

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

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Технологія оптичного розпізнавання рукописних символів в системах  
електронного документообігу»

Виконав: студент 2 курсу, групи 1КІ-20м  
напряму підготовки (спеціальності)  
123 — «Комп'ютерна інженерія»  
\_\_\_\_\_ Осельський В.В.

Керівник: к.т.н., доц. каф. ОТ

\_\_\_\_\_ Савицька Л.А.

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

Опонент: д.т.н., проф. каф МБІС

\_\_\_\_\_ Яремчук Ю.Є.

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

**Допущено до захисту**

Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д.

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Рівень вищої освіти II-й (магістерський)  
Галузь знань 12 — Інформаційні технології  
Спеціальність 123 — «Комп'ютерна інженерія»  
Освітня програма — «Комп'ютерна інженерія»

### **ЗАТВЕРДЖУЮ**

Завідувач кафедри  
обчислювальної техніки  
\_\_\_\_\_ проф., д.т.н. О.Д. Азаров

« \_\_\_ » \_\_\_\_\_ 2021 р.

### **З А В Д А Н Н Я** **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ** Осельському Вадиму Валерійовичу

1 Тема роботи «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу»  
керівник роботи Савицька Людмила Анатоліївна, к.т.н., доцент,  
затверджені наказом вищого навчального закладу від 24.09.2021р. №227.

2 Строк подання студентом роботи 17.12.2021 р.

3 Вихідні дані до роботи: роздільна здатність зображення тексту не менше 300 dpi, оптична щільність зображення — не менше 2,5Т, модель кольорів для представлення зображення — RGB, кількість градацій яскравості зображення — 256.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, огляд і аналіз існуючих методів розпізнавання рукописних символів, розробка методу розпізнавання рукописних символів, розробка послідовності розпізнавання рукописних символів, розробка програми розпізнавання рукописних символів, розрахунок економічної доцільності створення програми розпізнавання рукописних символів..

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): методи розпізнавання символів, схема послідовності розпізнавання символів, структурна схема програми розпізнавання рукописних символів.

6 Консультанти розділів роботи представлені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1,2,3	Савицька Л. А., к.т.н., доцент		
4	Кавецький В. В., к.т.н., доцент		

7 Дата видачі завдання 07.09.2021 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2 — Календарний план

№	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	07.09.21	
2	Огляд існуючих методів розпізнавання рукописних символів	08.09-09.09.21	
3	Аналіз та вибір методів розпізнавання рукописних символів	10.09-28.09.21	
4	Розробка способу розпізнавання рукописних символів	29.09-11.10.21	
5	Розробка послідовності розпізнавання рукописних символів	12.10-21.10.21	
6	Розробка програми розпізнавання рукописних символів	22.10-31.10.21	
7	Розробка інтерфейсу та тестування програми розпізнавання рукописних символів	01.11-10.11.21	
8	Підготовка матеріалів та опис розробки засобів розпізнавання рукописних символів	11.11-16.11.21	
9	Розрахунок економічної частини роботи	17.11-30.11.21	
10	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12-06.12.21	
11	Аналіз виконання роботи, висновки, додатки	07.12-16.12.21	
12	Перевірка якості виконання магістерської роботи та усунення недоліків	17.12.21	

Студент \_\_\_\_\_ Осельський В. В.  
Керівник роботи \_\_\_\_\_ Савицька Л. А.

## АНОТАЦІЯ

УДК 004.9

Осельський В.В. Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2021, 115 с.

На укр.мові. Бібліогр.: 39 назв, рис.6, табл. 12.

У магістерській кваліфікаційній роботі розроблені засоби розпізнавання рукописних символів у системах електронного документообігу.

В магістерській роботі розглянуті методи розпізнавання рукописних текстів та послідовність розпізнавання рукописних символів тексту, запропоновано об'єднати згорткову нейронну мережу та структурний метод розпізнавання, що підвищує достовірність розпізнавання тексту, розроблені послідовність та програма для розпізнавання рукописних текстів.

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

Графічна частина складається з 7 плакатів із результатами експериментальних досліджень.

Ключові слова: формування ознак символів, сегментація, контур символа, згорткова нейрона мережа.

## ANNOTATION

Oselskyi V. Technology of optical recognition of handwritten symbols in electronic document management systems. 123 — computer engineering, educational program — computer engineering. Vinnytsia: VNTU, 2021, 115 p.

In Ukrainian. Bibliogr .: 39 titles, fig. 6, table. 12.

In the master's qualification work developed means of recognition of handwritten symbols in electronic document management systems.

The master's thesis deals with methods of handwriting recognition and sequence of handwritten text recognition, proposed to combine a convolutional neural network and a structural method that increases the reliability of text recognition, developed a sequence and program for handwriting recognition.

The qualification work also performs economic calculations to justify the feasibility of developing a new software product.

The graphic part consists of 7 posters with the results of experimental research.

Keywords: formation of character signs, segmentation, contour of the symbol, convolutional neural network.

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....</b>	<b>8</b>
<b>ВСТУП.....</b>	<b>9</b>
<b>1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ ТЕКСТОВИХ ДОКУМЕНТІВ.....</b>	<b>13</b>
1.1 Загальний підхід до оптичного розпізнавання тексту.....	13
1.2 Ознаки символів для автоматичного розпізнання.....	16
1.3 Аналіз методів розпізнавання символів.....	18
1.4 Використання штучної нейронної мережі.....	23
1.5 Послідовність розпізнавання символів.....	26
1.6 Підходи до розпізнавання рукописних символів.....	30
<b>2 РОЗРОБКА ЗАСОБІВ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ.....</b>	<b>43</b>
2.1 Послідовність розпізнавання рукописного тексту.....	43
2.2 Нейронні мережі.....	47
2.3 Використання бібліотек із відкритим кодом.....	55
<b>3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ.....</b>	<b>59</b>
3.1 Вибір інструментальних засобів програмування.....	59
3.2 Розробка архітектури програми розпізнавання.....	62
3.3 Розробка програми розпізнавання рукописних символів.....	63
3.4 Перевірка якості програми розпізнавання.....	68
<b>4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ.....</b>	<b>71</b>
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	71
4.2 Визначення рівня конкурентоспроможності розробки.....	75
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	78
4.3.1 Витрати на оплату праці.....	78
4.3.2 Відрахування на соціальні заходи.....	81
4.3.3 Сировина та матеріали.....	81

					<i>08-23.МКР.008.00.000 ПЗ</i>					
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу. Пояснювальна записка			<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
<i>Розроб.</i>		<i>Осельський В. В.</i>						6	104	
<i>Перевір.</i>		<i>Савицька Л. А.</i>						1КІ-20м		
<i>Реценз.</i>		<i>Яремчук Ю.Є.</i>								
<i>Н. Контр.</i>		<i>Швець С.І.</i>								
<i>Затверд.</i>		<i>Азаров О. Д.</i>								

4.3.4 Розрахунок витрат на комплектуючі.....	82
4.3.5 Спецустаткування для наукових (експериментальних) робіт.....	82
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт ..	83
4.3.7 Амортизація обладнання, програмних засобів та приміщень.....	84
4.3.8 Паливо та енергія для науково-виробничих цілей .....	85
4.3.9 Службові відрядження .....	86
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	87
4.3.11 Інші витрати.....	87
4.3.12 Накладні (загальновиробничі) витрати .....	87
4.4 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	88
ВИСНОВКИ .....	94
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	96
ДОДАТОК А Технічне завдання.....	100
ДОДАТОК Б Лістинг програми .....	104
ДОДАТОК В Послідовність розпізнавання символів документів .....	108
ДОДАТОК Г Методи розпізнавання рукописних символів .....	109
ДОДАТОК Д Послідовність розпізнавання тексту .....	111
ДОДАТОК Е Алгоритм розпізнавання рукописних символів .....	113
ДОДАТОК Ж Структура програми розпізнавання рукописних символів....	113
ДОДАТОК К Протокол перевірки навчальної (кваліфікаційної) роботи .....	115

					08-23.МКР.008.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БНМ — багатосарова нейронна мережа,

ЕОМ — електронна обчислювальна машина,

ЗНМ — згорткова нейронна мережа,

НМ — нейронна мережа,

ПЗ — програмне забезпечення.



## ВСТУП

Методи автоматичного розпізнавання образів та їх практична реалізація у системах оптичного читання текстів (OCR або OpticalCharacterRecognition) є проявом однієї із найбільш чудових технологій штучного інтелекту. Під системою OCR-системою сприймається автоматичне розпізнавання за допомогою спеціальних програм зображень символів друкованого або ж рукописного тексту й переведення їх у такий формат, що придатний для подальшого оброблення редакторами текстів, текстовими процесорами та інше. Під OCR-системою іноді розуміють пристрої автоматичного читання тексту або ж оптичного розпізнавання символів. У промисловому використанні пристроїв OCR передбачається оброблення документів хорошої й середньої якості, до яких можуть бути віднесені бланки статистичного обліку, бланки різних податкових декларацій, різноманітного виду банківські рахунки та інше [1-3].

Завдання переведення інформації із друкованих паперових на електронні носії є дуже актуальним не тільки у рамках потреб, що виникають в різних системах електронного документообігу. Сучасні інформаційні технології суттєво спрощують доступ до різноманітних інформаційних ресурсів за виконання такої умови, що вони будуть переведені в більш зручний для оброблення електронний вигляд. Найбільш швидким й простим є здійснення операції сканування текстових документів із використанням сканерів. У підсумку отримуємо цифрове зображення початкового документа в вигляді нового графічного файлу. Але значно кращим, в порівнянні з його графічним представленням, є текстове подання цієї інформації. Цей варіант дозволяє досить суттєво зменшити витрати на обсяги зберігання й передачі інформації, та створює додаткові можливості для реалізації усіх можливих сценаріїв використання й аналізу електронних документів. Тому із практичної точки зору значний інтерес представляє саме переведення наявних паперових носіїв у текстові електронні документи.

Дуже складним стає це завдання переведення документів при виконанні операцій автоматизованого розпізнавання рукописних текстів, яке

в даний час поки ще так і не вирішено в повному обсязі та вважається одним із найбільш актуальних й складних завдань розпізнавання даних [4].

Значний обсяг даних, що використовується для розпізнавання рукописних символів, вимагає формування нових й подальшого вдосконалення вже існуючих підходів до оброблення сформованих цифрових зображень рукописних символів текстових документів із метою виділення й розпізнавання рукописних символів. Значна сфера використання методів оброблення зображень рукописних текстових документів для розпізнавання у них рукописних символів вказує на **актуальність** цієї проблеми й вимагає подальших досліджень та вдосконалень.

**Метою дослідження** магістерської роботи є вдосконалення методів виділення та розпізнавання рукописних символів у текстових документах.

**Задачі дослідження** магістерської роботи:

— здійснити аналіз методів та способів побудови систем розпізнавання рукописних символів;

— запропонувати кращий підхід для розпізнавання рукописних символів;

— створити послідовність та розробити програму для обробки зображення для виділення та розпізнавання рукописних символів;

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

**Об'єкт дослідження** магістерської роботи є процес отримання даних про рукописний текстовий документ шляхом знаходження у виділеному фрагменті зображення рукописних символів та їх подальшого розпізнавання.

**Предметом дослідження** магістерської роботи є методи оброблення зображення текстового документу для виділення та розпізнавання рукописних символів.

**Методи дослідження** магістерської роботи: використовувались методи теорії множин для формування множини ознак для розпізнавання, методи

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

**Наукова новизна** отриманих результатів магістерської роботи полягає у тому, що удосконалено метод оброблення отриманого зображення текстового документу для виділення та розпізнавання рукописних символів тексту, який відрізняється від відомих підходів виконанням виділення характерних базових ліній символів та їх порівняння із еталонними зображеннями із використанням нейронної мережі, що дозволяє підвищити швидкодію розпізнавання символів у отриманому тексті.

**Практичне значення** одержаних результатів магістерської роботи:

- створено послідовність оброблення цифрових зображень для виділення та розпізнавання рукописних символів текстових документів;
- розроблено програму оброблення зображення для виділення рукописних символів та їх розпізнавання у рукописних текстах.

Матеріали магістерської роботи під назвою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» **подані до опублікування** на Всеукраїнську науково-практичну інтернет-конференцію «Молодь в науці: дослідження, проблеми, перспективи (МН-2022)».

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

Частиною загального процесу розпізнавання образів є система оброблення та розпізнавання текстових документів, а також виділення рукописних символів з урахуванням їх властивостей. Цей розділ магістерської роботи присвячений аналізу методів виділення та розпізнавання рукописних символів та обговоренню інструментів, які використовуються для виконання операцій розпізнавання рукописного тексту в цифрових зображеннях.

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

Хоча сьогодні більшість документів створюється із допомогою комп'ютерів, завдання створення повністю електронної системи документообігу ще далеко не завершено. Як правило, існуючі системи охоплюють діяльність окремих організацій, а обмін даними між організаціями здійснюється за допомогою класичних паперових документів. Завдання перекладу інформації з паперового на електронний носій є актуальним не лише з точки зору вимог до систем робочого процесу. Сучасні інформаційні технології можуть значно спростити доступ до людських інформаційних ресурсів при перекладі в електронний формат. Найпростіший і швидкий спосіб — сканувати документи за допомогою сканера. В результаті виходить цифрове зображення документа, графічний файл. Текстове представлення інформації краще, ніж графічне. Ця опція значно зменшує зусилля, необхідні для зберігання та передачі інформації, і дозволяє реалізувати всі можливі сценарії використання та аналізу електронних документів. Тому переклад картки в електронний текстовий документ становить практичний інтерес. На вхід системи розпізнавання надходить растрова карта сторінки документа. Для алгоритмів розпізнавання бажано, щоб зображення для входу було найвищої якості. Коли зображення мають шум, різкість і низький контраст, алгоритми виявлення ускладнюються. Крок

сегментації дуже важливий у цьому підході, оскільки методи класифікації на основі контурів не вдаються без надійної сегментації. Завершення відбувається за традиційною поетапною системою, але деякі етапи були незвичними на той час і можуть продовжуватися. Першим кроком є посилення на компонентний аналіз, який збереже контури компонентів.

Найпростішим рішенням цієї проблеми можливе просте попиксельне порівняння двох зображень, щоб знайти різницю. Дійсно, цей варіант відображає всі відмінності, але чи всі вони дійсно важливі? Як виявилось, ні. Деякі відмінності, які може розрізнити машина, насправді незначні, оскільки вони абсолютно невидимі для людини. Наприклад, проста мова може бути причиною того, що тести «провалюються», тобто показують результат, який нас не задовольняє. Проблема згладжування символів є досить серйозною для візуального тестування, оскільки різні машини можуть виконувати цей процес за різними алгоритмами. Це, наприклад, буква «А» висотою 8 пікселів (рис. 1.1).

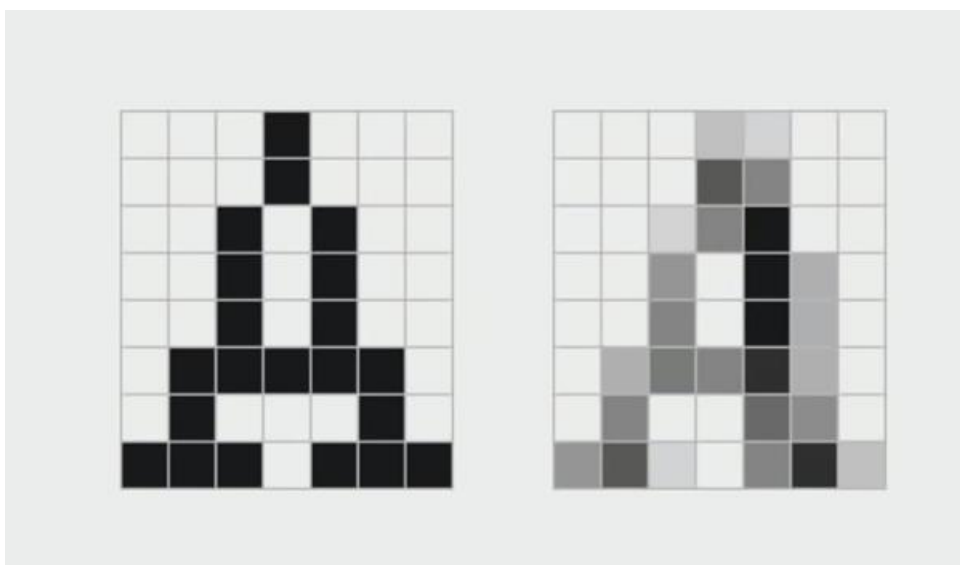


Рисунок 1.1 —Зображення великої літери «А»

І перший малюнок без згладжування, а другий з ним. Це відоме як згладжування відтінків сірого. Як бачимо, різні алгоритми згладжування та різні конфігурації комп'ютера дають нам майже різні зображення з точки зору машини. Для нас ця різниця незначна, тому що в обох випадках текст знаходиться в одних і тих самих місцях, а текст абсолютно однаковий. Тому

нам потрібно розробити систему, яка не чутлива до цього впливу. Основною проблемою в цих випадках є оптимізований алгоритм розпізнавання символів, який може розпізнавати символи правильно і швидко. Виходячи з цього, предметом дослідження є алгоритм розпізнавання символів, а об'єктом дослідження є сама нейронна мережа.

Більшість сучасних OCR працюють на трьох рівнях: шрифти, слова та сторінки. Однак, відповідно до принципів IPA, ABBYY запровадила ще один рівень у FineReader: весь багатосторінковий документ. Спочатку це було необхідно для правильного відтворення логічної структури, яка в сучасних документах стає дедалі складнішою [5-8].

З цієї причини була розроблена ADRT (Adaptive Document Recognition Technology), технологія для аналізу та синтезу документа на логічному рівні. Зрештою, це допомагає FineReader працювати якомога ближче до оригіналу [9]. Системи, подібні до AbbyFineReader, включають SimpleOCR, Readiris, OmniPage та CuneiForm.

Особливої уваги заслуговує Tesseract — система розпізнавання тексту, розроблена компанією Hewlett-Packard, яка безкоштовно розповсюджується Google з 2006 року і дуже швидко розвивається останніми роками [10].

