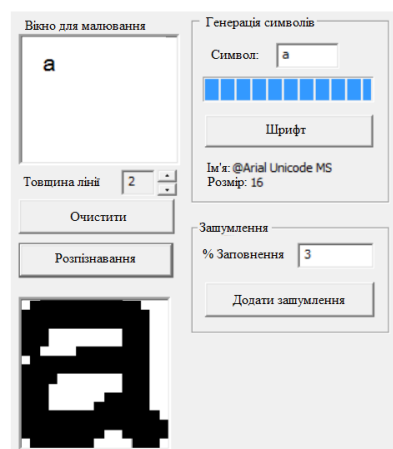


Рисунок 4.4 – Результат генерування символу

При натисканні кнопки Розпізнавання виконується розпізнавання тестового символу, результат чого виводиться у відповідному вікні (рис. 4.5), де, зокрема, наведено ймовірність віднесення символу до відповідних класів (символів базового алфавіту).

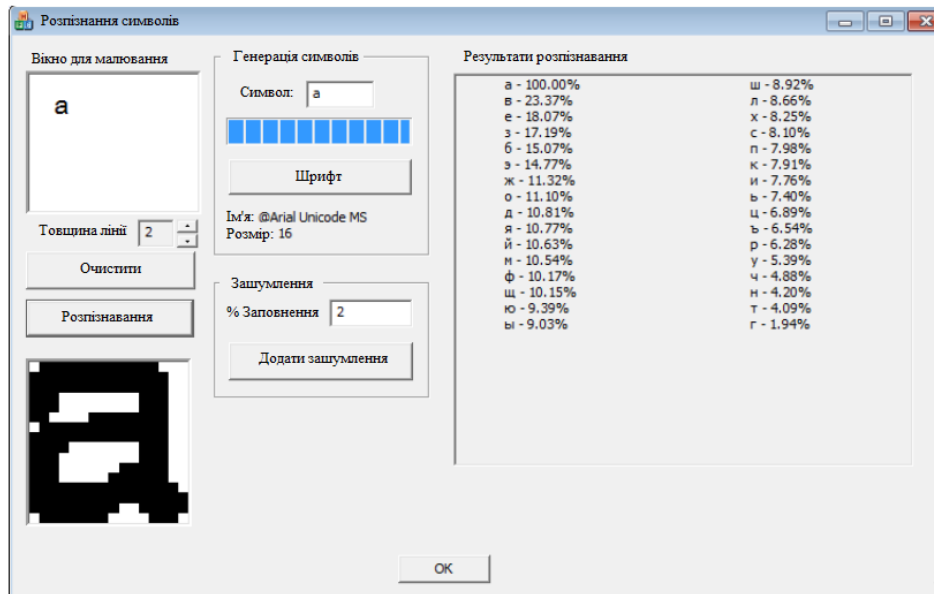


Рисунок 4.5 – Форма результату розпізнавання символів

4.5 Тестування програмного забезпечення для розпізнавання текстової інформації

Виконано тестування програмного забезпечення для розпізнавання текстової інформації.

Виявлено, що програма для підтримки процесу розпізнавання текстової інформації функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу управління розпізнавання текстової інформації.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу розпізнавання текстової інформації.

4.6 Висновки за розділом 4

Описано програмне забезпечення для розпізнавання текстової інформації. Виконано тестування розробленого програмного забезпечення для розпізнавання текстової інформації. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління розпізнавання текстової інформації.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розпізнавання образів.

Визначено, що розпізнавання текстової інформації є технологією, яка дозволяє автоматично перетворювати текст із зображень у формат, що може бути прочитаний та оброблений комп'ютером. Описано процес розпізнавання текстової інформації, що передбачає обробку даних, отриманих зі сканованих документів, фотографій або файлів із зображеннями, перетворюючи текст з них у машино зчитуваний формат. Проаналізовано основні види програмних засобів для розпізнавання текстової інформації. Описано галузі практичного застосування розпізнавання текстової інформації.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для розпізнавання текстової інформації. Проте деякі програмні засоби для розпізнавання текстової інформації можуть не забезпечувати прийнятну точність розпізнавання символів, що обумовлює потребу у витратах додаткового часу для корекції отриманих результатів. Іншим недоліком є те, що інтеграція та використання деяких програм для розпізнавання ТІ може бути дорого вартісною, вартість професійного програмного забезпечення та його налаштування під конкретні потреби компанії може бути значною, особливо якщо йдеться про великі обсяги документів. Також існують проблеми з розпізнаванням спеціальних символів, що робить використання таких програм менш ефективним у певних галузях, таких як наука, технології або інженерія, де використовуються подібні символи. При обробці багатомовних документів програмне забезпечення для розпізнавання текстової інформації може не завжди коректно перемикатися між мовами, що призводить до помилок в обробці та зменшення якості отриманих результатів. Тому актуальною є розробка програмного забезпечення для розпізнавання текстової інформації.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання текстової інформації.

Для реалізації програмного забезпечення для розпізнавання текстової інформації обрано мову програмування C++, яка забезпечує високу продуктивність і ефективність використання ресурсів системи. Крім того, C++ має широкий набір бібліотек для обробки зображень і розпізнавання тексту, таких як OpenCV і Tesseract, що спрощує інтеграцію готових рішень і прискорює розробку програмного забезпечення для розпізнавання текстової інформації. Ці бібліотеки вже містять ефективні алгоритми для обробки графічних даних, що дозволяє фокусуватися на побудові конкретного рішення, а не на розробці базових алгоритмів з нуля.

