

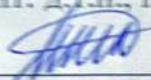
Вінницький національний технічний університет
Факкультет інформаційних електронних систем
Кафедра біомедичної інженерії та оптико-електронних систем

Бакалаврська кваліфікаційна робота на тему:

ДОСЛІДЖЕННЯ ТА РОЗРОБКА СИСТЕМИ ДІАГНОСТИКИ РАКУ
МОЛОЧНОЇ ЗАЛОЗИ

Виконав: студент 4 курсу,
групи БМІ-206,
спеціальності 163 Біомедична інженерія
 Подольнчук П. С.

Керівник: д-р філос., ст. викл. каф. БМІОЕС
 Карась О. В.
« 12 » 06 2024 р.

Рецензент: д.т.н., проф., професор каф. ІРТС
 Семенов А. О.
« 14 » 06 2024 р.

Допущено до захисту

Зав. кафедри БМІОЕС

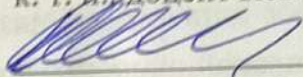
Коваль Л. Г. 

« 12 » 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інформаційних електронних систем
Кафедра біомедичної інженерії та оптико-електронних систем
Бакалавр
Спеціальність 163 Біомедична інженерія

ЗАТВЕРДЖУЮ
завідувач кафедри БМІОЕС
к. т. н., доцент Коваль Л. Г.

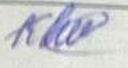

«11» 03 2024 р.

ЗАВДАННЯ

на бакалаврську кваліфікаційну роботу
студенту Подолянчуку Павлу Сергійовичу, гр. БМІ-206

1. Тема БДР «Дослідження та розробка системи діагностики раку молочної залози», керівник роботи ст. викл. каф. БМІОЕС Карась Олександр Володимирович, д-р філософії, затверджені наказом ВНТУ від 11 березня 2024 року № 80.
2. Строк подання студентом роботи – до 05.06.2024 р.
3. Вихідні дані для роботи: система на основі нейронної мережі, база даних зображень BreakHis, розподіл зображень на навчальну і тестувальну вибірку, інженерна і наукова література за темою.
4. Зміст роботи (перелік питань, які потрібно розробити): 1. Аналітичний огляд питання; 2. Розроблення системи класифікації гістологічних зображень; 3. Дослідження системи на базі даних зображень; 4. Опрацювання питань охорони праці
5. Зміст ілюстративної частини (з точним зазначенням обов'язкових креслеників): не передбачена.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Спеціальна частина	Карась О. В. д-р філос., ст. викл. кафедри БМІОЕС	 11.03.2024	 05.06.2024
Охорона праці	Дембіцька С. В., д. п. н., професор кафедри БЖДПБ	 18.03.2024	 25.05.2024

7. Дата видачі завдання 11 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів БДР	Строк виконання етапів БДР	Примітки
1	Формування та затвердження теми бакалаврської дипломної роботи (БДР)	До 10.03.2024	
2	Виконання спеціальної частини БДР. Перший рубіжний контроль виконання БДР	До 05.04.2024	
3	Виконання спеціальної частини БДР. Другий рубіжний контроль виконання БДР	До 10.05.2024	
4	Виконання розділу «Охорона праці»	До 26.05.2024	
5	Попередній захист БДР	05.06.2024	
6	Нормоконтроль БДР	12.06.2024	
7	Рецензування БДР	18.06.2024	
8	Захист БДР	19.06.2024	

Студент


(підпис)

Подольчук П. С.

Керівник роботи


(підпис)

Карась О. В.

АНОТАЦІЯ

Подольанчук П. С. БДР на тему «Дослідження та розробка системи діагностики раку молочної залози». Вінниця : ВНТУ, 2024. 81 с.

Українською мовою. 8 рис., 13 табл., бібліогр. 40 найм.

В бакалаврській кваліфікаційній роботі розроблено автоматизовану інформаційну медичну систему підтримки прийняття рішень на основі нейромережових технологій для діагностування раку молочної залози за допомогою класифікації гістологічних мікроскопічних зображень. В роботі розглянуто сучасний стан розвитку інформаційних систем медичної діагностики, проведено аналіз інформаційних технологій класифікації зображень, реалізовано і досліджено систему автоматизованої діагностики, точність якої склала 90-92 %.

ABSTRACT

Podolianchuk P. S. Bachelor thesis on the topic «Research and development of a breast cancer diagnosis system». Vinnytsia : VNTU, 2024. 81 p.

In Ukrainian. Figs. 8, Tables 13, Refs. 40.

In the bachelor's thesis, an automated information medical decision support system was developed based on neural network technologies for diagnosing breast cancer using the classification of histological microscopic images. The paper examines the current state of development of medical diagnostic information systems, analyzes image classification information technologies, implements and investigates an automated diagnostic system, the accuracy of which was 90-92 %.

