

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук і кібернетики
Кафедра математичної інформатики
Центр науково-дослідних і дослідно-конструкторських розробок
ТОВ "Самсунг Електронікс Україна Компані"

Дипломна робота
на здобуття ступеня магістра
за спеціальністю 122 Комп'ютерні науки

РОЗРОБКА АЛГОРИТМІВ СЕГМЕНТАЦІЇ РУКОПИСНИХ ТАБЛИЧНИХ СТРУКТУР

Виконала студентка II курсу магістратури
Чернега Анастасія Олександрівна

_____ (підпис)

Науковий керівник:
професор, доктор фізико-математичних наук
Терещенко Василь Миколайович

_____ (підпис)

Консультант (ментор ЦНДДКР ТОВ "Самсунг
Електронікс Україна Компані"):
старший інженер-програміст
Железняков Дмитро Валентинович

_____ (підпис)

Засвідчую, що в цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студентка

_____ (підпис)

Роботу розглянуто й допущено до захисту на
засіданні кафедри математичної інформатики
" ____ " _____ 2021 р.

протокол № _____

Завідувач кафедри
Терещенко Василь Миколайович

_____ (підпис)

РЕФЕРАТ

Обсяг роботи 52 сторінки, 35 ілюстрацій, 3 таблиці, 41 джерело посилань.

СЕГМЕНТАЦІЯ ТАБЛИЧНИХ СТРУКТУР, СЕГМЕНТАЦІЯ МАТЕМАТИЧНИХ МАТРИЦЬ, ДІАГРАМА ВОРОНОГО, РУКОПИСНІ МАТЕМАТИЧНІ ВИРАЗИ, ОНЛАЙН РУКОПИСНЕ ВВЕДЕННЯ, ПРОФАЙЛІНГ ПРОЕКЦІЙ.

Об'єктом роботи є процес сегментації рукописних табличних структур. Предметом роботи є розроблені методи сегментації табличних структур:

1. Модифікований метод профайлінгу проекцій.
2. Площинні діаграми Вороного для сегментації табличних структур.

Метою роботи є розробка алгоритмів сегментації табличних структур до розпізнавання, що можуть бути інтегровані у сучасну систему онлайн розпізнавання рукописних математичних виразів для сегментації математичних матриць. Методи дослідження: аналіз існуючих алгоритмів сегментації, спостереження і аналіз різних варіантів рукописних математичних матриць, розробка нових алгоритмів, експериментальна перевірка розроблених алгоритмів, інтеграція алгоритмів у систему онлайн розпізнавання рукописних математичних виразів. Інструменти розроблення: мова програмування C++, інтегроване середовище розробки CLion для мови C++, система онлайн розпізнавання математичних рукописних виразів.

Результати роботи: виконано загальний огляд систем розпізнавання математичних рукописних виразів, проаналізовано вимоги для модуля сегментації матриць, проаналізовано існуючі методи сегментації, які можна взяти за основу розроблюваних методів, проведено спостереження і аналіз можливих варіантів написання матриць, розроблено програмні модулі для проведення сегментації математичних матриць, розроблено програму для тестування ефективності методів сегментації матриць, проведено експериментальну перевірку запропонованих методів, проведено інтеграцію методів у систему розпізнавання.

Метод модифікованого профайлінгу проєкцій перевершив інші опубліковані результати за всіма метриками на матричних наборах даних CROHME 2014 і CROHME 2016. Частка матриць, повністю правильно відсегментованих цим методом становить 94.19% на наборі даних CROHME 2016.

Програмні модулі сегментації можуть використовуватися в системах онлайн розпізнавання рукописних математичних виразів та інших системах, що вимагають сегментацію рукописного введення табличної структури. Запропоновані методи також можна застосувати до інших табличних структур, наприклад, рукописних таблиць.

ЗМІСТ

1	Огляд підходів до розпізнавання математичних виразів та методи сегментації двовимірних рукописних структур	9
1.1	Розвиток систем розпізнавання онлайн рукописних виразів .	9
1.2	Методи сегментації математичних матриць	11
1.3	Метод профайлінгу проекцій	13
1.4	Діаграми Вороного для задачі сегментації	14
2	Система розпізнавання	16
2.1	Опис системи розпізнавання онлайн рукописних математичних виразів	16
3	Модифікований метод профайлінгу проекцій для рукописних матриць	19
3.1	Опис базового метода	19
3.2	Опис модифікованого метода	21
3.3	Застосування динамічного профайлінгу проекцій	25
4	Сегментація рукописних матриць на основі площинної діаграми Вороного	27
4.1	Побудова площинної діаграми Вороного	27
4.2	Запропонований метод сегментації	28
5	Оцінка запропонованих методів	38
5.1	Результати експериментів	38
5.2	Часова ефективність	39
5.3	Обговорення результатів сегментації модифікованим методом профайлінгу проекцій	40
5.4	Майбутній розвиток сегментації модифікованим методом профайлінгу проекцій	41
5.5	Обговорення результатів сегментації з використанням площинної діаграми Вороного	42

5.6	Майбутній розвиток сегментації з використанням площинної діаграми Вороного	43
	Література	47

ВСТУП

Актуальність теми. Найбільш природний і знайомий великій кількості людей з дитинства спосіб зберегти певну інформацію – це рукописне введення. Для різних сфер застосування було розроблено велику кількість нотацій і способів представлення, таких як таблиці, блок-схеми, хімічні формули, нотний запис, математичні вирази та багато інших. Широка розповсюдженість електронних пристроїв з сенсорними екранами та цифрових ручок, зростаючий інтерес до інтерфейсів типу “папір і ручка” [9, 14, 40], а також тенденція до оцифрування даних викликає необхідність у системах розпізнавання різних видів рукописного введення, у тому числі і математичних виразів. Стандартні способи введення (клавіатура, миша) більш пристосовані для введення одновимірних мов. Рукописне введення, згідно з результатами досліджень користувачів [9, 14, 40], при хорошій якості розпізнавання краще підходить для введення двовимірних мов, до яких належить і математична нотація. На ринку представлено ряд програмних продуктів для мобільних пристроїв, що надають можливість працювати з інтерфесами типу “папір і ручка”: “Nebo” [3], “MyScript Calculator” [2], “Microsoft Math Solver” [1]. “Microsoft Math Solver” має більше 10 мільйонів встановлень на Google Play [4].

Математична нотація широко використовується як для вирішення побутових задач широким загалом, так і в багатьох спеціалізованих сферах: математика, інженерія, фінанси, фізика та інші. Отже, розпізнавання математичних виразів є однією з важливих підзадач розпізнавання рукописного введення.

Для повноцінної роботи системи розпізнавання математичних виразів необхідна підтримка різних видів математичних виразів, в тому числі і математичних матриць. Однією з найбільш перспективних сфер застосування систем розпізнавання рукописних математичних виразів є освіта. На тлі пандемії освіта глобально перейшла в дистанційний формат, що робить тему роботи як ніколи актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів сегментації рукописних математичних матриць. Для цього потрібно розв'язати наступні завдання:

1. Дослідити наявні методи сегментації математичних матриць.
2. Дослідити наявні методи, які можна потенційно адаптувати для сегментації математичних матриць.
3. Розробити модифікації обраних методів.
4. Експериментально перевірити розроблені модифікації.
5. Проаналізувати результати експериментів.
6. Інтегрувати методи у систему розпізнавання рукописних математичних виразів.

Об'єкт дослідження. Об'єктом дослідження є сегментація двовимірних рукописних структур.

Методи дослідження. Основними методами дослідження є аналіз, спостереження, гіпотеза та експеримент. Спочатку аналізуються існуючі методи сегментації двовимірних рукописних структур. Потім проводяться спостереження різноманітних випадків написання рукописних матриць. Після цього робиться ряд припущень про покращення базових методів для кращої ефективності відповідно до спостережень. Розробляються модифіковані версії методів і експериментально перевіряється їхня ефективність.

Наукова новизна одержаних результатів. Одержані результати для модифікованого методу профайлінгу проєкцій перевищують наявні результати сегментації інших систем розпізнавання на датасетах CROHME 2016 [24] та CROHME 2014 [23]. Сегментація за допомогою площинних діаграм Вороного є першою спробою застосувати діаграми Вороного для сегментації онлайн рукописних табличних структур для двовимірних мов. Цей підхід вирішує деякі складні випадки написання матриць, однак загальна ефективність є нижчою.

Практичне значення одержаних результатів. Розроблені методи ефективно інтегруються у систему розпізнавання рукописних математичних виразів, забезпечуючи підтримку розпізнавання математичних матриць. Додавання такого модуля у систему, що використовує рекурентні нейронні мережі дозволить підтримувати розпізнавання матриць без обов'язкового перенавчання нейронної мережі.

Апробація результатів магістерської. Апробація розроблених методів проведена за місцем роботи автора.

Публікації.

1. Модифікований метод профайлінгу проєкцій:
Yakovchuk, O., Cherneha, A., Zhelezniakov, D., & Zaytsev, V. (2020, August). Methods for Lines and Matrices Segmentation in RNN-based Online Handwriting Mathematical Expression Recognition Systems. In 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) (pp. 255-261). IEEE.
2. Оцінка вимог для інтерфейсів сучасних систем онлайн розпізнавання рукописного введення:
Zhelezniakov, D., Cherneha, A., Zaytsev, V., Ignatova, T., Radyvonenko, O., & Yakovchuk, O. (2020, March). Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition. In Proceedings of the 25th International Conference on Intelligent User Interfaces (pp. 212-220).

Розділ 1

ОГЛЯД ПІДХОДІВ ДО РОЗПОЗНАВАННЯ МАТЕМАТИЧНИХ ВИРАЗІВ ТА МЕТОДИ СЕГМЕНТАЦІЇ ДВОВИМІРНИХ РУКОПИСНИХ СТРУКТУР

1.1 Розвиток систем розпізнавання онлайн рукописних виразів

Існує два основних підкласа у розпізнаванні рукописних математичних виразів: онлайн та офлайн розпізнавання. Задачею офлайн розпізнавання є перетворення математичних виразів, представлених у вигляді зображення, у формат, що може бути оброблений комп'ютером, наприклад, L^AT_EX. Онлайн розпізнавання працює з динамічним представленням рукописного введення, що записується за допомогою пристрою з сенсорним екраном. Користувач пише, використовуючи перо або палець, і траєкторії рухів по екрану знімаються у вигляді послідовності точок. Послідовність точок, записана між моментом притиснення пера до екрану та моментом відриву пера від екрану називається *штрих*. Представленням рукописного математичного виразу у задачі онлайн розпізнавання є послідовність штрихів. Приклад такого виразу показано на рис. 1.1

Таким чином, на відміну від офлайн, онлайн математичні вирази містять часову інформацію.

Для розв'язання задачі онлайн розпізнавання математичних виразів запропоновані різні варіанти систем розпізнавання. Перші підходи, як зазначає Zhang [36], зазвичай склалися з трьох основних послідовних кроків:

1. *Сегментація символів*. Штрихи групуються таким чином, щоб в одній групі опинилися тільки ті штрихи, що формують даний символ.

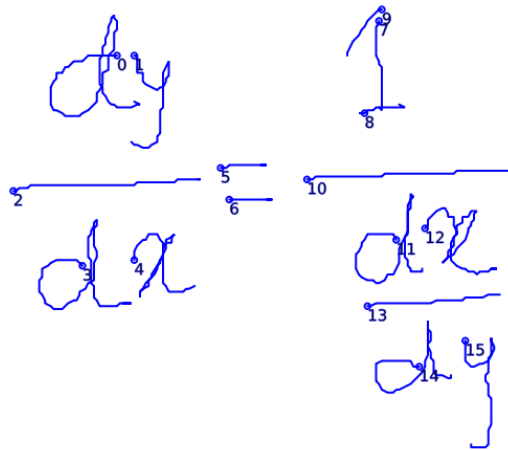


Рис. 1.1: Приклад онлайн рукописного математичного виразу

2. *Розпізнавання символів.* Кожна група штрихів має бути класифікована як один із можливих символів. Математична нотація може використовувати більше 1500 символів [8], тому дана задача є доволі складною.
3. *Структурний аналіз.* На цьому етапі проходить аналіз площинних та семантичних відношень між символами, і будується математичний вираз. Часто побудова відбувається на основі правил граматики математичної нотації.

Методи такого типу були запропоновані Kosmala and Rigoll [18], Matsakis [22], Okamoto [27], Winkler et al. [32].

Інший пласт підходів характерний тим, що замість послідовного розв'язання вказаних вище задач, відбувається інтегроване розв'язання: одночасно відбувається знаходження найкращих варіантів сегментації, розпізнавання символів та розпізнавання двовимірної структури, контрольоване граматиною математичної нотації. Ці підходи можна вважати інтегрованими підходами, що засновуються на граматиці. Такі системи розробили Awal et al. [7], Celik and Yanikoglu [10], Yamamoto et al. [33], Álvaro et al. [41].