Для створення програмного забезпечення для розпізнавання текстової інформації обрано середовище розробки Visual Studio, що пропонує потужний і зручний інтерфейс для розробників, який дозволяє ефективно управляти складними проектами, зокрема проектами для розпізнавання тексту. Крім того, Visual Studio має вбудовану підтримку C++, що є важливою перевагою для розробки продуктивного програмного забезпечення для розпізнавання тексту. Ще однією перевагою Visual Studio є можливість використання інтегрованої системи контролю версій, яка дозволяє легко керувати змінами в коді та співпрацювати з іншими розробниками.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для розпізнавання текстової інформації.

Практична цінність полягає у тому, що розроблено програмне забезпечення для розпізнавання текстової інформації.

Запропоновано структуру програмного забезпечення для розпізнавання текстової інформації. Визначено, що програмне забезпечення для розпізнавання текстової інформації складається з програмних модулів для

розпізнавання зображень, модулів для малювання символів та моделювання зашумлень, а також модулів для роботи з файлами зображень.

Описано функціонування програмного забезпечення для розпізнавання текстової інформації.

Розроблено базу даних програмного забезпечення для розпізнавання текстової інформації, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази даних обрано такі: користувачі (Users), інформація про зображення (ImagesInfo), інформація про програмні сесії (SessionsInfo).

Описано особливості реалізації програмного забезпечення для розпізнавання текстової інформації. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації.

Виконано тестування розробленого програмного забезпечення для розпізнавання текстової інформації. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління розпізнавання текстової інформації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. What is optical character recognition [Electronic resource]. – Access mode: <https://www.ibm.com/think/topics/optical-character-recognition>.
2. OCR meaning: why is OCR software important [Electronic resource]. – Access mode: <https://www.adobe.com/acrobat/guides/what-is-ocr.html>.
3. Optical character recognition [Electronic resource]. – Access mode: <https://www.techtarget.com/searchcontentmanagement/definition/OCR-optical-character-recognition>.
4. Optical Character Recognition – A Complete Guide [Electronic resource]. – Access mode: <https://revolveai.com/optical-character-recognition/>.
5. Optical Character Recognition – The 2024 Guide [Electronic resource]. – Access mode: <https://viso.ai/computer-vision/optical-character-recognition-ocr/>.
6. What is Optical Character Recognition [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/what-is-optical-character-recognition-ocr/>.
7. Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection [Electronic resource]. – Access mode: <https://onlinelibrary.wiley.com/doi/10.1002/eng2.12832>.
8. A History and Theory of Textual Event Detection and Recognition [Electronic resource]. – Access mode: <https://ieeexplore.ieee.org/document/9245483>.
9. Best OCR Software of 2024 [Electronic resource]. – Access mode: <https://www.adamenfroy.com/best-ocr-software>.
10. Best OCR Software [Electronic resource]. – Access mode: <https://www.g2.com/articles/best-ocr-software>.
11. Top-Rated OCR Software [Electronic resource]. – Access mode: <https://www.capterra.com/resources/top-rated-OCR-software/>.
12. CamScanner [Electronic resource]. – Access mode: <https://www.camscanner.com/>.
13. Rossum [Electronic resource]. – Access mode: <https://rosum.ai/>.

14. DocuPhase [Electronic resource]. – Access mode: <https://www.docuphase.com/>.
15. C++ Introduction [Electronic resource]. – Access mode: https://www.w3schools.com/cpp/cpp_intro.asp.
16. C++ Features [Electronic resource]. – Access mode: <https://www.javatpoint.com/cpp-features>.
17. OpenCV is the world's biggest computer vision library [Electronic resource]. – Access mode: <https://opencv.org/>.
18. Microsoft Visual Studio. Package consumption workflow [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/nuget/consume-packages/overview-and-workflow>.
19. Visual Studio: Community, Professional, or Enterprise [Electronic resource]. – Access mode: <https://pangea.ai/resources/visual-studio-community-professional-or-enterprise>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу розпізнавання текстових образів.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів розпізнавання текстової інформації», затверджене наказом Національного університету «Запорізька політехніка» № 398 від 07 жовтня 2024 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу розпізнавання текстової інформації.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підтримки процесу розпізнавання текстової інформації забезпечує виконання таких функцій:

- підтримка можливості авторизації користувачів;
- можливість завантаження графічного файлу, що містить текстову інформацію у вигляді зображення;
- можливість розпізнавання текстової інформації, поданої у вигляді зображення;
- підтримка можливості створення зображень символів для можливості їх подальшого розпізнавання;

- можливість виведення ймовірності віднесення символу, що розпізнається, до відповідних класів (символів);
- підтримка можливості автоматичного генерування символів за допомогою різних шрифтів;
- збереження результатів розпізнавання текстової інформації, поданої у вигляді зображення.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для розпізнавання текстової інформації повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для розпізнавання текстової інформації повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для розпізнавання текстової інформації необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення для розпізнавання текстової інформації необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 4 Гб і більше;

- 1–2 ГБ для встановлення програми та допоміжного програмного забезпечення;
- багатоядерний процесор з тактовою частотою 3,0 ГГц або вище;
- встановлена операційна система сімейства Windows.

A.3.6 Вимоги до маркування і пакування

Програма для підтримки процесу розпізнавання текстової інформації може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для розпізнавання текстової інформації».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для розпізнавання текстової інформації є інформація про користувача, файл із зображенням у вигляді тексту.

Вихідними даними програмного забезпечення для розпізнавання текстової інформації є тексти, що являють собою результати розпізнавання вхідних зображень.

