

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**ФАКУЛЬТЕТ ЕЛЕКТРОНИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# **КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

**на тему:**

**«Інтелектуальна технологія розпізнавання рукописних  
СИМВОЛІВ»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Шелехов І.В.**

**Студента групи ІН.м-92**

**Хекало В. С.**

**СУМИ 2020**

Сумський державний університет  
(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук  
Спеціальність 122 «Комп'ютерні науки»

Затверджую:  
зав.кафедрою \_\_\_\_\_  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ**

Хеґало Владиславу Сергійовичу  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інтелектуальна технологія розпізнавання рукописних символів

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи) \_\_\_\_\_

3. Вхідні данні до проекту (роботи) \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналіз проблеми та постановка задачі 2) Інформаційно-екстремальна інтелектуальна технологія 3) Розробка інформаційного і програмного забезпечення інтелектуальної системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	<i>Аналіз проблеми та постановка задачі</i>		
2.	<i>Інформаційно-екстремальна інтелектуальна технологія</i>		
3.	<i>Розробка інформаційного і програмного забезпечення інтелектуальної системи</i>		
4.	<i>Оформлення пояснювальної записки до дипломної роботи</i>		

Студент – дипломник \_\_\_\_\_ (підпис)

Керівник проекту \_\_\_\_\_ (підпис)

## РЕФЕРАТ

**Записка:** 43 стор., 16 рис., 1 додаток, 11 джерел.

**Об'єкт дослідження** – слабоформалізований процес розпізнавання рукописних символів.

**Мета роботи** — розробка алгоритму ієрархічного розпізнавання рукописних символів в рамках інформаційно-екстремальної інтелектуальної технології (ІЕІТ).

**Методи дослідження** — метод функціонально-статистичних випробувань, методи теорії інформації і теорії розпізнавання образів.

**Результати** — розроблено інформаційне та програмне забезпечення системи ієрархічного розпізнавання рукописних символів. При цьому розглянуто питання обчислення інформаційного критерію функціональної ефективності (КФЕ) таких систем в рамках інформаційно-екстремальної інтелектуальної технології. Розроблений алгоритм реалізовано у програмному забезпеченні створеному за допомогою інструментального програмного середовища MATLAB

СИСТЕМА РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ,  
ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, НАВЧАЛЬНА МАТРИЦЯ,  
МЕТОД ФУНКЦІОНАЛЬНО-СТАТИСТИЧНИХ  
ВИПРОБУВАНЬ, ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА  
ІНТЕЛЕКТУАЛЬНИЙ ТЕХНОЛОГІЯ.

## ЗМІСТ

ВСТУП .....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
1.1 ОСНОВНІ ЕТАПИ І ПРИНЦИПИ РОЗПІЗНАВАННЯ СИМВОЛІВ.....	8
1.2 АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ РОЗПІЗНАВАННЯ СИМВОЛІВ.....	8
1.3 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ .....	17
2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ.....	18
2.1 ОСНОВНІ ПОЛОЖЕННЯ І ВИЗНАЧЕННЯ ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНОЇ ІНТЕЛЕКТУАЛЬНОЇ ТЕХНОЛОГІЇ.....	18
2.2 МАТЕМАТИЧНІ МОДЕЛІ СИСТЕМ РОЗПІЗНАВАННЯ СИМВОЛІВ .....	21
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	25
3.1 ФОРМУВАННЯ ВХІДНОГО МАТЕМАТИЧНОГО ОПИСУ .....	25
3.2 АЛГОРИТМ ОЦІНКИ ЕФЕКТИВНОСТІ ІЄРАРХІЧНОГО КЛАСИФІКАТОРА .....	27
3.3 КОРОТКИЙ ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .....	29
3.4 АНАЛІЗ РЕЗУЛЬТАТІВ.....	35
ВИСНОВКИ.....	42
СПИСОК ЛІТЕРАТУРИ.....	43
ДОДАТОК.....	45

## ВСТУП

Цифрова обробка зображень є актуальним розділом досліджень в галузі теоретичної інформатики, як в сфері фундаментальних наук, так і в сфері практичного застосування. Відомими прикладами виникнення завдання обробки зображень в сфері фундаментальної науки є астрономія, фізика елементарних частинок, фотографія. Практичне застосування даного завдання пов'язано з радіолокацією, медициною, географією, криміналістикою, широким спектром завдань обліку, обробки та систематизації даних. Можна з упевненістю констатувати, що в даний час цифрова обробка зображень проникла в усі сфери людської діяльності. У зв'язку з цим спільне завдання обробки зображень розпадається на досить широкий клас підзадач. Однією з таких підзадач є задача розпізнавання машинописного і рукописного тексту. За останні роки досягнуто суттєвого прогресу в розпізнаванні стандартизованого (машинописного) тексту. Розроблено алгоритми і ряд програмних продуктів, що дозволяють з високим ступенем точності розпізнавати друкований текст. Інша працювати з алгоритмами і програмними комплексами для розпізнавання рукописного тексту. Так, існуючі програмні продукти призначені, в основному, для введення спеціальних форм або анкет, заповнених від руки і вирішення інших вузько спеціалізованих завдань.

У той же час рукописний текст є найбільш природним для людини способом збереження і подальшого використання інформації. З використанням сучасних засобів обчислювальної техніки, глобальних мереж і засобів обміну повідомленнями роль рукописного тексту в повсякденному комунікації зростає значно. Найпростішими прикладами є поштові адреси на конвертах, шкільні завдання, медичні довідки та висновки, заповнені від руки анкети і форми, підписи на документах і банківських чеках, людське спілкування за допомогою листа. Велике значення проблема розпізнавання рукописного тексту має в архівній справі. В даний час досить складно

оцінити загальну кількість існуючих рукописних архівних документів, які вже зараз вимагають перекладу в цифровий формат і розпізнавання.

Однак переклад рукописних документів в електронну форму шляхом прямого набору тексту оператором з паперового носія складний і має на увазі використання значних людських ресурсів та спеціалізованих програмних оболонок. З цієї причини існує реальна потреба в створенні автоматизованих систем розпізнавання рукописних документів, які не потребують великих трудовитрат з боку оператора.

Весь спектр перерахованих проблем, а також виникають нові технологічні завдання визначають актуальність розробки алгоритмів і методів цифрової обробки рукописних текстів, що і є предметом дослідження цієї роботи. Система розпізнавання рукописних символів за траєкторією, за якою рухається перо при написанні символу, була створена на базі інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології).

# 1 ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Основні етапи і принципи розпізнавання символів

Задача автоматичного аналізу та класифікації документів є досить актуальною для сучасної інформаційної системи керування. У відповідності з [2] для розв'язання задач подібного класу існує узагальнена регламентована методологія, котра умовно розбиває весь процес класифікації на такі функціональні блоки:

- аналіз документа з наступним виділенням окремих полів зображення;
- низькорівнева обробка зображення з метою відокремлення шумів, вирівнювання рядків, корекція нахилу, тощо;
- сегментація рядків тексту на слова та символи;
- розпізнавання символів;
- розпізнавання слів та речень.

У загальному випадку комплексне розв'язання задачі автоматичної класифікації документів є досить складною через комплексність та масовість, але за при введенні обмежень (конкретизації класу задачі) воно значимо спрощується. В даній роботі є такі спрощення. По-перше, розроблявся підхід до розпізнавання лише символів, по-друге, множину класів розпізнавання було звужено лише до друкованих латинських літер верхнього регістру – всього 26 елементів.

## 1.2 Аналітичний огляд методів розпізнавання символів

Системи розпізнавання реалізуються як класифікатори, які використовують різні методи: шаблонні (растрові); прізнаковие; структурні [2]. У класифікаторі шаблонного типу за допомогою критерію порівняння визначається, який з шаблонів вибрати з бази. Найпростіший критерій - мінімум точок, що відрізняють шаблон від досліджуваного зображення. До переваг шаблонного класифікатора відносяться гарне розпізнавання дефектних символів ( «розірваних» або «склесених»), простота і висока

швидкість розпізнавання. Недоліком є необхідність настройки системи на типи і розміри шрифтів.

У прізнакових класифікаторах аналіз проводиться тільки по набору чисел або ознак, що обчислюються по зображенню. Цей метод дозволяє розпізнавати різні зображення символів, тобто різні підчерки шрифти і т.д. Цей метод неминуче викликає деяку втрату інформації, так як використовується топологічний уявлення, що відбиває інформацію про взаємне розташування структурних елементів символу. Ці дані можуть бути представлені в графовой формі. При цьому даний метод забезпечує інваріантність щодо типів і розмірів шрифтів. Недоліками є складність розпізнавання дефектних символів і повільна робота.

Основою структурно-плямовий методу є структурно-плямовий еталон [12]. Він має вигляд набору плям з попарними відносинами між ними. Зазначене подання невідчутно до різних накресленням і дефектів символів. Алгоритм заснований на поєднанні шаблонного і структурного методів розпізнавання образів. При аналізі зразка виділяються ключові точки об'єкта - так звані «плями». Як плям, наприклад, можуть виступати: кінці ліній; вузли, де сходяться кілька ліній; місця зламів ліній; місця перетину ліній; крайні точки. Після виділення характерних точок визначаються зв'язку між ними - відрізок, або дуга. Таким чином, підсумкове опис є граф, який і служить об'єктом пошуку в бібліотеці «структурно-плямових еталонів». При пошуку встановлюється відповідність між ключовими точками зразка і еталона, після чого визначається ступінь деформації зв'язків, необхідна, щоб привести шуканий об'єкт до порівнює еталонному зразку. При цьому, менший ступінь необхідної деформації передбачає велику ймовірність правильного розпізнавання символу. Далі розглянемо етапи обробки зображення.

1. Передобробка. На цьому етапі виконуються наступні завдання: підвищення якості зображення за рахунок фільтрації, шумозаглушення і інших, що мають на меті підвищити якість зображення. На цьому етапі

відбувається очищення зображення від дефектів сканування. Зокрема, на самому початку роботи до зображення з метою шумозаглушення часто застосовується фільтр Гаусса. Важливу роль відіграє порогова бінаризація, тобто переклад зображення в чорно-білий формат з кольорового або відтінків сірого [3]. Це дозволяє різко розділити текст і фон, спрощує в подальшому застосування багатьох алгоритмів, а також позбавляє від деяких шумів на зображенні. При цьому використовується гістограма яскравості зображення тексту, на якому спостерігається два піки: високий пік, відповідний білому тлу, тобто кольором паперу, і пік в області темних пікселів, відповідних яскравості символів тексту.

2. Виділення регіону інтересу. На цьому етапі на бінаризованими зображенні виділяється безпосередньо область, на якій знаходиться розпізнається текст, і відкидаються елементи, текстом не є [3,4]. До них відносяться такі об'єкти, як ляпи, плями на папері, не видалені в процесі бінаризації, картинки і ін. Для їх видалення можна, наприклад, виділяти компоненти зв'язності на зображенні, обчислювати геометричні ознаки і на їх основі класифікувати компоненту зв'язності як частина тексту або дефект, використовуючи методи машинного навчання або евристики.

3. Сегментація і нормалізація тексту. На цьому етапі текст розділяється, або сегментується, на зручні для аналізу складові частини [5]. Найбільш природними процесами на даному етапі є поділ тексту на окремі рядки (сегментація рядків) і поділ рядків на слова (сегментація слів), а також, теоретично, поділ слів на елементарні складові частини. Крім того, на даному етапі проводиться нормалізація тексту приведення виділених складових частин до деякого стандартного виду для зниження варіативності і можна було легко розпізнати.

Сегментація рядків. Завдання сегментації (поділу) рядків в машинопечатних документах на сьогоднішній день вважається повністю вирішеною. Але в задачах при поділі рядків в загальному випадку виникають

складності, що не дозволяють безпосередньо застосовувати алгоритми, придатні для машинопечатних текстів:

- рядки не тільки можуть не бути паралельними, а й можуть згинатися;
- різні рядки можуть бути занадто близькі, а елементи тексту, що належить різним рядкам, можуть накладатися один на одного.

Наприклад, якщо коефіцієнт форми області (відносина квадрата її периметра до площі) менше деякого значення, а площа більше деякого значення, то це з великою ймовірністю дефект (тому що рукописний текст зазвичай є деякою кривою).

Виділення базових ліній. Ці методи засновані на ідеї, що людина пише або по, або поверх деякої уявної лінії. Дані методи намагаються апроксимувати цю лінію, а потім відновити по ній рядок. У перетворенні Хафа виділяються прямі, якщо вони не дуже викривлені. Перетворення Хафа застосовується до центрів компонент зв'язності пікселів тексту. Такий підхід вимагає, щоб рядки тексту були близькі до прямих, але зате дозволяє виділяти рядки, розташовані в довільному місці і йдуть під довільними кутами.

