

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ЯКОВЧУК ОЛЕГ КОСТЯНТИНОВИЧ

УДК 004.021:004.023:004.93

ДИСЕРТАЦІЯ

МЕТОДИ ВИРІВНЮВАННЯ РУКОПИСНОГО ТЕКСТУ НА ОСНОВІ РОЗПІЗНАВАННЯ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ ТА СТРУКТУРНОГО АНАЛІЗУ СИМВОЛІВ

122 – Комп'ютерні науки
Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

_____ О.К. Яковчук
Науковий керівник: Рогоза Валерій Станіславович,
доктор технічних наук, професор

Київ – 2024

АНОТАЦІЯ

Яковчук О.К. Методи вирівнювання рукописного тексту на основі розпізнавання з використанням машинного навчання та структурного аналізу символів. Кваліфікаційна наукова праця на правах рукопису.

Дисертаційна робота на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», 2024.

Метою дисертаційного дослідження є підвищення ефективності вирішення комплексу задач, що виникають при вирівнюванні рукописного тексту.

Разом із зростанням популярності рукописного введення постає завдання ефективної обробки рукописної інформації, зокрема рукописного тексту. Однією із задач є забезпечення структурної коректності та зрозумілості тексту, представленого в рукописному вигляді. Зокрема, актуальним завданням є вирішення проблеми вирівнювання рукописного тексту, при цьому зберігаючи формат, структуру рукописних елементів та їх текстовий зміст. Важливим є також збереження ідентичності та унікальності написаного тексту, що визначаються персональними особливостями написання та почерку автора. Водночас створювані рішення повинні мати змогу працювати в умовах обмеження обчислювальних ресурсів, на персональних пристроях, таких як смартфони, розумні годинники, інтерактивні дошки, монітори з рукописним введенням та інші гаджети.

У даній дисертаційній роботі акцент робиться на вирішенні задачі вирівнювання рукописного тексту зі збереженням початкового стилю написання, без перетворення рукописного тексту в друковану форму. В дослідженні використовуються передові методи та технології для обробки рукописного тексту, включно з розпізнаванням та сегментацією тексту. Дослідження в напрямках обробки рукописного тексту, разом з

візуальним покращенням тексту для підвищення його ясності та чіткості, має важливе значення для подальшого вдосконалення систем рукописного введення та їх використання в різних сферах, від освітнього до професійного використання. Розвиток вирівнювання рукописного тексту відкриває нові перспективи для зручної та ефективної роботи з рукописною інформацією, що стає невід'ємною частиною сучасного цифрового життя.

В першому розділі дисертаційної роботи описується постановка задачі вирівнювання рукописного тексту, визначаються основні проблеми, що існують на сьогоднішній день та потребують уваги для ефективного вирішення поставлених задач. Визначається підзадача оцінки рукописного тексту на розбірливість та вирівняність, а також підзадача порівняння двох рукописних текстів для визначення якості результатів вирівнювання. Проводиться ґрунтовний аналіз існуючих досліджень та передових робіт як в області обробки рукописного введення, так і конкретно для задачі вирівнювання тексту, аналіз підходів до розпізнавання та сегментації рукописного тексту для визначення найбільш релевантних для використання в системі, що розроблюється в цій роботі. Розглядається загальна структура запропонованого рішення.

В другому розділі представлено рішення, що пропонується використовувати в системі вирівнювання для проведення розпізнавання та сегментації рукописного введення. Досліджуються принципи роботи рекурентних нейронних мереж для задачі розпізнавання рукописного тексту. Описується ідея довгої короткочасної пам'яті, представляється архітектура мережі з використанням довгої короткочасної пам'яті, що дає змогу вирішувати відразу два завдання: сегментації та класифікації символів. Досліджується принцип нейромережевої часової класифікації. Представляється додаткова мережа для уточнення та покращення результатів сегментації символів. Представлено додаткові алгоритми для приєднання відтермінованих символів, пошуку та виправлення помилок сегментації.

В третьому розділі приведено опис основної частини системи вирівнювання рукописного тексту. Представляється запропонований новий метод вирівнювання, що використовує структурний аналіз символів. Цей розділ включає в себе розгляд процесу попередньої обробки вхідних даних, представлення метрик текстового рядка, запропонованого розбиття символів на класи, виконання знаходження класів символів, обрахунку метрик всіх символів та фінальних метрик вирівняного рядка, проведення переміщення символів на знайдені відповідні позиції. Детально розглядається процес виконання кожного кроку, виконується розгляд розроблених алгоритмів та підходів, що застосовуються для структурного аналізу символів. Результатом даного розділу є спроектована та розроблена система для структурного аналізу рукописного тексту, що дозволяє виконувати поставлену задачу вирівнювання.

В четвертому розділі представлені основні експерименти для аналізу якості роботи розробленої системи. Описується набір експериментальних даних та його базові характеристики. Проводиться оцінка точності системи розпізнавання. Виконуються обчислювальні експерименти, на основі яких підтверджено загальну спроможність системи успішно вирівнювати рукописний текст. Проводиться аналіз успішних прикладів роботи системи та допущених помилок при вирівнюванні, оцінка продуктивності роботи системи в середовищі з обмеженими ресурсами. Також розроблена система порівнюється з наявними підходами до поліпшення зовнішнього вигляду та вирівнювання рукописного тексту, результат підтвердив визначені переваги розробленої системи та її практичну цінність.

Основним результатом цієї роботи є розроблена система вирівнювання рукописного тексту, яка базується на запропонованому методі вирівнювання рукописного тексту на основі розроблених алгоритмів розпізнавання та сегментації тексту, а також з використанням структурного аналізу символів, що враховує індивідуальні особливості почерку, зв'язне написання символів та дозволяє зберегти

початковий стиль написання тексту. Розроблена система вирівнювання дає змогу розв'язувати наступні задачі по вирівнюванню текстів українською мовою: вирівнювання одного рядка рукописного тексту, вирівнювання текстів, що складаються з багатьох рядків, вирівнювання документів, що містять різні типи текстових блоків, вирівнювання текстової інформації з можливістю підтримувати різні типи контенту, такі як специфічні символи, математичні формули, адреси електронної пошти, номери телефонів і т.д. Особливістю розробленої системи є можливість до масштабування та розширення для підтримки нових мов, типів контенту, введення окремих правил для особливих задач вирівнювання. Перевагою розробленої системи є здатність вирівнювати текст з врахуванням персональних особливостей написання користувача.

Розроблену систему вирівнювання тексту було успішно інтегровано в тестовий мобільний застосунок, який надає змогу користувачу вводити рукописний текст, отримувати результат розпізнавання тексту та результат вирівнювання тексту. Проведено експериментальні дослідження, що підтвердили здатність системи виконувати всі поставлені завдання. Система показала високі характеристики якості роботи та швидкодії в умовах роботи на пристроях з обмеженнями обчислювальних ресурсів. В результаті розроблена система може бути інтегрована до інших мобільних додатків для обробки та вирівнювання рукописного тексту. Підтримка роботи з українським текстом дає потенційні можливості для використання системи в різноманітних галузях нашої країни, таких як освіта, медицина, рекламний бізнес і т.д. На сьогодні існує тенденція цифровізації документообігу в різних сферах діяльності, в тому числі між державою та громадянами, де також може мати велику користь система обробки рукописного тексту для покращення візуального сприйняття та розбірливості написаного тексту. Запропоновані та розроблені у дисертаційній роботі методи було використано при розробці комерційних проєктів, зокрема мобільного додатку для смартфонів Samsung у вигляді функції вирівнювання рукописного тексту для різних мов.

Наукова новизна отриманих результатів.

В рамках цього дослідження було:

- Вперше запропоновано метод вирівнювання рукописного тексту, наукова новизна якого полягає у використанні розроблених алгоритмів розпізнавання та сегментації тексту, які базуються на теорії машинного навчання, а також у проведенні структурного аналізу символів, що враховує індивідуальні особливості почерку, зв'язне написання символів та дозволяє зберегти початковий стиль написання тексту.
- Вдосконалено метод сегментації рукописного тексту, що ґрунтується на використанні легковісної рекурентної нейронної мережі та динамічного програмування, який відрізняється від існуючих застосуванням додаткових алгоритмів, які дають змогу знаходити та виправляти помилки сегментації, визначати відповідні символи для відтермінованих штрихів, що дозволяє підвищити точність маркування кожного символу.
- Вперше розроблено метод вирівнювання для тексту, написаного українською мовою, наукова новизна якого полягає у використанні алгоритмів, що базуються на структурному аналізі символів, та визначаються окремо для кожного символу, а також у можливості реалізувати вирівнювання рукописних літер для різних стилів написання.

Практичне значення отриманих результатів.

Практичне значення результатів, отриманих у ході дисертаційного дослідження, зводиться до наступного переліку:

- Розроблені метод та алгоритми впроваджено в інструментальну систему вирівнювання рукописного тексту, що дає змогу розв'язувати задачі вирівнювання одного рядка рукописного тексту та текстів, що складаються з багатьох рядків, на пристроях з обмеженнями обчислювальних ресурсів.

- Розроблена система забезпечує високу точність результатів, що дозволяє використовувати її для знаходження правильних позицій всіх символів та їх коректного вирівнювання, що продемонстровано на прикладі її використання в тестовому мобільному додатку для вирівнювання рукописного тексту українською мовою.
- Представлений у цій роботі метод вирівнювання рукописного тексту впроваджений у комерційний додаток для смартфонів Samsung у вигляді повноцінної функції вирівнювання для різних мов.

Ключові слова: вирівнювання рукописного тексту, обробка рукописного введення, онлайн розпізнавання цифрових чорнил, класифікація символів, структурний аналіз, сегментація рукописного введення, рекурентна нейронна мережа, машинне навчання, штучний інтелект, інтелектуальне введення інформації, людино-комп'ютерна взаємодія.

ANNOTATION

Oleg Yakovchuk. Methods for handwriting text alignment based on recognition with the usage of machine learning and symbols structural analysis. Qualification scientific work as manuscript.

Doctor of Philosophy dissertation under 122 “Computer Science” specialty. – National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, 2024.

The aim of the dissertation research is to enhance the efficiency in addressing the tasks arising during the handwritten text alignment.

As handwritten input gains increasing popularity, the need for effectively processing handwritten information, especially handwritten text, becomes apparent. One of the tasks is to provide clearness and structural integrity of the handwritten text. Specifically, aligning handwritten text while keeping the format, structure of textual elements, and their content is crucial. Additionally, it is essential to preserve the individuality and distinctiveness of the written text, characterized by the author's unique handwriting traits. Furthermore, solutions must be capable of operating within computational resource limitations, catering to personal devices like smartphones, smartwatches, interactive whiteboards, handwriting input-enabled monitors, and other gadgets.

This dissertation focuses on the task of aligning handwritten text while preserving its original writing style, without converting handwritten text into printed form. Advanced methods and technologies for processing handwritten text are utilized in the research, including text recognition and segmentation. The research in the area of handwritten text processing and text visual enhancement for clarity and sharpness increase is significant for further refining handwriting input systems and their utilization across various domains, from educational to professional use. The development of handwritten text alignment opens up

new prospects for convenient and effective handling of handwritten information, which has become an integral part of modern digital life.

The first chapter of the dissertation describes the formulation of the handwritten text alignment task, identifying the key challenges existing today that require attention for the effective resolution of the stated objectives. A subtask is defined for evaluating the legibility and alignment of handwritten text, as well as a subtask for comparing two handwritten texts to determine the quality of alignment results. A thorough analysis of existing research and state-of-the-art works is conducted both in the field of handwriting processing and specifically for the task of text alignment. Approaches to handwriting recognition and segmentation are analyzed to determine the most relevant ones for use in the system being developed in this work. The overall structure of the proposed solution is also examined.

The second chapter presents the solution to be used in the alignment system for handwriting recognition and segmentation. The main principles of recurrent neural networks for handwritten text recognition are investigated. The concept of long short-term memory (LSTM) is described, and a network architecture utilizing LSTM is presented, resolving simultaneous segmentation and character classification tasks. The principle of connectionist temporal classification (CTC) is explored. An additional network is introduced for refining and improving the results of character segmentation. Additional algorithms are presented for handling delayed strokes, searching, and correcting segmentation errors.

The third chapter provides a description of the main component of the handwritten text alignment system. A novel alignment method is introduced, utilizing structural analysis of symbols. This section encompasses the discussion of the preprocessing of input data, the representation of text line metrics, the proposed symbols classes description, the algorithm of symbol class identification, the calculation of all symbols metrics, and the final metrics of the aligned text line. Additionally, this chapter covers the process of moving symbols to their corrected positions. Each step is thoroughly examined, including the developed algorithms and approaches applied for the structural analysis of the symbols. The outcome

of this chapter is the designed and developed system for the text structural analysis that enables the accomplishment of the alignment task.

The fourth chapter encompasses the primary experiments conducted to analyze the performance quality of the developed system. It describes the dataset used for experimentation along with its basic characteristics. An assessment of the accuracy of the recognition system is provided. Computational experiments are executed to confirm the overall capability of the system in successfully aligning handwritten text. Analysis of successful system operation instances and errors encountered during alignment is conducted, along with an evaluation of the system's performance in resource-constrained environments. Additionally, the system is compared with existing approaches to handwritten text recognition and beautification, validating the identified advantages of the developed system and its practical value.

The primary outcome of the research is the developed handwritten text alignment system, which relies on the proposed method of alignment based on the algorithms for text recognition and segmentation. It utilizes symbols structural analysis, taking into account individual handwriting characteristics and connected writing, thus preserving the original writing style. The developed alignment system enables the resolving of task for Ukrainian text alignment, including aligning single-line texts, multi-line texts, aligning documents with different types of text blocks, and aligning handwriting of various content types, such as specific symbols, mathematical formulas, email addresses, phone numbers, etc. A significant feature of the developed system is its scalability and adaptability for supporting new languages, content types, and the possibility to incorporate specific rules for unique alignment tasks. An advantage of the developed system is its capability to align text while considering the user's personalized writing characteristics.

The developed text alignment system has been successfully integrated into a test mobile application, enabling users to input handwritten text and receive recognition and alignment results. Experimental studies have confirmed the system's capability to fulfill all

the assigned tasks. The system has exhibited high-quality performance and efficiency, even when operating on devices with limited computational resources. Consequently, the developed system can be integrated into other mobile applications for processing and aligning handwritten text. The support for Ukrainian text offers potential opportunities for the system's utilization across various sectors within our country, including education, healthcare, advertising, etc. There is currently a trend towards digitalizing document workflows in different fields of activity, including interactions between the state and citizens. In such contexts, a handwriting text processing system could greatly benefit visual perception and legibility of written text. Moreover, the methods proposed and developed in the dissertation were used in the development of commercial projects, in particular, the mobile application for the Samsung smartphones, serving as a separate feature for the alignment of the text in various languages.

The scientific novelty of the results includes:

- For the first time, a method for handwritten text alignment has been proposed, the scientific novelty of which lies in the utilization of the developed algorithms for text recognition and segmentation, that is based on machine learning theory, coupled with symbols structural analysis. This approach considers individual handwriting characteristics and connected writing, thereby preserving the original writing style.
- The segmentation method for handwritten text based on the usage of a lightweight recurrent neural network and dynamic programming has been enhanced, and it distinguishes from the existing approaches by adding additional algorithms with the capability to identify and correct segmentation errors, as well as determine the corresponding symbols for the delayed strokes. This enhancement contributes to improving the accuracy of symbol labeling.
- For the first time, a method for aligning the text written in Ukrainian has been developed, the scientific novelty of which lies in the utilization of the algorithms that are based on the structural analysis of the symbols, and provide the individual

alignment rules for each symbol. This approach allows the handwritten letters alignment across different writing styles.

The practical significance of the results includes:

- The developed method and algorithms have been integrated into the handwriting alignment system, enabling the alignment of single-line handwritten text as well as multi-line texts on devices with limited computational resources.
- The implemented system ensures high accuracy of results, allowing the determination of correct positions for all symbols and their proper alignment. This was demonstrated by its utilization in a test mobile application for the Ukrainian handwritten text alignment.
- The handwriting alignment method presented in this work has been incorporated into the commercial application for the Samsung smartphones, serving as a separate feature for the alignment of the text in various languages.

Keywords: handwritten text alignment, handwriting processing, online digital ink recognition, symbols classification, structural analysis, handwriting segmentation, recurrent neural network, machine learning, artificial intelligence, intelligent input, human-computer interaction.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях України:

- [1] Yakovchuk, O., Vasin, M. (2023). Increasing the accuracy of handwriting text recognition in medical prescriptions with generative artificial intelligence. *Technology Audit and Production Reserves*, 4 (2 (72)), 18–22. DOI: <https://doi.org/10.15587/2706-5448.2023.284998>
- [2] Яковчук О.К. (2023). Побудова швидкої та легковісної рекурентної нейронної мережі для вирішення задачі розпізнавання рукописних жестів. *Таврійський науковий вісник. Серія: технічні науки*, (4), 87-93. DOI: <https://doi.org/10.32782/tnv-tech.2023.4.11>
- [3] Yakovchuk, O., Rogoza, W. (2024). An overview of statistical and neural-based line segmentation methods for offline handwriting recognition task. *Technology Audit and Production Reserves*, 1 (2 (75)), 14–19. DOI: <https://doi.org/10.15587/2706-5448.2024.298405>

Міжнародні патенти:

- [4] D. Zhelezniakov, O. Yakovchuk, D. Olynik, V. Zaitsev., Y. Yakishyn, A. Cherneha (2021). Electronic Device and Method for Modification of Content. WIPO PCT WO2021246570A1. Publication Date 2021.12.09.
<https://patents.google.com/patent/WO2021246570A1/en>
- [5] A. Cherneha, N. Sakhnenko, D. Zhelezniakov, O. Yakovchuk, V. Volkova, V. Zaitsev (2022). Method, Electronic Device and Storage Medium for Adjusting Document Style. Patent No. PN146948, Application Date 2022.12.02.

Статті у збірниках за матеріалами конференцій та тези доповідей:

- [6] D. Zhelezniakov, A. Cherneha, V. Zaytsev, T. Ignatova, O. Radyvonenko, and O. Yakovchuk. (2020). Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition. *25th International Conference on Intelligent User Interfaces (ACM IUI '20)*. 212–220. DOI: <https://doi.org/10.1145/3377325.3377482>
- [7] O. Yakovchuk, A. Cherneha, D. Zhelezniakov and V. Zaytsev. (2020). Methods for Lines and Matrices Segmentation in RNN-based Online Handwriting Mathematical Expression Recognition Systems. *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, Lviv, Ukraine. 255-261. DOI: <https://doi.org/10.1109/DSMP47368.2020.9204273>
- [8] K. Korovai, D. Zhelezniakov, O. Radyvonenko, O. Yakovchuk, I. Deriuga, N. Sakhnenko. (2023). Recognition-Independent Handwritten Text Alignment Using Lightweight Recurrent Neural Network. In *SIGGRAPH Asia 2023 Posters (SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 30, 1–2. DOI: <https://doi.org/10.1145/3610542.3626136>

ЗМІСТ

АНОТАЦІЯ.....	2
ANNOTATION.....	8
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	13
ЗМІСТ.....	15
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	18
ВСТУП	19
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ, АНАЛІЗ НАЯВНИХ ПІДХОДІВ ТА ЗАГАЛЬНА СТРУКТУРА ЗАПРОПОНОВАНОГО РІШЕННЯ	29
1.1 Проблеми вирівнювання рукописного тексту.....	29
1.1.1 Особливості рукописного тексту.....	29
1.1.2 Оцінка якості вирівнювання.....	31
1.2 Наявні підходи до вирівнювання та покращення рукописного тексту.....	32
1.2.1 Методи на основі статистичного підходу.....	33
1.2.2 Поліпшення рукописного тексту на основі токенів.....	35
1.2.3 Методи на основі синтезу рукописного тексту.....	36
1.2.4 Вирівнювання за допомогою рекурентної нейронної мережі	37
1.3 Існуючі методи розпізнавання рукописного тексту	39
1.3.1 Напрямки в розпізнаванні рукописного тексту.....	39
1.3.2 Основні підходи до онлайн розпізнавання рукописного тексту	40
1.3.3 Методи попередньої обробки вхідних даних	41
1.4 Існуючі методи сегментації рукописного тексту	42
1.4.1 Сегментація рядків.....	43
1.4.2 Сегментація символів	44
1.5 Постановка задач дослідження	46
1.6 Загальна структура запропонованого рішення	48
1.7 Висновки до розділу	49
РОЗДІЛ 2. РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЯ СИМВОЛІВ	51

2.1 Двонаправлена довга короткочасна пам'ять	51
2.1.1 Рекурентна нейронна мережа.....	51
2.1.2 Довга короткочасна пам'ять.....	53
2.1.3 Двонаправлена ДКЧП.....	53
2.1.4 Структура блоку ДКЧП	55
2.2 Нейромережева часова класифікація	58
2.2.1 Задача маркування послідовностей.....	58
2.2.2 Ідея нейромережевої часової класифікації	59
2.2.3 Кодування маркування послідовностей.....	60
2.2.4 Приклад роботи НЧК.....	62
2.3 Система розпізнавання та сегментації символів.....	63
2.3.1 Аналіз та обробка вхідних даних.....	63
2.3.2 Архітектура основної мережі класифікації символів	65
2.4 Покращення результатів сегментації символів	66
2.4.1 Додаткова мережа для збільшення точності сегментації.....	67
2.4.2 Динамічне програмування для інтерпретації результатів.....	68
2.4.3 Приєднання відтермінованих штрихів.....	70
2.4.4 Виправлення помилок сегментації	72
2.5 Висновки до розділу	73
<i>РОЗДІЛ 3. МЕТОД ВИРІВНЮВАННЯ З ВИКОРИСТАННЯМ СТРУКТУРНОГО</i>	
<i>АНАЛІЗУ СИМВОЛІВ</i>	75
3.1 Опис запропонованого методу вирівнювання	76
3.2 Структура розробленої системи вирівнювання	77
3.3 Попередня обробка вхідних даних	78
3.3.1 Побудова спрощених наборів точок.....	79
3.3.2 Побудова відповідностей наборів точок до кожного символу	81
3.3.3 Обрахування основних ознак вхідних штрихів.....	82
3.4 Метрики текстового рядка та розбиття символів на класи	84
3.4.1 Визначення класів символів.....	85
3.4.2 Уточнення класів символів	86
3.5 Визначення основних ліній для кожного символу.....	87
3.5.1 Визначення на основі класу символу	88

3.5.2	Структурний аналіз для визначення основних ліній	89
3.5.3	Аналіз локальних екстремумів	91
3.5.4	Аналіз сусідніх символів	93
3.5.5	Згладжування позицій символів в рядку	93
3.6	Підтримка зв'язного написання	94
3.7	Визначення результируючих метрик рядка	96
3.8	Визначення фінальної позиції та вирівнювання рядка	97
3.9	Висновки до розділу	99
РОЗДІЛ 4. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ		101
4.1	Опис експериментальних даних	101
4.2	Точність системи розпізнавання	102
4.3	Метрики для оцінювання вирівнювання тексту	104
4.4	Постановка обчислювальних експериментів	104
4.5	Аналіз результатів вирівнювання	106
4.5.1	Метрика за максимальною зміною відстаней	106
4.5.2	Метрика за середньоквадратичною похибкою зміни відстаней	107
4.5.3	Залежність від довжини рядка та кількості штрихів	108
4.5.4	Аналіз прикладів роботи системи	110
4.6	Практичне застосування та впровадження в мобільний застосунок	112
4.7	Оцінка продуктивності роботи системи на мобільному пристрої	113
4.8	Порівняння розробленої системи з наявними підходами	115
4.9	Висновки до розділу	116
ВИСНОВКИ		119
СПИСОК ІЛЮСТРАЦІЙ		122
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		125
Додаток А. ПРИКЛАДИ РОБОТИ СИСТЕМИ ВИРІВНЮВАННЯ		
РУКОПИСНОГО ТЕКСТУ		143
Додаток Б. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА		
ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ		146
Додаток В. АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ		148

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- МН – Машинне навчання (Machine Learning, ML)
- ГН – Глибинне навчання (Deep Learning, DL)
- СА – Структурний аналіз (Structural Analysis, SA)
- КЗ – Комп’ютерний зір (Computer Vision, CV)
- РВ – Рукописне введення (Handwriting Input, HI)
- ВРТ – Вирівнювання рукописного тексту (Handwritten Text Alignment, HTA)
- ОРС – Оптичне розпізнавання символів (Optical Character Recognition, OCR)
- РЦЧ – Розпізнавання цифрових чорнил (Digital Ink Recognition, DIR)
- 2Д – Двовимірний (Two-Dimensional)
- НМ – Нейронна мережа (Neural Network, NN)
- РНМ – Рекурентна нейронна мережа (Recurrent Neural Network, RNN)
- ВРВ – Вентильний рекурентний вузол (Gated Recurrent Unit, GRU)
- ДКЧП – Довга короткочасна пам’ять (Long Short-Term Memory, LSTM)
- ДДКЧП – Двонаправлена довга короткочасна пам’ять (Bidirectional Long Short-Term Memory, BLSTM)
- НЧК - Нейромережева часова класифікація (Connectionist Temporal Classification, CTC)
- Семпл – один зразок з набору вхідних даних (Sample)
- РДП – алгоритм Рамера-Дугласа-Пекера (Ramer–Douglas–Peucker algorithm, RDP)
- КСКП – коренева середня квадратична похибка (Root Mean Square Deviation, RMSE)

ВСТУП

Актуальність теми. У сучасному цифровому світі, де швидкість та зручність доступу до інформації стали важливими критеріями, роль рукописного введення та вирівнювання рукописного тексту набуває великого значення. Зі зростанням популярності сенсорних екранів та різноманітних периферійних пристроїв, які дозволяють введення за допомогою стилусів, актуальність цієї теми знаходиться в центрі уваги науковців та розробників. За статистикою, на сьогодні 6.92 мільярди людей є власниками смартфонів, що становить 85.74% всього населення. Ще не так давно, лише 7 років тому, це значення становило 49.4% від популяції планети [1].

Поява персональних мобільних пристроїв значною мірою сприяла активному дослідженню, просуванню та популяризації рукописних інтерфейсів, широко відомих як інтерфейси “олівець і папір” (“pencil and paper”). Такі інтерфейси пропонують природньо зрозумілу, просту та універсальну взаємодію з користувачем, дозволяючи створювати складні документи, що містять текстову інформацію, списки, таблиці, математичні вирази, діаграми, зображення та інші елементи. Розробники приділяють велику увагу розпізнаванню рукописного вмісту різних типів [2], визначенню намірів користувача та подальшій інтерпретації цих намірів [3]. В більшості випадків, основним результатом цієї обробки є перетворення рукописного введення в цифровий або друкований формат.

Рукописне введення стає ключовим елементом для взаємодії користувача з різноманітними пристроями, від смартфонів до планшетів та персональних комп'ютерів. Замість традиційного клавішного введення, рукописне введення дозволяє виражати ідеї та думки в найбільш природний спосіб – за допомогою власного почерку [4]. Це відкриває широкі можливості для творчості та особистого вираження, особливо в цифровому середовищі.

Задача цифрової обробки рукописного тексту розділяється на два основні напрямки – оптичне розпізнавання символів (Optical Character Recognition) [5] та розпізнавання цифрових чорнил (Digital Ink Recognition) [6]. Перший підхід базується на обробці сторінки рукописного документу, що представлена у вигляді набору пікселів, тобто виконує статичну обробку графічних зображень. Другий підхід працює з представленням рукописного документу у вигляді послідовності точок у двовимірному просторі, що описують траєкторії цифрового пера, тобто виконує обробку часових послідовностей точок. Розпізнавання рукописного введення з другим підходом також називається онлайн розпізнаванням. Кожен з підходів має свої складнощі та виклики, проте другий підхід надає більше інформації про рукописний текст і дозволяє виконувати більш складні задачі [7]. В даній роботі досліджується другий підхід по обробці рукописної інформації, яка отримується за допомогою цифрових чорнил.

Глобальний ринок цифрового пера у 2021 році оцінили у 2,04 мільярда доларів, і спрогнозували зростання за рівнем сукупного середньорічного темпу зростання (CAGR) в 13,3% від 2022 до 2030 року [8]. Використання цифрових чорнил набуває значної популярності по всьому світу завдяки впровадженню електронних сайтів та додатків, що використовують технологію рукописного введення. Прикладами можуть бути індустрії анімації, творчих мистецтв, сфери медицини, науки, освіти. Так, освітяни намагаються забезпечити навчання через онлайн-середовище, тим самим надаючи більше зручності студентам порівняно з традиційним офлайн-навчанням. Використання традиційної дошки часто не є ефективним, наприклад через низьку видимість написаного контенту. Таким чином, використання цифрового пера як альтернативи крейді або маркеру в результаті підвищує чіткість та ясність написаного тексту, що дає змогу отримувати інформацію з екрану легше та швидше [9].

Проводяться різноманітні дослідження по зручності використання рукописного введення [10], зокрема в роботі [3] було досліджено використання рукописного введення для вирішення повсякденних задач у смартфоні, наприклад створення

контакту, замітки в календарі, обрахунки в калькуляторі, виставлення таймера, будильника тощо. Порівняно середній час виконання задач за допомогою традиційного введення, рукописного введення та за допомогою голосу. Результати експериментів та опитування респондентів показали, що в порівнянні з традиційним введенням інформації, рукописне введення пришвидшує виконання поставлених задач на 20%, а також сприяє соціальному комфорту під час роботи [3].

Разом із зростанням популярності рукописного введення постає завдання ефективної обробки рукописної інформації, зокрема рукописного тексту. Однією із задач є забезпечення точності та структурної коректності тексту, написаного в рукописному вигляді. Зокрема, є актуальною проблема вирівнювання рукописного тексту, при цьому зберігаючи формат, структуру текстових елементів, їх текстовий зміст [11]. Важливим є також збереження ідентичності та унікальності написаного тексту, що надається йому автором за допомогою персональних особливостей написання та почерку. Водночас створювані рішення повинні мати змогу працювати в умовах обмеження обчислювальних ресурсів, на персональних пристроях, таких як смартфони, розумні годинники, інтерактивні дошки, монітори з рукописним введенням та інші гаджети.

Основними проблемами в задачах обробки рукописного тексту є в першу чергу сама природа рукописного введення інформації, що має велику варіативність вхідних даних та є унікальною для кожного користувача. Індивідуальний рівень професійних навичок рукописного введення, швидкість написання, наявність помилок, персональні особливості написання – всі ці параметри впливають на вхідні дані, які повинні бути враховані та коректно опрацьовані для ефективного вирішення задачі вирівнювання тексту [12].

У даній дисертаційній роботі акцент робиться на задачі вирівнювання рукописного тексту зі збереженням початкового стилю написання, без перетворення рукописного тексту в друковану форму. Існуючі підходи перетворення рукописного тексту в друковану форму шляхом розпізнавання мають потенційні проблеми, що

пов'язані з відсутністю рішень, які надають гарантовану максимальну точність розпізнавання. Таким чином, помилка в перетвореному тексті є цілком імовірною, в той же час виконання вирівнювання зі збереженням рукописної форми тексту є більш надійним варіантом з меншою ймовірністю виникнення помилки під час перетворення, навіть за умови використання результатів розпізнавання для проведення вирівнювання, оскільки запропонований метод використовує лише часткову інформацію з розпізнавання.

Актуальність роботи полягає у відсутності існуючих рішень для ефективного вирівнювання рукописного тексту зі збереженням початкового стилю написання, підтримкою зв'язного написання символів, індивідуальних особливостей написання різних символів, знаходженням результуючих метрик рядка. Перетворення рукописного тексту користувача у вирівняну форму покращує сприйняття власного рукописного введення, збільшує швидкість та легкість читання написаного тексту, поліпшує естетичність та загальний зовнішній вигляд рукописного документу. Також вирівнювання рукописного тексту може використовуватись для зменшення загального розміру рукописного тексту в документі або нотатці, шляхом зменшення як висоти кожного рядка під час вирівнювання, так і відступів між рядками за рахунок ущільнення вирівняних рядків. Це дозволяє більш ефективно використовувати простір в текстових документах та отримати додаткове вільне місце.

В дослідженні використовуються передові методи та технології для обробки рукописного тексту, включно з розпізнаванням та сегментацією тексту. Дослідження в напрямках обробки рукописного тексту, разом з візуальним покращенням тексту для збільшення ясності та чіткості, мають важливе значення для подальшого вдосконалення систем рукописного введення та їх використання в різних сферах, від освітнього до професійного використання. Розвиток вирівнювання рукописного тексту відкриває нові перспективи для зручної та ефективної роботи з рукописною інформацією, що стає невід'ємною частиною сучасного цифрового життя.

Аналіз існуючих підходів та методів до візуального поліпшення та вирівнювання рукописного тексту показав, що на сьогодні залишається невирішеною задача ефективного вирівнювання рукописного тексту з врахуванням особливостей різних почерків, підтримкою зв'язного написання, знаходженням результуючих метрик рядка, збереженням початкового стилю написання. Не розглянуто методи вирівнювання рукописного тексту, що використовують розпізнавання та сегментацію рукописних символів. Не розглянуто методи, що виконують вирівнювання тексту на основі переміщення рукописних штрихів з врахуванням маркування вхідної послідовності на символи. Не розглянуто методи вирівнювання рукописного тексту, написаного українською мовою.

Таким чином, дисертаційна робота присвячена вирішенню важливого **наукового завдання**, що має технічну спрямованість в галузі інформаційних технологій, а саме: розробка методів вирівнювання рукописного тексту на основі розпізнавання та сегментації тексту з використанням структурного аналізу символів.

Дисертаційна робота виконана відповідно до планів наукової і науково-технічної діяльності Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» і є частиною досліджень у межах науково-дослідних робіт кафедри системного проектування, а також в рамках вирішення поставлених задач у підрозділі із досліджень та розробки в галузі інтелектуального введення інформації в компанії «Самсунг РнД Інститут Україна».

Мета і завдання дослідження. Метою дисертаційного дослідження є підвищення ефективності вирішення комплексу задач, що виникають при вирівнюванні рукописного тексту.

Для досягнення поставленої мети в роботі виконуються наступні завдання:

1. Провести аналіз існуючих підходів до поліпшення та вирівнювання рукописного тексту, визначити їх основні переваги та недоліки.