Третім видом є системи, що проводять розпізнавання одразу цілого виразу, за допомогою кодувально-декодувальних нейронних мереж, що перетворюють послідовність штрихів у послідовність символів вихідного формату (end-to-end). Zhang et al. [35] запропонували використати кодувально-декодувальну нейронну мережу, в якій кодувальник складається з дво-

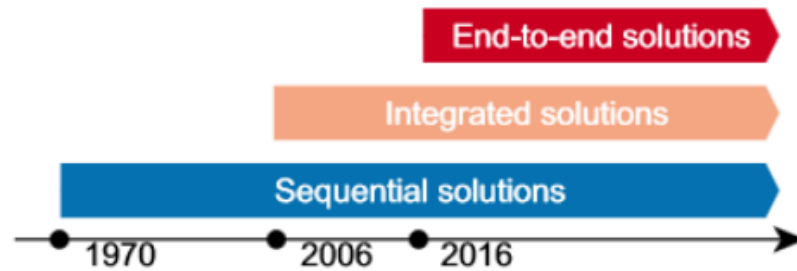


Рис. 1.2: Розвиток систем розпізнавання

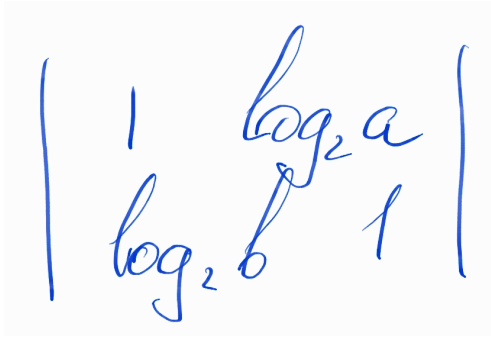
сторонніх рекурентних прошарків з комірками типу GRU, а декодувальник складається з односторонніх рекурентних прошарків з комірками типу GRU а механізмом уваги. У Zhang et al. [37] розроблено систему на основі рекурентної мережі типу BLSTM з використанням техніки CTC (Connectist temporal classification) [13], що дозволяє встановити відповідність між вхідними штрихами та виводом нейронної мережі.

Zhelezniakov et al. [38] узагальнено демонструють еволюційний розвиток різних типів систем розпізнавання на рис. 1.2.

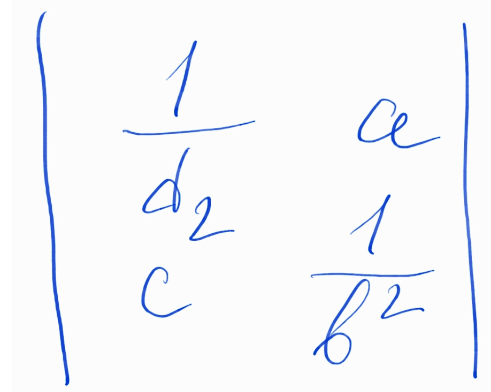
1.2 Методи сегментації математичних матриць

Підтримка математичних матриць є однією з вимог до повноцінних систем розпізнавання математичних виразів. Задача сегментації математичних матриць полягає у визначенні розмірності матриці та встановленні відповідності між символами або штрихами та елементами матриці: комірками, рядками і стовпчиками. Під час розв'язання даної задачі виникають різні складнощі та нюанси. Так, серед рукописних матриць зустрічаються перетини комірок (рис. 1.3а), неоднозначні варіанти написання (рис. 1.3б), написання зі скошеними рядками.

Також, матриці можуть бути неповними, містити трикрапки для позначення довільної кількості елементів, або позначати всі комірки вище або нижче діагоналі за допомогою лише одного символу. Підходи до сегментації математичних матриць були описані у ряді робіт. Розглядалися друквані офлайн матриці, рукописні офлайн матриці та рукописні онлайн матриці. Як вже зазначалося у попередній секції, значна кількість систем розпізнавання спочатку виконують сегментацію та розпізнавання



(а) Перетин матричних комірок



(б) Неоднозначність у приналежності символа до конкретної комірки

Рис. 1.3: Складні ситуації у сегментації рукописних матриць

символів. Відповідно, сегментатори матриць у таких системах вже мають інформацію про тип символа. У методі для офлайн матриць, запропонованому Li et al. [20], спочатку на основі обмежувальних паралелепіпедів символів та інформації про результати їх розпізнавання створюються групи символів для елементів матриці. Потім на основі знайдених елементів формуються рядки і стовпчики. У Kanahori and Suzuki [16] символи також спочатку об'єднуються у елементи на основі коефіцієнтів, спеціально підібраних для різних груп символів, а структура матриці розглядається як мережа елементів, з'єднаних між собою відношеннями. Цей метод застосовується для друкованих офлайн матриць, але підтримує сегментацію матриць зі скороченим записом та площинними елементами (елементами, що пишуть замість декількох суміжних однакових елементів, часто над чи під діагоналлю). Tausky et al. [31] працюють з онлайн рукописними матрицям. Вони групують символи у елементи на основі середньої ширини символів, а також окремо обробляють трикрапки. Також в статті пропонується визначення структури матриці шляхом кластеризації вертикальних та горизонтальних проєкцій центроїдів символів (рис. 1.4).

Метод Taria and Rojas [30] теж працює з онлайн матрицями, застосовується мінімальне кістякове дерево для знаходження рядків матриці, а розділення рядка на елементи відбувається вже на основі відстаней.

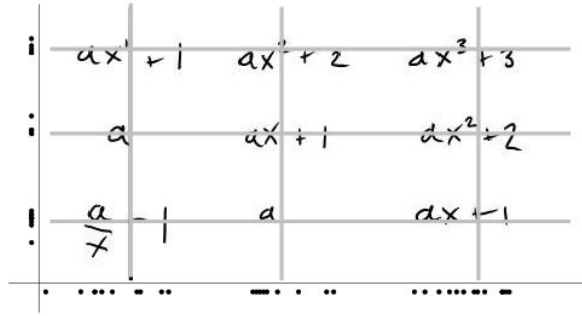


Рис. 1.4: Кластеризація центроїдів символів

1.3 Метод профайлінгу проекцій

Даний метод прийшов з області комп'ютерного зору. Він широко використовується для аналізу документів. Запропонований вперше для аналізу документів Na et al. [15], де описано використання обмежувальних прямокутників компонент зв'язності пікселів для побудови проекцій. Проекції використовувалися для сегментації рядків, слів та блоків тексту. Приклад побудови проекції для сегментації рядків зазначеної роботи показано на рис. 1.5.

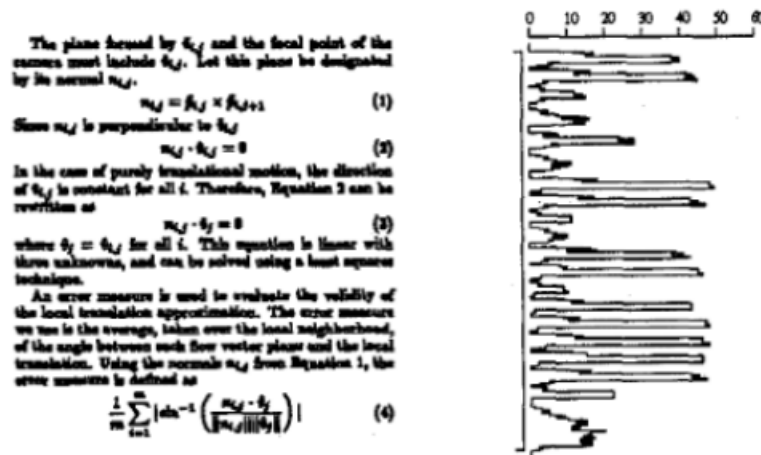


Рис. 1.5: Застосування профайлінгу проекцій для сегментації рядків тексту Na et al. [15].

Цікавим є застосування профайлінгу проекцій у Antonacopoulos and Karatzas [6] для задачі оцифрування архівних записів часів Другої Світової Війни, а саме транспортних листків та каталогових карток з нацистського концентраційного табору Штуттгоф. Документи містять друкований текст, і мають певну фіксовану структуру, і для підготовки до роз-

пізнавання лишалося провести сегментацію рядків тексту і потім літер. Оскільки літери для кожного класу документів мали певний шрифт, алгоритм підлаштували таким чином, щоб знаходити зразкову відстань між послідовними розділювачами шляхом побудови гістограм відстаней між усіма розділювачами і обранням такої, що зустрічається найчастіше.

dos Santos et al. [11] застосовують модифіковану версію профайлінгу проєкцій для сегментації рядків рукописного тексту. Запропонований алгоритм спочатку використовує порогове значення, пропорційне середній довжині рядка, щоб відфільтрувати хибні “пагорби” проєкцій, а потім відфільтровують хибні рядки по висоті за допомогою порогового значення, пропорційного середній висоті рядка. Приклад побудови проєкції для рукописних рядків зазначеної роботи показано на рис. 1.6.

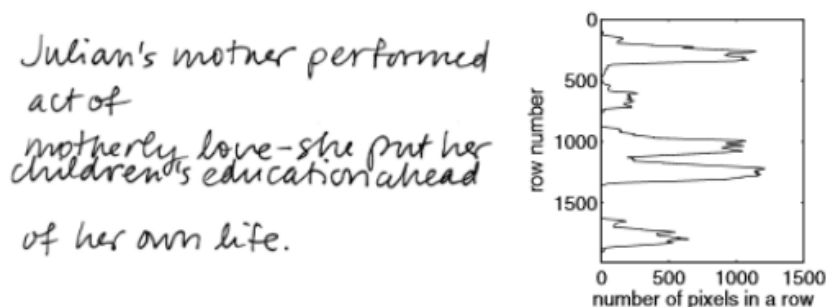


Рис. 1.6: Приклад побудови проєкції для рукописних рядків dos Santos et al. [11].

Zahour et al. [34] пропонують комбінацію метода часткового профайлінгу проєкцій з частковим обходом за контуром для сегментації рядків рукописного арабського тексту. Також, після основної сегментації проходить спеціальне призначення діакритичних знаків до відповідних рядків.

1.4 Діаграми Вороного для задачі сегментації

Діаграми Вороного широко застосовуються для різних задач сегментації.

Kise et al. [17] розробили метод на основі площинних діаграм Вороного, який визначає логічні секції документа. Для цього було введено спеціальні пороги, засновані на двох ознаках: площа комірки діаграми Вороного та мінімальна відстань для кожного ребра діаграми. Частина ідей з даної

публікації адаптовано в цій роботі для сегментації математичних матриць. Приклад результатів роботи цього метода показано на рис. 1.7.



Рис. 1.7: Результат сегментації документа методом Kise et al. [17].

Agrawal and Doermann [5] ґрунтуються на попередньому методі, але визначають динамічні пороги локально, краще адаптуючись до локальних варіацій у розмірі, орієнтації, пропусках. Також для покращення точності використовуються деякі о і із методу Docstrum [26].

Інша область, де застосовувалися діаграми Вороного для сементації - це сегментація мережі біологічних нейронів на зображеннях мікроскопії. Заснований на діаграмах Вороного алгоритм скелетонізації описаний у Kotsur and Tereshchenko [19].

Розділ 2

СИСТЕМА РОЗПІЗНАВАННЯ

В цьому розділі описано систему розпізнавання, складовою частиною якої має стати модуль сегментації рукописних математичних матриць.

2.1 Опис системи розпізнавання онлайн рукописних математичних виразів

Основні модулі і процес роботи системи зображено на рис. 2.1. Модуль сегментації ліній (Line Segmentation) на основі геометрії штрихів розподіляє їх на окремі вирази. Модуль виявлення матриць виявляє матриці у вхідних виразах. Це відбувається шляхом геометричного виявлення круглих, квадратних та прямих дужок відносно великого розміру, та формування пар дужок із них. Після цього штрихи, які знаходяться в області між парою дужок, надходять у модуль сегментації матриць (Matrix Segmentation). Сегментація матриць буде описана в наступних розділах.