Перетин елементів різних рядків являє собою проблему не тільки сегментації рядків, але і розпізнавання тексту, так як віднесення елемента до неправильної рядку очевидно погіршує його розпізнаваність. Пересічні компоненти є проблемою для методів горизонтальної проекції (так як вони збільшують значення профілю проекції в тих місцях, де повинен бути його мінімум) групіровочних методів (так як вони використовують зв'язкові компоненти пікселів тексту для побудови рядків), але слабо впливають на деякі методи виділення базових ліній. Для пошуку пересічних елементів з різних рядків можна використовувати такі ознаки, як розмір компонент зв'язності тексту, факт віднесення однієї компоненти до кількох рядках або, навпаки, не відносяться ні до якого рядка. Після знаходження таких сумнівних компонент потрібно визначити, чи належать вони до якоїсь рядку або ж їх потрібно декомпозувати на елементи, що відносяться до різних

рядках. Така вертикальна декомпозиція компонент складне завдання. Просте рішення полягає в розрізуванні компоненти на частини горизонтальними лініями, але можна застосувати і більш тонкі підходи, наприклад, виділення окремих штрихів.

3. Сегментація слів. На цьому етапі роботи системи розпізнавання виділені рядки тексту поділяються на окремі слова. На відміну від машинописного тексту, в якому відстань між словами більш-менш постійно, а інтервали між символами усередині слова набагато менше, ніж інтервали між словами, в рукописному тексті розмір інтервалів між словами може варіюватися в дуже широких межах. Компоненти зв'язності тексту, віднесені до одного рядка на попередньому етапі роботи системи розпізнавання, об'єднуються в слова на цьому етапі.

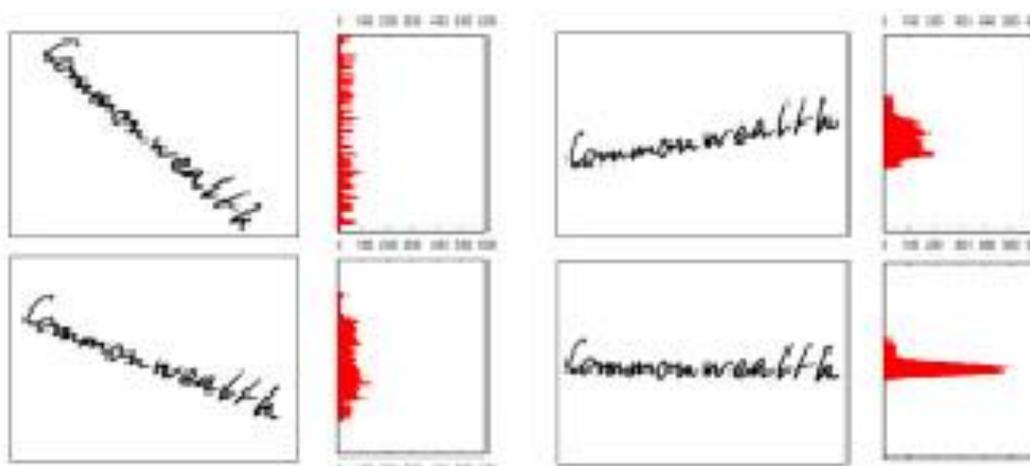


Рисунок 1.1 – Корекція рядків по горизонталі з використанням гістограми профілю

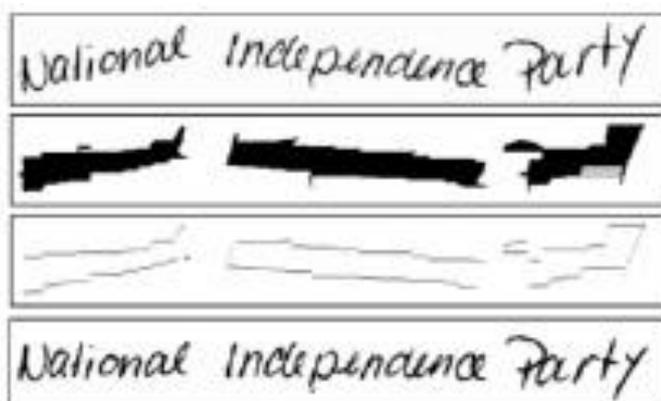


Рисунок 1.2 – Корекція рядків по горизонталі з використанням алгоритму лінійної регресії

4. Нормалізація. В силу високої варіативності накреслення слів їх розпізнавання є дуже складним процесом [12]. Нормалізація служить для приведення слова до деякого стандартного виду без значної втрати інформації, необхідної для розпізнавання. Одними з найбільш часто використовуваних методів нормалізації є метод корекції нахилу слова від горизонтальної та вертикальної лінії [12]. Найпростіший метод корекції по горизонталі полягає у виконанні максимізації його на деякому діапазоні (рис. 1.1). Існують і інші методи нормалізації, наприклад, корекція розміру і виділення скелета тексту, але вони застосовуються рідше.

Існують і інші методи, наприклад, засновані на згладжування і лінійної регресії (рис. 1.2).

#### Структурні методи

Алгоритм роботи цих методів полягає в розбитті вихідного образу на складові частини, що описуються як стабільні ідеальні елементи.

Одні з методів даного напрямку використовує для розпізнавання топологічний опис зображень. Іншими словами, еталон містить інформацію про взаємне положення окремих складових частин символу. При цьому стає неважливим розмір розпізнається букви і навіть шрифт, яким вона надрукована. Можливі зображення, що становлять той чи інший клас, можна уявити як результат гомеоморфних перетворень деякого еталонного зображення відповідно до цього класу. Завдання розпізнавання в цьому випадку може бути зведена до встановлення гомеоморфними пред'явленого зображення з одним з еталонних. Її можна виявити за допомогою топологічних інваріантів - таких властивостей зображення, які не змінюються при його гомеоморфних перетвореннях. Інваріантом, що дозволяє дати кількісну опис зображень, є, наприклад, кількість сходяться в точці ліній. Відповідний опис виходить обходом в певному порядку контурів зображення з одночасною фіксацією індексів точок. Встановлення гомеоморфними - власне розпізнавання - зводиться до порівняння описів пред'явленого зображення і еталонних зображень класів. Важливе значення топологічного

опису - його нечутливість до сильних деформацій зображення, що включає всі перетворення подібності, якщо пов'язувати з кожним зображенням деяку характерну точку, з якої починається обхід. Навчання топологічному коду складається у веденні набору еталонних описів. Встановлення гомеоморфними - власне розпізнавання - зводиться до порівняння описів пред'явленого зображення і еталонних зображень класів. Важливе значення топологічного опису - його нечутливість до сильних деформацій зображення, що включає всі перетворення подібності, якщо пов'язувати з кожним зображенням деяку характерну точку, з якої починається обхід. Навчання топологічному коду складається у веденні набору еталонних описів. Встановлення гомеоморфними - власне розпізнавання - зводиться до порівняння описів пред'явленого зображення і еталонних зображень класів. Важливе значення топологічного опису - його нечутливість до сильних деформацій зображення, що включає всі перетворення подібності, якщо пов'язувати з кожним зображенням деяку характерну точку, з якої починається обхід. Навчання топологічному коду складається у веденні набору еталонних описів.

Інші структурні методи реалізують алгоритм подієвого розпізнавання. Подієвий метод спирається на топологічну структуру об'єкта, що складається з ліній і не змінюється при малих деформаціях образу. Лінією називається частина образу, в кожному перетині якого є всього один інтервал. Лінії, вульгаризовані на деякій сітці, визначають події. Подієве уявлення є не тільки формальним набором ознак, але і адекватним топологічним описом. Навчання методу зводиться до складання списку еталонів на досить великій послідовності образів. При розпізнаванні для вихідного растру визначається подієвий уявлення, яким зіставляється еталонний клас.

Основною проблемою структурних методів розпізнавання є ідентифікація знаків, що мають дефекти.

Перевагами методу є здатність розпізнавання спотворених символів і швидкодія при малому алфавіті.

## Шаблонні методи

Алгоритм роботи шаблонних методів спирається на зіставлення вхідного графічного зображення, ідеального шаблоном. Першим етапом роботи шаблонного методу є перетворення відсканованого зображення в растрове. У процесі розпізнавання, перебираються шаблони, і обчислюється відстань від способу до шаблону. Клас, шаблони якого знаходяться на мінімальній відстані від вхідного образу, є результатом розпізнавання.

Дані методи діляться на два класи шрифтозавісіміє і шрифтонезавісіміє. Шрифтонезавісіміє методи використовують заздалегідь певні шаблони, і універсальні для всіх типів шрифтів. Однак при такому підході знижується ймовірність правильного розпізнавання. Шрифтозавісіміє алгоритми розраховані тільки на один тип шрифту, це підвищує якість їх роботи, але вони абсолютно не працездатні при використанні інших шрифтів. Були запропоновані методи для розпізнавання великих обсягів тексту, коли частина символів гарантовано розпізнається шрифтонезавісіміє методами, а потім на підставі розпізнаних символів будуються шаблони для шрифтозавісіміє алгоритмів.

При існуючому достатку друкованої продукції в процесі навчання неможливо охопити всі шрифти і їх модифікації.

До переваг даного алгоритму відносяться: простота реалізації, надійна робота в умовах відсутності перешкод, висока точність розпізнавання дефектних символів, швидкість при малому алфавіті.

До недоліків можна віднести: сильну залежність від шаблонів і складність підбору оптимальних шаблонів, неможливість розпізнати шрифт, що відрізняється від закладеного в систему, повільна робота при великій кількості перешкод, чутливість до обертання, шумів і спотворень.

## Ознакові методи

Дані методи базуються на тому, що зображення ставиться у відповідність  $N$ -мірний вектор ознак. Розпізнавання полягає в порівнянні його з набором еталонних векторів тієї ж розмірності. Ухвалення рішення

про приналежність образу того чи іншого класу, на підставі аналізу обчислених ознак, має цілий ряд строгих математичних рішень в рамках імовірнісного підходу. Тип і кількість ознак в чималому ступені визначають якість розпізнавання. Формування вектора відбувається під час аналізу зображення. Дану процедуру називають витяганням ознак. Еталон для кожного класу отримують шляхом аналогічної обробки символів навчальної вибірки.

До переваг методу можна віднести: простота реалізації, висока узагальнююча здатність, стійкість до зміни форми символів, високу швидкодію.

До недоліків методу відносяться: нестійкість до різних дефектів зображення, втрата інформації про символ на етапі отримання ознак.

#### Нейромеревеві методи

Нейромеревеві методи [8, 9] засновані на застосуванні різних типів штучних нейронних мереж. Ідея цих методів - моделювання роботи мозку людини. На вхід заздалегідь навченої нейронної мережі надходить вектор, який є поданням вхідного образу (пікселі, частотні характеристики, вейвлет). На виході нейрон, відповідний класу розпізнаного символу, видає максимальне значення функції активації. Або ж на вихід надходить безліч ключових характеристик зображення, які потім обробляються іншими системами. Навчання нейронних мереж відбувається на безлічі навчальних прикладів. Причому можливе навчання з учителем (персептрон) або самоорганізація (мережа Кохонена).

Перевагами методу є: здатність до узагальнення, висока швидкість роботи.

Недоліки: чутливість до обертань і спотворення символів, складність підбору навчальної вибірки і алгоритму навчання.

### **1.3 Аналіз проблеми та постановка задачі**

За результатами проведеного аналітичного огляду актуальною є задача розпізнавання рукописних символів. Метою роботи є розробка ефективного алгоритму розпізнавання рукописних символів в рамках інформаційно-екстремальної інтелектуальної технології. Для досягнення поставленої мети необхідно виконати такі завдання:

- 1) Сформувати вхідний математичний опис системи ієрархічного розпізнавання рукописних символів, що зданта навчатися.
- 2) Створити математичну модель навчання ієрархічного розпізнаванню рукописних символів в рамках інформаційно-екстремальної інтелектуальної технології.
- 3) Обрати критерій функціональної ефективності ієрархічної системи розпізнавання.
- 4) Розробити та програмно реалізувати алгоритми навчання ієрархічного розпізнавання рукописних символів.
- 5) Перевірити працездатність ієрархічної системи на задачі розпізнавання 20 рукописних символів.

