

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

\_\_\_\_\_ 19 травня 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 - Комп'ютерних наук,

освітньо-наукової програми «Інформатика»

на тему: **«Інформаційна технологія персоналізованого діагностування  
ранньої стадії раку передміхурової залози»**

здобувача групи ІН.м-11н Савченко Тараса Руслановича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело.

\_\_\_\_\_ Тарас САВЧЕНКО  
(підпис)

Керівник,  
Доктор технічних наук, професор

Анатолій ДОВБИШ

\_\_\_\_\_ (підпис)

**Суми – 2023**

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»  
В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**  
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-наукової програми «Інформатика»  
здобувача групи ІН.м-1 ІІн Савченко Тарас Русланович

1. Тема роботи: «Інформаційна технологія персоналізованого діагностування ранньої стадії раку передміхурової залози»

затверджую наказом по СумДУ від «08» травня 2023 р. № 0475-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 19 травня 2023 року

3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_

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

1) Порівняльний аналіз підходів для діагностування онкопатологій передміхурової залози.

2) Формування словника патоморфологічних ознак. 3) Інформаційний огляд сучасних

методів діагностування онкопатологій за гістологічними зображеннями.

4) Проектування інтелектуальної системи для персоналізованого діагностування онкології передміхурової залози на ранній стадії поширення захворювання

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Порівняльний аналіз підходів для діагностування онкопатологій передміхурової залози</i>		
2	<i>Формування словника патоморфологічних ознак</i>		
3	<i>Інформаційний огляд сучасних методів діагностування онкопатологій за гістологічними зображеннями</i>		
4	<i>Проектування інтелектуальної системи для персоналізованого діагностування онкології передміхурової залози на ранній стадії поширення захворювання</i>		

Здобувач вищої освіти \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

**Записка:** 83 стор., 32 рис., 1 додаток, 29 джерел.

**Обґрунтування актуальності теми роботи.** Сучасні інтелектуальні компоненти, які базуються на методах комп'ютерного зору, ставлять перед собою задачу локалізації ділянок біоптату, які мають підозру на рак. Тим не менш цього не достатньо для повноцінного діагностування онкології, адже патоморфологічні ознаки характеризуються не аби якої комплексністю, що виражається в їх взаємопов'язаності. Навіть досвідчені онкологи стверджують, що при прийнятті класифікаційних рішень виникає розбіжність між експертами, приблизно в 30%. Оскільки, не зважаючи на загально відомі шаблони поширення раку, на нього можуть впливати біомаркери, які унікальні для кожного окремого організму. Тому постає задача розробки персоналізованої системи діагностування, яка автоналаштується під конкретного пацієнта.

**Об'єкт дослідження.** Слабоформалізований процес розпізнавання патоморфологічних об'єктів на гістологічних знімках передміхурової залози.

**Мета роботи.** Розробка інтелектуальної системи для автоматизованого діагностування з персоналізованим вхідним математичним описом.

**Методи дослідження.** Детерміновано-статистичні методи розпізнавання образів, математичний аналіз вхідних даних, інформаційно-екстремальна інтелектуальна технологія, графічні методи обробки зображення.

**Результати.** Розроблений програмний комплекс здатний напів-автоматизовано формувати алфавіт класів розпізнавання для ієрархічного класифікатору інформаційно-екстремальної технології. За результатами роботи вдалось розрізнити ранню стадію розвитку раку від гіперплазії простати.

ІНФОРМАЦІЙНА ІНТЕЛЕКТУАЛЬНА СИСТЕМА, БІОІНФОРМАТИКА,  
ПЕРСОНАЛІЗОВАНЕ ДІАГНОСТУВАННЯ РАКУ, КОМП'ЮТЕРНИЙ ЗІР,  
МАШИННЕ НАВЧАННЯ

## ЗМІСТ

ВСТУП .....	8
1. ІНФОРМАЦІЙНИЙ ОГЛЯД НАУКОВО-ТЕХНІЧНОЇ ПРОБЛЕМАТИКИ ГАЛУЗІ .....	10
1.1. Інформаційний огляд підходів для діагностування онкопатологій передміхурової залози. ....	10
1.2. Словник патоморфологічних ознак.....	15
1.3. Сучасний стан застосування інтелектуальних систем для діагностування онкопатологій передміхурової залози.....	22
1.4. Формалізована постановка задачі інформаційного синтезу інтелектуальної системи функціонального діагностування. ....	33
2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ.....	35
2.1 Основні положення інформаційно екстремальної технології. ....	35
2.2. Математичні моделі машинного навчання.....	37
2.3. Оцінка функціональної ефективності машинного навчання системи розпізнавання.....	39
2.4. Лінійний алгоритм машинного навчання інформаційно екстремальної технології для системи розпізнавання. ....	42
2.5. Інформаційно-екстремальне машинне навчання системи функціонального діагностування за ієрархічною структурою даних. ....	46
3.АЛГОРИТМІЧНО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДІАГНОСТУВАННЯ ОНКОЛОГІЇ ПЕРЕДМІХУРОВОЇ ЗАЛОЗИ.....	49
3.1. Формування вхідного математичного опису персоналізованої системи діагностування.....	49
3.2. Результати моделювання.....	52
3.3. Короткий опис програмної реалізації .....	58
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
ДОДАТОК.....	68

## ВСТУП

Попри значний технологічний розвиток, який здійснило людство за останнє століття, онкологічні захворювання все ще залишаються однією з найбільш смертельних хвороб. Більше того сучасна медицина здатна надати максимально ефективне лікування проти онкопатологій різного генезу, лише на ранніх стадіях її розвитку. Чим довше розвиваються пухлини в тілі людини, тим вища смертність серед хворих. При цьому не існує таких профілактичних засобів чи порад, які здатні звести до мінімуму вірогідність появи онкології. Тому виходить, що це захворювання, яке складно лікувати, має великий відсоток смертності й відсутні методи, які здатні запобігти появі патологій в організмі. Єдина адекватна профілактика – це регулярне діагностування.

Таким чином, серед чоловічої частини населення найчастіше діагностується рак передміхурової залози. І хоч при профілактичному обстеженні, у більшості випадках, достатньо неінвазійних методів, наприклад, магнітно-резонансна терапія органів малого тазу, яка допомагає виявити і локалізувати новоутворення, проте для подальшої діагностики цього не достатньо. Для того щоб повноцінно оцінити стан тканин та дати припущення стосовно майбутнього перебігу захворювання, виконується біопсія ураженої ділянки залози. Хоча отриманий біоптат містить достатню кількість патоморфологічних ознак, проте працювати з сучасними цифровими знімками тканини достатньо важко. Зокрема із-за надмірності інформації на них. Адже лікарю спочатку потрібно локалізувати ураження, а рак предміхурової залози розвивається мультицентрично, тобто поширюється у межах декількох областей, а вже потім сформувані припущення щодо природи самої онкології. Це потребує часу та не аби якої уваги зі сторони лікаря.

Саме тому сучасна цифрова мікроскопія передбачає наявність в онкологічних лабораторіях програмно-алгоритмічного забезпечення, яке спрощує частину роботи лікаря при аналізі гістологічних знімків. Найчастіше - це виконання локалізації ділянок зразка тканини, які мають підозру на рак. Тим

не менш є і такі системи, які здатні брати на себе частину завдань пов'язаних з діагностуванням. Наприклад, робити припущення щодо стадії онкологічного процесу або класифікації новоутворень. Відповідно подібні реалізації передбачають комплексне використання як алгоритмів графічного аналізу так й моделей і методів машинного навчання.

Однак головною науково-методологічною проблемою на шляху створення повноцінної автоматизованої системи діагностування є не можливість алгоритмічно відтворити в повній мірі когнітивні функції людського мозку. Варто розуміти, що діагностика не закінчується на локалізації ураження. Для повноцінного аналізу необхідно комплексно оцінити усі наявні патоморфологічні ознаки, які при цьому можуть змінюватися під дією індивідуальних біомаркерів конкретного організму. Навіть серед лікарів часто зустрічаються розходження стосовно діагнозу. Щоб підвищити достовірність класифікаційних рішень системи необхідно наділити її властивістю персоналізації. Тобто переналаштування для конкретного організму.

Оскільки найбільш цінними результатами ідентифікація є знаходження онкології на ранній стадії, то необхідно, щоб система могла без помилково визначати її. Річ у тім, що при деяких умовах гістологічні зображення раку можуть мімікрувати під доброякісні новоутворення або навпаки. Тобто можливі ситуації пропущення онкології чи призначення не доцільного лікування пацієнту. Отже розрізнення патологій за рівнем їх небезпечності є однією з важливих характеристик інтелектуальної системи діагностування.

Таким чином під час дослідження необхідно розробити програмний комплекс, який здатний виконувати персоналізований підбір вхідного математичного опису, що дозволить навчити систему розпізнавати патоморфологічні ознаки конкретного пацієнту. Окрім цього буде зосереджена увага на ранніх стадіях онкології, тому необхідно забезпечити інтелектуальну компоненту модулем розрізнення раку від аденоми чи гіперплазії передміхурової залози.

# 1. ІНФОРМАЦІЙНИЙ ОГЛЯД НАУКОВО-ТЕХНІЧНОЇ ПРОБЛЕМАТИКИ ГАЛУЗІ

## 1.1 Інформаційний огляд підходів для діагностування онкопатологій передміхурової залози.

Головною задачею при діагностуванні онкопатологій є їх своєчасне попередження через виявлення клінічно значущих уражень тканин на ранніх стадіях розвитку новоутворень. При цьому важливо як оцінити ступінь захворювання так й передбачити майбутні ускладнення, що у свою чергу, дозволяє визначити рівень ризику прогресування, який напряду впливає на тип лікування пацієнта.

Одним із найрозповсюдженіших, неінвазивних, візуальних методів діагностичного обстеження раку передміхурової залози є магнітно-резонансна томографія (МРТ) органів малого тазу (рис.1.1), яке використовується для прямої локалізації уражень тканини. Це дозволяє з великою точністю виявити злоякісні новоутворення на різних стадіях патологічного процесу задовго до появи першої клінічної симптоматики. Ряд особливостей, таких як: швидкість, доступність та універсальність. Робить МРТ одним із найпопулярніших методів неінвазивного діагностування. При цьому як для первинної локалізації новоутворень, так й для подальшого моніторингу перебігу захворювання.



Рисунок 1.1. МРТ органів малого тазу

Хоч МРТ органів малого тазу (рис.1.1) є достатньо розповсюдженим варіантом скринінгу онкології, але він надає достатньо малу кількість інформації

про поточний стан хвороби. Єдине, що вдасться зробити за результатами такого діагностування – це локалізація новоутворення в передміхуровій залозі. Саме тому сучасні рентгенологічні дослідження раку простати проводяться за допомогою мультипараметричної магнітно-резонансної томографії (мпМРТ) [7, 8], яка зводиться до цілеспрямованої візуалізації лише одного органу. У нашому випадку передміхурової залози, яку досліджують за допомогою наступних рівнів візуального аналізу:

- **Дифузійно-зважене зображення (DWI)** з видимими коефіцієнтами дифузії, за допомогою яких вимірюється рухливість молекул води внаслідок броунівського руху. Загалом для раку передміхурової залози характерна підвищена щільність клітин, яка на DWI-зображеннях має вигляд зон з локально обмеженою дифузією вільної води. Візуально це виглядає як біла пляма. Окрім безпосередньої локалізації новоутворення можна використати знання про дифузію для визначення поточного стану хвороби.

- **T2-зважене зображення (T2W).** Використовується для окреслення анатомічно нормальних осередків передміхурової залози. За допомогою локальної води можна чітко продемонструвати перехідні та периферичні зони. Візуалізація T2W виявляє рак як фокальні області помірно низької інтенсивності сигналу.

- **Динамічне внутрішньовенне контрастне підсилення (DCE).** Для отримання цієї візуалізації необхідно ввести внутрішньовенно пацієнту спеціальний препарат гадолініум-ДТРА (діетилентріамінпентна оцтова кислота). Отриманні результати мпМРТ будуть відображати місцеве судинне оточення та корисні для оцінки васкуляризації в певних областях простати. Для раку передміхурової залози характерні ділянки аномальної неоваскуляризації, які можна відобразити шляхом аналізу змін, таких як раннє артеріальне посилення.



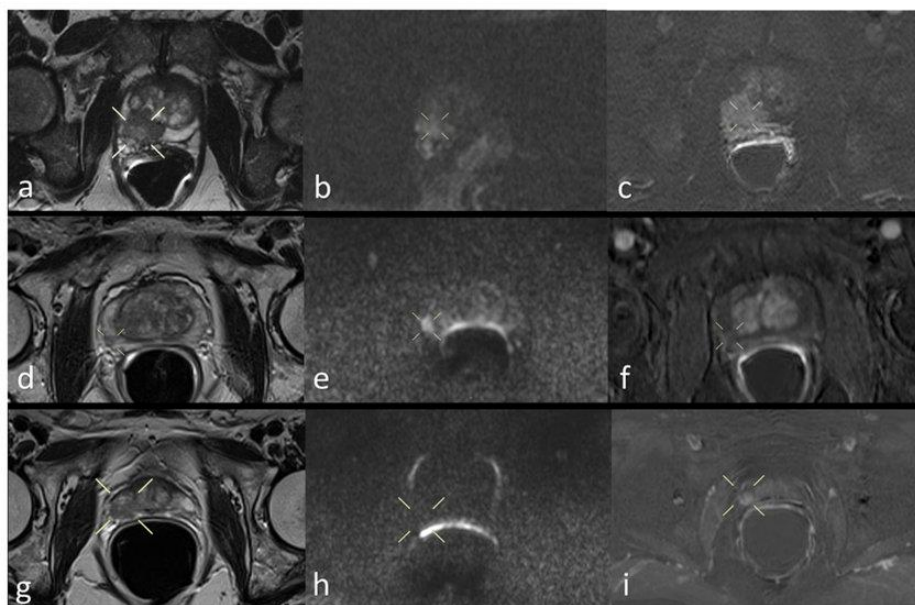


Рисунок 1.3. мпМРТ передміхурової залози для трьох пацієнтів [9]:

a-b) T2W, DWI, DCE приклад раку простати; d-f) T2W, DWI, DCE приклад гіперплазії простати; g-i) T2W, DWI, DCE приклад хибно негативного пацієнта;

Далеко не очевидною перевагою мпМРТ є мінімальне втручання людського фактору. Тобто можливо створити уніфікований, детермінований словник ознак з відповідними параметрами, які будуть однозначно характеризувати рак на знімках, навіть якщо вони зроблені в різних лабораторіях. «Система звітності та даних про зображення передміхурової залози», також відома під аббревіатурою PI-RADS [10], це міжнародна група, яка займається стандартизацією продуктивності МРТ простати. По своїй суті це система, яка містить характеристики та оцінки всіх можливих осередків внутрішньозалозистих вузлів, виявлених на знімках МРТ. Окрім цього враховуються технічні стандарти для апаратного забезпечення, сканування та протоколи для отримання та інтерпретації зображень.

Хоч мпМРТ має свої переваги, насамперед для довгострокового моніторингу перебігу захворювання, але це не найкращий спосіб для встановлення поточного стану хвороби. МРТ не може передати індивідуальні біологічні маркери, які здатні вплинути на розвиток новоутворень. Тому для більш точного аналізу та прогнозування використовується наступний алгоритм:

- 1) Аналіз крові на простат-специфічний антиген (PSA);
- 2) У випадку, коли після проведення PSA є підозри на рак, робиться магнітно-резонансна томографія для локалізації ураження простати;
- 3) Після визначення зони інтересу виконується біопсія з подальшим аналізом ураженої ділянки методами оптичної мікроскопії;

У наш час набирає популярності метод, так званої, повнослайдової цифрової мікроскопії (whole-slide-imaging, WSI) [11]. Ця технологія передбачає зйомку усієї тканини, розміщеної на предметному склі, результуюче зображення якої має гігапксельну роздільну здатність. Патоморфолог оцінює WSI зображення виділяючи здорову та аномальну тканину, аналізуючи будову клітин. У свою чергу це дозволяє призначити пацієнту більш персоналізоване лікування. Тим не менш, на результат аналізу біопсії впливає людський фактор. Вважається, що присутня розбіжність у розмірі 30% при маркуванні патологоанатомами ракових ділянок.

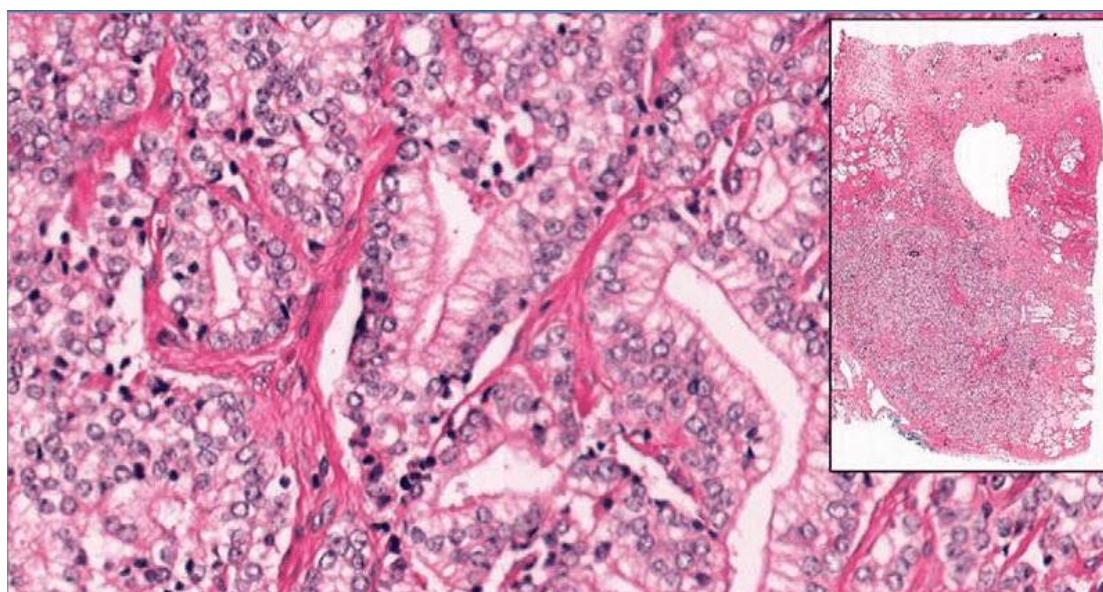


Рисунок 1.3. Приклад повнослайдового зображення

Основною перевагою WSI над класичним підходом оптичної мікроскопії, де лікар власноруч виконує аналіз через мікроскоп – є можливість реалізувати високоефективне сканування за рахунок швидкої локалізації зони інтересу тканини. Окрім цього будь-який лікар може запросити консультацію у своїх колег надавши їм відповідне повнослайдове зображення. Але все таки головним

плюсом WSI є їх висока інформативність ознак розпізнавання, які дозволяють поставити правильний діагноз та призначити індивідуальний план лікування.

Незважаючи на очевидні переваги, WSI - це все ще достатньо нова технологія і далеко не всі лабораторії згодні її використовувати. Хоч повнослайдові зображення і здатні пришвидшити та підвищити точність діагностування онкологій. Тим не менш їх візуальний аналіз є достатньо трудомістким, адже містить в собі велику кількість гістологічних ознак. Тому для подолання цієї проблеми варто використовувати програмне забезпечення, яке базується на принципах машинного навчання та розпізнавання образів у вигляді системи автоматизованої діагностики.

Спільним недоліком будь-якого методу класичної мікроскопії є порушення структури тканини взятої під час біопсії, що призводить до втрачання об'єму зразка. Більше того онкологія має ряд специфічних мета-ознак, які подекуди складно побачити на двовимірному зрізі ураженої тканини, що, у деякому сенсі, являє собою лише проекцію новоутворення. У такому випадку складно оцінити справжню її форму, розмір осередку раку та, наприклад, щільність. А скажімо звивисті структури, такі як лімфатичні судини чи залози взагалі не можливо точно зафіксувати в двовимірному зрізі, хоча їх стан може бути предикатом чи малої кількості захворювань. Для подолання цих проблем пропонується використання методів світло-листової мікроскопії, що передбачає комплексний трьохвимірний гістологічний аналіз тканини.