Сьогодні у світі існує багато різноманітних програм для розпізнавання символів, але слід зазначити, що здатність читати неякісний текст, прочитаний людиною, поки що не під силу комп'ютеру. Кожен друкований текст має певну характеристику: символ, яким він набраний. В результаті існує два типи алгоритмів розпізнавання символів: шрифтовий і незалежний від шрифту. Шрифтозалежні алгоритми використовують важливу інформацію про шрифт із літерами. Це означає, що програма повинна мати повний зразок тексту, надрукований цим шрифтом. Програма вимірює та аналізує різні властивості шрифтів і вставляє їх у свою базу даних опорних характеристик. Існує три основні методи визначення тексту.

Шаблонний метод; Методи шаблону перетворюють зображення креслення на растрове зображення, порівнюють його з усіма доступними шаблонами в базі даних і вибирають шаблон із найменшою кількістю різних

точок із вхідним зображенням. Шаблонні методи досить стійкі до дефектів зображення і мають високу швидкість обробки вхідних даних, але надійно розпізнаються лише символи, шаблони яких відомі. І якщо розпізнаний символ дещо відрізняється від еталонного, методи моделі можуть бути неправильними навіть при обробці дуже якісних зображень.

Структурний метод; Структурно-методичне розпізнавання зберігає інформацію не про сучасний сценарій персонажа, а про його топологію. Стандарт містить інформацію про взаємне розташування окремих компонентів. Перевагою процедури є стійкість до зміщення та повороту символу на невеликий кут, різні стилістичні варіанти символів. Однак, якщо кут повороту становить кілька десятків градусів, цей метод не можна використовувати для розпізнавання символів. За допомогою цього методу такі властивості, як розмір розгорнутих літер і навіть шрифти, які використовуються для друку, невідомі. Однак головною проблемою цього методу є ідентифікація, яка містить деякі помилки [11].

## 1.2 Ознаки символів для автоматичного розпізнавання

Методи малювання засновані на вирівнюванні зображення за вектором малювання. Суть розпізнавання щодо вектора символів із серією прикладів векторів одного типу. Перевагами методу є проста реалізація, хороша узагальнюваність, висока швидкість розпізнавання. Недоліком цього методу є висока чутливість дефектів зображення. Крім того, функції функцій мають ще один недолік: при створенні функції втрачається деяка інформація про символ. Вибір функції незалежний, тому інформація про взаємне розташування елементів креслення втрачається [12].

Довідкові методи. Референтні методи передбачають порівняння конкретного невиявленого виду з набором стандартів. Для цього використовують нейронні мережі, які необхідно заздалегідь зібрати зі стандартами. Існує кілька алгоритмів порівняння тексту зі стандартним.

Найпростіший спосіб — порівняти піксель за пікселем, але він вимагає порівнянних площин зображення. Інші варіанти включають накладання та

зміщення, які вирівнюють зображення один з одним. Методи перевірки рукописного та друкованого тексту на перший погляд схожі, але істотно відрізняються. На відміну від шрифтів, стандарти рукописного введення є лише приблизними прикладами. Це значно підвищує ймовірність помилок [13, 14].

У типовій системі оптичного розпізнавання символів (OCR) введені символи зчитуються та скануються за допомогою оптичного сканера. Далі кожен символ ідентифікується і розподіляється, а отримана матриця попередньо обробляється — тобто згладжується, фільтрується і нормалізується. Після попередньої обробки присвоюються відмітні ознаки, на основі яких проводиться класифікація.

Алгоритм скелетизації — це спосіб розпізнавання окремих бінарних зображень на основі побудови скелетів цих зображень і вибору скелета ребер і вузлів. Потім створюється таблиця з пов'язаними зображеннями на основі співвідношення між ребрами, їх кількістю та кількістю вузлів. Наприклад, скелет кола — це вузол, скелет букви «П» має три ребра і два вузли, а ребра — 2: 2: 1. У програмуванні цей метод має кілька можливих реалізацій [15].

Нейронна мережа. Цей метод був дуже популярний у 1960-1970-х роках, його легко було заповнено та відхилено, оскільки солідна кількість вимагає значної обчислювальної потужності, яка зазвичай не зустрічається на простих мобільних платформах. Однак слід пам'ятати, що нейронні мережі іноді дають дуже цікаві результати, через які вони не мають структури, тим більше, що деякі нейронні мережі здатні розпаковувати інваріантність. Наприклад, мережі на основі Neosonetron можуть ідентифікувати певні характеристики та зламати їх так, ніби вони повертаються [16-19].

Інваріантні числа. З геометрії зображень ви можете вибрати різні числа, які є незмінними за розміром зображення та поворотом, а потім створити таблицю, яка відповідає цим числам конкретного зображення (майже як алгоритм скелетизації). Прикладами інваріантних чисел є число Ейлера, ексцентриситет, орієнтація (у сенсі положення головної осі інерції відносно деяких інваріантів).



Порівняння у відсотках по пунктах зі стандартом. Попередня обробка потрібна для досягнення незмінності розміру та положення, а потім порівняння з підготовленою стандартною базою даних зображень — якщо збігається більше одного конкретного тегу, ми вважаємо, що зображення буде розпізнане [20].

### 1.3 Аналіз методів розпізнавання символів

Порівняйте малюнок і модель. Ця група методів заснована на операції прямого порівняння зображень, отриманих з тесту, і доступних опорних символів. Розраховується ступінь подібності отриманого зображення з кожним із еталонів. Отримана в результаті тестова класифікація символу заснована на методі найближчого сусіда. Існують також інші методи порівняння текстових зображень, тобто методи послідовного співвіднесення та фільтрації зображень.

З практичної точки зору ці методи прості в реалізації і використовуються багатьма комерційними системами OCR. Однак при «фронтальній» реалізації методу кореляції навіть невелика темна пляма, що впала на зовнішній контур символу, може значно погіршити результат розпізнавання. Тому для досягнення хорошої якості розпізнавання в системах, які використовують узгодження шаблонів, використовуються інші спеціальні методи порівняння зображень.

Основна модифікація алгоритму порівняння моделей використовує представлення моделей у вигляді набору певних логічних правил.

У групі ознакових методів статистичні ознаки функцій формуються на основі аналізу різноманітних статистичних розподілів. Найвідоміші методи цієї групи використовують обчислення моментів і розрахунок точок перетину [21].

Моменти різного порядку успішно використовуються в різних областях комп'ютерного зору як дескриптори форми виділених ділянок і об'єктів. У розпізнаванні символів тексту значення моментів набору «чорних» точок відносно вибраного центру використовуються як набір ознак. У цьому типі

застосування найчастіше зустрічаються порядкові, центральні та нормовані моменти.

Для цифрового зображення, що зберігається у двовимірній матриці, порядковий момент є функцією координат кожної точки на зображенні. Центральні моменти є функцією відстані точки від центру ваги символу, де  $x_0$  і  $y_0$  — координати центру ваги. Нормовані центральні моменти отримують шляхом ділення центральних моментів на моменти нульового порядку. Слід зазначити, що терміни, як правило, пропонують нижчий рівень визнання. Центральний і нормований моменти кращі за перетворення зображень через більшу інваріантність.

У процесі вирізання позначки формуються шляхом підрахунку того, як часто і як зображення символу перетинається з виділеними лініями, намальованими під певними кутами. Цей метод широко використовується в комерційних системах, оскільки він інваріантний до спотворення та невеликих стилістичних варіацій у написанні символів, має досить високу швидкість і не вимагає великих обчислювальних витрат [22].

Метод зон ділить область рамки, яка охоплює символ, на області і використовує щільність точок у різних областях як набір характеристик. Зображення розбиваються на зони, обчислюються ваги пікселів кожної із зон, а також вектор відстані від опорних векторів опорних символів.

У методі матриці суміжності характерними вважаються значення частот поєднання «чорних» і «білих» елементів разом у різних геометричних комбінаціях.

Метод ознак локусів використовує в якості знака частоту, з якою вертикальний і горизонтальний вектори перетинають сегменти, вибрані для кожної світлової точки у фоновій області символу. У цій групі також є ряд інших методів.

Серед сучасних технологій розпізнавання, заснованих на різних типах перетворень, є методи, що використовують дескриптори Фур'є символів і дескриптори частотних меж.

Переваги методів з використанням перетворення Фур'є-Меллена полягають у тому, що вони інваріантні щодо масштабу, обертання та трансляції символу. Основним недоліком цього методу є нечутливість до різких стрибків яскравості на кордонах символу, напр. у спектрі просторових частот важко відрізнити символ «O» від символу «Q» тощо. У той же час може бути корисним фільтрувати шуми по межах символу.

Функції текстури зазвичай використовуються, щоб виділити загальну структуру зображення. Вони описують геометричні та топологічні властивості символу [22]. Найпростішим можна уявити ідею структурного розпізнавання текстового символу по відношенню до завдання автоматичного зчитування поштових індексів. У таких «модельних» символів положення кожного можливого сегмента обведення відоме заздалегідь і один символ не менше відрізняється від іншого, ніж наявність чи відсутність такого повного обведення. Подібна проблема виникає з керуванням простими рідкокристалічними дисплеями. У таких системах відбір структурних компонентів зводиться до аналізу елементів раніше відомої форми (групи сегментів, які потрібно виявити).

У структурних системах розпізнавання для складніших символів часто використовуваними ознаками є також лінії, за допомогою яких ідентифікуються такі ознаки зображення: кінцеві точки, точки перетину, замкнуті кільця та їх положення відносно рамки, що покриває символ.

Основна перевага методів розпізнавання структури визначається їх стійкістю до зміщення, масштабування та повороту символу на невеликий кут, а також можливими спотвореннями та різними стильовими варіаціями та незначним спотворенням символів, що аналізуються.

Розрізняють шрифтозалежні та шрифтонезалежні алгоритми. Ці методи засновані на концепції одного типу даних. Тобто фокусується на символах, які вже відомі, наприклад, метод малювання [23]. Алгоритми, що залежать від символів, використовують апріорну символну інформацію, яка друкує певні літери, цифри або символи. Це означає, що система розпізнавання повинна мати повний зразок тексту, надрукований шрифтом. Програма

вимірює та аналізує різні риси характеру та зберігає їх у своїй базі даних опорних рис. Коли цей процес завершено, оптичний розпізнавач символів готовий розпізнати цей символ. Цей процес можна назвати резюме. Навчання повторюється для деяких персонажів, залежно від сфери застосування.

До недоліків цього методу можна віднести наступні фактори. Алгоритм повинен заздалегідь знати шрифт, який він отримає для розпізнавання, тобто він повинен зберігати різні властивості цього шрифту в своїй базі даних. Якість розпізнавання тексту, надрукованого випадковим символом, прямо пропорційна відношенню цього символу до символів, що містяться в базі даних системи розпізнавання. Якщо в процесі навчання є великий тиск, охопити всіх персонажів та їх зміни неможливо. Наприклад, сучасні комп'ютеризовані системи друку документів використовують близько 200 різних символів, і ця кількість з кожним роком збільшується. Іншими словами, цей фактор обмежує універсальність таких алгоритмів.

Для розпізнавання потрібен блок налаштування для певного символу. Очевидно, що цей пристрій внесе свою частку помилок в інтегральну оцінку якості розпізнавання, або ж функція налаштування шрифту буде залишена за користувачем [24].

Програма на основі алгоритму символів розпізнає символи користувача зі спеціальними знаннями про символи в цілому, їх групи та відмінності між ними, символи, якими друкується документ користувача. Зауважте, що не існує звичайного способу дізнатися, які шрифти використовував для друку ваш документ, якщо паперовий документ не був створений користувачем і ввів його невідомим способом. Потреба в спеціальних знаннях звужує коло потенційних користувачів і приводить їх до організацій, які мають відповідний персонал.

З іншого боку, характерний підхід має значну перевагу, тому активно використовується і, ймовірно, буде використовуватися в майбутньому. Тобто, розробляючи апріорну інформацію про символи, можна створити досить точні й надійні алгоритми розпізнавання. Загалом, при створенні алгоритму розпізнавання набору символів надійність і якість розпізнавання

символів є інтуїтивно зрозумілими, а значення, яке можна математично виразити з точністю. Це значення визначається як відстань у будь-якому метричному просторі від опорного символу, наданого програмі в процесі навчання, до символу, який програма намагається розпізнати.

Іншим класом алгоритмів є безсимвольні, тобто алгоритми, які не мають попередніх знань про символи, які надаються їм як вихідні дані. Такі алгоритми вимірюють і аналізують різні властивості (характеристики), властиві буквам, які практично не змінюються в будь-якому символі [25]. У деяких випадках залежний від символів алгоритм може не мати кривої навчання. У цьому випадку властивості символу вимірюються, кодуються і зберігаються в базі даних користувачем. Однак на практиці цей варіант рідко вирішує проблему розпізнавання. Більш загальним методом створення бази даних є підготовка програми до вилучення символів у режимі реального часу.

До недоліків такого підходу можна віднести наступні фактори. Насправді, досяжна якість розпізнавання тексту нижча, ніж за допомогою алгоритмів шрифтів. Це тому, що ступінь узагальнення при вимірюванні властивостей шрифту набагато кращий, ніж у шрифтозалежних алгоритмах. Фактично це означає, що різні припущення та нерівності при вимірюванні властивостей символів для роботи алгоритмів без символу можуть бути в 2—20 разів більшими, ніж символи.

Це можна вважати великим успіхом, якщо алгоритм є достатнім і фізично бездоганим без символу, тобто надійність коефіцієнта розпізнавання природно виходить з основного алгоритму алгоритму. Часто доводиться погодитися, що оцінка точності відсутня або створена штучно. Штучна оцінка передбачає, що вона істотно не узгоджується з ймовірністю правильного виявлення алгоритмом.

Переваги цього підходу тісно пов'язані з його недоліками. Основними перевагами є універсальність, адаптивність і простота використання програм.

Універсальність. З одного боку, це означає можливість використання такого підходу в тих випадках, коли потенційна різноманітність персонажів, які можуть потрапити в систему як оригінали, досить велика. З іншого боку,

завдяки своїй узагальнювальності, такі алгоритми можуть екстраполювати знання, накопичені поза шаблоном навчання, тобто постійно розпізнавати символи, які далекі від символів у шаблоні навчання.

Процес навчання залежних від шрифту алгоритмів зазвичай простіший і більш інтегрований у тому сенсі, що зразок не розбивається на різні класи (символи, шпильки тощо). Однак не обов'язково підтримувати різні умови для співіснування цих класів у базі даних об'єктів. Іншим проявом адаптивності є той факт, що часто можна створити майже повністю автоматичні програми навчання.

Комфорт у використанні програми. Якщо програма заснована на шрифтозалежних алгоритмах, користувачеві не потрібно нічого знати про сторінку, яку він хоче вставити в пам'ять комп'ютера і передати ці знання програмі. Це також спрощує інтерфейс користувача програми через відсутність набору параметрів і діалогових вікон, які використовуються для навчання та керування функціональною базою даних. У цьому випадку процес розпізнавання може бути представлений користувачеві як «чорний ящик» (користувач не має можливості контролювати хід процесу розпізнавання або в будь-якому випадку змінювати його). В результаті розширюється коло потенційних користувачів, у тому числі з мінімальною комп'ютерною грамотністю.

#### 1.4 Використання штучної нейронної мережі

Для розпізнавання символів часто використовуються штучні нейронні мережі. Алгоритми, розроблені для нейронних мереж розпізнавання символів, не часто будуються так: зображення символу, яке є вхідним зображенням для читання, зменшується до певного стандартного розміру. Зазвичай зображення зменшується до 16x16 пікселів.

Значення яскравості у вузлах нормалізованого зображення застосовується як вхідні параметри нейронної мережі. Кількість вихідних параметрів нейронної мережі такий самий, як кількості розпізнаних символів. Результатом розпізнавання є символ, що відповідає найвищому значенню

джерела нейронної мережі. Підвищення надійності таких алгоритмів зазвичай передбачає пошук більш інформативної функціональності введення або ускладнення структури нейронної мережі.

Від вибору структури та параметрів нейронної мережі значною мірою залежить надійність виявлення та потреба програми в обчислювальних ресурсах. Зображення зображень зменшуються до одного розміру (28x28 пікселів). Отримане зображення надсилається на вхід нейронної мережі, яка має три внутрішні шари і 10 вузлів верхнього шару. Нижні шари мережі не пов'язані між собою. Нижні вузли мають загальний набір ваг. На думку розробників, це повинно підвищити здатність нижніх шарів мережі розрізняти основні функції зображення. Невикористані масштаби видаляються, щоб збільшити здатність мережі до узагальнення та зменшити кількість необхідних обчислень і пам'яті. Це зменшує кількість незалежних параметрів на чотири. Навчання нейронної мережі виконується на наборі з 7300 символів, тест з багатьох 2000 символів. Похибки виявлення становлять приблизно 1% для навчального набору і 5% для тесту [18].

Замість значень яскравості у вузлах нормованої сітки в якості вхідних параметрів нейронної мережі можна використовувати значення, що характеризують різницю яскравості. За допомогою цих вхідних параметрів ви можете краще визначити межі літер. Об'єкти, що сприймають, зменшуються до 16x16 пікселів. Потім вони додатково обробляються, щоб виділити ділянки з найбільшою різницею в яскравості. Одним із широко використовуваних методів для підвищення точності розпізнавання є одночасне використання кількох модулів розпізнавання та подальша інтеграція результатів (наприклад, за допомогою координації). Дуже важливо, щоб алгоритми, які використовуються цими модулями, були максимально незалежними. Цього можна досягти за допомогою модулів розпізнавання, які в основному використовують різні алгоритми розпізнавання, а також спеціального відбору навчальних даних.

Один з таких методів був запропонований кілька років тому і заснований на використанні трьох модулів виявлення. Перший модуль

викладається як зазвичай. Другий модуль перевіряє символи, відфільтровані першим модулем, так що другий модуль бачить комбінацію символів, 50% з яких перший модуль розпізнає правильно, а 50% — неправильно. Нарешті, третій модуль вивчає символи, які відрізняють результати розпізнавання 1-го та 2-го модулів. Під час тесту всі три модулі використовують розпізнані символи. Оцінено продуктивність усіх трьох модулів. Визнається персонаж з найбільшою сумою балів.

Як правило, алгоритм розпізнавання заснований на розділенні сітки із зображенням букв основних ознак і на подальшому використанні штучної нейронної мережі для оцінки близькості вхідного зображення до символів заданого набору букв. Результатом є набір оцінок, які відображають ступінь близькості розпізнаного символу до символів заданого набору символів [25].

