

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

ТЕХНОЛОГІЯ ОПТИЧНОГО РОЗПІЗНАВАННЯ ДРУКОВАНИХ  
СИМВОЛІВ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ

Виконав: студент 2 курсу, групи 2КІ-22м  
спеціальності 123 — «Комп'ютерна інженерія»

Дідич Дідич О. Д.

Керівник: Д.Т. н., проф. каф. ОТ

Мартинюк Мартинюк Т. Б.

« 14 » 12 2023 р.

Опонент: к.т.н., доц. каф. ПЗ

Ракитянська Ракитянська Г. Б.

« 15 » 12 2023 р.

Допущено до захисту

Завідувач кафедри ОТ

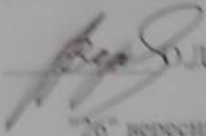
Азаров к.т.н., проф. Азаров О. Д.  
« 15 » 12 2023 р.

## ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Галузь знань — Інформаційні технології  
Освітній рівень — магістр  
Спеціальність — 123 Комп'ютерна інженерія  
Освітній професійна програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

  
О.Д. Азаров  
"26" вересня 2023 р.

### ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Діачу Олександр Дмитровичу

1 Тема роботи «Технологія оптичного розпізнавання друкованих символів на основі нейронної мережі» керівник роботи Мартинюк Тетяна Борисівна д.т.н., проф. каф. ОТ, затверджено наказом вищого навчального закладу від «18» вересня 2023 року № 247.

2 Строк подання студентом роботи: 08.12.23.

3 Вихідні дані до роботи: роздільна здатність зображення тексту не менше 300 dpi, модель кольорів для представлення зображення — RGB, кількість градацій яскравості зображення — 256.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз методів та засобів розпізнавання друкованих символів текстових документів, розробка технології та вибір засобів розпізнавання друкованих символів, розробка програмних засобів розпізнавання друкованих символів, тестування та оцінка ефективності функціонування створеної програми, розрахунок економічної доцільності створення програми розпізнавання друкованих символів.

6 Консультанти розділів проекту (роботи)

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Дата	
		Підпис, завдання видав	завдання прийняв
1-4	д. т. н., професор Мартинюк Тетяна Борисівна	<i>Т. Мартинюк</i>	<i>Т. Мартинюк</i>
5	к. е. н., професор кафедри ЕПВМ Небава Микола Іванович	02.11.2023	02.12.2023

*Небава* *Небава*

7 Дата видачі завдання 26.09.23

8 Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2— Календарний план

№ з/п	Назва етапів проекту (роботи)	Сррж виконання етапів проекту (роботи)	Примітка
1	Пошук та огляд інформаційних джерел	03.10.2023р.	виконано
2	Огляд та аналіз існуючих методів і засобів виділення і розпізнавання друкованих символів	14.10.2023р.	виконано
3	Результати проведених досліджень та методів виділення і розпізнавання друкованих символів	25.10.2023р.	виконано
4	Проектування нейромережевої технології виділення і розпізнавання друкованих символів	19.11.2023р.	виконано
5	Розрахунок економічної доцільності створення програми виділення і розпізнавання друкованих символів	02.12.2023р.	виконано
6	Оформлення пояснювальної записки і презентації	04.12.2023р.	виконано
7	Попередній захист	07.11.2023р.	виконано
8	Перевірка «антиплагіат»	05.12.2023р.	виконано
9	Аналіз виконання роботи, висновки, додатки	05.12.2023р.	виконано
10	Оформлення пояснювальної записки та ілюстративного матеріалу	06.12.2023р.	виконано
11	Перевірка якості виконання магістерської роботи та усунення недоліків	07.12.2023р.	виконано

Студент *Ділич* Ділич Олександр Дмитрович

Скривник *Т. Мартинюк* д.т.н., професор Мартинюк Тетяна Борисівна

## АНОТАЦІЯ

УДК 004.93

Дідич О. Д. Технологія оптичного розпізнавання друкованих символів на основі нейронної мережі. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма комп'ютерна інженерія. Вінниця: ВНТУ, 2023. 110 с.

На укр. мові. Бібліогр.: 37 назв; рис. 10; табл. 5.

Магістерську кваліфікаційну роботу присвячено розробці нейромережевої технології розпізнавання друкованих символів. Здійснено огляд існуючих методів та засобів виділення та розпізнавання друкованих символів текстових документів, визначено їх переваги та недоліки. Розроблено технологію виділення і розпізнавання друкованих символів, в основі якої знаходиться використання моделі згорткової нейронної мережі. Розроблено програму виділення і розпізнавання друкованих символів на основі використання згорткової нейронної мережі. Виконано навчання нейронної мережі для виконання розпізнавання друкованих символів. Проведено тестування створеного програмного застосунку для визначення його продуктивності. Проведено економічний розрахунок доцільності виконання нової розробки, обчислені фінансові затрати та визначено економічні переваги від впровадження запропонованого рішення у вигляді завершеного програмного продукту.

Ключові слова: друкований символ, згорткова нейронна мережа, навчання мережі.

## **ABSTRACT**

Didych O. D. Character recognition of text documents using a neural network. Master's thesis in specialty 123 - computer engineering, computer engineering education program. Vinnytsia: VNTU, 2023. 110 p.

In Ukrainian language. Bibliographer: 37 titles; fig.: 10; table 5.

The master's thesis is devoted to the development of a neural network technology for recognizing printed characters. An overview of the existing methods and means of selection and recognition of printed symbols of text documents was carried out, their advantages and disadvantages were determined. A technology for selection and recognition of printed characters has been developed, which is based on the use of a convolutional neural network model. A program for selection and recognition of printed characters based on the use of a convolutional neural network has been developed. A neural network was trained to recognize printed characters. The created software application was tested to determine its performance. An economic calculation of the feasibility of implementing a new development was carried out, the financial costs were calculated and the economic benefits from the implementation of the proposed solution in the form of a completed software product were determined.

Keywords: printed character, convolutional neural network, network learning.

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ ТЕКСТОВИХ ДОКУМЕНТІВ</b> .....	11
1.1 Загальний підхід до оптичного розпізнавання тексту.....	11
1.2 Аналіз методів розпізнавання друкованих символів .....	13
1.3 Використання штучної нейронної мережі.....	21
1.4 Аналіз засобів розпізнавання текстових документів .....	25
<b>2 РОЗРОБКА ТЕХНОЛОГІЇ ТА ВИБІР ЗАСОБІВ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ</b> .....	32
2.1 Етапи розпізнавання друкованих символів.....	32
2.2 Розробка послідовності розпізнавання друкованих символів.....	38
2.3 Використання згорткової нейронної мережі.....	44
<b>3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ</b> .....	54
3.1 Розробка архітектури програми розпізнавання символів.....	54
3.2 Попередня обробка текстового зображення.....	57
3.3 Модуль програми створення та навчання нейронної мережі.....	60
<b>4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ СТВОРЕНОЇ ПРОГРАМИ</b> .....	67
4.1 Тестування роботи програмного продукту .....	67
4.2 Перевірка якості роботи програми розпізнавання тексту.....	71
<b>5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ</b> .....	75
5.1 Комерційний та технологічний аудит науково-технічної розробки....	75

					08-54.МКР.027.00.000 ПЗ			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		Дідич О. Д.			Технологія оптичного розпізнавання друкованих символів на основі нейронної мережі	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>		Мартинюк Т. Б.					6	110
<i>Опонент</i>		Ракитянська Г.Б.				ВНТУ, гр. 2КІ-22м		
<i>Н. Контроль</i>		Швець С.І.						
<i>Затверджую</i>		Азаров О.Д.						

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи .....	79
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором .....	84
<b>ВИСНОВКИ</b> .....	91
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</b> .....	93
<b>ДОДАТОК А</b> Технічне завдання .....	97
<b>ДОДАТОК Б</b> Лістинг програми сегментації тексту.....	101
<b>ДОДАТОК В</b> Лістинг програми роботи нейронної мережі.....	103
<b>ДОДАТОК Г</b> Послідовність розпізнавання символів .....	106
<b>ДОДАТОК Д</b> Методи розпізнавання тексту .....	107
<b>ДОДАТОК Е</b> Системи оптичного розпізнавання тексту.....	108
<b>ДОДАТОК Ж</b> Структура програми розпізнавання символів.....	109
<b>ДОДАТОК И</b> Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень .....	110

					08-54.МКР.027.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## ВСТУП

Одним із напрямів розпізнавання образів є їх реалізація у системах оптичного розпізнавання текстів (Optical Character Recognition, OCR-системах), що являє собою один із найбільш яскравих й дієвих технологій штучного інтелекту. Система OCR реалізує автоматичне розпізнавання при допомозі спеціально розроблених програм зображень символів друкованого або ж рукописного тексту й переведення його в формат, який можна використати для подальшого оброблення редакторами текстів або текстовими процесорами та інше. Іноді під OCR розуміють власне пристрій для автоматичного зчитування тексту або ж оптичного розпізнавання символів [1-3].

Сучасні інформаційні технології обробки і передачі даних дозволяють суттєво спростити доступ до різного роду інформаційних ресурсів, що накопичені на теперішній час людством, при виконанні умови, що вони будуть переведені і подані у більш зручному електронному вигляді. Найбільш швидким й простим є виконання операції по скануванню документів із використанням сканерів, у результаті роботи якого отримуємо документ у вигляді графічного файлу. Але кращим варіантом є текстове подання даних. Цей підхід дозволяє істотно зменшити витрати на збереження й передачу даних та додатково дозволяє проводити операції по аналізу електронних документів. Ось чому із практичної точки зору значний інтерес представляє саме переведення та подання паперових носіїв у вигляді текстових електронних документів.

Тим не менше, створення кожного нового програмного засобу в цій області залишається творчим завданням й вимагає проведення додаткових досліджень через наявність специфічних вимог по вирішенню задач швидкодії, надійності розпізнавання та обсягів пам'яті, якими визначається кожне конкретне завдання по розпізнаванню символів друкованих текстів [4].

Дослідження по виділенню й розпізнаванню символів друкованих текстів характеризуються великим розмаїттям у можливості вибору засобів для їх вирішення. Традиційними завданнями цього процесу обробки текстових



документів є задачі виявлення й розпізнавання друкованих символів, що мають відому форму [5].

Набагато складнішим стає це завдання при здійсненні автоматизованого розпізнавання друкованих текстів із дефектами, яке у даний час поки ще не вирішено у повній мірі для такого типу текстів, та є однією з найбільш **актуальних** й складних задач розпізнавання даних.

**Метою дослідження** магістерської роботи є вдосконалення технології оптичного розпізнавання друкованих символів на основі нейронної мережі.

**Задачі дослідження** магістерської роботи:

— провести аналіз відомих методів обробки цифрових зображень для пошуку й розпізнавання друкованих символів текстового документу;

— запропонувати поліпшену технологію обробки цифрових зображень текстових документів для виділення й подальшого розпізнавання друкованих символів цих документів;

— сформувати послідовність та розробити програму обробки зображення для розпізнавання друкованих символів текстових документів;

— виконати обґрунтування доцільності виконання нового наукового рішення, розрахувати економічні витрати для створення програмних засобів розпізнавання друкованих символів текстових документів та визначити переваги від впровадження нового програмного продукту.

**Об'єктом дослідження** магістерської роботи є процес отримання даних про текстовий документ шляхом знаходження у виділеній сцені його зображення й розпізнавання друкованих символів.

**Предметом дослідження** магістерської роботи є методи обробки цифрових зображень текстових документів для виділення й розпізнавання друкованих символів.

**Методи дослідження** магістерської роботи: метод аналітичного огляду, метод порівняння і аналогій, метод моделювання, метод об'єктно-орієнтованого програмування для розробки програмного додатку. Використовувались методи теорії множин для формування множини ознак для

розпізнавання, методи диференційного числення для налаштування мережі, 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 технологію обробки цифрового зображення текстового документу для виділення й розпізнавання друкованих символів тексту, яка відрізняється від відомих підходів послідовним виконанням етапів сегментації зображення, використанням нейронної мережі й шаблонного методу, що дозволяє підвищити точність розпізнавати друкованих символів у отриманому зображенні тексту, у томі числі і символів із дефектами.

**Практичне значення** одержаних результатів магістерської роботи: сформована послідовність обробки цифрових зображень для виділення й розпізнавання друкованих символів текстових документів, розроблено програму обробки зображення для виділення друкованих символів текстових документах та їх подальшого розпізнавання.

**Апробація** результатів магістерської роботи: зроблено доповідь на Всеукраїнській ЛІІІ науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ-2024).

За результати магістерської роботи **опубліковано** 1 наукову працю [6].

## **1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ ТЕКСТОВИХ ДОКУМЕНТІВ**

У цьому розділі магістерської роботи проведено аналіз методів розпізнавання символів текстових документів у отриманому зображенні й розглянуто засоби, які застосовуються для здійснення операцій по розпізнаванню тексту.

### **1.1 Загальний підхід до оптичного розпізнавання тексту**

Для організацій, які працюють з великими обсягами різноманітних даних, їх обробка та подальша обробка є щоденною та терміною потребою. У зв'язку з постійним збільшенням обсягу даних і різноманітністю їх форматів, сьогодні існує особлива потреба в автоматичній обробці цих даних, щоб підвищити ефективність роботи та звільнити людські ресурси від рутинних механічних завдань.

Хоча зараз більшість документів створюється на комп'ютерах, завдання створення повністю електронного документообігу ще не завершено. Сучасні системи часто охоплюють діяльність окремих організацій, а обмін даними між організаціями здійснюється за допомогою традиційних паперових документів.

Завдання перетворення інформації з паперового носія на електронний є актуальним не лише в рамках вимог, що впливають із системи документообігу. Сучасні інформаційні технології значно спрощують доступ до накопичених людством джерел інформації за умови їх переведення в електронний вигляд. Найпростіший і швидкий спосіб — сканувати документи за допомогою сканера. Результатом є цифрове зображення документа – графічний файл. Письмове представлення інформації має перевагу над графічним. Ця опція значно скорочує витрати на зберігання та передачу інформації, дає можливість реалізувати всі можливі сценарії використання та аналізу електронних документів. Тому переклад паперових документів на електронні на практиці викликає занепокоєння.

Інструменти оптичного розпізнавання символів (OCR) використовуються для видобування та редагування даних зі сканованого паперу, фотографій з камери та файлів PDF, які містять лише зображення. Програмне забезпечення OCR витягує символи із зображень, перетворює їх на слова та збирає речення, щоб ви могли отримати доступ до оригінального тексту та змінити його. Це також усуває необхідність ручного введення даних.

Системи оптичного розпізнавання символів використовують апаратне та програмне забезпечення для перетворення фізичних друкованих документів у машиночитаний текст. Текст копіюється або зчитується апаратними засобами (наприклад, оптичним сканером або спеціальною платою), а подальша обробка зазвичай виконується програмним забезпеченням. Програмне забезпечення оптичного розпізнавання символів (OCR) зазвичай обробляє великі растрові зображення сторінок, надісланих зі сканера. При цьому більшість систем створюють шаблони для різноманітних креслень. Після розпізнавання кількох слів програмне забезпечення визначає шрифт, який буде використано, і шукає лише пари слів, які відповідають цьому шрифту. У деяких випадках програмне забезпечення використовує числове значення (масштаб) частини символу для визначення нового шрифту. Це покращує ефективність виявлення. Вхідними даними для системи розпізнавання є растрове зображення сторінок документа.

Для алгоритмів розпізнавання бажано, щоб якість вхідного зображення була найкращою. Якщо зображення шумне, різке та має низький контраст, завдання алгоритму розпізнавання ускладнюється. Етап сегментації є важливим у цьому методі, оскільки методи класифікації на основі меж є неефективними без надійної сегментації. Обробка відбувається за традиційною покроковою процедурою аналізу текстових документів. Першим кроком є зв'язування аналізу компонента, де збережено ескіз компонента.

Найпростішим вирішенням цієї проблеми може бути просте попиксельне порівняння двох зображень, щоб знайти відмінності. І насправді цей варіант дійсно покаже всі відмінності, але чи так вони важливі? Виявляється ні. Деякі відмінності, які можуть виявити машини, насправді незначні, тому що люди

взагалі не можуть їх виявити. Так, наприклад, простий текст може стати причиною того, що тести «проваляться», тобто дадуть результат, який нас не влаштовує. Проблема згладжування шрифтів є досить серйозною для перевірки зображень, оскільки різні машини можуть виконувати цей процес за різними алгоритмами.

Як бачимо, різні алгоритми згладжування та різні налаштування комп'ютера дають нам майже різні зображення з точки зору машини. Але для нас ця різниця не має значення, тому що в обох випадках текст знаходиться в одному місці і, до того ж, текст повністю ідентичний. Тому нам потрібно розробити систему, нечутливу до такого впливу. Основна проблема в цих випадках полягає в тому, що алгоритм розпізнавання символів оптимізовано, щоб мати змогу розпізнавати символи точно та швидко.

Розпізнавання структурованих (друкованих) символів на різних зображеннях забезпечує вирішення багатьох наукових і прикладних завдань при ідентифікації різних типів об'єктів. Сучасні методи розпізнавання символів поділяються на типові завдання, такі як розпізнавання тексту та спеціальні завдання розпізнавання символічної інформації, прикріпленої до поверхонь різних об'єктів. Розглянемо найвідоміші і поширені методи розпізнавання символів.

## 1.2 Аналіз методів розпізнавання друкованих символів

Зараз у світі існує багато різноманітних програм розпізнавання символів. Однак важливо зазначити, що здатність людини читати низькоякісний текст все ще перевершує можливості комп'ютерів. Весь друкований текст має певні характеристики набраного шрифту. Тому існує два типи алгоритмів розпізнавання символів: на основі шрифту і незалежний від шрифту. Залежні від шрифту алгоритми використовують важливу інформацію про шрифт символів. Це означає, що ваша заявка повинна містити повну копію тексту цим шрифтом. Ця програма вимірює й аналізує різні характеристики шрифтів і

додає їх до бази даних довідкових характеристик. Основними методами виявлення тестів є:

Модельний метод є метод зразка, який перетворює зображення одного символу на растрове зображення, порівнює його з усіма доступними зразками в базі даних і вибирає зразок, який найменше відрізняється від вхідного зображення. Методи шаблону дуже стійкі до візуальних помилок і швидко обробляють вхідні дані, але надійно розпізнаються лише шрифти, шаблони яких відомі. Крім того, якщо виявлений шрифт дещо відрізняється від еталонного шрифту, метод шаблону може бути неправильним, навіть якщо мова йде про зображення дуже високої якості. Класифікатор сітки або шаблону послідовно порівнює символ із набором стандартів і накладає зображення. Еталони — це спеціально підготовлені зображення, кожне з яких поєднує в собі ескізи багатьох варіантів написання того чи іншого символу.

Першим кроком у методі шаблону є перетворення сканованого зображення на растрове. Потім він порівнюється з усіма шаблонами, доступними в базі даних системи. Найбільш відповідним шаблоном вважається той, який найменше відрізняється від досліджуваного зображення. Шаблони для кожного класу зазвичай визначаються шляхом усереднення зображень символів. Ці методи забезпечують дуже високу точність виявлення дефектів тексту (склеювання або розрив).

Недоліком цього методу є те, що він не може розпізнати шрифти, які відрізняються (розміром, нахилом або малюнком) від шрифтів, вбудованих у систему. Алгоритми, засновані на шаблонних методах, повинні заздалегідь знати, які шрифти будуть представлені для розпізнавання. Зважаючи на велику кількість друкованих матеріалів, неможливо охопити всі шрифти та їх модифікації в процесі навчання. Іншими словами, цей фактор обмежує універсальність таких методів.

Шаблонні або еталонні методи передбачають порівняння вибраного невпізнаного символу з набором деяких стандартних символів. Для цього можна також використати нейронні мережі, які необхідно заздалегідь

заповнити стандартними наборами символів. На теперішній час існують кілька алгоритмів порівняння отриманого тексту зі стандартним набором еталонів.