## 2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

### 2.1 Основні положення і визначення інформаційно-екстремальної інтелектуальної технології

Інформаційно-екстремальна інтелектуальна технологія [9-10] (ІЕІТ) – інформаційно-екстремальний метод аналізу та синтезу здатної навчатися (самонавчатися) СППР, що ґрунтується на прямій оцінці інформаційної спроможності системи і гіпотезі нечіткої компактності реалізацій образу, яка допускає в загальному випадку перетин класів розпізнавання. ІЕІТ є модифікацією методу функціонально-статистичних випробувань (МФСВ), доповнюючи його оптимізаційними процедурами різних рівнів, поєднуючи таким чином, методи багатокритеріальної ентропійної оптимізації з оптимізацією геометричних параметрів класів розпізнавання в єдину методологію. Метод орієнтовано на використання сучасних обчислювальних комплексів і призначено для розв'язання практичних задач контролю та керування слабо формалізованими процесами шляхом автоматичної класифікації їх функціональних станів за умови апріорної невизначеності.

Основна ідея МФСВ полягає в оптимізації структурованих просторово-часових параметрів функціонування СППР шляхом трансформації в процесі навчання відношення схожості на нечіткому розбитті простору ознак розпізнавання на класи у відношення еквівалентності.

Класом розпізнавання (образом)  $X_m^o$  називається множина відбитих властивостей  $m$ -го функціонального стану СППР і відношень між її елементами. Клас розпізнавання – топологічна категорія, яка задається в просторі ознак розпізнавання областю  $X_m^o \subset \Omega_B$  ( $\Omega_B$  – простір ознак розпізнавання).

Детерміновано-статистичний підхід до моделювання систем вимагає завдання систем нормованих (експлуатаційних) і контрольних допусків на ознаки розпізнавання. Нехай  $X_1^o$  – базовий клас, який характеризує максимальну функціональну ефективність системи керування, тобто є найбільш бажаним для особи, що приймає рішення.

$$\text{ц} \quad x_m^{(j)} = \langle x_{m,1}^{(j)}, \dots, x_{m,j}^{(j)}, \dots, x_{m,N}^{(j)} \rangle, \quad j = \overline{1, n_{\min}},$$

де  $x_{m,i}^{(j)}$  –  $i$ -та координата вектора, яка приймає одиничне значення, якщо значення  $i$ -ої ознаки знаходиться в полі допусків  $\delta_{k,i}$ , і нульове значення, якщо не знаходиться;  $n_{\min}$  – мінімальна кількість випробувань, яка забезпечує репрезентативність навчальної вибірки.

При обґрунтуванні гіпотези компактності (чіткої, або нечіткої) за геометричний центр класу  $X_m^o$  приймається вершина двійкового еталонного вектора  $x_m$ .

Еталонний вектор-реалізація  $x_m$  – це математичне сподівання реалізацій класу  $X_m^o$ . Подамо детермінований вектор  $x_m$  у вигляді структурованого двійкового (бінарного) вектора:

$$x_m = \langle x_{m,1}, \dots, x_{m,i}, \dots, x_{m,N} \rangle, m = \overline{1, M},$$

де  $x_{m,i}$  –  $i$ -та координата вектора, яка приймає одиничне значення, якщо значення  $i$ -ої ознаки знаходиться в нормованому полі допусків  $\delta_{m,i}$ , і нульове значення, якщо не знаходиться.

$$l_{m,d} = \frac{d_m^*}{d(x_m \oplus x_c)} \quad (2.1)$$

Процес навчання полягає в мінімізації виразу (2.1).

В МФСВ, який ґрунтується на допущенні гіпотези компактності (чіткої або нечіткої) реалізацій образу, як наближення "точної" роздільної гіперповерхні для класу  $X_m^o$  розглядається гіперсфера, центром якої є еталонний  $x_m \in X_m^o$ , а радіусом  $d_m$  – кодова відстань, яка у просторі Хеммінга визначається за формулою:

$$d_m = \sum_{i=1}^N (x_{m,i} \oplus \lambda_{m,i}), \quad (2.2)$$

де  $x_{m,i}$  –  $i$ -та координата вектора  $x_m \in X_m^o$ ;  $\lambda_{m,i}$  –  $i$ -та координата деякого вектора-реалізації  $\lambda_m$ , вершина якого знаходиться на роздільній гіперповерхні класу  $X_m^o$ ;  $\oplus$  – операція складання за модулем два.

Алгоритм навчання за МФСВ полягає в реалізації багатоцикличної ітераційної процедури оптимізації структурованих просторово-часових параметрів функціонування СППР шляхом пошуку глобального максимуму усередненого за алфавітом  $\{X_m^o\}$  значення КФЕ навчання.

Оптимальною кодовою відстанню (радіусом) між вектором  $x_m$  і контейнером  $K_m^o$  називається екстремальне значення  $d_m^*$ , яке визначає максимум інформаційного КФЕ  $E_m^* = \max_{\{d\}} E_m$ , де  $\{d\}$  – послідовність збільшень радіуса контейнера  $K_m^o$ .

Побудова оптимальної в інформаційному сенсі гіперповерхні у вигляді гіперсфери за МФСВ зводиться до оптимізації радіуса роздільної гіперсфери  $d_m$ , яка відбувається за ітераційним алгоритмом (2.3):

$$d_m(k) = d_m(k-1) + \begin{cases} h, & \text{if } E_m[d(k)] \geq E_m[d(k-1)]; \\ 0, & \text{if } \text{else,} \end{cases} \quad (2.3)$$

де  $k$  – змінна числа збільшень радіуса гіперповерхні;  $d_m(0) = 0$ ;  $h$  – крок збільшення.

Процедура закінчується при знаходженні екстремального значення критерію  $E_m^* = \max_{\{d\}} E_m$ , де  $\{d\} = \{0, 1, \dots, d(x_m \oplus x_{m+1})\}$  – множина радіусів концентрованих гіперсфер, центр яких визначається вершиною еталонного вектора  $x_m \in X_m^o$ ;  $x_{m+1}$  – еталонний вектор найближчого (до  $X_m^o$ ) класу  $X_{m+1}^o$ .

## 2.2 Математичні моделі систем розпізнавання символів

Розглянемо математичну модель інтелектуальної системи, яка реалізує класичну задачу розпізнавання образів за МФСВ [9]. Математична модель повинна включати як обов'язкову складову частину вхідний математичний опис, який подамо на рівні системного аналізу у вигляді теоретико-множинної структури  $\Delta = \langle G, T, \Omega, Z, \Psi, \Pi, \Phi \rangle$ , де  $G$  – простір вхідних сигналів (факторів);  $T$  – множина моментів часу зняття інформації;  $\Omega$  – простір ознак розпізнавання;  $Z$  – простір можливих станів;  $Y$  – множина сигналів після первинної обробки інформації;  $\Pi: G \times T \times \Omega \rightarrow Z$  – оператор переходів, що відбиває механізм зміни станів під дією внутрішніх і зовнішніх збурень;  $\Phi: G \times T \times \Omega \times Z \rightarrow Y$  – оператор оброблення зображення (формування вибіркової множини  $Y$  на вході СППР), який є реакцією на внутрішні і зовнішні збурення. Як універсум випробувань  $W$  розглядається декартовий добуток:  $W = G \times T \times \Omega \times Z$ .

Для чіткого детермінованого розбиття  $\mathfrak{R}^{[M]}$  потужності  $M$ , оператор побудови розбиття  $\eta$  задає відображення  $\eta: Y \mapsto \mathfrak{R}^{[M]}$ . Априорно оператором  $\nu: Y \rightarrow L^{[M]}$  будується покриття  $L^{[M]} = \{X_n^a\}$ , яке визначає абетку класів розпізнавання і на якому на відміну від розбиття відношення еквівалентності відсутнє. Оператор  $\alpha: \mathfrak{R}^{[M]} \rightarrow L^{[M]}$  переводить апостеріорно розбиття  $\mathfrak{R}^{[M]}$  в покриття  $L^{[M]}$  і замикає комутативне кільце. Перевірка гіпотези про належність реалізацій образу здійснюється оператором класифікації  $\Psi: \mathfrak{R}^{[M]} \rightarrow I^{[M+1]}$ , де  $I^{[M+1]} = \{\gamma_1, \gamma_2, \dots, \gamma_M, \gamma_{M+1}\}$  – множина допустимих гіпотез (рішень). При цьому гіпотеза  $\gamma_{M+1}$  означає відмову від класифікації. У загальному випадку математичну модель будь-якої системи керування, яка розв'язує задачу класифікаційного аналізу, подамо у вигляді діаграми відображень відповідних множин (рис. 2.1):

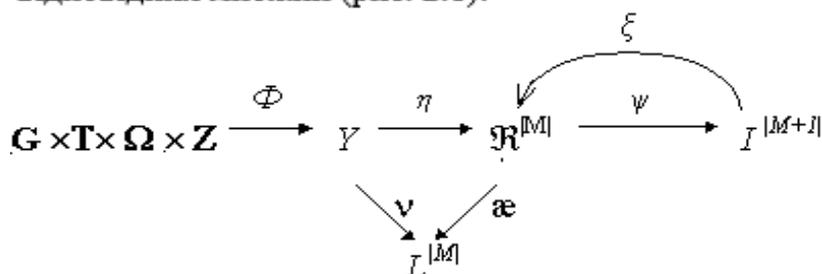


Рисунок 2.1 – Математична модель навчання системи розпізнавання

Оператор зворотного зв'язку  $\xi$  корегує геометричні параметри розбиття  $\mathfrak{R}^{[M]}$  з метою поліпшення точнісних характеристик. Саме спосіб реалізації оператора  $\xi$  породжує різні методи та підходи до розв'язання проблеми адаптивного навчання системи керування.

При обґрунтуванні гіпотези нечіткої компактності має місце нечітке розбиття  $\tilde{\mathfrak{R}}^{[M]} \subset \Omega$ . Тоді замість оператора  $\eta$  застосуємо оператор  $\theta$  нечіткої факторизації простору ознак:  $\theta: Y \rightarrow \tilde{\mathfrak{R}}^{[M]}$ . Нехай оператор класифікації  $\psi: \tilde{\mathfrak{R}}^{[M]} \rightarrow I^{[l]}$  визначає перевірку основної статистичної гіпотези про належність реалізацій  $\{x_m^{(j)} \mid j=\overline{1, n}\}$  класу  $X_m^o$ , де  $l$  – кількість статистичних гіпотез. Принциповою відмінністю МФСВ від відомих методів автоматичної класифікації є ітераційна оптимізація процесу навчання за інформаційним КФЕ. Для обчислення КФЕ оператор  $\gamma: I^{[l]} \rightarrow \mathfrak{Z}^{[q]}$  шляхом оцінки прийнятих гіпотез  $\gamma l$  формує множину ТХ  $\mathfrak{Z}^{[q]}$ , де  $q=l2$  – кількість точносних характеристик (ТХ), а оператор  $\varphi: \mathfrak{Z}^{[q]} \rightarrow E$  обчислює множину значень інформаційного КФЕ, який є функціоналом від ТХ. Ітераційний процес оптимізації геометричних параметрів розбиття  $\tilde{\mathfrak{R}}^{[M]}$  реалізується оператором  $r: E \rightarrow \tilde{\mathfrak{R}}^{[M]}$  шляхом пошуку максимуму КФЕ  $E_m^* = \max_{\{d\}} E_m(d)$ , де  $\{d\}$  – множина кроків навчання розпізнаванню реалізацій класу  $X_m^o$ . Структурна діаграма процесу навчання за МФСВ для випадку нечіткого розбиття має вигляд:

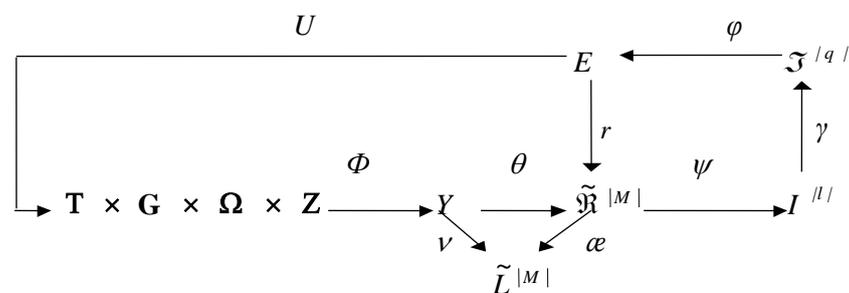


Рисунок 2.2 –Процес навчання за МФСВ у випадку нечіткого розбиття

Таким чином, на рис. 2.2 контур операторів  $\boxed{\psi \rightarrow \gamma \rightarrow \varphi \rightarrow \tau}$  безпосередньо оптимізує геометричні параметри розбиття  $\tilde{\mathfrak{R}}^{|\mathcal{M}|}$ . Оператор  $U: E \rightarrow G \times T \times \Omega \times Z$  регламентує процес навчання. Оскільки в цьому контурі також застосовано процедуру обчислення інформаційного критерію, то він дозволяє розв'язувати задачу оптимізації параметрів плану навчання, які визначають, наприклад, обсяг і тривалість випробовувань, потужність словника ознак, черговість подання для навчання класів розпізнавання та інше.

