

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу
Кафедра системного проектування**

До захисту допущено:

Завідувач кафедри

_____ Вадим МУХІН

«__» _____ 2022 р.

**Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Інтелектуальні сервіс-орієнтовані розподілені обчислювання»
зі спеціальності 122 «Комп'ютерні науки»
на тему: «Дослідження та розробка методів сегментації для задачі
розпізнавання рукописного тексту»**

Виконав:

студент ІV курсу, групи ДА-82

Васильюк Данило Ігорович _____

Керівник:

ас.

Яковчук Олег Констянтинович _____

Консультант з економіки:

д.е.н., доцент

Роцина Н. В. _____

Рецензент:

д.т.н., професор

Аушева Н. М. _____

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

Студент _____

Київ – 2022

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра системного проектування

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Інтелектуальні сервіс-орієнтовані розподілені обчислювання»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Вадим МУХІН

«__» _____ 2022 р.

ЗАВДАННЯ

на дипломну роботу студенту

Василюку Данилу Ігоровичу

1. Тема роботи «Дослідження та розробка методів сегментації для задачі розпізнавання рукописного тексту», керівник роботи Яковчук Олег Константинович, ас., затверджена наказом по університету від «__» _____ 20__ р. № _____
2. Термін подання студентом роботи – 10 червня 2022 р.
3. Вихідні дані до роботи: наукові публікації з методами сегментації рукописного тексту, відкритий програмний код реалізацій методів сегментації на платформі GitHub, датасет із зображеннями рукописного тексту “Computer Vision Lab-Database”, мова програмування Python 3.10, бібліотека функцій та алгоритмів комп'ютерного зору та обробки зображень OpenCV, бібліотека високорівневих математичних функцій numpy.

4. Зміст роботи:

- ознайомитись з проблемою сегментації рукописного тексту в сучасних інформаційних системах;
- дослідити існуючі методи та підходи для сегментації об'єктів, рукописного та нерукописного тексту;
- запропонувати та розробити алгоритм сегментації, реалізувати його програмно;
- провести тестування власної реалізації та існуючих методів на датасеті CVL. Провести оптимізацію параметрів алгоритму / моделі;
- проаналізувати отримані результати, визначити переваги та недоліки власного рішення, подальші напрямки роботи.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

- Презентація до захисту роботи

6. Консультанти розділів роботи^{1*}

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Роцина Н. В., професор, д.е.н.		

7. Дата видачі завдання 01.02.2022 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.02.2022	
2	Збір інформації	30.03.2022	
3	Підготовка теоретичної частини роботи	30.04.2022	
4	Дослідження алгоритмів, існуючих підходів	11.05.2022	
5	Розробка алгоритму сегментації	21.05.2022	
6	Тестування алгоритму, підбір параметрів	30.05.2022	
7	Аналіз отриманих результатів	05.06.2022	
8	Оформлення дипломної роботи	09.06.2022	
9	Отримання допуску до захисту та подача роботи в ДЕК	10.06.2022	

Студент
Керівник

Д.І. Василюк
О.К. Яковчук

^{1*} Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломної роботи.

АНОТАЦІЯ

бакалаврської дипломної роботи Василюка Данила Ігоровича на тему
“Дослідження та розробка методів сегментації для задачі розпізнавання
рукописного тексту”

Метою роботи є дослідження методів сегментації та створення алгоритму для отримання точної та швидкої сегментації зображення з рукописним текстом. Актуальність теми дослідження обумовлена відсутністю ідеального підходу сегментації, який стабільно працював на великій вибірці зображень, не залежачи від факторів картинки. В роботі проаналізовано наступні методи сегментації: методи аналізу різності яскравостей, нарощування областей і текстурні методи. Розроблено алгоритм сегментації картинок з рукописним текстом на окремі текстові лінії та окремі слова для подальшого використання отриманих сегментів у задачі розпізнавання. Було проведено налаштування параметрів для точнішої сегментації на заданій вибірці картинок, показано, як працює алгоритм і які результати він видає. В результаті досліджень було розроблено та налагоджено алгоритм і порівняно його зі вже існуючими, а підсумками стали ґрунтовні висновки щодо основних переваг і недоліків розробленого алгоритму та наступні кроки у подальшій його розробці.

Загальний обсяг роботи: 74 сторінок, 33 ілюстрації, 9 таблиць, 25 посилання.

Ключові слова: сегментація рукописного тексту, задача розпізнавання рукописного тексту, розробка алгоритму, обробка цифрових зображень, метод аналізу різності яскравостей, коефіцієнт Жаккара.

ABSTRACT

for a bachelor thesis written by Vasyliuk Danylo Igorovich on “Research and development of segmentation methods for handwriting recognition tasks”

The aim of the work is to research the methods of segmentation and create an algorithm to obtain accurate and fast segmentation of images with handwritten text. The relevance of the research topic is due to the lack of an ideal segmentation approach, which has worked steadily on a large sample of images, regardless of the factors of the picture. The following methods of segmentation are analyzed in the work: methods brightness differences analysis, increasing areas method and texture methods. An algorithm for segmenting pictures with handwritten text into separate text lines and individual words has been developed for further use of the obtained segments in the recognition problem. The parameters were adjusted for more accurate segmentation on a given sample of images, it is shown how the algorithm works and what results it gives. As a result of research, an algorithm was developed and adjusted, compared with existing ones, and the results were thorough conclusions about the main advantages and disadvantages of the developed algorithm and the next steps in its further development.

Total amount of research: 74 pages, 33 figures, 9 tables, 25 references.

Keywords: handwriting segmentation, handwriting recognition problem, algorithm development, digital image processing, brightness difference analysis method, Jacquard coefficient

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
1 ДОСЛІДЖЕННЯ МЕТОДІВ СЕГМЕНТАЦІЇ	11
1.1 Класифікація	13
1.2 Загальна математична модель сегментації	15
1.3 Аналіз існуючих методів сегментації зображень	17
1.3.1 Методи аналізу різності яскравостей.	17
1.3.1.1 Граничний метод	17
1.3.1.2 Виділення меж	18
1.3.1.3 Аналіз кольору	18
1.3.2 Методи нарощування областей	18
1.3.2.1 Центроїдне зв'язування	18
1.3.2.2 Хвильовий метод	19
1.3.2.3 Метод водорозділів	19
1.3.2.4 Злиття-розщеплення	20
1.3.3 Текстурні методи	20
1.3.3.1 Матриці збігів	20
1.3.3.2 Метод мозаїки Вороного	21
1.4 Огляд існуючих рішень	22
1.4.1 Amazon.....	22
1.4.2 Google	24
1.4.3 Apple	25
1.5 Висновки	26
2 РОЗРОБКА АЛГОРИТМУ	28
2.1 Аналіз документа та теорія масштабного простору.....	28
2.2 Попередня обробка	29
2.3 Сегментація сторінки документу	29
2.3.1 Алгоритм Кенні	29
2.4 Сегментація текстових ліній.....	30
2.5 Аналіз плями	32

2.5.1 Неоднорідні фільтри Гауса.....	32
2.6 Вибір масштабу.....	34
2.6.1 Вибір η	36
2.6.2 Вибір σ_y	37
2.7 Вилучення та постобробка плями	38
2.8 Висновки	38
3 АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПАРАМЕТРІВ АЛГОРИТМУ	40
3.1 Опис тестових даних.....	40
3.2 Опис оцінки алгоритму.....	41
3.3 Опис програмної реалізації.....	42
3.4 Аналіз результатів	44
3.5 Тестування та підбір параметрів	46
3.6 Приклади роботи алгоритму з довільними зображеннями	50
3.7 Порівняння з існуючими рішеннями	52
3.8 Висновки	53
4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ	55
4.1 Постановка задачі техніко-функціонального аналізу.....	55
4.2 Обґрунтування функцій та параметрів програмного продукту	56
4.2.1 Варіанти реалізації основних функцій	56
4.3 Обґрунтування системи параметрів ПП.....	58
4.3.1 Опис параметрів	58
4.3.2 Кількісна оцінка параметрів	59
4.3.3 Аналіз експертного оцінювання параметрів	60
4.4 Аналіз рівня якості варіантів реалізації функцій	63
4.5 Економічний аналіз варіантів розробки ПП.....	64
4.6 Висновки	68
ВИСНОВКИ	70
ПЕРЕЛІК ПОСИЛАНЬ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

RGB (скорочено від англ. Red, Green, Blue – червоний, зелений, синій)	адаптивна колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори
Датасет (англ. dataset)	набір даних
Пляма (англ. blob)	область у цифровому зображенні, яка вирізняється за властивостями, такими як яскравість або колір, порівняно з навколишніми областями
XML (англ. Extensible Markup Language)	розширювана мова розмітки, стандарт побудови мов розмітки ієрархічно структурованих даних
CNN (англ. Convolutional Neural Network)	згорткова нейронна мережа
Detr (англ. Detection Transformer)	архітектура, що використовує CNN для обробки плоского зображення

ВСТУП

У наш час розпізнавання образів знаходить все більше застосування у повсякденному житті. Розпізнавання звуків і тексту значно спрощує взаємодію людини з комп'ютером, розпізнавання друкованого тексту використовується для перекладу документів на електронну форму. Це в свою чергу набуває значного використання в таких сферах як медицина, наука, військова справа тощо. На сьогоднішній день велика частина інформації зберігається у електронному виді. Пошук та отримання необхідних знань відбуваються набагато простіше завдяки напівавтоматичним системам навігації по різних корпусах тестів, зображень та відео. Для навігації за рукописними документами необхідно виділити об'єкти, присутні на зображенні, такі як текст, математичні формули, зображення. Таким чином, однією з задач комп'ютерного зору є задача розпізнавання рукописного тексту. Цій задачі передують сегментація відсканованого документа. Саме сегментація дозволяє визначити необхідні об'єкти на зображенні, які далі будуть передаватись для їх збереження чи розпізнавання.

На даний момент вже існують різні алгоритми сегментації, але деякі з них неактуальні через свою неточність, ще частина доволі повільна, інші – доволі складні в розробці та аналізі. Отже, актуальність теми дослідження обумовлена відсутністю ідеального алгоритму для сегментації зображення.

Метою роботи є створення алгоритму сегментації, який давав би прийнятну якість і швидкодію для сегментації зображень. Сегментація в першу чергу повинна розділяти вихідне зображення на тло, текст та замальовки. Для досягнення поставленої мети необхідно розв'язати такі задачі:

- 1) проаналізувати сучасні методи сегментації зображень;
- 2) розробити та відлагодити алгоритм сегментації, визначивши параметри, що дадуть найбільшу точність;
- 3) розробити критерій для оцінки якості.

Об'єктом дослідження є процес визначення рукописного тексту на зображенні. Предметом дослідження є алгоритм сегментації.

Наукова новизна дипломної роботи полягає в тому, що було запропоновано та розроблено алгоритм, який досягає доволі високої швидкодії, а також найкращої точності для сегментації зображення з найбільшим відсотком співпадінь з реальними даними на тестовій вибірці.

1 ДОСЛІДЖЕННЯ МЕТОДІВ СЕГМЕНТАЦІЇ

Процес кластеризації зображення, тобто пошуку в ньому однорідних областей, називається сегментацією. Сегментація, при якій області розбиття не перетинаються, називається тесе́ляцією. Її ціль полягає в спрощенні або зміні представлення зображень, щоб його було легко аналізувати в подальшому, а результатом є безліч сегментів, які показують все зображення. Інакше кажучи, кожен піксель відмічений деякою міткою деякого класу.



Рисунок 1.1 – Результат сегментації зображення [22]

Сегментація зображення документа на його основні сутності, а саме, рядки і слова, розглядається як нетривіальна задача для вирішення розпізнавання рукописних документів. Труднощі, які виникають в рукописних документах роблять процедуру сегментації складним завданням. Різні види труднощів зустрічаються в сегментації текстових рядків і процедурі сегментації слів. У випадку процедури сегментації по лініям основні труднощі включають різницю в куті перекосу між рядками на сторінці або навіть уздовж одного текстового рядка, перекриваючи слова та суміжні рядки іншим текстом. Крім того, часта поява

акцентів у багатьох мовах (наприклад, у французькій та грецькій), розділових знаків, нерівномірний інтервал слів також ускладнює сегментацію. Оскільки сегментація це процес поділу зображення на області, кожна з яких має містити пікселі, які однорідні за певною ознакою і мають спільні візуальні характеристики, однорідність можна визначити, аналізуючи різні якісні показники, такі як яскравість, текстура, колір, міри близькості об'єктів на зображенні, а також по глибині, якщо мова йде про аналіз тривимірного простору [2].

He said these concerned Mr. Weaver's
 alleged association with organizations black-
 listed by the government. Immediately
 Mr. Kennedy rushed a letter to Senator
 Robertson saying the Federal Bureau of In-
 vestigation had reported on Mr. Weaver.
 He believed he would perform "outstanding
 service" in his post. Senator Robertson's
 committee has to pass Mr. Weaver's
 nomination before it can be con-
 sidered voted by the full Senate.

Рисунок 1.2 – Результат сегментації зображення, що містить рукописний текст

[22]

Донедавна ідентифікація та сегментація не були пов'язані між собою, тому їх вважають окремими етапами обробки зображення. Така точка зору зараз вважається необґрунтованою, оскільки відображення зображення у вигляді набору окремих сегментів завжди вирішує конкретну проблему. Більше того,

якщо ми зможемо точно сформулювати те, що нам потрібно розпізнати, ми зможемо визначити, який вигляд матиме сегментація. Тому ефективність універсальних методів сегментації часто можна значно підсилити за рахунок використання важливої для конкретної проблеми апріорної інформації [1].

Для сегментації зображень було розроблено досить багато універсальних алгоритмів і методів. Для корисності результатів ці методи необхідно поєднати зі спеціалізованими знаннями з предметної області, щоб ефективно розв'язати часткову проблему теми.

1.1 Класифікація

Дуже часто методи сегментації розділяють на ті, що виділяють області однорідних яскравості та кольору, та ті, які визначають однорідність деяких властивостей, найчастіше – текстури. Методи першого типу та їх можливості відомі доволі гарно. В публікаціях останніх років розглядаються здебільшого методи другого типу.

Також важливим критерієм є можливість класифікувати методи сегментації. Вони можуть бути як задані заздалегідь (такі як бібліотека текстур, еталонів тощо), так і не задані, які необхідно отримати в процесі роботи алгоритму сегментації [3].

Грубою область називається, коли вона може бути охарактеризовано як «об'єкт» або «фон». Така область часто використовується при незаданих заздалегідь характеристиках. В іншому випадку, коли таку область можна охарактеризувати більшою кількістю термінів, область називається розмальовкою і вона використовується при відомих характеристиках. Їх також можна поєднувати при сегментації: спочатку використовувати грубу, а після багатозначну.

Якщо відомі еталони об'єктів, застосовують кореляційні методи, які є ефективними в системах прикладного телебачення. Часто їх використовують в

області розпізнавань зображень. При існуванні постійних відмінностей в параметрах областей зображень використовують порогові методи. Відмінними від порогових методів є методи нарощення областей, адже вони особливо ефективні коли присутній стійкий зв'язок всередині більшості сегментів. При чітких та стабільних границях використовується метод виділення границь.