ДОДАТОК Б
Фрагмент тексту програми

```

class CbtDlg : public CDlg
{
public:
    CbtDlg();
    entm { INGDD = INGDD_AB OUTB OX }; protected:
    virtual void DDatExchng(CDbtExchng* pDX); protected:
    DECLARE_MSG_MAP()
};
CbtDlg::CbtDlg() : CDlg(CbtDlg::INGDD)
{}
void CbtDlg::DDatExchng(CDbtExchng* pDX)
{
    CDlg::DDatExchng(pDX);
}
BEGIN_MSG_MAP(CbtDlg, CDlg) END_MSG_MAP()
CcrDlg CcrDlg::CcrDlg(CWnd* pPrhnt /*=NULL*/)
: CDlg(CcrDlg::INGDD, pPrhnt)
, gsChr(_T(MM))
, gNsLvl(0)
{
    gbPntChr=false;
    gchPn=0;
    gFtnt=0;
    rozmFtnt=20;
    ghIcon = AfxGetApp()-
    >LoadIcon(IDR_MAINFRAME);
}
CcrDlg::~CcrDlg()
{
    for(int i=0; i<rthhrs.Height(); i++) delete [] msChrs[i]; delete
    [] msChrs; if(gchPn) {
        delete gchPn; gchPn=0;
    }
    if(gFtnt) {
        delete gFtnt; gFtnt=0;
    }
}
void CcrDlg::DDatExchng(CDbtExchng* pDX)
{
    CDlg::DDatExchng(pDX); DXCntrl(pDX, KDICHARS, gstChars);
    DXCntrl(pDX, KDISPIN_TICNS, gspTicns);
    DXCntrl(pDX, KDIEDIT_TICNS, gedTicns);
    DXCntrl(pDX, KDSTATIC_FTNT, gstFtnt);
    DXCntrl(pDX, KDIRSLTS, gsrslts);
    DXCntrl(pDX, KDIANLZE_PICTURE, gstAnlzePic);
    DXTxt(pDX, KDIEDIT_CHAR, gsChr); DXCntrl(pDX, KDIPROGRESS_CHARS,
    gprthhrs);
}

```

```

DXTxt(pDX, KDINOISE_LEVEL_EDIT, gNsLvl);
}
BEGIN_MSG_MAP(CcrDlg, CDlg) OVWGSYSCOMMAND()
OVWGPAINT() OVWGQUERYDRAGICON() //}}AFX_MSG_MAP OVWGMOUSEMOVE()
OVWGLBUTTONDOWN() OVWGLBUTTONUP() OVBNklkm(KDIBTN_CLEAR,
&CcrDlg::OnBnClickedBtnClear)
OVBNklkm(KDIBTN_FTNT,
&CcrDlg::OnBnClickedBtnFtnt)
OVBNklkm(KDIBTN_ANLZ,
&CcrDlg::OnBnClickedBtnAnlz)
OVEN_CHANGE(KDIEDIT_CHAR,
&CcrDlg::OnEnChangeEditChar)
OVBNklkm(KDIADD_NOISE_BUTTON,
&CcrDlg::OnBnClickedAddNoiseButton) END_MSG_MAP()
CcrDlg::OnInitDialog()
CDlg::OnInitDialog();
ASSERT((IDGABX & 0xFFF0) == IDGABOUTB_OX);
ASSERT(IDGABX < 0xF000); CMn* pSMn = GetSystemMenu(FALSE); if
(pSMn != NULL)
{
CString strAbtMn;
strAbtMn.LoadString(IDS_ABX); if (!strAbtMn.IsEmpty())
{
pSMn->AppendMenu(MGFSEPARATOR);
pSMn->AppendMenu(MGFSTRING, IDGAB_OUTB_OX, strAbtMn);
}
}
StIcon(ghIcon, TRUE);
StIcon(ghIcon, FALSE);
gprthhrs.SetRange(0,NMBRCRS); gprthhrs.SetPos(0); srand(
(unsigned)time( NULL ) ); short Tics=2;
gspTics.SetRange32(1,20); gspTics.SetBuddy(&gedTics);
gspTics.SetPos32(Tics);
CRct rthh;
gstChars.GetClientRcht(&rthh);
msChrs = new int *[rthh.Height()]; for(int i=0; i<rthh.Height();
i++)
msChrs[i] = new int [rthh.Width()];
grisChars.CreateCompatibleBitmap(gstChar
s.GetDC(),rthh.Width(),rthh.Height());
gmmdcChrs.CreateCompatibleDC(gstChars
GetDC());
gmmdcChrs.SelectObject(&grisChars);
gchBrush.CreateSolidBrush(clWhite);
gmmdcChrs.FillRcht(&rthh,&gchBrush); if(gchPn) {
delete gchPn;
}
gchPn = new CPen(PS_SOLID, Tics, clBlack);

```

```

gmmdcChrs.SelectObject(gchPn); CRct
rAnlz(0,0,MATROZMX,MATROZMY);
gmemdcAnlz.SelectObject(&gbmpAnlz);
gmemdcAnlz.FillRcht(&rAnlz,&gchBrush
);
gmemdcAnlz.SelectObject(gchPn); return TRUE;
void CcrDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
if ((nID & 0xFFF0) == IDGAB OUTB OX)
{
CbtDlg dlgAbout; dlgAbout.DoModal();
}
else
{
CDlg::OnSysCommand(nID, lParam);
}
}
void CcrDlg::OnPaint()
{
if (IsIconic())
{
CPaintDC dc(this);
SendMsg(WGICONERASEBKGD,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);
int cxIcn = GetSstmMetrics(SGCXICN);
int cylcn = GetSstmMetrics(SGCYICN);
CRct rcht;
GetClientRcht(&rcht);
int x = (rcht.Width() - cxIcn + 1) / 2;
int y = (rcht.Height() - cylcn + 1) / 2;
dc.DrawIcon(x, y, ghIcon);
}
else
{
CDlg::OnPaint();
CPaintDC dc(&gstChars);
CRct rthh;
gstChars.GetCli entRcht(&rthh); dc.BitBlt( 0, 0, rthh.Width(),
rthh.Height(), &gmmdcChrs, 0, 0, SRCCOP );
CPaintDC dcAnlz(&gstAnlzePic);
gstAnlzePic.GetClientRcht(&rAnlz); dcAnlz.StretchBlt(0, 0,
rAnlz.Width(), rAnlz.Height(), &gmemdcAnlz, 0, 0, MATROZMX,
MATROZMY, SRCCOP);
}
}
HCURSR CcrDlg::OnQueryDragIcon()
{
return static_cast<HCURSR>(ghIcon);
}