Розпізнаваний набір символів може містити літери та цифри. Вхідні дані для розпізнавання символів перетворюються у вимір. Особливістю реалізованого алгоритму є використання нейронної мережі з великою кількістю вхідних функцій. Оригінальне зображення виділяє основні особливості, що характеризують різницю яскравості у вузлах сітки. Нейронна мережа має внутрішній шар, який містить 100 вузлів і пов'язаний універсально, тобто кожен внутрішній  $X_i$  вузол підключений до всіх вхідних вузлів, а кожен вузол вищого рівня підключений до всіх вузлів внутрішнього рівня. Щоб зменшити кількість обчислень розпізнавання, для кожного зображення розпізнаного символу використовуються не всі вхідні символи, а лише частина з них, тобто  $X_i$  вектор вхідних параметрів нейронної мережі сильно розбавлений [26].

Для роботи системи розпізнавання тексту необхідно вставити цифровий файл зі відсканованим текстом (текстовий елемент). Наступним етапом системи розпізнавання тексту є так звана «сегментація» та попередня обробка даних. На цьому етапі роботи відбувається вирівнювання елементів тексту. Текстовими елементами можуть бути фотографії з певними фрагментами тексту, частинами відсканованих сторінок і подібними параметрами цифрового макета. Після вирівнювання всі ці елементи мають



бути звільнені від непотрібних частинок, тобто текст повинен складатися виключно із «знайомих» для системи символів, а решта — для того, щоб очистити поле зору системи або просто пропустити такі елементів. Потім потрібна процедура бінаризації, яка полягає в ізоляції фону та інших елементів для читання як білих елементів, а потрібний текст вводиться в систему лише як чорний текст. Потім система розпізнавання відокремлює стовпці та рядки тексту, ділить безперервний текст на слова та змінює розмір зображення для легкого порівняння з усталеними «посилальними» словами та символами.

У майбутньому розпізнавання тексту здійснюватиметься двома способами за допомогою символу, а саме методом порівняння зображень та адаптивним процесом розпізнавання. Кожен з них має певні переваги та недоліки, а також різні оптимальні ситуації для їх використання, розглянуті нижче.

Як тільки ви виходите з програми, необхідним і дуже важливим кроком є збільшення швидкості роботи програми і підвищення точності отриманих даних. Для цього додайте кілька словників, які допоможуть системі певного набору символів сформувавши слова, які найкраще відповідають логіці всього тексту, і виправити деякі невеликі неточності в словах. Ще один важливий спосіб підвищити точність вихідних даних — включити в граматичні правила розпізнану систему букв, щоб правильно відрізнити іменники від дієслів та інших подібних відмінків.

### 1.5 Послідовність розпізнавання символів

Типова система розпізнавання рукописного тексту складається з кількох основних етапів, а саме: попередня обробка зображення, сегментація, виявлення ознак, класифікація та пост обробка [24].

Після отримання зображення система розпізнавання отримує зображення як вхідне зображення. Зображення повинно мати певний формат, наприклад у форматах JPEG, BMP, PNG тощо. Це зображення можна

отримати за допомогою сканера, цифрової камери або іншого відповідного цифрового пристрою введення.

Наступним є попередня обробка — це серія операцій, що застосовуються над відсканованим вхідним зображенням. Це значно покращує та сегментує зображення. Послідовність операцій, що виконуються над зображенням. Процес бінаризації перетворює чорно-біле зображення на двійкове зображення за допомогою методу глобального порогового значення. Розширення країв у двійковому зображенні здійснюється шляхом відбілювання, розширення зображення та заповнення в ньому отворів — це операції, які виконуються на останніх двох кроках, щоб отримати попередньо оброблене зображення, придатне для сегментації.

На етапі сегментації зображення рядок символів розбивається на підсторінки окремого символу. У пропонованій системі попередньо оброблене вхідне зображення сегментується на окремі символи, кожному символу присвоюється номер у процесі маркування. Цей символ надає інформацію про кількість символів на малюнку.

Фаза класифікації є частиною процесу прийняття рішень у системі розпізнавання. Нейронна мережа, що може застосовуватися у роботі, використовується для класифікації та розпізнавання рукописних символів. Вводиться сегментоване зображення останнього кроку. Загальна кількість нейронів у вихідному шарі відповідає кількості символів конкретного алфавіту, оскільки система розробляється для розпізнавання заданого алфавіту.

Фаза постобробки є завершальною фазою системи розпізнавання. Тут відповідні розпізнані символи друкуються у структурованому тексті шляхом обчислення еквівалентного значення ASCII за допомогою індексу розпізнавання тестових шаблонів.

Необхідні дані піддаються різним фазам попередньої обробки, які використовуються в описовій фазі аналізу символів. Попередня обробка спрямована на отримання даних, які зручно використовувати для систем оптичного розпізнавання символів. Основними завданнями попередньої

обробки є фільтрація шуму, морфологічні операції, бінаризація, вирівнювання країв і кроки розпізнавання.

Фільтрація шуму є важливою частиною програми розпізнавання символів. Під час сканування документа додатковий шум може забруднити відскановані зображення, і ці зображення низької якості впливають на наступний етап обробки документа. Тому для покращення якості зображення необхідний етап попередньої обробки, перш ніж перейти до наступних кроків обробки документів. Шум може спричинити віддалені сегменти помилок, довгий міжрядковий інтервал тощо. Важливо усунути всі ці помилки для кращої інформації. У зображеннях є багато типів шумів. Додатковий шум, відомий як «шум солі та перцю», складається з чорних і білих крапок, розкиданих по всьому зображенню, які можна знайти практично в будь-якому документі. Методи шумозаглушення можна розділити на дві основні групи.

Фільтрація — спрямована на видалення шумів і зменшення незначних плям, які зазвичай виникають на грубій поверхні запису та/або викликані низькою частотою дискретизації пристрою збору даних.

Морфологічні операції — зазвичай використовуються як інструмент обробки зображень для отримання компонентів зображення, корисних для представлення та опису форми сайту. Морфологічні операції можна успішно використовувати для видалення шумів у зображенні документа через погану якість паперу та чорнила та неправильні рухи рук.

Бінаризація чорно-білих зображень шрифтів є важливим кроком у розпізнаванні шрифтів в автономному режимі. Хороша бінаризація полегшує сегментацію та розпізнавання символів. Процес бінаризації перетворює початкове зображення в двійкове зображення.

Виявлення країв використовується для подальшої ідентифікації об'єктів. Краї характеризують межі об'єктів і тому корисні для сегментації, реєстрації та ідентифікації об'єктів. Виявлення країв зображення значно зменшує обсяг даних і відфільтровує непотрібну інформацію, зберігаючи важливі структурні властивості зображення. Існує багато способів

визначення країв. Однак найрізноманітніші методи можна розділити на дві категорії: метод градієнта і метод Лапласа. Метод градієнта виявляє краї, шукаючи максимум і мінімум у першому отриманому зображенні. Метод Лапласа шукає нульові перетини у другому отриманому зображенні, щоб знайти ребра.

Виявлення спотворень — в результаті процесу сканування може виникнути асиметрія. Існує кілька поширених способів виявлення бічних протягів; деякі покладаються на пошук зв'язаних компонентів і пошук центральних кутів, що з'єднують їхні центри тяжіння. Необхідно усунути асиметрію, оскільки це знизить точність документа. Розраховується кут перекосу, а похилі лінії створюються горизонтально, щоб бути кутом перекосу.

Оцінка та нормалізація зображення: рукописний текст зазвичай позначається курсивом. Зокрема, символи можуть бути нахилені як справа наліво, так і навпаки. Крім того, різні варіанти можуть з'являтися не тільки в тексті, але і в межах одного слова. виправлення перекосу не впливає на текстуру слів, і отримані слова є природними. Skew Normalization використовується для нормалізації всіх символів до стандартної форми. Найпоширенішим способом оцінки ухилу є обчислення середнього кута сусідніх вертикальних елементів.

У процесах розпізнавання символів сегментація є найважливішим процесом. Сегментація проводиться для того, щоб поділити зображення на окремі символи. Поділ рукописного слова на різні області (верхню, середню та нижню) та символи є складнішим, ніж у друкованих документах. В основному це пов'язано з різницею між символічними відстанями, нахилом, нахилом, розміром, тобто рукописними літерами. Іноді компоненти двох послідовних символів можуть бути залучені або перекриватися, і ця ситуація ускладнює завдання сегментації. В індійських мовах через різну символіку верхньої та нижньої зон часто спостерігається таке дотик чи накладання. Сегментація є важливим кроком, оскільки ступінь поділу слів, рядків або символів безпосередньо впливає на швидкість розпізнавання тексту [23].

Існує два види сегментації: зовнішня і внутрішня.

Зовнішня сегментація: розбиває макет сторінки на логічні одиниці. Зовнішня сегментація — це вибір різних блоктів, наприклад абзаців, речень або слів. Це найважливіша частина аналізу документів. Аналіз і розпізнавання документів спрямований на автоматичне вилучення інформації, представленої на папері, і спрямований на розуміння людини з самого початку. Сегментація сторінки є одним із важливих кроків у аналізі макетів і особливо складна зі складними макетами. Розмітка сторінки аналізується у два етапи: перша фаза — це структурний аналіз, під час якого зображення розбивається на блоки компонентів документа (абзац, рядок, слово тощо). Другий — функціональний аналіз, який використовує макет, розміри та різні правила розмітки, щоб показати функціональний зміст компонентів документа. Потім сторінку сегментують, шукаючи області з чорно-білими текстурами.

Внутрішня сегментація — це операція, яка намагається розділити рядок символів на часткові зображення окремих символів. Хоча ці методи еволюціонували за останнє десятиліття і з'явилися різні техніки, сегментація рукописних символів на символи залишається невирішеною проблемою.

## 1.6 Підходи до розпізнавання рукописних символів

В останні роки розпізнавання рукописних символів було одним із найбільш захоплюючих і перспективних напрямків досліджень у сфері обробки зображень та розпізнавання образів. Кілька дослідників зосередилися на нових методах та ідеях, які б скоротили час обробки та забезпечили б більшу точність виявлення. Загалом розпізнавання рукописних символів поділяється на два типи: офлайн і онлайн. При розпізнаванні в автономному режимі рукописні символи зазвичай скануються за допомогою оптичного сканера, а готовий лист доступний у вигляді зображення. Онлайн-методи перевершують протилежні офлайн-методи через доступність даних часу в онлайн-методах. Однак такі програми, як сортування пошти, обробка банківських чеків, читання документів і розпізнавання поштових адрес,

вимагають офлайн-техніки розпізнавання рукописного тексту. Тому офлайнове розпізнавання рукописного тексту продовжує залишатися активною та перспективною сферою досліджень і вимагає вивчення нових методик, які б підвищили точність розпізнавання символів.

Офлайн — розпізнавання тексту, вже написаного на папері. Текст подається у вигляді скану або фотографії документа, сторінки книги тощо. Це складніший спосіб, оскільки у випадку онлайн-методу можна простежити за процесом написання тексту та побудувати на його основі алгоритм розпізнавання. Складність проблеми розпізнавання рукописного тексту полягає в різноманітності написання, форми, розміру літер у різних мовах. Крім того, папір з текстом може містити «шуми»: Дефекти паперу, сторонні плями, які ще більше ускладнюють весь процес.

Вилучення ознак все ще є одним з основних блоків створення послідовності розпізнавання рукописного тексту. Ефективність системи розпізнавання залежить головним чином від виділення ефективних ознак і від вибору відповідного класифікатора. Крок виділення ознак аналізує одиницю сегмента (частини) тексту та вибирає набір ознак, які можна використовувати для однозначної ідентифікації цієї одиниці. Отримані ознаки потім використовуються як вхідні дані для класифікатора рукописного введення. Ці функції є зменшеним уявленням вмісту зображення, яке зосереджується на підтримці особливостей, які є найбільш важливими для завдання ідентифікації. Метою вилучення ознак є визначення шаблонів за допомогою мінімальної кількості ознак, які ефективні для розпізнавання класів шаблонів. Методи вилучення ознак можна в основному розділити на три основні категорії: статистичні, структурні та глобальної трансформації [15].

Представлення зображення документа за допомогою статистичного розподілу точок певною мірою враховує зміни стилю. Зазвичай це краще для легкого пошуку та швидкого обчислення, а також коли реконструкція зображення не потрібна.

Зображення значка символу поділено на блоки, які називаються зонами. Якщо горизонтальну площину розділити на  $m$  ліній, а вертикальну — на  $n$  ліній, кількість утворених зон дорівнює  $m \cdot n$ . Потім розраховується середня інтенсивність рівня сірого для певної ділянки зони. Інтенсивність зони блоку обмежена певною областю, а не всім зображенням. Форма символу однозначно ідентифікує кожен набір символів і тому використовується як ознака.

Інша популярна статистика — це кількість разів, коли відрізок лінії перетинає шлях, що моделює символ. Зображення символу поділено на  $k$  областей, і тому проекція в горизонтальному напрямку дає горизонтальну проекцію  $i$ , так само, у вертикальному та діагональному напрямках, вертикальну та діагональну проекції, відповідно.

Геометричні фігури на ілюстрації представляють бенгальську цифру вісім у стандартній формі (ліворуч) і спотвореним почерком (праворуч). Слід зазначити, що навіть з такими спотвореннями клітинна проекція обох символів дуже схожа.

У структурному методі розпізнавання різні локальні та глобальні властивості набору символічних зображень можуть бути представлені геометричними та топологічними ознаками, які дуже стійкі до стилю та спотворення почерку.

Символ представлений виділенням і підрахунком різних топологічних ознак, таких як максимальна, мінімальна та крайня точки, нижче і вище порога, кількість петель, кількість горизонтальних і вертикальних відрізків і кінцеві точки.

Інший підхід, коли символ представлений геометричними властивостями, такими як відношення ширини до висоти, відносна відстань від першого до останнього пікселя зверху вниз і може включати відстань між пікселями зліва направо.

У підході дерева та графу символи перетворюються на набір топологічних примітивів, включаючи лінії, кільця, перетини, кінцеві точки тощо. Потім ці примітиви формуються за допомогою реляційних атрибутів

або графіків. Ці графіки представлені двома методами. Перший метод використовує координати фігури символу. Другий метод полягає в тому, щоб абстрактно представляти вершини, що відповідають штрихам, і дуги, відповідні відношенню між штрихами. Деревя використовуються для представлення символів на основі різноманітних функцій, які мають ієрархічний зв'язок.

Глобальні методи трансформації перетворюють зображення в компактну форму, що зменшує розмір об'єкта. Перетворення зображення може бути представлено лінійною комбінацією загалом добре визначених наборів синусів і косинусів. Перші  $N$  коефіцієнтів перетворення Фур'є використовуються як характеристичний вектор певного символу. Оскільки доступне зворотне перетворення, зображення можна відновити.

Вейвлет-перетворення — це метод послідовного розкладання, який дозволяє нам представляти сигнал з різними рівнями роздільної здатності. Сегменти зображення документа, які можуть відповідати літерам або словам, характеризуються вейвлет-коефіцієнтами, які відповідають різним рівням роздільної здатності. Потім ці коефіцієнти надсилаються до класифікатора для розпізнавання.

Існує щонайменше два підходи до розпізнавання рукописного тексту, які дають прийнятний результат: використання прихованої марковської моделі та штучної нейронної мережі (ШНМ). На практиці також використовується гібридний підхід з використанням двох одночасних підходів.

Підготовка рукописного документа до розпізнавання включає виправлення, бінаризацію порогу та видалення шумів.

Порогова бінаризація — це процес відокремлення фону від об'єкта, у даному випадку тексту. Результат — чорний текст на білому тлі.

Зниження шуму: видаляє артефакти із зображення, не впливаючи на написаний текст.

Також проводиться сегментація рядків, слів, символів. Це поділ тексту на рядки, слова та символи, щоб полегшити їх розпізнавання за допомогою



ANN. Чим менше в тексті рядків, які виглядають як прямі, тим гірше працює алгоритм сегментації рядків. Елементарна сегментація слів працює за принципом, що інтервал між словами більше, ніж між буквами.

Змінюючи кут нахилу, форму букв випрямляють. Кут нахилу: кут між вертикаллю та напрямком букви. Фаза виправлення — це напрямок, щоб максимально зменшити цей кут.

Не всі алгоритми по розпізнаванню символів вимагають сегментації. Розглянемо різні способи розбиття тексту на слова.

Сторінка тексту розбивається на рядки, потім рядок на слова, відокремлені пробілом. Потім до тексту застосовуються фільтри, щоб видалити шум і визначити межі слів. Текст розбивається на складові — обчислюються компоненти та відстань між їх центрами. Як параметр алгоритму використовується порогове значення відстані, яке можна додатково вибрати залежно від успіху результатів.

Наступний підхід також базується на розрахунках відстані. Support Vector Machine знаходить шар, який розділяє два різні типи даних (символів), а потім використовує поріг, щоб розбити текст на слова.

Лінгвістична модель допомагає алгоритму розпізнавання покращувати результат, використовуючи відомі мовні фрази. Лінгвістична модель може передбачити наступне слово на основі попереднього і запропонувати варіанти з різною ймовірністю. Наприклад, «є» більш імовірно, ніж «ти». Таку модель можна навчити на великій кількості тексту, обчислюючи повторення одного слова за іншим. Звичайно, немає сильної впевненості в тому, що мовна модель може передбачити наступне слово для будь-якого тексту в усіх випадках, тому модель тренується на тексті тієї ж теми, що і розпізнаваний. Використання мовної моделі також є необов'язковим.

Алгоритм пошуку префіксів (дерево префіксів) для слова швидкий і легкий для розуміння. Як ви можете собі уявити, він використовує структуру дерева, де ребра є літерами, а вершина містить атрибут слова. Таким чином, слово представляється у вигляді шляху від кореневого вузла (першої букви слова) через краї — складені літери слова вгору з атрибутом слова.