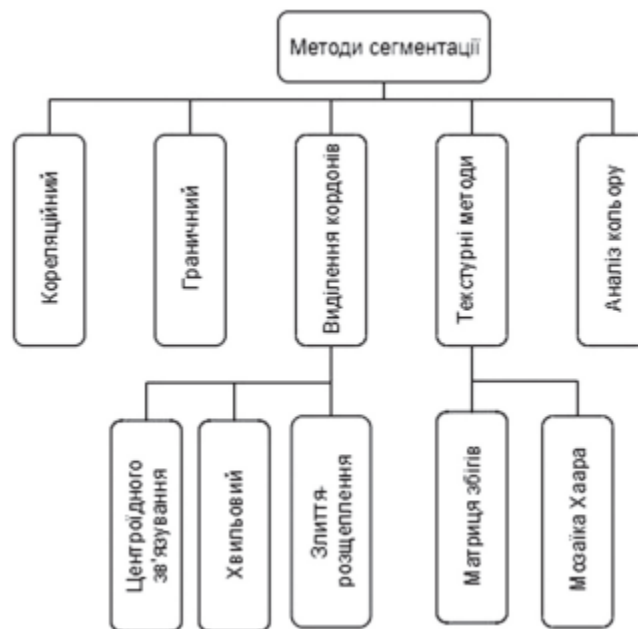


Рисунок 1.3 – Класифікація методів сегментації зображень [7]

Для опису і сегментації зображень (однорідності, шорсткості, регулярності) використовують текстурні методи, які умовно діляться на дві категорії: статичні та структурні. Прикладом статичного підходу є використання матриць збігів; структурного – мозаїка Вороного.

Залежно від технічних засобів отримання функції освітлення і принципу обробки розрізняють групи кольорових зображень, а значення функцій задаються в кольоровому форматі, зазвичай з використанням формату RGB. Також зображення можуть бути напівколірними, в кожній точці має одне значення – рівень сірого. Бінарні зображення – це зображення, які представлені двома

категоріями – контурами та виділеними кольоровими точками, які називаються зрізами. Для сегментації можна використовувати набір координат окремих точок, що визначають об'єкт або поверхню в комп'ютерній графіці, і цей спосіб введення називається векторним. Щоб сформулювати задачу сегментації, розглянемо функцію зображення, яка визначає розподіл світла у фіксованому полі зору. Цифрова обробка зображень вимагає просторової вибірки та квантування послідовних значень зображення шляхом переведення координат елементів у дискретні значення та розрахунку яскравості цих елементів з фіксованою кількістю рівнів.

1.2 Загальна математична модель сегментації

Нехай $D(n \times m)$ – це чисельне поле зору, растр, на якому задано зображення $V(i, j)$, при чому $D_k \subset D$ – область k -го об'єкту, де $k = 1, 2, \dots, s$. Інформація про зображення звичайно вводиться та зберігається у вигляді матриці, що задана на чисельному дискретному полі зору D . Вхідною інформацією будемо вважати матрицю V розміром $n \times m$, де n та m – число рядків та стовпців дискретного поля зору D . $D_1 \cup D_2 \cup \dots \cup D_s \cup D_\phi = D, D_i \cap D_j = \emptyset, i \neq j$.

Зображення $V(i, j)$ реальної сцени являє собою сукупність зображень окремих об'єктів та фону. Представимо його у вигляді:

$$V(i, j) = H_1(i, j) + \dots + H_s(i, j) + H_\phi(i, j) \quad (1.1)$$

де s – кількість об'єктів сцени; $H_k(i, j)$ – зображення k -го об'єкта, $k = 1, \dots, s$;

$H_\phi(i, j)$ – зображення фону;

Функції H приймають нульове значення, коли значення (i, j) не належить області k -го об'єкта. В полі зору об'єкти не повинні перетинатися між собою. Сусідніми називають дві точки растра D із координатами (p, q) та (i, j) . якщо для кожної пари точок $e_{k1}, e_{k2} \in E$ у множині E знайдеться така послідовність точок $e_{k1}, \dots, e_{k3}, \dots, e_{kn}, \dots, e_{k2} \in E$, то така множина називається 4-зв'язною. Кожна пара таких точок є сусідніми. Кожна 4-зв'язна множина точок одночасно є 8-зв'язною,

але не навпаки. Відповідно цій інформації про зв'язність точок можна розбити задачу сегментації на частини. Першою підзадачею стає груба сегментація, а саме побудова функції характеристик з метою об'єднання областей сегментованих об'єктів.

$$\mu(i, j) = \begin{cases} 1, \text{ якщо } (i, j) \in D_1 \cup \dots \cup D_s \\ 0, \text{ якщо } (i, j) \in D_\phi \end{cases} \quad (1.2)$$

Ця функція відповідає за виділення фону.

Другою підзадачею є зафарбування – створення розмітки для бінарного зображення, яке є комплексним зображенням деяких об'єктів. Така сегментація виконується з меншою долею евристики, ніж груба. Поняття «об'єкт» для бінарного зображення уже можна вважати строго формалізованим. Областю об'єкта є множина точок растру (i, j) , для яких виконується умова $\mu(i, j) = 1$, ця множина зв'язна і не знаходиться всередині іншої множини з більшою кількістю точок [8].

Таке розподілення використовується за умови, що об'єкти та їх точки не конфліктують та не пересікаються. В інакшому разі відображення картинки у вигляді растру виконується в залежності від її границі. Якщо $E \subset D$ – деяка область, то її точка $(i, j) \in D$ називається граничною, якщо не всі її сусіди відносно принципу зв'язності знаходяться в області E [8]. Можна вважати відомою і саму область E , тільки тоді, коли вона є зв'язною і відома її границя. Для цього потрібно спочатку скористатися одним із методів комп'ютерної графіки відновлення набору пікселів за границею зображення. Для цього будується зображення:

$$\begin{aligned} \pi: D &\rightarrow \{0, 1, \dots, s\}, \\ \pi(i, j) &= \begin{cases} 1, \text{ якщо } (i, j) \in \delta D_1, 1 \in \{1, \dots, s\} \\ 0, \text{ якщо } (i, j) \notin \delta D_1 \cup \dots \cup \delta D_s \end{cases} \end{aligned} \quad (1.3)$$

1.3 Аналіз існуючих методів сегментації зображень

У алгоритмах сегментації такі параметри як надійність і достовірність обробки залежать від того, наскільки враховано такі характеристики: розподіл яскравості зображення на всьому зображенні, включаючи його фон та окремі ділянки, форма об'єктів зображення та кількість перепадів яскравості.

Основним методом сегментації є процес визначення однорідних ділянок і окреслення об'єктів зображення. Сьогодні існує багато комбінацій методів, крім використання додаткових знань про форму, структуру, просторове розташування та характеристики місцевості. Таким чином, методи можна розділити на групи, кожна група сегментується за допомогою однієї характеристики зображення, а сам алгоритм поділяється на методи, що використовують порогове значення, текстуру, масштабування, злиття зрізів і кластерний аналіз [5].

1.3.1 Методи аналізу різності яскравостей.

1.3.1.1 Граничний метод

Нехай дано зображення $B(i, j)$, $s=1$ (один об'єкт), яскравість точок котрих є в межах $[T_1, T_2]$, а яскравості точок фону в цей проміжок не входять. Якщо $B(i, j) \in [T_1, T_2]$, то припускаємо, що точка (i, j) належить області об'єкта, в інакшому разі – області фону. У випадку $s > 1$ мають бути відомі відрізки $[T_1^k, T_2^k]$, яскравості k -х об'єктів знаходяться в межах даних відрізків. Ці відрізки не повинні перетинатись. Розмітка точок здійснюється шляхом відображення (1.3).

Проблема полягає у визначенні граничного значення. Для цього використовується аналіз гістограм яскравостей. Для випадку з одним об'єктом ($s = 1$) на гістограмі має бути два максимуми. Граничне значення обирається між цими двома верхніми межами (максимумами). На практиці для побудови та аналізу гістограм використовуються більш складні методи.

1.3.1.2 Виділення меж

За допомогою методу сегментації об'єкти передаються їх межами. Граничними точками вважаються точки різких перепадів функції освітлення. Чисельне диференціювання використовується для знаходження граничних точок [6]. Найпоширенішим є градієнтний метод, також відомий як метод контрастування або просторового диференціювання [4]. Використовуючи фільтр на зображенні, можливо отримати зображення градієнтів. Від вихідного воно відрізняється підкресленими перепадами яскравості. Точка (i, j) належить до контуру у випадку, коли яскравість зображення градієнтів перевищує деяке граничне значення, яке можна визначити за гистограмою.

1.3.1.3 Аналіз кольору

Сегментація за допомогою аналізу кольору заснована на його впізнанні. Властивостями є, до прикладу, три компоненти (координати) від функції випромінювання $B(\lambda)$ в кожній точці (i, j) : $C_1 = \int_{\lambda} B(\lambda)K_1(\lambda)d\lambda$, $1=1, 2, 3$. Спектральні криві чутливості $K_1(\lambda)$, $K_2(\lambda)$, $K_3(\lambda)$ можуть відповідати функціям трьох видів колбочок людського ока, але на практиці використовується багато інших систем кодування кольорових компонент, у т.ч. двох- та чотирьохкомпонентні системи. Сегментація може відбуватися даними методами – покомпонентно, а впізнання – шляхом дешифрації значень координат кольору [5].

1.3.2 Методи нарощування областей

1.3.2.1 Центроїдне зв'язування

Щоб скористатися цим методом, потрібна апріорна інформація про об'єкт, а саме, одна або декілька початкових точок. Задаються початкові точки a_1, \dots, a_k , яким відповідно зазначаються мітки $\lambda_1, \dots, \lambda_k$. Точки, позначені λ_i , утворюють множину S_i [7]. Після вибору початкових точок відбувається процес розмітки, під час якого розглядаються всі точки A множин S_i . Якщо точка N , сусідня з A ,

дорівнює $|B(A)-B(N)| < T$ і не має мітки, то точку N позначають λ_i . Після розподілу на області можливе проведення злиття областей, тобто застосування точкам з мітками λ_x та λ_y єдиної мітки $\min(\lambda_x, \lambda_y)$. У будь-якому порядку позначень цей метод найкраще підходить для простих зображень (з одним об'єктом) [7]. Для більш складних зображень використовується так званий хвильовий метод перегляду точок.

1.3.2.2 Хвильовий метод

Після вибору початкових точок відбувається процес, що складається з ітерацій. На кожній з ітерацій розглядаються точки множин S_i , за винятком точок, доданих в S_i на даній ітерації. Для точки (i, j) розглядаються її сусідні точки. Якщо умова з попереднього пункту виконується, то одній з них можна призначити мітку λ_i . Після того як було виконано аналіз для всіх точок множини S_i , крім тих, що були додані на даній ітерації, здійснюється аналіз точок з S_{i+1} . Точки множини S_i , що додані на k -й ітерації, називаються фронтом $F_k(\lambda_i)$, об'єднання – хвилею [7].

1.3.2.3 Метод водорозділів

Однією з модифікацій хвильового методу є метод водорозділів. Запроваджується множина точок $S = \{i, j, B(i, j)\}$, що називається поверхнею. На поверхні вводять поняття шляху від точки S_m до точки S_n .

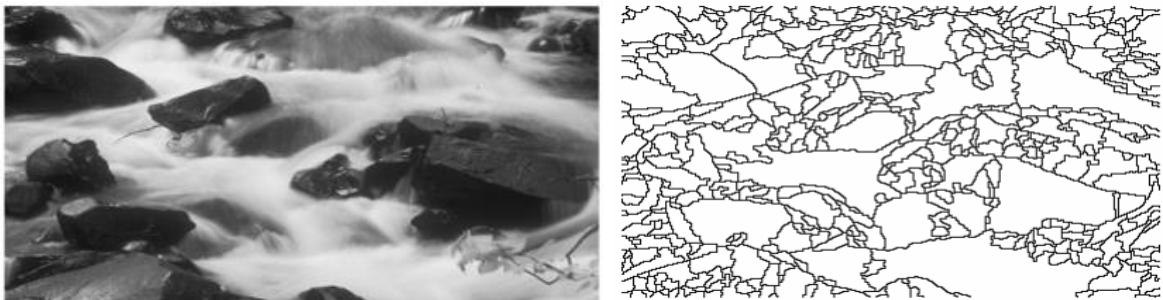


Рисунок 1.4 – Результат сегментації методом водорозділів [24]

Послідовність $\{S_m, S_{m+1}, \dots, S_{n-1}, S_n\}$, де S_i є сусідньою до S_{i+1} , називається шляхом. Шляхом, що не зростає, називається така послідовність $\{S_i\}$, що $\forall s_m(i_m, j_m, V(i_m, j_m)), s_n(i_n, j_n, V(i_n, j_n))$: $m \geq n \leftrightarrow V(i_m, j_m) \leq V(i_n, j_n)$.

Якщо не існує незростаючого шляху з початком в точці s , тоді точка $s \in S$ називається локальним мінімумом. Наступним кроком є так званий процес заповнення басейнів. Треба знайти $V = \min_{i,j}(V(i, j))$. В циклі на кожній ітерації збільшується яскравість V на один, поки не буде досягнута максимальна яскравість. Під час кожної ітерації методом хвиль здійснюється розмітка точок на зображенні. В кінці кінців зображення розподіляється на так звані басейни. Іноді також виділяють точки водорозділу. Такі точки мають сусідів, що належать до двох і більше басейнів. Залежно від завдань, яскравість V можна збільшувати не до максимального значення V_{\max} , а до будь-якого граничного значення [7].

1.3.2.4 Злиття-розщеплення

Метод полягає в тому, щоб розділити зображення на квадрати певним чином. Потім здійснюється аналіз однорідності цих квадратів, здебільшого аналізується однорідність яскравостей. Якщо чотири сусідні квадрати задовольняють одній умові однорідності, вони об'єднуються в один спільний. Якщо навпаки – квадрат не задовольняє умові, він ділиться на чотири підквадрати. Результатом даного метода є структура даних з інформацією про ці об'єкти. Найчастіше такою структурою даних є граф, але інколи це може бути і саме зображення, поділене на квадрати, в межах яких всі пікселі мають однакову яскравість.

1.3.3 Текстурні методи

1.3.3.1 Матриці збігів

Даний метод належить до групи статистичних методів. Обчислюючи для кожної ділянки так званої матриці збігів, цей метод дозволяє визначити, чи мають ділянки зображень текстури одного класу.

Розглядається ділянка $N \times N$. Маємо множину яскравостей $\{B(i, j), i=1, \dots, N; j=1, \dots, N\}$ с G градаціями сірого. Визначається вектор зміщення $d=(dx, dy)$. Вводиться матриця збігів $G \times G$, що позначається P_d . Елемент (a, b) матриці P_d дорівнює кількості випадків, коли від точки з яскравістю a на відстані, визначеній вектором d , знаходиться точка з яскравістю b [7]. Формально це записується так:

$$P_d(a, b) = \sum_{r,s} p(a, b, (r, s), (t, v)) \quad (1.4)$$

де $p(a, b, (r, s), (t, v)) = \begin{cases} 1, \text{ якщо } B(r, s) = a, B(t, v) = b \\ 0, \text{ в інакшому разі} \end{cases}$

На основі матриці P_d можуть бути обчислені різноманітні характеристики:

$$\sum_a \sum_b P_d^2(a, b) \text{ (енергія)}$$

$$\sum_a \sum_b P_d(a, b) \log P_d(a, b) \text{ (ентропія)}$$

1.3.3.2 Метод мозаїки Вороного

Використовуючи цей метод, сегментація складається з трьох кроків: побудова примітивів, складання мозаїки, аналіз елементів мозаїки. Зазвичай примітиви до зображення отримуються за допомогою таких фільтрів, що й при виділенні границь. Потім обираються точки локальних максимумів, до яких застосовується метод нарощування, у результаті чого здобуваються компоненти із 8-ми пов'язаних елементів.