### 2.3 Математична модель формування ієрархічного класифікатора у рамках ІЕІ-технології

Аналізуючи п. 2.2 (базову математичну модель) було розроблено математичну модель для системи автоматичного формування ієрархічного класифікатора (рис. 2.3).

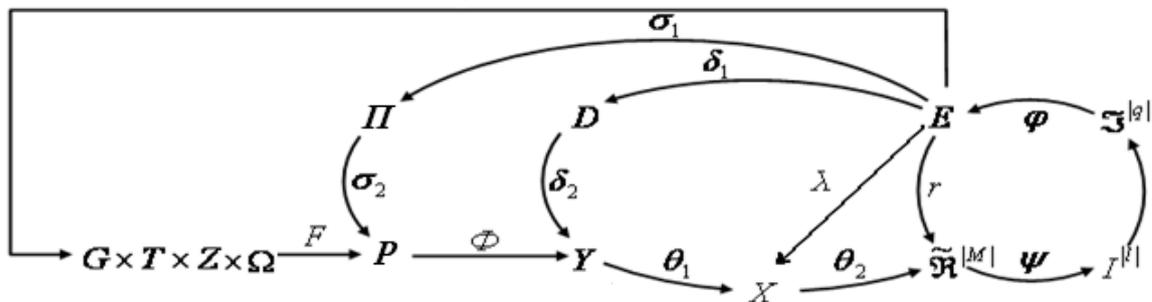


Рисунок 2.3 - Структурна діаграма процесу оптимізації ієрархічного класифікатора та рівнів селекції за ІЕІТ

Тут в базову математичну модель був додано новий контур оптимізації (з урахуванням вимог до контурів оптимізації генотипних параметрів за МФСВ), який відповідає за оптимізацію ієрархічного класифікатора СК, що навчається (рис. 2.4).



Рисунок 2.4 – Контур операторів процесу оптимізації ієрархічного класифікатора

Де  $P$  – множина правил формування ієрархій,  $\Pi$  – множина стратегій оптимізації цих правил, а  $\sigma_1$  та  $\sigma_2$  - це відповідні оператори, які регламентують процес побудови ієрархій.

Як критерій функціональної ефективності ієрархій використовується ентропійна міра Шеннона:

$$E = 1 + \frac{1}{2} \left( \frac{\alpha}{\alpha + D_2} \log_2 \frac{\alpha}{\alpha + D_2} + \frac{D_1}{D_1 + \beta} \log_2 \frac{D_1}{D_1 + \beta} + \frac{\beta}{D_1 + \beta} \log_2 \frac{\beta}{D_1 + \beta} + \frac{D_2}{\alpha + D_2} \log_2 \frac{D_2}{\alpha + D_2} \right).$$

де  $D_1, D_2, \alpha, \beta$  – точнісні характеристики процесу навчання: перша та друга достовірності, помилки першого та другого роду відповідно.

## 3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

### 3.1 Формування вхідного математичного опису

Оптимізацію параметрів навчання інтелектуальної системи будемо проводити загального випадку, коли алфавіт класів складається з шістнадцяти класів. Кожна реалізація в навчальній матриці представляє собою геометричний опис траєкторії, за якою переміщується перо при формуванні символу (рис. 3.1). Таким чином, в навчальній матриці фіксуються послідовність координат точок такої траєкторії.

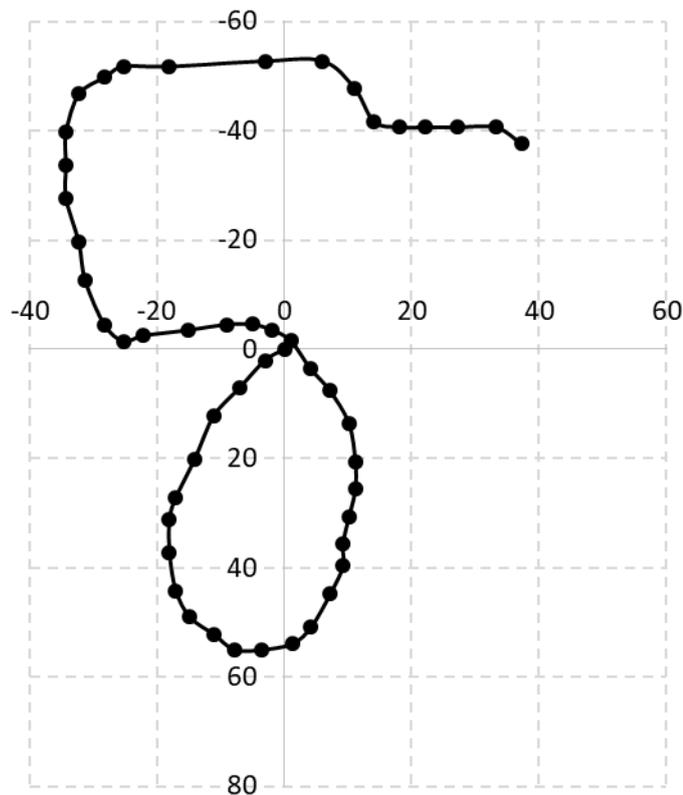


Рисунок 3.1 – Формування реалізації навчальної матриці

На рис. 3.1. темна суцільна лінія відображає символ, а сірі круги – точки, координати яких увійдуть до навчальної матриці. При цьому координати першої точки підпису будемо вважати  $(0; 0)$ . Оскільки навчальні матриці в інформаційно-екстремальній інтелектуальній технології повинні містити однакову кількість ознак розпізнавання для всіх класів, то кількість точок,

координати яких будуть відображатися в початковій матриці буде однаковою. В роботі використовується 50 таких точок. При цьому навчальну матрицю умовно можна розділити на дві частини: таку, що зберігає перші координати точок, та таку, що зберігає другі координати. Таким чином, кількість ознак розпізнавання в навчальній матриці складає 100. Слід зауважити, що в залежності від складності підпису, яка в більшості випадків пов'язана з довжиною траєкторії пера, буде змінюватися щільність таких точок, а їх кількість залишиться незмінною. Іншим параметром навчальної вибірки є кількість реалізацій, тобто прикладів написання кожного символу. При цьому мінімальний обсяг вибірки за інформаційно-екстремальною інтелектуальною технологією дорівнює 40 реалізацій. Для прискорення процесу формування навчальної матриці кожний підпис вводиться в одному екземплярі. Інші реалізації формуються автоматично шляхом часткової незначної зміни координат точок, сформованих за першою реалізацією. Приклад реалізації сформованої автоматично наведено на рис. 3.2.

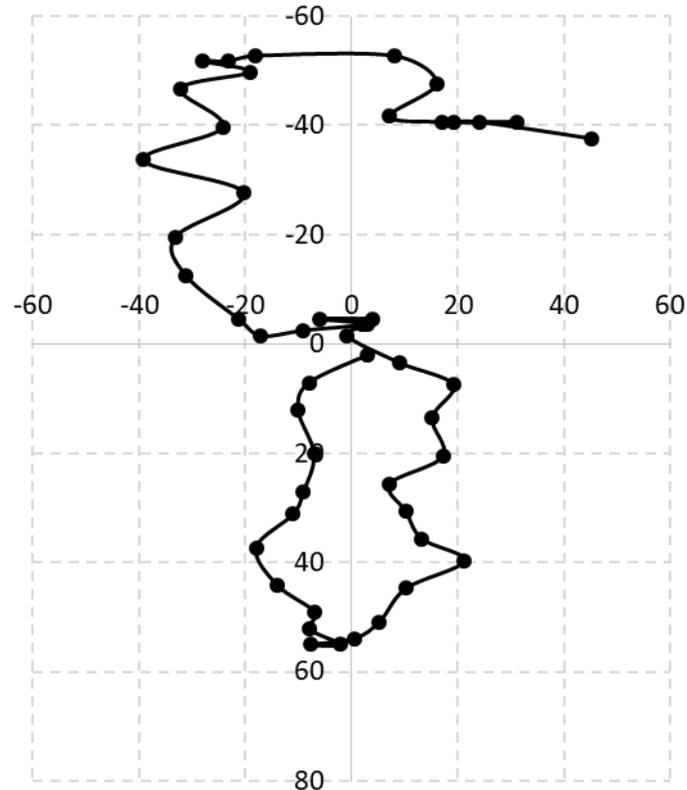


Рисунок 3.2 – Реалізація навчальної матриці, сформована автоматично за зображенням символу на рис. 3.1

### 3.2 Алгоритм оцінки ефективності ієрархічного класифікатора

Алгоритм оцінки ефективності ієрархічного класифікатора створено на основі базового алгоритму навчання за ІЕІТ, з деякими змінами, що обумовлено специфікою задачі. Структурна схема даного алгоритму зображена на рис. 3.3

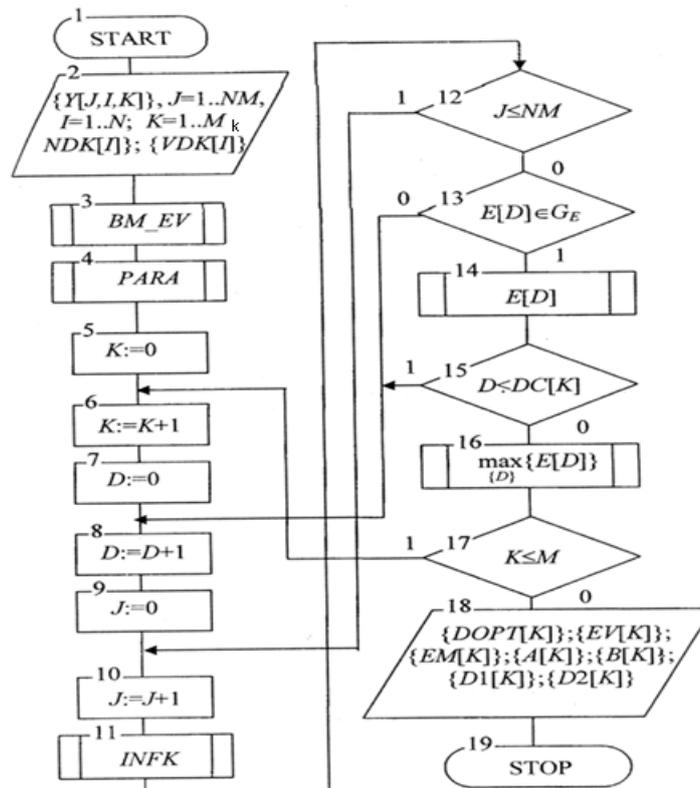


Рисунок 3.3 – Структурна схема алгоритму навчання для k-го рівня ієрархії

Тут показано такі вхідні дані:  $\{Y[J,I,K]\}$  – масив навчальних вибірок,  $J=1..NM$  – змінна кількості випробувань, де  $NM$  – мінімальний обсяг репрезентативної навчальної вибірки,  $I=1..N$  – змінна кількості ознак розпізнавання,  $K=1..M_k$  – змінна кількості ієрархій;  $\{NDK[I]\}$ ,  $\{VDK[I]\}$  – масиви нижніх і верхніх контрольних допусків на ознаки відповідно. Результатом реалізації алгоритму є:  $\{DOPT[K]\}$  – цілий масив оптимальних значень радіусів контейнерів класів розпізнавання у кодівій відстані

Хеммінга;  $\{EV[K]\}$  – масив еталонних двійкових векторів класів розпізнавання;  $\{EM[K]\}$  – дійсний масив максимальних значень інформаційного КФЕ процесу навчання;  $\{D1[K]\}$ ,  $\{A[K]\}$ ,  $\{B[K]\}$ ,  $\{D2[K]\}$  – дійсні масиви оцінок екстремальних значень точнісних характеристик процесу навчання для відповідних класів розпізнавання: перша вірогідність, помилки першого та другого роду і друга вірогідність відповідно. Змінна  $D$  є робочою змінною кроків навчання, на яких послідовно збільшується значення радіуса контейнера. У структурній схемі алгоритму (рис. 2.4.1) блок 3 формує масив навчальних двійкових вибірок  $\{X[J,I,K]\}$  шляхом порівняння значень елементів масиву  $\{Y[J,I,K]\}$  з відповідними контрольними допусками і формує масив еталонних двійкових векторів  $\{EV[K]\}$  шляхом статистичного усереднення стовпців масиву  $\{X[J,I,K]\}$  за при відповідному рівні селекції, який за умовчанням дорівнює  $\rho_m = 0,5$ . Блок 4 здійснює розбиття множини еталонних векторів на пари “найближчих сусідів”. Блок 11 обчислює на кожному кроці навчання значення інформаційного КФЕ і оцінки точнісних характеристик процесу навчання. При невиконанні умови блоку порівняння 12 блок 13 оцінює належність поточного значення критерію  $E[D]$  робочій області  $G_\epsilon$  визначення його функції і при позитивному рішенні блоку 13 це значення запам'ятовується блоком 14. При негативному рішенні блоку порівняння 15, в якому величина  $DC[K]$  дорівнює кодовій відстані між парою сусідніх еталонних векторів, блок 16 здійснює у робочій області  $G_\epsilon$  пошук глобального максимуму КФЕ –  $EM[K]$  і визначає для нього екстремальне значення радіуса гіперсфери –  $DOP1[K]$ . Аналогічно будуються оптимальні контейнери для інших класів. Якщо параметри навчання  $\{DOP1[K]\}$  і  $\{EV[K]\}$  є вхідними даними для екзамену, то значення КФЕ та екстремальних оцінок точнісних характеристик використовуються для аналізу ефективності процесу навчання. Таким чином, основною процедурою базового алгоритму навчання за ІЕП є обчислення на кожному кроці навчання статистичного інформаційного КФЕ і організація пошуку його глобального максимуму в робочій області визначення функції критерію.

