

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»  
ім. ІГОРЯ СІКОРСЬКОГО

факультет Інформатики та обчислювальної техніки  
(повне найменування інституту, факультету)

кафедра обчислювальної техніки  
(повна назва кафедри)

До захисту допущено  
Завідувач кафедри  
Стіренко С.Г.

( прізвище, ініціали )

(підпис)

“ \_\_\_ ” \_\_\_\_\_ 2019 р.

Дипломний проект  
освітньо-кваліфікаційного рівня “бакалавр”  
(назва ОКР)

з напрямку підготовки (спеціальності) 6.050103 « Програмна інженерія »  
(код та назва напрямку підготовки або спеціальності)

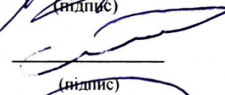
на тему «Система розпізнавання рукописних символів»

Виконав: студент 4 курсу, групи ІП-53  
Петрик Віталій Віталійович  
(прізвище, ім'я, по батькові)



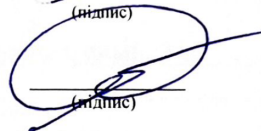
(підпис)

Керівник ст. викл. Сімоненко Андрій Валерійович  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)

Консультант нормоконтроль д.т.н., проф. Сімоненко В.П.  
(назва розділу) (вчені ступінь та звання, прізвище, ініціали)



(підпис)

Рецензент доц. каф. СТССС К.Т.Н., доц. Савка І.Т.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)

Київ - 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМ. ІГОРЯ СІКОРСЬКОГО»**

Факультет Інформатики та обчислювальної техніки  
(повне найменування інституту, факультету)

Обчислювальної техніки

Освітньо-кваліфікаційний рівень **бакалавр**

Напрямок підготовки **6.050103 « Програмна інженерія »**

ЗАТВЕРДЖУЮ

Завідувач кафедри

( прізвище ініціали )

( підпис )

“ \_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проект студента

Петрика Віталія Віталійовича  
( прізвище, ім'я, по батькові )

Тема проекту (роботи) Система розпізнавання рукописних символів,  
керівник проекту (роботи) ст. викл. Сімоненко А. В.,  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання )

затверджені наказом по університету від “23” 04 2019 року №1180-С

Термін здачі студентом закінченого проекту (роботи) 20 червня 2019 р.

Вихідні дані до проекту (роботи) технічна документація, теоретичні та  
статистичні дані з системи розпізнавання рукописних символів

Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)  
опис предметної області, дослідження та розробка системи розпізнавання  
рукописних символів, дослідження методів та алгоритмів розпізнавання рукопису  
Перелік графічного матеріалу (з точним позначенням обов'язкових креслень)  
схема алгоритму роботи системи, візуалізація процесу розпізнавання системи,  
діаграма прецедентів системи.

## **АНОТАЦІЯ**

В бакалаврській дипломній роботі розроблена система on-line розпізнавання рукописних символів, що дозволяє легко додавати символи будь-якого вигляду до списку тих, що розпізнаються, не потребуючі при цьому повторного тренування, перезавантаження чи оновлення.

Дана робота реалізувала on-line алгоритм розпізнавання рукописних символів, в основі якого лежить жадібний алгоритм співставлення точок. Простота реалізованого алгоритму дозволяє на його основі швидко створити власну систему розпізнавання символів, що має свої переваги і є конкурентоспроможною на невеликій множині відомих системі символів.

## **АННОТАЦИЯ**

В бакалаврской дипломной работе разработана система on-line распознавания рукописных символов, что позволяет легко добавлять символы любого вида в список распознаваемых, не требующая при этом повторной тренировки, перезагрузки или обновления.

Данная работа реализовала on-line алгоритм распознавания рукописных символов, в основе которого лежит жадный алгоритм сопоставления точек. Простота реализованного алгоритма позволяет на его основе быстро создать собственную систему распознавания символов, которая имеет свои преимущества и является конкурентоспособной на небольшом множестве известных системе символов.

## **ANNOTATION**

In this work for a Bachelor's Degree, we have developed the on-line handwriting recognition system, which allows you to easily add characters of any kind to the list of those that can be recognized without requiring retraining, rebooting, or updating.

This work has implemented the online handwriting recognition algorithm, which is based on the greedy point matching algorithm. The simplicity of the implemented algorithm allows you to quickly create based on it your own handwriting recognition system, which has its own advantages and is competitive on a small set of known symbols.

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ .....	6
1.1 Off-line підхід.....	7
1.2 On-line підхід .....	9
1.3 Існуючі проблеми.....	11
1.4 Пов'язані роботи .....	14
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	16
РОЗДІЛ 2. ОПИС МАТЕРІАЛІВ ТА МЕТОДІВ РІШЕННЯ.....	17
2.1 Модель рукописного образу .....	17
2.2 Процес розпізнавання рукописного образу.....	22
2.2.1 Попередня обробка .....	22
2.2.2 Вилучення особливостей.....	27
2.2.3 Класифікація.....	30
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	33
РОЗДІЛ 3. РЕАЛІЗАЦІЯ РІШЕННЯ ЗАДАЧІ .....	35
3.1 Реалізація процесу розпізнавання .....	36
3.2 Опис алгоритмів .....	45
3.3 Інструкція користувачеві.....	49
3.1 Тестування .....	52
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57

					<b>ІАЛЦ.467200.003 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата	Система розпізнавання рукописних символів.  Пояснювальна записка	Літ.	Арк.	Аркушів
Розробн.		Петрик В. В.						
Перевір.		Сімоненко А.В.					1	59
Н. Контр.		Сімоненко В.П.				НТУУ «КПІ», ФІОТ, ІП-53		
Затверд.		Стіренко С.Г.						

## ВСТУП

Рукопис - важлива модальність для взаємодії між людьми та комп'ютером. Для фахівців-медиків почерк є (як і раніше) найкращим природним методом документації. Розпізнавання рукописного тексту давно є основним напрямком досліджень в області комп'ютерних наук. З величезною повнотою смартфонів розпізнавання рукописного тексту стало новим імпульсом. За останнє десятиріччя зі стрімким розвитком потужності обчислювальних систем, розвитком алгоритмів машинного навчання та застосування нейронних мереж, кількість існуючих методів розпізнавання рукопису стрімко збільшилася. На сьогоднішній день знайти систему розпізнавання літер з точністю у 98% або більше не є складним завданням. Однак, дослідники все ще вдосконалюють алгоритми розпізнавання символів, намагаються покращити їх точність, підвищити швидкість їх роботи тощо.

Почерк – поняття, унікальне для кожної людини. Незважаючи на те, що рукописне слово може бути відтворене іншою людиною, у ньому існує певний рівень індивідуальності та унікальності, які неможливо зімітувати чи підробити. Наприклад, до таких особливостей почерку можуть відноситися: спосіб тримати перо при написанні слова, кількість використаних штрихів, їх нахил, рівень закругленості символів, кількість тиску на папір тощо. Оскільки комп'ютеризація стає все більш стрімкою на сьогоднішній день, розпізнавання рукописного тексту набуває все більшого значення в великій кількості різноманітних за своєю діяльністю галузь досліджень чи виробництва. Наприклад, автентифікація підписів у банках, розпізнавання поштових індексів на відісланих рукописних листах, судово-медичні докази тощо. Крім того, надання великомасштабним обчислювальним системам робити весь аналіз та автентифікацію в банку

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		3

чи інших агенціях за допомогою підпису на електронній поверхні зменшує кількість рутинної роботи працівникам.

Дослідження в області розпізнавання рукописного тексту, здається, досягли свого піку в кінці 1990-х років, але існують багато причини для відновлення інтересу до цієї теми, серед яких:

- Велике зростання використання мобільних обчислювальних пристроїв, багато з яких оснащені сенсорними екранами, які дозволяють легко записувати рукописний ввід.
- Бажання підтримувати більше символів, деякі з яких особливо важко набрати на клавіатурі, а для деяких з них взагалі немає стандартної розкладки клавіатури, що була б загальноприйнятою (наприклад, кхмерська мова).
- Нові досягнення в технологіях в таких областях, як розпізнавання мови, машинне навчання, машинний переклад і оптичне розпізнавання символів, які можуть бути застосовується до задачі розпізнавання рукописного введення.

Хоча перші комерційні системи розпізнавання рукописного введення були доступні в 1980-х роках, їх успіх і можливості були сильно обмеженими. Ми вважаємо, що розпізнавання рукописних символів чи образів до сих пір повністю не вирішена, і хотіли б коротко зупинитися на деяких проблемах:

- Між стилем письма існує сильна різниця. Наприклад, у різних груп людей та між окремими людьми переважає друкований стиль письма в порівнянні з рукописним стилем. До цього можна додати і вік особи, що пише: у дітей символи виходять невпевненими та неохайними через їх малий досвід у письмі; у похилих людей тремтять руки, що створює певних шум.
- Існує сильна мінливість навіть для однієї і тієї ж людини. В залежності від швидкості письма, охайності або обставин, при

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		4

яких вводиться текст. Наприклад, при ходьбі чи на екрані мобільного телефону за допомогою дотику пальця.

- Для сценаріїв з невідомою на момент створення системи множиною символів машинне навчання є достатньо проблемним засобом вирішення задачі, так само як і збір правильних даних навчання та охоплення всіх загальних вхідних символів.

Завдання читання рукописного тексту вимагає спеціальних навичок та інколи є не простою задачею. Нерідко люди не можуть прочитати навіть власний почерк. Але як комп'ютер розпізнає письмо особи?

Далі у цій роботі ми наведено огляд деяких існуючих методів розпізнавання та опишемо систему, яка була створена для розпізнавання рукописних символів та образів, множина яких заздалегідь невідома.

На вхід системи подається цифровий сигнал, представлення якого описано далі у роботі, від електронної поверхні чи спеціального електронного пера. Наша цільова платформа – це мобільні пристрої, що мають сенсорний екран, проте архітектура системи може бути використана і на інших платформах, які можуть отримати сигнал з електронної поверхні чи спеціального пера.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		5

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

З рукописним письмом можуть бути пов'язані кілька типів аналізу, розпізнавання та інтерпретації. Розпізнавання рукописного тексту – це завдання перетворення мови, перенесеної у власній вільній просторовій формі графічних знаків у символічне подання. Інтерпретація рукописного тексту – це завдання визначення сенсу рукописного тексту, наприклад, рукописної адреси. Ідентифікація рукописного тексту – це завдання визначення автора вибірки рукописного введення з набору письменників. Ідентифікація та перевірка – це процеси, що визначають характер особливостей написання конкретного письменника, тоді як розпізнавання та інтерпретація рукописного тексту є процесами, метою яких є відфільтрувати варіанти, щоб визначити повідомлення.

Дані рукописного тексту перетворюються в цифрову форму шляхом сканування письма на папері (чи з будь-якої іншої поверхні), або шляхом письма спеціальним електронним пером на електронній (сенсорній) поверхні. Ці два вище згадані підходи отримання образу рукописного тексту розрізняють як off-line та on-line відповідно.

На рис. 1.1 зображена візуалізація двох підходів.

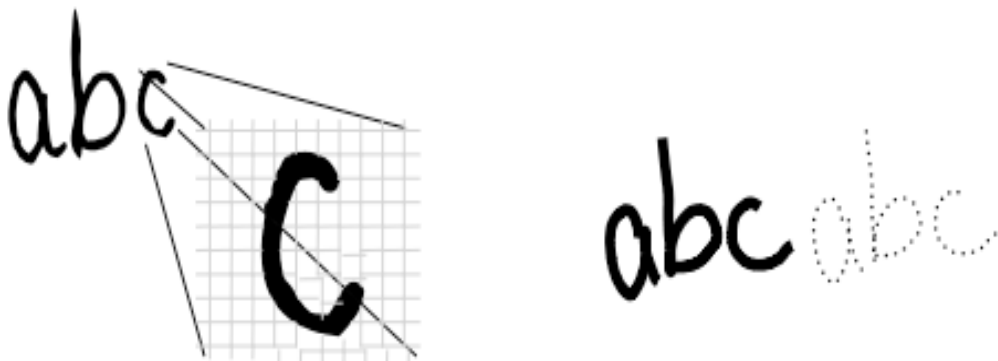


Рис. 1.1 - Off-line (ліворуч) та on-line (праворуч) розпізнавання [1]

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		6



## 1.1 Off-line підхід

В off-line підході на вхід системи подається зображення тексту, яке супроводжується подальшим оптичним скануванням (оптичним розпізнаванням символів – Optical character recognition, OCR [2]) або інтелектуальним розпізнаванням символів (далі Intelligent Character Recognition, ICR), тобто за допомогою штучного інтелекту з використання машинного навчання [3][4].

