

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

(повне найменування інституту)

Кафедра обчислювальної техніки

(повна назва кафедри)

**Пояснювальна записка**  
до магістерської кваліфікаційної роботи

\_\_\_\_\_магістра\_\_\_\_\_

(освітньо-кваліфікаційний рівень)

на тему: «Програмний засіб для розпізнавання пошкоджених друкованих текстів із використанням багаторівневого аналізу документа»

Виконала: студент 2 курсу, групи 1КІ-19м\_\_\_\_\_

напряму підготовки (спеціальності)

123 – «Комп'ютерна інженерія»\_\_\_\_\_

(шифр і назва напряму підготовки, спеціальності)

\_\_\_\_\_Поліщук К. В.\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_д.т.н., проф. Мартинюк Т. Б.\_\_\_\_\_

(прізвище та ініціали)

Вінниця

2020 р.

## АНОТАЦІЯ

Магістерська робота присвячена розробці засобів розпізнавання пошкоджених друкованих текстів із використанням багаторівневого аналізу документа.

В магістерській роботі розглянуті методи розпізнавання тестових документів та послідовність багаторівневого розпізнавання символів тексту, запропоновано об'єднати згорткову нейронну мережу та шаблонний метод розпізнавання, що підвищує достовірність розпізнавання тексту, розроблені алгоритм та програма для розпізнавання текстових документів.

У роботі також виконані економічні розрахунки по обґрунтуванню доцільності розробки нового програмного продукту.

## ANNOTATION

The master's thesis is devoted to the development of tools for recognizing damaged printed texts using multi-level document analysis.

The master's thesis discusses the methods of recognition of test documents and the sequence of multilevel recognition of text characters, proposed to combine a convolutional neural network and a template recognition method that increases the reliability of text recognition, developed an algorithm and program for recognizing text documents.

The master's thesis also contains economic calculations to justify the feasibility of developing a new software product.

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ</b> .....	12
1.1 Розпізнавання символів тексту як задача розпізнавання образів .....	12
1.2 Структура систем оптичного розпізнавання тексту.....	16
1.3 Аналіз методів розпізнавання символів.....	22
1.4 Способи перевірки достовірності розпізнавання символів .....	27
1.5 Аналіз засобів розпізнавання текстових символів.....	30
1.6 Порівняння систем оптичного розпізнавання символів .....	39
<b>2 РОЗРОБКА ПОСЛІДОВНОСТІ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ</b> .....	42
2.1 Розробка послідовності розпізнавання тексту.....	42
2.2 Розробка способу розпізнавання пошкоджених символів.....	49
2.3 Розробка алгоритму розпізнавання символів .....	52
2.4 Згортова нейронна мережа.....	54
2.5 Використання бібліотек із відкритим кодом.....	58
<b>3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ</b> .....	63
3.1 Вибір інструментальних засобів програмування.....	63
3.2 Розробка архітектури програми розпізнавання .....	66
3.3 Розробка програми розпізнавання символів .....	68
3.4 Перевірка якості роботи програми розпізнавання.....	73

					<i>08-23.МКР.010.00.000 ПЗ</i>						
					Програмний засіб для розпізнавання пошкоджених друкованих текстів із використанням	<i>Лім.</i>		<i>Маса</i>		<i>Масштаб</i>	
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>							
<i>Розробив</i>		<i>Поліщук К. В.</i>									
<i>Керівник</i>		<i>Мартинюк Т. Б.</i>									
<i>Рецензент</i>		<i>Яремчук Ю. Є.</i>				<i>Арк.б</i>		<i>Аркушів</i>			
<i>Н. контроль</i>		<i>Швець С. І.</i>			<i>ВНТУ, гр. 1КІ-19м</i>						
<i>Затвердж</i>		<i>Азаров О. Д.</i>									



## ВСТУП

Розвиток засобів комп'ютерної обробки інформації та розробка одного із напрямків комп'ютерних технологій, а саме методів машинного розпізнавання образів, дозволяє розширити коло виконуваних комп'ютерами завдань та зробити машинну переробку інформації ще більш інтелектуальною. Прикладами сфер застосування розпізнавання образів можуть служити системи розпізнавання мови, розпізнавання тексту, машинний зір, розпізнавання відбитків пальців та інше. Хоча ці завдання вирішуються людиною на підсвідомому рівні із великою швидкістю та достовірністю, до теперішнього часу ще не створено таких комп'ютерних програм, які вирішують їх настільки ж ефективно у загальному вигляді [1-3].

Методи автоматичного розпізнавання образів та їх реалізація у системах оптичного читання текстів (OpticalCharacterRecognition, OCR-системах) є однією із найбільш яскравих та дієвих технологій штучного інтелекту. Під системою OCR розуміється автоматичне розпізнавання при допомозі спеціально створених програм зображень символів друкованого або рукописного тексту та переведення його у формат, що придатний для подальшої обробки редакторами текстів, текстовими процесорами і т. п. Іноді також під OCR розуміють пристрій автоматичного читання тексту або оптичного розпізнавання символів. На теперішній час такі пристрої при використанні їх у різних сферах промисловості обробляють до 100 тисяч документів на добу. Промислове використання цих пристроїв передбачає оброблення документів хорошої та середньої якості, це можуть бути бланки статистичного обліку, бланки податкових декларацій, різного виду банківські рахунки і т. п. [4].

Завдання переведення інформації із паперових на електронні носії є актуальною не тільки у рамках потреб, що виникають у різних системах документообігу. Сучасні інформаційні технології дозволяють суттєво спростити доступ до різних інформаційних ресурсів, що накопичені людством, за виконання умови, що вони будуть переведені у більш зручний електронний

вигляд. Найбільш швидким та простим є виконання операції сканування документів за допомогою сканерів. У результаті роботи отримуємо цифрове зображення документа у вигляді графічного файлу. Але більш кращим, у порівнянні із графічним представленням, є текстове подання інформації. Цей варіант дозволяє істотно зменшити витрати на зберігання та передачу інформації, та додатково дозволяє реалізувати усі можливі сценарії використання та аналізу електронних документів. Тому з практичної точки зору найбільший інтерес представляє саме переведення паперових носіїв у текстовий електронний документ.

Так, наприклад, одна із відомих програм оптичного розпізнавання FineReader дозволяє обробляти друкований текст із незначною кількістю помилок, для ліквідації яких в подальшому вимагається втручання людини та проведення корекції отриманого тексту[5]. Тим не менш, створення кожного нового програмного додатку у цій області, як і раніше, залишається творчим завданням та вимагає проведення додаткових досліджень у зв'язку зі специфічними вимогами по вирішенню задач надійності розпізнавання, швидкодії та обсягом пам'яті, якими характеризується кожне конкретне завдання.

Ще складнішим стає це завдання при виконанні автоматизованого розпізнавання пошкоджених друкованих текстів, яке в даний час поки ще не вирішено у повній мірі та є однією з найбільш актуальних та складних задач розпізнавання даних.

Дослідження по пошуку та розпізнаванню символів пошкоджених друкованих текстів характеризуються великим розмаїттям у виборі засобів для їх вирішення. Традиційними завданнями цього процесу є задачі виявлення та розпізнавання символів, що мають відому форму із урахуванням впливу завад та спотворенням отриманих цифрових зображень текстових документів для їх подальшого оброблення [6].

Значний обсяг даних, що використовується для розпізнавання друкованих текстових символів, вимагає створення нових та подальшого

вдосконалення існуючих підходів до оброблення отриманих цифрових зображень текстових документів із метою виділення та у кінцевому етапі розпізнавання друкованих текстових символів. Виходячи із розглянутого, завдання подальшого вдосконалення методів виділення та розпізнавання текстових символів, що є кращими за критеріями достовірність, швидкість та ефективність, є актуальною задачею. Значна сфера застосування методів оброблення зображень текстових документів для розпізнавання друкованих символів вказує на **актуальність даної** проблеми та вимагає подальшого дослідження та вдосконалення.

**Мета дослідження** магістерської роботи — вдосконалення методів виділення та розпізнавання пошкоджених символів друкованих текстів із використанням багаторівневого аналізу документа.

**Задачі дослідження** магістерської роботи:

- здійснити аналіз способів побудови систем розпізнавання друкованих текстів;
- запропонувати кращий підхід для виділення та формування ознак друкованих символів для їх подальшого розпізнавання;
- створити алгоритм та програму обробки зображення для виділення та розпізнавання друкованих пошкоджених текстів;
- виконати обґрунтування доцільності виконання нового наукового рішення, розрахувати економічні витрати для створення програмних засобів розпізнавання текстових документів із багаторівневого аналізу та визначити переваги від впровадження нового програмного продукту.

**Об'єкт дослідження** магістерської роботи — процес отримання даних про текстовий документ шляхом знаходження у виділеній сцені його зображення та розпізнавання.

**Предмет дослідження** магістерської роботи — методи оброблення цифрового зображення для виділення та розпізнавання символів у пошкоджених друкованих текстах.

**Методи дослідження** магістерської роботи: використовувались методи



теорії множин для формування множини ознак для розпізнавання, методи диференційного числення для виділенню граничних контурів, методи математичної статистики для виконання аналізу отриманих результатів розпізнавання. У роботі використано принципи об'єктно-орієнтованого програмування для реалізації запропонованого підходу.

**Наукова новизна** отриманих результатів магістерської роботи полягає у тому, що удосконалено метод оброблення зображення текстового документу для виділення та розпізнавання символів тексту, який відрізняється від відомих підходів виконанням обчислення коефіцієнтів збігів символу зображення і його фону та їх порівняння із еталонними зображеннями, що дозволяє більш ефективно розпізнавати символи у отриманому пошкодженому друкованому тексті.

**Практичне значення** одержаних результатів магістерської роботи:

- створено алгоритм оброблення цифрових зображень для виділення та розпізнавання пошкоджених символів друкованих текстів;
- розроблено програму оброблення зображення для виділення символів та їх розпізнавання у пошкоджених друкованих текстах.

**Апробація** результатів магістерської роботи — зроблено доповідь на молодіжній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2019)».

По результатам роботи зроблені **публікації** в [7, 8].

# 1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ

У системах обробки та розпізнавання текстових документів виділення символів тексту та їх ознак є частиною загального процесу розпізнавання образів. Даний розділ магістерської роботи присвячено проведенню аналізу методів розпізнавання текстових документів у отриманому зображенні для виділення текстових символів та розглянуто засоби, які використовуються для виконання операцій по розпізнаванню тексту у цифрових зображеннях.

## 1.1 Розпізнавання символів тексту як задача розпізнавання образів

Розпізнавання тексту є одним з напрямків розпізнавання образів. В даному випадку під розпізнаванням розуміється співвіднесення зображення об'єкта (його образу) набору ознак самого об'єкта. Прикладами розпізнавання тексту є:

— виконання процесу переведення зображення тексту в цифрову форму (скановані книги, статті, журнали) для подальшої роботи з його цифровим аналогом;

- обробка анкетних бланків;
- обробка даних систем тестування знань;
- розпізнавання номерів машин.

Оптичне розпізнавання символів — це процес, який реалізує переведення зображення машинописного, друкованого або ж рукописного тексту в набір текстових даних, що представлені у електронному вигляді [9]. Якість виконання даного процесу розпізнавання символів залежить від стану початкового представлення даних, тобто зображення, що аналізується. Стан зображення текстового документу характеризується наявністю або відсутністю різного типу дефектів та відхилень при друку та переведенні документа у цифрову форму, що призводить до подальших складнощів при виконанні процесу розпізнавання.

Процес оптичного розпізнавання символів складається з етапів:

сприйняття, попередня обробка, сегментації і власне розпізнавання. На кожному з етапів вирішується проблема виявлення наявності певних спотворень та їх зменшення. Аналіз ефективності методів проводиться за наступним принципом: на вхід аналізованого методу подаються набори вихідних даних, після перетворення даних відбувається оцінка відповідності результату з еталонним значенням на підставі вибраних певних заздалегідь критеріїв.

Більшість сучасних OCR-систем проводить аналіз документа відповідно до одним з наступних принципів: top-down («зверху вниз») або bottom-up («знизу вгору»). Наприклад, аналіз і виділення об'єктів в документі відбувається зверху вниз, а генерація його електронної версії від низу до верху.

У даній роботі приділяється увага процесу розпізнавання символів і розгляду найбільш поширених методів, які використовуються при створенні засобів розпізнавання. На даний момент виділяють три основних підходи для вирішення задачі розпізнавання: шаблонний, структурний і ознаковий [10].

Одним із перших і досить широко застосованих є шаблонні методи розпізнавання символів. При використанні даної групи методів відскановане зображення переводиться в растрове (тобто зображення із точок), потім проводиться його порівняння з еталонними шаблонами, що відбулися сформовані в базі даних. Критерієм вибору шаблону є найменша кількість точок, відмінних від досліджуваного зображення. Шаблон для кожного класу отримують шляхом усереднення зображення символів навчальної вибірки. Перевагою шаблонних методів є досить висока точність розпізнавання символів, які мають різні види дефектів (від розривів у накреслені символів до різного виду склеювань). Основним недоліком даного виду розпізнавання є залежність від шрифту, який зустрічається в аналізованому зображенні. Шрифт повинен бути відомий завчасно, інакше буде неможливо вірно розпізнати досліджуваний символ із отриманого зображення. При існуючому розмаїтті та багатстві друкованої продукції в процесі проведення навчання практично неможливо охопити всі відомі шрифти та їх модифікації, а тому цей

фактор значно обмежує універсальність шаблонних методів розпізнавання символів.

Ще одна група методів — це структурні методи розпізнавання. Дана група методів використовує інформацію не про точкове написання символу, а про його топологічне представлення: до уваги береться взаємне розташування окремих складових частин аналізованого символу [9]. До основних переваг структурних методів розпізнавання символів можна віднести інваріантність по відношенню до типів та розмірів шрифтів [10]. Основною проблемою топологічних методів є ідентифікація виділених символів, що мають деякі дефекти (наприклад, розрив лінії або злиття сусідніх ліній), а також невисока швидкодія при роботі [6, 10].

Інша група методів відноситься до ознакових методів. Методи розпізнавання символів даної групи найбільш поширені при обробці текстових документів. Аналіз в них базується на тому, що зображенню символу ставиться у відповідність певний  $N$ -мірний вектор ознак. Процес розпізнавання полягає в послідовному порівнянні отриманого вектора із набором еталонних векторів тієї ж розмірності у створеній базі еталонних символів. Якість виконання процесу розпізнавання символів залежить від сформованих типів ознак та вибраної їх кількості. Формування вектора проводиться при проведенні аналізу підготовленого зображення, еталон для кожного класу отримують аналогічним способом шляхом оброблення символів із навчальної вибірки.

Основними перевагами даної групи розпізнавання символів є значна простота реалізації, достатня стійкість до змін форми символів, хороша узагальнююча здатність, досить висока швидкодія. З недоліків зазначених ознакових методів варто виділити нестійкість до можливих дефектів отриманого зображення, втрата частини інформації про символ на етапі виділення та отримання ознак, відсутність чітких та однозначно сформованих правил про процес формування ознак. Крім того, ознакові методи розпізнавання мають інший серйозний недолік - на етапі виділення та

формування ознак відбувається незворотна втрата частини інформації про контрольовані символи. Процес вилучення ознак ведеться незалежно від місця розташування символів, тому інформація про взаємне розташування елементів конкретного символу втрачається.

На ряду із цими методами існують також альтернативні методи, які не потребують попередньої сегментації, такі як ієрархічні приховані моделі Маркова та згорткові нейронні мережі. У зв'язку з виникненням нової хвилі популярності нейромережевих класифікаторів вони стали частіше використовуватися в дослідженнях по розпізнаванню тексту [11,12]. Основною перевагою використання нейромережевих технологій є їх хороша узагальнююча здатність, можливість використовувати контекстний аналіз та розпізнавати символ, ґрунтуючись на навколишніх його символах.

На сьогоднішній день системи автоматичного розпізнавання не досягають тих показників, які притаманні людському сприйняттю тексту, однак комбінування різних методів покращують ці показники. Наприклад, в OCR-системах АBBYY задіяні шість типів класифікаторів: растровий, контурний, просторових ознак, структурний, ознаковий диференціальний та структурно-диференційний, які застосовуються залежно від отриманих вхідних параметрів зображення. Точність розпізнавання цієї системи наближається приблизно до 96% [5].

Є ряд істотних проблем, пов'язаних із розпізнаванням рукописних та друкованих символів. Найбільш важливі з них такі:

- варіації розмірів і використаних масштабів та форм накреслення символів, різних шрифтів;
- різноманітність форм накреслення символів;
- спотворення елементів зображення символів.

Крім того, кожен окремий символ може бути написаний різними стандартними шрифтами, наприклад (Times, Courier, Gothic, Orator, Elite), а також значною кількістю різних нестандартних шрифтів, що використовуються в різних предметних областях. При цьому різні у написанні

символи можуть мати подібні обриси. Наприклад, «S» і «5», «U» і «V», «G» і «6», «Z» та «2».

Спотворення цифрових зображень текстових символів можуть бути викликані:

- спотвореннями при друку, в зокрема, неякісним друком, «злипанням» сусідніх символів, плямами і помилковими точками на тлі поблизу символів, у зміщенні символів відносно один одного, у розривах або зміні форми між частинами одного і того ж символу і т. п.;

- ефектами освітлення (тіні, відблиски і т. п.) при зйомці відеокамерою та різного типу оптичних спотвореннях, в тому числі неоднорідності освітлення аналізованого документу;

- зміною нахилу написання символів;

- зміщенням символів або ж їх частин щодо їх очікуваного положення в рядку;

- спотворенням форми символу за рахунок переведення зображення у цифрову форму із використанням так званої «грубої» дискретизації;

- появі різного виду помилкових знаків.

Отже, у підсумку, система оптичного розпізнавання тексту повинна виділяти на цифровому зображенні текстові області, виділяти в них окремі рядки, потім виділяти окремі символи, розпізнавати ці символи і при цьому бути нечутливою (стійкою) по відношенню до способу верстки, відстані між рядками і іншими параметрами друку.

## 1.2 Структура систем оптичного розпізнавання тексту

Системи розпізнавання текстових документів, що були отримані різними способами, мають деяку певну послідовність виконання своєї роботи. Ця послідовність розпізнавання текстових символів у отриманому документі складається із таких етапів.