Найпростішим варіантом є попиксельне порівняння, але для цього потрібні порівнювані шари зображення. Інші параметри включають накладення та накладання зміщення, коли зображення вирівнюються одне з одним. На перший погляд методи перевірки рукописних і друкованих символів можуть здатися схожими, але вони дуже різні. На відміну від шрифтів, стандарти рукописного тексту є лише приблизним прикладом. Це значно підвищує ймовірність помилки [6].

Порівняйте відсоток кожного бала з еталонним результатом. Необхідно виконати певну попередню обробку, щоб досягти незмінності розміру та позиції, а потім порівняти з підготовленою базою даних стандартів зображення: якщо збігів більше ніж заданий тег, ми вважаємо, що зображення буде розпізнано [9].

Система розпізнавання TypeReader використовує машинно-незалежний алгоритм на основі шаблонного підходу. Цей метод вимагає створення шаблону для кожного шрифту. Наприклад, програма TypeReader використовує 2100 різних розкладок символів.

Структурне розпізнавання символів зберігає інформацію не про фактичне написання символу тексту, а про топологію символів. Цей стандарт містить інформацію про взаємне розташування окремих компонентів. Перевагою цього методу є його толерантність до невеликих кутових переміщень і поворотів символів, а також до різних стилістичних варіацій шрифтів. Однак, коли кут повороту стає десятками градусів, розпізнавання символів цим методом стає неможливим.

За допомогою цього методу невідомі навіть такі характеристики, як розмір подвійних літер і шрифт, який використовується для друку. Однак основною проблемою цього методу є ідентифікація, яка має ряд недоліків [4].

Система розпізнавання Care Omni Page Professional використовує алгоритм, що базується на пошуку конкретних спільних характеристик

символів тексту. Ця система містить 100 різних алгоритмів для ідентифікації 100 різних символів: великих і малих літер від «А» до «Z», цифрових записів і знаків пунктуації. Кожен із цих алгоритмів прагне намалювати такі «об'єкти», як «острови», «півострів», точки, прямі поглиблення та дуги. Експертні системи також враховують горизонтальні та вертикальні проекції слідів літер і звертають увагу на основні характеристики згенерованих кривих, підсумовуючи кількість темних пікселів, які вони містять.

Методи структурного розпізнавання зберігають інформацію не про те, як пункт за пунктом написаний символ, а про його взаємопов'язану структуру. Іншими словами, стандарт містить інформацію про взаємне розташування окремих елементів символу [4, 5].

Зрозуміло, що розмір впізнаваної літери і навіть шрифт, яким вона надрукована, вже не мають значення. Але основною проблемою методів розпізнавання структур залишається ідентифікація маркерів з дефектами (наприклад, розриви рядків або злиття сусідніх рядків).

Розглянемо метод розпізнавання на конкретному прикладі [4]. Розпізнаний символ проходить процес кадрування (зменшення). Процедура фреймування та її використання в системах розпізнавання тексту вже давно вивчається та досліджується різними авторами, в які внесено багато робіт [5-8].

Кожен контур, отриманий із скелетного представлення, описується як послідовний набір спеціальних точок і так званий ланцюговий код, який складається з опорної точки, кількох кодів і таблиці напрямків від плчаткової точки до наступної точки, яку слід вказати. Спеціальними точками є кінцеві точки та точки розгалуження, тобто точки, де сусідні точки утворюють принаймні три зв'язані області. У прикладі на зображенні дві внутрішні схеми: одна точка і три розгалуження.

В описі результату використовується заокруглення, яке здійснюється шляхом видалення коротких ліній, об'єднання близько розташованих тріодів і руйнування дрібних внутрішніх контурів. Для зовнішньої схеми існує її тип або код топології. Для цього схема записується як послідовний набір певної



кількості точок, що відповідають петлі за годинниковою стрілкою. Потім шляхом перенумерації спеціальних точок і зміни початкової точки контуру робиться спроба визначити контур одним із основних типів.

Класифікатор структури виконує структурний аналіз символу, розбиває його на основні компоненти (сегменти, дуги, кола, точки) і формує точну схему аналізованого знаку. Потім отриману схему (що зображує структуру листа) порівнюють з еталоном. Цей класифікатор працює повільніше, ніж растрові та класифікатори об'єктів, але має високу точність.

Структурні описи, на відміну від декомпозицій, є органічно зрозумілими для розв'язувача задач розпізнавання об'єктів, можуть виконуватися обчислювальними засобами під час розпізнавання та не вимагають обчислень.

Структурні ознаки, які використовуються в структурному описі, є непохідними (тобто однозначно базовими), примітивними елементами (символами) образу ідентифікованого об'єкта. Поява структурних особливостей зумовлена проблемою розпізнавання зображення з його властивостями. Методи структурного розпізнавання символів зберігають інформацію не про пунктуацію символу, а про його топологію. Тобто стандарт містить інформацію про взаємне розташування окремих елементів символу [5, 6].

Зрозуміло, що розмір літери, що розпізнається, і навіть шрифт, яким вона надрукована, більше не мають значення. Але основною проблемою методів розпізнавання структур залишається ідентифікація маркерів з дефектами (наприклад, розриви рядків або злиття сусідніх рядків).

Розпізнаний символ подається на шаблон «стандартного» розміру та положення, який згладжується, тобто всі елементи, які призводять до його спотворення, усуваються із зображення шляхом заміни групових елементів зображення (зазвичай сусідніх елементів) їхніми еквівалентами і потім кодуються за допомогою рядків Фрімена, створюючи безперервні послідовності рядків. Отримане зображення піддається процедурі скелетонізації (зменшення) [6].

Отриманий ескіз скелетного представлення зображень як послідовний набір конкретних точок і ланцюжковий код схеми за годинниковою стрілкою, що складається з опорної точки, деякого коду та таблиці, що вказує шлях від наступної точки до опорної точки.

Спеціальними точками є кінцеві точки та точки розгалуження, тобто точки, де сусідні точки утворюють принаймні три зв'язані області. Потім шляхом перенумерації спеціальних точок і зміни початкової точки контуру робиться спроба визначити контур одним із основних типів. Оперуючи обмеженою кількістю атомарних (непохідних) елементів (примітивів), можна отримати описи різних об'єктів. Тобто для різних об'єктів ви можете мати однаковий набір непохідних елементів. Але для того, щоб опис і визначення непохідних елементів були повними, необхідно надати правила асоціації, які визначають методи побудови об'єкта з непохідних елементів. У результаті два елементи, які не однаково походять від різних об'єктів, можуть бути з'єднані один з одним за різними правилами. Це дозволить відрізнити предмети один від одного.

Загалом, для опису довільного об'єкта непохідні елементи об'єднуються в рядки (фрази) за окремим набором правил, характерних тільки для цього об'єкта. У результаті зв'язків із непохідних елементів (структурних особливостей) утворюється об'єкт, подібно до того, як речення в мові будуються сполучними словами, самі складаються з букв. Зокрема, структурні методи мають схожість із синтаксисом природних мов. Тому структурні ознаки ще називають мовними або синтаксичними. Тому опис кожного класу має бути набором непохідних послідовностей елементів із правилами з'єднання. Кожна з них є характеристикою структурної характеристики об'єктів даного класу.

Розпізнавання структурних символів під дією випадкового шуму зводиться до пошуку оптимального представлення розпізнаних символів. Функція якості полягає в пошуку найбільш ймовірного набору прихованих параметрів зображення. Наприклад, при розпізнаванні рядка тексту вимога

знайти найбільш імовірну послідовність літер еквівалентна вимозі мінімізувати кількість неправильно розпізнаних символів.

Зображення, що містять текст, таблиці та малюнки, створюються та читаються за певними правилами, набір яких можна формалізувати у певну граматику. Очевидно, що алгоритми, які розпізнають такі зображення, повинні покладатися на використання цих граматичних правил.

Методи створення таких граматик ґрунтуються на використанні теорії графів та методів теорії формальних мов та граматик.

У методі теорії графів зображення аналізованого тексту представлено у вигляді певним чином розміченого графа. Задачі розпізнавання символів ставляться як задачі знаходження ізоморфізму еталонного й вхідного графів, або ж ізоморфізму їх підграфів.

У методах теорії формальних мов і граматик зображення розглядається як певне слово у деякій формальній мові, яка описується за допомогою конструкцій, які подаються як узагальнення граматики Хомського. Розпізнавання символів полягає у знаходженні найкращого у певному значенні виводу зображення у цій заданій граматиці [7].

Але досить складною проблемою методів розпізнавання структур була і залишається ідентифікація маркерів із дефектами тексту (наприклад, розриви рядків тексту або ж злиття сусідніх рядків).

Класифікатор символів створює гіпотези на основі того, наскільки параметри символів подібні до еталонних значень. Працює на основі певних числових характеристик, таких як довжина периметра, кількість чорних плям у різних областях або в різних напрямках тощо. Методи об'єктів на основі зображень вирівнюються за векторами об'єктів. Суть розпізнавання полягає в порівнянні вектора ознаки з набором векторів вибірки того ж типу. Методи ознак ґрунтуються на тому факті, що  $N$ -вимірні вектори ознак прикріплені до зображення. Розпізнавання передбачає порівняння його з набором еталонних векторів того самого розміру. Задача розпізнавання, яка полягає у визначенні належності зображення до того чи іншого класу на основі аналізу обчислених

ознак, має низку строгих математичних рішень у рамках методів ідентифікації та ймовірності [12, 13].

У системах розпізнавання символів найчастіше використовується класифікація, заснована на обчисленні евклідової відстані між вектором ознак розпізаного символу та вектором ознак опорного дескриптора. Вид і кількість знаків багато в чому визначають якість ідентифікації. Формування вектора здійснюється під час аналізу попередньо підготовлених зображень. Цей процес називається вилученням ознак. Стандарт для кожного класу досягається шляхом обробки схожих символів.

Основними перевагами ознакових методів є проста реалізація, хороша здатність до узагальнення, хороша стійкість до зміни форми символу, низька кількість відхилень розпізнавання та висока швидкість. Найбільш серйозним недоліком цих методів є їх нестійкість перед різними дефектами зображення. Крім того, ознакові методи мають ще один серйозний недолік: на етапі виділення ознак частина інформації про символ безповоротно втрачається. Вилучення ознак виконується незалежно, тому інформація про взаємне розташування елементів символу втрачається.

У типовій системі оптичного розпізнавання символів введення символів зчитується й оцифровується із використанням оптичних сканерів. Потім кожен символ буде локалізовано та розподілено, а отриману матрицю буде попередньо оброблено, тобто згладжено, відфільтровано та нормалізовано. В результаті попередньої обробки виділяються особливі ознаки, після чого проводиться класифікація.

Алгоритм скелетонізації. Це метод розпізнавання одиничних бінарних зображень, заснований на побудові кадрів цих зображень і виділенні кадрів ребер і вузлів. Потім на підставі співвідношення ребер, їх кількості та кількості вузлів складається таблиця відповідних зображень. Наприклад, рамка кола буде вузлом, рамка літери «Р» матиме три ребра і два вузли, а ребра будуть 2: 2: 1. У програмуванні цей метод має кілька можливостей виконання [7].

Існують також методи, що засновані на використанні нейронних мереж. Ідея цих методів полягає в моделюванні людського мозку. На вхід попередньо навченої нейронної мережі надходить вектор, що представляє вхідне зображення (пікселі, частотні характеристики, довжина хвилі). На виході нейрон, що відповідає класу розпізнаного символу, видасть максимальне значення функції активації. Також генерується багато важливих характеристик зображення, які потім обробляються іншими системами. Нейронна мережа навчається на багатьох навчальних прикладах. Крім того, можна навчатися з учителем (перцептрон) або самоорганізовуватися (мережа Кохонена) [14].

Нейронна мережа — це структура зв'язаних елементів, на якій задані функції перетворення сигналу, а також коефіцієнти, які можна налаштовувати для певного характеру роботи. Деякі елементи структури вибираються як входи: приймають сигнали ззовні, деякі як виходи: формують отримані сигнали. Сигнал, що проходить через нейронну мережу, трансформується за формулами на елементах мережі та формує на виході відповідь [12, 13].

Нейронні мережі можна використовувати як класифікатори в системах розпізнавання тексту. Цей класифікатор можна навчити, регулюючи коефіцієнти на елементах мережі та таким чином намагаючись досягти ідеальних результатів розпізнавання. Нейронні мережі можна успішно використовувати в системах розпізнавання тексту, але багато обмежень перешкоджають їх широкому використанню. Щоб побудувати мережу, яка забезпечує розпізнавання кожного символу в тексті, необхідно побудувати досить велику мережу елементів, що призводить до великого споживання пам'яті. Під час процесу розпізнавання використовується навіть більше системних ресурсів, оскільки функції елементів мережі працюють з числами з плаваючою комою. Крім того, нейронні мережі потрібно навчити для всіх випадків, але це не гарантує точних результатів. І, нарешті, робота нейронної мережі розпізнавання тексту значною мірою залежить від конфігурації мережі та функцій, зазначених в елементах, що вимагає великих зусиль для побудови працюючої мережі.

### 1.3 Використання штучної нейронної мережі

Для розпізнавання символів часто використовуються штучні нейронні мережі. Алгоритми, розроблені для нейронних мереж розпізнавання символів, не часто проектуються таким чином. Зображення символів, що представляють вхідне зображення, яке зчитується, зменшено до стандартного розміру. Зображення зазвичай масштабуються до 16x16 пікселів. В якості вхідних параметрів нейронної мережі використовуються значення яскравості у вузлах нормалізованого зображення. Кількість вихідних параметрів нейронної мережі дорівнює кількості розпізнаних символів. Результатом розпізнавання є символ, що відповідає максимальному значенню джерела нейронної мережі. Підвищення надійності таких алгоритмів зазвичай передбачає пошук більш значущих вхідних характеристик і збільшення складності структури нейронної мережі.

Надійність виявлення та вимоги до обчислювальних ресурсів програми сильно залежать від структури нейронної мережі та вибору параметрів. Зображення форми зменшено до одного розміру (28x28 пікселів). Отримане зображення подається на вхід нейронної мережі з трьома внутрішніми шарами та 10 вузлами у верхньому шарі. Нижні рівні мережі підключені не повністю.

Підпорядковані вузли мають загальний набір ваг. За словами розробників, усе це спрямовано на покращення здатності нижніх рівнів мережі розрізняти основні ознаки зображення. Невикористані вагові коефіцієнти видаляються, щоб збільшити здатність мережі до узагальнення та зменшити обчислювальну пам'ять і необхідний обсяг пам'яті. Це зменшує кількість незалежних параметрів на чотири. Навчання нейронної мережі виконується на наборі з 7300 символів, а тестування – на наборі з 2000 символів. Похибка розпізнавання становить близько 1% у навчальній і близько 5% у тестовій [15].

Надійність виявлення та обчислювальні вимоги програми значною мірою залежать від структури нейронної мережі та вибору параметрів. Зображення символів приводяться до однієї форми, зменшені до одного розміру (28x28 пікселів). Отримане зображення надходить на вхід нейронної мережі з трьома

внутрішніми шарами та 10 вузлами у верхньому шарі. Нижній рівень мережі підключений не повністю. Підлеглі вузли мають загальний набір ваг. За словами розробників, усе це спрямовано на покращення здатності нижніх мережевих рівнів розрізняти ключові особливості зображення.

Невикористані ваги видаляються, щоб збільшити здатність мережі до узагальнення та зменшити вимоги до обчислень і пам'яті. Це зменшує кількість незалежних параметрів до чотирьох. Нейронна мережа навчена на наборі з 7300 символів і перевірена на наборі з 2000 символів. Похибка розпізнавання становить близько 1% при навчанні і близько 5% при тестуванні [15].

Один із цих методів був запропонований кілька років тому і базувався на використанні трьох модулів (машин) розпізнавання. Першу машину буде навчено нормально. Друга машина вивчає символи, які були виключені першою машиною, тому друга машина має суміш символів, які перша машина отримала на 50% правильних і на 50% неправильних. Ви бачите, що вони є. Нарешті, третій транспортний засіб вивчає символи, які відрізняють результати розпізнавання першого та другого транспортних засобів. Усі три машини використовують символи, розпізнані під час тестування. Буде оцінено роботу всіх трьох транспортних засобів. Розпізнається персонаж з найбільшим загальним балом.

Як правило, алгоритми розпізнавання засновані на поділі растру із зображенням літер ключових ознак, а потім за допомогою штучної нейронної мережі оцінюють близькість вхідного зображення до символів набору заданих букв. Результатом роботи є набір оцінок, які відображають, наскільки розпізнаний символ близький до символів даного набору символів [17]. Набір розпізнаваних символів, який може включати літери та цифри. Вхідні дані для розпізнавання символів перетворюються на один розмір. Відмінною особливістю реалізованого алгоритму є використання нейронної мережі з достатньо великою кількістю вхідних функцій. Вихідне зображення висвітлює основні ознаки, що характеризують різницю яскравості у вузлах растру.

Нейронна мережа має внутрішній рівень, що містить 100 вузлів, і є універсальним зв'язком, тобто кожен внутрішній вузол з'єднаний з усіма вхідними вузлами, а кожен вузол вищого рівня – з усіма внутрішніми вузлами. Щоб зменшити кількість обчислень розпізнавання, для кожного розпізнаваного зображення символу використовуються не всі вхідні символи, а лише його частина, тобто вектор вхідних параметрів нейронної мережі сильно розріджений [18].

Для роботи системи розпізнавання тексту необхідно імпортувати цифровий файл зі сканованим текстом (текстовий елемент). Наступний крок у системі розпізнавання тексту називається «сегментація» та попередня підготовка даних. На цьому етапі роботи відбувається вирівнювання елементів тексту. Текстовими елементами можуть бути зображення з певними текстовими розділами, відскановані розділи сторінок і подібні параметри цифрового макету. Після вирівнювання всі ці елементи необхідно видалити з непотрібних частин, тобто в тексті повинні бути лише ті символи, з якими система «знайома», а решта – звільнити поле зору системи або просто проігнорувати ці символи.

Далі потрібен процес бінаризації, який передбачає відокремлення фону та додаткових читабельних елементів як білих елементів, а необхідний текст вводиться в систему лише як чорний текст. Потім система розпізнавання відокремлює стовпці та рядки тексту, ділить суцільний текст на слова та масштабує зображення для легкого порівняння з усталеними «еталонними» словами та символами.

Надалі розпізнавання тексту виконується символом одним із двох методів — метод «порівняння із зображенням» та метод адаптивного розпізнавання. Кожен з цих методів має певні переваги та недоліки та різні найкращі ситуації для їх застосування [19].

Після отримання результату роботи програми необхідним і дуже важливим кроком є підвищення швидкості роботи програми та підвищення точності отриманих даних. Для цього потрібно додати кілька словників, які



допоможуть системі для певного набору символів складати слова, які найкраще відповідають логіці всього тексту, і виправляти дрібні неточності в словах. Крім того, важливим способом підвищення точності вихідних даних є включення систем розпізнавання символів у правила граматики, які правильно розрізняють іменники та дієслова тощо.

#### 1.4 Аналіз засобів розпізнавання текстових документів

На теперішній час на ринку присутня велика кількість різних програмних систем для аналізу, синтезу, редагування та подальшої обробки цифрових зображень із текстовими фрагментами. В даному параграфі проаналізовано наявне програмне забезпечення для виконання розпізнавання та переведення документів із графічного формату у текстовий.

Більшість сучасних систем OCR діють на трьох рівнях розпізнавання: символів, слів і сторінок. Однак система ABBYY, відповідно до сформованих принципів IPA, ввела у FineReader ще один рівень розпізнавання, який відповідає всьому багатосторінковому документу. У першу чергу все це знадобилося для коректного подання логічної структури документа [8].