Дані, отримані за цією формою, розглядаються як статичне зображення тексту. А ось розпізнавання рукописного вводу в off-line підході є порівняно важким завданням, оскільки різні люди мають різний почерк, який інколи навіть їх автори не розуміють.

На сьогоднішній день OCR системи в першу чергу орієнтовані на машинний друкований текст з можливістю розпізнавати велику кількість відомих шрифтів. Яскравим прикладом застосування OCR систем є оптичний зчитувач штрих-кодів. Хоча штрих-код не є різновидом тексту, він є різновидом символного зображення, яке має місце бути у контексті завдання розпізнавання символів.

Для розпізнавання з зображень рукописного тексту використовують ICR. Такі системи, як вже згадувалося вище, використовують штучні нейронні мережі, які тренують на великій кількості спеціально підготовлених зображень. Коефіцієнти сильно нелінійної функції, представлені у вигляді штучної нейронної мережі, з кожним «проходом» підготовленого для тренування екземпляру наближуються до значень, при яких так звана функція помилки (loss function або cost function) [5] наближується до нуля. Натреновані коефіцієнти зумовлюють здатність штучної нейронної мережі класифікувати вхідні дані до одного (чи

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		7

декількох в загальному випадку) класу із списку натренованих. Такими класами можуть бути і символи рукописного тексту, а вхідними даними – їх зображення. Такий підхід з використанням штучного інтелекту дозволяє розпізнавати не лише окремі символи, а й цілі слова.

Під час розпізнавання символів у off-line підході використовують велику кількість технік, деякими з яких є:

- звуження предметної області – наприклад, поштовий індекс адреси складається лише з символів арабських цифр 0-9, що значно зменшує кількість можливих ідентифікацій; так техніка допомагає значно збільшити точність системи розпізнавання;
- зменшення шуму зображення – виділити на зображенні лише корисну інформацію, тобто лише самі лінії символу, та позбутися зернистості зображення;
- витяг особливостей зображення – до списку таких особливостей можуть входити такі, як співвідношення сторін символу, кількість заповнених пікселів у верхній горизонтальній частині зображення, кількість заповнених пікселів у правій вертикальній частині зображення, кількість штрихів символу, середня відстань від центра зображення тощо.

Вище згадані техніки обробки зображення для off-line підходу розпізнавання символів підвищують точність розпізнавання та прискорюють роботу системи.

Варто зазначити, що даний підхід потребує спеціального обладнання для сканування зображення в деяких випадках використання OCR. Приклад з скануванням штрих-кодів наглядно це демонструє.

У випадку використання штучного інтелекту для розпізнавання символів потреби у спеціальному обладнанні немає. Проте розрахунки, що виконує нейронна мережа, потребують значної обчислювальної потужності пристрою, на якому встановлена система розпізнавання. До

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		8

того ж, штучна нейронна мережа потребує велику вибірку екземплярів для тренування, та часу на розрахунки необхідних їй коефіцієнтів. Тобто система з використанням штучного інтелекту не зможе розпізнавати символи інші, аніж ті, що її навчили. Перетренувати його – це великі затрати часу та обчислювальних ресурсів.

## 1.2 On-line підхід

В on-line підході на вхід системи подаються дані, отримані зі спеціальної сенсорної поверхні. На цій поверхні імітується письмо за допомогою спеціального електронного пера або за допомогою звичайного дотику пальця так, начебто письмо відбувається на звичайному листі паперу. Електронна поверхня посилає в систему сигнали, що характеризують поточне положення електронного пера чи пальця у даний момент часу. Таким чином, on-line підхід характеризується тим, що вхідні у систему дані можна представити у вигляді послідовності штрихів, які в свою чергу є послідовністю написаних точок  $(x, y, t)$  з позицією точки  $(x, y)$  та моментом її написання  $t$ [6].

Такий підхід дозволяє передати до системи такі особливості авторського письма як тиск на поверхню, швидкість написання, зміна напрямку штриха тощо. Деякі з таких особливостей можуть бути корисними, а деякі – ні.

Процес розпізнавання рукописного тексту в on-line підході можна розділити на кілька загальних кроків [7]:

- попередня обробка (preprocessing);
- вилучення особливостей (feature extraction);
- класифікація.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		9

Мета попередньої обробки полягає в тому, щоб відкинути неважливу, несуттєву чи надлишкову інформацію у вхідних даних, яка може негативно вплинути на процес розпізнавання [7]. Наприклад, до такого виду інформації можна віднести швидкість написання (адже різні люди пишуть один і той самий символ з різною швидкістю) та тремтіння пера під час проведення штриха (намалювати від руки ідеально пряму лінію неможливо). Попередня обробка зазвичай складається з нормалізації, відбору рівномірної вибірки чи нормальної вибірки вхідного сигналу, згладжування та зменшення шуму [8]. Усе для того, щоб зменшити відмінність між написанням одного і того ж самого символу.

Другий крок - вилучення особливостей письма (feature extraction). З дво- або тривимірного векторного поля, отриманого з алгоритмів попередньої обробки, витягуються дані вищого виміру. Метою цього кроку є виділення важливої інформації для розпізнавання [8]. Ці дані можуть включати в себе таку інформацію як, наприклад, тиск або зміни напрямку штриха.

Останній великий крок - класифікація. На цьому кроці і відбувається відображення оброблених даних до відповідного йому Unicode представлення.

Сучасні методи вирішення останнього кроку – класифікації – зумовлюють використання штучного інтелекту. Штучна нейронна мережа тренується вже не на зображеннях, як це відбувалося у off-line підході, а на багатовимірному векторі, що включає в себе вибірку з обробленого вхідного сигналу та вилучених особливостей письма.

Даний підхід вимагає спеціального обладнання – сенсорної поверхні та/або спеціального електронного пристрою для отримання вхідних даних на розпізнавання рукописного символу. Таким обладнанням, наприклад, володіють смартфони та подібні до них мобільні пристрої, що також мають сенсорний дисплей.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		10

У випадку використання штучного інтелекту для розпізнавання символів проблеми залишаються ті самі, що і у off-line підході. Проте розрахунки, що виконує система у даному підході, потребують меншої обчислювальної потужності пристрою, ніж у попередньому підході. Це зумовлено тим, що у off-line підході для подачі даних до штучної нейронної мережі необхідно провести значну попередню обробку на усьому зображенні. Попередня обробка цілого зображення, очевидно, набагато складніша за обробку певної множини точок, створеної вхідним сигналом електронного письма. До того ж, штучний інтелект має обробити ціле зображення та виділити у ньому його особливості, що є складнішою задачею, ніж класифікувати оброблені вектори даних, описані у цьому підрозділі.

### 1.3 Існуючі проблеми

Кожна людина у своєму рукописному стилі відрізняється певною множиною особливостей. До них можна віднести такі поняття як закруглення символів та штрихів, кут нахилу літер, кількість використаних штрихів тощо. Іноді вони так сильно відрізняються від особливостей інших осіб, що читачам, відмінним від письменника, важко прочитати написане. Бувають і випадки, коли деякі написані одним автором символи настільки схожі один на одного, що сам автор не може за часом їх відрізнити, а тому і прочитати свій власний почерк.

Багато людей пише літери з нечітким проміжком між ними. Як правило такі літери зливаються, що ускладнює їх відокремлення один від одного. Проблема відокремлення написаних символів один від одного називається проблемою сегментування. За допомогою сегментування можливо відокремити не тільки літери в одному слові (рис. 1.2), а й слова у

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		11

реченні чи навіть рядки написаного тексту. Для вирішення проблеми сегментування різного роду рукописних об'єктів, приклад яких зазначено вище, була розроблена велика низка методів та засобів.

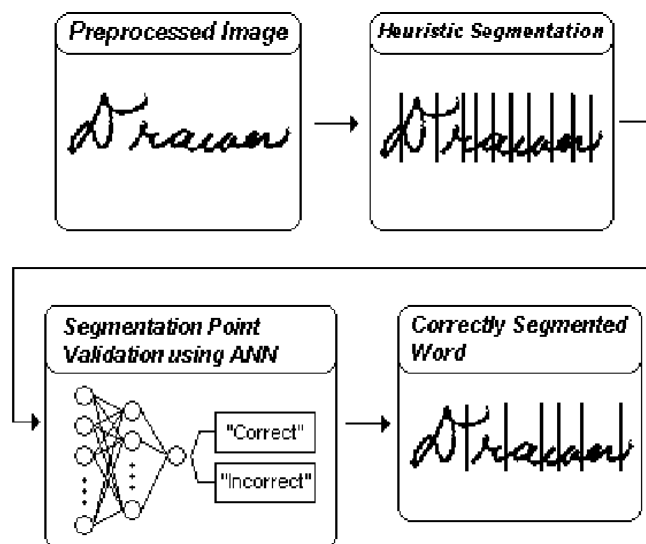


Рис. 1.2 - Приклад роботи сегментації літер у слові [9]

Коротко опишемо один із найпростіших методів сегментації під час off-line розпізнавання для загального розуміння поняття сегментації. Як вже було зазначено у цьому розділі, на вхід такої системи подається саме зображення чи фотографія рукописних символів чи тексту. Якщо це сегментація рядків, то після попередньої обробки зображення аналізують густину заповнених пікселів по усій множині можливих горизонтальних ліній. Очевидно, що рядок тексту матиме набагато більшу густину заповнених пікселів порівняно з рядками пробілу. Таким чином можливо отримати функцію залежності густини від номеру рядка, в якій можна дослідити локальні мінімуми та їх оточення. Аналогічним чином можна сегментувати слова у реченні та літери у слові, але вже по вертикальним лініям. Такий самий метод можна використовувати і при on-line підході розпізнавання, але з певними змінами через різницю у представленні даних.

Наприклад, можна аналізувати кількість точок вхідного сигналу для кожного дискретного відрізка.

Окрім проблеми невиразних пробілів між символами є і проблема схожості та контексту. Деякі символи, такі як «I» та «1» (один), «O» (літера) та «0» (нуль), «l» (мала літера «L») та «1» (один), «5» та «S», «6» та «G», виглядають схожими або навіть однаковими на письмі в залежності від стилю письма та почерку автора. Відрізнити їх інколи навіть людині складно. Зрозуміти чи то «O» (літера), чи то «0» (нуль) написано на поверхні можливо лише якщо відомий контекст, у якому такий символ зображений. Якщо цей символ зображений у складі якогось слова, то це – літера. Хоча відрізнити чи вона заголовна, чи мала можна лише по відношенню до кожного символу у слові розміру. Якщо він зображений у складі певного числа, то це – нуль. А якщо символ відокремлений, не є у складі слова чи числа, то зрозуміти його значення можливо лише знаючи середовище, в якому цей символ зображений. Визначення сенсу рукописного символу є завданням інтерпретації рукописі. Інтерпретація рукописного тексту – це дуже важливе та складне завдання, проте не є темою цієї роботи. Тому в рамках цієї роботи ми обмежимося лише неформальним визначенням цієї проблем, описаним на початку цього розділу.

Інколи рукописний символ не має власного представлення у вигляді єдиного Unicode символу. Він може бути вигаданим певним автором та використовуватися як його особистий авторський символ (рис. 1.3), а отже і мати власну назву, наприклад, «My Symbol». У таких випадках система розпізнавання потребує мати можливість додавати нові символні образи до списку тих, що можуть бути розпізнані системою.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		13

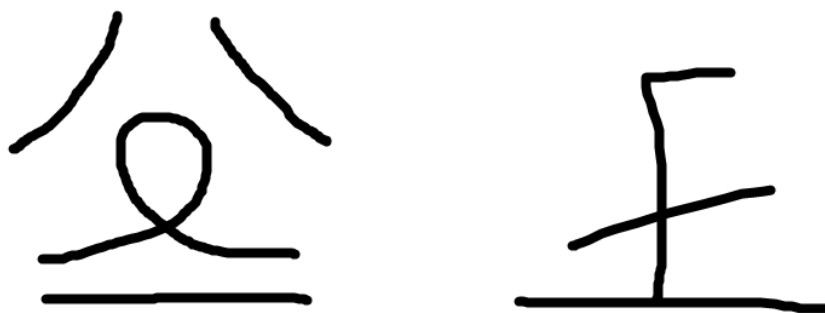


Рис. 1.3 - Приклад двох унікальних авторських символів

Варто зауважити, що така можливість є специфічною і обов'язково має бути оговорена під час проектування системи. Така проблема є яскравим прикладом, коли використання штучних нейронних мереж за прямим використанням є незручним, адже додавання нового класу символів до списку відомих мережі потребує переробки набору даних для тренування, що тягне за собою повторне тренування мережі. Повторне тренування мережі потребує великий обсяг часу та обчислювальної потужності.