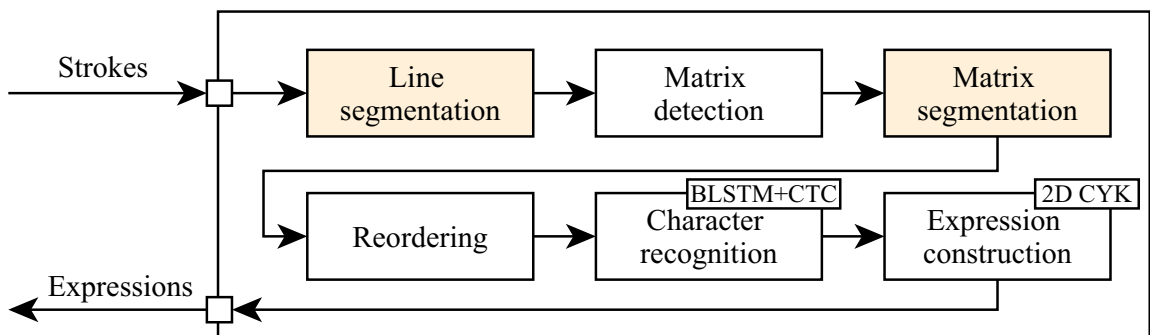


Рис. 2.1: Процес роботи системи розпізнавання.

Модуль розпізнавання символів (Character Recognition) проводить сегментацію та розпізнавання символів. Він заснований на рекурентній нейронній мережі із довгою короткочасною пам'яттю (BLSTM) та з нейро-

мережевою часовою класифікацією (Connectionist Temporal Classification (CTC)) [12, 21]. Побудова виразів (Expression Construction) здійснюється за допомогою стохастичної контекстно-вільної граматики (Probabilistic Context-Free Grammar(PCFG)) [25, 39] і також використовує N-грамну модель мови.

Довга короткочасна пам'ять виражається наступними рівняннями для комірки нейронної мережі:

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t),
\end{aligned} \tag{2.1}$$

де x_t – вхідний вектор на момент часу t ; f_t – вектор забувального вентиля; i_t – вектор входового вентиля; o_t – вектор виходового вентиля; h_t – вектор прихованого стану; \tilde{c}_t – вектор активації входу комірки; c_t – вектор стану комірки; W – матриця вагів; b – вектори параметрів упередження; σ_g – логістичний сигмоїд; σ_h – гіперболічний тангенс. Входом є нормалізований вектор ознак: Δx , Δy , а також показники притискання/відривання пера від екрану.

Нейромережева часова класифікація пов'язана із функцією оцінки та може буде представлена наступною формулою:

$$O(S) = -\ln \left(\prod_{(x,z) \in S} p(z|x) \right) = - \sum_{(x,z) \in S} \ln p(z|x), \tag{2.2}$$

де x – вхідна послідовність і z – позначки основної правди (ground truth).

Такий підхід розпізнає одразу цілий вираз, але встановлює відповідність між результатом та вхідною послідовністю, знаходячи в ній межі символів. Однак така мережа дуже залежить від порядку надходження вхідних штрихів, і під час тренування і роботи системи такий порядок повинен співпадати. Порядок, обраний для даної системи, такий: зверху вниз і зліва направо. Зафіксувавши порядок, можна отримувати кращі резуль-

тати від нейронної мережі за рахунок надання їй правильного контексту. Але, люди не завжди пишуть в такому порядку, часто зустрічається ситуація з відкладеними штрихами, наприклад, написання крапки над літерою 'i', або дописування індексів математичного виразу після написання основної частини всього виразу. У матрицях також немає жодної гарантії, що порядок вхідних штрихів буде таким, як ми визначили. А надання нейронній мережі штрихів не в тому порядку, як було під час навчання, значно погіршує результати.

Тому, сегментація матриць потрібна спершу для того, щоб надати нейронній мережі штрихи в правильному порядку. Потім, після розпізнавання символів результати сегментації використовуються знову на етапі побудови виразу. Саме через такі вимоги до алгоритму сегментації використання інформації про розпізнанні символи неможливе для первинної сегментації. Тому, запропоновані методи засновуються лише на геометричних ознаках.

Розділ 3

МОДИФІКОВАНИЙ МЕТОД ПРОФАЙЛІНГУ ПРОЕКЦІЙ ДЛЯ РУКОПИСНИХ МАТРИЦЬ

3.1 Опис базового метода

За спостереженнями, часто трапляються матриці, де певний рядок чи стовпчик значно чіткіше виражає структуру матриці, ніж інші рядки та стовпчики. Наприклад, на рис. 3.1 найкраще видно структуру у першому рядку і першому стовпчику. Це спостереження використовується у запропонованому алгоритмі шляхом вибору певного рядка чи стовпчика як шаблону сегментації.

$$R_{\alpha}(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Рис. 3.1: Матриця, структура якої краще виражена першим рядком і першим стовпчиком

Базовий алгоритм для даної модифікації - це відомий метод профайлінгу проєкцій. Кроки використання базового методу для сегментації штрихів виглядають наступним чином:

1. Обрахувати обмежувальний прямокутник кожного штриха.
2. Спроекувати обмежувальний прямокутник на вісь X та вісь Y.
3. Виділити на обох вісях сегменти з більшою концентрацією проєкцій.

4. Знайти перетини координат сегментів - орієнтовні прямокутники, що позначають розташування елементів матриці.
5. Розподілити штрихи до найближчих прямокутників.

Такий підхід базується на спостереженні, що матриці мають табличну структуру, і очікується, що навпроти рядків на вісі Y буде помітна більша концентрація проєкцій, а навпроти стовпчиків більша концентрація проєкцій буде на вісі X . Проєкції додаються на вісі таким чином: для обох вісей створюються масиви розмірністю в $x_{max} - x_{min}$ для вісі X і $y_{max} - y_{min}$ для вісі Y . Потім для кожного обмежувального прямокутника, обмеженого координатами x_{left} , x_{right} , y_{top} , y_{bottom} виконується збільшення значення масиву проєкцій для вісі X для усіх значень, що відповідають координатам від x_{left} до x_{right} , а для вісі Y - для усіх значень, що відповідають координатам від y_{top} до y_{bottom} . Після виконання цієї операції для всіх штрихів матриці, масиви проєкцій мають містити сегменти більших значень, розділені сегментами менших значень. Щоб визначати сегменти навпроти рядків та стовпчиків матриці, використовується змінне порогове значення T . Значення T залежить від величини найбільшого значення у масиві проєкцій: чим воно більше, тим більший поріг T , але не більше певного обмежувально значення T_{max} , яке визначається евристично. На рис. 3.2 помаранчевим пунктиром позначено поріг T , а червоним - сегменти з інтенсивністю проєкцій, що перевищує T . Після цього можна знайти перетини координат знайдених сегментів та знайти орієнтовні місця розташування елементів матриці. Далі кожен штрих відноситься до найближчого місця перетину координат сегментів і будується сегментація матриці.

Оскільки графік проєкцій може містити багато локальних мінімумів та максимумів, додатково відбувається відсіювання сегментів, які розділені занадто короткими відстанями. Для цього потрібно спочатку обрахувати всі проміжки між двома послідовними сегментами, і знайти найдовший з довжиною MAX_GAP . Якщо два послідовні сегменти s_i та s_{i+1} не задовольняють нерівності 3.1, то вони з'єднуються у один.

$$left(s_{i+1}) - right(s_i) > a \cdot MAX_GAP \quad (3.1)$$

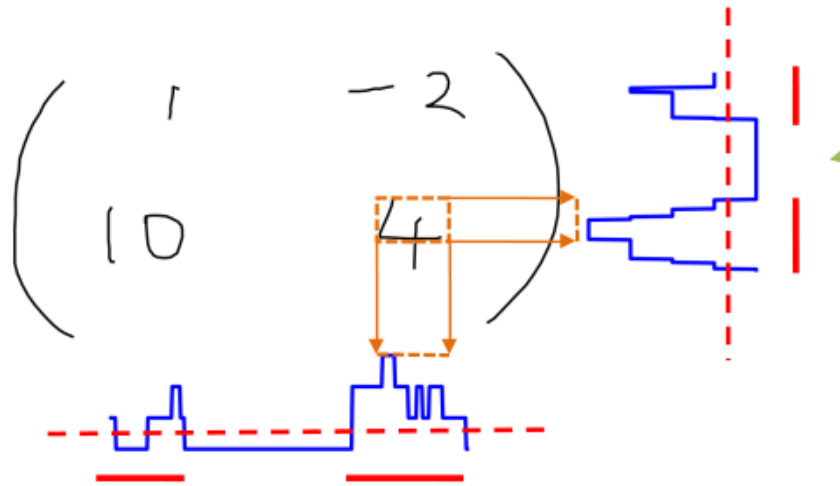


Рис. 3.2: Приклад знаходження сегментів з найбільш інтенсивними проєкціями

3.2 Опис модифікованого методу

Однак, часто зустрічаються матриці з таким написанням, що сегменти концентрації проєкцій знайти доволі складно, і часто є неоднозначності. Базова версія профайлінгу проєкцій не справляється з рукописними матрицями достатньо добре. До основних проблем можна віднести

- Нерівномірні пропуски між рядками або стовпчиками.
- Недостатні пропуски між рядками або стовпчиками.
- Велика варіативність у розмірах елементів однієї матриці.
- Матриці, скошено написані, коли рядки “повзуть” вгору або вниз.

Тому пропонується декілька модифікацій для створення методу, що краще справиться із задачею.

Вибір ведучої вісі. Згідно зі спостереженнями, у рукописних матрицях часто більш чітко розділення видно або між рядками, або між стовпчиками. Так, на рис. 3.3 можна побачити, що пропуски між стовпчиками більш очевидні та однакові, і, відповідно масив проєкцій для вісі X значно чіткіше виражає три пагорби, що відповідають трьом стовпчикам. На 3.4 показано матрицю, у якій більш чітко видно структуру рядків, і так само і проєкція

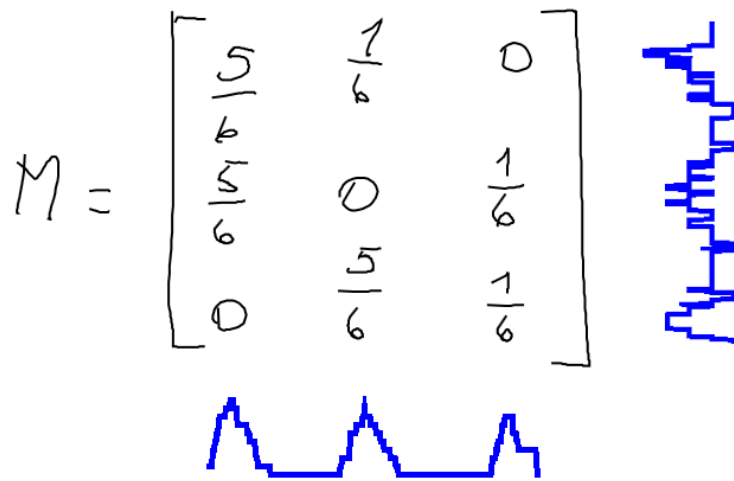


Рис. 3.3: Матриця, де стовпчики більш чітко розділені, ніж рядки

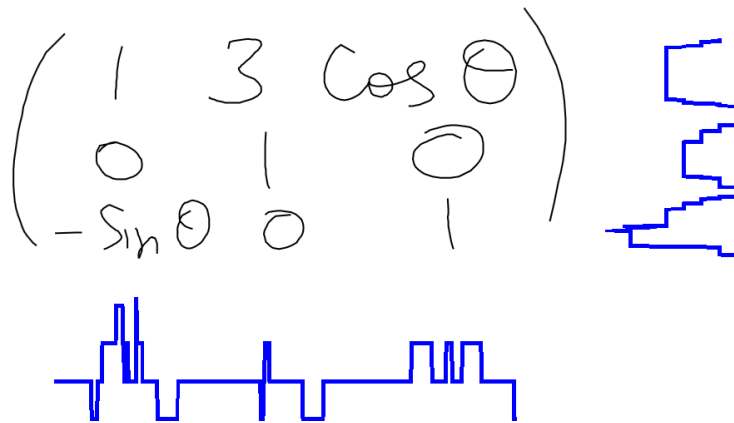
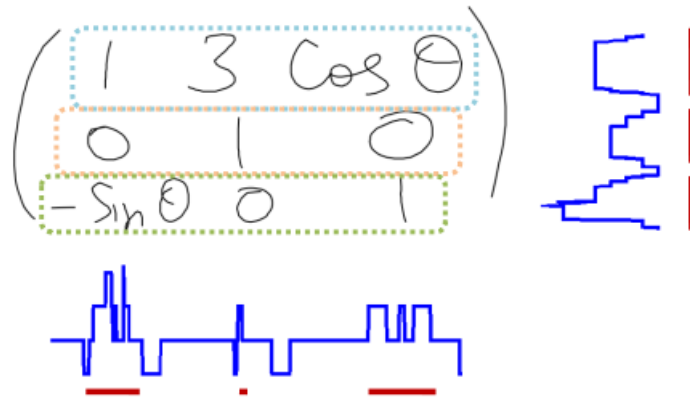


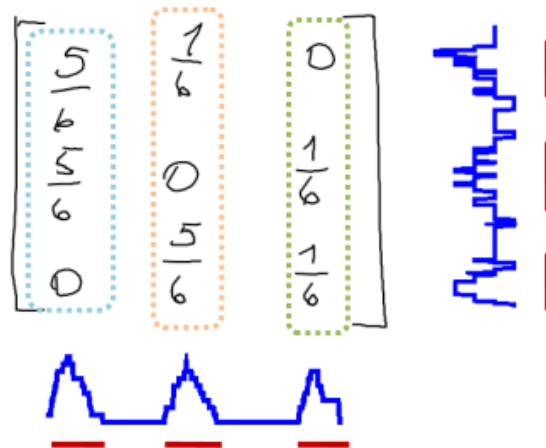
Рис. 3.4: Матриця, де рядки більше чітко розділені, ніж стовпчики

для вісі Y виглядає значно більш очевидною, ніж проекція для вісі X . Коли побудована проекція вийшла недостатньо чіткою, правильно визначити структуру матриці доволі складно. Тому, після побудови проекцій, модифікований метод обирає тільки одну вісь: ту, проекції для якої визначені, як більш чіткі. Обирається та вісь, довжини сегментів котрої мають меншу дисперсію.