ЗМІСТ

ВСТУП	5
1 ОГЛЯД СУЧАСНИХ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗОВАНОГО ДІАГНОСТУВАННЯ РАКУ МОЛОЧНОЇ ЗАЛОЗИ.....	8
1.1 Рак молочної залози як актуальна проблема.....	8
1.2 Інформатизація в медицині	12
1.3 Методи діагностики раку молочної залози	13
1.4 Сучасні технології автоматизованого аналізу і класифікації гістологічних зображень	15
2 РОЗРОБЛЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ДІАГНОСТУВАННЯ РАКУ МОЛОЧНОЇ ЗАЛОЗИ.....	19
2.1 Матеріали дослідження	19
2.2 Модель NEuro Fuzzy CLASSifier	20
2.3 Метод найкоротшого спуску.....	23
2.4 Метод диференціальної еволюції	31
2.5 Алгоритм стрибка в басейн.....	32
2.6 Згорткові нейронні мережі	34
3 ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ АНАЛІЗУ ГІСТОЛОГІЧНИХ ЗОБРАЖЕНЬ	40
3.1 Методологія дослідження	40
3.2 Експериментальна частина і обговорення результатів	40
4 ОХОРОНА ПРАЦІ	45
4.1 Технічні рішення щодо безпечного виконання роботи.....	45
4.2 Технічні рішення з гігієни праці та виробничої санітарії	48
4.2.1 Мікроклімат	48
4.2.2 Склад повітря робочої зони.....	49
4.2.3 Виробниче освітлення	50
4.2.4 Виробничий шум.....	51

4.2.5 Виробничі випромінювання.....	52
4.3 Пожежна безпека.....	53
4.3.1 Технічні рішення системи запобігання пожежі	54
4.3.2 Технічні рішення системи протипожежного захисту.....	55
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТКИ.....	63
Додаток А (довідковий) Приклади зображень з бази даних BreakHIS	64
Додаток Б (довідковий) Програмна реалізація системи	70
Додаток В (обов'язковий) Протокол перевірки на наявність текстових запозичень	80

ВСТУП

При вирішенні задач діагностики захворювань у медицині значну частину проблематики становить виділення діагностично значимих ознак для подальшої обробки даних і вибір методу класифікації ознак, за допомогою яких діагностуються різні патологічні стани. З розвитком і розповсюдженням систем підтримки прийняття рішень зростають вимоги до алгоритмів навчання і точності діагностики з їх допомогою. Надійність системи та простота використання впливають на стабільність і швидкість прийняття рішень, що може сприяти швидшому призначенню лікування та підвищенню шансів на його успішний результат. Перевага автоматизованих систем діагностики полягає в тому, що вони частково вирішують цю проблему. Методи класифікації і візуалізації дуже гнучкі та справляються із завданнями з невеликими витратами ресурсів.

Наразі медицина поступово збільшує використання машинного навчання, що створює нові професії у цій галузі. Медична галузь та охорона здоров'я зараз перебувають в процесі змін та швидко реагують на можливості нових технологій. Незважаючи на порівняно молодий вік медичної інформатики, яка існує менше 40 років, сьогодення медицина неможлива без інформаційних технологій, які стали невід'ємною частиною сімейної медицини, страхової медицини, єдиного інформаційного простору охорони здоров'я, її фінансового забезпечення, телемедицини, а також процесів інтеграції в загальноєвропейський медичний простір.

Сьогодні практично кожен етап діагностики потребує застосування інформаційних технологій. Водночас інтелектуальні інформаційні системи поки що обмежено застосовуються в практичній медицині. Інтелектуалізація автоматизованих систем дозволяє розширювати коло завдань, які вирішуються за допомогою комп'ютерів, і підвищувати рівень інтелектуальної підтримки сучасних фахівців-лікаря. Основним завданням використання цих

систем є імітацію роботи експерта в певній галузі. Застосування інтелектуальних систем у медицині безсумнівно сприяє прогресивному розвитку інформаційного потенціалу, який є універсальним засобом вирішення широкого кола завдань на різних етапах діагностики і лікування.

Автоматизовані системи оброблення медичних зображень дозволяють здійснювати детальний аналіз кожного пікселя з метою виявлення важливих діагностичних ознак навіть на ранніх, ще малопомітних стадіях розвитку захворювань. Особливо актуальною така задача є для діагностики ракових новоутворень та інших досліджень в онкології.

