

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра комп'ютерної інженерії

ЗАЯЦЬ Анастасія-Марія Володимирівна

**Програмний засіб розпізнавання тексту на основі
клітинних автоматів /
Software for text recognition based on cellular devices**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІзм-21
ЗАЯЦЬ Анастасія-Марія Володимирівна

Науковий керівник
к.т.н. Н.Я. Савка

ТЕРНОПІЛЬ - 2023

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний засіб розпізнавання тексту на основі клітинних автоматів» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 82 сторінки пояснюючої записки, 39 рисунків, 13 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка програмного засобу для розпізнавання тексту на основі клітинних автоматів.

Розглянуто задачу розпізнавання машинного тексту. Процедура розпізнавання включає етап виявлення ознак символів та класифікації. Проаналізовано методи та програмні засоби розпізнавання образів, виокремлено їх переваги та недоліки.

Охарактеризовано особливості функціонування клітинних автоматів. Зазначено перспективність їх застосування для розв'язування задачі розпізнавання тексту. Сформовано поняття клітинних автоматів з мітками та послідовності клітинних автоматів.

Розроблено алгоритми виокремлення характерних ознак символів, що включають виявлення кінців, петель та перетинів символів. На прикладі літер українського алфавіту показано приклад роботи алгоритмів.

Розроблено архітектуру програмного забезпечення на основі клітинних автоматів з мітками. Наведено результати роботи розроблених алгоритмів на основ образів слова та сторінки тексту із різними шрифтами. Проведено порівняльний аналіз параметрів розпізнавання розробленого програмного забезпечення, штучних нейронних мереж та FineReader.

Ключові слова: РОЗПІЗНАВАННЯ ТЕКСТУ, КЛІТИННІ АВТОМАТИ, ІНФОРМАТИВНІ ОЗНАКИ СИМВОЛІВ, МЕТОДИ КЛАСИФІКАЦІЇ.

RESUME

Qualification thesis “Software for text recognition based on cellular devices” in the specialty 123 "Computer Engineering" "Bachelor" education degree contains 82 pages of explanatory notes, 39 figures, 13 tables, 2 appendixes. The volume of graphic material is 2 sheets of A3 format.

The aim of the qualification project is to develop software for text recognition based on cellular automata.

The task of machine text recognition is considered. The recognition procedure includes the stage of character feature detection and classification. Pattern recognition methods and software are analyzed, their advantages and disadvantages are highlighted.

Features of the functioning of cellular automata are characterized. The perspective of their application for solving the task of text recognition is indicated. The concepts of cellular automata with labels and sequences of cellular automata are formed.

Algorithms for distinguishing characteristic features of symbols, including detection of ends, loops and intersections of symbols are developed. An example of how algorithms work is shown on the example of the letters of the Ukrainian alphabet.

Software architecture based on cellular automata with labels is developed. The results of the developed algorithms based on the image of word and page of text with different fonts are presented. Comparative analysis of the recognition parameters of the developed software, artificial neural networks and FineReader are carried out.

Keywords: TEXT RECOGNITION, CELLULAR AUTOMATA, INFORMATIVE SYMBOLS, CLASSIFICATION METHODS.

ЗМІСТ

Вступ.....	9
1 Розпізнавання тексту як напрямок розпізнавання образів.....	12
1.1 Аналіз задачі розпізнавання тексту.....	12
1.2 Аналіз засобів розпізнавання образів.....	13
1.3 Характеристика клітинних автоматів	16
1.4 Постановка задачі кваліфікаційної роботи.....	25
2 Алгоритми розпізнавання тексту на основі клітинних автоматів.....	27
2.1 Алгоритм опрацювання текстового образу	28
2.2 Алгоритм виділення кінців та петель символів.....	31
2.3 Алгоритм виділення перетинів, кінців та петель символів.....	38
3. Програмна реалізація правил клітинних автоматів.....	42
3.1 Архітектура програмного забезпечення.....	42
3.2 Реалізація алгоритму розпізнавання тексту на основі клітинних автоматів.....	50
3.3 Порівняльний аналіз результатів розпізнавання.....	55
4 Техніко-економічне обґрунтування розробки проєкту.....	60
4.1 Визначення витрат на оплату праці	60
4.2 Розрахунок ціни проєкту	67
4.3 Визначення економічної ефективності роробки проєкту	70
Висновки	73
Список використаних джерел	74
Додаток А Світлокопії публікації.....	79
Додаток Д Довідка про використання	82

					КР.КІ.9499965.00.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розробив		Зяць А.-М. В.			Літ.	Арк.	Акрушів
Перевір.		Савка Н.Я.			8	82	
Консульт.		Савка Н.Я.			ЗУНУ.ФКІТ. КІз-41		
Н. Контр.		Мельник Г.М.					
Затвердив		Дубчак Л.О.					

ВСТУП

Обчислювальні системи появилися досить давно з метою зменшення трудомісткої роботи людини, що вимагає складних обчислювальних ресурсів. Однією з таких задач є навчити обчислювальну машину розпізнавати образи, зокрема, розробка алгоритму для автоматизованого розпізнавання тексту на основі характерних ознак.

Задача розпізнавання текстів є досить складною з теоретичної та практичної сторони, зважаючи на кількість мов. Наприклад, людина для цього задіює весь комплекс своїх знань, вмінь та досвіду. Вона виявляє текст із сукупності сигналів органів чуття, виділяє кожен символ, характерні ознаки символів й на цій підставі, застосовуючи власний досвід формує висновок щодо значення символу та тексту в цілому.

Комп'ютер здатен помиляється при розпізнаванні образів набагато частіше, а ніж людина. Варто зазначити, що не існує на сьогодні абсолютно точного методу розпізнавання тексту та символів за їхнім зображенням. Комерційні проекти використовують власні запатентовані методи, проте вони не забезпечують достовірного розв'язання вказаної задачі.

Часто якісне вирішення потребує комплексного підходу до задачі. Процедура розпізнавання тексту включає підзадачі: фільтрація від шуму, виділення символів із текстового зображення, виокремлення характерних ознак символів їх порівняння із збереженими еталонами. Таким чином, кожна конкретна задача містить чимало варіантів розв'язання, проте лише деякі із них є оптимальними.

Досить розвиненим напрямом науки на сьогоднішній день є клітинні автомати [2, 4, 6, 7, 19, 33]. Особливий інтерес до клітинних автоматів проявляється насамперед завдяки їх простоті: на основі простих правил клітинні автомати можуть моделювати складну поведінку. Окрім того, такі автомати – це ідеальний варіант для паралельних обчислень, вони можуть ефективно

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

використовуватися в багатопроцесорних системах або бути реалізовані апаратно, оскільки основна властивість їхніх правил – локальність та однорідність.

Переваги клітинних автоматів є досить ефективними в системах розпізнавання тексту. Простота та однорідність правил може дозволити створювати складні системи на базі декількох логічних або математичних елементів і досягти результату з меншими витратами як обчислювальних ресурсів, так і пам'яті. Для цього необхідно виконати аналіз клітинних автоматів та їх властивостей, виділити основні характеристики клітинних автоматів, необхідних у вирішенні задач розпізнавання тексту, та розробити відповідні алгоритми.

Зважаючи на вищезазначене, метою кваліфікаційної роботи є розробка програмного засобу розпізнавання тексту на основі клітинних автоматів.

Об'єктом дослідження є процеси розпізнавання зображень.

Предметом дослідження є методи на основі клітинних апаратів для розпізнавання тексту.

Практична цінність полягає у розробці програмного засобу розпізнавання машинного тексту із високою точністю із невеликими часовими та обчислювальними витратами.

Досягнення мети кваліфікаційної роботи вимагає вирішення нижчеперелічених задач:

- охарактеризувати задачу розпізнавання тексту;
- проаналізувати існуючі підходи;
- проаналізувати особливості клітинних автоматів;
- проаналізувати системи клітинних автоматів;
- розробити алгоритм розпізнавання тексту на основі клітинних автоматів;
- розробити архітектуру програмної системи;
- реалізувати алгоритм розпізнавання тексту;
- провести експериментальні дослідження роботи алгоритму;
- охарактеризувати техніко-економічні показники розробки проекту.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

За результатами роботи опубліковано тези доповіді на VII науково-практичній конференції «Інтелектуальні ком'ютерні системи та мережі» [44].
Копії публікації наведено у додатку А.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1 РОЗПІЗНАВАННЯ ТЕКСТУ ЯК НАПРЯМОК РОЗПІЗНАВАННЯ ОБРАЗІВ

1.1 Аналіз задачі розпізнавання тексту

Розпізнавання тексту як один із напрямків розпізнавання образів є досить складною задачею із теоретичної та практичної точки зору. Незважаючи на те, що таку задачу легко виконують живі організми та людина, досить складно побудувати та технічно реалізувати таку штучну систему, щоб ефективно виконувала процес розпізнавання. При цьому під розпізнаванням розуміють співвідношення зображень об'єкта тобто його образу, до набору його ознак. Прикладом систем образного розпізнавання може бути загалом розпізнавання тексту або окремих його символів, людських осіб, розпізнавання мови, біометричних даних, штрих-кодів, номерів машин тощо.

Для задач розпізнавання тексту відносять: оцифрування зображень тексту (сканування) з метою подальшої роботи із отриманим цифровим аналогом, опрацювання анкетних бланків, розпізнавання машинних номерів, написів на об'єктах У такому випадку, варто зазначити, що задача текстового розпізнавання залишається актуальною навіть попри існування різноманітних систем, оскільки стовідсоткової універсальної системи розпізнавання тексту не існує.

На вході системи для текстового розпізнавання є зображення із текстом у графічному файлі. На виході системи – сформований текст, що виділений із вхідного зображення. Задача текстового розпізнавання включає підзадачі та підпроцеси [42].

1. Зображення, що подають на вхід системи, повинно бути відфільтроване від шуму та приведено до вигляду, що дозволяє ефективно виділення символів та їх розпізнавання.

2. Системі необхідно поділити зображення на блоки тексту, опираючись на особливості його вирівнювання та розподілення по декількох колонках.

3. Зображення із текстом потрібно поділити спочатку на зображення рядків, пізніше – на зображення символів для обробки кожного символу окремо. Після

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізації зазначеного кроку кожна система розпізнавання працює на основі власних специфічних алгоритмів.

4. Образ символу може опрацьовуватись повністю на основі порівняння із наявними шаблонами. У іншому випадку відбувається виокремлення характерних ознак символу: відбір характеристик та класифікація їх за критеріями, що наявні у системі. Результатом четвертого кроку є можливий варіант літери. Проте, здебільшого, системи на цьому припиняють свою роботу і продовжують процес розпізнавання із застосуванням інших методів й уточнюючи одержаний результат.

5. Кінцевий результат розпізнавання може бути незадовільним. Для кращих результатів у систему вбудовують блок навчання, який уможливорює задання прикладів образів різних літер в конкретному шрифті. Після завершення процесу навчання передбачають кращу якість розпізнавання тексту.

Варто зазначити, що система текстового розпізнавання не завжди повинна відповідати усім описаним крокам, проте основні етапи процесу розпізнавання є загальними для будь-якого алгоритму.

1.2 Аналіз засобів розпізнавання образів

Існуючі системи текстового розпізнавання є комерційними продуктами, тому багато алгоритмів, на яких вони базуються, приховані від загального доступу. Принцип роботи систем ґрунтується на певних стратегіях, але здебільшого алгоритм розпізнавання узагальному вигляді включає послідовне висунення та перевірку гіпотез. При цьому порядок висунення керований знаннями про досліджуваний об'єкт, закладеними у програму та результатами тестування попередніх гіпотез [39].

Розглянемо метод розпізнавання за шаблонами. Програмна система Optical Character Recognition (OCR), зазвичай, працює із великим растровим образом сторінки зі сканера. Більшість систем при цьому має шаблони, розроблені для

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

різних контурів образу. Після декількох розпізнаних слів, програмна система визначає шрифт, який використано, та шукає для нього відповідні пари. В деяких випадках програми для визначення нового шрифту, використовують числові значення частин символу. Це в свою чергу може покращити ефективність процесу розпізнавання [26, 43, 50].

Розпізнавальна програма TypeReader ґрунтується на машинно-залежних алгоритмах з врахуванням підходу на основі шаблонів, що вимагає розробки застосовує 2100 різних варіантів символічних контурів [35].

Структурний підхід також характерний для задач розпізнавання. Система Caere OmniPage Professional ґрунтується на алгоритмі знаходження загальних специфічних ознак символів. Така система включає 100 різних алгоритмів для ідентифікації 100 символів: верхнього та нижнього регістрів від «А» до «Z», символів пунктуації та чисел. Кожен алгоритм шукає характерні «особливості» креслень типу точок, «острівів», «півостровів», прямих, дуг, відбитків. Експертні системи розглядають також горизонтальні та вертикальні проекції літерних відбитків та звертають увагу на особливості у побудованих кривих, підсумовуючи при цьому кількість пік селів темного кольору. Нечіткий текст може бути проблемою для структурних алгоритмів, оскільки відсутній піксель призводить до розбиття довгих штрихів або кривих, а додаткова пляма може формувати петлю [5, 32].

Контекстне розпізнавання також присутнє у програмах систем розпізнавання. Системи часто включають словники, які надають довідку, проте не стійкі до відмов, коли мають справу із даними, які не занесено у словник.

Нейромережам у системі розпізнавання виділено особливу увагу. Нейромережі – це певна архітектура зв'язаних елементів на основі функції перетворення сигналу та коефіцієнтів налаштування роботи системи. Частина елементів структури є вхідними, що приймають сигнали ззовні, частина – вихідними, що формують кінцеві результуючі сигнали. Сигнал, який проходить через нейромережу, перетворюється на основі формул на елементи мережі й на виході отримуємо результуючий сигнал [5, 15, 23, 35, 41, 43].

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Нейромережа служить як класифікатор у системах розпізнавання тексту, який можна навчати, налаштовуючи коефіцієнти на основі елементів мережі й, таким чином, отримати якомога достовірніший результат розпізнавання.

Нейромережі успішно застосовуються у системах текстового розпізнавання, проте наявні певні недоліки, які унеможливають їх широке універсальне застосування. Для побудови архітектури нейромережі, яка б забезпечувала розпізнавання кожного символу, вимагає побудови достатньо великої складної мережі зв'язків елементів, що потребує великих затрат пам'яті. Функції елементів нейромережі працюють із числами з плаваючою крапкою, що також вимагає суттєвих затрат ресурсів при розпізнаванні. Варто зазначити, що нейромережі навчають на всі випадки, що, не завжди гарантує достовірний результат. В той же час, функціонування нейромережі для розпізнавання тексту суттєво залежить від самої конфігурації та типів функцій, заданих для елементів. Це вимагає достатньої кваліфікації розробника, щоб досягти бажаного ефекту.

Розв'язок задачі розпізнавання тексту на базі клітинних автоматів має особливу актуальність у сьогоднішні час, скільки не вимагають суттєвий обчислювальних ресурсів. У працях [20, 29, 47] описано спроби розв'язування задачі розпізнавання тексту на основі клітинних автоматів. Проте окрім висловлювань про можливість побудови клітинного автомата для розпізнавання тексту у дослідженнях не йдеться.

Розпізнавання тексту – це комплекс задач, які необхідно виконати, щоб одержати кінцевий результат – текст. Чинні комерційні системи розпізнавання тексту користуються набором алгоритмів, які у комплексі дають дуже точний результат.

Одна частина системи розпізнавання може функціонувати на основі нейронної мережі, інша частина може використовувати переваги клітинних автоматів, а третина системи – накопичувати статистику та на її основі видавати результат.

Комплекс заходів та алгоритмів, безперечно, дозволить отримати кращий результату, ніж окремо взятий принцип чи алгоритм. Клітинні автомати можуть

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

бути частиною такого комплексу. Вони мають безперечні переваги, такі як можливість паралельного обчислення, легкість та простота правил, на основі яких вони побудовані, можливість реалізації багатьох достатньо складних алгоритмів опрацювання зображень.

1.3 Характеристика клітинних автоматів

Історія клітинних автоматів починається понад півстоліття тому. Багато джерел вказують, що клітинний автомат – це результат роботи Джона фон Неймана та Кондара Цузе, які на початку сорокових років 20 століття розробили ідею універсального обчислювального середовища для побудови, аналізу та порівняння характеристик алгоритмів [2, 4, 6, 7, 13].

На сьогоднішній день клітинні автомати знайшли широке застосування у всіх сферах людської діяльності: математика, фізика, біологія, хімія, механіка, криптографія, гідродинаміка, газодинаміка тощо [22, 25, 27, 34, 48]. Основним напрямом застосування клітинних автоматів є моделювання динамічних процесів, наприклад, рух натовпу [] чи поширення теплових потоків. Деякі вчені вважають, що клітинні автомати – це принцип, за яким сама природа діє у повсякденному житті.