2. Дослідити сучасні методи розпізнавання та сегментації рукописного тексту, визначити ті, які дадуть найвищу точність класифікації та маркування вхідної послідовності символів.
3. Вдосконалити метод сегментації рукописного тексту для підвищення точності маркування кожного символу.
4. Розробити метод вирівнювання рукописного тексту на основі розпізнавання та сегментації тексту з використанням структурного аналізу символів.
5. Виконати програмну реалізацію представлених методів у вигляді інструментальної системи вирівнювання для розв'язування задачі вирівнювання рукописного тексту українською мовою.
6. Виконати експериментальне дослідження параметрів та оцінювання ефективності роботи системи вирівнювання рукописного тексту на мобільних пристроях з обмеженнями обчислювальних ресурсів.

Об'єкт дослідження. Об'єктом дисертаційного дослідження є процеси розпізнавання, сегментації та вирівнювання рукописного тексту.

Предмет дослідження. Предметом дисертаційного дослідження виступають методи для розпізнавання, сегментації та вирівнювання рукописного тексту.

Методи дослідження. Дисертаційне дослідження ґрунтується на системному аналізі результатів сучасних теоретичних і прикладних розробок у сфері інформаційних технологій. В дослідженні використовуються інтелектуальні методи обробки даних, теорія штучних нейронних мереж, методи статистичного аналізу, структурного аналізу, обчислювальної геометрії, машинного навчання, динамічного програмування. Описані методи були обрані з урахуванням їх практичності для використання при вирішенні задач розпізнавання, сегментації та обробки рукописного тексту.

Наукова новизна отриманих результатів:

- Вперше запропоновано метод вирівнювання рукописного тексту, наукова новизна якого полягає у використанні розроблених алгоритмів розпізнавання та сегментації тексту, які базуються на теорії машинного навчання, а також у проведенні структурного аналізу символів, що враховує індивідуальні особливості почерку, зв'язне написання символів та дозволяє зберегти початковий стиль написання тексту.
- Вдосконалено метод сегментації рукописного тексту, що ґрунтується на використанні легковісної рекурентної нейронної мережі та динамічного програмування, який відрізняється від існуючих застосуванням додаткових алгоритмів, які дають змогу знаходити та виправляти помилки сегментації, визначати відповідні символи для відтермінованих штрихів, що дозволяє підвищити точність маркування кожного символу.
- Вперше розроблено метод вирівнювання для тексту, написаного українською мовою, наукова новизна якого полягає у використанні алгоритмів, що базуються на структурному аналізі символів, та визначаються окремо для кожного символу, а також у можливості реалізувати вирівнювання рукописних літер для різних стилів написання.

Практичне значення отриманих результатів:

- Розроблені метод та алгоритми впроваджено в інструментальну систему вирівнювання рукописного тексту, що дає змогу розв'язувати задачі вирівнювання одного рядка рукописного тексту та текстів, що складаються з багатьох рядків, на пристроях з обмеженнями обчислювальних ресурсів.
- Розроблена система забезпечує високу точність результатів, що дозволяє використовувати її для знаходження правильних позицій всіх символів та їх коректного вирівнювання, що продемонстровано на прикладі її використання в

тестовому мобільному додатку для вирівнювання рукописного тексту українською мовою.

- Представлений у цій роботі метод вирівнювання рукописного тексту впроваджений у комерційний додаток для смартфонів Samsung у вигляді повноцінної функції вирівнювання для різних мов.

Практичне значення результатів роботи підтверджено актом впровадження в науково-дослідницькому центрі «Самсунг РнД Інститут Україна» (акт від 15.03.2024, Додаток В).

Особистий внесок здобувача.

Усі основні результати дисертаційного дослідження, представлені до захисту, одержані автором особисто. У публікаціях у співавторстві здобувачеві належать: дослідження існуючих підходів та методів, розробка методів та алгоритмів вирівнювання рукописного тексту на основі розпізнавання та сегментації, розробка додаткових інструментів для відлагодження та тестування розроблених методів, проведення експериментів та оцінювання ефективності розроблених методів.

Апробація матеріалів дисертації.

Основні положення та отримані наукові результати, що викладені в даній дисертаційній роботі, пройшли апробацію на трьох міжнародних науково-технічних конференціях:

“25th International Conference on Intelligent User Interfaces (ACM IUI), Cagliari, Italy, March 17–20, 2020”;

“IEEE Third International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, August 21–25, 2020”;

“16th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (ACM SIGGRAPH-Asia), Sydney, Australia, December 12–15, 2023”.

Публікації.

Статті у наукових фахових виданнях України:

[1] Yakovchuk, O., Vasin, M. (2023). Increasing the accuracy of handwriting text recognition in medical prescriptions with generative artificial intelligence. *Technology Audit and Production Reserves*, 4 (2 (72)), 18–22. DOI: <https://doi.org/10.15587/2706-5448.2023.284998>

[2] Яковчук О.К. (2023). Побудова швидкої та легкової рекурентної нейронної мережі для вирішення задачі розпізнавання рукописних жестів. *Таврійський науковий вісник. Серія: технічні науки*, (4), 87-93. DOI: <https://doi.org/10.32782/tnv-tech.2023.4.11>

[3] Yakovchuk, O., Rogoza, W. (2024). An overview of statistical and neural-based line segmentation methods for offline handwriting recognition task. *Technology Audit and Production Reserves*, 1 (2 (75)), 14–19. DOI: <https://doi.org/10.15587/2706-5448.2024.298405>

Міжнародні патенти:

[4] D. Zhelezniakov, O. Yakovchuk, D. Olynik, V. Zaitsev., Y. Yakishyn, A. Cherneha (2021). Electronic Device and Method for Modification of Content. WIPO PCT WO2021246570A1. Publication Date 2021.12.09. <https://patents.google.com/patent/WO2021246570A1/en>

[5] A. Cherneha, N. Sakhnenko, D. Zhelezniakov, O. Yakovchuk, V. Volkova, V. Zaitsev (2022). Method, Electronic Device and Storage Medium for Adjusting Document Style. Patent No. PN146948, Application Date 2022.12.02.

Статті у збірниках за матеріалами конференцій та тези доповідей:

[6] D. Zhelezniakov, A. Cherneha, V. Zaytsev, T. Ignatova, O. Radyvonenko, and O. Yakovchuk. (2020). Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition. *25th International Conference on Intelligent User Interfaces* (ACM IUI '20). 212–220. DOI: <https://doi.org/10.1145/3377325.3377482>

[7] O. Yakovchuk, A. Cherneha, D. Zhelezniakov and V. Zaytsev. (2020). Methods for Lines and Matrices Segmentation in RNN-based Online Handwriting Mathematical Expression Recognition Systems. *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, Lviv, Ukraine. 255-261. DOI: <https://doi.org/10.1109/DSMP47368.2020.9204273>

[8] K. Korovai, D. Zhelezniakov, O. Radyvonenko, O. Yakovchuk, I. Deriuga, N. Sakhnenko. (2023). Recognition-Independent Handwritten Text Alignment Using Lightweight Recurrent Neural Network. In *SIGGRAPH Asia 2023 Posters (SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 30, 1–2. DOI: <https://doi.org/10.1145/3610542.3626136>

Структура та обсяг дисертації. Дисертаційна робота складається з анотації, вступу, чотирьох розділів, висновків, списку ілюстрацій, списку використаних джерел та додатків. Загальний обсяг дисертації становить 148 сторінок, у тому числі: 103 сторінки основного тексту, 36 рисунків, 6 таблиць, список використаних джерел із 142 найменувань на 18 сторінках.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ, АНАЛІЗ НАЯВНИХ ПІДХОДІВ ТА ЗАГАЛЬНА СТРУКТУРА ЗАПРОПОНОВАНОГО РІШЕННЯ

Розділ присвячений аналізу поточного стану справ у основних темах, на яких фокусується дана дисертаційна робота. Описуються проблеми, що виникають в задачах обробки та покращення рукописного введення, які базуються на особливостях природи рукописних вхідних даних. Окремо розглядається задача оцінювання якості вирівнювання для подальшого коректного обґрунтування отримуваних результатів.

Проводиться ґрунтовний аналіз наявних підходів до поліпшення та вирівнювання рукописного тексту, визначаються основні переваги та недоліки існуючих рішень. Також аналізуються існуючі методи для виконання задач обробки вхідних даних, розпізнавання рукописного введення, сегментації рукописного тексту на рядки та сегментацію на символи, що є важливими частинами даного дослідження. Обґрунтовується вибір тих чи інших підходів до виконання поставлених задач.

Цей розділ включає в себе формулювання постановки задачі вирівнювання рукописного тексту, опис вхідних даних та встановлення основних вимог до розробленого рішення. Далі представляється загальна структура запропонованого рішення, яка описує основні складові та компоненти системи.

1.1 Проблеми вирівнювання рукописного тексту

1.1.1 Особливості рукописного тексту

Вирівнювання рукописного тексту, як одна зі складових обробки рукописної інформації, стикається з рядом викликів та проблем, які потребують уваги та нових ефективних методів вирішення. Ключові проблеми задачі вирівнювання рукописного тексту включають:

- Варіативність почерку. Різноманіття почерку серйозно ускладнює процес вирівнювання. Люди пишуть з різною швидкістю, стилем, нахилом, напрямком, що створює велику різноманітність форм та розмірів символів. Часто при

написанні до символів додаються особливі частини, такі як додаткові штрихи, гачки, заокруглення і т.д., що створюють унікальність та неповторність кожного почерку [12].

- Відстані між символами. Один і той самий символ може бути написаний різними людьми з великою варіативністю у відстанях між сусідніми символами. Це викликає проблеми в точному визначенні границь між символами під час вирівнювання [13].
- Зв'язне написання символів. Написання декількох символів разом, без відриву пера від поверхні, створює проблему з визначенням кордонів між символами, правильного положення цього набору символів, і потребує аналізу кожного випадку окремо.
- Вплив помилок техніки письма. Поспіх, помилки та непрофесійність в техніці письма можуть призводити до неправильностей у формі та структурі символів, що робить неочевидним їх коректне вирівнювання [14].
- Низька якість вхідних даних. Неправильне зчитування даних з сенсорних екранів чи планшетів може призводити до неточностей у визначенні координат точок та їх послідовності, утворенні додаткових хвостів та гачків, що ускладнює як обробку написання в цілому, так і його вирівнювання [15].
- Навколишні умови написання. Фактори, такі як освітлення, положення руки, або наявність перешкод можуть впливати на процес написання та змінювати характеристики написаного тексту [16].

Вирішення цих проблем вирівнювання рукописного тексту стає актуальним завданням, що вимагає розробки нових методів та технологій для підвищення точності та швидкості обробки рукописних даних.

1.1.2 Оцінка якості вирівнювання

Окремим питанням є оцінка результатів вирівнювання та аналіз якості роботи методів та алгоритмів. Немає однозначної метрики або параметра, який міг би оцінити рівність тексту або його розбірливість.

До цього питання частково відноситься поняття каліграфії - мистецтво писати розбірливим, гарним і чітким почерком, також каліграфією називають досконалу форму рукописного шрифту книг та документів. Проте в більшості випадків приклади каліграфії дуже відрізняються від звичайного тексту, який пишуть люди в повсякденному житті, і тому не завжди можуть вважатись єдиним правильним еталоном.

Зазвичай більшість людей навчається навичкам рукопису в школі, де їх вчать відтворювати рукопис по абетці, а також писати великі речення та тексти. Протягом цього часу формується базовий стиль написання, а в наступні роки він остаточно формується в унікальний набір особливостей рукописного тексту, що є індивідуальним почерком людини. В багатьох випадках особливості написання певних символів та літер можуть відрізнятись від загальноприйнятого стилю, що викликає питання щодо правильності та коректності таких прикладів.

В дослідженні [17] проводиться аналіз рукописного тексту вчителів молодших класів, що навчають учнів рукопису. Пропонується використання наступних параметрів рукописного тексту для оцінки його розбірливості: правильність написання символів (кількість допущених помилок), розмір літер, нахил рукописного тексту, міжсимвольні зв'язки, розтягування літер, візуальна видимість горизонтальних ліній тексту, відстані між словами. Кожен параметр оцінюється по шкалі від 1 до 3, відповідно отримання найбільшої оцінки 3 по всім параметрам буде визначати найкращу розбірливість рукописного тексту.

Робота [9] пропонує використовувати багатовимірну шкалу розбірливості (Multidimensional Legibility Scale), в [4] та [18] використовується шкала розбірливості рукописного тексту (Handwriting Legibility Scale), що є схожою. Вона оцінює рукопис

за 5 основними пунктами з оцінкою від 1 до 5: розбірливість, що визначається як загальне враження після першого прочитання тексту; зусилля, що витрачаються для першого прочитання тексту; загальна організація документу та компоновка, тобто позиціонування елементів один відносно одного; форма літер, що включає в себе консистентність в розмірі та нахилі; необхідність змін, що визначається як кількість елементів які потребують модифікації штрихів, розмірів, положення.

1.2 Наявні підходи до вирівнювання та покращення рукописного тексту

На сьогодні досягнуто значних успіхів у дослідженнях підходів та методів по обробці рукописної інформації.

Обробка рукописного тексту включає в себе велику кількість різних задач та сфер, в багатьох з них за останні роки досягнуто вражаючих результатів. Прикладами є розпізнавання тексту [19], [20], класифікація систем письма [21], ідентифікація автора [22], класифікація штрихів [23], розпізнавання математичних виразів [24], [25], [26], валідація підписів [27], [28], визначення захворювання на основі рукописного тексту [29], [30], розпізнавання та покращення рукописних ескізів [11], [31], [32], та інші.

Існує великий обсяг досліджень спрямованих на вдосконалення рукописного написання, з метою зробити його більш розбірливим та візуально привабливим для людського ока. Дослідники підходять до вирішення цієї задачі з різних точок зору, включаючи модифікацію структури, форми, позицій рукописних штрихів, стилізацію штрихів, виправлення перекосу та нахилу рядків, використовуючи техніки б'ютифікації тексту (text beautification), генерації рукописного тексту з іншими стилями. Окрім покращення рукописного тексту, розглядаються також методи візуального поліпшення інших типів рукописного введення, таких як фігури, блоки, діаграми і т.д. [33].

1.2.1 Методи на основі статистичного підходу

Початково спроби покращити рукописне введення полягали у встановленні набору суворих правил та обмежень до вхідних рукописних штрихів та послідовному коригуванні відповідно до цих обмежень, коли це можливо [34], [35]. Однак автори визнали, що використання статистичних методів для покращення рукописного введення довільних форм призводить до частих помилок, що дає небажані результати на практиці. В роботі [36] пропонується послідовно деформувати вхідні штрихи, використовуючи задані базові фігури та шаблони, для перетворення у ідеальні геометричні форми. Робота [37] пропонує виконувати пошук базової лінії тексту на основі вертикальної гістограми щільності точок. Інші техніки для покращення рукописного тексту пропонують згладжувати вхідні штрихи, зберігаючи при цьому основні виразні деталі [38].

Автори дослідження [39] пропонують використовувати клотоїдні сплайни [40] для представлення рукописних штрихів. Для кожної частини рукописного штриха застосовується пошук найкоротшого шляху в графі примітивів, що представлені клотоїдними сплайнами, які повинні забезпечувати відповідність всім підчастинам вхідних штрихів. Після цього вирішується задача нелінійної оптимізації для підставлення знайденого примітиву до конкретного штриха. Приклад роботи методу представлено на рис. 1.1. На думку авторів, незважаючи на те, що клотоїдні сплайни є математично та алгоритмічно складнішими за поліноміальні сплайни, результуюча вища точність представлення переважає зазначений недолік.

Автори дослідження [41] представили метод для генерації виразного та чіткого рукописного тексту та малюнків за допомогою використання набору віртуальних пензлів. Велика кількість робіт зосереджуються на корекції нахилу та повороту рукописного тексту методами комп'ютерного зору, так [42] та [43] застосовують перетворення Хафа, а в [44] використовується перетворення Радона.

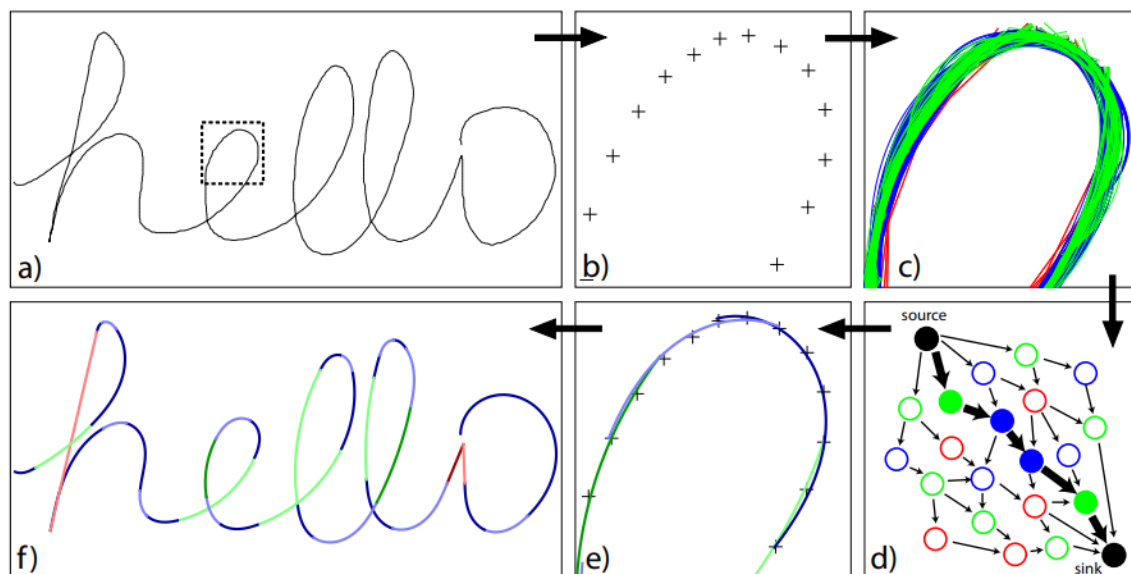


Рис. 1.1 – Покрокова візуалізація покращення рукописного тексту шляхом представлення рукописних штрихів через клотоїдні сплайни [39]

Інша робота [45] пропонує класифікувати екстремуми рукописних штрихів для знаходження ключових точок символів та використовувати методи геометричної деформації для покращення рукописного тексту. Структура запропонованої системи вирівнювання та приклад з результатом роботи наведено на рис. 1.2.

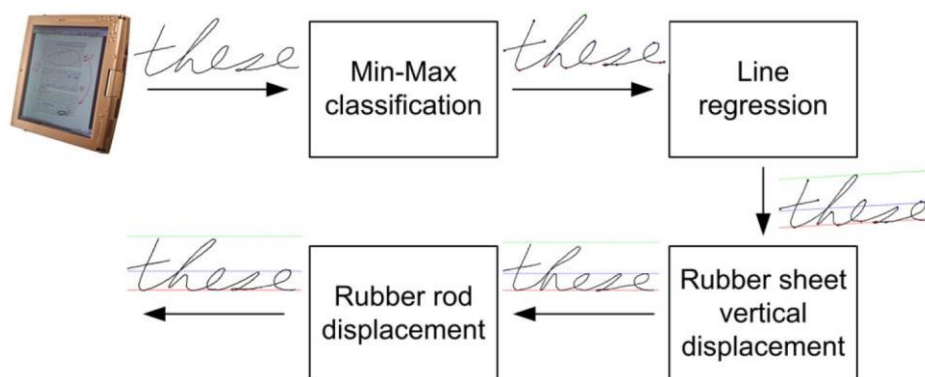


Рис. 1.2 – Структура системи та приклад вирівнювання на основі класифікації екстремумів та геометричних деформацій [45]

Метод вирівнювання, що пропонується в даній роботі, не використовує результат розпізнавання, і тому не має інформації про символи вхідного рядка. Автори

обґрунтовують це складністю точного вирішення задачі розпізнавання та сегментації рукописних символів. Для класифікації екстремумів на 5 класів в роботі запропонована архітектура згорткової нейронної мережі, отримана точність класифікації становить 86%. Наступним кроком виконується геометрична деформація символів по знайденим критичним точкам та знаходження текстових метрик вирівняного рядка. Основною проблемою даного підходу є відсутність інформації про символи, що вирівнюються, це негативно впливає на загальну якість вирівнювання, особливо помітно це може бути з різноманітним написанням символів з нижньою частиною. Також в результаті вирівнювання відбувається деформація вхідних штрихів, через це початковий стиль написання та почерк користувача можуть бути порушені.

1.2.2 Поліпшення рукописного тексту на основі токенів

Дослідження [47] по візуальному поліпшенню рукописного введення використовує розбиття всіх штрихів вхідного тексту на окремі складові частини - токени, які являються простими формами та фігурами, з яких складаються всі символи. Ідея полягає в кластеризації всіх токенів для групування їх за схожими ознаками. Далі, створивши набір “ідеальних” токенів, будь-який новий символ може бути візуально покращений, за допомогою модифікації його складових токенів через пошук найбільш підходящих з готового набору. Приклади результатів наведені на рис.

1.3. Перевагами даного підходу є можливість зберігати оригінальний стиль написання, проте основним недоліком, як зазначають автори, є необхідність мати досить великий набір готових “ідеальних” токенів, які будуть покривати різноманітні стилі написання та почерки, інакше для нових вхідних штрихів може бути не знайдено відповідний токен.

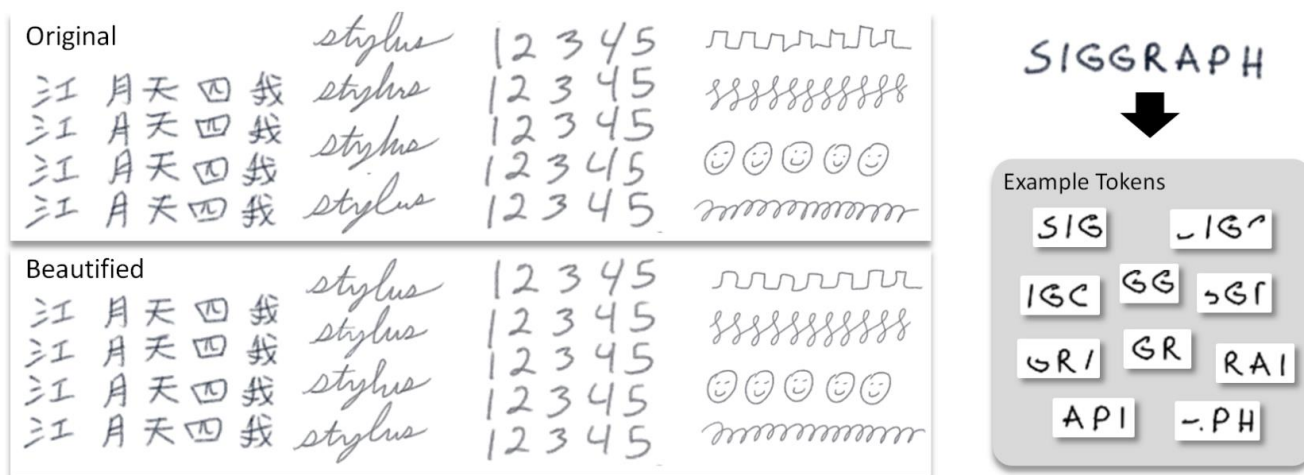


Рис. 1.3 – Приклади візуального покращення рукописних елементів та тексту за допомогою розбиття на токени [47]

1.2.3 Методи на основі синтезу рукописного тексту

В останній час було розроблено та представлено нові підходи, які зосереджуються навколо синтезу рукописного почерку, що також дають змогу контролювати стиль згенерованого тексту. Автори [46] представили архітектуру умовно-варіаційної рекурентної нейронної мережі для генерації рукописного тексту. Ця мережа призначена для надання згенерованому рукописному тексту конкретного заданого стилю, або перенесення унікального стилю користувача з одного тексту на інший. Таким чином пропонується надавати користувачам контроль на стилем написаного тексту з можливістю змінювати стиль як всього тексту, так і окремих його слів, покращуючи загальну виразність, рівність та зовнішній вигляд. Приклад роботи цієї мережі приведено на рис. 1.4. Варто зауважити, що даний підхід з генерацією тексту потребує попереднього розпізнавання тексту. Важливими недоліками даного підходу є часткова зміна або повна втрата початкового стилю написання користувача в результаті перетворення, погіршення якості результатів у випадку зв'язного написання символів. Також наведені результати не завжди дають рівномірний та покращений вигляд рукописного тексту через потенційні помилки та викиди під час роботи нейронної мережі.

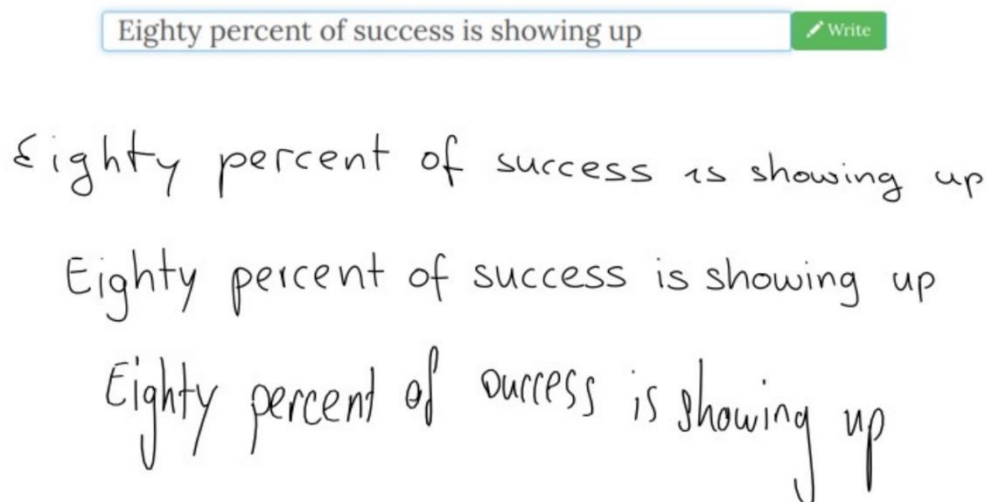


Рис. 1.4 – Приклад генерації рукописного тексту у різних заданих стилях за допомогою умовно-варіаційної рекурентної нейронної мережі [46]

Інша запропонована архітектура для зміни стилю рукописного тексту містить три рекурентні нейронні мережі: кодувальник вмісту, кодувальник стилю та декодувальник [48]. Незважаючи на те, що такі генеративні моделі мають потенціал для покращення рівності та чіткості тексту за рахунок заміни стилю на більш розбірливий, проте оригінальний стиль користувача може бути втрачений у процесі модифікації.

Підхід до покращення рукописного тексту шляхом очищення шуму за допомогою рекурентної нейронної мережі представлено в [49]. Основна ідея полягає в тренуванні нейронної мережі прибирати штучно згенеровані шуми під час навчання. Це дає можливість мережі модифікувати та покращувати рукописний текст, що подається на вхід, в різних аспектах, що будуть залежати від типу деформацій застосованих на етапах генерації штучного набору даних для навчання.

1.2.4 Вирівнювання за допомогою рекурентної нейронної мережі

В роботі [50] пропонується легковісна рекурентна нейронна мережа для вирівнювання рукописного тексту без використання інформації про результат розпізнавання. Основна ідея базується на припущенні, що нейронна мережа повинна

бути здатною самостійно визначати всі взаємозв'язки у вхідних даних, та проводити вирівнювання штрихів по базовій лінії без знання про сегментацію символів та самі символи.

Архітектура мережі визначається двошаровою двонаправленою моделлю GRU з одним прихованим шаром (рис. 1.5). Кількість тренувальних параметрів в мережі лише 40000, що дозволяє вважати таку модель дійсно легкою та швидкою.

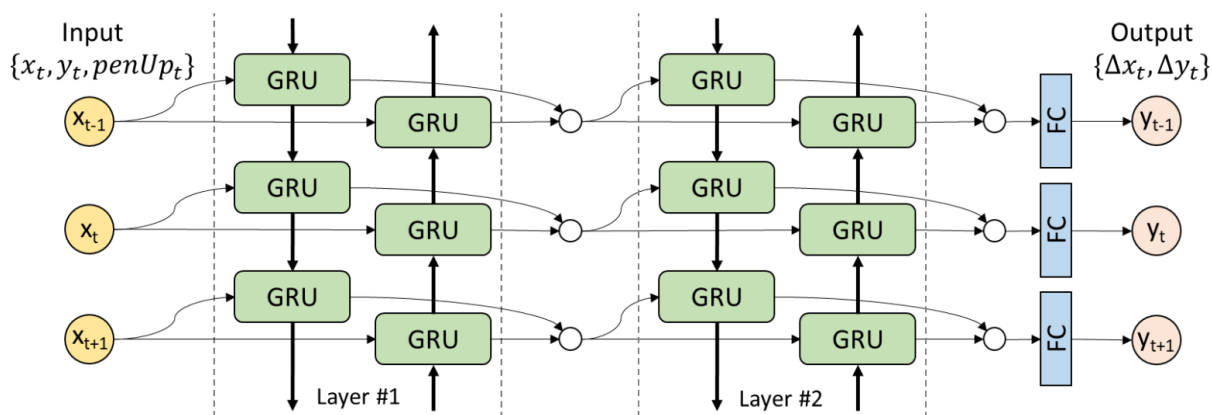


Рис. 1.5 – Архітектура запропонованої рекурентної нейронної мережі [50]

Основною проблемою цього підходу є необхідність мати тренувальний набір даних, що буде складатися з пар семплів (семпл – зразок з набору даних, що містить дані рукописного введення), один з яких є вхідним невирівняним набором штрихів, а інший повинен мати всі ті самі штрихи, переміщені у вирівняні позиції. У роботі визначається розмір тренувального набору даних в 530 тисяч семплів, що вказує на необхідність мати окрему готову систему для генерації такого набору даних. Відповідно, результат роботи натренованої нейронної мережі буде напряму залежати від якості згенерованих даних, тобто від якості роботи системи для його генерації. В наведених прикладах неуспішного вирівнювання помітні зміщення окремих символів та слів, які можуть бути викликані особливостями набору даних (рис. 1.6).

Ще одним недоліком даного підходу можна відмітити відсутність фіксованого результату, оскільки кожен наступний запуск нейронної мережі вирівнювання буде виконувати додаткову модифікацію та зміщення штрихів.

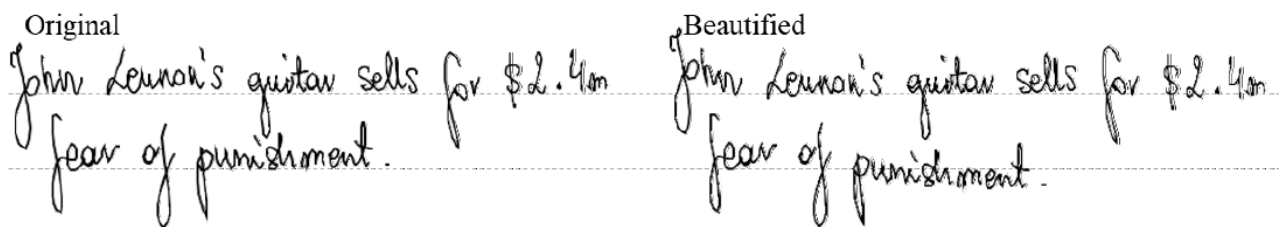


Рис. 1.6 – Приклади неуспішного вирівнювання за допомогою рекурентної нейронної мережі [50]

1.3 Існуючі методи розпізнавання рукописного тексту

В даній дисертаційній роботі пропонується підхід для вирівнювання, що використовує результати розпізнавання та сегментації рукописного тексту, тому було розглянуто передові дослідження в цих областях.

1.3.1 Напрямки в розпізнаванні рукописного тексту

В обробці рукописного введення зазвичай виділяють два основні підкласи цієї задачі: онлайн та офлайн обробка рукописних даних. Онлайн-розпізнавання використовує динамічне представлення рукописного введення у вигляді цифрових слідів від пера або рухів пальців. Офлайн-розпізнавання працює зі статичним представленням вхідного документа, наприклад зображенням, що може містити текст та інші види інформації. Також розглядаються комбіновані підходи розпізнавання [51].

Важливо зазначити, що методи розроблені для одного підходу можуть бути перенесені та застосовані і на інший підхід. Так, набір вхідних штрихів цифрових чорнил може бути конвертований у статичний формат за допомогою перетворення у піксельну форму, та нанесений на зображення з певним встановленим фоном. Також можливо використати онлайн методи розпізнавання для офлайн режиму. В роботі [52] розроблено алгоритм перетворення зображення у рукописні штрихи для подальшого використання онлайн методів. Ідея методу базується на проведенні бінаризації, скелетування, сегментації, реконструкції штрихів та фінальної нормалізації порядку

штрихів. Важливо, що для успішного застосування онлайн методів розпізнавання з такими перетвореними даними потрібно буде провести перенавчання моделі на цих нових даних.

В даній роботі ми зосереджуємось на онлайн методах обробки та розпізнавання рукописного введення.

1.3.2 Основні підходи до онлайн розпізнавання рукописного тексту

Дослідження напрямку онлайн розпізнавання рукописного тексту продовжується вже достатньо довгий час та має велику кількість публікацій та наукових праць. Оглядові роботи, такі як [53], [54] підсумовують загальний шлях розвитку технології. В дослідженнях [55], [56] визначається сучасний стан справ та найкращі передові рішення.

Розглядаються два основні підходи до розпізнавання рукописного тексту:

- 1) Класифікація символів на основі попередньої сегментації. Прикладами рішень на основі цього підходу є роботи [57], [58]. Основна ідея базується на проведенні сегментації вхідних штрихів для розбиття їх на окремі сегменти, що далі подаються на системи класифікації. Один з варіантів реалізації системи класифікації є використання нейронних мереж зі зворотнім поширенням помилки для розпізнавання символів по відсегментованим частинам. Дослідження [59] пропонує модель передачі токенів для аналізу результатів сегментації та декодування послідовності символів. Основним слабким місцем цього підходу в загальному є його акцентованість на відокремленості символів між собою у вхідних послідовностях, таким чином під час обробки інформації може не враховуватись важлива інформація про зв'язки у послідовностях.
- 2) Класифікація на основі аналізу часових рядів. Цей напрямок включає в себе використання прихованих марковських моделей (ПММ) [60], [61], [62], нейронних моделей з часовою затримкою (TDNN) [63], умовних випадкових полів (CRF) [64] та рекурентних нейронних мереж (PHM) [66], [65]. Останній

підхід є одним з найбільш цікавих для цієї задачі, особливо використання архітектури довгої короткочасної пам'яті [67], що дає досить високу точність розпізнавання та перевершує інші рішення [68].