Вибір шаблонного рядка чи стовпчика. Сегменти, побудовані на основі обраної ведучої вісі, відповідають або рядкам, або стовпчикам, тобто, на цьому етапі ми знаємо приблизні координати або рядків, або стовпчиків. Далі, відповідно до обраних сегментів, потрібно тимчасово розподілити всі штрихи матриці до найближчих рядків (стовпчиків). З утворених груп штрихів потрібно обрати такий рядок (стовпчик), що буде використовуватися як шаблон сегментації.



(а) Приклад тимчасового розділення матриці на рядки



(б) Приклад тимчасового розділення матриці на стовпчики

Рис. 3.5: Приклад тимчасового розділення матриці, необхідного для обрання шаблонного рядка (стовпчика)

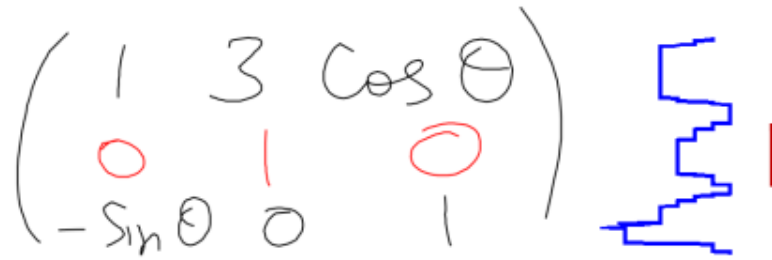
Приклад розподілення рядків та стовпчиків показано на рис. 3.5.

Тобто, саме по шаблону буде визначена кількість стовпчиків, якщо кількість рядків вже відома, і кількість рядків, якщо кількість стовпчиків вже відома. Було випробувано два способи вибору шаблону:

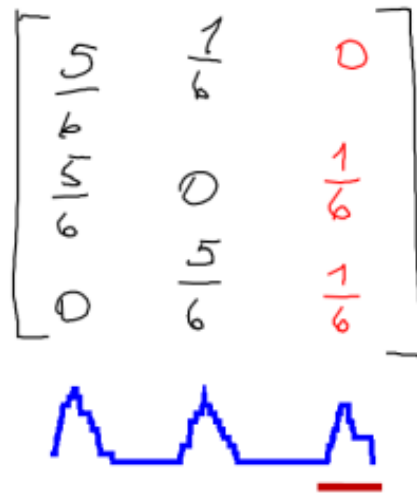
1. Обрати рядок (стовпчик), що містить найменшу кількість штрихів.
2. Обрати рядок (стовпчик), в якому знайдено найбільший пропуск.

Перший варіант заснований на міркуваннях, що при коротших виразах у матричних елементах, елементи будуть краще розділені пропусками, ніж при довших. Другий варіант заснований на припущенні, що найбільший пропуск скоріше за все трапиться у гарно розділеному пропусками рядку чи стовпчику.

Вибір шаблонного рядка та стовпчика першим способом показано на рис. 3.6. Обрані рядок і стовпчик позначені червоним кольором.



(а) Приклад вибору шаблонного рядка



(б) Приклад вибору шаблонного стовпчика

Рис. 3.6: Приклад тимчасового розділення матриці, необхідного для обрання шаблонного рядка (стовпчика)

Коли шаблонний рядок (стовпчик) обрано, потрібно визначити кількість елементів в ньому. В експериментах для цього знову використовувалася побудова проєкцій і знаходження сегментів, але тільки для однієї, потрібної вісі. Таким чином, після виконання попередньої дії, стає відомою N - кількість елементів у всіх інших рядках (стовпчиках). На основі цієї інформації сегментуються всі інші рядки (стовпчики) шляхом пошуку $N - 1$ найбільших пропусків у кожному з них.

До процесу побудови проєкцій також додана спеціальна обробка маленьких штрихів: їхні обмежувальні прямокутники збільшуються до певної заданої мінімальної величини.

Блок-схема описаного методу зображена на рис. 3.7.

3.3 Застосування динамічного профайлінгу проєкцій

Наступним витком розвитку методу стала заміна базового варіанту профайлінгу проєкцій на варіант зі змінним порогом. Якщо раніше поріг визначався рівномірним збільшенням порогу залежно від розміру глобального максимуму проєкції, то наступний підхід визначає поріг більш продуманим способом. Ptak et al. [28] запропонували встановлювати свій відносний поріг для кожного піка проєкції. Спочатку, розраховується профайлінг проєкцій, а потім всі значення сортуються від більшого до меншого. Далі, для кожного значення проєкції $H(X(i)) > \alpha \max(H)$, де H - проєкція, X - номер координати, $i = (1, n)$, α - параметр, що обирається евристично (дослідники використовували значення 0.1), виконуються наступні кроки:

1. Підрахувати відносний поріг $t_\alpha = tH(X(i))$.
2. Для координати $X(i)$ в оригінальній проєкції (до сортування) знайти такий інтервал R :

$$\begin{aligned} R = [x_1, x_2] : & (\forall x \in [x_1, x_2], H(x) \geq t_\alpha) \\ & \cup (x_1 \leq X(i) \leq x_2) \\ & \cup (H(x_1 - 1) < t_\alpha) \\ & \cup (H(x_2 + 1) < t_\alpha). \end{aligned}$$

3. Якщо R не перетинається із вже знайденими інтервалами, додати його до них.

В результаті буде знайдено сегменти, що відповідають координатам рядків (стовпчиків). Цей алгоритм використовується в запропонованому методі для пошуку сегментів. Ідея пошуку ведучої вісі та шаблонного рядка (стовпчика) залишається на інших кроках методу.



Рис. 3.7: Блок-схема модифікованого методу профайлінгу проєкцій

Розділ 4

СЕГМЕНТАЦІЯ РУКОПИСНИХ МАТРИЦЬ НА ОСНОВІ ПЛОЩИННОЇ ДІАГРАМИ ВОРОНОГО

4.1 Побудова площинної діаграми Вороного

Нехай $P = \{p_1, \dots, p_n\}$ - точки на площині, $d(p, q)$ - евклідова відстань між точками p і q . Тоді комірку Вороного можна описати формулою

$$V(p_i) = \{p \mid d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (4.1)$$

Тоді діаграмою Вороного є

$$V(P) = \{V(p_1), \dots, V(p_n)\} \quad (4.2)$$

Нехай $G = \{g_1, \dots, g_n\}$ - фігури на площині, що не перетинаються, $d(p, g_i)$ - евклідова відстань між точкою p і фігурою g , що визначається так:

$$d(p, g_i) = \min_{q \in g_i} d(p, q) \quad (4.3)$$

Тоді комірку площинної діаграми Вороного можна описати формулою

$$V(g_i) = \{p \mid d(p, g_i) \leq d(p, g_j), \forall j \neq i\} \quad (4.4)$$

А площинною діаграмою Вороного є

$$V(G) = \{V(g_1), \dots, V(g_n)\} \quad (4.5)$$

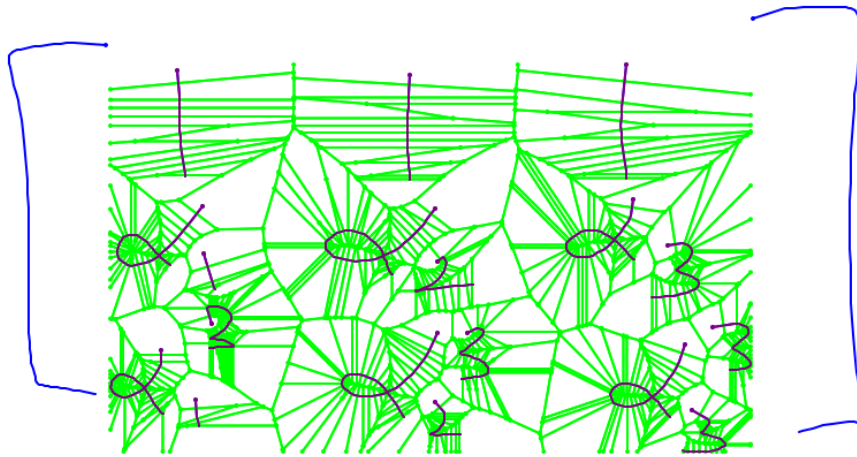
Побудувати наближену площинну діаграму Вороного можна за допомогою наступних кроків:

1. Обрати множину точок $P_i = \{p_{i1}, \dots, p_{im_i}\}$, що лежать на границях фігури g_i .
2. Побудувати звичайну діаграму Вороного з точок $P = P_1 \cup \dots \cup P_n$.
3. Для всіх i, j, k , видалити ребро побудованої діаграми, утворене точками p_{ij} та p_{ik} , якщо вони належать одній компоненті зв'язності.

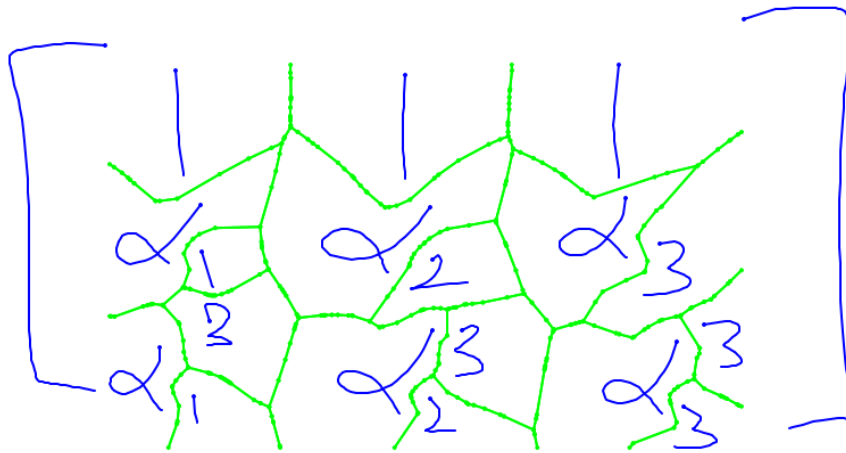
4.2 Запропонований метод сегментації

Запропонований метод заснований на площинній діаграмі Вороного, побудова якої описана у [29] і наведена у попередній секції. Нижче наведені основні кроки методу.

1. Побудувати звичайну діаграму Вороного на основі точок штрихів.
2. Об'єднати штрихи у компоненти зв'язності.
3. На основі результатів кроків 1 та 2 побудувати площинну діаграму Вороного.
4. Для кожного ребра в утвореній діаграмі, знайти *мінімальну відстань*.
5. Побудувати два графіки частот мінімальних відстаней: для вертикальних і для горизонтальних ребер.
6. Знайти порогове значення T , котре має розділити такі мінімальні відстані, що розділяють компоненти всередині одного елемента матриці, і такі, що розділяють компоненти, приналежні різним елементам матриці.
7. Підрахувати *вагу* для кожного ребра E_{ij} .
8. Знайти вхідні ребра - ребра, які мають бути початком або кінцем послідовності ребер, що розділяють рядки або стовпчики матриці.
9. За допомогою Алгоритму 1 знайти шляхи ребер, що розділяють рядки матриці, та шляхи ребер, які розділяють стовпчики матриці.
10. На основі знайдених шляхів побудувати сегментацію матриці.