Принцип клітинних автоматів ґрунтується на локальних взаємодіях. Всі елементи в будь-якій системі, діють за однаковими принципами. Спільно, результати роботи простих правил, на яких засновані взаємодії, що дозволяють отримувати складну поведінку. Джон фон Нейман дає таке визначення клітинним автоматам: « клітинні автомати є дискретними динамічними системами, поведінка яких повністю визначається термінами локальних залежностей. У значній мірі також це стосується великого класу безперервних динамічних систем, визначених рівняннями у частинних похідних. У цьому сенсі клітинні автомати в інформатиці є аналогом фізичного поняття «поля» клітинний автомат може вважатися як

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

стилізований світ. Простір, представлений рівномірною сіткою, кожен осередок чи клітина якої містить кілька бітів даних; час йде вперед дискретними кроками, а закони світу виражаються єдиним набором правил, скажімо, невеликою довідковою таблицею, за якою будь-яка клітина щокроку обчислює свій новий стан за станами її близьких сусідів.

Отже, закони системи є локальними та всюди однакові. «Локальний» означає, що для того, щоб дізнатися, що станеться тут за мить, досить подивитися на стан найближчого оточення: ніяка далекодія не допускається.

«Одинковість» означає, що закони скрізь одні й ті самі: можна відрізнити одне місце від іншого лише за формою ландшафту, а не за якоюсь різницею у законах» [47].

Формально клітинний автомат можна визначити як набір $\{G, Z, N, f\}$ [33], де G – метрика поля, у якому діє клітинний автомат;

Z – множина станів кожної клітини;

N – область клітини, що впливає стан цієї клітини;

f – правила клітинного автомата, які в математичному вигляді можуть бути записані $Z \times Z \mid N \mid Z$.

Властивістю клітинного автомата є: локальність правил, однорідність системи, завершення множини станів клітини, одночасність змін всіх клітин. Клітинні автомати можуть працювати у кількох вимірах: він може обробляти вектор станів клітин, площину чи тривимірну модель. У всіх випадках для візуалізації роботи автомата часто додають додатковий вимір – час, який дозволяє оцінити зміни станів поля у динаміці.

Існує кілька різновидів клітинних автоматів [44, 46]. Стандартні клітинні автомати урахуванням класичного визначення. Клітини поля такого автомата змінюють стан залежно від станів сусідніх клітин, і таким чином поле змінює свій стан. Приклади стандартних одномірних клітинних автоматів представлено на рисунку 1.1.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

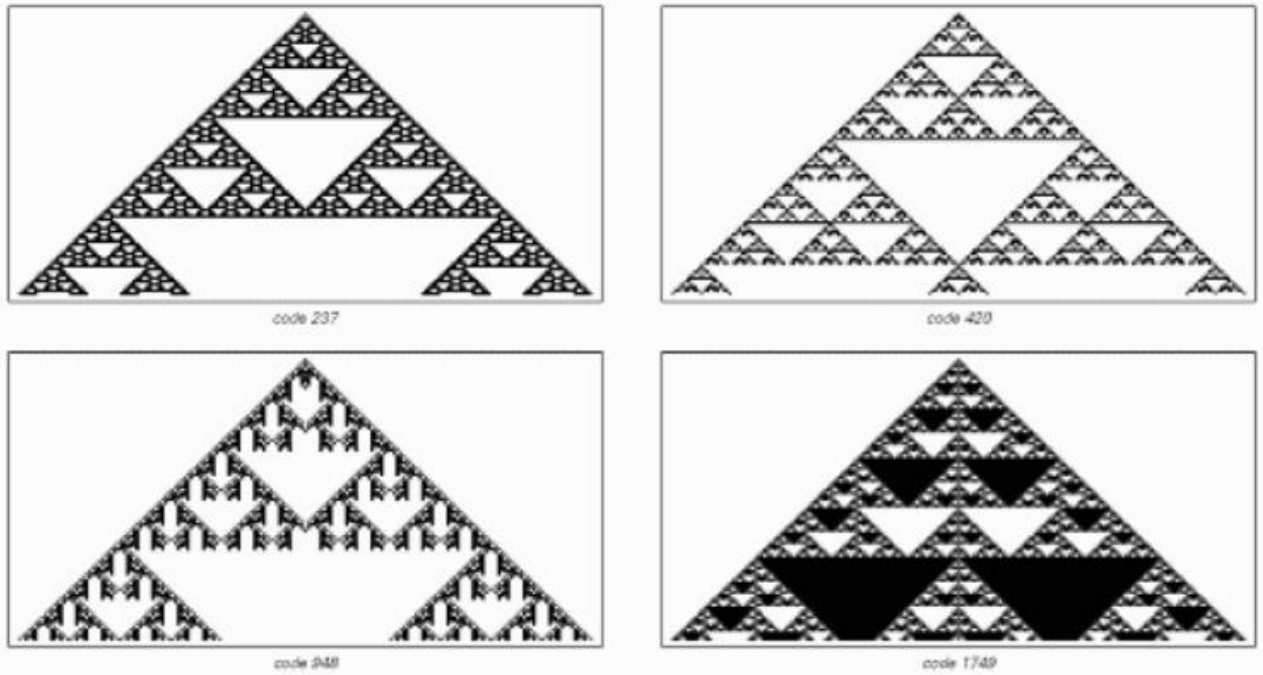


Рисунок 1.1 – Клітинні структури на основі стандартних клітинних автоматів

Мобільні клітинні автомати подібно до стандартних мають стан клітин, які змінюються з плином часу, але на відміну від них, додатково визначено одну активну клітини. Тільки активна клітина змінює стан у кожний момент часу. Приклад клітинної структури, породженої мобільним клітинним автоматично, зображено на рисунку 1.2. Тьюрингові машини на кшталт мобільного клітинного автомата мають лише одну активну клітину. Додаткова умова – визначення кількох станів активної клітини, як зображено на рисунку 1.3.

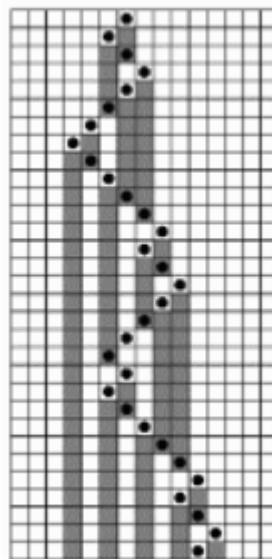


Рисунок 1.2 – Клітинна структура на основі мобільного клітинного автомата

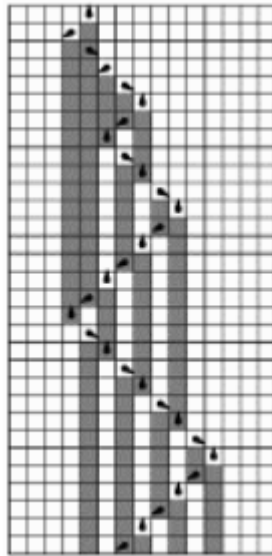


Рисунок 1.3 – Клітинна структура на основі машини Тьюринга

Система підстановок – це клітинний автомат, який на кожному кроці замінює клітину блоком інших клітин. Приклад автомата продемонстровано на рисунку 1.4.



Рисунок 1.4 – Клітинна структура на основі системи підстановок

Послідовні системи підстановки сканують послідовність точок на відповідність умовам кожного правила, при задоволенні правила відбувається підстановка та сканування починається наново, як зображено на рисунку 1.5.

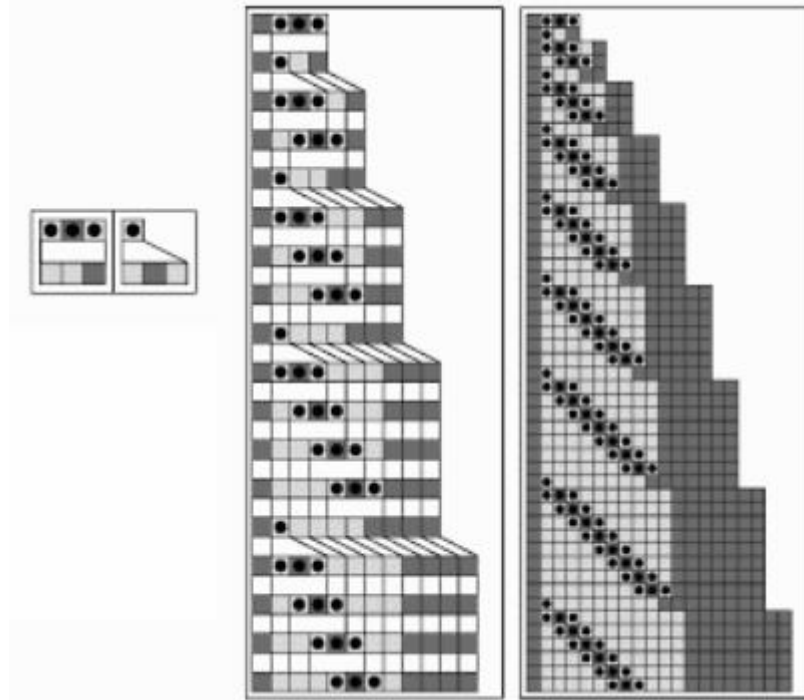


Рисунок 1.5 – Клітинна структура на основі послідовної системи підстановок

Правила, які видаляють елементи з початку ланцюжка клітин та додають інші клітини на кінець ланцюжків, називають системами тегів. Приклад подано на рисунку 1.6.

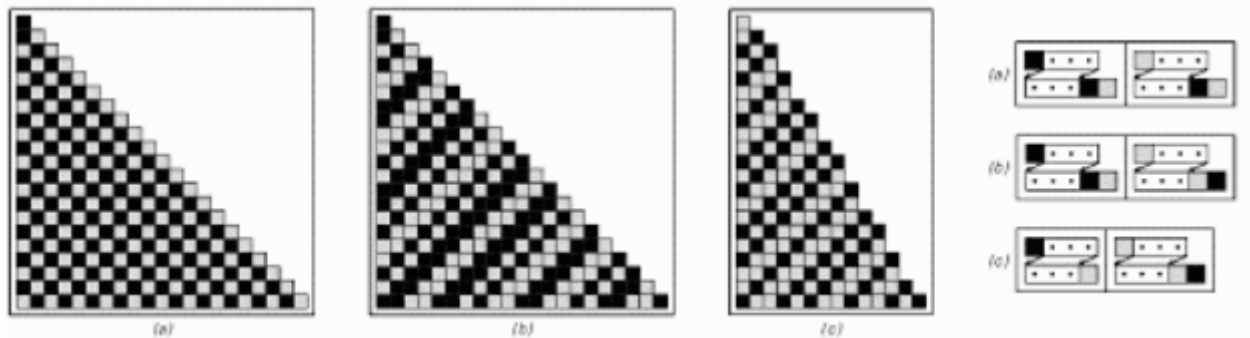


Рисунок 1.6 – Клітинна структура на основі системи тегів

Циклічність системи тегів може досягатися за рахунок послідовного використання різних систем правил. В результаті характер поведінки клітинного автомата стає специфічним, що видно із рисунку 1.7.

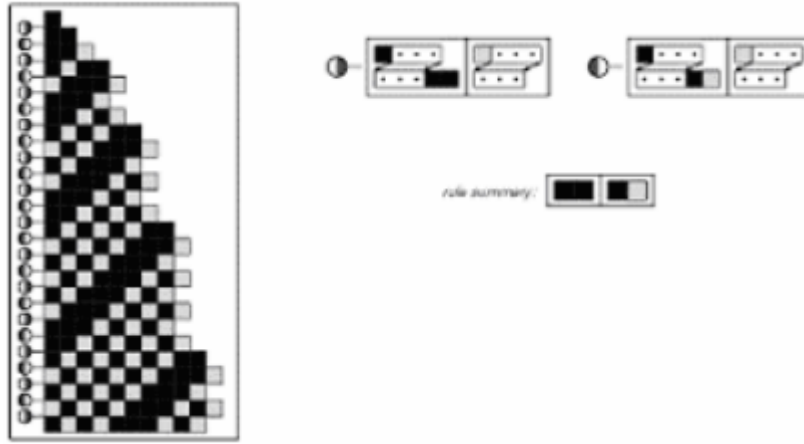


Рисунок 1.7 – Структура на основі циклічної системи тегів

Регістрові машини виконують задану програму. При успішному виконанні елемента програми відбувається заданий перехід, інакше перехід до наступного елемента. Приклад наведено на рисунку 1.8.

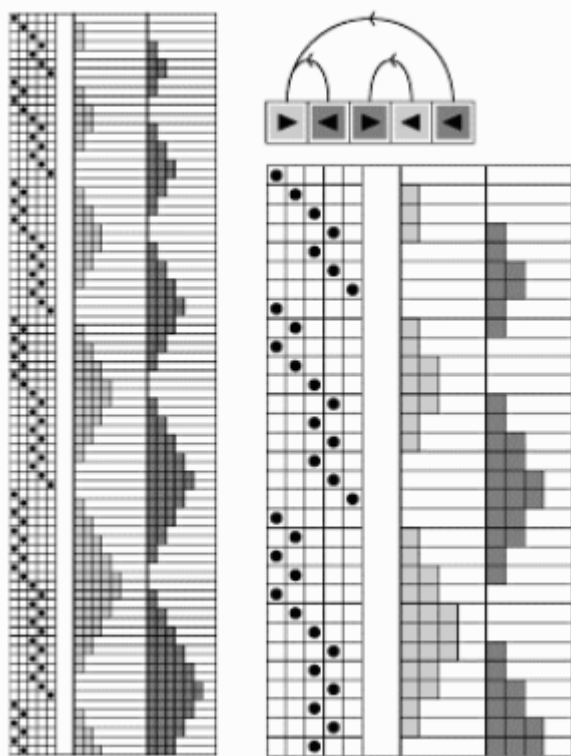


Рисунок 1.8 – Структура на основі реєстрової машини

У символній системі на кожному кроці кожен регіон, взятий у дужки, що трансформується за заданими правилами. В результаті цього формується послідовність. Приклад зображено на рисунку 1.9.

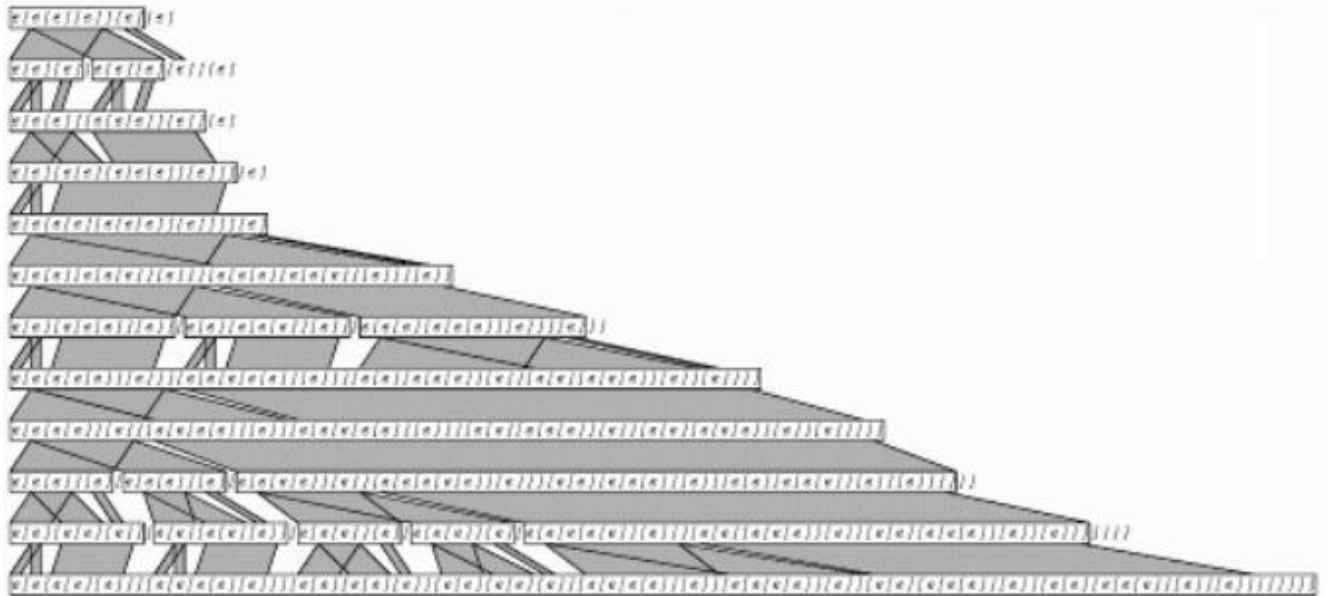


Рисунок 1.9 – Структура на основі символної системи

Причиною створення клітинних автоматів з мітками була необхідність досягнення заданих характеристик клітинних автоматів при розробці алгоритмів для текстового розпізнавання. Цей тип автоматів дозволяє ефективно реалізувати алгоритм виокремлення ознак символів. У процесі аналізу виявлено поняття пам'яті в теорії тьюрингових машин [19]. Воно визначає, що головка тьюрингової машини може зберігати деяке кількість даних, що використовуються правилами автомата машини для зміни стрічки машини або руху її головки. Це поняття при введенні деяких обмежень та переходу на теорію клітинних автоматів може бути також зведено до клітинних автоматів з мітками.