На відміну від традиційних методів флуоресцентної мікроскопії, які зазвичай використовують лише одну лінзу об'єктива для спостереження зразка тканини на предметному склі, світло-листовий підхід передбачає два об'єктива і роздільні шляхи освітлення зразка. Помістивши ці дві лінзи під кутом  $90^\circ$  можливо використовувати тонкий шар світла для оптичного, а не фізичного перерізу біоптату. Копланарне вирівнювання світлового листа по відношенню до фокальної площини об'єктива, забезпечує вибіркоче освітлення зразка. Це відрізняється від традиційних методів флуоресцентної мікроскопії, в яких часто

висвітлюється вся тканина для збору інформації лише з однієї точки або площини всередині зразка (рис.1.4)

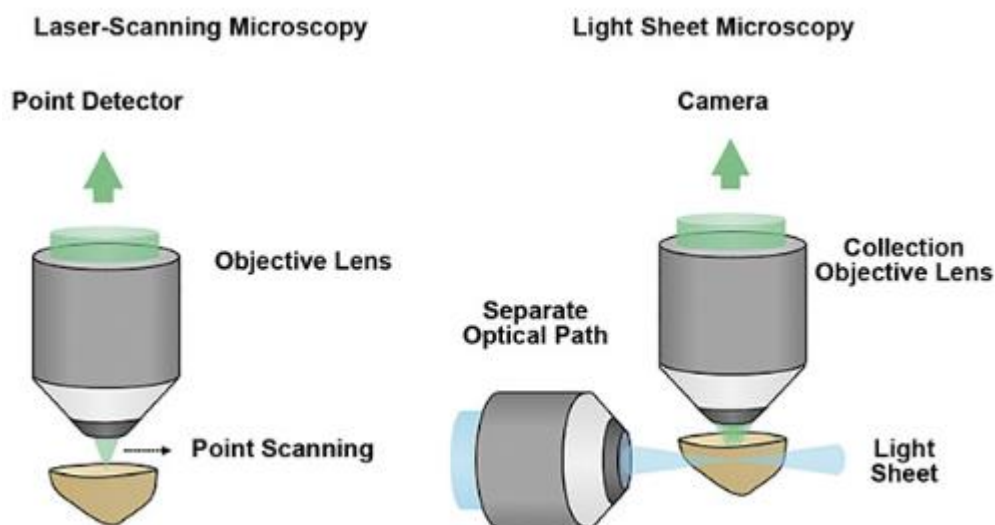


Рисунок 1.4. Схематичне порівняння точкової і лазерної мікроскопії [12]

Однозначно можна сказати, що перехід до трьохвимірного аналізу біоптатів має свої переваги. І найвагоміший з них – це багатократне збільшення інформативних ознак, що дозволяє набагато комплексніше виконувати діагностування. Проте цей підхід ще потребує не аби якого доопрацювання. Необхідне більш досконале обладнання й потрібні специфічні протоколи хімічної очистки біоптатів. Але найголовніше - потрібна достатня кількісна оцінка переваг трьохвимірного діагностування патологій в порівнянні з класичними методами.

## 1.2 Словник патоморфологічних ознак.

Для гістологічного прогнозування раку передміхурової залози лікарями патоморфологами використовується уніфікована шкала Глісона [13]. Яка дозволяє за допомогою суми деяких оцінок ураженої тканини прогнозувати подальший перебіг захворювання. Річ у тім, що онкологія передміхурової залози є не однорідною і досить часто містить більше однієї онкологічної ознаки. Тому для оцінки тканини обирається дві найбільш характерні ділянки. У випадку, коли вона дуже диференційована, то їй призначається 1 бал, у протилежному випадку 5. Таким чином сума Глісона знаходиться у межах  $1+1=2$  та  $5+5=10$ .

Візуально перший та другий рівень шкали Глісона достатньо схожі. Для першого ступеню диференційованості характерні добре окреслені вузли з щільно розташованими, рівномірно розподіленими клітинами. Зазвичай вони мають не великі розміри. Для порівняння, вже на третьому рівні клітини об'єднуються в значно більші структури. Також для перших рівнів характерна більш бліда цитоплазма залоз. Це важливо враховувати при ідентифікації, бо потрібно опиратися більше не на колір, а на форму. Найчастіше тканини цього зразка зустрічаються в передній частині простати перехідної зони.

На другому рівні клітини все ще зберігають невеликий розмір з мінімальною інфільтрацією по краях. Проте вони вже менш однорідні і далеко не так компактно розташовані. Що стосується інших ознак, то вони тотожні першому ступеню. Як правило, на практиці лікарі не вважають диференціювання, що зустрічаються на цих рівнях небезпечним.

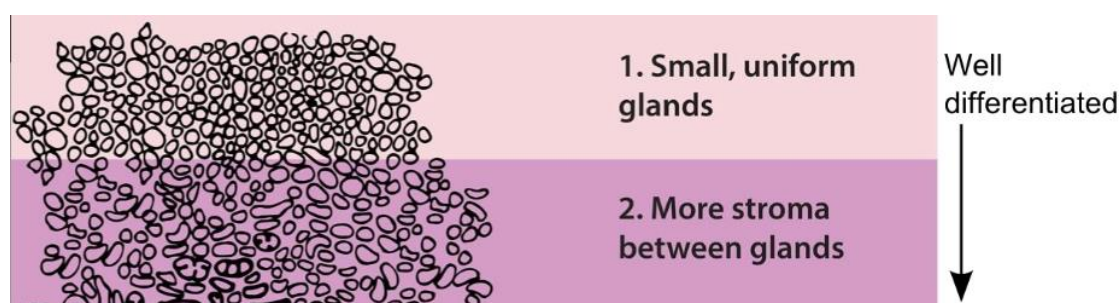


Рисунок 1.5. Шкала Глісона першого та другого рівня

Все що відповідає диференційованості вище другого рівня потребує уваги патоморфолога. Починаючи з третього для залоз стає характерна неоднорідність за розміром. Більшість з них невеликі, але з чітко виявленими контурами. При цьому можливий інфільтративний ріст в здорові тканини. На третьому ж рівні диференційованості варто звертати увагу на те, що клітини, у більшості випадках, не зростаються. Тобто вони деформуються, але не об'єднуються.



Рисунок 1.6. Шкала Глісона третього рівня

У деякому сенсі саме третій рівень диференційованості тканини є найбільш важливим при діагностуванні, адже він відповідає ранній стадії раку. Окрім цього він містить ознаки, які подібні до гіперплазії простати, тобто доброякісного новоутворення, що може вносити деяку невизначеність для діагностування. Четвертий та п'ятий рівень диференційованості є подібними між собою. Насамперед із-за того, що вони відповідають прогресуючій онкології і візуально достатньо легко визначаються, адже на цьому етапі більшість тканини буде враженою. Так для четвертого рівня характерне зростання клітин в великі конгломерати, які характеризуються не рівними контурами та щілиноподібними просвітами. П'ятий рівень передбачає, що вся ділянка вражена злоякісними новоутвореннями з, можливо, рідкісними вкрапленнями здорової тканини. Проте, що цікаво, уражені залози можуть бути досить однорідними, не зважаючи на п'ятий рівень диференційованості.

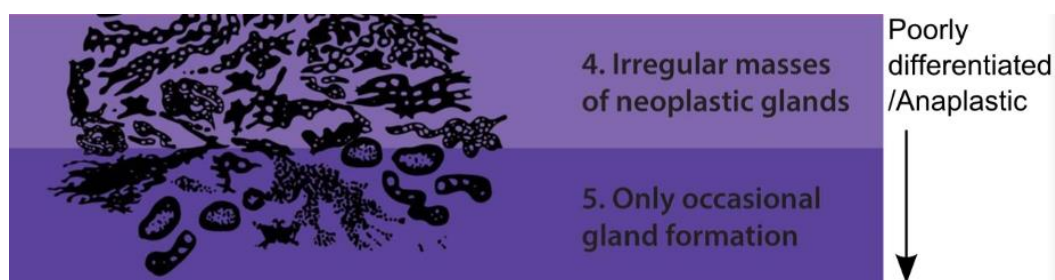


Рисунок 1.7. Шкала Глісона четвертого і п'ятого рівня

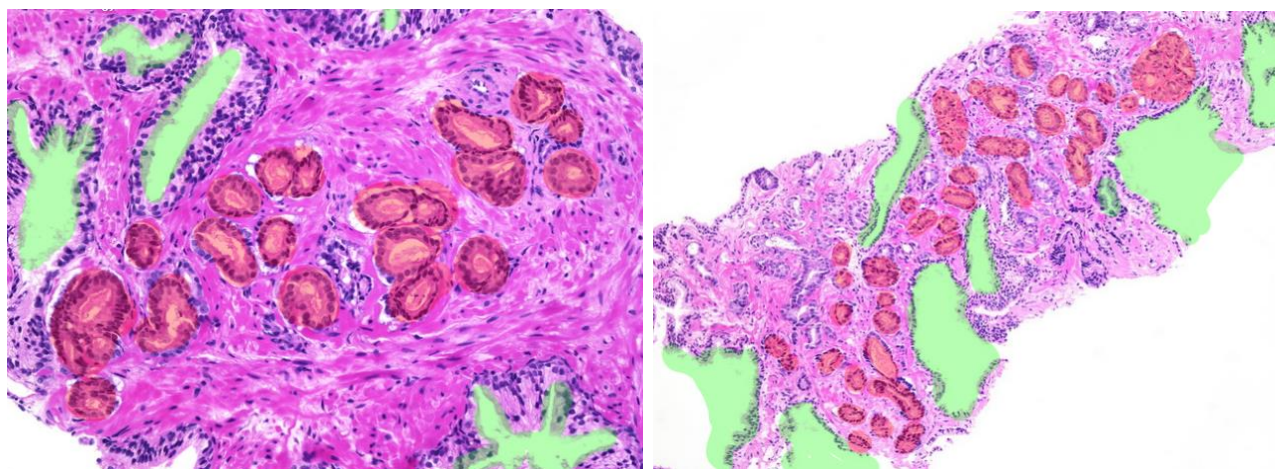
У більшості випадках, первинний аналіз біопсії передміхурової залози проводиться при малому збільшенні зразка тканини. Річ у тім, що ракові залози мають дещо спрощену структуру та формують чітко окреслені осередки овальної чи круглої форми. На зображенні 1.8.а виділенні якраз такі скупчення (червоний колір), які досить помітно контрастують з оточенням. Таким чином доброякісні тканини мають складну вузлувату структуру і папілярний тип розростання (зелений колір).

Невеликі та атипичні за формою залози, які знаходяться між доброякісними вузлуватими тканинами свідчать про інвазію і фактично це є діагностичною ознакою злоякісного новоутворення. Так на рисунку 1.8.б є

великі осередки такого руху клітин у сполучній тканині. При подальшому збільшенні повнослайдового зображення можна помітити, що злоякісні утворення характеризуються не тільки специфічною формою, але й помітною гіперхромією та збільшенням ядер в середині клітин. Виділяються внутрішні ядерця із-за просвіту муцину та відсутності зовнішнього шару базальних клітин. У свою чергу, це є достатньо розповсюдженою візуальною ознакою онкології, і вона часто використовується при локалізації уражень в тканині.

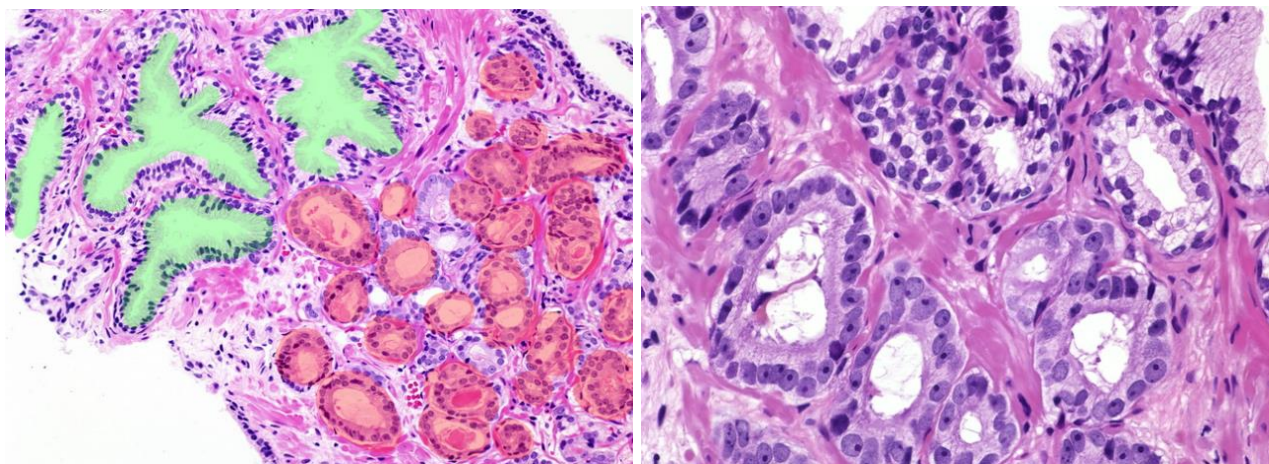
Злоякісні утворення перебувають у скупченнях, що також можна використати як гістологічну ознаку. На малюнку 1.8.в помітно, що доброякісні залози мають складний контур. У той же час в правій частині біоптату можна побачити скупчення маленьких злоякісних утворень, які знаходяться щільно один до одного.

Як зазначалося раніше, то ядерця, які можна помітити в тканинах уражених онкологією є досить важливою, але не стовідсотковою патоморфологічною ознакою. Наприклад, на малюнку 1.8.г можна побачити зліва макронуклеоли, які відсутні в клітинах в правій стороні зображення. Проте на подібне може вплинути ряд факторів, які потрібно враховувати при діагностиці, бо ядерця можуть маскуватися в злоякісних клітинах, або навпаки бути добре помітними в доброякісних.



а

б



В Г  
Рисунок 1.8. Набір гістологічних ознак [14]

На рисунку 1.9.а зібрано усі необхідні гістологічні ознаки для ідентифікації раку. Невеликі скупчення залоз, збільшенні ядра клітин з їх гіперхромією, добре помітні ядерця та внутрішньопросвітний муцин. Для порівняння можна подивитися на частково здорову залозу, яка помітна в правому нижньому кутку. Власне її ядра на порядок менші ніж у злоякісній.

Однією з ознак раку, яка характерна виключно для передміхурової залози, є периневральна інвазія (рис.1.9.б). Це злоякісне новоутворення, яке оточує нерв (жовтий колір) по його периметру. Ця ознака проявляється приблизно у 20% біопсій і є одним із способів розповсюдження онкології за межами передміхурової залози на перепростатичні м'які тканини. Вважається, що периневральна інвазія може свідчити про позапростатичне розповсюдження раку, але це піддається сумніву. На рисунку 1.9.в зображено скупчення клітин, які оточують два нерви.

Еозинофільні кристалоїди (рис.1.9.г) є досить слабкою ознакою раку передміхурової залози. Вони можуть приймати голко подібну, багатогранну або неправильну форму. Зазвичай їх пов'язують з злоякісними новоутвореннями, але це не однозначно, бо ця ознака не специфічна для аденокарциноми. Проте якщо вони присутні в тканині, то це може свідчити про високодиференційований рак. Приблизно в проміжку між 2 до 6 по шкалі Глісона.



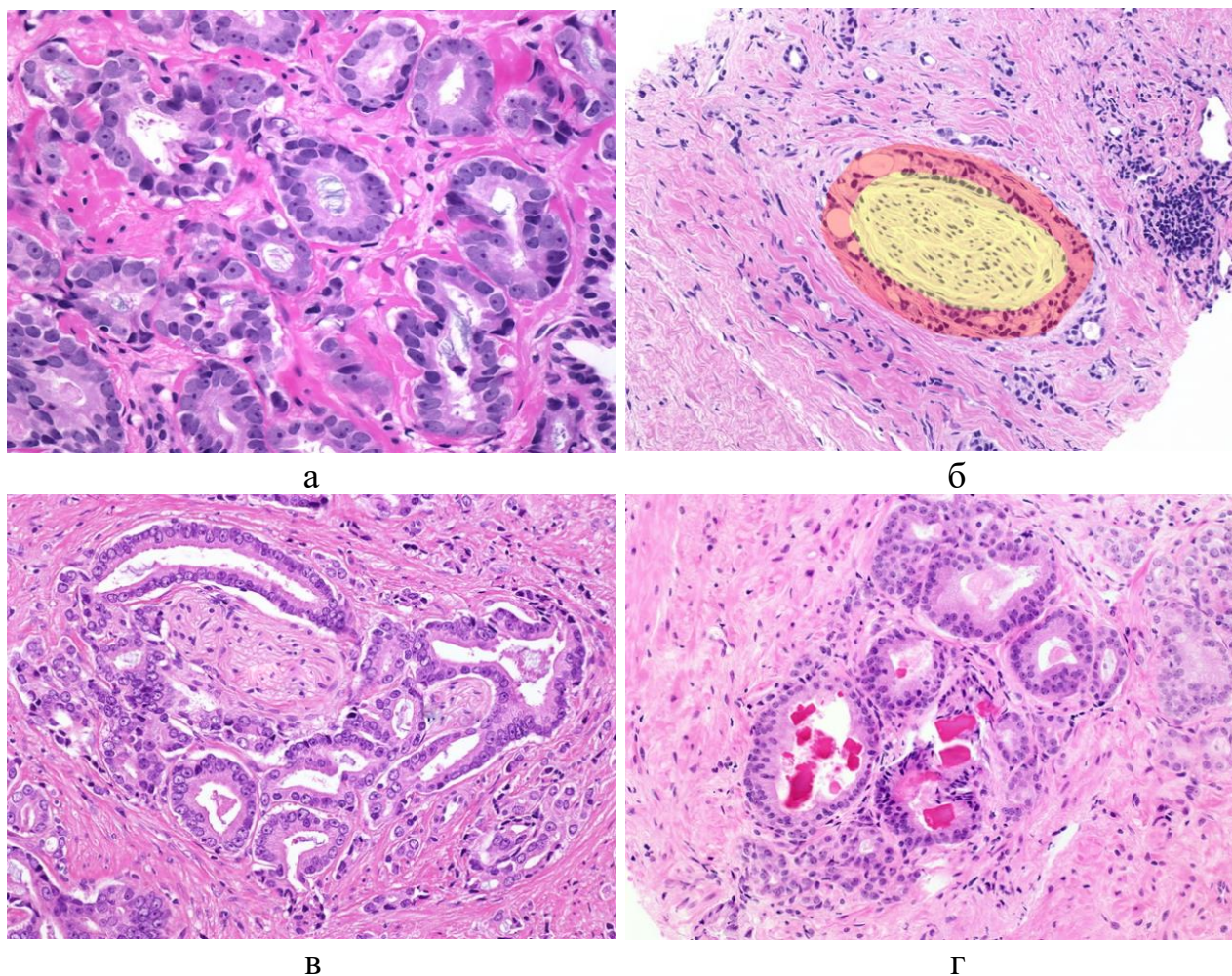


Рисунок 1.9. Набір гістологічних ознак [14]

Також однією з переліку досить не однозначних гістологічних ознак є внутрішньопросвітний синій муцин (рис.10.а). По-перше, його помітність залежить від конкретної лабораторії, тому не на всіх знімках його можна побачити. По-друге, він не характерний для раку предміхурової залози і може бути виявлений й в доброякісних утвореннях.

Характерною ознакою раку саме простати є колагенові мікрівузли (рис.1.10.б), які продукують велику кількість муцину. Вони нагадують павутину та присутні в середині та навколо злоякісних новоутворень. Частіше за все спричиняють видозмінення їх форми. Взагалі для раку притамані архітектурні ознаки, наприклад, карцинома передміхурової залози (рис.1.10.в). Вона створена з сукупності ракових клітин, які на просвіт виглядають як нирка або полумісяць.

Рак простати є мультицентричним / мультифокальним новоутворенням. Тобто, розповсюдження виконується одразу з декількох центрів. Мультифокальність реєструється в 60-100% випадках. Зазвичай є 2 або 3 окремих осередка ракових клітин. Проте вони не завжди розростаються рівномірно. Так, наприклад, на зображенні 1.10.г справа нормальна архітектура тканини майже повністю стерта домінуючим центром онкології

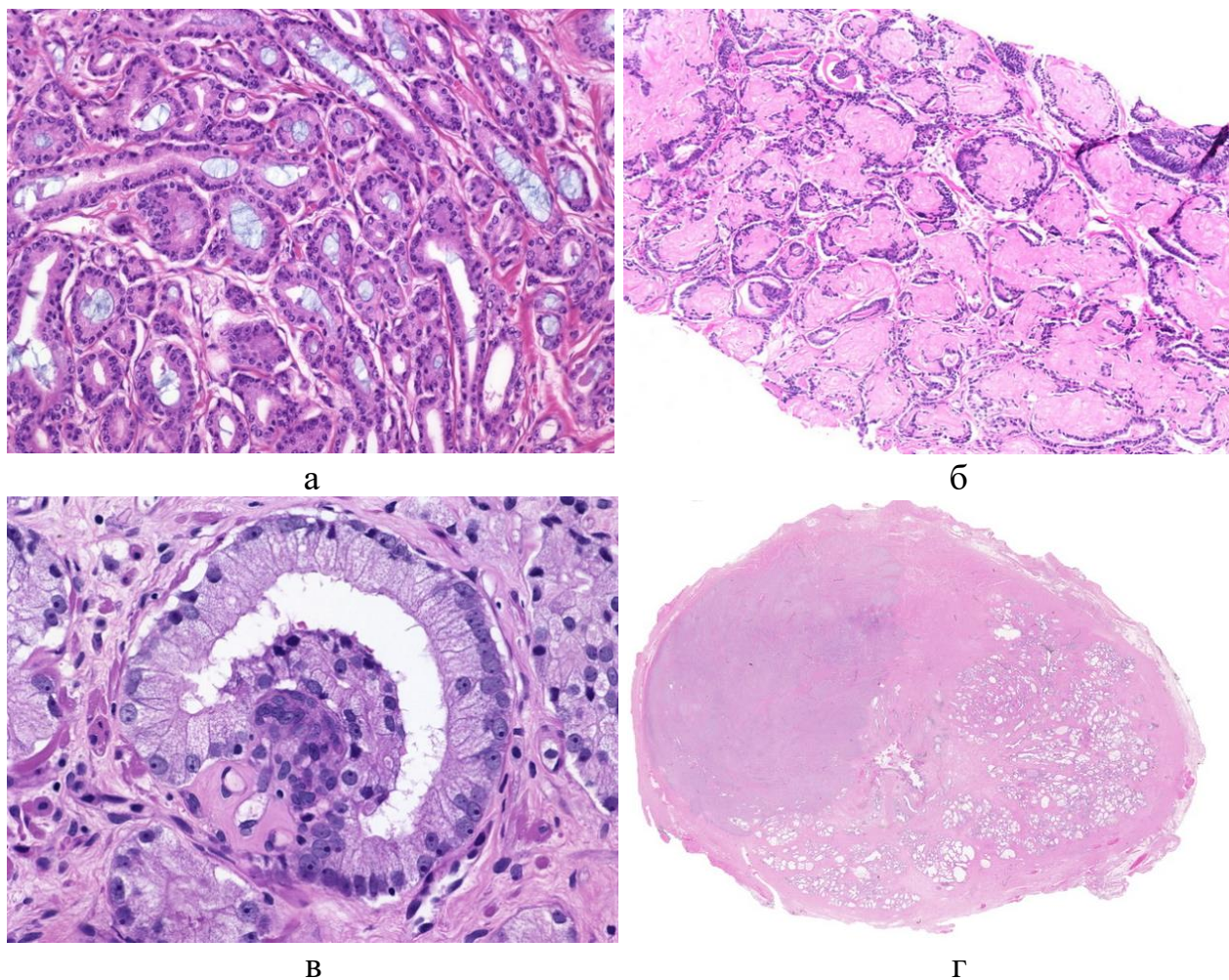


Рисунок 1.10. Набір гістологічних ознак [14]

Отже, найбільш візуально помітною патоморфологічною ознакою є гіперхромія ядер, оскільки цей маркер зустрічається майже в кожному гістологічному знімку, де є патологія. Тим не менш, цього не достатньо для діагностування. Необхідно визначити природу ураження, адже гіперхромія характерна не лише для онкології, а ще й для доброякісних новоутворень. Як додаткові ознаки можна використати розмір, форму, щільність скупчення залоз або інші структурні особливості тканини.