Першим етапом є пошук та виділення областей у отриманому документі, які мають текст, та вилучення різного роду малюнків та інших елементів, що

не є текстовими. Для вирішення цієї задачі використовують різного типу згорткові нейронні мережі, детектор Віоли-Джонса, методи класифікації на основі аналізу текстурних областей, признаки Тамура, різного типу детектори [13, 14]. Однак сучасні системи оптичного розпізнавання текстів OCR досить складно працюють для пошуку та виділення тексту у довільного типу зображеннях, так як розроблялися для роботи із однорідним структурованим текстом, які на фотографіях та відеозаписах реальних сцен зустрічаються досить рідко.

Наступним є етап покращення якості виділеного текстового фрагменту та виконання переведення його у бінаризовану форму представлення. На цьому етапі відбувається операція попереднього згладження зображення, визначення типів наявних завад та їх видалення. Потім визначається структура виділеного фрагменту тексту та порядок його читання.

Надалі здійснюється етап по виконанню операцій сегментації текстового документа послідовно спочатку на рядки, потім на слова та завершується виділенням окремих символів. Пошук рядків базується на виявленні періодичності та регулярності текстових областей та виконується на основі методу Хафа, методу зв'язаних компонент, аналізу горизонтальних, вертикальних та діагональних гістограм. Тепер для кожного символу застосовуються свій спосіб опису та виконується найбільш відповідальний етап його розпізнавання. Для розпізнавання символів реалізуються різні класифікатори, одними із яких є нейронні мережі, а теперішній час найчастіше використовуються згорткові нейронні мережі. Заключним етапом у багатьох системах розпізнавання є перевірка отриманих слів із наявним словником. Словарна перевірка виконується на основі стандартних або динамічно створених мовних словників, n-грам, що реалізовані у вигляді списків, дерев та графів [9, 15].

Сучасні системи оптичного розпізнавання тексту OCR складаються з таких основних блоків, які передбачають апаратну або ж програмну їх реалізацію [4]:

- блок сегментації (локалізації та виділення) елементів тексту;
- блок попередньої обробки зображення;
- блок виділення ознак;
- блок розпізнавання символів;
- блок постобробки результатів розпізнавання.

Ці блоки системи розпізнавання відповідають послідовним крокам обробки та аналізу зображень, що виконуються послідовно.

Спочатку здійснюється виділення текстових областей, рядків і розбиття зв'язаних текстових рядків на окремі знакомісця, кожне з яких відповідає одному текстовому символу.

Після розбиття (а іноді до або в процесі розбиття) символи, представлені у вигляді двовимірних матриць пікселів, піддаються згладжуванню, фільтрації з метою усунення завад, нормалізації розміру, а також іншим перетворенням з метою виділення утворюючих їх елементів або ж чисельних ознак, що використовуються згодом для їх розпізнавання.

Розпізнавання символів відбувається в процесі порівняння виділених характерних ознак з еталонними наборами та структурами ознак, які формувались і запам'ятовуються в процесі навчання системи на еталонних і або ж реальних прикладах текстових символів [16].

На завершальному етапі смислова або контекстна інформація може бути використана як для вирішення тих невизначеностей, що виникають при розпізнаванні окремих символів, що мають ідентичні розміри, так і для коригування помилково зчитаних слів і навіть фраз в цілому.

При надходженні зображення на розпізнавання можуть бути різні початкові умови: шуми, неправильна яскравість отриманого зображення, розмиття і інше. Саме тому попередня обробка є важливим етапом в процесі розпізнавання символів і дозволяє виробляти згладжування, нормалізацію, сегментацію і апроксимацію відрізків ліній.

Під згладжуванням в даному випадку розуміється велика група процедур обробки зображень. Зокрема, широко використовуються



морфологічні оператори заповнення і стоншення. Заповнення усуває невеликі розриви і прогалини. Потоншення представляє собою процес зменшення товщини лінії, в якій на кожному кроці області розміром в кілька пікселів ставиться у відповідність тільки один піксель «потоншеної лінії». Морфологічний спосіб реалізації подібних операцій на базі операторів розширення і стиснення Серра описаний в [17].

Також існує та описаний спеціальний алгоритм бінарної фільтрації зображень текстових символів, який отримав назву стирання бахроми [18]. Під «бахромою» тут розуміються нерівності кордонів символу, які заважають, по-перше, правильно визначити його розміри, а по-друге, спотворюють сам образ символу та заважають його подальшому розпізнаванню за контурною ознакою.

В технології OCR використовується і алгоритм бінарної фільтрації як «стирання бахроми». Його суть полягає в послідовному стирання крайніх елементів (наприклад, крайнього верхнього, нижнього, лівого і правого пікселів) на нерівностях біля кордонів символу, які найчастіше заважають точному їх визначенню, розмірів символу та подальшого його розпізнаванню за контурною ознакою. Якщо в якості апертури фільтра обрана околиця порядку розміром  $3 \times 3$  пікселя, то під крайнім верхнім пікселем розуміється такий піксель, в апертурі розміром  $3 \times 3$  пікселя якого спостерігаються такі поєднання:

$(000010100)$ ,  $(000010010)$ ,  $(000010001)$ ,  $(000010110)$ ,  $(000010011)$ ,  $(000010111)$ ,

де 0 кодує елемент фону, а 1 відповідно кодує елемент символу.

За допомогою повороту перерахованих матриць на  $90^\circ$ ,  $180^\circ$  та  $270^\circ$  знаходять додатково визначення відповідно крайнього лівого, крайнього нижнього і крайнього правого пікселів. При фільтрації з метою стирання бахроми всі вибрані таким чином крайні пікселі стираються. Стираються також ізольовані пікселі, які не мають сусідніх пікселів символу в апертурі  $3 \times 3$ . Решта пікселів, які не є крайніми або ізольованими, переносяться на

відфільтроване зображення без зміни.

За аналогією з описаним стиранням бахроми з одиниць можна ввести стирання бахроми з нулів. При цьому «крайові нулі», аперттури яких відповідають представленим вище матрицями із заміною одиниць на нулі та нулів на одиниці, також «стираються», тобто заміщуються одиницями.

Алгоритм комбінованого стирання бахроми одночасно оперує одиницями по символу і нулями по фону. Після використання даного алгоритму значно збільшується ймовірність точного розпізнавання символів. Для неправильно орієнтованих зображень символів застосовується геометрична нормалізація, що відповідає за усунення нахилів, перекосів окремих символів і за коригування їх довжини і ширини.

Геометрична нормалізація зображень документів має на увазі використання алгоритмів, що усувають нахили та перекози окремих символів, слів або ж рядків, а також включає в себе процедури, які здійснюють нормалізацію символів по висоті та ширині після відповідної їх попередньої обробки.

За допомогою заповнення можна усунути невеликі розриви та прогалини. Потоншення являє собою процес зменшення товщини лінії, в якій на кожному кроці області розміром в декілька пікселів ставиться у відповідність тільки один піксель «потоншеної лінії» [17].

Процедури сегментації здійснюють розбиття зображення документа на окремі області. Як правило, перш за все необхідно відокремити друкований текст від графіки і рукописних позначок. Далі більшість алгоритмів оптичного розпізнавання поділяють текст на символи і розпізнають їх окремо. Це просте рішення дійсно найбільш ефективно, якщо тільки символи тексту не перекривають один одного. Злиття символів може бути викликано типом шрифту, яким був набраний текст, поганим дозволом друкувального пристрою або високим рівнем яскравості, обраним для відновлення розірваних символів.

Додаткове розбиття текстових областей і рядків на слова доцільно в тому випадку, якщо слово є таким об'єктом, відповідно до якого виконується

розпізнавання тексту. Подібний підхід, при якому одиницею розпізнавання є не окремий символ, а ціле слово, складно реалізувати через велику кількість елементів, що підлягають запам'ятовуванню та подальшому розпізнаванню, але він може бути корисним та досить ефективним у окремих конкретних випадках, коли набір слів в кодовому словнику істотно обмежений за умовами завдання.

Під апроксимацією відрізків ліній розуміють складання графа опису символу у вигляді набору вершин і прямих ребер, які безпосередньо апроксимують ланцюжки пікселів вихідного зображення. Дана апроксимація здійснюється для зменшення обсягу даних і може використовуватися при розпізнаванні, заснованому на виділенні ознак, що описують геометрію і топологію зображення.

Однією з самих складних завдань в розпізнаванні образів є виділення ознак для кожного з них. На даний момент існує велика кількість систем ознак. Проблема полягає у виділенні саме тих ознак, які дозволять ефективно відрізнити один клас символів від всіх інших в даній конкретній задачі. Так, метод порівняння образу з еталоном вважається досить ефективним в розпізнаванні символів. В цьому випадку визначається ступінь подібності між образом та кожним із наявних еталонів. Класифікація текстового зображення символу відбувається за методом найближчого сусіда. Проте у даного кореляційного методу є суттєвий недолік: будь-які спотворення на отриманому зображенні символу можуть перешкодити грамотному розпізнаванню цього символу, тому тоді застосовуються інші, спеціальні способи порівняння образів.

Вважається, що виділення ознак є однією з найбільш важких і важливих завдань в розпізнаванні образів. Для розпізнавання символів може бути використано велику кількість різних систем ознак. Проблема полягає в тому, щоб виділити саме ті ознаки, які дозволять ефективно відрізнити один клас символів від усіх інших в даній конкретній завданню.

### 1.3 Аналіз методів розпізнавання символів

Співставлення зображення та шаблону. Ця група методів заснована на виконанні операції безпосереднього порівняння зображень отриманого тестового і наявного еталонного символів. При цьому обчислюється ступінь подібності між отриманим образом та кожним із еталонів. Класифікація отриманого тестового зображення символу відбувається за методом найближчого сусіда. Існують також інші методи порівняння текстових зображень, а саме — методи кореляції та узгодженої фільтрації зображень.

З практичної точки зору ці методи легко реалізувати, і багато комерційних систем OCR використовують саме їх. Однак при «лобовій» реалізації кореляційних методів навіть невелика темна плямка, що потрапила на зовнішній контур символу, може істотно вплинути на результат розпізнавання. Тому для досягнення хорошої якості розпізнавання в системах, що використовують зіставлення шаблонів, застосовуються інші, спеціальні способи порівняння зображень.

Одна з основних модифікацій алгоритму порівняння шаблонів використовує представлення шаблонів у вигляді набору певних логічних правил.

Наприклад, символ, що поданий у вигляді

0000000000

000aabb000

00aeffb00

0ae0000fb0

0ae0ii0fb0

0ae0ii0fb0

0ae0000fb0

0cg0000hd0

0cg0jj0hd0

0cg0jj0hd0

0cg0000hd0

00cgghhd00

000ccdd000

0000000000

може бути розпізнаний як число «нуль», якщо виконуються такі умови: (не менше 5 символів «а» є логічною «1» або ж не менше 4 символів «е» приймають значення логічної «1») та (не менше 4 символів «h» є логічною «1» або ж не менше 5 символів «d» є логічною «1») та (не менше 5 символів «с» є логічною «1» або ж не менше 4 символів «g» є логічною «1») та (не менше 4 символів «f» є логічною «1» або ж не менше 5 символів «b» є логічною «1») та (принаймні 3 символів «j» є логічним «0») та (принаймні 3 символів «i» є логічним «0») [18].

У групі ознакових методів статистичні характеристики ознак формуються на основі аналізу різних за статистикою розподілів. Найбільш відомі методики цієї групи використовують обчислення моментів та підрахунок перетинів [19].

Моменти різних порядків з успіхом використовуються в самих різних областях машинного зору як дескриптори форми виділених областей та об'єктів. У разі розпізнавання текстових символів в якості набору ознак використовують значення моментів сукупності «чорних» точок щодо деякого обраного центру. Найбільш загальноживаними в додатках такого роду є порядковий, центральний і нормований моменти.

Для цифрового зображення, що зберігається в двовимірному масиві, порядковий момент (1.1) є функціями координат кожної точки зображення наступного виду:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y), \quad (1.1)$$

де  $p, q \in \{0, 1, \dots, \infty\}$ ;

$M$  та  $N$  є розмірами зображення по горизонталі та вертикалі та  $f(x, y)$  є

яскравістю пікселя у точці із координатами  $(x, y)$  на зображенні [20].

Центральні моменти є функцією відстані точки від центру ваги символу:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - x_0)^p (y - y_0)^q f(x, y), \quad (1.2)$$

де відповідно  $x_0$  та  $y_0$  є координатами центру ваги.

Нормовані центральні моменти виходять в результаті ділення центральних моментів (1.2) на моменти нульового порядку.

Слід відзначити, що строкові моменти, як правило, забезпечують нижчий рівень розпізнавання. Центральні і нормовані моменти більш кращі внаслідок їх більшої інваріантності до перетворень зображень.

У методі перетинів ознаки формуються шляхом підрахунку того, скільки раз та яким чином відбувся перетин зображення символу з вибраними прямими, проведеними під певними кутами. Цей метод часто використовується в комерційних системах завдяки тому, що він інваріантний до дисторсії та невеликим стилістичним варіаціям написання символів, а також має досить високою швидкістю і не вимагає високих обчислювальних витрат [21].

Метод зон передбачає поділ площі рамки, що охоплює символ, на області та подальше використання щільності точок в різних областях як набір характерних ознак. На зображеннях виконується розбиття на зони, обчислюються піксельні ваги кожної із зон, а також вектор відстаней до еталонних векторів еталонних символів.

У методі матриць суміжності як ознаки розглядаються частоти спільної появи «чорних» та «білих» елементів у різних геометричних комбінаціях.

Метод характеристичних місць (characteristic-loci) використовує в якості ознак число разів, яке вертикальний та горизонтальний вектори перетинають вибрані відрізки ліній для кожної світлої точки в області фону символу. Існує також значна кількість інших методів даної групи.

Серед сучасних технологій розпізнавання, заснованих на різного виду

перетвореннях, виділяються методи, що використовують Фур'є-дескриптори символів, а також частотні дескриптори кордонів.

Переваги методів, що використовують перетворення Фур'є - Мелліна, пов'язані з тим, що вони мають інваріантність до масштабування, обертання і зсуву символу. Основний недолік цих методів полягає в нечутливості до різких стрибків яскравості на кордонах символів, наприклад, по спектру просторових частот складно відрізнити символ «O» від символу «Q» і т. п. В той же час, при фільтрації завад на кордонах символу ця властивість може виявитися корисною.

Структурні ознаки зазвичай використовуються для виділення загальної структури образу. Вони описують геометричні і топологічні властивості символу [10]. Найпростіше уявити ідею структурного розпізнавання символу тексту стосовно до задачі автоматичного зчитування поштових індексів. У таких «трафаретних» шрифтах положення кожного можливого відрізка-штриха заздалегідь відомо, і один символ відрізняється від іншого не менш ніж наявністю або відсутністю такого цілого штриха. Аналогічне завдання виникає і в разі контролю простих рідкокристалічних індикаторів. У таких системах виділення структурних складових зводиться до аналізу елементів заздалегідь відомого трафарету (набору відрізків, що підлягають виявленню).

В системах структурного розпізнавання більш складних шрифтів часто використовуваними ознаками також є штрихи, що застосовуються для визначення таких характерних особливостей зображення: кінцевих точок, точок перетину відрізків, замкнутих циклів, а також їх положення щодо рамки, яка охоплює символ. Розглянемо, наприклад, наступний спосіб структурного опису символу. Нехай матриця, яка містить потоншений символ, розділена на дев'ять прямокутних областей (у вигляді сітки 3x3), кожній із яких присвоєно літерний код від «A» до «I» (або від 0 до 8). Символ розглядається як набір штрихів. При цьому штрих, що з'єднує деякі дві точки в зображенні символу, може бути лінією (L) або кривою (C). Штрих вважається відрізком (дугою) кривої, якщо його точки задовольняють наступному виразу

$$B = \frac{1}{n} \sum_{i=1}^n \frac{ax_i + by_i + c}{\sqrt{a^2 + b^2}} > 0,69. \quad (1.3)$$

В іншому випадку із виразу (1.3) вважається, що це прямолінійний відрізок. У даній формулі  $(x_i, y_i)$  є точкою, що належить штриху;  $ax + by + c = 0$  — рівняння прямої, що проходить через кінці цього штриха, коефіцієнт 0,69 отриманий дослідним шляхом. Далі символ може бути описаний набором своїх відрізків і дуг. Наприклад, запис { «ALC», «ACD»} означає наявність прямої, що проходить із області «А» в область «С», та кривої, що проходить із області «А» відповідно в область «D».

Основна перевага структурних методів розпізнавання визначається їх стійкістю до зсуву, масштабування та повороту символу на певний невеликий кут, а також до можливих дісторсій та різних стильових варіацій та невеликим спотворенням аналізованих шрифтів.

#### 1.4 Способи перевірки достовірності розпізнавання символів

В існуючих системах OCR використовуються різноманітні алгоритми класифікації та віднесення ознак до різних класів. Вони істотно різняться залежно від прийнятих наборів ознак та застосовуваної по відношенню до них стратегії класифікації.

Для ознакової класифікації символів необхідно, в першу чергу, сформулювати набір еталонних векторів ознак по кожному з символів, що розпізнається. Для цього на стадії навчання системи оператор або розробник вводить у систему розпізнавання велику кількість зразків накреслення символів, супроводжуваних зазначеннями символу. Для кожного зразка система виділяє ознаки та зберігає їх у вигляді відповідного вектора ознак. Набір векторів ознак, що описують символ, називається класом, або кластером.

У процесі експлуатації системи OCR може з'явитися необхідність розширити сформовану раніше базу знань. У зв'язку з цим деякі системи мають можливість до навчання в реальному режимі часу.



Завданням власне процедури класифікації або розпізнавання, виконуваної в момент пред'явлення системі тестового зображення символу, є визначення того, до якого з раніше сформованих класів належить вектор ознак, отриманий для даного символу. Алгоритми класифікації засновані на визначенні ступеня близькості набору ознак розглянутого символу до кожного з класів. Правдоподібність одержуваного результату залежить від обраної метрики простору ознак. Найбільш відомою метрикою простору ознак є традиційна Евклідова відстань (1.4)

$$D_j = \sqrt{\sum_{i=1}^N (F_{ji}^L - F_i^l)^2}, \quad (1.4)$$

де  $F_{ji}^L$  — і-а ознака j-го еталонного вектора;

$F_i^l$  — і-а ознака зображення символу, що перевіряється [20].

При класифікації за методом найближчого сусіда символ буде віднесений до класу, вектор ознак якого найбільш близький до вектору ознак символу, що перевіряється. Слід враховувати, що витрати на обчислення в таких системах зростають зі збільшенням кількості використовуваних ознак і класів.