Принцип клітинної системи з мітками полягає в асоціації кожної клітини поля з одним або декількома мітками. Прикладом клітинного автомата з мітками може бути набір: поле, на якому визначено лише клітини чорного та білого кольорів, мітка, яка може бути присутня або відсутня у клітинах поля, та правила, що враховують наявність міток у сусідніх клітин. У разі визначення однієї мітки для клітинного автомата, кількість станів кожної клітини збільшується вдвічі у порівнянні з випадком без міток. У наведеному прикладі кількість станів кожної клітини дорівнює чотирьом. Правила клітинного автомата в такій системі виглядатимуть так, як представлено на рисунку 1.10.

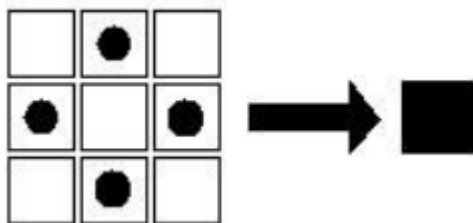


Рисунок 1.10 – Приклад правила клітинного автомата з мітками

У цьому прикладі використовується правило, засноване на восьми сусідах. Без міток кількість варіантів подібних правил (у разі визначення тільки двох кольорів клітин) буде $2^{10} = 1024$ (дев'ять клітин умови та одна – результату). Додаючи мітку, вийде $(2*2)^{10} = 1024^2 = 1048576$ варіантів правил. Зручніше, в даному випадку, розділяти двокрапкою нумерацію кольорів та міток. Тоді для правила, представленого на рисунку 1.10, номером може служити 1:340 (або у двійковому вигляді 0000000001:0101010100, де кожна цифра визначає стан клітини, клітини нумеруються зліва направо зверху вниз), де перше число визначає правило перетворення кольору, а друге число – правило зміни мітки.

Правило для поля, на якому визначено більше однієї позначки, нумерується аналогічним чином: для правила з двома мітками – x: y: z, де перше число визначає правило перетворення кольору, друге – перетворення однієї мітки, третє число – другої мітки.

Клітинний автомат із мітками дозволяє отримувати складні результати без зміни структури системи клітин. Наприклад, для заданого зображення, цей автомат дозволяє визначити деякі властивості зображення та виділити його характерні ознаки без зміни кольору.

Розглянемо формальний опис клітинного автомата мітками. Клітинний автомат із мітками – це набір $\{G, M, Z, N, f\}$.

1. G – кінцева дискретна метрична множина, що гарантує кінцеву відстань між клітинами.
2. M – кінцева множина міток, визначена для кожної клітини.
3. Z – кінцевий набір станів клітин.

4. N – кінцева множина, що визначає таку область клітини, що кожен елемент множини дозволяє визначити сусіда для кожної клітини. $|N|$ – кількість сусідніх клітин, які впливають на стан поточної.

5. f – правила клітинного автомата, що відповідають математичній функції переходів: $Z \times C \times Z \mid N \mid \times C \mid N \mid \rightarrow Z \times C$, де $C \subset M$.

Нижче показано, що клітинний автомат із мітками задовольняє вимоги класичного клітинного автомата. За визначенням клітинний автомат задається чотирма елементами $\{G^*, Z^*, N^*, f^*\}$, як зазначено вище. Елементи G та N клітинного автомата з мітками відповідають класичним елементам G^* та N^* .

Введемо $Z^\wedge = Z \times C$ в такий спосіб, що $|Z^\wedge| = |Z| \times |C|$. Множина Z^\wedge буде відповідати Z^* класичного визначення клітинного автомата, оскільки воно є кінцевим і визначає всі варіанти станів клітини автомата.

Аналогічно, $f^\wedge = Z^\wedge \times Z^\wedge \mid N \mid Z^\wedge$ буде відповідати f^* , оскільки враховуються всі стани на першому та другому часовому шарі. Таким чином, отриманий набір $\{G, Z^\wedge, N, f^\wedge\}$ клітинного автомата, в якому виконуються всі властивості: локальність правил (забезпечене N), однорідність системи на основі метрики G , кінцева множина станів клітини Z^\wedge , одночасність змін для всіх клітин, гарантоване набором правил f^\wedge .

Таким чином, клітинний автомат – це потужний інструмент виконання задач моделювання. Чітко визначивши всі його складові (метрику поля, стан клітин, кількість впливають на клітину сусідів та правила роботи автомата), можна вирішити величезну кількість задач, які мають не очевидні рішення [13]. На жаль, не всі задачі, вирішення яких можливе на основі клітинних автоматів уможлиблюють отримати рішення на основі їх невеликої кількості. Більш того, для вирішення частини задач може знадобитися послідовність з клітинних автоматів, кожен із яких вирішує спеціалізовану задачу.

Послідовність клітинних автоматів – це сукупність клітинних автоматів автоматів, кожен із яких заснований на специфічних правилах. Клітинні автомати запускаються у заданій послідовності та кожен черговий клітинний автомат

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

починає роботу на полі, який був сформований попереднім клітинним автоматично в послідовності.

Послідовність автоматів не обов'язково ускладнюватиме логіку роботи системи. Використання послідовності клітинних автоматів допомагає обходитися невеликим числом правил для кожного клітинного автомата під час вирішення нетривіальної задачі.

У разі використання клітинних автоматів з мітками в послідовності клітинних автоматів необхідно заздалегідь визначити однаковий набір M можливих міток клітин поля кожного автомата. Це допоможе підтримувати послідовність у працездатному стані.

Особливе місце займають мобільні клітинні автомати, які працюють на двовимірній системі, а також клітинні системи з мітками. Слід зауважити, що для досягнення бажаного результату розпізнавання варто застосовувати кілька типів клітинних автоматів.

1.4 Постановка задачі кваліфікаційної роботи

У результаті вище проведених досліджень, описано процес розпізнавання тексту, виділено основні складові цього процесу. Розпізнавання тексту це складна з точки зору формалізації, задача, яка потребує застосування засобів штучного інтелекту. Проаналізовано існуючі засоби та підходи до розпізнавання тексту й зазначено, що здебільшого вони не дають достовірного результату та характеризуються високою обчислювальною складністю. Охарактеризовано перспективність застосування методів клітинних автоматів до розпізнавання тексту, як простих структур, що можна використовувати комплексно. При цьому визначено основні властивості та принципи, на основі яких можна використовувати клітинні автомати при розпізнаванні тексту. Таким чином, узагальнено мету дослідження – розробка прогресивної системи розпізнавання

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

тексту із застосуванням клітинних автоматів для підвищення якості та достовірності розпізнавання.

При цьому необхідно розробити такі пункти:

- дослідити процес обробки зображення на основі клітинних автоматів;
- проаналізувати системи клітинних автоматів;
- розробити алгоритм розбиття текстового зображення на символи символів із використанням клітинних автоматів;
- визначити принцип виокремлення ознак символів на основі клітинних автоматів;
- розробити алгоритм виокремлення ознак символів;
- визначити можливість побудови клітинного автомата у процесі класифікація ознак символів.
- розробити архітектуру програмної системи;
- реалізувати алгоритм розпізнавання тексту;
- провести експериментальні дослідження роботи алгоритму;
- охарактеризувати техніко-економічні показники розробки проєкту.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

2 АЛГОРИТМИ РОЗПІЗНАВАННЯ ТЕКСТУ НА ОСНОВІ КЛІТИННИХ АВТОМАТІВ

Вміння розрізняти є основною властивістю людини, а також живих організмів інших видів. Опис описує об'єкт. Розпізнавання конкретних образів людиною розглядають як психофізіологічну задачу, яка пов'язана із певним процесом взаємодії особи із деяким фізичним показником. Якщо особа розпізнає образ, вона реалізує процедуру індуктивного висновку та встановлює зв'язок між на основі асоціацій між своїм сприйняттям та певними узагальненими поняттями чи "орієнтирами", встановленими виходячи з минулого досвіду.

В сутності розпізнавання образів зводиться до питання відносних шансів щодо того, що вихідні дані відповідають тому чи іншому з відомих множин статистичних сукупностей, що визначаються минулим досвідом людини та надають орієнтири та апріорні дані для розпізнавання. Тоді задачу розпізнавання образів розглядають, як задачу пошуку відмінностей між вихідними даними, причому не через ототожнення з окремими образами, але зі своїми сукупностями. Це ототожнення здійснюється на основі пошуку ознак на множині об'єктів, що утворюють певну сукупність.

Пошук ознак є одним із найважливіших етапів у процесі розпізнавання образів та, зокрема, символів. До процесу виокремлення вирішити кілька задач: фільтрація зображення тексту від шуму, приведення до стану, який дозволяє виконати умови алгоритмів розпізнавання, та виділити з нього окремі зображення символів.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

2.1 Алгоритм опрацювання текстового образу

Обробка зображення – це задача зміни його характеристик для того, щоб алгоритми, що беруть участь у розпізнаванні тексту, працювали якісніше з меншою кількістю помилок. Крім того, для збільшення продуктивності роботи клітинних автоматів, задіяних у розпізнаванні, дуже критичним елементом є кількість станів клітин (кольорів точок) зображення.

У кваліфікаційній роботі клітинні автомати, задіяні у розпізнаванні символів, функціонують на основі двох станів клітин, відповідно чорно-білий колір пікселів зображення. У процесі переведення зображення у чорно-білий стан складові символів точки слід виділити серед фону. Для такої задачі використовують клітинний автомат, в якому кожна клітин авідповідає точці зображення, а локальний радіус для клітини дорівнює нулю.

Автомат реалізує три правила: переводить колір кожної точки зображення у відтінок сірого; зафарбовує клітину чорним, якщо вона темніша деякого граничного кольору; зафарбовує клітину білим кольором, якщо вона світліша за встановлений граничний колір. На рисунку 2.1 представлено правила описаного клітинного автомата, створені на основі програми, детально описаної у розділі 3.

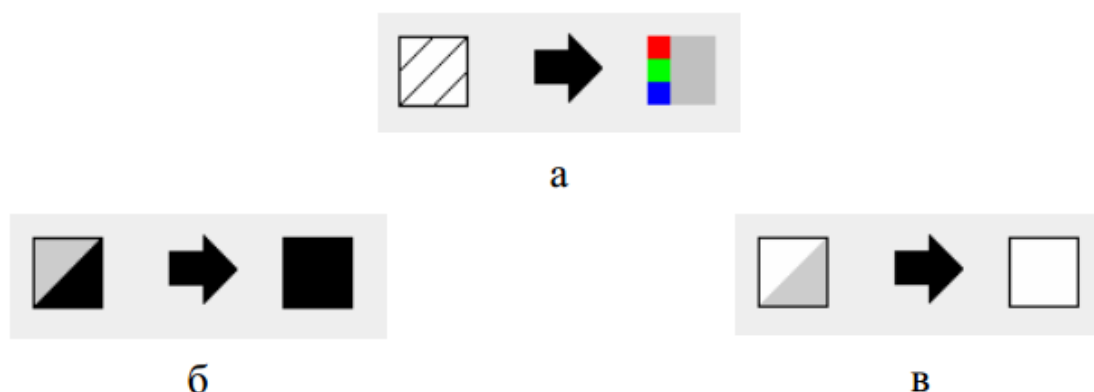


Рисунок 2.1 – Правила автомата переведення зображення тексту в чорно-білі кольори: а – переведення кольору точки у відтінок сірого, б – зафарбовування клітини чорним, якщо її відтінок темніший за граничний колір, в – зафарбовування клітини білим, якщо її відтінок світліший за граничний колір

Одна з невирішених на сьогоднішній день задача для систем розпізнавання тексту – його розподіл символи [32, 39, 42]. Найчастіше цю задачу вирішують у подібних системах в комплексі: попереднє розбиття текстового образу на окремі зображення символів і після цього визначається залежність між різними зображеннями з оцінкою відстані між ними. На етапі розпізнавання результати розбиття можуть уточнюватися з метою додаткового розбиття або об'єднання кількох отриманих зображень.

Уточнення можуть будуватися на основі контексту: якщо частина символів добре розпізнані, вони можуть вказати на нерозпізнаний, чи факт уточнення може ґрунтуватися на поганому розпізнанні отриманого зображення символу. Клітинні автомати ефективні при попередньому розбитті текстового зображення на зображення символів. В цьому випадку необхідно визначити два клітинних автомати з мітками, правила яких описано нижче.

1. Перший автомат ставить мітку на кожну чорну точку зображення у вигляді послідовно генерованих цілих чисел.

2. Другий автомат для кожної чорної точки переглядає локальну область одиничного радіусу і сам себе та виставляє мітку з мінімальною кількістю із області. При цьому стара мітка видаляється. Схеми автоматів представлено на рисунку 2.2.

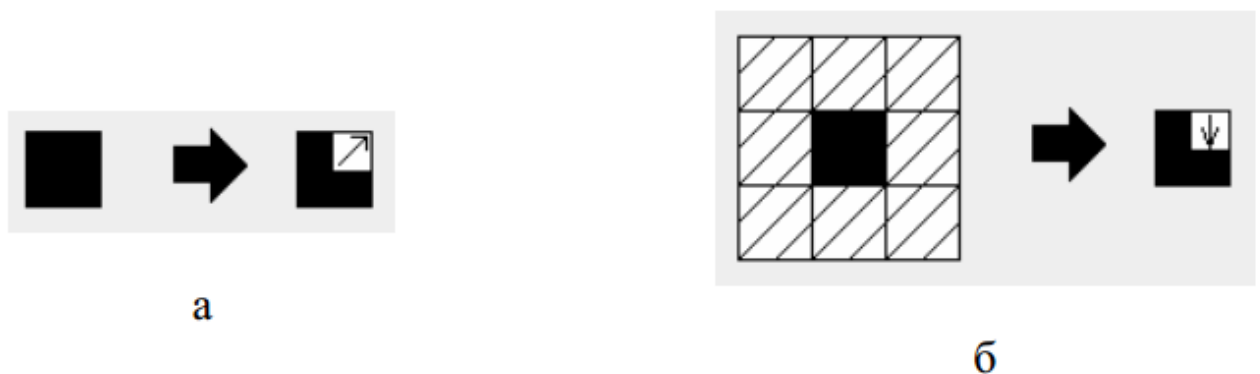


Рисунок 2.2 – Клітинні автомати на основі міток, що виділяють із зображення тексту символи: а – автомат генерації міток – чисел, б – автомат пошуку мітки з мінімальним числом

Після виокремлення ознак символів передбачається процес класифікації, який на підставі одержаних ознак визначить назву символу. Класифікація найчастіше ґрунтується на нейромережах, також застосовують статистичні методи, які на основі накопиченої інформації про ознаки можуть визначити символ.

Система розпізнавання передбачає наявність блоку навчання. Навчання системи безпосередньо пов'язане з класифікацією, воно дозволяє змінювати та виправляти коефіцієнти на основі асоціації результату класифікації з назвою символу. Для нейромереж та статистичних методів навчання проводиться у різний спосіб [41, 43].

Клітинні автомати також застосовують для класифікації ознак. Ідеєю класифікації при цьому є створення характерного клітинного автомата для кожної ознаки та її корекція з урахуванням певних ознак у процесі навчання. Ця концепція вимагає вивчення та подальших досліджень.

2.2 Алгоритм виділення кінців та петель символів

Перший алгоритм полягає у тому, що від верхнього краю символу вздовж точок, що становлять цей символ, пускається «хвиля». Ця «хвиля» може розділитися на складові, повторюючи контур зображення символу. У певний момент складові «хвилі» можуть зустрітися або загаснути на кінці символу. На рисунку 2.4 показано напрямок руху "хвилі" вздовж зображення символу "А" і показані місця старту "хвилі", згасання на кінцях символу та у місці зустрічі її двох складових.