### 1.3 Сучасний стан застосування інтелектуальних систем для діагностування онкопатологій передміхурової залози.

На сьогоднішній день існує декілька принципово різних підходів для реалізації інтелектуальної системи в рамках автоматизованої діагностики онкологій. Так класифікацію зображення тканини можна розглядати як процес локалізації ділянок, які вражені злякисними новоутвореннями. Більшість інтелектуальних систем діагностування (ІСД), які описані в сучасній науковій літературі, можна розділити на три абстрактних класи:

- *Системи напівавтоматизованої підтримки рішень.* По своїй суті алгоритми цієї групи здатні спрощувати роботу оператора, але самостійно вони не приймають класифікаційні рішення. Таким чином сюди можна віднести більшість методів обробки зображення, які сегментують його, знаходять контури, тощо. Фактично ці системи не є інтелектуальними, адже в них відсутній етап прийняття рішень й очевидним недоліком цього підходу є повна залежність від оператора, який повинен виконувати налаштування, перетворення зображення та оцінювати отриманий результат. Окрім цього ці системи характеризуються високою вірогідністю прийняття помилки першого роду, що обумовлює невиправдане лікування, або помилки другого роду - пропуск злякисного новоутворення на ранній стадії його розвитку.

- *Системи підтримки прийняття рішень з автоматизованими компонентами.* Це реалізації, які базуються на методах не глибокого машинного навчання, які при цьому здатні автономно приймати класифікаційні рішення в межах чітко поставленої підзадачі. Як приклад можна привести первинну локалізацію новоутворення на знімку МРТ. Тобто система визначає зону інтересу, але подальшу ідентифікацію та верифікацію вже виконує оператор. Алгоритми цього класу, такі як: випадковий ліс, метод опорних векторів, найпростіші нейронні мережі і т.д., можуть приймати в якості вхідних даних геометричні параметри попередньо сегментованих об'єктів методами графічної обробки зображення.

• **Системи автоматизованого діагностування.** У ідеалістичному вигляді – це програмне забезпечення, яке передбачає автономне діагностування пацієнта. Звісно, що в повноцінному вигляді, поки що, це не можливо реалізувати по ряду науково-методологічних причин, але передбачається, що така система буде використовувати алгоритми глибокого машинного навчання. При цьому обробка вхідних даних повинна проводитися максимально комплексно, щоб наблизити цей процес до когнітивних здібностей людини.

В незалежності від типу ІСД для подібних систем є характерною попередня обробка інформації для виокремлення найбільш інформативних ознак вхідного зображення. Видалення фону, візуального зашумлення та підвищення контрастності об'єкту інтересу є ключовими процесами, які варто враховувати при роботі з гістопатологічними зображеннями.

Одним із розповсюджених методів бінеризації та вилучення фону з WSI зображень [15, 16] є метод Оцу [17]. Ідея якого полягає у розділенні гістограми яскравості на два кластери з пороговим значенням, яке визначається за рахунок мінімізації зваженої дисперсії цих класів:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t), \text{ де}$$

$\omega_1(t), \omega_2(t)$  – вірогідності класів, які розділені порогом  $t \in [0, 250]$ ;

$\sigma_1^2(t), \sigma_2^2(t)$  – дисперсії класів, які розділені порогом  $t \in [0, 250]$ ;

У своїй роботі [17] автор аргументує існування двох варіантів знаходження порогу між кластерами. Перший – це мінімізація  $\sigma_w^2(t)$ . Другий – це максимізація дисперсії між класами:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2,$$

$$\mu(t) = \sum_{i=1}^t \frac{iP(i)}{\omega(t)},$$

$$\sigma_b^2(t) = \sum_{i=1}^t [i - \mu_b(t)]^2 \frac{P(i)}{\omega_b(t)}$$

Тобто у своїй основі ідея достатньо проста. Необхідно знайти розбиття між шумом та інформативними об'єктами на зображенні. У рамках патоморфології [16] це може використовуватися для виокремлення залоз в сполучній тканині. Тобто локалізації можливого осередку онкології.

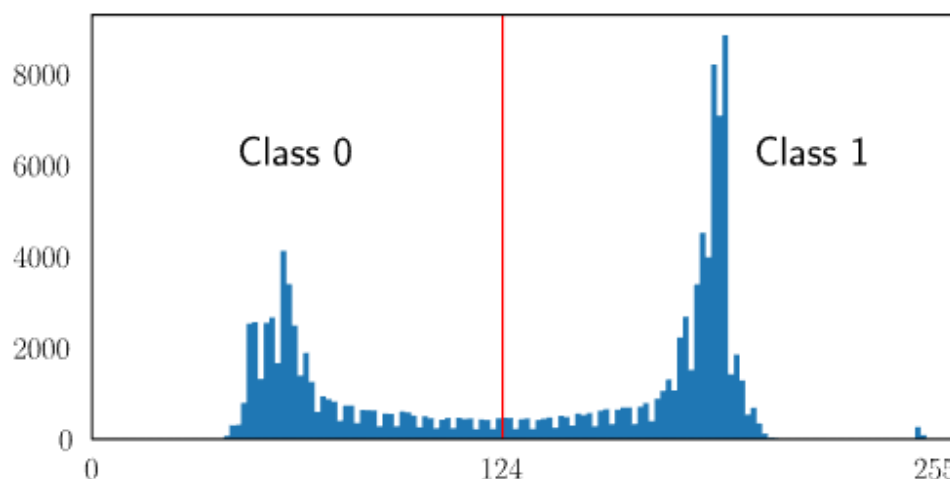


Рисунок 1.11. Гістограма яскравості розбита на два класи за методом Оцу

Взагалі проблема локалізації зони інтересу на WSI зображеннях є надактуальною. По-перше, цей процес сам по собі допомагає у роботі патогістолога виділяючи підозрілі на рак ділянки. По-друге, для навчання будь-якої інтелектуальної системи необхідна апостеріорна інформація. У WSI це зони з найбільш інформативними для нас візуальними ознаками.

Статистично приблизно 80% WSI зображень не містить корисної візуальної інформації про стан тканини. Обумовлено це специфікою отримання цих знімків, де більшу частину займає предметне скло, а в самому зразку осередки онкології мають мультицентричну природу у вигляді невеликих крапель в сполучній тканині. Звісно, якщо це не пізня стадія раку, коли вражена вже вся тканина. Один із варіантів визначення зони інтересу патоморфологічних ознак - це метод Blue Ratio [18]. У цифровій патології цей алгоритм використовується для локалізації ядер в клітинах, які приймають синє забарвлення при додаванні до зразка тканини препарату гематоксиліну та еозину (H&E), тоді як сполучна тканина набуває рожевого або червоного кольору. Blue Ratio зображення розраховується за наступною формулою:

$$BR = \frac{100 * B}{1 + R + G} * \frac{256}{1 + B + R + G}, \text{ де}$$

R, G, B – це червоний, зелений і синій канал RGB;

На практиці Blue Ratio використовується як маска для основного зображення, адже якщо придивитися до формули BR, то перша частина там відповідає за відтінок, а друга інтенсивність знімку. Результуюче зображення можна допрацювати графічними операціями по типу ерозія чи дилатація.

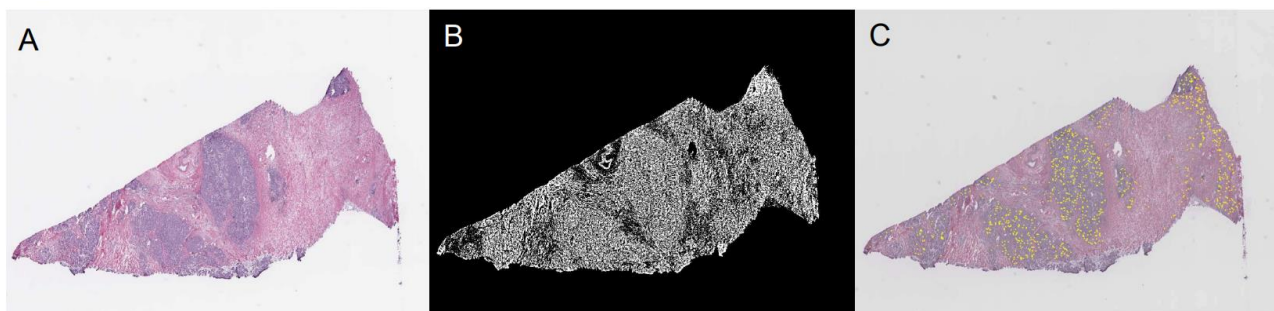


Рисунок 1.12. Приклад використання Blue Ratio: а) початкове WSI; б) бінеризація тканини за допомогою Blue Ratio; с) зона інтересу локалізована жовтим кольором;

Також для виділення зони інтересу у вигляді гіперхромних ядер на гістологічному зображенні використовується метод невід'ємної матричної факторизації (NMF) [22]. Цей алгоритм базується на припущенні, що зображення можна розкласти на інформативні частини. Нехай, вхідна матриця  $V$  форми  $mn$  буде розкладена на  $W$  і  $H$  з розмірністю  $mk$  і  $kn$  відповідно, а елементи  $W$  і  $H$  є додатними лише коли  $V$  додатна.

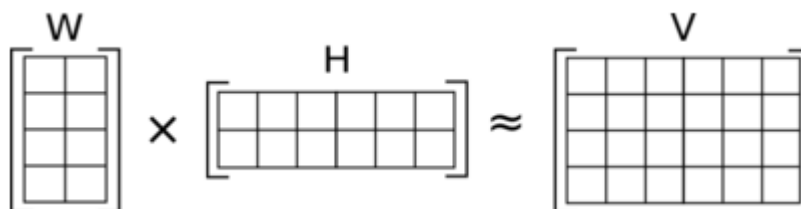


Рисунок 1.13. Приклад матриці NMF

У деякому сенсі NMF можна вважати алгоритмом машинного навчання без учителя. Оскільки метод передбачає, що результуюча матриця  $W$  буде

містити в собі ознаки вхідного зображення, а  $H$  найбільш інформативні з них, на думку системи. Для цього необхідні метрики, які допоможуть обчислити відстань між елементами.

Один із розповсюджених інформаційних критеріїв, що зустрічаються в практиці машинного навчання, є узагальнена дивергенція Кульбака–Лейблера [23]. По своїй суті це статистична міра, яка надає кількісну оцінку того наскільки один розподіл відрізняється від іншого. По мірі наближення значення розбіжності Кульбака – Лейблера до нуля, близькість відповідних ознак збільшується.

$$D_{KL}(x, y) = \begin{cases} x * \log\left(\frac{x}{y}\right) - x + y & x > 0, y > 0 \\ y & x = 0, y \geq 0 \\ \infty & \text{otherwise} \end{cases}$$

Також можна зустріти реалізації NMF в поєднанні з нормою Фробеніуса. Більше відомою під назвою; евклідова норма. Це квадратний корінь із суми абсолютних квадратів його елементів. Можна вважати це достатньо популярним способом визначення наскільки вдалою є апроксимація.

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}$$

Враховуючи усю надану теоретичну інформацію про NMF необхідно формалізувати її в одну цільову функцію, яка здатна знайти оптимальні ознаки для матриць  $W$  і  $H$ . Нам потрібно обчислити помилку апроксимації:

$$\frac{1}{2} \|A - WH\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m (A_{ij} - (WH)_{ij})^2$$

У процесі оптимізації ми будемо знаходити оптимальні, в інформаційному розумінні, матриці  $W$  і  $H$ , які на кожному кроці ітерації будуть змінюватися за наступними правилами:

$$W_{ic} \frac{(AH)_{ic}}{(WHH)_{ic}} \rightarrow W_{ic}, \quad H_{cj} \frac{(WA)_{cj}}{(WWH)_{cj}} \rightarrow H_{cj}$$

Тобто процес підбору оптимальних матриць  $W$  і  $H$  полягає в паралельному оновленні їх елементів. Простіше кажучи, метод буде змінювати початкове наближення до тих пір, доки добуток  $W$  і  $H$  не наблизиться до  $A$ . У свою чергу це можливо лише або при сходженні помилки апроксимації, або при досягненні максимальної кількості ітерацій.

Взагалі запропонована реалізація NMF є далеко не єдиною. Можна, наприклад, ініціалізувати матриці  $W$  і  $H$  випадковим чином, але тоді початкове наближення буде не передбачуване так само як і час сходження цільової функції. У таких випадках пропонується виконувати початкову кластеризацію вхідних даних, для спрощення процесу навчання. Або додати додатковий квантор оптимізації, наприклад, сингулярний розклад матриці (SVD).

Розглянемо NMF на прикладі. Припустимо необхідно визначити ознаки деякого зображення з обличчям, яке складається з  $p$  пікселів. Групуємо дані в один вектор, таким чином, щоб  $i$ -й запис відповідав  $i$ -му пікселю. Тоді  $X \in R^{pn}$  представляє  $p$  пікселів, а кожен із  $n$  стовпчиків є представленням свого зображення-ознаки.

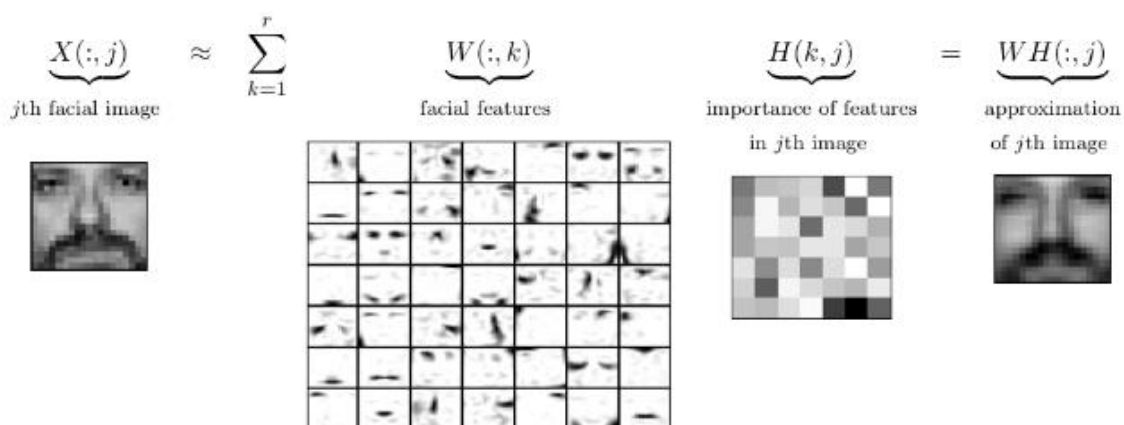


Рисунок 1.14. Приклад декомпозиції зображення за допомогою NMF

Взагалі графічних алгоритмів локалізації об'єкту інтересу є неймовірна велика кількість. Кожен з них працює у власному кольоровому діапазоні й з різною ефективністю підходить для своїх специфічних умов використання. У



рамках патоморфології однією з найпомітніших візуальних ознак - є гіперхромні ядра, які мають темно синій колір й виділяються на фоні сполучної тканини. На рисунку 1.15 зображено приклад попередньої обробки гістологічних зображень різними методами деконвуляції, які можуть бути використані для локалізації інформативних патоморфологічних ознак. Тим не менш вибір правильного графічного методу залежить від конкретних вхідних даних, а отже повинен обиратися експертом самостійно.

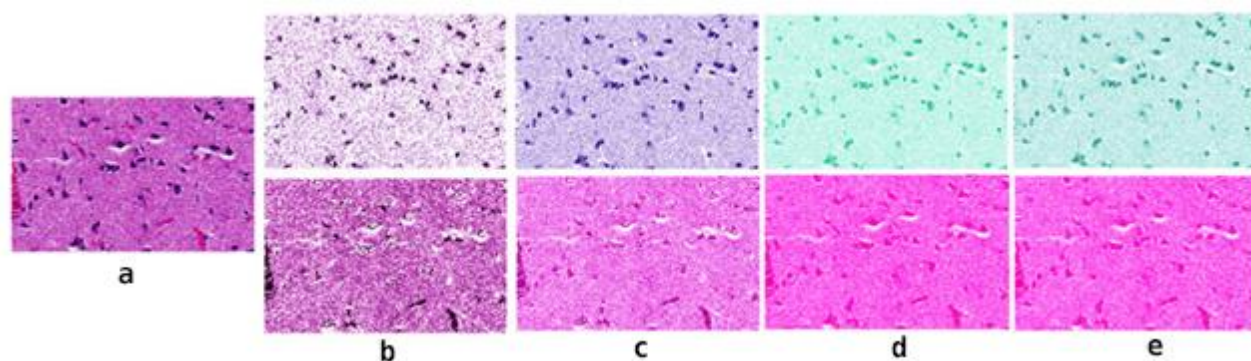


Рисунок 1.15. Приклад визначення гістологічної зони інтересу за допомогою графічних методів: а) початкове зображення; б) Khan [22]; в) Масенко [24]; д) SNMF [25]; е) NMF

Незважаючи на те, що методи локалізації зони інтересу, які були вже розглянуті, самі по собі можуть спростити роботу патоморфолога. Шляхом знаходження ділянок тканини, які мають підозри на рак. Тим не менш, це все ще не досконала система підтримки прийняття рішень лікаря, адже фактично ці алгоритми не виконують класифікацію даних. Для реалізації повноцінної інтелектуальної компоненти, необхідні більш продвинуті алгоритми машинного навчання. Такі як згорткові нейромережі (CNN). Їх головною перевагою є використання математичної згортки зображення, що дозволяє перетворювати його в форму, яку легше опрацювати, без втрати інформативності та інших параметрів початкового об'єкту, які можуть мати ключове значення при ідентифікації.

Згорткові нейронні мережі складаються із довільної кількості шарів штучних нейронів. Таким чином, намагаються структурно повторити будову

біологічного мозку, а отже і його когнітивні процеси. Кожен штучний нейрон – це математична функція, яка обчислює зважену суму вхідних сигналів і передає результат, у вигляді відповідної активації, далі. Наприклад, при розпізнаванні зображення перший шар CNN відповідає за основні параметри вхідного математичного опису, такі як горизонтальні або діагональні межі. Цей вивід подається далі, де вже знаходять більш специфічні ознаки, кути або контури. Власне глибина нейромережі буде обумовлена типом задачі та об'єктом безпосереднього інтересу. Тим не менш, останній шар, у всіх реалізація CNN, на основі усіх попередніх активацій, надає результат класифікації.

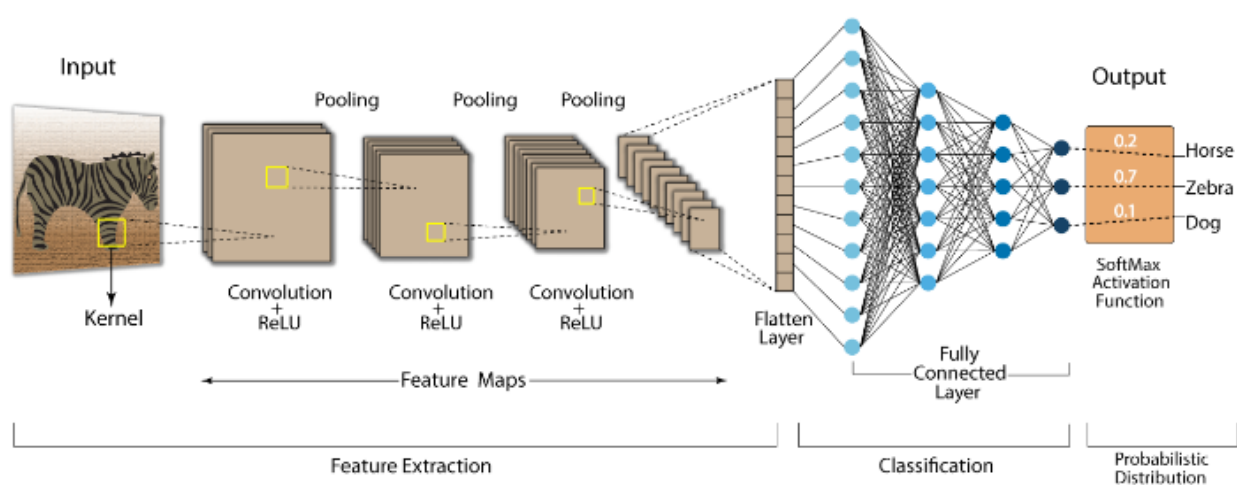


Рисунок 1.16. Приклад роботи згорткової нейромережі

Для покращення роботи нейромереж використовується алгоритм ковзного вікна, який дозволяє спростити обчислювальні задачі які мають складність  $O(n^2)$  до  $O(n)$ . Робиться це за рахунок швидкого обчислення тих елементів, які мають фіксоване «вікно». Це дозволяє повторно використовувати результат попереднього «вікна» для розрахунків наступного. З інженерної точки зору такий підхід дає можливість позбутися вкладених циклів, а з практичної це призводить до підвищення швидкодії системи.