Основною відмінністю даного алгоритму від базового є змінені процедура PARA. Наприклад на певному кроці (рис. 3.4) оптимізації ієрархії необхідно побудувати класифікатор для класу  $X^2$ .

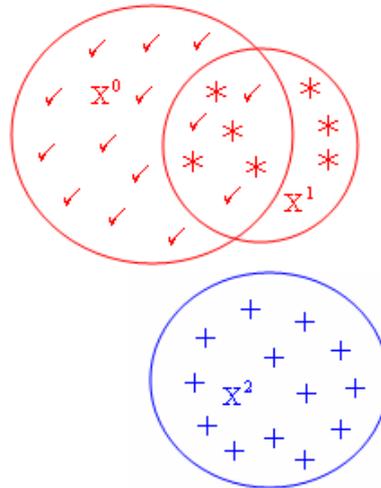


Рисунок 3.4 – Процес побудови контейнера на  $k$ -му рівні

При цьому класифікатор для класів  $X^0$  та  $X^1$  побудувати неможливо, тому що вони перетинаються в значній мірі. Класифікатор для цих класів ми будували на наступному рівні (рис. 3.5), тому що в даному випадку кодова відстань в процедурі PARA не враховується.

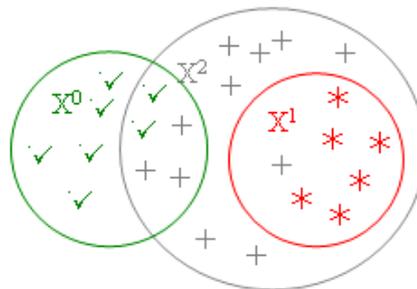


Рисунок 3.5 – Процес побудови контейнера на  $k+1$ -му рівні

### 3.3 Короткий опис програмної реалізації

MATLAB - це програма для програмування, моделювання та моделювання, яка об'єднує настільне середовище, налаштоване для ітеративного аналізу та розробки процесів з мовою програмування, яка безпосередньо виражає математичну та матричну математику.

Система автоматизованої інженерії (CAE) була створена компанією програмного забезпечення MathWorks, щоб допомогти користувачам аналізувати дані, розробляти алгоритми і створювати моделі. Він може похвалитися професійно розробленими інструментами, які пройшли ретельне тестування та повну документацію. Інтерактивні програми показують, як різні алгоритми працюють з даними користувачів. Технологія також масштабована, так що аналізи можуть працювати на кластерах, графічних процесорах та хмарах з мінімальними змінами коду, що виключає необхідність переписування або вивчення великих даних.

Рішення MATLAB для Deep Learning дозволяють користувачам проектувати, будувати і візуалізувати згорткові нейронні мережі, використовуючи лише кілька рядків. Навіть з обмеженими технічними знаннями користувачі можуть запускати розгорнуті моделі в 7 разів швидше, ніж інші подібні програми. Користувачі можуть отримати доступ до оновлених моделей, таких як GoogLeNet, VGG-16, VGG-19, AlexNet, ResNet-50, ResNet-101 і Inception-v3.

MATLAB інтегрована з різними рішеннями для обробки зображень і комп'ютерного зору. Користувачі можуть отримувати зображення та відео з апаратного забезпечення візуалізації, використовувати графічні інструменти для візуалізації та маніпулювання ними та мігрувати їх до вбудованого програмного забезпечення. Бібліотеки стандартних довідкових алгоритмів також допомагають генерувати новіші, більш свіжі ідеї.

MATLAB дозволяє легко використовувати методи обробки сигналів для вивчення та аналізу даних часових рядів. Програмне забезпечення забезпечує уніфікований робочий процес для розробки вбудованих систем і потокових додатків. Сигнали можуть бути отримані, виміряні і оцінені з декількох джерел. Алгоритми потоку для аудіо, смарт-сенсорів, приладів та пристроїв IoT можуть бути розроблені, перевірені та реалізовані з використанням широких можливостей MATLAB.

MATLAB є корисним для фінансових організацій, оскільки розробляє якісні, прозорі, документовані та відтворювані моделі ризиків і стрес-тестування. Він може похвалитися простими у використанні інструментами, які дозволяють створювати моделі лише за лічені дні, що є важливим з огляду на швидко змінюється регуляторне та бізнес середовище. Можна уникнути ризиків, пов'язаних з моделлю та операційними процесами, оскільки програмне забезпечення дозволяє користувачам включати кращі практики розробників, які розуміють ризик, при впровадженні контролю та автоматизації моделей. Єдиний стек моделі ризику може обслуговувати декілька режимів відповідності та функцій переднього та середнього офісу, що робить організації більш продуктивними.

MATLAB є корисним для дослідників і інженерів робототехніки. Технологія може розробляти і налаштовувати алгоритми, моделювати реальні системи і автоматично генерувати код, використовуючи одну інтегровану платформу. Користувачі можуть підключатися і управляти своїм роботом з розробленими алгоритмами. Також доступні інструменти для генерування апаратно-агностичних алгоритмів і підключення до операційної системи робота (ROS). Користувачі можуть надсилати керуючі сигнали або аналізувати декілька типів даних, підключаючи до датчиків і виконавців.

Алгоритми оптимізації в інформаційному розумінні функціональних параметрів інтелектуальної системи розпізнавання символів за інформаційно-екстремальною інтелектуальною технологією були реалізовані у вигляді набору m-функцій:

Формування бінарної навчальної матриці та еталонних векторів

```
function [BM,EV] = bmev(y,vd,nd);
```

де вхідні параметри:

y- навчальна матриця,

vd,nd- система контрольних допусків,

вихідні параметри:

BM- бінарна матриця,

EV- еталонні вектори.

Формування системи контрольних допусків

```
function [VD,ND] = dopuski(y,delta);
```

де вхідні параметри:

y- навчальна матриця,

delta- ширина поля контрольних допусків,

вихідні параметри:

VD – верхній допуск,

ND - нижній допуск.

Обчислення критерію функціональної ефективності та точносних характеристик

```
function [e,d1,d2,alfa,betta] = KFE(k1,k2,n);
```

де вхідні параметри:

k1,k2- кількість реалізацій базового та сусіднього класу в побудованому контейнері,

n - загальна кількість реалізацій,

вихідні параметри:

d1,d2,alfa,betta - значення точносних характеристик,

Обчисляє кількість реалізацій базового та сусіднього класу в побудованому контейнері

```
function [k1,k2] = KK(sk,N);
```

де вхідні параметри:

sk – масив кодівих відстаней ,

N – кількість ознак розпізнавання,

вихідні параметри:

k1,k2- кількість реалізацій базового та сусіднього класу в побудованому контейнері,

Формування масиву кодівих відстаней

```
function [SK] = kodR(ev,bm,kc,k);
```

де вхідні параметри:

$k_c$  – номер класу-сусіда,

$k$  – номер базового класу,

$bm$ - бінарна матриця,

$ev$  - еталонні вектори.

вихідні параметри:

$SK$  – масив кодових відстаней

Завантаження навчальної матриці

```
function [Y] = loadY(m,N,n);
```

де вхідні параметри:

$m$  – кількість класов,

$N$  – кількість ознак розпізнавання,

$n$  - загальна кількість реалізацій,

вихідні параметри:

$Y$  – навчальна матриця.

Оптимізація геометричних параметрів контейнерів класів

```
function [Emax,Dopt,D1,D2]=opt(ev,bm,kc,dc,rep);
```

де вхідні параметри:

$bm$ - бінарна матриця,

$ev$ - еталонні вектори.

$k_c$  – номер класу-сусіда,

$dc$  – кодова відстань між еталонними векторами,

$rep$  – змінна що вказує на необхідність формування звіту в процесі

навчання.

вихідні параметри:

$E_{max}$  – максимальне значення критерію функціональної ефективності,

$D_{opt}$  – оптимальне значення радіусу контейнера.

$D1, D2$  - значення точносних характеристик,

Оптимізація системи контрольних допусків

```
function [EmaxS,VDopt,NDopt,Emax,Dopt,D1,D2]=optDelta(y,deltaMax);
```

де вхідні параметри:

у- навчальна матриця,

deltaMax – максимальне значення ширини поля допусків.

вихідні параметри:

Emax – максимальне значення критерію функціональної ефективності,

D1,D2 - значення точносних характеристик,

EmaxS – максимально усереднене значення критерію функціональної ефективності,

VDopt – оптимальний верхній контрольний допуск,

NDopt – оптимальний нижній контрольний допуск,

Dopt – оптимальне значення радіусу контейнера.

Формування масиву номерів еталонних векторів сусідів та відстані до них

```
function [kc,dc] = para(ev);
```

де вхідні параметри:

ev- еталонні вектори.

вихідні параметри:

kc – номер класу-сусіда,

dc – кодова відстань між еталонними векторами.

Визначення робочої області критерію функціональної ефективності

```
function [t]=workD(d1,d2,dc,k);
```

де вхідні параметри:

k – номер класу,

d1,d2 - значення точносних характеристик,

dc – кодова відстань між еталонними векторами.

вихідні параметри:

t – робоча область.

dc – кодова відстань між еталонними векторами.

Програмний код наведено в додатку.

### 3.4 Аналіз результатів

Для перевірки працездатності системи було застосовано алгоритм паралельної оптимізації системи контрольних допусків, результати роботи якого наведено на рис. 3.6

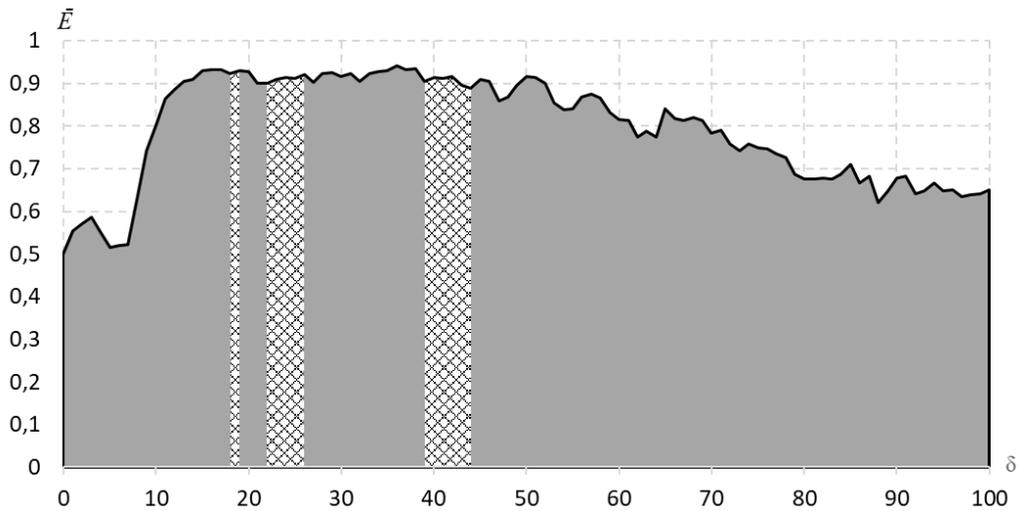
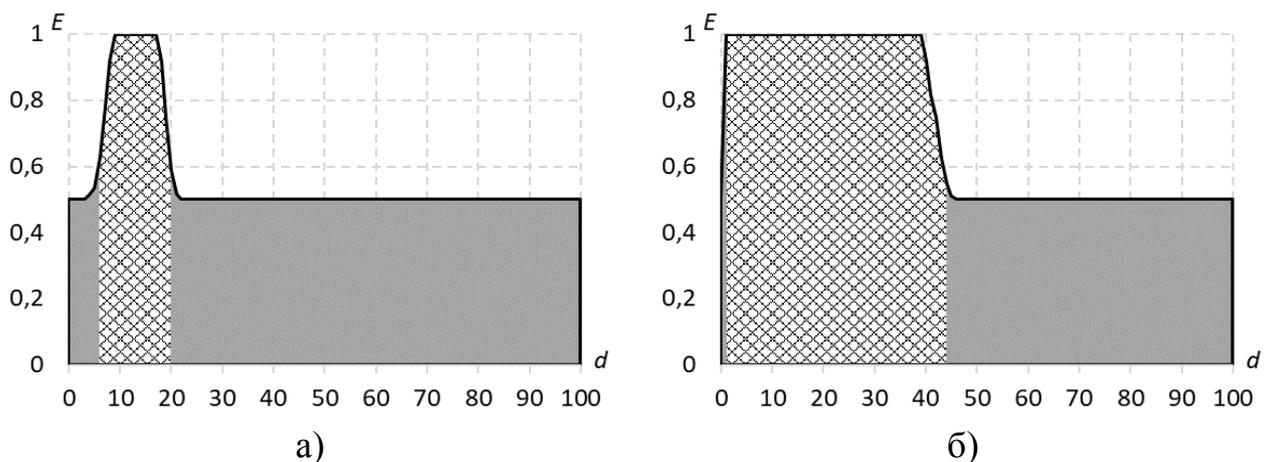
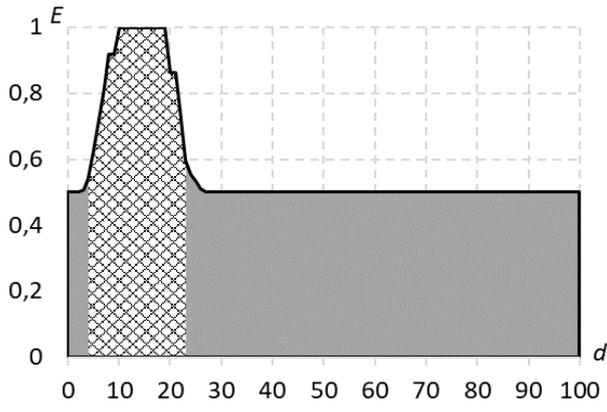