У поняття «хвилі» в даному алгоритмі вкладається декілька складових. "Фронт хвилі" – точки символу, які пересуваються від одного кінця зображення символу до іншого. "Шлейф хвилі" – це точки зображення, в яких у попередній час знаходився «фронт хвилі». «Точки пройденого шляху» – це точки зображення,

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

де був присутній "фронт хвилі", а потім і "шлейф хвилі", у цих точках процес не поновлюється.

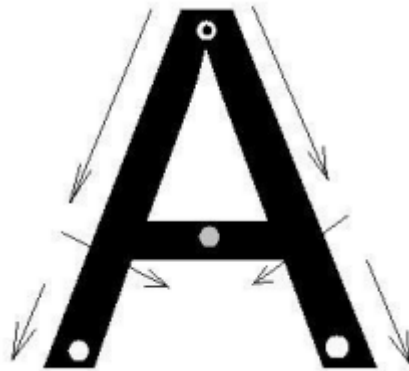


Рисунок 2.4 – Напрямок поширення «хвилі» вздовж символу «А» з відзначенням позицій кінців символів та петель

У початковий момент часу всі точки символу не позначені мітками. Перша точка хвилі позначається і алгоритм починає свою роботу. Алгоритм заснований на ідеї, що в процесі проходження хвилі в якийсь момент "фронт хвилі" згасне, тоді як шлейф все ще буде присутнім. Ця подія може статися лише на кінці символу або на місці зустрічі двох складових хвилі. Позиція шлейфу хвилі в цей момент запам'ятовується. Додатково подія зустрічі двох складових хвилі реєструється на підставі факту, що шлейфи двох складових хвилі у момент зустрічі не зв'язані між собою. Таким чином, запам'ятовується позиція петлі символу.

Схему роботи послідовності клітинних автоматів представлено на КР.КІ.07158/19.00.00.000.А1, у овальних блоках вказані клітинні автомати. Наведемо опис алгоритму на основі послідовності клітинних автоматів із мітками. Крім клітинних автоматів у послідовності є додаткові керуючі елементи. Стан клітини визначається її кольором. Крім кольору, клітина може містити одну або більше міток, які поділяються між собою кольорами.

Крок 1. На вхід послідовності клітинних автоматів надходить зображення символу.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Крок 2. Верхня ліва чорна клітинка зображення позначається червоною і сірою міткою.

Крок 3. Автомат "створення шлейфу": червона мітка замінюється синьою міткою.

Крок 4. Автомат створення фронту: чорні клітини без сірої мітки поряд із клітинами із сірою міткою позначаються сірою та червоною мітками.

Крок 5. Одна клітина поруч із червоною міткою замінює синю мітку зеленою.

Крок 6. Автомат «прохід по шлейфу»: всі клітини поруч із клітинами з зеленими мітками замінюють сині мітки зеленими.

Крок 7. Автомат «прохід фронтом»: всі клітини з червоною, але без зеленої мітки, що знаходяться поруч із клітинами із зеленою міткою, позначаються зеленою міткою; якщо клітина з червоною та зеленою міткою знаходиться поруч із клітиною із синьою міткою, то така клітина позначається помаранчевою міткою.

Крок 8. Якщо, починаючи з кроку 5, автомати не змінили стану однієї клітини, то перейти крок 9, інакше перейти крок 5.

Крок 9. Автомат «видалення зелених міток»: усі зелені мітки у клітин видаляються.

Крок 10. Якщо, починаючи з кроку 3, автомати не змінили стану однієї клітини, то перейти крок 11, інакше перейти крок 3.

Крок 11. Верхня ліва чорна клітка зображення позначається синьою міткою.

Крок 12. Автомат «видалення нижніх зайвих міток»: якщо над клітиною з синьою або помаранчевою міткою знаходиться клітина з такою ж міткою, то в цій клітці цю мітку видалити.

Крок 13. Автомат «видалення зайвих міток праворуч»: якщо зліва від клітини з синьою або помаранчевою міткою знаходиться клітина з такою ж міткою, то в цій клітині цю мітку видалити.

Крок 14. Якщо, починаючи з кроку 12, автомати не змінили стану однієї клітини, то завершити послідовність, інакше перейти на крок 12.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

На виході алгоритму деякі клітини зображення позначені мітками. Синя мітка означає, що в цьому місці символу кінець символу, а помаранчева мітка означає символ петлі.

У процесі роботи алгоритму клітини з червоними мітками можуть бути проінтерпретовані як фронт хвилі, сині клітини – як шлейф хвилі, а сірі клітини – як клітини пройденого шляху.

Нижче наведено формальні правила переходів клітинних автоматів з мітками. Для описуваних клітинних автоматів визначено п'ять різних міток: червоного, синього, сірого, зеленого та помаранчевого кольорів. Це означає, що стан клітини складається з шести складових: кольору та наявності/відсутності п'яти міток. Номер правила позначається шістьма числами, які означають стан клітини та присутність/відсутність мітки. Позиція кожної мітки в номері визначається на основі наведеного вище списку.

Деякі правила автоматів у схемах містять замість кольору символу штрихи – це означає набір правил із різними комбінаціями кольору/наявності міток клітини.

1. Автомат "створення шлейфу" має околицю N радіусу нуль – на стан клітини впливає лише сама клітина. Автомат містить лише одне правило: 3:2:1:1:0: 0. Схема правила вказана на рисунку 2.5.



Рисунок 2.5. – Схема правила клітинного автомата "створення шлейфу"

2. Автомат "створення фронту" має локальну область з радіусів в одну клітину і містить декілька правил. Схеми правил представлено на рисунку 2.6.

3. Автомат «прохід по шлейфу» заснований на області із сусідів одиничного радіусу і містить вісім правил, схеми яких зображено на рисунку 2.7.

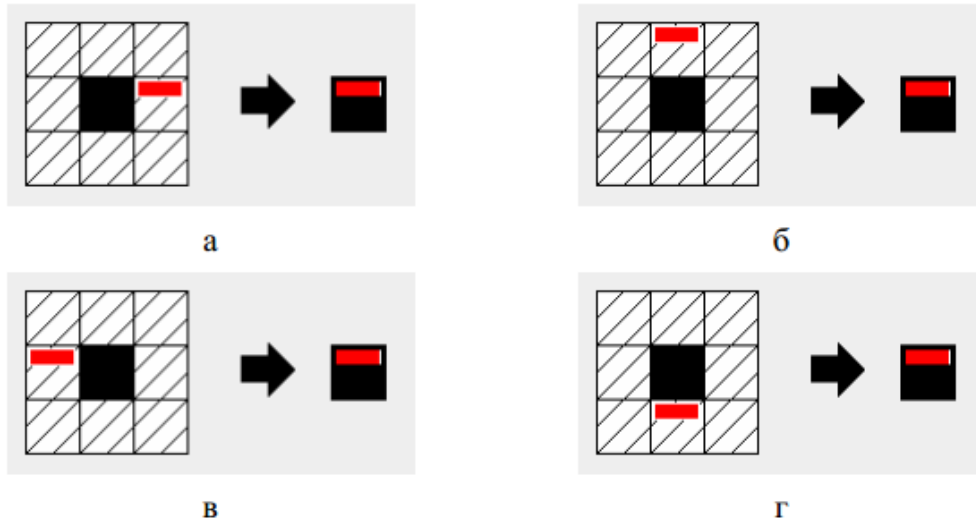


Рисунок 2.6 – Схеми правил клітинного автомата «створення фронту»

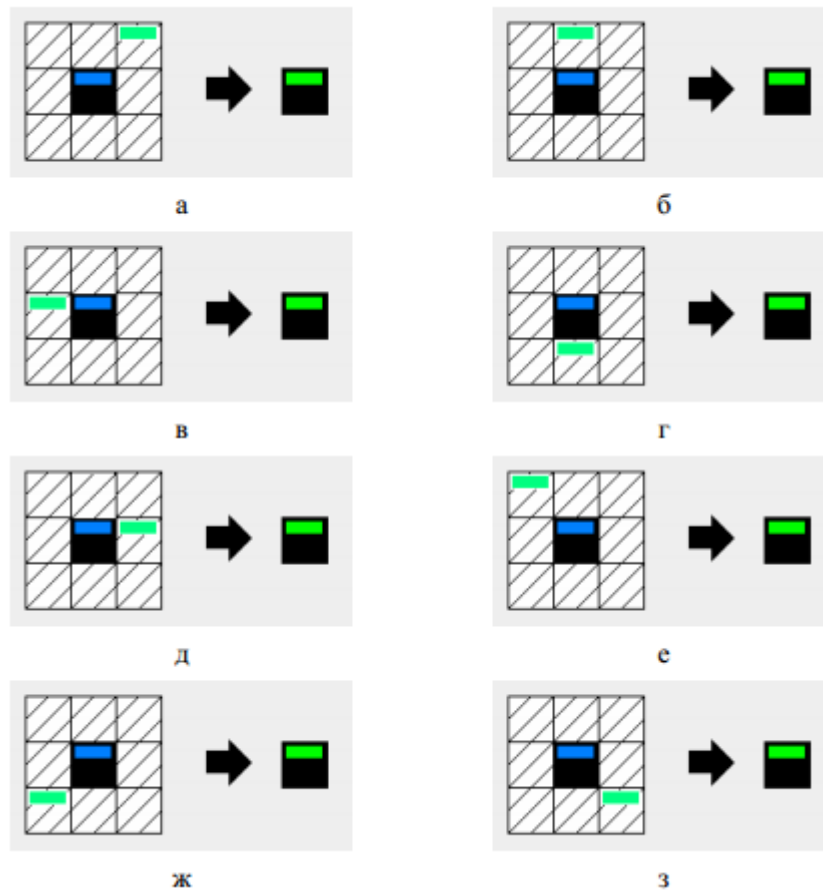


Рисунок 2.7 – Схеми правил клітинного автомата "прохід по шлейфу"

4. Схеми автомата «прохід фронтом» представлено на рисунку 2.8., автомат містить 12 правил.

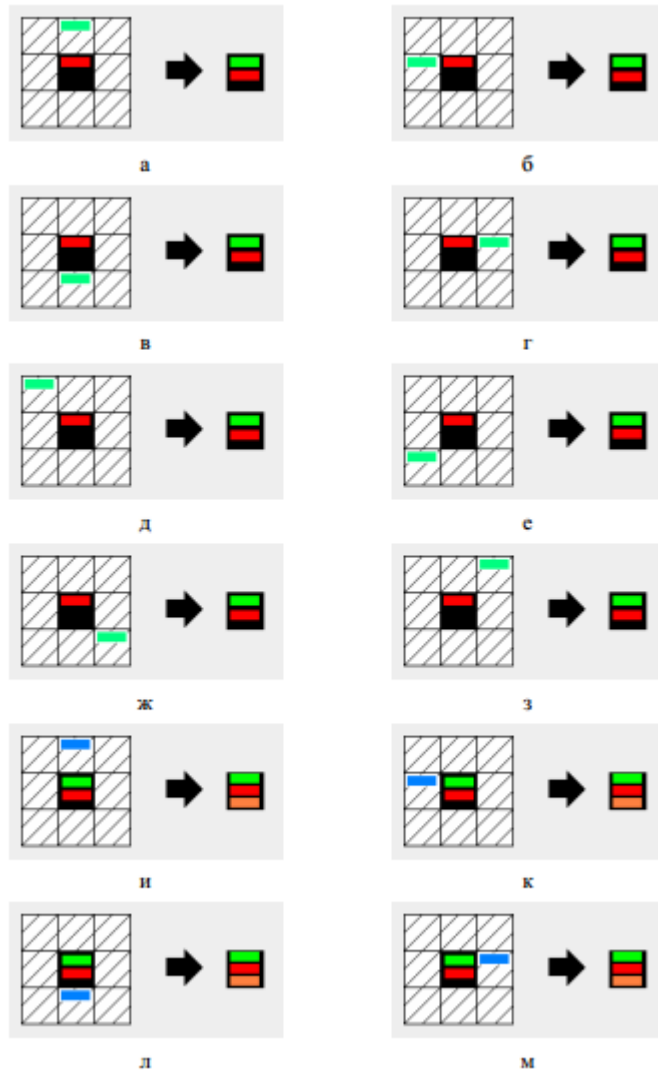


Рисунок 2.8 – Схеми правил клітинного автомата «прохід фронтом»

5. Автомат "видалення зелених міток" аналогічно автомату "створення" шлейфу» містить одне правило і має радіус локальної області – нуль. Схему правила автомата продемонстровано на рисунку 2.9.

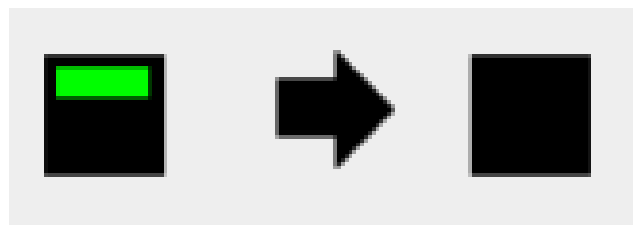


Рисунок 2.9 – Схеми правил клітинного автомата «видалення зелених міток»

6. Схеми правил клітинного автомата «видалення нижніх зайвих» міток» винесено на рисунок 2.10.

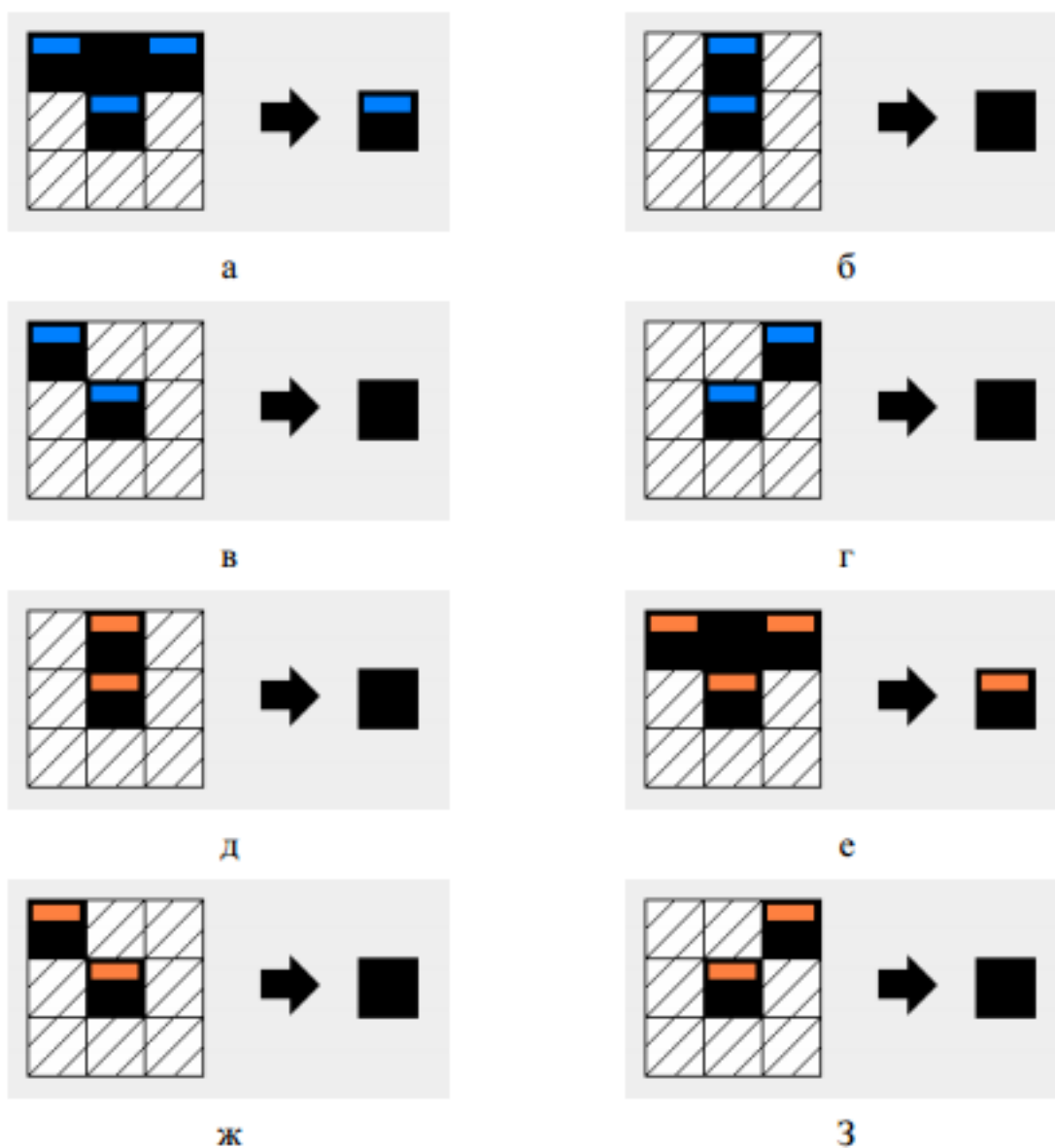


Рисунок 2.10 – Схеми правил автомата «видалення нижніх зайвих міток»

7. Схеми правил автомата «видалення зайвих міток праворуч» вміщено на рисунку 2.11.