Спільним недоліком більшості сучасних архітектур нейромереж є, так званий, *overfitting*. Це поняття стосується систем, які занадто добре моделюють навчальні дані. Тобто алгоритм настільки досконало вивчає деталі та шум, що це призводить до пониження ефективності при роботі з новими даними.

Проблема полягає у тому, що для масштабування нейромережі глибокого навчання є необхідним збільшувати навчальний датасет, але це якраз і спричиняє *overfitting* системи. Для CNN як перенасичення так й недонавчання призводить до зниження ефективності.

У загалом є два шляхи вирішення *overfitting*-у. Найпопулярніший з них – це метод повторної вибірки, який передбачає  $k$ -кратну перевірку навчальної множини. Відбувається це шляхом виконання операцій навчання-тестування на  $k$  різних підмножинах одного датасету. Більш універсальний, але далеко не такий ефективний метод – це використання перевірочних даних. По своїй суті це окрема підвибірка навчальної множини, яка не приймає участь безпосередньо у навчанні системи, але вже після його завершення перевіряє, на етапі тестування, наскільки гарно модель ідентифікує об'єкти, які ще ніколи не бачила.

Незважаючи на потужність та неймовірну популярність нейромереж, у сучасній культурі, вони все ще далекі від поняття «штучний інтелект». Хоч в основі CNN покладено просте розпізнавання ознак, проте для їх коректної роботи необхідна чи мала кількість навчальних даних, а єдиний адекватний спосіб підвищення вірогідності прийняття правильних класифікаційних рішень пов'язаний з масштабуванням нейромережі. Додаванням додаткових шарів в її структуру, яке призводить до підвищення обчислювального навантаження, що завжди пов'язано з лінійним, а іноді і експоненційним, зростанням часу навчання та перенавчання CNN.

Окрім цього хоч нейромережі і здатні класифікувати предмет інтересу, але вони все ще не спроможні повноцінно розрізняти контекст. Насамперед для цього необхідне розуміння метапараметрів, які далеко не завжди присутні в вхідному математичному описі, але які повинні бути частиною загальної бази знань будь-якої інтелектуальної компоненти. Наприклад, людина дивлячись на зображення здатна не лише класифікувати предмети на ньому, а й надати метаінформацію. Зробити деякі висновки з наявних засновків. Власне в цьому і

полягає глобальна задача діагностування онкології. Не лише констатація самих фактів про наявність чи відсутність деяких ознак раку, а й припущення стосовно його природи, оцінки інтенсивності поширення та прогнозування майбутнього перебігу хвороби.

Так у роботі [22] був використаний алгоритм локалізації зони інтересу NMF, але це було необхідно лише для того, щоб створити навчальний датасет для розроблювальної інтелектуальної системи. Розпізнавання ж самої онкології виконується за допомогою класифікатора на основі CNN. Як базову гістологічну ознаку автори обрали гіперхромні ядра. Проте це достатньо специфічна онко-особливість, яка може візуально відрізнитися в залежності від лабораторії або фенотипу самих ядер, які можуть бути наслідком як аномальної архітектури так й поганого фарбування зразка тканини. А неймережі, при обмеженості навчальної вибірки або поганого її анування, можуть бути дуже чутливими до такого роду невизначеності. Для подолання цієї проблеми автори [22] запропонували створити два класифікатори CNN з однаковою архітектурою, але один з них буде навчатися на везикулярних пробах, а інший на невезикулярних. Потім ці дві неймережі об'єднуються в одну шляхом додавання карт результуючих ймовірностей.

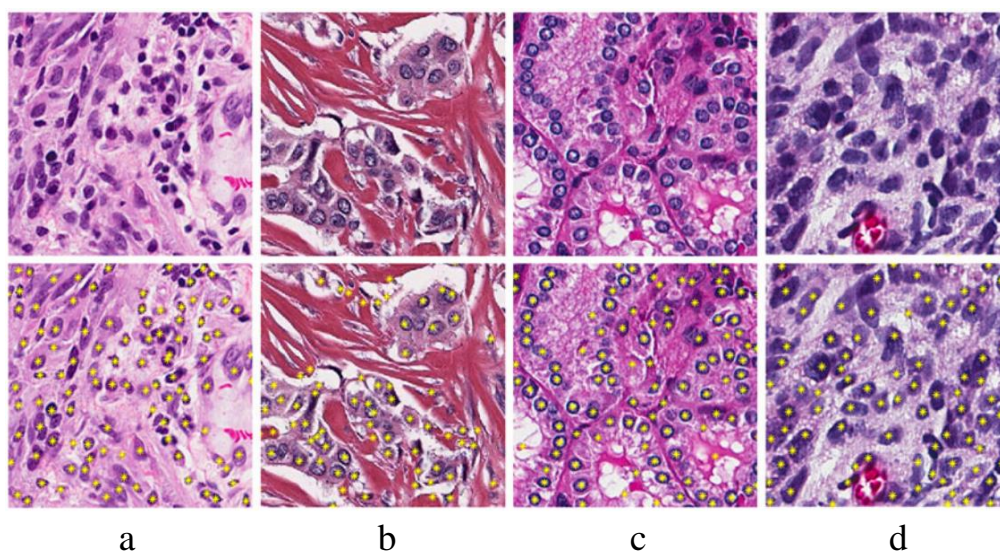


Рисунок 1.17. Результат виявлення гіперхромних ядер для різних тканин [22]:

- а) інфільтруючі лімфоцити в мультиформній гліобластомі; б-с) онкологія молочної залози; д) проліферуючі та плеоморфні пухлинні клітини;

Хоч гіперхромні ядра є дуже розповсюдженою гістологічною ознакою онкології, але при цьому вона є не над надійною при діагностуванні. Для більш комплексного аналізу природи і стану захворювання необхідно дещо більше ніж просте розпізнавання зображення. Зокрема бажано визначати форми скупчення та їх щільність у межах зразка. По декуди ці параметри напряду впливають на результат діагностування, адже дозволяють розрізнити аденому або гіперплазію від злоякісних новоутворень. У роботі [26] представлений достатньо цікавий підхід розпізнавання таких ознак за допомогою графів.

Автори локалізують зону інтересу, знаходять в середині неї об'єкт розпізнавання, а далі об'єднують їх в одну спільну структуру. Взагалі це достатньо специфічний підхід, але при цьому ефективний для визначення мета-параметрів осередку раку, як то щільність скупчення.

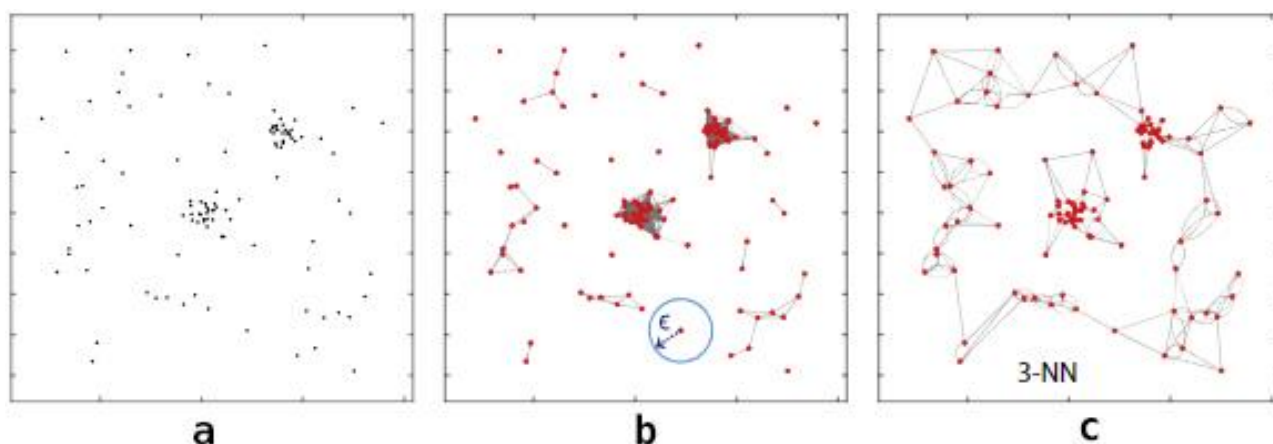


Рисунок 1.18. Графовий метод для аналізу онкології: а) початкова молекулярна локалізація; б)  $\epsilon$ -граф, використаний для побудови мережі, де кожен вузол з'єднаний з усіма іншими не-  $des$  в межах відстані  $\epsilon$ ; в) Графік  $kNN$ , використаний для побудови мережі, де кожен вузол з'єднаний лише з  $k$  найближчим сусідні вузли

Більшість сучасних ІСД намагаються використовувати максимальну кількість інформації на знімку тканини. Враховуючи структурні ознаки ураження, адже не достатньо локалізувати патологію, потрібно ще визначити її природу. У деяких випадках розрізнити рак від доброякісного новоутворення, адже візуально вони можуть бути схожими.

#### 1.4 Формалізована постановка задачі інформаційного синтезу інтелектуальної системи функціонального діагностування.

Нехай сформовано алфавіт  $\{X_m^o | m = \overline{1, M}\}$  з апріорно вірних класів розпізнавання автоматизованої системи діагностування патоморфології, які характеризують кадри зон інтересу на гістологічному зображенні. Для кожного із них побудовано відповідну тривимірну навчальну матрицю  $\|y_{m,i}^{(j)}\|$ , яка складається з значень яскравості кожного окремого пікселю. Тобто рядок  $\{y_{m,i}^{(j)} | i = \overline{1, N}\}$ , де  $N$  – кількість ознак, є структурованим вектором (далі в тексті просто реалізації), а стовпчик матриці – випадкова навчальна вибірка  $\{y_{m,i}^{(j)} | j = \overline{1, n}\}$   $i$ -ї ознаки з обсягом  $n$ .

Згідно принципів ІЕІ-технології, необхідно перетворити вхідну навчальну матрицю  $Y$  в, так звану, робочу бінарну  $X$ . Саме вона і буде змінюватися під час машинного навчання. За це відповідає вектор параметрів функціонування, який на пряму впливає на ефективність машинного навчання ІСД розпізнавати реалізації довільного класу  $X_m^o$  в бінарному просторі Хеммінга:

$$g_m = \langle x_m, d_m, \delta \rangle, \quad (1.1)$$

де  $x_m$  – усереднений вектор реалізацій класу  $X_m^o$ , який по сумісництву є центром гіперсферичного вирішального правила з радіусом  $d_m$ ;  $\delta$  - параметр оптимізації системи контрольних допусків на ознаки розпізнавання

Параметри функціонування, згідно принципів ІЕІ-технології, накладаються наступні обмеження:

- пікселі визначаються в інтервалі  $[0;255]$  градацій яскравості;
- допустима область значень радіусу  $d_m$  визначається нерівністю:

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

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

- параметр  $\delta$  має область значень, що задається наступною нерівністю:

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

де  $\delta_H$  - нормоване поле допусків на ознаки розпізнавання;

При функціонування системи в режимі екзамену, тобто ідентифікації кадрів гістологічних зображень, необхідно забезпечити перевірку прийняття класифікаційних рішень. Для цього, на етапі машинного навчання, потрібно оптимізувати відповідні параметри (1.1).

$$\bar{E}^* = \frac{1}{M} \sum_{m=1}^M \max_{G_E \cap \{k\}} E_m^{(k)}, \quad (1.2)$$

де  $E_m^{(k)}$  – значення інформаційного критерію, обчислене на  $k$ -му кроці машинного навчання;  $G_E$  – робоча область обчислення інформаційного критерію;  $\{k\}$  – множина кроків машинного навчання.

Для реалізації поставленої задачі необхідно виконати наступні кроки:

- реалізувати відповідні математичні моделі функціонування інтелектуальної компоненти в режимах машинного навчання та екзамену;
- реалізувати перевірку функціональної ефективності системи діагностування за допомогою обраного інформаційного критерію;
- оцінити функціональну ефективність розпізнавання кадрів зображення у рамках етапу екзамену;

У результаті маємо, що задача інтелектуально-інформаційної системи розпізнавання патоморфологічних ознак полягає в оптимізації її внутрішніх параметрів машинного навчання через наближення інформаційного критерію (1.2) до його максимально граничного значення.

## 2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

### 2.1 Основні положення інформаційно екстремальної технології аналізу даних.

Основна ідея машинного навчання системи розпізнавання, у рамках ІЕІ-технології, згідно до праць [1, 2], полягає в побудові, в рамках геометричного підходу, високо достовірних вирішальних правил шляхом оптимізації параметрів машинного навчання системи. При цьому здійснюється цілеспрямований пошук глобального максимуму функції статистичного інформаційного критерію в робочій області її визначення, під час процесу 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, 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 [1], здійснюється за багатоциклічною процедурою пошуку максимального граничного, усередненого за алфавітом  $\{X_m^o\}$ , значення інформаційного критерію оптимізації параметрів машинного навчання (2.1):

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

де  $E_m$  – інформаційний критерій оптимізації параметрів машинного навчання системи розпізнавати реалізації класу  $X_m^o$ ;  $G_{\xi}$  – допустима область значень  $\xi$ -ї ознаки розпізнавання;  $G_E$  – допустима область визначення функції інформаційного критерію.

Згідно праці [2] на алгоритм навчання (2.1) накладаються такі обмеження:



$$\left(\forall X_m^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left[X_m^o \neq \emptyset\right], \quad (2.2)$$

де  $\tilde{\mathfrak{R}}^{|\mathfrak{M}|}$  – розбиття простору ознак на класи з потужністю  $\text{Card } \tilde{\mathfrak{R}} = M$ ;

$$\left(\exists X_k^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left(\exists X_l^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left[X_k^o \neq X_l^o \rightarrow X_k^o \cap X_l^o \neq \emptyset\right]; \quad (2.3)$$

$$\left(\forall X_k^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left(\forall X_l^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left[X_k^o \neq X_l^o \rightarrow \text{Ker} X_k^o \cap \text{Ker} X_l^o = \emptyset\right], \quad (2.4)$$

де  $\text{Ker} X_k^o$  – ядро класу розпізнавання  $X_k^o$ ;  $\text{Ker} X_l^o$  – ядро класу розпізнавання;  $X_l^o$  – ядро найближчого сусіда для класу  $X_k^o$ ;

$$\begin{aligned} &\left(\forall X_k^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left(\forall X_l^o \in \tilde{\mathfrak{R}}^{|\mathfrak{M}|}\right) \left[X_k^o \neq X_l^o \rightarrow (d_k^* < d(x_k \oplus x_l)) \& \right. \\ &\quad \left. \& (d_l^* < d(x_k \oplus x_l))\right], \quad (2.5) \end{aligned}$$

де  $d_k^*$  – оптимальний радіус контейнера класу розпізнавання  $X_k^o$ ;  $d_l^*$  – оптимальний радіус для контейнера  $X_l^o$ ;  $d(x_k \oplus x_l)$  – кодова відстань між вектором  $x_k$ , усередненим за реалізаціями класу розпізнавання  $X_k^o$ , та аналогічним вектором  $x_l$  для  $X_l^o$ ;

$$\bigcup_{X_m^o \in \tilde{\mathfrak{R}}} X_m^o \subseteq \Omega_B; k \neq l; k, l, m = \overline{1, M}, \quad (2.6)$$

де  $\Omega_B$  – бінарний простір ознак розпізнавання.

Глибина машинного навчання визначається розмірністю вектору параметрів навчання. При цьому, внутрішній цикл реалізує, так званий, базовий алгоритм, призначенням якого є:

- 1) обчислення на кожному кроці навчання інформаційного критерію і пошук глобального максимуму його функції;
- 2) визначення оптимальних координат векторів  $\{x_m^* | m = \overline{1, M}\}$  і радіусів контейнерів класів розпізнавання  $\{d_m^* | m = \overline{1, M}\}$ ;

Таким чином, інформаційно-екстремальне машинне навчання полягає в наближенні на кожному кроці навчання інформаційного критерію (1.2) до його максимального граничного значення.

## 2.2. Математичні моделі машинного навчання.

Розглянемо категорійну модель машинного навчання розробленої СР у вигляді орієнтованого графа, де задіяні множини, що відображаються одна на одну відповідними операторами.

Категорійна модель включає вхідний математичний опис системи розпізнавання, який подано як структуру (2.7):

$$\Delta_B = \langle G, T, \Omega, Z, Y, X; f_1, f_2 \rangle, \quad (2.7)$$

де  $G$  – простір вхідних сигналів (факторів);  $T$  – множина моментів часу одержання інформації;  $\Omega$  – простір ознак розпізнавання;  $Z$  – простір станів системи, який визначає алфавіт класів розпізнавання;  $Y$  – вибіркова множина, яка утворює вхідну багатовимірну навчальну матрицю;  $X$  – бінарна навчальна матриця;  $f_1 : G \times T \times \Omega \times Z \rightarrow Y$  – оператор формування вхідної навчальної матриці  $Y$ ;  $f_2 : Y \rightarrow X$  – оператор трансформації вхідної навчальної матриці  $Y$  в бінарну  $X$ .

При цьому, відповідно до [2], декартів 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 БСР з оптимізацією відповідних параметрів – координат структурованого вектору (1.1) показано на рисунку 2.1.

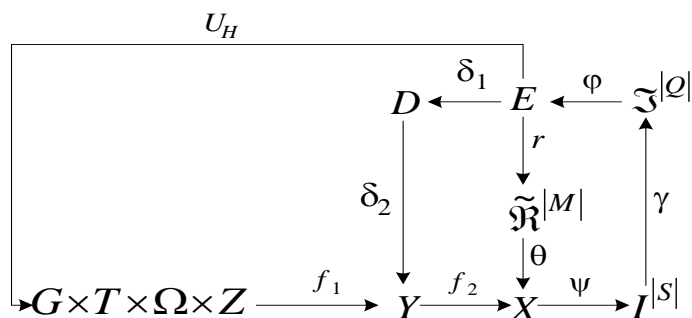


Рисунок 2.1 – Категорійна модель машинного навчання СР

На рисунку 2.1, згідно роботи [1], терм-множина  $E$  складається зі значень інформаційного критерію, які обчислювалися на кожному кроці машинного навчання і є загальними для всіх контурів оптимізації. Оператор  $r: E \rightarrow \tilde{\mathfrak{R}}^{|M|}$  буде на кожному кроці навчання розбиття  $\tilde{\mathfrak{R}}^{|M|}$ , яке відображається оператором  $\theta$  на нечіткий розподіл двійкових реалізацій 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  $\gamma$  визначає множину  $\mathfrak{S}^{|Q|}$  точнісних характеристик класифікаційних рішень, де  $Q = S^2$ , а оператор  $\Phi$  обчислює множину значень  $E$  інформаційного критерію оптимізації, який є функціоналом від точнісних характеристик. Контур оптимізації контрольних допусків на ознаки розпізнавання замикається через терм-множину  $D$ , елементами якої є значення системи контрольних допусків (СКД) на ознаки розпізнавання. Оператор  $U$  регламентує процес машинного навчання.

Автори [2] пропонують виконувати перевірку функціональної ефективності, інформаційно-екстремального машинного навчання, здійснюється при функціонуванні системи розпізнавання образів в режимі екзамену.

Категорійна модель подана у вигляді орієнтованого графа відображень множин, які застосовуються при функціонуванні розробленої системи в режимі класифікації об'єкту, показано на рис. 2.2.



належність класу  $X_m^o$  реалізації, що розпізнається, до повної ймовірності  $P_{f,m}^{(k)}$  помилкового прийняття рішень і різниці цих імовірностей:

$$E_m^{(k)} = \log_2 \frac{P_{t,m}^{(k)}}{P_{f,m}^{(k)}} * [P_{t,m}^{(k)} - P_{f,m}^{(k)}]. \quad (2.8)$$

Згідно з теоремою про повну ймовірність для двохальтернативної системи оцінок рішень при допущенні, згідно із принципом Лапласа-Бернуллі, що  $p(\mu_m) = p(\mu_c) = 0,5$ , маємо:

$$P_{t,m}^{(k)} = 0,5D_{1,m}^{(k)}(d) + 0,5D_{2,m}^{(k)}(d); P_{f,m}^{(k)} = 0,5\alpha_m^{(k)}(d) + 0,5\beta_m^{(k)}(d), \quad (2.9)$$

де  $D_{1,m}^{(k)}(d)$  – перша достовірність прийняття рішення на  $k$ -му кроці навчання;  $D_{2,m}^{(k)}(d)$  – друга достовірність;  $\alpha_m^{(k)}(d)$  – помилка першого роду;  $\beta_m^{(k)}(d)$  – помилка другого роду;  $d$  – дистанційна міра, яка визначає радіуси гіперсферичних контейнерів, побудованих в радіальному базисі Хеммінга.

