

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

МАГІСТЕРСЬКА
КВАЛІФІКАЦІЙНА РОБОТА

***на тему: «Інформаційна технологія розпізнавання
електронограм на просвічуючому електронному
мікроскопі»***

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Довбиш А.С.

Студент гр. ІНмз-92

Проценко С. І.

Суми 2021

Сумський державний університет

(назва вузу)

Факультет ІЗДВН Кафедра Комп'ютерних наук

Спеціальність 122 «Комп'ютерні науки»

Затверджую:

зав.кафедри _____

“ _____ ” _____ 20__ р.

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

Проценку Сергію Івановичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)

Інформаційна технологія розпізнавання електронограм на просвічуючому електронному мікроскопі.

затверджено наказом по інституту від “ _____ ” _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту (роботи)

26.01.2021 р.

3. Вхідні данні до проекту (роботи)

Архівні дані електронограм.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми дослідження. 2) Опис методів досліджень

3) Інформаційне, алгоритмічне та програмне забезпечення системи розпізнавання електронограм. 4) Додаток А.

5. Перелік демонстраційного матеріалу

1) Актуальність; 2) Мета; 3) Постановка задачі; 4) Категорійні моделі;

5) Критерій оптимізації параметрів машинного навчання;

6) Схеми алгоритму машинного навчання; 7) Результати моделювання;

8) Висновки.

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>Огляд літератури згідно теми диплома</i>		
2.	<i>Аналіз проблеми дослідження</i>		
3.	<i>Опис методу розв'язання поставленої задачі.</i>		
4.	<i>Інформаційний синтез системи розпізнавання електронограм, що навчається</i>		
5.	<i>Програмна реалізація алгоритму машинного навчання</i>		
6.	<i>Оформлення пояснювальної записки до дипломної роботи</i>		

Студент – дипломник

(підпис)

Керівник проекту

(підпис)

РЕФЕРАТ

Записка: - 75 стр., - рис., 3 табл., 11 бібліографічних джерел.

Мета роботи – підвищення функціональної ефективності машинного навчання системи розпізнавання електронограм, одержаних на просвічуючому електронному мікроскопі в режимі мікродифракції.

Об'єкт дослідження – процес машинного навчання системи розпізнавання електронограм.

Предмет дослідження – категорійні моделі, критерій та метод оптимізації параметрів машинного навчання, вирішальні правила та засоби інформаційної технології розпізнавання електронограм.

Метод дослідження – інформаційно-екстремальний метод машинного навчання системи розпізнавання 2D зображень.

Розроблено алгоритмічне та програмне забезпечення електроно-оптичної системи розпізнавання електронограм в рамках інформаційно-екстремальної інтелектуальної технології, яка ґрунтується на максимізації інформаційної спроможності системи в процесі машинного навчання. Як критерій оптимізації параметрів машинного навчання розглянуто модифікований критерій Кульбака. Досліджено вплив параметрів машинного навчання на функціональну ефективність електроно-оптичної системи розпізнавання електронограм.

ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ,
МАШИННЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ, ЕЛЕКТРОНОГРАМА,
ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, ВИРІШАЛЬНЕ ПРАВИЛО,
ЕЛЕКТРОННИЙ МІКРОСКОП

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	6
1.1 Фізичні основи отримання електронограм на просвічуючому електронному мікроскопі	6
1.2 Аналіз методів розпізнавання 2D зображень	15
1.3 Формалізована постановка задачі інформаційного синтезу системи розпізнавання електронограм	21
2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ.....	25
2.1 Основні положення, принципи та визначення інформаційно- екстремальної інтелектуальної технології аналізу даних	25
2.2 Базовий алгоритм інформаційно-екстремального машинного навчання системи розпізнавання.....	28
2.3 Інформаційний критерій оптимізації параметрів машинного навчання системи розпізнавання.....	32
3 ІНФОРМАЦІЙНЕ, АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЕЛЕКТРОНОГРАМ	37
3.1 Вхідний математичний опис системи розпізнавання електронограм	37
3.2 Категорійні моделі інформаційно-екстремального машинного навчання системи розпізнавання електронограм.....	41
3.3 Алгоритм інформаційно-екстремального машинного навчання з оптимізацією контрольних допусків на ознаки розпізнавання	43
3.4 Опис програмного забезпеченням системи розпізнавання електронограм.....	47
3.5 Результати моделювання.....	55
ВИСНОВКИ.....	62
СПИСОК ЛІТЕРАТУРИ.....	63
ДОДАТОК.....	64

ВСТУП

Розпізнавання електронограм різних речовин, одержаних в електронній мікроскопії у режимі мікродифракції, є важливою задачею якісного експрес-аналізу в металургії, кристалографії, геології та інше. Оскільки ручний аналіз електронограм вимагає значних часових витрат і високої професійної підготовки персоналу, то актуальною задачею є створення автоматизованих систем розпізнавання електронограм із використанням інтелектуальних інформаційних технологій аналізу даних. Розв'язання такої задачі дасть можливість підвищити точність та оперативність аналізу електронограм.

Незважаючи на інтенсивний розвиток методів розпізнавання образів машинна класифікація електронограм все ще залишається нерозв'язаною задачею через такі основні причини:

- нестационарність за яскравістю зображень електронограм;
- велика потужність алфавіту класів електронограм;
- багатовимірність словника ознак розпізнавання;
- велика ступінь перетину класів розпізнавання в просторі ознак;
- вплив неконтрольованих збурюючих факторів при виготовленні електронограм тощо.

Більшість відомих методів інтелектуального аналізу даних орієнтовано на розв'язання модельних задач, які виключають перетин класів і потребують статистичних стійкості та однорідності навчальної вибірки, що на практиці, як правило, не виконується.

В магістерській роботі підвищення функціональної ефективності системи розпізнавання електронограм шляхом машинного навчання системи в рамках інформаційно-екстремальної інтелектуальної технології, яка ґрунтується на максимізації інформаційної спроможності системи в процесі її машинного навчання.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Фізичні основи отримання електроннограм на просвічуючому електронному мікроскопі

Вивчення тонкої будови різноманітних матеріалів і керування їх структурою за допомогою електронного мікроскопа широко використовується як в наукових дослідженнях, така і в промисловому виробництві [1, 2].

Як відомо, електрони – це матеріальні частинки, які мають хвильові властивості. Довжина хвилі електрона λ та його імпульс p пов'язані співвідношенням де Бройля

$$\lambda = \frac{h}{p},$$

де h – стала Планка.

Якщо електрон прискорюється в полі з різницею потенціалів u , то довжина хвилі дорівнює

$$\lambda = \frac{h}{\sqrt{2meu}},$$

де m – маса електрона;

e – енергія електрона.

Для напруги 100 кВ довжина хвилі, що відповідає електрону з енергією 100 кеВ, дорівнює $3.7 \cdot 10^{-3}$ нм, для 50 кеВ – $5.3 \cdot 10^{-3}$ нм. Отже, електрони, подібно до хвиль, можуть дифрагувати, інтерферувати і цим подібні до звичайного світла. Здебільшого рух електронів зручно зображати як поширення світлових хвиль у вигляді променів або нормалей до фронту хвиль.

Однією з основних властивостей електронних пучків є їхня взаємодія з електричними та магнітними полями. При проходженні пучка електронів через область з електричним або магнітним полем виникають ефекти, аналогічні проходженню світла через області з різними показниками заломлення, хоча при цьому спостерігаються і певні відмінності.

Таким чином, електричні та магнітні поля спеціальної конфігурації можуть відігравати роль “лінзи” для електронної хвилі.

У сучасній електронній мікроскопії найбільше поширення здобули так звані магнітні лінзи. Як і у випадку оптичної лінзи, точки об'єкта і точки зображення лежать поза магнітним полем лінзи. Застосування полюсних наконечників дає змогу концентрувати магнітне поле в малій області, що значно збільшує заломлюючу силу (а отже, і світлосилу) лінзи. Фокусна відстань такої лінзи f пропорційна відношенню H/ϑ , де ϑ - швидкість електронів; H - напруженість поля. Беручи до уваги специфіку процесу, проходження електронів через магнітне поле, магнітні лінзи завжди є збірними. Розсіюючих магнітних лінз немає.

Важливою частиною електронного мікроскопа є джерело електронів, яке випромінює потужний пучок електронів і одночасно прискорює їх в електростатичному полі. Цей вузол називається електронною гарматою. Він складається з катода, що забезпечує емісію електронів, керуючого (направляючого) електрода та анода. Досить потужні катоди з густиною струму більше $2 \cdot 10^4$ А/см² виготовляють з вольфраму у вигляді ω -подібних ниток. Регулюючи струм розжарення нитки та величину потенціалу на керуючому електроді, отримують інтенсивні пучки електронів з досить малою апертурою.

На рис 1.1 наведена схема дифракції в електронному мікроскопі, коли він працює в режимі електроннографа «а» та у зворотній решітці «б» [1].

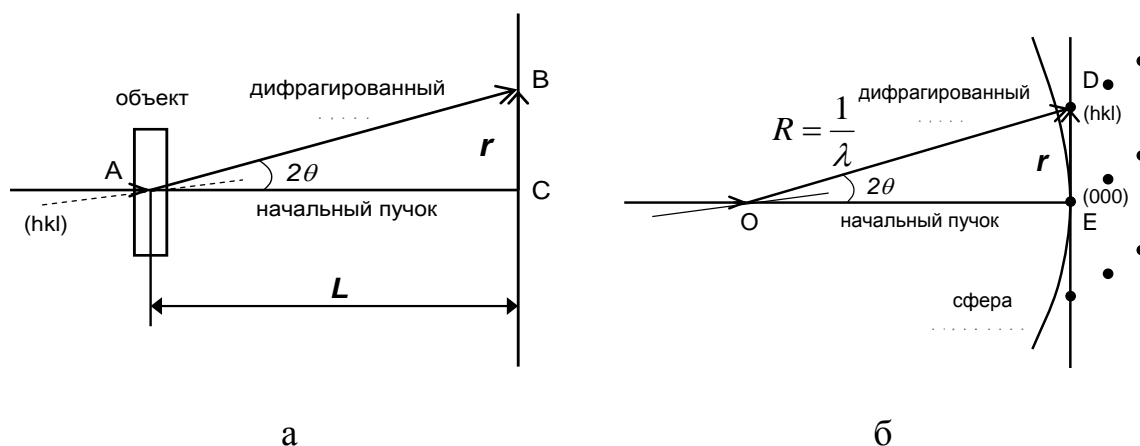


Рисунок 1.1 – Схема дифракції в електронографії

На рис. 1.1 прийнято такі позначення:

L - відстань від об'єкта до фотопластини;

r - відстань на фотопластині від сліду початкового пучка до дифрагированого;

(hkl) - індекси площини, що відбиває;

θ - кут Вульфа – Брегга;

\mathbf{H}_{hkl} - вектор зворотної решітки.

Сферу відбиття заміняють площиною, оскільки трикутники ABC й ODE подібні, то має місце співвідношення

$$\frac{|\mathbf{H}_{hkl}|}{1/\lambda} = \frac{r}{L}. \quad (1.1)$$

Із співвідношення (1.1) відстань r визначається як

$$r = |\mathbf{H}_{hkl}| L \lambda. \quad (1.2)$$

Із запису (1.2) видно, що електронограма являє собою перетин зворотної решітки, що проходить через початковий вузол (000) , у масштабі $L\lambda$.

Використовуючи властивість вектора зворотної решітки:

$$|\mathbf{H}_{hkl}| = \frac{1}{d_{hkl}},$$

де d_{hkl} – міжплощинна відстань, можна записати основну формулу електроннографії:

$$2rd_{hkl} = 2L\lambda. \quad (1.3)$$

У формулі (1.3) введено множник 2 для зручності вимірювання на практиці $2r$.

При даній величині прискорювального потенціалу величина $2L\lambda$ є константою приладу. Оскільки довжина хвилі в електроннографії строго не визначена (залежить від прискорювального потенціалу, його стабілізації), то у випадку точних вимірів $2L\lambda$ обчислюється для кожної електроннограми. Це можливо завдяки використанню еталонної речовини з відомим набором міжплощинних відстаней d_{hkl} . Для прикладу як еталони розглянемо речовини: NaCl, Mg, NH₄Cl. На рис. 1.2 наведена електроннограма еталона NaCl.

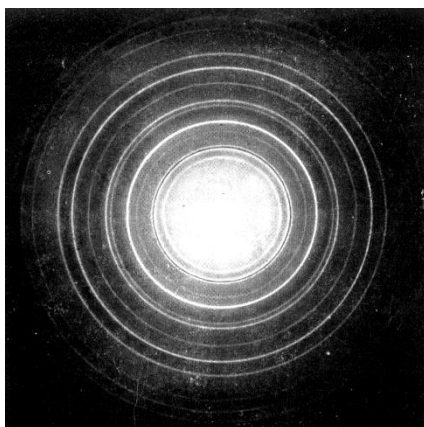


Рисунок 1.2 – Електроннограма від еталона NaCl

Залежно від характеру зйомки розрізняють два види електроннограм - електроннограми на проходження – коли використовувався об'єкт просвічується електронним променем та електроннограми на відбиття – коли електронний промінь майже ковзає уздовж досліджуваної поверхні зразка. У тому й іншому випадку в положення, що відбивається, попадають атомні площини майже паралельні падаючому пучку електронів.

Розрізняють наступні типи електроннограм [2].

1. Електроннограми полікристала - концентричні кільця виходять від безладно розташованих на підкладці кристаликів (рис. 1.3).

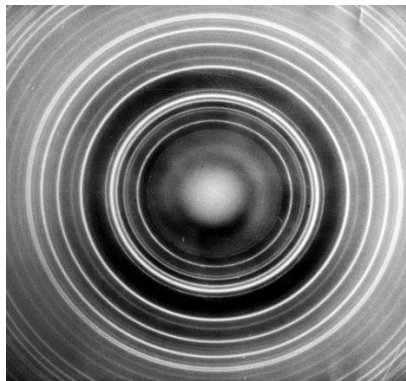


Рисунок 1.3 – Електроннограма полікристала

2. Точкові електроннограми з рефlekсами у вигляді плям. Виходять від монокристалів або монокристалічних зростків (мозаїка) з кутовим розкидом блоків порядку $2 - 3^\circ$ (рис. 1.4).

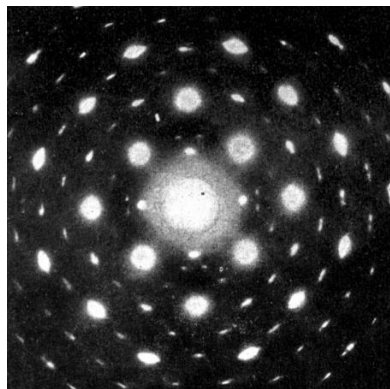


Рисунок 1.4 – Електроннограма від мозаїчного монокристалу

3. Електронограми від текстур з рефlekсами у вигляді кілець або дуг; виходять від закономірно орієнтованих на підкладці кристаликів, у яких певна грань паралельна підкладці, але самі вони безладно розподілені по азимуті (рис. 1.5).