```

```

}
void CcrDlg::OnMouseMove(UINT nFlgs, CPnt pnt)
{
if( gbPntChr && nFlgs&MK_LBUTTON )
CRct rthh;
gstChars.GetWindowRcht(&rthh);
ScreenToClient(&rthh);
if( rthh.PtlnRcht(pnt) ) {
gmmdcChrs.LineTo(pnt.x- rthh.left, pnt.y-rthh.top);
Invalidate();
UpdateWindow();
}
}
CDlg::OnMouseMove(nFlgs, pnt);
}
void CcrDlg::OnLButtonDown(UINT nFlgs, CPnt pnt)
{
if( nFlgs&MK_LBUTTON ) {
CRct rthh;
gstChars.GetW indowRcht(&rthh); ScreenToClient(&rthh);
if( rthh.PtlnRcht(pnt) ) { if(gchPn) {
delete gchPn;
}
int
thi ckhjnes=gspTicns.GetPos32();
gchPn = new CPen(PS_SOLID,
Ticns, clBlack);
gmmdcChrs.SelectObject(gchPn); gbPntChr=true;
gmmdcChrs.MoveTo(pnt.x- rthh.left, pnt.y-rthh.top);
}
}
CDlg::OnLButtonDown(nFlgs, pnt);
}
void CcrDlg::OnLButtonUp(UINT nFlgs, CPnt pnt)
if( gbPntChr ) {
gbPntChr=false;
}
CDlg::OnLButtonUp(nFlgs, pnt);
}
void CcrDlg::OnBnClickedBtnClear()
{
CRct rthh;
gstChars.GetClientRcht(&rthh);
gmmdcChrs.FillRcht(&rthh,&gchBrush);
Invalidate();
UpdateWindow();
}
}
void CcrDlg::OnBnClickedBtnChars()

```

```

{
void CcrDlg::OnBnClickedBtnFtnt()
{
CFtntDialog fdjlg;
if( fdjlg.DoModal() == IDOK )
{int rozm = fdjlg.GetRozm();
LOGFONT lf; fdjlg.GetCurrentFtnt(&lf); if(lf.lfFaceName[0] ) {
if(gFtnt) {
delete gFtnt;
}
gFtnt = new CFtnt(); if( gFtnt->CreateFtntIndircht(&lf) ) {
CRct rthh;
gstChars.GetClientRcht(&rthh);
gmmdcChrs.FillRcht(&rthh,&gchBrush);
rozxFtnt=rozX/10;
gstFtnt.SetWindowText(s);
gmmdcChrs.SelectObject(gF ont);
char ch=bgnChr;
gprthhrs.SetPos(0);
for(int i=0;
i<NMBRCRS; i++) {
s.Format("%c",ch); if(rozxFtnt<=48) {
gmmdcChrs.TextOut(20,10,s);
} else {
gmmdcChrs.TextOut(20,0,s);
}
FindChar(MasSimpleChars[i]);
gmmdcChrs.FillRcht(&rthh,&gchBrush);
ch++;
gprthhrs.SetPos(i+1);
}
gsChr=bgnChr;
UpdateDbt(FALSE);
} else {
delete gFtnt; gFtnt=0;}
}
}
}
bool CcrDlg::IsBlack(KLRTREF klrt)
{
return ( ((klrt>>23)&1) + ((klrt>>15)&1) + ((klrt>>7)&1) <= 1 );
}
void CcrDlg::CreateMatrix(BYTE Rslt[MATRROZMY] [MATRROZMX],
int xLft, int xRght, int yTp, int yBtm )
{
int i,j;
CRct rthh;
rthh=rthhrs;

```

```

if( (yBttm - yTp)*(xRght - xLft) == 0 ) {
xRght = -1; yTp = -1; yBttm = -1;
for(j=0; j<rthh.Height(); j++) {
for(i=0; i<rthh.Width(); i++) {
}
}
if(yTp == j) {
break;
}
}
for(j=rthh.Height()-1; j>=0; j--) {
for(i=0; i<rthh.Width(); i++) {
break;
}
}
if(yBttm == j) { break;
}
for(i=0; i<rthh.Width(); i++) {
for(j=0; j<rthh.Height(); j++) {
if(msChrs[j][i]) { xLft=i; break;
}
}
if(xLft == i) { break;
}
}
for(i=rthh.Width()-1; i>=0; i--) {
for(j=0; j<rthh.Height(); j++) {
if(msChrs[j][i]) { xRght=i; break;
}
}
if(xRght == i) { break;
}
}
if( (yBttm - yTp)*(xRght - xLft) == 0 )
}
int W = xRght - xLft + 1; int H = yBttm - yTp + 1;
for( j=yTp; j<=yBttm; j++ ) {
for( i=xLft; i<=xRght; i++ ) {
IsBlack(gmmdcChrs.GetPixel(i,j)) ) {
if(msChrs[j][i]) { nSmb1++;
}
}
}
}
double Prtcnt = (double)(nSmb1) / (H*W); Prtcnt *= 0.7;
int x[MATRROZMX+1];
for(i=0; i<=MATRROZMX; i++)
{x[i]=((W-1)*i+MATRROZMX/2)/MATRROZMX;
}
int y[MATRROZMY+1]; for(j=0; j<=MATRROZMY; j++)