Вхід класифікатора може отримувати зображення окремих слів, а також цілих рядків. ШНМ складається з шарів. Тут відбувається вся магія та математика: по-перше, він використовує згорткові нейронні мережі (ЗНМ) — операції згортки та об'єднання та повторювані нейронні мережі (ПНМ), один із типів LSTM, mdlstm, IDCN. Суть згортки полягає в тому, що кожен фрагмент зображення елемент за елементом множиться з матрицею (ядром) згортки і результат підсумовується і записується в аналогічному місці вихідного зображення (формується карта ознак) [27]. Процес групування може значно зменшити розмір зображення. Групування інтерпретується наступним чином: якщо деякі об'єкти вже були ідентифіковані в попередній операції згортки, таке детальне зображення більше не потрібне для подальшої обробки і стискається в менш детальне зображення — розмір створених карт об'єктів зменшується. Кінцевий результат також залежить від набору даних— набору шаблонів зображень для кожної літери, написаних іншим почерком.

Пост-обробка дозволяє виправити текст після класифікації. На даний момент ми маємо лише текстову інформацію без вихідних графічних даних. Наприклад, текст розбивається на слова, а потім перевіряється на наявність у словнику. Якщо в словнику немає слова, але воно схоже на слово, вам буде запропоновано варіант виправлення. Виправлення також можна тренувати паралельно за допомогою класифікатора.

Постобробка пропонує середнє покращення результатів на 15% для деяких алгоритмів, але може і погіршити результати (до 3-6%).

Актуальним завданням залишається розробка системи розпізнавання рукописних символів. Багато програмних рішень, широко доступних на ринку, не повністю вирішують проблему. Якщо рукописний текст можна розпізнати, то розпізнавання супроводжується багатьма помилками та неточностями. Крім того, більшість існуючих програм є платними, що ускладнює їх використання.

## 2 РОЗРОБКА ЗАСОБІВ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ

### 2.1 Послідовність розпізнавання рукописного тексту

Розпізнавання рукописного тексту відбувається в кілька етапів:

Попередня обробка зображення (preprocessing): На цьому етапі зображення обробляється з метою покращення його якості та приведення його у форму, придатну для сегментації.

Сегментація: на цьому етапі текст на зображенні вибирається і розбивається на його компоненти. Як правило, текст обробляється ієрархічно: спочатку вибираються окремі рядки, потім окремі слова, потім символи або частини символів.

Вилучення ознак: на цьому етапі створюються описи ознак частин, визначених на етапі сегментації.

Класифікація: На цьому етапі на основі описів ознак, які були створені на етапі виділення ознак, система приймає рішення щодо того, який відомий клас повинен класифікувати вибраний елемент на етапі сегментації.

Оброблення результатів (preprocessing): На цьому етапі побудова кінцевого тексту відбувається на основі результатів класифікації виділених частин тексту.

Попередня обробка зображення. Під час попередньої обробки здійснюється покращення зображення, яке приводить зображення в найбільш підходящий вигляд для подальшої автоматичної обробки [12]. Підвищення якості зображення для подальшого розпізнавання тексту включає усунення дефектів зображення та відокремлення тексту від фону [15].

Видалення дефектів. Помилки усуваються за допомогою стандартних методів обробки зображень. Найбільш часто використовуваним фільтром для видалення шуму є фільтр Гаусса для придушення високочастотного шуму і медіанний фільтр для видалення шуму [24].

Відділення тексту від фону є окремим випадком завдання виділення об'єкта на зображенні. Проблема можна сформулювати так: починаючи з

текстового зображення  $A$ , потрібно побудувати двійкове зображення  $B$  такого ж розміру з  $B(i, j) = 1$ , якщо піксель  $(i, j)$  належить до тексту на  $A$ , інакше  $B(i, j) = 0$ . Це перетворення зображення дає можливість подальшого використання аналізу з'єднаних компонентів, контурів, скелетів тощо.

Найпоширенішим методом, який використовується для відокремлення тексту від фону, є бінаризація порога. Для зображення  $I(i, j)$  — це яскравість пікселя з координатами  $(i, j)$ . Порогова бінаризація зображення — це покрокове перетворення  $f(i, j)$ , що

$$f(i, j) = 1, \text{ якщо } I(i, j) \geq d, 0, \text{ якщо } I(i, j) < d,$$

де  $d$  називається порогом бінаризації.

Зазвичай на гістограмі яскравості текстового зображення спостерігаються два піки: високий пік у області світлого пікселя (відповідає фону, тобто паперу) і нижчий пік у темній області пікселів (до тексту). Тому завдання пошуку порогу яскравості, тобто пікселів із яскравістю вище цього значення (фон) як чорний і нижче (текст), як білий (ця «інверсія» кольору виконується для спрощення застосування багатьох алгоритмів, в майбутньому) проблема полягає в пошуку оптимального значення між двома піками гістограми. Існують добре досліджені методи вирішення цієї проблеми, наприклад метод Оцу [28] та його варіанти. Результатом етапу попередньої обробки зображення є двійкове зображення білих чисел на чорному тлі, яке відповідає вихідному зображенню.

Сегментація. На етапі сегментації окремі цифри вибираються на зображенні, отриманому під час попередньої обробки. Результатом кроку сегментації має бути послідовність однозначних зображень, які потім надсилаються на вхід процедури виділення ознак. Ця стратегія, відома як стратегія явної сегментації, краще справляється з проблемою розпізнавання рукописних послідовностей цифр, ніж неявна сегментація, при якій виділення цифр, виділення ознак і класифікація здійснюються паралельно [24]. Призначити номерні групи. Після того, як текст був відокремлений від фону під час попередньої обробки, можна виділити групи чисел за допомогою аналізу зв'язаних компонентів. Якщо ми розглядаємо

білі пікселі як кутові точки графіка і з'єднуємо кутові точки, що відповідають сусіднім пікселям, ребрами, стає можливим вибрати компоненти зв'язку, які відповідають одній або кільком цифрам, що перетинаються. Існують стандартні й добре відомі алгоритми пошуку для визначення споріднених компонентів, наприклад пошук у глибину та пошук ушир.

Після того, як текст був відокремлений від фону під час попередньої обробки, можна виділити групи чисел за допомогою аналізу зв'язаних компонентів. Якщо ми розглядаємо білі пікселі як кутові точки графіка і з'єднуємо кутові точки, що відповідають сусіднім пікселям, ребрами, стає можливим вибрати компоненти зв'язку, які відповідають одній або кільком цифрам, що перетинаються. Існують стандартні й добре відомі алгоритми пошуку для визначення споріднених компонентів, наприклад пошук у глибину та пошук ушир.

Розділення зчеплених символів. Після виділення компонента підключення на діаграмі зображення вам потрібно вирішити, чи складається цей компонент з однієї чи кількох символів і, якщо більше, як їх розділити. Компонент зв'язності зазвичай сегментується на числа у два етапи: спочатку генеруються можливі типи сегментації компонента підключення, а потім вибирається найкращий. Побудова можливих методів сегментації зазвичай здійснюється за допомогою евристичних методів. Вибір найкращого методу сегментації може здійснюватися як підсистемою сегментації (у цьому випадку на вхід підсистеми виділення ознак і класифікації надходить лише один набір вхідних даних), так і підсистемою класифікації (у цьому випадку кілька наборів вхідних даних). дані надходять до системи виділення ознак та підсистеми класифікації, і рішення про те, яка сегментація була виконана правильно, приймається на основі результатів розпізнавання).

Для побудови явної сегментації часто використовуються два типи представлень даних: контур і скелет. Контур об'єкта — це набір усіх пікселів, які належать об'єкту і мають сусідній піксель, який не підходить для об'єкта. Точки в крайньому правому і лівому куті поділяють його на верхній і нижній

шляхи. Опукла область контуру називається долиною, а увігнута — пагорбом.

Методи сегментації, засновані на аналізі раніше визначених меж компонентів зв'язності, наведено в [23]. У цих методах на основі деяких міркувань щодо контурів компонентів зв'язності вибираються потенційні точки перетину, через які потім проводять лінії перетину, що відповідають певним умовам. У [15] можливі точки перетину встановлюються там, де відстань між верхнім і нижнім контурами перевищує порогове значення, а найкраще перетин вибирається як найкоротший шлях у графі з вершинами на можливих перетинах. У [15] точки можливого перетину вибираються евристично як точки, в яких профіль здійснює максимальний поворот за годинниковою стрілкою при русі проти годинникової стрілки.

Вилучення ознак. Нехай  $X$  — множина розглянутих об'єктів,  $Y$  — скінченна множина. Існує функція  $y: X \rightarrow Y$ , значення якої відоме лише на кінцевій вибірці  $X = \{(x_1, y_1), \dots, (x_m, y_m)\}$ . Необхідно побудувати функцію  $K: X \rightarrow Y$  для функції  $y$ . Функція  $K$  називається класифікатором.

Загалом систему розпізнавання рукописних даних можна представити у вигляді трьох підсистем: підсистеми вилучення ознак, підсистеми розпізнавання та підсистеми прийняття рішень, до якої належить цей об'єкт.

Вилучення ознак — це перетворення вхідних об'єктів в однорідну, компактну та зручну форму з втратою більшої частини інформації, що міститься в об'єкті, що мало впливає на класифікацію. Спосіб вилучення ознак залежить від типу та вихідного коду об'єктів і вибирається вручну.

Підсистема розпізнавання — це алгоритм, який розділяє простір ознак на частини, що відповідають заданим класам.

Розробка, тестування та налагодження системи введення тексту неможливі без створення бази даних символів. Шрифтова основа є однією з основних частин системи розпізнавання рукописного тексту. Представлення символів у цій базі даних залежить від того, як розпізнається рукописне зображення. Інтерпретація розрахованих ймовірностей відбувається за допомогою процесу прийняття рішень, який є окремим від розпізнавання.

Почерк обробляється в два етапи. На першому кроці вихідне зображення зводиться до універсальної векторної форми. Для цього необхідно виконати такі перетворення, які призводять до інваріантного представлення даних:

- переведіть зображення у відтінки сірого;
- зменшення чорно-білого зображення;
- аналіз і видалення шумів на картинці;
- виконання скелетизації чорно-білих зображень;
- векторизація зображення.

На цьому ж етапі вирішується задача сегментації рукописного тексту на вході на основі структурного підходу [3], який передбачає опис складних об'єктів за допомогою більш простих підоб'єктів.

Підзавданням є розробка техніки обробки зображень, а саме визначення сфери інтересу (роботи). Лінії діють як робоча область, яку потім потрібно розбити на слова. Через високий рівень узгодженості форм почерку багатьох людей розбивати слова на літери є виснажливим завданням.

Другим кроком є класифікація зображення. Завдання класифікації полягає в тому, щоб показати, чи належить вхідне зображення до одного із задалегідь визначених класів, або порівняти векторне представлення (опис) введеного рукописного символу з векторним представленням (описом) еталонного символу в базі даних.

## 2.2 Нейронні мережі

У машинному навчанні згортка нейронна мережа (CNN) є різновидом штучної прямолінійної нейронної мережі, в якій модель зв'язку її нейронів заснована на організації зорової кори тварин, окремих нейронів, влаштованих на основі область, яка охоплює поле зору, частково перекриваючись [16]. Згорткові нейронні мережі часто використовуються в розпізнаванні зображень і відео, в системах відліку [18] та в обробці природної мови.

У розпізнаванні зображень згорткові нейронні мережі (MNN) складаються з кількох шарів невеликих груп нейронів, які обробляють

частини вхідного зображення, так звані рецептивні поля. Вихідні дані цих збірок потім укладаються так, щоб їхні вхідні області перекривалися, щоб краще відображати вихідне зображення; це повторюється для кожного з цих шарів. Стекування з перекриттям дозволяє ZNM толерувати паралельні передачі вхідного зображення [18].

Згорткові мережі можуть включати локальні або глобальні шари підвибірки, які поєднують вихідні дані нейронних кластерів [19]. Вони також складаються з кількох комбінацій скручених і повністю з'єднаних шарів, використовуючи нелінійність точки в кінці кожного шару. Щоб зменшити кількість вільних параметрів і покращити узагальнення, в невеликих ділянках входу вводиться операція згортки.

Ідея згорткових нейронних мереж полягає в чергуванні згорткових шарів (C-шарів), шарів підвибірки (S-шарів) і наявності повністю зв'язаних шарів (F-шарів) на виході. Перша топологія згорткової нейронної мережі (LeNet) розроблена Яном Лекуном [18].

Архітектура групи (називається AlexNet) обговорювалася в груповій роботі. Вони використовували відносно просту планування в порівнянні з сучасними архітектурами. Мережа складалася з 5 згорткових шарів, шарів підвибірки та трьох повністю пов'язаних шарів [26].

Ця архітектура охоплює 3 основні парадигми: місцеве сприйняття, загальні підходи та підвибірки [26].

У локальному сприйнятті мається на увазі, що на вході нейрона є не все зображення (або вихід попереднього шару), а лише частина його області. Це дозволяє зберігати топологію зображення від одного рівня до іншого.

Підвибірка зменшує розмірність згенерованих карт об'єктів. У цій архітектурі мережі передбачається, що інформація про наявність потрібної ознаки важливіша за точне знання її координат, тому вибирається максимум із кількох сусідніх нейронів карти ознак і приймається за нейрон карти ознак зменшеного розміру. Завдяки цій операції мережа стає більш інваріантною порівняно з масштабом вхідного зображення, а також прискорює подальші обчислення.



Потім, повторюючи кілька шарів згортки та багаторазову підвибірку, будується згортка нейронна мережа. Як правило, після проходження кількох шарів карта об'єктів вироджується у вектор або навіть скаляр, але таких карт об'єктів є сотні. На виході мережі часто встановлюють кілька шарів повністю зв'язаної нейронної мережі (персептрон), на вхід якої подаються підсумкові карти характеристик.

Якщо двовимірне ядро згортки проходить по одноканальному зображенню на першому шарі, то на внутрішніх шарах, які складаються з багатьох карт об'єктів, береться тривимірний вимір ядра [розмір ядра  $e$  — це кількість карт функцій у поточному шарі, які (після проходження функції активації) утворюють карту ознак наступного шару, що відповідає цьому ядру згортки.

Процес навчання GPU значно збільшив кількість параметрів мережі. Отримана мережа отримала назву AlexNet. Із точки зору топології мережі, це майже те саме, що LeNet, лише збільшений у тисячу разів. Було додано набагато більше рівнів згортки, і розмір ядер згортки зменшується від входу в мережу до виходу. Це пов'язано з тим, що пікселі на початку сильно корелюють і область прийому, безумовно, може бути збільшена, але ми все одно втрачаємо мало інформації. Потім застосовується перетягування, збільшуючи щільність непов'язаних областей. На наступному рівні має сенс взяти трохи меншу площу рецептора. В результаті отримуємо таку піраміду із шарів  $11 \times 11 \rightarrow 5 \times 5 \rightarrow 3 \times 3$ .

Архітектура ZF Net дуже схожа на архітектуру AlexNet, за винятком кількох незначних змін. AlexNet був навчений на 15 мільйонах зображень, а ZF Net — лише на 1,3 мільйонах зображень. Замість фільтрів  $11 \times 11$  на першому рівні (точно так, як реалізовано в AlexNet), ZF Net використовує фільтри  $7 \times 7$  [25] та зменшені проходи. Причина цієї зміни полягає в тому, що менший розмір фільтра на першому рівні згортки допомагає зберігати велику кількість вихідної інформації про піксель у вхідному обсязі [26].

У конкурсі ILSVRC в 2014 році перемогу отримала згорткова нейронна мережа Google — GoogLeNet з показником помилок 6,7% [26].

GoogLeNet був однією з перших архітектур CNN, яка дійсно відхилилася від загального підходу простого поєднання шарів згортки та підвибірки в послідовну структуру. Автори статті [27] також підкреслили, що ця нова модель приділяє велику увагу використанню пам'яті та енергії.

Щоб уникнути перекваліфікації, були використані інші методи, деякі з яких тепер є стандартними для глибоких мереж: DropOut (RIP), Data Augmentation і ReLu [27].

Rethinking the Inception Architecture for Computer Vision — це топологія, яка з'явилася у 2015 році. У своїй статті [27] автори розглянули різні архітектури на практиці та розробили чотири принципи побудови глибоких нейронних мереж для комп'ютерного зору.

Уникайте вузьких місць у представленні: не знижуйте різко розмірність візуалізації даних, це потрібно робити плавно від початку мережі до класифікатора і виходу. Високорозмірні уявлення необхідно обробляти локально для збільшення розмірності: недостатньо поступово зменшувати розмірність, необхідно використовувати принципи аналізу та групування суміжних областей. Територіальні угоди можна і потрібно розбити на ще менші фактори: це економить ресурси та призводить до збільшення розміру мережі. Необхідно підтримувати баланс між глибиною і шириною сітки: не збільшуйте глибину сітки окремо від ширини і навпаки; обидва розміри повинні бути збільшені або зменшені однаково.

Давно відомо, що якість такої моделі зростає до певної межі, а потім знижується, якщо просто поставити більше рівнів. Ця задача називається проблемою деградації та мереж, отриманих шляхом складання великої кількості шарів: простих або плоских мереж. Дослідники з китайського підрозділу Microsoft змогли знайти топологію, в якій якість моделі підвищується за рахунок додавання нових шарів. Він отримав назву ResNet[18].

Крім класичних згорткових нейронних мереж, існують також регіональні згорткові нейронні мережі (Region Based CNN [27]). Поява R-CNN була більш вражаючою, ніж будь-яка попередня стаття про нові

архітектури ZNM. Оскільки перша стаття R-CNN була процитована понад 1600 разів, Росс Хіршік та його команда з Університету Берклі зробили одне з найбільш вражаючих досягнень у сфері комп'ютерного зору. Як впливає з назви, Fast R-CNN і Faster R-CNN [27] попрацювали над тим, щоб зробити модель швидшою та більш придатною для сучасних завдань розпізнавання об'єктів.

Метою R-CNN є вирішення проблеми виявлення об'єктів. Для конкретного зображення ми хочемо мати можливість малювати обмежувальні рамки на всіх об'єктах. Процес можна розділити на дві основні складові: рівень опіки регіону та фазу класифікації.

Зокрема, для R-CNN використовується вибіркового пошуку. Вибірковий пошук виконує функцію генерування 2000 різних областей, які найімовірніше містять об'єкт. Після того, як набір пропозицій регіонів був зібраний разом, ці пропозиції потім «обрізаються» до розміру зображення, яке можна надіслати на вхід CNN (у цьому випадку AlexNet), який витягує векторний елемент для кожного регіону. Цей вектор потім використовується як набір лінійних вхідних даних SVM, які навчаються для кожного класу, що виводить класифікацію [27].