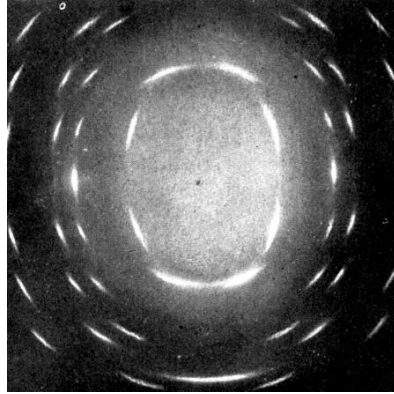


Рисунок 1.5 – Електронограма від текстури

4. Електронограми з Кікучі-лініями виходять від зроблених монокристалів з більшим розміром блоків і малою розорієнтацією (рис. 1.6).

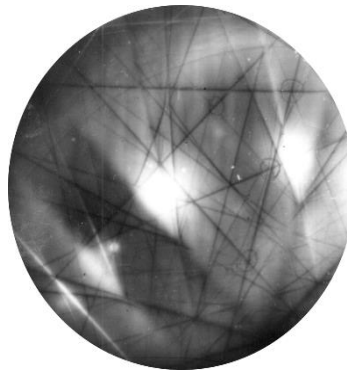


Рисунок 1.6 – Електронограми з Кікучі- лініями

Приведемо приклад ручної розшифровки електронограм від полікристалу. З огляду на, те що електронограма - плоский перетин зворотної решітки, необхідно з'ясувати який геометричний образ буде мати зворотна решітка полікристала. Він буде визначатися характером упорядкованості

розташування кристалів у зразку, і оскільки полікристал представляє агрегат безладно орієнтованих кристаликів, його зворотна решітка, у результаті обертання зворотної решітки монокристалу (тривимірна періодична система крапок - вузлів hkl) навколо вузла (000) у всіх напрямках, представляє систему концентричних сфер, вкладених друг у друга. Центр сфер - вузол (000) (рис. 1.7).

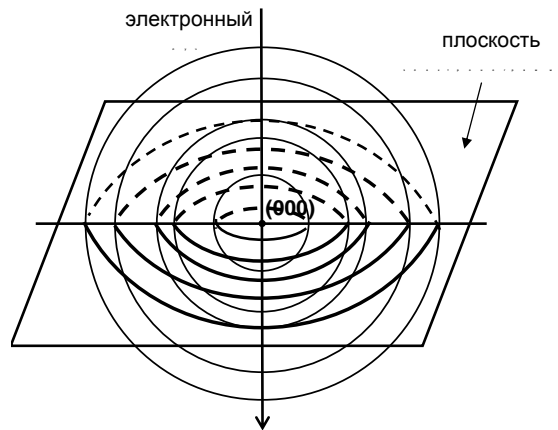


Рисунок. 1.7 – Утворення електронограм від полікристалу

Електронограма – перетин системи сфер площиною, що проходить через вузол (000) і перпендикулярно падаючому пучку, тобто система концентричних кіл. При зміні кута нахилу зразка до пучка електронограма не змінюється. Кожне кільце електронограми відповідає певному вектору зворотної решітки H_{hkl} , що губить всі ознаки просторового розташування щодо інших таких же векторів. Таким чином, електронограма полікристалу характеризується набором H_{hkl} , тобто набором міжплосинних відстаней $\{d_{hkl}\}$, властивим даній кристалічній решітці.

Електронограми полікристала застосовуються як у структурному аналізі (визначення атомної структури невідомих кристалів), так й у фізичному матеріалознавстві. Особливо інформативним є їх застосування для тонких плівок.

Перелічимо деякі завдання, для вирішення яких використовуються електронограми полікристала [2].

1. Ідентифікація речовини, фазовий аналіз. Вирішення цього завдання засноване на тому, що кожній речовині, фазі властивий свій набір міжплощинних відстаней.

Таблиці міжплощинних відстаней великої кількості речовин наведені в довіднику по рентгеноструктурному аналізі полікристала М.И. Миркіна.

2. Вимірюючи напівширину ліній і проводячи її аналіз, можна визначити розміри кристаліків, тобто вивчати субструктуру.

3. У випадку кристалів високої симетрії - кубічних кристалів - по електроннограмам можна визначати тип решітки Браве.

Електроннограми з Кікучі-лініями утворені звичайно парами ліній, темних й світлих, причому світла лінія звичайно лежить ближче до центра (рис. 1.5).

Поява на електроннограмах Кікучі-ліній свідчить про досконалість монокристалічного зразка. Розсіювання здобуває динамічний характер. Це проявляється в ослабленні центрального пучка. Розглянемо пояснення утворення Кікучі - ліній, засноване на елементарному розгляді динамічного розсіювання. Внаслідок сильної взаємодії електронів з речовиною створюються умови для дуже сильного розсіювання падаючого пучка електронів. Значна частина електронів у результаті багаторазових не пружних зіткнень із атомами решітки гублять енергію на виникнення електромагнітного випромінювання атомів, іонізацію й т.д. Ці електрони утворюють тло, що відрізняється майже рівномірною інтенсивністю на пластинці. Однак, деяка частина падаючих електронів взаємодіє з речовиною без істотного обміну енергією, значно змінюючи первісний напрямок. Такі розсіяні електронні пучки можуть надалі в результаті правильного Вульфа - Бреґґівського відбиття знову вже закономірно змінити свій напрямок на тлі не пружно розсіяних електронів, тобто створити відповідно знижену й підвищену інтенсивність.

На рис. 1.8 наведена схема такої взаємодії електронного пучка з речовиною [2].

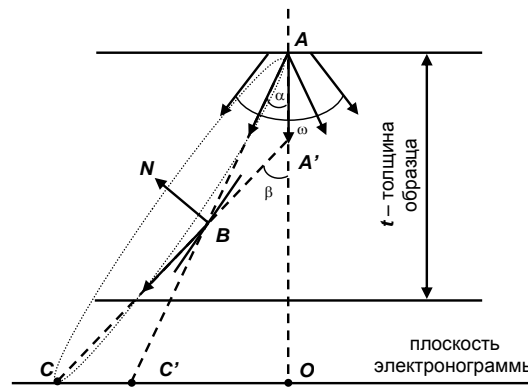


Рисунок 1.8 – Схема взаємодії електронного пучка з речовиною

На рис. 1.8 прийнято такі позначення:

АТ - напрямок падаючого пучка;

ω - кутовий інтервал пружного розсіювання електронів, які на електронограмі утворюють рівномірне тло;

АВ - електронний промінь, який опинився під кутом Вульфа-Брегга стосовно площини (hkl);

ВР - відбитий промінь.

У точках С и С' утворюється відповідно підвищена й знижена інтенсивність тла. Загалом кажучи, відбиті промені СВ розташовані по конусу, вісь якого перпендикулярна площини (hkl). Площина електронограми перетне конус відбитих променів по гіперболічній кривій малої кривизни, майже прямої. Тому на електронограмі спостерігають прямі лінії світлі С' ближче до центру (т. ПРО) і темні С. У кутовому інтервалі ω можуть виявитися промені А'В, які також як й АВ падають до площини (hkl) під кутом Вульфа - Брегга. Тоді відбитий промінь ВС', що відповідає падаючому А'В компенсує недолік електронів у т. С'. Однак цього не відбувається, тому що інтенсивність променю А'В менше АВ, тому що кут $\alpha < \beta$, а інтенсивність пружно розсіяних електронів швидко падає зі збільшенням кута розсіювання.

Як видно з схеми, наведеної на рис. 1.8 кожна пара світла - темна Дікучі-лінії відповідає певній площині (hkl). Дослідивши лінії на електроннограмах, можна визначити орієнтацію зразка по відношенню падаючого пучка. Причому точність у визначенні орієнтації перевищує точність при використанні крапкових електроннограм. Електроннограми з Кікучі-лініями використовуються для контролю за ступенем кристалічної досконалості поверхні. В електронній мікроскопії електроннограми з Кікучі - лініями використовуються для [2]:

1. Калібрування кута нахилу гоніометра.
2. Визначення дифракційних умов контрасту, зокрема, визначення знака параметра \mathbf{g} , що характеризує відхилення від точного виконання Вульфа – Бреґґівського відбиття.

Широке застосування електроннограми з Кікучі-лініями знаходять у мікроелектроніці для визначення ступеня кристалічної досконалості тонких плівок.

1.2 Аналіз методів розпізнавання 2D зображень

Розпізнавання образів — науково-технічний напрямок, пов'язаний з розробкою методів і побудовою систем на базі ЕОМ для встановлення належності деякого об'єкта (предмета, процесу, явища, ситуації, сигналу) до одного із заздалегідь виділених класів об'єктів (образу). Процес розпізнавання базується на співставленні ознак, характеристик досліджуваного об'єкта з ознаками, характеристиками інших відомих об'єктів, у результаті чого робиться висновок про найбільш правдоподібну їхню відповідність.

Серед методів розпізнавання образів можна виділити статистичні методи розпізнавання образів, що ґрунтуються, головним чином, на байєсівському вирішальному правилі [3 – 5]. Суть цього правила полягає у порівнянні обчислених в процесі навчання системи керування апостеріорних імовірностей $p(\mu_m / \gamma_l)$, де μ_m – подія, що відображає дійсну належність

реалізації образу класу X_m^o ; γ_l – гіпотеза про належність реалізації класу розпізнавання X_m^o , $m, l = \overline{1, M}$. Саме завдяки статистичному підходу відбувся стрімкий розвиток методів і алгоритмів машинного навчання, головною задачею якого був набір статистичних даних з метою формування апіорних імовірностей. Обчислення апостеріорних імовірностей за апіорними здійснюється за відомою формулою Байеса:

$$p(\mu_m / \gamma_l) = \frac{p(\mu_m) p(\gamma_l / \mu_m)}{p(\gamma_l)}, \quad (1.5)$$

де $p(\gamma_l / \mu_m)$ – умовна апіорна ймовірність прийняття гіпотези γ_l за умови існування основної гіпотези μ_m ;

$p(\gamma_l)$ – повна ймовірність прийняття гіпотези, яка визначається за теоремою про повну ймовірність як

$$p(\gamma_l) = \sum_{m=1}^M p(\mu_m) p(\gamma_l / \mu_m).$$

За байесівським вирішальним правилом гіпотезу γ_l належить віднести до класу X_m^o , якщо апостеріорна ймовірність для цього класу максимальна. Але оскільки оцінка апостеріорних імовірностей здійснюється за вибіркоvim методом математичної статистики, то це обумовлює ризик помилки особливо у тих випадках, коли центри розсіювання реалізацій образів знаходяться близько один до одного. Тому на практиці приймається рішення за умови мінімізації функції умовного ризику:

$$R(a_i / \gamma_l) = \sum_{m=1}^M \lambda(a_i / \mu_m) p(\mu_m / \gamma_l), \quad (1.6)$$

де $\lambda(a_i / \mu_m)$ – функція втрат, яка визначає втрати a_i , обумовлені прийняттям

гіпотези γ_1 за умови, що реалізація образу, яка розпізнається, належить класу X_m^o .

Побудова розбиття \mathfrak{X} простору ознак на класи розпізнавання за байєсівським правилом здійснюється шляхом обчислення класифікатором роздільних функцій і вибору гіпотези γ_1 , для якої роздільна функція є найбільшою. Застосування класифікатора Байєса на практиці є ускладненим, оскільки вигляд функцій щільності ймовірностей, як правило, невідомий. Тому з метою використання класифікатора Байєса розвинулося два підходи: параметричний, при якому функції щільності подаються в аналітичному вигляді, що не завжди можливо, і непараметричний підхід, при якому непотрібне апріорне знання вигляду функції щільності ймовірностей.

Відсутність на практиці повної апріорної інформації про стан об'єкту керування, є основною причиною низької достовірності розпізнавання за статистичними методами. Найбільш суттєві результати за статистичними методами в області контролю та керування отримано в тих галузях науки і техніки, де є можливість набору репрезентативної статистики при забезпеченні статистичної однорідності та стійкості, які витікають з центральної граничної теореми теорії ймовірностей. Саме при виконанні умов цієї теореми є можливість застосування добре розробленого апарату теорії статистичних рішень [3]. Суть усіх статистичних методів навчання полягає в знаходженні роздільної функції, спосіб побудови якої задає спосіб розбиття простору на класи розпізнавання. Задача навчання трактується як вибір серед можливих вирішальних правил такого, яке мінімізує середній ризик помилкового розпізнавання. З метою збільшення оперативності статистичних алгоритмів класифікації логічним є застосування алгоритмів мінімізації емпіричного середнього ризику, в яких функція емпіричного ризику обчислюється за випадковою та незалежною вибіркою малого обсягу.

Але ефективність розв'язання задачі навчання шляхом мінімізації функції емпіричного ризику, як це показано в праці [2], на практиці прямо залежить від умов існування рівномірної збіжності статистичної похибки до заданого значення в класі подій.

У статистичних методах розпізнавання етап машинного навчання характеризується значною обчислювальною трудомісткістю та недостатньою оперативністю, обумовленою необхідністю виконання умов рівномірної збіжності емпіричних частот до відповідних імовірностей. Облік специфіки окремих задач розпізнавання дозволяє підвищити оперативність етапу навчання. Але в цілому оперативність розв'язання задачі машинного навчання як на основі градієнтних методів, так і інших відомих модифікацій методу стохастичної апроксимації коливається в значних межах. Найбільш суттєві результати аналізу та синтезу складних систем за статистичними методами отримано у тих галузях науки і техніки, де є можливість набору репрезентативної статистики за умови забезпечення статистичних однорідності та стійкості експериментальних даних. При цьому зміст усіх теоретико-статистичних методів навчання полягає у знаходженні роздільної функції, спосіб побудови якої задає спосіб розбиття простору ознак на класи розпізнавання. Основною заслугою статистичного підходу до розпізнавання образів є започаткування розвитку теорії машинного навчання. При цьому модельність статистичних методів автоматичної класифікації так само становить певну методологічну цінність, оскільки дозволяє дослідити механізм прийняття рішень.

У теперішній час одним із популярних напрямків розпізнавання зображень є застосування штучних нейронних мереж, які відносяться до класу структурних методів розпізнавання [6 – 8]. Найбільш поширеними штучними нейронними мережами (ШНМ), які використовуються для аналізу візуальних зображень є згорткові нейронні мережі [6]. Такі ШНМ використовують відносно невелику попередню обробку в порівнянні з іншими алгоритмами класифікації зображень. Це значить що мережа вивчає

ознаки, які в традиційних алгоритмах були створені власноруч. Дана незалежність від попередніх значень і людських зусиль при розробці функцій є їх перевагою.

Згорткова нейронна мережа складається із вхідного і вихідного шарів, а також декількох прихованих шарів. Приховані шари CNN зазвичай складаються з декількох згорткових шарів, які використовують з множенням або іншими скалярними операціями. Функція активації зазвичай являється шаром Rectifier, за яким слідує додаткові згортки, такі як шари об'єднання, повністю зв'язані шари та шари нормалізації, які називають прихованими шарами, оскільки їх входи та виходи маскуються функцією активації і кінцевою згорткою.