```

```

{y[j]=((H-1)*j+MATROZMY/2)/MATROZMY;
}
MATROZMYxMATROZMX int khj, ki;
for(khj=0; khj<MATROZMY; khj++) {
for(ki=0; ki<MATROZMX; ki++) { nSmb1=0; for( j=yTp+y[khj];
j<=yTp+y[khj+1]; j++ ) {
for( i=xLft+x[ki];
i<=xLft+x[ki+1]; i++ ) {
IsBlack(gmmdcChrs.GetPixel(i,j)) ) {
if(msChrs[j][i]) { nSmb1++;
}
}
}
if( (double)(nSmb1) /      max(1,((x[ki+1] - x[ki] + 1) * (y[khj+1]
- y[khj] + 1))) > Prtcnt )
{
Rslt[khj][ki]=1;
} else {
Rslt[khj][ki]=0;
}
}
}
}
bool UDgrtr ( CAnlzeRslt &elem1, CAnlzeRslt &elem2 ) {
return elem1.Res > elem2.Res;
}
bool UDlssr ( CAnlzeRslt &elem1, CAnlzeRslt &elem2 )
{
return elem1.Res < elem2.Res;
}
MasChr double CcrDlg::CalcDstnc(int j0, int i0, BYTE
MasChr[MATROZMY][MATROZMX])
{
int i=0, j=0, i1, j1, sgn=1; int stht=0; int mhtb=1; int
roz=__min(MATROZMX,MATROZMY); double dst = roz*sqrt(2.0);
while( mhtb<=roz ) { switch(stht) { case 1:
break;
case 2:
j=-1; i=-1;
break;
}
j1=j0-j*mhtb;
i1=i0-i*mhtb;
if( j1>=0 && j1<MATROZMY && i1>=0 && i1<MATROZMX &&
MasChr[j1][i1] ) {
sgn = (i1<i0 || j1<j1) ? -1:1;
dst = sqrt(1.0*j*j + 1.0*i*i) * mhtb
sgn;

```

```

return dst;
stht++; if(stht > 6) { stht=1
}
}
return dst;      mhtb++;

}

void CcrDlg::GetMsChrs()
{
int ij;
for( j=0; j<rthhrs.Height(); j++ ) {
for( i=0; i<rthhrs.Width(); i++ ) {
if(
IsBlack(gmmdcChrs.GetPixel(i,j)) ) {
}
    msChrs[j][i]=1;
} else {
msChrs[j][i]=0;
}
}

void CcrDlg::FindChar(BYTE Rslt[MATRROZMY] [MATRROZMX])
{
int xLft, xRght, yTp, yBttm, i, j;
CRct rthh;
GetMsChrs();
xLft = -1;
for( j=0; j<rthh.Height(); j++ ) {
for( i=0; i<rthh.Width(); i++ ) {
nSmb1++;
}
}
double Prtcnt = 0.16; if(nSmb1<1/Prtcnt) { return;
}
MATRROZMYxMATRROZMX
yTp = -1; yBttm = -1;*/ int khj,ki;
(int)(sqrt(nSmb1*Prtcnt)+0.5); double prtc;
for(khj=0; khj<=rthh.Height()-ntm; khj++) {
for(ki=0; ki<=rthh.Width()-ntm; ki++) { nSmb1=0; for(j=0; j<ntm;
j++) {
for(i=0; i<ntm; i++) { if(
IsBlack(gmmdcChrs.GetPixel(ki+i,khj+j)) ) {
if(msChrs[khj+j][ki+i]) {
nSmb1++;
}
}
}
}
prtc=(double) (nSmb1)/(ntm*ntm);

```

```

if( prtc > Prtcnt ) {
yTp=khj;
break;
}
}
if(yTp == khj) {
yTp += (ntm+1)/2;
}
}
yTp += (int)(ntm*(1-prtc)+0.5); break;

for(khj=rthh.Height()-ntm; khj>=0; khj--) {
for(ki=0; ki<=rthh.Width()-ntm; ki++) { nSmb1=0; for(j=0; j<ntm;
j++) {
for(i=0; i<ntm; i++) {
if( IsBlack(gmmdcChrs.GetPixel(ki+i,khj+j)) ) {
if(msChrs[khj+j][ki+i]) {
nSmb1++;
}
}
}
prtc=(double)(nSmb1)/(ntm*ntm);
if( prtc > Prtcnt ) { yBttm=khj; break;
}
}
if(yBttm == khj) {
yBttm += (ntm+1)/2; yBttm += (int)(ntm*prtc+0.5); break;
}
}
for(khj=0; khj<=rthh.Height()-ntm; khj++) { nSmb1=0; for(j=0;
j<ntm; j++) {
for(i=0; i<ntm; i++) {
if( IsBlack(gmmdcChrs.GetPixel(ki+i,khj+j)) ) {
if(msChrs[khj+j][ki+i]) {
nSmb1++;
}
}
}
prtc=(double)(nSmb1)/(ntm*ntm);
if( prtc > Prtcnt ) { xLft=ki; break;
}
}
if(xLft == ki) {
xLft += (ntm+1)/2;
xLft += (int)(ntm*(1-prtc)+0.5);
break;
}
}
for(ki=rthh.Width()-ntm; ki>=0; ki--) {