В роботі [58] також допускається комбінація цих двох підходів, при якій на початковому етапі відбувається сегментація штрихів, а далі для їх класифікації використовується нейронна модель з часовою затримкою. В роботі [69] досліджуються дифузійні алгоритми та проблеми оптимізації структурних алгоритмів розпізнавання.

На основі проведеного аналізу, в цій роботі пропонується використовувати підхід оснований на РНМ, а саме двонаправлену РНМ з нейромережевою часовою класифікацією (НЧК) [93]. Дане рішення обумовлене головною особливістю моделей двонаправлених РНМ зберігати та використовувати контекстну інформацію протягом тривалого періоду часу, беручи до уваги як попередні, так і наступні частини вхідних послідовностей. Розглянуті сучасні дослідження демонструють можливість даного методу успішно виконувати такі задачі як класифікація текстів [51], розпізнавання мовлення [70], тексту [71]. А використання функції оцінки НЧК дає змогу вірно передбачати розташування символів у вхідних послідовностях та відокремлювати повторювані символи за допомогою введення додаткового роздільного символу.

1.3.3 Методи попередньої обробки вхідних даних

Важливим етапом в кожному з підходів є попередня обробка вхідних даних. Сучасні роботи [72], [73] пропонують та перевіряють різні комбінації та способи передобробки даних для отримання найкращих результатів розпізнавання. Найбільш поширеними кроками передобробки є нормалізація розмірів, щільності точок, корекція повороту текстових ліній, нахилу символів, проведення згладжування штрихів, видалення шумів [74], апроксимація точок для спрощення кривих, додавання пропущених точок для відновлення штрихів.

Пошук та аналіз різноманітних ознак в даних рукописного введення також залишається темою для досліджень вже довгий час. Традиційні підходи опираються на знаходження та використання різноманітних ознак для представлення ними кожної точки вхідної послідовності [63]. Такі представлення гарно підходять для методів аналізу часових послідовностей. Другий тип ознак рукописного тексту базується на глобальних ознаках, що характеризують вже цілі послідовності точок. Такий варіант необхідний для методів, що проводять попередню сегментацію вхідних послідовностей. Також ці два типи ознак можна комбінувати, обраховуючи глобальні ознаки на основі отриманих поточкових ознак, наприклад будуючи гістограму ознак на всій послідовності точок.

Опис найбільш розповсюджених ознак, що використовуються для аналізу характеристик та особливостей рукописного тексту, можна знайти в роботах [63], [75], [76]. Прикладами таких ознак є нормалізовані координати, кривизна штрихів, відношення сторін, наявність закручених частин, лінійність написання, точки перегинів, точки перетину штрихів, швидкість написання, наявність надстрочних та підстрочних частин символів, направленість написання, кількість штрихів і т.д..

1.4 Існуючі методи сегментації рукописного тексту

Сегментація рукописного введення є основним етапом попередньої обробки в контексті більшості систем розпізнавання рукописного тексту. Завдання сегментації може включати сегментацію символів [77], сегментацію слів [78], сегментацію рядків [79]. Останній крок є одним з початкових етапів передобробки даних для роботи з документами, що надаються на вхід системи. Застосування цього етапу дасть змогу проводити вирівнювання комплексних документів зі складною структурою, завдяки їх розбиттю на окремі рядки, що будуть прийматись на вхід наступними модулями системи.

1.4.1 Сегментація рядків

Загальна продуктивність системи розпізнавання рукописного тексту напряду залежить від якості результатів виявлення рядків тексту. Будь-які помилки на цьому етапі впливають на точність сегментації слів та на наступні процедури розпізнавання тексту. Сегментація текстових рядків документів все ще залишається однією з найскладніших проблем у розробці надійних систем розпізнавання рукописного тексту [79].

Метою задачі сегментації лінії є розділення рукописного тексту на сегменти, кожен з яких містить ізольовану повну лінію з усіма відповідними штрихами. Тобто на виході блок сегментації повертає список наборів штрихів, кожен з яких визначає всі штрихи, що належать одній лінії рукописного тексту.

Сегментація рядків тексту на сьогодні залишається складною задачею зі своїми викликами, особливо у випадках багатоблокових документів, невіривняних текстів, рядків із різними кутами повороту тощо. Рукописний текст часто може містити тісно розташовані рядки, коли нижні штрихи верхнього рядка можуть перетинатися зі штрихами рядка нижче. Особливості природи рукописних даних, такі як відмінності між стилями написання одних і тих самих символів, недотримання міжрядкових інтервалів, наявність шумів, помилок та помарок при написанні, також створюють проблеми для вирішення задачі сегментації. Окремим складним випадком є рукописний текст, що написано одним штрихом, без підняття пера. У таких випадках межі між символами та словами можуть бути невизначені чітко, що також ускладнює точне сегментування тексту. Деякі з прикладів показано на рис. 1.7.

Задача сегментації рядків також може застосовуватися до інших типів документів, таких як списки, формули та таблиці, кожен з яких має свої особливі вимоги сегментації. Наприклад, документи з формулами повинні бути розбиті на окремі рядки з формулами, враховуючи, що одна формула може складатися з кількох рівнів з елементами ділення [83].

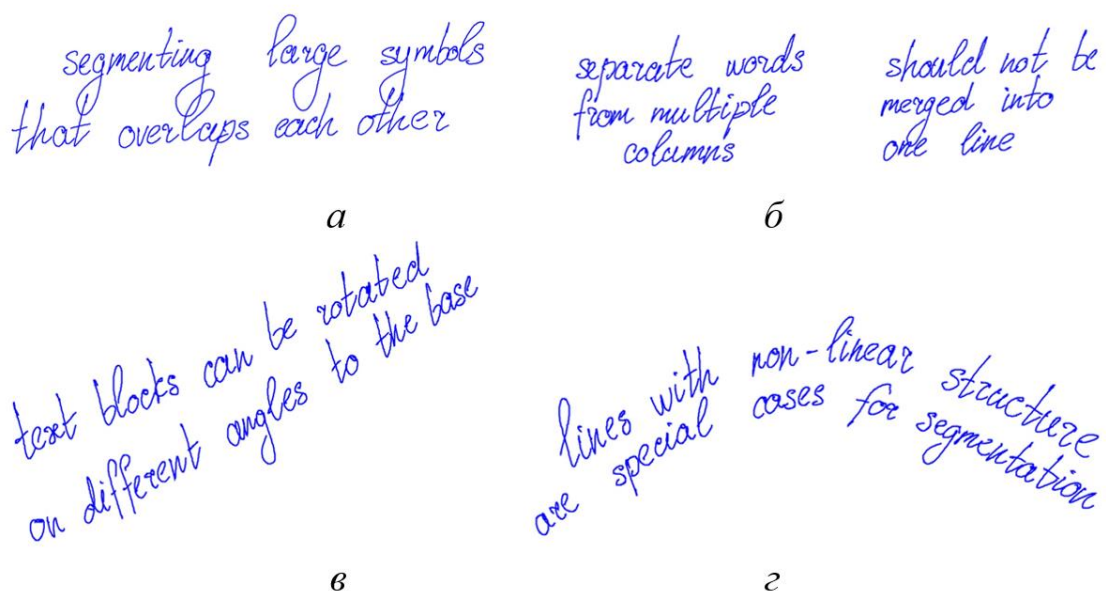


Рис. 1.7 – Приклади складних випадків для задачі сегментації рукописного тексту: а – перетин штрихів між рядками; б – структура колонок; в – написання тексту під кутом; г – нелінійна структура рядків [79]

Існує багато методів сегментації по лініям: статистичні методи групування на основі перевірки відстаней між штрихами, сегментація на базі результатів попередньої класифікації символів [80], методи пошуку мінімальних кістякових дерев (spanning trees) [81], за допомогою побудови проєкцій [83], та методи на основі нейронних мереж, які також можуть застосовуватись до вхідних даних у форматі зображень, де разом із сегментацією вони відразу вирішують завдання виявлення тексту [82]. В цій роботі використовуються методи сегментації ліній на основі побудови проєкцій [83].

1.4.2 Сегментація символів

На цьому етапі система отримує на вхід набір штрихів, що відносяться до одного рядка рукописного тексту, використовуючи результати попереднього етапу сегментації по лініям, або ж використовуючи тільки вхідні дані, якщо встановлено режим single-line, коли на вхід завжди приходить єдиний рядок тексту.

Реалізація даного етапу сильно залежить від обраного підходу до розпізнавання. У роботах [55], [58] розглядається проведення сегментації як окремого кроку, що передує наступним етапам розпізнавання. Інші підходи пропонують неявну сегментацію, що виконується паралельно з етапом розпізнавання, за допомогою використання нейромережевої часової класифікації [67], або підходів кодування-декодування, що тренуються на основі оцінки максимальної правдоподібності [84].

Оглядова робота [85] описує базові методи в напрямку сегментації символів, в роботі [86] пропонується використання дерев рішень, в [87] на основі знайдених векторів ознак будується сегментаційний граф, в якому вирішується задача знаходження оптимального шляху. Іншими прийнятими підходами до сегментації символів в роботах [90], [91] є метод опорних векторів, рекурентні нейронні мережі [88], динамічне програмування [89], [92].

Основні складнощі в задачі сегментації символів виникають через такі фактори:

- можливість зв'язного написання багатьох символів разом без відриву пера;
- форма рукописного символу залежить від позиції символу в слові та від наступних символів;
- сусідні символи в одному слові можуть перетинатися;
- відкладені штрихи, тобто такі, що були додані до даного символу після написання інших символів (напр. для символів 'i', 'i');
- варіативність та унікальність стилів написання різних людей.

Окрему увагу варто приділити саме наявності відкладених штрихів у вхідних послідовностях, оскільки не всі методи враховують даний фактор. Одним із запропонованих варіантів вирішення є намагання знайти та прибрати такі відкладені штрихи у вхідній послідовності [77].

Важливо зазначити, що більшість розглянутих підходів базуються на вирішенні задачі сегментації як попереднього кроку передобробки для успішного подальшого виконання розпізнавання та класифікації символів. Таким чином, основним критерієм перевірки якості роботи методів є саме результат класифікації, а не проміжний етап

сегментації, результат якого може не відповідати правильним межам кожного символу, при цьому давати достатній результат для правильного розпізнавання.

Проте існує набір задач, в яких також важливо отримати точні результати сегментації, наприклад розпізнавання складних послідовностей символів з комплексними просторовими зв'язками між собою, це і математичні та хімічні формули, музичні нотації тощо [25]. Іншим варіантом може бути задача подальшої роботи з кожним символом окремо, наприклад поліпшення рукописних символів [45] або вирівнювання тексту, що розглядається в даній роботі. Тому для таких задач розглядаються методи точної сегментації, які можуть також використовувати попередній результат розпізнавання рукописного тексту [77].

1.5 Постановка задач дослідження

Задача вирівнювання рукописного тексту може бути сформульована як геометрична модифікація вхідних даних рукописного введення для покращення читабельності, ясності та зовнішнього вигляду вирівняного тексту.

Рукописний текст представляється у вигляді наборів штрихів в порядку їх написання. Кожен штрих – це неперервна послідовність точок, що визначають траєкторію написання. Кожен штрих починається в момент дотику пера та закінчується в момент піднімання пера.

На рис. 1.8 продемонстровано приклад вирівнювання рукописного тексту, що складається з 3 рядків.

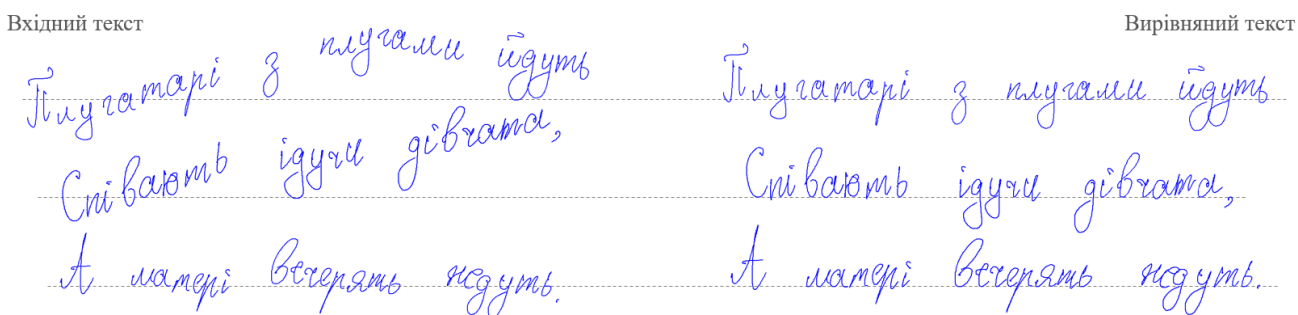


Рис. 1.8 – Приклад вирівнювання рукописного тексту

Для досягнення поставленої мети дисертаційного дослідження в наступних розділах розв'язуються такі задачі:

1. Вдосконалити метод сегментації рукописного тексту для підвищення точності маркування кожного символу.
2. Розробити метод вирівнювання рукописного тексту на основі розпізнавання та сегментації тексту з використанням структурного аналізу символів.
3. Виконати програмну реалізацію представлених методів у вигляді інструментальної системи вирівнювання для розв'язування задачі вирівнювання рукописного тексту українською мовою.
4. Виконати експериментальне дослідження параметрів та оцінювання ефективності роботи системи вирівнювання рукописного тексту на мобільних пристроях з обмеженнями обчислювальних ресурсів.

Було встановлено наступні основні вимоги до системи вирівнювання рукописного тексту, яка розробляється:

1. На вхід система приймає набір штрихів оригінального рукописного тексту, на вихід повертає набір штрихів вирівняного тексту.
2. Система повинна покращувати (або не погіршувати) вхідний рукописний текст для більшості випадків.
3. Додатково система повинна повертати результуючі метрики для кожного вирівняного рядка, це 4 основні лінії: верхня межа, ікс-висота, базова лінія, нижня межа.
4. Середній час вирівнювання одного рядка рукописного тексту не повинен перевищувати 1 секунди на сучасному мобільному пристрої, при довжині рядка не більше 60 символів.
5. Розмір моделей та скомпільованої бібліотеки не повинен перевищувати 10 Мб.

1.6 Загальна структура запропонованого рішення

Рішення для вирівнювання рукописного тексту, що пропонується в дисертаційній роботі, базується на проведенні сегментації символів та розпізнаванні тексту. Загальний процес вирівнювання можна розбити на три основні етапи, вони зображені на рис. 1.9.

На першому етапі проводиться розпізнавання та сегментація всіх вхідних штрихів у документі з основною задачею – визначення всіх рукописних символів у тексті. Початково на цьому етапі виконується комплексний аналіз вхідних штрихів: відокремлення нетекстових штрихів, ідентифікація текстових блоків, розділення блоків на окремі текстові рядки. Також оцінюється та коригується кут повороту для кожного рядка тексту. Далі кожен окремий рядок тексту направляється в систему розпізнавання. Тут система створює проміжні вихідні дані, що містять набір всіх розпізнаних символів разом із відповідними їм сегментами вхідних точок.



Рис. 1.9 – Загальна структура запропонованого рішення

На другому етапі відбувається безпосередньо структурний аналіз всіх символів в рядку тексту, який служить для визначення нових коректних позицій окремих символів, і далі всього рядка в цілому. Проводиться аналіз всієї доступної інформації про текстовий документ для визначення фінальних позицій рядка.

На останньому етапі проводиться модифікація штрихів рядка по обрахованим позиціям, що були отримані на попередніх етапах.

Запропоноване рішення являє собою повноцінну систему вирівнювання рукописного тексту, що здатна виконувати поставлену задачу без використання сторонніх компонентів та сервісів. Також окремі складові даного рішення можуть бути впроваджені в інші системи, наприклад виконання структурного аналізу символів для визначення нових позицій символів може бути використане в існуючих системах обробки рукописного тексту, які вже виконують задачі класифікації та сегментації тексту, а отже таке впровадження не потребуватиме додаткових ресурсів на розпізнавання тексту.

Запропонований в дисертаційній роботі метод для вирівнювання тексту дає змогу вирівнювати рукописний текст в документі, що може складатися з різної кількості текстових рядків, блоків, містити заголовки, списки та додаткові нетекстові елементи.

1.7 Висновки до розділу

У рамках першого розділу дисертаційної роботи описано постановку задачі вирівнювання рукописного тексту, визначено основні проблеми, що існують на сьогоднішній день та потребують уваги для ефективного вирішення поставленої задачі. Основними проблемами є в першу чергу сама природа рукописного введення інформації, яка має велику варіативність вхідних даних та є унікальною для кожного користувача. Індивідуальний рівень професійних навичок рукописного введення, швидкість написання, наявність помилок, персональні особливості написання – всі ці параметри впливають на вхідні дані, що повинні бути враховані та коректно опрацьовані для вирішення задачі вирівнювання тексту.

Окремо визначено підзадачу оцінки рукописного тексту на розбірливість та вирівняність, а також підзадачу порівняння двох рукописних текстів для визначення якості результатів вирівнювання. З наведених досліджень видно, що в більшості

випадків оцінка рукописного тексту виконується безпосередньо людьми за допомогою візуального огляду та базуючись на основних обраних критеріях.

Проведено ґрунтовний аналіз існуючих досліджень та передових робіт як в області обробки рукописного введення, так і конкретно для задачі вирівнювання тексту. Перші роботи по даній тематиці зосереджувались в основному на обробці рукописного вводу методами комп'ютерного зору, деякі з досліджень використовують методи розбиття штрихів на токени, кластеризацію, класифікацію екстремумів та інші методи. Останнім часом все популярнішим стає напрямок розвитку та використання генеративних нейронних мереж, що дозволяє також виконувати синтез та генерацію рукописного тексту, таким чином стиль вхідного рукописного тексту може бути замінений на інший, за допомогою попереднього розпізнавання цього тексту. Проте ці методи наразі не надають можливості вирівнювати складні частини тексту з достатньою якістю та точністю.

Проведено аналіз підходів до розпізнавання та сегментації рукописного тексту для визначення найбільш релевантних для використання в системі, що розроблюється в цій роботі. Окремо розглянуто задачу сегментації рукописного тексту на рядки, як першу складову комплексу передобробки вхідних даних системи. Проведено аналіз існуючих методів сегментації на символи, що є важливим етапом для отримання точних вхідних даних для посимвольного структурного аналізу.

Враховуючи переваги, недоліки та припущення існуючих методів, в даній дисертаційній роботі пропонується розробка нової системи для вирівнювання рукописного тексту, що буде базуватися на використанні результатів розпізнавання та сегментації тексту, для вирішення поставленої задачі з високою точністю та з можливістю використання в середовищах з обмеженнями. Загальна структура запропонованого рішення описується в секції 1.6, наведено інформацію про основні її компоненти. Детальний огляд складових частин системи проводиться в наступних розділах.

РОЗДІЛ 2. РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЯ СИМВОЛІВ

Розділ присвячений вирішенню задачі розпізнавання та сегментації рукописних символів. Система розпізнавання та сегментації не є розробкою автора, і подається для розуміння дисертаційної роботи в цілому. Основний підхід, що представлено в підрозділах 2.1 – 2.3, базується на використанні рекурентних нейронних мереж з двонаправленою довгою короткочасною пам'яттю та нейромережевою часовою класифікацією, що пропонується в роботі [93].

Досліджуються принципи роботи довгої короткочасної пам'яті, обґрунтовується додавання вічкових з'єднань в їх структуру для уникнення швидкого затухання градієнта похибок. Розглядається підхід нейромережевої часової класифікації для маркування послідовностей.

Покращення результатів сегментації базується на ідеї, представлений в роботі [77]. В пунктах 2.4.1 та 2.4.2 описується додаткова мережа та використання алгоритму динамічного програмування для уточнення та покращення результатів сегментації символів. В 2.4.3 та 2.4.4 представлено опис розроблених додаткових алгоритмів для приєднання відтермінованих символів, пошуку та виправлення помилок сегментації.

2.1 Двонаправлена довга короткочасна пам'ять

2.1.1 Рекурентна нейронна мережа

Рекурентність нейронної мережі означає, що вихід на поточному кроці обчислень стає входом на наступному кроці. На кожному елементі послідовності модель враховує не тільки поточний вхід, але і ту інформацію, що модель запам'ятала про попередні елементи.

Такий тип пам'яті дозволяє мережі вивчати довгострокові залежності у вхідних послідовностях, тобто в результаті вона може враховувати попередній контекст вхідних даних під час обробки. Саме ця властивість є критично важливою при роботі з рукописним введенням, де, наприклад, наступні символи в слові тісно пов'язані з

попередніми символами, або слова в реченні також пов'язані послідовністю введення. Рекурентність мережі дає змогу проявляти динамічну поведінку в часі, та призначена для імітації людського способу обробки послідовностей: аналізуючи речення, ми розглядаємо його повністю, а не окремі слова незалежно одне від одного.

Схема розгорнутої рекурентної нейронної мережі зображена на рис. 2.1. В кожен момент часу t на вхід мережі поступає нове значення x_t з вхідної послідовності. Далі воно переходить на прихований шар, застосовуючи значення вектору ваг u . На поточний елемент прихованого шару h_t окрім вхідних даних також надходить інформація з попереднього стану прихованого шару h_{t-1} , це з'єднання має вагу v . Результат роботи прихованого шару надходить на вихідний шар із застосуванням вектору ваг w , таким чином на виході мережі отримуються вектори ймовірностей кожного з класів.

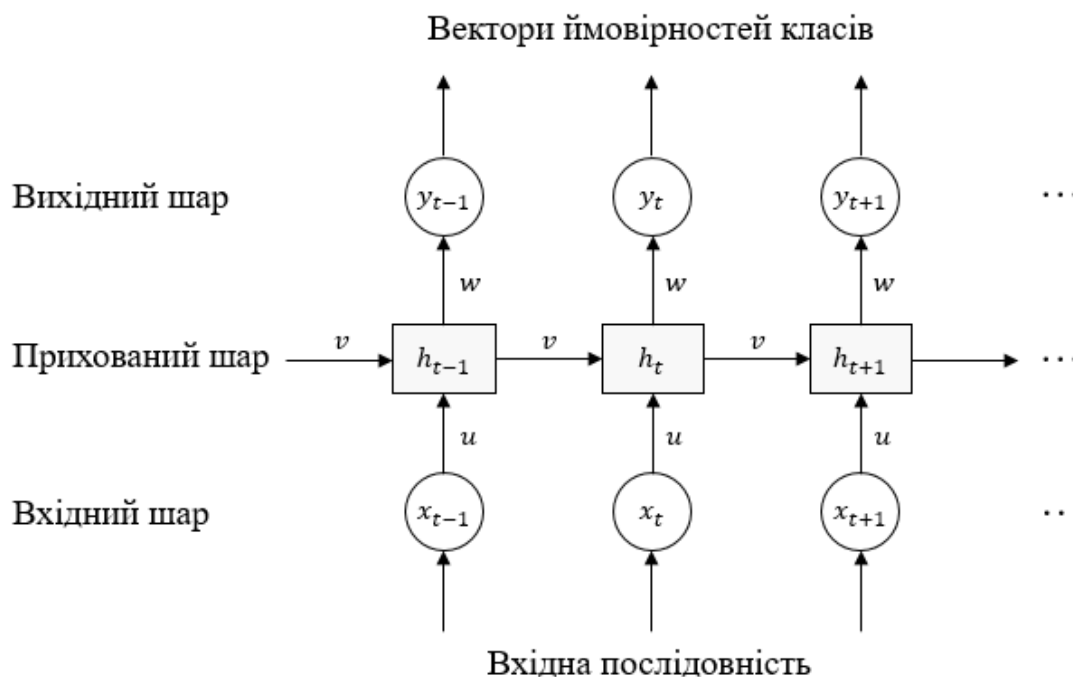


Рис. 2.1 – Схема розгорнутої рекурентної нейронної мережі

2.1.2 Довга короткочасна пам'ять

Ідею довгої короткочасної пам'яті (ДКЧП) вперше було запропоновано Хохрайтером та Шмідхубером в роботі [94]. Цей підхід спрямований на ефективне вирішення існуючої проблеми зникання градієнту у традиційних рекурентних мережах. Мережа з ДКЧП фокусується на запам'ятовуванні довгострокових залежностей у послідовностях даних, маючи змогу навчатися для виконання задач дуже глибокого навчання (very deep learning) [95], які потребують використання інформації, що була отримана задовго до поточного моменту часу [96], [97].

Використання ДКЧП дає змогу мережі будувати дуже довгі зв'язки між пов'язаними подіями, які можуть мати між собою 1000 та більше обчислювальних кроків. Це досягається шляхом введення в структуру мережі забувальних вентилей (forget gates). Їх задачею є визначення релевантності поточного значення комірки пам'яті, таким чином цей клапан може або зберегти це значення як важливе, або відкинути як непотрібне.

Описана особливість використання ДКЧП робить оптимальним її застосування для задач аналізу часових послідовностей та послідовних вхідних даних, навіть у випадках коли вхідні дані не завжди є інформативними, мають поєднання низькочастотних і високочастотних складових та великі часові проміжки між важливими подіями [98].

2.1.3 Двонаправлена ДКЧП

Покращення попереднього підходу базується на твердженні, що для ефективного аналізу часової послідовності необхідно оцінювати не тільки інформацію з минулої частини послідовності, а і майбутній контекст вхідних даних, тобто оброблювати вхідну інформацію в обох напрямках.

Двонаправлена рекурентна нейронна мережа була запропонована Шустером та Паліваном у 1997 році [99]. Ідея полягає у використанні скінченної вхідної послідовності для передбачування або маркування кожного елемента цієї

послідовності на основі минулого та майбутнього контексту цього елементу. Це виконується шляхом поєднання виходів двох РНМ, одна з яких виконує обробку послідовності зліва направо, а інша справа наліво. Виходи обох мереж поєднуються для визначення вихідних ймовірностей класів.

Двонаправлена довга короткочасна пам'ять (ДДКЧП) – це архітектура нейронних мереж, що поєднують ідею використання блоків ДКЧП та двонаправленості рекурентної мережі. Ця архітектура на сьогодні є однією з найбільш популярних для вирішення багатьох задач, таких як обробка природної мови [100], машинний переклад [101], розпізнавання мовлення [102], [103], [104], синтез мовлення з тексту [106], прогнозування часових рядів [108], вивчення граматики [97], [107], розпізнавання дій та активності людини [109], [110], розпізнавання емоцій [111], розпізнавання рукописного вводу [105].

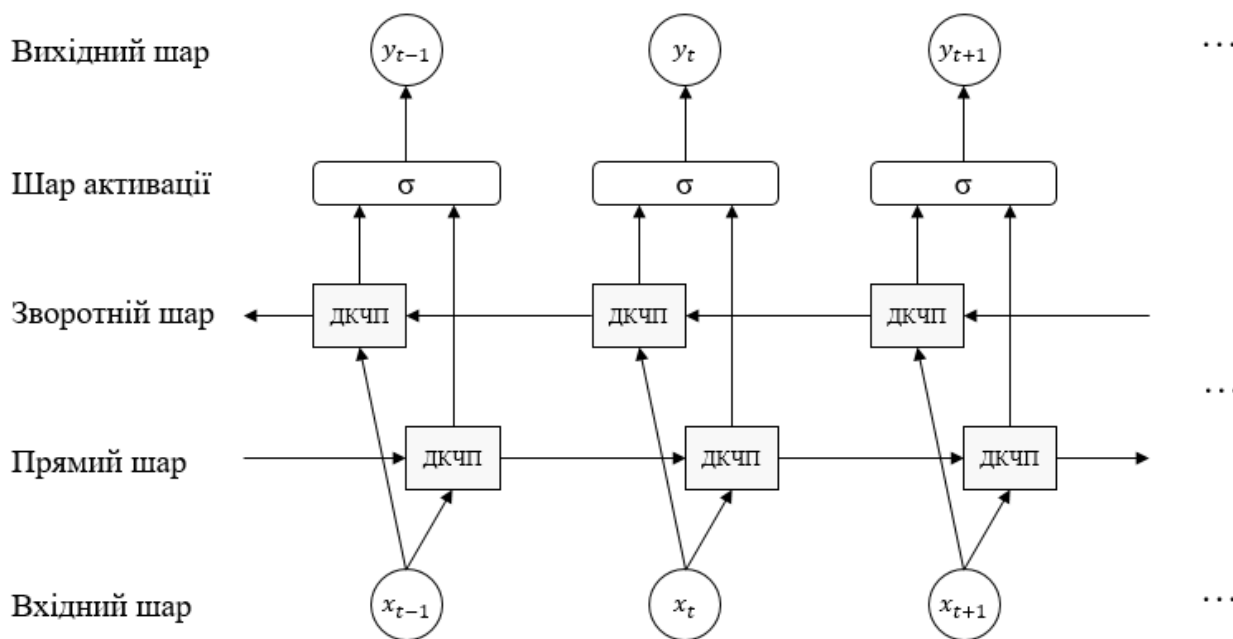


Рис. 2.2 – Схема розгорнутої двонаправленої рекурентної нейронної мережі

На рис. 2.2 зображено загальну схему рекурентної мережі, що була використана для побудови системи розпізнавання. Мережа містить 2 окремі приховані шари з вузлами ДКЧП для прямого та зворотнього напрямків. Кожен шар виконує обробку

вхідних даних тільки в одному напрямку та передає результат на шар активації. Важливо, що прямий та зворотній шар не мають жодних зв'язків між собою та працюють незалежно один від одного. Після поєднання результатів з обох шарів в шарі активації, результат передається на вихідний шар. Таким чином, представлена двонаправлена РНМ виконує розпізнавання кожної точки вхідної послідовності, аналізуючи минулий та майбутній контексти.

2.1.4 Структура блоку ДКЧП

Детально розглянемо структуру блоків ДКЧП, з яких складаються приховані шари мережі.

Основною складовою блоку ДКЧП є клітина пам'яті, що забезпечує зберігання інформації протягом тривалого періоду часу, та вентиля, що регулюють вхідні та вихідні потоки інформації. Початково ДКЧП містив лише два типи вентилів – вхідні та вихідні. В роботі [112] було додано забувальний вентиль, а в роботі [113] запропоновано ідею вічкових з'єднань (peerhole connections), за допомогою яких вентиля можуть бачити стан клітини пам'яті. В різних роботах, в залежності від задач, структура блоку ДКЧП може варіюватися: вічкові з'єднання можуть додаватися лише до деяких вентилів, до блоку може додаватись об'єднаний забувальний та вхідний вентиль, тоді процес забування інформації відбувається лише тоді, коли на вхід поступає нова інформація, необхідна для запам'ятовування. В даній роботі було використано блок ДДКЧП з трьома вентилями та однією клітиною пам'яті, його структура зображена на рис. 2.3.

Вхідний вентиль регулює кількість вхідної інформації, що надходить на зберігання до клітини пам'яті. Забувальний вентиль вирішує яка інформація буде відкинута з попереднього стану. Вихідний вентиль контролює інформацію, що використовується для обчислення активації виходу блоку, базуючись на інформації з минулого та поточного станів.

Активація вхідного вентиля визначає кількість поточної вхідної інформації, що буде отримуватись блоком в даний момент часу. Значення 0 означає закриті вхідні ворота і відсутність вхідної інформації, значення 1 означає врахування всієї інформації. Активація забувального вентиля визначає кількість інформації, що буде відкинута з попереднього стану, і, відповідно, кількість яка буде збережена. Активація вихідного вентиля визначає скільки інформації з поточного стану буде враховано на виході блоку. З'єднання стрілок у чорних крапках на рисунку визначає множення відповідних значень.

Опишемо рівняння для визначення роботи моделі ДДКЧП.

Для кожного моменту часу t будемо використовувати наступні позначення:

- $x_t \in \mathbb{R}^d$ – вектор вхідних даних, де значення d визначає кількість вхідних ознак мережі
- i_t – вектор активації вхідного вентиля
- f_t – вектор активації забувального вентиля
- o_t – вектор активації вихідного вентиля
- c_t – вектор комірки пам'яті
- h_t – вектор прихованого шару, або вихідний вектор блоку

Значення вагових матриць позначають W , вектори зміщення позначають b , це параметри які повинні вивчатись впродовж навчання мережі. Всі початкові значення встановлюються як 0. Функція активації σ є логістичною сигмоїдою, функція активації τ – гіперболічний тангенс.

Тоді модель ДДКЧП визначається наступними рівняннями:

Вхідний вентиль:

$$i_t = \sigma_i(x_t * W_{xi} + h_{t-1} * W_{hi} + c_{t-1} * W_{ci} + b_i) \quad (2.1)$$

Забувальний вентиль:

$$f_t = \sigma_i(x_t * W_{xf} + h_{t-1} * W_{hf} + c_{t-1} * W_{cf} + b_f) \quad (2.2)$$

Клітина пам'яті:

$$c_t = f_t c_{t-1} + i_t \tau_c (x_t * W_{xc} + h_{t-1} * W_{hc} + b_c) \quad (2.3)$$

Вихідний вентиль:

$$o_t = \sigma_i (x_t * W_{xo} + h_{t-1} * W_{ho} + c_t * W_{co} + b_o) \quad (2.4)$$

Вихід блоку:

$$h_t = o_t \tau_h (c_t) \quad (2.5)$$

Процес тренування такої мережі базується на тренуванні з вчителем з використанням ітеративного градієнтного спуску комбінованого зі зворотним поширенням помилки, змінюючи всі ваги ДКЧП мережі в пропорції відповідно до похідної отриманої похибки. Так звана константна карусель помилок (constant error carousel) [115], що з'являється з додаванням вічкових з'єднань між клітиною пам'яті та всіма вентилями, під час навчання буде безперервно повертати отриману помилку до кожного вентиля блоку, доки вони не навчаться відсікати її. Це дає змогу обходити проблему швидкого зникнення градієнта похибок [116] та дозволяє ефективно навчати мережу для послідовностей з тривалими паузами між значеннями.

Для поставленої в даній роботі задачі точної сегментації вхідних послідовностей рукописного тексту, здатність ДДКЧП враховувати контекст у двох напрямках є надзвичайно корисною. Це дає змогу мережі запам'ятовувати не тільки особливості написання окремих символів, а й виконувати їх правильне маркування, з врахуванням геометричних та семантичних особливостей сусідніх символів.

2.2 Нейромережева часова класифікація

2.2.1 Задача маркування послідовностей

Однією з основних поставлених задач до системи розпізнавання та сегментації є виконання маркування вхідної послідовності на окремі часові кроки, кожен з яких належить одній з міток класу. Традиційні рекурентні мережі, що призначені для виконання задачі класифікації, не здатні напряму вирішувати задачу маркування.