Алгоритм розпізнавання образів навчання з першого разу (One-Shot learning) – алгоритм категоризації об'єктів, який зустрічається в основному в комп'ютерному зорі [9]. Оскільки для більшості алгоритмів категоризації об'єктів на основі машинного навчання потребують сотні або тисячі зображень, то в працях [40] запропоновано алгоритми так званого одноразового машинного навчання за одним або декількома зображеннями. При цьому ознаки зображення повинні бути зменшені до необхідного мінімуму, але без втрати інформації. Кожний піксель рецепторного поля містить декілька біт інформації про колір. Для отримання вектору зображення відбувається декілька стадій обробки:

- перетворення в монохром за формулою $(Red+Green+Blue)/3$;
- визначається градієнт для кожної точки матриці;
- знаходиться найбільш важливий в ваговому відношенні область градієнта;
- проводиться пошук ланцюгів векторів які покривають найбільш важливі області градієнта;
- потім відбувається зациклювання кроків для отримання мінімальної кількості векторів які несуть в собі максимальну інформацію.

Зазначений алгоритм характеризується високою оперативністю машинного навчання, але має невисоку достовірність через довільні умови формування зображень і притаманні нейроподібним структурам недоліки:

– чутливість до багатовимірності словника ознак і алфавіту класів розпізнавання;

– негнучкість, що ускладнює перенавчання системи розпізнавання при розширенні алфавіту класів розпізнавання

Таким чином, розпізнавання зображень є однією з важливих наукових проблем, вирішення якої має практичне значення в багатьох галузях соціально-економічної сфери українського суспільства. Все більше і більше з'являється як наукових, так і промислових комплексів, які потребують розпізнавання зображень із високою точністю, що в свою чергу потребує створення і реалізації машинних алгоритмів розпізнавання зображень. Однією із важливих науково-технічних задач є автоматизація розпізнавання електронограм, яка має практичне значення в металургії, виготовлення кристалів та інше. Вирішення такої задачі дасть можливість швидше приймати рішення, зробить технологічний процес менш дорогим та трудомістким. На сучасному рівні розвитку теорії розпізнавання образів машинна класифікація електронограм все ще залишається нерозв'язаною задачею через такі основні причини:

- нестационарність за яскравістю зображень електронограм;
- велика потужність алфавіту класів електронограм;
- велика ступінь перетину класів розпізнавання в просторі ознак;
- вплив неконтрольованих збурюючих факторів.

Більшість відомих алгоритмів розпізнавання зображень орієнтована на розв'язання модельних задач, які виключають перетин класів розпізнавання і потребують статистичних стійкості та однорідності навчальної матриці, що на практиці, як правило, не виконується. Шляхом вирішення цієї проблеми є розв'язання задачі в рамках інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології) аналізу даних, яка ґрунтується на прямій оцінці

інформаційної спроможності системи розпізнавання в процесі машинного навчання [10, 11].

1.3 Формалізована постановка задачі інформаційного синтезу системи розпізнавання електронограм

Розглянемо в рамках ІЕІ-технології формалізовану постановку задачі інформаційного синтезу здатної навчатися системи розпізнавання електронограм. Нехай сформовано алфавіт $\{X_m^o \mid m = \overline{1, M}\}$ класів розпізнавання, які характеризують M електронограм різних матеріалів. Для кожного класу розпізнавання сформовано тривимірну навчальну матрицю $\|y_{m,i}^{(j)}\|$ яскравості пікселів рецепторного поля цифрового зображення електронограми, в якій рядок $\{y_{m,i}^{(j)} \mid i = \overline{1, N}\}$, де N – кількість ознак розпізнавання, є структурованим вектором ознак відповідного класу розпізнавання (далі в тексті просто реалізація), а стовпчик матриці – випадкова навчальна вибірка $\{y_{m,i}^{(j)} \mid j = \overline{1, n}\}$ з обсягом n .

Відомо, що концепція ІЕІ-технології полягає в перетворенні вхідної навчальної матриці Y в робочу бінарну матрицю X , яка змінюється в процесі машинного навчання. Тому для бінарного простору Хеммінга задано вектор параметрів функціонування, які впливають на функціональну ефективність машинного навчання БСР розпізнавати реалізації класу розпізнавання X_m^o :

$$g_m = \langle x_m, d_m, \delta \rangle, \quad (1.7)$$

де x_m – усереднений за ансамблем реалізацій вектор, вершина якого визначає центр гіперсферичного контейнера класу розпізнавання X_m^o ; d_m – радіус гіперсферичного контейнера класу розпізнавання X_m^o , який відновлюються в радіальному базисі простору ознак розпізнавання;

δ – параметр, величина якого дорівнює половині симетричного поля контрольних допусків на ознаки розпізнавання, якими є значення яскравості в пікселях.

На параметри функціонування системи, які будемо далі називати параметрами машинного навчання, накладаються відповідні обмеження:

- область значень яскравості пікселів знаходиться в інтервалі $[0; 255]$ градацій яскравості;
- область значень радіуса контейнера класу розпізнавання X_m^o задається нерівністю

$$d_m < d(x_m \oplus x_c),$$

де $d(x_m \oplus x_c)$ – міжцентрова відстань між реалізацією x_m і найближчою реалізацією x_c сусіднього класу X_c^o ;

- область значень параметра δ задається нерівністю

$$\delta < \delta_H / 2,$$

де δ_H – нормоване поле допусків на ознаки розпізнавання.

На рис. 1.9 показано двобічне симетричне поле контрольних допусків на y_i -ту ознаку розпізнавання.

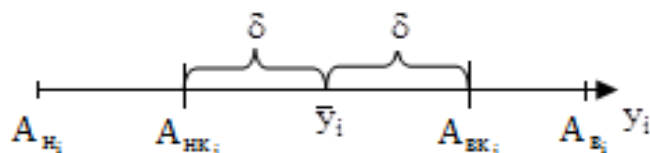


Рисунок 1.9 – Поле допусків на ознаку розпізнавання

На рис. 1.9 прийнято такі позначення:

\bar{y}_i – номінальне (усереднене) значення ознаки y_i ;

$A_{H,i}, A_{B,i}$ – нижній і верхній нормовані допуски на ознаку y_i ;

$A_{HK,i}, A_{BK,i}$ – нижній та верхній контрольні допуски на ознаку y_i ;

δ – параметр, який дорівнює половині симетричного поля контрольних допусків.

Необхідно в процесі машинного навчання системи розпізнавання оптимізувати параметри машинного навчання (1.7), які забезпечують максимальне значення інформаційного критерію оптимізації в робочій (допустимій) області визначення його функції:

$$\bar{E}^* = \frac{1}{M} \sum_{m=1}^M \max_{G_E \cap \{k\}} E_m^{(k)}, \quad (1.8)$$

де $E_m^{(k)}$ – значення інформаційного критерію навчання системи розпізнавати реалізації класу X_m^o , обчислене на k -му кроці машинного навчання;

G_E – робоча область обчислення інформаційного критерію;

$\{k\}$ – множина кроків машинного навчання.

При функціонуванні системи розпізнавання в режимі екзамену необхідно перевірити функціональну ефективність машинного навчання.

Для розв'язання поставленої задачі необхідно виконати такі завдання:

- сформулювати вхідний математичний опис системи розпізнавання електронограм;
- розробити математичні моделі функціонування системи розпізнавання електронограм в режимах машинного навчання та екзамену;
- розробити алгоритми машинного навчання системи розпізнавання електронограм з оптимізацією системи контрольних допусків на ознаки розпізнавання;
- за результатами машинного навчання побудувати вирішальні правила;

- розробити алгоритм функціонування системи в режимі екзамену і оцінити його функціональну ефективність.

Таким чином, задача інформаційного синтезу здатної навчатися системи розпізнавання полягає в оптимізації параметрів її машинного навчання шляхом наближення глобального максимуму інформаційного критерію (1.8) до його максимального граничного значення.

2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

2.1 Основні положення, принципи та визначення інформаційно-екстремальної інтелектуальної технології аналізу даних

Одним з перспективних підходів до аналізу та синтезу системи розпізнавання зображень є використання ідей та методів інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології), що ґрунтується на максимізації інформаційної спроможності системи в процесі її машинного навчання [10, 11]. Перевага ІЕІ-технології полягає на відміну від методів, побудованих на дистанційних мірах, в застосуванні як міри схожості інформаційних критеріїв оптимізації. Крім того, відмінність інформаційно-екстремального машинного навчання полягає в трансформації вхідної навчальної матриці, сформованої в евклідовому просторі ознак розпізнавання, в робочу бінарну навчальну матрицю, що дозволяє шляхом її цілеспрямованої зміни адаптувати вхідний математичний опис системи розпізнавання до високо достовірних вирішальних правил.

Основна ідея машинного навчання системи розпізнавання у рамках ІЕІ-технології згідно з працями [10, 11] полягає в побудові в рамках геометричного підходу високо достовірних вирішальних правил шляхом оптимізації параметрів машинного навчання системи. При цьому здійснюється цілеспрямовано пошук глобального максимуму функції статистичного інформаційного критерію в робочій області її визначення в процесі відновлення в радіальному базисі бінарного простору ознак контейнерів класів розпізнавання. Шляхом оптимізації контрольних допусків на ознаки розпізнавання здійснюється перетворення вхідної навчальної матриці в робочу бінарну матрицю, яка в процесі машинного навчання змінюється з метою адаптації вхідного математичного опису до максимальної достовірності класифікаційних рішень.

Побудова високо достовірних вирішальних правил в рамках ІЕІ-технології здійснюється за багатоциклічною процедурою пошуку

максимального граничного, усередненого за алфавітом $\{X_m^o \mid m = \overline{1, M}\}$ інформаційного критерію оптимізації параметрів машинного навчання [10]:

$$g_\xi^* = \arg \max_{G_\xi} \{ \max_{G_{\xi-1}} \{ \dots \{ \max_{G_1 \cap G_E} \frac{1}{M} \sum_{m=1}^M E_m \} \dots \} \}, \quad (2.1)$$

де E_m – інформаційний критерій оптимізації параметрів машинного навчання;

G_ξ – допустима область значень ξ -го параметра машинного навчання;

G_E – робоча (допустима) область визначення функції інформаційного критерію.

У процедурі (2.1) внутрішній цикл реалізує так званий базовий алгоритм, призначенням якого є:

- 1) обчислення на кожному кроці машинного навчання інформаційного критерію і пошук глобального максимуму його функції;
- 2) визначення оптимальних геометричних параметрів контейнерів класів розпізнавання, які відновлюються в радіальному базисі простору ознак.

Враховуючи, що розбиття $\tilde{\mathfrak{R}}^{|M|}$ бінарного простору ознак Хеммінга на класи розпізнавання є нечітким, то на нього накладаються такі обмеження [10]:

- 1) $(\forall X_m^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq \emptyset]$;
- 2) $(\exists X_k^o \in \tilde{\mathfrak{R}}^{|M|}) (\exists X_l^o \in \tilde{\mathfrak{R}}^{|M|}) [X_k^o \neq X_l^o \rightarrow X_k^o \cap X_l^o \neq \emptyset]$;
- 3) $(\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|}) (\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|}) [X_k^o \neq X_l^o \rightarrow Ker X_k^o \cap Ker X_l^o \neq \emptyset]$;
- 4) $\cup_{X_m^o \in \mathfrak{R}} X_m^o \subseteq \Omega_B; k \neq l; k, l, m = \overline{1, M}$. (2.2)

Оскільки, у бінарному просторі контейнер класу розпізнавання представлений у формі гіперпаралелепіеду, то для його представлення в

радіальному базисі можна використати гіперсферу, що включає усі вершини гіперпаралелепіеду. При цьому усереднений структурований вектор ознак $x_m \in X_m^0$ визначає центр гіперсферичного контейнера класу розпізнавання, а d_m – радіус контейнера, який розраховується за формулою для знаходження відстані Хеммінга:

$$d_m = d(x_m \oplus \lambda) = \sum_{i=0}^N (x_{m,i} \oplus \lambda_i),$$

де $x_{m,i}$ – i -та координата еталонного вектора x_m ;

λ_i – i -та координата вектора λ , вершина якого належить контейнеру класу розпізнавання .

У радіальному базисі простору ознак відновлення оптимального контейнера класу розпізнавання X_m^o відбувається шляхом перетворення його контейнера в гіперсферичний габарит, радіус якого змінюється на кожному кроці машинного навчання, наприклад, за процедурою

$$d_m(k) = [d_m(k-1) + h \mid d_m(k) \in G_m^d],$$

де k – крок збільшення радіуса;

h – величина збільшення радіуса контейнера;

G_m^d – область допустимих значень радіуса d_m контейнера.

Припустимо, що класи X_k^0 і X_l^0 – найближчі сусіди, тобто мають мінімальну міжцентрову відстань $d(x_k \oplus x_l)$, де x_k та x_l – усередненні вектори ознак розпізнавання цих класів відповідно. Для виключення ситуації, коли центр одного класу потрапляє у межі контейнера іншого, умови (2.2) доповнюються виразом [10]

$$\begin{aligned} (\forall X_k^0 \in \tilde{\mathfrak{R}}^{|M|}) (\forall X_l^0 \in \tilde{\mathfrak{R}}^{|M|}) [X_k^0 \neq X_l^0 \rightarrow (d_k^* < d(x_k \oplus x_l)) \& \\ \& (d_l^* < d(x_k \oplus x_l))], \end{aligned} \quad (2.3)$$

де d_k^* та d_l^* – оптимальні радіуси контейнерів класів розпізнавання X_k^0 і X_l^0 відповідно.

Таким чином, згідно з процедурою (2.1) інформаційно-екстремальне машинне навчання полягає в ітераційній оптимізації параметрів функціонування системи розпізнавання зображень, в процесі якої здійснюється пошуку глобального екстремуму усередненого значення критерію (1.8).

2.2 Базовий алгоритм інформаційно-екстремального машинного навчання системи розпізнавання

Базовий інформаційно-екстремальний алгоритм оптимізації параметрів машинного навчання системи розпізнавання реалізується у внутрішньому циклі структурованої ітераційної процедури машинного навчання (2.1), що й обумовило його назву. Призначенням базового алгоритму навчання є:

- оптимізація геометричних параметрів контейнерів класів розпізнавання;
- обчислення інформаційного критерію оптимізації параметрів машинного навчання системи;
- пошук глобального максимуму інформаційного критерію у робочій (допустимій) області визначення його функції.

Категорійну модель інформаційно-екстремального машинного навчання системи розпізнавання за базовим алгоритмом представимо у вигляді орієнтованого графу відображення операторами одна на одну відповідних множин, які застосовуються в процесі машинного навчання. Вхідний математичний опис здатної навчатися системи розпізнавання подамо у вигляді структури [10]

$$I_{\text{ex}} = \langle G, T, \Omega, Z, Y, X; f_1, f_2 \rangle, \quad (2.4)$$

де G – простір вхідних факторів, які визначають умови виконання експерименту;

T – множина моментів часу зняття інформації;

Ω – простір ознак розпізнавання;

Z – простір можливих станів, який формує алфавіт класів розпізнавання;

Y – вхідна навчальна матриця, яка відбиває властивості об'єкту дослідження;

X – робоча бінарна навчальна матриця, яка в процесі мінімаційно-екстремального машинного навчання адаптується до максимальної повної ймовірності прийняття правильних діагностичних рішень;

f_1 – оператор формування вхідної навчальної матриці Y ;

f_2 – оператор формування робочої бінарної навчальної матриці X .