Рисунок 1 – Результати оптимізації системи контрольних допусків.

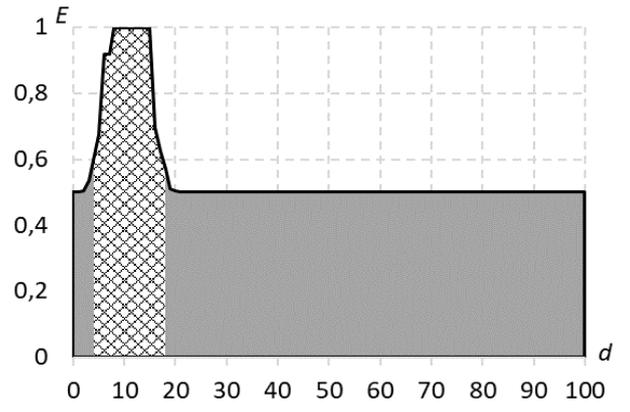
Аналіз рис. 3.6 показує, що максимальне значення усередненого КФЕ дорівнює 0,92224. При цьому оптимальне значення параметру, що визначає ширину поля контрольних допусків,  $\delta = 18$ .

Результати оптимізації геометричних параметрів вирішальних правил при оптимальній системі контрольних допусків наведено на рис. 3.7

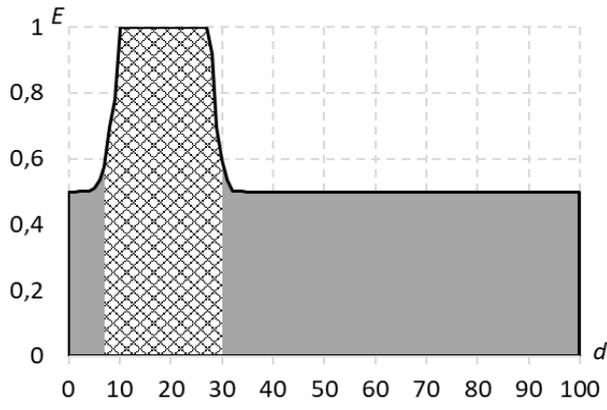




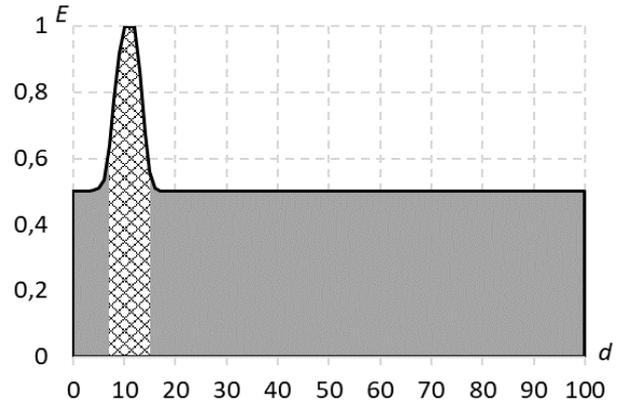
В)



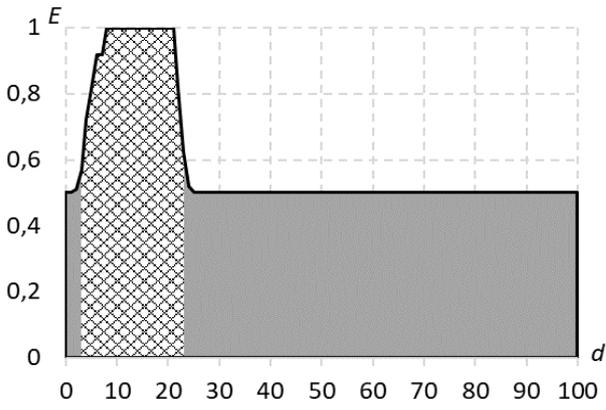
Г)



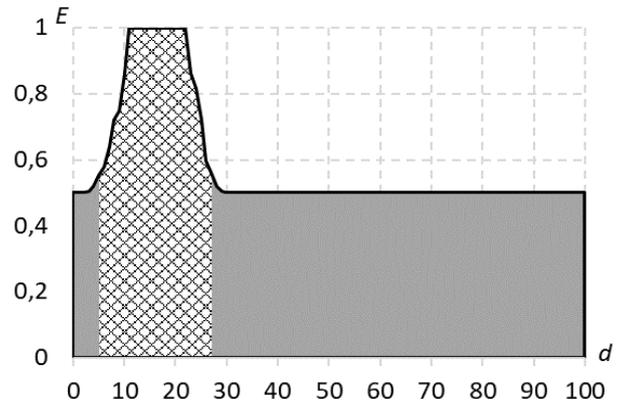
Д)



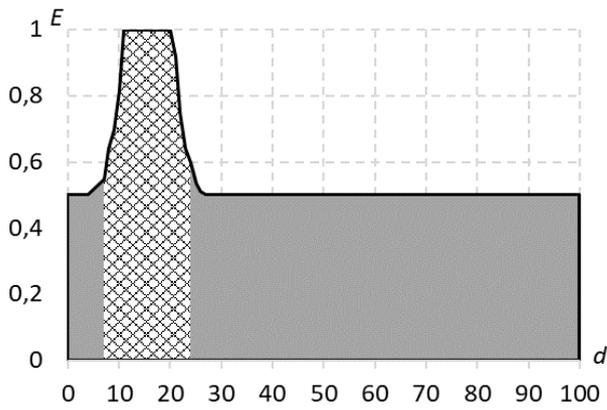
е)



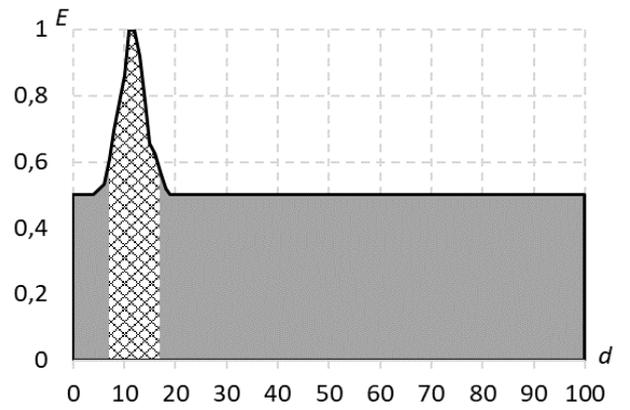
є)



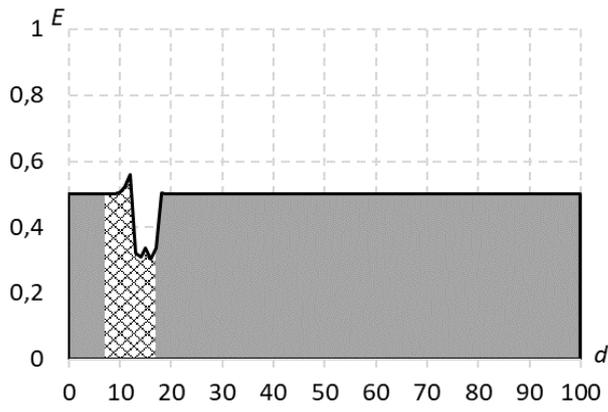
ж)



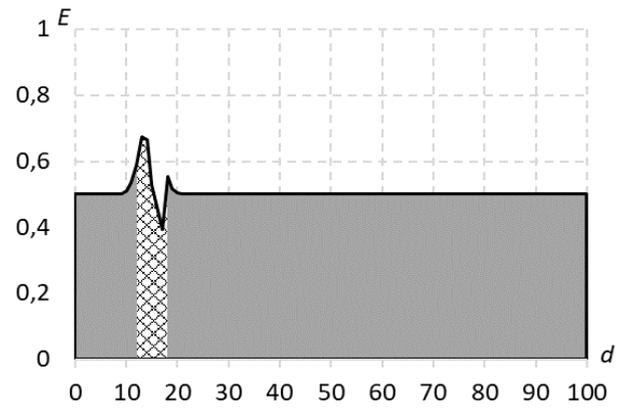
з)



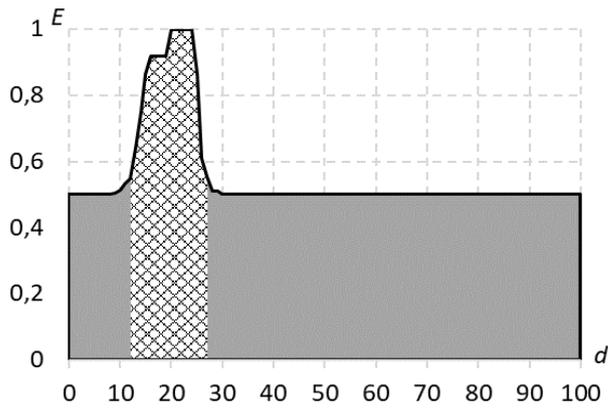
и)



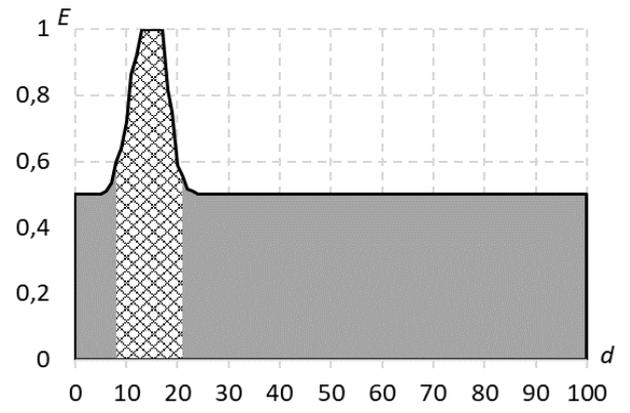
i)



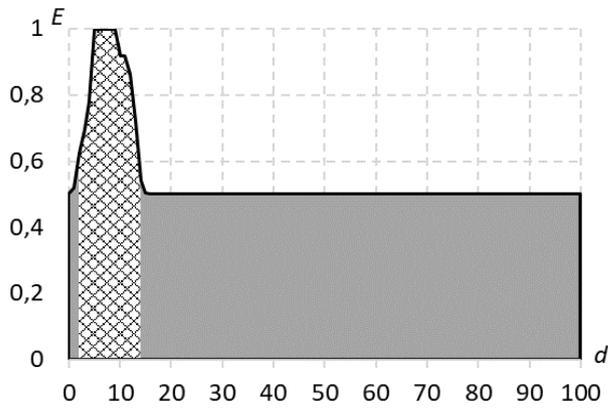
ii)