```

```

for(khj=0; khj<=rthh.Height()-ntm; khj++) { nSmb1=0; for(j=0;
j<ntm; j++) {
for(i=0; i<ntm; i++) {if( IsBlack(gmmdcChrs.GetPixel(ki+i,khj+j))
) { if(msChrs[khj+j][ki+i]) {
nSmb1++;
}
}
}
prtc=(double) (nSmb1)/(ntm*ntm);
if( prtc > Prtcnt ) { xRght=ki; break;
}
}
if(xRght == ki) {
xRght += (ntm+1)/2; xRght += (int) (ntm*prtc+0.5); break;
}
}
CreateMatrix(Rslt, xLft, xRght, yTp, yBttm );
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів розпізнавання текстової інформації

Виконав: студент групи КНТ-213м
Керівник: к.т.н., доцент

Микола КУШНАРЕНКО
Матвій ІЛЬЯШЕНКО

Рисунок В.1 – Слайд 1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

- **Об'єкт дослідження** – процес розпізнавання образів.
- **Предмет дослідження** – програмні засоби для розпізнавання текстової інформації.
- **Метою роботи** є дослідження та розробка програмного забезпечення для розпізнавання текстової інформації.

2

Рисунок В.2 – Слайд 2

ЗАВДАННЯ РОБОТИ

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання текстової інформації;
- розробити модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації;
- здійснити проектування програмного забезпечення для розпізнавання текстової інформації;
- створити програмне забезпечення для розпізнавання текстової інформації;