Категорійну модель інформаційно-екстремального машинного навчання системи розпізнавання за базовим алгоритмом показано на рис. 2.1 [10].

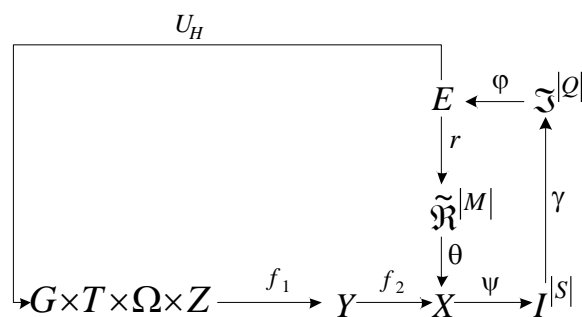


Рисунок 2.1 – Категорійна модель машинного навчання за базовим алгоритмом

На рис. 2.1 терм-множина E , елементами якої є обчислені на кожному кроці машинного навчання значення інформаційного критерію, згідно з принципом повної композиції є загальною для всіх контурів оптимізації параметрів навчання. Оператор $r: E \rightarrow \tilde{\mathfrak{R}}^{|M|}$ на кожному кроці машинного

навчання відновлює в радіальному базисі бінарного простору діагностичних ознак контейнери класів розпізнавання, які утворюють розбиття $\tilde{\mathfrak{R}}^{|M|}$. Оператор θ відображає розбиття $\tilde{\mathfrak{R}}^{|M|}$ на нечіткий розподіл апріорно класифікованих двійкових реалізацій класів розпізнавання. Далі оператор $\psi: X \rightarrow I^{|S|}$, де $I^{|S|}$ – множина гіпотез, перевіряє основну статистичну гіпотезу $\gamma_1: x_m^{(j)} \in X_m^o$. Оператор γ визначає множину точнісних характеристик $\mathfrak{Z}^{|Q|}$, де $Q = S^2$, а оператор ϕ обчислює множину значень E інформаційного критерію оптимізації, якій є функціоналом від точнісних характеристик. Контур оптимізації контрольних допусків замикається через терм-множину D елементами якої є значення контрольних допусків на діагностичні ознаки. Оператор u регламентує процес машинного навчання.

Вхідною інформацією для навчання за базовим алгоритмом є дійсний в загальному випадку тривимірний масив реалізацій класів розпізнавання $\{y_{m,i}^{(j)} \mid m = \overline{1, M}; i = \overline{1, N}; j = \overline{1, n}\}$; значення параметра поля контрольних допусків δ на ознаки розпізнавання і рівні квантування $\{\rho_m\}$ координат усереднених двійкових векторів-реалізацій, які за замовчуванням дорівнюють 0,5 для всіх класів розпізнавання.

Таким чином, базовий алгоритм навчання є ітераційною процедурою пошуку глобального максимуму інформаційного критерію оптимізації параметрів машинного навчання в робочій області визначення його функції:

$$d_m^* = \arg \max_{G_E \cap \{d\}} E_m^*(d).$$

На рис. 2.2 на прикладі побудови оптимального гіперсферичного контейнера класу розпізнавання X_1^o показано геометричну інтерпретацію реалізації базового алгоритму інформаційно-екстремального машинного навчання системи розпізнавання.

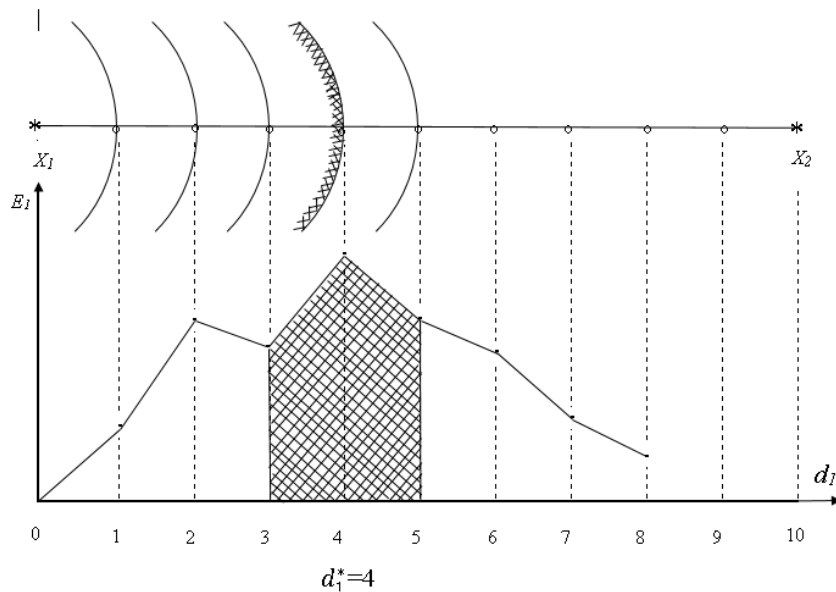


Рисунок 2.2 – Схема геометричної інтерпретації реалізації базового інформаційно-екстремального алгоритму машинного навчання

На рис.2.2 показано в бінарному просторі Хеммінга вершини усередненого двійкового вектора-реалізації x_1 класу розпізнавання X_1^o і усередненого двійкового вектора-реалізації x_2 класу розпізнавання X_2^o , який є найближчим сусідом для класу розпізнавання X_1^o . На схемі кодова відстань Хеммінга між векторами x_1 і x_2 дорівнює $d = (x_1 \oplus x_2) = 11$. У процесі машинного навчання за базовим алгоритмом здійснюється покрокове збільшення радіусу d_1 контейнера класу розпізнавання X_1^o на одну кодову одиницю. При цьому згідно з умовою (6.7) на величину радіусу d_1 накладається обмеження

$$d_1 < d(x_1 \oplus x_2) - 1.$$

На кожному кроці навчання обчислюється значення інформаційного критерію E_1 оптимізації радіусу контейнера класу розпізнавання X_1^o . Як оптимальний радіус приймається його екстремальне значення глобального

максимуму критерію E_1 , обчислене в робочій (допустимій) області визначення функції інформаційного критерію. На рис. 2.2 робоча область обчислення інформаційного критерію позначена заштрихованою ділянкою. Для двохальтернативних рішень робоча область існує за умови, що їх перша і друга достовірності перевищують відповідно помилки першого та другого роду. Аналіз рис. 2.2 показує, що максимальне значення інформаційного критерію E_1^* отримано на четвертому кроці машинного навчання, тобто оптимальний радіус контейнера класу розпізнавання X_1^o дорівнює $d_1^* = 4$ (тут і далі в кодових одиницях).

Таким чином, основною функцією базового алгоритму машинного навчання у рамках ІЕІ-технології є обчислення на кожному кроці навчання інформаційного критерію і організація пошуку його глобального максимуму в робочій області визначення функції критерію з метою визначення оптимальних геометричних параметрів розбиття простору ознак на класи розпізнавання. При гіперсферичному контейнері класів розпізнавання такими параметрами при інформаційно-екстремальному машинному навчанні за базовим алгоритмом є оптимальні кодові відстані $\{d_m^*\}$ і оптимальні усередненні вектори-реалізації $\{x_m^*\}$ для заданого алфавіту $\{X_m^o\}$.

2.3 Інформаційний критерій оптимізації параметрів машинного навчання системи розпізнавання

Розглянемо модифікацію диференціальної інформаційної міри Кульбака, яка подається як добуток відношення правдоподібності Λ на міру відхилень відповідних розподілів імовірностей.

Розглянемо логарифмічне відношення повної ймовірності $P_{t,m}^{(k)}$ правильного прийняття рішень про належність реалізацій класів X_m^o і X_c^o контейнеру $K_{m,k}^o \in X_m^o$ до повної ймовірності помилкового прийняття рішень $P_{f,m}^{(k)}$, яке для двохальтернативної системи оцінок рішень має вигляд [10]

$$\Lambda = \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} = \log_2 \frac{p(\mu_m)p(\gamma_{1,k}/\mu_m) + p(\mu_c)p(\gamma_{2,k}/\mu_c)}{p(\mu_m)p(\gamma_{2,k}/\mu_m) + p(\mu_c)p(\gamma_{1,k}/\mu_c)}, \quad (2.5)$$

де $p(\mu_m)$ – безумовна ймовірність появи реалізації класу X_m^o ;

$p(\mu_c)$ – безумовна ймовірність появи реалізації найближчого (сусіднього)

класу X_c^o ;

$\gamma_{1,k}$ – гіпотеза про належність реалізації класу розпізнавання X_m^o ;

$\gamma_{2,k}$ – альтернативна гіпотеза.

Із урахуванням (1.13) при допущенні згідно із принципом Лапласа-Бернуллі, що $p(\mu_m) = p(\mu_c) = 0,5$, і після переозначення апіорних умовних імовірностей відповідними точнісними характеристиками загальна міра Кульбака остаточно набирає вигляду [10]

$$\begin{aligned} E_{K,m}^{(k)} &= \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} * [P_{t,m}^{(k)} - P_{f,m}^{(k)}] = \\ &= \left| \begin{array}{l} P_{t,m}^{(k)} = 0,5D_{1,m}^{(k)}(d) + 0,5D_{2,m}^{(k)}(d) \\ P_{f,m}^{(k)} = 0,5\alpha_m^{(k)}(d) + 0,5\beta_m^{(k)}(d) \end{array} \right| = 0,5 \log_2 \left(\frac{D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * \\ & * [(D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)) - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))] = \\ & = \log_2 \left(\frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * [1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))]. \quad (2.6) \end{aligned}$$

Нормовану модифікацію критерію (1.14) можна подати у вигляді

$$E_{K,m}^{(k)} = \frac{E_{K,m}^{(k)}}{E_{K,\max}^{(k)}}, \quad (2.7)$$

де $E_{K_{\max}}^{(k)}$ – максимальне значення інформаційного критерію (2.6) при

$$D_{1,m}^{(k)}(d) = D_{2,m}^{(k)}(d) = 1 \quad \text{і} \quad \alpha_m^{(k)}(d) = \beta_m^{(k)}(d) = 0.$$

При оптимізації параметрів функціонування системи розпізнавання у процесі машинного навчання за ІЕІ-технологією нормування критеріїв оптимізації не є обов'язковим, оскільки тут розв'язується задача пошуку екстремальних значень параметрів навчання, які відповідають глобальному максимуму інформаційного критерію у робочій області його визначення. Але нормування критеріїв оптимізації є доцільним при порівняльному аналізі результатів досліджень і при оцінці ступеня близькості реальної системи розпізнавання до потенційної.

Оскільки інформаційний критерій є функціоналом від точнісних характеристик, то при репрезентативному обсязі навчальної вибірки необхідно користуватися їх оцінками [10]:

$$D_{1,m}^{(k)}(d) = \frac{K_{1,m}^{(k)}}{n_{\min}}; \quad \alpha_m^{(k)}(d) = \frac{K_{2,m}^{(k)}}{n_{\min}}; \quad \beta_m^{(k)}(d) = \frac{K_{3,m}^{(k)}}{n_{\min}}; \quad D_{2,m}^{(k)}(d) = \frac{K_{4,m}^{(k)}}{n_{\min}}, \quad (2.8)$$

де $K_{1,m}^{(k)}$ – кількість подій, які правильно визначають належність “своїй”

реалізацій класу розпізнавання X_m^o ;

$K_{2,m}^{(k)}$ – кількість подій, які помилково визначають неналежність “своїй”

реалізацій класу розпізнавання X_m^o ;

$K_{3,m}^{(k)}$ – кількість подій, які помилково визначають належність “чужої”

реалізацій класу розпізнавання X_m^o ;

$K_{4,m}^{(k)}$ – кількість подій, які правильно визначають неналежність “чужої”

реалізацій класу розпізнавання X_m^o ;

n_{\min} – мінімальний обсяг репрезентативної навчальної вибірки.

Після підстановки відповідних позначень (2.8) у вираз (1.4) одержимо робочу формулу для обчислення в рамках ІЕІ-технології ентропійного критерію машинного навчання системи розпізнаванню реалізацій класу X_1^o :

Робоча модифікація критерію Кульбака після відповідної підстановки оцінок (8.57) у вираз (2.6) набирає вигляду [10]

$$E = \frac{1}{n} \log_2 \left\{ \frac{2n + 10^{-r} - [K_2^{(k)} + K_3^{(k)}]}{[K_2^{(k)} + K_3^{(k)}] + 10^{-r}} \right\} \left[n - (K_2^{(k)} + K_3^{(k)}) \right], \quad (2.9)$$

де 10^{-r} – достатньо мале число, яке вводиться для уникнення поділу на нуль.

Розглянемо схему обчислення коефіцієнтів $K_2^{(k)}$ і $K_3^{(k)}$ у формулах (2.6) і (2.5) [10]. На рис. 2.3 показано структуру навчальної матриці при побудові оптимального контейнера для класу X_1^o . Навчальна матриця послідовно складається з векторів реалізацій $\{x_1^{(j)}\} \in X_1^o$ і $\{x_2^{(j)}\} \in X_2^o$ відповідно.

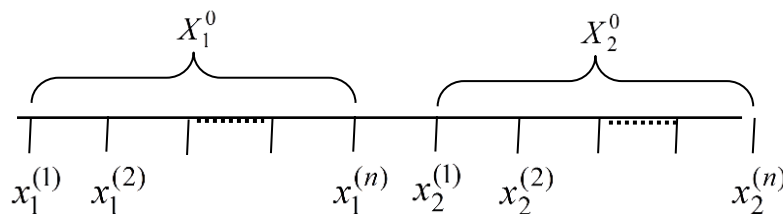


Рисунок 2.3 – Структура навчальної матриці

Алгоритм обчислення коефіцієнтів $K_2^{(k)}$ і $K_3^{(k)}$ має такий предикатний вигляд [10]:

$$(\forall X_1^o \in \mathfrak{R}^{|\Lambda|}) (\forall X_2^o \in \mathfrak{R}^{|\Lambda|}) [\text{if } x_1^{(j)} \in X_1 \text{ then} \\ K_1(j) := K_1(j-1) + 1 \text{ else } K_2(j-1) + 1];$$

$$(\forall X_1^o \in \mathfrak{R}^{|\Lambda|}) (\forall X_2^o \in \mathfrak{R}^{|\Lambda|}) [\text{if } x_2^{(j)} \in X_1 \text{ then} \\ K_3(j) := K_3(j-1) + 1 \text{ else } K_4(j) := K_4(j-1) + 1] .$$

Таким чином, інформаційні критерії (2.6) і (2.7) є функціоналами як від точнісних характеристик рішень, що приймаються, так і від дистанційних критеріїв, тобто їх можна розглядати як узагальнення відомих статистичних і детермінованих (дистанційних) критеріїв оптимізації параметрів машинного навчання системи розпізнавання зображень.