Розпізнавання рукописного символу все ще вважається відкритою проблемою для дослідження завдяки значним індивідуальним варіаціям зовнішнього вигляду одного і того самого символу.

#### 1.4 Пов'язані роботи

Розпізнавання рукописного тексту – це задача, яка має довгу історію дослідження, для вирішення якої було застосовано велика кількість різноманітних технологій, включаючи скінченні автомати [10], приховані Марковські моделі (Hidden Markov Models, HMMs) [11], різноманітні нейронні мережі [6][12], основані на геометричних особливостях письма статистичні класифікатори [13] та співставлення з шаблоном (template



matching) [14]. Ці технології використовувалися як в on-line, так і в off-line підходах.

Витончені алгоритми розпізнавання на основі штучних нейронних мереж[6][12] вимагають численних підготовлених екземплярів для тренування і є непридатними для використання, якщо набір символів не може бути визначений заздалегідь перед використанням. До того ж, практика показує, що машинне навчання може бути досить непрозорою темою для великої кількості програмістів, і співставлення зі шаблонами (template matching) не є винятком.

Деякі системи розпізнавання використовують тільки певні геометричних властивості кінцевого написаного символу. Прикладами таких властивостей є описаний прямокутник до символу або опуклість кривої [15]. Хоча такі системи можуть обробляти різні складні геометричні фігури, але вони не здатні обробити таку, наприклад, як рукописний ієрогліф.

Використаний у цій роботі метод вирішення задачі розпізнавання рукописного образу може бути використаний в ситуаціях, коли набір рукописних символів, що необхідно розпізнавати, заздалегідь невідомий; може швидко та без зайвих перетренувань додати новий символ до списку відомих системі; є легкий та прозорий для розуміння; який можна швидко реалізувати; який ефективно працює на невеликій кількості відомих символів.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		15

## ВИСНОВКИ ДО РОЗДІЛУ 1

Розпізнавання рукописного тексту - це завдання перетворення мови, перенесеної у власній просторовій формі графічних знаків, у символічне подання. Від виду вхідних даних до системи розпізнавання залежить підхід щодо попередніх обчислень та подальшу обробку даних. Існують два різних видів розпізнавання: off-line та on-line.

Off-line методи розпізнавання виконують свої передбачення, використовуючи зображення рукописного символу чи тексту.

On-line методи розпізнавання характеризуються тим, що вхідні дані представляються у вигляді послідовності штрихів, які в свою чергу є послідовністю написаних точок  $(x, y, t)$  з позицією точки  $(x, y)$  та моментом її написання  $t$ [6].

Для вирішення задачі розпізнавання рукописного тексту історією було запропоновано багато методів вирішення. В даній роботі описується on-line метод розпізнавання рукописної фігури, образу чи символу, список яких може бути невідомим заздалегідь. Вона дозволяє легко додати новий символ, образ чи фігуру до списку відомих системі. До того ж, така система є легкою та прозорою для розуміння та освоєння програмістом. Вона не потребує специфічних знань в області машинного навчання, а тому є конкурентоспроможним рішенням.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		16

## РОЗДІЛ 2.

### ОПИС МАТЕРІАЛІВ ТА МЕТОДІВ РІШЕННЯ

Як вже описано у першому розділі, методи розпізнавання рукописного тексту класифікуються на off-line та on-line розпізнавання. Розпізнавання рукописного тексту в режимі off-line виконується з використанням растрового зображення рукописного тексту. On-line розпізнавання рукописного тексту, в свою чергу, використовує сигнал, що описує траєкторію пера на електронній сенсорній поверхні чи за допомогою спеціального електронного пера.

Наша система використовує on-line метод розпізнавання рукописного образу чи символу. Тому даний розділ детально розглядає особливості on-line методів, їх теоретичне підґрунтя та модель рукописного образу.

#### 2.1 Модель рукописного образу

Усі письмові мови містять у собі певну множину символів, що дозволяють описувати цілі слова (наприклад, китайські ієрогліфи) або лише його частину (тобто, літера), та множини символів для пунктуації. Рукописний символ зазвичай складається з певної кількості різних штрихів, виконаних автором. Тут штрих - це шлях кінчика пера з моменту його дотику з поверхнею письма до моменту їх розриву. Літера, ієрогліф чи символ пунктуації може складатися з декількох штрихів. При чому в деяких мовах, наприклад, у китайської та японської, порядок написання та напрямок написання штрихів є невід'ємною частиною зображуваного символу (рис. 2.1). У таких випадках під час розпізнавання символів

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		17

порядок написання штрихів та їх напрямок також використовують як характеристику символів. Проте у більшості мов, що містять у собі алфавіт, такі особливості написання не є значущими, що робить задачу складнішою, адже кількість варіацій на один символ неймовірно велика.

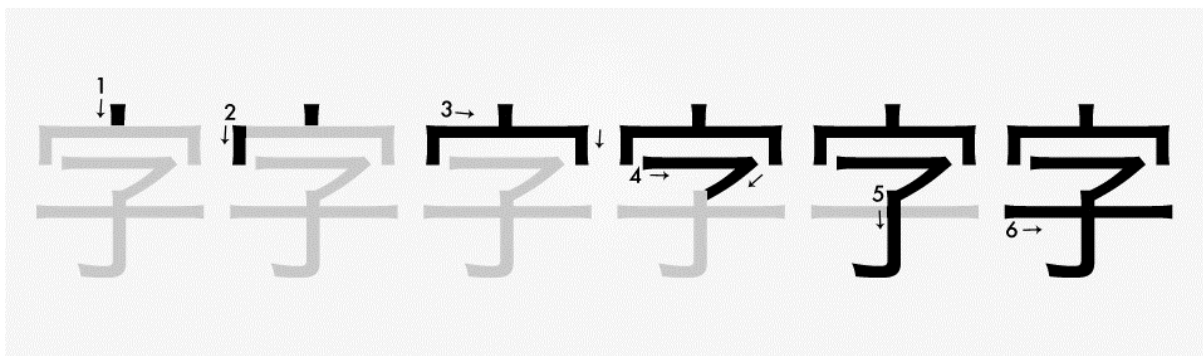


Рис. 2.1 - Порядок написання китайського ієрогліфу

Основний принцип розпізнавання, що лежить у основі більшої частини існуючих методів вирішення цієї задачі, полягає в тому, що різниця між різними символами більша, ніж різниця між кількома варіаціями одного і того ж символу. Приклад різного написання букви «R» можна побачити на рис. 2.2. Однак, бувають і винятки, приклади яких були зазначені у першому розділі. Деякі символи, такі як «O» (літера) та 0 (нуль), як правило, виглядають однаковими, відрізнити які можливо тільки за допомогою контексту, в якому вони знаходяться.



Рис. 2.2 - Різне написання однієї літери

Якщо різниця між різними символами більша, ніж різниця між кількома варіаціями одного і того ж символу, то це означає, що, маючи певні критерії для оцінки схожості чи, навпаки, розбіжності символів, можна визначити, який рукописний образ є на вході системи. Для цього необхідно для початку визначити цифрове представлення рукописного образу. Проаналізувавши це представлення можна спробувати виділити саме ті критерії, по яким можна здійснювати порівняння.

On-line підхід розпізнавання рукописного символу характеризується тим, що вхідні у систему дані можна представити у вигляді послідовності штрихів, які в свою чергу є послідовністю написаних точок  $(x, y, t)$  з позицією точки  $(x, y)$  та моментом її написання  $t$  [6].

Таким чином, ми моделюємо неперервний вхідний сигнал рукописного введення літери за допомогою електронної сенсорної поверхні чи спеціального електронного пера. Цей сигнал відображає траєкторію пишучого пера, що дотикається до поверхні.

Тепер сформуємо формальне математичне представлення такого сигналу. Його запис має вигляд:

$$X(t) = (x(t), y(t))^T,$$

де  $X(t)$  – сигнал пишучого пера чи поверхні;

$t$  – момент часу, в який була намальована точка, з проміжку  $T$ ;

$x(t)$  та  $y(t)$  – відповідні координати точки у момент часу  $t$ ;

$T$  – проміжок часу, на якому існує сигнал  $X(t)$ .

Якщо електронна сенсорна поверхня чи спеціальне електронне перо має можливість вимірювати тиск, що створюється письменником під час письма, то до описаного сигналу додається простір значень тиску  $p(t)$  [8]:

$$X(t) = (x(t), y(t), p(t))^T$$

Проте передати неперервний сигнал за допомогою цифрових пристроїв неможливо, а тому він проходить етап дискретизації. Після

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		19

процесу дискретизації сигнал  $X(t)$  розглядається як послідовність значень відповідного виміру сигналу, що обрані для кожної різної дискретної точки часу  $t \in T$ .

Для подальшого викладу змісту цієї роботи необхідно ввести такі поняття як нормальна вибірка сигналу та рівномірна вибірка значень, якими ми будемо оперувати усю наступну частину пояснювальної записки.

Нормальна вибірка вхідного сигналу – це послідовність значень сигналу, що були зафіксовані у різні дискретні моменти часу, але з незмінним часовим інтервалом.

Визначимо поняття нормальної вибірки формально. Нехай з вхідного сигналу  $X(t)$  маємо вибірку  $t_0, t_1, \dots, t_{m-1}$  таку, що задовольняє нерівність  $0 \leq t_0 < t_1 < \dots < t_{m-1}$ . Якщо часові точки розташовані на однаковій відстані один від одної, тобто  $|t_i - t_{i-1}| = \tau$  для всіх  $i \in \{1, 2, \dots, m\}$  та деякої константи  $\tau > 0$ , то такий вхідний сигнал ми називаємо нормальною вибіркою.

Визначимо поняття рівномірної вибірки формально. Нехай  $d(X(t_{i-1}), X(t_i)) = \sqrt{(x(t_i) - x(t_{i-1}))^2 + (y(t_i) - y(t_{i-1}))^2}$  буде Евклідовою відстанню по координатах  $x(t)$  і  $y(t)$  між точками  $X(t_{i-1})$  та  $X(t_i)$ . Тоді вибірка значень, що задовольняє рівність  $d(X(t_{i-1}), X(t_i)) = \delta$  для деякої константи  $\delta > 0$  та  $i \in \{1, \dots, n\}$ , називається рівномірною вибіркою сигналу  $X(t)$ , при чому  $t_n \leq t_m$ .

Графічне зображення нормальної вибірки та рівномірної вибірки одного і того самого сигналу можна порівняти на рис. 2.3.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		20

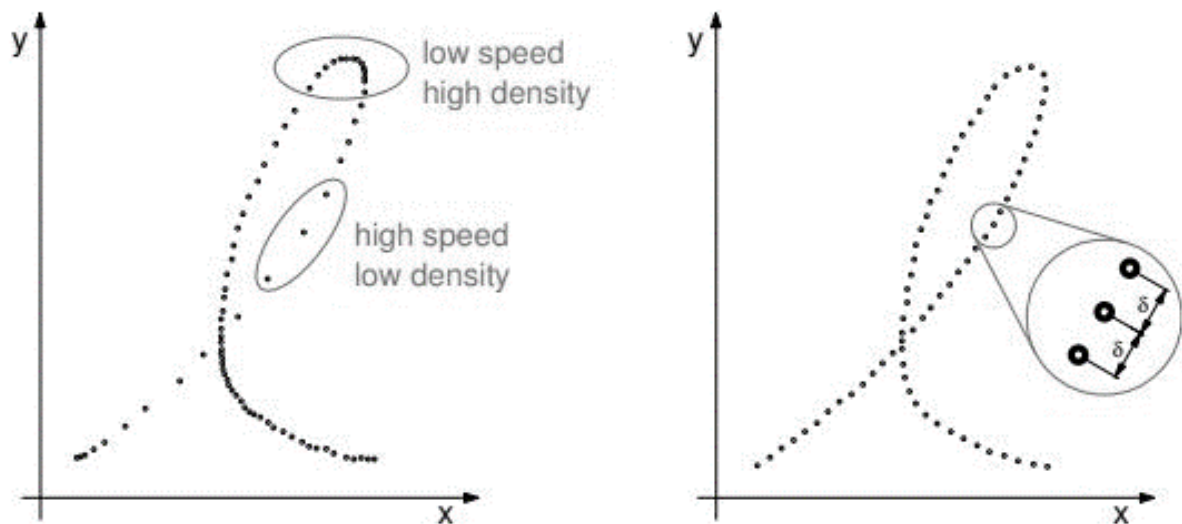


Рис. 2.3 - Приклад нормальної вибірки вхідного сигналу (ліворуч) та його рівномірної вибірки (праворуч) [8]

Кожна з описаних вибірок має свої переваги та недоліки. На рис. 2.3 чітко помітна різниця між двома вибірками вхідного сигналу. Нормальна вибірка зберігає таку особливість рукопису як швидкість письма: чим більша купчастість послідовних точок у певному регіоні двовимірного простору, тим менша швидкість письма лінії у ньому, і навпаки. Для прикладу, на рис. 2.3 нормальної вибірки помічені характерні регіони швидкого та повільного письма. Проте рівномірна вибірка краще зберігає форму написаного символу, хоч і не відображає швидкість письма.