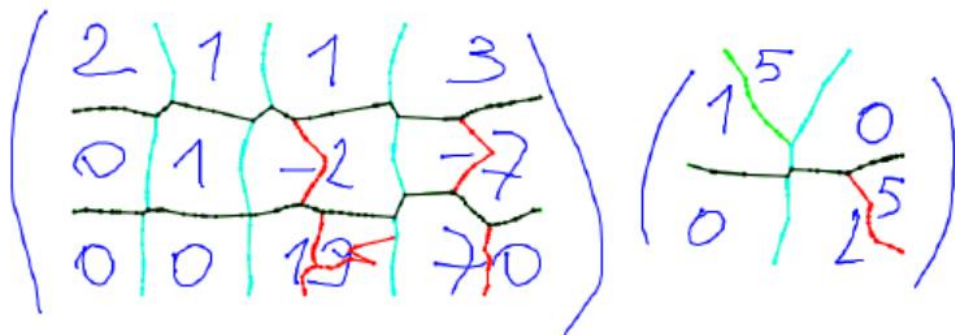


Рисунок 1.5 – Приклад правильної сегментації методом мозаїки Вороного [23]

Компоненти або точки локальних максимумів, отримані таким чином, визначають як примітиви. Потім будується мозаїчне розбиття Вороного для примітивів. Розглянемо побудову для випадку, коли примітиви є точками [7].

Нехай дана множина S з трьох і більше примітивів. Нехай не всі точки лежать на одній прямій. Розглянемо пару точок P та Q . Побудуємо пряму – геометричне місце точок, що рівновіддалені від P та Q . Отримаємо дві напівплощини H^P_Q та H^Q_P . Для будь-якої точки P можна здійснити таке розбиття з усіма $Q \in S$. Перетин визначає багатокутник, всі точки якого є ближчими до P , аніж до будь-якої іншої точки. Такий багатокутник називають багатокутником Вороного для даної точки. Розглядають множину багатокутників на діаграмі Вороного. Багатокутники із загальними властивостями об'єднують в області. Для розрахунку властивостей часто використовується центр ваги й момент площі багатокутника. Момент площі $(p + q)$ -го порядку для багатокутника відносно примітива з координатами (x, y) визначають так:

$$m_{pq} = \iint_R (x - \bar{x})^p (y - \bar{y})^q dx dy, \quad (1.5)$$

де $(p + q) = 0, 1, 2, \dots$; x, y – координати центру ваги багатокутника.

Часто використовують ознаки:

$$f_1 = m_{00}; f_2 = \sqrt{\bar{x}^2 + \bar{y}^2}; f_3 = \arctg\left(\frac{\bar{y}}{\bar{x}}\right) \quad (1.6)$$

1.4 Огляд існуючих рішень

1.4.1 Amazon

Алгоритм класифікації зображень SageMaker – це контрольований алгоритм навчання, який аналізує лише цілі зображення, класифікуючи їх за однією з кількох вихідних категорій. Алгоритм виявлення об'єктів – це контрольований алгоритм навчання, який виявляє та класифікує всі екземпляри об'єкта на зображенні. Він вказує розташування та масштаб кожного об'єкта на зображенні за допомогою прямокутної рамки.

Алгоритм семантичної сегментації SageMaker забезпечує підхід на рівні пікселів до розробки програм комп'ютерного зору. Він позначає кожен піксель на зображенні міткою класу із попередньо визначеного набору класів. Тегування є основоположним для розуміння сцен, що має вирішальне значення для все більшої кількості додатків комп'ютерного зору, таких як, наприклад, самокеровані транспортні засоби, діагностика медичних зображень і зондування роботів.

Оскільки алгоритм семантичної сегментації класифікує кожен піксель на зображенні, він також надає інформацію про форми об'єктів, що містяться в зображенні. Вихід сегментації представлений у вигляді зображення у відтінках сірого, яке називається маскою сегментації. Маска сегментації — це зображення у відтінках сірого з такою ж формою, що й вхідне зображення [11].

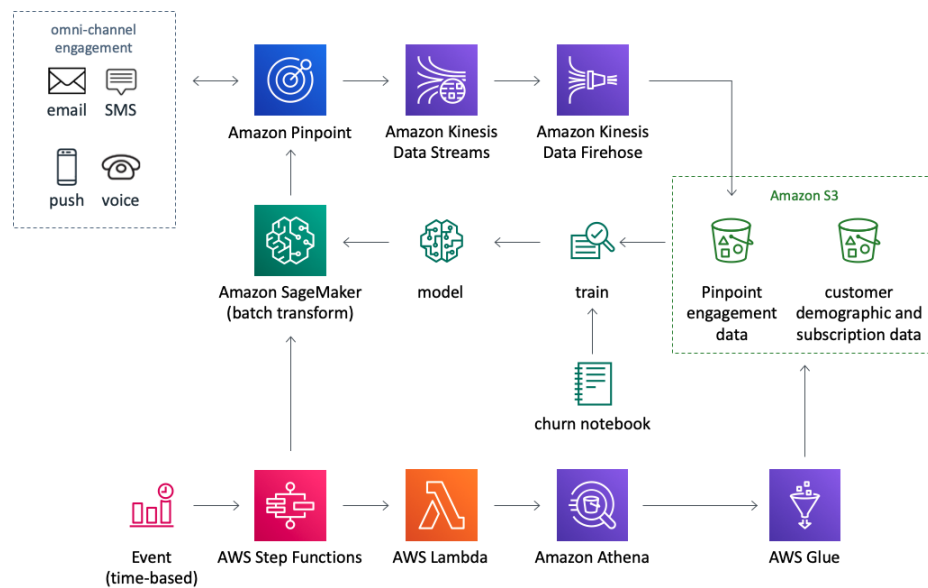


Рисунок 1.6 – Архітектура системи, яка використовує алгоритм сегментації SageMaker [11]

Алгоритм семантичної сегментації SageMaker побудований з використанням фреймворку MXNet Gluon і набору інструментів Gluon CV. Він

надає вибір із трьох вбудованих алгоритмів для навчання глибокої нейронної мережі. Можна використовувати алгоритм Fully-Convolutional Network (FCN), алгоритм Pyramid Scene Parsing (PSP) або DeepLabV3. Кожен із трьох алгоритмів складається з двох окремих компонентів:

1. Backbone (або кодер) – мережа, яка створює надійні карти активації функцій.
2. Декодер – мережа, яка створює маску сегментації із закодованих карт активації.

1.4.2 Google

Google з нуля розробив хмарні TPU для прискорення передових програм машинного навчання (ML), від розпізнавання зображень до моделювання мови та навчання з підкріпленням. Нещодавно компанія розробила хмарні TPU для сегментації зображень — процесу ідентифікації та маркування областей зображення на основі об'єктів або текстур, які вони містять — шляхом випуску високопродуктивних реалізацій TPU двох станів. Сучасні моделі сегментації, Mask R-CNN і DeepLab v3+ як відкритий вихідний код.

Mask R-CNN – це двоступенева модель сегментації екземплярів, яку можна використовувати для локалізації кількох об'єктів на зображенні до рівня пікселя. Перший етап моделі витягує ознаки (відмітні шаблони) з вхідного зображення для створення пропозицій регіону, які, ймовірно, містять об'єкти, що представляють інтерес. Другий етап уточнює та фільтрує ці пропозиції регіону, прогнозує клас кожного об'єкта високої достовірності та створює маску рівня пікселя для кожного об'єкта.

Точність під час тренування Mask R-CNN буде відрізнятись в залежності від застосування: для деяких швидкість навчання може бути головним пріоритетом, тоді як для інших визначаються пріоритети тренування з

максимально можливою точністю, навіть якщо тренування потребує більше часу і відповідні витрати необхідні для досягнення цього порогу точності [10].



Рисунок 1.7 – Результат роботи розпізнавання об’єктів алгоритмом Mask R-CNN після сегментації зображення [10]

1.4.3 Apple

Камера (в iOS та iPadOS) для створення зображень покладається на широкий спектр технологій для розуміння сцени. Зокрема, розуміння вмісту зображень на рівні пікселів, також відоме як сегментація зображень. Сегментація людей та оцінка глибини дають змогу портретному режиму, який імітує такі ефекти, як мала глибина різкості та сценічне освітлення. Сегментація осіб і шкіри забезпечує семантичне відтворення в групових знімках до чотирьох осіб, оптимізуючи контраст, освітлення та навіть відтінки шкіри для кожного об’єкта окремо. Фотографічні стилі створюють персональний вигляд фотографій, вибірково застосовуючи коригування потрібних ділянок за допомогою масок сегментації, зберігаючи відтінки шкіри. Сегментація неба та сегментація шкіри потужні алгоритми зменшення шуму та різкості для кращої якості зображення в областях з низькою текстурою.

Apple вибрали архітектуру Detection Transformer (DETR) як базову для її бажаного дизайну детектора.

По-перше, DETR не вимагає постобробки, як більшість архітектур. Зокрема, DETR уникає необхідності немаксимального стиску (NMS) для видалення повторюваних передбачень і декодування координат на основі прив'язки. NMS непотрібний, завдяки неявній дедуплікації Transformer декодера у цій моделі. Декодування позиції на основі прив'язки не потрібно, завдяки глобальній (не відносної) регресії координат [9].

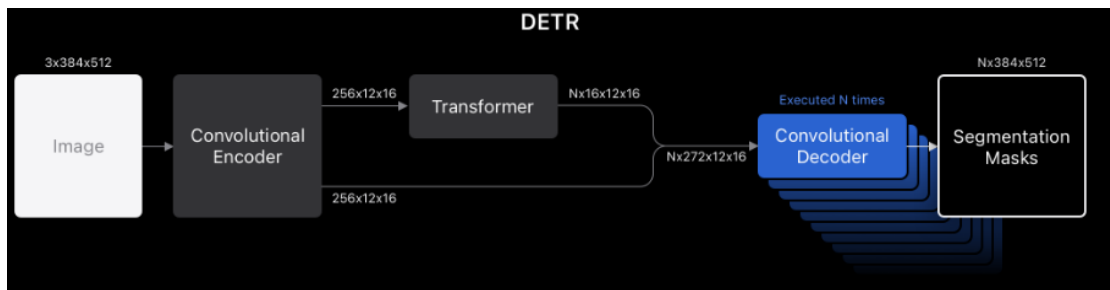


Рисунок 1.8 – Архітектура моделі DETR сегментації компанії Apple [9]

По-друге, DETR є високоефективним при оцінці регіонів інтересу (RoIs). Двоетапні підходи, такі як Mask R-CNN, оцінюють тисячі прив'язних RoI, перш ніж пересилати сотні пропозицій з найвищим рейтингом на другий етап. Замість цього обмежується кількість RoI в оригінальній моделі DETR на порядок величини (з конфігурації за замовчуванням 100), і отримується незначне погіршення ефективності виявлення для цільового розподілу зображень (<5 осіб на сцені).

1.5 Висновки

У цьому розділі було розглянуто основні методи сегментації зображень, проаналізовано, які процеси відбуваються на кожному кроці кожного метода. Досліджено важливість процесу сегментації в задачі розпізнавання об'єктів та способи застосування різних алгоритмів в складних системах.

Сегментація поки не є до кінця алгоритмізованим процесом для довільних зображень. Дуже часто виникає ряд проблем, такі як складний фон зображення, відмінність положення зору між еталонним зображенням та вхідним, а також відмінності в їх освітленні, кольоровій гамі, кількості випадкових перешкод тощо. Еталонні та вхідні зображення також часто відрізняються геометричними перетвореннями, різними проєкціями та кутами зображень. Тому існує ціла низка алгоритмів, кожен з яких використовується при різних умовах.

На сьогодні найпоширенішими методами сегментації є методи, які використовують визначення однорідності в яскравості, кольорів, текстур зображень. Також часто використовують методи, засновані на нейронних мережах, але вони залежать від натренованості моделі і не завжди можуть впоратися з нетривіальним зображенням. Для сегментації зображень на текстові компоненти, часто користуються методами аналізу кольору та границь, але для їх ефективної роботи зображення має бути належним чином відскановано та містити малу кількість дефектів.

Для обрання найефективнішого методу сегментації виникає необхідність проведення попереднього аналізу даних, але така можливість надається вкрай рідко. Тому актуальною стає потреба у створенні нескладного в реалізації алгоритму, який показує свою ефективність на різних типах вхідних даних і не залежить від попередньо визначеної моделі.

2 РОЗРОБКА АЛГОРИТМУ

2.1 Аналіз документа та теорія масштабного простору

Теорія масштабного простору розглядає важливість масштабу в будь-якому фізичному спостереженні, тобто об'єкти та ознаки мають значення лише в певних масштабах. У масштабному просторі, починаючи з вихідного зображення, генеруються послідовно згладжені зображення вздовж розміру масштабу. Декількома дослідниками було показано, що функція Гауса однозначно генерує простір лінійного масштабу зображення, коли накладаються певні умови [13].

Вважається, що масштабний простір також забезпечує ідеальну основу для аналізу документів, тобто документ складається з об'єктів у кількох масштабах. Інтуїтивно, у більш тонкому масштабі ми маємо символи, а в більшому — маємо слова, словосполучення, рядки та інші конструкції [14]. Тоді хотілось би мати масштаб, у якому можна виводити слова із зображення документа. Представлення простору лінійної шкали безперервного сигналу з довільними розмірами складається з побудови одного параметра. Нехай $f: R^2 \rightarrow R$ представляє будь-який заданий сигнал. Тоді масштабний простір має вигляд $I: R^2 \times R_+ \rightarrow R$ що визначається, дозволяючи представленню простору масштабу в нульовому масштабі бути рівним вихідному сигналу $I(\cdot; 0) = f$ і для $t > 0$

$$I(\cdot; t) = G(\cdot; t) \star f \quad (2.1)$$

де $t \in R_+$ параметр масштабу, G – ядро Гауса, яке в двомірному просторі записується як

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{(2\sigma^2)}} \quad (2.2)$$

де $\sigma = \sqrt{2t}$

2.2 Попередня обробка

Для сегментації зображення необхідно привести відповідне зображення до стандартизованого вигляду, а саме змінити його висоту і ширину до деякої константи, але такої, щоб подальші алгоритми не мали проблем з розділенням об'єктів між собою. Також необхідним кроком є приведення зображення у формат RGB [15].

2.3 Сегментація сторінки документу

Сегментація сторінки документу від фону зображення дозволяє прибрати зайві об'єкти з метою визначення сторінки з текстом. Сам алгоритм ділиться на наступні кроки:

1. Накладення фільтрів на зображення, таких як маска сірого, розмиття з ціллю прибрати зайві невеликі об'єкти.
2. Використання алгоритму Кенні задля виділення границь.
3. Проведення морфологічної трансформації розширення – елемент пікселя дорівнює «1», якщо принаймні один піксель під ядром дорівнює «1». Так збільшується біла область на зображенні або збільшується розмір об'єкта переднього плану. Зазвичай, у таких випадках, як видалення шуму, ерозія супроводжується розширенням.
4. Знаходження найбільш відповідного контуру сторінки.
5. Визначення координат сторінки і передача їх для подальшої сегментації ліній.

2.3.1 Алгоритм Кенні

Джон Кенні взяв за мету розробити оптимальний алгоритм виявлення границь. Цей алгоритм має задовольняти наступним критеріям:

- правильне виявлення положення границі елементу;
- підвищення відношення сигнал/шум (елемент – сигнал, задній фон – шум);

- однозначне положення границі

Основними етапами алгоритму є:

1. Накладання фільтру розмиття на зображення з метою видалення шуму.
2. Пошук градієнтів зображення (там де найбільше значення градієнту, там і границі. Оскільки вони часто мають різні напрямки, алгоритм Кенні використовує фільтри для визначення ребер в попередньо розмитому зображенні) [16].

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.3)$$

$$\theta = \arctg\left(\frac{G_y}{G_x}\right) \quad (2.4)$$

3. Подвійна порогова фільтрація.