Для цього й була спеціально розроблена технологія ADRT (Adaptive Document Recognition Technology), яка здійснює аналіз і синтез документа на логічному рівні. У кінцевому підсумку дана технологія допомагає зробити результат роботи програми FineReader максимально подібним на оригінал [14].

Безумовно лідером є програма для розпізнавання тексту Abbyy FineReader, але для того, щоб повністю підтвердити це, проведемо порівняльний аналіз ще декількох програм для розпізнавання тексту. У результаті маємо отримати можливість, по-перше, оцінити стан сучасного ринку доступних програмних систем для розпізнавання тексту, та по-друге, вказати на сильні і слабкі сторони кожної із цих програмних систем.

ABBYY FineReader 12 — це професійне програмне забезпечення, яке дозволяє розпізнавати текст із графічних файлів і конвертувати його в різні редаговані формати. Програма точно визначає текст і переводить його в новий

формат. Це дозволяє уникнути необхідності перероблення тексту вручну, зменшуючи витрати на обробку. Ця програмна система використовується в компаніях і організаціях, які регулярно працюють з паперовими документами. Abbyy FineReader конвертує файли pdf, tiff, jpeg у pdf, doc, xls, txt тощо.

Результати тестування показують, що ця система дуже добре обробляє документи та перетворює їх із графічного у текстовий формат. Це програмне забезпечення зарекомендувало себе як якісний продукт. Слід також зазначити, що основні поля, з яких у майбутньому будуть витягуватися дані з документа, добре відомі та безпомилкові, а значить, основне завдання оптимізації та автоматизації можна виконати вже зараз.

Плюси: висока якість доставки тексту, широкий вибір форматів вхідних і вихідних документів, простий і зрозумілий інтерфейс користувача. Недоліки: програмна система платна, відкритий доступ до вихідного коду програми відсутній.

Системи, подібні до Abbyy FineReader, включають: SimpleOCR; OmniPage; Прочитати; Клинопис. Free Online Service OCR — це безкоштовний онлайн-сервіс для розпізнавання тексту в різних форматах. Це дуже легко і просто у використанні для будь-якого користувача, немає необхідності завантажувати та встановлювати програму на комп'ютер. Цей продукт підтримує кілька мов для розповсюдження. Текст можна перевести в один із трьох форматів: Microsoft Word, Microsoft Excel і звичайний текст. В основному цих форматів достатньо для швидкого розповсюдження тексту.

З отриманих результатів можна зробити висновок, що цей сервіс має високий рівень доставки тексту. Цей сервіс є хорошим способом швидкого розпізнавання та передачі графічних файлів у текстовому форматі. Однак, через специфічну функціональність цього продукту, його не можна використовувати для реальної оптимізації бізнес-процесів.

Плюси: Простота використання, швидкий доступ до пристроїв з доступом до Інтернету, багато мов для розпізнавання. Недоліки: на відміну від

завантаженої програми оптичного розпізнавання символів, цей продукт не має таких повних функцій.

CuneiForm — це програма, яка використовується для розпізнавання текстових документів і подальшого перетворення їх у редаговану форму. Цей продукт є безкоштовним і доступним для будь-якого користувача. Завдяки програмам можна отримати файл у новому форматі зі зміненим текстом. Виходячи з того, що ми побачили вище, ми дійшли висновку, що ці програмні продукти не підтримують один із найпопулярніших форматів файлів, наприклад Microsoft Word (розширення .doc) і Acrobat Reader DC (розширення .pdf). Таким чином, цей продукт буде програвати в плані використання порівняно з аналогом ABBYY FineReader 12.

Результати тестування показують, що OCR CuneiForm є низькоякісним програмним забезпеченням оптичного розпізнавання тексту. Перекладати основну частину слова на незрозумілий набір символів або взагалі не розпізнавати їх. Середня кількість розпізнаних слів становить трохи менше 90%.

Він досить поганої якості. Переваги: ця програма безкоштовна, з доступом до вихідного коду програми (використовуючи їх, ви можете запустити програму для аналізу графічних документів за допомогою коду, написаного на C#). Недоліки: низька якість розпізнавання. Аналіз виконується без налаштування активних полів для розпізнавання, тобто програми перекладають лише текст у тих місцях графічного файлу, які вони визначили. Суть розпізнавання без коригування полягає в тому, що користувач не витрачає час на вибір того чи іншого поля, тому ми не витрачаємо час на обробку.

Якщо не потрібні сканування, експорт і пакетний режим, ви можете скористатися програмою ABBYY Screenshot Reader. Він захопить будь-яку область екрана та перетворить її на звичайний текст за лічені секунди. Як і його «старший брат», продукт оснащений словником і перекладачем, але коштує в рази дешевше. Безперечно, CuneiForm працює краще, ніж будь-яка інша

безкоштовна служба розпізнавання тексту. Продукт має схожі функції з утилітами АBBYY, але трохи простіше.

Readiris Pro і Freemore OCR можна вважати альтернативами FineReader і CuneiForm для західного ринку. Зображуючи схожість, ви зрозумієте, наскільки програми принципово схожі. Це правда, у Readiris є великий козир: красива система для читання рукописного тексту. Acrobat розроблено для інших цілей, але вбудована функція розпізнавання працює добре, навіть якщо вона менш практична, ніж інші рішення в нашому прикладі.

Scanitto Pro і RiDoc створять документ, зображення або аркуш паперу з символами, які досить швидко читатимуть текст. Утиліта щадна і не вимагає багато ресурсів.

FineReader — комерційний продукт оптичного розпізнавання символів, розроблений компанією АBBYY, беззаперечним лідером у галузі оптичного розпізнавання символів. FineReader базується на запатентованій технології оптичного розпізнавання символів АBBYY OCR, яка ліцензована провідними світовими компаніями, такими як Fujitsu, Panasonic, Xerox і Samsung. Ця програма дозволяє конвертувати зображення друкованих документів у різноманітні редаговані електронні формати. Версія 12 підтримує розпізнавання тексту на 190 мовах, 48 з яких мають вбудовану перевірку орфографії, може зберігати вихідне форматування документа, має багато функцій попередньої обробки зображень і багато налаштувань параметрів. Однак головною його перевагою є майже безпомилкова точність визначення. Зараз програмою користуються понад 20 мільйонів користувачів по всьому світу [5].

CuneiForm — система оптичного розпізнавання текстів компанії CognitiveTechnologies. На даний момент CuneiForm позиціонується як шрифтонезалежна система перетворення електронних копій паперових документів та графічних файлів у редагований вигляд з можливістю збереження структури та гарнітури шрифтів оригінального документа в автоматичному або напівавтоматичному режимі. Система включає дві програми для одиночної і пакетної обробки електронних документів [6].

Існує також система оптичного розпізнавання символів OCRopus, яка, починаючи із версії 0.4, спочатку націлена на перетворення великих обсягів документів в електронний формат, враховуючи власне ядро розпізнавання. Це пакет програмного забезпечення для розпізнавання тексту з відкритим кодом, що розповсюджується за ліцензією Apache 2.0. Ця програма дозволяє підключати додаткові модулі для аналізу макета вашого вмісту та покращення попередніх зображень.

За задумом розробника, за допомогою OCRopus має бути можливість построково визначати текстовий вміст у графічному файлі та конвертувати його у звичайний текстовий формат для подальшого редагування. Крім друкованого тексту програма може розпізнавати і рукописні матеріали.

OCRopus в даний час доступна лише для GNU/Linux. В даний час OCRopus використовує тільки інтерфейс командного рядка, приймаючи вказівки на вхідні зображення з текстом, та виводячи дані у форматі hOCR, TXT [17].

FreeOCR — безкоштовна програма для розпізнавання відсканованого тексту. Вона працює не тільки с файлами зображень, але і з pdf файлами і безпосередньо зі сканером. Для сканування необхідно тільки мати підключений сканер. Основним плюсом програми є її повна автоматизація і відсутність будь-яких налаштувань. Підтримуються формати JPEG, GIF, TIFF BMP і PDF. Також, існує ліміт на 10 зображень у годину. Стверджується, що система здатна розпізнавати більшість східноєвропейських мов, в тому числі українську.

Окремо слід виділити Tesseract — це система розпізнавання текстів, що у розроблялася Hewlett-Packard, а з 2006 є вільною і поширюється компанією Google та дуже стрімко розвивається у останні роки [3].

Tesseract OCR — це система розпізнавання тексту з відкритим кодом (OCR), доступна за ліцензією Apache 2.0. Його можна використовувати безпосередньо або (для програмістів) через API для вилучення друкованого тексту із зображень. Підтримує багато мов. Tesseract не має вбудованого графічного інтерфейсу, але деякі з них доступні на сторонніх сайтах

розробників. Tesseract сумісний з багатьма мовами програмування та фреймворками за допомогою оболонки, знайденої тут. Його можна використовувати з наявним аналізом макета для розпізнавання тексту у великих документах або із зовнішніми детекторами тексту для розпізнавання тексту із зображень окремих рядків тексту.

Tesseract 3.x залежить від багатоетапного процесу, в якому можна виділити етапи: пошук слів; пошук по рядку; оцінки персонажів. Пошук слів виконується шляхом упорядкування рядків тексту в краплі, а рядки та області скануються на наявність тексту фіксованої висоти або пропорційного тексту. Рядки тексту поділяються на різні слова залежно від інтервалу між символами. Потім акредитація відбувається в два етапи. Під час першого проходу ми намагаємося розпізнати кожне слово по черзі. Кожне задоволене слово передається адаптивному класифікатору як навчальні дані. Потім адаптивний класифікатор точніше розпізнає текст унизу сторінки. Оновлення механізму Tesseract — це спроба очистити код і додати нову модель LSTM. Вхідне зображення обробляється за допомогою вирівняних прямокутників, які вводяться в модель LSTM і генерують результати.

Більшість існуючих систем OCR показують прекрасні результати при обробці чорного тексту на світлому (білому) фоні. Що стосується текстових документів, то це не є критичним. Але якщо використати таку систему розпізнавання для обробки нестандартних зображень, таких як паспорти, посвідчення водія або номерних знаків автомобілів, то можна зіткнутися із неможливістю отримати коректні дані без здійснення попередньої обробки зображення текстового документу. Багато проблем, що розглянуті вище, можна усунути на етапі виконання попередньої обробки зображення.

Отже, підбиваючи підсумок розглянутого, до методу оптичного розпізнавання тексту в отриманому зображенні можна висунути такі вимоги:

- шукати та виділяти на цифровому зображенні області із текстом;
- потім у цих областях виділяти окремі рядки, а надалі окремі символи;

— виконувати розпізнавання цих символів і бути нечутливим та стійким по відношенню до способу друку, відстані між рядками у документі й іншими параметрами друку.

Ці завдання мають бути вирішені у даній магістерській роботі.

## **2 РОЗРОБКА ТЕХНОЛОГІЇ ТА ВИБІР ЗАСОБІВ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ**

У даному розділі магістерської роботи розроблена послідовність розпізнавання друкованих символів та сформована нейронна мережа для розпізнавання друкованих символів.

### **2.1 Етапи розпізнавання друкованих символів**

На теперішній час засоби та технології штучного інтелекту широко використовуються у різних галузях діяльності людей. Однією із таких найбільш відомих технологій штучного інтелекту є системи оптичного розпізнавання текстів (OCR-системи). Під системою OCR розуміється автоматичне розпізнавання із використанням спеціальних програм цифрових зображень символів друкованого, машинописного або ж рукописного текстового документу. У деяких випадках під OCR-системою розуміють окремий пристрій для оптичного розпізнавання символів тексту [11].

Системи розпізнавання текстових символів реалізуються як класифікатори, що використовують такі методи, як шаблонні або растрові, ознакові й структурні [12]. Шаблонний класифікатор за допомогою критерію порівняння символу із шаблоном визначається, який із шаблонів слід вибрати із бази. Самий простий критерій порівняння, це мінімум точок, що відрізняють вибраний шаблон від отриманого зображення символу. До переваг цього шаблонного класифікатора можна віднести досить непогане розпізнавання символів із дефектами, простота й висока швидкість розпізнавання. Суттєвим недоліком є необхідність налагодження системи на типи й розміри шрифтів.

У класифікаторі об'єктів аналіз виконується лише для набору чисел або об'єктів, обчислених із зображення. Цей метод дозволяє розпізнавати різні дизайни значків, тобто різні почерки, шрифти тощо. Цей спосіб неминуче призводить до втрати частини інформації через використання топологічного представлення, яке відображає інформацію про взаємне розташування



структурних елементів символу. Ці дані можна представити в графічній формі. Водночас цей спосіб забезпечує незмінність щодо типу та розміру шрифту. Недоліками є значні труднощі з розпізнаванням зіпсованих символів і досить повільна робота.

На теперішній час сформувалася певна послідовність розпізнавання тексту. При розгляді методів та способів розпізнавання текстів можна виділити таку загальну послідовність їх роботи, що складається із таких етапів.

Перший крок — знайти текстове поле, щоб воно виділялося на вихідному зображенні. Для вирішення цієї проблеми використовуються дескриптор НОГ [9], детектор Віоли Джонса, метод класифікації на основі аналізу ознак текстури [10], ознака Томура [11]. Слід зазначити, що сучасні системи оптичного розпізнавання символів погано підходять для завдання виявлення тексту в довільних зображеннях, оскільки спочатку вони були розроблені для роботи із зображеннями, які містять переважно Повністю структурований текст систематично. У кращому випадку це рідко можна побачити на фотографіях і відео, які використовуються в реальному світі. справа.

По-перше, всі системи оптичного розпізнавання символів здійснюють покращення якості та аналіз оброблених зображень. Серед них спеціальні фільтри використовуються для відновлення пошкоджених зображень, такі як використання еліптичного розсіювання [19], фільтри видалення розмитості [20].

На другому етапі по розпізнаванню текстів виконується робота по виявленню та виділенню власне тексту у зображенні. Відомі такі підходи: виділення країв й виділення кутів [12], на основі використання інформації про контури (скелетизації) [13], на основі даних про колір (метод гістограм) [16], методи на основі аналізу інваріантних моментів [15] й різні алгоритми адаптивної бінаризації, у тому числі відомі алгоритми бінаризації від Otsu, Niblack, Bernsan [18]; на основі проведення аналізу текстурної інформації, використання штучних нейронних мереж або експертних систем [19].

Перш ніж виконувати операції розпізнавання тексту на отриманому зображенні, нам потрібно покращити якість зображення, щоб збільшити

ймовірність точного розпізнавання кожного символу на цьому зображенні. Для цього виконайте такі дії: Спочатку виконуються операції з видалення шуму (це можна зробити за допомогою медіанного фільтра) і згладжування елементів зображення (для цього можна використовувати фільтр Гауса). Тепер ми можемо приступити до згладжування меж символів на отриманому зображенні. Тут часто використовуються операції, які постійно розмивають (розмивають) і додатково розширюють (розширюють, розширюють) вихідне зображення. Отримане зображення перетворюється на градації сірого, а для тексту на зображенні виконується корекція бічного світла. Для цих завдань алгоритм SSR може бути використаний як один із модифікованих алгоритмів логарифмічної корекції зображення[19]. Цей алгоритм балансує яскравість зображення, зберігаючи локальний контраст між яскравими та темними ділянками зображення. Алгоритми покращення якості попередньо згладжують зображення, щоб визначити типи присутніх шумів і видалити їх. Сегментація — це процес поділу тексту на рядки, слова та символи.

Лінійний пошук зазвичай базується на періодичності та регулярності текстових полів і виконується на основі методу Хафа, методу зв'язаних компонент [15] та аналізу горизонтальних, вертикальних та діагональних розмірних графів. Сучасні алгоритми спрямовані на вирішення таких завдань: Знайти та розташувати текст, визначити структуру тексту та визначити кут нахилу рядків тексту. Завдання визначення структури документа зазвичай передбачає вибір однорідних блоків тексту, зображень, графіки, таблиць тощо. Алгоритм, який класифікує блоки на текстові та нетекстові, дозволяє визначити, чи є виділений блок текстом, зображенням, графікою, таблицею тощо. Першим етапом будь-якого алгоритму виявлення є етап попередньої обробки.

Ця фаза виконує такі завдання: Покращення якості зображення за допомогою фільтрації, усунення перешкод та інших операцій для покращення якості зображення. На цьому етапі помилка сканування зображення має бути усунена.

І найголовніше, на початку роботи до отриманого зображення часто застосовують фільтр Гаусса для видалення шумів. Важливу роль відіграють порогові бінарні операції, тобто перетворення зображення з кольорового чи сірого у чорно-білий формат [17]. Це дозволяє чітко розділити отримане зображення на текст і фон, що ще більше спрощує використання багатьох алгоритмів, а також усуває деякі шуми на зображенні. Цей крок використовує гістограму інтенсивності отриманого текстового зображення. На цій гістограмі виділяються два піки. Один відповідає білому фону, який насправді є кольором паперу, а піковий діапазон становить .Темніші пікселі відповідають яскравості текстового символу.

Наступним кроком буде вибір тексту на зображенні як області інтересу. На цьому етапі роботи з бінарними зображеннями виділяються області, що містять розпізнаний текст, і видаляються нетекстові елементи в отриманому зображенні [15, 16]. Ці елементи включають такі об'єкти, як кадри зображень без тексту, точки на папері, які не видаляються під час бінаризації. Наприклад, щоб видалити їх, виберіть кріплення на зображенні, обчисліть геометричні елементи та використовуйте кріплення як текстові частини або класифікуйте текстові частини на основі цього. Використовуйте машинне навчання або евристичні методи. підхід. У майбутньому ми почнемо сегментувати та нормалізувати абзаци тексту. На цьому етапі текст розділяється або сегментується на компоненти, придатні для аналізу [17]. Найбільш природною дією на цьому етапі є розбиття тексту на окремі рядки (line split), потім розділення рядків на окремі слова (word split), а потім розділення виділеного слова на складові частини. Основна частина символічної форми.

Крім того, на цьому етапі текст стандартизується, а вибрані компоненти зводяться до стандартного формату, щоб зменшити варіативність і ще більше спростити процес розпізнавання. Етапи сегментації або поділу рядків у надрукованих машиною документах тепер вважаються повністю виправленими. Однак розгляд граничних задач загалом породжує низку складних проблем, які не дозволяють безпосередньо застосовувати відповідні

алгоритми до друкованого тексту. Наприклад, рядки можуть бути не тільки непаралельними, але й кривими, різні рядки можуть бути занадто близькими одна до одної, а текстові елементи, що належать до різних ліній, можуть накладатися.

На цьому етапі виконується сегментація текстового документу до рівня окремих символів. Будемо проводити її за три поетапних кроки.

Виділення рядка - вихідне зображення тексту необхідно «обрізати» на смуги необхідної ширини. Сегментація слова – у зображенні рядка тексту позначаємо зображення слова.

Сегментація символів — ми проводимо межі символів на зображенні слова. Іноді для вибору базової лінії використовується процедура. Така операція використовується в тих випадках, коли при отриманні текстового документа сталася значна деформація внаслідок деформації під час сканування. Ці методи засновані на ідеї, що спочатку потрібно перетнути уявну лінію. У таких методах вони спочатку намагаються наблизитися до цього рядка, а потім відновлюють рядок текстових документів по ньому. У цьому випадку перетворення Хафа використовується для вибору прямих ліній, якщо вони не надто вигнуті. Перетворення Хафа застосовано до центроїдів зв'язних компонентів текстових пікселів [19].

Можливість перетину різних лінійних елементів зі значними дефектами при розпізнаванні сканованих документів є проблемою не тільки для сегментації рядків, а й для розпізнавання тексту, оскільки призначення зображення елемента з погано вираженою лінією значно знизить його впізнаваність. Перехресні компоненти ускладнюють використання методів пошуку горизонтальної проекції (оскільки вони переоцінюють значення профілю проекції в місцях, де повинно бути його мінімальне значення) і методів кластеризації, але мало впливають на деякі методи пошуку.