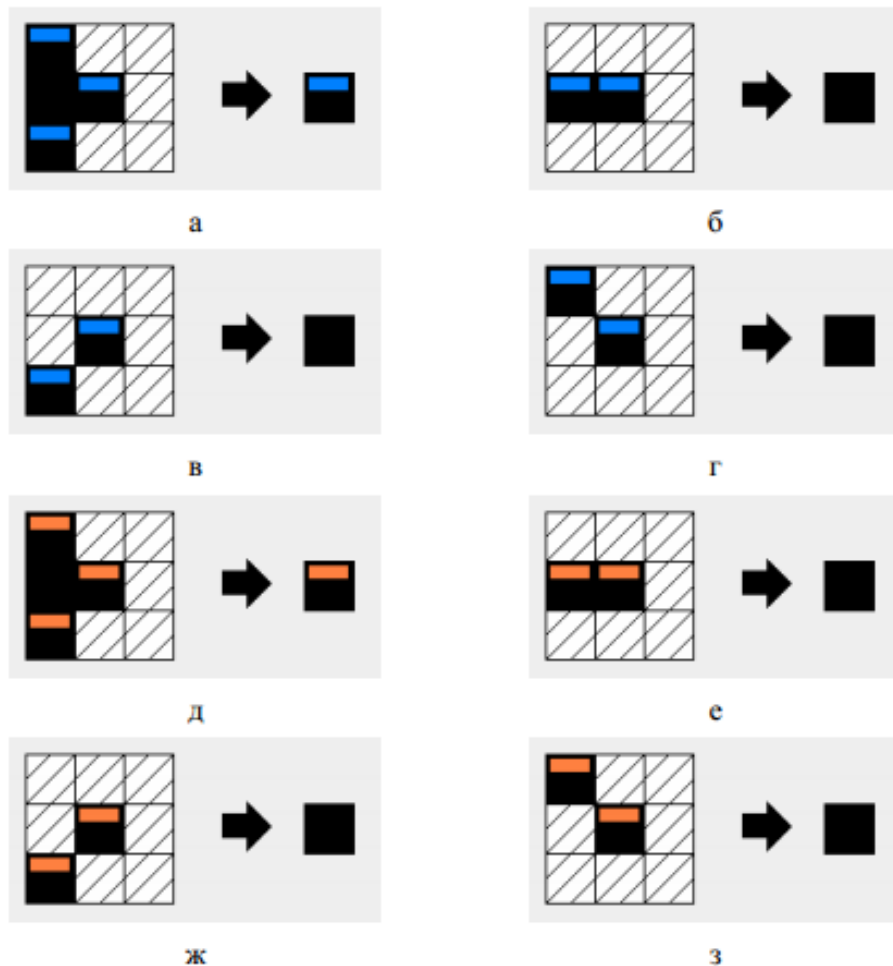


Рисунок 2.11 – Схеми правил клітинного автомата «видалення зайвих міток справа»

Таким чином, алгоритм виділення кінців та тель символів – це один із способів розпізнавання виокремлення ознак символів для текстового розпізнавання. У наступному підрозділі опишемо алгоритм роботи клітинних автоматів з мітками, що враховує також перетин.

2.3 Алгоритм виділення перетинів, кінців та петель символів

Друга стратегія є удосконаленою першою і розвиває її. У процесі поширення «хвилі» у момент, коли визначається клітина кінця символу або

клітина зустрічі двох складових «хвилі», генерується у відповідь «хвиля» – «ехо» вздовж уже пройдених клітин. «Відлуння» проходить шлях у зворотному напрямку і відзначає клітини, яких початкова хвиля була поділена на складові. Це повинно дозволити знайти позиції перетинів відрізків, з яких складаються символи.

Друга стратегія дозволяє визначити позиції перетинів ліній у зображеннях символів. Розповсюдження хвилі можна побачити із рисунка 2.12.

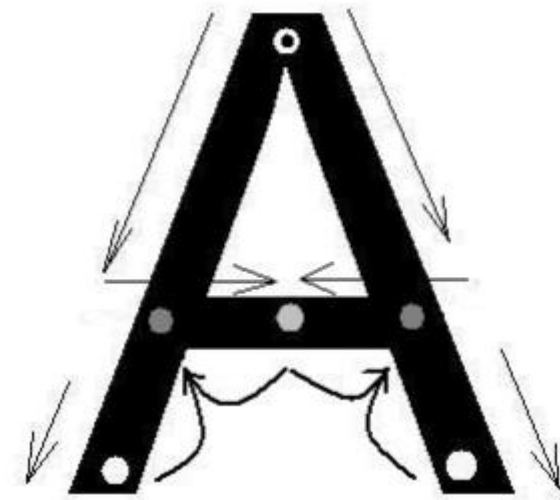


Рисунок 2.12 – Напрямок поширення «хвилі» вздовж символу «А» з відзначенням позицій кінців символів та петель, поширення «еха» з відзначенням позиції перетинів

Блок –схему алгоритму роботи послідовності клітинних автоматів показано на КР.КІ.07158/19.00.00.000.А2. Наведемо неформальний опис алгоритму на основі послідовності клітинних автоматів із мітками.

Крок 1. На вхід послідовності клітинних автоматів надходить зображення символу.

Крок 2. Верхня ліва чорна клітинка зображення позначається червоною і сірою міткою.

Крок 3. Автомат "створення шлейфу": червона мітка замінюється синьою міткою.

Крок 4. Автомат створення фронту: чорні клітини без сірої мітки поряд із клітинами із сірою міткою позначаються сірою та червоною мітками.

Крок 5. Одна клітина поруч із червоною міткою змінює синю мітку у зелений колір.

Крок 6. Автомат «прохід по шлейфу»: всі клітини поруч із клітинами з зеленими мітками замінюють сині мітки зеленими.

Крок 7. Автомат «прохід фронтом»: всі клітини з червоною, але не з зеленою міткою, що знаходяться поруч із клітинами із зеленою міткою, позначаються зеленою міткою; якщо клітина з червоною та зеленою мітками знаходиться поруч із клітиною із синьою міткою, то така клітина позначається помаранчевою міткою.

Крок 8. Якщо, починаючи з кроку 5, автомати не змінили стан однієї клітини, то перейти крок 9, інакше перейти крок 5.

Крок 9. Автомат «видалення зелених міток»: усі зелені мітки у клітин забираються.

Крок 10. Якщо в одній із клітин є синя мітка, то перейти на крок 11, інакше крок 20.

Крок 11. Автомат "створення еха": синю мітку клітин замінити на коричневий і блакитний колір, до червоної позначки символів додати коричневу мітку.

Крок 12. Автомат "створення шлейфу еха": коричнева мітка замінюється жовтою міткою.

Крок 13. Автомат «створення фронту еха»: чорні клітини із сірою міткою, але без фіолетової мітки поруч із клітинами з фіолетовою міткою позначаються фіолетовою та коричневою мітками.

Крок 14. Одна клітина поруч із коричневою міткою змінює жовту мітку на малиновий колір.

Крок 15. Автомат «прохід по шлейфу еха»: всі клітини поруч із клітинами з малиновими мітками замінюють жовті мітки малиновими.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Крок 16. Автомат «прохід фронтом еха»: всі клітини з коричневою, але не з малиною міткою, що знаходяться поряд з клітинами з малиною міткою, що позначаються малиною міткою; якщо клітина з коричневою та малиною мітками знаходиться поруч із клітиною з жовтою міткою, то така клітина позначається пурпуровою міткою.

Крок 17. Якщо, починаючи з кроку 14, автомати не змінили стану однієї клітини, то перейти крок 18, інакше перейти крок 14.

Крок 18. Автомат «видалення малинових міток»: всі малинові мітки у клітин забираються.

Крок 19. Якщо, починаючи з кроку 12, автомати не змінили стану однієї клітини, то перейти крок 20, інакше перейти крок 12.

Крок 20. Якщо, починаючи з кроку 3, автомати не змінили стану однієї клітини, то перейти крок 21, інакше перейти крок 3.

Крок 21. Верхня ліва чорна клітка зображення позначається синьою міткою.

Крок 22. Автомат «видалення нижніх зайвих міток»: якщо над кліткою з синьою або помаранчевою міткою знаходиться клітина з відповідно синьою або помаранчевою міткою, то в цій клітині цю мітку видалити.

Крок 23. Автомат «видаення зайвих міток праворуч»: якщо ліворуч від клітки з синьою або помаранчевою міткою знаходиться клітина з відповідно синьою або помаранчевою міткою, то в цій клітині цю мітку видалити.

Крок 24. Якщо, починаючи з кроку 22, автомати не змінили стану однієї клітини, то завершити послідовність, інакше перейти на крок 22.

Описаний алгоритм може здатися, на перший погляд, громіздким, але, насправді, він містить дві складові, аналогічні першому алгоритму. Правила автоматів даного алгоритму аналогічні правилам автоматів першого алгоритму виокремлення ознак символів.

Перевагою такого алгоритму вважається виділення більшої кількості ознак символів, ніж за допомогою першого алгоритму. Однак, другий алгоритм виконує більше кроків і тому працює довше.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРАВИЛ КЛІТИННИХ АВТОМАТІВ

Для проведення досліджень із застосування клітинних автоматів, клітинних автоматів з мітками та послідовностей клітинних автоматів розроблено програму мовою програмування Java. Архітектура програми дозволила створювати, зберігати та завантажувати клітинні автомати з мітками та без них, а також послідовності клітинних автоматів.

Нижче описано структуру основних частин програми, які безпосередньо пов'язані з роботою клітинних автоматів. Кожна сутність у програмі на основі принципів об'єктно-орієнтованого проектування представляється у вигляді класу з набором властивостей та дій, що виконуються нею. Наприклад, клітинний автомат, правило та клітина клітинного автомата є класами.

3.1 Архітектура програмного забезпечення

Програмна система створена для досліджень роботи різних клітинних автоматів. Її розробку здійснено з урахуванням доповнення окремих компонентів, тому й архітектура програми ґрунтується на модульному принципі. У програмі крім блоків, які відповідають за реалізацію клітинних автоматів та їх послідовностей, включені допоміжні елементи, що спрощують роботу з нею. Ці елементи: блок запису та завантаження, мовний блок.

У програмі є можливість запису та завантаження різних сутностей: картинок, клітинних автоматів, клітинних послідовностей автоматів, результати навчання системи. Основний принцип запису та завантаження – реалізація інтерфейсу XMLTransferable всіма збереженими в файлах елементами та наявність у них конструктора, який сприймає запис у форматі XML. Діаграму класів, що реалізують цей принцип, показано на рисунку 3.1. Програма

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

передбачає різні мови інтерфейсу. Для додавання додаткової мови, достатньо реалізувати інтерфейс локалізації.

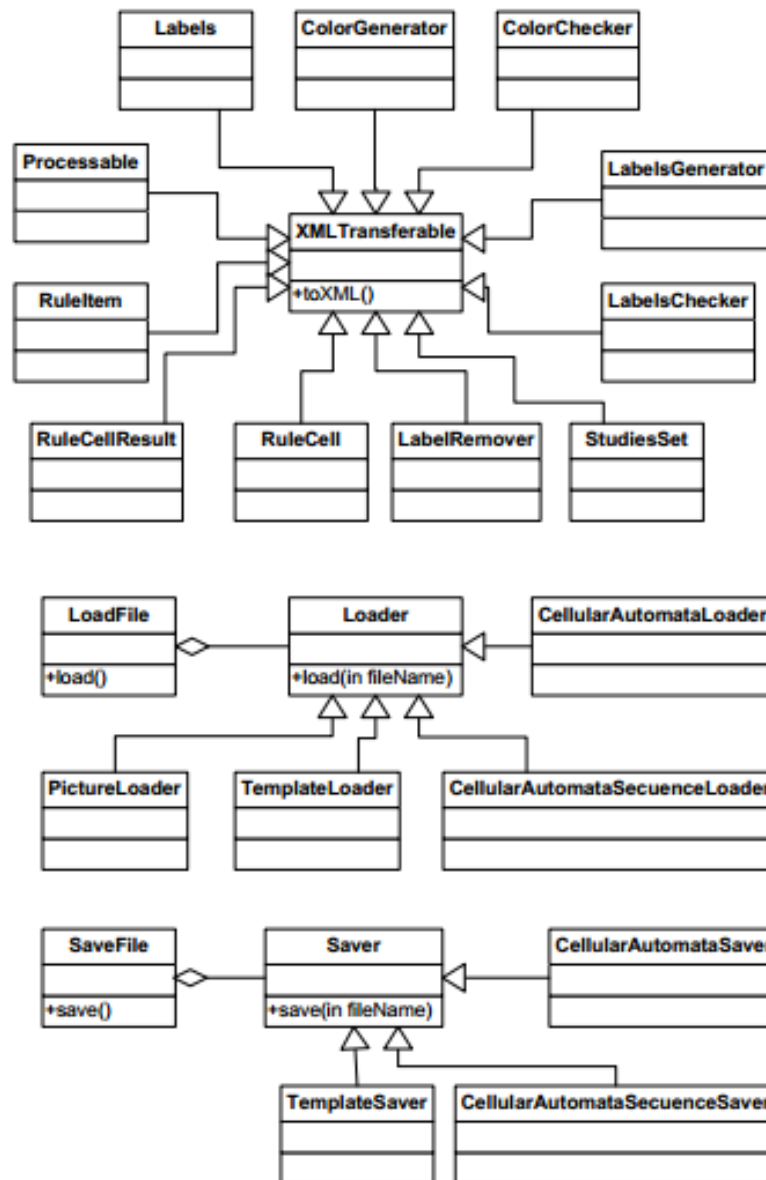


Рисунок 3.1 – Діаграма класів блоку запису і загрузки

Клас XMLTransferable – інтерфейс перетворення об'єктів у XML формат. LoadFile – клас для завантаження об'єкта з файлу. Loader – інтерфейс для завантаження об'єкта певного типу з файлу. SaveFile – клас для збереження об'єкта у файл. Saver – інтерфейс для збереження об'єкта певного типу у файл.

Клітинний автомат можна визначити як набір правил, причому у клітинного автомата має бути визначено хоча б одне правило. Тому автомат може містити

правило за замовчуванням і, можливо, інші правила. Кожне правило – це сукупність перевірки стану клітини та сусідніх клітин та визначення нового стану клітини. Якщо визначено локальний радіус, на якому сусідні клітини впливають на стан даної, рівний одиниці, то правило буде містити дев'ять клітин, які необхідно перевірити на задоволення умов правила, і результат дії правила – результуюча клітина.

Перевірка клітини на задоволення умов правила поділяється на перевірку стану клітини – його кольору та на перевірку наявності чи відсутності певних міток. Перевірка кольору клітин може відбуватися кількома способами: клітина повинна мати чітко визначений колір, колір темніший цього, колір світліший від даного, колір на відріжку і будь-який колір.

Аналогічно перевірка міток задається перевіркою на наявність всіх чи однієї із заданих міток, відсутністю всіх або певних міток. Визначення нового стану клітини означає визначення кольору та міток, які будуть визначені для клітини як результат виконання правила. Колір клітини може бути залишений тим самим або чітко заданим. Аналогічно, мітки можуть бути залишені колишніми або додані нові. Додатково мітки, знайдені у клітині на етапі перевірки в рамках цього правила можуть бути видалені. Діаграму класів блоку клітинного автомата представлено рисунку 3.2.

Опишемо, який клас за яку функцію відповідає. Cellular Automata – базовий клас, що описує роботу клітинного автомата. RuleItem – клас, що описує правило клітинного автомата. RuleCell – клас, що описує умови, накладені на клітину, у рамках правила клітинного автомата. ColorChecker – абстрактний клас, що перевіряє колір клітини. AnyColorChecker – клас, що перевіряє колір клітини: приймає будь-який колір клітини. FixedColorChecker – клас, що перевіряє колір клітини: приймає фіксований колір клітини. BetweenColorsChecker – клас, що перевіряє колір клітини: приймає колір клітини між двома заданими.