- виконати тестування розробленого програмного забезпечення для розпізнавання текстової інформації.

3

Рисунок В.3 – Слайд 3

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Критерій порівняння	CamScanner	Rossum	DocuPhase
Підтримка великої кількості мов	+	+-	-
Використання засобів штучного інтелекту	+-	+	+-
Низька ціна підписки	-	+	+-
Ефективність роботи з великою кількістю документів	-	+	+-
Засоби інтеграції	+-	+	+-
Інтелектуальне обрізання та автоматичне покращення	+	+-	-

4

Рисунок В.4 – Слайд 4

ВИБІР МОВИ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	C++	Python	Java
Висока продуктивність	+	+-	+
Широкий набір бібліотек для обробки зображень	+	+	-
Можливість компіляції виконуваних файлів	+	-	+-
Зручність для створення ПЗ для розпізнавання текстової інформації	+	-	-
Ефективність використання ресурсів системи	+	+-	+-

5

Рисунок В.5 – Слайд 5

ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	Dev-C++	Eclipse
Зручність використання інтегрованої системи контролю версій	+	-	-
Великий набір плагінів і розширень	+	-	+
Потужний і зручний інтерфейс для розробників	+	+-	+-
Підтримка розробки на мові C++	+	+	+-
Засоби інтеграції з іншими інструментами	+	+-	+-

6

Рисунок В.6 – Слайд 6