Одна з методик, що дозволяють поліпшити метрику подібності [20], заснована на статистичному аналізі еталонного набору ознак. При цьому в процесі класифікації більш надійним ознакам у виразі (1.5) віддається більший пріоритет:

$$D_j = \sqrt{w_i \sum_{i=1}^N (F_{ji}^L - F_i^l)^2}, \quad (1.5)$$

де  $w_i$  — вага і-ї ознаки.

Інша методика класифікації, що вимагає знання апріорної інформації про ймовірнісної моделі тексту, заснована на використанні формули Байеса. З правила Байеса виходить, що розглянутий вектор ознак належить класу «j», якщо відношення правдоподібності  $\lambda$  більше, ніж ставлення апріорної ймовірності класу j до апріорної ймовірності класу i.

У відповідальних системах OCR якість розпізнавання, що отримується при розпізнаванні окремих символів, не рахується достатньою. У таких системах необхідно використовувати також контекстну інформацію. Використання контекстної інформації дозволяє не тільки знаходити помилки, але і виправляти їх.

Існує велика кількість додатків OCR, що використовують глобальні та локальні позиційні діаграми, триграми, n-грами, словники і різні поєднання всіх цих методів [21]. Розглянемо два підходи до вирішення цього завдання: словник і набір бінарних матриць, що апроксимують структуру словника.

Доведено, що словникові методи є одними з найбільш ефективних при визначенні способів усунення помилок класифікації окремих символів. При цьому після розпізнавання всіх символів деякого слова словник проглядається в пошуках цього слова, з урахуванням того, що воно, можливо, містить помилку. Якщо слово знайдено в словнику, то це не говорить про відсутність помилок. Помилка може перетворити одне слово, яке перебуває в словнику, в інше, що також входить в словник. Така помилка не може бути виявлена без використання смислової контекстної інформації: тільки вона може підтвердити правильність написання. Якщо слово в словнику відсутнє, вважається, що в слові допущена помилка розпізнавання. Для виправлення помилки вдаються до заміни такого слова на найбільш схоже слово зі словника. Виправлення не проводиться, якщо в словнику знайдено декілька відповідних кандидатур для заміни. В цьому випадку інтерфейс деяких систем дозволяє показати слово користувачеві і запропонувати різні варіанти рішення, наприклад, виправити помилку, ігнорувати її і продовжувати роботу або внести це слово в словник [9].

Головний недолік у використанні словника полягає в тому, що операції пошуку і порівняння, що застосовуються для виправлення помилок, вимагають значних обчислювальних затрат, що зростають зі збільшенням обсягу словника. Через це весь процес завершальної фази обробки стає вкрай неефективним.

Деякі розробники з метою подолання труднощів, пов'язаних з використанням словника, намагаються виділяти інформацію про структуру слова з самого слова. Така інформація говорить про ступінь правдоподібності n-грам (символьних послідовностей, наприклад, пар або трійок букв) в тексті, які також можуть бути глобально позиціоновані, локально позиціоновані або взагалі не позиціоновані. Наприклад, ступінь достовірності непоціонованої пари букв може бути представлена у вигляді бінарної матриці, елемент якої дорівнює 1 тоді і тільки тоді, коли відповідна пара букв є в деякому слові, що входить в словник.

Позиційна бінарна діаграма  $D_{ij}$  є бінарною матрицею, яка визначає, яка з пар букв має ненульову ймовірність виникнення в позиції  $(i, j)$ . Набір всіх позиційних діаграм включає бінарні матриці для кожної пари положень.

Таким чином, технологія OCR дозволяє досить точно розпізнавати текст на зображеннях, крім варіантів, коли безпосередньо текст спотворений. Це можуть бути випадки перекриття символами один одного або схожості образів одних символів на інші (наприклад, поєднання букв «LI» може бути розпізнано, як «U»). Але до сих пір не розроблена технологія, що дозволяє в автоматизованому режимі розпізнати текст без помилок, тому в даному питанні ще потрібне втручання людини.

### 1.5 Аналіз засобів розпізнавання текстових символів

Завдання розпізнавання текстової інформації при переведенні рукописного або ж друкованого тексту в електронну форму є одним із найважливіших складових будь-якого проекту, яке має на меті отримати автоматизацію обігу документів чи запровадження безпаперових технологій. Однак одночасно це завдання є одним із найбільш складних та наукоємних задач процесу повністю автоматизованого аналізу зображень. Навіть людина, яка читає рукописний текст, в середньому вчиняє понад 4% помилок. При цьому, в найбільш відповідальних та надійних додатках OCR потрібно забезпечувати вищу надійність розпізнавання тексту (що має сягати близько

99% та більше) навіть при поганому рівні друку та за виконання операції конвертації вихідного тексту у цифровий формат.

В останні десятиріччя, завдяки використанню сучасних здобутків комп'ютерних технологій, були розроблені нові методи обробки зображень та розпізнавання образів. Завдяки цьому стало можливим створення ряду промислових систем розпізнавання друкованого тексту, які задовольняють основні вимоги для систем автоматизації документообігу.

Система оптичного розпізнавання тексту OCR (optical character recognition) — це конвертація різного типу зображень (друкарського, машинописного, рукописного) у електронне подання у вигляді тексту, щоб мати можливість для обробки, аналізу інформації та проведення його подальшого редагування. По іншому можна сказати, що — це процес отримання тексту із зображення. У програмному представленні, OCR — це розроблений на різних мовах програмування алгоритм для обробки зображення, що програмним шляхом розпізнає текстові символи [21].

Основні призначення OCR-систем полягають в аналізі растрової інформації (тобто відсканованого символу) та присвоєнні певному сегменту зображення відповідного символу. По завершенню процесу розпізнавання зображення OCR-системи повинні бути здатними зберігати форматування вихідних документів, привласнювати у належному місці атрибут абзацу, зберігати графіку, таблиці і т.д. Більшість сучасних програм розпізнавання підтримують усі відомі графічні та текстові формати та формати електронних таблиць, а деякі навіть підтримують такі формати, як HTML та PDF.

Переважає більшість OCR-систем мають елементарний автоматичний режим сканування зображення та розпізнавання (Scan and Read). Окрім цього, вони підтримують також режим розпізнавання зображень із файлів. Однак для того, аби досягти кращих із можливих для певної системи результатів, бажано (а іноді й обов'язково) попередньо вручну налаштувати систему на конкретний вид тексту, якість паперу та макет бланка. Працюючи з OCR-системою дуже важливою є зручність вибору мови розпізнавання і виду матеріалу, що

розпізнається, (матричний принтер, газета, друкарська машинка, факс, і т. д.), а також інтуїтивна зрозумілість інтерфейсу для користувача [16]. При розпізнаванні текстів, що в них використано не одну мову, а декілька, ефективність розпізнавання буде залежати від уміння OCR-системи формувати осередки мов. Нижче наведено основні характеристики найбільш поширених та популярних систем розпізнавання символів.

Програмний засіб ABBYY FineReader є визнаним лідером за якістю і кількістю технологій, що задіяні для оптичного розпізнавання, ущільнення і переведення. Програма працює зі сканером та підтримує такі формати зображень як jpg, png, jpeg, bmp, gif. ABBYY FineReader — відмінно реалізований додаток для розпізнавання тексту, а також для його сканування. Оновлена версія програми відмінно справляється із завданням переведення паперових документів та різноманітних зображень у електронний вид [22]. Програмний засіб може похвалитися доволі зручним оформленням інтерфейсу. Після того, як конвертація документів буде завершена, користувач може працювати із текстовими даними у будь-якому редакторі.

Основні можливості ABBYY FineReader:

- це засіб розпізнавання тексту, що передбачає роботу із 179 мовами світу;
- вбудовані інструменти, які відповідають за збільшення якості отриманих зображень і фото;
- висока точність розпізнавання зображень за допомогою технології Abbyu OCR;
- додаток підтримує операційну систему Windows 10, а також Windows 7 та 8;
- можливість переведення готового тексту в різні формати PDF, DOC, RTF, XLS, а також HTML;
- утиліта підтримує розпізнавання зображень, отриманих із сканерів, цифрових фотоапаратів та навіть із смартфона;
- зареєстровані користувачі ліцензійного ABBYY FineReader мають

доступ до сервісу Abbyu Finereader Online.

Переваги:

- розробник додатку надає регулярні оновлення своєї утиліти для сканування і розпізнавання тексту;
- обробка документів, що були сфотографовані на мобільний телефон;
- одночасна робота з пакетом сканів та зображень;
- програмне забезпечення здатне автоматично розпізнавати різні мови в одному документі;
- має високі показники точності оптичного розпізнавання текстових даних;
- збереження високої вихідної якості зображення документів;
- може підтримуватися будь-яка актуальна система Windows;
- особливо високі показники роботи з редакторами Microsoft Office.

Попри того, що додаток має доволі таки багато переваг, він має і певні недоліки, до яких належить:

- пробна версія додатку має обмежений термін дії та урізаний функціонал на фоні ліцензійної версії;
- продукти компанії Abbyu потребують обов'язкової реєстрації та поширюються тільки за допомогою підписки;
- в самій програмі Abbyu Fine Reader досить важко редагувати текст, тому краще одразу розпізнавати його з зображення та конвертувати у Microsoft Word;
- конвертація часто знищує нормальне форматування, переміщує колонтитули для вихідного друкованого тексту.

Ще одним засобом для оптичного розпізнавання текстів є OCR CuneiForm [23]. Це мабуть, найкраща безкоштовна програма, що є аналогом FineReader. Вона не настільки розвинена, однак при цьому менш складна і менш вимоглива. Це досить проста і зручна утиліта для виділення та захоплення тексту із приємним зовнішнім виглядом та інтуїтивно зрозумілим інтерфейсом. Вона вміє сканувати документи в Microsoft Word чи інший

додаток із офісного пакету, обробляти одночасно декілька файлів різних форматів (як приклад, jpeg, tiff і т.д.), перевіряти відредагований текст на наявність помилок.

OCR Cuneiform [23] швидко може розпізнавати як відскановані, так і сфотографовані тексти. В процесі розпізнавання текстів, обробці піддається величезний спектр друкованих шрифтів, але при цьому зберігається вихідна форма електронних документів. Готовий результат, що отриманий в додатку OCR, можна відправити для подальшого опрацювання в текстових редакторах. Функціонал OCR Cuneiform можна порівнювати з більш відомим додатком Abbyy Finereader, окрім того, що скачати першу можна на повністю безкоштовній основі.

Можливості OCR CuneiForm:

- повне збереження вихідної структури шрифтів;
- оптичне розпізнавання різноманітних текстових документів;
- пакетна обробка різних документів;
- безкоштовна програма Cuneiform забезпечує ефективне та швидке розпізнавання текстів та символів;
- конвертація електронних графічних документів та паперових документів в зручний для редагування вид;
- перетворення графічних файлів, різного тексту та ксерокопій факсів у формати файлів для пакету Microsoft Office;
- робота із уже розпізнаним текстом, передбачає аналіз документа та зручний пошук текстових блоків, зображень, таблиць.

Переваги:

- має зручну систему оптичного розпізнавання більшості існуючих друкованих шрифтів;
- форматування будь-якої структури файлів у зручний формат для відомих офісних програм та текстових редакторів;
- повний аналіз відсканованих документів та зручна система пошуку потрібної таблиці, будь-якого зображення чи тексту;

- розробники періодично оновлюють програму, розширюючи її опції та корисні інструменти;
- висока якість розпізнавання графічної і текстової інформації.

Як недолік можна виділити можливе пригальмовування програми Cuneiform під час розпізнавання документів.

Програма Readiris Pro [24] є надзвичайно потужним, але і недешевим програмним засобом розпізнавання тексту зі сканера, документа форматів pdf, djvu, tiff чи jpeg. Єдиний серед аналогів включає в себе інтелектуальну систему розпізнавання відсканованого паперового документа чи рукописного тексту, при цьому, виконує подібну роботу якісно, на досить високому рівні. Каліграфічно вірний почерк сприймається утилітою добре, українська і російська мови підтримуються.

Основні характеристики програми Readiris:

- хороша якість розпізнавання на більш ніж 100 мовах світу, включаючи таблиці та рукописні документи;
- містить можливість отримати текст із картинки в різних форматах, також із електронної книги формату djvu;
- розпізнаний текст можна експортувати не тільки у Microsoft Office Word, Excel, але і в PDF, OpenOffice, XPS;
- має інструмент компресії, що стискає зображення зі сканера в 200 разів без втрат якості;
- програма платна, однак проста у використанні — всі її дії виконуються в два натискання миші.

Програма Freemore OCR [21], так як і решта безкоштовних програм розпізнавання тексту, не забезпечує використання великої кількості функцій, але при цьому і не вимагає від користувачів придбання дорогої ліцензії. Відповідно до заявлених характеристик, технологія Side-by-Side допомагає програмою коректно відокремлювати картинку від символів, щоб не захоплювати зайві блоки документу. Утиліта зберігає розпізнані документи у текстовому форматі doc, txt чи pdf.



### Особливості додатку Freemore OCR:

- дає можливість розпізнавати документи, pdf-файли та зображення майже всіх форматів;
- відсутнє повноцінне розпізнавання українського тексту, тільки латиниця;
- меню програми зрозуміле та просте, для роботи в програмі не потрібні поглиблені знання англійської мови;
- програмний засіб регулярно оновлюється із офіційного сайту розробника.

Ще одна програма Tesseract — це вільна система розпізнавання тексту, що активно розробляється Hewlett-Packard, і яка із 2006 року спонсорується корпорацією Google [25].

Обробка зображень у Tesseract відповідає традиційному покроковому методу, однак деякі кроки відрізняють цю систему від решти інших. Першим кроком є проведення аналізу зв'язності компонентів, у результаті чого їх можна відокремити. Попри деяку обчислювальну складність цього процесу, його перевагою є виділення контурів окремих елементів зображення. Це робить процес розпізнавання не складнішим, аніж розпізнавання чорного тексту на білому фоні.

Потім виділені контури вкладаються у комірки, які організовуються в текстові рядки. Сортування та обробка комірок по  $x$ -координатам дає дозвіл привласнити їх винятковим рядкам тексту.

Якщо шрифт тексту є моно широким, то він відразу розбивається на окремі символи методом накладання рівномірної сітки. Текст, що має пропорційний шрифт, розбивається на окремі слова методом пошуку прогалів між ними. Після цього виконується розпізнавання у два проходи. При першому проході виконується спроба розпізнати кожне слово. Надійно та достатньо розпізнані слова додаються у адаптивний класифікатор у якості тренувального набору, згодом вони використовуються при подальшому розпізнанні слів, які знаходяться нижче у тексті. Після першого проходження

слів по текстовому документі здійснюється ще одне проходження з урахуванням всіх внесених у класифікатор слів.

Для слів, що не були достатньо розпізнані через різного типу перетини чи незв'язність символів, виконується обробка обрисів самих символів. Для розбиття символів, що перетинаються, формуються кандидати на точки розбиття. Потім вони встановлюються на ввігнутих вершинах полігональної апроксимації контурів (рис. 1.1). Ті із кандидатів, що не покращують точності розпізнавання, відкидаються.

При цьому, якщо точність розпізнавання все ще є недостатньо високою, то в такому випадку виконується поєднання розбитих символів. Асоціатор робить перевірку з різними варіантами поєднання контурів на підвищення точності розпізнавання тексту.

Основним недоліком даного програмного продукту є значна складність роботи із деформованим або пошкодженим текстом, при якому достовірність розпізнавання зменшується на 20-30% [25].

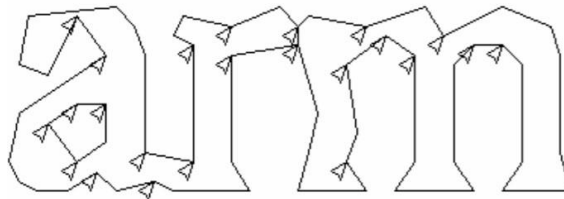


Рисунок 1.1 — Кандидати на точки розбиття

Програма OCRopus — аналізатор документів, що випущений Німецьким Інститутом Дослідження Штучного Інтелекту згідно ліцензії Apache 2.0. Програмний засіб OCRopus був розроблений спеціально для експлуатації у проектах з оцифрування досить великих обсягів документів, таких як інтернет-архіви або бібліотеки та Google Books. Завдяки модульному підходу системи можна вносити власні зміни в певні етапи розпізнавання тексту [21].

Основна архітектура додатку OCRopus є строго випереджаючою (тобто виключає повторення певних етапів розпізнавання) та має три головні функції:

виконання аналізу макета, розпізнавання текстових рядків та статистичне мовне моделювання.

Програмний засіб OCRopus забезпечує типовий набір інструментів бінаризації та обробки зображень у відтінках сірого. До того ж, OCRopus надає чотири інструменти, що менш часто зустрічаються в аналогічних системах:

- морфологічні інструменти кодування повторів символів дають змогу виконувати операції із використанням великих масок та із використанням непрямокутної маски з великою швидкістю;

- ефективний локальний адаптивний пороговий алгоритм, виконаний на основі інтегральних зображень;

- алгоритм із високою точністю виявлення та усунування перекосу;

- інструмент для визначення зон без тексту для їхнього видалення.

OCRopus містить два методи для розпізнавання рядків тексту. Перший метод має в основі алгоритм розпізнавання рядків як у системі Tesseract. Для другого методу використовуються багатошарові перцептрони для розпізнавання символів. Розпізнавання здійснюється у декілька етапів, спершу відбувається спроба надмірної сегментації вхідного рядка, а далі йде висунування гіпотез стосовно розпізнаних символів, та, нарешті, обидва результати розпізнавання та геометричні відносини між символними гіпотезами зображуються у вигляді графа.

Третім основним компонентом додатку OCRopus є статистичне мовне моделювання. Засіб OCRopus містить у собі відкриту OpenFST бібліотеку, в якості основи для своїх статистичних інструментів мовного моделювання, що представляє мовну модель як зважені кінцеві перетворювачі. Також їх можна розглядати наче форму «зваженого регулярного виразу» із можливістю виконувати обмежені заміни.