Різні методи вирішення задачі розпізнавання символу з вхідного сигналу потребують різні види його представлення. Наприклад, штучна нейронна мережа може вилучити певні особливості рукописного символу завдяки збереженню такої характеристики, як швидкість письма, і використовувати її для задачі ідентифікації почерку.

Отже, маємо модель цифрового представлення рукописного символу чи образу, яку тепер можна аналізувати. Тепер необхідно описати процес аналізу цієї моделі, що дозволить нам класифікувати (тобто розпізнати) вхідний сигнал.

## 2.2 Процес розпізнавання рукописного образу

Згадуючи перший розділ, процес розпізнавання рукописного тексту можна розділити на кілька загальних кроків [8]:

- попередня обробка (*preprocessing*);
- вилучення особливостей (*feature extraction*)
- класифікація (*classification*).

Розглянемо кожний цей крок окремо.

### 2.2.1 Попередня обробка

Мета попередньої обробки полягає в тому, щоб відкинути неважливу, несуттєву чи надлишкову інформацію у вхідних даних, яка може негативно вплинути на процес розпізнавання [8]. На попередню обробку припадає велика частина обчислень під час розпізнавання. До її складу може входити нормалізація, відбір рівномірної вибірки чи нормальної вибірки вхідного сигналу, згладжування та зменшення шуму [8]. Усе для того, щоб зменшити відмінність між написанням одного і того ж самого символу. Розглянемо кожний етап окремо.

Процес нормалізації вхідного сигналу полягає у вилученні варіації розміру вхідного символу та інформації про його відносну позицію з вхідних даних. До нього обов'язково входить такі перетворення як масштабування та зміщення.

Масштабування під час попередньої обробки вхідних даних необхідний для того, щоб автор чи письменник символу, який задається на розпізнавання, мав повну свободу у виборі його розміру. Так як кожна людина може писати символи у різному розмірі (наприклад, хтось ширше,

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		22



хтось вище, а хтось, навпаки, пише символи дуже дрібно), необхідно звести кожний символ до одного і того самого розміру. Така процедура прибирає варіативність розмірів символів перед початком їх розпізнавання, зводячи їх усіх до одного базового розміру, який очікує система розпізнавання.

Зміщення – це один із процесів попередньої обробки, необхідний для того, щоб представляти вхідні дані однаково відносно одного і того самого початку координат. Потреба у такому перетворенні вхідних даних зумовлена тим, що написаний символ може розміститися у будь-якій зоні сканування сигналу. Наприклад, зверху електронної сенсорної панелі, знизу, десь в кутку, посередині тощо. Тому вхідні дані необхідно змістити до одного і того самого початку координат.

Також до попередньої обробки вхідних даних може входити процес кореляції нахилу письма. Корекція нахилу, чи інколи називають корекцією зміщення базової лінії [16], - це процес, що спрямований на зведення напрямку чи нахилу письма до горизонтальної лінії.

Проте сучасні пристрої для введення рукописного символу чи образу (сенсорні електронні екрани) мають певного роду підказки чи індикатори, які вказують на те, де необхідно написати літеру, слово чи речення. Вони можуть бути у формі ліній, на яких необхідно писати, або навіть певного прямокутного контуру, який обмежує зону розпізнавання. Тому, зазвичай, рукописний символ чи образ має правильну орієнтацію і його не потрібно повертати перед початком розпізнавання. Але якщо це не так, і користувачеві надається можливість писати у будь-якому напрямку та під будь-яким кутом, то вхідні дані можуть мати певний кут перекосу, який необхідно виправити.

Якщо кореляція нахилу має місце під час попередньої обробки даних, то це означає, що необхідно визначити кут  $\alpha_0$ , який вказує на скільки лінія основи та напрямок написаного слова чи речення нахилені

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		23

відносно горизонтальної лінії. Визначивши цей кут та повернувши вхідні дані на значення цього кута, кожна літера слова чи речення будуть нахилені на вертикальний кут  $\varphi_0$ . Цей кут також необхідно визначити та вирівняти відносно вертикальної осі.

Приклади описаних вище кутів для кореляції зображені на рис. 2.4.

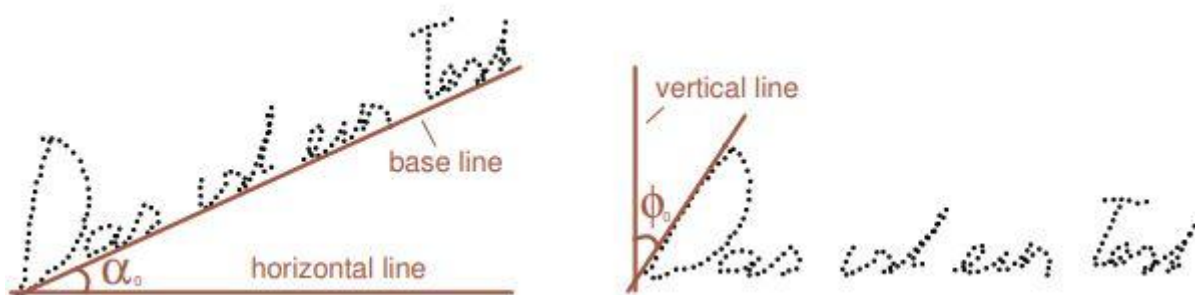


Рис. 2.4 – Візуалізація кутів помилки  $\alpha_0$  (ліворуч) та  $\varphi_0$  (праворуч) при корекції. [8]

Варто зазначити, що процедура корекція куту нахилу можлива лише тоді, коли вхідні дані формують слово чи речення, де можливо визначити загальний кут нахилу усієї конструкції. Якщо вхідні дані формують один єдиний символ, то кут його нахилу визначити неможливо. До того ж, цей кут нахилу може бути єдиним показником, що відрізняє два різних символів. До такого прикладу можна віднести літери англійського алфавіту «d» та «p», «b» та «q», велика літера «A» та математичний символ «V», або літери «U» та «C».

Ще одним важливим етапом попередньої обробки вхідних даних є відбір рівномірної вибірки.

Частіше за все процедура відбору рівномірної вибірки, яка іноді називається фільтруванням[16], виконується після нормалізації і також може бути останнім етапом попередньої обробки. Проте, досить часто попередню обробку починають саме з відбору рівномірної вибірки, оскільки це процес низької вартості, який може суттєво підвищити

продуктивність та точність інших етапів попередньої обробки. Особливо великий вплив відбір рівномірної вибірки має на методи корекції куту нахилу, які використовують проекції вхідних даних уздовж осі абсцис та ординат.

Існує, як правило, дві причини для застосування процедури відбору рівномірної вибірки. Одна з них полягає у видаленні непотрібних деталей з форми (занадто багато точок, що представляють малюнок). Інша причина - видалити артефакти з форми, такі як швидкість руху. Слід зауважити, що швидкість руху пера під час малювання символу може бути цікавою характеристикою на етапі вилучання особливостей. У цьому випадку швидкість руху пера слід видобути та зберігати перед застосуванням процедури відбору.

Зазвичай, безперервний вхідний сигнал  $X(t)$ , як правило, після дискретизації, стає дискретним сигналом, що має властивості нормальної вибірки. В результаті цього утворюється рівномірно поділена по часовому простору послідовність точок. У деяких випадках, наприклад, у підході on-line розпізнавання з спеціальною сенсорною електронною дошкою, цифрові дані не можуть бути рівномірними як у часі, так і в просторі, через похибку зчитування сигналу чи інші недоліки апаратного обладнання. Проте ціль процедури відбору рівномірної вибірки залишається незмінною, а саме відбір точок, які мають бути рівномірно розташовані в просторі, в ідеалі, не спотворюючи даних. Приклад даних до і після процесу вибірки можна побачити на рис. 2.5.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		25

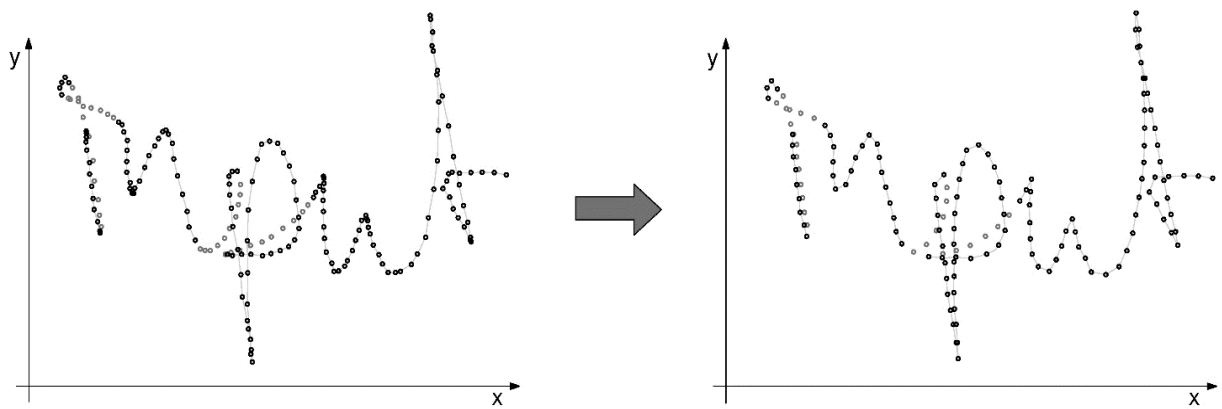


Рис. 2.5 - Візуалізація процесу повторного відбору. Нормально відібрані дані (ліворуч) та його еквівалентна рівномірна вибірка (праворуч). [8]

До процесу попередньої обробки вхідних даних також відноситься етап видалення повторних точок. Для того щоб зменшити кількість даних на вході системи, а отже пришвидшити певні етапи попередньої обробки та подальші обчислення системи, вхідний сигнал фільтрується на наявність однакових за координатами  $x(t)$  та  $y(t)$ , повторюваних точок або на наявність точок, що знаходяться в деякому невеликому околі будь-якої іншої точки. Такі точки є прикладом несуттєвої інформації.

Останніми розглянутими у цій роботі засобами попередньої обробки даних стають згладжування та зменшення шуму.

Нестійкий рух пера та тремтіння руки під час написання символу викликає тремтіння в почерку, що створює певний шум у вхідному сигналі, який заважає коректному процесу розпізнавання. Цей тип шуму видаляється за допомогою методів згладжування та зменшення шуму.

Згладжування та зменшенням шуму вхідного сигналу є однією і тією самою задачею, адже згладити сигнал означає зменшити кількість деталей, які він представляє. Шум у термінах on-line розпізнавання рукописної літери – це зазвичай невелике спотворення вхідного сигналу, викликане тремтінням руки та пишучого пера під час написання символу, або присутність незначних малопомітних надлишкових деталей у ньому,

викликаних, наприклад, випадковим дотиком до електронної сенсорної поверхні.

Зазвичай, більшість методів видалення шуму та згладжування використовують принцип зваженого усереднення, виходячи з сусідніх значень. Маючи часову послідовність значень, її можна згладити, якщо під час перетворення кожне значення нової послідовності буде зваженим середнім значення в певному околі цього значення з вхідної послідовності. Інколи, деякі методи згладжування усереднюють не окіл конкретного значення вхідної послідовності, а його попередні значення. На такому принципі працюють сімейство методів рухомого середнього.

Для видалення шуму та для згладжування вхідного сигналу можна використовувати фільтр Гаусах[17], функції сімейства рухомого середнього[17], фільтр Калмана[18] тощо.

### 2.2.2 Вилучення особливостей

Другий крок процесу розпізнавання – це вилучення особливостей письма (feature extraction).

Метою цього кроку є виділення важливої інформації для процесу розпізнавання [8]. Під час процесу вилучення особливостей з отриманих даних, оброблених кроком попередньої обробки, обраними з численної кількості методів намагаються виділити дані вищого абстрактного рівня.

Дані, отримані на етапі вилучення особливостей, можуть включати в себе таку інформацію як, наприклад, тиск пера на поверхню під час зображення символу, швидкість руху пера, зміни напрямків штрихів у символі, кількість штрихів, використаних при написанні символу, кількість закруглень, радіус закруглень, кількість кутів у символі, кількість вільних

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		27

кінців, проекції символу на вісь абсцис та на вісь ординат і будь-які інші можливі геометричні особливості символу.