Автоматизовані системи можуть аналізувати медичні зображення з високою точністю, допомагаючи виявити навіть найменші ознаки раку. Вони здатні знаходити патерни, які можуть бути пропущені людським оком, що знижує ймовірність помилкових негативних та позитивних результатів.

Системи автоматизованої діагностики можуть швидко обробляти великі обсяги медичних даних, що дозволяє значно скоротити час на постановку діагнозу. Це особливо важливо в умовах великого навантаження на медичні заклади.

Автоматизовані системи можуть бути використані в регіонах з обмеженим доступом до висококваліфікованих медичних фахівців. Це дозволяє надавати високоякісну діагностичну допомогу в малозабезпечених районах, гірській та сільській місцевості.

Інтелектуальні системи можуть виступати як інструмент підтримки прийняття рішень, надаючи лікарям додаткову інформацію та рекомендації. Це сприяє більш обґрунтованим рішенням щодо лікування пацієнтів.

Автоматизація діагностичних процесів може знизити витрати на медичні послуги шляхом підвищення ефективності роботи медичного персоналу та зменшення потреби в дорогих додаткових обстеженнях.

Автоматизовані системи забезпечують стандартизований підхід до діагностики, що зменшує ризик варіабельності результатів між різними медичними закладами та фахівцями.

Збір та аналіз великих обсягів даних за допомогою автоматизованих систем сприяє науковим дослідженням у галузі онкології, допомагаючи виявляти нові підходи до лікування та профілактики раку молочної залози.

Метою бакалаврської кваліфікаційної роботи є дослідження та розроблення алгоритмічно-програмних засобів підтримки прийняття рішень лікарем при діагностиці раку молочної залози у жінок для забезпечення ранньої постановки діагнозу, що в свою чергу підвищує ефективність лікування.

Виходячи з мети, було виділено такі завданнями роботи:

- проаналізувати сучасний стан систем автоматизованої діагностики раку молочної залози;
- запропонувати архітектурну організацію і структуру нейронної мережі для аналізу гістологічних зображень;
- реалізувати нейромережову систему для діагностики раку молочної залози;
- провести дослідження розробленої системи на навчальній та тестовій вибірці гістологічних мікроскопічних зображень.

Об'єкт БКР – процес діагностування онкологічних патологій молочної залози.

Предмет БКР – автоматизована система підтримки прийняття рішень лікарем при діагностиці раку молочної залози на основі нейронних мереж.

1 ОГЛЯД СУЧАСНИХ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗОВАНОГО ДІАГНОСТУВАННЯ РАКУ МОЛОЧНОЇ ЗАЛОЗИ

1.1 Рак молочної залози як актуальна проблема

Рак молочної залози – захворювання, при якому патологічні клітини у тканинах молочної залози починають неконтрольовано ділитися та утворюють пухлину. Без лікування пухлина може поширитися в інші області організму і призвести до смерті. Класифікація на стадії є доволі умовною і описується параметрами, показаними у табл. 1 [1]. Орієнтовний розмір пухлин показано на рис. 1.

Таблиця 1 – Стадії розвитку раку молочної залози

Стадія	0 (нульова)	1 (перша)	2 (друга)	3 (третя)	4 (четверта)
Розмір пухлини	Дуже малий	До 2 см	2–5 см	Понад 5 см	Будь-який
Лімфатичні вузли	Не уражені	Не уражені	Уражені	Уражені, раkdосyгає м'язів та шкіри	Уражені
Метастази	Немає	Немає	Немає	Немає	Метастази у будь-якій частині тіла
Виживання протягом 5 років	100 %	100 %	87 %	61 %	20 %



Рисунок 1 – Орієнтовний розмір пухлини залежно від стадії

Схематичне зображення онкологічного ураження молочних залоз показано на рис. 2.

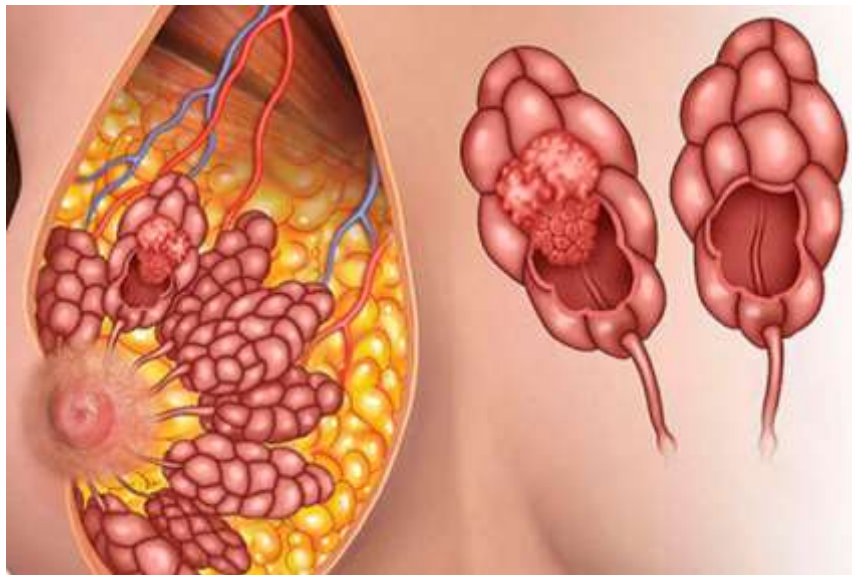


Рисунок 2 – Ураження молочних залоз пухлиною [2]