(а) Звичайна (точкова) діаграма Вороного, *крок 1*



(б) Площинна діаграма Вороного, *крок 3*

Рис. 4.1: Приклад побудови площинної діаграми Вороного на рукописній математичній матриці

На кроці 1 будується звичайна діаграма Вороного, використовуючи точки, з яких складаються штрихи. Оскільки площинна діаграма оперує з фігурами, що не перетинаються, то на кроці 2 потрібно об'єднати штрихи, що перетинаються, в компоненти зв'язності. На кроці 3 ті ребра, що утворені двома точками, що належать одній компоненті зв'язності, видаляються. Таким чином, у результуючій діаграмі ребро E_{ij} розділяє компоненти зв'язності c_i та c_j , і складається з послідовності сегментів e_1, \dots, e_k , котрі є ребрами точкової діаграми Вороного. Ці кроки показано на рис. 4.1.

Kise et al. [17] застосовують мінімальну відстань як одну із характеристик для того, щоб розрізнити відстані між символами і відстані між елементами документа. Для рукописних матриць припустимо, що відстані між компонентами зв'язності всередині однієї матричної комірки є значно

меншими, ніж відстані між компонентами зв'язності, що належать різним сусіднім коміркам. Також припустимо, що варіативність у кожній з цих двох груп відстаней порівняно невелика. Отже, адаптуємо ідею із розглянутої статті таким чином: ребра, що відповідають більшим відстаням, мають ділити сусідні комірки, а ребра, що відповідають меншим відстаням - штрихи в одній комірці.

Відповідно, на кроці 4 для кожного ребра E_{ij} площинної діаграми розраховується мінімальна відстань. E_{ij} складається з ребер точкової діаграми e_1, \dots, e_n , кожне e_k з яких утворене парою точок p_{ik} та p_{jk} , а мінімальна відстань визначається наступним чином

$$\minDistance(E_{ij}) = \min_{1 \leq k \leq n} distance(p_{ik}, p_{jk}) \quad (4.6)$$

де $distance(a, b)$ - евклідова відстань.

На кроці 5 будується два графіка частот мінімальних відстаней. Обрано будувати два графіка тому, що рукописні матриці часто мають відмінні відстані між сусідніми комірками по вертикалі і по горизонталі. Тому, дані для побудови графіків обираються окремо для вертикальних та горизонтальних проміжків таким чином: якщо обмежувальні паралелепіпеди двох сусідніх компонент зв'язності c_i та c_j перетинаються за координатою X, то мінімальна відстань для ребра E_{ij} додається до даних вертикальних проміжків, а якщо вони перетинаються за координатою Y, то відстань додається до даних для горизонтальних відстаней. Для обох наборів відстаней будуємо і згладжуємо частотний графік. У експериментах було випробувано медіанний фільтр та згладжування рухомим середнім, і рухоме середнє виявилось кращим вибором для даної задачі.

Згідно зі спостереженнями, в багатьох випадках на такому графіку буде два очевидних "пагорба": лівий відповідає за відстані всередині матричного елемента, правий - за відстані між різними матричними елементами. Відповідно, якщо кожен елемент матриці містить лише один штрих, то очікується лише один пагорб. Також, на графіку може бути і більше, ніж два пагорби. Тоді лівий так само відноситься до відстаней всередині матричного елемента, а всі інші - до відстаней між елементами. Рис. 4.2 демонструє графік частот відстаней для матриці з рис. 4.1.

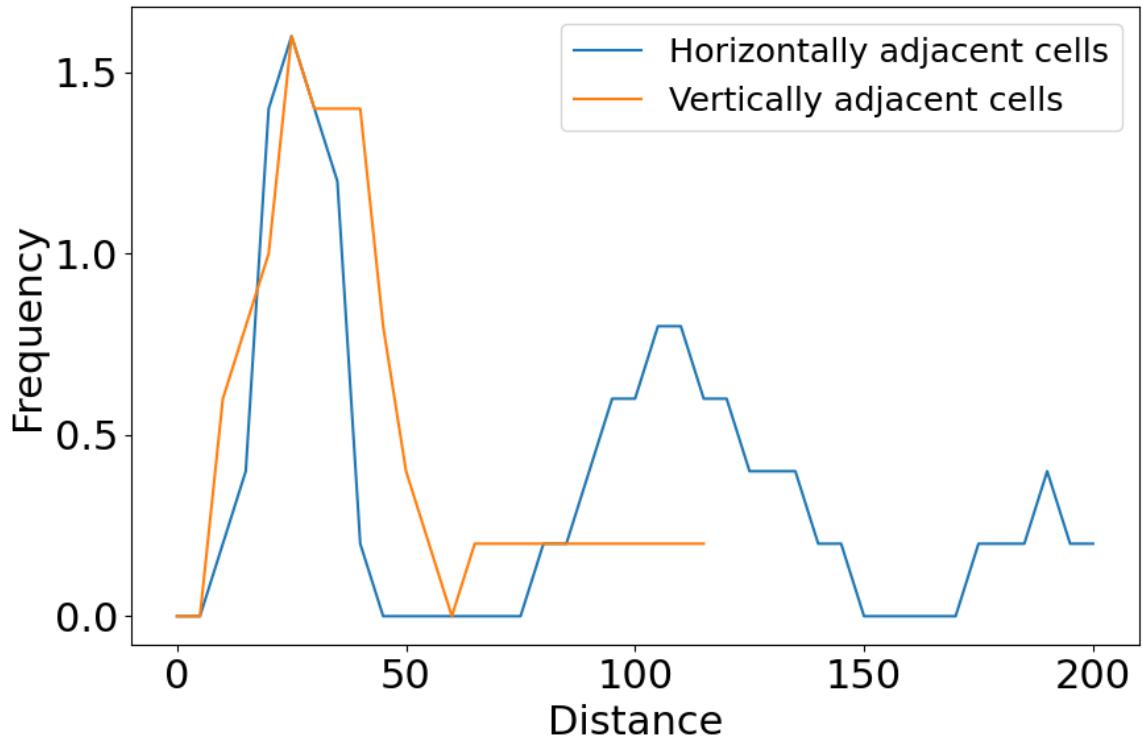


Рис. 4.2: Приклад графіка частот мінімальних відстаней для матриці

Особливим випадком при побудові графіка частот відстаней є горизонтальні штрихи, що означають знак мінус. Такі штрихи, розташовані у сусідніх по вертикалі комірках, мають мінімальну відстань значно більшу, ніж усі інші компоненти. Тому, було вирішено виключити компоненти, що утворюють горизонтальну лінію, з підрахунку графіка частот вертикальних відстаней. Варто зазначити, що горизонтальні штрихи можуть репрезентувати не тільки знак мінус, а й риску дробу або риску зверху. Однак, такі символи у виразах розташовані на невеликій вертикальній відстані від штрихів, від яких вони залежать (чисельник та знаменник дробу, символ, над яким проведено риску). Тому, вилучення їх із статистики в загальному випадку не спотворює графік частот відстаней.

На кроці 6 потрібно визначити поріг T , який розділить відстані між компонентами елемента матриці і між компонентами різних елементів матриці. У експерименті T обиралося як точка-кінець лівого пагорба.

Далі, T використовується на кроці 7 для підрахунку ваги w_{ij} для кожного ребра E_{ij} :

$$w_{ij} = \frac{\minDistance(E_{ij})}{T} \quad (4.7)$$

В ідеалі, якщо відстані всередині елемента матриці і відстані між елементами матриці можна розділити одним пороговим значенням, і якщо це порогове значення знайдено правильно, то ребра, для яких $w_{ij} < 1$ - це ребра між компонентами всередині елемента матриці, а ребра, для яких $w_{ij} \geq 1$ - ребра між компонентами з різних елементів матриці. Приклади матриць, де розділення відбулося правильно, показано на рис. 4.3.

На наступних рисунках ребро E_{ij} позначене зеленим, якщо $w_{ij} < 1$, червоним, якщо $w_{ij} \geq 1$.

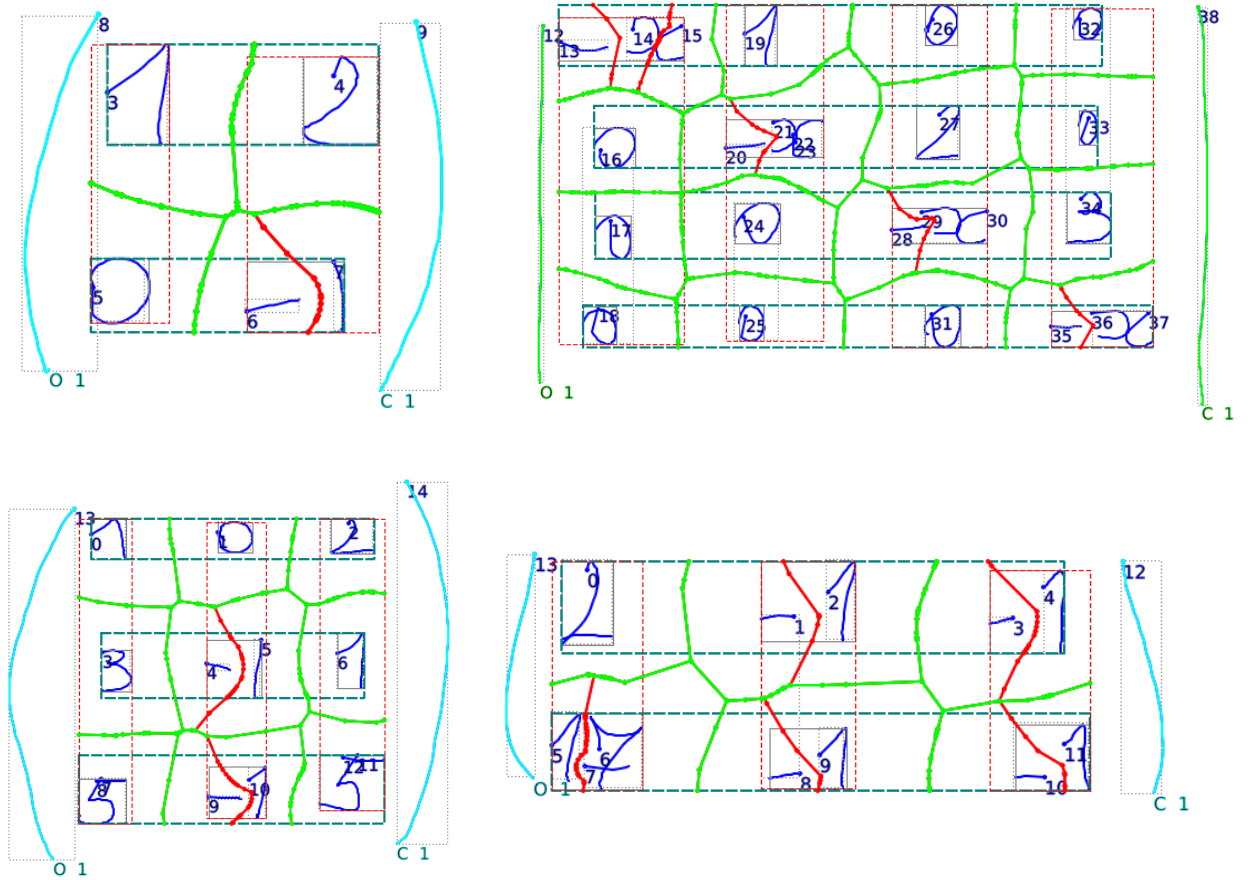


Рис. 4.3: Приклади матриць, ребра яких правильно розділені

Приклади матриць, де деякі ребра, що насправді розділяють компоненти всередині одного матричного елемента, віднесено до ребер, які розділяють різні елементи, показано на рис. 4.4. Така помилка не є критичною, тому що якщо ми не будемо розглядати червоні ребра під час побудови шляхів, алгоритм все одно зможе побудувати правильні шляхи.