## 1.6 Порівняння систем оптичного розпізнавання символів

Вище розглянуті OCR-системи являються найбільш популярними та

відомими, однак із огляду на активний розвиток області розпізнавання, схожих систем є дуже багато. Проведемо порівняльний аналіз по представлених вище системах оптичного розпізнавання символів. Для оцінювання якості OCR-систем існують різні критерії оцінки. Потік усіх операцій, що входять у процес оптичного розпізнавання, можна поділити на два етапи, що із великою значимістю впливають на кінцевий результат: перший — це структурний аналіз (тобто сегментація документа, класифікація регіонів) та другий — власне розпізнавання тексту. Для кожного із етапів існують власні методи оцінки якості [21]:

- оцінювання результатів розпізнавання за стандартами;
- оцінювання результатів розпізнавання без еталонів.
- оцінювання результатів структурного аналізу документа;

У таблиці 1.1 представлені порівняння систем по наведеним проблемам та із урахуванням способів оцінки їх якості роботи.

З таблиці 1.1 видно, що жодна із розглянутих систем розпізнавання тексту найбільш повно не вирішує поставлені проблеми. При передачі у обробку зображення низької якості, яке має символи із перекрученою формою, існує висока ймовірність помилкового розпізнавання.

Оскільки OCR-системи мають і технічні характеристики, так як і будь-яке інше ПЗ, то слід провести порівняння за даними параметрами також (див. таблиця 1.2).

Розглянуті системи, як уже було сказано раніше, належать до списку найбільш відомих та широко розповсюджених OCR систем. Є ряд рейтингів, що визначають найкращі системи розпізнавання зображень. Відповідно до даних рейтингів однією з кращих програм вважається Abby FineReader, у якій поєднуються структурні та растрові методи розпізнавання із додатковим використанням лінгвістичних методів.

Таблиця 1.1 — Порівняння оптичних систем розпізнавання

Система	Форми	і	Спотворення	Варіації розмірів
---------	-------	---	-------------	-------------------

	креслення символів	зображень	та масштабів
ABBYY Fine Rider	Так	Ні	Так
OCR CuneiForm	Так	Ні	Ні
Readiris Pro	Так	Ні	Так
Freemore OCR	Так	Так	Ні

Таблиця 1.2 — Порівняння OCR-систем по технічним параметрам

Назва системи	Остання версія (рік)	Ліцензія	Операційна система	Кількість мов	Вихідний формат
ABBYY Fine Rider	2017	Потрібна покупка	Windows, Linux, Mac OS X та інші. UNIX-подібні	192	DOC, DOCX, TXT, PPTX, RTF, XLS, XLSX, HTML, PDF, CSV, DjVu, EPUB, ODT, FB2
Readiris Pro	2017	Потрібна покупка	Windows, Mac OS X	137	Word, PDF, HTML, TXT, Excel, ePub
OCR CuneiForm	2011	Безкоштовна	Windows, Linux, Mac OS X та інші. UNIX-подібні	22	TXT, DOC, DOCX
Freemore OCR	2019	Безкоштовна	Windows	1	DOC, DOCX, PDF, TXT

Алгоритми роботи більшості комерційних програм розпізнавання тестових документів як правило не розкриваються. Але аналіз їх роботи та сфери застосування вказують на використання структурних методів розпізнавання у багатьох системах розпізнавання тексту із подальшим доповненням лінгвістичними методами. Таке поєднання методів

розпізнавання тексту дозволяє досягти точності розпізнавання до 98% [21]. Але ще кращі результати можна отримати при застосуванні методів розпізнавання на основі загорткових нейронних мереж. При використанні таких мереж досягнуті результати розпізнавання звичайних текстових документів на рівні до 99,5% [12]. Слід також відмітити, що нейронні мережі мають значний вигреш у розпізнаванні образів порівняно із статистичними класифікаторами, такими як, наприклад, Байєсовський. Але при розпізнаванні текстових документів із значною кількістю пошкоджених символів всі ці підходи показують гірші результати розпізнавання та потребують подальшого вдосконалення.

## 2 РОЗРОБКА ПОСЛІДОВНОСТІ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ

### 2.1 Розробка послідовності розпізнавання тексту

Методи автоматичного розпізнавання образів та їх реалізація в системах оптичного читання текстів (OCR-системах) є однією із найбільш відомих та поширених технологій штучного інтелекту [11]. Під цим у наведеному трактуванні під системою OCR розуміється автоматичне розпізнавання за допомогою спеціальних програм зображень символів друкованого або рукописного тексту. Іноді під OCR розуміють пристрій оптичного розпізнавання символів або автоматичного читання тексту [11].

Відзначимо такі особливості розпізнавання тексту, істотні з точки зору OCR-систем: шрифти та різноманітність розмірів символів; спотворення в зображеннях символів (розриви образів символів, наприклад, при збільшенні зображення; злипання сусідніх символів і ін.); перекося при скануванні; сторонні включення в зображеннях; поєднання фрагментів тексту на різних мовах; велике різноманітність класів символів, які можуть бути розпізнані тільки при наявності додаткової контекстної інформації.

Як відомо, системи розпізнавання реалізуються як класифікатори, що використовують різні методи: шаблонні (растрові), ознакові або структурні [10]. У класифікаторі шаблонного типу за допомогою критерію порівняння визначається, який з шаблонів вибрати з бази. Самий простий критерій - мінімум точок, що відрізняють шаблон від досліджуваного зображення. До переваг шаблонного класифікатора відносяться гарне розпізнавання дефектних символів («розірваних» або «склеєних»), простота і висока швидкість розпізнавання. Недоліком є необхідність налаштування системи на типи і розміри шрифтів.

У ознакових класифікаторах аналіз проводиться тільки по набору чисел або ознак, що обчислюються по зображенню. Цей метод дозволяє розпізнавати різні накреслення символів, тобто різні почерки, шрифти і т.д. Цей метод

неминуче викликає деяку втрату інформації, так як використовується топологічний уявлення, що відбиває інформацію про взаємне розташування структурних елементів символу. Ці дані можуть бути представлені в графовой формі. При цьому даний метод забезпечує інваріантність щодо типів і розмірів шрифтів. Недоліками є значні труднощі розпізнавання дефектних символів та досить повільна робота.

Основою структурно-плямового методу є структурно-плямовий еталон [26]. Він має вид набору плям з попарними відносинами між ними. Дане представлення невідчутно до різних накреслень та дефектів символів. Алгоритм заснований на поєднанні шаблонного і структурного методів розпізнавання образів. При аналізі зразка виділяються ключові точки об'єкта — так звані «плями». У якості плям, наприклад, можуть виступати: кінці ліній; вузли, де сходяться кілька ліній; місця зламів ліній; місця перетину ліній; крайні точки.

Після виділення характерних точок визначаються зв'язку між ними — відрізок або дуга. Таким чином, підсумковий опис являє собою граф, який і служить об'єктом пошуку в бібліотеці «структурно-плямових еталонів». При пошуку встановлюється відповідність між ключовими точками зразка та еталону, після чого визначається ступінь деформації зв'язків, необхідна, щоб привести шуканий об'єкт до порівнювати еталонному зразку. При цьому, менша ступінь необхідної деформації передбачає велику ймовірність правильного розпізнавання символу. Далі розглянемо етапи обробки зображення.

Початковим етапом усіх алгоритмів розпізнавання є етап попередньої обробки. На цьому етапі покладається виконання таких завдань: підвищення якості зображення за рахунок фільтрації, видалення завад та інші операції, що мають на меті підвищити якість зображення. На цьому етапі відбувається очищення зображення від дефектів сканування. І, зокрема, на самому початку роботи до отриманого зображення із метою видалення завад часто застосовується фільтр Гауса. Важливу роль відіграє операція бінаризації по



порогу, тобто переведення зображення в чорно-білий формат з кольорового або відтінків сірого [17]. Це дозволяє різко розділити отримане зображення на текст та фон, що надалі спрощує використання багатьох алгоритмів, а також додатково позбавляє від деяких завад на зображенні. Для цього етапу використовується гістограма яскравості отриманого зображення тексту, на якому виділяються два піки: високий пік, що відповідає білому фону, тобто фактично кольору паперу, та пік в області темних пікселів, що відповідають яскравості символів тексту.

Наступним етапом є виділення тексту на зображенні як регіону інтересу. На цьому етапі роботи на бінаризованому зображенні виділяється безпосередньо область, на якій знаходиться текст, що підлягає розпізнаванню, та відкидаються такі елементи, що не є текстом у отриманому зображенні [27, 28]. До таких елементів відносяться такі об'єкти, як плями, картинки, плями на папері, що не видалені в процесі бінаризації та ін. Для їх видалення можна, наприклад, виділяти компоненти зв'язності на зображенні, обчислювати геометричні ознаки та на їх основі виконувати класифікацію компонентів зв'язності як частину тексту або можливий дефект, використовуючи методи машинного навчання або евристичні підходи.

Надалі приступають до виконання операцій сегментації та нормалізації текстових фрагментів. На цьому етапі текст розділяється, або ж сегментується, на зручні для аналізу складові частини [29]. Найбільш природними діями на даному етапі є поділ тексту на окремі рядки (сегментація рядків), потім поділ рядків на окремі слова (сегментація слів), а надалі поділ виділених слів на елементарні складові частини у вигляді символів. Крім того, на даному етапі проводиться нормалізація тексту та приведення виділених складових частин до деякого стандартного виду для зниження варіативності та подальшої можливості спрощення операції розпізнавання.

Етап сегментації або ж поділу рядків у друкованих машинним способом документах на сьогоднішній день вважається повністю вирішеною. Але при розгляді задачі на поділ рядків у загальному випадку виникають певні

складності, що не дозволяють безпосередньо застосовувати алгоритми, які придатні для машинописних текстів:

— рядки не тільки можуть не бути паралельними, але і також можуть згинатися;

— різні рядки можуть бути розташовані занадто близькі, а елементи тексту, що належить різним рядкам, можуть накладатися друг на одного.

Наприклад, якщо коефіцієнт форми області (відносини квадрата її периметра до площі) менше деякого значення, а площа більше деякого значення, то це з великою ймовірністю є дефектом.

Іноді використовують процес для виділення базових ліній. Така операція застосовується у випадках, коли при отриманні текстового документа відбулися значні спотворення при деяких перекосах при його скануванні. Ці методи засновані на ідеї, що попередньо слід зайти деяку уявну лінію. У таких методах намагаються спочатку апроксимувати цю лінію, а вже потім відновити по ній рядок текстового документа. У такому випадку використовують перетворення Хафа для виділення прямих, якщо вони є не дуже викривленими. Перетворення Хафа застосовується до центрів компонент зв'язності пікселів тексту [17].

Можливий перетин елементів різних рядків при значних дефектах при отриманні відсканованого документа являє собою проблему не тільки сегментації рядків, але і також для розпізнавання тексту, так як віднесення елементу зображення до невірно визначеного рядка значно погіршує його розпізнавання. Пересічні компоненти є проблемою для використання методів пошуку горизонтальної проекції (так як вони збільшують значення профілю проекції у тих місцях, де повинно бути значення його мінімуму) та методів групування, але слабо впливають на деякі методи виділення базових ліній [9].

Для пошуку елементів, що пересікаються, із різних рядків можна використовувати такі ознаки, як розмір компонент зв'язності тексту, факт віднесення однієї компоненти до кількох рядків або ж факту, що компонент не відносяться до жодного рядку. Після знаходження таких сумнівних

компонентів слід визначити, чи належать вони до якогось рядку або ж їх потрібно розділити на такі елементи, що відносяться до різних рядках. Така вертикальна декомпозиція компонентів є досить складним завданням. Простіше рішення полягає в розрізуванні компоненти на частини горизонтальними лініями, але можна використовувати і більш тонкі підходи, наприклад, виділення окремих штрихів.

Наступним важливим етапом є виконання операції сегментації слів. На цьому етапі роботи системи розпізнавання виділені рядки текстового документа поділяються на окремі слова. На відміну від машинописного тексту, в якому відстань між словами більш-менш постійна, а інтервали між символами усередині слова набагато менші, ніж інтервали між словами, в інших типах текстових документів розмір інтервалів між словами може змінюватися у досить широких межах. Виділенні компоненти зв'язності тексту, віднесені до одного рядка на попередньому етапі роботи системи розпізнавання, на цьому етапі об'єднуються у слова.

Ще одним етапом роботи системи розпізнавання текстових документів є нормалізація отриманого зображення. В силу високої варіативності накреслення символів слів їх розпізнавання є досить складним процесом [26]. Нормалізація служить для приведення слова або кожного символу слова до деякого стандартного вигляду без значної втрати інформації, необхідної для виконання етапу подальшого розпізнавання. Одними із найбільш часто використовуваних методів нормалізації є метод корекції нахилу слова від горизонтальної та вертикальної лінії та приведення до деяких визначених розмірів. Найпростіший метод корекції слова по горизонталі полягає у виконанні операції максимізації яскравості його зображення по деякому діапазону. Існують і інші методи нормалізації, наприклад, корекція розміру та виділення скелету тексту, але вони застосовуються рідше. Також є і інші методи, наприклад, засновані на згладжуванні та лінійній регресії [18].

Отже, розпізнавання тексту повинно включати в себе такі етапи.

Зображення, що надходить на вхід системи, повинно бути очищено від

завад та приведенне до такого вигляду, що дозволяє ефективно виділяти окремі символи та розпізнавати їх.

Система повинна розбити зображення на окремі блоки тексту, ґрунтуючись на особливостях його вирівнювання та розподілу по декільком колонкам.

Зображення із текстом має бути спочатку розділене на зображення рядків, а потім на зображення символів для того, щоб в подальшому виконати оброблення кожного символу окремо. Після виконання даного кроку різні системи розпізнавання текстів працюють за своїми притаманними їм специфічними алгоритмами.

Зображення символу може оброблятися як єдине ціле, для цього виконується порівняння із наявними шаблонами, або ж розпізнається за допомогою нейронних мереж або виділяються попередньо визначені характеристики зображуваного символу.

На виході четвертого кроку роботи системи з'являється можливий варіант літери. Однак зазвичай системи на цьому не зупиняються та продовжують роботу на основі використання інших методів, послідовно уточнюючи отриманий результат.

Перед виконанням операції розпізнавання тексту у отриманому зображенні слід попередньо поліпшити його якість для збільшення ймовірності вірного розпізнавання кожного із символів цього зображення. Для цього виконуються такі операції. Спочатку здійснюється операція по видаленню завад (це можна виконати із застосуванням медіанного фільтру) та згладженню елементів зображення (для цього можна скористатися фільтром Гауса).

Потім можна перейти до операції по згладженню контурів символів у отриманому зображенні. Тут досить часто використовують операції послідовного використання метода ерозії (erode) та наступного процесу розширення (дилатації, dilate) до початкового зображення. Отримане зображення переводиться у градації сірого кольору та виконується процес

вирівнювання освітлення сторінки тексту зображення. Для цих задач можна скористатися алгоритмом SSR як одним із модифікованих алгоритмів по логарифмічній корекції зображення. Даний алгоритм вирівнює яскравість зображення, одночасно зберігаючи локальну контрастність у місцях як яскраво, так і погано освітлених частинах зображення.

На першому етапі у всіх системах оптичного розпізнавання символів виконуються поліпшення якості і аналіз зображення, поданого на обробку. У тому числі застосовуються спеціальні фільтри відновлення пошкоджених зображень, наприклад, за допомогою гіпоеліптичної дифузії [10], фільтри, що усувають змазування [17].

На другому етапі виконується робота по виявленню текстів на зображеннях. Відомі наступні підходи: на основі використання контурної інформації (скелетизації), виділення країв та виділення кутів [17], методи на основі інваріантних моментів [29]; на основі колірної інформації (метод гістограм), аналіз головних компонент [26] і різні алгоритми адаптивної бінаризації, у тому числі алгоритми бінаризації від Niblack, Sauvola, Chistian, Bernsan, Otsu тощо [30, 31]); на основі аналізу текстурної інформації (метод опорних векторів, штучні нейронні мережі [12], експертні системи). На наступному етапі виконується сегментація тексту. Будемо проводити її в три етапи.

Виділення рядків — вихідне зображення тексту необхідно «розрізати» на смуги-рядки потрібної ширини.

Сегментація слів — в зображенні текстового рядка виділяємо зображення слів.

Сегментація символів — в зображенні слова проводимо кордони символів.

## 2.2 Розробка способу розпізнавання пошкоджених символів

На теперішній час відомо значна кількість способів розпізнавання символів текстового документу, що представлений у відсканованому вигляді.

Відомий спосіб комп'ютерного розпізнавання об'єктів за патентом РФ [32], відповідно з яким попередньо приводять зображення об'єкта, що вводиться в комп'ютер, до нормального, стандартного для даного способу стану шляхом зміни масштабу, повороту в необхідне положення, центрування, вписування в прямокутник необхідного розміру, на екран монітора виводять зображення об'єкта, що розпізнається, у перетворене в зображення, виконане в градаціях — різних ступенях яскравості одного кольору, наприклад червоного, і на нього послідовно, по чергово накладають зображення шаблонів, що зберігаються в пам'яті комп'ютера, виконаних, наприклад, в градаціях зеленого кольору, що дозволяє побачити в зоні перекриття зображень зображення іншого кольору, відмінного від перших двох кольору, яке і фіксують як розпізнане в разі тотожних, ідентичних, а значить мають однаковий контур зображень, що розпізнається, та шаблону. В принципі розпізнаваним об'єктом може бути цифровий або буквенний символ.

Відомий також спосіб автоматичного розпізнавання із використанням електронних засобів символів, таких як букви і або ж цифри, що віддруковані на будь-якому матеріалі, який включає структури із сильною контрастністю [33]. Цей спосіб включає в себе такі етапи: налагодження, формування моделей символів, розпізнавання, реєстрація моделі фону разом із фоном отриманого зображення, потім виділення моделі зареєстрованого фону із елементарного зображення фону, комбінування для кожного положення символу моделі букв і або ж цифр із елементарним зображенням відповідного фону, створення комбінованих моделей, зіставлення невідомих символів із комбінованими моделями, розпізнавання кожного невідомого символу як відповідного символу, комбінована модель якого найкращим чином накладається на нього відповідно до технології «порівняння із шаблоном».

Даний спосіб добре зарекомендував себе в том випадку, коли заздалегідь відомі вихідні параметри комбінації символів, тобто вид і розміри шрифту, кількість символів, і знайшов застосування в розпізнаванні номерів цінних паперів, зокрема грошей. Однак розпізнавання стає вкрай складним, якщо

мова йде про символи з невідомими параметрами шрифту і фону.

Відомий також спосіб розпізнавання номерних знаків транспортних засобів [34], який полягає в формуванні вихідного зображення передбачуваної області знаходження набору символів номерних знаків, виділення прямокутної області розміщення набору символів номерних знаків, виявленні у виділеній області зон знаходження окремих позицій символів і розпізнаванні окремих символів на виявлених позиціях. При цьому розпізнавання окремих символів здійснюється за допомогою набору нейронних мереж, що організовані в два каскади.