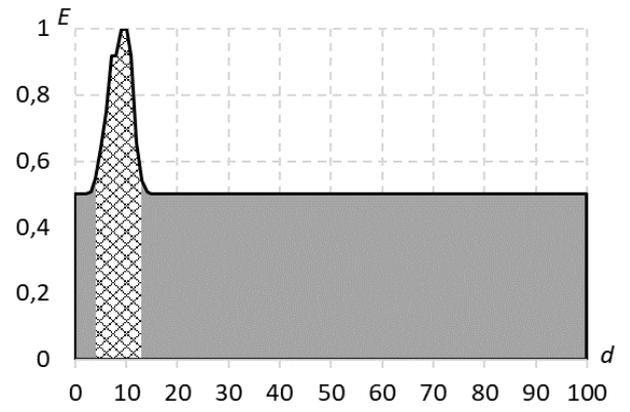
iii)



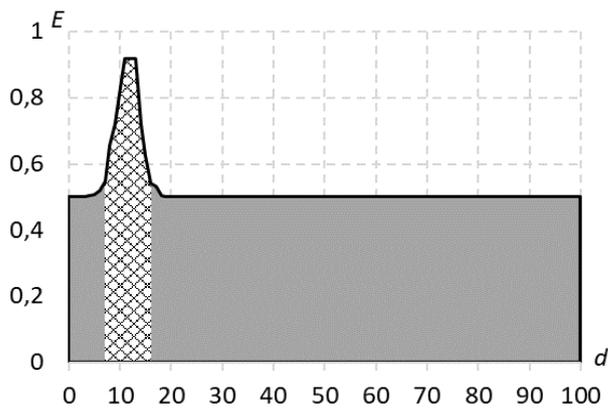
iv)



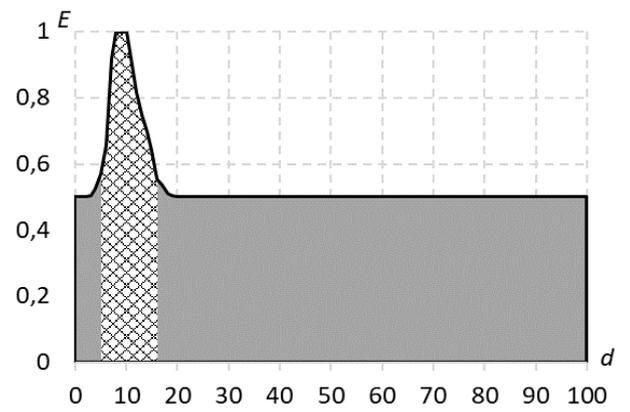
v)



vi)



vii)



viii)

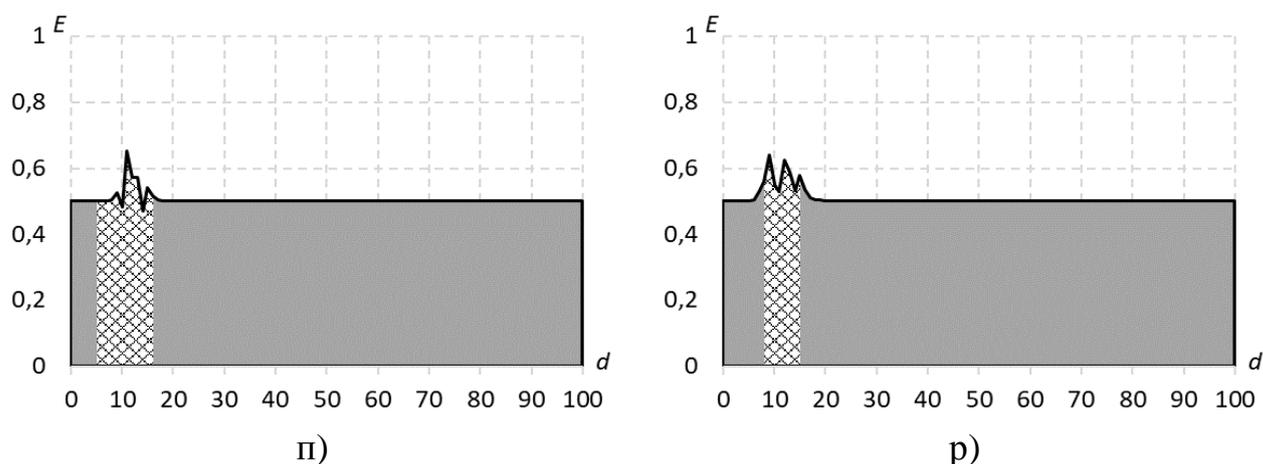


Рисунок 3.7 – Результати оптимізації геометричних параметрів класів: а) літера «а», б) літера «б», в) літера «в», г) літера «г», д) літера «д», е) літера «е», з) літера «э», ж) літера «ж», з) літера «з», и) літера «и», і) літера «і», ї) літера «ї», й) літера «й», к) літера «к», л) літера «л», м) літера «м», н) літера «н», о) літера «о», п) літера «п», р) літера «р».

Аналіз рис. 3.7 показує, що 15 вирішальних правил характеризуються максимальним значення КФЕ, яке дорівнює 1, а точнісні характеристики набувають однакових значень  $D1=1$ ,  $D2=1$ . При цьому вирішальні правила для класів 11, 12, 17, 19, 20 не є безпомилковими. Вони характеризуються такими максимальним значення КФЕ: клас 11 – 0.5604 ( $D1=0,57$ ,  $D2=1$ ), клас 12 – 0.67499 ( $D1=0,80$ ,  $D2=1$ ), клас 17 – 0.91729 ( $D1=0,57$ ,  $D2=0,97$ ), клас 19 – 0.65305 ( $D1=0,88$ ,  $D2=0,97$ ), клас 20 – 0.63904 ( $D1=0,75$ ,  $D2=1$ ).

При цьому, для класу літера «а» оптимальним є радіус  $d=15$  (найближчий клас літера «г», міжцентрова відстань 22), для класу літера «б» оптимальним є радіус  $d=19$  (найближчий клас літера «ї», міжцентрова відстань 40), для класу літера «в» оптимальним є радіус  $d=9$  (найближчий клас літера «з», міжцентрова відстань 25), для класу літера «г» оптимальним є радіус  $d=16$  (найближчий клас літера «е», міжцентрова відстань 18), для класу літера «д» оптимальним є радіус  $d=18$  (найближчий клас літера «к», міжцентрова відстань 28), для класу літера «е» оптимальним є радіус  $d=15$  (найближчий клас літера «г», міжцентрова відстань 16), для класу літера «е»

оптимальним є радіус  $d=15$  (найближчий клас літера «л», міжцентрова відстань 25), для класу літера «ж» оптимальним є радіус  $d=9$  (найближчий клас літера «и», міжцентрова відстань 24), для класу літера «з» оптимальним є радіус  $d=8$  (найближчий клас літера «в», міжцентрова відстань 24), для класу літера «и» оптимальним є радіус  $d=17$  (найближчий клас літера «к», міжцентрова відстань 22), для класу літера «і» оптимальним є радіус  $d=12$  (найближчий клас літера «ї», міжцентрова відстань 14), для класу літера «ї» оптимальним є радіус  $d=11$  (найближчий клас літера «і», міжцентрова відстань 14), для класу літера «й» оптимальним є радіус  $d=10$  (найближчий клас літера «и», міжцентрова відстань 29), для класу літера «к» оптимальним є радіус  $d=10$  (найближчий клас літера «і», міжцентрова відстань 21), для класу літера «л» оптимальним є радіус  $d=16$  (найближчий клас літера «м», міжцентрова відстань 23), для класу літера «м» оптимальним є радіус  $d=15$  (найближчий клас літера «л», міжцентрова відстань 23), для класу літера «н» оптимальним є радіус  $d=10$  (найближчий клас літера «и», міжцентрова відстань 15), для класу літера «о» оптимальним є радіус  $d=10$  (найближчий клас літера «е», міжцентрова відстань 15), для класу літера «п» оптимальним є радіус  $d=20$  (найближчий клас літера «и», міжцентрова відстань 23), для класу літера «р» оптимальним є радіус  $d=19$  (найближчий клас літера «д», міжцентрова відстань 23).

Таким чином, сформований класифікатор 20 рукописних літер не є безпомилковим. Виділемо класи 11, 12, 17, 19, 20 на другий рівень ієрархії.

Застосуємо алгоритм паралельної оптимізації системи контрольних допусків для другого рівня ієрархії. Результати роботи даного алгоритму наведено на рис. 3.8

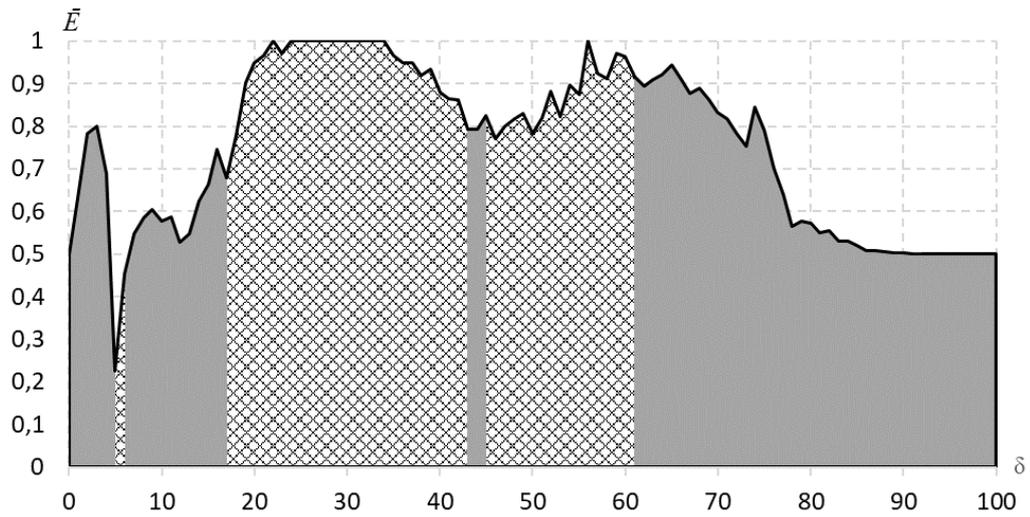
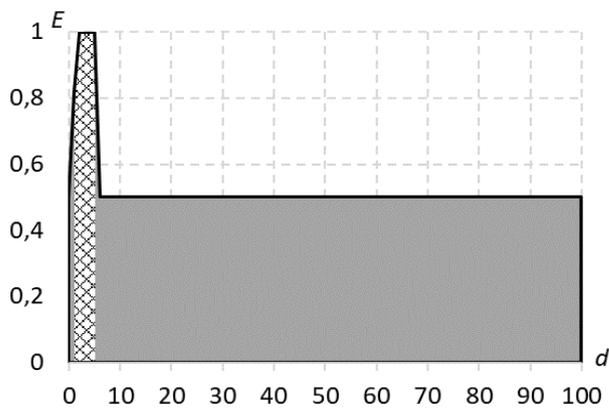
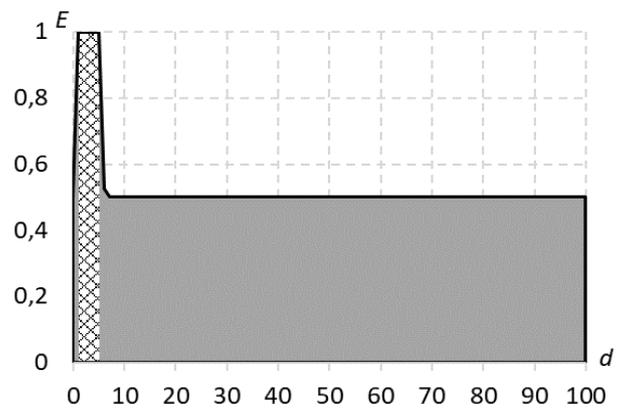


Рисунок 3.8 – Результати оптимізації системи контрольних допусків для другого рівня ієрархії

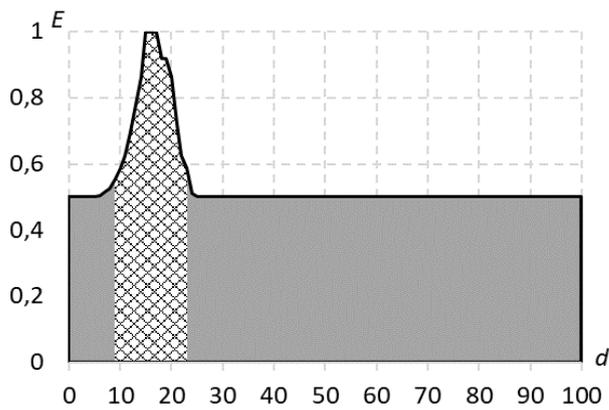
Аналіз рис. 3.8 показує, що максимальне значення усередненого КФЕ дорівнює 1. При цьому оптимальне значення параметру, що визначає ширину поля контрольних допусків,  $\delta = 22$ .