Щоб знайти пересічні елементи з різних рядків, ви можете використовувати такі функції, як розмір елементів, що з'єднують текст, чи належить елемент до кількох рядків чи до жодної лінії.

Знайшовши такі проблемні компоненти, вам потрібно визначити, чи належать вони до певного сімейства, чи їх потрібно розділити на елементи, що належать до різних сімейств. Вертикальний аналіз таких компонентів є дуже складним завданням. Простішим рішенням є використання горизонтальних ліній для поділу композиції на частини. Однак ви також можете застосувати більш тонкий підхід, наприклад, виділити окремі штрихи. Наступним важливим кроком є виконання операції поділу слова. На цьому етапі системи розпізнавання виділені рядки текстового документа розбиваються на окремі слова. На відміну від друкованого тексту, де інтервал між словами приблизно постійний, а інтервал між літерами в слові набагато менший, ніж інтервал між словами, в інших типах текстових документів розмір інтервалу між словами значно змінюється. Існує ймовірність.

Виділені текстові елементи зв'язку, призначені рядку на попередньому етапі системи розпізнавання, на цьому етапі будуть об'єднані в слова. Іншим кроком у системі розпізнавання текстових документів є нормалізація отриманого зображення. Через високу варіативність у застосуванні вербальних символів їх розпізнавання є досить складним процесом [19]. Мета нормалізації полягає в тому, щоб привести слово або кожен символ у слові до стандартної форми без значної втрати інформації, необхідної для наступного етапу розпізнавання. Одним із найпоширеніших методів нормалізації є виправлення градієнта слова вздовж горизонтальних і вертикальних ліній і доведення його до певних заданих розмірів. Найпростіший спосіб закріпити слово по горизонталі - це максимально збільшити яскравість зображення в певному діапазоні. Існують інші методи нормалізації, такі як зміна розміру та виділення текстових кадрів, але вони використовуються рідше. Існують також інші методи, наприклад, засновані на згладжуванні та лінійній регресії [20].

Існує два алгоритми для отримання опису ознак символу: один аналізує оригінальне зображення символу як ознаку, а інший аналізує обчислені характеристики, такі як вертикальні хорди [14], приблизні контури та рамки маркування [15], класифіковані за класами.

Щоб зменшити розмірність простору ознак, часто використовуються методи головних компонент і лінійний дискримінантний аналіз. Для розпізнавання символів реалізовано різні класифікатори на основі нейронних мереж [13] і згорткових нейронних мереж [7, 19], які наразі є найпопулярнішими.

Словникова перевірка здійснюється на основі стандартних або динамічно створюваних мовних словників, n-грам, реалізованих у вигляді списків, дерев або графіків [12, 16].

Отже, процес розпізнавання текстового документу повинен включати такі етапи роботи.

Пошук області зображення, що містить текст, та його локалізація. Попереднє покращення якості локалізованої області зображення та її бінаризація. Виявлення структури знайденого та виділеного блоку тексту у зображенні та визначення порядку прочитання. Сегментація тексту на рядки, слова й символи. Формування ознакового опису кожного символу зображення. Розпізнавання окремих символів текстового зображення. Словникова перевірка для поліпшення якості розпізнаного текстового фрагменту.

Зображення, що надходить на вхід системи розпізнавання, повинно бути очищено від можливих завад й приведене до такого вигляду обробки, що дозволяє ефективно виділити окремі символи тексту й розпізнавати їх.

Програмний засіб повинен розділити зображення на окремі фрагменти тексту, наприклад, у вигляді абзаців, використовуючи особливості його вирівнювання й розподілу по декількох колонках.

Надалі зображення із виділеним текстом має бути розділене на зображення рядків тексту, а потім на зображення слів та окремих символів для того, щоб у подальшому виконати розпізнавання кожного із символів окремо. Після виконання даного кроку виділення символів різні системи розпізнавання текстових документів працюють за своїми певними їм специфічними алгоритмами.

Виділене зображення символу може оброблятися як єдиний цілий елемент, для цього можна скористатися порівнянням із наявними шаблонами, можна виконати розпізнавання за допомогою штучних нейронних мереж або ж виділити попередньо визначені певні характеристики кожного із символів для розпізнавання.

На виході такого кроку роботи засобів розпізнавання з'являється можливий варіант символу у вигляді літер, цифр або інших елементів. Як правило, засоби розпізнавання на цьому не зупиняються й продовжують роботу по аналізу текстового документу на основі використання інших методів та підходів, послідовно деталізуючи та уточнюючи отриманий результат роботи.

Завершальним етапом може бути використання словника для остаточного уточнення виділеного слова із наявною множиною слів у вибраному словнику.

## 2.2 Розробка послідовності розпізнавання друкованих символів

Послідовність отримання зображення для розпізнавання тексту складається із декількох базових етапів, таких як виділення області тексту у отриманому зображенні, попередня обробка виділеного фрагмента зображення, сегментація, виділення ознак, класифікація та розпізнавання символів тексту та завершальна обробка (так звана пост-обробка). Загальна послідовність обробки зображення приведена на рисунку 2.1. Ці етапи покладемо в основу формування технологічної послідовності обробки зображення для виділення та розпізнавання друкованих символів.

Засіб розпізнавання тексту отримує зображення текстового документа у якості початкового вхідного зображення. Зображення повинне бути подане у певному форматі, такому як BMP, JPEG, PNG або іншому. Таке зображення може бути отримане із використанням сканера, цифрової камери або ж будь-якого іншого пристрою формування та введення цифрового зображення.

Наступний етап — це попередня обробка. Вона складається із ряду операцій, що здійснюються над отриманим вхідним зображенням. Це значно

покращить і сегментує зображення. Послідовність операцій, які виконуються над зображенням, наведена на рисунку 2.1.

Процес бінаризації використовує методи глобального порогового значення для перетворення чорно-білих зображень у двійкові. Розширення країв бінаризованого зображення виконується шляхом відбілювання. Масштабування зображення та заповнення отворів — це операції, які виконуються на останніх двох етапах для отримання попередньо оброблених зображень, придатних для сегментації.

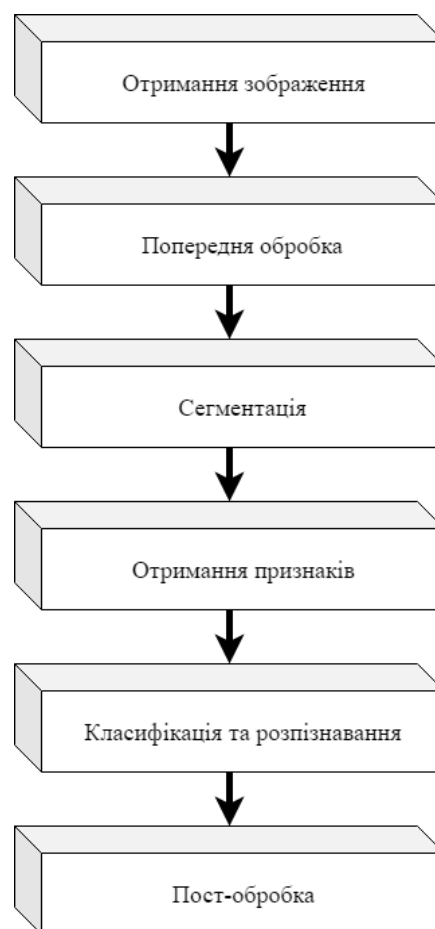


Рисунок 2.1 — Загальна схема розв’язання задачі розпізнавання

Необхідно, щоб дані пройшли кілька етапів попередньої обробки, які будуть використані на етапі опису символічного аналізу. Метою попередньої обробки є отримання даних, придатних для системи оптичного розпізнавання



символів. Основним завданням попередньої обробки є фільтрація шумів, посилення контрастності та бінарізація зображення.

Під час сканування документів відскановані зображення можуть мати додатковий шум, і ці зображення низької якості вплинуть на наступний етап обробки документів. Тому етап попередньої обробки необхідний для покращення якості зображення перед передачею його на наступні етапи обробки документа. Шум може спричинити хибні виняткові сегменти, розриви довгих рядків тощо, тому важливо видалити всі ці помилки, щоб отримати кращу інформацію. Зображення мають багато типів шуму. Додатковий шум називається «шум солі та перцю», що складається з чорних і білих крапок, розкиданих по всьому зображенню, яке зустрічається в більшості документів. Методи зменшення шуму можна розділити на дві основні групи: фільтрація та морфологічна обробка.

Метою фільтрації є усунення шуму та зменшення зайвих точок, які зазвичай виникають через нерівні поверхні запису та/або низьку частоту дискретизації пристрою збору даних. Морфологічні операції часто використовуються як інструмент обробки зображень для отримання елементів зображення, корисних для представлення та опису форми області. Операції трансформації можна успішно використовувати для видалення шумів із зображень документів, спричинених низькою якістю паперу та чорнила, а також сторонніми рухами рук. Бінарізація чорно-білих зображень символів є важливим кроком у розпізнаванні символів офлайн. Хороша двійкова система полегшує сегментацію та розпізнавання символів.

Процес бінарізації перетворює оригінальне зображення на двійкове. Сегментація зображення ділить послідовність символів на окремі підсторінки символів. У запропонованій системі попередньо оброблене вхідне зображення поділяється на окремі символи, і процес маркування використовується для призначення номера кожному символу. Це позначення надає інформацію про кількість символів у зображенні. Ребра характеризують межі об'єктів і тому корисні для сегментації об'єктів, реєстрації та виявлення. Виявлення країв

зображення значно зменшує обсяг даних і відфільтровує непотрібну інформацію, зберігаючи важливі структурні характеристики зображення.



Рисунок 2.2 — Алгоритм попередньої обробки зображення

Існує багато способів визначення країв. Однак більшість різних методів можна розділити на дві категорії: градієнтні методи та методи Лапласа. Метод градієнта визначає краї шляхом знаходження максимального та мінімального значень першого похідного зображення. Метод Лапласа знаходить ребра шляхом пошуку перетинів нуля у другому похідному зображенні.

Виявлення спотворень — можлива асиметрія через сканування. Існує кілька поширених способів виявлення зміщення сторінок. Деякі компоненти покладаються на ідентифікацію зв'язаних компонентів і знаходження

середнього кута, що сполучає їхні центри. Асиметрія знижує точність документа, і її слід усунути. Кут нахилу обчислюється, і лінія нахилу проводиться горизонтально до кута нахилу. Сегментація є найважливішим процесом у методах розпізнавання символів. Сегментація ділить зображення на окремі символи.

Розподілити рукописні слова на різні області (верхню, середню, нижню) і літери складніше, ніж у друкованих документах. Насамперед це пов'язано з відмінностями між символічним інтервалом, нахилом, нахилом і розміром.Х.

Рукописний напис. У деяких випадках компоненти двох послідовних символів можуть бути задіяні або накладатися, і ця ситуація ускладнює завдання сегментації.В індійських мовах цей контакт або накладення часто відбувається через чергування символів верхньої та нижньої зони.Сегментація є важливим кроком, оскільки ступінь поділу слів, рядків або символів безпосередньо впливає на швидкість розпізнавання тексту.

Існують два типи сегментації — зовнішня та внутрішня. Зовнішні сегменти формуються шляхом розбиття макету сторінки на логічні блоки. Потім йде поділ різних блоків тексту на, наприклад, абзаци, речення або слова. Це є найважливішою частиною аналізу документів.

Аналіз та розпізнавання документів має на меті автоматичне вилучення інформації, представленої на папері, та вирішення її відповідно до людського розуміння з самого початку. Сегментація сторінки є важливим кроком в аналізі макета, але це особливо складно, коли ви маєте справу зі складними макетами. Макети сторінок аналізуються в два етапи.

Першим кроком є структурний аналіз, де зображення ділиться на блоки елементів документа (абзаци, рядки, слова тощо).

Другий — функціональний аналіз. Він визначає макет, розмір і функціональний зміст елементів документа за допомогою різних правил розмітки.

Потім сегментуйте сторінку, шукаючи області з білою та чорною текстурою. Внутрішня сегментація — це операція, яка намагається розділити

рядок на субфрейми окремих символів. Хоча за останнє десятиліття ці методи різко розвинулися та з'явилося багато різних методів, розділення рукописних шрифтів на символи залишається відкритою проблемою.

Крок класифікації на основі виділення ознак символів є частиною процесу прийняття рішень у створеній послідовності розпізнавання. Для класифікації й розпізнавання друкованих символів використовується нейронна мережа. Сегментоване текстове зображення із останнього кроку подається на вхід нейронної мережі. Загальна кількість нейронів у вихідному шарі мережі становить 44, оскільки запропонований програмний засіб призначений для виявлення та розпізнавання текстових документів на основі кириличного алфавіту.

Етап після обробки (пост обробки) є передостаннім етапом запропонованої послідовності розпізнавання. Тут виявлені та розпізнані символи подаються в структурованому тексті шляхом визначення відповідного еквівалентного значення у кодовій таблиці ASCII із використанням індексів розпізнаних символів тестового шаблону.

Завершальним етапом роботи системи розпізнавання символів є використання словника для остаточного уточнення виділеного слова шляхом його порівняння із наявною множиною слів у даному словнику.

### 2.3 Використання згорткової нейронної мережі

Для розпізнавання символів виділеного текстового зображення пропонується використати згорткову нейронну мережу.

Згорткові нейронні мережі (CNN) — це клас глибоких штучних нейронних мереж прямого поширення, які застосовуються для аналізу зображень. Ці мережі створюються за аналогією з біологічними процесами, вони є варіантами багатшарових персептронів і вимагають мінімальної попередньої обробки. ШНМ — це тип багаторівневої нейронної мережі, яка називається «згорткова мережа» від назви операції згортки. ШНМ складається з вхідного та вихідного шарів, а також кількох прихованих шарів.

Зазвичай вхідний шар просто перетворює 2D-зображення в 3D-матрицю. Таким чином, тривимірна матриця подається на вході окремого шару, але її розмір можна змінювати на виході шару.

На відміну від загальної моделі багаторівневої нейронної мережі (MNN), приховані шари ШНМ часто є спеціалізованими і включають згорткові шари, шари об'єднання, повністю зв'язані шари, шари зменшення дискретизації та шари вибірки.

Загальна структура ЗНМ формується із багатьох шарів. Вхідне зображення подається на перший шар, а надалі сигнал проходить через послідовну серію згорткових шарів, де він поступово згортається й субдискретизується. Ви можете створити карту об'єктів, чергуючи шари. З кожним наступним рівнем розмір цієї карти зменшується, але при цьому збільшується кількість каналів. Після проходження через певну кількість шарів карта об'єктів перетворюється на скалярну або векторну, і ЗНМ створює сотні таких карт об'єктів. Після проходження таких згорткових шарів вводиться кілька наступних шарів перцептрона (повністю зв'язаної нейронної мережі), на вхідні дані яких подається остаточна сформована карта ознак.

На рисунку 2.3 показана типова архітектура згорткової нейронної мережі.

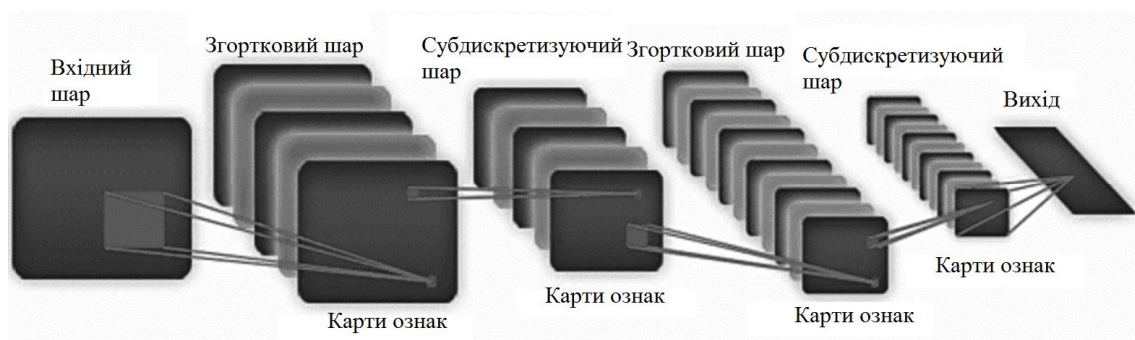


Рисунок 2.3 – Структура згорткової нейронної мережі

Основними компонентами такого типу штучних нейронних мереж є згорткові шари, вони застосовують до вхідного зображення операцію згортки й передають результат для опрацювання наступному шару мережі. Згортка імітує

реакцію окремого взятого нейрону на зоровий стимул. У кожного згорткового шару є такі базові характеристики:

- $F$  — вказує на розмір рецептивного поля, яке відповідає за кількість нейронів у попередньому шарові, що потім будуть зв'язані із одним нейроном у цьому шарі мережі;

- $K$  — вказує на глибину, тобто на кількість нейронів у цьому шарі (що характеризує глибину матриці на виході мережі);

- $S$  — крок мережі, який контролює ступінь перекриття рецептивних полів;

- $P$  — вводиться розмір нульового доповнення, що здійснює контроль розміру вихідної матриці мережі.

Якщо виникне потреба, то вхід на краях зображення буде доповнено нулями. У мережі кожен нейрон згорткового шару представляється фільтром (ядром). Цей фільтр є матрицею, що має розмір  $F_1 \times F_2 \times F_3$ , її параметри можуть змінюються під час навчання мережі. Із використанням цього фільтру формується двомірна карта активації мережі, яка є зрізом вихідної матриці згорткового шару.

У ЗНМ використовуються спільні ваги у згорткових шарах, що вказує на те, що для кожного рецептивного поля шару мережі використовується один й той же фільтр у вигляді банку ваг, це зменшує обсяг необхідної пам'яті й поліпшує продуктивність роботи мережі .

У згортковому шарі нейронної мережі над фрагментами зображення, які подаються на вхід цього шару, здійснюється операція згортки шляхом обчислення зваженої суми фрагментів аналізованого зображення навколо заданого та визначеного ядра згортки:

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

де  $f$  — початкова матриця аналізованого зображення;

$g$  — власне ядро згортки зображення;

$k$  — порядковий номер рядка ядра згортки;

$l$  — порядковий номер стовпця ядра згортки.

Основа роботи таких ЗНМ ґрунтується на послідовному чергуванні шарів згортки (convolution layers) й шарів субдискретизації (subsampling layers або ж інша назва pooling layers). Структура такої нейронної мережі є однонаправленою, у ній відсутні зворотні зв'язки та у її основі покладено принцип використання багатьох шарів. При навчанні такої ЗНМ застосовується метод зворотного поширення помилки у шарах мережі.

Робота ЗНМ базується на принципі переходу від конкретних рис аналізованого зображення до розпізнавання більш абстрактних деталей і понять більш високого рівня. Як правило, ЗНМ налаштовуються, щоб сформувати бажану ієрархію абстрактних характеристик, відкидаючи неважливі деталі та підкреслюючи важливі. Функціонування ЗНМ забезпечується двома основними елементами: роллю фільтра, який виявляє конкретні особливості, і карти функцій, яка описує аналізоване зображення.

Базова модель ЗНМ складається з трьох типів шарів: згортковий шар, шар підвибірки та «звичайний» шар нейронної мережі у формі персептрона. Персептрон — це штучний нейрон, який фактично являє собою повноцінну нейронну мережу, де кожен наступний нейрон у мережі пов'язаний з усіма нейронами попереднього рівня, де ці зв'язки мають власні вагові коефіцієнти. У такій нейронній мережі операція згортки виконується невеликою ваговою матрицею, яка переміщається по всьому вхідному зображенню, представленому цим нейронам, і формує сигнал активації, що активується після кожного зсуву для нейрона наступного шару з подібною позицією. Ця вагова матриця для різних нейронів у вхідному шарі використовується як певне ядро згортки. Його представлення — це графічне кодування певної характеристики зображення. Наступний шар, утворений операцією згортання цієї вагової матриці, вказує на наявність певного об'єкта, ідентифікованого в обробленому шарі, і вказує його координати, таким чином формуючи карту характеристик.