У способі [35] автоматизованого розпізнаванні ідентифікаційного номеру, що полягає у формуванні початкового зображення передбачуваної області знаходження ідентифікаційного номеру, в виділенні прямокутної області розміщення ідентифікаційного номеру, виявленні у виділеній прямокутній області зон знаходження окремих позицій ідентифікаційного номеру та розпізнаванні символів на виявлених позиціях цього номеру, відповідно попередньо сформоване вихідне зображення очищають за допомогою морфологічного фільтру, а після виявлення зон знаходження окремих позицій ідентифікаційного номеру визначають за збереженими фрагментами символів координати приблизних центрів кожної позиції і приблизний параметр шрифту, використаного в процесі нанесення ідентифікаційного номеру, виходячи з якого формують початковий набір шаблонів символів з приблизними шрифтом номеру, що розпізнається. Процес розпізнавання символів на виявлених позиціях номеру виробляють шляхом послідовного накладення шаблонів зі змінними параметрами шрифту символів на зони знаходження позицій номеру з циклічним зсувом шаблонів щодо знайдених приблизних центрів позицій і порівняльного аналізу, в ході якого для кожної позиції ідентифікаційного номеру виявляють шаблон, який має максимальне значення коефіцієнту збігу з фрагментами символу, що знаходиться на даній позиції.

Технічний результат полягає в оптимізації процесу підгонки шаблонів

до фрагментів символів ідентифікаційного номеру з невідомими завчасно параметрами. Підгонка здійснюється за рахунок зміни геометричних параметрів шрифту шаблонів і прецизійного пошуку оптимального місця їх розташування в зонах перебування символів. При цьому сам процес підгонки істотно полегшується за рахунок попереднього очищення зображення ідентифікаційного номеру від фону, завдяки чому фрагменти символів стають більш виділеними [36]. В результаті забезпечується достовірність розпізнавання за допомогою вибору тих шаблонів, з якими відбувається максимально можливий збіг фрагментів символів по кожній позиції ідентифікаційного номеру.

Тоді послідовність розпізнавання символів буде такою.

Отримуємо зображення, у ньому виконуємо фільтрацію від можливих завад (цей етап поліпшує сприйняття фону), переводимо зображення із отриманого кольорового у бінарний формат та виділяємо області, що відповідають сторінкам текстового документу. Наступним етапом є розподіл виділеної сторінки тексту на окремо взяті рядки (це можна назвати сегментацією рядків). Потім приступаємо до розподілу рядків на окремі слова (це так звана сегментація слів). Надалі переходимо до найбільш відповідальної частини роботи процесу розпізнавання — до поділу слів на окремі символи та розпізнавання цих символів.

Виконуємо поділ слів на окремі символи та проводимо їх нормалізацію до необхідних розмірів для виконання розпізнавання. Виконуємо етап розпізнавання символів із використанням нейронної мережі глибокого навчання. Якщо символ на деякій позиції не був розпізнаний через значну ступінь спотворення, то тоді переходимо до етапу розпізнавання із використанням порівняння із наявною базою шаблонних символів.

Визначається тип алфавіту, який був використаний на попередньому етапі розпізнавання (латиниця чи кирилиця). Визначається ширина деформованого символу по отриманому зображенню. Цей параметр є досить стабільним для конкретного представлення символів алфавіту. По значенню



цієї ширини вибирається набір символів із визначеного алфавіту. Проводиться послідовний перебір шаблонів із вибраної групи символів. При накладанні обчислюється коефіцієнт збігу для кожного примірника символу із даної групи. Це буде коефіцієнт збігу між отриманим та еталонним символом. Коефіцієнт збігу визначається шляхом логічного множення отриманого символу із кожним еталонним символом вибраної групи символів. Із виконанням операції логічного додавання до всіх пікселів отриманого силуету символу визначається коефіцієнт відхилення фону даного символу від отриманого фону зображення. Це буде коефіцієнт відхилення або ж відмінності отриманого фону символу від фону еталонного символу. Порівняння сум цих коефіцієнту збігу та коефіцієнту відхилення фону із кожним із еталонних символів дасть можливість вибрати із визначеної групи еталонних символів такий символ, який має найбільше значення із отриманої групи та є найбільш подібним до аналізованого символу. Це збільшує ймовірність розпізнавання символу. Запропонована послідовність розпізнавання символів лежить у основі програми розпізнавання символів текстових документів.

### 2.3 Розробка алгоритму розпізнавання символів

У основі алгоритму вибору та розпізнавання символу із пошкодженим силуетом лежить припущення про пошук найкращого збігу за кількістю пікселів із отриманим символом та послідовним перебором із найбільш ймовірними еталонними символами. Виконання пошуку максимально ймовірного символу матиме таку послідовність дій:

2.3.1 Визначаємо тип шрифту по попередніх визначених символах (латиниця чи кирилиця).

2.3.2 Визначаємо ширину інтервалу, яку займає даний символ.

2.3.3 Виконуємо звертання до відповідного набору символів що будуть використовуватись для розпізнавання.

2.3.4 Вибираємо перший символ із визначеного набору.

2.3.5 Визначаємо ступінь збігів по накладанню матриці еталонного символу із матрицею опису вибраного символу.

2.3.6 Підраховуємо кількість збігів для силуету символу у вибраному еталонному зображенні.

2.3.7 Підраховуємо кількість збігів для фону символу у вибраному еталонному зображенні.

2.3.8 Знаходимо сумарну кількість збігів для силуету та фону.

2.3.9 Запам'ятовуємо отримане значення.

2.3.10 Переходимо до вибору наступного еталонного символу.

2.3.11 Повторюємо дії пунктів по визначенню сумарної кількості збігів для вибраного нового еталонного символу.

2.3.12 Порівнюємо результати двох попередніх обчислень.

2.3.13 Фіксуємо більше значення із отриманого порівняння.

2.3.14 Переходимо до наступного еталонного символу.

2.3.15 Повторюємо дії пунктів по визначенню сумарної кількості збігів для вибраного нового еталонного символу та його порівняння із попереднім символом.

2.3.16 Перевіряємо, чи всі еталонні символи із вибраного набору символів пройшли операцію порівняння.

2.3.17 Виводимо силует символу, який отримав найбільшу кількість збігів при порівнянні еталонних символів із аналізованим символом.

2.3.18 Кінець алгоритму.

2.3.19 Запропонований алгоритм лежить у основі створення програмного засобу по розпізнаванню символів із пошкодженим силуетом, який розглянутий у наступному розділі даної магістерської роботи. Визначення коефіцієнтів збігу символу зображення та фону зображення дозволяє збільшити достовірність розпізнавання пошкоджених символів. Зображення розробленого алгоритму наведено у додатку Д даної роботи.

2.4 Згорткова нейронна мережа

На теперішній час відомі різні типи штучних нейронних мереж. Це Нейронні мережі зберігання та вибірки великої пам'яті, згорткові нейронні мережі та інші. Досить широке застосування останнім часом отримали згорткові нейронні мережі.

Згорткова нейронна мережа (ЗНМ), це штучна нейронна мережа із спеціальною архітектурою та будовою, що виконує ефективне розпізнавання об'єктів цифрових зображень та відноситься до складу технологій глибокого навчання. ЗНМ відносяться до типу багат шарової нейронної мережі, яка свою назву «згорткова мережа» отримала через назву виконання операції згортки. У згортковому шарі мережі над зображеннями, які надходять на вхід цього шару, виконується операція згортки шляхом знаходження зваженої суми фрагментів зображення навколо заданого ядра згортки:

$$(f \cdot g)[m, n] = \sum_{k, l} (f[m - k, n - l] \cdot g[k, l]) \quad (2.1)$$

де  $f$  — початкова матриця зображення;

$g$  — власно ядро згортки;

$k$  — номер рядка ядра згортки;

$l$  — номер стовпця ядра згортки [11].

Принцип роботи таких ЗНМ полягає у послідовному чергуванні шарів згортки (англійською *convolution layers*) та шарів субдискретизації (*subsampling layers* або інша назва *pooling layers*). Структура такої мережі є однонаправленою без введення зворотних зв'язків та її основі покладено принцип багат шаровості. При навчанні ЗНМ використовується метод зворотного поширення помилки.

У основі роботи ЗНМ лежить принцип переходу від конкретних особливостей аналізованого зображення до виявлення більш абстрактних деталей та понять більш високого рівня. Зазвичай ЗНМ самоналаштовується та формує необхідну ієрархію абстрактних ознак, відкидаючи незначні деталі та виділяючи суттєве.

Робота ЗНМ забезпечується двома основними елементами, у ролі яких

виступають фільтри (filters) для виявлення певних ознак та карти ознак (feature maps), що описують аналізоване зображення.

Модель ЗНМ складається із трьох типів шарів: згорткових (convolutional) шарів, субдискретизуючих (subsampling) шарів та прошарку «звичайної» нейронної мережі у вигляді перцептрона.

Перцептроном є такий штучний нейрон, який фактично представляє повнозв'язану нейронну мережу, у якій кожний наступний нейрон мережі пов'язаний із усіма нейронами попереднього шару, у якому такі зв'язки мають свої вагові коефіцієнти. У такій мережі із нейронів операцію згортки здійснює невеличка матриця ваг, яка переміщається по всьому вхідному зображенню, що подане на ці нейрони, та формує сигнал активації після виконання кожного зсуву для нейрону наступного шару із аналогічною позицією. Ця матриця ваг для різних нейронів вхідного шару використовується як певне ядро згортки. Її представлення є графічним кодуванням певної ознаки зображення. Наступний шар, який формується в результаті цієї операції згортки матрицею ваг, вказує на наявність певної визначеної ознаки в шарі, що обробляється, та вказує її координати, таким чином формуючи карту ознак (англійською Feature map).

У ЗНМ набір ваг не один, а їх ціла гама, що кодують елементи аналізованого зображення (наприклад лінії під різними кутами та дуги). Ці ядра згортки є унікальними для даної мережі, тому що вони формуються методом зворотного навчання, а не задаються вручну людиною-оператором. При проходженні ЗНМ набором ваг формується набір ознак, що робить цю нейромережу багатоканальною. При обробленні шару матрицею ваг її пересувають як правило не на розмір самої матриці, а на певну невелику відстань. Наприклад, якщо матриця ваг має розмірність  $5 \times 5$ , то її переміщують на один або ж два нейрони замість п'яти, щоб не втратити та пропустити шукані ознаки.

Операція субдискретизації (Subsampling або Pooling) використовується для зменшення розмірів сформованих карт ознак. Після виконання цієї процедури формується рішення про те, чи є шукана ознака у даному фрагменті

зображення, що є важливішим точного знання її координат, і тому із сусідніх нейронів із їх карт ознак вибирається максимальне значення та це приймається за один нейрон тепер уже ущільненої карти ознак зменшеної розмірності. Ця операція дозволяє пришвидшити обчислення та додатково мережа стає більш нечутливою до масштабу отриманого зображення.

Типова структура ЗНМ складається із великої кількості шарів. Вхідне зображення подається на початковий шар, а надалі сигнал проходить серію згорткових шарів, в яких відбувається згортка та субдискретизація. Почергове проходження шарів дозволяє сформувати карти ознак, у кожному наступному шарі ця карта зменшується у розмірі, але водночас при цьому зростає кількість каналів. Після проходження деякої кількості шарів карта ознак перетворюється у скаляр або ж вектор, але таких карт ознак у ЗНМ формуються сотні. Після проходження таких загорткових шарів додатково вводяться декілька шарів перцептронів (повнозв'язаної нейронної мережі), на виході яких подаються сформовані кінцеві карти ознак [37]. На рисунку 2.1 приведена типова архітектура згорткової нейронної мережі.

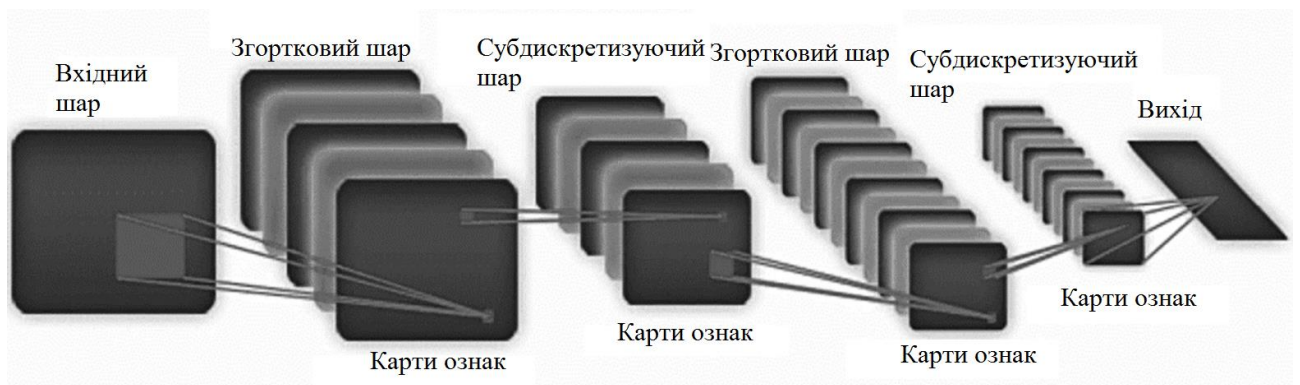


Рисунок 2.1 — Структура згорткової нейронної мережі

Ядро згортки мережі є певним набором вагових коефіцієнтів. У підсумку виконання даної операції отримуємо зображення у вигляді карти ознак. Залежно від обраної матриці для виконання операції згортки карта ознак виділятиме певні характеристики оброблюваного вхідного зображення. Щоб отримати найкращі характеристики вхідного зображення використовують кілька різних ядер згортки, і тому на виході згорткового шару мережі

формуються кілька різних карт ознак. Ядро згортки фактично є фільтром або ж вікном, яке переміщається по усій попередній карті та виділяє певні ознаки об'єктів.

Шар субдискретизації у мережі представляє собою операцію нелінійного ущільнення (нелінійне перетворення) отриманої карти ознак, при цьому певна група пікселів (як правило розміром  $2 \times 2$  пікселів) ущільнюється до одного пікселя. Найчастіше для здійснення цього перетворення використовують функція знаходження максимуму. Для перетворення виділять прямокутники або квадрати, які не перекриваються, і кожен із яких замінюється одним пікселем, в якості якого вибирають піксель із максимальним значенням у аналізованому фрагменті зображення. Операція субдискретизації дозволяє у підсумку суттєво зменшити просторовий розмір аналізованого зображення. Цю операцію можна трактувати таким чином: якщо на попередньому етапі виконання згортки були виявлені певні ознаки зображення, то для наступного етапу обробки таке зображення уже не потрібно, і виконується операція ущільнення зображення до менш детального. До того ж виконання операції фільтрації тепер уже непотрібних деталей зображення допомагає мережі не перенавчатися. Шари згортки та субдискретизації чередуються і шар субдискретизації вставляється після попереднього шару згортки перед наступним шаром виконання згортки.

Після кількох таких проходжень згортки зображення та його ущільнення за допомогою виконання операції субдискретизації мережа переформатовується від конкретної сітки пікселів із високою роздільною здатністю до більш абстрактних сформованих карт ознак, та на кожному наступному шарі зростає число каналів та зменшується розмірність аналізованого зображення. У підсумку отримуємо великий набір вихідних каналів, що зберігають тепер уже невелику кількість даних або ж навіть тільки один параметр, які інтерпретуються як самі абстрактні поняття, що виявлені із початкового аналізованого зображення.

Ці отримані дані об'єднуються та передаються на звичайну

повнозв'язану нейронну мережу, яка теж може складатися із декількох шарів. При цьому повнозв'язні шари вже втрачають просторову структуру пікселів та мають невелику розмірність по відношенню до кількості пікселів початкового зображення.

## 2.5 Використання бібліотек із відкритим кодом

Система Open Computer Vision Library, скорочено OpenCV, позначає відкриту бібліотеку комп'ютерного зору, яка має реалізуватись в реальному часі [38]. OpenCV може використовуватися для розв'язання задач взаємодії на рівні «людина-комп'ютер»; для спостереження рухів, відслідковування переміщення камери, їхнє структурування та розуміння; для ідентифікації об'єктів, їхньої сегментації та розпізнавання, у тому числі для розпізнавання облич і жестів. Однією із перспективних галузей використання бібліотеки OpenCV є робототехніка, у тому числі і мобільні роботи.

Основними модулями бібліотеки є наведені нижче модулі (рис.2.2). Ядро `sxcore` виконує наступні операції:

- підтримку матричної алгебри, генераторів випадкових чисел DFT, DST, математичних функцій;
  - базові операції над багатовимірними числовими масивами;
  - підтримку базових функцій 2D графіки;
  - підтримку більш складних структур даних: динамічно зростаючих послідовностей, графів, розріджених масивів;
  - запис та відновлення структур даних в / з XML / YAML форматів;
- Ще один модуль CV — це модуль для оброблення зображень і виконання операцій комп'ютерного зору. Його основними функціями є:

- виконання базових операцій над зображеннями (геометричні перетворення, перетворення колірних просторів, фільтрація і т. д.)
- структурний аналіз (плоскі розбиття, опис форм);
- аналізування зображень (пошук контурів, гістограми, вибір відмінних ознак, морфологія);

- аналіз руху, стеження за об'єктами;
- виокремлення об'єктів, зокрема осіб;
- калібрування камер;

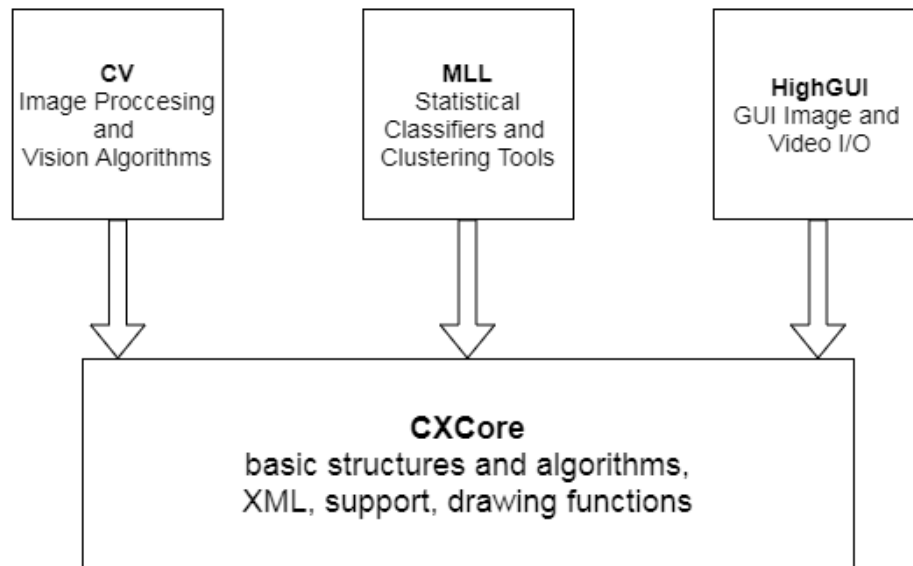


Рисунок 2.2 — Архітектура бібліотеки OpenCV

HighGUI є модулем для введення та виведення зображень та відео, а також для створення користувацького інтерфейсу. Він необхідний для виконання наступних операцій:

- захоплення відео із камер та із відео файлів, читання та запис статичних зображень;
- функції для організації зрозумілого та простого інтерфейсу користувача (на даний час більшість додатків використовують HighGUI).