Після підстановки виразу (2.9) в формулу (2.8) отримаємо

$$E_m^{(k)} = 0,5 \log_2 \left( \frac{D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * \{ [D_{1,m}^{(k)}(d) + D_{2,m}^{(k)}(d)] - [\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)] \}. \quad (2.10)$$

Виразимо першу і другу достовірності:

$$D_{1,m}^{(k)}(d) = 1 - \alpha_m^{(k)}(d); D_{2,m}^{(k)}(d) = 1 - \beta_m^{(k)}(d). \quad (2.11)$$

Після підстановки виразів (2.11) в формулу (2.10) отримаємо:

$$E_m^{(k)} = \log_2 \left( \frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * [1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))]. \quad (2.12)$$

Нормована модифікація критерію (2.12) має вигляд (2.13)

$$E_m^{(k)} = \frac{E_{Km}^{(k)}}{E_{K\max}^{(k)}}, \quad (2.13)$$

де  $E_{\max}^{(k)}$  – значення інформаційного критерію, яке приймається при підстановці  $D_{1,m}^{(k)}(d) = D_{2,m}^{(k)}(d) = 1$  і  $\alpha_m^{(k)}(d) = \beta_m^{(k)}(d) = 0$  у формулу (2.12).

Розглянемо процедуру обчислення інформаційного критерію (2.12). Оскільки, інформаційний критерій є функціоналом від точнісних характеристик, то при репрезентативному, але обмеженому обсязі навчальної вибірки, користуються їх оцінками:

$$\alpha_m^{(k)}(d) = \frac{K_{1,m}^{(k)}}{n_{\min}}; \quad \beta_m^{(k)}(d) = \frac{K_{2m}^{(k)}}{n_{\min}}, \quad (2.14)$$

де  $K_{1,m}^{(k)}$  – кількість подій, при яких реалізації, що належать класу  $X_m^o$ , помилково до нього не відносяться;  $K_{3,m}^{(k)}$  – кількість подій, при яких помилково відносяться, до класу розпізнавання  $X_m^o$ , реалізації сусіда  $X_c^o$ ;  $n_{\min}$  – мінімальний обсяг навчальної вибірки, який визначається згідно з працею [13].

Після підстановки відповідних оцінок точнісних характеристик (2.14) у вираз (2.12) одержимо робочу формулу для обчислення критерію Кульбака:

$$E_m^{(k)} = \frac{1}{n} \log_2 \left\{ \frac{2n + 10^{-r} - [K_1^{(k)} + K_2^{(k)}]}{[K_1^{(k)} + K_2^{(k)}] + 10^{-r}} \right\} \left[ n - (K_{12}^{(k)} + K_3^{(k)}) \right], \quad (2.15)$$

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

На практиці у формулі (2.15) значення параметра  $r$  вибирається із інтервалу  $1 < r \leq 3$ , тобто дорівнює кількості знаків мантиси критерію  $E_m^{(k)}$ .

Обчислення коефіцієнтів  $K_{1,m}^{(k)}$  і  $K_{2,m}^{(k)}$ , при класифікації  $j$ -ої реалізації, здійснювалося за процедурами:

$$\text{if } x_m^{(j)} \notin X_m^o \text{ then } K_1(j) := K_1(j-1) + 1; \text{ if } x_c^{(j)} \in X_m^o \text{ then } K_3(j) := K_3(j-1) + 1$$

При цьому віднесення, наприклад, реалізації  $x_c^{(j)}$  до класу розпізнавання  $X_m^o$  здійснюється за правилом:

- 1) обчислюється кодова відстань  $d[x_m \oplus x_c^{(j)}]$ ;
- 2) порівняння: якщо  $d[x_m \oplus x_c^{(j)}] \leq d_m$ , то  $x_c^{(j)} \in X_m^o$ , інакше –  $x_c^{(j)} \notin X_m^o$ ;

Таким чином, інформаційні критерії (2.12) і (2.13) є функціоналами від точнісних характеристик класифікаційних рішень, які в свою чергу залежать від дистанційних параметрів. Тобто, вище наведені модифікації критерію Кульбака можна розглядати як узагальнення відомих статистичних і дистанційних критеріїв близькості класів розпізнавання.

#### 2.4. Лінійний алгоритм машинного навчання інформаційно екстремальної технології для системи розпізнавання.

Базовий алгоритм, як правило, не забезпечує високої функціональної ефективності машинного навчання, оскільки контрольні допуски можуть бути не оптимальними в інформаційному розумінні. Згідно з категорійною моделлю (Рис. 2.1) інформаційно-екстремальний алгоритм машинного навчання системи розпізнавання з оптимізацією системи контрольних допусків (2.16) подано у вигляді ітераційної процедури пошуку глобального максимуму усередненого за алфавітом класів розпізнавання інформаційного критерію в робочій (допустимій) області визначення його функції.

$$\delta_K^* = \arg \max_{G_\delta} \{ \max_{G_E \cap \{k\}} \bar{E}^{(k)} \}, \quad (2.16)$$

де  $\delta_K^*$  – оптимальний параметр поля контрольних допусків;  $G_\delta$  – допустима область значень параметра  $\delta$  поля контрольних допусків;  $\{k\}$  – впорядкована, за часом множина, кроків машинного навчання.

На рисунку 2.3 показано двобічне симетричне поле контрольних допусків.

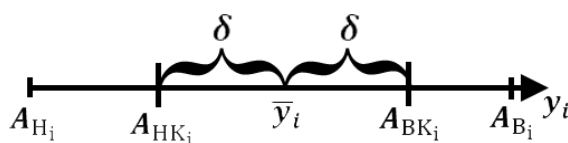


Рисунок 2.3 – Поле контрольних допусків на ознаку розпізнавання

Згідно рисунку 2.3 приймаємо такі позначення:  $\overline{y}_i$  – номінальне (усереднене) значення ознаки  $y_i$ ;  $A_{H,i}, A_{B,i}$  – нижні та верхні нормовані допуски на ознаку  $y_i$ ;  $A_{HK,i}, A_{BK,i}$  – нижні та верхні контрольні допуски на ознаку  $y_i$ ;  $\delta$  – параметр, який дорівнює половині симетричного поля контрольних допусків.

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 СКД.

Розглянемо схему алгоритму машинного навчання з оптимізацією контрольних допусків на діагностичні, за процедурою (2.16):

- 1) онулення лічильника класів розпізнавання:  $m := 0$ ;
- 2)  $m := m + 1$ ;
- 3) онулення лічильника зміни параметра  $\delta$  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 до правил:

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

- 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  $\rho_m$  – рівень квантування координат двійкового вектору  $x_m$ , 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  $X_c^o$ ;

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  $\delta^*$  і відповідних нижніх  $A_{H,i}^*$  і верхніх  $A_{B,i}^*$  контрольних допусків на всі ознаки розпізнавання;

Отримані в процесі паралельної оптимізації екстремальні значення параметрів машинного навчання є квазіоптимальними, оскільки вони змінювалися, при кожному кроці навчання, на однакову величину для всіх ознак одночасно. Для підвищення функціональної ефективності СР було реалізовано алгоритм 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 (2.17).

$$\delta_{K,i}^* = \arg \otimes_{l=1}^L \left\{ \max_{G_{\delta i}} \left[ \frac{1}{M} \sum_{m=1}^M \max_{G_{E_m \cap \{k\}}} E_m^{(l)}(d_m) \right] \right\}, i = \overline{1, N}, \quad (2.17)$$

де  $L$  – кількість прогонів процедури послідовної оптимізації контрольних допусків, обумовлених неоптимальними стартовими величинами для всіх ознак;  $\otimes$  – символ операції повторення;

Машинне навчання СР, 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 оптимальними геометричними параметрами контейнерів класів розпізнавання було побудовано продукційні вирішальні правила, які мають наступний вигляд:

$$\begin{aligned} (\forall X_m^o \in \mathfrak{R}^{|M|}) (\forall x^{(j)} \in \mathfrak{R}^{|M|}) [if (\mu_m > 0) \& (\mu_m = \max\{\mu_m\}) \\ then x^{(j)} \in X_m^o \quad else x^{(j)} \notin X_m^o], \end{aligned} \quad (2.18)$$

де  $x^{(j)}$  – вектор, що розпізнається;  $\mu_m$  – функція належності вектора  $x^{(j)}$  до контейнеру класу розпізнавання  $X_m^o$ .

У виразі (2.18) функція належності, для гіперсферичного контейнера класу розпізнавання  $X_m^o$ , визначається за формулою (2.19)

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \quad (2.19)$$

де  $x_m^*$ ,  $d_m^*$  – оптимальні параметри машинного навчання: усереднена двійкова реалізація і радіус гіперсферичного контейнеру, відповідно.

Таким чином, на екзамені визначається, за вирішальними правилами (2.18), належність реалізації класу, що розпізнається, одному із класів, які задані алфавітом. При цьому, вирішальні правила, через малу обчислювальну трудомісткість, відрізняються високою оперативністю.

## **2.5. Інформаційно-екстремальне машинне навчання системи функціонального діагностування за ієрархічною структурою даних.**

Основною перевагою інформаційно-екстремальної технології, над нейроподібними методами інтелектуального аналізу даних, є побудова класифікаційних правил у рамках геометричного підходу. Насамперед ця особливість допомагає побудувати майже інваріантні класифікаційні правила, які будуть не чутливі до довільного положення об'єкту інтересу в кадрі розпізнавання. Тим не менш залишається не вирішеною проблематика багатовимірності алфавіту класів, які в процесі функціонування системи можуть досягати довільних значень. Адже добре відомо, що при збільшенні класифікаційних ознак – збільшується ступінь перетину класів розпізнавання. Так у праці [1] ступінь перетину характеризується відношенням повної ймовірності  $P_f$  помилкових класифікаційних рішень до повної ймовірності  $P_t$  прийняття правильних класифікаційних рішень, то в рамках збільшення потужності алфавіту статистично достовірність класифікації буде зменшуватися. Як один із можливих варіантів подолання цієї проблеми є перехід від лінійного класифікатору, до структурованого. Наприклад ієрархічного.

У рамках інформаційно-екстремальної технології структурний класифікатор прийнято будувати як дерево [30], де кожна наступна страта містить в собі атрибути верхніх, у нашому випадку це класи розпізнавання. Тобто ідея ідентифікації буде полягати у переході від більшої абстракції до меншої. При цьому кожен клас, який ієрархічно знаходиться вище, кожного разу повертається в свою страту більш нижнього ярусу. Оскільки цей принцип відрізняється від класичного рекурсивного дерева, то таку структуру будемо називати декурсивною (рис. 2.4).

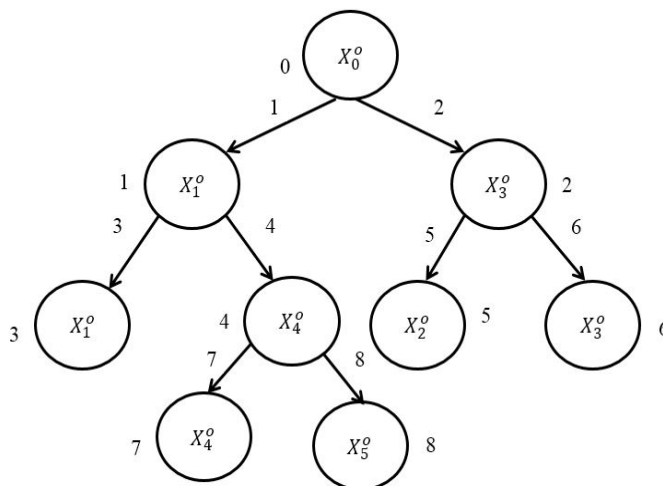


Рисунок 2.4 – Приклад декурсивної ієрархічної структури даних

Відповідно до рисунку 2.4, вершини декурсивного дерева є класи розпізнавання, які належать наступному алфавіту:  $\{X_{h,s,m}^o\}$ , де  $h$  - це номер ярусу дерева;  $s$  - порядковий номер страти  $h$ -го рівня;  $m$  - це порядковий номер класу в алфавіті. Таким чином верхній ярус отриманого дерева на рисунку 2.4 складається з однієї страти, в яку входять два класи:  $X_{1,1,1}^o$ ,  $X_{1,1,3}^o$ . Ярусом нижче знаходиться вже дві страти, в кожній з яких по два класи  $X_{2,1,1}^o$ ,  $X_{2,1,4}^o$  в першій і  $X_{2,2,2}^o$ ,  $X_{2,2,3}^o$  відповідно в другій. Варто помітити, що при такій організації класи  $X_{1,1,1}^o$  і  $X_{1,1,3}^o$  перенесено у відповідні страти нижнього ярусу. Таким чином запропонована структура дозволяє реорганізувати лінійний класифікатор для п'яти класів таким чином щоб класифікація виконувалася попарно. У свою чергу такий підхід дозволяє підвищити ймовірність побудови безпомилкових, за навчальними даними, класифікаційних рішень через зменшення перетину класів.

Категорійну модель машинного навчання системи діагностування з ієрархічним класифікатором, відповідно до роботи [30], представимо у вигляді орієнтованого графу. На відмінну від лінійного класифікатора (рис.2.1), ієрархічний (рис.2.5) має додатковий контур оптимізації, який замикається через структурну множину  $H$ .

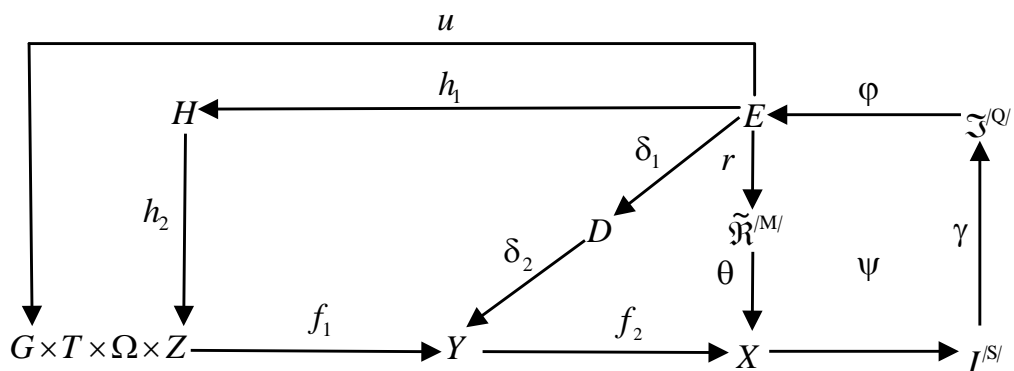


Рисунок 2.5 – Категорійна модель ієрархічного навчання

Відповідно до рисунку 2.5 оператор  $h_1$  обирає необхідний ярус і страту з множини  $H$ , ці дії задають план машинного навчання. У свою чергу  $h_2$  змінює простір можливих технічних станів об'єкти ідентифікації, що впливає на алфавіт класів розпізнавання.

Виходить, що кінцева модель ієрархічного машинного навчання, у рамках ІЕІ-технології, полягає в побудові багаторівневої структури. Кожен рівень якої складається з лінійного класифікатора, який визначає подальший рух кадрів розпізнавання. Тим не менш, при такому підході з'являється питання розташування алфавіту класів в ієрархічному дереві. Найпростіший спосіб структуризації – це експертна оцінка. Тобто деяка людина, за допомогою своїх суб'єктивних знань може розмістити класи, наприклад, за рівнем їх абстракції. Більш загальні вище, більш конкретні нижче. Проте це можна зробити і автоматично. Експериментально було підтверджено [3], що чим менша дисперсія класу розпізнавання, тим більш однозначний на ньому об'єкт. Цим принципом можна скористатися для встановлення ієрархічних зв'язків серед класів розпізнавання.

### **3. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДІАГНОСТУВАННЯ ОНКОЛОГІЇ ПЕРЕДМІХУРОВОЇ ЗАЛОЗИ**

#### **3.1.Формування вхідного математичного опису персоналізованої системи діагностування онкологій**

Однією з актуальних проблем сучасної біоінформатики є реалізація автономних систем діагностування та лікування хвороб найрізноманітнішого генезу. Однак вирішення цієї задачі ускладнюється тим фактом, що будь-який організм має свої власні біомаркери, які впливають на перебіг внутрішніх процесів. Саме тому медична інтелектуальна компонента повинна бути наділена властивістю персоналізації аналізу даних. Тобто виділення найбільш характерних діагностичних ознак для конкретного пацієнта або випадку.

Щоб реалізувати персоналізовану систему, для задачі ідентифікації гістологічних зображень, було запропоновано наступний алгоритм дій:

- 1) формування класифікатору з алфавіту, який створений автоматично за допомогою методу кластер аналізу;
- 2) виділення зони інтересу на зображенні шляхом використання методів графічної обробки;
- 3) використання класифікатору на попередньо виділеній зоні інтересу;

Перед початком роботи необхідно сформувати датасет з кадрів розпізнавання, які будуть використанні для організації алфавіту класів. Вхідним матеріалом, згідно сформованого принципу персоналізації, є зображення тканини, які отримані від однієї людини. Таким чином вдасться виділити патморфологічні ознаки, які характерні для конкретного пацієнта.

Датасет складається з 543 гістологічних кадрів розміром 35 на 35 пікселів. Взявши як основну інформаційну характеристику – дисперсію яскравості, використаємо відомий алгоритм кластеризації k-means, щоб для початку розбити простір датасету на дві частини. Одна, теоретично, буде відповідати за

здорову тканину, а інша за підозру на онкологію. Оскільки нас більше цікавить другий кластер, то зробимо ще одне розбиття у його межах, щоб деталізувати ознаки. Тобто на даному етапі у нас є три кластери, два з яких ієрархічно пов'язані між собою. Відповідно класифікатор, який потрібно побудувати буде мати наступний вигляд, рисунок 3.1. Де  $X_1^0$  – клас, який найкраще описує перший кластер,  $X_2^0$  - клас, який найкраще описує другий кластер,  $X_3^0$  – це клас, який є підкласом  $X_2^0$ .

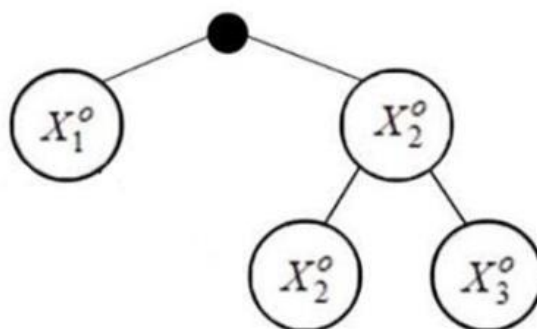


Рисунок 3.1 – Модель ієрархічного класифікатору

Для того щоб сформувати алфавіт класів розпізнавання було запропоновано використати ІЕІ-технологію, як інструмент для обрання найкращого математичного опису. Для цього кадри, в кожному з кластерів, було згруповано на  $p$ -груп, з яких обрано представника, що є найближчим до усередненого значення яскравості у межах групи. Далі за допомогою нормованого інформаційного критерію (2.1), під час паралельного машинного навчання, визначалася найкраща пара кадрів для кожної з страт. Таким чином вдалося сформувати вхідний математичний опис, який, з інформаційної точки зору, є максимально достовірним. При цьому, що важливо, алфавіт класів формується на попередньо впорядкованих множинах датасету гістологічних ознак, що дозволяє обирати класи в напівавтоматичному режимі.

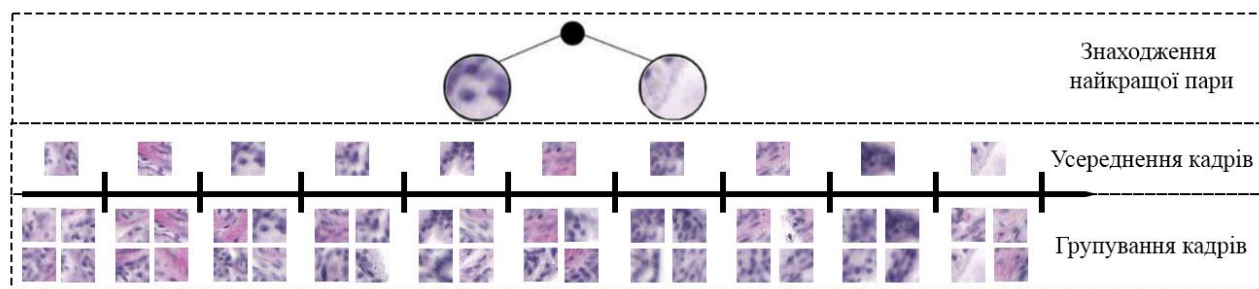


Рисунок 3.2 – Приклад обрання пари  $X_1^0$  і  $X_2^0$

Головною перевагою такого підходу є автоматичне формування алфавіту класів розпізнавання. Перш за все це мінімізує людське втручання, яке впливає на доцільність вибору гістологічних кадрів. Замість цього система сама вирішує, які з них є найбільш інформативними, а отже й максимально придатними для ролі вхідного математичного опису. Людині залишається лише логічно аргументувати форму ієрархічного класифікатору і дати назви кожному з обраних класів розпізнавання.

Окрім цього первинне, нечітке розбиття датасету, за допомогою методів кластеризації, дозволяє виконати початковий аналіз вибірки. Сформувані припущення про кількість класів розпізнавання на даному гістологічному знімку. Такий підхід надає лікарю, ще на етапі початкової обробки біоптату, зробити деякі висновки про характер ураження. За допомогою, наприклад, гістограм яскравості подивитися наскільки однорідною є тканина і чи є в ній ознаки онкологічного характеру.