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. Під час редагування шару за допомогою вагової матриці шар зазвичай переміщується на невелику фіксовану відстань, а не на розмір самої матриці. Наприклад, якщо вагова матриця має розміри 5x5, вона буде зміщена на 1 або 2 нейрони замість 5, щоб уникнути втрати або пропуску функцій, які ви шукаєте. Рівень згортки застосовує операцію згортки до вхідних даних і передає результат до наступного рівня. Ядром згорткової мережі є певний набір вагових коефіцієнтів. Результатом цієї операції є зображення формі карти об'єктів.

Залежно від матриці, вибраної для операції згортання, карта функцій виділяє певні особливості обробленого вхідного зображення. Щоб отримати найкращі характеристики вхідного зображення, використовується кілька різних ядер згортки, а на виході згорткового рівня мережі формується кілька карт з різними характеристиками. Ядро згортки насправді є фільтром або вікном, яке переміщується по попередній карті, щоб вибрати певні властивості об'єкта.

Згортка моделює реакцію одного нейрона на візуальний стимул. Кожен згортковий нейрон обробляє дані лише для свого рецептивного поля. Параметри рівня складаються з набору навчальних фільтрів (або ядер), які мають невелике сприйнятливим поле, але охоплюють всю глибину вхідного контейнера. Під час підгонки кожен фільтр масштабується відповідно до ширини та висоти вхідного об'єму, а також обчислюється скалярний добуток фільтра та вхідних даних, де створюється двовимірний збудження фільтра. Це дозволяє мережі дізнаватися, які фільтри активуються, коли певний тип функції виявляється в певному просторовому місці входу. Агрегування карт збудження всіх фільтрів за глибиною максимізує продуктивність згорткових шарів. Таким чином, кожен запис у вихідному відсіку також можна розуміти як



вихід нейрона, який реагує на невелику вхідну область і має спільні параметри з нейронами в одній карті стимулів. Рівень згортки застосовує операцію згортки до вхідних даних і передає результат до наступного рівня. Ядром згорткової мережі є набір конкретних вагових коефіцієнтів. Результатом цієї операції є зображення у формі карти об'єктів.

Залежно від матриці, вибраної для операції згортання, карта функцій виділяє певні особливості обробленого вхідного зображення. Щоб отримати найкращі характеристики вхідного зображення, використовується кілька різних ядер згортки, а на виході згорткового рівня мережі формується кілька карт з різними характеристиками. Ядро згортки насправді є фільтром або вікном, яке переміщається по попередній карті, щоб вибрати певні властивості об'єкта. Згортка моделює реакцію окремого нейрона на візуальний стимул. Кожен згортковий нейрон обробляє дані лише для свого рецептивного поля. Параметричний рівень складається з набору навчальних фільтрів (або ядер), які мають невелике сприйнятливим поле, але охоплюють всю глибину вхідного діапазону. Під час передачі кожен фільтр масштабується на основі ширини та висоти вхідного контейнера та відображається шляхом обчислення скалярного добутку фільтра та вхідних даних.

Таким чином, мережа знає, які фільтри активуються, коли певний тип функції виявляється в певному просторовому розташуванні на вході. Агрегування карт збудження всіх фільтрів за глибиною забезпечує повну потужність згорткового шару. Таким чином, вхід до вихідного відсіку також можна розуміти як вихід нейрона, який реагує на невелику вхідну область і має спільні параметри з нейронами в одній карті стимулів.

Рівень підвибірки в мережі — це операція нелінійного стиснення (нелінійного перетворення) отриманої карти ознак, у той час як певна група пікселів (зазвичай розміром  $2 \times 2$  пікселя) стискається в один піксель.

Зазвичай для виконання цього перетворення використовується функція пошуку максимуму. Для перетворення вибираються прямокутники або квадрати, які не перекриваються, і кожен прямокутник замінюється пікселем,

вибраним як піксель із найбільшим значенням у сегменті аналізованого зображення.

Остаточна операція субдискретизації може значно зменшити просторовий розмір аналізованого зображення. Цю операцію можна зрозуміти так: якщо на попередньому кроці згортки були виявлені певні особливості зображення, то для наступного кроку обробки це зображення більше не потрібне, і обробка зображення під час операції стиснення виконується на нижчому рівні. Крім того, виконання операції фільтрації непотрібних на даний момент деталей зображення допомагає мережі не навчатися заново.

Шар зменшення дискретизації (недобірки) змінює розмір карти функцій. Це дозволяє спростити обчислення, а також може запобігти перетренованості. Вважається, що наявність знака важливіше точного розуміння його координат. Група пікселів (зазвичай розміром  $2 \times 2$ ) стискається в один піксель шляхом нелінійного перетворення. Таким чином, фільтр  $2 \times 2$  застосовується до кожного фрагмента вхідної матриці, а 4 числа замінюються їх найбільшим числом. Результатом є напіввимірна матриця [15]. Це перетворення впливає на прямокутники чи квадрати, які не перетинаються. Ця операція об'єднання значно зменшує просторовий об'єм зображення.

Крім стиснення з функцією максимуму, використовуються і інші функції - середнє значення, нормалізація L2, але як показує практика, найкращим є об'єднання з функцією максимуму.

Згортковий шар і шар зниження дискретизації чергуються, а шар зниження дискретизації вставляється після попереднього згорткового шару перед наступним згортковим шаром. Після кількох проходів згортання та стиснення зображення шляхом виконання операції зменшення дискретизації мережа переформатується з конкретної сітки пікселів високої роздільної здатності в більш абстрактно згенеровані карти функцій, із кожним наступним шаром, що збільшує кількість каналів і зменшує розмір зображення.

В результаті ми отримуємо великий набір вихідних каналів, які тепер зберігають невелику кількість даних або навіть один параметр. Унікальні

числа, що розуміються як найбільш абстрактні поняття, виявлені з початково проаналізованого зображення.

Після декількох послідовних чергувань згорткового та підвибіркового шарів використовують один або декілька повнозв'язних шарів (fullyconnected) нейронів. Нейрони у повнозв'язному шарі з'єднані з усіма нейронами попереднього шару, як у звичайних нейронних мережах прямого поширення. 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.

Повністю зв'язаний рівень з'єднує кожен нейрон одного шару з кожним нейроном наступного рівня та надає мітку високого рівня. Такий шар є різновидом шару традиційної багатошарової нейронної мережі (МНМ). Використовуйте повністю зв'язаний шар, який визначається на основі отриманих ознак. Цей рівень працює точно так само, як традиційна багатошарова нейронна мережа персептрона. Повністю з'єднані шари з'єднують кожен нейрон одного шару з кожним нейроном наступного шару. Рівень злиття ZNM спрямований на локальне або глобальне злиття та об'єднання виходу кластерів нейронів одного шару з нейронами наступного рівня.

Агрегація — це тип нелінійної підвибірки з ідеєю, що точне розташування об'єкта не так важливо, як його приблизне розташування відносно інших об'єктів. Рівень об'єднання має ефект поступового зменшення просторового розміру представлення, щоб зменшити кількість параметрів і обчислювальних зусиль у мережі та, таким чином, також контролювати перенавчання. В архітектурі ZNM зазвичай періодично вставляють шар об'єднання між послідовними згортковими шарами. Глобальні операції забезпечують інший тип інваріантності, ніж паралельна передача. Рівень об'єднання працює незалежно на кожному вхідному зрізі глибини та зменшує

його просторовий розмір. Максимальне об'єднання (максимальне об'єднання) використовує максимальне значення кожного кластера нейронів із попереднього рівня. Він розділяє вхідне зображення на набір прямокутників, які не перекриваються, і відображає максимальне значення для кожної з цих субрегіонів.

Усічений лінійний одиничний рівень (рівень випрямленої лінійної одиниці, ReLU) застосовує функцію передачі ненасиченого ReLU.

Це покращує нелінійні властивості функції прийняття рішення та всієї мережі, не впливаючи на сприйнятливі поля згорткового рівня. Інші функції, такі як насичений гіперболічний тангенс і сигмоїдна функція, використовуються для покращення нелінійності. Зрівняльний шар зменшує просторовий розмір входу до розміру каналу. Рівень втрат визначає, як потяг обробляє різницю між прогнозованими та фактичними мітками класу, і зазвичай є останнім шаром. Для різних завдань можна використовувати різні функції втрат. Нормовані експоненціальні втрати (softmax) застосовуються для прогнозування одного класу серед  $K$  взаємовиключних класів. Сигмоїдна втрата крос-ентропії використовується для прогнозування  $K$  незалежних значень ймовірності в інтервалі  $[0, 1]$ . Евклідові втрати застосовуються до регресії на мітках із дійсними значеннями в діапазоні  $(-\infty, +\infty)$ .

Основна відмінність між ZNM і BNM полягає в наявності просторової організації (тривимірної ємності нейронів) у CNM, локальних зв'язків між нейронами в суміжних шарах для забезпечення просторової локальності та виявлення ознак незалежно від загальних параметрів (ваги) для їх положення в полі зору та рівень зв'язку.

Основна відмінність ZNM від BNM полягає в наявності просторової організації (тривимірної ємності нейронів) і локальних зв'язків між нейронами суміжних шарів у CNM для забезпечення локальності. Просторове розташування, глобальні параметри (ваги) дозволяють ідентифікувати ознаки незалежно від положення в полі зору та в межах пов'язаного шару. Поєднання цих властивостей дозволяє CNM краще узагальнювати візуальні завдання.

Зважений розподіл значно зменшує кількість вільних параметрів, які вивчає мережа, зменшує пам'ять, необхідну для роботи мережі, і дозволяє навчати більші та потужніші мережі. Традиційне зворотне розповсюдження помилок дозволяє навчати згорткову нейронну мережу, вибираючи ваги з фільтра. Навчання ZNM виконується з використанням техніки зворотного поширення помилок.

У даному розділі була сформована послідовність обробки текстового документу та описані вимоги до згорткової нейронної мережі, яка буде виконувати розпізнавання виділених символів у текстовому документі.

### **3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ДРУКОВАНИХ СИМВОЛІВ**

Процес виділення і розпізнавання окремо взятих друкованих символів сканованих текстових документів виконується за ряд етапів, послідовність виконання яких та їх реалізація у вигляді програмного продукту розглянуті у даному розділі магістерської роботи.

#### **3.1 Розробка архітектури програми розпізнавання символів**

На основі запропонованої послідовності обробки зображення із текстом у структурі програмного засобу мають бути модулі, що послідовно із отриманого зображення текстового документа виділяють сторінки, потім у них виділяють текстовий фрагмент і виконують сегментацію виділеного фрагменту із текстом на абзаци, рядки, слова і символи і потім виконують розпізнавання символів. У нашому випадку процес розпізнавання символів покладається на нейронну мережу. Таку нейронну мережу необхідно попередньо налаштувати на розпізнавання символів тексту. Частина текстового документа, у якій символи будуть не розпізнані, маємо виконати із залученням додаткових засобів.

Отже, виходячи із такої послідовності розпізнавання друкованих символів, програмний засіб повинен складатися із таких модулів: модуль виділення сторінок, модуль попередньої обробки текстового фрагменту зображення, модуль поділу текстового фрагменту на абзаци, рядки, слова і символи, модуль розпізнавання символів на основі нейронної мережі та модуль для остаточного розпізнавання друкованих символів із деякими дефектами. Також є ще один модуль для навчання згорткової нейронної мережі на виконання процедури розпізнавання друкованих символів та модуль виведення результатів розпізнавання символів.

На модуль попередньої обробки зображення покладається виконання таких функцій. Отримуємо фрагмент зображення із друкованим текстом, у цьому фрагменті виконуємо фільтрацію від можливих завад й морфологічну

фільтрацію зображення, виконуємо переведення зображення із кольорового у бінарне подання та послідовно виділяємо ті області, що відповідають окремим сторінкам текстового документу. На наступному кроці виконуємо розподіл виділеної сторінки текстового документу на окремо взяті рядки із залученням гістограм розподілу яскравості отриманого зображення на вертикальну проекцію. Наступним кроком є розподіл рядків на окремі слова із залученням гістограм розподілу яскравості виділеного рядка на горизонтальну проекцію. Надалі переходимо до розділення виділених слів на окремі символи й визначення їх місця розташування. Після виконання операції поділу слів на окремі символи тексту проводимо їх нормалізацію до бажаних необхідних розмірів для виконання операції подальшого розпізнавання із використанням нейронної мережі. На цьому етапі процес виконання розпізнавання символів тексту здійснюємо із підключенням згорткової нейронної мережі глибокого навчання. Якщо через значну ступінь спотворення певний символ на деякій позиції не був розпізнаний, то тоді приступаємо до виконання роботи наступного модуля нашої програми: етапу розпізнавання символу із використанням порівняння виявленого деформованого символу із набором шаблонних символів.

Розпізнавання символів із дефектами виконується у такій послідовності. При роботі цього модуля програми спочатку визначаємо тип алфавіту, який був задіяний на попередньому етапі розпізнавання тексту. Визначаємо ширину деформованого символу у отриманому фрагменті зображення. По значенню цієї ширини символу вибирається набір символів із бази даних уже визначеного алфавіту. Послідовно проводиться перебір і порівняння цих шаблонів із вибраної еталонної групи символів. При кожному накладанні обчислюється коефіцієнт збігу для даного примірника символу із вибраної групи. Таким чином буде визначено коефіцієнт збігу між отриманим й еталонним символами. Коефіцієнт збігу обчислюється шляхом виконання операції логічного множення аналізованого символу із кожним символом із вибраної групи еталонних символів. Аналогічним способом визначається коефіцієнт відхилення фону

виділеного символу від отриманого фону фрагменту зображення, також шляхом логічного множення, але тепер уже фону отриманого фрагменту зображення по відношенню до фону еталонного зображення. Таким чином визначаємо коефіцієнт відхилення фону отриманого символу тексту від фону вибраного еталонного символу. Порівняння різниць таких коефіцієнтів дає можливість вибрати із розглянутої групи еталонних символів той символ, у якого є найбільше значення різниць коефіцієнтів із отриманої групи і тоді даний символ є найбільш подібним із аналізованим символом.

Модуль для навчання згорткової нейронної мережі для здійснення процесу розпізнавання символів для свого налаштування використовує бази еталонних символів Letter Dataset, EMNIST Dataset [23] та сформований набір символів, які дозволили виконати навчання нейронної мережі для здійснення операцій розпізнавання друкованих та у деякій мірі й рукописних символів.

Для реалізації запропонованих підходів по розпізнаванню друкованих символів було вибрано таку досить поширену 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 роботи програміста та має прекрасну читабельність коду. Синтаксис ядра Python є мінімалістичним. У 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. Враховуючи такі особливості цієї мови вона була вибрана для створення програмного продукту.



Розроблена послідовність розпізнавання друкованих символів текстового документа приведена у додатку Г даної магістерської роботи.

### 3.2 Попередня обробка текстового зображення

Перед початком виконання дій по переведенню отриманого файлового документу у окремі сторінки імпортуємо деякі стандартні функції із бібліотеки `numpy` та `image`. Створюємо функцію `osr` для розділення початкового текстового документу у форматі `pdf` на окремі сторінки. Перевіряємо, чи поданий файл документу сформований у форматі `pdf`, якщо ні, то переводимо його у такий формат.

```
def ocr_pages(file):
    if not is_this_pdf(file):
        file = create_pdf_dummy(file)
        file = compose_pdf(file)
```

Формуємо пустий масив для зберігання майбутніх результатів розпізнавання `ocr_res = []` і конвертуємо його у масив сторінок.

Виділяємо кожну сторінку, визначаємо ширину і висоту кожної із цих сторінок, отримуємо результат з моделі відповідно до зображення, де `OcrData` - увесь текст, що знайдений, `X`, `Y` - відповідно ширина і висота виділеного тексту та `TextBlocks` - рамки знайденого тексту.

Перевіряємо чи вхідний файл є `.pdf` файлом із виконанням такої послідовності дій

```
def is_this_pdf(file):
    for byte_ind in range(min(len(file), 1024)-3):
        if file[byte_ind]==0x25 and file[byte_ind+1]==0x50 and
file[byte_ind+2]==0x44 and file[byte_ind+3]==0x46:
            return True
    return False
```

Наступним етапом виділення зображення із текстом є переведення його у бінарне представлення, знаходження меж області тексту та приведення виділеного зображення до бажаного розміру. Для реалізації даного модуля програми скористаємося бібліотеками `numpy` та `image`. Для здійснення процесу бінаризації використаємо функції `convertToBW` і `toggleOnesAndZeros`, при цьому чорному кольору буде відповідати рівень логічної 1, а білому рівень логічного 0.

```
def convert ToBW (image In):
    blackAndWhite = image In. Convert ('1')
    blackAndWhite = np.array (blackAndWhite)*1
    return blackAndWhite
def toggleOnesAndZeros (blackAndWhite)
    return (blackAndWhite^1)
```

У процесі попередньої обробки текстового зображення скористаємося функцією визначення границь області зображення із текстом, для чого виділимо межі правої, лівої, верхньої і нижньої меж зображення. Для виділення лівої та правої межі зображення тексту скористаємося такою послідовністю дій.

```
def crop (blackAndWhiteToggled):
[numbler OfRowPixels, numbler OfColumnPixels] = blackAndWhite Toggled.shape
vertikalSumOfBlackPixels = np.Sum (blackAndWhiteToggled, axis = 0)
leftDetected = False
for i in range (0, numberOfColumnPixels):
    if vertical SumOfBlackPixels [i] > 0 and leftDetected == False:
        leftDetected == True
        left = i
    elif vertical SumOfBlackPixels [i] > 0 and leftDetected == True
        right = i
```

Для виділення верхньої та нижньої межі зображення тексту скористаємося такою послідовністю дій.

```
horizontalSumOfBlackPixels = np.Sum (blackAndWhiteToggled, axis = 1)
topDetected = False
for i in range (0, numberOfRowsPixels):
    if horizontalSumOfBlackPixels [i] > 0 and topDetected == False:
        topDetected == True
        top = i
    elif horizontalSumOfBlackPixels [i] > 0 and topDetected == True:
        botton = i
```

Завершальним етапом у процесі попередньої обробки зображення є виконання нормалізації, за допомогою якої виконаємо зміну зображення до бажаного розміру. Для виконання цієї процедури по зміні розміру зображення скористаємося функцією `resize` із бібліотеки `image`. Одним з аргументів цієї функції є застосування фільтру `resample`, який слід використати для виконання повторної вибірки. У нашому випадку використаємо `HAMMING`, так як він дозволяє створити більш чітке зображення, ніж із процедурою `BILINEAR`, та не має дислокацій на локальному рівні подання, як з використанням `BOX`. Функція нормалізації матиме такий вигляд

```
def normalize (character _in, width, height):
    normalized = character _in resize ((width, height), im. HAMMING)
    Normalized Array = np. array(normalized)
    return Normalized Array
```

Наступним етапом буде формування модуля вилучення рядків та символів. Для створення даного модуля використовуємо також бібліотеки `numpy` і `image`, і також необхідно виконати імпорт раніше створеного модуля попередньої обробки зображення. На початковому етапі відбувається

розпізнавання абзаців. Для пошуку та виділення нового абзацу використовуємо пробіли, кількість яких має бути більшою за визначену кількість пікселів.

```
def cropParagraphs(numberOfLines,topOfLines,bottonOfLines):
    lokationOfNewLines = []
    for i in range(1, numberOfLines):
        whiteSpaceDistance = topOfLines[i] – bottonOfLines[i-1]
        if whiteSpaceDistance > 60:
            locationOfNewLines.append(i-1)
    return locationOfNewLines
```

На наступному кроці відбувається розпізнавання рядків. Для цього виконується пошук перших і останніх чорних пікселів у рядку, за рахунок чого здійснюється створення списку, у який вносяться усі виявлені рядки текстового зображення (Додаток Б до магістерської роботи).