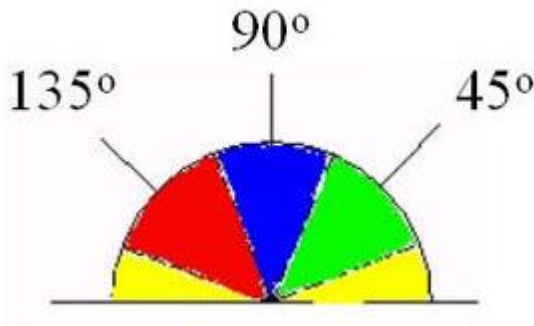


Рисунок 2.1 – Напрямок градієнту [16]

2.4 Сегментація текстових ліній

Сегментація текстових ліній дозволяє розділяти висхідні та низхідні лінії. Часто рядки складаються з ряду горизонтальних компонентів зліва направо. Методи проекційного профілю широко використовуються в сегментації рядків і слів для машинних друкованих документів. У цій техніці одновимірною функцією значень пікселів отримується шляхом проектування двійкового зображення на горизонтальну або вертикальну вісь [12]. Ми використовуємо модифіковану версію того ж алгоритму, розширену на зображення у відтінках сірого. Нехай $f(x,$

y) — це значення інтенсивності пікселя (x, y) на зображенні у шкалі сірого. Тоді ми визначаємо профіль вертикальної проекції як

$$P(y) = \sum_{x=0}^W f(x, y) \quad (2.5)$$

де W – ширина зображення.

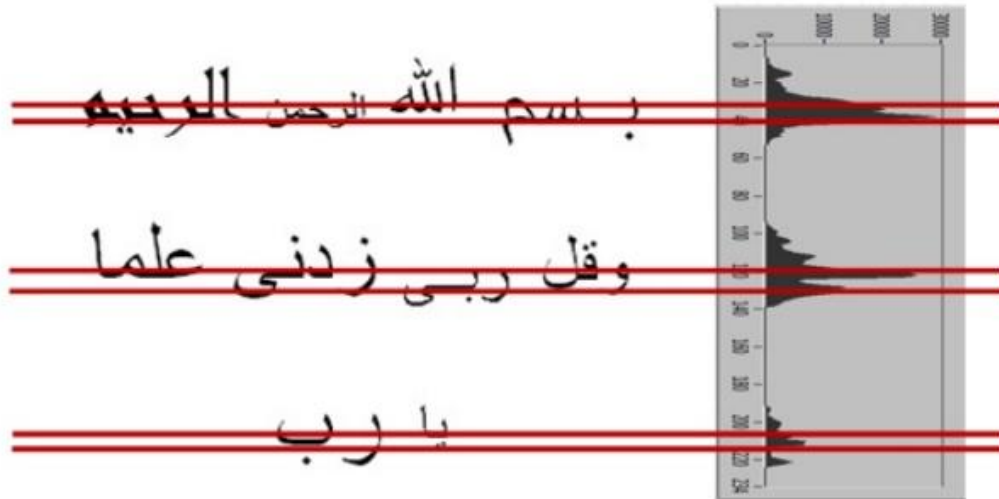


Рисунок 2.2 – Результат сегментації ліній і відповідні їм проекційні профілі [25]

Чіткі локальні піки в профілі відповідають білому простору між рядками, а окремі локальні мінімуми відповідають тексту. Таким чином, сегментація ліній передбачає виявлення положення локальних максимумів. Проте проекційний профіль має ряд хибних локальних максимумів і мінімумів [14]. Таким чином, функція проекції $P(y)$ згладжується за допомогою фільтра Гауса (низьких частот), щоб усунути помилкові тривоги та зменшити чутливість до шуму. Тоді локальні максимуми отримують з першої похідної функції проекції шляхом вирішення для y такого, що:

$$P'(y) = P(y) * G_y = 0 \quad (2.6)$$

Техніка сегментації рядків стійка до змін у розмірі рядків і була протестована на широкому діапазоні рукописних сторінок. Наступним кроком

після сегментації лінії є створення масштабного простору лінійних зображень для аналізу плями.

2.5 Аналіз плями

Тепер ми розглядаємо кожне зображення рядка окремо, щоб виділити слова. Зображення слова складається з окремих символів, пов'язаних символів або їх комбінації. Ми хотіли б об'єднати ці підрозділи в єдину значущу сутність, якою є слово. Цього можна досягти, сформувавши зображення у вигляді плями. Пляму можна розглядати як зв'язану область в просторі. Традиційним способом формування плями є використання лапласіана Гауса (LOG), оскільки LOG є популярним оператором і часто використовується для виявлення плями та різноманітних завдань масштабного аналізу зображень. Алгоритм, що розроблюється, використовує диференціальний вираз, подібний до LOG для створення масштабного представлення для виявлення плями. Однак наш диференціальний вираз відрізняється тим, що ми об'єднуємо часткові похідні Гауса другого порядку вздовж двох орієнтацій на різних масштабах.

2.5.1 Неоднорідні фільтри Гауса

Деякі властивості, які характеризують письмо, використовуються для формулювання підходу до фільтрації слів. Максимуми в просторі масштабу відбуваються в масштабі, пропорційному просторовим розмірам плями [14]. Якщо ми поспостерігаємо за словом, то можемо побачити, що його просторовий об'єм визначається наступним:

1. Окремі символи визначають висоту (вимір Y) слова
2. Довжина (вимір X) визначається кількістю символів у ньому.

Слово, як правило, містить більше одного символу і має співвідношення сторін більше одного. Оскільки розмір x слова більше розміру y , частота просторової фільтрації також має бути вищою у розмірі y порівняно з розміром x . Ці знання специфічної області дозволяють нам перейти від ізотропних

(однаковий масштаб в обох напрямках) до анізотропних операторів. Ми вибираємо, щоб масштаб розміру x був більшим за розмір y , щоб відповідати просторовій структурі слова. Визначимо анізотропний фільтр Гауса як

$$G(x, y; \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \quad (2.7)$$

Можна визначити коефіцієнт множення η як $\eta = \frac{\sigma_x}{\sigma_y}$

Середнє співвідношення сторін або коефіцієнт множення η лежить від трьох до шести для більшості рукописних десяти документів. Також анізотропний фільтр Гауса (виміряний як просторова протяжність утворених плям) є максимальним в цьому діапазоні. Для наведеного вище рівняння Гауса другий анізотропний гаусівський диференціальний оператор $L(x, y; \sigma_x, \sigma_y)$ визначається як

$$L(x, y; \sigma_x, \sigma_y) = G_{xx}(x, y; \sigma_x, \sigma_y) + G_{yy}(x, y; \sigma_x, \sigma_y) \quad (2.8)$$

Масштабне представлення лінійних зображень будується шляхом згортання зображення за допомогою (2.9). Розглянемо двовимірне зображення $f(x, y)$, тоді відповідне вихідне зображення має вигляд

$$I(x, y; \sigma_x, \sigma_y) = L(x, y; \sigma_x, \sigma_y) * f(x, y) \quad (2.9)$$

Основні ознаки, які виникають із представлення простору в масштабі, є плямами (тобто з'єднані області або світліші, або темніші за фон). Знак I може бути використаний для класифікації поверхні тривимірної інтенсивності на передній і задній план. Для прикладу розглянемо лінійне зображення на рис. 2.3(а). На малюнках показано зображення плями $I(x, y; \sigma_x, \sigma_y)$ зі збільшенням значень масштабу. На рис. 2.3(б) показано, що в меншому масштабі зображення плями складається з плями символів. Коли ми збільшуємо масштаб, плями символів породжують плями слів (рис. 2.3(в) і рис. 2.3(г)). Це свідчить про явище злиття в пляму. Видно, що для певних значень масштабу плями, а отже, і слова правильно розмежовані (рис. 2.3(г)). Подальше збільшення значення масштабу

може не обов'язково призводити до злиття плям слів разом, а також спостерігається інше явище, наприклад розщеплення. Ці цифри показують, що існує масштаб, за яким можна окреслити більшість слів.

2.6 Вибір масштабу

Аналіз масштабного простору не вирішує проблему вибору масштабу. Рішення цієї проблеми залежить від конкретного застосування та вимагає використання попередньої інформації для керівництва процедурою вибору шкали. Максимальна реакція як у масштабі, так і в просторі отримується в масштабі, пропорційному розміру об'єкта [14].

Зображення документа складається з таких структур, як символи, слова та рядки різного масштабу. Однак, у порівнянні з іншими типами зображень, зображення документів мають унікальну властивість: для вилучення конкретного типу структури не потрібні великі зміни в масштабі. Наприклад, усі слова по суті близькі один до одного за своїм масштабом і, отже, можуть бути вилучені без великої зміни параметра масштабу. Отже, існує шкала, де кожне окреме слово утворює окрему пляму. Тоді вихідний сегмент (пляма) максимальний при цьому значенні параметра масштабу. Цей масштаб є функцією вертикального розміру слова, якщо співвідношення сторін фіксоване.

Міра плями визначається так, щоб вона складалася з просторової протяжності та контрасту. Потім створюється дерево масштабних плям, щоб відстежувати окремі плями в масштабах [14]. У моєму аналізі відстеження окремих плям у різних масштабах не є актуальним питанням і не є доцільним з точки зору обчислень через наявність великої кількості плям, що представляють символи та слова. Також неможливо визначити, чи відповідає екстремум плямі символу чи плямі слова, і, як згадувалося раніше, варіація найкращого масштабу для слова невелика. Тому ми розглядаємо пляму як зв'язану область в просторі і

вимірюємо її просторову протяжність, але не надаємо їй ніякого об'ємного значення.



а – зображення лінії; б - зображення лінії при $\sigma_y = 1, \sigma_x = 2$; в - зображення лінії при $\sigma_y = 2, \sigma_x = 4$; г - зображення лінії при $\sigma_y = 4, \sigma_x = 16$; д - зображення лінії при $\sigma_y = 6, \sigma_x = 36$

Рисунок 2.3 – Зображення лінії та вихідне зображення при різних масштабах

Просторова протяжність як характеристика плями доступна для обчислень і вона зміщується з масштабом, даючи максимум, коли плями символів зливаються, утворюючи слова. Це узгоджується з інтуїтивним міркуванням про те, що контур слова на правильній шкалі спостереження має бути максимальний, оскільки кожна пляма має лише діапазон масштабів, щоб проявити себе.

2.6.1 Вибір η

Параметри σ_y та σ_x намагаються охопити просторові розміри слова. Важливою характеристикою слова є його співвідношення сторін. Був проведений ручний аналіз кількох зображень і показано, що середнє співвідношення сторін слова в зображенні документа становить приблизно 3-6. Раніше ми визначили коефіцієнт множення як $\eta = \frac{\sigma_x}{\sigma_y}$.

Аналіз кількох зображень показує, що для константи σ_y максимуми протяжності отримано для η , що лежить в діапазоні від 3 до 6. Зображення лінії та відповідний графік показано на рис. 2.4. На цьому рисунку отримано максимум в області між 3-6. Цей аналіз разом із спостереженням, що середнє співвідношення сторін слова лежить в діапазоні 3-6 дозволяє вибрати значення η в діапазоні 3-6. Для подальшого аналізу вибрано значення $\eta = 5$.

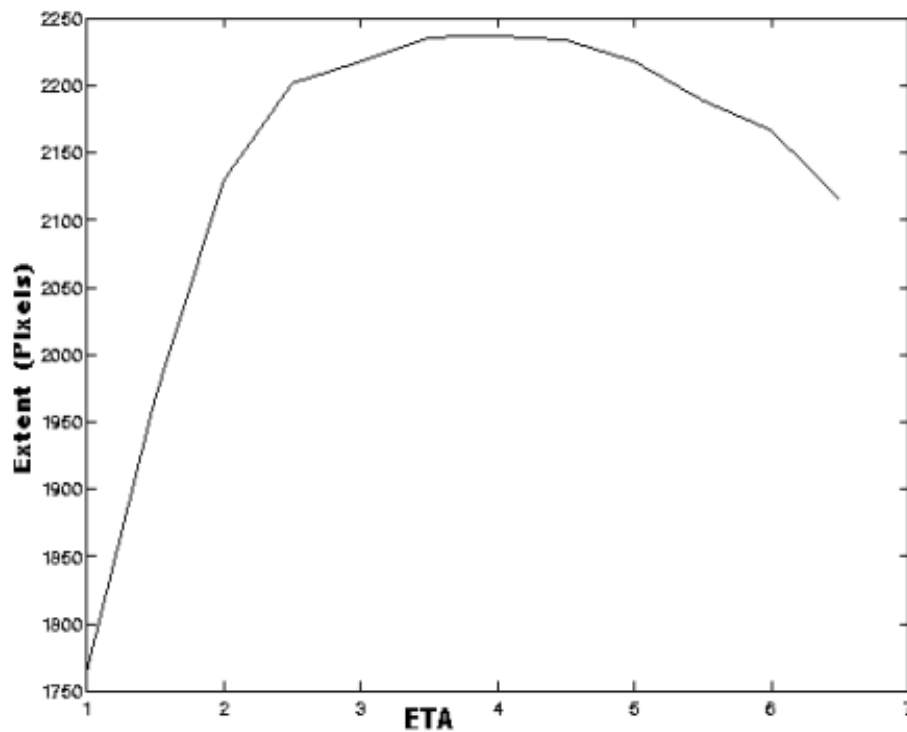
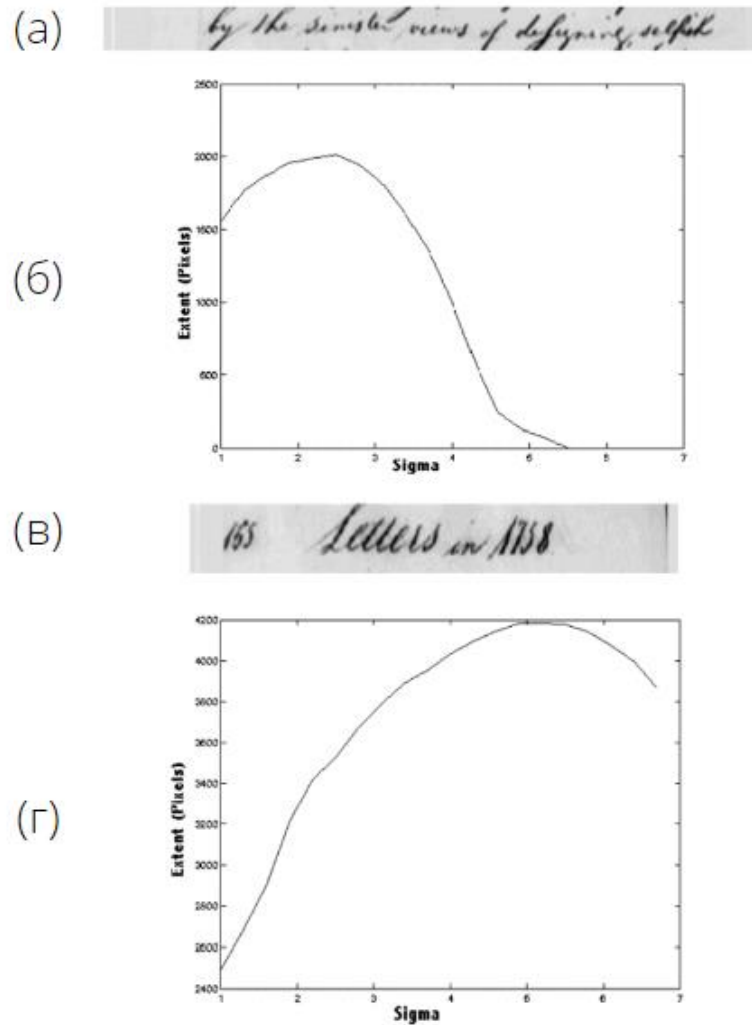


Рисунок 2.4 – Графік відношення протяжності до η , $\sigma_y=2$

2.6.2 Вибір σ_y



а - приклад лінії з меншою висотою; б - графік залежності протяжності (а) від σ_y ; в - приклад лінії з більшою висотою; г - графік залежності протяжності (в) від σ_y ;

Рисунок 2.5 – Залежність параметрів від висоти лінії

На рис. 2.5 показано зображення ліній та відповідні графіки залежності протяжності від σ_y для постійної η . Як видно на рисунку 2.5, загальна протяжність

демонструє пік, який залежить від σ_y . Також показано, як зміщується пік із зміною розміру (висоти) символів. Експериментально було встановлено, що σ_y (масштаб y) є функцією висоти слів (яка пов'язана з висотою рядка). Оцінка σ_y отримується за допомогою висоти лінії, тобто

$$\sigma_y = k \times \text{висота лінії} \quad (2.10)$$

де $0 < k < 1$.