Тим не менш, під час лабораторного експерименту було виявлено, що у більшості випадках для ідентифікації новоутворень достатньо лише трьох класів розпізнавання, організованих в наступну структуру (рис.3.1). Річ у тім, що більшість патоморфологічних ознак є неоднорідними і для їх виявлення не достатньо лише класифікувати кадри зображення, потрібно аналізувати знімки комплексно. Тому у рамках дослідження було запропоновано зосередитися на ранніх стадіях раку передміхурової залози. Приблизно третій рівень за шкалою Глісона, адже більш вищі рівні характеризуються низьким ступенем диференціації зразка тканини. Як правило пізні стадії онкології дуже легко



ідентифікуються, а ось ранні характеризуються високою розбіжністю при прийнятті класифікаційних рішень. Як мінімум, гіперплазія або аденома може мімікрувати під злоякісне новоутворення. Тому подальша робота буде зосереджена на розрізненні ранньої стадії раку та доброякісних пухлин.

Для виконання цієї задачі нам достатньо буде ієрархічної структури наступного вигляду (рис. 3.3), де  $X_1^0$  відповідає за не ураженні ділянки сполучної тканини,  $X_2^0$  – це просвіти біоптату, які з'являються при розрізанні залоз,  $X_3^0$  – ділянки з підозрою на рак. Алгоритм формування алфавіту класів був аналогічний описаному раніше (рис.3.2).

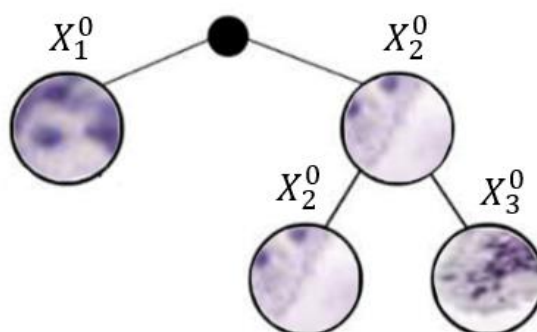


Рисунок 3.3 – Вхідний математичний опис у вигляді ієрархічної структури

### 3.2. Результати моделювання

Незважаючи на сформований алфавіт класів розпізнавання, у вигляді ієрархічної структури, цього не достатньо для діагностування онкології за гістологічними зображеннями тканини. Адже аналізувати ракові новоутворення необхідно комплексно, приймаючи до уваги найрізноманітніші мета-ознаки. Таким чином, враховуючи, що у межах дослідження розглядається онкологія на ранній її стадії, приблизно третій рівень за шкалою Глісона, то основна проблема для ідентифікації раку за кадрами тканини – це його візуальна подібність до аденоми або гіперплазії в передміхуровій залозі. Один із можливих способів розрізнити злоякісне новоутворення від доброякісного – це проаналізувати щільність скупчення залоз та їх розміри.

Для початку реалізуємо ієрархічний класифікатор розпізнавання гістологічних зображень, у рамках інформаційно-екстримальної технології. Для

цього потрібно створити два лінійних класифікатори між  $X_1^0$  і  $X_2^0$  й  $X_2^0$  і  $X_3^0$  відповідно. Таким чином на рисунку 3.4 показано графік усередненого, за алфавітом класів розпізнавання, нормованого інформаційного критерію (2.13) від параметру  $\delta$ , що відповідає половині симетричного поля контрольних допусків, при їх паралельній оптимізації. На цьому графіку і на кожному подальшому робочу область визначення функції критерію виділено темним кольором.

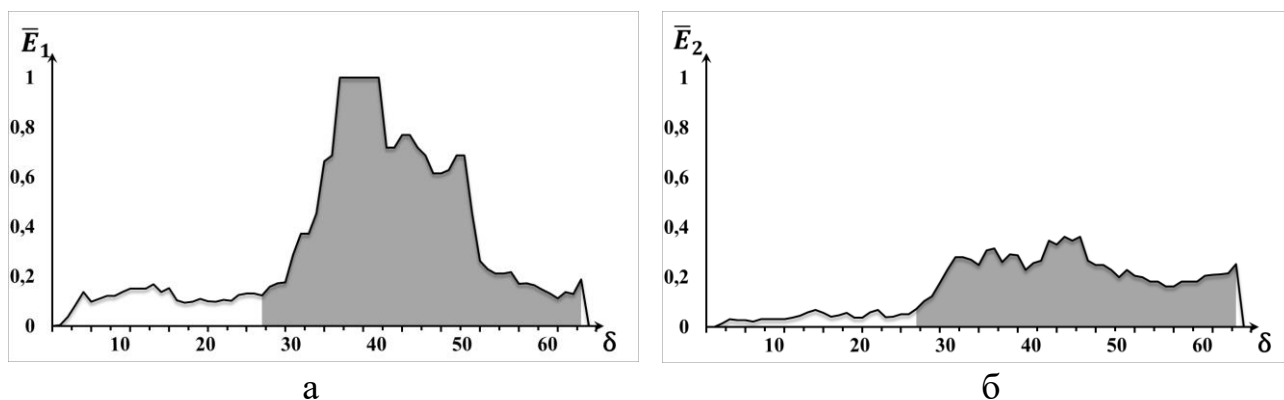


Рисунок 3.4 – Графік залежності інформаційного критерію (2.13) від параметра поля контрольних допусків між (а) класами  $X_1^0$  і  $X_2^0$  й (б) класами  $X_2^0$  і  $X_3^0$

З рисунку 3.4.а видно, що верхня страта ієрархічного класифікатору, яка складається з класів  $X_1^0$  і  $X_2^0$  набуває максимального граничного значення інформаційного критерію вже при  $\delta = 40$ . У той час як для другої страти (рис.3.4.б) критерій набуває значення лише в  $\bar{E}^* = 0.31$  при  $\delta = 46$ . Тому окремо для цього випадку потрібно провести послідовну оптимізацію контрольних допусків.

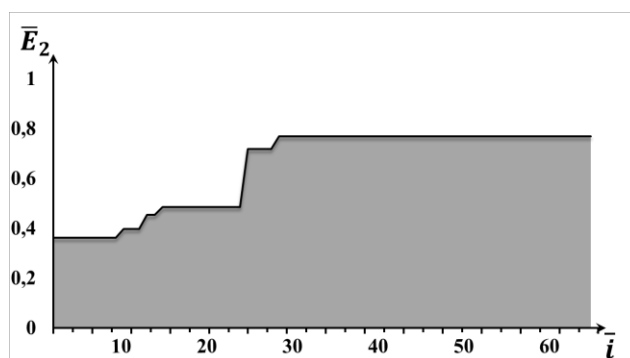


Рисунок 3.5 – Графік зміни інформаційного критерію (2.13) в процесі послідовної оптимізації контрольних допусків на ознаки розпізнавання

З рисунку 3.5 можна помітити, що значення інформаційного критерію для другої страти значно підвищилося, у порівнянні з паралельною оптимізацією і дорівнює  $\bar{E}^* = 0.76$ . При цьому досягти цього вдалося вже на першому прогоні, який визначається відношенням кількості ітерацій  $\bar{i}$  до кількості ознак  $N$ .

Знаючи параметри при яких класифікатори набувають своїх максимальних значень інформаційного критерію (2.13), побудуємо графіки залежності між ним та радіусів контейнерів розпізнавання (рис.3.6), бо саме вони дозволяють отримати оптимальні вирішальні правила (2.18).

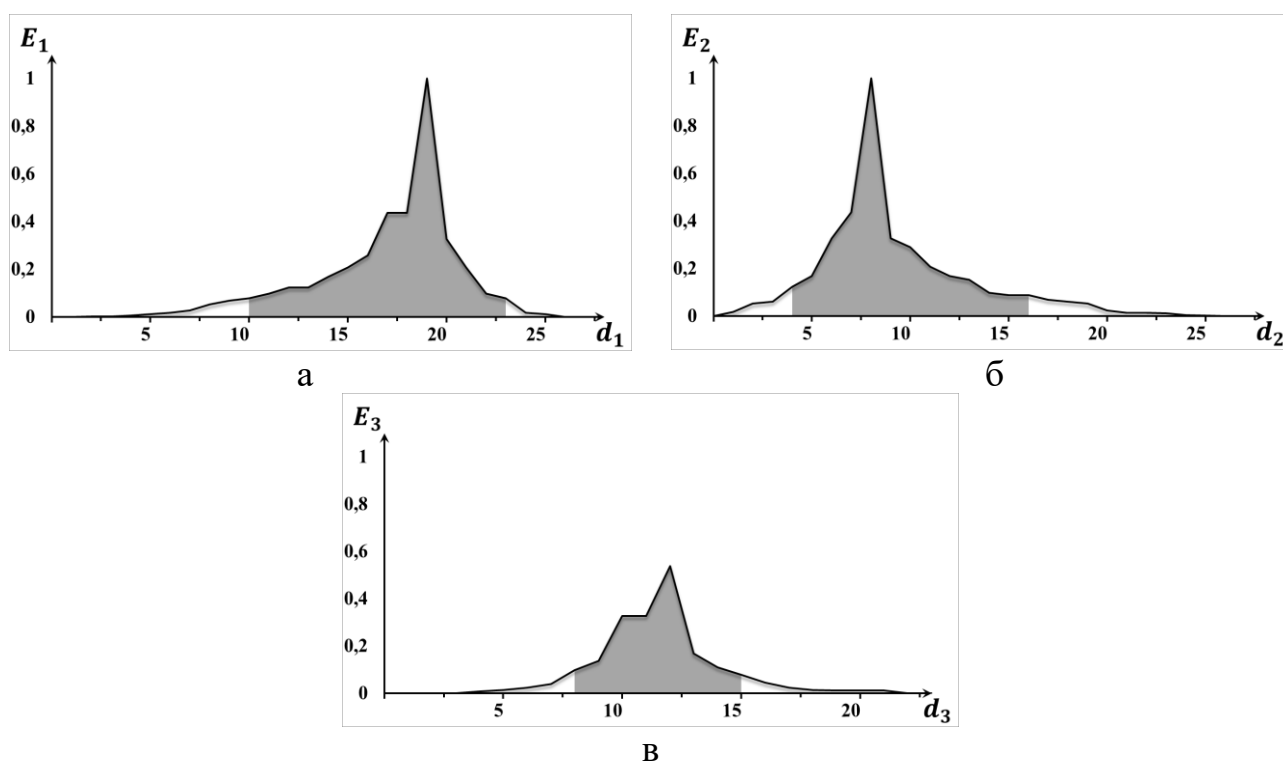


Рисунок 3.6 – Графік залежності критерію (2.13) від радіусів контейнерів класів розпізнавання: а – клас  $X_1^0$ ; б – клас  $X_2^0$ ; в – клас  $X_3^0$ ;

Згідно отриманого результату (рис.3.6) оптимальні значення радіусів контейнерів класів розпізнавання дорівнюють:  $d_1^* = 19$  (тут і далі в кодових одиницях) – для класу  $X_1^0$ ;  $d_2^* = 8$  для класу  $X_2^0$ ;  $d_3^* = 12$  для класу  $X_3^0$ .

І хоч побудований класифікатор може ідентифікувати зображення, проте цього, як зазначалося раніше, не достатньо для повноцінного діагностування гістологічних знімків. Адже результат, який подібний до того, що

продемонстрований в роботі [4], у вигляді сегментованої оцифровки, не є достатньо інформативним для лікаря. Окрім цього ранні стадії раку можуть бути візуально схожі на доброякісні новоутворення і інтелектуальна компонента повинна вміти їх розрізняти. Саме томи ІЕІ класифікатор буде використаний лише як складова комплексної системи підтримки прийняття рішень, у вигляді інструменту для локалізації зони інтересу.

Дві основні мета-ознаки, які дозволять розрізнити аденому від злоякісного новоутворення є: розмір та щільність розміщення залоз. Для початку потрібно перевести RGB (рис.3.6.A) зображення в HSV (рис.3.6.B). Оскільки при такій кольоровій схемі набагато краще буде видно гіперхромні ядра, які характерні для новоутворення будь-якого генезу. Через бінаризацію отриманого HSV-зображення (рис.3.6.C) ці ядра стануть більш помітні і вже на бінарному знімку, за допомогою інформаційно-екстремального класифікатору, визначимо зону інтересу (рис.3.6.D).

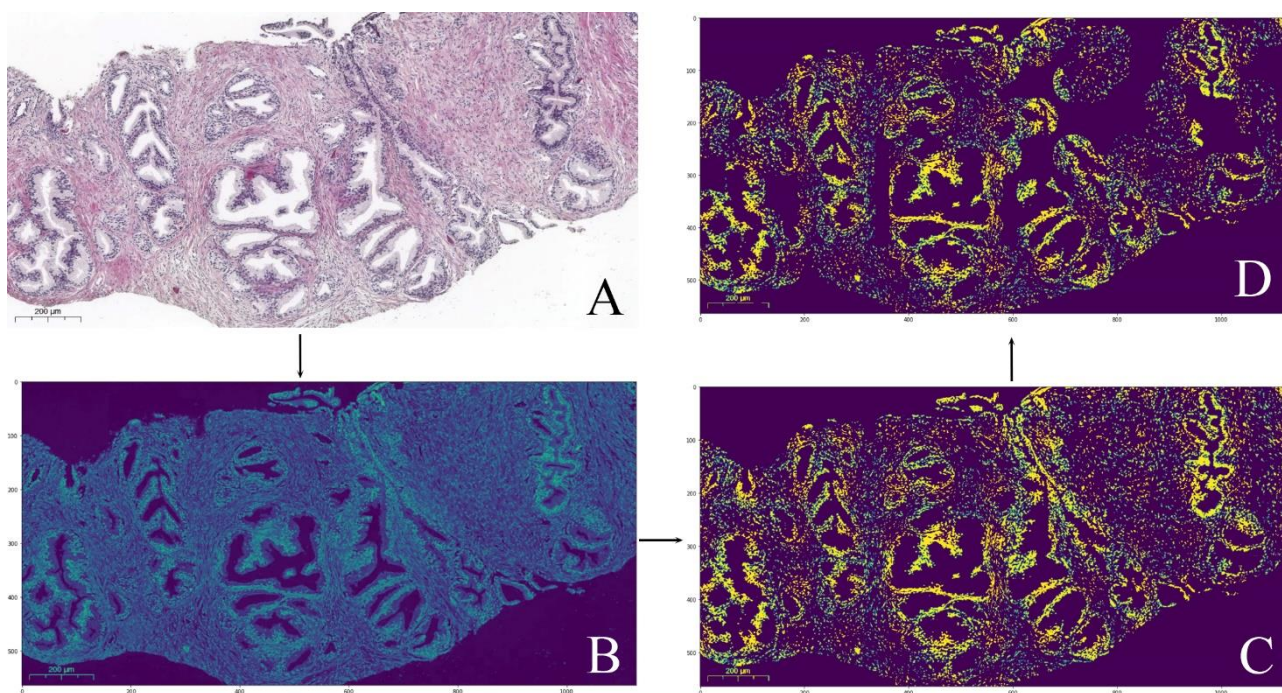


Рисунок 3.6 – Визначення зони інтересу на зображенні: А) початковий гістологічний знімок; В) гістологічний знімок в кольоровій схемі HSV; С) бінаризоване HSV-зображення; D) бінаризована зона інтересу;

Хоч гіперхромні ядра і характерні для новоутворень, проте це далеко не єдина їх візуальна ознака. Тому не достатньо просто бінаризувати зображення і сподіватися, що гіперхромія вкаже на вражені ділянки тканини. Саме тому у цій реалізації (рис.3.6) і використовується інформаційно-екстремальний класифікатор, який є гарантією того, що зона інтересу буде визначена правильно й дійсно буде містити в собі інформативні патоморфологічні ознаки.

Власне після локалізації зони інтересу необхідно визначити межі уражених залоз. Для цього спочатку потрібно позбутися від шуму на бінаризованому знімку (рис. 3.7.A). Далі, за допомогою морфологічних операцій обробки зображення, реконструюємо залози у бінарному просторі (рис.3.7.B) через визначення їх приблизної форми та області поширення.

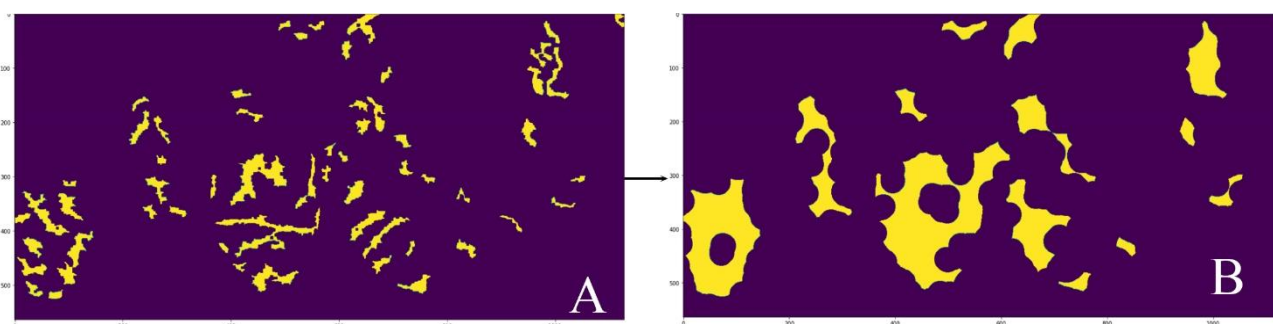


Рисунок 3.7 – Процес відтворення залоз в бінарному просторі: А) видалення зашумлення; В) реконструкція залоз;

Описаний алгоритм дій необхідний був для того, щоб визначити положення залоз та їх розміри. Частково ці характеристики вже можна оцінити на зображенні 3.7.B. Тим не менш, для спрощення подальшого діагностування, бажано визначити щільність скупчення залоз. Для цього необхідно розрахувати центр кожної та знайти найближчих до неї сусідів. При цьому, апріорно відомо, коли центри знаходяться не далі ніж  $250\mu\text{m}$ , то вони утворюють конгломерат. Таким чином на рисунку 3.8. зображено результуюче зображення, яке дозволяє перевірити ефективність машинного навчання. Червоним кольором зображено тіло залози, помаранчевим - область розповсюдження, а чорними точками – центри новоутворень.

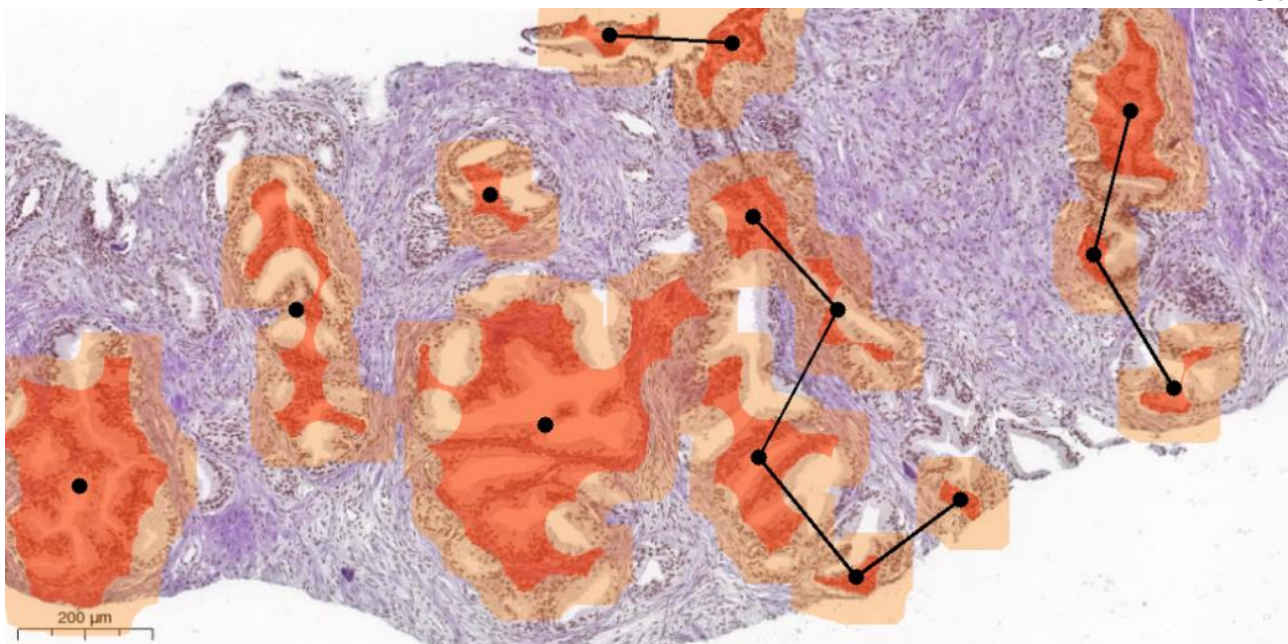


Рисунок 3.8 – Результат ідентифікації доброякісного новоутворення

На рисунку 3.8. видно, що залози, в переважній більшості, є великими. При цьому їх розміщення не характеризується достатньою щільністю поширення. У свою чергу, це означає, що на знімку не рак на ранній своїй стадії. Це гіперплазія передміхурової залози, а отже доброякісне новоутворення. Для чистоти експерименту проведемо усі вище описані дії, з тим самим алфавітом класів розпізнавання, але вже для іншого знімку.