Пухлинні клітини раку молочної залози починають рости в молочних протоках та/або часточках залозистої тканини. Найбільш рання форма (*in situ*) не є небезпечною для життя і може бути виявлена на ранніх стадіях. Клітини пухлини можуть поширюватися у прилеглі тканини молочної залози (інвазія). У міру подальшого зростання пухлини виникає об'ємне утворення або ущільнення.

Інвазивний рак може поширюватися в лімфатичні вузли або інші органи (метастазувати). Метастази можуть бути небезпечними для життя та смертельними.

Вибір лікування залежить від індивідуальних особливостей пацієнта, типу пухлини та ступеня її поширення. Лікування є комбінованим і включає хірургічне втручання, променеву терапію та прийом препаратів.

У 2022 році рак молочної залози був діагностований у 2,3 мільйонів жінок і викликав 670 тисяч випадків смерті у всьому світі. Рак молочної залози зустрічається у всіх країнах світу у жінок будь-якого віку після досягнення

статевої зрілості, при цьому показники захворюваності збільшуються із віком. Розрахункові дані свідчать про різючі диспропорції у тягарі раку молочної залози в різних країнах залежно від рівня розвитку людського потенціалу. Так, у країнах з дуже високим індексом розвитку людського потенціалу (ІРЛП) рак молочної залози діагностується у кожної 12 жінки протягом життя, а смерть від нього настає у кожної 71 жінки.

З іншого боку, у країнах з низьким ІРЛП рак молочної залози діагностується лише у кожної 27 жінки протягом життя, проте від нього вмирає кожна 48 жінка.

Розрізняють такі типи раку молочної залози:

- інвазивна протокова карцинома;
- інвазивна лобулярна (часточкова) карцинома;
- запальний рак молочної залози;
- рак соска;
- внутрішньопотокова карцинома;
- лобулярна (часточкова) карцинома.

Останні два типи є менш поширеними. На рис. 3 показана орієнтовна локалізація та відмінності типів раку.

Найсерйознішим фактором ризику розвитку раку молочної залози є приналежність до жіночої статі. Приблизно 99% випадків раку молочної залози виникає серед жінок та 0,5–1 % випадків серед чоловіків. Лікування раку грудної залози у чоловіків проводиться відповідно до тих самих принципів ведення захворювання, що і у разі раку молочної залози у жінок.