Приклади матриць, де деякі ребра, що насправді розділяють різні елементи, віднесено до ребер, які розділяють компоненти всередині одного

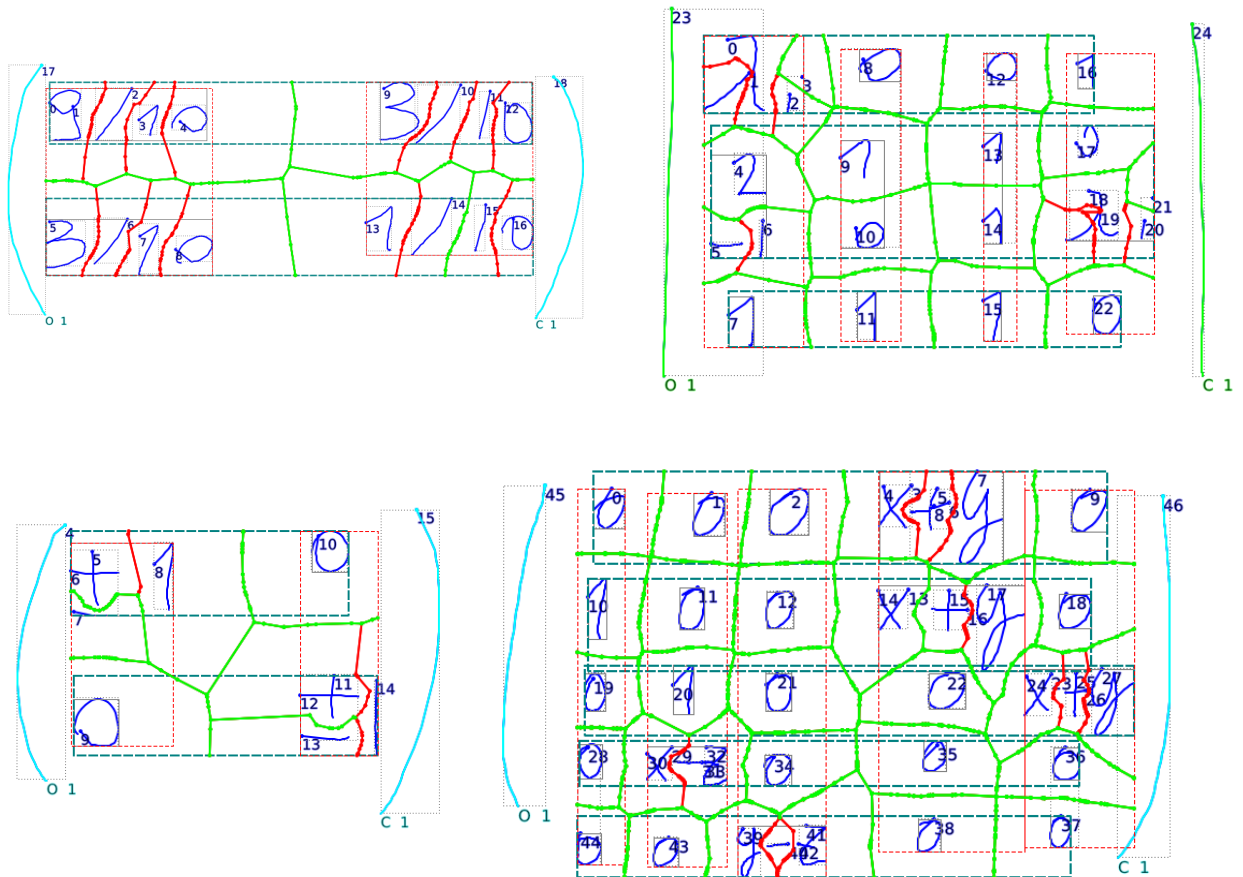


Рис. 4.4: Приклади матриць, де ребра між компонентами всередині елемента матриці помилково віднесені до ребер між різними елементами матриці

матричного елемента, показано на рис. 4.5. Ця проблема більш серйозна, тому що якщо ми не будемо використовувати червоні ребра для побудови шляхів, то побудувати правильну сегментацію не вийде взагалі.

Далі потрібно знайти шляхи з ребер, що розділяють стовпчики і шляхи, що розділяють рядки. Щоб знайти шляхи, потрібно спочатку знайти їхні початки і кінці - входи, що і відбувається на кроці 8. Ребра входи, відповідно, будуть використовуватися, як перші та останні ребра в шляхах. Обираються вони таким чином:

1. Ребра, один з кінців яких не приєднаний до жодного іншого ребра, і координати якого співпадають з лівою обмежувальною границею матриці, розглядаються як початок горизонтального шляху.
2. Ребра, один з кінців яких не приєднаний до жодного іншого ребра, і координати якого співпадають з правою обмежувальною границею матриці, розглядаються як кінець горизонтального шляху.

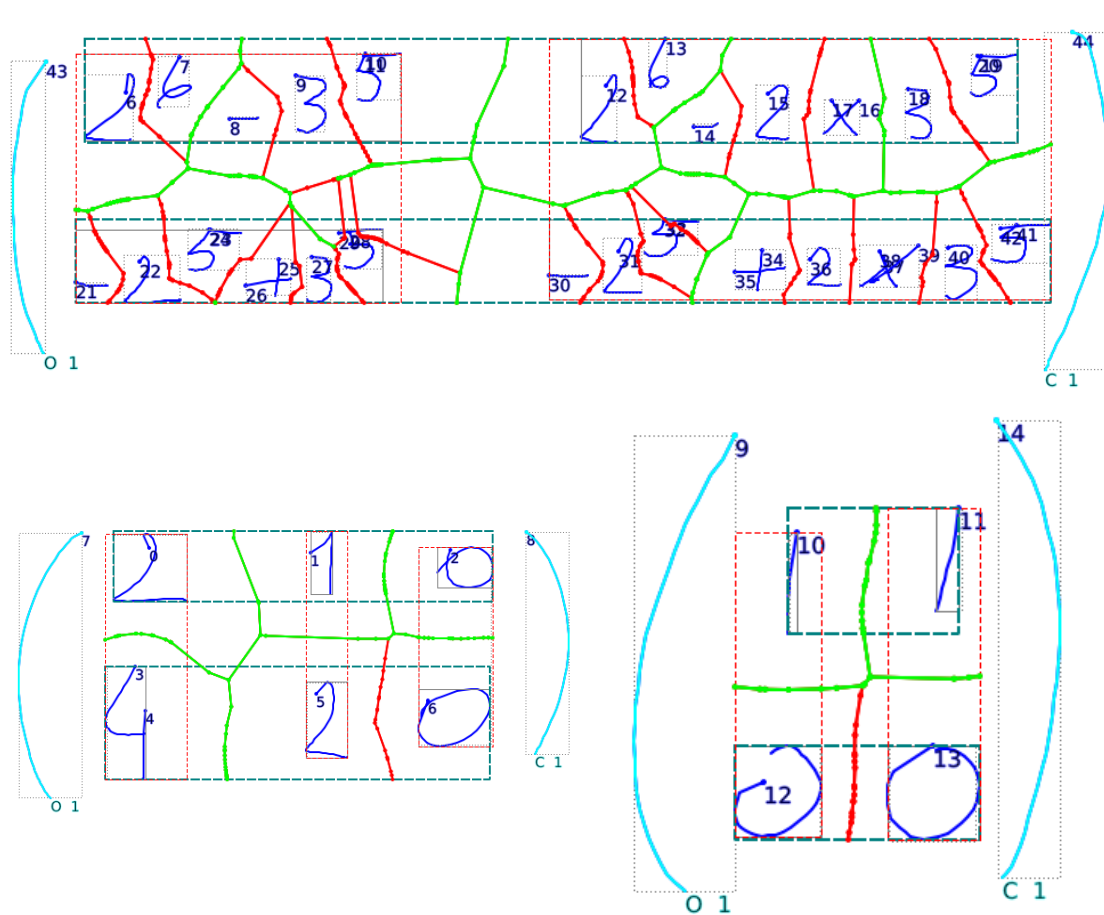


Рис. 4.5: Приклади матриць, де ребра між різними елементами матриці помилково віднесені до ребер між компонентами всередині елемента матриці

3. Ребра, один з кінців яких не приєднаний до жодного іншого ребра, і координати якого співпадають з верхньою обмежувальною границею матриці, розглядаються як початок вертикального шляху.
4. Ребра, один з кінців яких не приєднаний до жодного іншого ребра, і координати якого співпадають з нижньою обмежувальною границею матриці, розглядаються як кінець вертикального шляху.

Для вибору входів і виходів використовуються ті ребра, які визначені ребрами між різними матричними елементами, тобто, вага яких $w_{ij} \geq 1$. Якщо кількості перших і останніх ребер відрізняються, можуть бути додані і ребра з вагою $t < w_{ij} < 1$, де t - це мінімальна допустима вага для додаткових ребер входів. t визначається евристично, і в експериментах до даної роботи $t = 0.8$. Однак, навіть після додавання додаткових ребер, кількість входів і виходів все ще може не співпадати. Для внутрішніх ребер

шляху можуть використовуватися ребра, що $w_{ij} \geq t_{inner}$, де $t_{inner} < 1$ - певний поріг, що трохи пом'якшує чітке розділення на червоні та зелені ребра. Для таких ребер, що $t_{inner} \leq w_{ij} < 1$, можна додати додатковий штраф при підрахунку довжини шляху.

Для знаходження найкращих варіантів шляхів запропоновано Алгоритм 1.

Алгоритм приймає на вхід список можливих ребер-початків шляхів f_i, \dots, f_n та список можливих ребер-кінців шляхів l_1, \dots, l_m . Оскільки кількість входів і виходів на цьому етапі може бути різною, алгоритм формує кандидатів шляхів. Кандидати шляхів будуються рекурсивно, із суміжних ребер, від ребра-початку до ребра-кінця. Таким чином зазвичай для кожної пари початку і кінця можна знайти декілька варіантів шляху. Обирається як кандидат лише один з мінімальною довжиною L , яка визначається наступним чином

$$L = \sum_{p=1}^N \frac{\text{length}(E_p)}{\min(1, aw_p)} \quad (4.8)$$

де a - це коефіцієнт, що визначає вклад ваги ребра, ребро E_p - це таке ребро E_{ij} , що знаходиться в шляху на місці p за порядком, w_p - вага даного ребра. У експериментах було обрано $a = 0.6$.

Кожна можлива комбінація початкових і кінцевих ребер застосовується та оцінюється, і обирається та, що має найменше значення S . Наприклад, припустимо, що в певній матриці маємо два вхідних лівих ребра (f_1, f_2) і три вихідних правих ребра (l_1, l_2, l_3) . Можливі комбінації в такому випадку: $(f_1 - l_1, f_2 - l_2)$, $(f_1 - l_1, f_2 - l_3)$ та $(f_1 - l_2, f_2 - l_3)$. Щоб обрати, яка комбінація найбільш вдала, для кожної комбінації розраховуються значення S

$$S = \sum_{k=1}^N L_k \quad (4.9)$$

де L_k - це довжина найкоротшого шляху для пари входів p_k ($first_i, last_j$).

Алгоритм застосовується двічі: для пошуку вертикальних шляхів та для пошуку горизонтальних шляхів. Коли найкращі комбінації шляхів знайдено, горизонтальні використовуються як розділення між рядками матриці, а вертикальні - як розділення між стовпчиками матриці. На кроці

Algorithm 1 Знаходить найкращу комбінацію розділюючих шляхів

```
1: procedure FINDPATHS( $F[f_1, \dots, f_n], L[l_1, \dots, l_m], weights$ )
2:    $pathsNum \leftarrow \min(n, m)$ 
3:    $combinations \leftarrow \text{GenCombinations}(pathsNum, \max(n, m))$ 
4:   for  $c$  in  $combinations$  do
5:     SortAscending( $c$ )
6:   end for
7:    $bestCombination \leftarrow []$ 
8:    $minS \leftarrow \text{MAXFLOAT}$ 
9:   for  $c$  in  $combinations$  do
10:     $combinationPaths \leftarrow []$ 
11:     $S \leftarrow 0$ 
12:    for  $i$  from 1 to  $pathsNum$  do
13:      if  $n \geq m$  then
14:         $paths \leftarrow \text{FindPaths}(F[c[i]], L[i])$ 
15:      else
16:         $paths \leftarrow \text{FindPaths}(L[i], F[c[i]])$ 
17:      end if
18:       $minL \leftarrow \text{MAXFLOAT}$ 
19:      for  $path$  in  $paths$  do
20:         $L = \text{CalcWeightedLength}(path, weights)$ 
21:        if  $L < minL$  then
22:           $minL \leftarrow L$ 
23:           $bestPath \leftarrow path$ 
24:        end if
25:      end for
26:       $combinationPaths.append(bestPath)$ 
27:       $S \leftarrow S + minL$ 
28:    end for
29:    if  $S < minS$  then
30:       $minS \leftarrow S$ 
31:       $bestCombination \leftarrow combinationPaths$ 
32:    end if
33:  end for
34:  return  $bestCombination$ 
35: end procedure
```

10 на основі цих шляхів формується сегментація матриці. Рис. 4.6 показує приклади знаходження шляхів. Горизонтальні шляхи позначено чорним кольором, а вертикальні - блакитним.