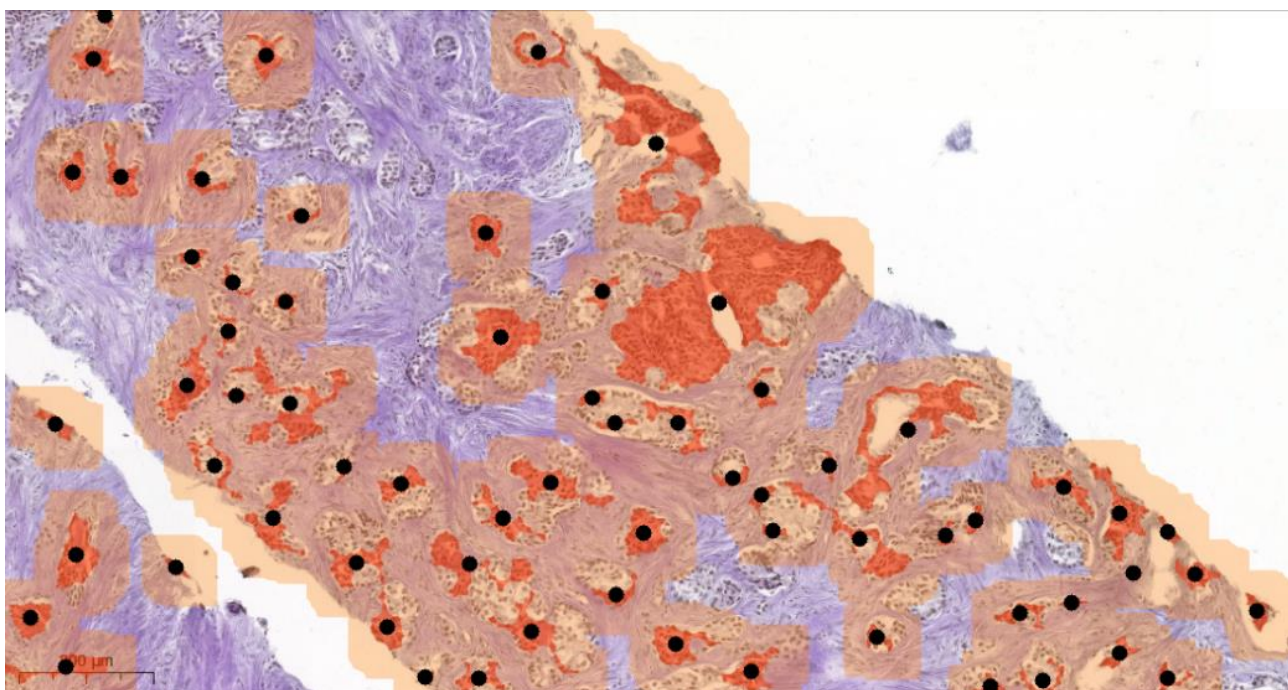


Рисунок 3.9 – Результат ідентифікації злоякісного новоутворення

На цей раз ми бачимо, що на рисунку 3.9 щільність залоз значно вища, а розміри менші. При цьому вони поступово заповнюють собою усю сполучну тканину і починають утворювати конгломерати, що характерно для ранніх стадій раку.

Перше, що потрібно зробити для покращення системи підтримки прийняття рішень – це додати додаткові мета-ознаки розпізнавання. Так, наприклад, онкологічні залози схильні до поступового збільшення та спрощення своєї форми під час перебігу захворювання. Як правило, вони мають еліптичну форму, у той час як аденома характеризується більш складним, деревоподібним зовнішнім виглядом. Це можна використати при подальшому дослідженні.

Окрім цього на етапі формування персоналізованого алфавіту класів розпізнавання можна додати нові метрики. Наприклад, у роботі [3] було продемонстровано, що дисперсія яскравості класів, на пряму впливає на їх компактність. Таким чином, при формуванні ієрархічного класифікатора можна розмістити алфавіт згідно дисперсії, чим більша різниця між класами тим далі будуть їх центри в просторі ознак.

Також сама парадигма інформаційно-екстремальної технології передбачає можливість поглиблення машинного навчання шляхом оптимізації або додавання нових параметрів. Таким чином можна оптимізувати розмір кадрів розпізнавання [5] або оптимальні їх RGB ваги [6].

### **3.3. Короткий опис програмної реалізації**

Система персоналізованого діагностування онкологій передміхурової залози була реалізована за допомогою поєднання двох мов програмування. Python для графічної обробки зображень та C# для інформаційно-екстремальної технології. Нижче наведено усі класи та модулі, які були використані.

Клас `imageSplitting` був написаний на мові програмування C# і необхідний для формування навчального датасету шляхом розбиття вхідного гістологічного зображення на кадри фіксованого розміру.

Таблиця 3.1 Опис складових класу imageSplitting

Методи	
Назва	Призначення
<code>static private Bitmap optimal_map_exam(Bitmap map, int W, int H)</code>	Зміна розміру вхідного зображення, щоб можна було його розбити на кадри розміром $N \times n$ .
<code>static void Main(string[] args)</code>	Ітеративний процес розбиття вхідного зображення на кадри однакового розміру

Клас SimpleClassifier реалізований за допомогою С# відповідає за організацію алфавіту класів ієрархічного класифікатору та оцифровку зображення на етапі екзамону. Використовується для локалізації зони інтересу на гістологічних знімках.

Таблиця 3.2 Опис складових класу SimpleClassifier

Методи	
Назва	Призначення
<code>static void Main(string[] args)</code>	Завантаження вхідного математичного опису
<code>static private Bitmap optimal_map_exam(Image map, int N, int n)</code>	Зміна розміру вхідного зображення, щоб можна було його розбити на кадри розміром $N \times n$ .
<code>static void exam(machine_learn topHeap, machine_learn subClass)</code>	Екзамен для ієрархічного класифікатору
<code>static void make_fragment_exam_map(int res, int x, int y, ref Bitmap map_num)</code>	Розмітка кадрів розпізнавання

Клас classSelection відповідає за вибір найкращої пари для конкретної страти ієрархічного класифікатору.

Таблиця 3.3 Опис складових класу classSelection

Методи	
Назва	Призначення
<code>public classSelection(List&lt;string&gt; paths)</code>	Конструктор, який отримує на вхід шлях до директорії з датасетом серед якого потрібно обрати пару найбільш придатних класів
<code>public classSelection(string path)</code>	Перевантажений конструктор. Використовується, коли один із класів вже відомий і потрібно обрати йому пару з деякого датасету



<code>List&lt;decart_coord&gt; sort(List&lt;decart_coord&gt; sample)</code>	Сортування завантажених датасетів за дисперсією їх яскравості
<code>float averageValueInterval(List&lt;decart_coord&gt; interval)</code>	Розрахунок інтервалів між кадрами датасету
<code>void findBestCouple(Bitmap left, List&lt;decart_coord&gt; right)</code>	Знаходження найкращої пари класів, коли один з них вже відомий
<code>public void findClasses(int intervals, Bitmap bestClass1)</code>	Розбиття датасету на інтервали при умові, що один із класів вже відомий
<code>void findBestCouple(List&lt;decart_coord&gt; left, List&lt;decart_coord&gt; right)</code>	Пошук найкращої пари класів
<code>public void findClasses(int intervals)</code>	Розбиття датасету на інтервали яскравості

Клас `decart_coord` зберігає інформацію про алфавіт клас розпізнавання.

Таблиця 3.4 Опис складових класу `decart_coord`

Методи	
Назва	Призначення
<code>public decart_coord(Bitmap source)</code>	Конструктор, який приймає на вхід зображення класу
<code>public void makeEV()</code>	Створення еталонного вектору на основі класу розпізнавання
<code>public void makeBM(int[] vd, int[] nd)</code>	Бінаризація класу розпізнавання
<code>void make_RGB()</code>	Розбиття класу на R, G, B складові та обчислення дисперсії й середнього значення яскравості

Клас `teach_decart` клас, який виконує оптимізацію параметрів машинного навчання в рамках методу інформаційно-екстремальної технології.

Таблиця 3.5 Опис складових класу `teach_decart`

Методи	
Назва	Призначення
<code>public teach_decart(ref List&lt;decart_coord&gt; sender)</code>	Конструктор, який приймає на вхід алфавіт класів розпізнавання
<code>void realization()</code>	Організація вхідного математичного опису
<code>bool local_dopusk (int k, int delta)</code>	Зміна поля контрольних допусків при послідовній оптимізації
<code>void make_para()</code>	Розбиття класів на пари, за принципом найближчого сусіда

<code>double INFK(int INFK_d, ref float INFK_d1, ref float INFK_betta, int k)</code>	Розрахунок інформаційного критерію. У даній реалізації це нормалізована міра Кульбака
<code>void make_sk(int k)</code>	Відстані між гіперсферами класів розпізнавання
<code>void make_myDo()</code>	Пошук оптимальних радіусів вирішальних правил
<code>bool propusk_KFE(int radius, int k)</code>	Перевірка робочої області визначення критерію
<code>public float average_KFE()</code>	Знаходження середнього значення критерію
<code>public void make_dopusk()</code>	Зміна контрольних допусків для паралельної їх оптимізації
<code>public void parallel_dopusk()</code>	Процес паралельної оптимізації контрольних допусків
<code>public void posledovatel_dopusk()</code>	Процес послідовної оптимізації контрольних допусків

Клас `machine_learn` клас, який відповідає за правильну організацію роботи `teach_decart` й `decart_coord`, а також реалізує режим екзамену у вигляді класифікації вхідних кадрів.

Таблиця 3.6 Опис складових класу `machine_learn`

Методи	
Назва	Призначення
<code>public void make_fragment(Bitmap bitmap)</code>	Конвертування вхідного зображення в матрицю яскравості
<code>public void realization_decart(bool posl)</code>	Організація машинного навчання з необхідними кванторами оптимізації внутрішніх параметрів
<code>public int exams_decart(Bitmap bitmap)</code>	Ідентифікація вхідних кадрів

Клас `Program` відповідає за визначення структури ієрархічного класифікатора та оцифровку результуючого зображення.

Таблиця 3.7 Опис складових класу `Program`

Методи	
Назва	Призначення
<code>static void exam(machine_learn topHeap, machine_learn subClass)</code>	Оцифровка зображення
<code>static void Main(string[] args)</code>	Організація класифікатора
<code>static void make_fragment_exam_map(int res, int x, int y, ref Bitmap map_num)</code>	Нанесення цифри класу на кадр

## ВИСНОВКИ

Головною метою дослідження було подолати проблему перетину патоморфологічних класів розпізнавання в просторі гістологічних ознак. На прикладі аденоми і раку передміхурової залози. Насамперед задача ускладнюється тим фактом, що візуально ці два типи патології достатньо схожі між собою і щоб їх розрізнити було використано метаінформацію про них. Такі як: величина і щільність розміщення залоз в тканині.

Таким чином було розроблено комплексну систему діагностування онкології, яка здатна розрізнити рак, на ранній стадії розвитку, від гіперплазії в передміхуровій залозі. Це було досягнуто шляхом поєднання інформаційно-екстремальної технології, яка використовувалася для локалізації зони ураження тканини, й графічних методів обробки, які допомогли відтворити початкову форму залоз в бінарному просторі. Результуюче зображення містить інформацію про площу ураження, центри залоз та візуально демонструє щільність їх скупчення з приблизними межами поширення патології.

Інтелектуальна компонента, на основі інформаційно-екстремальної технології, була реалізована через ієрархічний класифікатор. Це дозволило підвищити точність розпізнавання шляхом реорганізації алфавіту класів в більш оптимізовану структуру. При цьому вдалося збільшити точність прийняття класифікаційних рішень. Навіть для достатньо абстрактних класів, таких як: підозра на онкологію. У свою чергу це означає, що розроблена система має властивість узагальнення інформації і здатна ідентифікувати абстракції при малій кількості навчальної вибірки.

Тим не менш, інтелектуальна компонента використовувалася лише для локалізації зони інтересу гістологічного зображення, у вигляді ураження онкології. У межах якої, за допомогою методів графічної обробки зображення, було відтворено форму та розміри патології. Такий багатосаровий процес потрібний, насамперед, для того, щоб діагностування однозначно проводилося в зоні інтересу, визначення якої може бути ускладненим із-за вимог до

технологічного процесу створення гістологічних знімків. Тому спочатку виконується пошук абстракції «підозра на онкологію», а вже потім аналізуються залози в середині локалізованої зони.

Оскільки візуально рак й доброякісна патологія можуть бути візуально схожі, тобто мати деякі подібні патоморфологічні ознаки, то для їх розрізнення було запропоновано використовувати метаінформацію про обрані класи розпізнавання. На прикладі розміру, форми та щільності розміщення залоз в тканині. Це дозволяє підвищити вірогідність прийняття класифікаційних рішень під час безпосереднього діагностування, адже ідентифікація гістологічного знімку проводиться не по окремим кадрам, а комплексно. Під час дослідження було експериментально доведено життєздатність такого підходу до розпізнавання морфологічних образів.

Проте основною перевагою реалізованої системи є її властивість персоналізації внутрішніх параметрів функціонування. За допомогою кластер-аналізу будується вхідний математичний опис, який є унікальним для кожного окремого знімку тканини, а отже який буде враховувати індивідуальні біомаркери пацієнта при діагностуванні.

Подальші етапи розробки пов'язані з розширенням словника патоморфологічних ознак. Наприклад, додаванням модулю класифікування патології за розміром залоз чи формою їх просвіту. Також структура ієрархічного класифікатору потребує деякої оптимізації для її автоматизованого використання. Це передбачає виокремлення особливостей патоморфологічних ознак, які б дозволили, з інженерної точки зору, уніфікувати ієрархію між ними. При цьому потрібно дослідити використання різних метрик при формування алфавіту класів розпізнавання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А. С Довбиш, *Основи проектування інтелектуальних систем*, Суми: СумДУ, 2009.
2. Довбиш А.С. Інтелектуальні інформаційні технології в електронному навчанні / А.С. Довбиш, А.В. Васильєв, В.О. Любчак. – Суми: Видавництво СумДУ, 2013.– 177 с
3. Naumenko, I., Piatachenko, V., Myronenko, M., & Savchenko, T. (2022). Information-Extreme Machine Learning of an On-board Ground Object Recognition System with a Choice of a Base Recognition Class.
4. Dovbysh, A., Shelehov, I., Romaniuk, A., Moskalenko, R., & Savchenko, T. (2023). Decision-making support system for diagnosis of oncopathologies by histological images. *Journal of Pathology Informatics*, 14, 100193.
5. Protsenko, O., Savchenko, T., Myronenko, M., & Prikhodchenko, O. (2020, May). Informational and extreme machine learning for onboard recognition system of ground objects. In 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 213-218). IEEE.
6. Naumenko, I., Myronenko, M., & Savchenko, T. (2021). Information-extreme machine training of on-board recognition system with optimization of RGB-component digital images. *Radioelectronic and Computer Systems*, (4), 59-70.
7. Cancer Research UK // Multiparametric MRI (mpMRI) scan for prostate cancer. URL: <https://www.cancerresearchuk.org/about-cancer/tests-and-scans/multiparametric-mri> (10.05.2023)
8. Parker C, Castro E, Fizazi K, Heidenreich A, Ost P, Procopio G, Tombal B, Gillessen S; ESMO Guidelines Committee. Prostate cancer: ESMO Clinical Practice Guidelines for diagnosis, treatment and follow-up. *Ann Oncol*. 2020 Sep;31(9):1119-1134. doi: 10.1016/j.annonc.2020.06.011. Epub 2020 Jun 25. PMID: 32593798.

9. Pesapane, F.; Acquasanta, M.; Meo, R.D.; Agazzi, G.M.; Tantrige, P.; Codari, M.; Schiaffino, S.; Patella, F.; Esseridou, A.; Sardanelli, F. Comparison of Sensitivity and Specificity of Biparametric versus Multiparametric Prostate MRI in the Detection of Prostate Cancer in 431 Men with Elevated Prostate-Specific Antigen Levels. *Diagnostics* 2021, 11, 1223. <https://doi.org/10.3390/diagnostics11071223>

10. Turkbey B, Rosenkrantz AB, Haider MA, Padhani AR, Villeirs G, Macura KJ, Tempany CM, Choyke PL, Cornud F, Margolis DJ, Thoeny HC, Verma S, Barentsz J, Weinreb JC. Prostate Imaging Reporting and Data System Version 2.1: 2019 Update of Prostate Imaging Reporting and Data System Version 2. *Eur Urol.* 2019 Sep;76(3):340-351. doi: 10.1016/j.eururo.2019.02.033. Epub 2019 Mar 18. PMID: 30898406.

11. Farahani, Navid, Anil V. Parwani, and Liron Pantanowitz. "Whole slide imaging in pathology: advantages, limitations, and emerging perspectives." *Pathol Lab Med Int* 7.23-33 (2015): 4321.

12. Glaser et al. (2019). Multi-immersion open-top light-sheet microscope for high-throughput imaging of cleared tissue. *Nat Commun*, Vol. 10, Article No. 2781

13. Prostate cancer foundation // Gleason Score and Grade Group. URL: <https://www.pcf.org/about-prostate-cancer/diagnosis-staging-prostate-cancer/gleason-score-isup-grade/> (10.05.2023)

14. WebPathology Visual Survey of surgical pathology // Prostate. URL: <https://www.webpathology.com/category.asp?category=1> (10.05.2023)

15. J. T. Kwak and S. M. Hewitt, "Nuclear architecture analysis of prostate cancer via convolutional neural networks," *IEEE Access*, vol. 5, pp. 18526–18533, 2017

16. E. Arvaniti, K. S. Fricker, M. Moret, N. Rupp, T. Hermanns, C. Fankhauser, N. Wey, P. J. Wild, J. H. Rüschoff, and M. Claassen, "Automated Gleason grading of prostate cancer tissue microarrays via deep learning," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, Dec. 2018

17. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979
18. O. J. del Toro, M. Atzori, S. Otálora, M. Andersson, K. Eurén, M. Hedlund, P. Rönquist, and H. Müller, "Convolutional neural networks for an automatic classification of prostate tissue slides with high-grade Gleason score," *Proc. SPIE*, vol. 10140, Mar. 2017, Art. no. 101400O
19. Савченко, Т. Р. "Інформаційна технологія розпізнавання об'єктів. Машинне навчання бортової системи розпізнавання наземних об'єктів." (2021).
20. Гриненко, О. В. Моделі та методи інформаційної технології машинного навчання бортової системи безпілотного літального апарату з використанням симулятора. MS thesis. Сумський державний університет, 2022.
21. Dovbysh, A., Naumenko, I., Myronenko, M., & Savchenko, T. (2020). Information-extreme machine learning on-board recognition system of ground objects with the adaptation of the input mathematical description. In *CMIS* (pp. 913-925).
22. G. Litjens, C. I. Sánchez, N. Timofeeva, M. Hermsen, I. Nagtegaal, I. Kovacs, C. H. van de Kaa, P. Bult, B. van Ginneken, and J. van der Laak, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," *Sci. Rep.*, vol. 6, no. 1, Sep. 2016, Art. no. 2686
23. Csiszar, I. "I-Divergence Geometry of Probability Distributions and Minimization Problems". *Ann. Probab.* 3 (1): 146–158. doi:10.1214/aop/1176996454
24. M. Khan et al., "A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1729–1738, Jun. 2014. [Online]. Available: <http://dx.doi.org/10.1109/TBME.2014.2303294> [26] M. Macenko et al., "A method for normalizing histology slides for quantitative analysis," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano to Macro*, Jun. 2009, pp. 1107–1110. [Online]. Available: [http:// dx.doi.org/10.1109/ISBI.2009.5193250](http://dx.doi.org/10.1109/ISBI.2009.5193250)

25. Rabinovich et al., "Unsupervised color decomposition of histologically stained tissue samples," in Proc. Annu. Conf. Neural Inf. Process. Syst., 2004, pp. 667–674. [Online]. Available: <http://papers.nips.cc/paper/2497-unsupervised-color-decomposition-of-histologically-stained-tissue-samples.pdf>

26. Khater, Ismail M., Ivan Robert Nabi, and Ghassan Hamarneh. "A review of super-resolution single-molecule localization microscopy cluster analysis and quantification methods." *Patterns* 1.3 (2020): 100038.

27. Dovbysh, A., Savchenko, T., Moskalenko, R., & Romaniuk, A. (2022, September). The application of artificial intelligence in the diagnosis of prostate cancer. In *VIRCHOWS ARCHIV* (Vol. 481, No. SUPPL 1, pp. S302-S303). ONE NEW YORK PLAZA, SUITE 4600, NEW YORK, NY, UNITED STATES: SPRINGER.

28. Denysenko, A. P., Savchenko, T. R., Dovbysh, A. S., Romaniuk, A. M., & Moskalenko, R. A. (2022). *Artificial Intelligence Approach in Prostate Cancer Diagnosis: Bibliometric Analysis*.

29. Рябцева, С. Н., Ковалев, В. А., Малышев, В. Д., Семеник, И. А., Дервянко, М. А., Москаленко, Р. А., ... & Романюк, А. Н. (2020). Разработка алгоритма поиска опухолевых областей на основе обработки полнослайдовых гистологических изображений рака молочной железы. Доклады Белорусского государственного университета информатики и радиоэлектроники, 18(8), 21-28.

30. Зимовець В. І. Моделі, методи та інформаційна технологія функціонального діагностування багатоканатної шахтної підйомної машини [Електронний ресурс] : автореф. дис. ... канд. техн. наук : спец. 05.13.06 / Вікторія Ігорівна Зимовець ; [наук. керівник Довбиш А. С.] ; Нац. техн. ун-т "Харків. політехн. ін-т". – Харків, 2021. – 20 с. – Бібліогр.: с. 17-19. – укр.