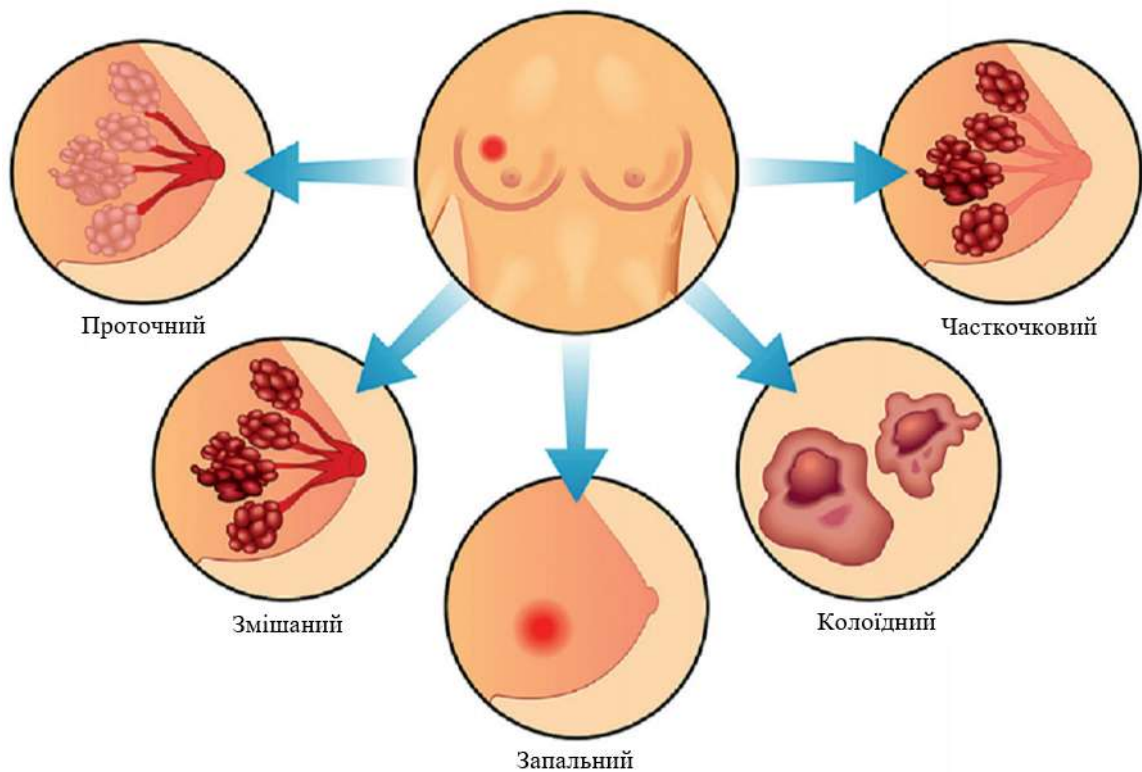


Рисунок 3 – Типи раку молочної залози

Ризик розвитку раку молочної залози підвищується за наявності ряду факторів, таких як більш зрілий вік, ожиріння, шкідливе вживання алкоголю, сімейний анамнез раку молочної залози, анамнез радіаційного впливу, репродуктивний анамнез (наприклад, вік початку менструацій та вік першої вагітності), вживання тютюну та гормональна терапія під час постменопаузи. Приблизно в половині випадків рак молочної залози вражає жінок, які за винятком статевої приналежності (жіночої статі) та віку (старше 40 років) не піддаються впливу інших факторів ризику розвитку раку молочної залози, що чітко визначаються.

Сімейний анамнез раку молочної залози збільшує ризик розвитку захворювання, проте у більшості жінок з діагнованим раком молочної залози немає близьких родичів, які хворіли на цей вид раку [3]. Відсутність підтвердженого сімейного анамнезу раку молочної залози не завжди означає, що жінка наражається на менший ризик.

Ризик розвитку раку молочної залози збільшують певні успадковані високопенетрантні мутації в генах, причому домінантними з них є мутації в генах BRCA1, BRCA2 і PALB-2 [4]. При виявленні мутацій у цих головних генах доцільно розглянути можливість вжиття таких профілактичних заходів, як хірургічне видалення обох молочних залоз чи хіміопрофілактика.

1.2 Інформатизація в медицині

Медичні інформаційні технології, які колись були теоретичними та ексклюзивними для деяких клінік, сьогодні стали невід'ємною частиною повсякденної медичної практики. Інтелектуальна інформаційна система є одним із видів автоматизованих інформаційних систем, яку також називають системою, заснованою на знаннях. У процесі моделювання та побудови таких систем використовується поняття моделі предметної області, яка на основі системи знань забезпечує автоматичний вибір оптимального алгоритму для вирішення завдань.

У загальному випадку всі системи, що використовують знання, можна розділити на системи, які вирішують завдання аналізу, і системи, які вирішують завдання синтезу. Основна відмінність завдань аналізу від завдань синтезу полягає в тому, що в задачах аналізу множина рішень може бути перелічена і включена в систему, тоді як у завданнях синтезу множина рішень потенційно необмежена і будується з рішень компонентів або проблем.

До завдань аналізу належать інтерпретація даних, діагностика, підтримка ухвалення рішень; до завдань синтезу відносяться проєктування, планування, управління. Існують також комбіновані завдання, такі як вивчення, моніторинг, прогнозування. Відомі мови взаємодії та подання знань в інтелектуальних системах (Prolog, Lisp, QBE) зазвичай мають вузькоспеціалізовану спрямованість. У цій роботі будемо розглядати систему для задач медичної діагностики.

Відомо, що особливістю таких систем є автоматизація вибору і прийняття оптимальних рішень на основі попередньо набутого людиною досвіду і раціонального аналізу зовнішніх факторів, описаних в термінах моделі предметної області [5]. Автоматизація процесів медичної діагностики, як один з найважливіших напрямів медицини, відіграє значну роль у підвищенні надійності і точності діагностики захворювань.