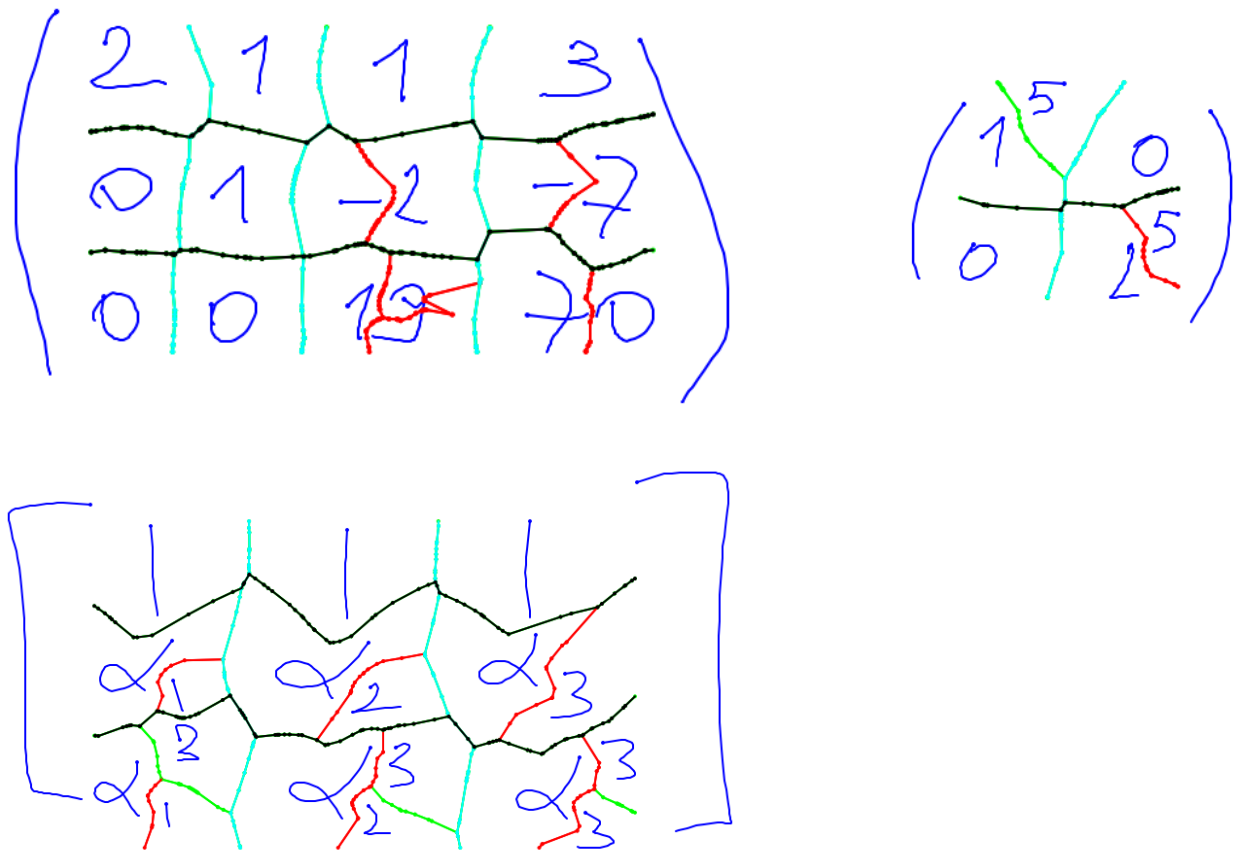


Рис. 4.6: Приклади результатів знаходження розділюючих вертикальних і горизонтальних шляхів

На рисунках бачимо, що, не дивлячись на те, що частина ребер, що ділять штрихи всередині однієї матричної комірки, позначена зеленим, шляхи побудовані правильно. Зелений колір ребра означає, що воно було розглянене під час побудови шляхів. Проте, завдяки оцінці довжин шляхів, такі шляхи не були обрані в результаті.

Розділ 5

ОЦІНКА ЗАПРОПОНОВАНИХ МЕТОДІВ

5.1 Результати експериментів

Для оцінки запропонованих методів використано відкриті датасети математичних виразів CROHME 2014 [23] тестовий матричний датасет та CROHME 2016 [24] тестовий матричний датасет. Датасети CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions) публікуються в рамках одноіменного змагання з розпізнавання рукописних математичних формул. Порівняння відбувається з опублікованими в підсумках змагань результатами [23, 24]. CROHME 2014 тестовий матричний датасет містить 123 вирази і 175 матриць, CROHME 2016 тестовий матричний датасет містить 250 виразів і 363 матриці. Для оцінки саме сегментації припускається, що виявлення матриць відбулося повністю правильно, і на вхід реалізованого метода подаються штрихи, що належать кожній матриці. Міра повноти (recall rate) підраховується комірок, рядків та стовпчиків. Рядок, стовпчик чи комірка вважаються правильними, якщо всі штрихи співпадають з тими, що визначені в основній правді (ground truth). Формула для підрахунку повноти рядків:

$$Row Recall = \frac{Correct Rows}{Total Rows} \quad (5.1)$$

Формула для підрахунку повноти стовпчиків:

$$Column Recall = \frac{Correct Columns}{Total Columns} \quad (5.2)$$

Формула для підрахунку повноти елементів матриці:

$$Cell Recall = \frac{Correct Cells}{Total Cells} \quad (5.3)$$

Табл. 5.1: Результати сегментації на CROHME 2014

	Діаграма Вороного	Метод проєкцій	System 1	System 3
Cell Recall Rate	81.58	92.00	55.35	71.07
Row Recall Rate	94.58	95.52	70.59	92.00
Column Recall Rate	83.13	92.53	50.84	69.16
Segmentation Success Rate	78.28	88.57	–	–

Табл. 5.2: Результати сегментації на CROHME 2016

	Діаграма Вороного	Метод проєкцій	MyScript	Wiris
Cell Recall Rate	78.83	94.08	87.49	84.68
Row Recall Rate	85.74	95.92	95.61	87.16
Column Recall Rate	88.32	97.71	90.71	82.22
Segmentation Success Rate	78.17	94.19	–	–

Також, введено більш узагальнену метрику для оцінки сегментації матриці:

$$Success\ Rate = \frac{\sum_{i=1}^N f(m_i)}{N}, \quad (5.4)$$

$$f(m) = \begin{cases} 1, & \text{each stroke is in proper matrix cell} \\ 0, & \text{otherwise} \end{cases}$$

Результати наведено у табл.5.1 і табл.5.2

5.2 Часова ефективність

Проведено заміри часу роботи алгоритму на матричних датасетах CROHME 2014 та CROHME 2016 (табл. 5.3). Тестування відбувалося на комп'ютері з процесором Intel Core i7-2700K CPU, 3.50GHz. Програма для окремого тестування сегментації матриць запускала в один потік.

Табл. 5.3: Результати замірів часу роботи методів

	Метод проєкцій		Діаграма Вороного	
	Час усього, мс	Час на матрицю, мс	Час усього, мс	Час на матрицю, мс
CROHME 2014	479	2.737	1235	7.057
CROHME 2016	924	2.545	3152	8.683

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x+0 & 0 \\ 0 & 0 & 0 & t+0 & 0 & 0 \\ m+0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 2 & 0 \\ t+1 & 1/2 \end{pmatrix}$$

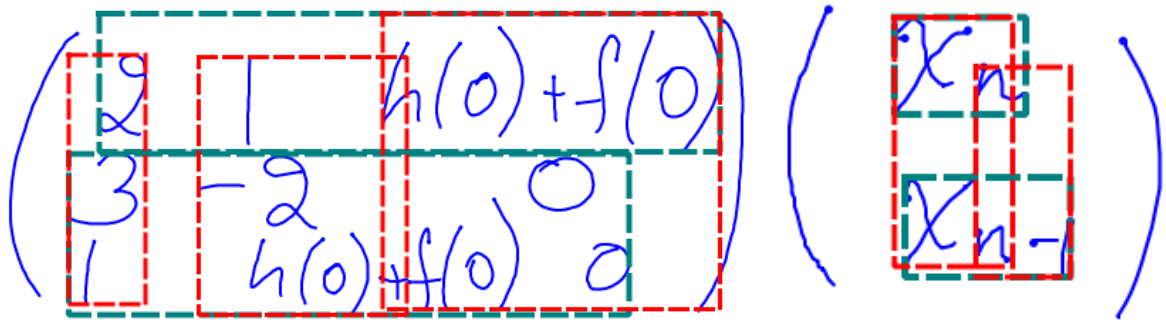
$$\begin{pmatrix} 3-x & 0 & 0 \\ 1 & 3-x & 0 \\ 0 & 0 & 4-x \end{pmatrix} \quad \begin{array}{ccc|c} 0 & 1 & 0 & f(x) \\ 1 & 0 & 1 & 0 \\ g(x) & 1 & 0 & 1 \\ 0 & h(x) & 1 & 0 \end{array}$$

Рис. 5.1: Приклади коректно відсегментованих матриць модифікованим методом профайлінгу проєкцій

5.3 Обговорення результатів сегментації модифікованим методом профайлінгу проєкцій

Загалом метод продемонстрував хороші результати, і справився з більшістю рукописних матриць. Приклади деяких матриць, які були сегментовані правильно, показано на рис. 5.1.

Досліджуючи роботу запропонованого метода, було виявлено ряд обмежень в ньому. По-перше, виявлення чітких “пагорбів” на проєкціях сильно ускладнюється, якщо обмежувальні прямокутники різних рядків або стовпчиків сильно перетинаються. Ця проблема частково усувається запропонованим методом завдяки тому, що за основу береться інформація про проєкції тільки на одній вісі, ті, що більш чіткі. Однак, зустрічаються матриці, де перетинаються і рядки, і стовпчики, і тоді метод може спрацювати помилково, як на рис. 5.2а. Інша ситуація, де трапляються помилки сегментації - це матриці, в яких більшість рядків (стовпчиків) мають велику кількість штрихів, а деякі - значно меншу. В такому випадку проєкції для таких рядків (стовпчиків) можуть бути більше схожими не на “пагорби”, а на “долини”, і не будуть виявлені, як сегменти. Метод також потребує



(а) Помилкова сегментація, спричинена перетинами рядків і стовпчиків (б) Помилкова сегментація у матриці розмірності $n \times 1$

Рис. 5.2: Приклади помилок сегментації модифікованим методом профайлінгу проекцій

покращень у сегментації матриць-рядків та матриць-стовпчиків. Проблема в тому, що в проекціях таких матриць часто є розділюючі “долини”, але важко визначити, чи вони справжні, бо в матрицях більшого розміру алгоритм відштовхується від найдовшої “долини”, а в таких матрицях зробити це неможливо. Приклад проблемної сегментації для матриці-стовпчика показано на рис. 5.2б. Оскільки така проблема найчастіше виникала саме з верхніми і нижніми індексами, додатково було додана пост-обробка, яка включала в себе геометричну перевірку на такі випадки.

5.4 Майбутній розвиток сегментації модифікованим методом профайлінгу проекцій

Даний метод використовує тільки геометричну інформацію, тому не може вирішити ситуації, коли правильну сегментацію неможливо однозначно визначити з геометрії, але можна з використанням семантичної інформації. Тому, даний метод можна використати як чорнову сегментацію, і вже після розпізнавання виразів уточнити її на основі семантичної інформації. Важливою є подальша робота над методом для підтримки матриць з прощинними елементами та пропущеними елементами.

5.5 Обговорення результатів сегментації з використанням площинної діаграми Вороного

Метод надає правильні результати для матриць, де відстані між рядками та стовпчиками є достатніми, в матрицях, що акуратно написані. Одна з основних переваг цього метода є здатність правильно відсегментувати матриці зі скошеним написанням, такі, як показано на рис. 5.3. Модифікований метод проєкцій, навпаки, погано справляється з дуже скошеними матрицями. Іншою сильною стороною метода на основі діаграми Вороного є те, що на відміну від модифікованого методу проєкцій, він зазвичай правильно сегментує матриці, де більшість рядків чи стовпчиків містять велику кількість штрихів, а деякі - значно меншу. Це пояснюється тим, що даний метод базується не на інтенсивності вираження рядків і стовпчиків, а на відстанях між сусідніми штрихами.

Основний і найбільш очевидний недолік використання площинної діаграми Вороного є робота методу в тому випадку, коли штрихи з різних елементів перетинаються. В такому випадку штрихи будуть об'єднані в одну компоненту зв'язності ще на початковому етапі роботи методу. Приклад такої матриці можна побачити на рис. 5.4. Перетин штрихів, однак, також означає перетин обмежувальних прямокутників, а, отже, такі матриці можуть становити певну складність і для модифікованого методу профайлінгу проєкцій, хоча все одно можуть бути ним правильно сегментовані. Також, такі матриці можуть бути проблемою для методів, що групують штрихи або символи в елементи матриці на основі перетину обмежувальних прямокутників. Іншим проблемним випадком є матриці, де деякі штрихи одного матричного елемента знаходяться на більшій відстані від штрихів цього самого елемента, ніж від штрихів іншого елемента. Зазвичай таке написання може зустрітися у випадку елементів з верхніми та нижніми індексами. Ще одне обмеження методу - матриці, площинна діаграма Вороного для котрих будується таким чином, що два або більше різні шляхи мають починатися з одного й того ж ребра. В цій ситуації неочевидно, як визначити кількість роздільних ліній. Приклад такої матриці показано на рис.5.5. Колонки в ній відділені правильно, а горизонтальний розділюючий шлях знайдено всього один через те, що зліва і справа є тільки одне вхідне

ребро.