### 3.3 Модуль програми створення та навчання нейронної мережі

Для формування даного модуля слід спочатку підключити бібліотеки `numpy` та `tensorflow`, а також імпортувати попередньо створені модуль попередньої обробки і модуль виділення символів і рядків, які будуть необхідними для реалізації ряду функцій.

Навчання нейронної мережі покладається на функцію `MixerBlock`. Із використанням вкладених циклів для обчислення цільової помилки символів базового алфавіту, які потрібно налаштувати (тобто фактично навчити), та кількості навчальних зразків здійснюється процес попередньої обробки, обчислення похибки для усіх вихідних нейронів вихідного шару мережі, уточнення вагових коефіцієнтів нейронів і обчислення середньо квадратичної помилки.

Створюємо повноз'єднаний блок для моделі нейронної мережі із назвою `MixerBlock`, виконуємо ініціалізацію параметрів кількості вимірів для цього блоку.

```
class MlpBlock(nn.Module):
    mlp_dim: int
```

Формуємо функцію виклику екземпляра класу для повноз'єданого шару нейронної мережі, що об'єднує всі параметри в  $N \times N$  матрицю. Використовуємо нелінійну функцію активації Gaussian Error Linear Units. Для цього повноз'єднаний шар, що з'єднується в кількість каналів вхідного зображення, переводиться у нове подання. Наприклад, якщо в нас зображення (224, 224, 3), тобто звичайне RGB зображення, то пайплайн цього блоку буде мати вхід (224, 224, 3) та буде переведений у вигляд (224, 224, mlp\_dim) \* gelu.

У шарі нормалізації для уникнення проблеми, коли числа у вагах стають занадто великими і крок для градієнту стає надто нестабільний, міняємо осі вхідного зображення місцями.

```
@nn.compact
def __call__(self, x):
    y = nn.LayerNorm()(x)
    y = jnp.swapaxes(y, 1, 2)
```

Створюємо модуль MlpBlock для виконання операції змішування токенів (патчів, фрагментів зображення). Міняємо ось Y вхідного положення, додаємо інформацію, отриману з перемішування токенів, до основного зображення та виконуємо нормалізацію.

```
y = MlpBlock(self.tokens_mlp_dim, name='token_mixing')(y)
y = jnp.swapaxes(y, 1, 2)
x = x + y
y = nn.LayerNorm()(x)
```

Надалі додаємо інформацію, отриману з перемішування каналів, до основного зображення.

```
return x + MlpBlock(self.channels_mlp_dim, name='channel_mixing')(y)
```

Основний клас моделі `MlpMixer` має такі характеристики, як `patches` - набір фрагментів зображення, `num_classes` - кількість категорій для класифікації, `num_blocks` - кількість міксер-блоків, `hidden_dim` - кількість фільтрів для конволюційного шару.

Формуємо конволюційний шар для згортання і розділення зображення на фрагменти, виконуємо конкатенацію фрагментів зображення по осям висоти та ширини, здійснюємо ініціалізацію та проведення `num_blocks` операцій з вхідним зображенням, виконуємо нормалізацію та усереднення.

```
for _ in range(self.num_blocks):
    x = MixerBlock(self.tokens_mlp_dim, self.channels_mlp_dim)(x)
    x = nn.LayerNorm(name='pre_head_layer_norm')(x)
    x = jnp.mean(x, axis=1)
```

Формуємо вихідний результат у вигляді імовірнісного розподілу по класам. Перевіряємо функцію оновлення градієнту, що використовується для тренування нейронної мережі. Використовуємо функцію крос-ентропії для мультикласового розпізнавання

```
def cross_entropy_loss(*, logits, labels):
    logp = jax.nn.log_softmax(logits)
    return -jnp.mean(jnp.sum(logp * labels, axis=1))
```

Формуємо крос-ентропію з імовірнісними одиницями (`logits`), що є необробленим вектором імовірностей конкретного вхідного зображення. Виконуємо оновлення вагів відповідно до розрахованого градієнту, де `l` - конкретні значення, `g` - їх градієнт.

```
l, g = utils.accumulate_gradient(
    jax.value_and_grad(loss_fn), params, batch['image'], batch['label'],
    accum_steps)
```

Для отримання інформації з датасету для навчання моделі нейронної мережі вказуємо конфігурацію моделі, клас моделі та вказуємо кількість кроків тренування, що модель має пройти.

```
def train_and_evaluate(config: ml_collections.ConfigDict, workdir: str):
    dataset_info = input_pipeline.get_dataset_info(config.dataset, 'train')
    model_cls = {'Mixer': models.MlpMixer}[config.get('model_type', 'ViT')]
    model = model_cls(num_classes=dataset_info['num_classes'],
**config.model)
    total_steps = config.total_steps
```

Формуємо графік зміни швидкості тренування, чим вона менша тим більш прискіплива модель до змін. На початку тренування бажано мати якнайбільшу швидкість тренування, адже так модель зможе запам'ятати найбільш загальні патерни датасету, для покращення точності використовується менша швидкість тренування.

Виконуємо глобальну нормалізацію та ініціалізацію оптимізатора для знаходження глобального мінімуму цільової loss-функції. Здійснюємо ініціалізацію функції оновлення градієнтів. Нульовий крок визначає, що тренування проводиться з нуля.

```
update_fn_repl = make_update_fn(
    apply_fn=model.apply, accum_steps=config.accum_steps, tx=tx)
update_rng_repl = flax.jax_utils.replicate(jax.random.PRNGKey(0))
initial_step = 0
```

Здійснюємо операцію проведення тренування нейронної мережі, виконуємо розрахунок градієнтів та здійснюємо оновлення ваг нейронів відповідно до виконаних розрахунків.

```
for step, batch in zip(
    range(initial_step, total_steps + 1),
```

```

input_pipeline.prefetch(ds_train, config.prefetch)):
with jax.profiler.StepTraceAnnotation('train', step_num=step):
    params_repl, opt_state_repl, loss_repl, update_rng_repl = update_fn_repl(
        params_repl, opt_state_repl, batch, update_rng_repl)
return flax.jax_utils.unreplicate(params_repl)

```

Для виконання у подальшому оптичного розпізнавання символів здійснюємо ініціалізацію алфавіту. Для роботи функції по ініціалізації вагових коефіцієнтів і зміщення використовуємо функцію `numpy.random.random`, яка виконує повернення випадкового плаваючого значення у заданому діапазоні, який для нашого випадку знаходиться від -0.5 до 0.5. Обчислюємо ваги з вхідного до прихованого шару, та послідовно здійснюємо зміщення прихованого та вхідного шару мережі.

Для активізації нейронів прихованого і вхідного шару мережі задіємо функцію активізації, у ролі якої виступає логістична сигмоїдна функція.

```

def logActivation(sum):
    out = 1 / (1 + np.exp(-sum))
    return out

```

Обчислення значень вихідних сигналів для усіх нейронів мережі з вхідного до прихованого шару здійснюється із використанням вкладених циклів. Враховується кількість прихованих шарів і кількість входів мережі, потім обчислюється добуток вагових коефіцієнтів по відношенню до вхідних сигналів й до загальної суми усіх дендритів додається зміщення. Надалі, із використанням логістичної функції виконується обчислення вихідного сигналу кожного прихованого нейрона мережі та відбувається надсилання його для зберігання до масиву `outputOfHiddNeurons`. Для формування розповсюдження сигналу через мережу описуємо початкові змінні.

```

def feedForward(normalized_data, Wi_h, Wh_o, Bh, Bo):
    n_h = 0

```



Далі йде обробка послідовності розповсюдження сигналу з вхідного до прихованого шару.

Обчислені вихідні сигнали із прихованого шару перемножуємо на вагові коефіцієнти із прихованого до вихідного шару мережі. Знаходимо вхід для k-го вихідного нейрона мережі шляхом додавання зміщення, і за допомогою логістичної функції обчислюємо значення виходу для вихідних нейронів мережі та зберігаємо отримані дані у вихідному.

Реалізація процесу коригування вагових коефіцієнтів виконано із використанням функції із алгоритмом зворотного поширення. На першому етапі відбувається збереження значень попередніх вагових коефіцієнтів. Зворотній пошук здійснюється із вихідного до прихованого шару. Обчислення значень коригування знаходиться як добуток швидкості навчання з урахуванням помилки на поточному виході нейрона мережі, сигмоїдної похідної та виходом поточного прихованого нейрона мережі.

Наступним етапом процесу налаштування мережі є зворотній пошук із прихованого до вхідного шару мережі. Виконується обчислення похибки для кожного вихідного нейрону по відношенню до поточного прихованого нейрону мережі. Із використанням обчислення у циклі по усім вхідним ваговим коефіцієнтам, що відносяться до поточного прихованого нейрона мережі відбувається розрахунок загальної дельта-помилки відносно ваги, яку слід скоригувати. Ці обчислені вагові коефіцієнти з'єднують вхідний і поточний прихований шари мережі (див. Додаток В до магістерської роботи).

Для кожного виходу із вихідних нейронів мережі обчислюємо похибку від цільового виходу. Обчислення середньоквадратичної похибки здійснюється за таким підходом.

```
factualError = 0
```

```
for x in range(0,64):
```

```
    rootMeanSquare = 0.5 * outMistake[x]**2
```

```
    factualError = factualError + rootMeanSquare
```

Для виведення зображення розпізнаних символів використовуємо функцію recognizeChar. Розпізнаним символом текстового зображення є нейрон із найбільшим результатом.

Лістинг програми по навчанню нейронної мережі приведено у Додатку В магістерської роботи.

У даному розділі магістерської роботи було розроблену структурну схему програмного продукту для виконання операції розпізнавання друкованих символів текстового документу. Були визначені базові модулі програмного продукту та їх функціональне призначення. На основі сформованої послідовності розпізнавання текстової частини документа були виконані розробка модуля програми сегментації текстового документа для виділення символів для розпізнавання, був розроблений модуль програми по виконанню розпізнавання символів на основі згорткової нейронної мережі та виконано його налаштування. Розроблений програмний продукт може бути використаний для розпізнавання друкованих символів текстових документів.

## 4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ СТВОРЕНОЇ ПРОГРАМИ

Для оцінка ефективності функціонування створеної програми були використані різні типи документів із текстовим наповненням. Результати проведених досліджень по оцінюванню якості розпізнавання тексту приведені у цьому розділі магістерської роботи.

### 4.1 Тестування роботи програмного продукту

Для тестування роботи сформованого програмного продукту були відібрані різні типи документів із чорним та кольоровим поданням тексту, із різними типами шрифтів та розмірів символів у них. Були створені різні зображення з друкованими та невеликими фрагментами рукописних символів.

Перевірку працездатності програми виконаємо на різних типах документів із різними шрифтами. Один із таких фрагментів документів, що має різні відступи, поля, але однаковий тип шрифту із різними розмірами символів, приведений на рис. 4.1. Слова у цьому фрагменті документу, що були підготовлені для розпізнавання, поміщені у відповідний прямокутник та розділені проміжками одне від одного. Розпізнаний текст, який можна вивести у вигляді текстового документу із шрифтом типу Times New Roman, приведений на рис 4.2.



Рисунок 4.1 — Фрагмент документу для розпізнавання

```
'Опис\n'
'документів, що надаються юридичною особою\n'
'державному реєстратору\n'
'для проведення реєстраційної дії\n'
'“Державна реєстрація новоутвореної шляхом заснування юридичної\n'
'особи”\n'
```

Рисунок 4.2 — Фрагмент розпізнаної частину документа

Інший фрагмент документа, що має різні відступи, поля, різний тип шрифту із різними розмірами символів, що сформований у різних кольорах, приведений на рис. 4.3. Слова у цьому фрагменті документа, що були підготовлені для розпізнавання, як і у попередньому варіанті, також поміщені у відповідний прямокутник та розділені проміжками одне від одного. Розпізнаний фрагмент тексту, який після виконаних операцій по розпізнаванню можна вивести у вигляді текстового документа, приведений на рис 4.4.



Рисунок 4.3 — Фрагмент різноколірного документа для розпізнавання

Символ малого герба України, який не відноситься до групи символів текстових документів, не було розпізнано.

```
--Recognized text--

('НАЦІОНАЛЬНА РАДА УКРАЇНИ\n'
 'З ПИТАНЬ ТЕЛЕБАЧЕННЯ І РАДІОМОВЛЕННЯ\n'
 'Видана ТОВАРИСТВУ З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ\n'
 '«ЮНІВЕРС ТБ», м. Славутич Київської області\n'
 'Строк дії ліцензії\n'
 'Дата видачі ліцензії\n')
```

Рисунок 4.4 — Фрагмент розпізнаної частину документу

Інша частина тексту із значно меншим розміром шрифту приведена на рис. 4.5. Виведення даних розпізнаного тексту виконується як звичайний текст, виділяючи тільки літери великі та прописні. Фрагмент рукописного тексту, написання якого максимально наближене до друкованого варіанту, також був розпізнаний. Це отримано завдяки навчання нейронної мережі на значному наборі даних, що мають у тому числі варіанти написання рукописних символів у вигляді цифр. Тут також зображення малого герба України як частина документу не було розпізнано, так як він не відноситься до тих символів, що навчена система здатна розпізнавати.

Розпізнавання здійснюється із документу, поданого у форматі pdf-файлу. Документ попередньо розбивається на окремі сторінки. Потім у межах однієї сторінки відбувається виділення рядків та пошук у них слів. Розпізнавання символів здійснюється у межах слів, і розпізнані символи групуються у попередньо встановлені межі слів. Слова, що у початковому документі були розміщені у одному рядку, після розпізнавання групуються у той же один рядок із послідовності символів, що обмежуються таким чином розпізнана група слів \n'.



Рисунок 4.5 — Фрагмент документа із частиною рукописного тексту

Такий підхід дозволяє при виведенні на друк розпізнаного тексту початкового документа зберегти за необхідності його вихідну форму.

Розпізнаний текст приведеної частини текстового документа приведено на рис. 4.6. Приведений текст початкового документа дозволяє зробити висновок про те, що наведені документи були розпізнані вірно. Але таке якісне розпізнавання відбувалося не у всіх випадках. У деякій частині виконаних досліджень по розпізнаванню змішаних форм текстових документів 100% відсоткової точності розпізнавань не вдалось досягнути. Фрагмент таких документів приведені на рис. 4.7.

```
--Recognized text--
C:\n'
'НАЦІОНАЛЬНА КОМІСІЯ,\n'
'ЩО ЗДІЙСНЮЄ ДЕРЖАВНЕ РЕГУЛЮВАННЯ У СФЕРІ РИНКІВ\n'
'ФІНАНСОВИХ ПОСЛУГ\n'
'12.2017\n'
'РОЗПОРЯДЖЕННЯ\n'
'м. Київ\n'
'Про видачу ТОВАРИСТВУ З ОБМЕЖЕНОЮ\n'
'ВІДПОВІДАЛЬНІСТЮ\n'
'ліцензії Тра\n'
'провадження господарської діяльності з\n'
'надання фінансових послуг (крім професійної\n'
'діяльності на ринку цінних паперів)\n'
'>>\n'
'№₂\n'
'За результатами розгляду заяви про отримання ліцензії на провадження\n'
'господарської діяльності з надання фінансових послуг (крім професійної\n'
'діяльності на ринку цінних паперів) ТОВАРИСТВА З ОБМЕЖЕНОЮ\n'
'ВІДПОВІДАЛЬНІСТЮ\n'
```

Рисунок 4.6 — Результати розпізнавання

```
'Нашснальна\n'
'рада\n'
'укратни\n'
'з\n'
```

Рисунок 4.7 — Фрагмент результатів розпізнавання із помилками

#### 4.2 Перевірка якості роботи програми розпізнавання тексту

Для оцінювання якості роботи систем й алгоритмів розпізнавання тексту у документах можна використати такі категорії як точність й достовірність. Точність роботи системи розпізнавання текстових документів може бути охарактеризована як ймовірність вірного розпізнавання вибраного тексту у аналізованому об'єкті. Однак у загальному випадку визначити дійсне значення ймовірності розпізнавання тексту досить складно. У такому випадку значення точності розпізнавання тексту можна визначити як апостеріорна ймовірність вірного розпізнавання символів тексту для деякого визначеного набору початкових текстових даних. При такому підході до знаходження точності розпізнавання завдання ефективного розпізнавання тексту може бути визначено як завдання досягнення максимуму апостеріорної ймовірності правильного знаходження та розпізнавання символів із визначеного набору текстових даних.

Іншим ще одним показником оцінювання якості роботи системи розпізнавання тексту може бути достовірність або ж надійність розпізнавання. Оцінювання значення достовірності роботи системи розпізнавання тексту відображає її здатність апріорно визначати рівень точності власного отриманого результату роботи.

По відношенню до розпізнавання текстових документів, у яких певна кількість символів у словах може бути не розпізнана, можна скористатися таким підходом. У еталонному слові із кількістю  $s$  символів можуть бути помилкові символи. Для оцінки якості розпізнавання таких символів скористаємося підходом по визначенню коефіцієнта помилок  $K_s$  у символах, який обчислюється за таким виразом

$$K_s = \frac{b_s + e_s + d_s}{c}, \quad (4.1)$$

де  $b_s$  — кількість помилково вставлених символів;  
 $e_s$  — кількість помилково заміщених символів;  
 $d_s$  — кількість помилково видалених символів;  
 $c$  — кількість усіх символів у еталонному поданні [34].

Із аналогічним підходом визначається коефіцієнт помилок  $K_{sl}$  по відношенню до слів у еталонному зразку, який можна обрахувати таким виразом (4.2)

$$K_{sl} = \frac{b_w + e_w + d_w}{w}, \quad (4.2)$$

де  $b_w$ ,  $e_w$  й  $d_w$  є відповідно кількість помилково при обробці текстового документа вставлених, заміщених й видалених слів;

$w$  — загальна кількість слів у еталонному зразкові документа [34].

Для знаходження точності розпізнавання символів  $T_s$  у початковому документі можна скористатися виразом

$$T_s = 1 - K_s. \quad (4.3)$$

Можна також коефіцієнт помилок у обробленому документі обчислити через значення мінімальної кількості операцій по відношенню до символів, які необхідно виконати, щоб отриманий набір символів після ряду дій привести до еталонного набору. Такий варіант визначення помилок має назву відстані по Левенштейну. Але якщо у отриманій після розпізнавання послідовності символів для приведення її стану до еталонного значення документа необхідно ще й виконати перестановку деяких із символів, то у такому випадку сумарна кількість усіх виконаних операцій має назву відстані по Дамерау-



Левенштейну [34]. Для нашого варіанта визначення точності розпізнавання символів скористаємося виразом (4.3), що дозволяє обчислити точність розпізнавання символів у обробленому документі.

Для перевірки якості створеного програмного продукту була сформована навчальна вибірка, де були використані шрифти Times New Roman, Verdana й Arial. До складу вибірки були включені літери латинського й кириличного алфавітів, розділові знаки й арабські цифри. Всього для перевірки роботи було використано у різному представленні 460 символів.

Для розширення послідовності контролю були також задіяні інші типи шрифтів, такі як Lucida Fax, Calibri, Candara та Gadugi. У цьому випадку усього додатково було задіяно 360 символів.