DarkerThenColorCheker – клас, що перевіряє колір клітини: приймає колір клітини темніший від заданого. LighterThenColor – клас, що перевіряє колір клітини: приймає колір клітини світліше за задане. LabelChecker – абстрактний

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

клітини: не змінює колір клітини. FixedColorGenerator – клас для генерації кольору клітини: виставляє заданий колір клітини. MiddleGreyColorGenerator – клас для генерації кольору клітини: виставляє середній арифметичний колір клітини. LabelGenerator – абстрактний клас для генерації міток клітини. DoNothingGenerator – клас для генерації міток клітини: не додає мітки. AddLabelGenerator – клас для генерації міток клітини: додає мітки до клітини. IncreaseGenerator – клас для генерації міток клітини: виставляє мітку, наступну з невикористаних. MinimumOfGenerated – клас для генерації міток клітини: виставляє мітку, мінімальну із сусідніх. LabelRemover – абстрактний клас для видалення міток клітини. У таблиці 3.1 наведено опис полів базового класу клітинного автомата

Таблиця 3.1 – Опис базового класу роботи клітинного автомата

Поле/метод	Опис
String name	Ім'я клітинного автомата
String description	Опис клітинного автомата
ContinueCondition	Умови продовження роботи автомата (автомат не буде зупинятися, доки виконується умова)
RuleItem defaultRuleItem	Правило клітинного автомата по замовчуванню
List<RuleItem> ruleItems	Правила клітинного автомата
void initialize (AutomataSequence automataSequence)	Ініціалізація клітинного автомата у на початку роботи послідовності клітинних автоматів
ProcessResult process (MatrixElement[][] source)	Запуск клітинного автомата на полі, заданому матрицею станів клітин
ProcessResult processStep (MatrixElement[][] source)	Запуск кроку клітинного автомата на полі, заданому матрицею станів клітин

Моделююча програма необхідна для запуску послідовності клітинних автоматів. Реалізація цього процесу передбачає наявність можливості запуску всієї послідовності в цілому, запуску окремого клітинного автомата та кроку клітинного автомата. Крім клітинних автоматів у послідовності можуть бути включені інші елементи: елемент розділення зображення тексту на окремі зображення символів, елемент перевірки умов та переходу за даною умовою на інший елемент послідовності, елемент, який виставляє мітку на задану клітину (початкова умова), елемент, що фіксує результат.

Діаграму класів послідовності клітинних автоматів показано на рисунку 3.3.

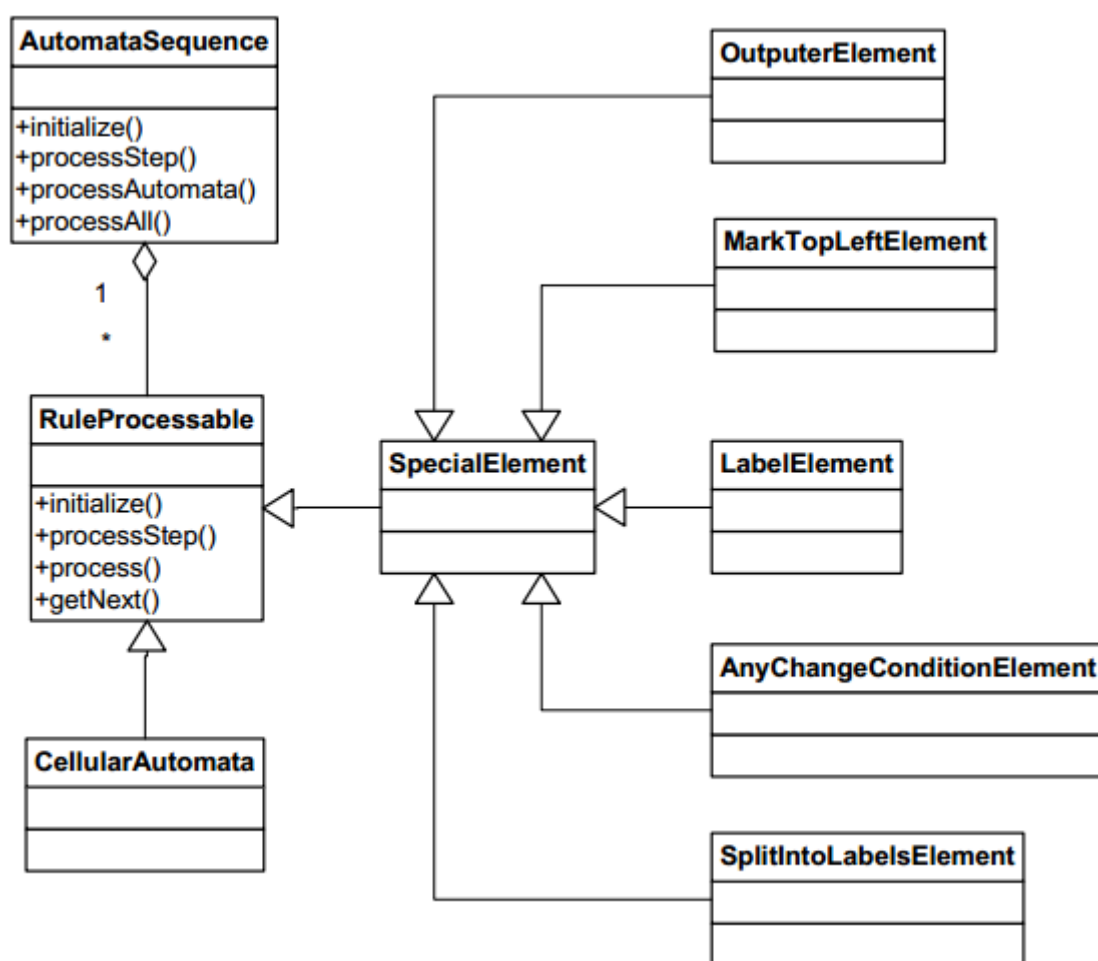


Рисунок 3.3 – Діаграма класів послідовності клітинних автоматів

Опишемо класи блоку послідовності клітинних автоматів.

AutomataSequence – базовий клас, що описує роботу послідовності клітинних автоматів. RuleProcessable – абстрактний клас, який описує роботу

елемента послідовності. `SpecialElement` – абстрактний клас, який описує роботу спеціального елемента послідовності. `LabelElement` – клас, що визначає мітку у послідовності, визначений для умовних елементів. `AnyChangeConditionElement` – клас – умовний елемент, що перекладає роботу послідовності на мітку за умови виконання дії в одному з клітинних автоматів між міткою та умовою. `SplitIntoLabelsElement` – клас, що поділяє матрицю клітин на кілька матриць залежно від виставлених у них міток. `MarkTopLeftElement` – клас, що мітить верхню ліву чорну клітинку в матриці. `OutputerElement` – клас, що виводить результат роботи послідовності на екран.

У програмній системі блок навчання та класифікації заснований на статистичному методі: на етапі навчання система накопичує знання про ознаки символів, але на етапі класифікації – вибирає необхідний символ на основі цих знань за виділеними ознаками. Під ознаками в даному випадку розуміють наявність у символу перетинів, кінців і петель та їх взаємне розташування.

Завдяки продуманій архітектурі, у програмному забезпеченні передбачено можливість доповнення існуючого інструменту класифікації іншими. Наприклад, можна використовувати алгоритми нейронних мереж або алгоритми, що базуються на клітинних автоматах. Діаграму класів блоку навчання та класифікації показано на рисунку 3.4.

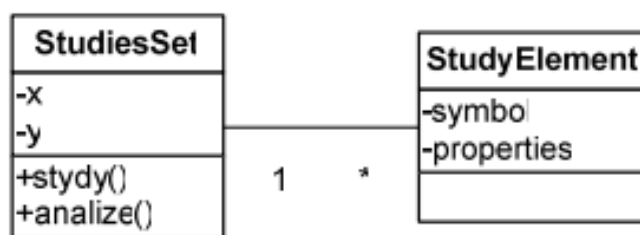


Рисунок 3.4 – Діаграма класів блоку навчання та класифікації

У таблиці 3.2 представлено опис класів блоку навчання та класифікації.

Таблиця 3.2 – Опис класів блоку навчання та класифікації

Клас/поле/метод	Опис
StudiesSet – базовий клас, що описує роботу блоку навчання та розпізнавання	
List<StudiesElement> elements	Елементи навчання системи
int x	Індекс поділу по горизонталі
int y	Індекс поділу по вертикалі
void study(MatrixElement[][] matrix, List<Labels> acceptLabels, String text)	Навчання системи на виділених ознаках, заданих матрицею клітин і списком міток, що розпізнаються, а також текст, що відповідає цим ознакам
List<AnalyzeResult> analyze (MatrixElement[][] matrix, List<Labels> acceptLabels)	Розпізнавання ознак, заданих матрицею клітин та списком позначок, що розпізнаються. Повертає результат аналізу, відсортований за імовірністю
StudyElement – клас – елемент навчання системи	
Map<Integer, Map<Integer, Integer>>> numbers	Частота подання ознаки щодо міток та координат
int getNumbers(int l, int x, int y)	Повертає частоту ознак щодо координат та мітки

Програмну систему побудовано на посторінковій основі, де сторінки із заданою функціональністю йдуть послідовно одна за одою. Всього у програмі п'ять сторінок: завантаження зображення, створення та зміна правил клітинних автоматів, комбінування клітинних автоматів у послідовності, запуску послідовності та розпізнавання тексту щодо даних, отриманих у результаті роботи послідовності клітинних автоматів.

Завантаження зображення відбувається на першій сторінці. Зображення є вхідними даними для послідовності клітинних автоматів. На цій сторінці для зручності створена можливість збільшення або зменшення зображення. На другій сторінці можливе створення клітинних автоматів та їх правил. Список правил

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		







відображається у правій частині вікна програми, з якого можна вибрати конкретне правило для редагування в основній частині вікна. Сторінка комбінування клітинних автоматів дозволяє скласти послідовності запуску клітинних автоматів та визначати додаткову логіку запуску: непослідовні переходи до наступних автоматів за умовою, розбиття зображення, визначення початкової умови для роботи клітинних автоматів. Четверта сторінка програми служить для запуску послідовності клітинних автоматів. Після завантаження послідовності з пам'яті програми або з файлу послідовність можна запускати на виконання. При цьому доступні три режими роботи: повний запуск усієї послідовності, запуск автомата на виконання та запуск кроку автомата. На кожному кроці в основному вікні можна переглянути попередні результати роботи. Сторінка розпізнавання використовує дані, отримані в результаті запуску послідовності клітинних автоматів, для навчання та розпізнавання тексту символів. Для цього використовуються мітки, якими відзначалися ознаки символів, та метод розпізнавання, заданий у програмі.

3.2 Реалізація алгоритму розпізнавання тексту на основі клітинних автоматів

Програма дозволяє реалізовувати різні алгоритми на клітинних автоматах та клітинних автоматах з мітками, з подальшим запуском та перевіркою працездатності. Зокрема, програма дозволяє виділити ознаки символів із їхніх образів та класифікувати їх. На основі розробленого програмног забезпечення досліджено розроблені алгоритми на основі клітинних автоматів для різних символів шрифтів. У таблиці 3.3 представлено частину цього дослідження.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.3 – Дослідження роботи алгоритмів виділення ознак символів для різних шрифтів

Характеристики тексту	Перший алгоритм виділення	Другий алгоритм виділення ознак символів
Символи невеликого розміру шрифта Arial		
Символи великого розміру шрифта Arial		
Символи великого розміру шрифта Times New Roman		

Результати показали стійкість роботи алгоритму для різних розмірів одного й того ж шрифту. У той же час для різних шрифтів алгоритми виділяють кілька відмінних комбінацій ознак та їх розташування, що підтверджує необхідність обліку шрифту при використанні систем OCR з урахуванням даних алгоритмів.

Розроблена програма побудована з орієнтацією на розширюваність: врахована можливість додавання нових елементів, які необхідні для проведення досліджень. Продуктивність системи багато в чому залежить від заданої послідовності клітинних правил і безпосередньо обчислюється з кількості кроків, що виконуються клітинними автоматами. Основний час роботи програми займає обхід поля для реалізації роботи клітинних автоматів. Для оптимізації алгоритму

обходу слід враховувати принцип локальності правил та реалізовувати принцип паралельного обчислення.

Оскільки програмне забезпечення розроблено на основі мови Java, то враховано можливості запуску програми на будь-якій платформі, а також із Інтернет-браузера на основі технології Applet. Швидкісні характеристики програмного забезпечення для першого алгоритму розпізнавання тексту представлено у таблиці 3.4.

Таблиця 3.4 – Швидкісні характеристики програми

Слова для обробки послідовності	Розмір зображення у пікселях	Кількість символів	Кількість ітерацій автоматів	Час роботи, с
Документування	236x30	16	10947	6
Алфавіт, окрім букв Й, І, Ї	454x44	30	20250	13
Текст	561x56	76	46006	16

Невисокі характеристики роботи визначаються моделюючим характером розробленої програми. Таким чином, алгоритми, що використовуються, вимагають оптимізації та доопрацювання.

Навчання системи полягає у запуску алгоритмів виділення ознак символів на спеціально створеному прикладі. Цей приклад повинен містити, по можливості, всі символи, що обробляються системою. Найбільш оптимально проводити навчання на зображенні алфавіту. Після обробки послідовністю клітинних автоматів, у кожному образі символу алфавіту буде виділено ознаки символів.

Користувач програми асоціює набір ознак конкретного символу і ця асоціація зберігається. Надалі інформацію про відповідність ознак певним символам можна буде використовувати у процесі розпізнавання текстів, написаних тими самими шрифтами, як і навчальний текст.

Зважаючи на вищезазначене, використаємо український алфавіт для навчання. Це зображення завантажується на першій сторінці програми. Після запуску послідовності клітинних автоматів, які перетворюють зображення на чорно-біле, поділяють зображення тексту на образи окремих символів та виділяють ознаки зображень. При цьому використовуємо перший алгоритм виокремлення ознак символів. У результаті кожного символу з'являється набір ознак. На рисунку 3.5 схематично зображено ознаки перших чотирьох букв алфавіту.

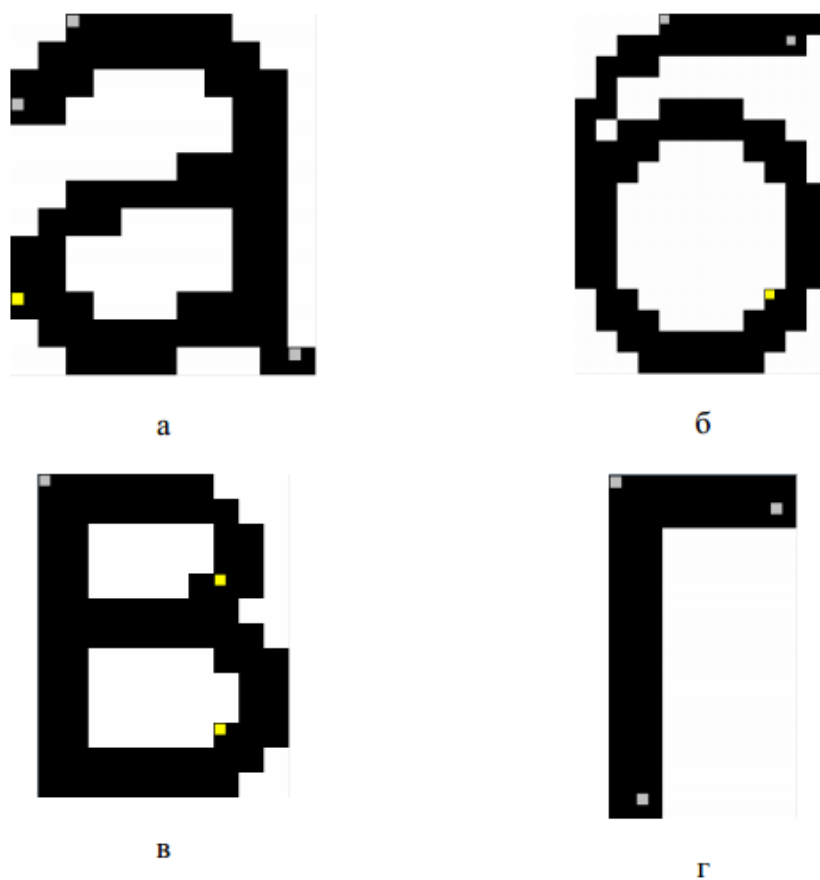


Рисунок 3.5 – Ознаки (розташування кінців та петель) символів алфавіту

Після виокремлення ознак їм у відповідність необхідно виставити символи. Для цього необхідно вибрати, які мітки в зображенні відповідають ознакам символів. Після визначення міток запускаємо навчання, при цьому потрібно ввести символ, відповідний зображенню. Як тільки всі символи визначені користувачем, дані можна зберегти у файл і потім використовувати в блоці розпізнавання текстів.