1.3 Методи діагностики раку молочної залози

При діагностуванні онкологічних захворювань грудей використовуються такі методи інструментальної діагностики [6, 7, 8]:

- пальпація та зовнішній огляд (важливим є також самообстеження для виявлення будь-яких змін або аномалій в тканині молочних залоз);
- ультразвукове дослідження грудей (використовується для вивчення структури тканин молочних залоз і визначення, чи є виявлене утворення рідким (кіста) або твердим (пухлина));
- біопсія та гістологічне дослідження (включає забір зразків тканини молочної залози для лабораторного аналізу; існує кілька видів біопсії: тонкоголкова аспіраційна біопсія (FNA), біопсія товстою голкою (core biopsy), вакуумна біопсія, хірургічна біопсія);
- мамографія (рентгенологічне обстеження молочних залоз, яке дозволяє виявити пухлини, що не відчуються на дотик; скринінгова мамографія використовується для регулярного обстеження жінок без симптомів; діагностична мамографія проводиться для уточнення аномалій, виявлених під час скринінгу або клінічного обстеження; цифрова мамографія дозволяє створити тривимірне зображення молочних залоз, що дозволяє краще бачити структуру тканин і виявляти пухлини);
- комп'ютерна томографія (КТ);

- біохімічні тести на онкомаркери (визначає наявність специфічних білків у крові);
- імуногістохімічний аналіз (ІНС) – визначає наявність специфічних білків у тканині пухлини, що допомагає встановити підтип раку молочної залози та підібрати відповідне лікування;
- магнітно-резонансна томографія (МРТ) – забезпечує детальне зображення молочних залоз і може бути корисною для обстеження жінок з високим ризиком раку молочної залози або для уточнення результатів інших методів;
- комбіноване дослідження з використанням позитронно-емісійної та комп'ютерної томографії (ПЕТ-КТ), що дає зображення підвищеної точності для локалізації невеликих або розподілених новоутворень;
- аналіз крові на гормони;
- генетичне тестування – аналіз генів BRCA1 і BRCA2 (інколи також PALB2, PTEN, TP53) для виявлення спадкової схильності до раку молочної залози;
- лабораторні дослідження соскових виділень.
- сцинтиграфія (метод радіонуклідної візуалізації тканин грудей з використанням радіофармпрепаратів, призначений для виявлення місць патологічного порушення обміну речовин в м'яких тканинах);
- дуктографія (введення рентген-контрастної речовини у протоки молочних залоз через сосок за допомогою спеціального катетера з подальшим мамографічним знімком у двох проєкціях, що дає інформацію для уточнення виявлених змін і диференціальної діагностики, яка дозволяє розрізнити запальний процес та новоутворення у протоках);
- молекулярна діагностика (включає аналізи на зразок Oncotype DX, які оцінюють експресію генів у пухлині і допомагають передбачити ризик рецидиву та ефективність хіміотерапії).

Кожен із описаних методів має своє призначення і застосовується з різною частотою. Однак найбільш точним і достовірним вважається гістохімічний аналіз зразків тканин, отриманих шляхом біопсії. Часто ці зображення аналізуються лікарем, що збільшує час діагностики та узалежнює результати від досвіду і кваліфікації фахівця-діагноста. Тому розроблення автоматизованої системи для аналізу мікроскопічних зображень гістологічного матеріалу тканин молочної залози є актуальним завданням.

1.4 Сучасні технології автоматизованого аналізу і класифікації гістологічних зображень

На сьогодні рак є значною проблемою для здоров'я в усьому світі. Згідно з даними Міжнародного агентства з дослідження раку (IARC), яка входить у Всесвітню організацію охорони здоров'я (ВООЗ), у 2012 році було зареєстровано 8,2 мільйона смертей від раку, і очікується, що до 2030 року кількість нових випадків захворювання зросте до 27 мільйонів [9]. Серед усіх типів раку, рак молочної залози є другим за поширеністю серед жінок. Крім того, спостерігається висока смертність у порівнянні з іншими видами раку.

Останні досягнення в молекулярній біології та прогрес у дослідженнях раку молочної залози, а також відкриття нових молекулярних маркерів, підтверджують, що гістопатологічний аналіз залишається основним методом діагностики раку молочної залози [10]. Незважаючи на значні досягнення у діагностичних технологіях, остаточний діагноз раку молочної залози, включаючи класифікацію та постановку діагнозу, продовжує здійснюватися лікарями, які проводять візуальний огляд гістологічних зразків під мікроскопом. Сучасні досягнення в технологіях обробки зображень та машинного навчання дозволяють розробляти системи автоматичної діагностики (SAD), які можуть значно полегшити роботу діагностів, зробити

її більш продуктивною, об'єктивною та послідовною, виключити людський фактор.

Класифікація гістологічних зображень патологічних тканин на основі різних шаблонів, відповідних для неракових або ракових станів тканини, є основною метою в системах аналізу зображень для автоматичної діагностики раку. Головним викликом для таких систем є здатність до ефективної обробки складних гістопатологічних знімків.