Згорткові нейронні мережі використовують просторово-локальну кореляцію, яка забезпечує локальну модель зв'язку між нейронами сусідніх шарів. Іншими словами, входи блоків, прихованих у шарі  $m$ , беруться з підмножини блоків у шарі  $m-1$  із просторово суміжними рецептивними полями [26]. Графічно це показано на рисунку 2.1.

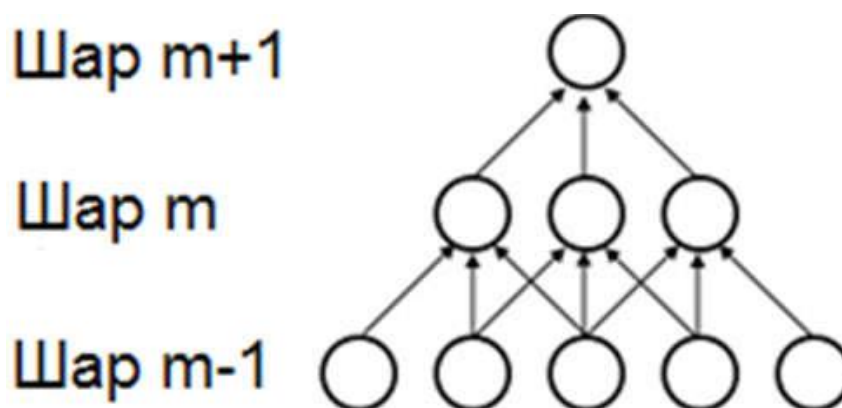


Рисунок 2.1 — Зв'язність між нейронами суміжних шарів

Зріз  $m-1$  є аналогією входу в сітківку. На рисунку (2.1) вище елементи  $m$ -шару мають 3 широкі рецептивні поля у вхідному шарі, і, отже, лише 3 з'єднані з сусідніми нейронами наступного шару.

Нейрони шару  $m + 1$  мають подібну здатність з'єднуватися з нейронами наступного шару. Його сприйнятливим полем щодо наступного шару також називається 3, але його сприйнятливим полем щодо входу дорівнює 5 [27]. Кожен блок не реагує на зміни за межами свого рецепторного поля по відношенню до сітківки.

Таким чином, архітектура гарантує, що вилучені «фільтри» виробляють найсильнішу реакцію на просторову та локальну природу вхідного сигналу.

Проте структура багатьох із цих шарів призводить, як показано вище, до нелінійних «фільтрів», які стають все більш «глобальними» (тобто реагують на велику площу пікселів). Наприклад, нейрон у прихованому шарі  $m + 1$  може кодувати нелінійну ознаку шириною 5 (з точки зору простору пікселів).

У згорткових нейронних мережах кожен високочастотний фільтр реплікується по всьому полю зору. Ці репліковані блоки мають однакову параметризацію (вектор ваги та переміщення) і утворюють карту об'єктів.

Однією з основних переваг мереж згортки є використання спільної ваги в шарах згортки, що означає, що для кожного пікселя шару використовується один і той же фільтр (ваговий вектор); це зменшує як необхідні вимоги до пам'яті, так і продуктивність [19]. Вага кожного кольору є загальною. Така реплікація блоків дозволяє ідентифікувати ключові характеристики незалежно від їх положення в полі зору. Крім того, загальні шкали підвищують ефективність за рахунок зменшення мережевих змінних. Ці обмеження моделі дозволяють краще узагальнити діяльність комп'ютерного зору.

Карта ознак отримується шляхом багаторазового застосування функції згортки до підобластей усього зображення. Іншими словами, я згортаю вхідне зображення з лінійним фільтром, додаючи зміщення, а потім застосовую нелінійну функцію.

У порівнянні з іншими алгоритмами класифікації зображень, згорткові нейронні мережі використовують відносно мало попередньої обробки. Це означає, що мережа відповідає за навчання розроблених вручну фільтрів традиційним алгоритмам. Незалежність у розвитку функціональних можливостей від апріорних знань і людських зусиль є великою перевагою мережі ZNM.

Перші два типи шарів (згортка, підвибірка), чергуючись один з одним, утворюють вхідний вектор властивостей для багат шарового персептрона.

Неофіційно цю операцію можна описати наступним чином: розмір вікна ядра проходить із заданим кроком (зазвичай 1) усі зображення зазвичай, на кожному кроці, елемент за елементом, множить вміст вікна з ядром, результат підсумовується і записується в матрицю результатів [20].

При цьому, залежно від способу обробки країв вихідної матриці, результат може бути меншим за вихідне (дійсне) зображення, такого ж розміру (рівним) або більшим (ціле).

Згортковий шар реалізує ідею так званих локальних рецептивних полів, тобто кожен вихідний нейрон пов'язаний лише з певною (невеликою) ділянкою вхідної матриці і тому моделює деякі характеристики зору людини [20]. У спрощеній формі цей рівень можна описати такою формулою:

Варто також зазначити, що хоча рівень згортки зменшує кількість параметрів порівняно з повнозв'язаним рівнем, він використовує більше гіперпараметрів — параметрів, які вибираються до навчання [27]. На рисунку 2.2 показано схематичне зображення принципу роботи складного шару.

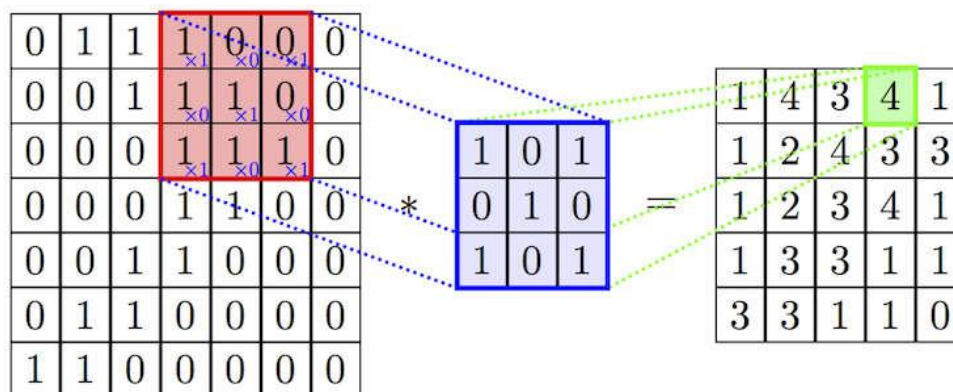


Рисунок 2.2 — Принцип роботи складного шару

Зокрема, вибираються такі гіперпараметри: глибина, заповнення та рівні субдискретизації.

Глибина — скільки ядер і коефіцієнтів зсуву задіяно шар; висота і ширина кожного сердечника; крок (зміщення) — на скільки переміщується ядро на кожному кроці при обчисленні наступного пікселя отриманого зображення. Зазвичай він вважається рівним 1, і чим вище його значення, тим менше вихідний розмір зображення;

Заповнення: зверніть увагу, що згинання з ядром розміром більше 1x1 зменшить розмір вихідного зображення. Оскільки загалом бажано зберегти розмір вихідного зображення, малюнок доповнюється нулями по краях.

Рівні субдискретизації в згорткових нейронних мережах зменшують розмір вхідної карти ознак (зазвичай у 2 рази) [20]. Це можна зробити декількома способами, в даному випадку ми розглядаємо метод вибору максимального елемента (max pooling) — вся карта розбивається на клітинки 2x2, з яких вибирається максимальне значення. Тому під час підвибірки зображення значно зменшується, але його зміст не змінюється або змінюється незначно.

Схематичне зображення процесу підвибірки показано на рисунку 2.3.

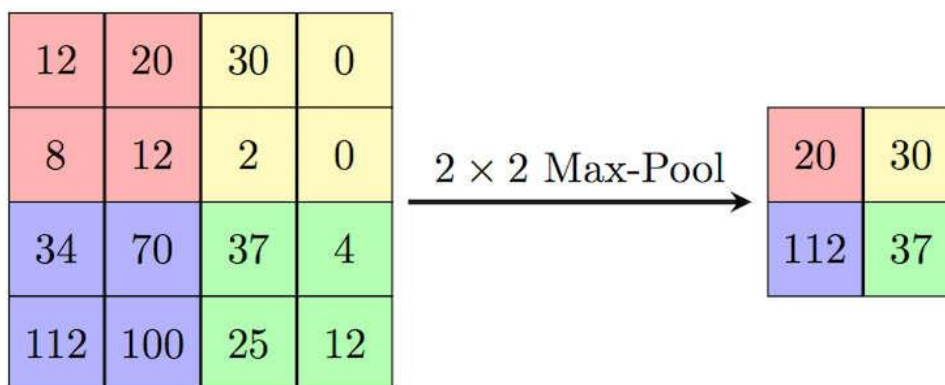


Рисунок 2.3 – Процес підвибірки

І останнє, але не менш важливе, шари в нейронних мережах із оболонкою є повністю зв'язаними шарами. Вхідні нейрони цього шару підключені до всіх активацій попереднього шару. Цей рівень обчислює

рейтинг класу та повертає вектор розмірності (кількість класів). Отже, повністю зв'язаний шар є багатошаровим перцептронним шаблоном (MLP).

Потім, після аналізу вихідного повністю зв'язаного вектора шару, результатом згорткової нейронної мережі буде клас, який набере найвищий бал у цьому векторі.

Типову архітектуру CNN для класифікації зображень можна розділити на дві окремі частини: ланцюжок повторюваних рівнів Conv  $\rightarrow$  Pool (іноді з кількома рівнями згортки одночасно), за якими слідує кілька повністю пов'язаних рівнів (кожен піксель зображення, розрахований як незалежний вхід), кінцем якого є модель Soft Max, оптимізована для перехресної ентропії [26].

Як правило, після кожного згорткового або повністю зв'язаного шару функція активації (наприклад, ReLU) застосовується до всіх виходів [26].

### 2.3 Використання бібліотек із відкритим кодом

Бібліотечна система Open Computer Vision, спрощена як OpenCV, означає відкриту бібліотеку комп'ютерного зору, яку необхідно відтворювати в реальному часі [29]. OpenCV можна використовувати для вирішення широкого спектру проблем взаємодії “людина-комп'ютер”; спостерігати за рухами, відстежувати, структурувати та розуміти рухи камери; для розпізнавання об'єктів, сегментації та ідентифікації, включаючи розпізнавання обличчя та жестів. Одним з найбільш перспективних напрямків використання бібліотеки OpenCV є робототехніка, зокрема мобільна робота.

Основними модулями бібліотеки є такі модулі. Ядро `sxcore` виконує такі операції:

- підтримка матричної алгебри, генераторів випадкових чисел DFT, DST, математичних функцій;
- основні операції над багатовимірними числовими масивами;
- підтримка основних функцій 2D-графіки;

— підтримка більш складних структур даних: динамічно зростаючі послідовності, графіки, розріджені масиви;

Запис та відновлення структур даних у / з форматів XML / YAML.

Інший модуль CV — це модуль обробки зображень і операцій комп'ютерного зору. Його основні функції:

— виконувати базові операції із зображеннями (геометричні перетворення, перетворення колірного простору, фільтрація тощо).

— розрахунок конструкції (плоські перегородки, опис форм);

— аналіз зображень (пошук контурів, гістограм, виділення відмітних ознак, морфології);

— відстеження об'єктів, аналіз руху, ;

— поділ предметів, особливо людей;

— калібрування камер;

HighGUI — це модуль для введення та виведення зображень і відео, а також для створення інтерфейсу користувача. Необхідно зробити такі операції:

— записувати відео з камер та відеофайли, читати та записувати нерухомі зображення;

— функції з організації зрозумілого і простого інтерфейсу користувача (в даний час більшість додатків використовують HighGUI).

— Модуль Svaux вміщує в себе експериментальні та деякі старі функції.

За допомогою цього модуля виконується наступні операції:

— просторовий зір: стерео калібрування, самокалібрування;

— пошук та опис рис особи людини;

— пошук збігу стерео, клацань за графіком;

— приховані ланцюги Маркова;

— опис текстури;

— порівняння форми та конструкції скелетів.



З точки зору програмного забезпечення бібліотека OpenCV є набором недорогих, але потужних операцій для обробки пікселів у зображеннях. Реалізує алгоритми калібрування камери; Знак відстеження; впізнавання та формування персонажів; тривимірна реконструкція об'єктів; їх сегментація та розпізнавання; аналіз форм; аналіз руху.

Для реалізації своїх властивостей бібліотека працює з набором типів даних, включаючи дані типу масиву: CvMat (матриця), IplImage (зображення), CvSet (набір), CvGraph (графік), CvSeq (послідовність); змішані типи, такі як CvHistogram; Допоміжні типи даних: CvPoint (2D точка), CvTermCriteria (критерій завершення ітераційного процесу), CvSize (розміри); CvMoments (просторові моменти); IplConvKernel (ядро коагуляції)[39].

Бібліотека OpenCV в основному використовується в операційній системі MS Windows, хоча є також реалізацію бібліотеки для інших операційних систем, включаючи Linux, iOS та інші.

Інша бібліотека LTI або LTI lib — це об'єктно-орієнтована бібліотека алгоритмів та структур даних [30]. Вона широко використовується для обробки зображень та комп'ютерного зору. Її основні сфери застосування— комп'ютерний зір для робототехніки, розпізнавання об'єктів, голос і жести. Основна мета цієї бібліотеки — створити об'єктно-орієнтовану бібліотеку на C++, щоб спростити використання програмного коду та надати швидкі алгоритми, які можна використовувати у реальних додатках.

Існує також бібліотека AForge.NET з відкритим вихідним кодом, яка заснована на мові програмування C# і призначена для комп'ютерного зору. Спектр роботи цієї бібліотек широкий: обробка зображень, нейронні мережі, машинне навчання, нечітка логіка та багато іншого [31]. Бібліотека складається з низки компонентів, які допомагають працювати у певному напрямку, наприклад: V. AForge.Neuro - бібліотека для роботи та виконання дій та різних операцій з нейронними мережами. Однак основним недоліком цієї бібліотеки є її обмежений об'єм, пов'язаний з представленням її

документації англійською мовою, який не широко використовується при її використанні.

Ще одна бібліотека VXL — це серія бібліотек, написаних на C++, які реалізують технологію комп'ютерного зору та наукові функції дослідження[32]. Бібліотека написана на ANSI/ISO C++ і призначена для портативних платформ. Бібліотека складається з наступних компонентів: VNL — робота з чисельними алгоритмами та контейнерами; BIL — робота з картинками; VGL — геометрія точок, кривих та інших елементарних об'єктів у двовимірному та тривимірному просторах та ін. Але набору модулів у цій бібліотеці недостатньо для вирішення наших завдань.

Підсумовуючи, найбільш оптимальною з розглянутих бібліотек буде використання бібліотеки OpenCV, так як її швидкість досить висока і включає велику кількість функцій, які можна використовувати для розпізнавання тексту та обробки зображень всього документа. Тому, щоб створити програмний продукт для розпізнавання текстових документів, ми концентруємось на використанні бібліотеки.

У цьому розділі магістерської роботи запропонована послідовність розпізнавання рукописних символів документів, визначена нейронна мережа для виділення рукописних символів у отриманому тексті та їх розпізнавання.

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

У цьому розділі розглянуто послідовність виконання розпізнавання окремо взятих та виділених символів у сканованих рукописних документах, яка здійснюється за ряд етапів та їхню програмну реалізацію.

### **3.1 Вибір інструментальних засобів програмування**

Для реалізації розпізнавання рукописних символів наведеними вище підходами було обрано мову програмування Python. Вибір цієї мови розробки для створення даного програмного продукту обумовлений тим, що це дуже зручна і ідеально функціональна мова для реалізації складних математичних операцій. Python — це загальна мова програмування високого рівня, яка зосереджена на покращенні продуктивності розробника та якою дуже зручно писати й читати код [33]. Синтаксис ядра Python мінімалістичний. При цьому, стандартна бібліотека містить велику кількість корисних функцій. Python підтримує структуроване, об'єктно-орієнтоване, обов'язкове, аспектно-орієнтоване та функціональне програмування. Основними характеристиками архітектури є повний самоаналіз, динамічне введення, механізм обробки винятків, автоматичне управління пам'яттю, високорівневі структури даних, підтримка багатопоточної обробки. Мова підтримує поділ програми на окремі модулі, які в свою чергу можуть бути об'єднані в пакети.

Інтерпретатор CPython є найкращою реалізацією Python, яка підтримує більшість активних платформ. Він розповсюджується безкоштовно за ліцензією Python Software Foundation, що дозволяє без обмежень використовувати його в будь-якому додатку, в тому числі в приватному. Python є портативним і працює майже на всіх відомих платформах, від мейнфреймів до КПК. Є версії для Microsoft Windows, майже всіх версій UNIX-подібних систем (включаючи FreeBSD і Linux), Windows Mobile, Symbian і Android.

Проте, на відміну від багатьох інших портованих систем, Python підтримує специфічні для платформи технології (такі як Microsoft COM/DCOM). Існує також Jython, спеціальна версія Python для віртуальної машини Java, яка дозволяє інтерпретатору працювати в будь-якій системі, яка підтримує Java. Однак класи Java можна використовувати безпосередньо з Python і навіть писати на Python. Код компілюється та виконується на машині Java (JVM), що дозволяє використовувати код Python із програм Java. Розроблений прототип системи розпізнавання тексту зі спотвореними символами має продемонструвати функціональність алгоритму розпізнавання вмісту пошкоджених текстів. Програмне забезпечення реалізовано на вищих мовах програмування Java та Python.

Наступними основними перевагами мови програмування Python є такі сторони. Це добре розроблена мова програмування, яка включає в себе сучасні тенденції програмування з нуля. Крім того, мова програмування Python динамічно розвивається, процес додавання нових конструкцій до мови добре налагоджений і продовжує охоплювати методи аспектно-орієнтованого програмування, функціонального програмування тощо, залишаючись при цьому внутрішньо сумісним і послідовним. Крім того, мову Python легко програмувати.

Мова Python використовує простий для читання синтаксис (порівняно із мовами C++, PHP) й дозволяє вам легко читати чужий код і вільно розуміти ваш давно написаний код. Численні бібліотеки з кодом на будь-який випадок (наприклад, робота з малюнками, електронними таблицями Excel або Twitter).

Python дуже портативний і може працювати на всіх поширених операційних системах і багатьох архітектурах: на міні-комп'ютерах Windows, MacOS, Linux і Arduino. Система залежностей досить добре продумана, і розгортання програм в іншій системі є простим й зрозумілим.