Чим більше особливостей можна вилучити з вхідних даних, тим вужче множина відповідних до них символів. Виділення таких особливостей допомагає підвищити швидкість роботи наступного третього етапу розпізнавання – класифікації – та/або підвищити точність розпізнавання системи.

Проте, очевидно, що використання великої кількості шуканих особливостей призводить до підвищення складності роботи системи та сповільнює процес розпізнавання. Тому необхідно ретельно відібрати саме ті характеристики, які матимуть найкращий вплив на швидкість та точність роботи етапу класифікації. Знайти той самий баланс між швидкістю його роботи та точністю розпізнавання.

Розглянемо поверхнево методи вилучення деяких особливостей, щоб мати уявлення про роботу цього етапу.

Тиск пера на електронну поверхню письма під час зображення символу вимірюється самим електронним пером та/або самою електронною поверхнею, якщо таке апаратне забезпечення має можливість виконувати такі вимірювання, і подається на вхід системи як один із вимірів сигналу  $X(t)$ , описаного на початку цього розділу.

Для етапу вилучення особливостей цікавими є моменти часу та/або зона двомірного простору з найбільшим чи найменшим тиском пера на електронну поверхню. Можна, наприклад, визначити, що найбільше значення тиску досягалася у першій третині часового виміру, або значення тиску досягло мінімуму у нижній правій чверті двовимірного простору тощо.

Відносна швидкість руху пера під час зображення символу розраховується з вхідного сигналу як відношення пройденого шляху до часу його подолання і обов'язково розраховується до початку етапу

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		28

попередньої обробки, який може видалити таку характеристику під час своєї роботи (наприклад, після застосування методу відбору рівномірної вибірки).

Зазвичай швидкість руху пера не є суттєвою характеристикою символу під час написання, адже вона є індивідуальною для кожної людини. Проте, таку інформацію зазвичай використовують в методах машинного навчання для тренування та класифікації відповідного символу штучною нейронною мережею.

Цікавою та зазвичай корисною особливістю написаного символу є кількість штрихів, використаних для його написання. Нагадаємо, що штрих – це шлях кінчика пера з моменту його дотику з поверхнею письма до моменту розриву їх контакту. Підраховуючи кількість розривів контакту пера з поверхнею можна отримати кількість штрихів, використаних для написання символу. Маючи статистику кількості використаних штрихів на кожний з відомих системі символів, можна покращити точність розпізнавання, не сповільнюючи її роботу. Наприклад, очевидним є той факт, що для написання літери англійського алфавіту «С» найчастіше використовують лише один штрих. В той час як для написання літери «Т» – два штриха.

Іншою характерною особливістю символу може бути його проекції на вісь абсцис та вісь ординат. Проекція символу на вісь абсцис – це функція, яка відображає залежність кількості точок символу на певному фіксованому проміжку осі абсцис від цього проміжку. Аналогічно і для проекції символу до осі ординат. Якщо поділити вісь ординат та вісь абсцис на фіксовану кількість відрізків фіксованої довжини, то можна отримати гістограми відповідних проекцій символу, пара яких може характеризувати символ.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		29

### 2.2.3 Класифікація

Класифікація – це процес співставлення нових для системи вхідних даних з однією із множини відомих системі категорій (класів).

Якщо описувати у термінах розпізнавання рукописного символу, то класифікація – це процес співставлення нових для системи вхідних даних, що відображають написаний від руки символ, із цифровою текстовою назвою символу із списку відомих системі. Тобто, якщо подати на класифікацію отримані дані з етапів попередньої обробки та вилучення особливостей, то цей етап має відповісти, яку назву мають вхідні до системи дані, що відображають рукописний символ чи образ.

Етап класифікації може бути реалізованим як дуже простим (наприклад, вибір за отриманими особливостями з таблиці відповідну назву символу), так і дуже складним (наприклад, використання методів машинного навчання).

Прямим та найпростішим способом класифікувати вхідні дані системи on-line розпізнавання рукописного символу – це зіставити отримані на етапі вилучення особливостей дані з вже підготовленою таблицею відомих значень та обрати той запис у таблиці, що найбільше відповідає отриманим особливостям. Такий метод класифікації є простим у реалізації, хоча він вимагає повного проходу по усій таблиці відомих системі значень. При великому обсязі відомих системі символів та/або великій кількості вилучених особливостей пошук найкращого кандидата є повільним. До того ж, вилучені особливості не завжди однозначні для кожного символу, що створює ситуацію, в якій таблиця вже не двовимірна (кількість записів у таблиці на кількість відомих системі особливостей), а тривимірна (конкретна колонка особливості певного запису таблиці є множиною підходящих значень вказаної особливості для вказаного запису таблиці). Проте, така реалізація дозволяє додавати нові записи до таблиці

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		30



під час використання системи. Тобто, це означає, що така реалізація дозволяє створювати нові відомі для системи символи, які вона зможе розпізнавати.

Більш сучасні методи класифікації вхідних даних використовують машинне навчання. Такі системи використовують штучні нейронні мережи, які тренують на множині великої кількості спеціально підготовлених прикладів.

Штучна нейронна мережа у корні своєї сутності – це сильно нелінійна функція, яка є композицією певної кількості більш простих функцій. Такі більш прості функції у термінах машинного навчання представляють так звані функції активації слоїв мережі. Ця нелінійна функція (далі називатимемо функцією-гіпотезою) параметризована великою кількістю коефіцієнтів, які під час тренування мережі змінюються та корегуються системою так, щоб значення самої функції наближувалися до значень з таблиці істинності. Ця таблиця істинності формує відповідність вхідних тренуючих даних до бажаних значень, які штучна нейронна мережа має формувати після тренування.

З кожним «проходом» спеціально підготовленого екземпляру через процес тренування за допомогою алгоритму зворотного розповсюдження помилки усі коефіцієнти функції-гіпотези наближуються до значень, при яких так звана функція помилки (loss function або cost function) [5] наближується до нуля. Ця функція помилки характеризує на скільки точним є значення функції-гіпотези відносно значень таблиці істинності в залежності від переданих їй коефіцієнтів. Чим більше значення функції помилки, тим менша точність функції-гіпотези при зазначених коефіцієнтах. З кожним наступним кроком чи ітерацією процесу тренування, значення функції помилки мають спадати. Якщо значення функції помилки протягом процесу тренування зростають, то це сигналізує про помилки у процесі тренування.

Натреновані коефіцієнти зумовлюють здатність штучної нейронної мережі класифікувати вхідні дані до відповідного класу подібних символів. Такий підхід з використанням штучного інтелекту дозволяє зменшити етап вилучення особливостей або взагалі від нього відмовитися. Це зумовлено тим, що під час тренування штучна нейронна мережа скорегує свої коефіцієнти так, що її функції активації будуть автоматично виділяти найважливіші для неї особливості вхідних даних.

У випадку використання штучного інтелекту для розпізнавання символів існує певна кількість обмежень.

По-перше, штучна нейронна мережа потребує велику множину екземплярів для тренування. Чим більша ця множина даних та чим більш вона різноманітна, тим ефективніше відбуватиметься процес тренування. Створення такої вибірки даних є непростим завданням.

По-друге, штучна нейронна мережа може класифікувати вхідні дані тільки до тих, на які вона натренована. Додавання нових символів чи образів до списку відомих штучному інтелекту потребує корегування множини екземплярів для тренування та повторного процесу тренування.

По-третє, тренування штучної нейронної мережі потребує значного часу та обчислювальних можливостей. Тому корегування списку відомих символів кінцевому користувачу такої системи є неможливим.

Штучні нейронні мережі дозволяють досягнути великої точності у процесі розпізнавання рукописних символів, яку можна порівняти з людською здатність їх впізнавати. Проте їх використання є ефективним у тому випадку, коли множина символів, яку система здатна розрізнити, відома заздалегідь і не змінюється під час роботи.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		32

## ВИСНОВКИ ДО РОЗДІЛУ 2

On-line підхід розпізнавання рукописного символу характеризується тим, що вхідні у систему дані можна представити у вигляді послідовності штрихів, які в свою чергу є послідовністю написаних точок  $(x, y, t)$  з позицією точки  $(x, y)$  та моментом її написання  $t$  [6].

Таким чином, ми моделюємо неперервний вхідний сигнал рукописного введення літери за допомогою електронної сенсорної поверхні чи спеціального електронного пера. Цей сигнал відображає траєкторію пишучого пера, що дотикається до поверхні. Маючи модель цифрового представлення рукописного символу чи образу, можна починати аналізувати вхідний сигнал.

Процес розпізнавання рукописного символу в on-line підході можна розділити на кілька загальних кроків [8]:

- попередня обробка;
- вилучення особливостей;
- класифікація.

Мета попередньої обробки полягає в тому, щоб відкинути неважливу, несуттєву чи надлишкову інформацію у вхідних даних, яка може негативно вплинути на процес розпізнавання [8]. Наприклад, до такого виду інформації можна віднести швидкість написання та тремтіння пера під час проведення штриха. Цей крок дозволяє зменшити відмінність між написанням одного і того ж самого символу.

Метою процесу вилучення особливостей є виділення важливої інформації для розпізнавання [8]. Дані, отримані на цьому етапі, можуть включати в себе таку інформацію як, наприклад, тиск пера на поверхню під час зображення символу, зміни напрямків штрихів у символі, кількість штрихів, використаних при написанні символу, кількість закруглень,

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		33

радіус закруглень, кількість кутів у символі, кількість вільних кінців, проекції символу на вісь абсцис та на вісь ординат і будь-які інші можливі геометричні особливості символу. Чим більше особливостей можна вилучити з вхідних даних, тим вужче множина відповідних до них символів.

Останній крок – класифікація. На цьому кроці і відбувається відображення оброблених даних до відповідного йому Unicode представленню. Результат роботи цього кроку і є результатом роботи системи.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		34

### РОЗДІЛ 3. РЕАЛІЗАЦІЯ РІШЕННЯ ЗАДАЧІ

У другому розділі ми описали модель рукописного символу чи образу та розглянули процес його розпізнавання з детальним викладом мети та призначення кожного з кроків процесу. Враховуючи весь викладений матеріал вище, настав час описати нашу реалізацію системи on-line розпізнавання рукописних символів.

На вхід системи on-line розпізнавання подається цифровий сигнал від електронної поверхні чи спеціального електронного пера. Тому цільова платформа нашої системи – це мобільні пристрої, що мають сенсорний екран. Проте теоретично основа нашої системи може бути використана і на інших платформах, які можуть отримати сигнал з електронної поверхні чи спеціального пера.

Такий вибір цільової платформи зумовлений великою популярністю смартфонів та планшетів у світі, а також дуже активною розробкою для них програмного забезпечення. Дуже велику частину такого програмного забезпечення складають ігри для мобільних пристроїв та прикладні програми для веб-сервісів. Кожна з цих категорій однозначно може мати намір застосовувати певні механізми розпізнавання рукописних символів чи образів, що створюються на екранах мобільних телефонів, планшетів тощо.

Велика кількість сучасні методів розпізнавання рукописних символів, такі як приховані Марковські моделі [11], різноманітні нейронні мережі [6][12], статистичні класифікатори [13], зазвичай вимагають значних технічних знань для їх проектування та розробки систем з їх використанням. Проте переважна більшість команд розробки програмного забезпечення на мобільні пристрої у зазначених вище категорія не мають

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		35

таких знань і не готові витратити свої ресурси на підготовку свого особового складу.

Наша робота пропонує недорогий, простий для розуміння, швидкий у розробці, проте ефективний підхід для розпізнавання рукописних символів чи образів на невеликій множині відомих системі категорій. Цей підхід включає лише прості геометричні обчислення та зрозуміле внутрішні представлення. До того ж, наша система має можливість додавати нові символи до списку тих, що вона може розпізнати. Така можливість є надзвичайно важливою і, мабуть, ключовою в нашій системі.

Отже, на вхід подається послідовність цифрових сигналів  $X(t)$ , яка відображає послідовність написаних пером штрихів. Кожний штрих є послідовністю написаних точок  $(x, y, t)$  з позицією точки  $(x, y)$  та моментом її написання  $t$ .

Як зазначалося у цій роботі раніше, процес розпізнавання рукописної літери умовно розділяють на три етапи: попередня обробка вхідних даних, вилучення особливостей вхідних даних та їх класифікація. Проте наша система не використовує жодний з заявлених методів вилучення особливостей і пропускає цей крок у зв'язку з специфікою класифікаційного методу. Цей метод буде описаний нижче.