2.7 Вилучення та постобробка плями

Наступним кроком є зіставлення плями з вихідним зображенням, щоб знайти слова. Широко використовується процедура полягає в тому, щоб закласти пляму в обмежувальну рамку, яку можна отримати за допомогою аналізу зв'язаних компонентів. У відображенні плями локалізація не підтримується. Також частини слів, особливо висхідні та низхідні, втрачені через попередні операції сегментації та згладжування рядків (розмивання). Тому наведена вище обмежувальна рамка розширюється у вертикальному напрямку, щоб включати ці верхні та нижні частини. На цьому етапі використовується фільтр для видалення невеликих частин через шум.

2.8 Висновки

У розділі було описано процес розробки складових частин алгоритму, розглянуто передобробку зображення та кожен етап сегментації, проведено аналіз їх теоретичної складової.

Для коректної роботи ключовим є попередня обробка зображення, а саме приведення його до стандартизованого виду зміною його розмірів і приведення у формат RGB. Наступним етапом є визначення координат сторінки з метою відкидання фону і подальшою роботою саме з сторінкою документа для визначення координат тексту. Для подальшої сегментації ліній необхідно визначити їх проекційний профіль. Це дозволяє знайти на зображенні ділянки з текстом і простір між рядками. Через те, що ряд локальних мінімумів і

максимумів є хибним у проєкційному профілі, було вирішено використання фільтру низьких частот Гауса.

Далі розглядається важливий пункт, а саме аналіз плями. Для формування плями використовується анізотропний фільтр Гауса, який залежить від таких параметрів: відношення ширини слова до його висоти і просторові розміри слова. Ці параметри мають бути визначені окремо для зображення. Було експериментально визначено для низки картинок ці параметри та побудовано графіки їх залежності. В середньому відношення довжини до ширини дорівнює 4, а просторова висота плями описується окремою функцією, яка залежить від висоти плями. Для використання константи в якості просторової висоти проводиться стандартизація зображення, з метою отримання однакових висот плям.

Основним етапом після аналізу плями є її вилучення та постобробка. Після накладення фільтрів не є проблемою знайти координати контурів плям. Постобробка полягає в створенні обмежувальної рамки для плями, задля подальшої роботи вже з чітко окресленою лінією, словом або символом.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПАРАМЕТРІВ АЛГОРИТМУ

3.1 Опис тестових даних

CVL є загальнодоступним датасетом для пошуку авторів, ідентифікації автора та визначення слів. Датасет складається з 7 різних рукописних текстів (1 німецький і 6 англійських текстів). Загалом у наборі даних взяли участь 310 авторів, 27 з яких написали 7 текстів, а 283 письменники написали 5 текстів. Для кожного тексту доступне кольорове зображення rgb (300 dpi), що містить рукописний текст і надрукований зразок тексту. Унікальний ідентифікатор ідентифікує автора, тоді як обмежувальні рамки для кожного окремого слова зберігаються у файлі XML.

706-1

Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows - only hard and with luminous edges - and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said "my universe": but now my mind has been opened to higher views of things.

Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows - only hard and with luminous edges - and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said "my universe": but now my mind has been opened to higher views of things.

Рисунок 3.1 – Зразок зображення з датасету CVL

CVL складається із зображень зі швидкісним рукописним текстом німецькою та англійською мовами, які були обрані з літературних творів. Усі сторінки мають унікальний ідентифікатор автора та номер тексту (розділений тире) у верхньому правому куті, а потім надрукований зразок тексту. Під друкованим текстом людей попросили написати текст, використовуючи нижній аркуш із лінійками, щоб запобігти згортанню рядків тексту. Макет відповідає стилю бази даних IAM [17].

3.2 Опис оцінки алгоритму

Коефіцієнт Жаккара (коефіцієнт флористичної спільності) — коефіцієнт, запропонований у 1901 році Полем Жаккаром. Він використовується для оцінки подібностей множин, в пошуку плагіату [18].



Рисунок 3.2 – Приклад виявлення знаку зупинки на зображенні. Прогнозована обмежувальна рамка намальована червоним кольором, а обмежувальна рамка, що відповідає дійсності, – зеленим [18]

Коефіцієнт Жаккара вимірює подібність між множинами і визначається як міра спільної частини, поділена на міру об'єднання множин:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3.1)$$

де $0 \leq J(A, B) \leq 1$

Будь-який алгоритм, який надає передбачені обмежувальні рамки як вихідні дані, може бути оцінений за допомогою коефіцієнту Жаккара. Оцінка коефіцієнту Жаккара > 0.5 зазвичай вважається хорошим прогнозом. Передбачені обмежувальні рамки, які значно перекриваються з реальними обмежувальними рамками, мають вищі оцінки, ніж з меншим перекриттям. В алгоритмі не цікавить точна відповідність (x, y)-координат, основною метою є якомога більша збіжність двох прямокутників. Це робить коефіцієнт Жаккара відмінною метрикою для оцінки користувацьких детекторів об'єктів.

3.3 Опис програмної реалізації

Алгоритм сегментації було реалізовано, використовуючи мову програмування Python версії 3.9. Python було обрано, тому що це інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією [19]. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду [19].

В якості бібліотеки для оброблення зображень було використано модуль OpenCV2 – бібліотеку алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

На рис. 3.3 та на рис. 3.4 можна побачити покрокові операції для сегментації спочатку сторінки, а після цього і ліній. Для датасету CVL сегментація сторінки не є обов'язковим пунктом, адже датасет надає вже готовий документ для сегментації ліній.



Рисунок 3.3 – Блок-схема алгоритму сегментації сторінки



Рисунок 3.4 – Блок-схема алгоритму сегментації ліній

3.4 Аналіз результатів

Демонстрацію результатів сегментації сторінки можна побачити на рис. 3.5. Розроблений алгоритм також трансформує вихідне зображення для подальшої сегментації об'єктів.

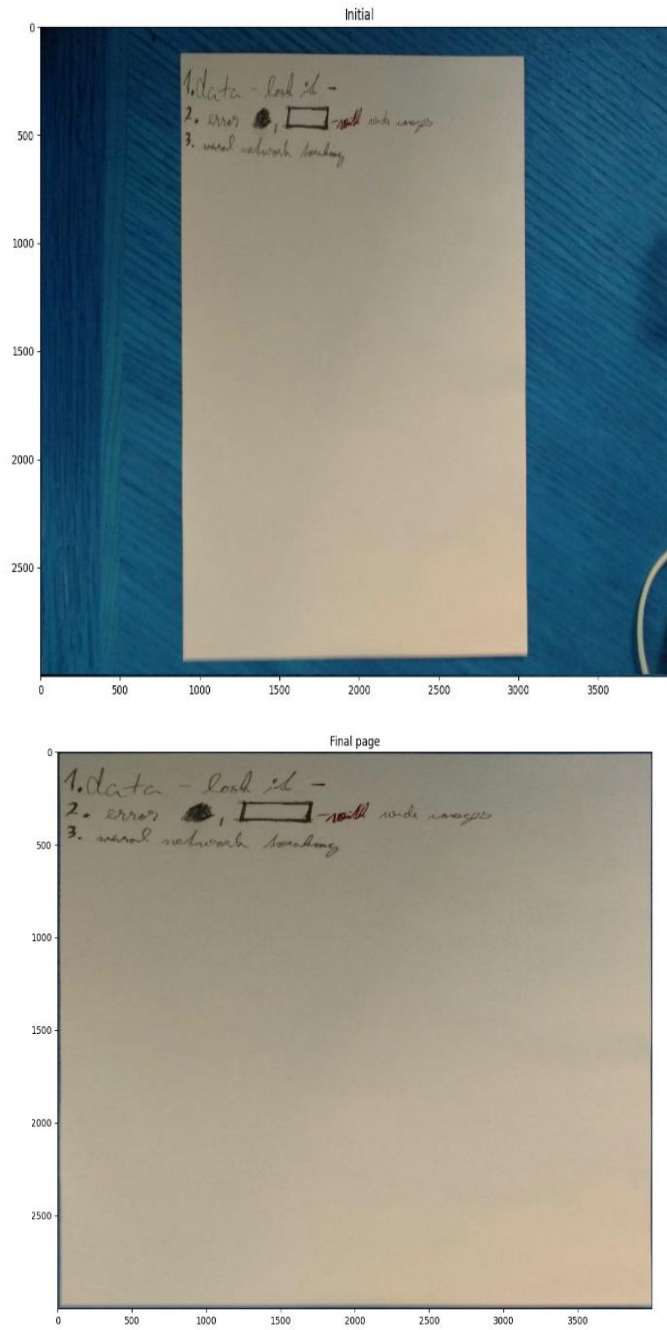


Рисунок 3.5 – Результат сегментації сторінки

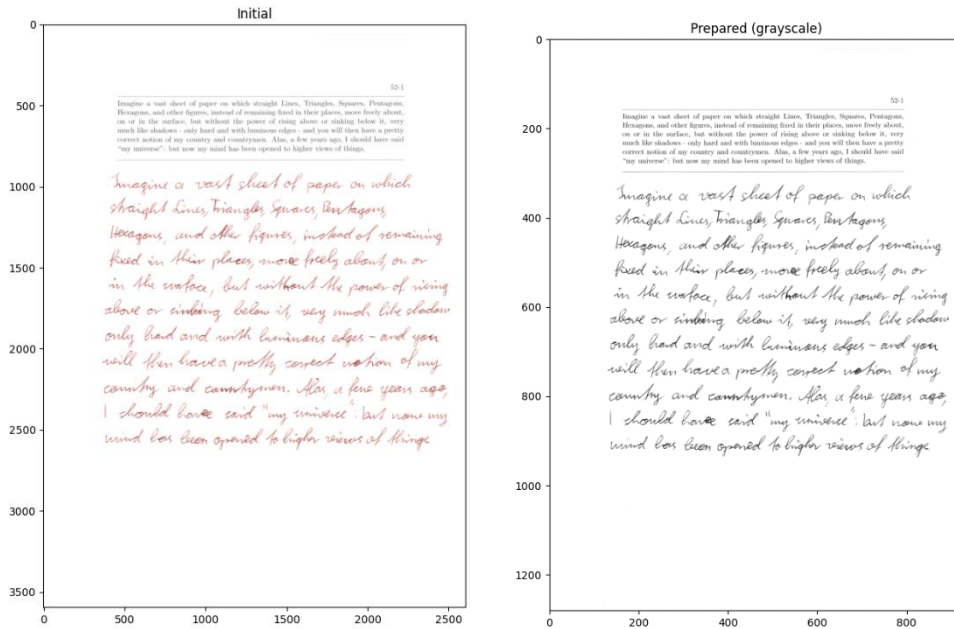


Рисунок 3.6 – Початкове зображення (ліворуч) та нормоване зображення по розмірам, у відтінках сірого (праворуч)

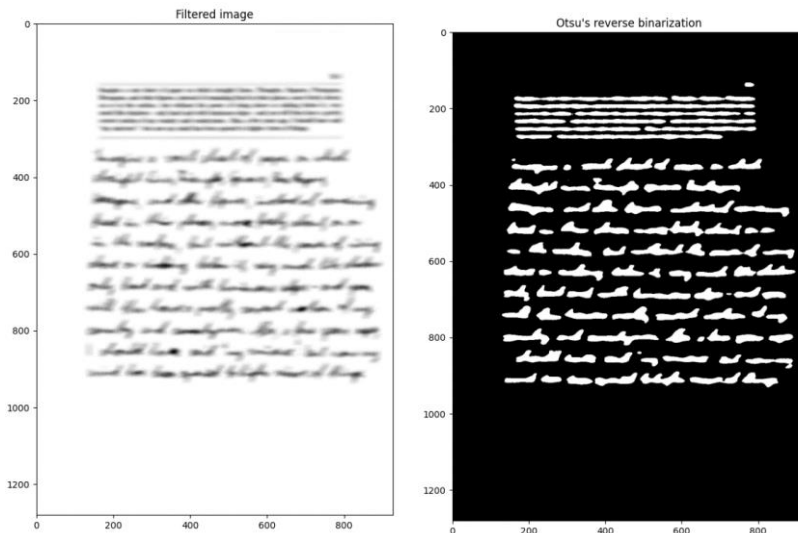


Рисунок 3.7 – Зображення після накладання фільтру Гауса (ліворуч) та зображення після бінаризації (праворуч)

На рис. 3.6 – рис. 3.8 зображені поетапні результати роботи алгоритму. Зображення сторінки взято із датасету CVL.

Як можна побачити з рис. 3.8, сегментація показує гарний результат. Порівнюючи з даними з датасету, було знайдено 88% елементів, а коефіцієнт Жаккара дорівнює 0.65, що є доволі гарним результатом.

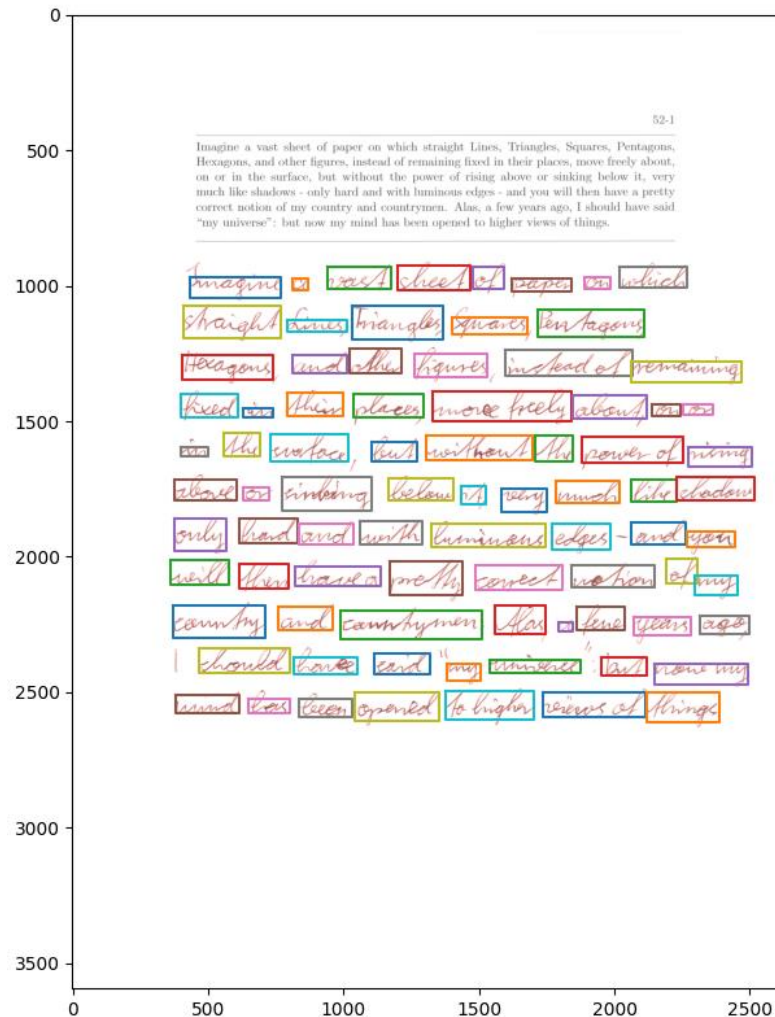


Рисунок 3.8 – Результат сегментації зображення по словам

3.5 Тестування та підбір параметрів

Для алгоритму використовуються 3 основні параметри:

- σ – параметр контролює ширину стандартне відхилення функції Гауса вздовж x -напрямку. Малі значення можуть призвести до виявлення множини об'єктів на слово (завищена сегментація), тоді як

великі значення можуть призвести до розділення слова на підсегменти (занижена сегментація);

- θ – приблизне співвідношення між шириною та висотою слова;
- resize_height – висота, яка буде у зображення з відповідно пропорційною шириною задля однакової сегментації зображень з різними параметрами, за замовчуванням 1260.