Послідовність дій у програмі, відповідно розпізнаванню тексту практично аналогічна діям, необхідним для навчання: необхідний запуск послідовності клітинних автоматів для перетворення зображення на чорно-біле, розділення зображення тексту на образи окремих символів та виокремлення ознак символів із зображень. Для прикладу взято образ слова «документування», як показано на рисунку 3.6.

ДОКУМЕНТУВАННЯ

Рисунок 3.6 – Зображення слова для розпізнавання

Після завершення роботи послідовності клітинних автоматів (четверта сторінка моделюючої програми) із зображень символів слова виділено ознаки. Приклади ознак перших двох букв слова представлено на рисунку 3.7.

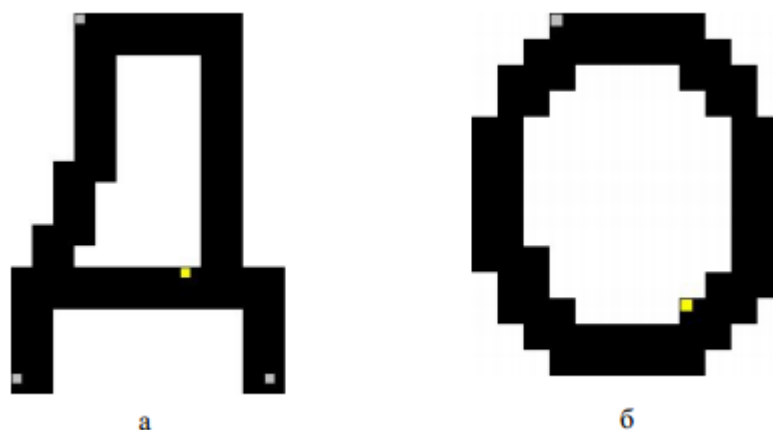


Рисунок 3.7 – Ознаки (розташування кінців та петель) літер: а – Д, б – О

Після виділення ознак символів на п'ятій сторінці програми запускається розпізнавання тексту на основі цих ознак. Для цього аналогічно процесу навчання вибираються мітки, які відповідають за ознаки символів, і завантажується файл шаблону, який містить інформацію про навчання на алфавіті. Результатом розпізнавання буде текст, кожний символ якого визначено із заданою точністю на основі статистичного методу.

Як видно з прикладу, результат містить помилку в дев'ятому символі, що означає нестачу інформації для коректного розпізнавання. Більш точний результат можна отримати на основі другого алгоритму виокремлення ознак символів.

3.3 Порівняльний аналіз результатів розпізнавання

Основним завданням розроблених алгоритмів у процесі розпізнавання тексту – це обробка зображення та виокремлення ознак. Більшість подібних систем є комерційними продуктами, тому інформацію про алгоритми, які вони використовують, не розміщує у публічному доступі. Такі системи через закритість інформації для порівняння із розробленими алгоритмами застосувати неможливо.

У світі представлено кілька відкритих програмних засобів з розпізнавання тексту. На жаль, практично всі з них характеризуються відсутністю документації до методів та алгоритмів, які вони використовують. Деякі програми перебувають у стадії розробки (ImageTextEditor, OpenOCR), інші – спеціально оптимізовані під Unix системи, частина використовує алгоритми, що оптимізовані під конкретні мови (Kognition для німецької мови, Tesseract для англійської мови) [18, 32].

Найбільш опрацьованим відкритим програмним засобом є Tesseract [18]. Таку систему розроблено ще в 1985 року спочатку Hewlett-Packard, потім університетом Невади в Лас-Вегасі (University of Nevada in Las Vegas) та доопрацьовувалась компанією Google, яка у серпні 2006 року відкрила вихідний тексти програми.

Програма добре розпізнає тексти на основі шрифтів без засічок, для чого використовує комплекс алгоритмів, оптимізованих для символів англійського алфавіту. У порівнянні з Tesseract розроблені в кваліфікаційній роботі алгоритми не залежать від мови і служать для дослідження поведінки клітинних автоматів у

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

системах текстового розпізнавання. Проте Tesseract працює швидше від розробленої програми на основі клітинних автоматів, що лише підтверджує дослідницький характер роботи.

У порівнянні з нейронними мережами, алгоритми, представлені в роботі, виконують інше завдання, а саме виділяють показники символів з їх образів. У той час як нейронні мережі використовуються для класифікації виділених ознак. Нейронні мережі можуть використовуватись у комплексі з розробленими алгоритмами виокремлення ознак символів. У цьому випадку вони вирішуватимуть задачі класифікації ознак, виділених алгоритмами на основі клітинних автоматів.

Реалізація алгоритмів розпізнавання тексту на основі клітинних автоматів порівняно з реалізацією системи розпізнавання тексту на основі нейронних мереж. При цьому, якщо програма на основі нейронних мереж показала швидкодію в середньому більшу порівняно із програмою на основі клітинних автоматів, то відсоток помилок виявився нижчим у двічі в алгоритмів на основі клітинних автоматів.

У таблиці 3.5 представлено порівняльну характеристику роботи розробленого програмного забезпечення на основі клітинних автоматів із програмним забезпеченням на основі нейронної мережі та програмним продуктом FineReader компанії АBBYY [1].

Таблиця 3.5. – Порівняльна характеристика роботи програм на основі клітинних автоматів, нейронних мереж та продукту FineReader

Тестові значення	Розрблене ПЗ		Нейронні мережі		FineReader	
	Час роботи, с.	% помилок	Час роботи, с.	% помилок	Час роботи, с.	% помилок
Образ слова «документування», розмір 236*30	6	4	2	9	1	0

Образ українського алфавіту	13	3	4	11	1	3
Образ тексту українською мовою (76 символів), розміри 561*56	16	11	5	17	3	1
Образ сторінки тексту (742 символи)	83	12	40	16	4	2

З наведених у таблиці 3.5 даних можна зробити декілька висновків, які наведено нижче.

1. Характеристика якості розпізнавання вища у розробленого програмного забезпечення в порівнянні із програмною системою основі нейронних мереж. FineReader перевершує якість розпізнавання у порівнянні із розглянутими програмами для порівняння.

2. За швидкістю роботи лідирує програма FineReader, потім програмне забезпечення на основі нейронних мереж та з невеликим відставанням від останньої знаходиться розроблена програма.

Високі показники точності розпізнавання тексту продуктом FineReader визначаються використанням комплексу ресурсів по розпізнаванню, зокрема, активне використання словників [1]. Також можна додати, що FineReader є закінченим готовим продуктом компанії АBBYY, що удосконалюється протягом декількох років.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, представлене порівняння носить інформативний характер і не є основою для висновків щодо ефективності роботи розробленого програмного забезпечення на основі клітинних автоматів.

Невисокі характеристики швидкості роботи розробленої програми в якості OCR-системи визначається моделюючим характером програми. Під час розробки програми основним завданням було дослідження алгоритмів на основі клітинних автоматів у системі розпізнавання тексту з можливістю поетапного, покрокового огляду результатів. Оптимізація алгоритмів розпізнавання значно збільшить продуктивність моделюючої програми. Таку оптимізацію можна досягнути шляхом реалізації кількох принципів:

- запровадження узагальнених координат;
- виключення тих частин полів, які не використовуються, але у яких працюють клітинні автомати;
- об'єднання точок зображення у клітини великого розміру тощо.

Проте, варто зазначити, що для розпізнавання тексту не важливим є показник швидкості розпізнавання, тут ключовим параметром є якість розпізнавання. Як показують результати порівняльного аналізу, розроблені алгоритми на основі клітинних автоматів з мітками та послідовності клітинних автоматів в основі програмного забезпечення уможливають отримати достовірніші результати розпізнавання машинного тексту. Зокрема, при розпізнаванні сторінки тексту достовірність вища на 4 %.

Розроблені алгоритми обробки зображення, ґрунтуються на поділі образу тексту на зображення окремих символів. Алгоритми виокремлення ознак символів дозволяють виконати частину завдач процесу розпізнавання тексту. Ці алгоритми базуються на послідовності клітинних автоматів з мітками, що дає їм усі переваги використання клітинних автоматів.

1. Розроблені алгоритми можуть брати участь у паралельних обчисленнях.
2. Алгоритми виділяють унікальні ознаки символів та дозволяють на їх основі реалізувати подальшу класифікацію.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Алгоритми стійкі до розмірів символів, але залежить від шрифту тексту, особливо, пр. наявності наявність засічок.

4. Алгоритми ґрунтуються на простих правилах і не містять складних обчислень, особливо що стосується обчислень із плаваючою крапкою.

Алгоритми можуть бути задіяні у комплексному розв'язуванні задачі розпізнавання тексту, де частина алгоритмів використовуватиме інший принцип. Розроблені алгоритми реалізовані та протестовані на основі програмного забезпечення. Результати підтвердили їх працездатність та можливість подальшого удосконалення. Таким чином, показано перспективність застосування клітинних автоматів у системах розпізнавання тексту.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ТЕХНІКО – ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОЕКТУ

4.1 Визначення витрат на розробку програмного забезпечення

Розроблене програмне забезпечення призначене для розпізнавання тексту на основі виділення ознак символів. Алгоритм роботи послідовності клітинних автоматів з мітками, на якому базується розроблене програмне забезпечення характеризується високою швидкістю реалізації, що уможливлює отримати результат розпізнавання літери із високою точністю.

Витрати на розробку і впровадження програмних засобів (K) включають [5]:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмних засобів, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програми вирішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників;
- витрати на відрахування у спеціальні державні фонди;
- витрати на покупні вироби;
- витрати на придбання спецобладнання для проведення експериментальних робіт;
- накладні витрати;
- інші витрати.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування програмного засобу. Перелік необхідної програмної документації визначено відповідно до ДСТУ 3008-95 та включає:

- текст програми;
- керівництво користувача, яке включає інструкцію користувача;

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

- опис програми – відомості про логічну і фізичну модель, відомості щодо функціонування програми;
- пояснювальна записка – схема алгоритму, загальний опис алгоритму або функціонування програми, а також обґрунтування прийнятих технічних і технічно-економічних рішень.

Перш за все визначаємо стадії розробки програмного засобу. У таблиці 4.1 відображено інформацію щодо етапів технологічного процесу розробки проекту

Таблиця 4.1 – Стадії розробки програмного засобу

№ п/п	Назва операції (стадії)	Виконавець, посада	Середній час виконання операції, год.
1	Підготовка, складання ТЗ	Менеджер проекту	3
2	Розробка макету системи	Team lead, Дизайнер (2)	2
3	Створення алгоритму системи	Архітектор (1)	3
4	Розробка системи	Програміст (6), дизайнер (2)	140
5	Тестування продукту	Тестувальник (3)	10
6	Попереднє представлення для замовника	Менеджер проекту	4
7	Представлення реалізованої системи	Менеджер проекту	1
Всього		14	163

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год.;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

За умов, якщо середньогодинну ставку розробника не відомо, її можна розрахувати за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

де C_{ij}^0 – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати;

$PЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го розряду, год.

Результати розрахунків записуємо у таблицю 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці при розробці проекту розглянутого типу

№ п/п	Посада виконавця	Час розробки, год.	Погодинна заробітна плата, грн	Витрати на оплату праці, грн
1	Back-end розробник (2)	25	350	17 500
2	Front-end розробник (2)	25	300	15 000
4	Team lead(1)	15	400	6 000
5	Python розробник(1)	35	350	12 250
6	Менеджер(1)	10	450	4 500
7	Тестувальник(3)	10	250	2 500
8	Дизайнер(2)	20	300	12 000
9	Архітектор(1)	3	225	675
	Всього	143		70425

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки виконавцем кваліфікаційної роботи є студент, то він є і розробником і тестувальником і дизайнером. Таким чином, оплата його праці – це стипендія без додаткових нарахувань, надбавок та премій. Стипендія студента становить 1400 грн. Зважаючи на це, вартість проекту включає стипендію студента, а також витрати керівника на керівництво розробкою проекту та консультанта із написання техніко-економічного розділу. У таблицю 4.3 записуємо витрати на розробку програмного засобу у вигляді написання кваліфікаційної роботи.

Таблиця 4.3 – Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год (за весь рік)	Витрати на розробку, грн.
1	Керівник КР, доцент	8	409	3272
2	Консультант з техніко-економічного розділу, доцент	1	409	409
3	Студент	150	22	3300
Разом				6981

У таблиці 4.4 наведено витрати на матеріали та комплектуючі вироби при виконанні кваліфікаційної роботи.

Загальна сума витрат на матеріальні ресурси B_M визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.4)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Таблиця 4.4 – Розрахунок витрат на матеріали та комплектуючі

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	уп	200,0	1	200,00	20,0	220,0
2	Ручка кулькова	шт	10,0	1	10,00	1,0	11,0
3	Олівець простий	шт	8	2	16,00	1,6	17,6
4	Диски CD-R	шт	9	2	18,00	1,8	19,8
5	Зошит, 24 арк.	шт	10	1	10	1	11
6	Тонер для принтера	уп	80	1	80	8,0	88
Разом							367,40

Накладні витрати включають три групи витрат: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками від витрат на оплату праці. При цьому накладні витрати складають 70% від заробітної плати:

$$H = 70425 \cdot 0,75 = 49297,5 \text{ грн.}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 6981 \cdot 0,1 = 698,1 \text{ грн.}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_{KB} + I. \quad (4.4)$$

$$K_1 = 6981 + 754,6 + 367,40 + 698,1 = 8801,1 \text{ грн.}$$

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.5)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год (приймаємо 6);

$t_{від}$ – час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 25 днів, середній щоденний час роботи на комп'ютері – 1 год., вартість години роботи комп'ютера дорівнює 6 грн. Звідси витрати на відлагодження та експлуатацію розраховуємо:

$$K_2 = 6 \cdot 25 = 150 \text{ грн.}$$

Оскільки розробка проекту включає застосування засобів обчислювальної техніки, то розрахуємо витрати на електроенергію, а результати розрахунків занесемо у таблицю 4.5.

Загальну суму витрат на електроенергію розраховуємо за формулою:

$$B_E = \sum_{i=1}^n P_i \cdot k_i \cdot T_i \cdot Ц, \quad (4.6)$$

де P_i – паспортна потужність i -го електрообладнання, $кВт$;

k_i – коефіцієнт використання потужності i -го електрообладнання (приймається 0.7...0.9);

T_i – час роботи i -го устаткування за весь період розробки, год;

$Ц$ – ціна електроенергії, грн / кВт*год;

i – тип електрообладнання;

n – кількість електрообладнання.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.5 – Витрати на електроенергію

Назва устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнання, год	Ціна Електроенергії, грн, кВт/год	Сума, грн
Комп'ютер	0.5	0.7	163	1,68	95,85

До амортизації основних фондів включається сума амортизаційних відрахувань від вартості обладнання і приладів, що використовуються при розробці програмного продукту. Амортизаційні відрахування розраховуємо за формулою:

$$B_{AM} = \sum_{i=1}^n \frac{B_i \cdot H_i \cdot T_i}{100 \cdot T_{E\Phi i}}, \quad (4.7)$$

де B_i – вартість i -го устаткування, грн.;

H_i – річна норма амортизації i -го устаткування, %;

T_i – час роботи i -го устаткування за весь період розробки, год.;

$T_{E\Phi i}$ – ефективний фонд часу роботи i -го устаткування за рік, год / рік;

i – тип устаткування;

n – кількість устаткування.

Усі розрахунки заносимо у таблицю 4.6.

Таблиця 4.6 – Амортизація основних фондів

Найменування устаткування	Вартість устаткування, грн	Річна норма амортизації, %	Ефективний фонд часу роботи обладнання, год / рік	Час роботи обладнання для розробки системи, год	Сума, грн.
Комп'ютер	20000	60	1700	163	1150,6
Разом амортизаційні відрахування					1150,6

На основі отриманих даних в результаті обчислень складаємо кошторис

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

витрат на розробку програмного забезпечення і заносимо їх у таблицю 4.7.

Таблиця 4.7 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	6981
2	Витрати на електроенергію	95,85
3	Витрати на куповані вироби	367,40
4	Амортизаційні відрахування	1150,6
5	Інші витрати	698,1
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	150
Разом		9442,95

Розрахувавши капітальні вкладення у розробку проекту, перейдемо до обчислення можливої продажної ціни.

4.2 Розрахунок ціни проекту

Для оцінки економічної ефективності розробленого програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення. Для цього визначимо експлуатаційні витрати на робробку проекту.