МОДЕЛЬ ВЗАЄМОДІЇ КОРИСТУВАЧА З ПЗ ДЛЯ РОЗПІЗНАВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

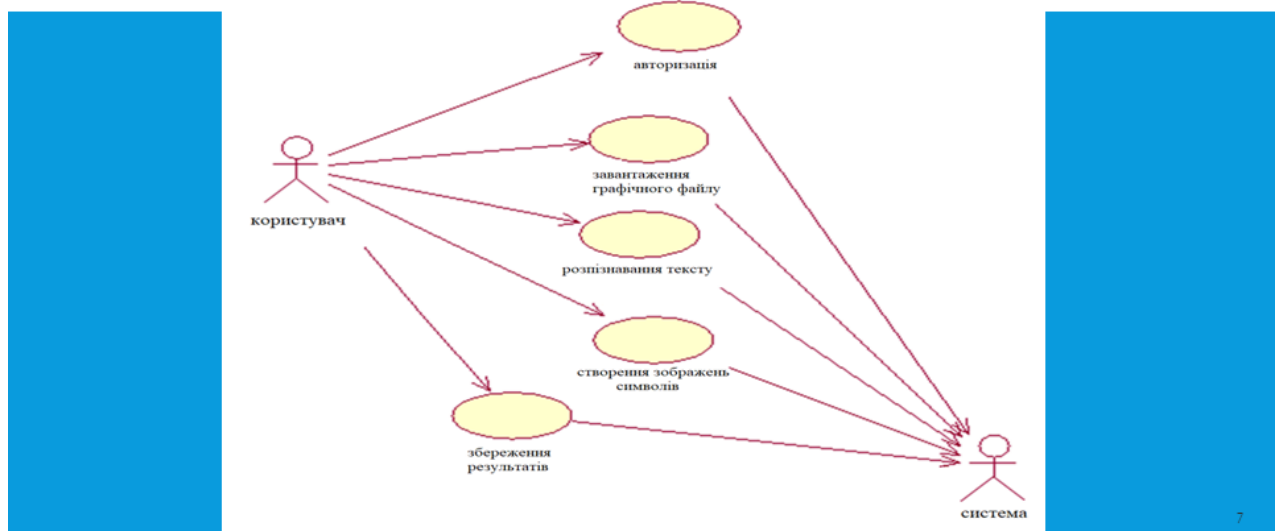


Рисунок В.7 – Слайд 7

СТРУКТУРА ПЗ ДЛЯ РОЗПІЗНАВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

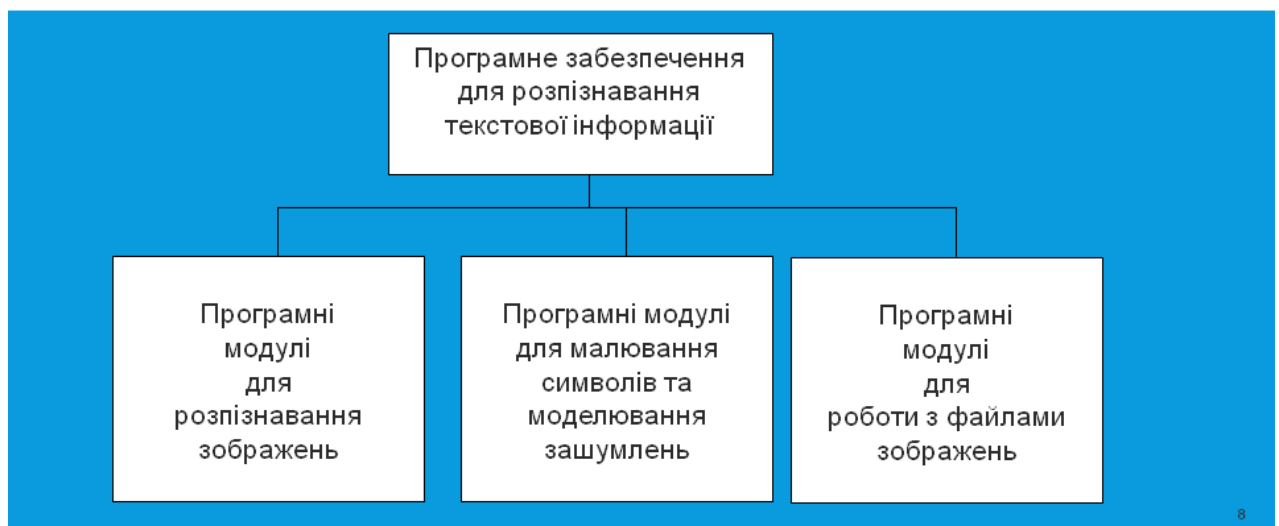


Рисунок В.8 – Слайд 8

СХЕМА ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

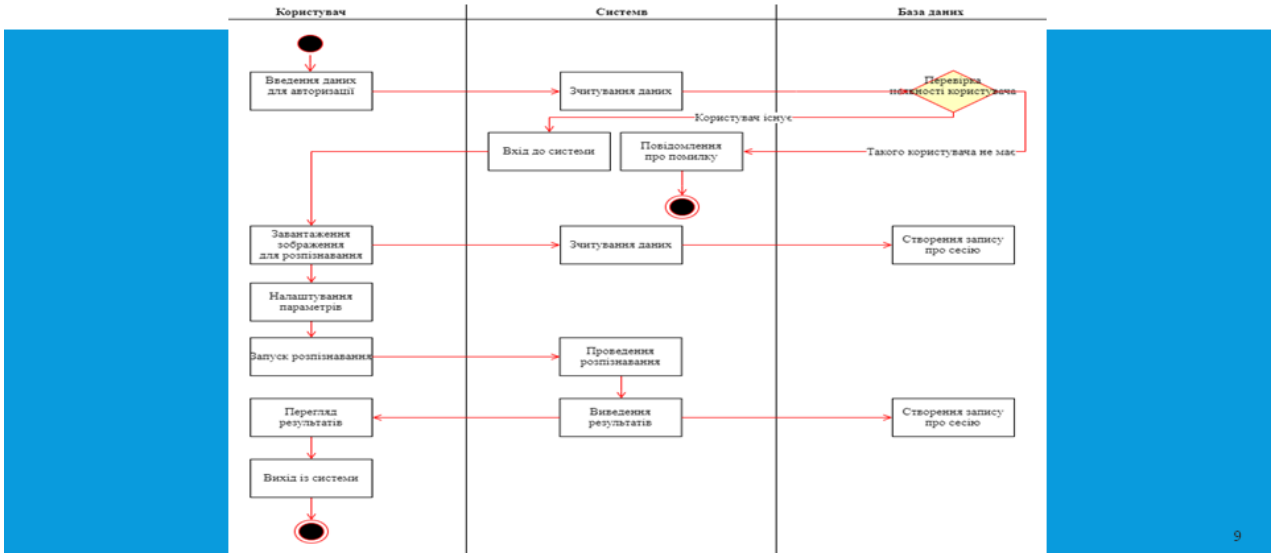


Рисунок В.9 – Слайд 9

СХЕМА БАЗИ ДАНИХ

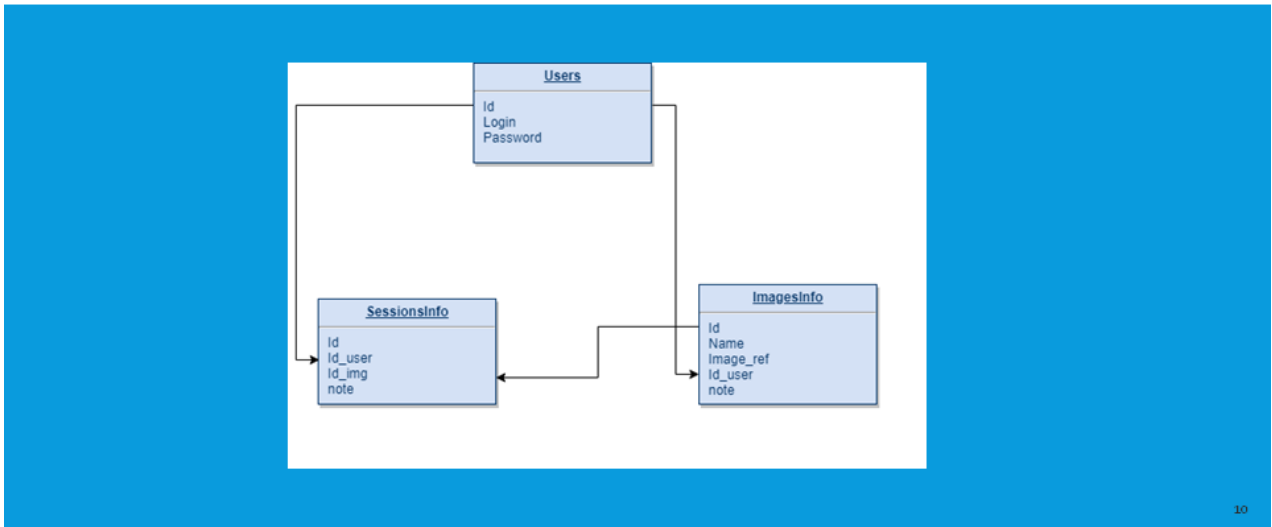


Рисунок В.10 – Слайд 10

РОБОТА З ПРОГРАМОЮ. РЕЄСТРАЦІЯ. ГОЛОВНА ФОРМА

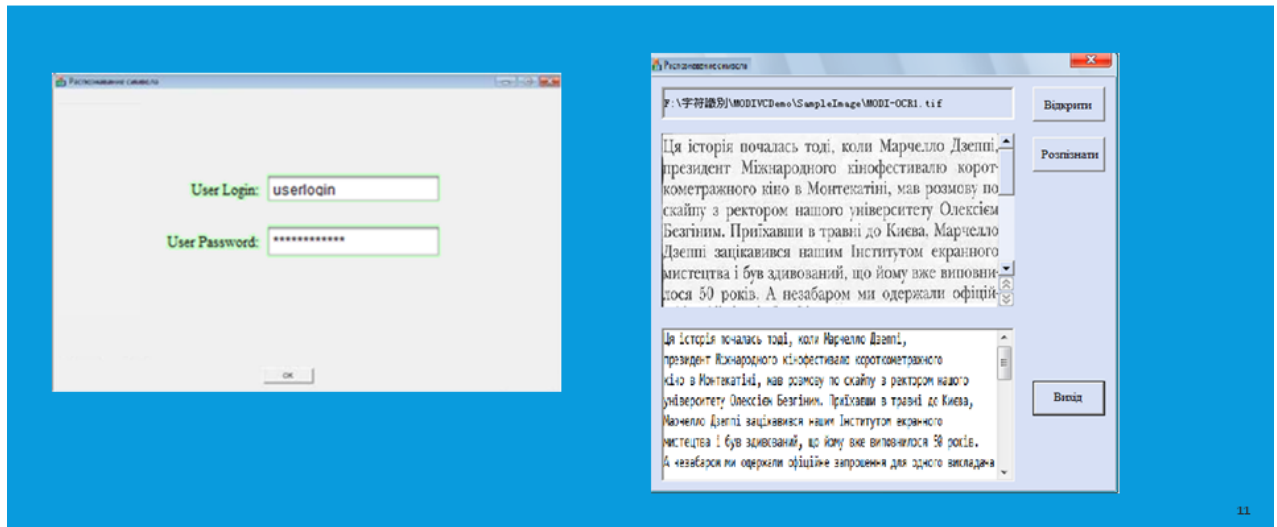


Рисунок В.11 – Слайд 11

РОБОТА З ПРОГРАМОЮ. ФОРМИ МАЛЮВАННЯ ТА ГЕНЕРУВАННЯ СИМВОЛІВ

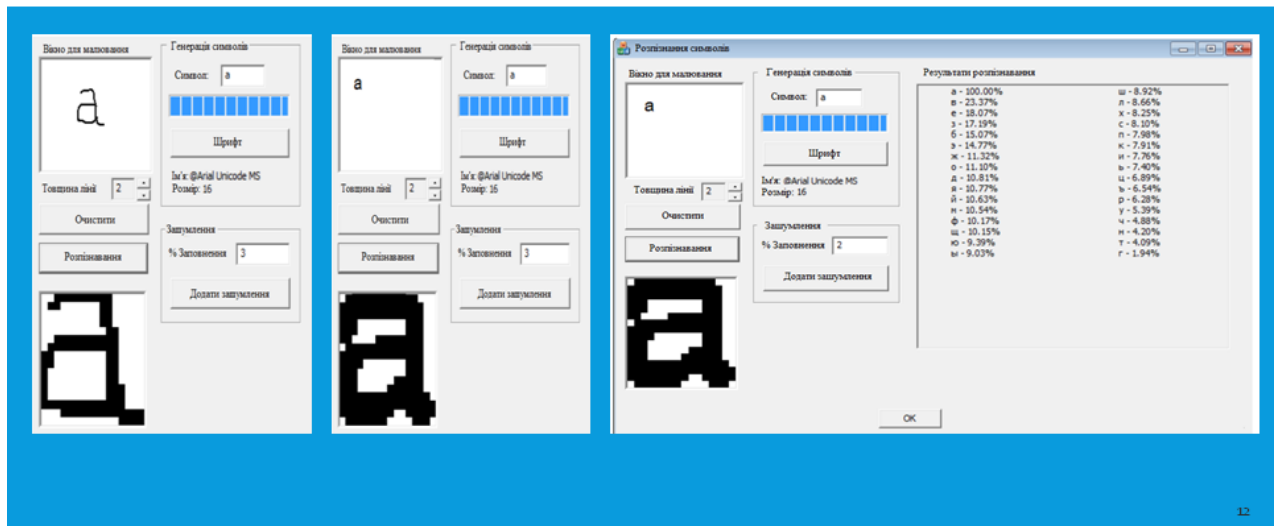


Рисунок В.12 – Слайд 12

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розпізнавання образів.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для розпізнавання текстової інформації, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для розпізнавання текстової інформації.

Практична цінність полягає у тому, що розроблено програмне забезпечення для розпізнавання текстової інформації.

Виконано тестування розробленого програмного забезпечення для розпізнавання текстової інформації.

13

Рисунок В.13 – Слайд 13