Вхідні тренувальні дані для задачі класифікації зазвичай представляються у вигляді пар (x, y) , де x – вхідний елемент, а y – відповідна мітка або клас для цього елемента. В задачі маркування вхідними даними є не окремі точки, а послідовності точок x , а виходами мережі є відповідні мітки y зі скінченного алфавіту міток класів. Таким чином, навчання такої мережі повинно виконуватись лише на тренувальних даних з чітким розбиттям вхідних послідовностей точок на незалежні мітки класів. Це накладає вимогу, що всі тренувальні дані повинні бути попередньо відсегментовані та розмічені, а виходи мережі повинні додатково оброблюватись для визначення фінальних міток. Проте створення таких наборів тренувальних даних достатнього розміру є дуже складним, оскільки потребує ручної анотації кожної точки вхідних послідовностей.

Для вирішення задачі маркування послідовностей з використанням РНМ було запропоновано декілька підходів, одним з яких є їх комбінація з прихованими марковськими моделями (ПММ) в гібридному підході [117]. Ця система використовує ПММ для моделювання довготривалу послідовну структуру даних, та нейронні мережі для виконання локалізованих класифікацій. Це дозволяє автоматично виконувати сегментацію послідовності під час тренування та перетворювати результат класифікації мережею в послідовності міток класів. Основними недоліками такої системи є як унаслідування основних проблем самих ПММ, так і зниження ефективності використання основного потенціалу РНМ для моделювання послідовностей.

2.2.2 Ідея нейромережевої часової класифікації

В роботах [93] та [67] було запропоновано метод нейромережевої часової класифікації (НЧК). Його ідея полягає у допомозі рекурентній нейронній мережі з вивченням залежностей між вхідними послідовностями та вихідними класами без потреби мати пряму відповідність всіх елементів послідовності до всіх міток класів. Цей підхід може застосовуватись для маркування вхідних послідовностей довільної

довжини до вихідних послідовностей довільної довжини. Важливо, що НЧК дозволяє обходити необхідність в точній анотації всіх тренувальних даних. Основною прикладною областю використання цього методу є задача аналізу часових послідовностей, наприклад розпізнавання голосу, де модель повинна перетворити мовлення у цифровий текст [118], або розпізнавання рукописного тексту, особливо у випадках зі зв'язним написанням довгих послідовностей тексту [119].

Навчання РНМ з використанням НЧК виконується на основі наступних вхідних даних: матриці ознак для вхідної послідовності та відповідного вектору з мітками класів, що для нашої задачі являють собою текстові символи. Ця вхідна інформація подається без прив'язки кожного символу послідовності до конкретного моменту часу.

2.2.3 Кодування маркування послідовностей

Скінченний набір всіх міток класів будемо позначати алфавітом L . Також для коректного маркування послідовностей з повторюваними символами необхідно ввести додатковий елемент – символ *blank*. Розширений алфавіт L' визначається як $L \cup \text{blank}$. Під час кодування послідовності символ *blank* завжди додається між символами що повторюються, а також може бути доданий довільну кількість разів у будь-яку іншу позицію послідовності. На етапі декодування всі символи *blank* будуть видалені, залишаючи у вихідній послідовності лише значущі символи. Таким чином є можливість створювати та комбінувати різні варіації маркування однієї і тієї ж вхідної послідовності символів.

Задача НМ полягає у навчанні визначати закодовану послідовність символів. Для вхідної послідовності x довжиною T з навчального набору даних S , маємо y_k^t – активацію вихідного вузла k в момент часу t . Можемо інтерпретувати це значення y_k^t як ймовірність отримати мітку класу $k \in L'$ в момент часу t . Будь-яка послідовність $\pi \in L'^T$ називається шляхом, її ймовірність визначається наступною формулою:

$$p(\pi|x, S) = \prod_{t=1}^T y_{\pi_t}^t, \forall \pi \in L'^T \quad (2.6)$$

Визначимо функцію декодування $D : L'^T \rightarrow L^{\leq T}$, що визначає набір всіх допустимих маркувань $L^{\leq T}$. Умовна ймовірність кожного маркування $l \in L^{\leq T}$ обчислюється на основі підсумовування ймовірностей всіх побудованих шляхів, що дають однаковий результат декодування, оскільки всі шляхи є взаємовиключними, та представляється формулою:

$$p(l|x) = \sum_{\pi \in D^{-1}(l)} p(\pi|x) \quad (2.7)$$

Результатом декодування є вектор кандидатів сегментації символів $X = [O_{start}, O_{end}, Z]$, де O_{start} позначає індекс початкової точки з вхідної послідовності, O_{end} – індекс кінцевої точки. Вектор $Z = [(L, V)]$ визначає отримані варіанти класифікації символів, де L – мітка класу, V – її ймовірність. Під час декодування обираються всі результуючі варіанти, ймовірності яких більші певного встановленого порогового значення V_{dec} , окрім символу *blank*.

Для навчання такої мережі за допомогою градієнтного спуску необхідно визначити цільову функцію втрат. Вона базується на принципі найбільшої схожості, а саме виконує задачу максимізації логарифмічних ймовірностей до цільових міток, та визначається як негативна ймовірність правильного маркування всіх навчальних послідовностей з тренувального набору S . Нехай z – очікуваний результат для вхідної послідовності x , тоді цільова функція НЧК визначається формулою:

$$O(S) = -\ln \left(\prod_{(x,z) \in S} p(z|x) \right) = - \sum_{(x,z) \in S} \ln p(z|x) \quad (2.8)$$

2.2.4 Приклад роботи НЧК

Розглянемо приклад роботи НЧК для виконання задачі сегментації рукописного тексту, що наведено на рис. 2.4. На вхід подаються рукописні штрихи, що відповідають символам 'м', 'і', 'с'. Два символи складаються з одного штриха, та один символ складається з двох штрихів. В цьому прикладі всі штрихи написані в правильному порядку написання, тобто зліва направо та без переривання послідовності одного символу. Проблема з відтермінованими штрихами буде розглянута окремо.

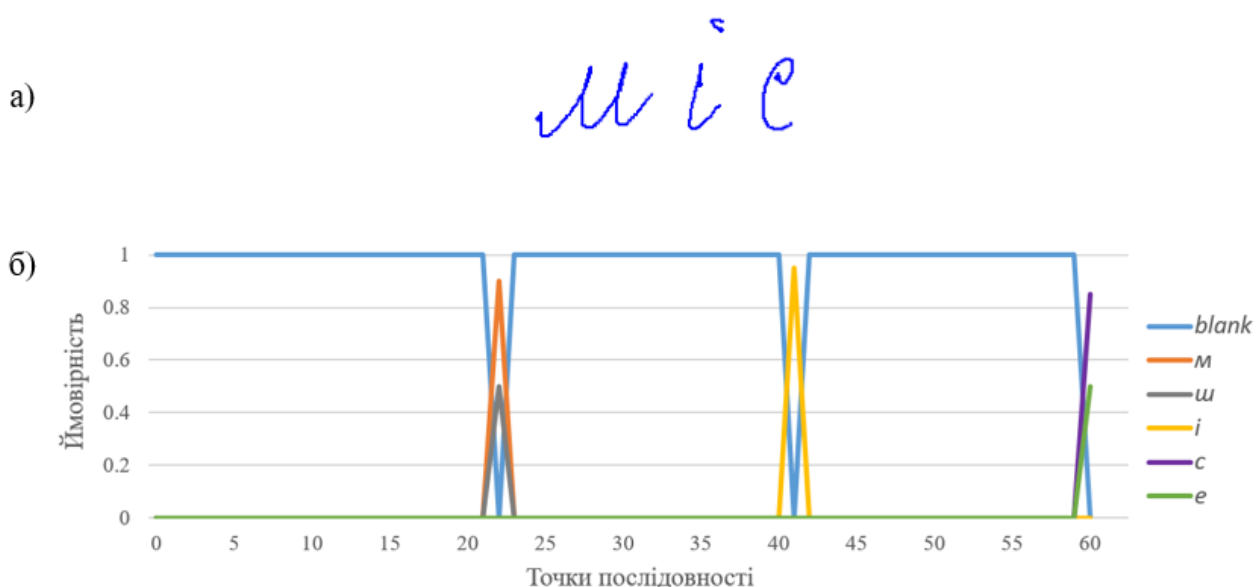


Рис. 2.4 – Приклад роботи нейромережевої часової класифікації: а) вхідні точки рукописного введення; б) результат класифікації та маркування послідовності рекурентною нейронною мережею з НЧК

На виході рекурентної нейронної мережі з НЧК ми отримуємо результати класифікації символів та сегментації вхідної послідовності. Для першого та третього символів результат класифікації містить по 2 варіанти, а саме літера 'м' також класифікувалась як літера 'ш' з меншою ймовірністю, та літера 'с' має другим варіантом класифікації символ 'е' через специфічний стиль написання. Літера 'і' має однозначний результат класифікації з високою ймовірністю. Також у варіантах

результатів є символ blank, що маркує всі інші часові мітки, окрім явних всплесків всіх значущих символів, що дозволяє ефективно сегментувати послідовності символів між собою. Тобто результат сегментації визначається на основі завершення всплеску символу з найбільшою ймовірністю, окрім символу blank, позначаючи кінець одного символу та початок наступного.

2.3 Система розпізнавання та сегментації символів

Система розпізнавання та сегментації символів, що є складовим компонентом всієї системи вирівнювання рукописного тексту, виконує задачі попереднього аналізу вхідних даних та безпосередньо розпізнавання та сегментацію вхідного рукописного тексту. Далі розглянемо основні складові цієї системи та архітектуру самої мережі.

2.3.1 Аналіз та обробка вхідних даних

В систему розпізнавання можуть включатись різні додаткові компоненти для аналізу та передобробки вхідних даних, що дає змогу виконувати більш складні задачі розпізнавання та оброблювати більш складні документи. Включення цих модулів в розроблену систему вирівнювання регулюється поставленими задачами та особливостями вхідних даних. Опишемо основні такі компоненти:

- Класифікація вхідних штрихів. Основною задачею є відділення всіх текстових штрихів від будь-яких інших, наприклад штрихів що відносяться до малюнків, графіків, формул, діаграм, декоративних елементів тощо [120]. Також на цьому етапі може проводитись розділення та ідентифікація окремих текстових блоків у випадку складних та комплексних документів з блоковою або колоночною структурою. Такий аналіз базується на архітектурі ієрархічних згорткових і рекурентних нейронних мереж [121], [122]. Також для цієї задачі можуть застосовуватись інші ієрархічні глибокі згорткові мережі з механізмом уваги [123], [124], [125].

- Класифікація мови рукописного тексту. У випадках коли вхідні дані можуть містити текст, що складається зі слів різних мов, необхідно виконувати їх розділення для коректного розпізнавання, а також для правильного вирівнювання у випадках, коли ці мови мають різні правила розміщення символів [126]. Ця задача вирішується статистичним класифікатором, що аналізує виходи ДДКЧП з використанням НЧК [67].
- Впорядковування послідовності штрихів. Досить частою проблемою в обробці рукописного введення є наявність відкладених штрихів, коли послідовність написання штрихів порушується, і частина штрихів може дописуватись в неправильному порядку, віддалено в часі від інших штрихів свого ж символа. Правильний порядок штрихів є критично важливим для коректної роботи РНМ з ДДКЧП та НЧК, які вимагають суворої відповідності послідовностей у навчальних даних. Також для деяких задач є потреба підтримувати довільний порядок введення всіх штрихів тексту. Ці задачі вирішуються за допомогою алгоритмів впорядковування послідовностей штрихів [127]. Одним з підходів є впорядковування штрихів на основі горизонтальних та вертикальних відстаней та перетинів штрихів [128].
- Сегментація на рядки тексту. Задачею цього кроку є кластеризація кожного текстового блоку на окремі ізольовані рядки, що містять усі відповідні рукописні штрихи тексту. Сегментація на лінії виконується на основі побудови проекцій точок на осі координат та аналізу часових та просторових зв'язків між групами точок [83].
- Визначення кута нахилу рядка. Корекція кута нахилу може знадобитись у випадках дуже нахилених ліній для коректного розпізнавання такого тексту. Базується на обчисленні центрів гравітацій частин рядка та аналізі просторових відношень між штрихами [129], [130], [131].

Також виконується попередня обробка вхідних даних вже безпосередньо для входу до нейронної мережі. Сюди належить видалення дублюючих точок, спрощення

послідовностей штрихів, видалення шумів та коротких гачків, згладжування ліній, нормалізація по позиціям та розмірам [132], [133]. Ці методи дублюються з попередньою обробкою вхідних даних для системи вирівнювання, та детально описуються в розділі 3.3.

2.3.2 Архітектура основної мережі класифікації символів

Розпізнавання та сегментація символів виконується за допомогою рекурентної нейронної мережі з використанням двонаправленої довгої короткочасної пам'яті [134] та нейромережевої часової класифікації [93]. Структура мережі складається 9 шарів, а саме: з вхідного шару, двох прямих та двох зворотніх шарів ДКЧП, двох об'єднуючих шару для поєднання результатів шарів ДКЧП, повнозв'язного шару та шару декодування НЧК. Архітектура мережі представлена на рис. 2.5.

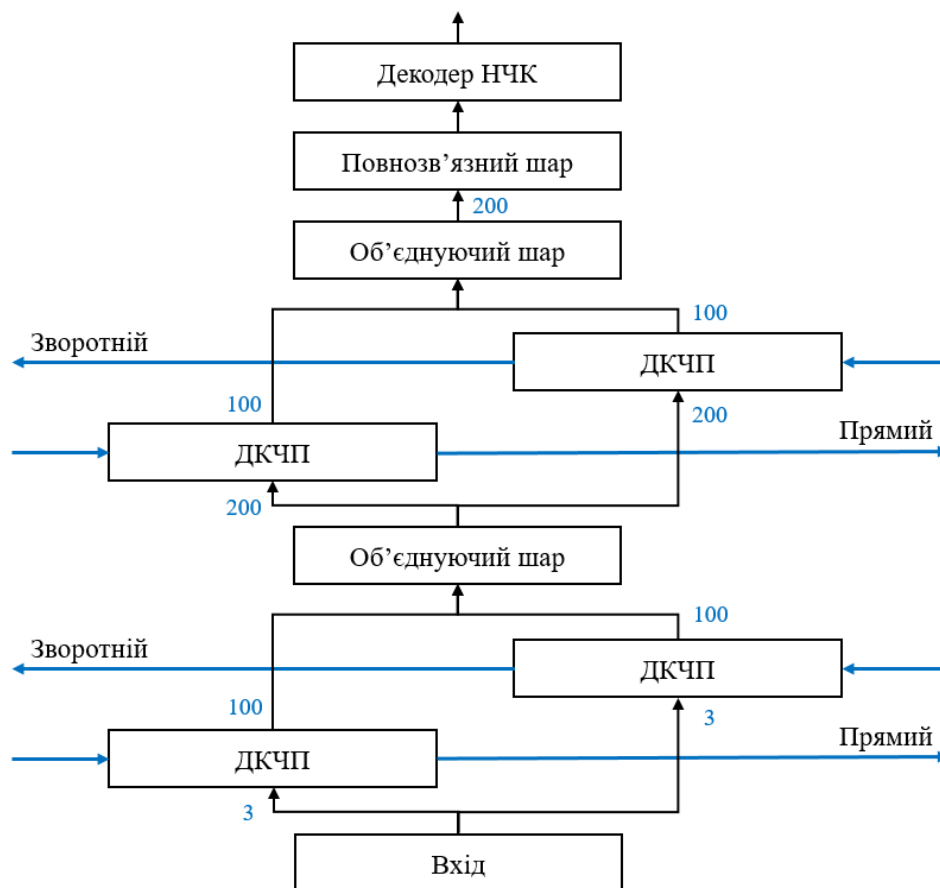


Рис. 2.5 – Архітектура рекурентної нейронної мережі для розпізнавання та сегментації символів

Вхідний шар отримує нормалізовані вектори ознак послідовностей, кожен вектор складається з 3 ознак: дельта по координаті x , дельта по координаті y , мітка початку або кінця штриха. На виході декодер НЧК обирає результуючі символи за найбільшими значення ймовірностей, надаючи також інформацію про початкові та кінцеві позиції відповідного маркування вхідної послідовності до розпізнаного символу, виконуючи таким чином задачу сегментації. Загальна кількість ваг мережі становить близько 300 000.

Навчання мережі відбувається із застосуванням міні-пакетного градієнтного спуску та алгоритму оптимізації Adadelta [135], який забезпечує надійну сходимість параметрів. Також під час навчання активується Dropout шар з $p = 0.5$, що призначений для підвищення продуктивності навчання. До навчальних наборів вводяться загальні вимоги, що визначаються застосуванням ДДКЧП, зокрема бажана наявність вхідних послідовностей що містять лише один символ тексту, для забезпечення високої точності сегментації.

2.4 Покращення результатів сегментації символів

Застосування традиційного підходу НЧК виконує сегментацію послідовностей точок на основі всплесків ймовірностей символів, вважаючи їх одночасно закінченням одного символу та початком наступного. Загалом такий підхід дає прийнятну точність сегментації, проте виникають часті випадки, коли початок нового символу не збігається з піком ймовірності від попереднього символу. В результаті, розбиття послідовностей на символи буде давати помилку, призначаючи частину точок помилковому символу, порушуючи геометрію символів та пропускаючи цю помилку на наступні компоненти системи вирівнювання, що є критичним для їх роботи.

В цьому підрозділі представлено вдосконалення методу сегментації рукописного тексту, особливістю якого є здатність до знаходження та виправлення помилок сегментації та визначення відповідних символів для відтермінованих штрихів, що дозволяє підвищити точність маркування кожного символу.

2.4.1 Додаткова мережа для збільшення точності сегментації

Для збільшення точності сегментації було використано ідею, запропоновану в роботі [77], яка полягає у застосуванні додаткової рекурентної нейронної мережі для коректування результатів сегментації, отриманих від основної мережі класифікації символів. Результат сегментації представляється у вигляді точок сегментації, що визначають перехід між об'єктами класифікації всередині вхідної послідовності, в нашому випадку між текстовими символами.

Задача додаткової РНМ полягає у класифікації точок вхідних штрихів на N класів, де $N - 1$ класів визначають потенційні сегменти символів, та нульовий клас, зарезервований для маркування відтермінованих штрихів, тобто написаних окремо від інших штрихів відповідного символу. Для даної задачі було використано 4 класи, яких достатньо для ефективного виконання поставленої задачі сегментації тексту. Збільшення кількості класів може бути необхідним для підтримки більш складних структур рукописного введення. Таким чином всі точки, що належать одному символу, позначаються першим класом, всі точки наступного символу – другим класом, і так далі, в результаті всі набори точок кожного символу послідовно позначаються одним з трьох доступних класів. Нульовим класом позначено всі точки відтермінованих штрихів. Така реалізація маркування символів є мовонезалежною.

Додаткова мережа являє собою легковісну РНМ, що складається з семи шарів: вхідний шар, два прямі та два зворотні шари ДДКЧП, один зв'язуючий шар та вихідний шар. На вхід ця мережа приймає вектор з чотирьох ознак, три з яких є аналогічними до вхідних ознак основної мережі, та одна ознака що містить вихідні дані з основної мережі, тобто попередній результат сегментації вхідної послідовності у вигляді набору точок сегментації. Мережа містить близько 14000 ваг.

Мережа тренується мінімізувати помилку крос-ентропії відносно цільових даних за допомогою функції активації Softmax [136], що знаходиться на останньому шарі мережі. Важливо відмітити, що для тренування такої мережі потребується точна розмітка сегментації вхідних даних з розділенням послідовностей точок на окремі

символи. У випадку відсутності доступних розмічених даних, як наприклад у випадку з українською мовою, з'являється необхідність у створенні великого тренувального набору даних. Проте ця задача значно полегшується з використанням наявних попередніх результатів сегментації від основної мережі. Вони можуть бути використані для їх ручного перегляду та коректування, що спрощує цю задачу відносно сегментації всіх вхідних даних з нуля.

2.4.2 Динамічне програмування для інтерпретації результатів

Вихідні результати додаткової мережі представляються у вигляді ймовірностей $\tilde{P}(\omega_t)$ для кожної точки часової послідовності t відносно кожного з класів ω . Ці результати повинні бути коректно опрацьовані для визначення фінальних позицій точок сегментації. Для цього сформулюємо задачу декодування максимальної правдоподібності (maximum likelihood decoding), де $\{\omega_t^0\}_{t=1,T} = \omega_1^0, \dots, \omega_T^0$ – послідовність класів сегментів зі спеціальним обмеженням для кількості сегментів K :

$$L(\{\omega\}_{t=1,T}; \{(x, y)\}_{t=1,T}) = \prod_{t=1}^T \tilde{P}(\omega_t) \quad (2.9)$$

$$\{\omega_t^0\}_{t=1,T} = \underset{\{\omega_t\}_{t=1,T}}{\operatorname{argmax}} L(\{\omega\}_{t=1,T}; \{(x, y)\}_{t=1,T}) \quad (2.10)$$

Базуючись на тому, що нам відома кількість сегментів, для вирішення цієї задачі в роботі [77] застосовується підхід динамічного програмування [92], [137]. Для цього вводяться додаткові правила для можливих переходів між класами сегментів:

- 1) Перший клас може бути змінений тільки другим або нульовим класом.
- 2) Другий клас може бути змінений тільки третім або нульовим класом.
- 3) Третій клас може бути змінений тільки першим або нульовим класом.
- 4) Відкладений штрих, помічений нульовим класом, може бути виключно цілісним штрихом без роз'єднань сегментації.
- 5) Нульовий клас може бути замінений тільки попереднім класом у послідовності.

Можливі переходи між всіма чотирма класами для проходження динамічного програмування зображені на рис. 2.6.

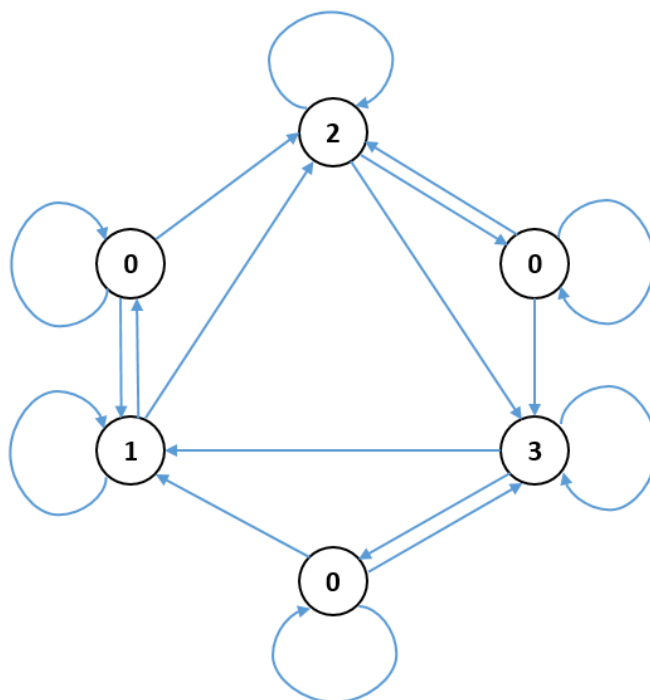


Рис. 2.6 – Допустимі переходи між чотирма класами в процесі коректування результатів сегментації додатковою мережею

Визначимо функцію вартості пройденого шляху як $dp_t(k, s)$ від сегменту $k = \overline{1, K}$ точки $t = \overline{1, T}$, де K – загальна кількість сегментів, T – загальна кількість точок.

Початкові значення параметрів:

- $t = 0$
- $k = 1$
- $s = 1$
- $dp_0(1, 1) = c_0(1)$

Загальний крок динамічного програмування визначається наступними рекурентними функціями, де ПШ – початок штриха:

$$dp_{t+1}(k, 1) = \max \begin{cases} dp_t(k, 1) + c_{t+1}(k \bmod N + 1); \\ dp_t(k - 1, 1) + c_{t+1}((k - 1) \bmod N + 1); \\ dp_t(k - 1, 0) + c_{t+1}((k - 1) \bmod N + 1) \text{ якщо ПШ}; \\ dp_t(k, 0) + c_{t+1}(k \bmod N + 1) \text{ якщо ПШ}; \end{cases} \quad (2.11)$$

$$dp_{t+1}(k, 0) = \max \begin{cases} dp_t(k, 0) + c_{t+1}(0); \\ dp_t(k, 1) + c_{t+1}(0); \end{cases} \quad (2.12)$$

Кожен наступний стан параметрів задається формулою:

$$(k_{t+1}, s_{t+1}) = \underset{(k,s)}{\operatorname{argmax}}(dp_{t+1}(k_t, 0), dp_{t+1}(k_t, 1), dp_{t+1}(k_{t+1}, 1)) \quad (2.13)$$

В результаті вирішення задачі декодування максимальної правдоподібності послідовностей класів за допомогою динамічного програмування [77], отримуємо найбільш ймовірний відкоректований результат сегментації вхідної послідовності, разом з результатом класифікації символів від основної РНМ. Важливою особливістю такого підходу є точність в ізолюванні відтермінованих штрихів у вхідній послідовності. Ця додаткова мережа дає змогу покращити точність сегментації та виправляти значну частину помилок сегментації, які допускає основна мережа класифікації.

2.4.3 Приєднання відтермінованих штрихів

Додаткова мережа для уточнення сегментації на виході повертає відповідність сегментів точок до відповідних символів, а також окремі сегменти для відтермінованих штрихів, для яких є невідомою відповідність до конкретного символа розпізнаної послідовності.

Для вирішення задачі приєднання відтермінованих штрихів до відповідних символів було розроблено алгоритм приєднання на основі аналізу відстаней між штрихами та символами. Для всіх символів алфавіту визначено спеціальні регіони та позиції, куди з найбільшою ймовірністю можуть потрапляти відтерміновані штрихи. Наприклад, для символів 'і', 'ї', 'й' розташування очікуваної позиції відтермінованих

штрихів є їх верхня частина, для символу ‘А’ – середня частина, для символів ‘ц’, ‘ц’ – нижня частина, і т.д.

Ймовірність приналежності відтермінованого штриха до кожного з символів послідовності визначається на основі функції розрахунку дистанції, після цього виконується приєднання даного штриха до відповідного символа, який найкраще підходить по параметрам приєднання та по геометричному розташуванню. При відсутності символів, що задовольняють умовам приєднання, такий відтермінований штрих вхідної послідовності вважається шумом та може бути відкинутим.

Граничні значення по можливим відстаням між штрихами визначаються відносно значення висоти медіанного штриха з вхідної послідовності та за допомогою коефіцієнтів, що підбираються емпіричним шляхом. Використання висоти медіанного штриха дозволяє алгоритму коректно працювати з будь-якими вхідними даними, незалежно від розмірів написаного тексту.

Таким чином, всі відтерміновані штрихи аналізуються даним алгоритмом та приєднуються до одного з символів розпізнаної послідовності, або відкидаються.

На рис. 2.7 наведено процес побудови сегментації разом з використанням додаткової мережі для покращення точності сегментації та алгоритмом приєднання відтермінованих штрихів. На третьому проміжному кроці прикладу чорним кольором відмічено відтермінований штрих, що надійшов з додаткової мережі, – крапку над символом ‘і’. Алгоритм приєднання відтермінованих штрихів успішно приєднав цей штрих до відповідного символа. На четвертому кроці зображено фінальний результат сегментації, а також результат класифікації символів.

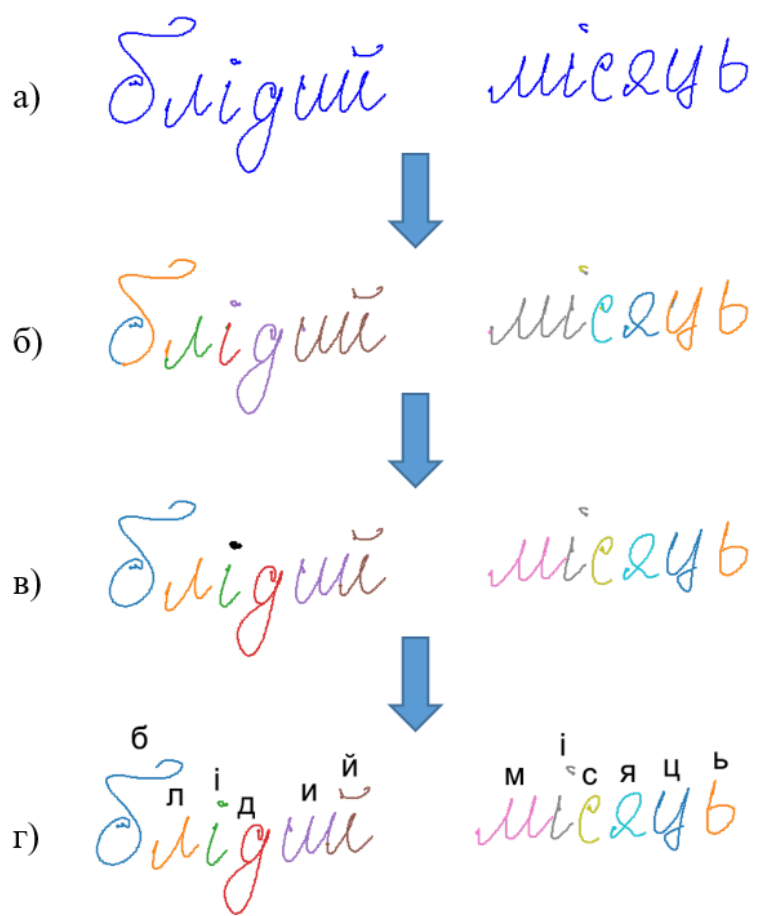


Рис. 2.7 – Приклад розпізнавання та побудови сегментації: а) вхідна послідовність штрихів; б) попередній результат сегментації після виходу з основної мережі класифікації; в) результат сегментації з додатковою мережею для покращення сегментації; г) результат сегментації з алгоритмом приєднання відтермінованих штрихів та результат класифікації символів

2.4.4 Виправлення помилок сегментації

Розроблена система успішно виконує сегментацію вхідних послідовностей для різних почерків та стилей написання. Проте через природу рукописного введення, досягнути абсолютної точності сегментації для даної задачі майже неможливо. В окремих випадках система може допускати помилки у визначенні сегментації, наприклад при зв'язному написанні деяких окремих частин символів, при пропусканні символів або їх частин, при допусканні помилок в написанні символів.

Це може вплинути як на результат сегментації одного символу, так і потягнути за собою порушення сегментації всієї послідовності рядка.

Для виправлення таких дрібних помилок сегментації було розроблено два додаткові алгоритми, які базуються на пошуку малих сегментів, відокремлених від інших основних частин свого символу, що можуть бути помилками сегментації та потребувати корекції. Перший алгоритм аналізує всі пари сусідніх символів на наявність спільних сегментів, що займають кінець одного символу та початок іншого. Якщо один з таких сегментів знаходиться на краю одного з символів та має достатньо малу кількість точок відносно загальної кількості точок всього символу, такий сегмент вважається помилковим та прибирається з символу, а звільнені точки повністю призначаються іншому символу, що володіє більшою частиною даної мітки сегментації. Другий алгоритм працює за схожою логікою, проте аналізує сусідні зв'язки вже між окремими словами, та має більші коефіцієнти на кількості точок у сумнівних сегментах для виправлення помилок сегментації слів у реченні.

Впровадження цих алгоритмів дало змогу виправляти дрібні похибки в результатах сегментації та покращило загальну якість сегментації, що є важливим для ефективної роботи всієї системи вирівнювання рукописного тексту.

2.5 Висновки до розділу

В цьому розділі було представлено систему розпізнавання та сегментації рукописного введення.

Досліджено загальну ідею рекурентних нейронних мереж, використання блоків довгої короткочасної пам'яті, концепцію двонаправленості РНМ. Детально розглянуто структуру блока ДКЧП. В якості основної мережі для класифікації символів було використано мережу з двома прихованими шарами ДДКЧП. Такий тип мережі дозволяє ефективно вирішувати задачі аналізу та розпізнавання часових послідовностей. Вічкові з'єднання між клітиною пам'яті та всіма вентилями всередині блоку ДКЧП дозволяють мережі вивчати залежності у вхідних послідовностях навіть

в дуже нерівномірних вхідних даних з різною часовою затримкою. Також в цій мережі було використано шар нейромережевої часової класифікації та відповідну функцію втрат, що дають змогу маркувати вхідну послідовність символів відповідно до класифікованих символів.

Для підтримки роботи системи з вхідними даними, що містять рукописні документи складної структури з блоками та колонками, текст написаний різними мовами або під великим кутом, до системи додаються додаткові компоненти, такі як класифікація вхідних штрихів, класифікація мови рукописного тексту, визначення кута нахилу рядка.

Точність результатів сегментації є критичним параметром в загальній структурі системи вирівнювання, оскільки напряду впливає на правильність подальшого структурного аналізу символів та коректність вирівнювання рукописного тексту. Тому для збільшення точності маркування вхідної послідовності загальний метод сегментації покращується за допомогою поєднання декількох складових. В систему впроваджується додаткова легковісна РНМ, що виконує корекцію попередньої сегментації. Для інтерпретації виходів з цієї мережі використовується алгоритм динамічного програмування. Основною науковою новизною є розроблені додаткові алгоритми для покращення сегментації: алгоритм приєднання відтермінованих штрихів для символів українського алфавіту та алгоритми знаходження та виправлення помилок сегментації, допущених на попередніх кроках.

Систему було натреновано для розпізнавання та сегментації рукописних символів українського алфавіту. Розроблена система має можливість отримувати більш точні результати класифікації символів та сегментації вхідної послідовності, а отже може використовуватись у загальній структурі системи вирівнювання рукописного тексту.

РОЗДІЛ 3. МЕТОД ВИРІВНЮВАННЯ З ВИКОРИСТАННЯМ СТРУКТУРНОГО АНАЛІЗУ СИМВОЛІВ

У цьому розділі проводиться опис основної частини системи вирівнювання рукописного тексту. Представлено запропонований новий метод вирівнювання, що використовує структурний аналіз символів. Базуючись на поєднанні вхідних даних з результатами розпізнавання та сегментації, отримання яких детально описано в розділі 2, є можливість проводити аналіз геометрії окремих символів, що і являє собою основу структурного аналізу.

Завдання структурного аналізу полягає у визначенні у-координати для кожного символа, яка буде використана для встановлення всіх символів рядка на одну горизонталь (базову лінію) для отримання вирівняного тексту.

Отримуючи на вхід результати розпізнавання та посимвольної сегментації рукописного тексту, система будує відповідність наборів вхідних точок до кожного розпізнаного символа. Базуючись на цьому, проводиться структурний аналіз часових рядів послідовностей точок з використанням індивідуальних алгоритмів для кожного специфічного символа та/або груп символів, застосовуються методи обчислювальної геометрії, апроксимації послідовностей, дослідження екстремумів у двовимірному просторі.