## ДОДАТОК

*imageSplitting*

```

using System.Drawing;
using System.Drawing.Imaging;

namespace imageSplitting
{
    class Program
    {
        static private Bitmap optimal_map_exam(Bitmap map, int W, int H)
        {
            Size new_size = new Size();
            for (int i = 0; ; i++)
            {
                //ширина карты, что подходит для этого кадра
                if ((map.Width - i) % W == 0)
                {
                    new_size.Width = map.Width - i;
                    break;
                }
            }
            for (int i = 0; ; i++)
            {
                //высота карты, что подходит для этого кадра
                if ((map.Height - i) % H == 0)
                {
                    new_size.Height = map.Height - i;
                    break;
                }
            }

            Bitmap result = new Bitmap(map, new_size.Width, new_size.Height);

            return result;
        }
        static void Main(string[] args)
        {
            int x = 0, y = 0;
            int count = 0;
            int N = 35;
            int n = 35;
            var mainMap = new Bitmap(SplittingImage.Properties.Resources.G0_1);
            mainMap = optimal_map_exam(mainMap, N, n);

            while (count <= (mainMap.Width * mainMap.Height) / (n * N))
            {
                Bitmap bitmap = new Bitmap(N, n);

                using (var gr = Graphics.FromImage(bitmap))
                    gr.DrawImage(mainMap, 0 - x, 0 - y);

                bitmap.Save(count.ToString() + ".JPG", ImageFormat.Jpeg);

                count++;
                x += N;
                if (x >= mainMap.Width)
                {
                    x = 0;
                    if (y >= mainMap.Height - n) break;
                    else y += n;
                }
            }
        }
    }
}

```

```

    }
}
}
}

```

### *SimpleClassifier*

```

using machine_learning;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;

namespace SimpleClassifier
{
    class Program
    {
        static void Main(string[] args)
        {
            machine_learn topHeap = new machine_learn();
            List<string> path = new List<string>() {
                @"E:\MyDev\magistrWork\PythonScript\class1",
                @"E:\MyDev\magistrWork\PythonScript\class2"};
            classSelection selectionTopHeap = new classSelection(path);
            selectionTopHeap.findClasses(4);

            topHeap.make_fragment(selectionTopHeap.getBestClass(0));
            topHeap.make_fragment(selectionTopHeap.getBestClass(1));
            topHeap.realization_decart(true);

            machine_learn subClass = new machine_learn();
            splittingExam(topHeap, 0, "SubClassFragments\\");
            classSelection selectionSubClass = new
classSelection("SubClassFragments\\");
            selectionSubClass.findClasses(3, selectionTopHeap.getBestClass(0));
            subClass.make_fragment(selectionSubClass.getBestClass(0));
            subClass.make_fragment(selectionSubClass.getBestClass(1));
            subClass.realization_decart(true);

            selectionTopHeap.getBestClass(0).Save("1.1" + ".JPG", ImageFormat.Jpeg);
            selectionTopHeap.getBestClass(1).Save("2.1" + ".JPG", ImageFormat.Jpeg);
            selectionSubClass.getBestClass(0).Save("1.2" + ".JPG", ImageFormat.Jpeg);
            selectionSubClass.getBestClass(1).Save("2.2" + ".JPG", ImageFormat.Jpeg);

            exam(topHeap, subClass);
        }
        static private Bitmap optimal_map_exam(Image map, int N, int n)
        {
            Size new_size = new Size();
            for (int i = 0; ; i++)
            {
                if ((map.Width - i) % N == 0)
                {
                    new_size.Width = map.Width - i;
                    break;
                }
            }
            for (int i = 0; ; i++)

```

```

        {
            if ((map.Height - i) % n == 0)
            {
                new_size.Height = map.Height - i;
                break;
            }
        }

        Bitmap result = new Bitmap(map, new_size.Width, new_size.Height);

        return result;
    }
    static void splittingExam(machine_learn machine, int interest, string directory)
    {
        int x = 0, y = 0;
        int count = 0;
        int N = machine.getDecart(0).getColumns();
        int n = machine.getDecart(0).getRows();
        Bitmap mainMap = new Bitmap(optimal_map_exam(Properties.Resources.G0, N,
n));
        while (count <= (mainMap.Width * mainMap.Height) / (n * N))
        {
            Bitmap bitmap = new Bitmap(N, n);

            using (var gr = Graphics.FromImage(bitmap))
                gr.DrawImage(mainMap, 0 - x, 0 - y);

            int res = machine.exams_decart(bitmap);
            if(res == interest)
                bitmap.Save(directory + count.ToString() + ".JPG",
ImageFormat.Jpeg);

            count++;
            x += N;
            if (x >= mainMap.Width)
            {
                x = 0;
                if (y >= mainMap.Height - n) break;
                else y += n;
            }
        }
    }
    static void exam(machine_learn topHeap, machine_learn subClass) {
        int x = 0, y = 0;
        int count = 0;
        int N = topHeap.getDecart(0).getColumns();
        int n = topHeap.getDecart(0).getRows();

        Bitmap mainMap = new Bitmap(optimal_map_exam(Properties.Resources.G3, N,
n));
        Bitmap numberMap = new Bitmap(mainMap);
        using (StreamWriter sw = new StreamWriter("map.txt"))
        {
            string output = "";
            while (count <= (mainMap.Width * mainMap.Height) / (n * N))
            {
                int res;
                Bitmap bitmap = new Bitmap(N, n);

                using (var gr = Graphics.FromImage(bitmap))
                    gr.DrawImage(mainMap, 0 - x, 0 - y);

                res = topHeap.exams_decart(bitmap);
                if (res == 0)
                {

```

```

        res = subClass.exams_decart(bitmap);
        if (res == 1) res = 2;
    }
    output += " " + res.ToString();

    make_fragment_exam_map(res, x, y, ref numberMap);

    count++;
    x += N;
    if (x >= mainMap.Width)
    {
        x = 0;
        if (y >= mainMap.Height - n) break;
        else y += n;

        sw.WriteLine(output);
        output = "";
    }
    }
    sw.WriteLine(output);
}

numberMap.Save("resNum.jpg", ImageFormat.Bmp);
}
static void make_fragment_exam_map(int res, int x, int y, ref Bitmap map_num)
{
    Graphics gr = Graphics.FromImage(map_num);
    gr.DrawString($"{res + 1}", new Font("Segoe UI", 31, FontStyle.Bold), new
SolidBrush(Color.White), x + 7, y);
    gr.DrawString($"{res + 1}", new Font("Segoe UI", 25, FontStyle.Bold), new
SolidBrush(Color.Black), x + 10, y + 4);
}
}
}
}

```

### *classSelection*

```

using machine_learning;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SimpleClassifier
{
    class classSelection
    {
        List<List<decart_coord>> fragments = new List<List<decart_coord>>();
        List<Bitmap> bestCouple = new List<Bitmap>();

        public Bitmap getBestClass(int i) {
            return bestCouple[i];
        }
        public classSelection(List<string> paths) {
            foreach(string path in paths){
                List<decart_coord> bitmaps = new List<decart_coord>();
                foreach (string pathImg in Directory.GetFiles(path, "*.JPG",
SearchOption.AllDirectories)) {
                    Bitmap bm = new Bitmap(pathImg);
                    decart_coord decart = new decart_coord(bm);
                    bitmaps.Add(decart);
                }
            }
        }
    }
}

```

```

        fragments.Add(bitmaps);
    }
}
public class Selection(string path) {
    List<decart_coord> bitmaps = new List<decart_coord>();
    foreach (string pathImg in Directory.GetFiles(path, "*.JPG",
SearchOption.AllDirectories))
    {
        Bitmap bm = new Bitmap(pathImg);
        decart_coord decart = new decart_coord(bm);
        bitmaps.Add(decart);
    }
    fragments.Add(bitmaps);
}
List<decart_coord> sort(List<decart_coord> sample) {
    int len = sample.Count();
    for (int i = 1; i < len; i++)
    {
        for (int j = 0; j < len - i; j++)
        {
            if (sample[j].getDispersion() > sample[j + 1].getDispersion()) {
                var tmp = sample[j];
                sample[j] = sample[j + 1];
                sample[j + 1] = tmp;
            }
        }
    }
    return sample;
}
float averageValueInterval(List<decart_coord> interval) {
    float avr = 0;
    foreach (decart_coord elem in interval)
        avr += elem.getAvr();
    return avr / interval.Count();
}
void findBestCouple(Bitmap left, List<decart_coord> right)
{
    float maxE = -1;
    int radius = -1;

    foreach (decart_coord rightElem in right)
    {
        machine_learn machine = new machine_learn();
        machine.make_fragment(left);
        machine.make_fragment(rightElem.getImg());

        machine.realization_decart(false);
        if (machine.getTeachDecart().average_KFE() > maxE && radius <
machine.getTeachDecart().getRadius(0))
        {
            maxE = machine.getTeachDecart().average_KFE();
            radius = machine.getTeachDecart().getRadius(0);
            bestCouple = new List<Bitmap>();

            bestCouple.Add(left);
            bestCouple.Add(rightElem.getImg());
        }
    }
}
public void findClasses(int intervals, Bitmap bestClass1)
{
    sort(fragments[0]);
    float step = (fragments[0][fragments[0].Count() - 1].getDispersion() -
fragments[0][0].getDispersion()) / intervals;
}

```

```

List<List<decart_coord>> spacing = new List<List<decart_coord>>();
List<decart_coord> space = new List<decart_coord>();
int numInterval = 1;
foreach (decart_coord selection in fragments[0])
{
    if (selection.getDispertion() <= fragments[0][0].getDispertion() +
(step * numInterval))
        space.Add(selection);
    else
    {
        spacing.Add(space);
        space = new List<decart_coord>();
        space.Add(selection);
        numInterval++;
    }
}
List<decart_coord> sampleAvr = new List<decart_coord>();
foreach (List<decart_coord> interval in spacing)
{
    float avr = averageValueInterval(interval);
    decart_coord neighbourAvr = new decart_coord();
    float neighbourError = 9999999;

    foreach (decart_coord elem in interval)
    {
        if (Math.Abs((avr - elem.getDispertion())) < neighbourError)
        {
            neighbourAvr = elem;
            neighbourError = Math.Abs((avr - elem.getDispertion()));
        }
    }
    sampleAvr.Add(neighbourAvr);
}

findBestCouple(bestClass1, sampleAvr);
}
void findBestCouple(List<decart_coord> left, List<decart_coord> right)
{
    float maxE = -1;
    int radius = 999;
    foreach (decart_coord leftElem in left)
    {
        foreach (decart_coord rightElem in right)
        {
            machine_learn machine = new machine_learn();
            machine.make_fragment(leftElem.getImg());
            machine.make_fragment(rightElem.getImg());

            machine.realization_decart(false);
            if (machine.getTeachDecart().average_KFE() > maxE && radius >
machine.getTeachDecart().getRadius(0))
            {
                maxE = machine.getTeachDecart().average_KFE();
                radius = machine.getTeachDecart().getRadius(0);
                bestCouple = new List<Bitmap>();

                bestCouple.Add(leftElem.getImg());
                bestCouple.Add(rightElem.getImg());
            }
        }
    }
}
}
public void findClasses(int intervals) {
    List<List<decart_coord>> semiClasses = new List<List<decart_coord>>();
    foreach (List<decart_coord> sample in fragments) {

```



```

}

public void realization_decart(bool posl)
{
    teachDecart = new teach_decart(ref decarts);

    teachDecart.parallel_dopusk();
    if (posl)
        teachDecart.posledovatel_dopusk();
}

public void make_fragment(Bitmap bitmap)
{
    decart_coord decart = new decart_coord(bitmap);

    for (int i = 0; i < decart.getColumns(); i++)
    {
        for (int j = 0; j < decart.getRows(); j++)
        {
            Color color = bitmap.GetPixel(i, j);
            int pixel = (color.R + color.G + color.B) / 3;

            decart.setBitMap(i, j, pixel);
        }
    }

    decarts.Add(decart);
}

public decart_coord examDecart;
public int exams_decart(Bitmap bitmap) {
    float[] res = new float[decarts.Count];
    examDecart = new decart_coord(bitmap);

    for (int i = 0; i < examDecart.getColumns(); i++)
    {
        for (int j = 0; j < examDecart.getRows(); j++)
        {
            Color color = bitmap.GetPixel(i, j);
            int pixel = (color.R + color.G + color.B) / 3;
            examDecart.setBitMap(i, j, pixel);
        }
    }
    examDecart.makeBM(teachDecart.getVD(), teachDecart.getND());

    for (int k = 0; k < decarts.Count; k++)
    {
        res[k] = 0;
        if (teachDecart.getRadius(k) != 0)
        {
            for (int j = 0; j < examDecart.getColumns(); j++)
            {
                int d_exam = 0;
                for (int i = 0; i < examDecart.getRows(); i++)
                    d_exam += Math.Abs(decarts[k].getElemEV(i) -
examDecart.getElemBM(i, j));
                res[k] += 1 - (d_exam / teachDecart.getRadius(k));
            }
            res[k] /= examDecart.getRows();
        }
    }
    int k_max = -1;
    double mu_max = 0;
    for (int i = 0; i < decarts.Count; i++)
    {
        if (mu_max < res[i])

```



```

        {
            mu_max = res[i];
            k_max = i;
        }
    }
    return k_max;
}
}
}

```

## decart\_coord

```

using System;
using System.Drawing;

namespace machine_learning
{
    public class decart_coord
    {
        Bitmap img;
        int[,] bitMap;
        int[,] binaryMap;
        int[] etalonVektor;

        int N;
        int n;
        float ro = 0.5f;
        float dispertion;
        float avr;

        int[,] RGB;
        public decart_coord(Bitmap source)
        {
            img = source;

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

            etalonVektor = new int[N];
            binaryMap = new int[N, n];
            bitMap = new int[N, n];

            RGB = new int[N, 3];

            make_RGB();
        }
        public decart_coord() {
        }
        public Bitmap getImg() {
            return img;
        }
        public int getColumns() {
            return N;
        }
        public int getRows()
        {
            return n;
        }
        public void setBitMap(int i, int j, int val) {
            bitMap[i, j] = val;
        }
        public int getElemBitMap(int i, int j) {

```

```

        return bitMap[i, j];
    }
    public int[,] getBitmap()
    {
        return bitMap;
    }
    public int[,] getBM() {
        return binaryMap;
    }
    public void setBitmap(int[,] bitmap) {
        for (int i = 0; i < N; i++)
            for (int j = 0; j < n; j++)
                bitMap[i, j] = bitmap[i, j];
    }
    public int getElemEV(int i) {
        return etalonVektor[i];
    }
    public int getElemBM(int i, int j)
    {
        return binaryMap[i, j];
    }
    public float getDispertion() {
        return dispertion;
    }
    public float getAvr()
    {
        return avr;
    }
    public int getRGB(int i, int j) {
        return RGB[i, j];
    }
    public void makeEV()
    {
        for (int i = 0; i < N; i++)
        {
            float ev_sum = 0;
            for (int j = 0; j < n; j++)
                ev_sum += binaryMap[i, j];
            if ((ev_sum / n) >= ro) etalonVektor[i] = 1;
            else etalonVektor[i] = 0;
        }
    }
    public void makeBM(int[] vd, int[] nd)
    {
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < n; j++)
            {
                int tmp = bitMap[i, j];
                if (bitMap[i, j] >= nd[i] && bitMap[i, j] <= vd[i])
                    binaryMap[i, j] = 1;
                else
                    binaryMap[i, j] = 0;
            }
        }
    }
    void make_RGB()
    {
        avr = 0;
        for (int i = 0; i < N; i++)
        {
            int R = 0, G = 0, B = 0;
            for (int j = 0; j < n; j++)
            {
                Color color = img.GetPixel(i, j);

```

```

        R += color.R;
        G += color.G;
        B += color.B;
    }
    RGB[i, 0] = R / n;
    RGB[i, 1] = G / n;
    RGB[i, 2] = B / n;

    avr += (RGB[i, 0] + RGB[i, 1] + RGB[i, 2]) / 3;
}
avr /= N;

float distance = 0;
for (int i = 0; i < N; i++)
    distance += (float)Math.Pow(((RGB[i, 0] + RGB[i, 1] + RGB[i, 2]) / 3)
- avr, 2);
    dispersion = distance / N;
}
}
}

```

### *teach\_decart*

```

using System;
using System.Collections.Generic;

namespace machine_learning
{
    public class teach_decart : AbstractTeach
    {
        List<decart_coord> decarts = new List<decart_coord>();
        int[] para_d;
        int[] para;
        int[,] sk;
        int[,] sk_para;
        float[] em;
        int[] my_do;

        int[] vd;
        int[] nd;

        int limitD = 80;
        int d;
        int used_class = 0;
        int[] d_posl;

        public teach_decart(ref List<decart_coord> sender)
        {
            foreach (var item in sender)
                decarts.Add(item);
            sk = new int[2, decarts[0].getRows()];
            sk_para = new int[2, decarts[0].getRows()];

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

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

            vd = new int[decarts[0].getColumns()];
            nd = new int[decarts[0].getColumns()];

            d_posl = new int[decarts[0].getColumns()];

```

```

        sender = decarts;
    }
    public decart_coord getDecart(int i) {
        return decarts[i];
    }
    public override int getColumns() {
        return decarts[0].getColumns();
    }
    public override int getRows()
    {
        return decarts[0].getRows();
    }
    public override int getRadius(int i) {
        return my_do[i];
    }
    public float getAverageRadius()
    {
        int avr = 0;
        for (int k = 0; k < decarts.Count; k++)
            avr += my_do[k];
        return avr/decarts.Count;
    }
    public int[] getVD() {
        return vd;
    }
    public int[] getND()
    {
        return nd;
    }
    public override int getDelta() {
        return d;
    }
    public override float getEM(int i) {
        return em[i];
    }
    public override int getLimitD() {
        return limitD;
    }
    public override int getElemRGB(int k, int i, int j) {
        return decarts[k].getRGB(i, j);
    }
    void realization()
    {
        foreach (var decart in decarts)
            decart.makeBM(vd, nd);
        for (int k = 0; k < decarts.Count; k++)
            decarts[k].makeEV();

        make_para();
        make_myDo();
    }
    bool make_dopusk(int k, int delta)
    {
        for (int i = 0; i < decarts[k].getColumns(); i++)
        {
            int sum = 0;
            for (int j = 0; j < decarts[k].getRows(); j++)
                sum += decarts[k].getElemBitMap(i, j);

            sum /= decarts[k].getRows();

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

            if (vd[i] > 255 || nd[i] < 0)

```

```

        {
            limitD = delta;
            return false;
        }
    }
    return true;
}
void make_para()
{
    for (int k = 0; k < decarts.Count; k++)
    {
        para_d[k] = 0;
        int ev_sum1 = decarts[k].getColumns();
        for (int k1 = 0; k1 < decarts.Count; k1++)
        {
            if (k1 != k)
            {
                int ev_sum = 0;
                for (int i = 0; i < decarts[k].getColumns(); i++)
                {
                    if (decarts[k].getElemEV(i) != decarts[k1].getElemEV(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 < decarts[k].getRows(); INFK_j++)
    {
        if (sk[0, INFK_j] <= INFK_d) k1++;
        if (sk[1, INFK_j] <= INFK_d) k4++;
    }
    INFK_d1 = (float)k1 / (float)decarts[k].getRows();
    INFK_beta = (float)k4 / (float)decarts[k].getRows();

    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 < decarts[k].getRows(); j++)
    {
        sk[1, j] = 0;
        sk[0, j] = 0;
        sk_para[1, j] = 0;
        sk_para[0, j] = 0;
        for (int i = 0; i < decarts[k].getColumns(); i++)
        {
            int pixel = decarts[0].getElemBitMap(i, j);
            if (decarts[k].getElemEV(i) != decarts[para[k]].getElemBM(i, j))
                sk[1, j] += 1;

            int EV = decarts[k].getElemEV(i);

```

```

        int second = decarts[k].getElemBM(i, j);
        if (decarts[k].getElemEV(i) != decarts[k].getElemBM(i, j))
            sk[0, j] += 1;

        if (decarts[para[k]].getElemEV(i) != decarts[para[k]].getElemBM(i,
j))
            sk_para[1, j] += 1;
        if (decarts[para[k]].getElemEV(i) != decarts[k].getElemBM(i, j))
            sk_para[0, j] += 1;
    }
}
}
void make_myDo()
{
    float t_d1 = 0, t_beta = 0;

    for (int k = decarts.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 < decarts[k].getColumns(); 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)
            {
                d_correct = (float)d / para_d[k];
                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;
}
public float average_KFE()
{
    float avr = 0;
    for (int k = 0; k < decarts.Count; k++)
        avr += em[k];
    avr /= decarts.Count;
    return avr;
}
public void parallel_dopusk()
{

```

```

int now_class = 0;
float em_last_d = 0;
float radius = 0;

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

    float em_now = average_KFE();
    if (propusk_KFE(my_do[now_class], now_class))
    {
        float t_d1 = 0, t_beta = 0;
        INFK(my_do[now_class], ref t_d1, ref t_beta, now_class);
    }

    if (em_now > em_last_d && propusk_KFE(my_do[now_class], now_class))
    {
        em_last_d = em_now;
        d = delta;
        radius = my_do[now_class];
    }
    else if (em_now == em_last_d && radius < my_do[now_class] &&
propusk_KFE(my_do[now_class], now_class))
    {
        em_last_d = em_now;
        d = delta;
        radius = my_do[now_class];
    }
}

make_dopusk(now_class, d);
realization();
}

bool local_dopusk(int i, int delta)
{
    int sum = 0;
    for (int j = 0; j < decarts[used_class].getRows(); j++)
        sum += decarts[used_class].getElemBitMap(i, j);

    sum /= decarts[used_class].getRows();

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

    if (vd[i] > 255 || nd[i] < 0)
        return false;
    return true;
}

public void posledovatel_dopusk()
{
    float last_res_G = 0, now_res_G = 0;

    for (int i = 0; i < decarts[used_class].getColumns(); i++)
        d_posl[i] = d;
    now_res_G = average_KFE();

    do
    {
        last_res_G = now_res_G;

        for (int i = 0; i < decarts[used_class].getColumns(); i++)
        {
            float last_res_L = 0;

```

```

float radius = 0;

for (int delta = 1; delta < limitD; 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[used_class];
    }

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

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

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

```