Програми Python дуже схожі на програми Java і можуть вільно взаємодіяти з Java-кодом. Програми Java зазвичай перекладаються в

спеціальний байт-код, тому вони запускаються на будь-якій архітектурі комп'ютера за допомогою віртуальної машини Java.

Перевагою цього методу виконання програми є абсолютна незалежність байт-коду від операційної системи та обладнання, а це означає, що ви можете виконувати Java-код на будь-якій системі, для якої існує віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки, в якій виконання програми повністю контролюється віртуальною машиною. Якщо будь-які операції, які перевищують встановлені дозволи програми (наприклад, підключення до іншого комп'ютера або спроба несанкціонованого доступу до даних), негайно зупинить програму. Одним із недоліків концепції віртуальної машини є значне зниження продуктивності системи.

Численні вдосконалення цієї мови трохи підвищили продуктивність програм Java. Використання технології трансляції байт-коду в машинному коді безпосередньо під час виконання програми (технологія JIT) з можливістю збереження версій класів у машинному коді. Широке використання платформно-орієнтованого коду (родного коду) у стандартних бібліотеках. Використання апаратного забезпечення, яке пропонує прискорену обробку байт-коду [34].

Python має потужні та однорідні концепції абстракції для типів даних і значень. Крім того, мова має гнучкі симетричні конструкції домішок для складання типів і класів. Таким чином ви можете розкласти об'єкти, порівнявши їх із зразком; Приклади та вирази були узагальнені для підтримки природної обробки XML-документів. Загалом, ці конструкції дозволяють легко створювати окремі компоненти, які використовують бібліотеки Python без використання спеціальних мовних конструкцій.

Крім того, ця система використовує бібліотеку OpenCV, яка містить алгоритми машинного навчання, як допомогу в попередній обробці отриманих зображень текстових документів.

### 3.2 Розробка архітектури програми розпізнавання

Структура програми розпізнавання рукописного введення складатиметься з трьох основних модулів та одного додаткового модуля: модуля попередньої обробки зображень, модуля виділення символів для розпізнавання у нейронній мережі та модуля розпізнавання символів з використанням нейронної мережі. Крім того, буде ще один окремий четвертий модуль для налаштування нейронної мережі для виконання процесу розпізнавання символів.

У модулі попередньої обробки зображень виконуються наступні функції. Ми отримуємо зображення рукописного тексту, фільтруємо його від можливих перешкод та використовуємо морфологічні фільтри, переводимо зображення з отриманого кольору у двійковий формат та по черзі виділяємо області, що відповідають сторінкам текстового документа. Далі у другому модулі розділимо виділену сторінку тексту на окремі рядки за допомогою гістограм вертикального розподілу яскравості зображення. Потім починаємо розбивати рядки на окремі слова за допомогою гістограм розподілу яскравості виділеного рядка по горизонталі. Далі ми поділяємо виділені слова на окремі символи та визначаємо їхнє положення. Після розподілу слів на окремі символи ми нормалізуємо їх до необхідного розміру для подальшої реалізації процесу розпізнавання за допомогою нейронної мережі. Процес розпізнавання символів відбувається під час підключення нейронної мережі глибокого навчання. Якщо символ не був розпізнаний в одному місці через значне відхилення при написанні, ми переходимо до наступного етапу роботи програми: етапу розпізнавання, який використовує розпізнавання символу із застосуванням структурного методу.

На даному етапі використовуємо нормалізовані попередньо символи рукописного тексту. Їх розділяємо на зони розміром 4x4, у межах яких по зоно описуємо лінії. У межах кожного фрагменту описуємо тип лінії, який знаходиться у цій зоні. Це може бути пряма, вигнута лінія або пересічення двох і більше ліній. Набір таких структурованих ознак дозволить описані літери алфавіту.

Тепер виконуємо послідовний пошук та співставлення описаного зображення літери із структурним описом шаблону із вибраної групи символів. Якщо є збіг по перекриттю ліній у вибраній зоні аналізу, то розраховується коефіцієнт узгодженості для даної зони аналізованого символу. Коефіцієнт узгодженості визначається логічним множенням аналізованої зони символу на зону опорного символу виділеної групи символів. Таким чином визначаємо коефіцієнт відповідності між отриманим символом та символами набору. Порівняння відмінностей цих коефіцієнтів дозволяє вибрати із опорних символів символ, що має найбільше значення з отриманої групи та найбільш схожий на аналізований символ.

Модуль використовує базу даних еталонних символів MNIST [35], яка складається з різних символів, щоб настроїти нейронну мережу таким чином, щоб вона виконувала процес розпізнавання символів.

Пропонована послідовність розпізнавання символів наведена у Додатку Д до магістерської роботи.

### 3.3 Розробка програми розпізнавання рукописних символів

Програма для розпізнавання рукописних символів виконує ряд послідовних дій, які дозволяють у кінцевому підсумку виділити та розпізнати конкретний символ. Робота програми починається із отримання відсканованого документу рукописного тексту. Наступним є етапи розподілу тексту на рядки, слова та виділення у них окремих букв. Розділення отриманого тексту на окремі букви є одним із важливих кроків роботи програми по виконанню операції розпізнавання символів. Для цього використаємо можливості відкритої бібліотеки OpenCV та виберемо одну із її функцій для виділення контурів `findContours`. Ця функція дозволить провести операції по сегментації рядків, слів та виділення окремих символів.

Перед початком роботи програми по розпізнаванню символів необхідно налаштувати нейронну мережу. Для розпізнавання виділених символів використовуємо нейронну мережу. Також скористаємося базою даних символів MNIST, яка у своєму складі має набір різних символів.

Для роботи нейронної мережі необхідно спочатку імпортувати усі модулі, які будуть необхідні для початкового тренування нашої моделі. Скористаємося також бібліотекою Keras, яка у своєму складі вже має певну кількість підготовлених модулів. Серед цих модулів є також і база MNIST. Таким чином імпортуємо необхідні для роботи модулі та будемо використовувати їх у нашому процесі.

Перед початком роботи слід задіяти модуль навчання створеної нейронної мережі.

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
```

Він функціонує таким чином. На вході нейронна мережа оперує фрагментом виділеного зображення символу розміром 28x28 біт, а на виході має відповідно 64 виходи, Якщо символ буде розпізнаний, то на одному із виходів формується значення логічної одиниці, що констатує факт знаходження та розпізнавання відповідного символу. Також створюємо модель загорткової нейронної мережі.

```
batch_size = 128
epochs = 10
model = Sequential()
model.add(Conv2D(32, kernel_size=(3,
3),activation='relu',input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
```

Використовуємо класичну згорткову нейронну мережу, яка виділяє та аналізує певні ознаки виділеного символу текстового зображення,



використовуючи для цього фільтри розміром 32 й 62. До виходу цієї мережі підключена звичайна лінійна мережа MLP, на яку і покладено завдання формувати остаточний результат розпізнавання. Дані, які використовуються для навчання мережі, необхідно спочатку певним чином переформатувати. Для цього тренувальні дані переводимо у формат (60000, 28, 28, 1) та використовуємо їх для підготовки мережі для навчання.

```
num_classes = 10
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
input_shape = (28, 28, 1)
```

Наступний етап, це створення моделі загорткової нейронної мережі. Така мережа складається із згорткових шарів та шарів попередньої вибірки. Така мережа краще функціонує із даними, що подані у вигляді матриці. Із-за цієї причини такого типу мережі використовуються для обробки зображень. Ще один шар використовується для вилучення окремих нейронів під час навчання, щоб зменшити ймовірність перенавчання.

```
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss=keras.losses.categorical_crossentropy, optimizer=keras.
optimizers.Adadelta(), metrics=['accuracy'])
```

Тепер переходимо до етапу навчання мережі. Для нашого випадку скористаємося базою даних для навчання нейронної мережі MNIST. Для виконання операції читання даних із цієї бази та її подальшої роботи скористаємося можливостями бібліотеки `idx2numpy`, й підготуємо необхідні дані для навчання й подальшої перевірки їх нейронною мережею.

```
import idx2 numpy
emnist_path = '/home/Documents/ TestApps/keras/emnist/'
X_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
train- images- idx3- ubyte')
```

Для навчання нашої мережі використовуємо підготовлені набори даних, варіанти подання символів у яких представлені як звичайні масиви.

Для подальшого виконання операції розпізнавання символів завантажуюмо підготовлену модель мережі та визиваємо функцію `predict_classes`. Завершальна функція по налаштування мережі матиме такий вигляд:

```
def img_to_str (model: Any, image_file: str):
    letters = letters_extract (image_file)
    return s_out
```

Тепер приступаємо до процесу оброблення текстового документа. Для виділення тексту із отриманого відсканованого зображення у форматі сканера `png` та виконуємо його відкривання із використанням однієї із стандартних процедур (`cv2.imread`), що є у бібліотеці `OpenCV` [29]. Наступним кроком є переведення його у градації сірого кольору й виконання процесу бінаризації зображення із використанням тепер уже процедури `cv2.cvtColor` у поширеній моделі кольорів `RGB`, потім змінюємо розміри процедурою (`cv2.erode`) й виконуємо формування контурів рукописних символів за допомогою процедури `cv2.threshold`.

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

Здійснюємо операцію послідовного виділення контурів на отриманій сторінці, спочатку рядків, потім слів та переходимо до окремих виділених символів.

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

Виконуємо процедуру формування ієрархії виділених контурів символів по їх складності.

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

У результаті виконаних дій отримуємо ієрархічне дерево контурів із застосуванням параметру `cv2.RETR_TREE`. Першим йде контур загальної картини отриманого зображення текстової сторінки, потім формуються контури слів і завершуємо контуром окремих символів. Таким чином фактично виконуємо сегментацію текстової сторінки до рівня окремо взятого символу. У нашого випадку процес розпізнавання необхідно зупинити на контурах окремих символів, тому й зупиняємося на цьому рівні представлення текстового документа.

Наступним кроком у обробці текстового документа буде збереження кожної із виділених букв у масштабі розміром 28x28 біт, так як база еталонних символів MNIST [35] працює у такому масштабі, а також нейронна мережа для розпізнавання рукописних символів. Додатково розставляємо координати виділених рукописних символів по осі X, щоб із виділених координат надалі сформувати пропуски між цими символами.

```
def letters_ extract(image_ file: str, out_ size= 28) -> List [Any]:
```

Для цього змінюємо розмір отриманого фрагменту зображення символу до квадратного розміру.

```
size_ max = max (w, h)
letter_ square = 255 * np.ones (shape= [size_ max, size_ max], dtype=
np.uint8)
if w >h:
    Форматуємо розмір зображення символу при перегляді зверху вниз.
    y_ pos = size_ max//2 - h //2
```

```
letter_square[y_pos:y_pos + h, 0:w] = letter_crop
elif w < h:
```

Також форматуємо розмір зображення символу при перегляді зліва на право у виділеному рядку.

```
x_pos = size_max//2 - w //2
```

```
letter_square[0:h, x_pos:x_pos + w] = letter_crop
```

```
else:
```

```
letter_square = letter_crop
```

```
letters.append ((x, w, cv2.resize (letter_square, (out_size, out_size),
interpolation= cv2.INTER_AREA)))
```

Завершуємо сортування отриманого масиву символів по координаті X.

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