Цей розділ включає в себе представлення самого методу та загальної структури розробленої системи, розгляд процесу попередньої обробки вхідних даних, що виконується за допомогою різних методів, представлення метрик текстового рядка, запропонованого розбиття символів на класи, виконання знаходження класів символів, обрахунку метрик всіх символів та фінальних метрик вирівняного рядка, проведення переміщення символів на знайдені відповідні позиції. Детально розглядається процес виконання кожного кроку, виконується розгляд розроблених алгоритмів та підходів, що застосовуються для структурного аналізу символів.

3.1 Опис запропонованого методу вирівнювання

В цій частині описується вперше запропонований метод вирівнювання рукописного тексту на основі розроблених алгоритмів розпізнавання та сегментації тексту, а також з використанням структурного аналізу символів, що враховує індивідуальні особливості почерку, зв'язне написання символів та дозволяє зберегти початковий стиль написання тексту.

Метод отримує на вхід послідовність вхідних штрихів рукописного рядка тексту та результати розпізнавання та сегментації цих штрихів, що містять набір всіх класифікованих символів та відповідні сегментовані частини кожного символу.

Запропонований метод визначається чіткою послідовністю наступних кроків:

1. Сформувати набори відповідних точок для кожного розпізнаного символу на основі результатів розпізнавання та сегментації, виконати попередню обробку вхідних даних рукописних штрихів.
2. Провести попереднє визначення класів символів за статичною таблицею класів символів.
3. Виконати уточнення класу для неоднозначних символів.
4. Визначити позиції основних ліній для кожного символу на основі структурного аналізу послідовностей точок.
5. Виконати об'єднання символів в групи для забезпечення підтримки зв'язного написання символів.
6. Обчислити загальні метрики для всього рядка тексту.
7. Визначити фінальну позицію вирівняного рядка тексту в документі на основі контекстної інформації про весь документ та обраної стратегії вирівнювання.
8. Виконати геометричне перетворення штрихів для переміщення всіх символів на свої нові позиції.

На виході метод повертає послідовність вирівняних штрихів рукописного рядка тексту, а також результуючі метрики рядка, що визначають його позицію та геометричні параметри.

3.2 Структура розробленої системи вирівнювання

В рамках дослідження було розроблено систему вирівнювання рукописного тексту, що базується на запропонованому методі. В цій частині описуються основні компоненти системи та представлена загальна структура побудованої системи.

Процес роботи системи вирівнювання зображено на рис. 3.1.



Рис. 3.1 – Загальна структура розробленої системи вирівнювання

Приймаючи на вхід вхідні штрихи, для подальшої роботи з цими даними необхідно поєднати їх з результатами розпізнавання та сегментації, що були отримані на попередніх кроках, для формування наборів точок, що відносяться до кожного символу. Далі проводиться статичний аналіз, базуючись на отриманих розпізнаних символах, що містяться у вхідному рядку тексту. Використовується статична таблиця класів символів для попереднього визначення класу для кожного символу, що характеризує його вертикальне положення в текстовому рядку. Для символів з різними можливими значеннями класів проводиться уточнення класу. Наступним

етапом є визначення основних ліній для кожного символу. На цьому етапі проводиться структурний аналіз послідовностей точок для точного визначення базової лінії для кожного символу. Далі виконується об'єднання символів в групи символів та обчислення основних ліній для груп. Використовуючи отриману інформацію про символи, обчислюються результуючі метрики для текстового рядка, а також визначається його фінальна позиція після вирівнювання з врахуванням контекстної інформації про весь документ. На фінальному етапі виконується вирівнювання штрихів, тобто переміщення їх в потрібні позиції. На виході система повертає набір вирівняних штрихів та результуючі метрики вирівняного рядка, що далі використовуються для вирівнювання інших об'єктів в документі. В результаті роботи системи користувач отримує рукописний текст, що вирівняно на основі обраної стратегії вирівнювання.

В наступних частинах детально розглядається кожен крок побудованої системи з описом розроблених методів та підходів.

3.3 Попередня обробка вхідних даних

Задача попередньої обробки даних полягає у підготовці вхідних даних для зручної та оптимальної роботи наступних етапів. Вхідними даними на цьому етапі є самі штрихи вхідної послідовності тексту, а також результати розпізнавання та сегментації.

До методів попередньої обробки штрихів можуть належати видалення дублюючих точок, перегруповання послідовності, нормалізація розмірів, згладжування штрихів, видалення коротких гачків, спрощення послідовності і т.д. Конкретні методи та їх порядок обирається в залежності від особливостей вхідних даних. Основні методи попередньої обробки детально розглянуті в роботах [50], [51], і мають загальну логіку виконання. Розглянемо основні важливі етапи попередньої обробки.

3.3.1 Побудова спрощених наборів точок

Кількість та щільність точок у вхідних даних сильно залежить від самого типу рукописного введення (цифровим пером або пальцем), апаратних характеристик пера та екрану. У випадках, коли сторонні фактори можуть впливати на цілісність даних введеного штриха таким чином, що в результаті не всі введені точки будуть записані та подані у вхідній послідовності, одним з кроків попередньої обробки може застосовуватися додавання пропущених точок для подальшої коректної роботи алгоритмів обробки [94]. Додавання точок виконується між сусідніми точками, що мають відстань більшу ніж певне визначене граничне значення, яке залежить від середньої відстані між всіма точками в штрихах та від висоти символів в рядку. Траєкторія для додавання пропущених точок може визначатися як лінійна у простому випадку, або ж за допомогою кривих Безьє.

Оскільки асимптотична складність роботи алгоритмів для обробки штрихів на пряму залежить від кількості точок в штрихах, є критично важливим зменшити цю кількість, водночас залишаючи ту мінімальну кількість точок, що необхідна для коректного аналізу штрихів [138]. Тобто задача полягає в апроксимації вхідних штрихів для оптимізації подальшого аналізу, з мінімальним впливом на якість та точність.

Для ефективної апроксимації штрихів з вхідних даних використовується метод Рамера-Дугласа-Пекера **Error! Reference source not found.**, [140]. Цей метод дозволяє визначити значущі точки, які зберігають форму штриху, зменшуючи кількість точок та полегшуючи подальший аналіз. Ступінь спрощення контролюється єдиним параметром ϵ , який визначає максимальну відстань між початковими точками та спрощеною кривою.

Відстань між сусідніми точками обраховується за формулою:

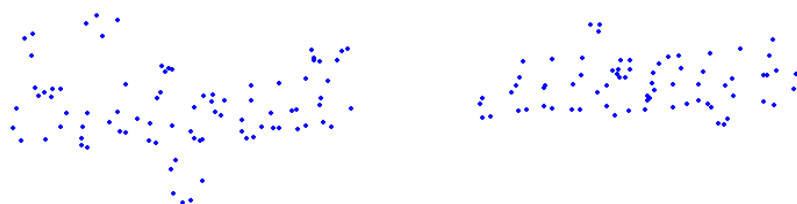
$$D = |P_i - P_{i-1}| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (3.1)$$

Асимптотична складність даного алгоритму спрощення може наближатися до квадратичної у найгірших випадках, проте в середньому випадку є меншою.

Параметр спрощення визначається з врахуванням поточної висоти вхідного рядка тексту для коректної роботи з різними наборами вхідних даних. Приклад роботи спрощення штрихів зображено на рис. 3.2.



а) поточкове представлення всіх штрихів вхідного семпла, кількість точок: 2856



б) результат апроксимації алгоритмом РДП, кількість точок: 158

Рис. 3.2 – Приклад побудови спрощеного набору точок
з вхідним рядком тексту “*блідий місяць*”

Таким чином, для даного прикладу кількість точок була зменшена приблизно в 18 разів, при збереженні основних критичних точок, які необхідні для точного визначення основних ліній кожного символу. Варто зауважити, що кількість вхідних точок семпла залежить від фізичних параметрів пристрою на якому створювався цей семпл. Зі збільшенням розмірів, роздільної здатності екрану, частоти зчитування точок екраном та стилусом, буде збільшуватись кількість точок у вхідному семплі.

Отже, спрощення точок дає можливість алгоритмам структурного аналізу отримувати дані у однаковому форматі, зважаючи на можливі відмінності по щільності точок у вхідних даних.

3.3.2 Побудова відповістей наборів точок до кожного символу

Для подальшої роботи з послідовностями точок необхідно провести їх прив'язку до символів вхідного тексту, таким чином буде отримана можливість посимвольної обробки вхідних даних.

Для цього була створена модель посимвольного зберігання інформації про послідовності точок. До кожного символу належить набір сегментів, що відповідає за послідовність точок. Кожен сегмент визначається початковою та кінцевою позиціями, які містять інформацію про індекс точки та індекс відповідного штриха з вхідного набору штрихів. Схематично ці зв'язки позначені на рис. 3.3.

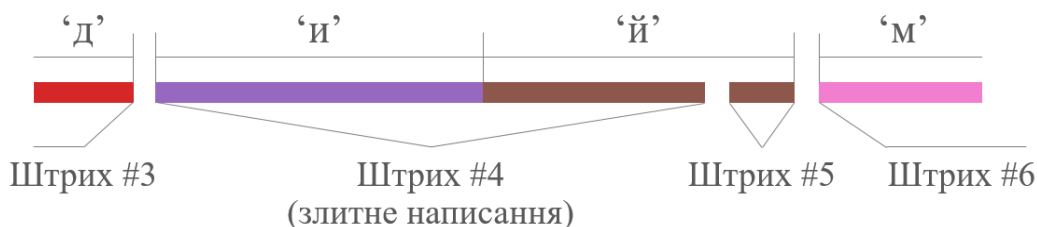


Рис. 3.3 – Модель зберігання інформації з відповідністю наборів точок до кожного символу

Результати розпізнавання та сегментації, які отримуються на вхід, містять цю інформацію про відповідність штрихів до символів. Задача цього кроку полягає в обробці цієї інформації та перетворення її в зручну для використання модель, описану вище. В результаті кожен текстовий символ буде мати набір сегментів, що відповідають цьому символу згідно до результатів сегментації. Приклад з візуалізацією інформації про сегментацію зображено на рис. 3.4.



а) результати розпізнавання та сегментації вхідного семплу



б) візуалізація сегментованих штрихів, штрих №4 має зв'язне написання та відноситься до двох символів

Рис. 3.4 – Приклад побудови відповідностей сегментів вхідних штрихів до розпізнаних символів

3.3.3 Обрахування основних ознак вхідних штрихів

Для подальшої роботи алгоритмів обробки штрихів система потребує попереднього обрахування потрібних ознак штрихів [141], [76] для загальної оптимізації роботи, оскільки ці ознаки можуть використовуватися в багатьох місцях системи, тому не повинні обраховуватися кожен раз окремо.

Для вхідного штриха S визначимо всю послідовність його точок як $\{s_i \dots s_j\}$.

Визначимо основні нотації для базових характеристик штриха S :

- u_x, u_y – одиничні вектори співнаправлені з осями x та y , вважаємо, що вісь y направлена вниз

- s_1, s_n – перша та остання точки в послідовності написання штриха
- $v = \overrightarrow{s_1 s_n}$ – визначає напрямок написання штриха
- $\|v\| = \|s_1 s_n\|$ – довжина штриха
- $\mu = \frac{1}{n} \sum_{i=1}^n s_i$ – центр тяжіння по всім точкам штриха

Вектор напрямку штриха дає інформацію про загальну динаміку напису штриха. Визначається на основі першої та останньої точок штриха та залежить від відстані між ними, обраховується за формулами:

$$DV_x = \frac{\overrightarrow{v_x} * \overrightarrow{u_x}}{\|v\|}, VN_y = \frac{\overrightarrow{v_y} * \overrightarrow{u_y}}{\|v\|} \quad (3.2)$$

Кут початкового вектора визначається на основі перших точок траєкторії штриха, за умови якщо їх не менше трьох. Вектор між першою та третьою точками: $w = \overrightarrow{s_1 s_3}$. Для обрахунку використовується вже апроксимований набір точок штриха. Формули кута початкового вектора:

$$IA_x = \frac{\overrightarrow{w_x} * \overrightarrow{u_x}}{\|w\|}, PK_y = \frac{\overrightarrow{w_y} * \overrightarrow{u_y}}{\|w\|} \quad (3.3)$$

Якщо попередні ознаки є динамічними характеристиками штриха, що визначаються самим процесом написання штриха, то наступні ознаки є візуальними, які фокусуються на зовнішньому вигляді результату написання.

Середнє відхилення всіх точок штриха є ознакою, що не залежить від напрямку написання, та визначається відносно центру тяжіння штриха μ за формулою:

$$GD = \frac{1}{n} \sum_{i=1}^n \|s_i \mu\| \quad (3.4)$$

Усереднений напрямок штриха обраховується на основі всіх його відрізків за наступною формулою:

$$AD = \frac{1}{n-1} \sum_{i=1}^{n-1} \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \quad (3.5)$$

Ступінь кривизни штриха визначається обрахунком всіх локальних значень кривизни. Визначення кута θ_i в кожній точці штриха, окрім першої та останньої, визначається враховуючи 2 відрізки, що з'єднуються в цій точці, та обраховується за формулою:

$$\theta_i = \arccos\left(\frac{\overrightarrow{s_{i-1}s_i} \cdot \overrightarrow{s_i s_{i+1}}}{\|\overrightarrow{s_{i-1}s_i}\| \|\overrightarrow{s_i s_{i+1}}\|}\right) \quad (3.6)$$

На основі значень по всьому штриху визначається середнє значення кривизни штриха:

$$CU = \sum_{i=2}^{n-1} \theta_i \quad (3.7)$$

Відповідно, для штриха, що являє собою пряму лінію, ступінь кривизни буде дорівнювати нулю. Для штрихів з великою кількістю поворотів траєкторії цей критерій буде мати великі значення.

З отриманих вище значень визначається ступінь перпендикулярності, що може виявляти специфічні зміни траєкторії штриха, такі як прямі кути, обраховується на формулою:

$$PE = \sum_{i=2}^{n-1} \sin^2(\theta_i) \quad (3.8)$$

Максимальний кут зміни траєкторії штриха дає змогу виявляти різкі зміни траєкторії, що також є важливим параметром в алгоритмах обробки штрихів:

$$MA = \max_{i=2}^{n-1} \theta_i \quad (3.9)$$

3.4 Метрики текстового рядка та розбиття символів на класи

Для опису характеристик позицій та розмірів рядка є потреба ввести метрики текстового рядка. Для цієї задачі в даній роботі використовуються 4 основні горизонтальні лінії з наступними назвами: верхня лінія, ікс-лінія, базова лінія, нижня лінія. Геометричне положення текстового рядка та його розміри можна представити за допомогою цих основних ліній: верхня та нижня лінії охоплюють усі символи, які присутні в текстовому рядку зверху та знизу відповідно, лінія ікс-висоти обмежує маленькі символи зверху, а базова лінія визначає основу, на якій розташовуються всі символи.

3.4.1 Визначення класів символів

На базі основних ліній рядка виконано розділення всіх символів на класи, що визначають очікуване вертикальне розташування кожного символу: *‘Високий’*, *‘Великий’*, *‘Низький’*, *‘Довгий’*, *‘Верхній’*, *‘Центральний’*, *‘Нижній’*. Представлення класів символів показано на рис. 3.5.

Для всіх символів кожної підтримуваної мови створюється окрема таблиця відповідності кодів символів до класів, можлива реалізація: хеш-таблиця. Тоді асимптотика визначення класу символу буде $O(1)$.

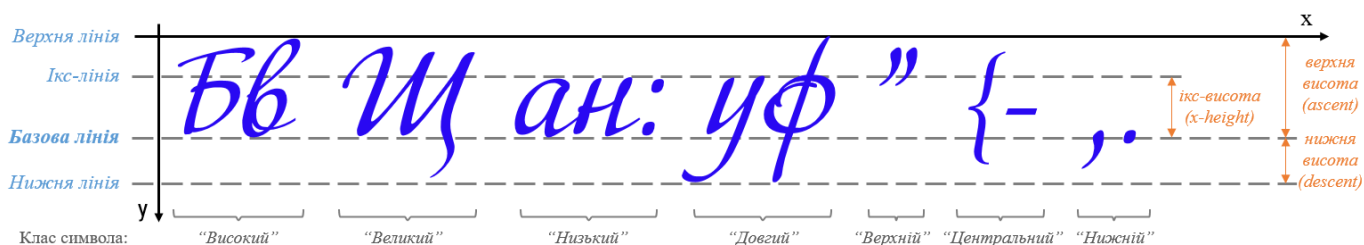


Рис. 3.5 – Визначення основних ліній текстового рядка (позначені синім),
результуючих метрик рядка (позначені помаранчевим) та визначення класів
символів

Наприклад, клас *‘Високий’* означає, що символ має бути розташований між верхньою лінією та базовою лінією, це може бути будь-яка велика літера, цифра або великий символ який знаходиться на базовій лінії, наприклад *‘&’*.

Символи з класом *‘Великий’* мають складнові штрихи що можуть розташовуватися між верхньою та нижньою лініями, це символи *‘Ц’* та *‘Щ’*.

‘Низькі’ символи — це звичайні маленькі літери та символи, що розташовуються між лінією ікс-висоти та базовою лінією, тобто *‘а’*, *‘з’*, *‘л’*, і т.д.

Клас *‘Довгий’* представляє символи з очікуваним розташуванням між ікс-лінією та нижньою лінією, такими символами є наприклад *‘у’*, *‘ф’*, *‘р’*.

Деякі знаки пунктуації, такі як крапки та коми, мають клас *‘Нижній’*, тобто вони повинні знаходитись прямо на базовій лінії.

Символи апостроф, зірочка та лапки належать класу *‘Верхній’*, тому їхнє очікуване розташування може бути між верхньою лінією та ікс-лінією, залежно від розміру символу та типів суміжних символів.

Клас *‘Центральний’* призначається символам, що мають бути відцентровані вздовж осі, такі як математичні оператори, різні типи дужок тощо.

3.4.2 Уточнення класів символів

В деяких випадках клас символу може відрізнитись залежно від індивідуального стилю написання, тому не всі символи мають однозначний визначений клас. Прикладом є літера *‘з’*, яка може мати клас *‘Довгий’*, або ж клас *‘Низький’* з іншим стилем рукописного написання (рис. 3.6), тому її не можна використовувати для визначення основних ліній символу без додаткового уточнення його класу.

клас літери *‘з’*: *“Довгий”*



клас літери *‘з’*: *“Низький”*



Рис. 3.6 – Приклад написання літери *‘з’* в різному стилі

Уточнення класів для неоднозначних символів виконується на основі аналізу геометричних ознак символу та його розташування відносно сусідніх символів для

точного визначення класу. Символи що потребують уточнення класу: ‘з’ (“Довгий” / “Низький”), ‘д’ (“Високий” / “Довгий”).

Псевдокод алгоритму уточнення класу символа по його розташуванню наведено нижче.

Алгоритм 3.1: Уточнення класу символа по його розташуванню

```

if isNeighbourClass == Високий or Низький then
    if symbolBottom – neighboursBottom > 0.3 * symbolHeight then
        | symbolClass = Довгий;
    else
        | symbolClass = Низький;
    end
else if isNeighbourClass == Довгий then
    if neighboursBottom – symbolBottom < 0.5 * symbolHeight then
        | symbolClass = Довгий;
    else
        | symbolClass = Низький;
    end
else
    | symbolClass = GetClassByGeometry(symbol);
end
/* Символ не має сусідніх літер */

```

3.5 Визначення основних ліній для кожного символа

Основною задачею даного етапу є визначення базової лінії для кожного символа, що дасть можливість дізнатись значення зміщення положення кожного символа відносно загальної базової лінії та виконати вирівнювання. Додатково відбувається визначення інших основних ліній в тих випадках, коли це можливо, що допомагає збільшити точність визначення результуючих метрик рядка та покращити якість вирівнювання.

Процедура визначення основних ліній символів застосовує 3 окремі методи, які застосовуються послідовно один за одним, уточнюючи основні лінії символа, якщо попередній метод не зміг визначити їх остаточно. Виконання всіх трьох методів гарантує, що кожен символ в результаті отримає значення базової лінії.

3.5.1 Визначення на основі класу символу

На цьому етапі для кожного символу визначаються попередні значення основних ліній, які можна отримати з класу символу. На основі класу можемо визначити не більше 2 ліній для символу – це верхня та нижня сторони прямокутника, що обмежують всі сегменти символу. Відповідність основних ліній, які можуть бути визначені, до границь прямокутника символу показані в таблиці 3.1.

Таблиця 3.1 – Визначення основних ліній символу на основі класу

Клас символу	Верхня сторона прямокутника	Нижня сторона прямокутника
<i>‘Високий’</i>	Верхня лінія	Базова лінія
<i>‘Великий’</i>	Верхня лінія	Нижня лінія
<i>‘Низький’</i>	Ікс-лінія	Базова лінія
<i>‘Довгий’</i>	Ікс-лінія	Нижня лінія
<i>Інші</i>	Верхня лінія	Нижня лінія

Наприклад, для великої літери *‘Д’* можемо визначити значення верхньої лінії та базової лінії. Для маленької літери *‘ц’* – ікс-лінію та нижню лінію. Для символу *‘Ю’* – значення верхньої та нижньої ліній. Приклади зображені на рис. 3.7.

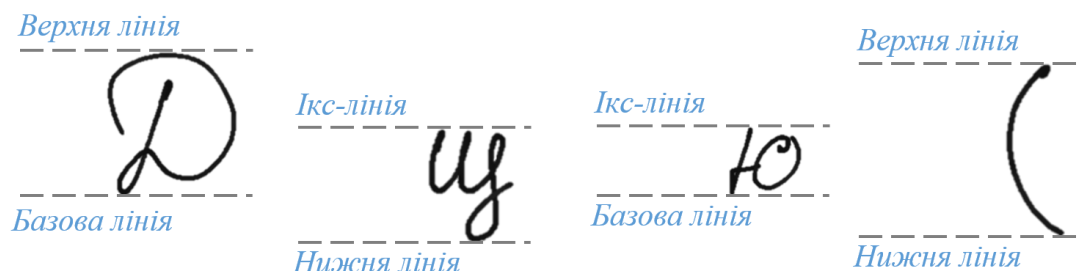


Рис. 3.7 – Приклади визначення основних ліній базуючись на класі символу

Проте значення основних ліній включно з базовою лінією, що обчислені цим методом, є достатніми лише для частини символів, які не мають опціональних та

додаткових штрихів що змінюють форму та структуру символу. Прикладами таких символів є літери 'o', 'b'. Для іншої частини символів необхідне більш точне визначення основних ліній.

3.5.2 Структурний аналіз для визначення основних ліній

На основі визначених класів символів визначаються попередні показники основних ліній символів. Однак велика частина символів потребує додаткового аналізу для більш точного визначення базової лінії. Причиною є індивідуальні особливості написання, коли одні й ті ж символи мають різні варіанти написання, що відрізняються кількістю штрихів, кривих, локальних екстремумів і критичних точок. Ці особливості написання можуть впливати на розміщення базової лінії в символі, відповідно більш точне її визначення буде покращувати якість вирівнювання. Приклади визначення основних ліній символів на основі додаткового структурного аналізу символів зображено на рис. 3.8. Бачимо, що частина символів не може отримати точне визначення базової лінії лише на основі свого класу, та потребує більш точного визначення основних ліній через індивідуальні особливості написання, а саме додаткові штрихи, хвости, завитки тощо.

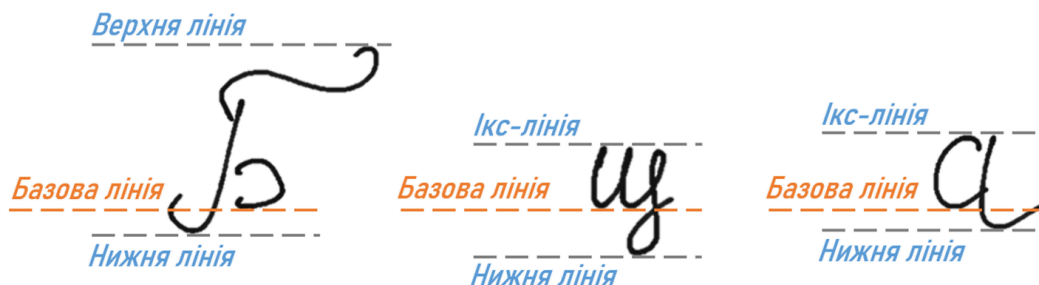


Рис. 3.8 – Приклади визначення основних ліній на основі структурного аналізу символів

Для кожного особливого символу або групи символів були розроблені індивідуальні алгоритми аналізу геометричних ознак, що охоплюють більшість стилів написання. Опис цих алгоритмів визначено в таблиці 3.2.

Таблиця 3.2 – Алгоритми структурного аналізу для наборів символів

Особливість набору символів	Алгоритм визначення основних ліній	Символи
Великі символи з опорою на визначену очікувану кількість кінцівок	Визначення базової лінії на основі середнього значення по N найбільшим локальним екстремумам	<i>А, Ж, К, Л, М, Н, П, Т, Х, Ш, Ю, Я</i>
Маленькі символи з опорою на визначену очікувану кількість кінцівок		<i>ж, к, л, м, н, п, т, х, ш, ю, я</i>
Великі символи з можливим великим гачком зліва	Визначення базової лінії на основі ітеративного пошуку найбільш підходящого екстремуму по набору параметрів (розмір, положення, кути)	<i>Б, В, Д</i>
Маленькі символи з можливим довгим хвостом справа		<i>а, и, ѣ</i>
Символи класу 'Довгий' з вираженою основною овальною частиною	Визначення базової лінії на основі пошуку найбільшої верхньої овальної частини по розміру та кутам	<i>д, р, у</i>
Символи класу 'Довгий' з невираженою основною овальною частиною	Визначення базової лінії на основі пошуку хвостової частини за допомогою аналізу кутів	<i>ф, з, ц, щ</i>
Символи з діакритичним знаком	Визначення ікс-лінії на основі пошуку та відсікання діакритичної частини символа	<i>і, ї, ѣ</i>

Алгоритми, що описані в таблиці, визначаються спільно для наборів символів, проте можуть регулюватись індивідуально для кожного символа за допомогою підбору коефіцієнтів та інших параметрів для основних ознак, які використовуються на етапі аналізу структури символа. Набір цих алгоритмів формує собою метод вирівнювання для символів тексту, написаного українською мовою.

3.5.3 Аналіз локальних екстремумів

Частина символів мають структуру, для якої характерним є опирання символа на базову лінію за допомогою або нижніх кінцівок символа, або його основної частини овальної форми. Для таких символів були розроблені алгоритми, які використовують аналіз екстремумів за віссю Y.

Введемо поняття U-кривої, яка використовується при аналізі локальних екстремумів. Для кожного локального максимуму, який не представляє закінчення штриха, існують два сусідніх мінімуми, які будуть визначати одну U-криву. Завдання полягає в тому, щоб знайти цільову U-криву, яка містить базову лінію цього символа. Це можна зробити шляхом ітераційного відкидання невідповідних локальних максимумів, поки не залишиться лише один. У цьому пошуку використовуються такі параметри U-кривої, як її ширина, висота, положення, внутрішні кути тощо. Якщо в кінці алгоритму пошуку залишається більше однієї кривої, для визначення базової лінії використовується нижній лівий (або правий) локальний максимум в залежності від типу символа. Цей алгоритм використовується для символів 'a', 'u' та інших.

Для символів із очікуваною кількістю нижніх кінцівок (символи 'л', 'н', 'М', 'П' тощо) використовується аналіз локальних максимумів, де базова лінія обчислюється як середнє N локальних максимумів, обраних як нижні кінці символу.

Символи, які можуть мати відділені хвости, наприклад 'ц', 'Щ', спочатку перевіряються на наявність окремих штрихів, які належать до хвоста. У випадку наявності, ці штрихи відхиляються, а базова лінія визначається за алгоритмом аналізу очікуваної кількості кінцівок, для символів що залишились і являють собою основну частину символу. У іншому випадку, маємо ситуацію що символ написаний одним штрихом. Тоді проводиться аналіз U-кривих, вони порівнюються за розмірами та позицією для визначення тієї, що належить хвосту, для її відсікання.

Символи 'd', 'p', 'y' мають схожі особливості, що дозволяє їм використовувати спільні частини алгоритмів аналізу екстремумів. Детально розглянемо одну з можливих реалізацій алгоритму визначення базової лінії на основі пошуку найбільшої

верхньої овальної частини по розміру та кутам на прикладі маленької літери ‘у’. Якщо ‘у’ написана двома прямими штрихами, базова лінія визначається як їх перетин. В іншому випадку використовується аналіз екстремумів з ітераційним відкиданням, і в результаті другий за рахунком невідкинтий локальний максимум використовується як базова лінія. Псевдокод алгоритму приведено нижче.

Алгоритм 3.2: Визначення базової лінії по основній овальній частині

```

extremums = findAllLocalExtremums(symbol.approxStrokes);
extremums = filterExtremumsBySize(extremums);
if extremums.max.size < 2 then
    | return not success;                                /* Символ з 1 екстремумом - вирівнювання по сусіднім */
end
else if extremums.max.size == 2 then
    | if symbol.strokes.size > 1 then                        /* Базова лінія по перетину штрихів */
        | return findBaselineByIntersection(symbol);
    | else                                                    /* Базова лінія по другому максимальному екстремуму */
        | symbol.baseline = extremums.secondMax;
    | end
else if extremums.max.size > 2 then
    | i = 1;
    | maxUCurve = i;
    | maxUCurveHeight = 0;
    | while i ≠ extremums.max.size do
        | if extremums.max > baselineThreshold then
            | | continue;                                /* Пропускаємо екстремуми в нижній частині символу */
        | end
        | uCurvePrevHeight = extremums.max[i] - extremums.max[i].previousMin;
        | uCurveNextHeight = extremums.max[i] - extremums.max[i].nextMin;
        | uCurveHeight = max(uCurvePrevHeight, uCurveNextHeight);
        | if uCurveHeight > maxUCurveHeight then
            | | maxUCurve = i;
            | | maxUCurveHeight = uCurveHeight;          /* Знаходимо екстремум з найбільшою U-кривою */
        | end
    | end
    | symbol.baseline = extremums[maxUCurve];
end
return success;

```

Крім базової лінії визначаються також інші показники основних ліній, коли це можливо. Наприклад, символ ‘і’ та інші символи з діакритичними знаками використовуються для визначення лінії ікс-висоти, знаходячи та відокремлюючи штрихи крапок від штрихів основної частини.

3.5.4 Аналіз сусідніх символів

Для деяких специфічних стилів написання деяких символів, структура та форма яких не містить конкретних точок, що визначають базову лінію символу, неможливо знайти точну позицію базової лінії на основі структурного аналізу лише одного цього символу. Також ця ситуація може виникнути у випадках, коли алгоритми структурного аналізу зіткнулись з неможливістю коректно завершити своє виконання. Прикладами таких ситуацій є специфічний запис певних символів, коли символ не містить локальних максимумів в тих позиціях, на яких вони повинні бути за традиційними правилами написання. Приклад такого написання літери 'р' зображено на рис. 3.9.



Рис. 3.9 – Приклад визначення базової лінії для специфічного написання символу 'р' в слові "крок" на основі сусідніх символів

В таких випадках застосовується знаходження коректної позиції символу на основі аналізу сусідніх символів. Виконується обчислення базової лінії таким чином, щоб вирівняти цей символ відповідно до позицій базових ліній попередньо вирівняних сусідніх символів, або ж до лінії ікс-висоти, у випадку відсутності сусідніх символів, у результаті вирівнювання яких ми впевнені.

3.5.5 Згладжування позицій символів в рядку

Розроблено набір алгоритмів для пошуку можливих помилок обчислення основних ліній та покращення якості вирівнювання тексту шляхом згладжування позицій символів з врахуванням індивідуальних особливостей написання.

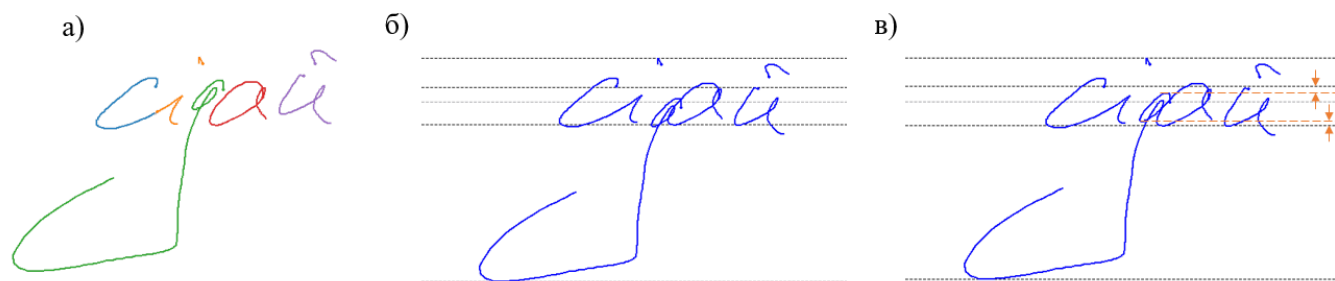


Рис. 3.10 – Приклад роботи згладжування позиції символу ‘d’: а) вхідні штрихи; б) вирівнювання штрихів виконано по визначеній базовій лінії символів; в) позицію символу ‘d’ згладжено на основі однакової відстані від основної частини символу до ікс-лінії та до базової лінії рядка (помаранчеві стрілки)

На рис. 3.10 зображено приклад роботи згладжування позиції символів. На основі загальних правил вирівнювання та визначених базових ліній для всіх символів рядка, символ ‘d’ було вирівняно по його базовій лінії. Проте алгоритм згладжування виявив занадто різкий стрибок символу ‘d’ через низьку висоту його основної частини. Позицію символу було скореговано, встановленням однакової відстані від основної частини символу до ікс-лінії та до базової лінії рядка.

3.6 Підтримка зв’язного написання

Для підтримки окремого складного випадку - зв’язного написання декількох символів без відриву пера, введемо поняття групи символів. Група символів – це мінімальний набір символів, які не мають спільних штрихів з жодною іншою групою.