Модуль Svaux містить експериментальні та деякі застарілі функції. За допомогою даного модулю виконується набір таких операцій:

- просторовий зір: стерео калібрація, самокалібрація;
- знаходження та опис рис обличчя людини;
- пошук стерео-відповідності, кліки у графах;
- приховані Марківські ланцюги;
- опис текстур;
- порівняння форм та побудова скелетонів.



Із програмної точки зору бібліотека OpenCV є набором низьковитратних, та водночас, високопродуктивних операцій по обробці пікселів на зображеннях. Вона реалізує алгоритми калібрування камери; відслідковування ознак; виявлення та формування ознак; тривимірної реконструкції об'єктів; їхньої сегментації та розпізнавання; аналізу форм; аналізу рухів.

Для реалізації своїх властивостей бібліотека оперує набором типів даних, зокрема дані типу масивів: CvMat (матриця), IplImage (зображення), CvSet (множина), CvGraph (граф), CvSeq (послідовність); змішані типи, як приклад CvHistogram; допоміжні типи даних: CvPoint (2D-точка), CvTermCriteria (критерій закінчення ітеративних процесів), CvSize (розміри); CvMoments (просторові моменти); IplConvKernel (ядро згортання) [39].

Бібліотека OpenCV головним чином використовується на операційній системі MS Windows, хоча також існують і реалізації бібліотеки для інших ОС, зокрема Linux, iOS та інших.

Існує також бібліотека із відкритим кодом AForge.NET, яка створена на основі мови програмування C# та призначена для роботи у області комп'ютерного зору. Спектр роботи цієї бібліотеки достатньо широкий: обробка зображень, нейронні мережі, машинне навчання, нечітка логіка та інше [40]. Бібліотека складається із ряду компонентів, які дозволять працювати у певному напрямку, як наприклад AForge.Neuro — бібліотека для роботи та виконання дій та різних операцій із нейронними мережами. Але головним недоліком цієї бібліотеки є обмежена сфера її застосування, що пов'язано із представленням її документації англійською мовою та невелика спільна по її використанню.

Ще одна бібліотека LTI або LTI-lib є об'єктно-орієнтованою бібліотекою алгоритмів та структур даних [41]. Цю бібліотеку часто використовують для обробки зображень та у сфері комп'ютерного зору. Найважливішою сферою її застосування є комп'ютерний зір для технологій робототехніки, розпізнавання об'єктів, голосу та жестів. Головною задачею цієї бібліотеки є створення об'єктно-орієнтованою бібліотекою на мові C++ для спрощення

використання коду програм та забезпечення швидких алгоритмів, які можна було використати у реальних додатках.

Інша бібліотека VXL є набором бібліотек, написаних мовою C++ та призначених для реалізації технологій комп'ютерного зору та наукових досліджень [42]. Бібліотека була написана у ANSI/ISO C++ та призначена для роботи із портативними платформами. До складу бібліотеки входять такі складові: VNL — робота із чисельними алгоритмами та контейнерами; VIL— робота із зображеннями; VGL — геометрія точок, кривих та інших елементарних об'єктів у двомірних та тривимірних просторах та інші. Але набір модулів цієї бібліотеки не є достатнім для вирішення наших задач.

Підводячи підсумок, із розглянутих бібліотек найбільш оптимальним буде використання бібліотеки OpenCV, так як її швидкодія є досить високою, до її складу входить значна кількість функцій, що можуть бути використанні для розпізнавання тексту та оброблення зображення документу у цілому. Тому для створення програмного продукту по розпізнаванню текстових документів зупиняємося на використанні бібліотеки.

## 3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ

Розпізнавання окремо взятих та виділених символів сканованих текстових документів здійснюється за ряд етапів, послідовність виконання яких та їх програмна реалізація розглянуті у даному розділі.

### 3.1 Вибір інструментальних засобів програмування

Для реалізації вище запропонованих підходів для розпізнавання друкованих символів було вибрано мову програмування Python. Вибір цієї мови розробки для створення даного програмного продукту зумовлений тим, що це досить зручна функціональна мова, що відмінно підходить для реалізації складних математичних операцій. Мова Python є високорівневою мовою програмування загального призначення, що орієнтована на підвищення продуктивності розробника та читабельність коду. Синтаксис ядра Python є мінімалістичним. Однак в той же час стандартна бібліотека містить великий обсяг корисних функцій. Мова Python підтримує структурне, об'єктно-орієнтоване, імперативне, аспектно-орієнтоване і функціональне програмування. Основними архітектурними рисами є повна інтроспекція, динамічна типізація, механізм обробки виключень, автоматичне керування пам'яттю, високорівневі структури даних, підтримка багатопоточних обчислень. У мові підтримується розбиття програми на окремі модулі, що, в свою чергу, можна об'єднувати у пакети.

Інтерпретатор CPython є еталонною реалізацією Python, який підтримує більшість платформ, що активно використовуються на даний час [43]. Він вільно поширюється із ліцензією Python Software Foundation License, яка дозволяє використовувати його без ніяких обмежень у будь-яких додатках, включаючи пропрієтарні. Python є портованим та працює майже на усіх відомих платформах — від мейнфреймів до кишенькових персональних комп'ютерів. Існують версії під Microsoft Windows, практично під усі варіанти UNIX-подібних систем (включаючи FreeBSD та Linux), Windows Mobile,

Symbian, а також Android.

Однак, на відміну від багатьох інших портованих систем, Python має підтримку особливих для даної платформи технологій (як приклад, Microsoft COM / DCOM). Окрім того, існує Jython — це спеціальна версія Python для віртуальної машини Java, яка дозволяє інтерпретатору запускатися на будь-якій системі, яка підтримує Java. Разом з цим класи Java можуть безпосередньо використовуватися із Python та навіть бути написаними на мові Python. При цьому код буде компілюватися та виконуватися на java-машині (JVM), що дозволяє використовувати Python-код із Java-програм. Розроблений прототип системи розпізнавання текстів, що містять спотворені символи, є призначеним для демонстрації роботи алгоритму розпізнавання вмісту пошкоджених текстів. Програмний засіб реалізовано на високорівневих мовах програмування Java та Python.

Основними перевагами мови програмування Python є такі аспекти.

Це добре спроектована мова програмування, яка вмістила у собі сучасні тенденції у програмуванні «із нуля». Окрім того, мова програмування Python динамічно розвивається, у ній процес додання нових конструкцій в мову добре налагоджений, та продовжує вбирати у себе прийоми аспектно-орієнтованого програмування функціонального програмування та іншого, при цьому залишаючись назад-сумісним та внутрішньо несуперечливим. Крім того, мова Python легка у виконанні програмування.

У мові Python використовується синтаксис, який легко читається (у порівнянні із C ++, PHP) та дозволяє з легкістю читати чужий код, вільно розбиратися у давно написаному власному коді.

Дуже велика кількість бібліотек із кодом на будь-які випадки (наприклад, робота з зображеннями, таблицями Excel або мережею Twitter).

Мова Python має значну переносимість та може працювати із усіма поширеними на теперішній час операційними системами та на значній кількості архітектур — Windows, MacOS, Linux, а також на міні-комп'ютерах Arduino. Система залежностей досить добре продумана, розгортання додатків

на іншій системі відбувається легко та без ніяких проблем.

Програми на Python багато у чому схожі на Java-програми, та можуть вільно взаємодіяти із Java-кодом. Програми Java зазвичай транслюються у спеціальний байт-код, саме тому вони, за допомогою віртуальної Java-машини, працюють на будь-якій комп'ютерній архітектурі.

Перевагою такого способу виконання програм є абсолютна незалежність байт-коду від операційної системи та обладнання, що дозволяє виконувати Java-код на будь-якій системі, для якої існує відповідна віртуальна машина. Ще одна важлива особливість технології Java — це гнучка система безпеки, у рамках якої виконання додатку повністю контролюється віртуальною машиною. Якщо будь-які операції, що перевищують встановлені дозволи програми (для прикладу, з'єднання з іншим комп'ютером або спроба несанкціонованого доступу до даних) викликають негайне переривання виконання програми. До недоліків концепції віртуальної машини можна віднести значне зниження продуктивності роботи системи.

Ряд удосконалень у цій мові дещо збільшили швидкодію виконання програм на Java:

- використання технології трансляції байт-коду у машинний код безпосередньо під час виконання програми (JIT-технологія) із можливістю збереження версій класу у машинному коді;

- широке застосування платформи-орієнтованого коду (native-код) у стандартних бібліотеках;

- використання апаратних засобів, що забезпечують пришвидшену обробку байт-коду [44].

У Python вбудовані потужні та одноманітні концепції абстракцій як для типів даних, так і для значень. Окрім того, мова має гнучкі симетричні конструкції домішок для композиції типажів та класів. Це дає можливість виробляти декомпозицію об'єктів шляхом їх порівняння із зразком; зразки та вирази при цьому були узагальненими для підтримки природної обробки XML-документів. В загальному, ці конструкції дозволяють легко формувати

самостійні компоненти, що використовують бібліотеки Python, при цьому, не використовуючи спеціальні мовні конструкції.

Також, у даній системі використовується бібліотека OpenCV, що містить алгоритми машинного навчання, у якості допоміжного засобу для попередньої обробки отриманих зображень.

### 3.2 Розробка архітектури програми розпізнавання

Структура програми по розпізнаванню друкованих символів буде складатися із трьох основних та одного додаткового модулів: модуль попередньої обробки зображення, модуль розпізнавання символів із використанням нейронної мережі та модуль для остаточного розпізнавання символів із деформованою структурою. Крім того, буде ще один окремий четвертий модуль для налаштування нейронної мережі на виконання процесу розпізнавання символів.

У модулі попередньої обробки зображення будуть виконуватися такі функції. Отримуємо зображення друкованого тексту, у ньому виконуємо фільтрацію від можливих завад та морфологічну фільтрацію, переводимо зображення із отриманого кольорового у бінарний формат та послідовно виділяємо області, що відповідають сторінкам текстового документу. Надалі виконуємо розподіл виділеної сторінки тексту на окремо взяті рядки із використанням гістограм розподілу яскравості зображення по вертикалі. Потім приступаємо до розподілу рядків на окремі слова із використанням гістограм розподілу яскравості виділеного рядка по горизонталі. Надалі переходимо до поділу виділених слів на окремі символи та визначення їх місця розташування. Після виконання поділу слів на окремі символи проводимо їх нормалізацію до необхідних розмірів для подальшого виконання процесу розпізнавання із використанням нейронної мережі. Процес виконання розпізнавання символів здійснюємо із підключенням нейронної мережі глибокого навчання. Якщо символ на деякій позиції не був розпізнаний через значну ступінь спотворення, то тоді переходимо до роботи наступного модуля

програми: етапу розпізнавання із використанням порівняння деформованого символу із наявною базою шаблонних символів.

При роботі цього модуля програми визначається тип алфавіту, який був використаний на попередньому етапі розпізнавання (латиниця чи кирилиця). Визначається ширина деформованого символу по отриманому зображенню. По значенню цієї ширини вибирається набір символів із визначеного алфавіту. Проводиться послідовний перебір шаблонів із вибраної групи символів. При накладанні обчислюється коефіцієнт збігу для кожного примірника символу із даної групи. Це буде коефіцієнт збігу між отриманим та еталонним символом. Коефіцієнт збігу визначається шляхом логічного множення отриманого символу із кожним еталонним символом вибраної групи символів. Таким же чином визначається коефіцієнт відхилення фону даного символу від отриманого фону зображення. Це буде коефіцієнт відмінності отриманого фону символу від фону еталонного символу. Порівняння різниць цих коефіцієнтів дасть можливість вибрати із визначеної групи еталонних символів такий символ, який має найбільше значення із отриманої групи та є найбільш подібним до аналізованого символу.

Модуль для налаштування нейронної мережі на виконання процесу розпізнавання символів для своєї роботи використовує еталонну базу даних символів EMNIST, яка у своєму складі має 64 різних символи ( А, В, С, ..., 0, 1 та інші) [45].

Запропонована послідовність розпізнавання символів приведена у додатку Д даної магістерської роботи.

### 3.3 Розробка програми розпізнавання символів

Розділення тексту на окремі букви є одним із кроків роботи програми по розпізнаванню символів. Для цього скористаємося відкритою бібліотекою OpenCV та однією із її функцій `findContours`.

Для виділення тексту із зображення отримуємо зображення у форматі `png` та відкриваємо його із використанням стандартної процедури (`cv2.imread`)

із бібліотеки OpenCV [38]. Потім переводимо його у градації сірого кольору та виконуємо процес бінаризації зображення із використанням процедури `cv2.cvtColor` у моделі кольорів RGB та змінюємо розміри (`cv2.erode`) і виконуємо пошук контурів символів за допомогою `cv2.threshold`.

```
image_file = "text.png"
img = cv2.imread(image_file)
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY)
img_erode = cv2.erode(thresh, np.ones((3, 3), np.uint8), iterations = 1)
```

Виконуємо операцію по послідовному виділенню контурів спочатку сторінок, потім рядків та слів та окремо взятих символів.

```
contours, hierarchy = cv2.findContours(img_erode, cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)
```

```
output = img.copy()
for idx, contour in enumerate(contours):
    (x,y,w,h) = cv2.boundingRect(contour)
```

Здійснюємо процедуру формування ієрархії контурів по їх складності.

```
if hierarchy[0][idx][3] == 0:
    cv2.rectangle(output, (x, y), (x+w, y+h), (80, 0, 0), 1)
cv2.imshow("Input", img)
cv2.imshow("Enlarged", img_erode)
cv2.imshow("Output", output)
cv2.waitKey(0)
```

У підсумку отримуємо ієрархічне дерево контурів (параметр `cv2.RETR_TREE`). Спочатку йде контур загальної картини зображення, потім контур слів, а потім контури окремих символів. Це фактично виконана сегментація текстової сторінки до окремо взятого символу. Для нашого процесу розпізнавання необхідно зупинитися на контурах окремих символів, тому і виділяємо цей рівень представлення.

Наступним кроком буде збереження кожної із виділених букв текстового



документу у масштабі 28x28 біт, так як у такому масштабі працює база еталонних символів EMNIST [45] та нейронна мережа для розпізнавання символів. Також розставляємо координати виділених символів по осі X, щоб надалі із виділених координат виділити пропуски між символами.

```
def letters_extract(image_file: str, out_size=28) -> List [Any]:
```

Змінюємо розмір отриманого фрагменту зображення до розміру квадрату.

```
size_max = max (w, h)
```

```
letter_square = 255 * np.ones (shape= [size_max, size_max], dtype=
np.uint8)
```

```
if w > h:
```

Збільшуємо розмір зображення зверху до низу зліва на право. Потім виконуємо сортування отриманого масиву за X координатою

```
letters.sort (key= lambda x:x[0], reverse=False)
```

```
return letters
```

Після виконання сортування символів по X координаті зберігаємо їх у вигляді tuple (x, w, letter), щоб потім із сформованих відстаней виділити пропуски. Перевіряємо правильність роботи.

Використовуємо нейронну мережу для розпізнавання символів. Для цього скористаємося базою даних символів EMNIST, яка у своєму складі має різні символи (A, B, C, ..., 0, 1 та інші).

Модуль навчання створеної нейронної мережі функціонує таким чином. Нейронна мережа має відповідно виходи та на вході оперує фрагментом зображення розміром 28x28 біт, та на одному із виходів формує значення логічної 1 у випадку знаходження та розпізнавання відповідного символу. Створюємо модель нейронної мережі.

```
from tensorflow import keras
```

```
from keras import optimizers
```

```
from keras.models import Sequential
```

```
from keras.layers import Convolution2D, Max Pooling2D, Dropout, Dense,
```

## Reshape, Flatten, LSTM, Batch Normalization

```

from keras import backend as K
from keras.optimizers import SGD, RMSprop, Adam
from keras.constraints import maxnorm
import tensorflow as tf
def emnist_model():
    model = Sequential()
    model.add(Convolution2D(filters=32, kernel_size=(3, 3), padding=
'valid', input_shape=(28, 28, 1), activation='relu'))
    model.add(Convolution2D(filters=64, kernel_size=(3, 3), activation =
'relu'))
    return model

```

Це класична згорткова нейронна мережа, яка виділяє певні ознаки виділеного зображення із використанням фільтрів розміром 32 та 62, та до виходу якої під'єднана лінійна мережа MLP, яка формує остаточний результат.

Тепер переходимо до етапу навчання мережі. Тут скористаємося базою даних EMNIST. Для читання даних із цієї бази та її роботи скористаємося бібліотекою `idx2numpy`, та підготуємо необхідні дані для навчання та подальшої перевірки нейронною мережею.

```

import idx2numpy
emnist_path = '/home/Documents/TestApps/keras/emnist/'
X_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
train-images-idx3-ubyte')
y_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
train-labels-idx1-ubyte')
X_test = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-test-
images-idx3-ubyte')
y_test = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-test-
labels-idx1-ubyte')

```

Виконуємо нормалізацію зображення. Для навчання мережі

підготовлені два набори даних, символи у яких представлені у вигляді звичайних масивів, Для навчання використовується частина бази даних.