Для тестування використовується додатковий параметр boxes_delta . Він відповідає за знаходження прямокутника в результатах датасету. Бралось 1000 випадкових зображень з датасету CVL. Для оцінки використовувався коефіцієнт Жаккара.

Таблиця 3.1 – Результат сегментації для різних значень σ , при $\theta = 4$

σ	Коефіцієнт Жаккара	Кількість знайдених елементів, %
5	0.65	89.4
6	0.69	89.8
7	0.71	90.1
8	0.725	90.9
9	0.72	89.8
10	0.714	88.1
11	0.7	88

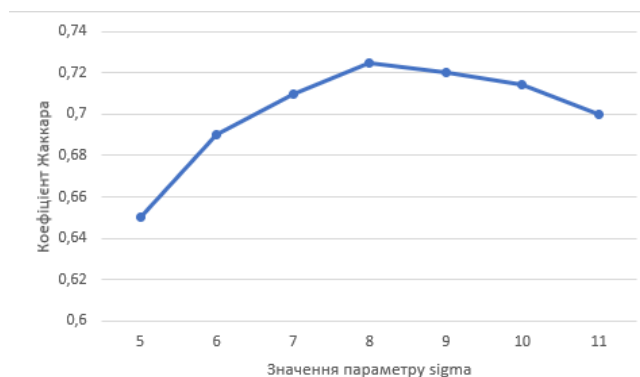


Рисунок 3.9 – Графік залежності коефіцієнту Жаккара від σ

Таблиця 3.2 – Результат сегментації для різних значень theta, при sigma 8

theta	Коефіцієнт Жаккара	Кількість знайдених елементів, %
3	0.73	90
4	0.725	89.9
5	0.681	88.8
6	0.67	88.6

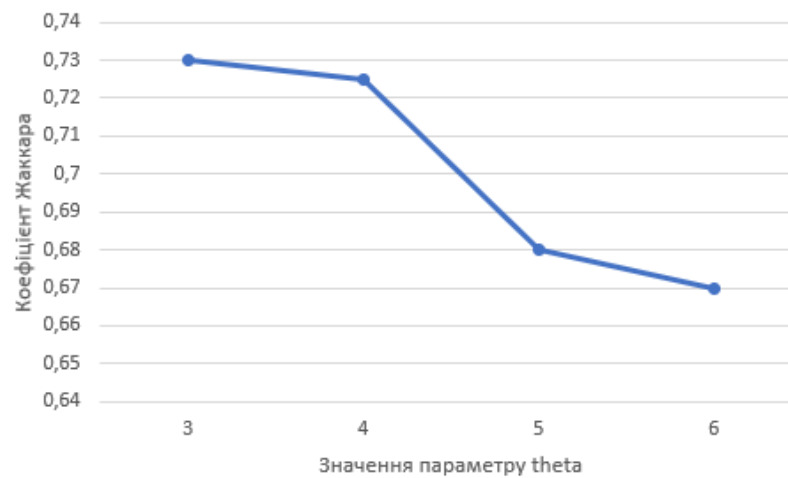


Рисунок 3.10 – Графік залежності коефіцієнту Жаккара від theta

Таблиця 3.3 – Результат сегментації для різних значень resize_height, при sigma 8, theta 3

resize_height	Коефіцієнт Жаккара	Кількість знайдених елементів, %
960	0.65	88
1020	0.66	88.1
1080	0.69	88.8
1140	0.71	89.2
1200	0.72	89.8
1260	0.72	90

Продовження таблиці 3.3

resize_height	Коефіцієнт Жаккара	Кількість знайдених елементів, %
1320	0.73	90.1
1380	0.71	89.4
1440	0.72	89

Як можна побачити з рис. 3.9 та рис. 3.10, найкращими значеннями для параметрів σ – 8, для θ – 3. В результаті визначення найкращих значень цих параметрів можна знайти найбільш кращий параметр `resize_height`.

При найкращих параметрах ($\sigma = 8$, $\theta = 3$, `resize_height` = 1320) було отримано коефіцієнт Жаккара 0.73, а кількість знайдених елементів 90.6%.

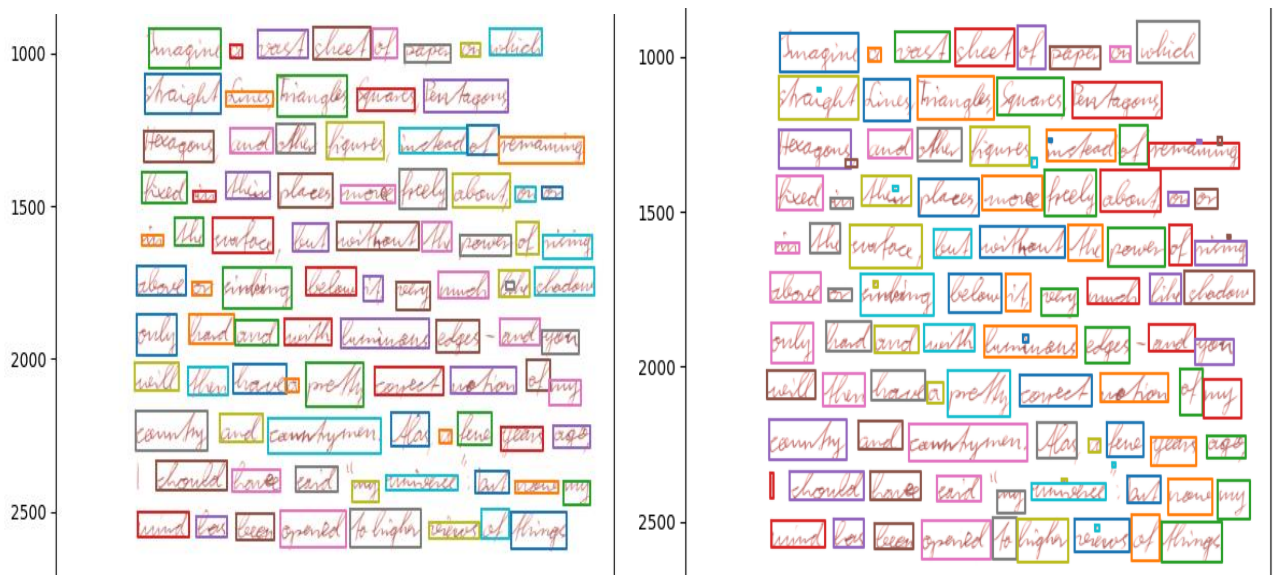


Рисунок 3.11 – Порівняння сегментованих зображень реалізованим алгоритмом (ліворуч) з даними із датасету (праворуч)

Кількість знайдених елементів не перевищує 90% для датасету. Це можна пояснити тим, що датасет не надає ідеальних сегментованих контурів слів. На

рис. 3.11 явно видно, що деякі елементи як крапки, коми, штрихи надаються як окремі елементи. В реалізованому ж алгоритмі такі об'єкти часто або не виділяються, або потрапляють в контур найбільшого сусіднього елементу.

3.6 Приклади роботи алгоритму з довільними зображеннями

На основі знайдених раніше найкращих параметрів для алгоритму можна проводити сегментацію довільного зображення з використанням цих параметрів. Для цієї задачі необхідно спочатку знайти координати документа, на якому міститься текст, а потім визначити координати слів.

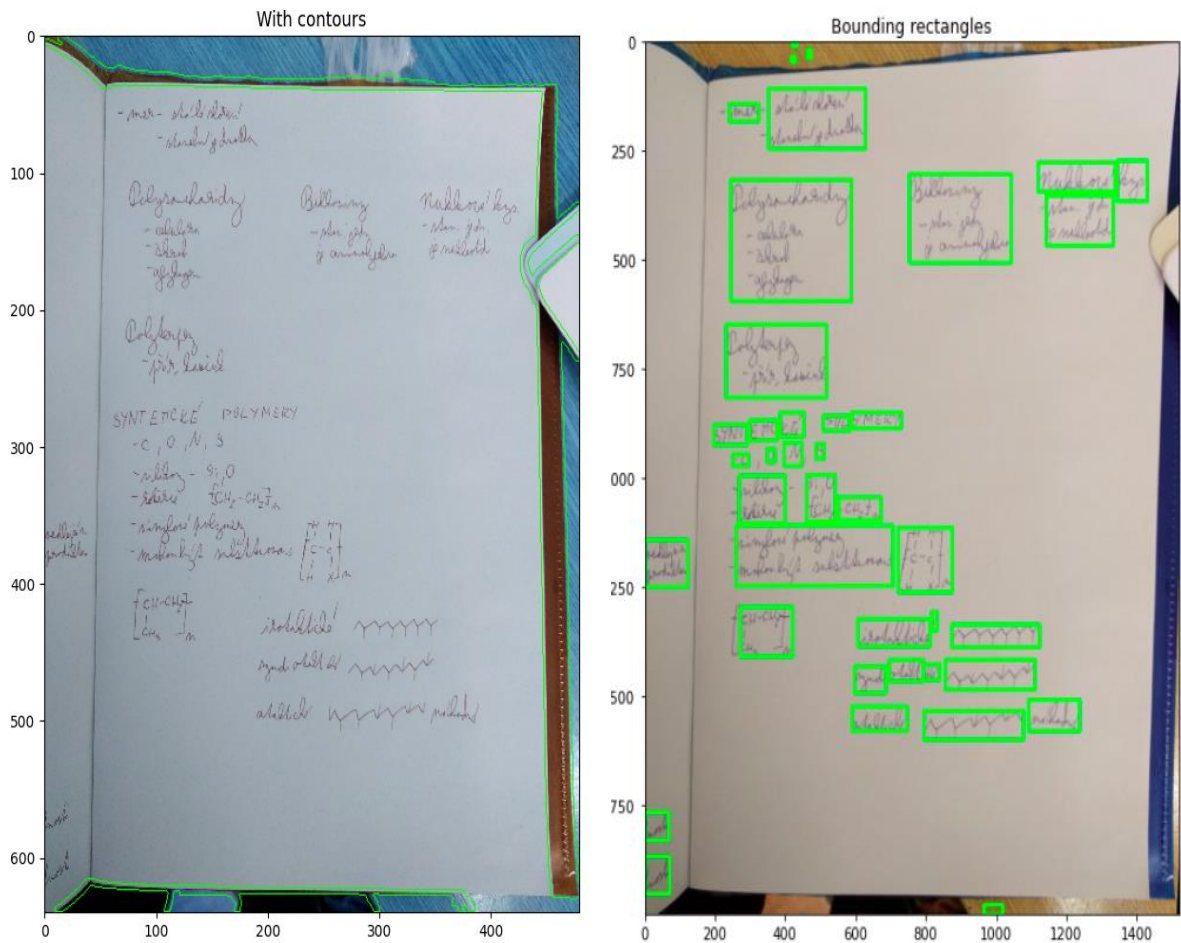


Рисунок 3.12 – Результат сегментації сторінки довільного зображення (ліворуч – сегментація сторінки документа від фону, праворуч – сегментація текстових рядків від фону)

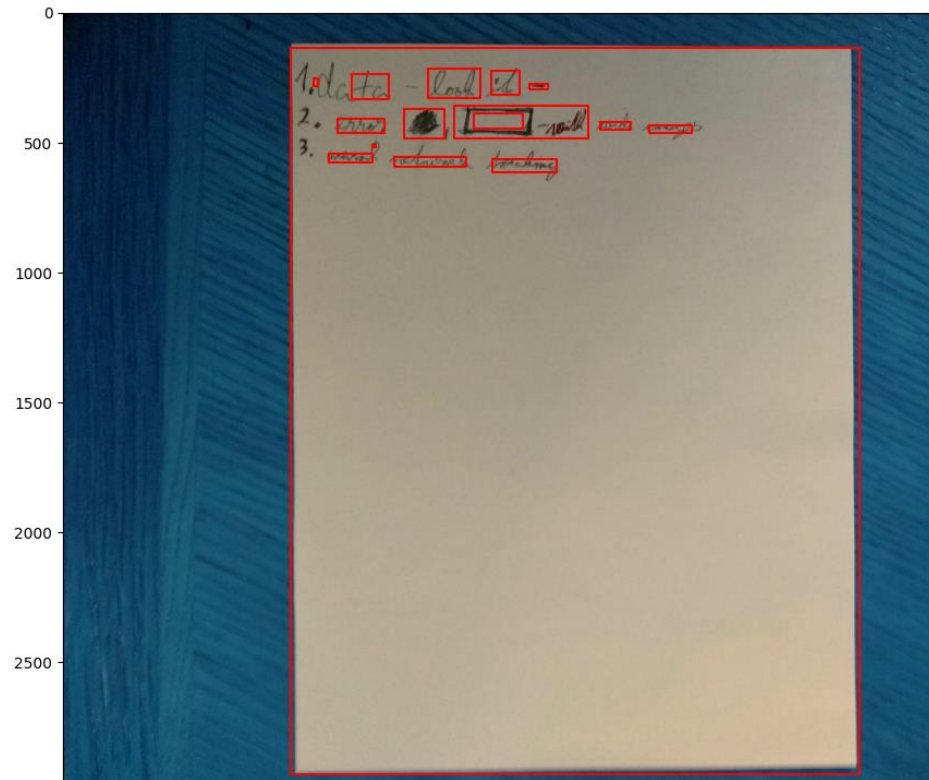


Рисунок 3.13 – Результат сегментації іншого довільного зображення

На рис 3.12 можна побачити доволі вдалий результат сегментації текстових рядків від фону зображення. Виділено всі області зображення, на яких присутній текст, але також є контури, які не несуть ніякої інформації, бо позначають порожню область. Це може бути результатом того, що розмиття не дозволяє прибрати весь шум картинки. Цю проблему можна вирішити відсіканням надмалих контурів, бо, як правило, текстові області навіть на стиснутому зображенні займають більше місця.

Для картинки з рис. 3.13 необхідно оптимізувати параметри вручну, адже області зверху зліва на документі були окреслені неправильно.

Здебільшого алгоритм показує свою ефективність на довільних зображеннях, знаходячи області з математичними формулами, малюнками та

словами. Не завжди успішно лінії розділяються між собою, але це зв'язано з тим, що на документі ці лінії йдуть без жодного відступу.

3.7 Порівняння з існуючими рішеннями

На рис. 3.14-3.15 представлені результати роботи алгоритму від Google, який використовує двоступеневу модель сегментації Mask R-CNN. Як можна побачити, всі елементи знайдені. Крім того, деякі з них мають контур відразу з нахилом, що робить результат сегментації більш точним.

Для датасету CVL також знайдено коефіцієнт Жаккара, який дорівнює 0.75, при цьому кількість знайдених елементів сягає 94%. Порівнюючи розроблений алгоритм з алгоритмом від Google, можна зробити висновок, що алгоритм від Google має кращі показники в якості і кількості знаходження, а також він розділяє великий об'єкт на декілька малих. У випадках, коли треба знаходити великий об'єкт, такий як математичне рівняння або матрицю, можна використовувати розроблений алгоритм.