3 ІНФОРМАЦІЙНЕ, АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЕЛЕКТРОНОГРАМ

3.1 Вхідний математичний опис системи розпізнавання електронограм

Реалізацію методу інформаційно-екстремального машинного навчання системи розпізнавання розглянемо на прикладі розпізнавання отриманих на просвічуючому електронному мікроскопі електронограм, показаних на рис.3.1 – рис. 3.4.

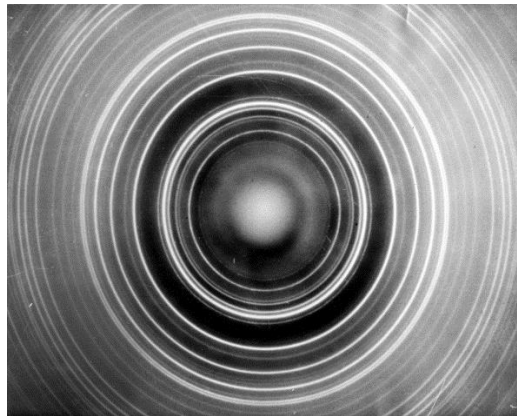


Рисунок 3.1 – Електронограма полікристала

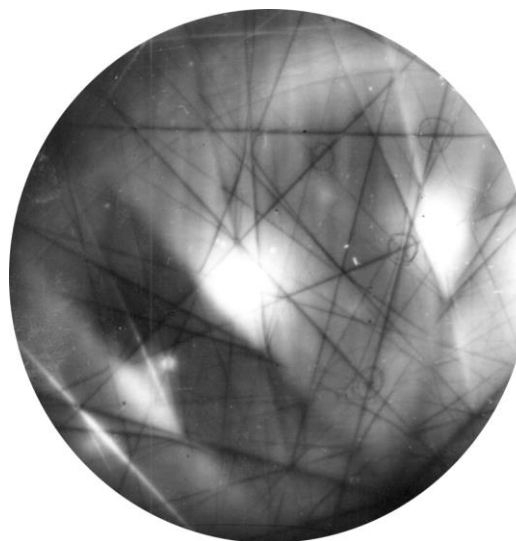


Рисунок 3.2 – Електронограма з Кікучі-лініями

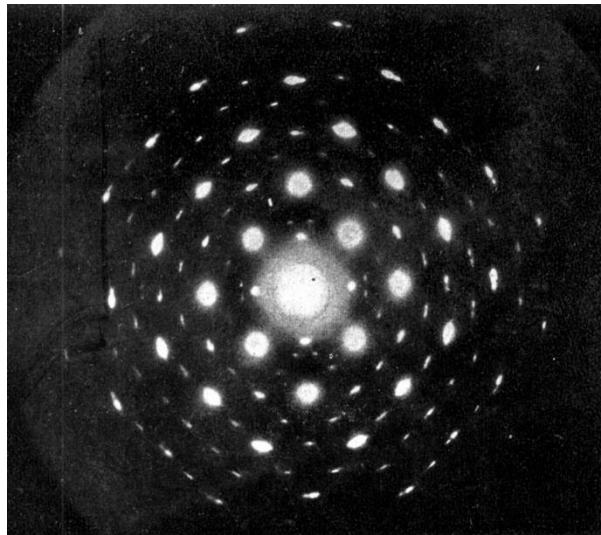


Рисунок 3.3 – Електронограма мозаїчного монокристалу

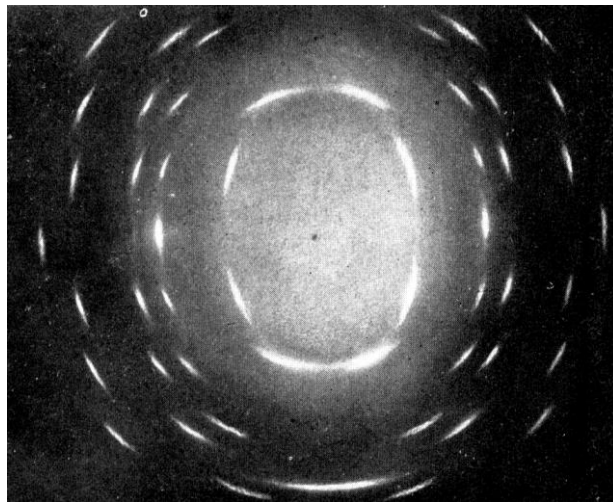


Рисунок 3.4 – Електронограма текстури

З метою забезпечення інваріантності вирішальних правил до зсуву і повороту об'єктів в рецепторному полі, розмір якого дорівнював 200×200 пікселів, здійснювалося формування вхідної навчальної матриці шляхом оброблення цифрового зображення електронограми в полярній системі координат. При цьому усереднена яскравість пікселів кожного кола зчитування, побудованого навколо геометричного центру кадру, обчислювалася за формулою [11]

$$\Theta_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \theta_i, \quad (3.1)$$

де Θ_j – усереднене значення яскравості пікселів, що входять у коло зчитування j -го радіусу, $j = \overline{0, R}$;

θ_i – значення яскравості RGB-складової в i -му пікселі рецепторного поля зображення електронограми;

N_j – загальна кількість пікселів в j -му колі зчитування;

R – радіус кола зчитування.

На рис.3.5 показано схематично оброблення зображення електронограми в полярних координатах за формулою (3.1).

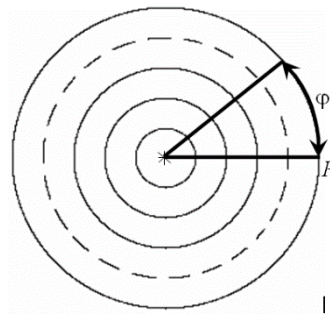


Рисунок 3.5 – Схема оброблення зображення електронограми в полярних координатах

При обробленні зображення об'єкту визначення геометричного центру електронограми здійснювалося за формулою

$$i_c = \frac{S_n}{n} = \left(\frac{a_1 + a_n}{2} \right), \quad (3.2)$$

де S_n – сума арифметичної прогресії для послідовності $\{a_1, \dots, a_n\}$ пікселів рецепторного поля зображення електронограми;

3.2 Категорійні моделі інформаційно-екстремального машинного навчання системи розпізнавання електронограм

На відміну від нейроподібних структур моделі інформаційно-екстремального машинного навчання системи розпізнавання будуються в рамках функціонального підходу до моделювання когнітивних процесів, притаманних людині при формуванні та безпосередньо прийнятті класифікаційних рішень.

Розглянемо категорійну модель машинного навчання системи розпізнавання електронограм у вигляді орієнтованого графа, в якому задіяні в процесі машинного навчання множини відображаються одна на одну відповідними операторами.

Категорійна модель включає вхідний математичний опис системи розпізнавання у вигляді структури (1.6). При цьому декартовий добуток $G \times T \times \Omega \times Z$ утворює універсум випробувань, який є джерелом інформації для формування вхідної навчальної матриці Y . Категорійну модель інформаційно-екстремального машинного навчання системи розпізнавання електронограм з оптимізацією параметрів машинного навчання – координат структурованого вектора (1.6) показано на рис. 3.7 [11].

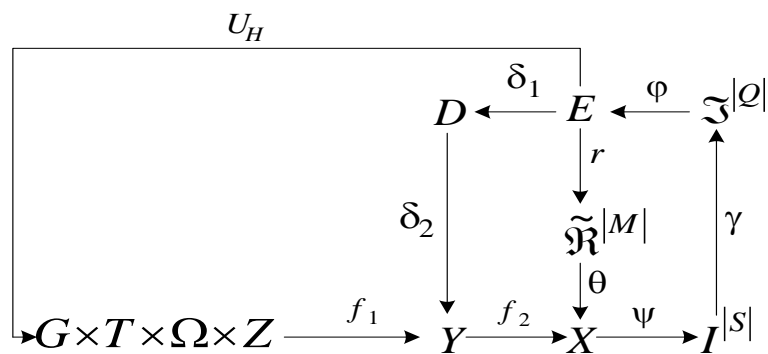


Рисунок 3.7 – Категорійна модель машинного навчання системи розпізнавання

На рис. 3.7 терм-множина E , елементами якої є обчислені на кожному кроці машинного навчання значення інформаційного критерію, є загальною для всіх контурів оптимізації параметрів машинного навчання. Оператор

$r: E \rightarrow \tilde{\mathfrak{R}}^{|M|}$ буде на кожному кроці навчання розбиття $\tilde{\mathfrak{R}}^{|M|}$, яке відображається оператором θ на нечіткий розподіл двійкових реалізації матриці X . Далі оператор $\psi: X \rightarrow I^{|S|}$, де $I^{|S|}$ – множина S гіпотез, перевіряє основну статистичну гіпотезу $\gamma_1: x_{m,i}^{(j)} \in X_m^o$. Оператор γ визначає множину $\mathfrak{S}^{|Q|}$ точнісних характеристик класифікаційних рішень, де $Q = S^2$, а оператор ϕ обчислює множину значень E інформаційного критерію оптимізації, якій є функціоналом від точнісних характеристик. Контур оптимізації контрольних допусків на ознаки розпізнавання замикається через терм-множину D , елементами якої є значання системи контрольних допусків на ознаки розпізнавання. Оператор u регламентує процес машинного навчання.

Перевірка функціональної ефективності інформаційно-екстремального машинного навчання здійснюється при функціонуванні системи розпізнавання в режимі екзамену, алгоритм якого аналогічний алгоритму функціонування системи безпосередньо в робочому режимі.

Категорійну модель у вигляді орієнтованого графа відображень множин, які застосовуються при функціонуванні системи розпізнавання в режимі екзамену, показано на рис. 3.8 [11].

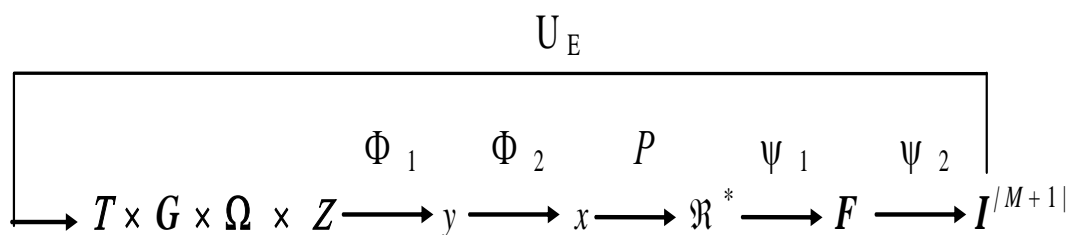


Рисунок 3.8 – Категорійна модель функціонування системи розпізнавання в режимі екзамену

У категорійній моделі (рис. 3.8) оператор Φ_1 із джерела інформації формує екзаменаційну реалізацію класу, що розпізнається, аналогічну за

структурою реалізації з навчальної матриці. Оператор Φ_2 за отриманими на етапі машинного навчання оптимальними контрольними допусками на ознаки розпізнавання формує двійкову реалізацію x , а оператор P відображає вектор-реалізацію, що розпізнається, на побудоване на етапі машинного навчання оптимальне розбиття \mathcal{R}^* класів розпізнавання. Оператор Ψ_1 для кожної реалізації обчислює значення побудованих на етапі машинного навчання вирішальних правил і формує терм-множину F , а оператор Ψ_2 за максимальним значенням вирішального правила відносить вектор, що розпізнається, до одного із класів заданого алфавіту $\{X_m^o\}$. Призначенням оператора U_E є регламентація процесу екзамену.

Розглянуті категорійні моделі відбивають притаманні людині перетворення інформації, які мають місце при когнітивних процесах формування та прийняття класифікаційних рішень. Тому вони розглядаються як узагальнені структурні схеми алгоритмів інформаційно-екстремального машинного навчання системи розпізнавання.

3.3 Алгоритм інформаційно-екстремального машинного навчання з оптимізацією контрольних допусків на ознаки розпізнавання

Базовий алгоритм як правило не забезпечує високої функціональної ефективності машинного навчання, оскільки контрольні допуски, як правило, можуть бути не оптимальними в інформаційному розумінні. Тому обов'язковою процедурою інформаційно-екстремального машинного навчання є оптимізація контрольних допусків на ознаки розпізнавання.

Згідно з категорійною моделлю (рис. 3.6) інформаційно-екстремальний алгоритм машинного навчання системи розпізнавання електронограм з оптимізацією системи контрольних допусків подається у вигляді ітераційної процедури пошуку глобального максимуму усередненого за алфавітом класів розпізнавання інформаційного критерію в робочій (допустимій) області визначення його функції [11]

$$\delta_K^* = \arg \max_{G_\delta} \{ \max_{G_E \cap \{k\}} \bar{E}^{(k)} \}, \quad (3.3)$$

де δ_K^* – оптимальний параметр поля контрольних допусків;

G_δ – допустима область значень параметра δ поля контрольних допусків;

$\{k\}$ – впорядкована за часом множина кроків машинного навчання.

Вхідною інформацією для алгоритму машинного навчання є масив $\{y_{m,i}^{(j)}\}$ і система полів нормованих допусків $\{\delta_{H,i}\}$ на ознаки розпізнавання, яка задає область значень відповідних контрольних допусків.

Розглянемо схему алгоритму машинного навчання з оптимізацією контрольних допусків на діагностичні ознаки за процедурою (3.3):

- 1) онулення лічильника класів розпізнавання: $m := 0$;
- 2) $m := m + 1$;
- 3) онулення лічильника зміни параметра δ поля контрольних допусків: $\delta := 0$;
- 4) $\delta := \delta + 1$;
- 5) обчислюються нижні $A_{H,i}$ і верхні $A_{B,i}$ контрольні допуски на ознаки розпізнавання відповідно за правилами

$$A_{H,i} = y_i - \delta; \quad A_{B,i} = y_i + \delta;$$

6) онулення лічильника кроків зміни радіуса гіперсферичного контейнера: $k := 0$;

7) $k := k + 1$;

8) формується тривимірний масив бінарної навчальної матриці X , елементи якої обчислюються за правилом

$$x_{m,i}^{(j)}[k] = \begin{cases} 1, & \text{якщо } A_{HK,i}[k] < y_{m,i}^{(j)} < A_{BK,i}[k]; \\ 0, & \text{якщо інакше} \end{cases}$$

9) формування масиву усереднених двійкових векторів-реалізацій $\{x_m\}$, елементи яких визначаються за правилом

$$x_{m,i} = \begin{cases} 1, & \text{if } \frac{1}{n} \sum_{j=1}^n x_{m,i}^{(j)} > \rho_m; \\ 0, & \text{if } \text{else,} \end{cases}$$

де ρ_m – рівень квантування координат двійкового вектора x_m , який за замовчуванням дорівнює 0,5.