```
return letters
```

По завершенню операції сортування символів по координаті X зберігаємо їх у вигляді функції tuple (x, w, letter), щоб по завершенню розпізнавання із сформованих відстаней виділити пропуски для кожного із символів та перевіряємо правильність роботи.

У підсумку отримуємо матрицю опису розпізнаних символів. Лістинг програми по розпізнаванню символів наведений у додатку Б магістерської роботи.

### 3.4 Перевірка якості програми розпізнавання

Для перевірки якості роботи програми розпізнавання символів використаємо підготовлений набір даних. Він включатиме літери алфавіту, набір цифр та розділові знаки. Також для оцінювання роботи скористаємося категоріями як точність розпізнавання та надійність виявлення символів. Такі категорії, як точність та надійність, використовуються для оцінки якості систем та алгоритмів розпізнавання тексту. Точність системи розпізнавання тексту показує можливість правильного розпізнавання вибраного об'єкта. Проте справжнє значення ймовірності розпізнавання у випадку досить складно визначити, й у разі точність розпізнавання визначається

апостеріорної ймовірності правильного розпізнавання текстових символів для набору введених текстових даних. При такому підході завдання тестового розпізнавання може бути сформульована як завдання максимізації апостеріорної ймовірності правильного розпізнавання символів вибраного набору текстових даних.

Ще один показник якості системи розпізнавання — надійність чи надійність розпізнавання. Оцінка надійності системи виявлення відбиває її здатність апіорі оцінити рівень точності своїх результатів.

Для оцінки якості розпізнавання використовуємо коефіцієнт помилок у символах  $K_s$ , який визначається так:

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

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

$c$  — кількість символів в еталоні [36].

Відповідно точність розпізнавання  $T_s$  у символах визначається як

$$T_s = 1 - K_s. \quad (3.2)$$

З іншого боку, коефіцієнт помилок можна обчислити, визначивши мінімальну кількість операцій, які мають бути виконані, щоб привести результуючий набір символів еталонний набір. Це число називається відстанню Левенштейна. Якщо в отриманій послідовності символів потрібна перестановка деяких символів, щоб привести її до еталонного значення, то загальна кількість усіх операцій згідно з Дамерау-Левенштейном [36] називається відстанню. У разі ми використовуємо вираз (3.2), яке визначає точність розпізнавання в символах.

Для створення навчального прикладу використовувалися шрифти з бібліотеки. Вибірка складалася з букв кирилиці, розділових знаків і цифр. Усього було використано 640 знаків.

У результаті 640 використаних символів було розпізнано 607 (94,8%) символів. Не було розпізнано лише 33 символічних зображень, до яких потім був застосований метод розпізнавання структур. Деякі символи були визначені неправильно через помилку визначення розміру символу. Це

можна пояснити використанням нормалізації зменшення символів до стандартного розміру, що робить символи схожими. Детальний аналіз неправильно розпізнаних символів показує, що більшість із них лягає на шрифт із максимально можливою чутливістю до регістру. Це можна пояснити тим, що символи в цьому шрифті явно відрізняються формою від символів в інших шрифтах, які використовувалися для формування нейронної мережі.

Найбільше спотворень показало, що розмір знаків показав рукописний текст, набраний шрифтом під значним кутом, найменше спотворення показав шрифт, максимально наближений до друкованого шрифту. Це призводить до висновку, що розпізнавання тексту у шрифті, введеному у дослідження вибірка буде найточнішою. Інші шрифти розпізнаються, якщо їх символи трохи відрізняються від шрифтів, призначених вихідним шаблоном нейронної мережі.

Щоб розпізнати спотворені рукописні символи, ми використовуємо підготовлений набір символів, в якому кількість спотворень різна і визначається у відсотках на основі виразів (3.1) та (3.2) щодо правильного представлення символів. Результати такого експерименту показані малюнку. Як показано, якщо перекручування перевищує 30%, розпізнавання символів стає практично неможливим.

Проведені експериментальні дослідження показують, що розроблений програмний продукт для розпізнавання рукописних символів з певними спотвореннями може бути використаний різних типах систем перекладу текстових документів в електронний вигляд.

У цьому розділі роботи була вибрана мова та інструментальні засоби програмування, сформована структура програмного продукту та розроблена програма по розпізнаванню рукописних символів, приведені результати експериментальної перевірки реалізованого підходу по розпізнаванню рукописних символів.

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

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### **4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» є оцінювання науково-технічного

рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [37].

Таблиця 4.1 — Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	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 працездатність 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 аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
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					



## Завершення таблиці 4.1

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

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 — Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	2	2	2
3. Ринкові переваги (ціна продукту)	2	2	2

## Завершення таблиці 4.2

4. Ринкові переваги (технічні властивості)	2	1	1
5. Ринкові переваги (експлуатаційні витрати)	1	1	1
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	1	1	1
8. Практична здійсненність (наявність фахівців)	3	3	3
9. Практична здійсненність (наявність фінансів)	3	4	4
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	34	34	33
Середньоарифметична сума балів $СБ_c$	33,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [37].

Таблиця 4.3 — Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» становить 33,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

Запропоновано процес розпізнавання рукописних символів який розділений на два етапи: на першому етапі використовується нейронна мережа, а для символів, які сильно спотворені та не були розпізнані на

першому етапі, застосувати шаблонний метод розпізнавання із аналізом силуету символу та фону, на якому він знаходиться. Це збільшує достовірність розпізнавання. Для реалізації розпізнавання рукописних символів наведеними вище підходами було обрано мову програмування Python. Вибір цієї мови розробки для створення даного програмного продукту обумовлений тим, що це дуже зручна і ідеально функціональна мова для реалізації складних математичних операцій. Python — це загальна мова програмування високого рівня, яка зосереджена на покращенні продуктивності розробника та якою дуже зручно писати й читати код. Синтаксис ядра Python мінімалістичний. При цьому, стандартна бібліотека містить велику кількість корисних функцій. Python підтримує структуроване, об'єктно-орієнтоване, обов'язкове, аспектно-орієнтоване та функціональне програмування. Основними характеристиками архітектури є повний самоаналіз, динамічне введення, механізм обробки винятків, автоматичне управління пам'яттю, високорівневі структури даних, підтримка багатопоточної обробки. Мова підтримує поділ програми на окремі модулі, які в свою чергу можуть бути об'єднані в пакети.

#### 4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою (4.1) [37]

$$q_i = \frac{P_i}{P_{\text{базі}}} \quad (4.1)$$

де  $q_i$  — одиничний параметричний індекс, розрахований за  $i$ -м параметром;

$P_i$  — значення  $i$ -го параметра виробу;

$P_{\text{базі}}$  — аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 — Основні техніко-економічні показники аналога та розробки, що проектується

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Точність розпізнавання	%	86	93,8	1,1	0,3
Кількість генерованих областей пошуку	-	1650	2000	1,21	0,4
Кількість шарів мережі	-	3	5	1,67	0,05
Рівень ущільнення	-	3	4	1,33	0,07
Можливість оперування набором типів даних	-	9	12	1,33	0,18
Експлуатаційні витрати	грн	21	17	0,81	0,45
Ціна	грн	2560	2120	0,83	0,55

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 — пристрій відповідає нормам і стандартам; 0 — не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою (4.2) [37]:

$$I_{\text{НП}} = \prod_{i=1}^n q_i, \quad (4.2)$$

де  $I_{\text{НП}}$  — загальний показник конкурентоспроможності за нормативними параметрами;

$q_i$  — одиничний (частинний) показник за  $i$ -м нормативним параметром;

$n$  — кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому  $I_{\text{НП}} = 1$ .

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра (4.3) [37]

$$I_{ТП} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (4.3)$$

де  $I_{ТП}$  — груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

$q_i$  — одиничний параметричний показник  $i$ -го параметра;

$\alpha_i$  — вагомість  $i$ -го параметричного показника,  $\sum_{i=1}^n \alpha_i = 1$ ;

$n$  — кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{ТП} = 1,1 \cdot 0,3 + 1,21 \cdot 0,4 + 1,67 \cdot 0,05 + 1,33 \cdot 0,07 + 1,33 \cdot 0,18 = 1,23.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою (4.4) [37]

$$I_{ЕП} = \sum_{i=1}^m q_i \cdot \beta_i, \quad (4.4)$$

де  $I_{ЕП}$  — груповий параметричний індекс за економічними показниками;

$q_i$  — економічний параметр  $i$ -го виду;

$\beta_i$  — частка  $i$ -го економічного параметра,  $\sum_{i=1}^m \beta_i = 1$ ;

$m$  — кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{ЕП} = 0,81 \cdot 0,45 + 0,83 \cdot 0,55 = 0,82.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою (4.5) [37]:

$$K_{ИТ} = I_{ИП} \cdot \frac{I_{ТП}}{I_{ЕП}}, \quad (4.5)$$

$$K_{\text{ИИТ}} = 1 \cdot 1,23 / 0,82 = 1,50.$$

Інтегральний показник конкурентоспроможності  $K_{\text{ИИТ}} > 1$ , отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

#### 4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

#### Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою (4.6) [37]

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де  $k$  — кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  — місячний посадовий оклад конкретного дослідника, грн;

$t_i$  — число днів роботи конкретного дослідника, дн.;

$T_p$  — середнє число робочих днів в місяці,  $T_p = 24$  дні.

$$Z_o = 12220,00 \cdot 24 / 24 = 12220,00 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 — Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12220,00	509,17	24	12220,00
Інженер-розробник програмного забезпечення	12110,00	504,58	22	11100,83
Консультант (графоаналітик)	12100,00	504,17	10	5041,67
Технік	7170,00	298,75	10	2987,50
Всього				31350,00

### Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» розраховуємо за формулою (4.7) [37]:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.7)$$

де  $C_i$  — погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  — час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою (4.8) [37]:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.8)$$

де  $M_M$  — розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=2379,00$  (грн.);

$K_i$  — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [46];

$K_c$  — мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  — середнє число робочих днів в місяці, приблизно  $T_p = 24$  дн;

$t_{зм}$  — тривалість зміни, год.

$$C_1 = 2379,00 \cdot 1,10 \cdot 1,65 / (24 \cdot 8) = 22,49 \text{ (грн.)}$$

$$З_{р1} = 22,49 \cdot 9,20 = 206,90 \text{ (грн.)}$$

Таблиця 4.6 — Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка електронно-обчислювального обладнання	9,20	2	1,10	22,49	206,90
Підготовка робочого місця розробника програмного забезпечення	7,60	2	1,10	22,49	170,92
Інсталяція програмного забезпечення	6,20	5	1,70	34,76	215,49
Монтаж оптичних пристроїв (сканери, блоки автоподачі тощо)	6,50	3	1,35	27,60	179,40
Введення кодів програмних модулів	11,00	3	1,35	27,60	303,60
Підготовка програмного забезпечення	8,00	5	1,70	34,76	278,05
Тестування	4,50	2	1,10	22,49	101,20
Всього					1455,55

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою (4.9) [37]:

$$З_{дод} = (З_o + З_p) \cdot \frac{H_{дод}}{100\%}, \quad (4.9)$$

де  $H_{дод}$  — норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$З_{дод} = (31350,00 + 1455,55) \cdot 10 / 100\% = 3280,55 \text{ (грн.)}$$



### 4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою (4.10) [37]:

$$Z_n = (Z_o + Z_p + Z_{од}) \cdot \frac{H_{зп}}{100\%} \quad (4.10)$$

де  $H_{зп}$  — норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (31350,00 + 1455,55 + 3280,55) \cdot 22 / 100\% = 7938,94 \text{ (грн.)}$$

### 4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу».

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою (4.11) [37]:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{вj} \quad (4.11)$$

де  $H_j$  — норма витрат матеріалу  $j$ -го найменування, кг;

$n$  — кількість видів матеріалів;

$C_j$  — вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  — коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  — маса відходів  $j$ -го найменування, кг;

$C_{вj}$  — вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 4,00 \cdot 160,00 \cdot 1,1 - 0,000 \cdot 0,00 = 704,00 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір 500-80	160,00	4,00	-	-	704,00
Папір для записів А5	54,00	4,00	-	-	237,60
Органайзер офісний Office	212,00	3,00	-	-	699,60
Канцелярське приладдя (набір офісного працівника)	185,00	3,00	-	-	610,50
Картридж для принтера Canon LBP	923,00	1,00	-	-	1015,30
Диск оптичний Optivisio CD-RW	14,00	3,00	-	-	46,20
Flesh-пам'ять DATA 64 GB	209,00	2,00	-	-	459,80
Тека для паперів	95,00	3,00	-	-	313,50
Інше	100,00	1,0	-	-	110,00
Всього					4196,50

#### 4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_6$ ), які використовують при проведенні НДР на тему «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» відсутні.

#### 4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою (4.12) [37]:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (4.12)$$

де  $C_i$  — ціна придбання одиниці спекустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$  — кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  — коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  — кількість найменувань устаткування.

$$B_{\text{спец}} = 19000,00 \cdot 1 \cdot 1,11 = 21090,00 \text{ (грн.)}$$

Отримані результати зведемо до таблиці 4.8:

Таблиця 4.8 — Витрати на придбання спекустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Сканер CANON DR-C240 (0651C003)	1	19000,00	21090,00
Всього			21090,00

#### 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою (4.13):

$$B_{\text{прг}} = \sum_{i=1}^k C_{\text{іпрг}} \cdot C_{\text{прг.}i} \cdot K_i, \quad (4.13)$$

де  $C_{\text{іпрг}}$  — ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прг.}i}$  — кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  — коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  — кількість найменувань програмних засобів.

$$V_{\text{прг}} = 12670,00 \cdot 1 \cdot 1,1 = 13937,00 \text{ (грн.)}$$

Отримані результати зведемо до таблиці 4.9:

Таблиця 4.9 — Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
База даних згорткової нейронної мережі (MNN)	1	12670,00	13937,00
Всього			13937,00

#### 4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою (4.14) [38]:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.14)$$

де  $Ц_{\text{б}}$  — балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$  — термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{в}}$  — строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (28750,00 \cdot 1) / (3 \cdot 12) = 798,61 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 — Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн

Завершення таблиці 4.10

Персональний комп'ютер	28750,00	3	1	798,61
Робоче місце розробника	9560,00	5	1	159,33
Пристрої виводу інформації	8624,00	4	1	179,67
Оргтехніка	9241,00	4	1	192,52
Приміщення лабораторії	265000,00	25	1	883,33
ОС Windows 11	7459,00	2	1	310,79
Прикладний пакет Microsoft Office 2019	6786,00	2	1	282,75
Засоби передачі даних	5265,00	2	1	219,38
Всього				3026,38

## 4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховують за формулою (4.15):

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (4.15)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 4,50$ (грн.) [39];

$K_{впi}$  – коефіцієнт, що враховує використання потужності,  $K_{впi} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,65 \cdot 200,0 \cdot 4,50 \cdot 0,95 / 0,97 = 585,00 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.11.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер	0,65	200,0	585,00
Робоче місце розробника	0,20	190,0	171,00
Пристрої виводу інформації	0,02	50,0	4,50
Оргтехніка	0,80	10,0	36,00
Засоби передачі даних	0,05	190,0	42,75
Сканер CANON DR-C240 (0651C003)	0,07	100,0	31,50
Всього			870,75

#### 4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою (4.16):

$$B_{cb} = (Z_o + Z_p) \cdot \frac{H_{cb}}{100\%}, \quad (4.16)$$

де  $H_{cb}$  — норма нарахування за статтею «Службові відрядження», приймемо  $H_{cb} = 20\%$ .

$$B_{cb} = (31350,00 + 1455,55) \cdot 20 / 100\% = 6561,11 \text{ (грн.)}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою (4.17):

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.17)$$

де  $H_{cn}$  — норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 35\%$ .

$$B_{cn} = (31350,00 + 1455,55) \cdot 35 / 100\% = 11481,94 \text{ (грн.)}$$

#### 4.3.11 Інші 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» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою (4.18):

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.18)$$

де  $H_{ie}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{ie} = 60\%$ .

$$I_e = (31350,00 + 1455,55) \cdot 60 / 100\% = 19683,33 \text{ (грн.)}$$

#### 4.3.12 Накладні (загальновиробничі) витрати

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 (загально виробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою (4.19):

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.19)$$

де  $H_{нзв}$  — норма нарахування за статтею «Накладні (загально виробничі) витрати», прийmemo  $H_{нзв} = 125\%$ .

$$B_{нзв} = (31350,00 + 1455,55) \cdot 125 / 100\% = 41006,94 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи на тему «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» розраховуємо як суму всіх попередніх статей витрат за формулою (4.20):

$$B_{заг} = Z_o + Z_p + Z_{оод} + Z_n + M + K_v + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_v + B_{нзв}. \quad (4.20)$$

$$B_{заг} = 31350,00 + 1455,55 + 3280,55 + 7938,942713 + 4196,50 + 0,00 + 21090,00 + 13937,00 + 3026,38 + 870,75 + 6561,11 + 11481,94 + 19683,33 + 41006,94 = 165878,99 \text{ (грн.)}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою (4.21):

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.21)$$

де  $\eta$  — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta = 0,95$ .

$$ЗВ = 165878,99 / 0,95 = 174609,47 \text{ (грн.)}$$

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

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.



Результати дослідження проведені за темою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

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

$\Delta N$  — збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Таблиця 4.12 Кількість споживачів

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	2000	3000	2500	1500

$N$  — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 35000 осіб;

$C_0$  — вартість програмного продукту у році до впровадження результатів розробки, прийmemo 2100,00 грн;

$\pm \Delta C_0$  — зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 20,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [37] (4.22):

$$\Delta \Pi_i = (\pm \Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.22)$$

де  $\lambda$  — коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  — коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo  $\rho = 40\%$ ;

$\vartheta$  — ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році  $\vartheta=18\%$ .

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (20,00 \cdot 35000,00 + 2120,00 \cdot 2000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1344865,60 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (20,00 \cdot 35000,00 + 2120,00 \cdot 5000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 3076312,00 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (20,00 \cdot 35000,00 + 2120,00 \cdot 7500) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 4519184,00 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (20,00 \cdot 35000,00 + 2120,00 \cdot 9000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 5384907,20 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки (4.23):

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.23)$$

де  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau=0,14$ ;

$t$  — період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 1344865,60/(1+0,14)^1 + 3076312,00/(1+0,14)^2 + 4519184,00/(1+0,14)^3 + \\ &+ 5384907,20/(1+0,14)^4 = 1179706,67 + 2367122,19 + 3050320,48 + 3188297,35 = \\ &= 9785446,68 \text{ (грн.)} \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки формула (4.24):

$$PV = k_{инв} \cdot 3B, \quad (4.24)$$

де  $k_{инв}$  — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв} = 1,5$ ;

$3B$  — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 174609,47 (грн.)

$$PV = k_{инв} \cdot 3B = 1,5 \cdot 174609,47 = 261914,20 \text{ (грн.)}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме (4.25):

$$E_{абс} = III - PV \quad (4.25)$$

де  $III$  — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 9785446,68 грн;

$PV$  — теперішня вартість початкових інвестицій, 261914,20 (грн.)

$$E_{абс} = III - PV = 9785446,68 - 261914,20 = 9523532,48 \text{ (грн.)}$$

Внутрішня економічна дохідність інвестицій  $E_е$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки (формула 4.26):

$$E_е = Tж \sqrt{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.26)$$

де  $E_{абс}$  — абсолютний економічний ефект вкладених інвестицій, 9523532,48 грн;

$PV$  — теперішня вартість початкових інвестицій, 261914,20 грн;

$T_{жс}$  — життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_e = T_{жс} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 9523532,48/261914,20)^{1/4} = 1,47.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій розрахована за формулою (4.27):

$$\tau_{мин} = d + f, \quad (4.27)$$

де  $d$  — середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні  $d = 0,11$ ;

$f$  — показник, що характеризує ризикованість вкладення інвестицій, приймемо  $0,16$ .

$\tau_{мин} = 0,11 + 0,16 = 0,27 < 1,47$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_e$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки визначений за формулою (4.28):

$$T_{ок} = \frac{1}{E_e}, \quad (4.28)$$

де  $E_e$  — внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 1,47 = 0,68 \text{ (р.)}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Технологія оптичного розпізнавання рукописних

символів в системах електронного документообігу» становить 33,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,50 рази.

Також термін окупності становить 0,68 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Технологія оптичного розпізнавання рукописних символів в системах електронного документообігу».

## ВИСНОВКИ

Існуючі системи розпізнавання текстових документів із високою достовірністю дозволяють переводити такі документи у електронну форму. Але при появі у таких документах рукописних символів значно зростає відсоток помилок розпізнавання. Ця проблема на теперішній час є досить актуальною. Дана магістерська робота присвячена вирішенню цієї задачі по розробці програмних засобів розпізнавання текстів із рукописними символами у системах електронного документообігу.

У першому розділі магістерської роботи був виконаний огляд існуючих систем та методів оптичного розпізнавання текстових документів, проведено порівняльний аналіз засобів розпізнавання, вказано на основні недоліки та обґрунтовано необхідність розробки програмного продукту для розпізнавання рукописних символів у системах електронного документообігу. Проведений аналіз різних методів та засобів розпізнавання символів друкованих документів дозволив виявити проблеми, що існують в даній області та уточнити області застосування різних методів, їх переваги та недоліки в застосуванні до вирішення завдань по розпізнаванню рукописних символів.

У другому розділі магістерської роботи запропонована послідовність розпізнавання рукописних символів документів, визначена нейронна мережа для виділення рукописних символів у отриманому тексті та їх розпізнавання. Процес розпізнавання рукописних символів запропоновано розділити на два етапи на першому етапі використовується процес по підготовці документа до виділення рядків, слів та символів, а на другому етапі використовується нейронна мережа для розпізнавання уже підготовлених та виділених символів. Така послідовність дозволяє збільшити достовірність розпізнавання рукописних символів.

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

У четвертому розділі магістерської кваліфікаційної роботи було здійснено обґрунтування доцільності виконання нового наукового вирішення поставленої проблеми розпізнавання рукописних символів, виконано розрахунок необхідних економічних витрат для реалізації запропонованих засобів виділення та розпізнавання рукописних символів текстових документів та визначено комерційні переваги від впровадження запропонованого програмного продукту в системах розпізнавання рукописних символів.

Розроблений у магістерській роботі програмний продукт можна буде використовувати в засобах розпізнавання рукописних текстових документів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методы компьютерной обработки изображений / Под ред. В. А. Сойфера. — М.: Физматлит, 2003. — 784 с. — ISBN 5-9221-0270-2.
2. Желтов С. Ю. Обработка и анализ изображений в задачах машинного зрения / С. Ю. Желтов. — М.: Физматкнига, 2010. — 672 с.
3. Что такое OCR [Электронный ресурс] — Режим доступа: [https://www.abbyy.com/ru-ru/finereader/what-is-ocr/?\\_\\_c=1](https://www.abbyy.com/ru-ru/finereader/what-is-ocr/?__c=1).
4. Хаустов П. А. Алгоритмы распознавания рукописных символов на основе построения структурных моделей / П.А. Хаустов // Компьютерная оптика. — 2017. — Т. 41, № 1. — С. 67—78.
5. Що таке ABBYY FineReader [Электронный ресурс] — Режим доступа: <https://help.abbyy.com/uk-ua/finereader/12/overview>
6. Сравнение систем оптического распознавания. [Электронный ресурс] — Режим доступа: [https://en.wikipedia.org/wiki/Comparison\\_of\\_optical\\_character\\_recognition\\_software](https://en.wikipedia.org/wiki/Comparison_of_optical_character_recognition_software).
7. OCR CuneiForm для Windows [Электронный ресурс] — Режим доступа: <http://softobase.com/ru/ocr-cuneiform>.
8. FineReader [Электронный ресурс]. — Режим доступа: [http://ru.wikipedia.org/wiki/ABBYY\\_FineReader](http://ru.wikipedia.org/wiki/ABBYY_FineReader).
9. Программа Readiris Pro [Электронный ресурс] — Режим доступа: <https://readiris.ru.uptodown.com/windows>
10. 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.
11. Липкина, А. Распознавание текста по структуре скелета букв / А. Липкина. — М.: МГУ им. Ломоносова, 2018. — 31 с.
12. Моругов А. М. Методы распознавания символов. / А.М. Моругов, С.В.Волков //Труды Международного симпозиума «Надежность и качество», 2017, том 1. С. 76—81.



13. Титов Ю. В. Об искажении символов при сканировании // Сб. трудов ИСА РАН «Системный подход к управлению информацией». М.: КомКнига, 2006, Том № 23. — С. 260—288.
14. Кристофер М. Бишоп. Распознавание образов и машинное обучение. М.: Диалектика, 2020. — 624 с
15. 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.
16. Хайкин С. Б. Нейронные сети: полный курс. — 2-е изд. Пер. с англ. — М.: Издательский дом «Вильямс», 2006. — 1104 с.
17. Мокшин В. В. Разработка системы распознавания символов с использованием нейронной сети / В. В. Мокшин, Л. Р. Габдрахманова // Современные инновации в науке и технике. — 2014. — № 4. — С. 223—225.
18. Shi B., Bai X., Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition // IEEE transactions on pattern analysis and machine intelligence. 2017, vol. 39, № 11, pp. 2298—2304.
19. Крысова И. В., Обработка изображений с помощью нейронных сетей/ И. В. Крысова, С. В. Пасечник, П. С. Бугаенко // XXXIX Гагаринские чтения. Том 4. — М.: МАТИ, 2013. — С. 144-146.
20. Busta M., Neumann L., Matas J. Deep text spotter: an end-to-end trainable scene text localization and recognition framework // Proceedings of the IEEE International Conference on Computer Vision. — 2017, pp. 2204—2212.
21. Ye Q., Doermann D. Text detection and recognition in imagery: A survey // IEEE transactions on pattern analysis and machine intelligence. — 2015, vol. 37, no. 7, pp. 1480—1500.
22. Uchida S. Statistical Deformation Model for Handwritten Character Recognition // Advances in Digital Document Processing and Retrieval. — 2014, pp. 157—174.

23. Чочиа П.А. Сегментация изображений на основе анализа расстояний в пространстве признаков. // Автометрия. — 2014. — Vol. 50. — No.6. — P. 97—110.
24. Болотова Ю. А. Обзор алгоритмов детектирования текстовых областей на изображениях и видеозаписях / Ю.А. Болотова, В.Г. Спицын, П.М. Осина // Компьютерная оптика. —2017. — Т. 41, № 3. — С. 441—452.
25. Погребняк И. В. Распознавание символов на изображениях, содержащих искажения / И. В. Погребняк, А. Ю. Тропченко // Международный научно-исследовательский журнал. — 2017. — Выпуск № 05 (59), часть 3, с.81—86.
26. Jaderberg, M. Reading text in the wild with convolutional neural networks / M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman // International Journal of Computer Vision. — 2016. — Vol. 116, Issue 1. — P. 1—20.
27. Shaoqing Ren Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun // arXiv:1506.01497v3 [cs.CV]. — 2016, 14p.
28. Otsu N. A threshold selection method from gray-level histograms // IEEE transactionson systems, man, and cybernetics. — 1979. — vol. 9, no. 1, pp. 62—66.
29. OpenCV library [Электронный ресурс]. — Режим доступа: <https://opencv.org/>.
30. LTI-Lib [Электронный ресурс]. — Режим доступа: <http://ltilib.sourceforge.net/doc/homepage/index.shtml>.
31. AForge.NET: Framework [Электронный ресурс]. — Режим доступа: <http://www.aforgenet.com/framework/>.
32. OpenCV vs VXL vs LTI: Performance Test - AI Shack [Электронный ресурс]. — Режим доступа: <http://www.aishack.in/tutorials/opencv-vs-vxl-vs-lti-performance-test/>.
33. Коэльё Л. Построение систем машинного обучения на языке Python. / Коэльё Л. П., Ричерт В. — Перевод с английского. — М.: ДМК Пресс, 2015. — 364с. — ISBN 978-5-97060-330-7.

34. Маккинли У. Python и анализ данных. — Перевод с английского. — М.: ДМК Пресс, 2015. — 482 с. — ISBN 978-5-97060-315-4.
35. The MNIST dataset. [Электронный ресурс]. — Режим доступа: <https://www.nist.gov/itl/iad/image-group/mnist-dataset>.
36. Смирнов С.В. Критерии оценки качества результатов оптического распознавания / С.В. Смирнов // Сборник материалов XVI Международной научно-практической конференции «Перспективы развития информационных технологий». — Новосибирск: Издательство ЦРНС, 2013. — С. 33—38.
37. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.
38. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка — Вінниця : ВНТУ, 2016. — 113 с.
39. 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  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

### **ЗАТВЕРДЖУЮ**

Завідувач кафедри ОТ

\_\_\_\_\_ проф., д.т.н. О. Д. Азаров

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

### **ТЕХНІЧНЕ ЗАВДАННЯ**

на виконання магістерської кваліфікаційної роботи  
«Технологія оптичного розпізнавання рукописних символів в системах  
електронного документообігу»

08-23.МКР.008.00.000 ТЗ

Науковий керівник: к.т.н., доцент

\_\_\_\_\_ Савицька Л.А.

Магістрант групи 1КІ-20м

\_\_\_\_\_ Осельський В.В.

Вінниця 2021

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

1.1 Актуальність даного дослідження визначається необхідністю вирішення основних проблем існуючих засобів пошуку та розпізнавання рукописних символів текстових документів. Особливо актуальним дане дослідження є у зв'язку з тенденціями до впровадження таких засобів у різноманітних додатках для пошуку та розпізнавання рукописних символів у системах електронного документообігу.

1.2 Наказ про затвердження теми магістерської кваліфікаційної роботи.

## 2 Мета і призначення МКР

2.1 Мета роботи полягає у підвищенні ефективності розпізнавання рукописних символів текстових документів. Підвищення ефективності розпізнавання рукописних символів текстових документів досягається шляхом більш точної перевірки об'єктів на зображенні із набором еталонних символів.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи із подальшим впровадженням та розвитком продукту..

## 3 Вихідні дані для виконання МКР

— запропонувати нові підходи для реалізації методу розпізнавання рукописних символів текстових документів.;

— вхідні дані: роздільна здатність зображення тексту не менше 300 dpi, оптична щільність зображення – не менше 2,5Т, кількість градацій яскравості зображення – 256.

## 4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

— запропонувати нові підходи для реалізації методу пошуку та розпізнавання рукописних символів текстових документів.

— розробити послідовність для розпізнавання рукописних символів текстових документів у системах електронного документообігу.

— виконати розробку програмного забезпечення для розпізнавання рукописних символів, провести його тестування. Схеми послідовності

розпізнавання рукописних символів та лістинги програми представити в додатках до роботи.

## 5 Етапи МКР та очікувані результати таблиця А.1

Таблиця А.1 — Етапи виконання роботи

№	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	07.09.21	09.09.21	Вступ
2	Аналіз способів виділення та розпізнавання рукописних символів	10.09.21	28.09.21	розділ 1
3	Розробка способу та послідовності розпізнавання рукописних символів	29.09.21	21.10.21	Розділ 2, розробка способу
4	Розробка програми розпізнавання рукописних символів	22.10.21	10.11.21	Розділ 3, розробка програми
5	Практична реалізація, результати	11.11.21	16.11.21	Розділ 3
6	Розробка економічної частини	17.11.21	30.11.21	Розділ 4
7	Оформлення пояснювальної записки	01.12.21	17.12.21	ПЗ, презентація

6 Матеріали, що подаються до захисту МКР — пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив рецензента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами.

## 7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної 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

Вимоги викладені в «Положенні про порядок підготовки магістрів у Вінницькому національному технічному університеті» з урахуванням змін, що подані у бюлетені ВАК України № 9-10, 2011р., а також в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008-

2015, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

9 Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання до виконання отримав \_\_\_\_\_ Осельський В. В.

## ДОДАТОК Б

### Лістинг програми

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
image_file = "text.png"
img = cv2.imread(image_file)
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY)
img_erode = cv2.erode(thresh, np.ones((3, 3), np.uint8), iterations = 1)
contours, hierarchy = cv2.findContours(img_erode, cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)
output = img.copy()
for idx, contour in enumerate(contours):
    (x,y,w,h) = cv2.boundingRect(contour)
    if hierarchy[0][idx][3] == 0:
        cv2.rectangle(output, (x, y), (x +w, y +h), (80, 0, 0), 1)
cv2.imshow("Input", img)
cv2.imshow("Enlarged", img_erode)
cv2.imshow("Output", output)
cv2.waitKey(0)
def letters_extract(image_file: str, out_size= 28) -> List[Any]:
size_max = max(w, h)
letter_square = 255 * np.ones(shape= [size_max, size_max], dtype=
np.uint8)
if w >h:
    y_pos = size_max//2 - h //2
```



```

        letter_square[y_pos:y_pos + h, 0:w] = letter_crop
elif w < h:
    x_pos = size_max//2 - w //2
    letter_square[0:h, x_pos:x_pos + w] = letter_crop
else:
    letter_square = letter_crop
    letters.append ((x, w, cv2.resize (letter_square, (out_size, out_size),
interpolation= cv2.INTER_AREA)))
    letters.sort (key= lambda x:x[0], reverse= False)
return letters

from keras import backend as K
from keras.optimizers import SGD, RMSprop, Adam
from keras.constraints import maxnorm
import tensorflow as tf
def emnist_model():
    model = Sequential()
    model.add (Convolution2D (filters=32, kernel_size=(3, 3), padding= 'valid',
input_shape= (28, 28, 1), activation = 'relu'))
    model.add (Convolution2D (filters=64, kernel_size=(3, 3), activation = 'relu'))
    model.add (MaxPooling2D (pool_size= (2, 2)))
    model.add (Flatten())
    model.add (Dropout (0.25))
    model.add (Dense (512, activation= 'relu'))
    model.add (Dropout (0.5))
    model.add (Dense (len (emnist_labels), activation= 'softmax'))
    model.compile (loss='categorical_crossentropy', optimizer= 'adadelta', metrics=
['accuracy'])
    return model
num_classes = 10
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)