При використанні різних моделей машинного навчання, таких як нейронні мережі та метод опорних векторів (SVM), точність результатів діагностики становить від 76 % до 94 %, що було показано на наборі даних з 92 зображень [11].

У роботі [12] було запропоновано каскадний підхід до класифікації. На першому рівні каскаду здійснюється відсікання легких випадків (які класифікуються однозначно), а решта спрямовуються на другий рівень, де використовується більш складна система класифікації. В зазначеному дослідженні використано базу даних ізраїльського технологічного інституту, яка включає 361 зображення, а точність результатів склала 97 %. У іншому дослідженні ті ж автори оцінювали ансамбль однокласових класифікаторів на тій же базі даних, досягаючи точності близько 92 %. Більшість цих робіт спрямовані на класифікацію раку молочної залози за допомогою цілісних зображень (WSI) [13–15]. Однак широке впровадження цих технологій та інших форм цифрової патології (digital pathology) зіштовхується з високою вартістю застосування та використання технології, недостатньою продуктивністю для великих масштабів досліджень на рівні публічних клінік, власними алгоритмічними проблемами технології, недостатньо врегульованими правовими питаннями, а також інерційністю й недовірою лікарів-діагностів [16].

За останній час більшість досліджень у гістопатологічному аналізі раку молочної залози проводилися на невеликих вибірках даних, які часто були

недоступні науковому співтовариству. Покращення цієї ситуації представлено набором даних з 7909 гістопатологічних зображень тканин, отриманих від 82 пацієнтів [17]. У цьому дослідженні було оцінено різноманітні текстурні дескриптори та різні класифікатори, результатом чого була отримана точність на рівні від 80 % до 85 %, залежно від кількості зображень.

Як видно з результатів, представлених у роботах [18, 19], текстурні дескриптори можуть надати гарне уявлення про підготовку класифікаторів. Проте деякі вчені вказують на те, що основною слабкістю сучасних методів машинного навчання є саме ця технічна складність. Все це означає, що алгоритми машинного навчання повинні бути менш залежними від функціональної інженерії, здатні витягати та організовувати дискримінаційну інформацію з даних, тобто повинні бути здатні до вивчення зображень. Ідея навчання системи не нова, але вона стала життєздатною альтернативою в останнє десятиліття завдяки популярності графічних обробників з їх високою обчислювальною потужністю та відносно невеликою вартістю, яка була досягнута завдяки їх масовій паралельній архітектурі. Також доцільно звернути увагу на використання попередньо навченої моделі, яка навчалась на іншому наборі даних.

Поміж різноманітних підходів, широко використовується 5-7 рівнева згорткова нейронна мережа (CNN), запропонована Лекуном [20], для досягнення передових результатів у розпізнаванні мікроскопічних зображень на основі аналізу їх текстури. Виявлено, що CNN здатні перевершувати в результатах традиційні текстурні дескриптори. Крім того, традиційний підхід до виявлення важливих ознак для класифікації на зображеннях патологічних тканин потребує значних зусиль та ефективного використання експертного знання, що часто призводить до високоінтегрованих рішень, специфічних для кожної задачі, що таким чином, невдало використовуються в інших контекстах. З урахуванням цього, в бакалаврській роботі будемо оцінювати глибокий підхід до вивчення проблеми гістопатологічної класифікації

зображень раку молочної залози і досліджувати можливість використання попередньо навчених моделей.

В роботі [21] було показано, що CNN досягає кращих результатів, ніж будь-які інші моделі машинного навчання, що навчались за допомогою текстурних сценаріїв на експериментальному наборі даних. При цьому найвища продуктивність досягається завдяки використанню комбінації різних CNN.

2 РОЗРОБЛЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ДІАГНОСТУВАННЯ РАКУ МОЛОЧНОЇ ЗАЛОЗИ

2.1 Матеріали дослідження

База даних BreakHis включає мікроскопічні зображення біопсії з доброякісних та злоякісних пухлин молочних залоз, які були отримані в клінічному дослідженні з січня по грудень 2014 року [22]. У цій базі міститься 9109 клінічно репрезентативних мікроскопічних зображень грудних пухлин з різними збільшеннями (збільшення у 40, 100, 200 та 400 разів), що були отримані від 82 пацієнток в лабораторії R&D в Бразилії, причому всі вимоги до етичності та анонімності особистих даних були дотримані.