Для виконання операції розпізнавання завантажуюмо створену модель та визиваємо функцію `predict_classes`.

```
model = keras.models.load_model('emnist_letters.h5')
def emnist_predict_img(model, img):
    img_arr = np.expand_dims(img, axis=0)
    img_arr = 1 - img_arr/255.0
    img_arr[0] = np.rot90(img_arr[0], 3)
    img_arr[0] = np.fliplr(img_arr[0])
    img_arr = img_arr.reshape((1, 28, 28, 1))
    result = model.predict_classes([img_arr])
    return chr(emnist_labels[result[0]])
```

Завершальна функція матиме вигляд:

```
def img_to_str(model: Any, image_file: str):
    letters = letters_extract(image_file)
    s_out = ""
    for i in range(len(letters)):
        dn = letters[i+1][0] - letters[i][0] - letters[i][1] if i < len(letters) - 1 else
0
        s_out += emnist_predict_img(model, letters[i][2])
        if (dn > letters[i][1]/4):
            s_out += ' '
    return s_out
```

Отримуємо матрицю опису нерозпізаного символу та вибираємо опис першого еталонного символу із вибраної групи еталонних символів.

Виконуємо операції логічного множення та логічного додавання елементів цих матриць.

```
for i in range(len(S)):
    for j in range(len(S[0])):
```

```

MZ[i][j] = S[i][j] logical_and E[i][j]
for i in range(len(S)):
for j in range(len(S[0])):
    MF[i][j] = S[i][j] logical_or E[i][j]

```

У отриманій матриці здійснюємо підрахунок кількості збігів отриманого символу із еталонним зображенням

```

MZ[i][j] = 'MZ'
S1 = 0
for i in range(N):
for j in range(M):
    if (MZ[i][j] == 1):
        S1 += 1

```

Аналогічну операцію виконуємо для визначення кількості відхилень фону отриманого символу від фону еталонного зображення

```

MF[i][j] = 'MF'
S0 = 0
for i in range(N):
for j in range(M):
    if (MF[i][j] == 0):
        S0 += 1

```

Знаходимо суму кількості збігів та відхилень.

```
S = S1 + S0
```

Визиваємо наступне еталонне зображення та проводимо такі ж самі обчислення. Порівнюємо значення отриманих сум та вибираємо більше. Фіксуємо порядковий номер отриманого еталонного зображення та переходимо до порівняння із наступним еталонним зображенням.

Такі операції виконуємо із усіма еталонними зображеннями вибраної групи еталонів. У результаті отримуємо еталонне зображення, яке має максимальну кількість збігів із отриманим зображенням. Це і буде результат розпізнавання символу. Розроблений варіант програми розпізнавання

символів текстового документа приведений у додатку Б магістерської роботи.

### 3.4 Перевірка якості роботи програми розпізнавання

Для оцінювання якості систем та алгоритмів розпізнавання тексту використовуються такі категорії як точність та достовірність. Точність роботи системи розпізнавання тексту показує ймовірність вірного розпізнавання вибраного об'єкту. Однак дійсне значення ймовірності розпізнавання у загальному випадку визначити досить складно і у такому випадку точність розпізнавання визначається апостеріорною ймовірністю вірного розпізнавання символів тексту для деякого набору вхідних текстових даних. При такому підході завдання розпізнавання тексту може бути сформульованою як задача максималізації апостеріорної ймовірності правильного розпізнавання символів із вибраного набору текстових даних.

Ще одним показником якості роботи системи розпізнавання є достовірність або надійність розпізнавання. Оцінка достовірності роботи системи розпізнавання відображає її здатність апріорно оцінювати ступінь точності власного отриманого результату.

Для оцінки якості розпізнавання скористаємося коефіцієнтом помилок  $K_s$  у символах, який визначається таким чином

$$K_s = \frac{b_s + e_s + d_s}{c}, \quad (3.1)$$

де  $b_s$ ,  $e_s$  та  $d_s$  відповідно кількість помилково вставлених, заміщених та видалених символів;

$c$  — кількість символів в еталоні [46].

Точність розпізнавання  $K_{sl}$  у словах можна представити таким виразом (3.2)

$$K_{sl} = \frac{b_w + e_w + d_w}{w}, \quad (3.2)$$

де  $b_w$ ,  $e_w$  та  $d_w$  відповідно кількість помилково вставлених, заміщених та видалених слів;

$w$  — кількість слів в еталонному зразку [46].

Відповідно точність розпізнавання  $T_s$  у символах визначається як:

$$T_s = 1 - K_s. \quad (3.3)$$

З іншої сторони коефіцієнт помилок можна обчислити через визначення мінімальної кількості операцій, які необхідно виконати, щоб привести отриманий набір символів до еталонного набору. Таке число має назву відстані по Левенштейну. Якщо у отриманій послідовності символів для доведення її до еталонного значення необхідно ще й виконати перестановку деяких символів, то тоді сумарна кількість всіх операцій має назву відстані по Дамерау-Левенштейну [46]. У нашому випадку скористаємося виразом (3.3), що визначає точність розпізнавання у символах.

Для створення навчальної вибірки були використані шрифти Arial та Times New Roman. До вибірки були включені букви латинського та кириличного алфавіту (за винятком букви й), розділові знаки та цифри. Всього було використано 340 символів.

Для генерації перевіряючої послідовності були також залучені інші шрифти, такі як Courier, Tahoma та Verdana. Всього також було використано 340 символів.

У результаті було розпізнано 319 (93,8%) символів із 340 використаних. Не були розпізнані всього 17 образів символів, до яких потім був застосований шаблонний метод розпізнавання. Частина символів було визначена невірно через похибку із-за визначення до якого регістру відноситься даний символ. Це можна пояснити використанням нормалізації до приведення символів до стандартного розміру, у результаті якої символи стають подібними. Детальний аналіз неправильно розпізнаних символів вказує на те, що більшість із них припадає на шрифт Impact. Це можна пояснити тим, що символи цього шрифту суттєво відрізняються по своїй формі від символів інших використаних шрифтів, які використовувалися для формування нейронної мережі.

У випадку, коли були вилучені символи шрифту Impact, вдалось розпізнати значно більшу кількість символів. По результатах розпізнавання

256 образів некоректно було розпізнано тільки 12 символів (4,7%).

Найбільшу кількість спотворень показав по розміру символів показав текст, що був набраний шрифтом Courier, найменшу кількість спотворень показав шрифт Times New Roman. Це дозволяє зробити висновок, що розпізнавання тексту, набраного шрифтом, що був внесений до навчальної вибірки, буде найбільш вірним. Інші шрифти будуть розпізнані, якщо їх символи будуть у незначній мірі відрізнятися від шрифтів, які були віднесені до початкової вибірки нейронної мережі.

Для розпізнавання спотворених символів скористаємося підготовленим набором символів, у яких кількість спотворень буде різною та визначеною у відсотках із використанням виразів (3.1) та (3.3) по відношенню до вірного представлення символів. Результати такого експерименту приведені на рисунку 3.1.

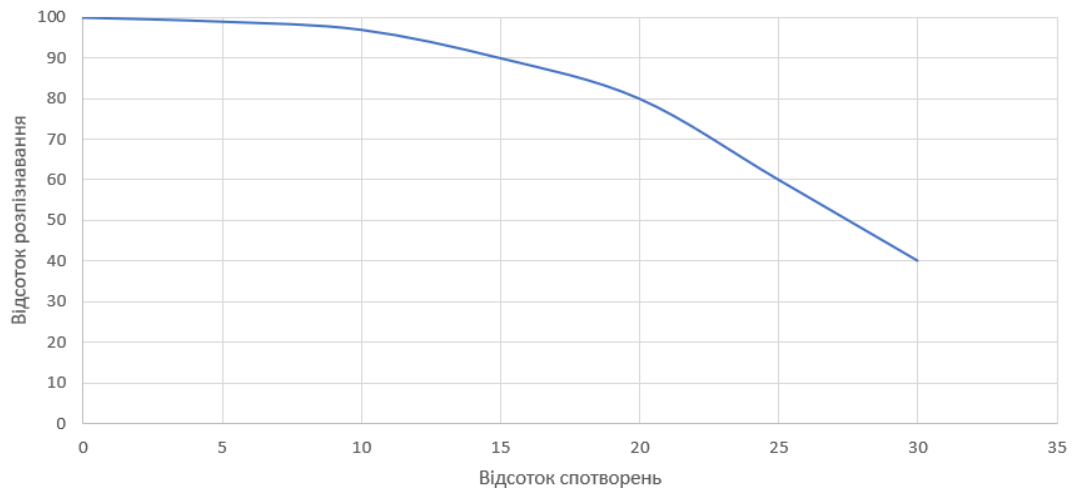


Рисунок 3.1 — Графік залежності відсотку розпізнавання символу від його спотворення

Як представлено, по наявності спотворень більше 30%, розпізнавання символів стає практично неможливою.

Виконані експериментальні дослідження дозволяють зробити висновок, що розроблений програмний продукт по розпізнаванню символів із певними спотвореннями може використовуватися у різного типу системах переведення текстових документів у електронну форму.





## **4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ**

Метою економічної частини магістерської кваліфікаційної роботи є довести економічну доцільність та ефективність впровадження розробки «Програмний засіб розпізнавання пошкоджених друкованих текстів із використанням багаторівневого аналізу документа», для цього необхідно виконати такі етапи:

- оцінити комерційний потенціал розробки;
- спрогнозувати витрати на виконання наукової роботи та впровадження її результатів;
- спрогнозувати комерційний ефект від реалізації результатів розробки;
- розрахувати ефективність вкладених інвестицій та період їх окупності.

### **4.1 Оцінювання економічного потенціалу розробки**

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Об'єктом дослідження магістерської кваліфікаційної роботи є розробка «Програмний засіб розпізнавання пошкоджених друкованих текстів із використанням багаторівневого аналізу документа».

Для проведення технологічного аудиту залучено трьох незалежних експертів: к.т.н., доц. кафедри ОТ Крупельницький Л.В., к.т.н., доц. кафедри ОТ Богомолів С.В., к.т.н. доц. кафедри ОТ Городецька О.С. Кожен з експертів повинен ознайомитися з запропонованою розробкою, та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Оцінювання комерційного потенціалу розробки здійснюється за критеріями, наведеними в таблиці 4.1.

Таблиця 4.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри- тері й	0	1	2	3	4
<b>Технічна здійсненність концепції:</b>					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
<b>Ринкові переваги (недоліки):</b>					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
<b>Ринкові перспективи</b>					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Кінець таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тері й	0	1	2	3	4
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу наведено в табл. 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта		
	Крупельницький Л.В.	Богомолов С.В.	Городецька О.С.
	Бали, виставлені експертами:		
1	4	3	3
2	3	2	2
3	4	3	3
4	4	4	3
5	3	2	3
6	2	2	4
7	2	4	3
8	3	2	2
9	1	1	3
10	4	4	4
11	4	3	3
12	4	4	4
Сума балів	СБ <sub>1</sub> =38	СБ <sub>2</sub> =34	СБ <sub>3</sub> =37
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{38 + 34 + 37}{3} = 36,3$		

Отже, з отриманих даних таблиці 4.2 видно, що середньоарифметична сума балів дорівнює 36,3, тобто нова розробка має рівень комерційного потенціалу вище середнього.

#### 4.2 Прогнозування витрат на виконання науково-дослідної роботи

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи може складатися з таких етапів:

- 1) розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;
- 2) розрахунок загальних витрат на виконання даної роботи;
- 3) прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

#### 4.2.1 Перший етап

Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, можна здійснити за такими статтями та формулами:

4.2.1.1 Основна заробітна плата кожного із розробників (дослідників)  $Z_o$ , якщо вони працюють в наукових установах бюджетної сфери:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн.}, \quad (4.1)$$

де  $M$  — місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн;

$T_p$  — число робочих днів в місяці; приблизно  $T_p = 21$  день;

$t$  — число робочих днів роботи розробника (дослідника).

Заробітна плата розробника:

$$Z_p = \frac{13000}{21} \cdot 60 = 37142,86 \text{ (грн)}.$$

Заробітна плата наукового керівника проекту:

$$Z_{HK} = \frac{8500}{21} \cdot 19 = 7690,48 \text{ (грн)}.$$

Витрати на оплату праці, основна заробітна плата:

$$Z_o = Z_p + Z_{HK} = 37142,86 + 7690,48 = 44833,34 \text{ (грн)}$$

Розрахунок основної заробітної плати спеціаліста здійснюється за критеріями, наведеними в таблиці 4.3.

Таблиця 4.3 — Розрахунки основної заробітної плати спеціаліста

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Розробник	13000	619,05	60	37142,86
Науковий керівник проекту	8500	404,76	19	7690,48
Всього:				44833,34

4.2.1.2 Додаткова заробітна плата  $Z_d$  всіх розробників та робітників, які брали участь у виконанні даної роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = (0,1 \dots 0,12) \cdot Z_o \quad (4.2)$$

Додаткова заробітна плата усіх робітників та розробників:

$$Z_d = 0,12 \cdot 44833,34 = 5380(\text{грн})$$

4.2.1.3 Нарахування на заробітну плату  $H_{зп}$  розробників та робітників, які брали участь у виконанні даної роботи, розраховуються за формулою:

$$H_{зп} = (Z_o + Z_p + Z_d) \cdot \frac{\beta}{100}, \quad (4.3)$$

де  $Z_o$  — основна заробітна плата розробників, грн.;

$Z_d$  — додаткова заробітна плата розробників, грн.;

$\beta$  — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %.

З 01.01.2016 року ставка єдиного внеску на загальнообов'язкове державне соціальне страхування встановлено 22%.

Обчислення розміру нарахування на заробітну плату здійснюється за формулою, що наведена нижче:

$$H_{3П} = (44833,34 + 5380) \cdot \frac{22}{100} = 11046,93 \text{ (грн)}$$

4.2.1.4 Амортизаційні відрахування обладнання можна розрахувати за формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн.}, \quad (4.4)$$

де  $Ц$  – загальна балансова вартість обладнання,  $Ц = 11300$  грн.;

$H_a$  – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що  $H_a = 20\%$ ;

$T$  – термін використання обладнання,  $T = 4$  міс.

Амортизаційні відрахування для персонального комп'ютера становить:

$$A = \frac{11300 \cdot 20}{100} \cdot \frac{4}{12} = 753,33 \text{ (грн)}$$

4.2.1.5 Витрати на матеріали  $M$ , що були використанні під час виконання роботи, розраховуються по кожному виду матеріалів за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_b \text{ грн.}, \quad (4.5)$$

де  $H_i$  — витрати матеріалу  $i$ -го найменування;

$Ц_i$  — вартість матеріалу  $i$ -го найменування, грн./кг.;

$K_i$  — коефіцієнт транспортних витрат,  $K_i = 1,12$ ;

$B_i$  — маса відходів матеріалу  $i$ -го найменування, кг;

$Ц_b$  — ціна відходів матеріалу  $i$ -го найменування, грн/кг;

$n$  — кількість видів матеріалів.

Затрати матеріалів, що були використані при розробці програмного засобу наведені у таблиці 4.4.

Таблиця 4.4 — Вартість матеріалів, що були використані для розробки програмного засобу

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	Шт.	700	1	700
Папір А4	Шт.	120	82	98,4
Ручка	Шт.	48	1	48
Всього				846,4

Загальна вартість витрат становить:

$$M = 846,4 \cdot 1,12 = 947,97 \text{ (грн)}$$

4.2.1.6 Витрати на силову електроенергію  $V_e$ , розраховується за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} \text{ грн.}, \quad (4.6)$$

де  $V$  — вартість 1кВт-год. електроенергії,  $V = 3,69$  грн/кВт;

$\Pi$  — установлена потужність ноутбука,  $\Pi = 0,1$  кВт;

$\Phi$  — фактична кількість годин роботи ноутбука,  $\Phi = 180$ ;

$K_{\Pi}$  — коефіцієнт використання потужності,  $K_{\Pi} = 0,72$ .

Витрати на силову енергію становлять:

$$V_e = 3,69 \cdot 0,1 \cdot 180 \cdot 0,72 = 51,11 \text{ (грн)}$$

4.2.1.7 Витрати на послуги, що використовуються при виготовленні дослідного зразка ПЗ наведені у таблиці 4.5.

Таблиця 4.5 — Послуги, що використовуються при виготовленні дослідного зразка

Найменування послуг	Термін використання, місяців	Ціна за місяць, грн.	Сума, грн.
1. Послуга «AWS хостинг», шт.	3	980	2940
2. Послуга «Доменне ім'я», шт.	12	29,2	350,4
Всього			3290,4 грн.



4.2.1.8 Інші витрати  $I_b$  можна прийняти як 200% від суми основної заробітної плати розробників, тобто:

$$V_{iH} = 2 \cdot Z_o \quad (4.7)$$

Інші витрати становлять:

$$V_{iH} = 2 \cdot 44833,34 = 89666,68(\text{грн})$$

4.2.1.9 Сума усіх витрат становить:

$$V_{iH} = 44833,34 + 5380 + 11046,93 + 753,33 + 947,97 + 51,11 + 3290,4 + 89666,68 = 152679,36 (\text{грн})$$

#### 4.2.2 Другий етап

Розрахунок загальних витрат на виконання роботи. Загальна вартість роботи визначається за  $V_{\text{заг}}$  формулою:

$$V_{\text{заг}} = \frac{B}{\alpha} \quad (4.8)$$

де  $\alpha$  — частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$V_{\text{заг}} = \frac{152679,36}{1} = 152679,36 (\text{грн})$$

#### 4.2.3 Третій етап

Прогнозування загальних витрат на виконання та впровадження результатів. Прогнозування загальних витрат  $ZB$  на виконання та впровадження результатів здійснюється за формулою:

$$ЗВ = \frac{В_{\text{заг}}}{\beta}, \quad (4.9)$$

де  $\beta$  — коефіцієнт, який характеризує етап (стадію) виконання даної роботи, на стадії розробки промислового зразка,  $\beta \approx 0,7$ .

$$ЗВ = \frac{152679,36}{0,7} = 218113,37(\text{грн})$$

#### 4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Виконання наукової роботи та впровадження її результатів буде здійснюватися протягом одного року. Позитивні результати від впровадження розробки очікуються протягом трьох років від впровадження розробки. Одним із основних позитивних результатів є зростання величини прибутку. Зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Збільшення чистого прибутку підприємства (організації)  $\Delta\Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{Я}} \cdot N + \Delta N \cdot \Pi_{\text{Я}})_i, \quad (4.10)$$

де  $\Delta\Pi_{\text{Я}}$  — покращення основного якісного показника від впровадження результатів розробки у даному році;

$n$  — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

$N$  — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$P_{я}$  — основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки.