### 3.1 Реалізація процесу розпізнавання

На початку роботи системи вхідні дані у вигляді послідовності написаних штрихів, які в свою чергу є послідовністю написаних точок  $(x, y, t)$  з позицією точки  $(x, y)$  та моментом її написання  $t$ , проходять через етап попередньої обробки, щоб видалити несуттєву та надлишкову інформацію з даних та привести до загально виду, який система може обробляти та аналізувати.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		36

Наша система не вирішує проблему розпізнавання рукописного тексту, речень, слів, їх інтерпретацію тощо. Вона вирішує проблему розпізнавання рукописної літери, а тому не включає в себе етап сегментації. Процес сегментації літер та символів у слові є достатньо цікавою задачею та є обов'язковим етапом для систем розпізнавання тексту. Хоча ця проблема не є темою нашої роботи, проте ми навели її поверхневий огляд у першому розділі.

В створеній системі етап попередньої обробки включає в себе відбір рівномірної вибірки та її нормалізацію, тобто масштабування та зміщення позицій точок вхідного сигналу до початку відліку двовимірної просторової системи координат.

В даній роботі крок відбору рівномірної вибірки виконується на початку попередньої обробки вхідного сигналу. Під час відбору точки кожного штриха проходять через функцію, яка відображає  $m$  точок нормальної вибірки на  $n$  точок рівномірної вибірки, де  $n < m$ . Кількість точок  $n$  у рівномірній вибірці є сталою величиною і впливає швидкість роботи системи. Ця величини обиралася експериментальним шляхом, результати яких будуть наведені далі у роботі.

За визначенням, описаним у другому розділі, рівномірна вибірка у контексті нашої задачі з вхідного сигналу – це така послідовність точок для кожного штриха, Евклідова відстань між позицій яких є однаковою для кожної пари сусідніх по часу точок штриха. Зазвичай, множина точок з вхідного сигналу не є достатньо повною для того, щоб обрати з неї рівномірну вибірку. Проте її можна створити, враховуючи траєкторію руху штрихів. Тому необхідно реалізувати такий алгоритм відбору рівномірної вибірки, який би це враховував.

Насправді часовий вимір вхідного сигналу для нас не є суттєвим. Важливим є лише порядок точок, в якому вони були зображені. Це, знову ж таки, зумовлено специфікою нашого класифікаційного методу. Тому в

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		37

нашій системі ми не фіксуємо час написання точки, ми фіксуємо лише їх порядок. Причина такого рішення буде пояснена у роботі далі, коли розглядатимемо реалізований класифікаційний метод.

Таким чином, щоб створити рівномірну вибірку, враховуючи траєкторію руху штрихів, необхідно для початку визначити ту відстань, на якій будуть розташовані їх точки. Знаючи сталу кількість точок рівномірної вибірки, не складно вирахувати цю відстань. Необхідно лише поділити значення шляху пера, що утворює зображення символу, на кількість відрізків між точками, необхідних для утворення рівномірної вибірки. Тобто,  $n - 1$ , де  $n$  – це кількість точок рівномірної вибірки. Таким чином ми отримуємо відстань, на якій мають бути розташовані точки у штрихах рівномірної вибірки.

Знаючи відстань між точками рівномірної вибірки, необхідно створити рівномірну вибірку, враховуючи траєкторію штрихів. Для цього послідовність точок кожного штриха можна було б інтерполювати та обрати на отриманій функції рівновіддалені точки, проте значення позицій цих точок не залежать один від одного, а залежать від іншої змінної. Пошук рівновіддалених точок  $(x, y)$  у такому випадку є аналітичною задачею, яку, нажаль, не кожний програміст може вирішити. До того ж, такий метод створення рівномірної вибірки не є ефективним через надлишкові обчислення для інтерполяції, тому ми пропонуємо використати більш простий та ефективний метод відбору рівномірної вибірки.

Основна ідея реалізованого методу відбору рівномірної вибірки полягає у тому, щоб видалити усі точки, у яких шлях по траєкторії штриха від останньої відібраної для рівномірної вибірки точки є меншим за необхідну. Перша точка, шлях до якої від останньої відібраної є не меншим за необхідну відстань, клонується, зсувається та додається до рівномірної вибірки. Зсув цієї точки відбувається на різницю між

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		38



необхідною для рівномірної вибірки відстанню та шляхом від останньої відібраної точки. Процес продовжується з новоствореної точки і повторюється до тих пір, поки не він не пройде усі точки вхідних даних. Перша точка штриха одразу додається до рівномірної вибірки, з якої і починається процес відбору.

Алгоритм відбору рівномірної вибірки у довільній формі наведений далі у цій роботі у підрозділі опису алгоритмів.

Після отримання рівномірної вибірки, ці дані проходять нормалізацію для видалення варіативності розмірів, розтягування чи стискання та позиції написаного символу чи образу. Таким чином, подальший процес розпізнавання позбувається залежності від вище зазначених характеристик зображеного символу.

В даній роботі крок нормалізації складається тільки з масштабування та зміщення.

Під час масштабування ширина та висота описуючого прямокутника, що описує усі точки вхідних даних, зводяться до співвідношення 1:1, а величина його ширини та висоти зводиться до фіксованого в системі значення. Це значення може бути будь-яким. Важливо лише те, щоб воно залишалось незмінним, фіксованим з моменту першого запуску системи. В нашій системі це значення дорівнює одиниці. Після або до масштабування вхідні дані зміщуються до початку координат так, щоб точки з найменшою позицією по відповідних координатах опинилися на відповідній осі. Таким чином, після кроку нормалізації значення координат точок вхідних даних знаходяться в діапазонах  $x \in [0..1]$  ,  $y \in [0..1]$  . Порядок виконання масштабування чи зміщення вхідних даних не важливий.

Алгоритми нормалізації, масштабування та зміщення наведені далі у цій роботі у підрозділі опису алгоритмів.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		39

Наступний етап, найбільший етап нашої системи, останній етап процесу розпізнавання рукописного символу чи образу, – етап класифікації.

Так як однією з вимог до нашої системи розпізнавання рукописного символу є можливість легко додавати нові символи до списку тих, що вона може розпізнати, ми вирішили використати метод класифікації, що можна віднести до категорії методів співставлення з шаблоном. Тобто для того, щоб додати новий символ до системи, необхідно сформувавши для нього шаблон, з яким система буде порівнювати у подальшій своїй роботі. Зазвичай, класифікаційний етап використовує вилучені корисні особливості з попередньо оброблених вхідних даних, щоб визначити категорію, до якої відноситься написаний символ чи образ. Проте часто вилучені особливості з символу чи образу не є однозначними для його характеристики і їх використання для формування нового шаблону системи може їй зашкодити. Наприклад, порядок та напрямки штрихів можуть відрізнятися серед користувачів системи, що пишуть один і той же самий символ чи образ. Ілюстрацію такого прикладу можна побачити на рис. 3.1. В такому випадку деякі методи класифікації мають перебрати всі можливі перестановки заданого багато штрихового символу, що спричиняє сильний приріст як у пам'яті, так і у часі виконання. Наприклад, 2-штриховий символ, такий як літера англійського алфавіту «X», має 8 перестановок, враховуючи порядок написання штрихів та їх можливі напрямки. Або 4-штриховий символ, наприклад, звичайний квадрат має  $4! \times 2^4 = 384$  різних перестановок. Або куб – символ, який зазвичай намальований з 9 штрихів, – буде мати у системі  $9! \times 2^9 = 185,794,560$  перестановок. Це надзвичайно велике число перестановок, яке необхідно врахувати під час класифікації, навіть для сучасних ноутбуків, не кажучи вже про мобільні пристрої, які обмежені у своєму апаратному забезпеченні.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		40

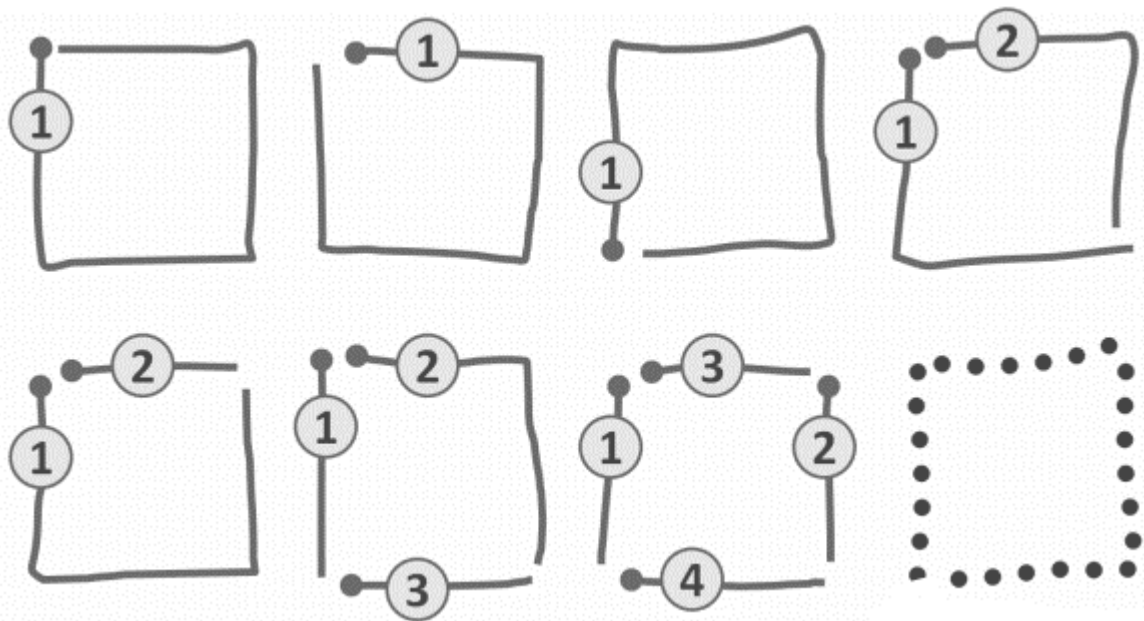


Рис. 3.1 - Навіть простий квадрат може бути нарисований багатьма способами. Однак усі артикуляції ігноруються, якщо дивитись на квадрат як на хмару точок. [19]

Для вирішення вищезазначених проблем ми представляємо символи у вигляді так званої "хмари точок" і, таким чином, ігноруємо змінну поведінку користувачів з точки зору порядку написання та напрямку штрихів (див. рис. 3.1). Якщо ігнорувати часовий вимір вхідного сигналу, тоді кількість використаних штрихів, їх порядок та напрямок написання стають нерелевантними. Залишається тільки просторова форма символу чи образу. Далі у цій роботі множину точок, що утворює відібрана рівномірна вибірка, будемо називати хмарою точок. Таке визначення акцентує увагу на форму, що утворена рівномірною вибіркою, а не на послідовність чи напрямок штрихів, використаних для зображення цієї форми.

Якщо система матиме певну кількість шаблонів, що будуть представляти список відомих їй символів чи образів, у вигляді таких хмар точок, то у такому випадку вона має лише зіставити кожну точку хмари попередньо обробленого символу-кандидата  $S$  з кожною точкою хмари символу-шаблону  $T$  та обчислити відповідну помилку зіставлення. За

традиційним підходом жадібних алгоритмів назва символу, що представлений у системі шаблоном  $T$ , який має найменшу помилку зіставлення з кандидатом  $C$ , і є результатом нашого класифікаційного методу.

Визначимо зіставлення (відповідність)  $M$  між двома хмарами точок  $C$  і  $T$  як множину пар точок, в якій кожна точка  $C_i \in C$  пов'язана з певною точкою  $T_j \in T$ . Тобто  $T_j = M(C_i)$  та  $C_i = M(T_j)$ . Якщо обидві хмари точок  $C$  та  $T$  є рівномірними вибірками з однаковою кількістю точок  $n$ , то відповідність  $M$  також буде складатися з точно  $n$  пар точок.

Визначимо помилку для зіставлення  $M$  як суму евклідових відстаней між точками для всіх пар відповідності  $M$ :

$$\sum_{i=1}^n \|C_i - T_j\| = \sqrt{(C_i^{(x)} - T_j^{(x)})^2 + (C_i^{(y)} - T_j^{(y)})^2},$$

де  $T_j = M(C_i)$ ;

$C_i^{(x)}$  та  $C_i^{(y)}$  – відповідні координати точки  $C_i$ ;

$T_j^{(x)}$  та  $T_j^{(y)}$  – відповідні координати точки  $T_j$ .

У цьому рівнянні індекс  $j$  залежить від  $i$ , так як помилка зіставлення рахується у контексті створеної відповідності  $M$  і пояснюється рівністю  $T_j = M(C_i)$ . Але для полегшення позначення ми виконуємо суму по  $i$  та відкидаємо додаткові формальності нотації, враховуючи, що точка  $C_i$  з хмари  $C$  має одну і лише одну відповідну точку  $T_j$  з хмари  $T$  за допомогою деякого алгоритму зіставлення, що реалізує  $M$ .