5.6 Майбутній розвиток сегментації з використанням площинної діаграми Вороного

Основним напрямком подальшої роботи є покращення методу шляхом усунення основних недоліків. Обійти ситуацію, коли різні шляхи починаються з одного ребра можна, зробивши певний попередній аналіз очікуваної структури матриці, а саме знайти найбільш ймовірну кількість рядків і стовпчиків. Справитися з проблемою далеко написаних верхніх та нижніх індексів можна спробувати, аналізуючи просторові відношення між штрихами і їх відносні розміри. Також, можна розробити більш точний спосіб знаходження порогового значення T , або пошуку найкращого шляху. Запропонований метод можна випробувати і на інших табличних структурах, наприклад, на рукописних таблицях.

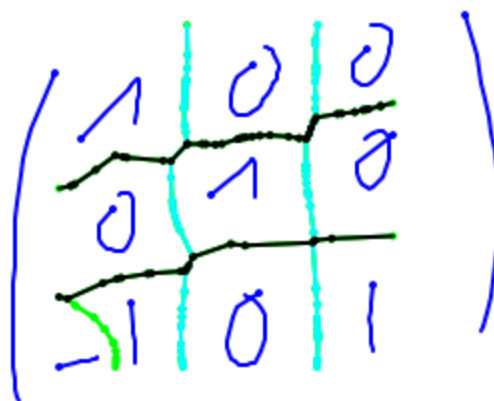


Рис. 5.3: Приклад скошено написаної матриці

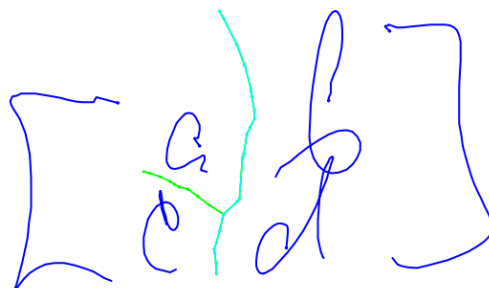


Рис. 5.4: Приклад матриці, де штрихи з різних елементів перетинаються

ВИСНОВКИ

В результаті роботи розроблено два життєздатних метода сегментації онлайн рукописних математичних матриць: модифікований метод профайлінгу проєкцій та метод сегментації на основі площинних діаграм Вороного. Модифікований метод профайлінгу проєкцій обійшов інші опубліковані результати за всіма метриками. Також, метод показав високу часову ефективність, тому підходить для інтеграції у систему розпізнавання на смартфоні. Метод на основі діаграм Вороного має високу точність для скошено написаних матриць, однак загалом потребує подальших поліпшень.

Обидва методи успішно інтегровані у систему онлайн розпізнавання рукописних математичних виразів. Однак, оскільки ефективність модифікованого методу профайлінгу проєкцій помітно вища, саме він працює в системі за замовчуванням. Інтеграція дозволила додати в систему підтримку розпізнавання матриць. На рис 5.6. показано роботу системи на смартфоні у випадку розпізнавання матриці.

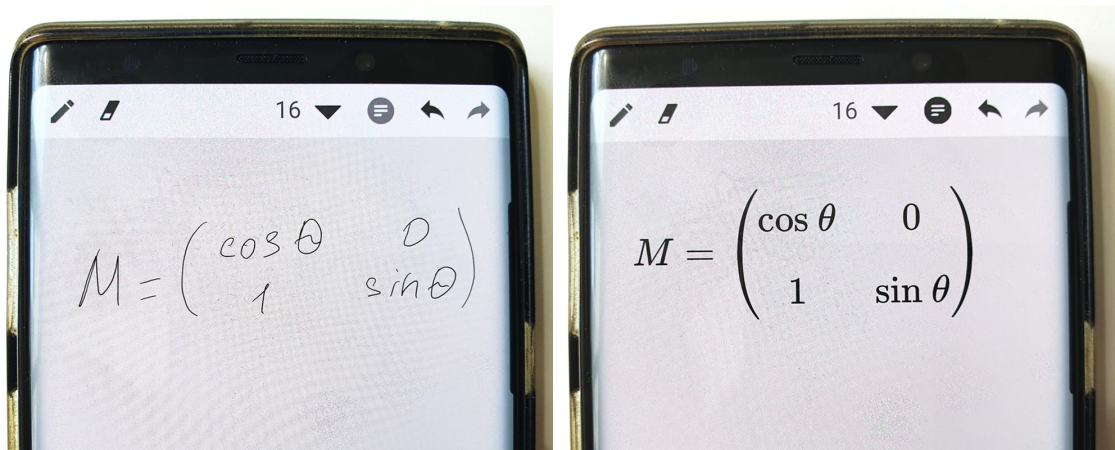


Рис. 5.6: Приклад роботи системи після інтеграції модуля сегментації матриць. Зліва - введення користувача, справа - результат сегментації та розпізнавання

Розроблені методи можуть також застосовуватися до інших табличних структур.

Подальші дослідження за темою можуть бути спрямовані на удосконалення методів для досягнення більш точної сегментації та для часової

оптимізації. Також, необхідно розробити логіку підтримки матриць зі скороченою формою запису: з пропущеними елементами та площинними елементами.

БІБЛІОГРАФІЯ

- [1] Microsoft math solver, 2019. URL <https://math.microsoft.com/>. Accessed: 2019-12-08.
- [2] Myscript calculator 2, 2019. URL <https://www.myscript.com/calculator>. Accessed: 2019-12-08.
- [3] Nebo, 2019. URL <https://www.nebo.app/>. Accessed: 2019-12-14.
- [4] Google play, 2021. URL <https://www.play.google.com>. Accessed: 2021-04-22.
- [5] Mudit Agrawal and David Doermann. Voronoi++: A dynamic page segmentation approach based on Voronoi and docstrum features. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1011–1015. IEEE, 2009.
- [6] A. Antonacopoulos and D. Karatzas. Document image analysis for world war ii personal records. In *First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings.*, pages 336–341, 2004. doi: 10.1109/DIAL.2004.1263263.
- [7] A. Awal, H. Mouchère, and C. Viard-Gaudin. A global learning approach for an online handwritten mathematical expression recognition system. *Pattern Recognit. Lett.*, 35:68–77, 2014.
- [8] Barbara Beeton, Asmus Freytag, and Murray Sargent III. Unicode support for mathematics. Technical report, The Unicode Consortium, 2017.
- [9] Jared N. Bott. *The WOZ Recognizer: A Tool For Understanding User Perceptions of Sketch-Based Interfaces*. PhD thesis, University of Central Florida, 2016. URL <https://stars.library.ucf.edu/etd/4884>.
- [10] M. Celik and B. Yanikoglu. Probabilistic mathematical formula recognition using a 2d context-free graph grammar. In *2011 International Conference*

- on Document Analysis and Recognition*, pages 161–166, 2011. doi: 10.1109/ICDAR.2011.41.
- [11] Rodolfo P dos Santos, Gabriela S Clemente, Tsang Ing Ren, and George DC Cavalcanti. Text line segmentation based on morphology and histogram projection. In *2009 10th International Conference on Document Analysis and Recognition*, pages 651–655. IEEE, 2009.
- [12] Alex Graves. Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 61–93. Springer, 2012.
- [13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL <https://doi.org/10.1145/1143844.1143891>.
- [14] Jiseong Gu and Geehyuk Lee. In-place-ink: Toward more direct handwriting interfaces. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces, ISS '16*, page 67–76, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342483. doi: 10.1145/2992154.2992184. URL <https://doi.org/10.1145/2992154.2992184>.
- [15] Jaekyu Ha, Robert Haralick, and Ihsin Phillips. Document page decomposition by the bounding-box project. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 1119–1122, 01 1995. doi: 10.1109/ICDAR.1995.602115.
- [16] T. Kanahori and M. Suzuki. Detection of matrices and segmentation of matrix elements in scanned images of scientific documents. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 433–437 vol.1, 2003. doi: 10.1109/ICDAR.2003.1227704.

- [17] Koichi Kise, Akinori Sato, and Motoi Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.
- [18] Andreas Kosmala and Gerhard Rigoll. On-line handwritten formula recognition using statistical methods. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, volume 2, pages 1306–1308. IEEE, 1998.
- [19] Dmytro Kotsur and Vasyl Tereshchenko. Voronoi-based skeletonization algorithm for segmenting the network of biological neurons. 04 2019. doi: 10.15588/1607-3274-2019-1-10.
- [20] C. Li, R. Zeleznik, T. Miller, and J. J. LaViola. Online recognition of handwritten mathematical expressions with support for matrices. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008.
- [21] Marcus Liwicki, Alex Graves, Santiago Fernández, Horst Bunke, and Jürgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.
- [22] Nicholas Elias Matsakis. *Recognition of handwritten mathematical expressions*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [23] Harold Mouchere, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 791–796. IEEE, 2014.
- [24] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 607–612. IEEE, 2016.

- [25] Francisco Álvaro Muñoz. *Mathematical Expression Recognition based on Probabilistic Grammars*. PhD thesis, Universitat Politècnica de València, 2015. URL <https://pdfs.semanticscholar.org/5605/a2d4196c8388a0d4bbf03cfbb3f6d847cee4.pdf>.
- [26] L. O’Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993. doi: 10.1109/34.244677.
- [27] Masayuki Okamoto. Recognition of mathematical expressions by using the layout structure of symbols. In *Proc. 1st Int. Conf. Document Analysis and Recognition, 1991*, pages 242–250, 1991.
- [28] Roman Ptak, Bartosz Zygadło, and Olgierd Unold. Projection-based text line segmentation with a variable threshold. *International Journal of Applied Mathematics and Computer Science*, 27(1):195–206, 2017.
- [29] K. Sugihara. Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. *CVGIP: Graphical Models and Image Processing*, 55(6): 522–531, 1993. ISSN 1049-9652. doi: <https://doi.org/10.1006/cgip.1993.1039>.
- [30] Ernesto Tapia and Raúl Rojas. Recognition of on-line handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance. In Josep Lladós and Young-Bin Kwon, editors, *Graphics Recognition. Recent Advances and Perspectives*, pages 329–340, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-25977-0.
- [31] David Tausky, George Labahn, Edward Lank, and Mirette Marzouk. Managing ambiguity in mathematical matrices. In *Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling, SBIM ’07*, page 115–122, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595939135. doi: 10.1145/1384429.1384454. URL <https://doi.org/10.1145/1384429.1384454>.
- [32] H-J Winkler, H Fahrner, and Manfred Lang. A soft-decision approach for structural analysis of handwritten mathematical expressions. In *1995*

International Conference on Acoustics, Speech, and Signal Processing, volume 4, pages 2459–2462. IEEE, 1995.

- [33] Ryo Yamamoto, Shinji Sako, Takuya Nishimoto, and Shigeki Sagayama. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. 10 2006.
- [34] Abderrazak Zahour, Bruno Taconet, Pascal Mercy, and Said Ramdane. Arabic hand-written text-line extraction. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 281–285. IEEE, 2001.
- [35] Jianshu Zhang, Jun Du, and Lirong Dai. A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition. pages 902–907, 11 2017. doi: 10.1109/ICDAR.2017.152.
- [36] Ting Zhang. *New Architectures for Handwritten Mathematical Expressions Recognition*. PhD thesis, 2017.
- [37] Ting Zhang, H. Mouchre, and Christian Viard-Gaudin. Using blstm for interpretation of 2-d languages: Case of handwritten mathematical expressions. 19:135–157, 12 2016. doi: 10.3166/DN.19.2-3.135-157.
- [38] Dmitry Zhelezniakov, Viktor Zaytsev, and Olga Radyvonenko. Online handwritten mathematical expression recognition and applications: A survey. *IEEE Access*, PP:1–1, 03 2021. doi: 10.1109/ACCESS.2021.3063413.
- [39] Dmytro Zhelezniakov, Viktor Zaytsev, and Olga Radyvonenko. Acceleration of online recognition of 2d sequences using deep bidirectional lstm and dynamic programming. In *International Work-Conference on Artificial Neural Networks*, pages 438–449. Springer, 2019.
- [40] Dmytro Zhelezniakov, Anastasiia Cherneha, Viktor Zaytsev, Tetiana Ignatova, Olga Radyvonenko, and Oleg Yakovchuk. Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page

212–220, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371186. doi: 10.1145/3377325.3377482. URL <https://doi.org/10.1145/3377325.3377482>.

- [41] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognition*, 51:135–147, 2016. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2015.09.013>. URL <https://www.sciencedirect.com/science/article/pii/S0031320315003441>.