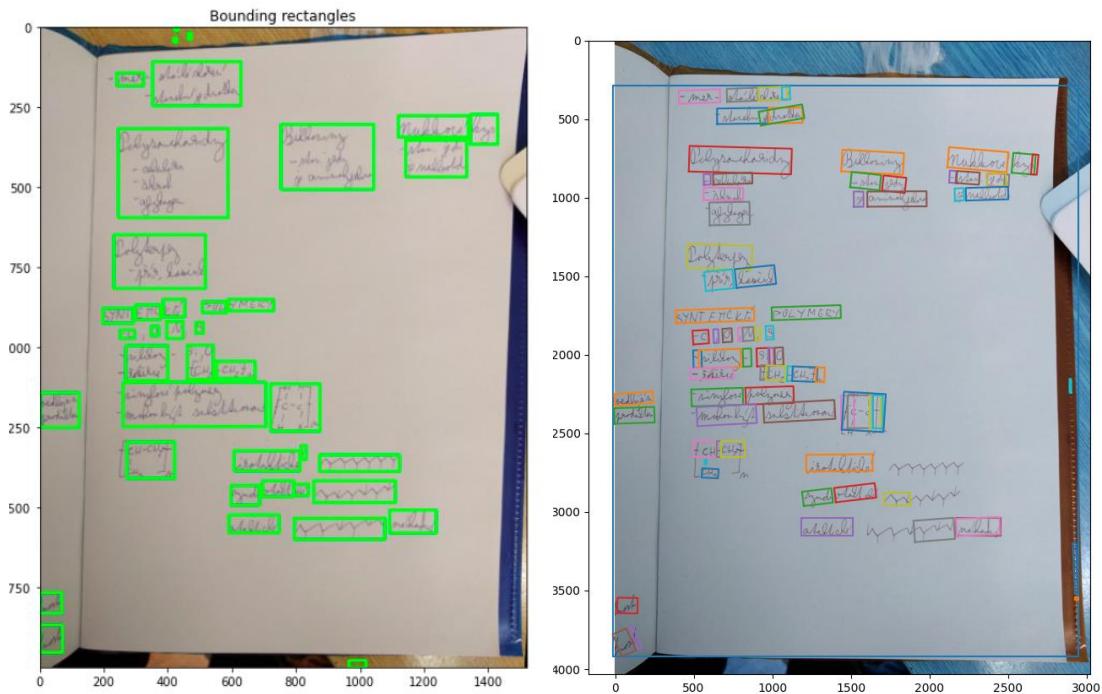


Рисунок 3.14 – Результат роботи алгоритму від Google(праворуч) та розробленого алгоритму (ліворуч) на довільному зображенні



Рисунок 3.15 – Результат роботи алгоритму від Google (праворуч) та розробленого алгоритму (ліворуч) на зображенні з CVL

3.8 Висновки

Було проведено аналіз результатів роботи алгоритму сегментації з використанням датасету CVL, а також продемонстровано результат сегментації на власному зображенні.

Датасет CVL було обрано через його велику базу зображень з рукописними даними. Значна кількість зображень з різним текстом і почерком дозволяє реалізувати алгоритм, який буде працювати стабільно і не буде залежати від мови і почерку автора.

В якості оцінки коректності роботи сегментації було використано коефіцієнт Жаккара, який дозволяє оцінити множину об'єктів між собою. В задачі

сегментації цей метод робить можливим оцінку контурів об'єктів між даними з датасету та тими, що віддає алгоритм.

Було продемонстровано як покроково працює алгоритм і з якими проміжними зображеннями він стикається, як виглядає зображення після нанесення різних фільтрів і як саме виділяються основні об'єкти від фону.

Методом підбору параметрів було визначено найбільш оптимальні параметри для алгоритму, при яких отримується найбільший коефіцієнт Жаккара та знаходиться найбільша кількість елементів. Результат роботи при найкращих параметрах було продемонстровано на довільному зображенні, всі необхідні об'єкти було знайдено.

Також було протестовано роботу алгоритму від Google. Значення коефіцієнту Жаккара показує, що алгоритм дуже добре справляється з задачею сегментації. Крім того, як показано з результатів роботи алгоритму, знайдені об'єкти мають кращий контур, що є кращим для подальшої задачі розпізнавання.

Загалом можна стверджувати, що запропонований алгоритм сегментації дуже гарно справляється зі своєю задачею. Він показує непоганий результат не тільки на підготованому датасеті, а і з реальними даними.

4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ

Описаний алгоритм сегментації зображень може бути застосованим для заміни вже існуючих підходів в сегментації рукописного тексту для подальшої задачі розпізнавання тексту. У даному розділі проводиться оцінка можливої реалізації технології з функціонально-вартісним аналізом її застосування. Програмний продукт був розроблений за допомогою мови програмування Python, є крос-платформним та не залежить від операційної системи.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями [20].

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. В даній роботі проводиться аналіз функцій розроблюваного програмного продукту та виявлення всіх витрат на їх реалізацію [21].

4.1 Постановка задачі техніко-функціонального аналізу

Для кожної функції визначається повний обсяг річних витрат та кількість робочих часів. На основі даних оцінок визначається кількісна характеристика джерел витрат [21]. Після визначення джерел витрат проводиться кінцевий розрахунок витрат на виробництво продукту.

До продукту були визначені наступні технічні вимоги:

- можливість запуску програми на персональних комп'ютерах із стандартною конфігурацією;

- висока швидкість обробки даних та відповідь користувачеві у реальному часі;
- зручність та простота взаємодії;
- загальна доступність програми широкому колу користувачів;
- мінімальні витрати на впровадження програмного продукту.

4.2 Обґрунтування функцій та параметрів програмного продукту

Головна функція F_0 – розробка програмного продукту, який буде набір моделей прогнозу та перевіряє їх точність на певному наборі даних. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір мови програмування;

F_2 – вибір готових бібліотек;

F_3 – вибір середі розробки.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

а) python;

б) c#.

Функція F_2 :

а) openCV2;

б) PIL.

Функція F_3 :

а) PyCharm;

б) Jupyter Notebook.

4.2.1 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1). Побудовано позитивно-негативну матрицю варіантів основних функцій на основі цієї карти (таблиця 4.1).

Морфологічна карта відображає всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів програмного продукту [21].

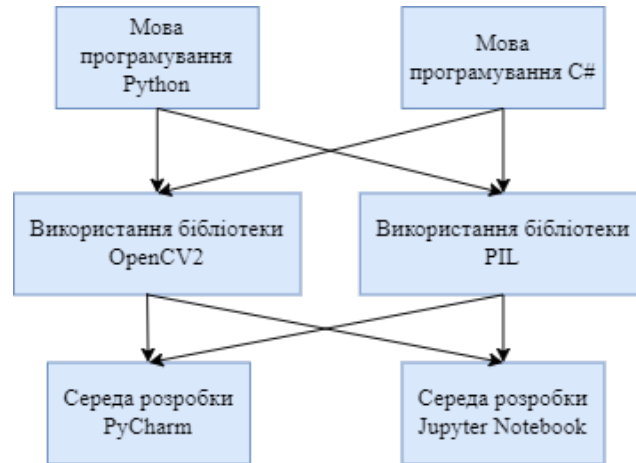


Рисунок 4.1 – Морфологічна карта

Таблиця 4.1 – Позитивно-негативна карта

Основні функції	Варіанти реалізації	Переваги	Недоліки
F ₁	А	Займає менше часу при написанні коду	Повільний
	Б	Код швидко виконується	Займає більше часу для написання коду
F ₂	А	Має великий функціонал	Повільніше працює
	Б	Невелика та швидка бібліотека	Функціоналу не дуже багато
F ₃	А	Має вбудований функціонал для відлагодження	Професійна версія платна
	Б	Легкий у створенні	Відсутня крос-платформеність

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F₁:

Оскільки поточна реалізація не враховує час виконання алгоритму, швидкість його розробки важливіша, тому варіант б) має бути відкинтий.

Функція F₂:

Оскільки планується розглядати широкий спектр різноманітних варіантів даних, та перший є більш зручним в даному випадку, то варіант б) має бути відкинтий.

Функція F₃:

Інтерфейс користувача не відіграє велику роль у даному програмному продукті, тому вважаємо а) та б) гідними розгляду.

Таким чином, будемо розглядати такі варіанти реалізації ПП.

F_{1a} – F_{2a} – F_{3a}

F_{1a} – F_{2a} – F_{3б}

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів ПП

4.3.1 Опис параметрів

Спираючись на дані про основні функції, які повинен реалізувати програмний продукт, та вимоги до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- X₁ – об'єм пам'яті, необхідний для збереження та обробки даних;

- X2 – потенційний об'єм програмного коду;
- X3 – швидкодія операцій мови програмування в залежності від обраної бібліотеки;
- X4 – час обробки даних.

4.3.2 Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту як показано у табл. 4.2.

Таблиця 4.2 – Основні параметри ПП

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметру		
			гірші	середні	кращі
Об'єм пам'яті для збереження та обробки даних	X1	Мб	35	15	5
Об'єм програмного коду	X2	кількість строк коду	2500	1800	1200
Швидкодія операцій мови програмування	X3	Оп/мс	20000	12000	3200
Час обробки даних	X4	мс	1500	740	300

За даними таблиці X будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5

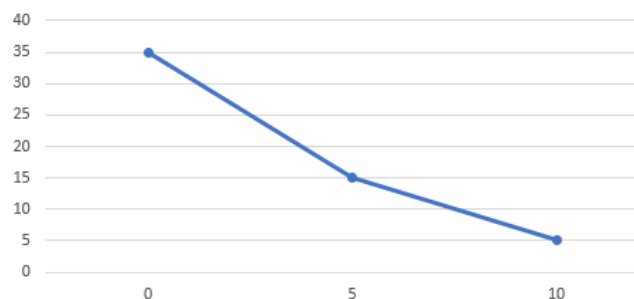


Рисунок 4.2 – X1, об'єм пам'яті для збереження та обробки даних

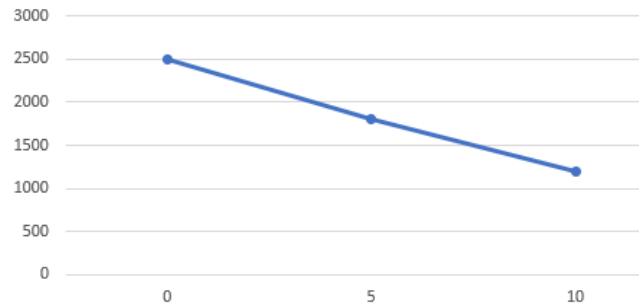


Рисунок 4.3 – X2, потенційний об'єм програмного коду

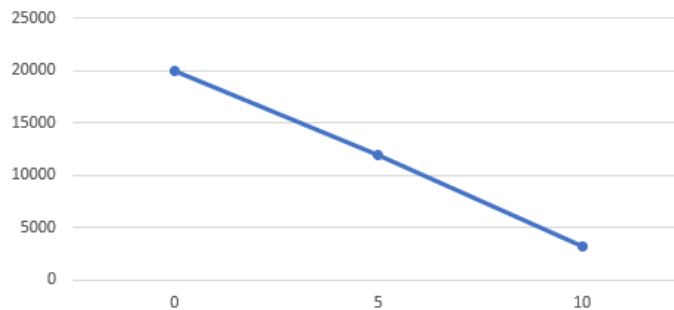


Рисунок 4.4 – X3, швидкодія операцій мови програмування в залежності від обраної бібліотеки

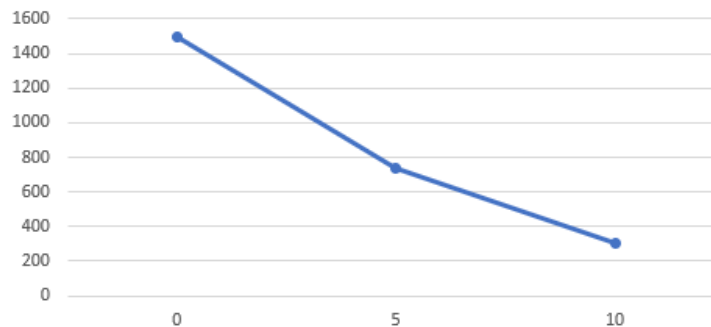


Рисунок 4.5 – X4, час обробки даних

4.3.3 Аналіз експертного оцінювання параметрів

Наступним етапом ФВА є ранжування показників. Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру, спираючись на поставлену перед проектом мету.

Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Параметр	Ранг параметра за оцінкою експерта							Сума рангів	Відхилення Δ_i	Δ_i^2
	1	2	3	4	5	6	7			
X1	4	4	3	4	3	3	4	25	-1.25	1.56
X2	6	6	6	5	6	6	6	41	14.75	217.56
X3	4	4	4	4	3	4	4	27	0.75	0.56
X4	2	2	1	2	2	2	1	12	-14.25	203
Разом	15	15	14	15	13	15	15	105	0	422.7

Для перевірки ступеню достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів наведені у формулах 4.1 та 4.2 відповідно:

$$R_i = \sum_{j=1}^N r_{ij} = 105 \quad (4.1)$$

де r_{ij} – ранг i -го параметра, визначений j -м експертом;

N – число експертів.

б) середня сума рангів T :

$$T = \frac{1}{n} R_i = 26.25 \quad (4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T \quad (4.3)$$

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^n \Delta_i^2 = 422.7 \quad (4.4)$$

д) коефіцієнт узгодженості (конкордації):

$$W = \frac{12S}{N^2 * (n^3 - n)} = \frac{12 * 422.7}{7^2 * (4^3 - 4)} = 1.72 > W_k = 0.67 \quad (4.5)$$

Ранжирування можна вважати достовірним, оскільки знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0.67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів, результати якого наведені у таблиці 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	<	<	<	<	<	<	0.5
X1 і X3	=	=	<	=	=	<	=	=	1
X1 і X4	>	>	>	>	>	>	>	>	1.5
X2 і X3	>	>	>	>	>	>	>	>	1.5
X2 і X4	>	>	>	>	>	>	>	>	1.5
X3 і X4	>	>	>	>	>	>	>	>	1.5

Числове значення, що визначає ступінь переваги і-го параметра над j-тим, a_{ij} , визначається за формулою:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases} \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості K_{B_i} за формулою:

$$K_{B_i} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (4.7)$$

де $b_i = \sum_{i=1}^n a_{ij}$ – вагомість і-го параметра за результатами оцінок експертів; a_{ij} – коефіцієнт переваги і-го на j-м параметром.

Відносні оцінки розраховуються декілька разів, поки наступні значення не будуть незначно відрізнятись від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{B_i} = \frac{b'_i}{\sum_{i=1}^n b'_i} \quad (4.8)$$

де $b'_i = \sum_{i=1}^n a_{ij} b_i$.

З таблиці 4.5 видно, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.	
	X1	X2	X3	X4	B_i	K_{B_i}	B_i^1	$K_{B_i}^1$
X1	1	0.5	1	1.5	4	0.25	14.5	0.24
X2	1.5	1	1.5	1.5	5.5	0.34	21.25	0.36
X3	1	0.5	1	1.5	4	0.25	14.5	0.24
X4	0.5	0.5	0.5	1	2.5	0.16	9.25	0.16
Разом					16	1	59.5	1

4.4 Аналіз рівня якості варіантів реалізації функцій

Рівень якості кожного варіанту виконання основних функцій визначається окремо.

Абсолютне значення параметру X1(об'єм пам'яті для збереження та обробки даних) обрано не найгіршим (не максимальним), тобто це значення відповідає варіанту а) 5 або б) 15 Мб.