Розбиття таких символів для окремого переміщення спотворить оригінальну послідовність написання та індивідуальність почерку, тому ми об’єднуємо їх в одну групу. Всі символи групи вирівнюються цілісно, для уникнення розривів зв’язків у зв’язному написанні. Використовуючи аналіз функцій символів з попереднього кроку, ми обчислюємо загальну метрику групового тексту за спеціальними правилами об’єднання. Нова базова лінія групи обчислюється як середнє базових ліній всіх символів групи. Інші лінії визначаються як мінімум для верхньої лінії та ікс-лінії, та

максимум для нижньої лінії. Розрахунок основних ліній групи при об'єднанні символів виконується за формулами:

$$\text{ВерхняЛініяГрупи} = \min_{i \in \text{Група}} \text{ВерхняЛініяСимвола}_i \quad (3.10)$$

$$\text{ІксЛініяГрупи} = \min_{i \in \text{Група}} \text{ІксЛініяСимвола}_i \quad (3.11)$$

$$\text{БазоваЛініяГрупи} = \text{avg}_{i \in \text{Група}} \text{БазоваЛініяСимвола}_i \quad (3.12)$$

$$\text{НижняЛініяГрупи} = \max_{i \in \text{Група}} \text{НижняЛініяСимвола}_i \quad (3.13)$$

Розглянемо визначення основних ліній на прикладі рядка “забезпечення основних”, виконаного зі зв'язним написанням символів (рис. 3.11).

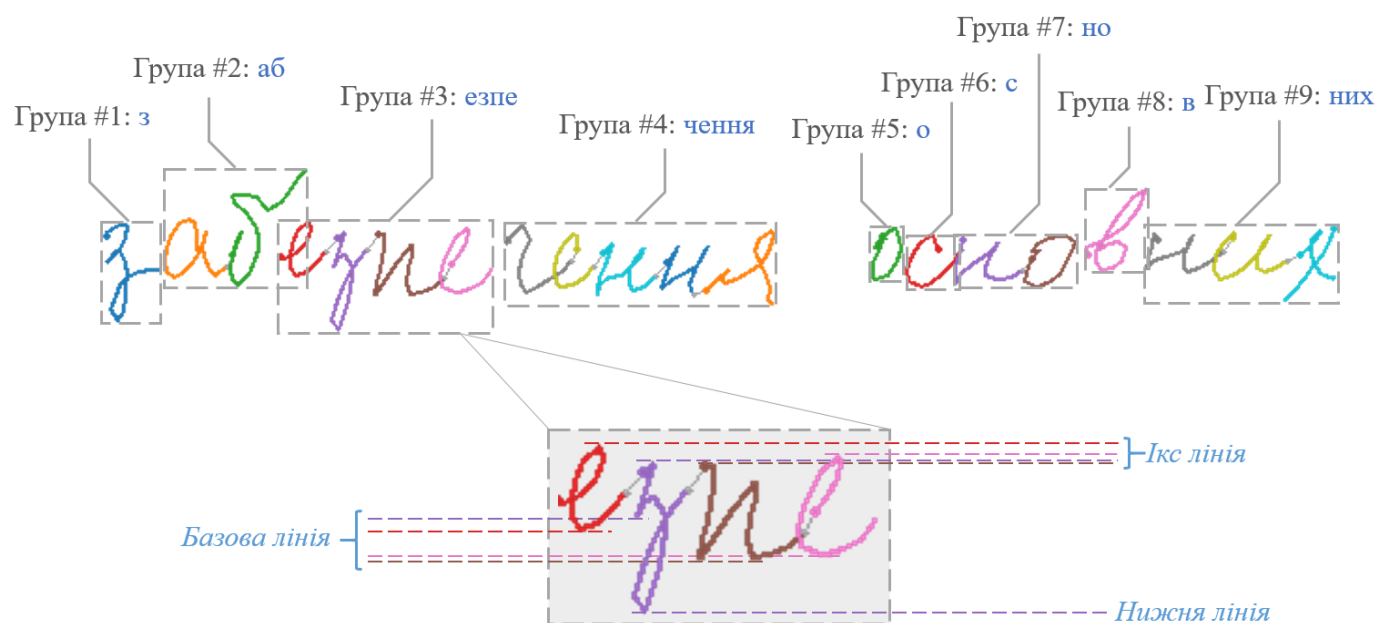


Рис. 3.11 – Приклад визначення основних ліній при об'єднанні символів в групи

Бачимо, що розрахунок значення базової лінії для групи #3, що складається з 4 символів - “езпе”, виконується на основі всіх значень базової лінії, визначених для кожного символу на попередніх етапах. Таким чином, базова лінія групи буде

середнім арифметичним всіх значень базових ліній символів цієї групи. Аналогічно обраховується значення ікс-лінії групи, проте береться вже не середнє значення, а мінімальне значення усіх ікс-ліній складових символів, оскільки ікс-лінія групи повинна визначати найвищу геометричну точку, яка обмежує всі літери та символи маленького регістру. Значення нижньої лінії в даному прикладі визначається тільки на основі символа 'з', оскільки це єдиний символ що має значення цієї лінії.

Використання груп символів у цьому алгоритмі є одним із ключових компонентів, що дозволяє зберегти стиль почерку користувача та зберегти природність вирівняних штрихів. Алгоритм гарантує, що базова лінія буде призначена для кожної групи, оскільки це має вирішальне значення для наступних етапів правильного визначення зсуву групи відносно загальної базової лінії.

Отже, використовуючи результати попередніх етапів, базуючись на попередньо визначених класах символів зі словника, уточненні класів, аналізі геометричних особливостей кожного символа та об'єднанні символів в групи, ми отримуємо точні значення основних ліній для кожної групи символів.

3.7 Визначення результируючих метрик рядка

Основним завданням цього кроку є отримання результируючих метрик текстового рядка, які є основними характеристиками, що описують вирівняний текстовий рядок. Вони представлені у формі абсолютних значень для визначення вертикальних розмірів ліній і не містять інформації про положення, за винятком загальної базової лінії.

Метрики текстового рядка обчислюються з врахуванням значень основних ліній всіх його складових груп. Значення верхньої лінії визначається як максимальна різниця між положеннями базової лінії та положеннями верхніх ліній в усіх групах. Значення нижньої лінії — це максимальна різниця між нижніми лініями груп і базовими лініями. Значення ікс-лінії визначається як середнє між базовими лініями груп і ікс-лініями. Цільове положення базової лінії для вирівнювання всього рядка

обраховується як середнє між базовими лініями всіх груп. Відповідні формули наведені нижче:

$$\text{ВерхняВисота} = \max_{i \in \text{Рядок}} (\text{БазоваЛініяГрупи}_i - \text{ВерхняЛініяГрупи}_i) \quad (3.14)$$

$$\text{НижняВисота} = \max_{i \in \text{Рядок}} (\text{НижняЛініяГрупи}_i - \text{БазоваЛініяГрупи}_i) \quad (3.15)$$

$$\text{ІксВисота} = \text{avg}_{i \in \text{Рядок}} (\text{БазоваЛініяГрупи}_i - \text{ІксЛініяГрупи}_i) \quad (3.16)$$

3.8 Визначення фінальної позиції та вирівнювання рядка

Отримані метрики текстового рядка можна використовувати для коректного вирівнювання декількох рядків та регулювання компонентів всього текстового документа в цілому, беручи до уваги різні параметри, такі як середня / максимальна висота символів, розміри / положення попередніх рядків тексту, статичний інтервал між рядками, інтерфейсні компоненти документа, інші типи об'єктів, такі як таблиці, списки, картинки і т.д., та інші фактори. Залежно від вибраної стратегії вирівнювання, текстові рядки можна розташувати таким чином, щоб підтримувати однаковий міжрядковий інтервал, однаковий інтерліньяж (відстань між базовими лініями), уникнення накладення рядків, вміщувати рядки в мінімальну обмежену область тощо.

Після визначення остаточного місця розташування лінії розраховане положення базової лінії використовується для виконання вирівнювання лінії, таким чином процедура зміщення штрихів для текстового рядка виконується лише раз, відразу на потрібні місця. Для кожної групи ліній значення зміщення обчислюється як різниця між цільовою базовою лінією та базовою лінією групи. Відповідні штрихи групи регулюються цим зміщенням на їхнє нове вирівняне положення. Приклади вирівнювання тексту за різними стратегіями вирівнювання наведені на рис. 3.12.



Рис. 3.12 – Приклад вирівнювання текстових рядків з різними стратегіями вирівнювання документу: а) вхідний текст; б) вирівнювання з мінімізацією загального розміру та уникненням перетинання штрихів ліній; в) вирівнювання зі збереженням однакового міжряджкового інтервалу; г) вирівнювання зі збереженням однакового значення інтерліньяжу (відстані між базовими лініями кожного рядка)

В результаті, виконавши процедуру переміщення кожного штриха вхідної послідовності текстового рядка на нову обраховану позицію для нього, ми отримуємо новий набір штрихів, який є вихідними даними системи вирівнювання. Додатковим виходом системи є результуючі метрики рядка, що визначають його позицію в документі та геометричні розміри.

Таким чином, користувач системи отримує вирівняний рядок рукописного тексту, який було встановлено у визначену позицію в документі на основі заданої ним стратегії вирівнювання, що використовує контекстну інформацію про весь документ та задані параметри вирівнювання.

3.9 Висновки до розділу

В цьому розділі було представлено основну складову частину системи – метод вирівнювання рукописного тексту, що базується на використанні структурного аналізу символів.

На початку розділу описано процес попередньої обробки даних, що включає в себе підготовку вхідних штрихів для їх подальшої ефективної та оптимальної обробки алгоритмами, а також обробку результатів розпізнавання та сегментації, які приймаються на вхід, для формування відповідностей наборів штрихів до розпізнаних символів.

Було представлено розділення літер українського алфавіту та символів на 7 основних класів, які визначають очікуване вертикальне розташування кожного символу в рядку тексту. Запропоновано алгоритми для уточнення класів деяких символів, які можуть мати різне розташування, залежно від написання. Визначено основні складові структурного аналізу та наведено загальну схему роботи системи. Ядром структурного аналізу є набір розроблених алгоритмів для знаходження основних ліній для кожного символу, що базується на результатах сегментації, та при визначення результату враховує індивідуальні особливості написання та різні варіації почерків.

Запропоновано ідею поєднання символів в групи символів, що дозволяє коректно вирівнювати зв'язно написані послідовності символів. На основі поєднання всіх результатів з попередніх кроків, обчислюються результуючі метрики текстового рядка та визначення його фінальної позиції. Розроблена система дає можливість вирівнювати набір рядків тексту за наперед визначеною стратегією. Можливі стратегії визначаються правилами розміщення вирівняних ліній, наприклад можуть вирівняти текст по заданому міжрядковому інтервалу, або ж виконати мінімізацію загального розміру тексту, звільнивши таким чином вільне місце в документі. Стратегії вирівнювання також можуть враховувати розміри та положення інших об'єктів в

документі, наприклад малюнків, графіків, діаграм, та регулюватись параметрами від користувача.

В результаті розроблено метод вирівнювання тексту, написаного українською мовою, особливість якого полягає у використанні алгоритмів, що визначаються окремо для кожного символу, та дозволяє реалізувати вирівнювання рукописних літер для різних стилів написання. Також результатом даного розділу є спроектована та розроблена система для структурного аналізу рукописного тексту, що дозволяє виконувати поставлену задачу вирівнювання. Аналіз результатів роботи розробленої системи проводиться в наступному розділі, разом з інтеграцією системи в мобільний застосунок та перевіркою виконання основних вимог, поставлених в дисертаційній роботі.

РОЗДІЛ 4. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

4.1 Опис експериментальних даних

Для проведення експериментів та перевірки роботи розробленої системи важливо використати тексти різних людей з різними почерками та стилями написання. Пошук наявних наборів даних українських текстів, виконаних цифровими чорнилами, у відкритому доступі результатів не дав. Тому було вирішено використати внутрішній закритий набір даних, що належить компанії «Самсунг РнД Інститут Україна». Частина з цього набору використовувалася під час тренування нейронних мереж для розпізнавання та сегментації, інша частина була виділена для тестування розроблених методів розпізнавання та вирівнювання.

Тестовий набір даних складається з 7432 семплів, що містять один рядок рукописного тексту українською мовою. У створенні набору приймали участь люди різного віку та статі, відповідно у ньому забезпечено різномайття рукописних стилів та почерків, що є важливим для нашої задачі. Детальний опис характеристик набору даних наведено у табл. 4.1.

Таблиця 4.1 – Опис характеристик тестового набору даних

	Мін.	Середня	Макс.
Довжина текстового рядка	3	20.34	71
Кількість штрихів	2	14.27	63
Кількість точок	39	487.57	2040
Кількість семплів	7432		

В тестовому наборі даних присутня кожна з літер українського алфавіту, розподіл кількостей кожної літери наведено на рис. 4.1. Бачимо, що кількість появ літери ‘г’ є найменшою серед усіх, що є очікуваним через її рідкість зустрічання в більшості слів. Ця та інші літери, що мають низьку кількість появ в тестовому наборі

даних, будуть потребувати додаткової ручної перевірки для пересвідчення в корестності роботи розроблених алгоритмів.

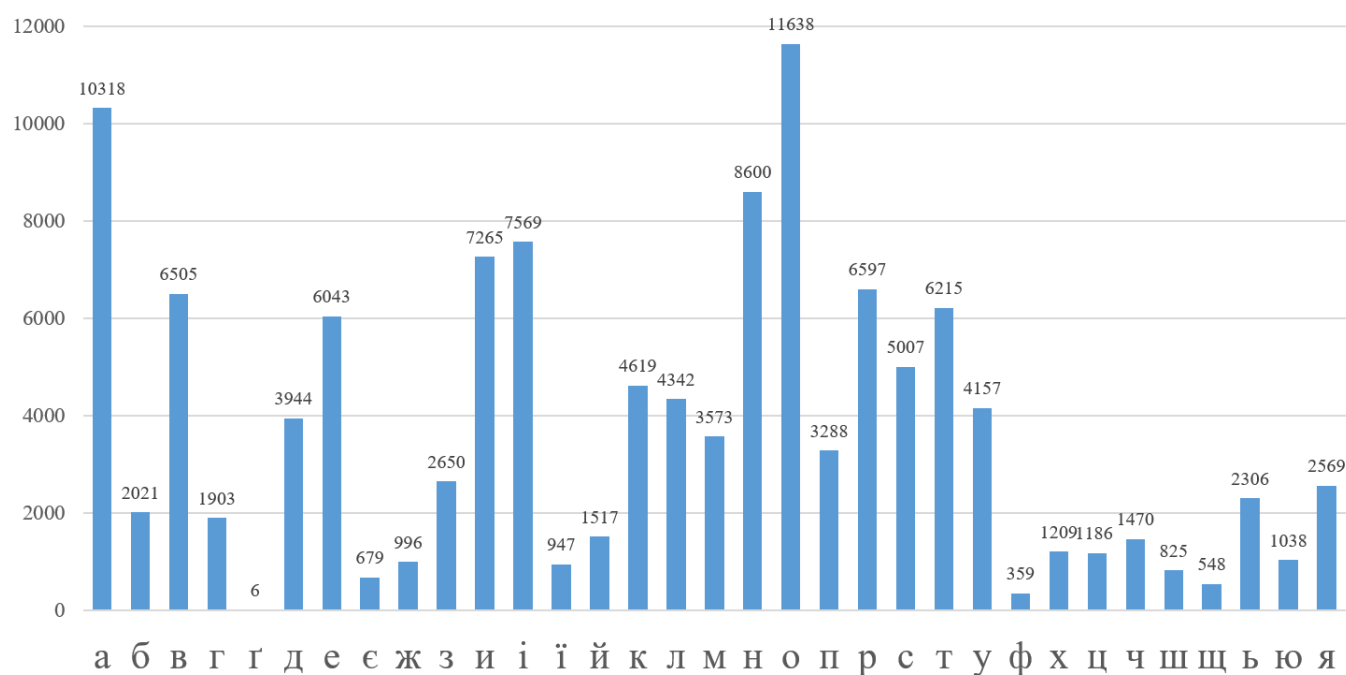


Рис. 4.1 – Розподіл кожної літери українського алфавіту по кількості зустрічань в тестовому наборі даних

Набір даних містить семпли з різною висотою текстових рядків та штрихів, тому для автоматизації процесу перевірки результатів вирівнювання важливим є нормування штрихів. Для цього було використано нормування всіх семплів з приведенням їх медіанного штриха до висоти 30. Таким чином, кожен семпл в результаті може мати різну висоту, проте висота більшості штрихів буде знаходитись біля цього визначеного значення.

4.2 Точність системи розпізнавання

Використане рішення для виконання розпізнавання та сегментації символів, що описується в розділі 2, було протестовано для оцінки точності розпізнавання.

Для оцінки точності класифікації символів використано найбільш розповсюджені та зручні метрики оцінки, а саме влучність (precision), повноту (recall)

та F1-міру. Кожна з метрик була застосована для визначення якості розпізнавання по окремим символам, по словам та по всьому реченню вхідного семпла. Отримані результати наведено в табл. 4.2.

Таблиця 4.2 Оцінка точності системи розпізнавання

Метрика	Тестовий набір даних
Влучність по символам	98.31 %
Повнота по символам	97.88 %
F1-міра по символам	98.1 %
Влучність по словам	87.86 %
Повнота по словам	89.19 %
F1-міра по словам	88.52 %
Точність розпізнавання символів	97.73 %
Точність розпізнавання слів	87.39 %
Точність розпізнавання речення	73.04 %

Також було перевірено гіпотезу, що система розпізнавання може покращити якість розпізнавання, якщо перед подачею семплів на розпізнавання проводити попереднє вирівнювання рукописного тексту. Для цього тестовий набір даних було вирівняно розробленою системою вирівнювання та виділено в окремий вирівняний набір даних, після цього обидва набори даних було оброблено системою розпізнавання для отримання та порівняння результатів класифікації. Результат розпізнавання на вирівняних семплах не покращився. Це пояснюється різницею у природі даних між семплами, на яких мережа проходила навчання, та які прийшли їй на вхід після вирівнювання, з потенційно втраченими важливими просторовими зв'язками.

4.3 Метрики для оцінювання вирівнювання тексту

Оцінка якості вирівнювання рукописного тексту сама по собі є складною задачею, оскільки параметр рівності тексту не є вимірюваною величиною, відповідно важко отримати цільові значення, з якими можна було б порівнювати результат.

Основним аспектом який повинен враховуватись при вирівнюванні є правильне положення кожного символу відносно базової лінії текстового рядка. Проте кожен символ має свої власні параметри позицій, які також сильно залежать від особливостей написання та додаткових елементів, які можуть додаватись до символу при написанні.

Було запропоновано декілька метрик оцінювання вирівнювання тексту. Серед них – аналіз зміни відстаней між сусідніми штрихами у семплі. Порівнюючи цю метрику з іншими можливими, такими як зміна загальної висоти семпла, зміна відстаней кожного штриха до загальної базової лінії, зміна відстаней кожного штриха до центру мас семпла, було виявлено, що дані метрики не можуть ефективно визначати помилки вирівнювання та вирізняти їх серед коректного вирівнювання з допустимою точністю.

Для порівняння вхідної послідовності точок з послідовністю вирівняних точок, в експериментах використовується обчислення Евклідової відстані у двовимірному просторі. Формула для обчислення відстані між точками p та q :

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (4.1)$$

4.4 Постановка обчислювальних експериментів

Обрана метрика зміни відстаней між сусідніми штрихами дає можливість якісно проаналізувати як загальну модифікацію всього семпла після вирівнювання, так і відловлювати конкретні випадки відносно великої зміни положення штрихів, що може

вказувати на можливу похибку вирівнювання. Також в експериментах важливо дослідити вплив можливих похибок розпізнавання та сегментації на загальні результати вирівнювання.

З описаною вище метрикою було проведено 2 основні експерименти для перевірки якості вирівнювання. Кожен з них виконаний на повному тестувальному наборі даних, з фільтруванням результатів за правильністю результату розпізнавання. Неспівпадіння мінімум одного символу з розпізнаного рядка тексту до правдивого рядка тексту з набору даних вважається помилкою розпізнавання. Важливо зауважити, що помилка розпізнавання в частині випадків може вести за собою помилку або неточність сегментації, відповідно і навпаки. Проте також в багатьох випадках помилка розпізнавання є лише неспівпадінням написаного тексту до розміченого правдивого рядка, що записаний в наборі даних.

Перший експеримент оцінює максимальну зміну відносного положення сусідніх штрихів, що дає змогу оцінити глобальну модифікацію семпла при вирівнюванні. Для кожного вхідного семплу обчислюється набір відстаней між кожним сусіднім штрихом. Далі семпл проводиться через систему вирівнювання, та на виході отримуємо вирівняний семпл з модифікованими штрихами. Для нього проводимо таке саме обчислення відстаней між сусідніми штрихами для отримання другого набору відстаней. Далі обидва набори відстаней співставляються для порівняння тих самих відношень між сусідами, і обраховується вихідний набір змін відстаней по модулю. Для аналізу глобальної зміни семпла перевіряється максимальне значення з усього набору змін відстаней.

В другому експерименті використовується коренева середньоквадратична похибка (КСКВ – RMSE) по всьому обчисленому набору змін відстаней. Середньоквадратична похибка є числом, обчисленим як сума квадратів обчислених залишків, поділена на кількість спостережень, тобто розглядає суму, на яку значення, передбачені оцінювачем, відрізняються від оцінюваних значень. RMSE додатково використовує корінь над середньоквадратичною похибкою. Це значення охоплює як

дисперсію оцінювача (наскільки широким є розкид оцінок від одного зразка даних до іншого), так і його зміщення (наскільки віддаленим є усереднене оцінене значення від істинного).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\Delta Input_i - \Delta Aligned_i)^2} \quad (4.2)$$

Середньоквадратичне відхилення є мірою якості оцінювача. Оскільки вона походить від квадрата евклідової відстані, її значення є завжди додатним, і зменшується, коли похибка наближається до нуля. Ця оцінка враховує довжину семпла та показує середнє значення модифікації по всьому семплу.

4.5 Аналіз результатів вирівнювання

Результати виконання обох експериментів, опис яких наведено вище, зображені на рис. 4.2 та рис. 4.3. На діаграмах на осі абсцис зображено розподіл по кількості семплів у відсотках відносно однієї з метрик, які оцінюють виконану зміну позицій штрихів вхідного семпла в результаті вирівнювання. Значення на осі ординат визначаються пікселями. Варто відзначити, що отримані діаграми по обом експериментам мають схожість з логарифмічним нормальним розподілом, при цьому останній стовпець відображається як аномальний сплеск, що можна вважати потенційною помилкою вирівнювання. Більший сплеск спостерігається якраз на випадках розпізнавання з помилкою, що підтверджує припущення про погіршення результатів вирівнювання при отриманні помилки розпізнавання.

4.5.1 Метрика за максимальною зміною відстаней

Графік розподілу семплів за максимальною зміною відстаней між сусідніми штрихами наведено на рис. 4.2.

По аналізу максимальних відстаней отримано 0.79% семплів з успішним розпізнаванням, в яких відбулась зміна відстані між сусідніми штрихами більше ніж

30 пікселів (висота медіанного штриха). Ці приклади мають підозру або на дуже нерівне початкове написання символів, або на допущену помилку при вирівнюванні. Такі зразки повинні бути окремо перевірені експертом.

Середнє значення максимальної зміни відносного положення сусідніх штрихів становить 14.56 пікселів, що являє собою приблизно 48.5% від значення медіанного штриха в семплі.

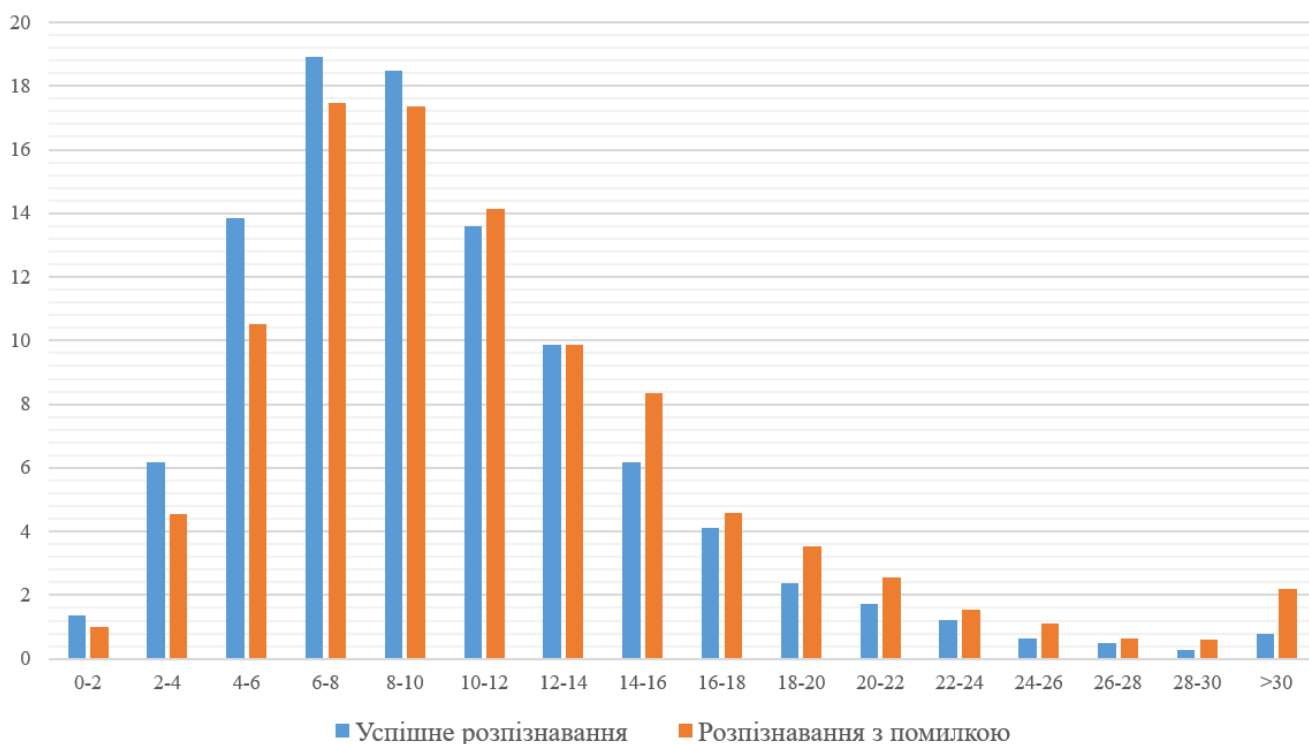


Рис. 4.2 – Розподіл семплів за максимальною зміною відстаней між сусідніми штрихами

4.5.2 Метрика за середньоквадратичною похибкою зміни відстаней

Графік розподілу семплів за середньоквадратичною похибкою (RMSE) по зміні відстаней між сусідніми штрихами наведено на рис. 4.3.

Аналіз за середньоквадратичною похибкою показав схожу тенденцію по розподілу вирівняних семплів. Майже 4.7% від всіх семплів отримали найбільше середнє зміщення більше 10 пікселів, інші семпли отримали менші зміни у структуру зв'язків між сусідніми штрихами. Така значна модифікація відносних відстаней між

сусідніми штрихами означає занадто різку зміну структури вхідної послідовності, що може свідчити про можливу помилку вирівнювання. Це також підтверджується тим, що кількість семплів з таким великим зміщенням більша якраз у випадку появи помилки при розпізнаванні. Середні значення по зсуву відстаней між сусідніми штрихами становлять 4.56 пікселів при успішному розпізнаванні, та 4.77 пікселів при розпізнавання з помилкою.

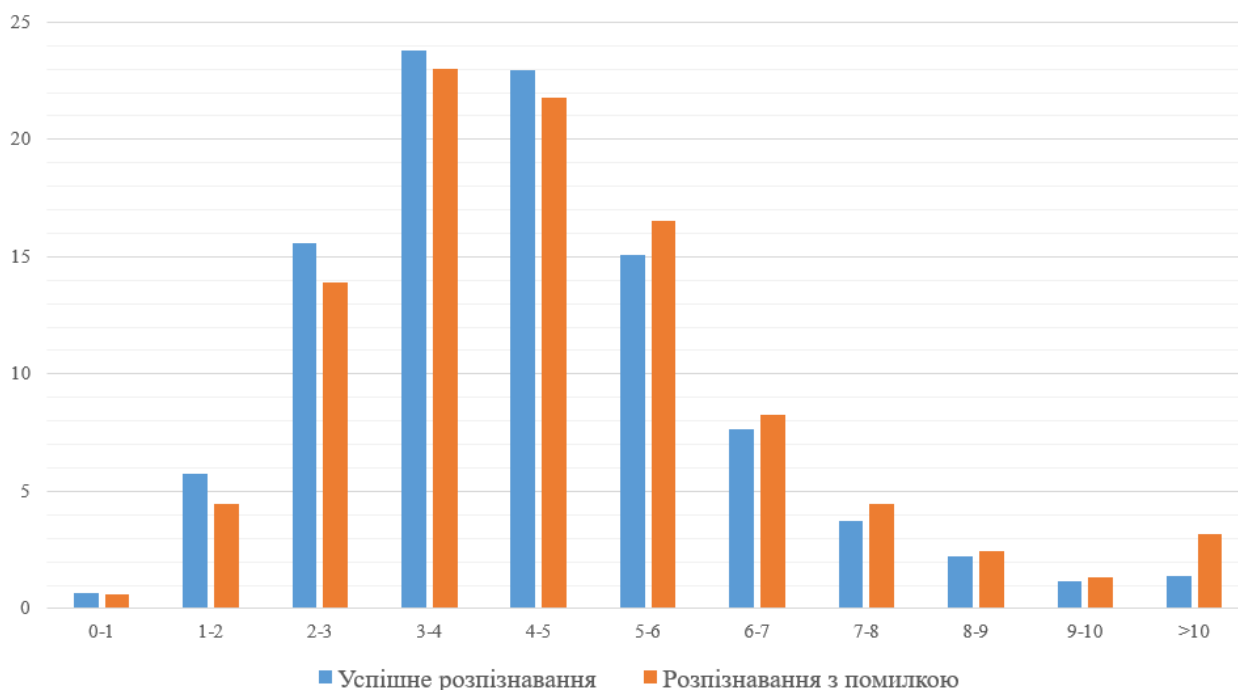


Рис. 4.3 – Розподіл семплів за середньоквадратичною похибкою (RMSE) по зміні відстаней між сусідніми штрихами

Отже, можна вважати, що розроблена система провела операцію вирівнювання вхідних даних, виконуючи зміщення штрихів в середньому на 15.4% відносно їх початкових розташувань.

4.5.3 Залежність від довжини рядка та кількості штрихів

Було також заміряно залежність середньоквадратичної похибки зміни відстаней між сусідами до довжини рядка семла та до кількості штрихів в семплі, результати зображені на рис. 4.4 та рис. 4.5.

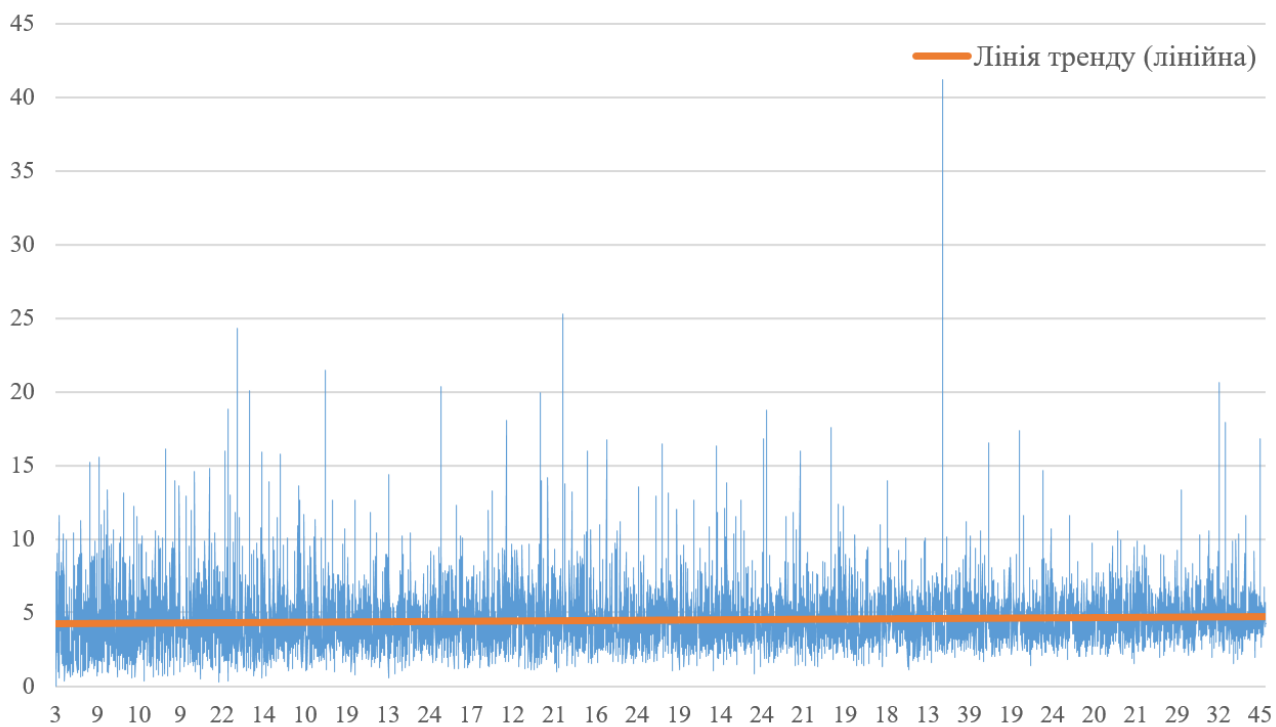


Рис. 4.4 – Залежність RMSE зміни відстаней між сусідніми штрихами відносно кількості штрихів у семплі

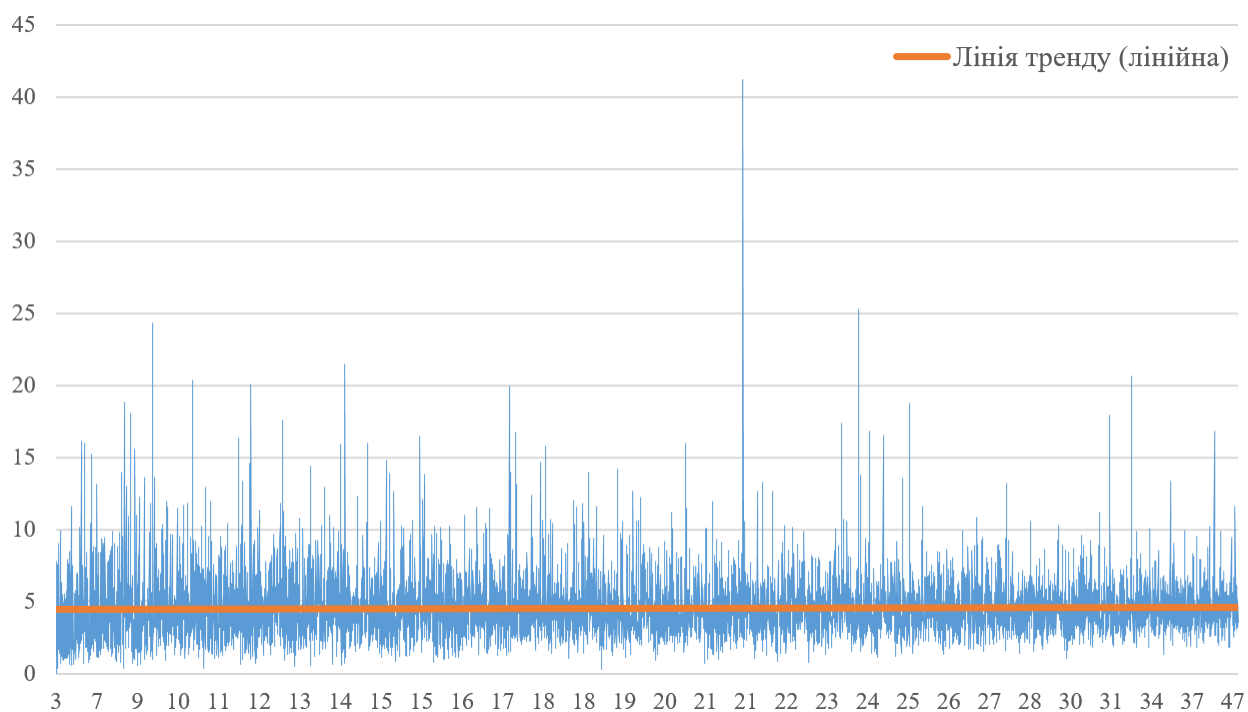


Рис. 4.5 – Залежність RMSE зміни відстаней між сусідніми штрихами відносно довжини рядка у семплі

Бачимо, що результати вирівнювання мають низьку залежність до довжини рядка або кількості штрихів у семплі, що є показником коректної роботи системи вирівнювання, оскільки потенційна помилка кроків розпізнавання, сегментації або вирівнювання не буде накопичуватися зі збільшенням семпла.