Експлуатаційні одноразові витрати по розробці програмного забезпечення і його аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.8)$$

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

$E_{2П}$ – вартість роботи комп’ютера для розробки програмного продукту (аналогу), грн.

Річні експлуатаційні витрати B_{EP} визначаються за формулою:

$$B_{EP} = E_{П} * N_{П}, \quad (4.9)$$

де $N_{П}$ – періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп’ютері визначається за формулою:

$$E_{1П} = \sum_{l=1}^n n_i t_i c_i, \quad (4.8)$$

де i – категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i – кількість працівників i -ої категорії, осіб;

t_i – трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i – середньогодинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення (4.3).

Трудомісткість підготовки даних для проектного рішення складає 4 год., для аналога 2 год.

Таблиця 4.8 – Розрахунок витрат на реалізацію програмного забезпечення

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Проектне рішення			
1	4	8	32
Аналог			
1	2	32	64

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t * S_{MG},$$

де t – витрати машинного часу для реалізації проектного рішення (аналогу), год.;

S_{MG} – вартість однієї години роботи комп'ютера, грн./год.

Зважаючи на вищенаписане, проверемо розрахунки:

$$E_{2\Pi} = 4 \cdot 6 = 24 \text{ грн.}, E_{2\Pi_a} = 2 \cdot 6 = 12 \text{ грн.};$$

$$E_{\Pi} = 32 + 24 = 56 \text{ грн.}, E_{\Pi_a} = 64 + 12 = 78 \text{ грн.};$$

$$B_{E\Pi} = 56 \cdot 252 = 14112 \text{ грн.}, B_{E\Pi_a} = 78 \cdot 252 = 19656 \text{ грн.}$$

Ціна програмного продукту – це витрати на придбання і експлуатацію програмного засобу за весь період його служби:

$$C_{\Pi} = K \cdot \left(1 + \frac{\Pi_p}{100}\right) + K_0 + K_k, \quad (4.9)$$

де K – кошторисна вартість;

Π_p – рентабельність;

K_0 – витрати на встановлення та освоєння програмного засобу на конкретному об'єкті, грн.;

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.

Зважаючи на вищеописане, розрахуємо ціну програмного засобу

$$C_{\Pi_a} = 141719,525 \cdot (1 + 0,3) = 184235,4 \text{ грн.}$$

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

$$Ц_{II} = 9442,95 \cdot (1 + 0,3) = 12275,8 \text{ грн.}$$

У наступному підрозділі проведемо аналіз економічної ефективності розробки програмного продукту.

4.3 Визначення економічної ефективності розробки проєкту

Для того, щоб побудувати таблицю показників економічної ефективності розробки програмного продукту, проведемо розрахунки необхідних показників.

Економічний ефект в сфері проєктування проєктного рішення розраховуємо за формулою:

$$E_{IP} = Ц_{II} - Ц_A, \quad (4.10)$$

$$E_{IP} = 184235,4 - 12275,8 = 171959,6 \text{ грн.}$$

Річний економічний ефект від експлуатації програмного продукту:

$$E_{KC} = B_{EA} - B_{EP}, \quad E_{KC} = 19656 - 14112 = 5544 \text{ грн.} \quad (4.11)$$

Дохід від розробки ПЗ у i -му періоді розраховуємо за формулою:

$$D_i = J_i(B_i - C_i), \quad (4.12)$$

де B_i – ціна продажу програмного продукту в i -му періоді;

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

C_i – собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

J_i – кількість ПЗ.

$$D_i = 1 \cdot (12275,8 - 9442,95) = 2832,85 \text{ грн.}$$

Економічний ефект полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів та розраховується за формулою:

$$E = \frac{D_i}{B_{\text{заг}}}. \quad (4.13)$$

$$E = \frac{2832,85}{9442,95} = 0,3.$$

Тоді термін окупності обчислюємо за такою формулою:

$$T = \frac{1}{E} \quad (4.14)$$

$$T = \frac{1}{0,3} = 3,3 \text{ р.}$$

Зважаючи на проведені розрахунки ефективності розробки програмного продукту, обчислимо сумарний ефект від розробки програмного продукту за формулою:

$$E = E_{IP} + E_{KC}$$

$$E = 171959,6 + 5544 = 177503,6 \text{ грн.}$$

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.9 – Показники економічної ефективності проектного рішення

№	Найменування	Значення показників	
		Аналог	Новий варіант
1	Капітальні вкладення	70425	9442,95
2	Ціна придбання	184235,4	12275,8
3	Економічний ефект в сфері проектування	-	171959,6
4	Економічний ефект в сфері експлуатації	-	5544
5	Дохід від розробки		2832,85
6	Сумарний ефект	177503,6	
7	Термін окупності проекту	3,3	

Отже, у цьому розділі проведено розрахунок витрат на розробку програмного забезпечення. Показники, що характеризують витрати на розробку програмного продукту порівняно із показниками, які характеризують програмний продукт із аналогічним функціональним призначенням.

Розроблене програмне забезпечення має суттєві переваги у порівнянні із аналогами, зокрема простота використання, швидкість проведення розрахунків, зручність.

Згідно із проведеними розрахунками, що обґрунтовують економічну ефективність розробки програмного продукту, можна зробити висновок, що розроблене програмне забезпечення є суттєво дешевшим, оскільки у ролі розробника виступає студент. Отримано економічний ефект у розмірі 177503,6 грн., що свідчить про економічну доцільність розробки і впровадження програмного забезпечення для аналізу розпізнавання тексту на основі клітинних автоматів.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано практичну розпізнавання машинного тексту на основі клітинних автоматів. При цьому отримано такі практичні результати.

1. Проаналізовано програмні засоби розпізнавання тексту, виокремлено їх переваги та недоліки. Зазначено, що для зважаючи на особливості розв'язування задачі розпізнавання тексту необхідне застосування комплексу алгоритмів.

2. Проаналізовано алгоритм опрацювання текстового образу на основі клітинного автомату, що включає три правила: переведення кожної точки зображення у відтінок сірого, замальовування клітини чорним, якщо її колір темніший за граничний та білим – колір світліший за граничний.

3. Досліджено особливості застосування клітинних автоматів у задачі розпізнавання тексту, проаналізовано кожне підзавдання процесу розпізнавання та оцінено можливість її вирішення за допомогою клітинних автоматів.

4. Охарактеризовано поняття клітинних автоматів з мітками та послідовностей клітинних автоматів й зазначено, що клітинні автомати з мітками мають спільні риси з поняттям пам'яті тьюрингових машин.

5. Розроблено алгоритми фільтрації зображення, сегментації зображення тексту на окремі образи символів та виокремлення ознак символів, які дозволяють проводити паралельні обчислення на важливих стадіях процесу розпізнавання тексту. Окрім того, використання механізму клітинних автоматів уможливило апаратну реалізацію алгоритмів.

6. Для реалізації розроблених алгоритмів розроблено програмне забезпечення, яке дозволило покроково оцінити ефективність створених алгоритмів та провести експерименти на основі задачі розпізнавання тексту на основі українського алфавіту.

7. Обґрунтовано техніко-економічні показники ефективності розробки програмного забезпечення для розпізнавання машинного тексту.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abbyy finereader pdf corporate:веб-сайт. URL: <https://abbyy.store/uk-ua/fr15-corporate/> (дата звернення 16.05.2023).
2. Bilan S.M. Evolution of two – dimensional cellular automata. New forms of presentation. 2021. Т. 3. № 1. Р. 85-90.
3. Jegadish Kumar K.J., Kesava Reddy K., Salivahanan S. Novel and Efficient Cellular Automata Based Symmetric Key Encryption Algorithm for Wireless Sensor Networks. International Journal of Computer Applications. 2011. Vol. 13. № 4. Р. 30-37.
4. John von Neumann. Theory of Self-Reproducing Automata. Urbana and London : University of Illinois Press, 1966. 25 p.
5. Klieshch Danil, Volodymyr Fedorchenko Volodymyr Аналіз підходів до розв’язання задач розпізнавання образів з використанням штучного інтелекту // Системи управління, навігації та зв’язку. Збірник наукових праць. –Полтава: ПНТУ, 2023. Т. 1 (71). С. 96-100.
6. Laura Diosan, Anca Andreica, Alina Enescu. The use of simple cellular automata in image processing. Studia Universitatis Babeş-Bolyai, Informatica. Jun 2017. Vol. 62. Issue 1. P. 5-14. 10 p.
7. Mainzer K., Chua L. The Universe as automaton. Springer, 2012. 112 p.
8. Mazoyer J., Terrier V. Signals in one-dimensional cellular automata. Theoretical Computer Science. 1999. 217(1). P. 53–80.
9. Mihaljevic M., Y. Zheng, and H. Imai. A cellular automaton based fast one-way hash function suitable for hardware implementation. Public Key Cryptography. 1998. Volume 1431. P. 217–234.
10. Nandi S., B. K. Kar, and P. P. Chaudhuri. Theory and applications of cellular automata in cryptography. IEEE Trans. Computers. 1994. 43(12). P. 1346–1357.
11. Nelles O. Nonlinear Systems Identification. Berlin: Springer, 2001. 785 p.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Norziana Jamil, Ramlan Mahmood, Muhammad Reza Z'aba. A New Cryptographic Hash Function Based on Cellular Automata Rules 30, 134 and OmegaFlip Network. 2012 International Conference on Information and Computer networks (ICICN 2012). 2012. Vol. 27. P. 163-169.

13. Sarkar P. A brief history of cellular automata. ACM Computing Surveys. 2000. Vol. 32, No. 1. P. 80-107.

14. Schiff J.L. Cellular automata. A Discrete View of the World. A John Wiley & Sons Inc., Publication. University of Auckland, 2008. 279 p.

15. Schmidhuber, J., Deep learning in neural networks: An overview// Neural Networks. 2015. №61. С. 58-69.

16. Seredynski F., Bouvry P., Zomaya A.V. Cellular automata computations and secret key cryptography. Parallel Computing. 2004. P. 753-766.

17. Sinkevych O. V. & Sokolovsky I. Y. Програмне та алгоритмічне забезпечення для роботи з клітинними автоматами // Науковий вісник НЛТУ України, 2018. Вип. 28(3). С. 137-141.

18. Tesseract OCR: веб-сайт. URL: <https://sourceforge.net/projects/tesseract-ocr/> (дата звернення 19.05.2023).

19. Wolfram Stephen. Statistical Mechanics of Cellular Automata // Reviews of Modern Physics: journal. 1983. Vol. 55. P. 601–644.

20. Wolfram S. Random sequence generation by cellular automata. Advances in Applied Mathematics. 1986. Vol. 7. P. 123-164.

21. Yamamoto K., Kokubo S., Nishinari K. Simulation for pedestrian dynamics by real-coded cellular automata (RCA) // Physica A. 2007. Vol. 379. P. 654-660.

22. Анікін В.В. Актуальність застосування клітинних автоматів для розробки комп'ютерних ігор // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я. 2019. Ч. IV. С. 127.

23. Бойко Б. Б. Використання нейронних мереж для розпізнавання касових документів: дипломна робота. Івано-Франківськ, 2022. С. 35.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

24. Валь Л.О., Жихаревич В.В., Остапов С.Е. Розробка і дослідження криптостійкого генератора бінарної послідовності на основі клітинних автоматів. Науковий вісник Чернівецького університету. Комп'ютерні системи та компоненти. 2009. Вип. 479. С.147-150.

25. Валь Л.О., Жихаревич В.В., Остапов С.Е. Застосування клітинних автоматів для генерування бінарних псевдовипадкових послідовностей. Науковий вісник Чернівецького університету. Комп'ютерні системи та компоненти. 2010. Т.1. Вип. 1. С.67-72.

26. Вилучення тексту із зображень за допомогою інструментів OCR): веб-сайт. URL : <https://apix-drive.com/ua/blog/useful/viluchennja-tekstu-iz-zobrazhen-za-dopomogoju-instrumentiv-ocr> (дата звернення 14.05.2023).

27. Волковський О. С., Стародінов М. О. Використання клітинного автомата для перетворення музичного ряду в візуальний // Науковий вісник ТДАТУ. 2014. Вип. 6. Т. 2. С. 108-117.

28. Воробйов Є.С, Павленко М.А., Хлебніков Є.Ю., Гладишев М.Г. Використання кліткового автомату у методі вибору варіанту маршруту польоту ударних літаків щодо ураження наземних цілей. Системи озброєння військова техніка. 2018. № 1(53). С. 84-90.

29. Жихаревич В. В., Миронів І. В., Остапов С. Е. Алгоритм розпізнавання символів тексту на основі конкуруючих клітинних автоматів // Радіоелектроніка, інформатика, управління. 2015. № 4. С. 39-44.

30. Жихаревич В.В., Миронів І.В., Остапов С.Е. Алгоритм розпізнавання символів тексту на основі конкуруючих клітинних автоматів. Радіоелектроніка, Інформатика, Управління. 2015. № 4 (35). С. 39-44.

31. Жихаревич В.В., Чайка Л.О., Камінська О.В., Овчар Р.І., Остапов С.Е. Система захищеного обміну даними на основі клітинних автоматів. Науковий вісник Чернівецького університету. Комп'ютерні системи та компоненти. 2011. Т.2. Вип. 1. С.15-20.

32. Інтелектуальна обробка документів (Intelligent Document Processing) та оптичне розпізнавання символів (Optical Character Recognition): Чи коректно ви

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

застосовуєте терміни?: веб-сайт. URL: <https://dms-solutions.co/uk/blog-uk/ocr-vs-intelligent-document-processing/> (дата звернення 17.05.2023).

33. Кобильник Т.П. Кліткові автомати як засіб моделювання складних систем // Фізико-математична освіта. 2018. Вип. 4. С. 71-75.

34. Макаренко О.С., Крушинський Д.А. Моделювання руху пішоходів на основі клітинних автоматів. Системні дослідження та інформаційні технології. 2010. № 1. С. 100-109.

35. Марченко О.О. Машинно-навчальні методи розпізнавання іменованих сутностей тексту // Proceedings of the 10th International Conference of Programming UkrPROG'2016. Kyiv, 2016. С. 150-157.

36. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

37. Методичні вказівки до оформлення курсових, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О.Дубчак / під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 34 с.

38. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2020. 60с.

39. Особливості систем та методів структурного розпізнавання текстових зображень: веб-сайт. URL: https://sci.ldubgd.edu.ua/bitstream/123456789/4275/1/st_UAD_2017.pdf (дата звернення 22.05.2023).

40. Остапов С.Е., Жихаревич В.В., Добровольський Ю.Г. Сучасні методи та засоби захисту інформації. Чернівці: ЧНУ, 2021. 72 с.

41. Парамуд Я.С., Яркун В.І. Метод розпізнавання символів на зображенні на основі згорткової нейронної мереж: веб-сайт. URL: <https://science.lpnu.ua/sites/default/files/journal->

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

paper/2019/sep/18610/182073vis905ksmzvichniy-96-105.pdf (дата звернення 14.05.2023).

42. Погибелева Л. В. Інформаційна технологія оптичного розпізнавання тексту з зображення для мобільного додатку: кваліфікаційна магістерська робота. Суми, 2021. 56 с.

43. Помаз С. А. Оптичне розпізнавання тексту за допомогою нейромережі : дипломна робота. Чернігів, 2021. 113 с.

44. Приклади моделей клітинних автоматів; веб-сайт. URL: <https://ukrayinska.libretexts.org> (дата звернення 19.05.2023).

45. Сич Т.А., Заяць А-М. В. Навчальні платформи як освітній інструмент сьогодення // Матеріали VII науково-прокритичної конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи і мережі». Тернопіль, 2023. С. 26.

46. Толкованов А.А. Огляд деяких моделей клітинних автоматів: веб-сайт. URL: <http://dspace.onua.edu.ua/bitstream/handle> (дата звернення 20.05.2023).

47. Хміль Ю.В. Аналіз зображень за допомогою клітинних автоматів: курсова робота. 2021. 54 с.

48. Чайка Л.О., Жихаревич В.В., Остапов С.Е. Криптостійкий генератор псевдовипадкової послідовності на основі клітинних автоматів. Вісник Київського національного університету імені Тараса Шевченка. Серія фізикоматематичні науки. 2011. Вип. 1. С.215-219.

49. Чередніченко В.О. Математичне та програмне забезпечення для дослідження властивостей тривимірних клітинних автоматів: магістерська дисертація. К., 2019. 116 с.

50. Що таке оптичне розпізнавання символів (OCR): веб-сайт. URL : <https://www.konicaminolta.ua/uk-ua/rethink-work/tools/what-is-optical-character-recognition-ocr> (дата звернення 18.05.2023).

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		