Абсолютне значення параметру X2 (об'єм програмного коду) відповідає технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X3 (швидкодія операцій мови програмування) відповідає варіанту а) 20000 або б) 3200 Оп/мс.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується за формулою:

$$K_{TP} = \sum_{i=1}^n K_{B_i} B_i \quad (4.9)$$

де n – кількість параметрів; K_{B_i} – коефіцієнт вагомості i -го параметра; B_i – оцінка i -го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1(X3)	А	3200	3.6	0.24	0.86
F2(X2)	А	1200	5.6	0.36	2
F3(X1, X4)	А	15	2.2	0.24	0.53
	Б	5	1.4	0.19	0.27

Використавши дані з таблиці 4.6, визначимо рівень якості кожного з варіантів:

$$K_{я1} = 0.86 + 2 + 0.53 = 3.39$$

$$K_{я2} = 0.86 + 2 + 0.27 = 3.13$$

Оскільки коефіцієнт якості у першому варіанті вище, варіант 1 є найкращим.

4.5 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки програмного продукту спочатку проведемо розрахунок трудомісткості. Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка алгоритму збору даних.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1, належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовуємо інформацію з довідника, а для завдання 2 використовуємо інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань. Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М} \quad (4.10)$$

де T_P – трудомісткість розробки програмного продукту; K_{Π} – поправочний коефіцієнт; $K_{СК}$ – коефіцієнт на складність вхідної інформації; K_M – коефіцієнт рівня мови програмування; $K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм; $K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання трудомісткість дорівнює: $T_P = 85$ людино-днів, виходячи із часових норм для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1. Поправочний коефіцієнт, який враховує вид вхідної інформації для першого завдання: $K_{\Pi} = 1.7$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації рівний $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.8$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 85 \cdot 1.7 \cdot 0.8 = 115.6 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для другого завдання, в якому використовується алгоритм третьої групи складності зі ступенем новизни Б, тобто $T_P = 26$ людино-днів, $K_{\Pi} = 0.9$, $K_{СК} = 1$, $K_{СТ} = 0.8$:

$$T_2 = 26 \cdot 0.9 \cdot 0.8 = 18.7 \text{ людино-днів.}$$

Оскільки загальна трудомісткість усіх варіантів реалізації збігається, їх можна об'єднати в одну групу.

Загальна трудомісткість складає:

$$T_0 = (115.6 + 18.7) \cdot 8 = 1074.4 \text{ людино-годин.}$$

В розробці беруть участь два програмісти з окладом 15000 грн., один фінансовий аналітик з окладом 10000 грн. Визначимо середню зарплату за годину за формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.}, \quad (4.11)$$

де M – місячний оклад працівників; T_m – кількість робочих днів на місяць; t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{15000 + 15000 + 10000}{3 \cdot 21 \cdot 8} = 79.37 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}}, \quad (4.12)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста; T_i – трудомісткість відповідного завдання; $K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$C_{\text{зп}} = 79.37 \cdot 1074.4 \cdot 1.2 = 102\,330.15 \text{ грн.}$$

Відрахування на соціальний внесок становить 22%:

$$C_{\text{св}} = C_{\text{зп}} \cdot 0.22 = 102\,330.15 \cdot 0.22 = 22\,512.46 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. Оскільки одна ЕОМ обслуговується одним програмістом з окладом 15000 грн. та коефіцієнтом зайнятості $K_3 = 0.2$ то для однієї машини отримаємо:

$$C_{\text{г}} = 12 \cdot M \cdot K_3 = 12 \cdot 15\,000 \cdot 0.2 = 36\,000 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{\text{зп}} = C_{\text{г}} \cdot (1 + K_3) = 36\,000 \cdot (1 + 0.2) = 43\,200 \text{ грн.}$$

Відрахування на соціальний внесок становить 22%:

$$C_{\text{св}} = C_{\text{зп}} \cdot 0.22 = 43\,200 \cdot 0.22 = 9\,594 \text{ грн.}$$

Амортизаційні відрахування розраховуємо за формулою при амортизації 25% та вартості ЕОМ – 9 000 грн.:

$$C_{\text{а}} = K_{\text{тм}} \cdot K_{\text{а}} \cdot \text{Ц}_{\text{пр}} = 1.15 \cdot 0.25 \cdot 9000 = 2\,586.5 \text{ грн.}$$

де K_{TM} – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; K_A – річна норма амортизації; $C_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо за формулою:

$$C_P = K_{TM} \cdot K_P \cdot C_{ПР} = 1,15 \cdot 0,05 \cdot 9000 = 517,5 \text{ грн.}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{EF} &= (D_K - D_B - D_C - D_P) \cdot t \cdot K_B, T_{EF} = \\ &= (365 - 104 - 8 - 16) \cdot 8 \cdot 0,9 = 1706,4 \text{ год.} \end{aligned}$$

де D_K – календарна кількість днів у році; D_B, D_C – відповідно кількість вихідних та святкових днів; D_P – кількість днів планових ремонтів устаткування; t – кількість робочих годин в день; K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{EL} = T_{EF} \cdot N_C \cdot K_3 \cdot C_{EL} = 1706,4 \cdot 0,65 \cdot 0,2 \cdot 3,03 = 627,15 \text{ грн.}$$

де N_C – середньо-споживча потужність приладу; K_3 – коефіцієнтом зайнятості приладу; C_{EL} – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{ПР} \cdot 0,67 = 9000 \cdot 0,67 = 6030 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть складати:

$$\begin{aligned} C_{EK} &= C_{ЗП} + C_{СВ} + C_A + C_P + C_{EL} + C_H = \\ &= 43200 + 9594 + 2586,5 + 517,5 + 627,15 + 6030 = 62555,15 \text{ грн.} \end{aligned}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{MG} = \frac{C_{EK}}{T_{EF}} = \frac{62555,15}{1706,4} = 36,66 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу складають:

$$C_M = C_{MG} \cdot T = 36.66 \cdot 1074.4 = 39\,387.5 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{ЗП} \cdot 0.67 = 102\,330.15 \cdot 0.67 = 68\,561.2 \text{ грн.}$$

Отже, вартість розробки програмного продукту за варіантами становить:

$$\begin{aligned} C_{ПП} &= C_{ЗП} + C_{СВ} + C_M + C_H = \\ &= 102\,330.15 + 22\,512.46 + 39\,387.5 + 68\,561.2 = 232\,791.31 \text{ грн.} \end{aligned}$$

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}} = \frac{K_K}{C_{ПП}} = \frac{3.39}{232791.31} = 1.5 \cdot 10^{-5}$$

4.6 Висновки

В даному розділі було проведено функціонально-вартісний аналіз впровадження технології, яка була досліджена в ході даної роботи. Процес аналізу поділений на дві умовні частини.

Спершу було проаналізовано програмний продукт з технічної точки зору: визначено основні функції та параметри ПП, що мають бути пріоритетними задля вибору найкращої реалізації.

В другій частині ФВА відбулось економічне обґрунтування альтернативних варіантів впровадження. Коефіцієнт ефективності став показником для порівняння запропонованих варіантів реалізації, які були виконані з урахуванням таких параметрів як трудомісткість, витрати на заробітну плату, електроенергію та накладні витрати.

З ФВА випливає, що найоптимальнішим варіантом серед альтернатив, що залишились після відбору двох варіантів, є перший варіант. У нього виявився

найкращий показник техніко-економічного рівню якості $K_{\text{TEP}} = 1.5 * 10^{-5}$,
вартість витрат для нього ставить $C_{\text{ПП}} = 232791.31$ грн.

Цей варіант реалізації ПП має такі параметри:

- мова програмування Python;
- використання бібліотеки OpenCV2;
- середовище розробки PyCharm.

ВИСНОВКИ

Метою роботи було дослідження та розробка методів сегментації для задачі розпізнавання рукописного тексту.

Було розглянуто поняття сегментації та її основні методи, які традиційно використовуються у задачі знаходження об'єктів на зображенні. Загалом можна стверджувати, що використання більшості алгоритмів залежить від факторів зображення, таких як яскравість, кольорова гама, кут нахилу об'єкта. Також існують нейронні мережі, які можуть справлятися з вище перерахованими факторами, але вони залежать від тренуваності моделі мережі, і тому не завжди можуть бути використані для того чи іншого зображення.

Дослідивши базові методи сегментації, було запропоновано новий алгоритм, що здатний знаходити об'єкт, порівнюючи яскравості об'єкта та фону. За допомогою фільтра Гауса можна розмити зображення так, щоб бінарна маска, що накладається на нього, дозволяла б з легкістю відрізнити потрібні об'єкти від фону та шуму. Задля коректного розмиття треба визначити основні параметри:

- відношення ширини слова до його висоти;
- просторові розміри слова.

Їх було експериментально визначено для низки картинок під час тестування алгоритму. Також було запропоновано алгоритм знаходження координат сторінки, що виконаний методом грубої сегментації, а саме накладанням стандартних фільтрів на зображення і подальшим знаходженням контуру документа.

Протестувавши розроблений алгоритм з існуючим від Google, були визначені його недоліки, а саме не завжди точне знаходження правильних координат кутів об'єкта і рідка побудова координат для ділянки зображення, в якому будь-який об'єкт відсутній. Але також він має ряд переваг, які можуть використовуватися за деяких умов: визначення великого об'єкта нероздільно, що краще впливає на знаходження рукописних малюнків, математичних формул, а

також невключення у результат сегментації надмалих об'єктів, які можна вважати шумом зображення.

Результатом роботи є відлагоджений та протестований алгоритм сегментації для подальшого розпізнавання тексту. Коефіцієнт Жаккара при цьому дорівнює 0.73 а кількість знайдених елементів 90.6% порівнюючи з даними із датасету CVL, що у рамках задачі сегментації є доволі гарним результатом.

Оскільки сегментація, особливо останнім часом, є доволі важливим етапом у задачі розпізнавання тексту, потенційні застосування даної роботи розширюються та можуть включати в себе:

- 1) розробку програмного забезпечення, що на основі завантаженого зображення буде проводити сегментацію та віддавати користувачу в якості відповіді координати контурів а також саме зображення, з накладеними на нього контурами;
- 2) використання отриманих результатів сегментації як даних, для порівняння з новими алгоритмами, підходами, а також для покращення продуктивності реалізованого алгоритму.

Крім застосування в задачах розпізнавання тексту, запропонований алгоритм можна налаштувати таким чином, що буде проходити розпізнавання й інших об'єктів. Але для цього треба впроваджувати нові кроки, притаманні для сфери знаходжених об'єктів.

Таким чином в роботі була розкрита актуальна науково-прикладна тема: створення нового алгоритму сегментації для подальшої задачі розпізнавання рукописного тексту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Mihajlovic I. Everything You Ever Wanted To Know About Computer Vision [Електронний ресурс] // Towards Data Science. – 2019. – Режим доступу до ресурсу: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>
2. Шантыр Д. С., Невдащенко М. В., Болдак О. А. Среда "быстрой" разработки систем компьютерной обработки изображений. Вестник НТУУ "КПИ": Информатика, управление и вычислительная техника. – 2012. – 169 с.
3. Parker J. R. Algorithms for Image Processing and Computer Vision. Wiley, 2nd edition. – 2010. – 504 p.
4. Susan S., Racha Devi K. M. Text area segmentation from document images by novel adaptive thresholding and template matching using texture cues. – 2019. – 869 p.
5. Marschener S. and others Fundamentals of Computer Graphics. A K Peters/CRC Press; 5th edition. – 2021. – 700 p.
6. Гонсалес Р., Вудс Р. Цифровая обработка изображений / пер. с англ. Л. И. Рубанова, П. А. Чочиа. Москва. – 2012. – 1104 с.
7. Ревякин А.М., Скурнович А.В. Подходы к разработке системы распознавания для решения задачи определения контента цифровых изображений. – Интернет журнал «НАУКОВЕДИНИЕ» Том 8, №4, 2016.
8. Sai Hareesh A., Chandrasekaran V. Exemplar-based color image inpainting: a fractional gradient function approach [Електронний ресурс] // Pattern Analysis and Applications. – 2014. – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s10044-012-0316-4>
9. On-device Panoptic Segmentation for Camera Using Transformers [Електронний ресурс] // Apple. – 2021. – Режим доступу до ресурсу: <https://machinelearning.apple.com/research/panoptic-segmentation>

10. What's in an image: fast, accurate image segmentation with Cloud TPUs [Электронный ресурс] / Pathak O. // Google. – 2019. – Режим доступа до ресурсу: <https://cloud.google.com/blog/products/ai-machine-learning/whats-in-an-image-fast-accurate-image-segmentation-with-cloud-tpus>
11. Semantic Segmentation Algorithm [Электронный ресурс] // Amazon. – Режим доступа до ресурсу: <https://docs.aws.amazon.com/sagemaker/latest/dg/semantic-segmentation.html> – Дата доступа: 01.05.2022.
12. Nati O., Meirav G., Sharon A., Achi B. "On Detection of Faint Edges in Noisy Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.42, no.4, pp.894-908. – 2020.
13. Hendel D., Johnston K., Patra R., Sen B. A machine-vision method for automatic classification of stellar halo substructure [Электронный ресурс] // Monthly Notices. – 2019. – Режим доступа до ресурсу: <https://academic.oup.com/mnras/article/486/3/3604/5475124>
14. Bohleber P., Barbante C., Pelillo M. Ice Core Science Meets Computer Vision: Challenges and Perspectives [Электронный ресурс] // Frontier in Computer Science. – 2022. – Режим доступа до ресурсу: https://www.researchgate.net/publication/352486928_Ice_Core_Science_Meets_Computer_Vision_Challenges_and_Perspectives
15. Tavoli R., Keyvanpour M. A Novel Word-Spotting Method for Handwritten Documents Using an Optimization-Based Classifier [Электронный ресурс] // Taylor & Francis Online. – 2017. – Режим доступа до ресурсу: <https://www.tandfonline.com/doi/full/10.1080/08839514.2017.1346964>
16. Canny edge detector [Электронный ресурс] // Wikipedia. – Режим доступа до ресурсу: https://en.wikipedia.org/wiki/Canny_edge_detector
17. An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting [Электронный ресурс] // CVL. – 2015. – Режим доступа до ресурсу:

- <https://cvl.tuwien.ac.at/research/cvl-databases/an-off-line-database-for-writer-retrieval-writer-identification-and-word-spotting/>
18. Jaccard Index [Електронний ресурс] // Wikipedia. – Режим доступу до ресурсу: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> – Дата доступу: 03.04.2022.
 19. Python [Електронний ресурс] // Wikipedia. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Python> – Дата доступу: 15.04.2022.
 20. Яловий Г.К. Економіка та організація виробництва / Яловий Г.К., Пашін В.П., Сичов В.С. – К.: "Політехніка". – 2004. – С. 80.
 21. Богданюк В.Є Методичні вказівки до виконання організаційно-економічного розділу дипломних проектів / Богданюк В.Є., Березовський К.В., Пашін В.П. – К.: НТУУ "КПІ". – 1999. – С. 66.
 22. Flor A. Text Segmentation [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://arthurflor23.medium.com/text-segmentation-b32503ef2613> – Дата доступу: 05.05.2022.
 23. Cherhera A., Zhelezniakov D., Tereshenko V. Segmentation of Handwritten Mathematical Matrices Using the Area Voronoi Diagram [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.researchgate.net/publication/354617169_Segmentation_of_Handwritten_Mathematical_Matrices_Using_the_Area_Voronoi_Diagram – Дата доступу 01.06.2022.
 24. Самойленко Д. Е. Структурная сегментация изображений / // Искусств. интеллект. – 2004. – № 4. – С. 521-528
 25. Mahmoud A. A. Mousa Arabic Character Segmentation Using Projection-Based Approach with Profile's Amplitude Filter [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: <https://arxiv.org/ftp/arxiv/papers/1707/1707.00800.pdf>