10) розбиття множини векторів $\{x_m\}$ на пари найближчих “сусідів”

$\mathfrak{R}_m^{[2]} = \langle x_m, x_c \rangle$, де x_c – усереднений вектор сусіднього класу X_c^0 ;

11) обчислюється інформаційний критерій;

12) якщо $k \leq N$, то виконується пункт 7, інакше – пункт 13;

13) якщо $\delta < \delta_H$, то виконується пункт 4, інакше – пункт 14;

14) визначається максимальне значення інформаційного критерію в робочій області визначення його функції, де перша і друга достовірності більше 0,5;

15) якщо виконується умова $m < M - 1$, то реалізується пункт 2, інакше – пункт 16;

16) визначається глобальний максимум усередненого інформаційного критерію \bar{E}^* в робочій області визначення його функції;

17) визначаються оптимальні значення параметра δ^* і відповідних нижніх $A_{H,i}^*$ і верхніх $A_{B,i}^*$ контрольних допусків на всі ознаки розпізнавання

Але отримані в процесі паралельної оптимізації екстремальні значення параметрів машинного навчання є квазіоптимальними, оскільки вони змінювалися на кожному кроці навчання на однакову величину для всіх ознак одночасно. Одним із шляхів підвищення функціональної ефективності системи розпізнавання є реалізація алгоритму машинного навчання з послідовною оптимізацією контрольних допусків. При цьому отримані на етапі паралельної оптимізації контрольні допуски приймалися як стартові при послідовній оптимізації, яка здійснювалася за процедурою [11]

$$\delta_{K,i}^* = \arg \bigotimes_{l=1}^L \left\{ \max_{G_{\delta_i}} \left[\frac{1}{M} \sum_{m=1}^M \max_{G_{E_m} \cap \{k\}} E_m^{(l)}(d_m) \right] \right\}, i = \overline{1, N}, \quad (3.4)$$

де L – кількість прогонів процедури послідовної оптимізації контрольних допусків, обумовлених неоптимальними стартовими величинами контрольних допусків для всіх ознак;

\bigotimes – символ операції повторення.

Машинне навчання системи розпізнавання з паралельно-послідовною оптимізацією контрольних допусків дозволяє підвищити достовірність класифікаційних рішень і при цьому суттєво підвищується оперативність машинного навчання, оскільки пошук глобального максимуму критерію здійснюється тільки в робочій області визначення його функції.

За отриманими в процесі машинного навчання оптимальними геометричними параметрами контейнерів класів розпізнавання було побудовано продукційні вирішальні правила, які мають вигляд

$$\begin{aligned} (\forall X_m^o \in \mathfrak{R}^{|M|}) (\forall x^{(j)} \in \mathfrak{R}^{|M|}) [if (\mu_m > 0) \& (\mu_m = \max\{\mu_m\}) \\ then x^{(j)} \in X_m^o \text{ else } x^{(j)} \notin X_m^o], \end{aligned} \quad (3.5)$$

де $x^{(j)}$ – вектор, що розпізнається;

μ_m – функція належності вектора $x^{(j)}$ класу розпізнавання X_m^o .

У виразі (3.5) функція належності для гіперсферичного контейнера класу розпізнавання X_m^o визначається за формулою [7]

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \quad (3.6)$$

де x_m^* – оптимальна усереднена реалізація класу розпізнавання X_m^o ;

d_m^* – оптимальний радіус гіперсферичного контейнера класу розпізнавання X_m^o .

Таким чином, на екзамені визначається за вирішальними правилами (3.5) належність реалізації класу розпізнавання, що розпізнається, одному із класів із заданого алфавіту. При цьому вирішальні правила через малу обчислювальну трудомісткість відрізняються високою оперативністю.

3.4 Опис програмного забезпечення системи розпізнавання електронограм

Алгоритм машинного навчання системи розпізнавання електронограм з оптимізацією системи контрольних допусків на ознаки розпізнавання було реалізовано на мові $C_{\#}$. Для зручності сприйняття і використання опис коду розробленого програмного забезпечення представлено у вигляді таблиць (3.1) – (3.3). Нижче наведено класи, які безпосередньо приймають участь у реалізації алгоритму машинного навчання.

Клас *polarCoord* обробляє зображення електронограм в полярній системі координат, будує спектри RGB-складових, за значеннями їх дискрет формує вхідну навчальну матрицю, взаємодіє і зберігає інформацію про клас.

Таблиця 3.1 – Опис складових класу polarCoord

ЗМІНИ			
Назва	тип	Модифікатор доступу	Призначення
img	itmap	private	Зберігає клас у вигляді зображення
matrix_size	int	private	Кількість кроків, які ми будемо виконувати при переході в декартову систему координат
max_radius	int	private	Максимальний радіус концентричного кола
zone	int[,]	private	Зберігає клас у вигляді матриці яскравості пікселів
binary_map_zone	int[,]	private	Зберігає первинну бінарну матрицю класу, яка була отримана під час паралельної оптимізації в декартовій системі координат
binary_map_polar	int[,]	private	Бінарна матриця для полярної системи координат.
polar_map	int[,]	private	Клас, який був переведений в полярну систему координат
etalon_vektor	int[]	private	Зберігає еталонний вектор класу

N	nt	private	Кількість ознак (стовпці)
n	nt	private	Кількість реалізацій (строки)
ro	loat	private	Рівень квантування координат двійкового вектору
center	oint	private	Центр полярної системи координат
Методи			
Назва		Призначення	
public PolarCoord(Bitmap source)		Конструктор класу. Приймає зображення і виконує первинну ініціалізацію змінних	
public int getElem_EV(int i)		Повертає значення еталонного вектору за його індексом.	
public int getElem_VMpolar(int i, int j)		Повертає значення бінарної карти за індексом.	
public int getColumns()		Повертає кількість колонок зображення (N)	
public int getRows()		Повертає кількість строк зображення (n)	
public int getPixel(int i, int j)		Повертає значення яскравості пікселя за його індексом.	
public void setPixel(int i, int j, int value)		Встановлює значення яскравості пікселя за його індексом.	

<code>public void setBinaryMapZone(int i, int j, int val)</code>	Встановлює, за індексом, значення для первинної бінарної матриці.
<code>public int getMatrixSize()</code>	Повертає значення <code>matrixSize</code> .
<code>public int getMaxRadius()</code>	Повертає максимальне значення радіусу для концентричних кіл.
<code>public int getElem_polarBM(int i, int j)</code>	Повертає, за індексом, значення бінарної матриці класу, який був переведений в полярну систему координат.
<code>public int getElem_polarMap(int i, int j)</code>	Повертає значення класу, який був переведений в полярну систему координат.
<code>public void make_polar_coord()</code>	Переведення класу до полярної системи координат
<code>int draw_circle(int d, double i_start, double i_end, ref int R, ref int G, ref int B)</code>	Знаходження пікселя за його полярними координатами.
<code>void find_center()</code>	Знаходження центру первинної бінарної матриці
<code>public void make_etalon_vektor()</code>	Створює еталонний вектор
<code>public void make_binary_map(int[] vd, int[] nd)</code>	Створює бінарну карту за наявними допусками
<code>public void make_exams_binary_map(int[] vd, int[] nd)</code>	Створення первинної бінарної матриці для екзаменаційного зображення.

Клас *teachPolar* виконує безпосереднє навчання в рамках інформаційно-екстремальної інтелектуальної технології для класів, які представлені в полярній системі координат.

Таблиця 3.2 – Опис складових класу *teachPolar*

ЗМІНИ			
Назва	Тип	Модифікатор доступу	Призначення
polars	List ◊	private	Колекція класів на яких навчається ІС
decartDat	Tea chDecart	private	Зберігає інформацію про первинне оброблення класів в декартовій системі
para_d	int[]	private	Відстань до сусіда
para	int[]	private	Список сусідів
sk	int[,]	private	0-й стовпчик – це відстані від сусідніх реалізацій до свого EV 1-й стовпчик – це відстані від свої реалізацій до свого EV
sk_para	int[,]	private	0-й стовпчик – це відстані від сусідніх реалізацій до сусіднього EV 1-й стовпчик – це відстані від свої реалізацій до сусіднього EV
em	float[]	private	Значення критерію для

			класів
my_do	int[]	private	Значення радіусів вирішальних правил
vd	int[]	private	Верхні допуски
nd	int[]	private	Нижні допуски
limit_d	int	private	Максимальне можливе значення дельта
d	int	private	Дельта
d_posl	int[]	private	Масив значень дельта для кожної ознаки класу
Методи			
Назва		Призначення	
Public TeachPolar(List<decartCoord> sender)		Конструктор класу. Приймає класи і виконує первинну ініціалізацію змінних	
public TeachDecart getDecart(int i)		Отримання інформації про первинне навчання	
public Bitmap getClass(int i)		Отримання зображення, яке представляє клас	
public int getRadius(int i)		Отримання радіусу для певного класу	
public int[] getVD()		Отримання масиву верхніх допусків	
public int[] getND()		Отримання масиву нижніх допусків	
void realization()		Процес машинного навчання, який виконує	

	створення бінарної карти, еталонних векторів, розбиття на пари і оптимізацію радіусів
void start()	Первинне навчання, яке виконується для створення початкової бінарної матриці. На її основі ми будемо переходити в полярну систему координат.
bool make_dopusk(int k, int delta)	Змінна допусків для k-го класу
void make_para()	Розбиття класів на пари
double INFK(int INFK_d, ref float INFK_d1, ref float INFK_betta, int k)	Обчислення критерію
void make_sk(int k)	Знаходження відстаней для своїх і сусідніх реалізацій
void make_myDo()	Оптимізація радіусів
bool propusk_KFE(int radius, int k)	Перевірка на робочу область
public float average_KFE()	Усереднене значення критерію
public void parallelOptimization()	Паралельна оптимізація СКД
bool local_dopusk(int i, int delta)	Зміна допусків для i -ї ознаки
public void sequentialOptimization()	Процес послідовної

	оптимізації СКД
--	-----------------

Клас *machineLearn* клас, організовує роботу машинного навчання (Клас *teachPolar*) з користувачем. Виконує екзамен.

Таблиця 3.3 – Опис складових класу *machineLearn*

ЗМІНИ			
Назва	Тип	Модифікатор доступу	Призначення
polars	List ◊	private	Колекція класів
teachPolar	TeachDec art	public	Екземпляр класу, який виконує машинне навчання і зберігає результати по ньому.
Методи			
Назва		Призначення	
public machineLearn()		Конструктор класу. Виконує первинну ініціалізацію змінних	
public void realizationPolar()		Послідовно викликає методи оптимізації вирішальних правил.	
public	void	Отримує значення яскравості пікселів із зображення	
make_fragment(Bitmap bitmap)			
public int	examsPolar(Bitmap bitmap)	Виконує алгоритм екзамену. Повертає значення класу до якого належить зображення.	

Користувацький інтерфейс і його функціональності не були описані у цьому розділі. Це пов'язано із тим, що розробник сам вирішує як він хоче застосувати класи, які наведені вище.

3.5 Результати моделювання

Алфавіт класів розпізнавання складався з чотирьох класів: клас X_1^o – електронограма полікристала (рис.3.1), клас X_2^o – електронограма з Кікучі-лініями (рис. 3.2), клас X_3^o – електронограма мозаїчного монокристалу (рис. 3.3) і клас X_4^o – електронограма текстури (рис. 3.4).

Спочатку було реалізовано алгоритм інформаційно-екстремального машинного навчання системи розпізнавання електронограм з паралельною оптимізацією системи контрольних допусків на ознаки розпізнавання за процедурою (3.3). На рис. 3.9 показано графік залежності усередненого критерію (1.17) від параметра δ (delta), отриманий при реалізації алгоритму машинного навчання з паралельною оптимізацією контрольних допусків.

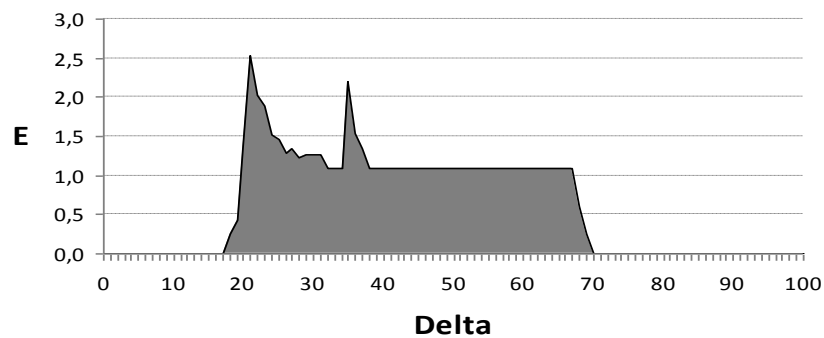


Рисунок 3.9 – Графік залежності усередненого критерію оптимізації від параметра поля контрольних допусків для лінійного класифікатора

Аналіз рис. 3.9 показує, що максимальне усереднене значення інформаційного критерію дорівнює 2,53, при оптимальному параметрі поля контрольних допусків $\delta^* = 21$ градацій яскравості. Оскільки граничне максимальне значення ненормованого критерію (1.17) при 30 реалізаціях

одного класу розпізнавання дорівнює $E_{гран}^* = 4,45$, то можна зробити висновок, що лінійний класифікатор вже при алфавіті із чотирьох класів розпізнавання не забезпечує високу достовірність класифікаційних рішень. Тому з метою зменшення впливу багатовимірності на функціональну ефективність машинного навчання доцільно перейти на ієрархічну структуру даних. На рис. 3.10 показано ієрархічну структуру даних для заданого алфавіту класів розпізнавання у вигляді декурсивного т дерева, в якому атрибут вершини верхнього ярусу передається на вершину його страти на нижньому ярусі.

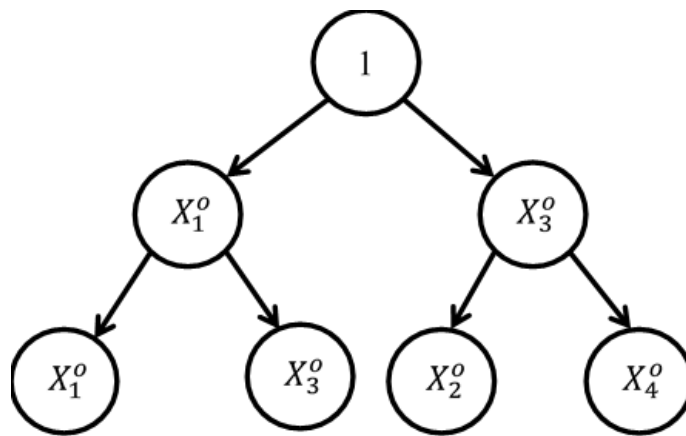


Рисунок 3.10 – Ієрархічна структура даних.

Аналіз рис.3.10 показує, що машинне навчання за процедурою (3.3) буде здійснюватися тепер на кожній страті тільки для двох класів розпізнавання.

На рис.3.11 показано графік залежності усередненого для класів розпізнавання першої фінальної страти нижнього ярусу критерію (2.9) від параметра поля контрольних допусків. На цьому графіку і далі робочу область визначення функції критерію позначено світлою ділянкою.