У результаті розпізнавання першої групи символів було розпізнано 437 (95,0%) символів з 460 використаних. У цій групі не були розпізнані всього 23 сформованих символів, до яких потім було використано шаблонний метод розпізнавання символів. Частина символів із цього набору було визначено невірно із-за похибка визначення розміру символу по відношенню до якого регістру слід віднести даний символ. Цей варіант розпізнавання можна пояснити здійсненням нормалізації розміру символів шляхом приведення їх до стандартного розміру, у результаті якої деякі символи стають подібними. Грунтовний аналіз таких неправильно розпізнаних символів указує на те, що більшість із нерозпізнаних символів припадає на шрифт типу Impact. Це можна пояснити тим фактором, що друковані символи цього типу шрифту значно відрізняються по своїй формі подання від символів інших задіяних шрифтів, які були використані для навчання згорткової нейронної мережі.

У випадку, коли із процесу розпізнавання були вилучені символи шрифту типу Calibri, удалось розпізнати більшу кількість заданих символів. У результаті розпізнавання сформованих додатково 360 образів символів некоректно було розпізнано всього 12 символів (3,3%). При виконанні розпізнавання одного й того ж тексту, що був зображений трьома різними

шрифтами задіяними у навчальній виборці, були отримані такі результати розпізнавання.

Найбільшу кількість невірних розпізнавань символів показав текст, що був сформований шрифтом типу Gadugi, найменшу кількість невірних розпізнавань показав текст, сформований шрифтом типу Times New Roman. Це дозволяє сформулювати висновок, що при розпізнаванні тексту, набраного тим шрифтом, що був введений до навчальної вибірки, буде показувати найкращий результат. Інші типи шрифтів будуть також розпізнані, якщо накреслення їх символів буде у незначній мірі відрізнятися від накреслення шрифтів.

Для визначення якості розпізнавання символів із дефектами типу розрив скористаємося підготовленим набором таких символів. Значення спотворення символу будемо визначати як відношення наявної кількості точок до повної кількості усіх точок, що представляє даний символ. Для кожного із символів кількість спотворень буде різною та була визначена у відсотках із застосуванням для обчислення виразів (4.1) й (4.3) по відношенню до вірного подання символів алфавіту. Усереднені результати виконання такого експерименту по виявленню точності розпізнавання кирилических символів показали, що після 25% спотворень розпізнавання символів тексту не дає бажаних результатів

У результаті проведених досліджень можна зробити висновок, що при наявності спотворень у символах більше 25%, точність розпізнавання символів різко знижується, і стає практично неможливим визначення конкретного типу символу. Підвищення точності розпізнавання можна буде збільшити тільки шляхом введення додаткового розпізнавання із використанням словників.

Проведені експериментальні дослідження по тестуванню та розпізнаванню спеціально сформованих символів дозволяють зробити заключення, що розроблений програмний засіб по розпізнаванню друкованих символів, у тому числі із деякими дефектами у вигляді розривів у поданні символів, може використовуватися у системах переведення сфотографованих та поданих у pdf-форматі текстових документів у електронну форму.

## **5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ЛЮДЕЙ У ВІДЕОПОТОЦІ**

Метою економічної частини магістерської кваліфікаційної роботи є довести економічну доцільність та ефективність впровадження наукової розробки, 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.

### **5.1 Комерційний та технологічний аудит науково-технічної розробки**

Метою даного розділу є проведення технологічного аудиту, в даному випадку програми оброблення зображення для виділення символів та їх розпізнавання у текстах. Метою роботи є вдосконалення методів виділення та розпізнавання друкованих символів текстових документів із використанням багаторівневого аналізу документа. Особливістю розробки є використання удосконаленого методу оброблення отриманого зображення текстового документу для виділення та розпізнавання символів тексту, який відрізняється від відомих підходів виконанням виділення характерних базових ліній символів та їх порівняння із еталонними зображеннями, що дозволяє більш ефективно розпізнавати символи у отриманому тексті.

Аналогом може бути система оптичного розпізнавання символів ABBYY FineReader PDF Standard за ціною 3229 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 5.1 — Критерії оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри тері й	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
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 аналогі	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
Ринкові перспективи					
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	Конкурентів немає
Практична здійсненність					

Таблиця 5.1 — Закінчення

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

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	3	4	3
Експлуатаційні витрати	4	4	3
Ринок збуту	3	4	4
Конкурентоспроможність	4	3	3
Фахівці з технічної і комерційної реалізації	3	4	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	3
Супровідна документація	2	3	4
Сума	41	43	42
Середньоарифметична сума балів	$(43+41+42) / 3 = 42$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що використовується удосконалений метод оброблення отриманого зображення

текстового документу для виділення та розпізнавання символів тексту, який відрізняється від відомих підходів виконанням виділення характерних базових ліній символів та їх порівняння із еталонними зображеннями, що дозволяє більш ефективно розпізнавати символи у отриманому тексті.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де  $M$  — місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_p$  — число робочих днів за місяць, 20 днів;

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

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	32500	1547,62	48	74285,714
Програміст	29500	1404,76	48	67428,571
Всього				141714,29

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 12 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 12 \% / 100 \% \quad (5.2)$$

$$Z_d = (141714,29 \cdot 12 \% / 100 \% ) = 17005,71 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (141714,29 + 17005,71) \cdot 22 \% / 100 \% = 34918,40 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_в} \cdot \frac{t_{виск}}{12} \text{ [грн.]} \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;



T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$  — термін використання під час розробки, місяців Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 29499 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 2,29 міс.

$$A_{\text{обл}} = \frac{29499}{2} \times \frac{2,29}{12} = 2809,492 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю,  $V_{\text{нем.ак.}} = 23353,2$  грн.

Таблиця 5.5 — Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	29499	2	2,29	2809,429
Офісне обладнання (меблі)	20499	4	2,29	976,143
Приміщення	900000	20	2,29	8571,429
Ліцензійна ОС, та спеціалізовані ліцензійні нематеріальні ресурси (Microsoft Visual Studio Professional 2022 - 23353,2 грн)	23353,2	2	2,29	2224,114
Всього				14581,11

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де  $V$  — вартість 1 кВт-години електроенергії для 1 класу підприємства,  $V = 6,2$  грн./кВт;

$\Pi$  — встановлена потужність обладнання, кВт.  $\Pi = 0,4$  кВт;

$\Phi$  — фактична кількість годин роботи обладнання, годин.

$K_{\Pi}$  — коефіцієнт використання потужності,  $K_{\Pi} = 0,9$ .

$$V_e = 0,9 \cdot 0,3 \cdot 8 \cdot 48 \cdot 6,2 = 642,816 \text{ (грн.)}$$

#### 5.2.6 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.6)$$

де  $H_{ib}$  — норма нарахування за статтею «Інші витрати».

$$I_b = 141714,29 \cdot 50\% / 100\% = 70857,14 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (5.7)$$

де  $H_{\text{нзв}}$  — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нзв}} = 141714,29 * 110 \% / 100 \% = 155886 \text{ (грн.)}$$

#### 5.2.8 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$V_{\text{заг}} = 141714,29 + 17005,71 + 34918,40 + 14581,11 + 23353,2 + 642,82 + 70857,14 + \\ + 155886 = 458958,39 \text{ грн.}$$

5.2.9 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \quad (\text{грн}), \quad (5.8)$$

де  $\eta$  — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то  $\eta=0,1$ ; технічного проектування, то  $\eta=0,2$ ; розробки конструкторської документації, то  $\eta=0,3$ ; розробки технологій, то  $\eta=0,4$ ; розробки дослідного зразка, то  $\eta=0,5$ ; розробки промислового зразка, то  $\eta=0,7$ ; впровадження, то  $\eta=0,9$ . Оберемо  $\eta = 0,5$ , так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 458958,39 / 0,5 = 917917 \text{ грн.}$$

### 5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

— а вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

— зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

— кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

— визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.9)$$

де  $\pm\Delta\Pi_0$  — розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

$N$  — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$\Pi_0$  — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,  $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$ ;

$\Pi_6$  — вартість програмного продукту у році до впровадження результатів розробки;

$\Delta N$  — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

$\lambda$  — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ .

$r$  — коефіцієнт, який враховує рентабельність продукту;

$\vartheta$  — ставка податку на прибуток, у 2023 році  $\vartheta = 18\%$ .

Припустимо, що при прогнозованій ціні 1200 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 100 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 13000 шт., протягом другого року – на 15000 шт., протягом третього року на 18000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*100 + (1200 + 100) * 13000) * 0,8333 * 0,38 * (1 - 0,18) = 4050799,838$$

грн.

$$\Delta\Pi_2 = (0*100 + (1200 + 100) * (13000 + 15000)) * 0,8333 * 0,38 * (1 - 0,18) =$$

9451866,289 грн.

$$\Delta\Pi_3 = (0*100 + (1200 + 100) * (13000 + 15000 + 18000)) * 0,8333 * 0,38 * (1 - 0,18) =$$

15528066,046 грн.

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 29030732,17 грн.

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

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.10)$$

де  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

$T$  — період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;

$t$  — період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$ПП = (4050799,838/(1+0,1)^1) + (9451866,289/(1+0,1)^2) + (15528066,046/(1+0,1)^3) = 3682545,31 + 7811459,743 + 11666465,85 = 23160470,9 \text{ грн.}$$

Далі розраховують величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} * ЗВ, \quad (5.11)$$

де  $k_{\text{інв}}$  — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{\text{інв}}=2\dots 5$ , але може бути і більшим;

ЗВ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 917917 = 1835833,55 \text{ грн.}$$

Тоді абсолютний економічний ефект  $E_{\text{абс}}$  або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (5.12)$$

$$E_{\text{абс}} = 23160470,9 - 1835833,55 = 21324637,35 \text{ грн.}$$

Оскільки  $E_{\text{абс}} > 0$  то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

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

$$E_{\text{г}} = T_{\text{жк}} \sqrt[1 + \frac{E_{\text{абс}}}{PV}] - 1, \quad (5.13)$$



де  $T_{ж}$  — життєвий цикл наукової розробки, роки.

$$\sqrt{E_b} = 3 (1 + 21324637,35/1835833,55 - 1) = 1,328$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де  $d$  — середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = (0,09...0,14)$ ;

$f$  — показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,5)$ .

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як  $E_b > \tau_{\min}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_b}, \quad (5.15)$$

$$T_{ок} = 1 / 1,328 = 0,75 \text{ р.}$$

Оскільки  $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,75 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких

складає 917917 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентно спроможним. Період окупності складе близько 0,75 роки.

## ВИСНОВКИ

Переведення текстових документів у електронну форму за допомогою оптичних систем розпізнавання широко використовується у різних сферах діяльності людей, таких як відділення банків, офіси фірм, у друкарській справі. Але при розпізнаванні у таких документах друкованих символів із дефектами значно збільшується відсоток помилок розпізнавання. Вирішення деяких аспектів цієї задачі із використанням нейронної мережі розглядається у даній магістерській роботі.

У першому розділі магістерської роботи був здійснений огляд відомих методів й систем, що використовуються для оптичного розпізнавання друкованих символів текстових документів, здійснено порівняльний аналіз найбільш поширених засобів розпізнавання текстових документів, перелічено їх основні недоліки й обґрунтовано необхідність створення нового програмного продукту для розпізнавання друкованих символів текстових документів. Здійснений аналіз розглянутих методів й засобів розпізнавання друкованих символів текстових документів дозволив виявити деякі проблеми, що існують у цій області та уточнити сфери примінення розглянутих методів, виявити їх переваги й недоліки у використанні до вирішення завдань по виявленню і розпізнаванню друкованих символів, у тому числі із деякими спотвореннями.

У другому розділі магістерської роботи сформована послідовність поетапного розпізнавання друкованих символів текстових документів. Запропоновано процес розпізнавання розділити на ряд етапів, де послідовно виконується попередня обробка текстового документу, поділ його на абзаци, рядки, слова і символи, використання нейронної мережі для розпізнавання, а для тих символів, які із певних причин не були визначені на попередньому етапі, застосувати додатковий підхід до розпізнавання із аналізом контуру символу й фону, на якому він розміщений.

У третьому розділі роботи була розроблена структура програмного продукту для розпізнавання друкованих символів текстових документів,

розроблена програма розпізнавання символів та виконано процес навчання нейронної мережі на сформованому тестовому наборі символів.

У четвертому розділі роботи приведені результати тестування створеного програмного продукту та експериментальної перевірки якості роботи запропонованого підходу.

У п'ятому розділі роботи були виконані розрахунки по обґрунтуванню доцільності здійснення нового наукового вирішення сформованої задачі по розпізнаванню друкованих символів, виконані розрахунки необхідних економічних витрат для створення запропонованих засобів виділення й розпізнавання друкованих символів і визначено комерційний ефект від впровадження запропонованого програмного рішення в оптичних системах для розпізнавання друкованих символів текстових документів.

Розроблений у магістерській роботі програмний продукт може бути використаний у оптичних засобах розпізнавання друкованих символів текстових документів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Chaudhuri A. Optical character recognition systems for different languages with soft computing. / A. Chaudhuri, K. Mandaviya, P. Badelia, S. K. Ghosh // Springer, 2017. - 260 p. [https://doi.org/10.1007/978-3-319-50252-6\\_1](https://doi.org/10.1007/978-3-319-50252-6_1)
2. Оптичне розпізнавання символів (OCR) [Електронний ресурс] – Режим доступу: <https://hashdork.com/uk/optical-character-recognition/>.
3. FineReader [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/ABBYY\\_FineReader](https://uk.wikipedia.org/wiki/ABBYY_FineReader)
4. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
5. Тимченко О. В. Нейромережеві методи розпізнавання зображень текстів /О. В. Тимченко, Б. М. Гавриш, Б. В. Дурняк // Поліграфія і видавнича справа. 2021, № 1 (81) – С.72-88.
6. 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.
7. Дідич О. Л. Розпізнавання символів текстових документів із використанням нейронної мережі О. Д. Дідич, Т. Б. Мартинюк, М. А. Очуров // Матеріали ІІІ науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ-2024)». — Електрон. текст. дані. — Вінниця, 2024. — Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2024/paper/view/19561>.
8. Згорткова нейронна мережа. [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Згорткова\\_нейронна\\_мережа](https://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа).
9. Shi B. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition / B. Shi, X. Bai, C. Yao // IEEE transactions on pattern analysis and machine intelligence. 2017, vol. 39, №. 11, pp. 2298-2304.

10. Busta M. Deep text spotter: an end-to-end trainable scene text localization and recognition framework / M. Busta, L. Neumann, J. Matas // Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2204-2212.
11. Ye Q. Text detection and recognition in imagery: A survey / Q. Ye, D. Doermann // IEEE transactions on pattern analysis and machine intelligence. 2015, vol. 37, № 7, pp. 1480-1500.
12. Uchida S. Statistical Deformation Model for Handwritten Character Recognition // Advances in Digital Document Processing and Retrieval. 2014, pp. 157-174.
13. Синєглазов В. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення / В. Синєглазов, О. Чумаченко [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <http://itcm.comp-sc.if.ua/2017/Sineglazov.pdf/>
14. Шаховська Н. Б. Розроблення архітектури системи оптичного розпізнавання символів з фотографій документів. / Н. Б. Шаховська, А. Шебеко // Вісник Хмельницького національного університету, №3, 2022 (309) с. 50-54.
15. Кушнір Н.О. Використання згорткових нейронних мереж у задачах розпізнавання та класифікації об'єктів зображень / Н.О. Кушнір, Т.М. Локтікова, А.В. Морозов, В.О. Юрченко // Технічна інженерія №1 (89), 2022. — С. 93 —100.
16. Тимченко О. В. Алгоритми та функції інформаційної системи розпізнавання символів на основі методів поліпшення зображень / О. В. Тимченко, І. О. Кульчицька, О. О. Тимченко // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. – Вип.69. – К.: 2013. – С.167-173.
17. Zahra Elhamraoui. Introduction to convolutional neural network. <https://medium.com/analytics-vidhya/introduction-to-convolutional-neural-network-6942c189a723>, May 28, 2020.

18. Huang Gao. Densely Connected Convolutional Networks / Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger // NASA ADS. — 2016. — Pp. 21–29.
19. Що таке ABBYY FineReader [Електронний ресурс] – Режим доступу: <https://help.abbyy.com/uk-ua/finereader/12/overview>.
20. OCR CuneiForm для Windows [Електронний ресурс] – Режим доступу: <http://softobase.com/ru/ocr-cuneiform>.
21. Програма Readiris Pro [Електронний ресурс] – Режим доступу: <https://readiris.ru.uptodown.com/windows>
22. Redmon Joseph. You Only Look Once: Unified, Real-Time Object Detection / Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi // The Computer Vision Foundation. — 2016. — Pp. 21–30.
23. LeCun Y. LeNet-5, convolutional neural networks. [Електронний ресурс] – Режим доступу: URL: <http://yann.lecun.com/exdb/lenet/>.
24. Comparison of optical character recognition software. [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Comparison\\_of\\_optical\\_character\\_recognition\\_software](https://en.wikipedia.org/wiki/Comparison_of_optical_character_recognition_software).
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. InftyReader. Офіціальний сайт. [Електронний ресурс] – Режим доступу: URL: <http://www.inftyreader.org/>.
27. Bolotova Yu. A. A review of algorithms for text detection in images and videos. / Yu. A. Bolotova, V. G. Spitsyn, P. M. Osina // Computer Optics 2017; 41(3): p. 441-452.
28. Larsson, Gustav. FractalNet: Ultra-Deep Neural Networks without Residuals / Gustav Larsson, Michael Maire, Gregory Shakhnarovich // NASA ADS. — 2016. — Pp. 31–41.
29. Зінченко О. В. Згорткові нейронні мережі для вирішення задач комп'ютерного зору. / О. В.Зінченко, О. С. Звенігородський, Т. М. Кисіль// Телекомунікаційні та інформаційні технології. 2022. № 2(75). С. 4-12.

30. Ковальчук А. М. Застосування згорткової нейронної мережі для розпізнавання рукописних символів / А. М. Ковальчук, Г. В. Марчук, Д. К. Марчук // Зб. наук. пр. «Вчені записки ТНУ імені В.І. Вернадського». Серія: технічні науки, Том 30 (69) Ч. 1 № 4, 2019, с. 68-73.
31. Keras. Офіційна документація. [Електронний ресурс] – Режим доступу: URL: <https://keras.io/>.
32. OpenCV library [Електронний ресурс] – Режим доступу: <https://opencv.org/>.
33. AForge.NET: Framework [Електронний ресурс]. – Режим доступу: <http://www.aforgenet.com/framework/>.
34. The EMNIST dataset. [Електронний ресурс]. – Режим доступу: <https://www.nist.gov/itl/iad/image-group/emnist-dataset>.
35. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
36. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.
37. Тарифи на електроенергію [Електронний ресурс]. - Режим доступу: <http://index.minfin.com.ua/tarif/electric.php>.



**ДОДАТОК А**

Міністерство освіти та науки України  
Вінницький національний технічний університет  
Інститут інформаційних технологій та комп'ютерної інженерії

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ОТ  
д.т.н., проф. Азаров О.Д.

" "

2023 р.

**ТЕХНІЧНЕ ЗАВДАННЯ**

на виконання магістерської кваліфікаційної роботи

Технологія оптичного розпізнавання друкованих символів на основі нейронної  
мережі

08-54.МКР.027.00.000.ТЗ

Науковий керівник: д.т.н., професор

\_\_\_\_\_ Мартинюк Т. Б.

Студент групи 2КІ-22м

\_\_\_\_\_ Дідич О. Д.

## 1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність дослідження пов'язана із широким застосуванням засобів по розпізнаванню символів друкованих текстів, створення кожного нового програмного засобу в цій області залишається творчим завданням й вимагає проведення додаткових досліджень через наявність специфічних вимог по вирішенню задач швидкодії, надійності розпізнавання символів та обсягів пам'яті, якими визначається кожне конкретне завдання по розпізнаванню символів друкованих текстів.

1.2 Наказ про затвердження теми МКР.