4.5.4 Аналіз прикладів роботи системи

Проведемо аналіз результатів роботи розробленої системи вирівнювання на прикладах з тестувального набору даних.

На рис. 4.6 наведено приклади успішного проведення вирівнювання. Система доводить свою здатність коректно вирівнювати тексти з різними стилями написання, включаючи як окреме написання кожного символу, так і зв'язне написання послідовностей символів.

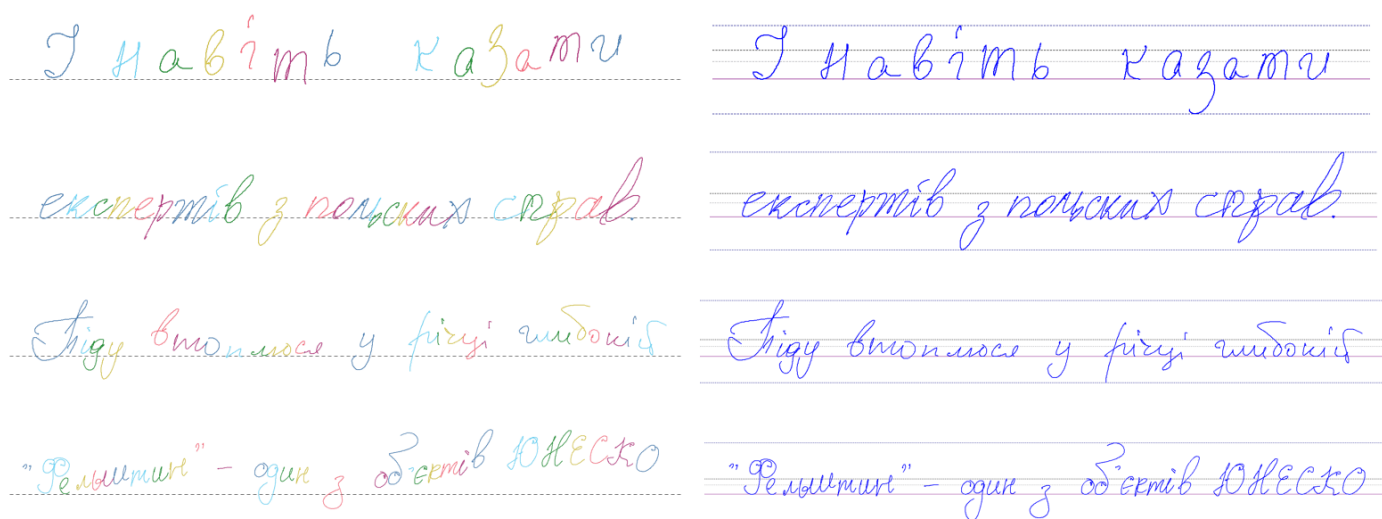


Рис. 4.6 – Приклади успішного вирівнювання

Інші приклади роботи системи вирівнювання наведено у Додатку А.

На рис. 4.7 наведено приклади некоректного вирівнювання з допущеними помилками. На першому прикладі літера 'в' вирівнюється по нижній частині символу за загальним правилом вирівнювання, проте через наявність додаткового нижнього гачка піднята трохи вище, ніж потрібно. На другому прикладі відбувалась помилка класифікації та маркування послідовності, а саме символ '9' класифіковано два рази,

що в результаті створило помилку у вирівнюванні. В третьому прикладі відбулась помилка сегментації послідовності, а саме невірна сегментація в символі “П” через написання, що схоже на іншу літеру, що також вплинуло на сегментацію декількох наступних символів. Четвертий приклад демонструє ситуацію з неочевидним правильним результатом вирівнювання через специфічні вхідні дані, що містять некоректно написаний символ ‘д’.



Рис. 4.7 – Приклади некоректного вирівнювання

Бачимо, що частина з прикладів некоректного вирівнювання пов’язана з помилками від попередніх етапів обробки, а саме розпізнавання та сегментації. Для інших помилок, що спричинені специфічним написанням деяких символів, які не підтримуються поточними версіями алгоритмів, зафіксовано задачі на майбутню модифікацію та покращення алгоритмів вирівнювання.

Важливо зазначити, що навіть при наявності помилок вирівнювання, вони залишилось локалізованими і не вплинули на вирівнювання іншої частини послідовності, яка в результаті отримала кращий вигляд, ніж початкове її написання.

4.6 Практичне застосування та впровадження в мобільний застосунок

Одним з основним цільових призначень розробленої системи є її використання в різноманітних пристроях з можливістю рукописного введення та підтримкою цифрового пера, одним з найрозповсюдженіших прикладів є смартфон. Розроблену систему було зібрано та скомпільовано як окрему бібліотеку під архітектуру ARM64. Між системою та інтерфейсом користувача створено додатковий провайдер на основі JNI (Java Native Interface), що дає можливість проводити обмін інформації між складовими частинами, написаними різними мовами, в даному разі на мовах програмування C++ та Java.

Розроблені складові було інтегровано в прототип мобільного застосунку під Android для тестування роботи вирівнювання рукописного тексту. Застосунок має простий інтерфейс з можливістю вводити рукописний текст в нижнє поле, за необхідності видаляти написаний текст, отримувати результат розпізнавання тексту, а також результат вирівнювання тексту, що відображається у верхньому полі після проходження часу неактивності (таймауту). Варто зазначити, що основа мобільного додатку розроблена всередині компанії «Самсунг РнД Інститут Україна». Приклад роботи системи інтегрованої в мобільний застосунок наведено на рис. 4.8, введений текст українською мовою успішно розпізнано та вирівняно.

Також було додатково проаналізовано здатність системи вирівнювання зменшувати загальну висоту рукописного документу. Варто зазначити, що зменшення буде відбуватись тільки на прикладах, символи в яких написані з певним розкидом по вертикалі. У випадках рівного написання тексту зменшення розміру відбуватись не буде, тому експеримент було проведено з відкиданням прикладів, де текст відразу було написано рівно (вирівнювання не дало модифікацій). В результаті виконання вирівнювання висота рядка тексту зменшується в середньому на 10.1%. Отже, в середньому на таке ж значення буде зменшуватись висота текстового документу при вирівнюванні з обраною стратегією статичного міжрядкового інтервалу зі значенням 0. Стратегія вирівнювання з мінімізацією розміру (дозволяє перетини рядків там, де

не відбувається накладання штрихів) дасть змогу зменшити висоту документу ще сильніше, як наслідок, ефективність використання простору в текстовому документі буде збільшено.

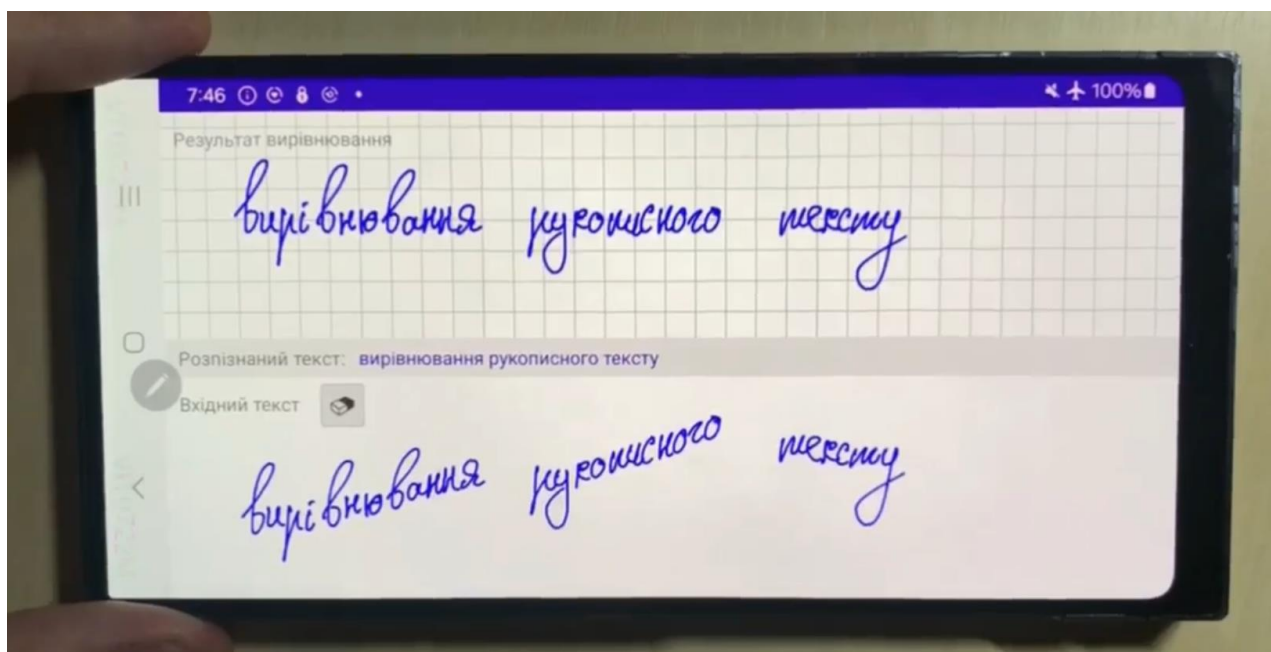


Рис. 4.8 – Приклад роботи системи інтегрованої в прототип мобільного застосунку

4.7 Оцінка продуктивності роботи системи на мобільному пристрої

Важливим критерієм оцінки розробленої системи є її спроможність ефективно працювати в середовищі з обмеженими ресурсами, в тому числі на мобільних пристроях. В цьому експерименті було оцінено продуктивність розробленого методу на прикладі пристрою, який є моделлю середнього класу та використовує процесор Mediatek Helio G80 з частотою 2 ГГц. Всі заміри проводились на CPU (центральний процесор), без використання пришвидшення обчислень за допомогою GPU (графічний прискорювач), обчислення проводились в однопоточному режимі. Було окремо заміряно час розпізнавання разом з сегментацією та час структурного аналізу. Результати наведені в табл. 4.3.

Таблиця 4.3 Оцінка продуктивності системи на мобільному пристрої

	Процесор з частотою 2.0 ГГц
Макс. час розпізнавання та сегментації	1441.25 мс
Макс. час структурного аналізу	2.015 мс
Середній час розпізнавання та сегментації	515.19 мс
Середній час структурного аналізу	0.62 мс
Середній час вирівнювання	515.81 мс

Час роботи всієї системи в середньому складає 0.52 с. У разі використання більш сучасних мобільних пристроїв з більшою потужністю процесора, швидкість виконання може збільшуватись майже в 4 рази. Також бачимо, що переважна більшість часу роботи системи припадає саме на процес розпізнавання та сегментації, а безпосередньо структурний аналіз та вирівнювання тексту займають менше 1% загального часу виконання. Це пояснюється різницею в кількостях обчислень, оскільки розпізнавання та сегментація включають в себе велику кількість процедур, в тому числі роботу двох рекурентних нейронних мереж. Проте використання цих компонентів також часто застосовується і для суміжних задач по обробці рукописного тексту, тому застосування їх в нашій системі є цілком виправданим та не є критичним з точки зору часу обчислень.

Розмір моделі розпізнавання та сегментації складає 4.7 Мб, а розмір бібліотеки системи вирівнювання у бінарному форматі, зібраної під архітектуру ARM64, становить 3.5 Мб, що в сумі задовольняє поставленим вимогам до розмірів системи.

Потенційними вдосконаленнями системи для оптимізації роботи та збільшення продуктивності є використання багатопоточності для зменшення часу обчислень, а також використання GPU для прискорення роботи наявних нейронних мереж.

4.8 Порівняння розробленої системи з наявними підходами

Для оцінки практичної цінності розробленої системи, було проведено порівняння з іншими дослідженими підходами до вирівнювання та покращення рукописного тексту. З усіх розглянутих методів в розділі 1 для порівняння було обрано найбільш схожі по поставленій задачі до вирівнювання рукописного тексту. Результати порівняння з визначеними основними відмінностями наведено в табл. 4.4.

Таблиця 4.4 Порівняння розробленої системи з наявними підходами

Підхід	Розпізнавання символів	Вирівнювання символів по базовій лінії	Елемент вирівнювання	Знаходження результуючих метрик рядка	Підтримка зв'язного написання	Збереження початкового стилю написання
Token Means [47]	Ні	Ні	Сегменти штриха	Ні	Ні	Ні
Apple Notes [142]	Так	Ні	Текстовий рядок одним цілим	Ні	Ні	Так
Ink normalization [45]	Ні	Так	Точки текстового рядка	Так	Частково	Частково
Alignment RNN [50]	Ні	Так	Окремі штрихи	Ні	Так	Так
Розроблена система	Так	Так	Штрихи символа	Так	Так	Так

Частина з існуючих підходів проводять вирівнювання тексту за допомогою розвороту всієї текстової лінії, або трансформацією розпізнаного тексту в друковане представлення за допомогою шрифтів [142]. Метод [47] виконує візуальне поліпшення рукописного введення на основі трансформації сегментів штрихів, таким чином отриманий результат не досягає поставленої в даній роботі задачі по вирівнюванню рукописного тексту. Метод [45] проводить вирівнювання рукописного тексту по базовій лінії за допомогою афінних та геометричних перетворень точок вхідних штрихів, розтягуючи їх та деформує, таким чином змінюються самі форми введених символів, втрачаючи початковий стиль написання користувача та

потенційно навіть зміст введеної інформації. Метод [50] показує достатньо хороші результати вирівнювання та задовольняє більшість поставлених задач в нашій роботі, проте не знаходить результуючі метрики вирівняного рядка. Проте основним недоліком є неможливість роботи даного методу без попереднього навчання моделі, яка потребує підготовленого набору даних зі згенерованими вирівняними наборами текстів. Варто зазначити, що підхід, запропонований в нашій роботі, може бути успішно використаний для генерації таких наборів даних, що будуть слугувати тренувальними даними для нейронних підходів.

Отже, результати порівняння показують, що розроблена система перевершує існуючі досліджені методи за більшістю поставлених вимог, даючи змогу знаходити результуючі метрики вирівняного тексту, підтримуючи вирівнювання тексту зі зв'язним написанням символів, залишаючи початковий стиль написання та почерк користувача без спотворень.

4.9 Висновки до розділу

В даному розділі було проведено основні експерименти для аналізу якості роботи розробленої системи.

Описано набір експериментальних даних та його базові характеристики. Проведено оцінку точності системи розпізнавання, система успішно розпізнала 73.04% з усіх семплів тестувального набору даних без жодної помилки, точність розпізнавання слів склала 87.39%, а символів – 97.73 %. Для порівняння результатів розпізнавання з іншими існуючими методами необхідна наявність підходящих наборів даних у відкритому доступі.

Для наступних етапів аналізу розробленої системи було задано метрики оцінювання якості вирівнювання тексту. Проведено обчислювальні експерименти, на основі яких підтверджено загальну спроможність системи успішно вирівнювати рукописний текст. Аналізуючи результати, виділено близько 13% семплів, де результат вирівнювання показав значне відхилення по зміщенню штрихів. Для

точного підтвердження або спростування помилки такі семпли потребують експертної перевірки коректності результатів. Також підтверджено пряму залежність якості вирівнювання від якості розпізнавання та сегментації, помилки яких можуть спричиняти невірну роботу алгоритмів визначення основних метрик для рукописних символів. Для семплів з правильним результатом розпізнавання символів визначено потенційну помилку вирівнювання у 0.79% семплів, базуючись на метриці максимальних відстаней між сусідніми символами. При допущеній помилці розпізнавання отримано 2.19% семплів зі значною модифікацією позицій штрихів. Отримані результати задовольняють поставлені вимоги по якості вирівнювання. Підтверджено низьку залежність результатів вирівнювання відносно довжини вхідного рядка та кількості штрихів у рядку, що потенційно дозволяє виконувати вирівнювання на ще довших рядках рукописного тексту.

На основі проведеного аналізу прикладів з некоректним вирівнюванням тексту визначено основні помилки вирівнювання. Однією з причин є похибки в розпізнаванні та сегментації вхідної послідовності рукописного тексту, що спричиняють невірну інтерпретацію та використання отриманих результатів. Іншою причиною є певна кількість нетипових випадків написання деяких символів, які не підпадають під визначені правила вирівнювання. Також частою проблемою є помилки в самому написанні користувачами деяких слів та символів з невідповідністю загальним правилам написання, тобто пропускання частини символа, зв'язному написанні частин символа які повинні бути написані окремо тощо. При цьому розроблена система успішно оброблює та вирівнює велику частину і таких прикладів з помилковим написанням, враховуючи можливі неточності написання користувачем. Також запропонований метод коректно вирівнює такі складні випадки, як рукописне введення зі зв'язним написанням символів, тексти з наявними відтермінованими штрихами, з наявними унікальними стилями написання окремих символів.

Виконано впровадження розробленої системи в мобільний застосунок, проведено оцінку продуктивності роботи системи в середовищі з обмеженими

ресурсами. Отримані результати підтвердили виконання системою всіх поставлених вимог та обмежень. Також було порівняно розроблену систему з наявними підходами до візуального поліпшення та вирівнювання тексту, результат підтвердив визначені переваги розробленої системи та її практичну цінність. В результаті розроблена система може бути інтегрована до мобільних застосунків для виконання задач вирівнювання рукописного тексту.

ВИСНОВКИ

Головним результатом дисертаційної роботи є створення нового методу для вирівнювання рукописного тексту, що враховує індивідуальні особливості написання, зв'язне написання символів та зберігає початковий стиль написання. Основна ідея базується на використанні методів розпізнавання та сегментації рукописного тексту.

У дисертаційній роботі отримано такі основні результати:

- Розроблено новий метод вирівнювання рукописного тексту, що базується на розгорнутій двонаправленій довгій короткочасній пам'яті, нейромережевій часовій класифікації для одночасного вирішення задач сегментації та розпізнавання символів, використанні структурного аналізу рукописних символів, який враховує індивідуальні особливості написання, зв'язне написання символів та зберігає початковий стиль написання. Розроблений метод дає можливість вирівнювати складний рукописний текст, що дозволяє зменшити загальний розмір документу, збільшити ефективність використання наявного простору та легкість сприйняття рукописного тексту.
- Вдосконалено метод сегментації рукописного тексту, що ґрунтується на використанні легкої рекурентної нейронної мережі та динамічного програмування, який відрізняється від існуючих застосувань додаткових алгоритмів, які дають змогу знаходити та виправляти помилки сегментації, визначати відповідні символи для відтермінованих штрихів, що дозволяє підвищити точність маркування кожного символу.
- Розроблено метод вирівнювання для тексту, написаного українською мовою. Запропоновано розбиття символів українського алфавіту на класи на базі основних ліній рядка, що визначають очікуване розташування для кожного символу. Розроблено алгоритм уточнення класів для певних символів, які мають

неоднозначне розташування в залежності від індивідуальних особливостей написання. Розроблено набір алгоритмів, що базуються на геометричному та структурному аналізі послідовностей символів для індивідуального вирівнювання рукописних літер українського алфавіту для різних стилів написання. Реалізовано пошук можливих помилок обчислення основних ліній, та згладжування позицій символів в рядку тексту. Розроблено розрахунок основних ліній вирівняного рядка для надання результуючих метрик разом з результатом вирівнювання. Запропоновано можливість вирівнювання багатьох рядків в тексті за різними стратегіями вирівнювання з врахуванням контекстної інформації про весь текстовий документ.

- Розроблено систему вирівнювання рукописного тексту для української мови та впроваджено її в тестовий мобільний застосунок. Проведено тестування та експериментально визначено ефективність системи в умовах обмежень обчислювальних ресурсів. Проведено аналіз результатів роботи системи з використанням різних метрик для визначення можливих помилок вирівнювання. Визначено, що система вирівнювання потенційно зробила помилку вирівнювання в 0.79% семплів при правильному результаті розпізнавання, що цілком задовольняє поставлені вимоги по якості роботи. Продемонстровано практичну цінність розробленої системи для можливостей побудови різноманітних мобільних додатків з інтерфейсом на основі пера.
- Представлений у цій роботі метод вирівнювання рукописного тексту впроваджений у комерційний додаток для смартфонів Samsung у вигляді повноцінної функції вирівнювання для різних мов.

Майбутніми напрямками дослідження в даній області можуть бути:

- Підвищення якості вирівнювання тексту за рахунок покращення точності розпізнавання та сегментації, що може досягатися покращенням архітектури

мереж, збільшенням кількості ваг, збільшенням розмірів та варіативності наборів даних для навчання.

- Розширення набору мов та систем письма, які підтримуються системою вирівнювання. Специфічні мови, як наприклад мова гінді, арабська мова, потребують інших алгоритмів та особливих правил по вирівнюванню рукописного тексту, які також можуть базуватись на структурному аналізі розпізнаних символів.
- Впровадження інших геометричних модифікацій штрихів для збільшення якості поліпшення зовнішнього вигляду рукописного тексту. Такими модифікаціями можуть бути поворот штрихів, розтягнення штрихів по осям координат, згладжування точок штрихів для плавності ліній тощо.

Також іншим наступним підходом у цій області може бути використання рекурентних нейронних мереж, що гарно показують себе для виконання різних задач по обробці рукописного тексту, для виконання задачі вирівнювання напрямку. Основним обмеженням такого підходу на сьогодні є необхідність мати великі набори готових вирівняних даних для тренування мережі, окремі для різних мов та систем письма. Це є проблемою за відсутності наявності інших методів до вирівнювання, які можуть бути використані для генерації тренувальних даних у вигляді пар “вхідні рукописні штрихи – вирівняні рукописні штрихи”. Відповідно, розроблена в даній дисертаційній роботі система має додаткову практичну цінність для вирішення задачі по створенню вирівняних наборів рукописних текстів відповідно до встановлених правил та обмежень.

СПИСОК ІЛЮСТРАЦІЙ

<i>Рис. 1.1 – Покрокова візуалізація покращення рукописного тексту шляхом представлення рукописних штрихів через клотоїдні сплайни [39]</i>	<i>34</i>
<i>Рис. 1.2 – Структура системи та приклад вирівнювання на основі класифікації екстремумів та геометричних деформацій [45].....</i>	<i>34</i>
<i>Рис. 1.3 – Приклади візуального покращення рукописних елементів та тексту за допомогою розбиття на токени [47].....</i>	<i>36</i>
<i>Рис. 1.4 – Приклад генерації рукописного тексту у різних заданих стилях за допомогою умовно-варіаційної рекурентної нейронної мережі [46].....</i>	<i>37</i>
<i>Рис. 1.5 – Архітектура запропонованої рекурентної нейронної мережі [50]</i>	<i>38</i>
<i>Рис. 1.6 – Приклади неуспішного вирівнювання за допомогою рекурентної нейронної мережі [50].....</i>	<i>39</i>
<i>Рис. 1.7 – Приклади складних випадків для задачі сегментації рукописного тексту: а – перетин штрихів між рядками; б – структура колонок; в – написання тексту під кутом; г – нелінійна структура рядків [79].....</i>	<i>44</i>
<i>Рис. 1.8 – Приклад вирівнювання рукописного тексту.....</i>	<i>46</i>
<i>Рис. 1.9 – Загальна структура запропонованого рішення</i>	<i>48</i>
<i>Рис. 2.1 – Схема розгорнутої рекурентної нейронної мережі.....</i>	<i>52</i>
<i>Рис. 2.2 – Схема розгорнутої двонаправленої рекурентної нейронної мережі</i>	<i>54</i>
<i>Рис. 2.3 – Структура вузла ДДКЧП з блоком пам'яті на одну комірку</i>	<i>56</i>
<i>Рис. 2.4 – Приклад роботи нейромережевої часової класифікації: а) вхідні точки рукописного введення; б) результат класифікації та маркування послідовності рекурентною нейронною мережею з НЧК.....</i>	<i>62</i>
<i>Рис. 2.5 – Архітектура рекурентної нейронної мережі для розпізнавання та сегментації символів.....</i>	<i>65</i>

<i>Рис. 2.6 – Допустимі переходи між чотирма класами в процесі коректування результатів сегментації додатковою мережею.....</i>	<i>69</i>
<i>Рис. 2.7 – Приклад розпізнавання та побудови сегментації: а) вхідна послідовність штрихів; б) попередній результат сегментації після виходу з основної мережі класифікації; в) результат сегментації з додатковою мережею для покращення сегментації; г) результат сегментації з алгоритмом приєднання відтермінованих штрихів та результат класифікації символів.....</i>	<i>72</i>
<i>Рис. 3.1 – Загальна структура розробленої системи вирівнювання</i>	<i>77</i>
<i>Рис. 3.2 – Приклад побудови спрощеного набору точок з вхідним рядком тексту “блідий місяць”.....</i>	<i>80</i>
<i>Рис. 3.3 – Модель зберігання інформації з відповідністю наборів точок до кожного символу</i>	<i>81</i>
<i>Рис. 3.4 – Приклад побудови відповідностей сегментів вхідних штрихів до розпізнаних символів</i>	<i>82</i>
<i>Рис. 3.5 – Визначення основних ліній текстового рядка (позначені синім), результуючих метрик рядка (позначені помаранчевим) та визначення класів символів.....</i>	<i>85</i>
<i>Рис. 3.6 – Приклад написання літери ‘з’ в різному стилі.....</i>	<i>86</i>
<i>Рис. 3.7 – Приклади визначення основних ліній базуючись на класі символу</i>	<i>88</i>
<i>Рис. 3.8 – Приклади визначення основних ліній на основі структурного аналізу символів.....</i>	<i>89</i>
<i>Рис. 3.9 – Приклад визначення базової лінії для специфічного написання символу ‘р’ в слові “крок” на основі сусідніх символів</i>	<i>93</i>
<i>Рис. 3.10 – Приклад роботи згладжування позиції символу ‘д’: а) вхідні штрихи; б) вирівнювання штрихів виконано по визначеній базовій лінії символів; в) позицію символу ‘д’ згладжено на основі однакової відстані від основної частини символу до ікс-лінії та до базової лінії рядка (помаранчеві стрілки).....</i>	<i>94</i>
<i>Рис. 3.11 – Приклад визначення основних ліній при об’єднанні символів в групи .</i>	<i>95</i>

<i>Рис. 3.12 – Приклад вирівнювання текстових рядків з різними стратегіями вирівнювання документа: а) вхідний текст; б) вирівнювання з мінімізацією загального розміру та уникненням перетинання штрихів ліній; в) вирівнювання зі збереженням однакового міжряджкового інтервалу; г) вирівнювання зі збереженням однакового значення інтерліньяжу (відстані між базовими лініями кожного рядка).....</i>	<i>98</i>
<i>Рис. 4.1 – Розподіл кожної літери українського алфавіту по кількості зустрічань в тестовому наборі даних.....</i>	<i>102</i>
<i>Рис. 4.2 – Розподіл семплів за максимальною зміною відстаней між сусідніми штрихами.....</i>	<i>107</i>
<i>Рис. 4.3 – Розподіл семплів за середньоквадратичною похибкою (RMSE) по зміні відстаней між сусідніми штрихами.....</i>	<i>108</i>
<i>Рис. 4.4 – Залежність RMSE зміни відстаней між сусідніми штрихами відносно кількості штрихів у семплі.....</i>	<i>109</i>
<i>Рис. 4.5 – Залежність RMSE зміни відстаней між сусідніми штрихами відносно довжини рядка у семплі.....</i>	<i>109</i>
<i>Рис. 4.6 – Приклади успішного вирівнювання.....</i>	<i>110</i>
<i>Рис. 4.7 – Приклади некоректного вирівнювання.....</i>	<i>111</i>
<i>Рис. 4.8 – Приклад роботи системи інтегрованої в прототип мобільного застосунку.....</i>	<i>113</i>

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] “How many smartphones are in the world?”. [Online]. Available: <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>. [Accessed: 04-Nov-2023].
- [2] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. (2021). Text recognition in the wild: A survey. *ACM Computing Surveys (CSUR)* 54, 2, Article 42 (mar 2021), 35 pages. <https://doi.org/10.1145/3440756>.
- [3] Viktor Zaytsev, Dmytro Zhelezniakov, Anastasiia Cherneha, and Olga Radyvonenko. (2023). A Study on the Usability of Handwriting Assistant for Smartphone’s Lock Screen. In *International Conference on Human-Computer Interaction. Design, Operation and Evaluation of Mobile Communications*, pp/ 106–123. 10.1007/978-3-031-35921-7_8
- [4] Caravolas M, Downing C, Hadden CL and Wynne C. (2020). Handwriting Legibility and Its Relationship to Spelling Ability and Age: Evidence From Monolingual and Bilingual Children. *Front. Psychol.* 11:1097. doi: 10.3389/fpsyg.2020.01097.
- [5] Islam, Noman & Islam, Zeeshan & Noor, Nazia. (2016). A Survey on Optical Character Recognition System. *ITB Journal of Information and Communication Technology*.
- [6] Indermühle, Emanuel. (2012). Analysis of Digital Ink in Electronic Documents. [Online]. Available: https://www.researchgate.net/publication/259460866_Analysis_of_Digital_Ink_in_Electronic_Documents. [Accessed: 04-Nov-2023].
- [7] Khobragade, R.N., Koli, N.A., Lanjewar, V.T. (2020). Challenges in Recognition of Online and Off-line Compound Handwritten Characters: A Review. In: Zhang, YD., Mandal, J., So-In, C., Thakur, N. (eds) *Smart Trends in Computing and*

- Communications. Smart Innovation, Systems and Technologies*, vol 165. Springer, Singapore. https://doi.org/10.1007/978-981-15-0077-0_38.
- [8] “GVR Report cover Digital Pen Market Size, Share & Trends Report Digital Pen Market Size, Share & Trends Analysis Report By Product (Scanning, Handwriting), By Usage (PC, Tablet, Smartphone), By Application, By Region, And Segment Forecasts, 2022 - 2030”. [Online]. Available: <https://www.grandviewresearch.com/industry-analysis/digital-pen-market>. [Accessed: 04-Nov-2023].
- [9] Duran, Erol & Karataş, Arda. (2019). Elimination of Writing Difficulty in Primary School: An Action Research. *International Journal of Progressive Education*. 15. 288-299. 10.29329/ijpe.2019.212.19.
- [10] C. Marlin “Lin” Brown. (1988). Comparison of typing and handwriting in “two-finger typists”. In *Proceedings of the Human Factors Society Annual Meeting*, Vol. 32. SAGE Publications Sage CA: Los Angeles, CA, 381–385.
- [11] Aishwarya Agarwal, Anuj Srivastava, Inderjeet Nair, Swasti Shreya Mishra, Vineeth Dorna, Sharmila Reddy Nangi, and Balaji Vasan Srinivasan. (2023). Sketch-Buddy: Context-Aware Sketch Enrichment and Enhancement. In *Proceedings of the 14th Conference on ACM Multimedia Systems (Vancouver, BC, Canada) (MM-Sys '23)*. Association for Computing Machinery, New York, NY, USA, 217–228. <https://doi.org/10.1145/3587819.3590980>.
- [12] Ronald T Kellogg. (2008). Training writing skills: A cognitive developmental perspective. *Journal of writing research* 1, 1 (2008), 1–26. <https://doi.org/10.17239/jowr-2008.01.01.1>.
- [13] Fitrianiingsih & Madenda, Sarifuddin & Ernastuti, Ernastuti & Widodo, Rini & Rodiah, Rodiah. (2017). Cursive Handwriting Segmentation using Ideal Distance Approach. *International Journal of Electrical and Computer Engineering (IJECE)*. 7. 2863. 10.11591/ijece.v7i5.pp2863-2872.