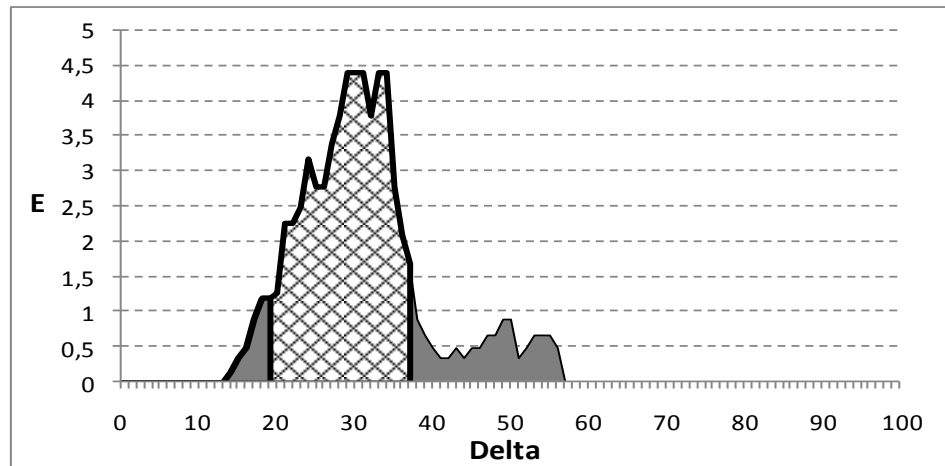
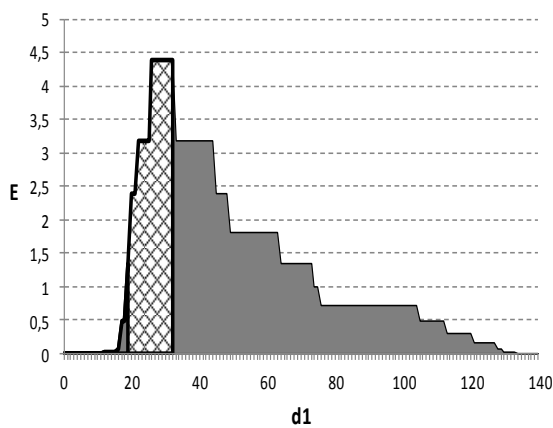


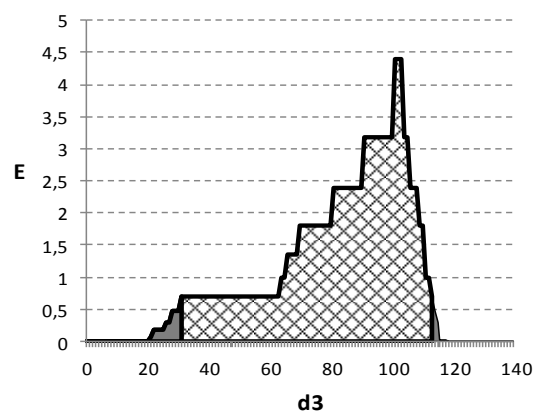
Рисунок 3.11 – Графік залежності усередненого критерію (2.9) від параметра поля контрольних допусків для першої фінальної страти

Аналіз рис. 3.11 показує, що максимальне усереднене значення критерію оптимізації дорівнює $E^* = 4,40$ при оптимальному параметрі поля контрольних допусків $\delta^* = 28$ градаций яскравості.

В процесі машинного навчання у внутрішньому циклі процедури (3.3) визначалися оптимальні геометричні параметри контейнерів класів розпізнавання, за яким будувалися вирішальні правила (3.15). На рис. 3.12 показано графіки залежності критерія (2.9) від радіусів контейнерів класів розпізнавання першої фінальної страти



а



б

Рисунок 3.11 – Графіки залежності критерія (2.9) від радіусів контейнерів класів розпізнавання: а – клас X_1^o ; б – клас X_3^o

Аналіз рис. 3.11 показує, що оптимальні радіуси контейнерів класів розпізнавання X_1^o і X_3^o відповідно дорівнюють $d_1^* = 24$ (тут і далі в козових одиницях) і $d_3^* = 102$. При цьому максимальне значення критерію оптимізації для першої фінальної страти дорівнює $E_1^* = 4,40$. При такому значенні критерію усереднена перша достовірність дорівнює $D_1^* = 0,96$, а помилка другого роду – $\beta = 0$.

На рис. 3.12 показано графік залежності усередненого для класів розпізнавання другої фінальної страти нижнього ярусу критерію (2.9) від параметра поля контрольних допусків.

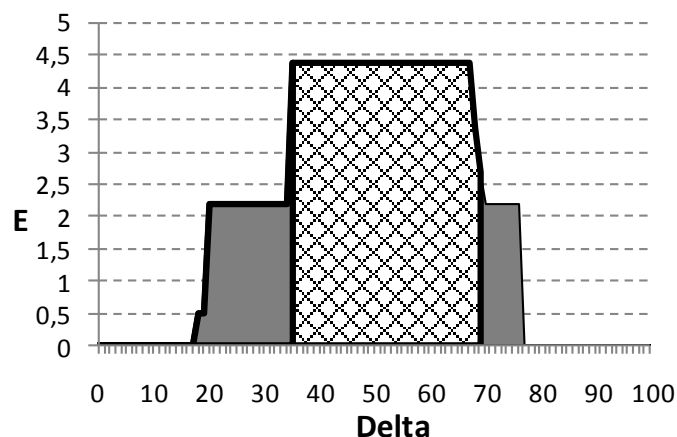


Рисунок 3.12 – Графік залежності усередненого критерію (2.9) від параметра поля контрольних допусків для першої фінальної страти

Аналіз рис. 3.12 показує, що максимальне усереднене значення критерію оптимізації дорівнює $E^* = 4,43$ при оптимальному параметрі поля контрольних допусків $\delta^* = 35$ градацій яскравості.

На рис. 3.11 і на рис. 3.12 на графіках присутні ділянки типу “плато”, на яких функція критерію не є взаємнооднозначною. Тому визначення

екстремального значення параметра здійснювалося при мінімальному значенні коефіцієнта нечіткої компактності [10]

$$\eta = \frac{d_m^*}{d(x_m \oplus x_c)},$$

де d_m^* – оптимальний радіус контейнера класу розпізнавання X_m^o ;

$d(x_m \oplus x_c)$ – міжцентрова відстань в кодових одиницях між найближчими сусідніми класами X_m^o і X_c^o .

На рис. 3.13 показано графіки залежності критерія (2.9) від радіусів контейнерів класів розпізнавання другої фінальної страти нижнього ярусу.

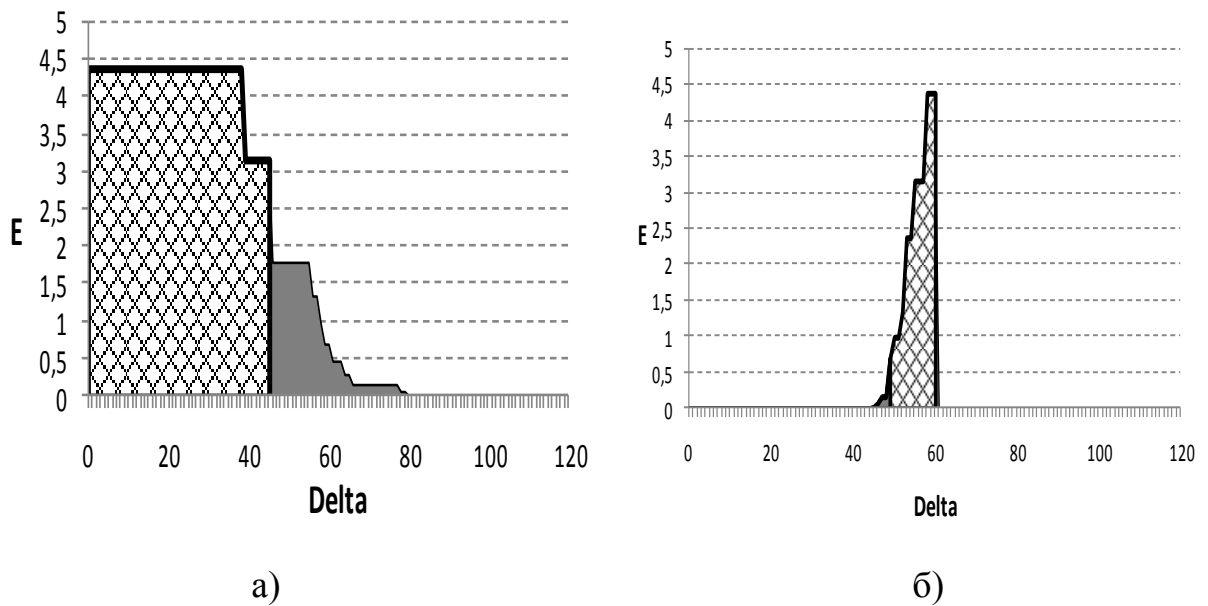


Рисунок 3.13 – Графіки залежності критерія (1.17) від радіусів контейнерів класів розпізнавання: а – клас X_2^o ; б – клас X_4^o

Аналіз рис. 3.13 показує, що оптимальні радіуси контейнерів класів розпізнавання X_1^o і X_3^o відповідно дорівнюють $d_1^* = 36$ і $d_4^* = 61$. При цьому

максимальне значення критерію оптимізації для другої фінальної страти так само дорівнює $E_1^* = 4,40$.

Аналіз результатів інформаційно-екстремального машинного навчання системи розпізнавання електронограм показує, що при використанні ієрархічного класифікатора максимальне значення інформаційного критерію оптимізації майже у двічі більше у порівнянні з лінійним алгоритмом навчання і наближається до максимального граничного значення.

Для перевірки функціональної ефективності машинного навчання системи розпізнавання було реалізовано режим екзамену із застосуванням .вирішальних правил (3.5). На рисю 3.14 показано інтерфейс програми екзамену, на якому система правильно розпізнала електронограму мозаїчний монокристал, оскільки функція належності (3.6) саме для цього класу розпізнавання мала максимальне додатне значення.

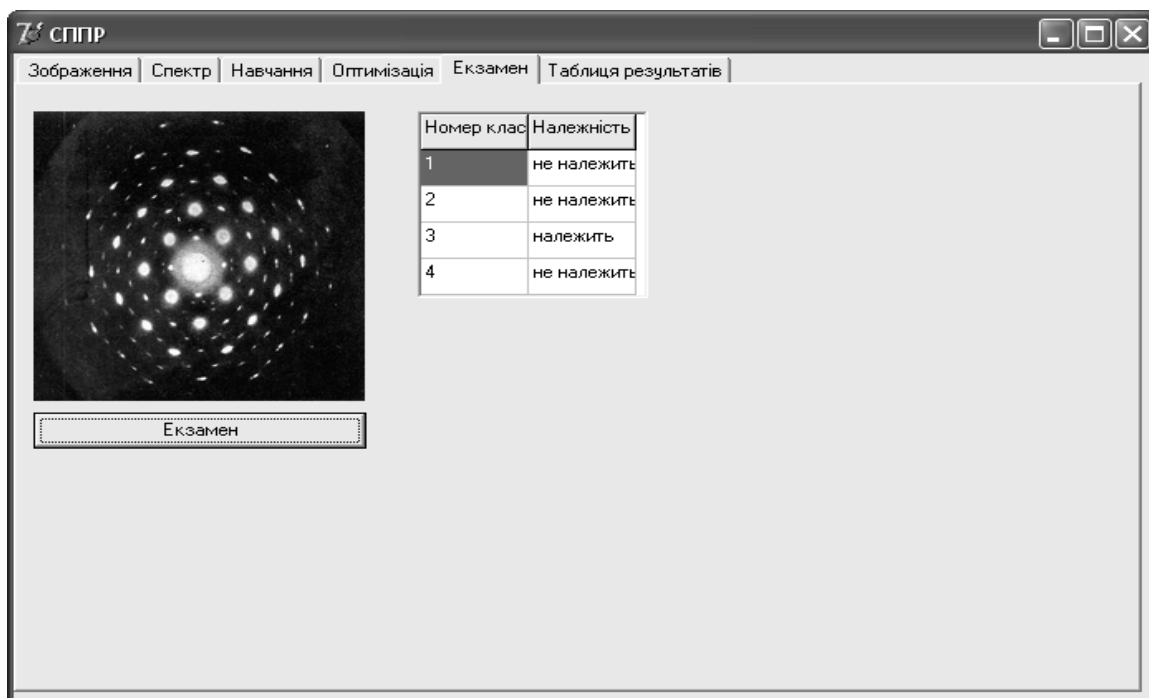


Рисунок 3.14 – Інтерфейс програми екзамену для розпізнавання реалізації класу розпізнавання X_3^0 – мозаїчний монокристал

Таким чином, за результатами фізичного моделювання підтверджена висока функціональна ефективність машинного навчання системи розпізнавання електронограм для заданого алфавіту класів розпізнавання.

ВИСНОВКИ

1. Розроблено метод інформаційно-екстремального машинного навчання системи розпізнавання електронограм за навчальною матрицею яскравості, сформованої шляхом оброблення зображень в полярних координатах, що дозволяє підвищити достовірність розпізнавання електронограм і забезпечує інваріантність вирішальних правил до довільних початкових умов формування досліджуваного зразка.

2. Побудовані за результатами машинного навчання в рамках геометричного підходу вирішальні правила є практично інваріантними до багатовимірності простору ознак розпізнавання.

3. Показано, що збільшення потужності алфавіту класів розпізнавання призводить до зменшення функціональної ефективності машинного навчання за лінійним алгоритмом. Тому для підвищення достовірності розпізнавання електронограм розроблено та програмно реалізовано алгоритм машинного навчання за ієрархічною структурою даних, що дозволило у порівнянні з лінійним класифікатором підвищити повну ймовірність правильного прийняття класифікаційних рішень.

4. Розроблене інформаційне, алгоритмічне та програмне забезпечення системи розпізнавання за результатами фізичного моделювання забезпечує високу достовірність розпізнавання електронограм в режимі екзамену.

СПИСОК ЛІТЕРАТУРИ

1. Ichimiya A., Cohen P.I. Reflection High Energy Electron Diffraction. - Cambridge: University Press, 2004. - 353 p.
2. Schwartz A.J., Kumar M., Adams B.L., Field D.P. (Eds.) Electron Backscatter Diffraction in Materials Science. - 2nd Edition. - Springer Science+Business Media, LLC, 2009. - 406 p.
3. Андерсон Т. Введение в многомерный статистический анализ: Пер. с англ./ Под ред. Б.В. Гнеденко. — М.: Физматгиз, 1963. — 400 с.
4. Duda R. O., Hart P. E., Stork D. G. Pattern Classification, second ed. John Wiley & Sons, New York, 2001.— 738 p.
5. Webb A.R. Statistical Pattern Recognition : Third edition / A.R. Webb, K.D. Copsey. – Chichester, UK : John Wiley & Sons. – 2011. – 666 p.
6. Artificial Neural Networks – Methodological Advances and Biomedical Applications : [Ed. by K. Suzuki]. – InTech, 2011. – 362 p.
7. Зайченко Ю. П. Основи проектування інтелектуальних систем : навчальний посібник / Ю. П. Зайченко. – К. : Видавничий дім «Слово», 2004. – 352 с.
8. Осовский С. Нейронные сети для обработки информации. Пер. с польск. / С. Осовский. – М.: Научно-техническое издательство «Горячая линия – Телеком», 2017. – 448 с.
9. Анализ данных и процессов / [А. А. Барсегян [и др.]. – [3-е изд.]. – СПб.: БХВ-Петербург, 2009. – 512 с.
10. Довбиш А.С. Інтелектуальні інформаційні технології в електронному навчанні / А.С. Довбиш, А.В. Васильєв, В.О. Любчак. – Суми: Видавництво СумДУ. – 2013. – 172 с.
11. Довбиш А.С. Основи проектування інтелектуальних систем: Навчальний посібник / А.С. Довбиш. – Суми: Видавництво СумДУ, 2009.