## 2 Мета МКР і призначення розробки

2.1 Мета роботи — аналіз сучасного стану і тенденцій розвитку зарубіжних і вітчизняних розробок у сфері оптичного розпізнавання тексту і вдосконалення технології розпізнавання друкованих символів текстових документів.

2.2 Призначення розробки — створення програмного продукту для виконання процесу розпізнавання друкованих символів текстових документів.

## 3 Вихідні дані для виконання МКР

3.1 Роздільна здатність зображення тексту не менше 300 dpi, модель кольорів для представлення зображення — RGB, кількість градацій яскравості зображення — 256.

3.2 Середовище розробки ПЗ Visual Studio Code.

## 4 Вимоги до виконання МКР

4.1 Провести аналіз різних типів технологій виявлення і розпізнавання оптичного розпізнавання тексту.

4.2 Вивчити принцип роботи та особливості систем розпізнавання текстових документів.

4.3 Запропонувати послідовність розпізнавання текстових документів.

4.4 Розробити технологію виділення і розпізнавання символів текстових документів.

## 5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз методів та засобів розпізнавання текстових документів	26.09.2023р.	30.09.2023р.	Розділ 1
2	Розробка нейромережевої технології розпізнавання текстових документів	01.10.2023р.	04.10.2023р.	Розділ 2
3	Розробка програми виділення і розпізнавання текстових документів	05.10.2023р.	18.10.2023р.	Розділ 3
4	Перевірка працездатності програмного продукту	19.10.2023р.	01.11.2023р.	Розділ 4
5	Розрахунок економічної доцільності створення програми виділення і розпізнавання текстових документів	02.11.2023р.	15.11.2023р.	Розділ 5
6	Апробація та впровадження результатів дослідження	16.11.2023р.	10.12.2023р.	Тези доповіді
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	19.11.2023р.	25.11.2023р.	ПЗ, графічний матеріал
8	Підготовка і підпис супроводжуючих документів, нормоконтроль та тест на плагіат	26.11.2023р.	08.12.2023р.	Оформлені документи

## 6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних

екзаменів, анотації до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

## 7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

## 8 Вимоги до оформлення МКР

### 8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія»;

— документи на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

## ДОДАТОК Б

## Лістинг програми сегментації тексту

```

import io
from convert import convert_pdf
from preprocess import create_pdf_dummy
from preprocess import compose_pdf
import model as m
model = m.MlpMixer()
def ocr_pages(file):
    if not is_this_pdf(file):
        file = create_pdf_dummy(file)
        file = compose_pdf(file)
    # пустий масив з майбутніми результатами
    ocr_res = []
    pages_as_images = convert_pdf(file)
    # проходимо по кожній сторінці
    for index, page in enumerate(pages_as_images):
        # визначаємо ширину і висоту сторінки
        width, height = page[0].size
        img_byte_arr = io.BytesIO()
        page[0].save(img_byte_arr, "jpeg")
        img_byte_arr = img_byte_arr.getvalue()
        # отримуємо результат з моделі відповідно до зображення
        # OcrData - увесь текст знайдений на фото
        # X, Y - ширина і висота відповідно
        # TextBlocks - рамки знайденого тексту
        data = model(img_byte_arr)
        if data:
            ocr_res.append({
                "OcrData": data,
                "Y": height,
                "X": width,
                "TextBlocks": page[1]
            })
    return ocr_res
def is_this_pdf(file):
    for byte_ind in range(min(len(file), 1024)-3):
        if file[byte_ind]==0x25 and file[byte_ind+1]==0x50 and
file[byte_ind+2]==0x44 and file[byte_ind+3]==0x46:
            return True
    return False
    def convert ToBW (image In):
        blackAndWhite = image In. Convert ('1')

```

```

    blackAndWhite = np.array (blackAndWhite)*1
    return blackAndWhite
def toggleOnesAndZeros (blackAndWhite)
    return (blackAndWhite^1)
def crop (blackAndWhiteToggled):
[numbner OfRowPixels, numbner OfColumnPixels] = blackAndWhite Toggled.shape
vertikalSumOfBlackPixels = np.Sum (blackAndWhiteToggled, axis = 0)
leftDetected = False
for i in range (0, numberOfColumnPixels):
    if vertical SumOfBlackPixels [i] > 0 and leftDetected == False:
        leftDetected == True
        left = i
    elif vertical SumOfBlackPixels [i] > 0 and leftDetected == True
        right = i
horizontalSumOfBlackPixels = np.Sum (blackAndWhiteToggled, axis = 1)
topDetected = False
for i in range (0, numberRowPixels):
    if horizontalSumOfBlackPixels [i] > 0 and topDetected == False:
        topDetected == True
        top = i
    elif horizontalSumOfBlackPixels [i] > 0 and topDetected == True:
        botton = i
        # нормалізація
def normalize (character _in, width, height):
    normalized = character _in resize ((width, height), im. HAMMING)
    Normalized Array = np. array(normalized)
    return Normalized Array
def cropParagraphs(numberOfLines,topOfLines,bottonOfLines):
    lokationOfNewLines = []
    for i in range(1, numberOfLines):
        whiteSpaceDistance = topOfLines[i] – bottonOfLines[i-1]
        if whiteSpaceDistance > 60:
            locationOfNewLines.append(i-1)
    return locationOfNewLines

```

## ДОДАТОК В

## Лістинг програми роботи нейронної мережі

```

import functools
import os
import time
from absl import logging
from clu import metric_writers
from clu import periodic_actions
import flax
from flax.training import checkpoints as flax_checkpoints
import jax
import jax.numpy as jnp
import ml_collections
import numpy as np
import optax
import tensorflow as tf

from vit_jax import checkpoint
from vit_jax import input_pipeline
from vit_jax import models
from vit_jax import utils
class MlpBlock(nn.Module):
    mlp_dim: int
    @nn.compact
    def __call__(self, x):
        y = nn.Dense(self.mlp_dim)(x)
        # застосування нелінійної функції активації Gaussian Error Linear Units
        y = nn.gelu(y)
        return nn.Dense(x.shape[-1])(y)
class MixerBlock(nn.Module):
    tokens_mlp_dim: int
    channels_mlp_dim: int
    @nn.compact
    def __call__(self, x):
        y = nn.LayerNorm()(x)
        y = jnp.swapaxes(y, 1, 2)
        y = MlpBlock(self.tokens_mlp_dim, name='token_mixing')(y)
        y = jnp.swapaxes(y, 1, 2)
        x = x + y
        # Нормалізація
        y = nn.LayerNorm()(x)
        return x + MlpBlock(self.channels_mlp_dim, name='channel_mixing')(y)
class MlpMixer(nn.Module):

```

```

patches: Any
num_classes: int
num_blocks: int
hidden_dim: int
tokens_mlp_dim: int
channels_mlp_dim: int
model_name: Optional[str] = None
@nn.compact
def __call__(self, inputs, *, train):
    # Конволюційний шар для згортання і розділення зображення на
фрагменти
    x = nn.Conv(self.hidden_dim, self.patches.size,
                strides=self.patches.size, name='stem')(inputs)
    x = einops.rearrange(x, 'n h w c -> n (h w) c')
    # ініціалізація та проведення num_blocks операцій з вхідним зображенням
    for _ in range(self.num_blocks):
        x = MixerBlock(self.tokens_mlp_dim, self.channels_mlp_dim)(x)
    # Нормалізація
    x = nn.LayerNorm(name='pre_head_layer_norm')(x)
    # Усереднення
    x = jnp.mean(x, axis=1)
    # Вихідний результат у вигляді імовірнісного розподілу по класам
    if self.num_classes:
        x = nn.Dense(self.num_classes, kernel_init=nn.initializers.zeros,
                    name='head')(x)
    return x
def make_update_fn(*, apply_fn, accum_steps, tx):
    # функція оновлення градієнту, що використовується для тренування
    def update_fn(params, opt_state, batch, rng):
        _, new_rng = jax.random.split(rng)
        dropout_rng = jax.random.fold_in(rng, jax.lax.axis_index('batch'))
        # крос-ентропія для мультикласового розпізнавання
        def cross_entropy_loss(*, logits, labels):
            logp = jax.nn.log_softmax(logits)
            return -jnp.mean(jnp.sum(logp * labels, axis=1))
        def loss_fn(params, images, labels):
            logits = apply_fn(
                dict(params=params),
                rngs=dict(dropout=dropout_rng),
                inputs=images,
                train=True)
            return cross_entropy_loss(logits=logits, labels=labels)
        # оновлення вагів відповідно до розрахованого градієнту
        l, g = utils.accumulate_gradient(
            jax.value_and_grad(loss_fn), params, batch['image'], batch['label'],

```



```

    accum_steps)
    g = jax.tree_map(lambda x: jax.lax.pmean(x, axis_name='batch'), g)
    updates, opt_state = tx.update(g, opt_state)
    params = optax.apply_updates(params, updates)
    l = jax.lax.pmean(l, axis_name='batch')
    return params, opt_state, l, new_rng
return jax.pmap(update_fn, axis_name='batch', donate_argnums=(0,))
def train_and_evaluate(config: ml_collections.ConfigDict, workdir: str):
    # отримання інформації з датасету
    dataset_info = input_pipeline.get_dataset_info(config.dataset, 'train')
    model_cls = {'Mixer': models.MlpMixer}[config.get('model_type', 'ViT')]
    model = model_cls(num_classes=dataset_info['num_classes'], **config.model)
    total_steps = config.total_steps
    lr_fn = utils.create_learning_rate_schedule(total_steps, config.base_lr,
                                                config.decay_type,
                                                config.warmup_steps)
    # глобальна нормалізація для знаходження глобального мінімуму цільової loss-
функції
    tx = optax.chain(
        optax.clip_by_global_norm(config.grad_norm_clip),
        optax.sgd(
            learning_rate=lr_fn,
            momentum=0.9,
            accumulator_dtype='bfloat16',
        ),
    )
    # ініціалізація функції оновлення градієнтів
    update_fn_repl = make_update_fn(
        apply_fn=model.apply, accum_steps=config.accum_steps, tx=tx)
    update_rng_repl = flax.jax_utils.replicate(jax.random.PRNGKey(0))
    initial_step = 0
    # проведення тренування, розрахунків градієнтів, та оновлення ваг відповідно
до розрахунків
    for step, batch in zip(
        range(initial_step, total_steps + 1),
        input_pipeline.prefetch(ds_train, config.prefetch)):
        with jax.profiler.StepTraceAnnotation('train', step_num=step):
            params_repl, opt_state_repl, loss_repl, update_rng_repl = update_fn_repl(
                params_repl, opt_state_repl, batch, update_rng_repl)
    return flax.jax_utils.unreplicate(params_repl)

```

**ДОДАТОК Г**  
Послідовність розпізнавання символів

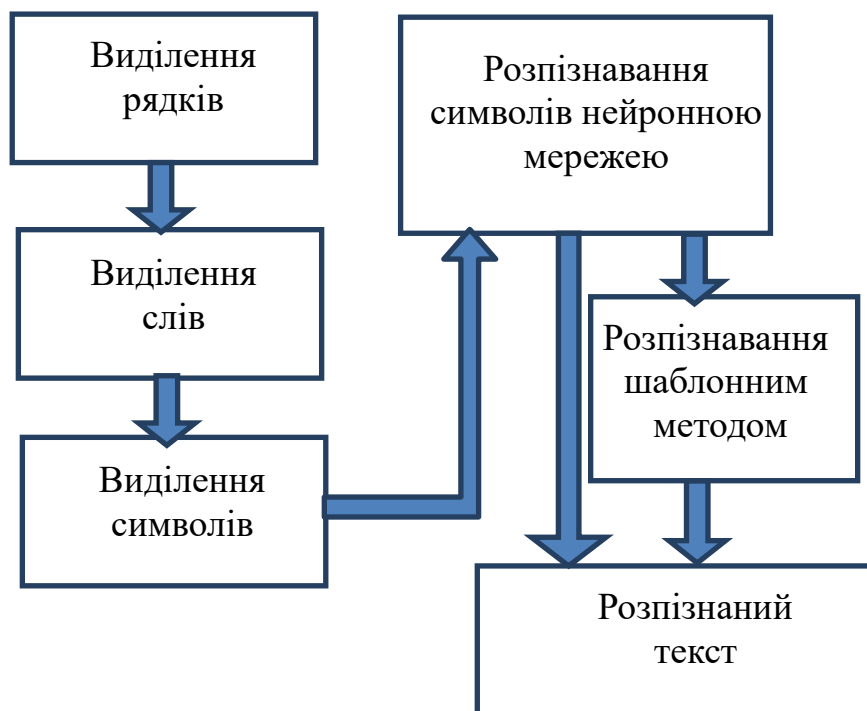


Рисунок Г.1 — Послідовність розпізнавання символів

ДОДАТОК Д  
Методи розпізнавання тексту



Рисунок Д.1 — Методи розпізнавання тексту

## ДОДАТОК Е

### Системи оптичного розпізнавання тексту

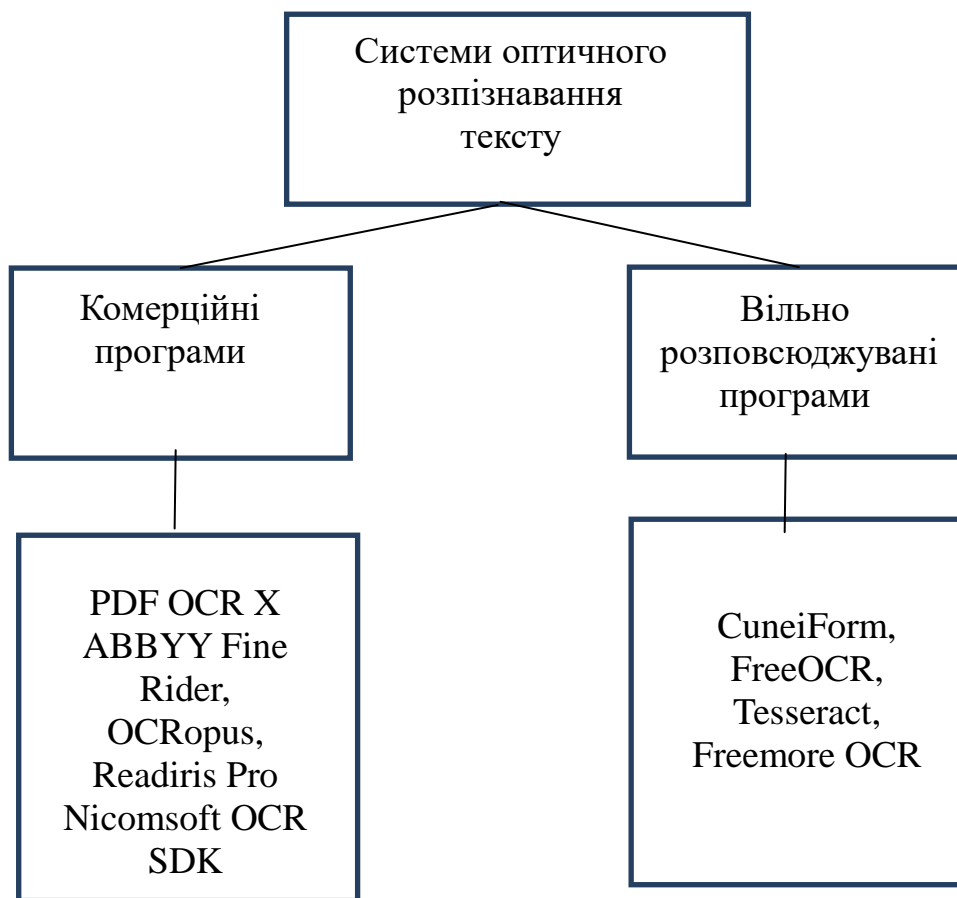


Рисунок Е.1 — Системи оптичного розпізнавання тексту

## ДОДАТОК Ж

## Структура програми розпізнавання символів

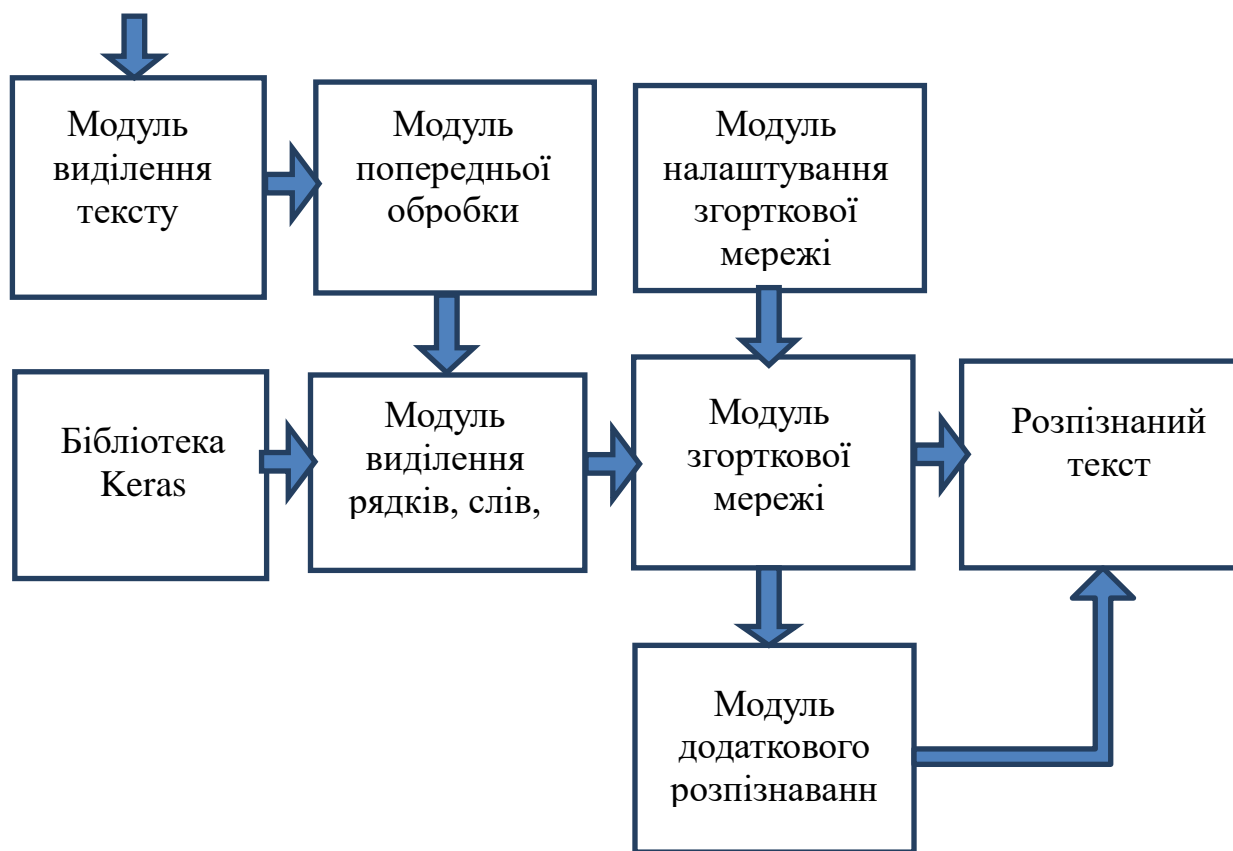


Рисунок Ж.1 — Структура програми розпізнавання символів

**ДОДАТОК И**

**ПРОТОКОЛ**

**ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ  
ЗАПОЗИЧЕНЬ**

Назва роботи: Технологія оптичного розпізнавання друкованих символів на основі нейронної мережі

Тип роботи: магістерська кваліфікаційна робота  
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки  
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 93,4 % Схожість 6,6 %.

Аналіз звіту подібності (відмітити потрібне):

Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_ Захарченко С.М.  
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи \_\_\_\_\_ Дідич О. Д.  
(підпис) (прізвище, ініціали)

Керівник роботи \_\_\_\_\_ Мартинюк Т. Б.  
(підпис) (прізвище, ініціали)