- [14] Steve Graham, Naomi Weintraub, and Virginia W Berninger. (1998). The relationship between handwriting style and speed and legibility. *The journal of educational research* 91, 5, 290–297. <https://doi.org/10.1080/00220679809597556>.
- [15] Mustafid, Ahmad & Younas, Junaid & Lukowicz, Paul & Ahmed, Sheraz. (2023). IAMonSense: multi-level handwriting classification using spatiotemporal information. *International Journal on Document Analysis and Recognition (IJDAR)*. 26. 1-17. [10.1007/s10032-023-00433-y](https://doi.org/10.1007/s10032-023-00433-y).
- [16] Kuznetsov, Yuri & Sdobnov, Anton & Meglinski, Igor & Harmelin, Alon & Kalchenko, Vyacheslav. (2019). Evaluation of handwriting peculiarities utilizing laser speckle contrast imaging. *Laser Physics Letters*. 16. [10.1088/1612-202X/ab43d7](https://doi.org/10.1088/1612-202X/ab43d7).
- [17] Tolga Erdogan, Ozge Erdogan. (2012). An Analysis of the Legibility of Cursive Handwriting of Prospective Primary School Teachers. *Social and Behavioral Sciences*, V.46, 2012, 5214-5218. <https://doi.org/10.1016/j.sbspro.2012.06.412>.
- [18] Nurhan Aktas. (2023). Does Primary Students’ Writing Ergonomics Affect Their Handwriting Legibility? *Language Teaching and Educational Research* 6, 1 (2023), 24–38. <https://doi.org/10.35207/late.1286665>.
- [19] Ajoy Mondal and CV Jawahar. (2023). ICDAR 2023 Competition on Indic Handwriting Text Recognition. In *Document Analysis and Recognition - ICDAR 2023*. Springer, Springer Nature Switzerland, Cham, 435–453. https://doi.org/10.1007/978-3-031-41679-8_25.
- [20] Harjeet Singh, Rajendra Kumar Sharma, and VP Singh. (2021). Online handwriting recognition systems for Indic and non-Indic scripts: a review. *Artificial Intelligence Review* 54 (2021), 1525–1579. <https://doi.org/10.1007/s10462-020-09886-7>.
- [21] Rabiaa Zitouni, Hala Bezine, and Najet Arous. (2023). Online handwritten scripts classification using fuzzy attributed relational graphs. *International Journal of Machine Learning and Cybernetics* (2023), 1–18. <https://doi.org/10.1007/s13042-023-01772-2>.

- [22] Zelin Chen, Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. (2021). Level online writer identification. *International Journal of Computer Vision* 129, 5 (2021), 1394–1409. <https://doi.org/10.1007/s11263-020-01414-y>.
- [23] Yan-Rong Wang, Da-Han Wang, Xiao-Long Yun, Yan-Ming Zhang, Fei Yin, and Shunzhi Zhu. (2023). A Shallow Graph Neural Network with Innovative Node Updating for Online Handwritten Stroke Classification. In *Document Analysis and Recognition - ICDAR 2023*. Springer, Springer Nature Switzerland, Cham, 3–19. https://doi.org/10.1007/978-3-031-41685-9_1.
- [24] Yejing Xie, Harold Mouchère, Foteini Simistira Liwicki, Sumit Rakesh, Rajkumar Saini, Masaki Nakagawa, Cuong Tuan Nguyen, and Thanh-Nghia Truong. (2023). ICDAR 2023 CROHME: Competition on Recognition of Handwritten Mathematical Expressions. In *Document Analysis and Recognition - ICDAR 2023*. Springer, Springer Nature Switzerland, Cham, 553–565. https://doi.org/10.1007/978-3-031-41679-8_33.
- [25] Dmytro Zhelezniakov, Viktor Zaytsev, and Olga Radyvonenko. (2021). Online Handwritten Mathematical Expression Recognition and Applications: A Survey. *IEEE Access* 9 (2021), 38352–38373. <https://doi.org/10.1109/ACCESS.2021.3063413>.
- [26] Anh Duc Le and Masaki Nakagawa. (2016). A system for recognizing online handwritten mathematical expressions by using improved structural analysis. *Int. J. on Doc. Anal. and Recog.* 19, 4 (2016), 305–319.
- [27] Moises Diaz, Miguel A Ferrer, Donato Impedovo, Muhammad Imran Malik, Giuseppe Pirlo, and Réjean Plamondon. (2019). A perspective analysis of handwritten signature technology. *Acm Computing Surveys (Csur)* 51, 6 (2019), 1–39. <https://doi.org/10.1145/3274658>.
- [28] Kar, Biswajit & Mukherjee, Anirban & Dutta, Pranab. (2017). Stroke Point Warping-Based Reference Selection and Verification of Online Signature. *IEEE*

- Transactions on Instrumentation and Measurement*. PP. 1-10.
10.1109/TIM.2017.2755898.
- [29] Moises Diaz, Momina Moetesum, Imran Siddiqi, and Gennaro Vessio. (2021). Sequence-based dynamic handwriting analysis for Parkinson's disease detection with one-dimensional convolutions and BiGRUs. *Expert Systems with Applications* 168 (2021), 114405. <https://doi.org/10.1016/j.eswa.2020.114405>.
 - [30] D'Alessandro, Tiziana & De Stefano, Claudio & Fontanella, Francesco & Nardone, Emanuele & Scotto di Freca, Alessandra. (2023). Feature Evaluation in Handwriting Analysis for Alzheimer's Disease Using Bayesian Network. *21st International Conference of the International Graphonomics Society, IGS 2023*. 10.1007/978-3-031-45461-5_9.
 - [31] David Ha and Douglas Eck. (2017). A Neural Representation of Sketch Drawings. ArXiv e-prints (apr 2017). arXiv:1704.03477 [cs.NE].
 - [32] Peng Xu, Timothy M. Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. (2023). Deep Learning for Free-Hand Sketch: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2023), 285–312. <https://doi.org/10.1109/TPAMI.2022.3148853>.
 - [33] Hse, Heloise & Newton, A., (2005). Recognition and beautification of multi-stroke symbols in digital ink. *Computers & Graphics*. 29. 533-546. 10.1016/j.cag.2005.05.006.
 - [34] Theo Pavlidis and Christopher J. Van Wyk. (1985). An Automatic Beautifier for Drawings and Illustrations. *SIGGRAPH Comput. Graph.* 19, 3 (jul 1985), 225–234. <https://doi.org/10.1145/325165.325240>.
 - [35] Wacef Guerfali and Réjean Plamondon. (1993). Normalizing and restoring on-line handwriting. *Pattern Recognition* 26, 3, 419–431.
 - [36] James Arvo and Kevin Novins. (2000). Fluid Sketches: Continuous Recognition and Morphing of Simple Hand-Drawn Shapes. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (San Diego, California,*

- USA) (UIST '00). Association for Computing Machinery, New York, NY, USA, 73–80. <https://doi.org/10.1145/354401.354413>.
- [37] Aida-zade, Kamil & Hasanov, Jamaladdin. (2009). Word base line detection in handwritten text recognition systems. *International Journal of Intelligent Systems and Technologies* 4:1 2009.
- [38] Yannick Thiel, Karan Singh, and Ravin Balakrishnan. (2011). Elasticurves: Exploiting Stroke Dynamics and Inertia for the Real-Time Neatening of Sketched 2D Curves (UIST '11). Association for Computing Machinery, New York, NY, USA, 383–392. <https://doi.org/10.1145/2047196.2047246>.
- [39] Ilya Baran, Jaakko Lehtinen, and Jovan Popović. (2010). Sketching clothoid splines using shortest paths. *Computer Graphics Forum* 29, 2 (2010), 655–664. <https://doi.org/10.1111/j.1467-8659.2009.01635.x>.
- [40] Vázquez-Méndez, Miguel & Casal Urcera, Gerardo. (2016). The Clothoid Computation: A Simple and Efficient Numerical Algorithm. *Journal of Surveying Engineering*. 142. 04016005:1-9. [http://dx.doi.org/10.1061/\(ASCE\)SU.1943-5428.0000177](http://dx.doi.org/10.1061/(ASCE)SU.1943-5428.0000177).
- [41] Jingwan Lu, Fisher Yu, Adam Finkelstein, and Stephen DiVerdi. (2012). Helping-Hand: Example-Based Stroke Stylization. *ACM Trans. Graph.* 31, 4, Article 46 (jul 2012), 10 pages. <https://doi.org/10.1145/2185520.2185542>.
- [42] Abdelhak Boukharouba. 2017. A new algorithm for skew correction and baseline detection based on the randomized Hough Transform. *Journal of King Saud university-computer and information sciences* 29, 1 (2017), 29–38. <https://doi.org/10.1016/j.jksuci.2016.02.002>.
- [43] Chandan Singh, Nitin Bhatia, and Amandeep Kaur. (2008). Hough transform based fast skew detection and accurate skew correction methods. *Pattern Recognition* 41, 12 (2008), 3528–3546. <https://doi.org/10.1016/j.patcog.2008.06.002>.
- [44] Jian xiong Dong, P. Dominique, A. Krzyyzak, and C.Y. Suen. (2005). Cursive word skew/slant corrections based on Radon transform. In *Eighth International*

- Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 478–483.
<https://doi.org/10.1109/ICDAR.2005.82>.
- [45] Patrice Y Simard, David Steinkraus, and Maneesh Agrawala. (2005). Ink normalization and beautification. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 1182–1187.
<https://doi.org/10.1109/ICDAR.2005.143>.
- [46] Emre Aksan, Fabrizio Pece, and Otmar Hilliges. (2018). DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (Montreal QC, Canada) (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14.
<https://doi.org/10.1145/3173574.3173779>.
- [47] C. Lawrence Zitnick. (2013). Handwriting Beautification Using Token Means. *ACM Trans. Graph.* 32, 4, Article 53 (jul. 2013), 8 pages.
<https://doi.org/10.1145/2461912.2461985>.
- [48] Shusen Tang and Zhouhui Lian. (2021). Write Like You: Synthesizing Your Cursive Online Chinese Handwriting via Metric-based Meta Learning. *Computer Graphics Forum* 40, 2 (2021), 141–151. <https://doi.org/10.1111/cgf.142621>.
- [49] Dohyeon Kim, Evgenii Bahtyn, Ivan Deriuga, Olga Radyvonenko, and Vadym Osadchiy. (2021). Method and electronic device for correcting handwriting input. US Patent 2021/0134029 A1. <https://patents.google.com/patent/US20210134029A1>.
- [50] K. Korovai, D. Zhelezniakov, O. Radyvonenko, O. Yakovchuk, I. Deriuga, N. Sakhnenko. (2023). Recognition-Independent Handwritten Text Alignment Using Lightweight Recurrent Neural Network. In *SIGGRAPH Asia 2023 Posters (SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 30, 1–2. DOI: <https://doi.org/10.1145/3610542.3626136>.
- [51] V. Yaremenko, W. Rogoza, V. Spitkovskyi. (2021). Application of neural network algorithms and naive bayes for text classification. *Journal of Theoretical and Applied Information Technology* 99 (1), Pages 125-134.

- [52] Chungkwong Chan. (2020). Stroke extraction for offline handwritten mathematical expression recognition. *IEEE Access* 8 (2020), 61565–61575.
- [53] Tappert, Charles & Suen, Ching & Wakahara, Toru. (1990). State of the art in on-line handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 12. 787 - 808. <http://dx.doi.org/10.1109/34.57669>.
- [54] Plamondon, Réjean. (2000). On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans Pattern Anal Mach Intell (T-PAMI). IEEE Trans. Pattern Anal. Mach. Intell.*. 22. 63-84. <http://dx.doi.org/10.1109/34.824821>.
- [55] D. Keysers, T. Deselaers, H. A. Rowley, L. -L. Wang and V. Carbune. (2017). Multi-Language Online Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1180-1194, 1. 10.1109/TPAMI.2016.2572693.
- [56] J. H. Kim and B. Sin. (2014). Online handwriting recognition. In *Proc. Handbook Doc. Image Process. Recognit.*, pp. 887–915.
- [57] Yaeger, Larry & Webb, Brandyn & Lyon, Richard. (1998). Combining Neural Networks and Context-Driven Search for Online, Printed Handwriting Recognition in the *NEWTON. AI Magazine*. 19. 73-90. http://dx.doi.org/10.1007/3-540-49430-8_14.
- [58] Pittman, James. (2007). Handwriting Recognition: Tablet PC Text Input. *Computer*. 40. 49 - 54. 10.1109/MC.2007.314. <http://dx.doi.org/10.1109/MC.2007.314>.
- [59] S. J. Young, N. H. Russell, and J. H. S. Thornton. (1989). Token passing: A simple conceptual model for connected speech recognition systems. Tech. Rep. CUED/F-INFENG/TR38, Cambridge University Engineering Department, 1989.
- [60] Lei Hu and Richard Zanibbi. (2011). HMM-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature. In *Proc. IEEE Int. Conf. on Doc. Anal. and Recog.* 457–462.

- [61] Andreas Kosmala and Gerhard Rigoll. (1998). On-line handwritten formula recognition using statistical methods. In *Proc. IEEE Int. Conf. on Pattern Recog.*, Vol. 2. 1306–1308.
- [62] H-J Winkler. (1996). HMM-based handwritten symbol recognition using on-line and off-line features. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process*, Vol. 6. 3438–3441.
- [63] Jaeger, Stefan & Manke, Stefan & Reichert, Juergen & Waibel, Alex. (2001). Online handwriting recognition: the NPen++ recognizer. *Int J Doc Anal Recognit* 3:169-180. IJDAR. 3. 169-180. <http://dx.doi.org/10.1007/PL00013559>.
- [64] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. Int. Conf. on Machine Learning*. 282–289.
- [65] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. JMLR.org, 2342–2350.
- [66] J. Schenk and G. Rigoll. (2006). Novel hybrid NN/HMM modelling techniques for on-line handwriting recognition. In *Proc. Int. Workshop Frontiers Handwriting Recognit.*, 2006, pp. 619–623.
- [67] Graves, Alex & Fernández, Santiago & Gomez, Faustino & Schmidhuber, Jürgen. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks. *ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning*. 2006. 369-376. 10.1145/1143844.1143891.
- [68] Graves, Alex & Liwicki, Marcus & Fernández, Santiago & Bertolami, Roman & Bunke, Horst & Schmidhuber, Jürgen. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE transactions on pattern analysis and machine intelligence*. 31. 855-68. <http://dx.doi.org/10.1109/TPAMI.2008.137>.

- [69] М.І. Шлезінгер, К.В. АНТОНЮК. (2011). Diffusion algorithms and structural recognition optimization problems. *Cybernetics and Systems Analysis, K., Наук.думка*, 2011.- 48, № 2.- с. 3-12.
- [70] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.* 4774–4778.
- [71] Hung Tuan Nguyen, Cuong Tuan Nguyen, and Masaki Nakagawa. (2018). ICFHR 2018– Competition on Vietnamese Online Handwritten Text Recognition using HANDS-VNOnDB (VOHTR2018). In *Proc. IEEE Int. Conf. on Frontiers in Handwr. Recog.* 494–499.
- [72] Simayi, Wujiahemaiti & Ibrayim, Mayire & Hamdulla, Askar. (2021). Study the preprocessing effect on RNN based online Uyghur handwritten word recognition. *Wireless Networks*. <https://link.springer.com/article/10.1007/s11276-021-02651-w>.
- [73] Xie, Canyu & Lai, Songxuan & Liao, Qianying & Jin, Lianwen. (2020). High Performance Offline Handwritten Chinese Text Recognition with a New Data Preprocessing and Augmentation Pipeline. *Lecture Notes in Computer Science*. LNIP, volume 12116. http://dx.doi.org/10.1007/978-3-030-57058-3_4.
- [74] Benoît Frénay and Michel Verleysen. (2013). Classification in the presence of label noise: a survey. *IEEE Transactions on neural networks and learning systems* 25, 5 (2013), 845–869.
- [75] Jaeger, Stefan & Liu, Cheng-Lin & Nakagawa, Masaki. (2003). The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition. *International Journal on Document Analysis and Recognition*. 6. 75-88. 10.1007/s10032-003-0107-y.
- [76] Delaye, Adrien & Anquetil, Eric. (2012). HBF49 feature set: A first unified baseline for online symbol recognition. *Pattern Recognition*. 46. -. 10.1016/j.patcog.2012.07.015.

- [77] Valentyna Volkova, Ivan Deriuga, Vadym Osadchyi, and Olga Radyvonenko. (2018). Improvement of Character Segmentation Using Recurrent Neural Networks and Dynamic Programming. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. 218–222.
<https://doi.org/10.1109/DSMP.2018.8478457>.
- [78] Sreedevi, I., Panda, J., Omayio, E. (2022). Word Segmentation by Component Tracing and Association (CTA) Technique. *Journal of Engineering Research*.
<https://doi.org/10.36909/jer.15207>.
- [79] Yakovchuk, O., Rogoza, W. (2024). An overview of statistical and neural-based line segmentation methods for offline handwriting recognition task. *Technology Audit and Production Reserves*, 1 (2 (75)), 14–19. DOI: <https://doi.org/10.15587/2706-5448.2024.298405>.
- [80] Kanahori, Toshihiro & Suzuki, M.. (2003). Detection of matrices and segmentation of matrix elements in scanned images of scientific documents. *International Conference on Document Analysis and Recognition*. 2003. 433 - 437 vol.1.
[10.1109/ICDAR.2003.1227704](https://doi.org/10.1109/ICDAR.2003.1227704).
- [81] Tapia, Ernesto & Rojas, Raul. (2003). Recognition of On-line Handwritten Mathematical Expressions Using a Minimum Spanning Tree Construction and Symbol Dominance. *Lecture Notes in Computer Science*. LNCS, volume 3088. 329-340. [10.1007/978-3-540-25977-0_30](https://doi.org/10.1007/978-3-540-25977-0_30).
- [82] Renton, G., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T. (2017). Handwritten text line segmentation using fully convolutional network. *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. pp. 5–9.
<https://doi.org/10.1109/ICDAR.2017.321>.
- [83] O. Yakovchuk, A. Cherneha, D. Zhelezniakov and V. Zaytsev. (2020). Methods for Lines and Matrices Segmentation in RNN-based Online Handwriting Mathematical Expression Recognition Systems. *2020 IEEE Third International Conference on*

- Data Stream Mining & Processing (DSMP)*, Lviv, Ukraine. 255-261. DOI: <https://doi.org/10.1109/DSMP47368.2020.9204273>.
- [84] Zhang, Jianshu & Du, Jun & Dai, Lirong. (2017). A GRU-based Encoder-Decoder Approach with Attention for Online Handwritten Mathematical Expression Recognition. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 902–907. IEEE (2017).
 - [85] R.G. Casey and E. Lecolinet. (1996). A Survey of Method and Strategies in Character Segmentation. *IEEE Trans. on PAMI*, vol. 18 (7), pp. 690- 706, 1996.
 - [86] S. P. Naeini, M. Khademi and A. Nikookar. (2012). A novel approach to segmentation of Persian cursive script using decision tree. *International Journal of Computer Theory and Engineering*, vol. 4 (3), p. 465, 2012.
 - [87] F. Naohiro, J. Tokuno and H. Ikeda. (2006). Online character segmentation method for unconstrained handwriting strings using off-stroke features. *Tenth International Workshop on Frontiers in Handwriting Recognition, IWFHR-10*, pp. 361-366, 2006.
 - [88] C. T. Nguyen and M. Nakagawa. (2015). An improved segmentation of online English handwritten text using recurrent neural networks. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, pp. 176-180, 2015.
 - [89] I. Mayire, H. Askar and T. Dilmurat. (2013). A Dynamic Programming Method for Segmentation of Online Cursive Uyghur Handwritten Words into Basic Recognizable Units. *Journal of Software*, vol. 10(8), pp. 2535- 2540, 2013.
 - [90] R. Ghosh. (2013). Stroke segmentation of online handwritten word using the busy zone concept. *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, pp. 54-59, 2013.
 - [91] N. Bhattacharya and U. Pal. (2012). Stroke segmentation and recognition from Bangla online handwritten text. *2012 International Conference on Frontiers in Handwriting Recognition*, pp. 740-745, 2012.
 - [92] Ibrayim, Mayire & Hamdulla, Askar & Tursun, Dilmurat. (2013). “A dynamic programming method for segmentation of online cursive Uyghur handwritten words

- into basic recognizable units”. *Journal of Software*, vol. 8 (10), pp. 2535-2540, 10.4304/jsw.8.10.2535-2540.
- [93] Alex Graves. (2012). Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*. 61–93.
- [94] Sepp Hochreiter and Jürgen Schmidhuber. (1997). Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [95] Jürgen Schmidhuber (2015). Deep learning in neural networks: An overview. *Neural Networks* 61 (2015): 85-117.
- [96] David B Fogel. (2006). Evolutionary computation: toward a new philosophy of machine intelligence. *IEEE Computing and Processing*. Vol. 1. John Wiley & Sons.
- [97] Felix A Gers and E Schmidhuber. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks* 12, 6 (2001), 1333–1340.
- [98] Felix Gers, Nicholas Schraudolph, and Jürgen Schmidhuber (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research* 3:115–143.
- [99] Mike Schuster and Kuldip K Paliwal. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.
- [100] Gillick, Dan & Brunk, Cliff & Vinyals, Oriol & Subramanya, Amarnag. (2015). Multilingual Language Processing From Bytes. *North American Chapter of the Association for Computational Linguistics*.
- [101] Sutskever, Ilya & Vinyals, Oriol & Le, Quoc. (2014). Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*. 4.
- [102] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.

- [103] Alex Graves and Jürgen Schmidhuber. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.
- [104] Rudregowda, Shashidhar & Patilkulkarni, Sudarshan & Murthy, Nishanth. (2021). Visual Speech Recognition using VGG16 Convolutional Neural Network. *Multimedia Tools and Applications*. Volume 80, pages 28941–28952, (2021), 10.21203/rs.3.rs-177220/v1.
- [105] Marcus Liwicki, Alex Graves, Santiago Fernández, Horst Bunke, and Jürgen Schmidhuber. (2007). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proc. IEEE Int. Conf. on Doc. Anal. and Recog.*
- [106] Bo Fan, Lijuan Wang, Frank K. Soong, and Lei Xie (2015). Photo-Real Talking Head with Deep Bidirectional LSTM. In *Proceedings of ICASSP 2015*.
- [107] Jürgen Schmidhuber, F Gers, and Douglas Eck. (2002). Learning nonregular languages: A comparison of simple recurrent networks and LSTM. *Neural computation* 14, 9 (2002), 2039– 2041.
- [108] Jürgen Schmidhuber, Daan Wierstra, and Faustino J Gomez. (2005). Evolino: Hybrid neuroevolution/optimal linear search for sequence prediction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*.
- [109] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atila Baskurt. (2011). Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*. Springer, 29–39.
- [110] Yong Du, Wei Wang, and Liang Wang. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1110–1118.
- [111] Liang Li, Xinge Zhu, Yiming Hao, Shuhui Wang, Xingyu Gao, and Qingming Huang. (2019). A hierarchical CNN-RNN approach for visual emotion

- classification. *ACM Transactions on Multimedia Computing, Communications, and Applications* (TOMM) 15, 3s, Article 97, 17 pages. <https://doi.org/10.1145/3359753>.
- [112] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation* 12, 10 (2000), 2451–2471.
- [113] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. (2002). Learning precise timing with LSTM recurrent networks. *Journal of machine learning research* 3, Aug (2002), 115–143.
- [114] Felix A Gers and Jürgen Schmidhuber. (2000). Recurrent nets that time and count. In Proceedings of the IEEE-INNS-ENNS *International Joint Conference on Neural Networks*. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Vol. 3. IEEE, 189–194.
<https://doi.org/10.1109/IJCNN.2000.861302>.
- [115] Wang, Xin & Liu, Yuanchao & Sun, Chengjie & Wang, Baoxun & Wang, Xiaolong. (2015). Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. Proceedings of the *7th International Joint Conference on Natural Language Processing* (Volume 1: Long Papers). 1343-1353.
10.3115/v1/P15-1130.
- [116] Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. (2001). "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies". In *Kremer and, S. C.; Kolen, J. F.. A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press.
- [117] Bourlard, Herve & Morgan, Nelson. (1994). Connectionist Speech Recognition: A Hybrid Approach. *Kluwer Academic Publishers*. 10.1007/978-1-4615-3210-1.
- [118] Julian Salazar, Katrin Kirchhoff, and Zhiheng Huang. (2019). Self-attention networks for connectionist temporal classification in speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP). IEEE, 7115– 7119.

- [119] Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner. (2019). Evaluating sequence-to-sequence models for handwritten text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1286–1293.
- [120] Viacheslav Khomenko, Andriy Volkoviy, Illya Degtyarenko, and Olga Radyvonenko. (2017). Handwriting Text/Non-Text Classification on Mobile Device. In *Proc. Int. Conf. on Art. Intel. and Pattern Recog.* 42.
- [121] Illya Degtyarenko, Ivan Deriuga, Andrii Grygoriev, Serhii Polotskyi, Volodymyr Melnyk, Dmytro Zakharchuk, and Olga Radyvonenko. (2021). Hierarchical Recurrent Neural Network for Handwritten Strokes Classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2865–2869. <https://doi.org/10.1109/ICASSP39728.2021.9413412>.
- [122] Andrii Grygoriev, Illya Degtyarenko, Ivan Deriuga, Serhii Polotskyi, Volodymyr Melnyk, Dmytro Zakharchuk, and Olga Radyvonenko. (2021). HCRNN: a novel architecture for fast online handwritten stroke classification. In *Document Analysis and Recognition – ICDAR 2021*. Springer, Springer International Publishing, Cham, 193–208. https://doi.org/10.1007/978-3-030-86331-9_13.
- [123] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. (2019). Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 82–92.
- [124] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis De-Coste, Wei Di, and Yizhou Yu. (2015). HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition. In *Proceedings of the IEEE international conference on computer vision*. 2740–2748.
- [125] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for*

- computational linguistics: human language technologies*. Association for Computational Linguistics, San Diego, California, 1480–1489.
<https://doi.org/10.18653/v1/N16-1174>.
- [126] Mioulet, Luc & Garain, Utpal & Chatelain, Clement & Barlas, Philippine & Paquet, Thierry. (2015). Language identification from handwritten documents. 676-680. 10.1109/ICDAR.2015.7333847.
- [127] Esma F Bilgin Tasdemir and Berrin Yanikoglu. (2019). A comparative study of delayed stroke handling approaches in online handwriting. *International Journal on Document Analysis and Recognition (IJDAR)* 22 (2019), 15–28.
<https://doi.org/10.1007/s10032-018-0313-2>.
- [128] Le Duc, Anh & Nguyen, Dai Hai & Nakagawa, Masaki. (2016). Modified X-Y Cut for Re-Ordering Strokes of Online Handwritten Mathematical Expressions. 2016 12th IAPR *Workshop on Document Analysis Systems (DAS)* 233-238. 10.1109/DAS.2016.19.
- [129] Bera, Suman & Kar, Radib & Saha, Souvik & Chakrabarty, Akash & Lahiri, Sagnik & Malakar, Samir & Sarkar, Ram. (2018). A One-Pass Approach for Slope and Slant Estimation of Tri-Script Handwritten Words. *Journal of Intelligent Systems*. preprint. 10.1515/jisys-2018-0105.
- [130] Ghosh, Rajib & Mandal, Gouranga. (2012). Skew Detection and Correction of Online Bangla Handwritten Word. *International Journal of Computer Science Issues*. 9.
- [131] John Freeman. (1975). The modelling of spatial relations. *Computer graphics and image processing* 4, 2 (1975), 156–171.
- [132] Bing Quan Huang, YB Zhang, and Mohand Tahar Kechadi. (2007). Preprocessing techniques for online handwriting recognition. In *Proc. IEEE Int. Conf. on Intel. Systems Design and Appl.* 793– 800.
- [133] Wacef Guerfali and Réjean Plamondon. (1993). Normalizing and restoring on-line handwriting. *Pattern Recognition*. 26, 3 (1993), 419–431.

- [134] Volkmar Frinken and Seiichi Uchida. (2015). Deep BLSTM neural networks for unconstrained continuous handwritten text recognition. In *2015 13th international conference on document analysis and recognition (ICDAR)*. IEEE, 911–915. <https://doi.org/10.1109/ICDAR.2015.7333894>.
- [135] Matthew D Zeiler. (2012). Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012).
- [136] Asadi, B., & Jiang, H. (2020). On Approximation Capabilities of ReLU Activation and Softmax Output Layer in Neural Networks. ArXiv, abs/2002.04060.
- [137] F. Chunsheng. (2009). From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM). *Final project report for ECE742 Stochastic Decision*. 2009.
- [138] Maheswari Visvalingam and James D Whyatt. (1993). Line generalisation by repeated elimination of points. *The cartographic journal* 30, 1 (1993), 46–51.
- [139] Li, Lelin & Jiang, Wanshou. (2010). An improved Douglas-Peucker algorithm for fast curve approximation. *Proceedings. 2010 3rd International Congress on Image and Signal Processing, CISP 2010*. 4. 10.1109/CISP.2010.5647972.
- [140] Singh, Sukhdeep & Chauhan, Vinod & Barney Smith, Elisa. (2020). A self controlled RDP approach for feature extraction in online handwriting recognition using deep learning. *Applied Intelligence*. 50. 10.1007/s10489-020-01632-4.
- [141] Zhen-Long Bai and Qiang Huo. (2005). A study on the use of 8-directional features for online handwritten Chinese character recognition. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 262–266.
- [142] “Draw or write in Notes on iPad”. [Online]. Available: <https://support.apple.com/en-gu/guide/ipad/ipada87a6078/ipados>. [Accessed: 01-Feb-2024].

**Додаток А. ПРИКЛАДИ РОБОТИ СИСТЕМИ
ВИРІВНЮВАННЯ РУКОПИСНОГО ТЕКСТУ**

Які організовувала мама

Які організовувала мама

а) “які організовувала мама”

Розквіт Карфагена припав на часи

Розквіт Карфагена припав на часи

б) “Розквіт Карфагена припав на часи”

попросіть будь ласка його!

попросіть будь ласка його!

в) “попросіть будь ласка його!”

70 (сімдесят)

70 (сімдесят)

г) “70 (сімдесят)”

встановлення нових рекордів

встановлення нових рекордів

г) “встановлення нових рекордів”

За "Народним календарем" (172 слова)

За "Народним календарем" (172 слова)

д) "за "Народним календарем" (172 слова)"

ХВОРОБА КАНАВАНА

ХВОРОБА КАНАВАНА

е) "Хвороба Канавана"

В 1866 році тут працював

В 1866 році тут працював

є) "в 1866 році тут працював"

Рис. Д1.1 Приклади роботи системи вирівнювання рукописного тексту

Додаток Б. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях України:

- [1] Yakovchuk, O., Vasin, M. (2023). Increasing the accuracy of handwriting text recognition in medical prescriptions with generative artificial intelligence. *Technology Audit and Production Reserves*, 4 (2 (72)), 18–22. DOI: <https://doi.org/10.15587/2706-5448.2023.284998>
- [2] Яковчук О.К. (2023). Побудова швидкої та легковісної рекурентної нейронної мережі для вирішення задачі розпізнавання рукописних жестів. *Таврійський науковий вісник. Серія: технічні науки*, (4), 87-93. DOI: <https://doi.org/10.32782/tnv-tech.2023.4.11>
- [3] Yakovchuk, O., Rogoza, W. (2024). An overview of statistical and neural-based line segmentation methods for offline handwriting recognition task. *Technology Audit and Production Reserves*, 1 (2 (75)), 14–19. DOI: <https://doi.org/10.15587/2706-5448.2024.298405>

Міжнародні патенти:

- [4] D. Zhelezniakov, O. Yakovchuk, D. Olynik, V. Zaitsev., Y. Yakishyn, A. Cherneha (2021). Electronic Device and Method for Modification of Content. WIPO PCT WO2021246570A1. Publication Date 2021.12.09. <https://patents.google.com/patent/WO2021246570A1/en>
- [5] A. Cherneha, N. Sakhnenko, D. Zhelezniakov, O. Yakovchuk, V. Volkova, V. Zaitsev (2022). Method, Electronic Device and Storage Medium for Adjusting Document Style. Patent No. PN146948, Application Date 2022.12.02.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

- [6] D. Zhelezniakov, A. Cherneha, V. Zaytsev, T. Ignatova, O. Radyvonenko, and O. Yakovchuk. (2020). Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition. *25th International Conference on Intelligent User Interfaces* (ACM IUI '20). 212–220. DOI: <https://doi.org/10.1145/3377325.3377482>
- [7] O. Yakovchuk, A. Cherneha, D. Zhelezniakov and V. Zaytsev. (2020). Methods for Lines and Matrices Segmentation in RNN-based Online Handwriting Mathematical Expression Recognition Systems. *2020 IEEE Third International Conference on Data Stream Mining & Processing* (DSMP), Lviv, Ukraine. 255-261. DOI: <https://doi.org/10.1109/DSMP47368.2020.9204273>
- [8] K. Korovai, D. Zhelezniakov, O. Radyvonenko, O. Yakovchuk, I. Deriuga, N. Sakhnenko. (2023). Recognition-Independent Handwritten Text Alignment Using Lightweight Recurrent Neural Network. In *SIGGRAPH Asia 2023 Posters* (SA '23). Association for Computing Machinery, New York, NY, USA, Article 30, 1–2. DOI: <https://doi.org/10.1145/3610542.3626136>

Додаток В. АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

«ЗАТВЕРДЖУЮ»

Заступник директора
з технологій штучного інтелекту,
Самсунг РнД Інститут Україна,
Литвиненко Сергій Юрійович



« 15 » березня 2024 року

АКТ

впровадження результатів досліджень дисертаційної роботи
Яковчука Олега Костянтиновича на тему «Методи вирівнювання рукописного тексту на
основі розпізнавання з використанням машинного навчання та структурного аналізу
символів» на здобуття наукового ступеня кандидата технічних наук за спеціальністю 122
– комп'ютерні науки

Підрозділ із досліджень та розробки у галузі інтелектуального введення інформації Самсунг РнД Інститут Україна інформує, що, у контексті вирішення завдань вирівнювання рукописного тексту, було використано результати опублікованих Яковчуком О.К. матеріалів наукових робіт за темою «Методи вирівнювання рукописного тексту на основі розпізнавання з використанням машинного навчання та структурного аналізу символів». Зокрема, методи вирівнювання, що виконують структурний аналіз символів і дають змогу знаходити точні позиції всіх символів в рядку на основі результатів розпізнавання та сегментації рукописного тексту. Особливістю даного підходу є можливість враховувати індивідуальні особливості почерку, злизне написання символів та зберігати початковий стиль написання.

Застосування запропонованого Яковчуком О.К. методу покращило ефективність вирішення комплексу задач, що виникають при вирівнюванні рукописного тексту, та дозволило підвищити якість продукту. Розглянуті та запропоновані у дисертаційній роботі методи було використано при розробці комерційних проєктів, зокрема мобільного додатку «Samsung Notes» для флагманської моделі смартфона Samsung Galaxy S24 у вигляді функції вирівнювання рукописного тексту для різних мов.

к.т.н., доцент, провідний інженер
Самсунг РнД Інститут Україна,
керівник лабораторії "Input Intelligence Lab"

 Радивоненко Ольга Сергіївна
« 15 » березня 2024 року