ДОДАТОК

Клас TeachPolar

```

using System;
using System.Collections.Generic;
using System.Drawing;

namespace machineLearning
{
    public class TeachPolar
    {
        List<PolarCoord> polars = new List<PolarCoord>();
        TeachDecart decartData;
        int[] para_d;
        int[] para;
        int[,] sk;
        int[,] sk_para;
        float[] em;
        int[] my_do;

        int[] vd;
        int[] nd;

        int limit_d = 80;
        int d;
        int[] d_posl;

        public TeachPolar(List<PolarCoord> sender) {
            foreach (var item in sender)
                polars.Add(item);
            sk = new int[2, polars[0].getMaxRadius()];
            sk_para = new int[2, polars[0].getMaxRadius()];

            para_d = new int[polars.Count];
            para = new int[polars.Count];

            em = new float[polars.Count];
            my_do = new int[polars.Count];

            vd = new int[polars[0].getMatrixSize()];
            nd = new int[polars[0].getMatrixSize()];

            d_posl = new int[polars[0].getMatrixSize()];

            start();
        }

        public TeachDecart getDecart() {
            return decartData;
        }
        public Bitmap getClass(int i)
        {

```

```

    return polars[i].img;
}
public int[] getVD()
{
    return vd;
}
public int[] getND()
{
    return nd;
}
public int getRadius(int i)
{
    return my_do[i];
}
public void realization() {
    foreach (var polar in polars)
        polar.make_binary_map(vd, nd);

    for (int k = 0; k < polars.Count; k++)
        polars[k].make_etalon_vektor();

    make_para();
    make_myDo();
}
void start() {
    List<DecartCoord> decarts = new List<DecartCoord>();
    foreach (var polar in polars)
    {
        DecartCoord decart = new DecartCoord(polar.img);
        for (int i = 0; i < polar.getColumns(); i++)
            for (int j = 0; j < polar.getRows(); j++)
                decart.setPixel(i, j, polar.getPixel(i, j));

        decarts.Add(decart);
    }

    decartData = new TeachDecart(decarts);
    decartData.parallelOptimization();
    for (int k = 0; k < polars.Count; k++)
    {
        for (int i = 0; i < polars[0].getColumns(); i++)
            for (int j = 0; j < polars[0].getRows(); j++)
                polars[k].setBinaryMapZone(i, j, decartData.getDecart(k).getElem_BM(i,j));

        polars[k].make_polar_coord();
    }
}
public bool make_dopusk(int k, int delta)
{
    for (int i = 0; i < polars[k].getMatrixSize(); i++)
    {
        int sum = 0;

```

```

    for (int j = 0; j < polars[k].getMaxRadius(); j++)
        sum += polars[k].getElem_polarMap(i, j);

    sum /= polars[k].getMaxRadius();

    vd[i] = sum + delta;
    nd[i] = sum - delta;
    if (vd[i] > 255 || nd[i] < 0)
    {
        limit_d = delta;
        return false;
    }
}
return true;
}
void make_para()
{
    for (int k = 0; k < polars.Count; k++)
    {
        para_d[k] = 0;
        int ev_sum1 = polars[k].getMatrixSize();
        for (int k1 = 0; k1 < polars.Count; k1++)
        {
            if (k1 != k)
            {
                int ev_sum = 0;
                for (int i = 0; i < polars[k].getMatrixSize(); i++)
                {
                    if (polars[k].getElem_EV(i) != polars[k1].getElem_EV(i))
                        ev_sum++;
                }
                if (ev_sum <= ev_sum1)
                {
                    ev_sum1 = ev_sum;
                    para[k] = k1;
                    para_d[k] = ev_sum1;
                }
            }
        }
    }
}
double INFK(int INFK_d, ref float INFK_d1, ref float INFK_beta, int k)
{
    int k1 = 0, k4 = 0;
    for (int INFK_j = 0; INFK_j < polars[k].getMaxRadius(); INFK_j++)
    {
        if (sk[0, INFK_j] <= INFK_d) k1++;
        if (sk[1, INFK_j] <= INFK_d) k4++;
    }
    INFK_d1 = (float)k1 / (float)polars[k].getMaxRadius();
    INFK_beta = (float)k4 / (float)polars[k].getMaxRadius();
}

```

```

float d1_b = INFK_d1 - INFK_beta;
double best_res = 1 * Math.Log((1 + 1 + 0.001) / (1 - 1 + 0.001));

return d1_b * Math.Log((1 + d1_b + 0.001) / (1 - d1_b + 0.001)) / best_res;
}
void make_sk(int k)
{
for (int j = 0; j < polars[k].getMaxRadius(); j++)
{
sk[1, j] = 0;
sk[0, j] = 0;
sk_para[1, j] = 0;
sk_para[0, j] = 0;
for (int i = 0; i < polars[k].getMatrixSize(); i++)
{
if (polars[k].getElem_EV(i) != polars[para[k]].getElem_BMpolar(i, j)) sk[1, j] += 1;
if (polars[k].getElem_EV(i) != polars[k].getElem_BMpolar(i, j)) sk[0, j] += 1;
if (polars[para[k]].getElem_EV(i) != polars[para[k]].getElem_BMpolar(i, j))
sk_para[1, j] += 1;
if (polars[para[k]].getElem_EV(i) != polars[k].getElem_BMpolar(i, j)) sk_para[0, j]
+= 1;
}
}
}
void make_myDo()
{
float t_d1 = 0, t_beta = 0;
for (int k = polars.Count - 1; k >= 0; k--)
{
em[k] = 0;
my_do[k] = 0;

make_sk(k);
float d_correct = 999;
float d_tmp;
for (int d = 1; d < polars[k].getMatrixSize(); d++)
{
double e = INFK(d, ref t_d1, ref t_beta, k);

if (e >= em[k] && t_d1 >= 0.5 && t_beta <= 0.5 && d < para_d[k])
{
if (e > em[k])
my_do[k] = d;
em[k] = (float)e;

d_tmp = (float)d / para_d[k];
if (e == em[k] && d_tmp < d_correct)
{
d_correct = d_tmp;
my_do[k] = d;
}
}
}
}
}

```

```

        if (e > em[k] && my_do[k] == 0)
            em[k] = (float)e;
    }
}
}
bool propusk_KFE(int radius, int k) {
    float t_d1 = 0, t_beta = 0;

    INFK(radius, ref t_d1, ref t_beta, k);
    if (t_d1 >= 0.5 && t_beta <= 0.5)
        return true;
    return false;
}
float average_KFE()
{
    float avr = 0;
    for (int k = 0; k < polars.Count; k++)
        avr += em[k];
    avr /= polars.Count;
    return avr;
}
public void parallelOptimization()
{
    double em_last_d = 0;
    float radius = 0;

    for (int delta = 1; delta <= limit_d; delta++)
    {
        if (!make_dopusk(0, delta)) break;
        realization();

        float em_now = average_KFE();
        if (em_now > em_last_d && propusk_KFE(my_do[0], 0))
        {
            em_last_d = em_now;
            d = delta;
            radius = my_do[0];
        }
        else if (em_now == em_last_d && radius <= my_do[0] && propusk_KFE(my_do[0],
0))
        {
            em_last_d = em_now;
            d = delta;
            radius = my_do[0];
        }
    }
    make_dopusk(0, d);
    realization();
}
bool local_dopusk(int i, int delta)
{
    int sum = 0;

```

```

for (int j = 0; j < polars[0].getMaxRadius(); j++)
    sum += polars[0].getElem_polarMap(i, j);

sum /= polars[0].getMaxRadius();

vd[i] = sum + delta;
nd[i] = sum - delta;

if (vd[i] > 255 || nd[i] < 0)
    return false;
return true;
}
public void sequentialOptimization()
{
    float last_res_G = 0, now_res_G = 0;
    for (int i = 0; i < polars[0].getMatrixSize(); i++)
        d_posl[i] = d;

    now_res_G = average_KFE();
    do
    {
        last_res_G = now_res_G;
        for (int i = 0; i < polars[0].getMatrixSize(); i++)
        {
            float last_res_L = 0;
            float radius = 0;

            for (int delta = 1; delta < 80; delta++)
            {
                if(!local_dopusk(i, delta)) break;
                realization();

                float now_res_L = average_KFE();
                if (now_res_L > last_res_L)
                {
                    last_res_L = now_res_L;
                    d_posl[i] = delta;

                    radius = my_do[0];
                }
                else if (now_res_L == last_res_L && radius < my_do[0])
                {
                    last_res_L = now_res_L;
                    d_posl[i] = delta;

                    radius = my_do[0];
                }
            }
            local_dopusk(i, d_posl[i]);
            realization();
        }
    }
}

```

```

        now_res_G = average_KFE();
    } while (Math.Abs(now_res_G - last_res_G) > 0.0001);
    }
}
}

```

Κλάσ PolarCoord

```

using System;
using System.Drawing;

namespace machineLearning
{
    public class PolarCoord
    {
        public Bitmap img;
        int matrix_size = 40;
        int[,] polar_map;
        int[,] binary_map_zone;
        int[,] binary_map_polar;
        int[,] zone;
        int[] etalon_vektor;

        int[,] RGB;

        int N;
        int n;
        float ro = 0.5f;
        int max_radius = 27;

        Point center;
        public PolarCoord(Bitmap source) {
            img = source;

            n = img.Height;
            N = img.Width;

            etalon_vektor = new int[matrix_size];
            binary_map_zone = new int[N, n];
            binary_map_polar = new int[matrix_size, max_radius];
            zone = new int[N, n];

            RGB = new int[matrix_size, 3];
        }

        public int getPixel(int i, int j)
        {
            return zone[i, j];
        }
        public void setPixel(int i, int j, int value)
        {
            zone[i, j] = value;
        }
        public void setBinaryMapZone(int i, int j, int val) {

```

```

    binary_map_zone[i, j] = val;
}
public int getColumns()
{
    return N;
}
public int getRows()
{
    return n;
}
public int getMatrixSize() {
    return matrix_size;
}
public int getMaxRadius() {
    return max_radius;
}
public int getElem_EV(int i)
{
    return etalon_vektor[i];
}
public int getElem_polarBM(int i, int j)
{
    return binary_map_polar[i, j];
}
public int getElem_polarMap(int i, int j) {
    return polar_map[i, j];
}
public int getElem_BMpolar(int i, int j) {
    return binary_map_polar[i, j];
}
public void make_polar_coord()
{
    find_center();

    polar_map = new int[matrix_size, max_radius];
    int angle = 0;
    int R = 0, G = 0, B = 0;

    for (int r = 0; r < matrix_size; r++)
    {
        for (int d = 0; d < max_radius; d++)
        {
            if (r == 0) polar_map[r, d] = draw_circle(d, 0, 360, ref R, ref G, ref B);
            else polar_map[r, d] = draw_circle(d, angle, angle + 15, ref R, ref G, ref B);
        }
        angle += 23;

        RGB[r, 0] = R / max_radius;
        RGB[r, 1] = G / max_radius;
        RGB[r, 2] = B / max_radius;

        R = G = B = 0;
    }
}

```



```

    }

}

int draw_circle(int d, double i_start, double i_end, ref int R, ref int G, ref int B)
{
    int r = 0, g = 0, b = 0;

    for (double angle = i_start; angle < i_end; angle++)
    {
        double x1 = d * Math.Cos(angle * Math.PI / 180f);
        double y1 = d * Math.Sin(angle * Math.PI / 180f);

        int ElX = (int)Math.Round(center.X + x1);
        int ElY = (int)Math.Round(center.Y + y1);

        if ((ElX >= 0) && (ElY >= 0) && (ElX < N) && (ElY < n))
        {
            Color color = img.GetPixel(ElX, ElY);
            r += color.R;
            r = r / 2;

            g += color.G;
            g = g / 2;

            b += color.B;
            b = b / 2;
        }
    }

    R += r;
    G += g;
    B += b;

    return (r + g + b) / 3;
}

void find_center()
{
    int count = 0;
    center.X = center.Y = 0;

    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (binary_map_zone[i, j] == 1)
            {
                center.X += i;
                center.Y += j;
                count++;
            }
        }
    }
}

```



```

{
public class MachineLearn
{
    List<PolarCoord> polars;
    public TeachPolar teachPolar;
    public MachineLearn()
    {
        polars = new List<PolarCoord>();
    }
    public void realizationPolar()
    {
        teachPolar = new TeachPolar(polars);
        teachPolar.parallelOptimization();
        teachPolar.sequentialOptimization();
    }
    public void make_fragment(Bitmap bitmap)
    {
        PolarCoord polar = new PolarCoord(bitmap);
        for (int i = 0; i < bitmap.Width; i++)
        {
            for (int j = 0; j < bitmap.Height; j++)
            {
                Color color = bitmap.GetPixel(i, j);
                polar.setPixel(i, j, (color.R + color.G + color.B) / 3);
            }
        }
        polars.Add(polar);
    }

    public int examsPolar(Bitmap bitmap)
    {
        float[] res = new float[polars.Count];
        PolarCoord examPolar = new PolarCoord(bitmap);

        for (int i = 0; i < examPolar.getColumns(); i++)
        {
            for (int j = 0; j < examPolar.getRows(); j++)
            {
                Color color = bitmap.GetPixel(i, j);
                examPolar.setPixel(i, j, (color.R + color.G + color.B) / 3);
            }
        }
        examPolar.make_exams_binary_map(teachPolar.getDecart().getVD(),
teachPolar.getDecart().getND());
        examPolar.make_polar_coord();
        examPolar.make_binary_map(teachPolar.getVD(), teachPolar.getND());

        for (int k = 0; k < polars.Count; k++)
        {
            res[k] = 0;
            for (int j = 0; j < examPolar.getMaxRadius(); j++)
            {

```

```
        int d_exam = 0;
        for (int i = 0; i < examPolar.getMatrixSize(); i++)
            d_exam += Math.Abs(polars[k].getElem_EV(i) - examPolar.getElem_polarBM(i,
j));
        res[k] += 1 - (d_exam / teachPolar.getRadius(k));
    }
    res[k] /= examPolar.getMaxRadius();
}
int k_max = -1;
double mu_max = -1;
for (int i = 0; i < polars.Count; i++)
{
    if (mu_max < res[i])
    {
        mu_max = res[i];
        k_max = i;
    }
}
return k_max;
}
}
```