В результаті впровадження результатів наукової розробки витрати на заробітну плату робітникам, що виявляють фейкові новини, зменшаться на 1200 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 1200 грн), а кількість достовірних матеріалів збільшиться: протягом першого року — на 350 матеріалів, протягом другого року — на 200 матеріалів, протягом третього року — на 100 матеріалів.

Кількість опублікованих матеріалів до впровадження наукової розробки складала 700 матеріалів, а прибуток, що отримало підприємство (організація) на один матеріал до впровадження наукової розробки – 400 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого продукту  $\Delta P_1$  протягом першого року складатиме:

$$\Delta P_1 = 1200 \cdot 700 + (400 + 1200) \cdot 350 = 1400000 \text{ (грн)}$$

Протягом другого року:

$$\Delta P_2 = 1200 \cdot 700 + (400 + 1200) \cdot (350 + 200) = 1720000 \text{ (грн)}$$

Протягом третього року:

$$\Delta P_3 = 1200 \cdot 700 + (400 + 1200) \cdot (350 + 200 + 100) = 1880000 \text{ (грн)}$$

#### 4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

В даному підрозділі необхідно кількісно спрогнозувати, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної роботи.

Розрахований комерційний ефект можливого впровадження розробки

ще не означає, що розробка буде впроваджена. Якщо збільшення прогнозованого прибутку від впровадження результатів наукової розробки є вигідним для підприємства, то це ще не означає, що вона зацікавить інвесторів. Основні показники, що визначають рентабельність фінансування розробки певним інвестором, є відносна і абсолютна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення послідовності таких робіт:

4.4.1 Розрахуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану раніше, тобто будемо вважати, що  $ZB = PV = 218113,37$ (грн).

4.4.2 Розрахуємо очікуване збільшення прибутку  $\Delta\Pi_i$ , що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta\Pi_1 = 1400000 \text{ (грн)}, \quad \Delta\Pi_2 = 1720000 \text{ (грн)}, \quad \Delta\Pi_3 = 1880000 \text{ (грн)}.$$

4.4.3 Для спрощення подальших розрахунків необхідно побудувати вісь часу, на яку наносять всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Якщо загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнюють 218113,37грн., а результати вкладених у наукову розробку інвестицій почнуть виявлятися вже вкінці другого року впровадження. То ці результати виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 1400000 грн. відносно базового року, у другому році — збільшення чистого прибутку на 1720000 грн. (відносно базового року), у третьому році — збільшення чистого прибутку на 1880000 грн. (відносно базового року).

Тоді рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.1.

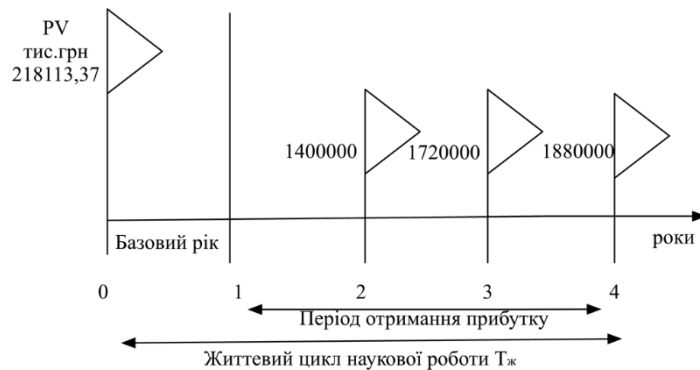


Рисунок 4.1 — Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4.4.5 Розрахуємо абсолютну ефективність вкладених інвестицій  $E_{\text{абс}}$ . Якщо  $E_{\text{абс}} \leq 0$  то результати від проведення наукових досліджень та їх впровадження буде збитковим і вкладати гроші в проведення цих досліджень ніхто не буде. Для цього показника скористаємося формулою:

$$E_{\text{абс}} = (\text{ПП} - \text{PV}), \quad (4.11)$$

де ПП — приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій  $PV = 3B = 218113,37$  грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.12)$$

де  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

T — період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований

рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

$t$  — період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$\begin{aligned} \text{ПП} &= \frac{1400000}{(1 + 0,1)^2} + \frac{1720000}{(1 + 0,1)^3} + \frac{1880000}{(1 + 0,1)^4} = \\ &= 1157024,79 + 1292261,46 + 1284065,3 = 3733351,55 \text{ (грн)} \end{aligned}$$

$$E_{\text{абс}} = (3733351,55 - 218113,37) = 3951464,92 \text{ (грн)}$$

Оскільки  $E_{\text{абс}} > 0$ , то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

4.4.6 Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_{\text{в}}$ . Для цього використаємо формулу:

$$E_{\text{в}} = T_{\text{ж}} \sqrt[3]{1 + \frac{E_{\text{абс}}}{\text{PV}}} - 1, \quad (4.13)$$

де  $T_{\text{ж}}$  — життєвий цикл наукової розробки, роки;

$E_{\text{абс}}$  — абсолютна ефективність вкладених інвестицій, грн;

$\text{PV}$  — теперішня вартість інвестицій  $\text{PV} = \text{ЗВ}$ , грн.

Далі, розрахована величина  $E_{\text{в}}$  порівнюється з мінімальною (бар'єрною) ставкою дисконтування  $\tau_{\text{мін}}$ , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування  $\tau_{\text{мін}}$  визначається за формулою:

$$\tau = d + f \quad (4.14)$$

де  $d$  — середньозважена ставка за депозитними операціями в комерційних банках;  $d = 0,16$ ;

$f$  — показник, що характеризує ризикованість вкладень;  $f = 0,05$ .

Якщо величина  $E_{\text{в}} > \tau_{\text{мін}}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки

здійснюватися не буде.

Спочатку спрогнозуємо величину  $\tau_{\text{мін}}$ . Припустимо, що за даних умов:

$$\tau_{\text{мін}} = 0,16 + 0,05 = 0,21$$

Тоді відносна (щорічна) ефективність вкладних інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_B = \sqrt[4]{1 + \frac{3951464,92}{218113,37}} - 1 = \sqrt[4]{19,12} - 1 = 1,09 \text{ або } 109\%$$

Оскільки  $E_B = 109\% > \tau_{\text{мін}} = 21\%$ , то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

4.4.7 Розраховують термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій  $T_{\text{ок}}$  можна розрахувати за формулою:

$$T_{\text{ок}} = \frac{1}{E_B} \quad (4.15)$$

Якщо  $T_{\text{ок}} < 3$ -х років, то фінансування даної наукової розробки в принципі є доцільним. В інших випадках потрібні додаткові розрахунки та обґрунтування.

Термін окупності становить:

$$T_{\text{ок}} = \frac{1}{1,09} = 0,91 \quad (4.16)$$

$T_{\text{ок}} < 3$ -х років, що свідчить про доцільність фінансування даної наукової розробки.

Рівень комерційного потенціалу розробки «Програмний засіб розпізнавання пошкоджених друкованих текстів із використанням багаторівневого аналізу документа» є вище середнього. Показники

ефективності показують, що даний програмний засіб є доцільним і буде цікавий для інвестора. Термін окупності розробленого проекту складає 0,91 року, тобто, менше 3-х років, що підтверджує доцільність вкладання коштів в дану розробку.



## ВИСНОВКИ

Існуючі системи розпізнавання текстових символів із високою достовірністю дозволяють переводити текстові документів у електронну форму. Але при появі у таких документах спотворених символів значно зростає відсоток помилок розпізнавання. Ця проблема на теперішній час є досить актуальною. Дана магістерська робота присвячена вирішенню цієї задачі по розробці програмних засобів розпізнавання пошкоджених символів друкованих документів.

У першому розділі магістерської роботи був виконаний огляд існуючих систем та методів оптичного розпізнавання текстових документів, проведено порівняльний аналіз засобів розпізнавання, вказано на основні недоліки та обґрунтовано необхідність розробки програмного продукту для розпізнавання пошкоджених символів друкованих документів.

Проведений аналіз різних методів та засобів розпізнавання символів друкованих документів дозволив виявити проблеми, що існують в даній області та уточнити області застосування різних методів, їх переваги та недоліки в застосуванні до вирішення завдань по розпізнаванню пошкоджених символів.

У другому розділі магістерської роботи запропонована послідовність багаторівневого розпізнавання текстових документів, розроблений алгоритм виділення символів та їх розпізнавання. Запропоновано процес розпізнавання розділити на два етапи на першому етапі використовувати нейронну мережу, а для символів, які спотворені та не були визначені на першому етапі, застосувати шаблонний метод розпізнавання із аналізом силуету символу та фону, на якому він знаходиться. Це збільшує достовірність розпізнавання.

У третьому розділі була вибрана мова програмування, розроблена структура програмного продукту, розроблена програма розпізнавання та приведені результати експериментальної перевірки запропонованого підходу.

У четвертому розділі магістерської кваліфікаційної роботи було здійснено обґрунтування доцільності виконання нового наукового вирішення

поставленої проблеми розпізнавання символів, виконано розрахунок необхідних економічних витрат для реалізації запропонованих засобів виділення та розпізнавання символів пошкоджених текстових документів та визначено комерційні переваги від впровадження запропонованого програмного продукту в оптичних системах розпізнавання текстових символів.

Розроблений програмний продукт може використовуватися в засобах розпізнавання текстових документів із спотвореними символами.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. — М.: Вильямс, 2004. — 928 с.
2. Фомин Я. А. Распознавание образов: теория и применение. / Я. А. Фомин. — М.: ФАЗИС, 2012. — 429 с.
3. Желтов С. Ю. Обработка и анализ изображений в задачах машинного зрения / С. Ю. Желтов. — М.: Физматкнига, 2010. — 672 с.
4. Что такое OCR [Электронный ресурс] — Режим доступа: [https://www.abbyy.com/ru-ru/finereader/what-is-ocr/?\\_\\_c=1](https://www.abbyy.com/ru-ru/finereader/what-is-ocr/?__c=1).
5. FineReader [Электронный ресурс]. — Режим доступа: [http://ru.wikipedia.org/wiki/ABBYY\\_FineReader](http://ru.wikipedia.org/wiki/ABBYY_FineReader).
6. Титов Ю. В. Об искажении символов при сканировании // Сб. трудов ИСА РАН «Системный подход к управлению информацией». М.: КомКнига, 2006, Том № 23. — С. 260-288.
7. Поліщук К. В. Розпізнавання пошкоджених друкованих текстів із використанням багаторівневого аналізу документа / К. В. Поліщук, М.А.Очкуров // Електронна наукова конференція ВНТУ «Молодь в науці: дослідження, проблеми, перспективи (МН-2021)». Режим доступа: <https://conferences.vntu.edu.ua/index.php/mn/mn2021>.
8. Поліщук К. В. Розпізнавання деформованих символів текстових документів // Електронна наукова конференція ВНТУ «Молодь в науці: дослідження, проблеми, перспективи (МН-2019)». Режим доступа: <https://conferences.vntu.edu.ua/index.php/mn/mn2019/paper/viewFile/8113/6786>
9. Моругов А.М. Методы распознавания символов. / А.М. Моругов, С.В.Волков //Труды Международного симпозиума «Надежность и качество», 2017, том 1. С. 76-81.
10. Жихаревич В. В. Аналіз методів розпізнавання символів тексту / В. В. Жихаревич, С. Е. Остапов, І. В. Миронів // Радіоелектронні і комп'ютерні системи. — 2016. — № 5. — С. 137—142.
11. Хайкин С. Б. Нейронные сети: полный курс. — 2-е изд. Пер. с англ. —

М.: Издательский дом «Вильямс», 2006. — 1104 с.

12. Мокшин, В. В. Разработка системы распознавания символов с использованием нейронной сети / В. В. Мокшин, Л. Р. Габдрахманова // Современные инновации в науке и технике. — 2014. — № 4. — С. 223—225.

13. Shi B., Bai X., Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition // IEEE transactions on pattern analysis and machine intelligence. 2017, vol. 39, no. 11, pp. 2298-2304.

14. Busta M., Neumann L., Matas J. Deep text spotter: an end-to-end trainable scene text localization and recognition framework // Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2204-2212.

15. Ye Q., Doermann D. Text detection and recognition in imagery: A survey // IEEE transactions on pattern analysis and machine intelligence. 2015, vol. 37, no. 7, pp. 1480-1500.

16. Uchida S. Statistical Deformation Model for Handwritten Character Recognition // Advances in Digital Document Processing and Retrieval. 2014, pp. 157-174.

17. Гонсалес Р. Цифровая обработка изображений / Гонсалес Р., Вудс Р. — М.: Техносфера, 2006. — 1072 с. ISBN 5-94836-028-8.

18. Липкина, А. Распознавание текста по структуре скелета букв / А. Липкина. — М.: МГУ им. Ломоносова, 2018. — 31 с.

19. Смирнов С.В. Корректировка ошибок оптического распознавания на основе рейтинго-ранговой модели текста / С.В. Смирнов // Труды СПИИРАН. — 2014. — Выпуск 4(35). — С. 64-82.

20. Горелик А. Л. Методы распознавания. / А. Л. Горелик, В. А. Скрипкин — 4-е изд. — М.: Высшая школа, 1984, 2004. — 262с.

21. Сравнение систем оптического распознавания. [Электронный ресурс] — Режим доступа: [https://en.wikipedia.org/wiki/Comparison\\_of\\_optical\\_character\\_recognition\\_software](https://en.wikipedia.org/wiki/Comparison_of_optical_character_recognition_software).

22. Що таке АBBYY FineReader [Электронный ресурс] — Режим доступа:

<https://help.abbyy.com/uk-ua/finereader/12/overview>

23. OCR CuneiForm для Windows [Электронный ресурс] — Режим доступа: <http://softobase.com/ru/ocr-cuneiform>.

24. Программа Readiris Pro [Электронный ресурс] — Режим доступа: <https://readiris.ru.uptodown.com/windows>

25. Smith, R.W. History of the Tesseract OCR engine: what worked and what didn't / R.W. Smith // Proceedings of SPIE. — 2013. — Vol. 8658. — pp. 186-195.

26. Болотова, Ю.А. Обзор алгоритмов детектирования текстовых областей на изображениях и видеозаписях / Ю.А. Болотова, В.Г. Спицын, П.М. Осина // Компьютерная оптика. — 2017. — Т. 41, № 3. — С. 441-452.

27. Погребняк И. В. Распознавание символов на изображениях, содержащих искажения / И. В. Погребняк, А. Ю. Тропченко // Международный научно-исследовательский журнал. — 2017. — Выпуск № 05 (59), часть 3, с.81-86.

28. Цифровая обработка изображений в информационных системах / И. С. Грузман, В. С. Киричук и др. — Новосибирск: НГТУ, 2002. — 352 с.

29. Методы компьютерной обработки изображений / Под ред. В. А. Сойфера. — М.: Физматлит, 2003. — 784 с. — ISBN 5-9221-0270-2.

30. Niblack W. An Introduction to Digital Image Processing, Prentice Hall. 1986.- P. 115-116.

31. Otsu N. A threshold selection method from gray-level histograms // IEEE transactionson systems, man, and cybernetics. 1979, vol. 9, no. 1, pp. 62-66.

32. Способ компьютерного распознавания объектов (патент РФ № 2191431 от 03.12.2009, МПК G06K 09/68).

33. Способ автоматического распознавания электронными средствами символов, таких как буквы и/или цифры, отпечатанных на любом материале, включающем структуры с сильной контрастностью (патент РФ № 2249251 от 20.06.2010, МПК G06K 09/80, 09/62).

34. Иванов Г. А. Автоматизация распознавания номерных знаков транспортных средств / Г. А. Иванов, А. А. Ларионов, Д. В. Панин // Комп'ютерне моделювання та інтелектуальні системи. — 2007. — С. 28-36.

35. Способ автоматизированного распознавания идентификационного номера (патент РФ № 2403817 от 05.10.2012, МПК G06K 09/68).

36. Багрова И. А. Выбор признаков для распознавания печатных кириллических символов / Багрова И.А., Грицай А.А., Сорокин С. В., Пономарев С. А., Сытник Д.А. // Вестник Тверского Государственного Университета 2010 г., С. 59-73.

37. Крысова И. В., Обработка изображений с помощью нейронных сетей/ И. В. Крысова, С. В. Пасечник, П. С. Бугаенко // XXXIX Гагаринские чтения. Том 4. — М.: МАТИ, 2013. — С. 144-146.

38. OpenCV library [Электронный ресурс] — Режим доступа: <https://opencv.org/>. — Загл. с экрана.

39. Прохоренок, Н. А. OpenCV и Java. Обработка изображений и компьютерное зрение / Н. А. Прохоренок. — СПб.: БХВ-Петербург, 2018. — 320 с.

40. AForge.NET: Framework [Электронный ресурс] — Режим доступа: <http://www.aforgenet.com/framework/>.

41. LTI-Lib [Электронный ресурс] — Режим доступа: <http://ltilib.sourceforge.net/doc/homepage/index.shtml>.

42. OpenCV vs VXL vs LTI: Performance Test — AI Shack [Электронный ресурс]. — Режим доступа: <http://www.aishack.in/tutorials/opencv-vs-vxl-vs-lti-performance-test/>.

43. Коэльё Л. Построение систем машинного обучения на языке Python. / Коэльё Л. П., Ричерт В. — Перевод с английского. — М.: ДМК Пресс, 2015. — 364с. — ISBN 978-5-97060-330-7.

44. Маккинли У. Python и анализ данных. — Перевод с английского. — М.: ДМК Пресс, 2015. — 482 с. — ISBN 978-5-97060-315-4.

45. The EMNIST dataset. [Электронный ресурс]. — Режим доступа: <https://www.nist.gov/itl/iad/image-group/emnist-dataset>.

46. Смирнов С.В. Критерии оценки качества результатов оптического распознавания / С.В. Смирнов // Сборник материалов XVI Международной

научно-практической конференции «Перспективы развития информационных технологий». — Новосибирск: Издательство ЦРНС, 2013. — С. 33—38.

47. Кавецький В. В. Економічне обґрунтування інноваційних рішень: Практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. — ВНТУ, 2013. — 110 с.

48. Адлер О. О. Методичні вказівки до підготовки та написання курсової роботи з дисципліни «Економічне обґрунтування інноваційних рішень» / Уклад. О. О. Адлер, І. В. Причепа, Н. М. Тарасюк. — Вінниця: ВНТУ, 2014. — 38 с.

49. Тарифи на електроенергію [Електронний ресурс]. — Режим доступу: <http://index.minfin.com.ua/tarif/electric.php>.