У цій роботі для співставлення точок з однієї хмари до іншої ми реалізували алгоритм жадібного співставлення зваженого точок, спираючись на роботу [20] та використовуючи запропоновані евристики у [21].

Спочатку наша система реалізовувала звичайний алгоритм жадібного співставлення. Його роботу поверхнево можна описати наступним чином. Для кожної точки  $C_i$  з першої хмари  $C$  необхідно знайти найближчу точку  $T_j$  з іншої хмари  $T$ , яка ще не була узгоджена. Після того, як точка  $C_i$  зіставлена з найближчою точкою  $T_j$ , повторюємо попередній крок доти, доки кожна точка з хмари  $C$  не матиме зіставлену пару з хмари  $T$ . Асимптотична складність такого алгоритму зіставлення дорівнює  $O(n^2)$ , якщо описані хмари мають однакову кількість точок  $n$ . Це очевидно, так як для кожної точки з хмари  $C$  потрібно виконати лінійний пошук найближчої точки для кожної неузгодженої точки з хмари  $T$ .

Проте після реалізації цього алгоритму ми помітили, що значення помилки зіставлення для одних і тих самих хмар точок відрізняються в залежності від порядку, в якому точки з хмари  $C$  шукають собі найближчу пару з другої хмари  $T$ . Це зумовлено тим, що перша точка з хмари  $C$ , що шукає собі найближчу пару з хмари  $T$ , має можливість обрати будь-яку пару, тому що жодна точка з хмари  $T$  ще не є узгодженою. В такому випадку остання точка з хмари  $C$  не матиме іншого вибору, аніж обрати єдину, що залишилася, неузгоджену точку з хмари  $T$ . Тоді пошук відповідності між двома хмарами точок, яка матиме найменшу можливу помилку зіставлення, потребує додаткових  $n$  повторів виконання зіставлення хмар, щоб кожна точка з хмари  $C$  мала можливість першою обрати собі найближчу пару. Асимптотична складність такого пошуку зіставлення з найменшою помилкою зростає до  $O(n^3)$ ,

Введемо параметр  $\varepsilon$ , який буде контролювати кількість повторних зіставлень двох хмар. Якщо  $\varepsilon = 0$ , то ми виконуємо зіставлення хмар лише один раз. Якщо  $\varepsilon = 1$ , то алгоритм зіставлення виконується  $n$  разів, даючи кожній точці хмари  $C$  можливість першою підібрати собі пару з хмари  $T$ ). Якщо  $0 < \varepsilon < 1$ , то алгоритм зіставлення виконується  $n^\varepsilon < n$  разів.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		43

Використовуючи цей формалізм, асимптотична складність нашої спроби знайти найкраще співставлення виражається як  $O(n^{2+\varepsilon})$ , де  $\varepsilon \in [0; 1]$ .

Для отримання більш точної оцінки помилки зіставлення введемо коефіцієнти, які зважать значення відстані між точками кожної узгодженої пари. Тоді значення помилки зіставлення двох хмар рахується як:

$$\sum_{i=1} w_i * \|C_i - T_j\|,$$

де  $w_i$  – коефіцієнт, що зважує відстань між точкою  $C_i$  та її парою  $T_j$ .

Коефіцієнт  $w_i$  відображає довіру до кожної пари  $(C_i, T_j)$  співставлення, обчислену під час жадібного пошуку пари. Перша утворена пара співставлення має коефіцієнт довіри  $w_i = 1.0$ , що означає, що ми впевнені в тому, що точка  $C_i$  знайшла найближчу пару з хмари  $T$ , бо ще немає узгоджених точок хмари  $T$  і перша точка має можливість обрати будь-яку з них. По мірі того, як алгоритм прогресує, залишається все менше та менше неузгоджених точок з хмари  $T$  для решти точок з хмари  $C$  для пошуку їх найближчої пари. Тому ми не можемо повною мірою довіряти цим зіставленням і відповідно зважуємо їх меншими коефіцієнтами, що лежать в межах  $(0..1]$ . Наприклад, остання точка, що підлягає співставленню, немає іншого вибору, аніж обрати точку, що залишилася. Тому впевненість в цьому співставленні також повинна бути мала.

Ми використали схему лінійного зважування, в якій коефіцієнти довіри отримують наступні значення:

$$w_i = 1 - \frac{i - 1}{n}$$

де  $i = 1..n$ .

Таким чином, наша система реалізує метод зіставлення з шаблоном у якості класифікаційного методу. Для того, щоб визначити категорію символу чи образу, до якої відносяться вхідні дані, необхідно зіставити їх з

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		44

кожним присутнім у системі шаблоном. Категорія шаблону, що матиме найменшу помилку зіставлення з вхідними даними, і є результатом нашого класифікаційного методу, а отже і усієї роботи системи.

Алгоритм реалізованого класифікаційного методу наведені далі у цій роботі у підрозділі опису алгоритмів.

### 3.2 Опис алгоритмів

У даному підрозділі ми наводимо у довільній формі реалізовані алгоритми.

Першим розглянутим алгоритмом є основним алгоритмом роботи реалізованої системи, що описує реалізований процес розпізнавання. На вхід алгоритму подаються хмара точок  $C$  та масив шаблонів  $T$ .

- Основний алгоритм роботи системи:

1. Початок
2. Виконати попередню обробку хмари точок  $C$
3.  $scores \leftarrow \infty$
4. Для кожного відомого шаблону  $T$ :
  - 4.1.  $d \leftarrow$  оцінка відповідності  $C$  до  $T$
  - 4.2. Якщо  $scores > d$ , то
    - 4.2.1.  $scores \leftarrow d$
    - 4.2.2.  $result \leftarrow$  шаблон  $T$
5. Знайдений відповідний шаблон  $result$  з оцінкою відповідності  $scores$
6. Кінець

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		45

- Алгоритм попередньої обробки

1. Початок
2.  $P \leftarrow$  рівномірна вибірка з вхідного сигналу  $C$
3. Масштабувати  $P$
4. Зміщення  $P$
5. Маємо нормалізовану хмару точок  $P$
6. Кінець

- Алгоритм створення рівномірної вибірки:

Алгоритм створення рівномірної вибірки утворює хмару точок, що задовольняє умові рівномірної вибірки. Результатом цього алгоритму є рівномірна вибірка, що містить  $n$  точок.

На вхід алгоритму подається послідовність точок  $C$ , з якої необхідно утворити рівномірну вибірку.

1. Початок
2.  $I \leftarrow PathLength(C) / n$
3.  $D \leftarrow 0$
4.  $result \leftarrow C_0$
5. Для кожної точки  $C_i \in C$ 
  - 5.1. Якщо  $C_i.strokeID = C_{i-1}.strokeID$ , то
    - 5.1.1.  $d \leftarrow$  Евклідова відстань між точками  $C_i$  та  $C_{i-1}$
    - 5.1.2. Якщо  $D + d \geq I$ , то
      - 5.1.2.1.  $q.x \leftarrow C_{i-1}.x + \frac{I-D}{d} \cdot (C_i.x - C_{i-1}.x)$
      - 5.1.2.2.  $q.y \leftarrow C_{i-1}.y + \frac{I-D}{d} \cdot (C_i.y - C_{i-1}.y)$
      - 5.1.2.3.  $q.strokeID \leftarrow C_{i-1}.strokeID$

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		46



5.1.2.4.  $Append(result, q)$

5.1.2.5.  $Insert(C, i, q)$

5.1.2.6.  $D \leftarrow 0$

5.1.3. **Інакше**

5.1.3.1.  $D \leftarrow D + d$

6. Маємо рівномірну вибірку  $result$

7. Кінець

- Алгоритм масштабування та зміщення хмари точок

1. Початок

2. Знайти мінімальні значення по осі  $x$  та по осі  $y$

3. Знайти максимальні значення по осі  $x$  та по осі  $y$

4.  $scaleX \leftarrow x_{max} - x_{min}$

5.  $scaleY \leftarrow y_{max} - y_{min}$

6. Для кожної точки  $C_i$  з хмари  $C$

$$6.1. C_i.x \leftarrow \frac{C_i.x - x_{min}}{scaleX}$$

$$6.2. C_i.y \leftarrow \frac{C_i.y - y_{min}}{scaleY}$$

7. Кінець

- Алгоритм оцінки відповідності хмари точок до шаблону:

Алгоритм оцінки відповідності зіставляє дві хмари точок (вхідних даних та шаблону), повторно виконуючи порівняння між їхніми точками та кожного разу починаючи з нової стартової точки. Результатом цього алгоритму є значення помилки відповідності.

На вхід алгоритму подаються хмара точок  $C$  та шаблон  $T$ .

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		47

1. Початок
2.  $step \leftarrow \lfloor n^{1-\varepsilon} \rfloor$
3.  $min \leftarrow \infty$
4. Для кожного  $i \in [0..n]$  з кроком  $step$ :
  - 4.1.  $d_1 \leftarrow$  відповідність  $C$  до  $T$  починаючи з точки  $i$
  - 4.2.  $d_2 \leftarrow$  відповідність  $T$  до  $C$  починаючи з точки  $i$
  - 4.3.  $min \leftarrow \text{minimum}(min, d_1, d_2)$
5. Знайдене значення оцінки відповідності хмар  $C$  та  $T$
6. Кінець

- Алгоритм оцінки відповідності двох хмар точок з початковим індексом:

Алгоритм оцінки відповідності зіставляє дві хмари точок починаючи з вказаної стартової позиції та обчислює мінімальну зважену помилку між точками першої та другої хмар. Результатом цього алгоритму є значення оцінки відповідності першої хмари точок до другої.

На вхід алгоритму подаються перша хмара точок  $C$ , друга хмара точок  $T$  та індекс стартової точки  $start$ .

1. Початок
2.  $i \leftarrow start$
3.  $sum \leftarrow 0$
4.  $min \leftarrow \infty$
5. Для кожної неузгодженої точки  $T_j \in T$ :
  - 5.1.  $d \leftarrow$  Евклідова відстань між точками  $C_i$  та  $T_j$
  - 5.2. Якщо  $d < min$ , то
    - 5.2.1.  $min \leftarrow d$
    - 5.2.2. Позначити точку  $T_j$  як узгоджена

$$5.3.w \leftarrow 1 - \frac{i-start}{n}$$

$$5.4.sum \leftarrow sum + w * min$$

$$5.5.i \leftarrow (i + 1) MOD n$$

6. **Поки**  $i \neq start$  повернутися до кроку 4
7. Знайдена значення помилки зіставлення  $C$  до  $T$
8. Кінець

### 3.3 Розробка програми та інструкція користувачеві

У ході цієї роботи був розроблений застосунок для мобільної платформи Android, що реалізовує інтерфейс для взаємодії з системою розпізнавання рукописних символів.

Метою розробки цього застосунку було створення графічного інтерфейсу для взаємодії з системою розпізнавання рукописних символів чи образів, який дозволяє продемонструвати можливості реалізованої системи та зібрати експериментальні та статистичні дані для подальшого вивчення, аналізу проблеми розпізнавання та покращення існуючих методів.

Вибір такої мобільної платформи зумовлений її великою популярністю. Мобільні пристрої, такі як смартфони та планшети, що використовують операційну систему Android, є одними із найпоширеніших у світі. Така розповсюдженість платформи дозволяє легко знайти суб'єкт для тестування нашої системи. До того ж, випускати розроблений застосунок на таку платформу легко і зручно. Тому для розробки графічного інтерфейсу для взаємодії з нашою системою була обрана мобільна платформа Android.

Головною точкою входу програми є діяльність MainActivity. Діяльності мобільної платформи Android відображають можливі екрани

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		49

застосунку. Так як наш застосунок – це технологічна демонстрація нашої системи для розпізнавання рукописної літери, для якої необхідна лише можливість зобразити бажаний символ, то він (застосунок) містить тільки один екран, на якому можна писати та виконувати певні маніпуляції над створеним зображенням. Тому наш застосунок складається лише з однієї головної діяльності.

Наша активність `MainActivity` містить полотно для написання рукописного символу, текстовий рядок, в якому буде відображатися текстова інтерпретація рукописного символу чи образу, та деяка кількість кнопок, необхідних для редагування рукопису чи контролю процесу розпізнавання.