```

```

input_shape = (28, 28, 1)
batch_size = 128
epochs = 10
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',
input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss=keras.losses.categorical_crossentropy,optimizer=keras.optimi
zers.Adadelta(),metrics=['accuracy'])
hist = model.fit(x_train, y_train, batch_size = batch_size, epochs=epochs,
verbose=1, validation_data=(x_test, y_test))

import idx2 numpy
emnist_path = '/home/Documents/ TestApps/keras/emnist/'
X_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
train- images- idx3- ubyte')
y_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
train- labels- idx1- ubyte')
X_test = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
test- images- idx3- ubyte')
y_test = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
test- labels- idx1- ubyte')
X_train = np.reshape(X_train, (X_train.shape [0], 28, 28, 1))
X_test = np.reshape(X_test, (X_test.shape [0], 28, 28, 1))
print(X_train.shape, y_train.shape, X_test.shape, y_test.shape,
len(emnist_labels))

```

```

X_train = X_train[:X_train.shape[0]]
y_train = y_train[:y_train.shape[0]]
X_test = X_test[:X_test.shape[0]]
y_test = y_test[:y_test.shape[0]]
X_train = X_train.astype(np.float32)
X_train /= 255.0
X_test = X_test.astype(np.float32)
X_test /= 255.0
x_train_cat = keras.utils.to_categorical(y_train, len(ernist_labels))
y_test_cat = keras.utils.to_categorical(y_test, len(ernist_labels))
model = keras.models.load_model('ernist_letters.h5')
def ernist_predict_img(model, img):
    img_arr = np.expand_dims(img, axis=0)
    img_arr = 1 - img_arr/255.0
    img_arr[0] = np.rot90(img_arr[0], 3)
    img_arr[0] = np.fliplr(img_arr[0])
    img_arr = img_arr.reshape((1, 28, 28, 1))
    result = model.predict_classes([img_arr])
    return chr(ernist_labels[result[0]])
def img_to_str(model: Any, image_file: str):
    letters = letters_extract(image_file)
    s_out = ""
    for i in range(len(letters)):
        dn = letters[i+1][0] - letters[i][0] - letters[i][1] if i < len(letters) - 1 else 0
        s_out += ernist_predict_img(model, letters[i][2])
        if (dn > letters[i][1]/4):
            s_out += ' '
    return s_out

```

## ДОДАТОК В

Послідовність розпізнавання символів текстових документів

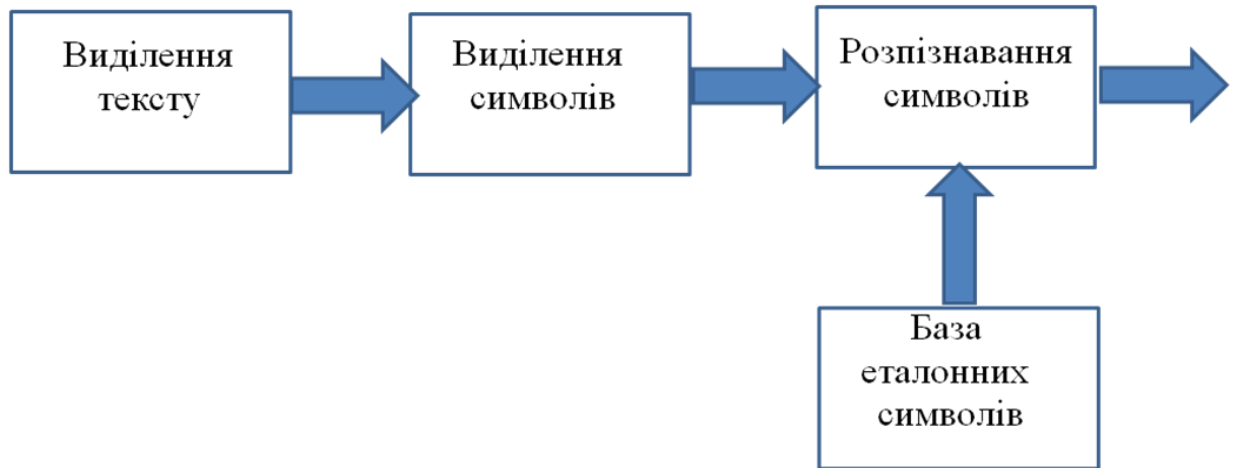


Рисунок В.1 — Послідовність розпізнавання символів текстових документів

## ДОДАТОК Г

### Методи розпізнавання тексту



Рисунок Г.1 — Методи розпізнавання тексту

## ДОДАТОК Д

### Системи оптичного розпізнавання тексту

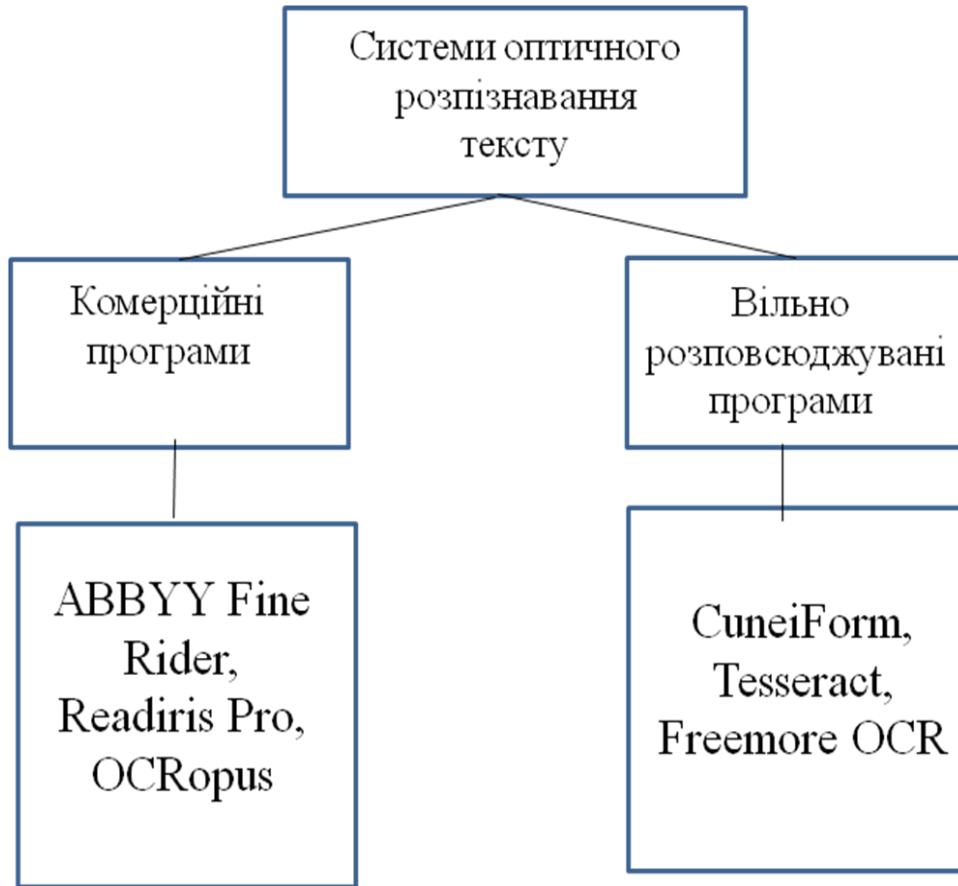


Рисунок Д.1 — Системи оптичного розпізнавання тексту

## ДОДАТОК Е

Структура програми розпізнавання рукописних символів



Рисунок Е.1— Структура програми розпізнавання рукописних символів

## ДОДАТОК Ж

### Послідовність виділення символу

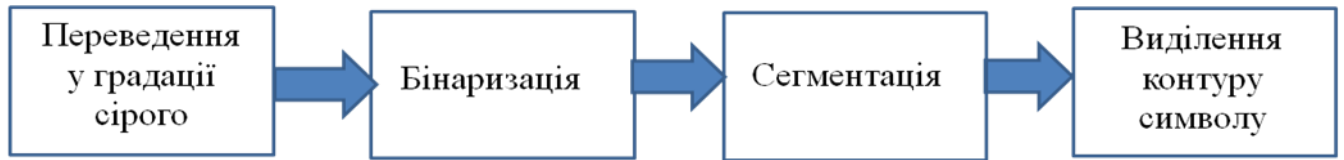


Рисунок Ж.1— Послідовність виділення символу



## ДОДАТОК К

### ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Технологія оптичного розпізнавання рукописних символів  
в системах електронного документообігу

Тип роботи: магістерська кваліфікаційна робота  
(кваліфікаційна роботи, курсовий проект (робота), реферат, аналітичний огляд, інше (вказати))

Підрозділ кафедра обчислювальної техніки, 1КІ-20м  
(кафедра, факультет (інститут), навчальна група)

Науковий керівник Савицька Л.А., доц. кафедри ОТ  
(прізвище, ініціали, посада)

#### Показники звіту подібності

Plagiat.pl(StrikePlagiarism)		Unicheck	
КП1		Оригінальність	92,1
КП2			
Тривога/Білі знаки	/	Схожість	7,9

#### Аналіз звіту подібності (відмінити подібне)

- 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 і відсутності самостійності її автора. Робот направити на доопрацювання.
- 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.

Заявляю, що ознайомлений(-на) з повним звітом подібності, який був  
згенерований Системою щодо роботи (додається)

Автор \_\_\_\_\_ Осельський В.В.  
(підпис) (прізвище, ініціали)

#### Опис прийнятого рішення

Ступінь оригінальності роботи відповідає вимогам, що висуваються до МКР

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