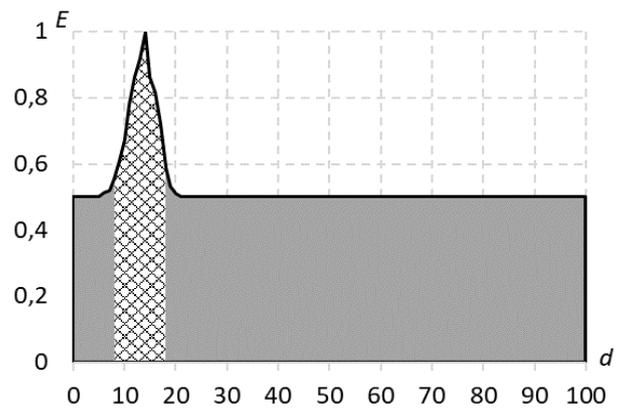
а)



б)



в)



г)

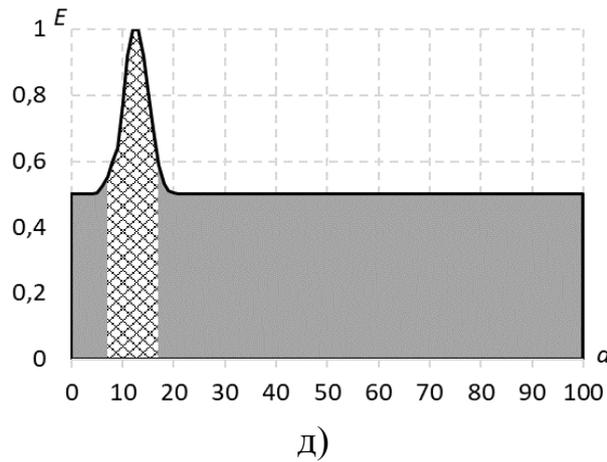


Рисунок 3.9 – Результати оптимізації геометричних параметрів класів: а) літера «і», б) літера «ї», в) літера «н», г) літера «п», д) літера «р».

Аналіз рис. 3.9 показує, що всі вирішальні правила характеризуються максимальним значенням КФЕ, яке дорівнює 1, а точнісні характеристики набувають однакових значень  $D1=1$ ,  $D2=1$ .

При цьому, для класу літера «і» оптимальним є радіус  $d=2$  (найближчий клас літера «ї», міжцентрова відстань 6), для класу літера «ї» оптимальним є радіус  $d=1$  (найближчий клас літера «і», міжцентрова відстань 6), для класу літера «н» оптимальним є радіус  $d=4$  (найближчий клас літера «р», міжцентрова відстань 22), для класу літера «п» оптимальним є радіус  $d=5$  (найближчий клас літера «р», міжцентрова відстань 19), для класу літера «р» оптимальним є радіус  $d=4$  (найближчий клас літера «р», міжцентрова відстань 19).

На рис. 3.10 подано оптимальну ієрархічну структуру вирішальних правил для розпізнавання 20 рукописних символів.

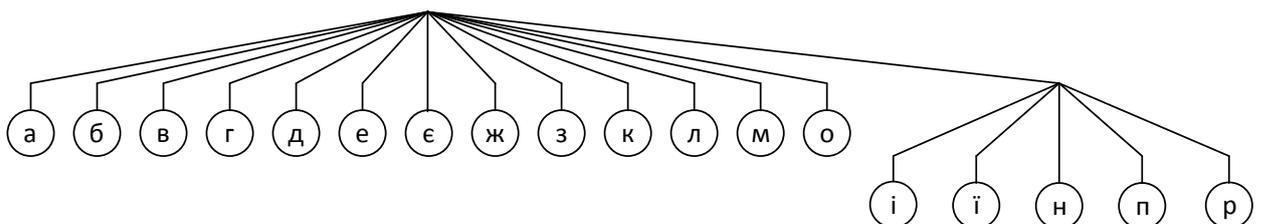


Рисунок 3.10 – Результати оптимізації ієрархічної структури вирішальних правил

## ВИСНОВКИ

В ході виконання кваліфікаційної магістерської роботи було розроблено і програмно реалізовано систему ієрархічного розпізнавання рукописних символів на базі інформаційно-екстремальної інтелектуальної технології. При цьому були виконані такі завдання:

- 1) Сформувано вхідний математичний опис системи ієрархічного розпізнавання рукописних символів, що зданта навчатися та оптимізувати ієрархічну структуру вирішальних правил.
- 2) Створено математичну модель навчання ієрархічного розпізнаванню рукописних символів в рамках інформаційно-екстремальної інтелектуальної технології.
- 3) Обрано критерій функціональної ефективності ієрархічної системи розпізнавання.
- 4) Розроблено та програмно реалізувано алгоритми навчання ієрархічного розпізнавання рукописних символів.
- 5) Перевірено працездатність ієрархічної системи на задачі розпізнавання 20 рукописних символів.

## СПИСОК ЛІТЕРАТУРИ

1. Ahmed S.B., Razzak M.I., Yusof R. Cursive Script Text Recognition in Natural Scene Images: Arabic Text Complexities. – Springer Nature Singapore, 2020. — 121 p.
2. Apprich C., Chun W.H.K., Cramer F., Steyerl H. Pattern Discrimination. – Minneapolis: Meson Press, 2018. — 139 p.
3. Горский Н., Анисимов В., Горская Л. Распознавания рукописного текста: от теории к практике.—СПб.: Политехника, 1997.—126 с.
4. Voigtländer P. Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks / P. Voigtländer, P. Doetsch, H. Ney // 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016. – P. 228-233.
5. Wimmer Z. System and method for superimposed handwriting recognition technology / Z. Wimmer, F. Perraud , P.–M. Lallican , G. Aradilla // United States Patent № US 10,007,859 B2. – Jun. 26, 2018. – P. 19.
6. Yaun S. Handwriting recognition in electronic devices / S. Yaun // United States Patent № US 8,760,389 B2. – Jun. 24, 2014. – P. 13.
7. Impedovo S. More than twenty years of advancements on Frontiers in handwriting recognition / S. Impedovo // Pattern Recognition, 2014. – Vol. 47, Is. 3. – P. 916-928.
8. Yadav P. Handwriting Recognition System-A Review / P. Yadav, S. Yadav // International Journal of Computer Applications, 2015. – Vol. 114, Is. 19. – P. 36-40.
9. Doermann D. Handbook of Document Image Processing and Recognition / D. Doermann, K. Tombre // Springer-Verlag London, 2014. – P. 1055.
10. Краснополюсовський А.С. Інформаційний синтез інтелектуальних систем керування: Підхід, що ґрунтується на методі функціонально-статистичних випробувань. – Суми: Видавництво СумДУ, 2004. – 261с.
11. Довбиш А.С. Основи проектування інтелектуальних систем: Навчальний посібник.— Суми: Видавництво Сум ДУ, 2009.— 171 с.

12. Neetu Bhatia E. Optical Character Recognition Techniques: A Review / E. Neetu Bhatia // International Journal of Advanced Research in Computer Science and Software Engineering, 2014. – Vol. 4, Is. 5. – P. 1219 -1223
13. Полюхин Д.А., Сальников И.И. Методы и этапы распознавания рукописного текста // Научное обозрение. Педагогические науки. – 2019. – № 3-2. – С. 71-74;
14. Хекало В.С Інформаційне та програмне забезпечення інтелектуальної системи розпізнавання рукописних символів.– <https://essuir.sumdu.edu.ua/browse?type=author&value=%D0%A5%D0%B5%D0%BA%D0%B0%D0%BB%D0%BE,%20%D0%92.%D0%A1>.

## ДОДАТОК

```

function [BM,EV] = bmev(y,vd,nd);
[m,N,n]=size(y);
for i=1:N
    BM(:,i,:)=(y(:,i,:)<=vd(i)) & (y(:,i,:)>=nd(i));
end;
EV=round(mean(BM,3));

function [VD,ND] = dopuski(y,delta);
sr=mean(y(2,:,:),3);
VD=sr+delta;
ND=sr-delta;
function [e,d1,d2,alfa,betta] = KFE(k1,k2,n);
d1=k1./n; %перша достовірність
betta=k2./n; %помилка другого роду
alfa=1-d1; %помилка першого роду
d2=1-betta; %друга достовірність
warning off;
e1=(k1./(k1+k2).*log2(k1./(k1+k2))); e1(find(isnan(e1)))=0;
e2=(k2./(k1+k2).*log2(k2./(k1+k2))); e2(find(isnan(e2)))=0;
e3=((n-k1)./(2*n-k1-k2)).*log2((n-k1)./(2*n-k1-k2));
e3(find(isnan(e3)))=0;
e4=((n-k2)./(2*n-k1-k2)).*log2((n-k2)./(2*n-k1-k2));
e4(find(isnan(e4)))=0;
warning on;
e=1+0.5*(e1+e2+e3+e4);
function [k1,k2] = KK(sk,N);

for d=1:N
    k1(d)=length(find(sk(1,:)<=d)); %k1 - кількість своїх реалізацій в
контейнері
    k2(d)=length(find(sk(2,:)<=d)); %k2 - кількість сусідніх реалізацій
в контейнері
end;
function [SK] = kodR(ev,bm,ks,k);
n=size(bm,3);
for j=1:n
SK(1,j)=sum(xor(ev(k,:),bm(k,:,j))); %поточного класу
SK(2,j)=sum(xor(ev(k,:),bm(ks(k),:,j))); %його сусіда
end;
function [Y] = loadY(m,N,n);
for k=1:m
fid = fopen([int2str(k) '.txt']);
Y(k,:,:) = fscanf(fid,'%g',[N n]);

```

```

    fclose(fid);
end;
function [Emax,Dopt,D1,D2]=opt(ev,bm,kc,dc,rep);
[m,N,n]=size(bm);
for k=1:m
[sk] = kodR(ev,bm,kc,k);
[k1,k2]=KK(sk,N);
[e,d1,d2,alfa,betta]=KFE(k1,k2,n);
t=workD(d1,d2,dc,k);
if isempty(t)
    Emax(k)=max(e);
else Emax(k)=max(e(t));
end;
d_o=find(e==Emax(k)); %декілька радіусів, за яких КФЕ максимальний
Dopt(k,:)=d_o(1); %серед них обираємо перший, тобто найменший
D1(k)=d1(Dopt(k));
D2(k)=d2(Dopt(k));
if rep
    fidd = fopen([int2str(k),'res.txt'],'w');
    for d=1:N
        fprintf(fidd,'%3d %6.5f %3.2f %3.2f\n',d, e(d), d1(d), d2(d));
    end;
    fprintf(fidd,'\n %3d %6.5f %3.2f %3.2f %3d %3d\n',d_o(1), Emax(k),
D1(k), D2(k), kc(k), dc(k));
    fclose(fidd);
end;
end;
function [EmaxS,VDopt,NDopt,Emax,Dopt,D1,D2]=optDelta(y,deltaMax);
fid = fopen('delta.txt','w');
for delta=1:deltaMax
    [vd,nd]=dopuski(y,delta);
    [bm,ev]=bmev(y,vd,nd);
    [kc,dc]=para(ev);
    [Emax,Dopt,D1,D2]=opt(ev,bm,kc,dc,0);
    Esr(delta)=mean(Emax);
    ws(delta)=all(and(D1>0.5,D2>0.5));
    %[delta, Esr(delta), ws(delta)]
    fprintf(fid,'%3d %6.5f %1d \n',delta, Esr(delta), ws(delta));
end;
plot(Esr);
wss=find(ws);

if isempty(wss)
    deltas=find(Esr==max(Esr));

```

```

else deltas=find(Esr==max(Esr(wss)));
end;
deltas=find(Esr==max(Esr(wss)));
deltaOpt=deltas(1);
[VDopt,NDopt]=dopuski(y,deltaOpt);
[bm,ev]=bmev(y,VDopt,NDopt);
[kc,dc]=para(ev);
[Emax,Dopt,D1,D2]=opt(ev,bm,kc,dc,1);
EmaxS=mean(Emax);
fclose(fid);
function [kc,dc] = para(ev);
m=size(ev,1);
for k=1:m
    %знаходимо сусідній клас
    for k_c=1:m
        if k==k_c para_k(k_c)=NaN; %до самого себе відстань не обчислюємо
        else
            para_k(k_c)=sum(xor(ev(k,:),ev(k_c,:)));
        end;
    end;
    dc(k)=min(para_k);
    kc_mas=find(para_k==dc(k)); %таких класів може бути декілька
    kc(k)=kc_mas(1);
end;
function [t]=workD(d1,d2,dc,k);
t=find(d1(1:dc(k)-1)>0.5&d2(1:dc(k)-1)>0.5);

```