Основний фрагмент інтерфейсу - це спеціальний об'єкт `PaintView`, що реалізовує інтерфейс класу `View`, який полегшує сканування та дискретизацію вхідного сигналу, утвореного завдяки взаємодії спеціального пишучого пера (яким може виступати, наприклад, звичайний палець) з електронною сенсорною поверхнею. Саме цей об'єкт дозволяє програмному застосунку правильно отримувати сигнал рукописного вводу. `PaintView` записує шлях, що складається з штрихів, які робить користувач, та зберігає у себе, очікуючи сигнал для подачі цих даних до системи розпізнавання.

Додаток містить кнопки «Clear» (з англ. «Очистити») та «Recognize» (з англ. «Розпізнай»).

Кнопка «Clear» очищає полотно, створене об'єктом `PaintView`, та будь-які зображені на ньому штрихи, образ чи малюнок, щоб надати простір для написання нового символу.

Кнопка «Recognize» посилає сигнал до головної діяльності `MainActivity`, що сигналізує про те, що введення символу закінчено, необхідно почати розпізнавання. Отримавши такий сигнал, діяльність передає отриману з об'єкту `PaintView` послідовність точок до об'єкту класу

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		50

Recognizer, який в свою чергу виконує усі зазначені у цій роботі етапи процесу розпізнавання. Як тільки об'єкту класу Recognizer закінчить свою роботу, головний потік діяльності отримає сигнал, що містить текстове представлення зображеного символу та значення помилки знайденої відповідності. Цей результат діяльність виводить у спеціальному текстовому полі, згаданому вище.

Також програмний застосунок містить кнопку «Undo» (з англ. «Відмінити»), кнопку «Teach me» (з англ. «Навчи мене») та кнопку «Sync» (з англ. «Синхронізувати»).

Кнопка «Undo» стирає останній написаний користувачем штрих, що часто є дуже зручною функцією, коли користувач не хоче витирати усі написані їм штрихи, але вважає, що остання (чи останні) штрихи вийшли невдалими, і бажає їх перемалювати.

Кнопка «Teach me» використовується лише тоді, коли користувача не задовольняє результат, виданий системою розпізнавання, або він має намір додати новий символ до списку відомих системі. В такому випадку користувач має можливість «навчити» систему розпізнавати даний символ чи образ, додавши до неї поточний зображений на об'єкті PaintView рукописний символ як новий шаблон для наступних процесів розпізнавання.

При натисканні на кнопку «Teach me» користувачеві відкривається діалогове вікно, що пропонує йому назвати створений шаблон символу. Після підтвердження назви, застосунок додає до свого списку шаблонів новий.

Кнопка «Sync» використовується для того, щоб синхронізувати свій список відомих системі шаблонів зі списком, що знаходиться на сервері. При натисканні на кнопку, програма автоматично відправить серверу запит на отримання списку існуючих шаблонів та синхронізує свій список з отриманим як тільки від сервера прийде відповідь.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		51

### 3.4 Тестування

У нашому дослідженні ми використовували один і той же натренований список шаблонів для тесту різних значень параметру  $\varepsilon$ , що контролює кількість повторних зіставлень, різної кількості представлених шаблонів  $T$  для кожного символу, та різної кількості точок  $n$  для рівномірної вибірки. В результаті чого маємо велику кількість різних реалізацій системи.

На початку тестуванні ми провели окремий дослід для визначення найоптимальнішого коефіцієнту  $\varepsilon$  при максимальній в наших тестах кількості точок для повторної рівномірної вибірки ( $n = 96$ ). Значення коефіцієнту обиралися з наступної множини  $\varepsilon \in \{0, 0,25, 0,50, 0,75, 1\}$ .

Точність розпізнання вхідних даних системою при значеннях параметру  $\varepsilon \geq 0,5$  досягала найвищих серед усіх значення, як і очікувалося, проте усі вони відрізнялися між собою несуттєво. Тому для подальших досліджень ми використовували значення параметра  $\varepsilon$  як 0,5.

Ми протестували розроблену нами системи на вплив кількості навчальних зразків для кожного символу на відсоток точності розпізнавання. Наша система досягає 90,5% точності для  $T \geq 3$  навчальних зразків на символ та тримається більше 93% при  $T \geq 5$ .

Також ми були зацікавлені в тому, скільки учасників мають надати зразки шаблонів для системи розпізнавання, щоб забезпечити високу точність роботи для нових користувачів. Тому ми попросили 15 осіб різного віку та статі прийняти участь у нашому дослідженні.

Наша система видала 84% точності розпізнавання з шаблонами, наданими лише одним учасником. Точність розпізнавання системи для нових користувачів піднялася до 92% з зразками шаблонів, наданими 4 учасниками тестування. Надавши системі шаблони 10 учасників тестування, точність її розпізнавання складала приблизно 95%.

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		52

Час класифікації кандидата займав 32 мс, але це результати при  $n = 96$ . Так ми вирішили дослідити ефективність роботи за різними значеннями  $n$ . Ми протестували нашу систему розпізнавання для наступних значень  $n \in \{8, 16, 32, 64, 96\}$ .

При  $n = 8$  ми отримали найменше значення точності розпізнавання (72%), а точність вище 80% почали отримувати при значення  $n \geq 32$ . Вимірювання часу роботи та точність роботи системи при  $n \geq 32$  показало, що при збільшенні  $n$  кількість операцій системи під час роботи стрімко зростає, проте і точність стає помітно вищою. Хоча при значенні  $n = 96$  наша система показувала найкращі результати точності розпізнавання (96%), проте зменшення значення  $n$  до 32 зменшило час її роботи приблизно у 8 разів.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		53

### ВИСНОВКИ ДО РОЗДІЛУ 3

На вхід системи on-line розпізнавання подається цифровий сигнал від електронної поверхні чи спеціального електронного пера. Тому цільова платформа нашої системи – це мобільні пристрої, що мають сенсорний екран. Проте теоретично основа нашої системи може бути використана і на інших платформах, які можуть отримати сигнал з електронної поверхні чи спеціального пера.

Наш процес розпізнавання складається з етапу попередньої обробки (що включає в себе крок відбору рівномірної вибірки, масштабування та зміщення) та етапу класифікації. Проте наша система не використовує жодний з заявлених методів вилучення особливостей і пропускає цей крок у зв'язку з специфікою класифікаційного методу.

Наш метод класифікації можна віднести до категорії методів співставлення з шаблонами. Тобто для того, щоб додати новий символ до системи, необхідно сформулювати для нього шаблон, з яким система буде порівнювати у подальшій своїй роботі.

Порівняння з існуючими шаблонами нашої системи виконується алгоритмом, в основі якого лежить алгоритм жадібного співставлення точок. Асимптотична складність нашого алгоритму розпізнавання рукописної літери складає  $O(T * n^{2+\epsilon})$ , де  $T$  – кількість тренуваних зразків (шаблонів) системи,  $n$  – кількість точок у рівномірній вибірці,  $\epsilon$  – параметр, що контролює кількість повторних зіставлень.

У ході нашої роботи був розроблений застосунок для мобільної платформи Android, метою якого було створення графічного інтерфейсу для взаємодії з системою розпізнавання рукописних символів чи образів, який дозволяє продемонструвати можливості реалізованої системи та

					<b>ІАЛЦ.467200.003 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		54



зібрати експериментальні та статистичні дані для подальшого вивчення, аналізу проблеми розпізнавання та покращення існуючих методів.

Під час тестування розробленої системи експериментальним шляхом було виявлені найоптимальніші значення параметрів нашої системи для роботи на мобільних пристроях, а саме використовувати параметр, що контролює кількість повторних зіставлень,  $\varepsilon = 0,5$  та кількість точок у рівномірній вибірці  $n = 32$ .

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		55

## ВИСНОВКИ

У ході даної роботи були проведені глибокі дослідження в області проблеми розпізнавання рукописних символів. Була опрацьована велика кількість літературних та фондових матеріалів.

Метою нашого цього дослідження було ознайомлення з існуючими методами обробки рукописних символів, інтерпретації письма у друкований текст, зрозумілий обчислювальним технологіям та, використовуючи отримані знання, реалізувати розпізнавальну систему для мобільних пристроїв.

Процес онлайн-розпізнавання рукописного тексту можна розділити на кілька загальних кроків: попередня обробка (*preprocessing*), вилучення характеристик (*feature extraction*) та класифікація (*classification*).

Наша система використовує метод жадібного співставлення точок в якості класифікатора. Ми змогли досягнути максимальної точності розпізнавання до 96% при великому обсязі навчальних зразків та достатній кількості точок рівномірної вибірки, що створюється на етапі попередньої обробки вхідного сигналу. Однак, для оптимізації ефективності обраного алгоритму розпізнавання ми пожертвували відносно невеликим відсотком точності, проте прискорили роботу програми.

Система була навчена на розпізнавання англійської абетки як великих, так і малих літер, тобто 62 рукописних символів. У складі системи входить набір з 210 навчальних зразків, що в середньому розподіляється по 3 зразки на кожний символ, що вважається невеликим набором.

Враховуючи усі вище згадані параметри, тестування кінцевої системи розпізнавання досягло 92% точності.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		56

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shankar Rao; J. Aditya, Handwriting Recognition - "Offline" Approach. Modern Trends in Intelligent Computing Systems (Nasmotics), 2005.
2. Schantz, Herbert F. (1982). The history of OCR, optical character recognition. [Manchester Center, Vt.]: Recognition Technologies Users Association. ISBN 9780943072012.
3. Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4\_9.
4. Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2
5. Berger, James O. (1985). Statistical decision theory and Bayesian Analysis (2nd ed.). New York: Springer-Verlag. ISBN 978-0-387-96098-2. MR 0804611.
6. Daniel Keysers, Thomas Deselaers, Henry A. Rowley, Li-Lun Wang, Victor Carbune (2017). Multi-Language Online Handwriting Recognition. Published in IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI:10.1109/TPAMI.2016.2572693
7. Huang, B.; Zhang, Y. and Kechadi, M.; Preprocessing Techniques for Online Handwriting Recognition. Intelligent Text Categorization and Clustering, Springer Berlin Heidelberg, 2009, Vol. 164, "Studies in Computational Intelligence" pp. 25–45.
8. Holzinger, A.; Stocker, C.; Peischl, B. and Simonic, K.-M.; On Using Entropy for Enhancing Handwriting Preprocessing, Entropy 2012, 14, pp. 2324-2350.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		57

9. Blumenstein, Michael and Brijesh Verma. “A new segmentation algorithm for handwritten word recognition.” IJCNN (1999).

10. Hong, P., Turk, M. and Huang, T.S. (2000) Constructing finite state machines for fast gesture recognition. Proc. ICPR '00. Los Alamitos, CA: IEEE Press, 691-694.

11. Anderson, D., Bailey, C. and Skubic, M. (2004) Hidden Markov Model symbol recognition for sketch-based interfaces. AAAI Fall Symposium. Menlo Park, CA: AAAI Press, 15-21.

12. Pittman, J.A. (1991) Recognizing handwritten text. Proc. CHI '91. New York: ACM Press, 271-275.

13. Cho, M.G. (2006) A new gesture recognition algorithm and segmentation method of Korean scripts for gesture-allowed ink editor. Information Sciences 176 (9), 1290-1303.

14. Kristensson, P.-O. and Zhai, S. (2004) SHARK2 : A large vocabulary shorthand writing system for pen-based computers. Proc. UIST '04. New York: ACM Press, 43-52.

15. Calhoun, C., Stahovich, T.F., Kurtoglu, T. and Kara, L.B. (2002) Recognizing multi-stroke symbols. AAAI Spring Symposium. Menlo Park, CA: AAAI Press, 15-23.

16. Guerfali, W.; Plamondon, R. Normalizing and restoring on-line handwriting. Pattern Recogn. 1993, 26, 419–431.

17. Shwetha, Dimple & Lokesh, Ramya. (2014). Comparison of Smoothing Techniques and Recognition Methods for Online Kannada Character Recognition System. 2014 International Conference on Advances in Engineering and Technology Research, ICAETR 2014. 10.1109/ICAETR.2014.7012888.

18. Kalman R.E. A new approach to linear filtering and prediction problems // Journal of Basic Engineering. — 1960. — Т. 82, вип. 1. — С. 35–45. — DOI:10.1115/1.3662552.

19. Vatavu, R.-D., Anthony, L. and Wobbrock, J.O. (2012). Gestures as point clouds: A SP recognizer for user interface prototypes. Proceedings of the ACM International Conference on Multimodal Interfaces (ICMI '12). Santa Monica, California (October 22-26, 2012). New York: ACM Press, pp. 273-280.

20. Stanek, Steven & Packard, Woodley. (2008). Greedy Point Match Handwriting Recognition.

21. Avis, D. A survey of heuristics for the weighted matching problem. Networks 13 (1983), 475–493.

					<i>ІАЛЦ.467200.003 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		59