Зображення були згенеровані з біопсійних слайдів тканин грудей, які забарвлені гематоксилином та еозином (HE). Ці зразки були зібрані хірургічною (відкритою) біопсією, були підготовлені для гістологічного дослідження та позначені й класифіковані лікарями лабораторії R&D. Процедура підготовки зразків відповідала стандартним вимогам, які широко використовуються у клінічній практиці. Основна мета полягала у збереженні оригінальної структури тканини та мікрокартини патології, наче при безпосередньому спостереженні через світловий мікроскоп.

Зображення були одержані за допомогою мікроскопічної системи Olympus BX-50 з окуляром 10x, об'єктивами різного збільшення та цифровою камерою Samsung SCC-131AN. Зображення збережені в трьохканальному кольоровому просторі TrueColor RGB (Red-Green-Blue) при збільшенні у 40, 100, 200 і 400 разів (глибина кольорів 24 розряди, кольоровий канал 8 біт).. Виділений прямокутник (доданий вручну тільки для ілюстративних цілей) представляє область інтересу, обрану патологом, яка буде деталізована на наступному збільшенні. На сьогоднішній день база даних містить 9109 зображень, які класифіковані як доброякісні та злоякісні пухлини. Розподіл

зображень представлено в табл. 2. У додатку А наведено зразки зображень з бази даних BreakHis з різним збільшенням.

Таблиця 2 – Розподіл зображень у базі даних BreakHis

Збільшення мікноспока	Кількість зображень		
	доброякісних пухлин	злаякісних пухлин	усього
40 разів	755	1505	2260
100 разів	754	1558	2312
200 разів	753	1544	2297
400 разів	748	1492	2240
Разом	3010	6099	9109
Пацієнти	24	58	82

2.2 Модель NEuro Fuzzy CLASSifier

У зв'язку з тим, що зображення часто є набором цифрових даних, що складаються з великої кількості елементів (пікселів), поєднання яких складно формалізувати чи описати, в автоматизованих діагностичних системах значного поширення набули нечіткі моделі обробки [23].

Нечіткі моделі в класифікації зображень використовують нечітку логіку для обробки неточностей і невизначеностей, які можуть бути присутніми в зображеннях. Ці моделі зокрема корисні в ситуаціях, де межі між різними класами є розмитими або їх важко чітко визначити.

Основними компонентами нечітких моделей у класифікації зображень є:

а) нечіткі множини, які використовують для представлення розмитих класів. Кожен піксель або сегмент зображення може належати до декількох класів одночасно з різним ступенем належності, що задається значенням від 0 до 1;

б) лінгвістичні змінні, які визначають параметри, що описують властивості зображень (наприклад, колір, текстура, форма) у нецифровій формі. Кожна змінна може приймати лінгвістичні значення (наприклад, «темний», «середній», «світлий»), які інтерпретуються через нечіткі множини;

в) нечіткі правила, які складають основу нечітких моделей і мають вигляд «якщо-то» (IF-THEN). Наприклад, «якщо область зображення є світлою і має гладку текстуру, то ймовірно, що це клас А»;

г) механізм нечітких висновків, який застосовує нечіткі правила до вхідних даних (зображень) для отримання ступеня належності кожного пікселя або сегменту до різних класів. Висновок генерується шляхом агрегування результатів застосування правил;

д) дефазифікація – процес перетворення нечітких висновків у конкретні класи. Вихідна нечітка множина перетворюється в чітке значення, яке визначає остаточний клас зображення або його сегменту.

Перевагами нечітких моделей у класифікації гістологічних зображень є: гнучкість (здатність обробляти невизначеності та неоднозначності, що часто зустрічаються на мікроскопічних зображеннях), інтерпретованість (нечіткі правила часто зрозумілі для людей, що спрощує пояснення або представлення рішень моделі;), адаптивність (легко адаптуються до різних типів зображень та задач шляхом налаштування правил і функцій належності).

Слід відзначити також і обмеження, пов'язані з використанням нечітких моделей: параметризація (вибір і налаштування нечітких правил та функцій належності можуть бути складними і вимагати експертного знання або великого набору даних); обчислювальна складність, яка може бути вищою (але не завжди) порівняно з іншими методами, особливо при великій кількості правил і змінних.

Використання в медичній діагностиці нечітких моделей є досить поширеним, особливо для класифікації медичних зображень, таких як рентгенівські знімки або гістопатологічні зображення [24], де межі між

різними тканинами можуть бути розмитими. Нечіткі моделі є потужним інструментом для класифікації зображень, особливо в умовах, де традиційні методи можуть бути недостатньо ефективними через присутність невизначеностей або неоднозначностей.

Розглянемо нечітку модель NEFCLASS.

Метою моделі NEuro Fuzzy CLASSifier (NEFCLASS) є отримання нечітких правил з множини даних, які можна розподілити на різні класи. Нечіткі правила описують дані в формі:

якщо $x \in X$. має функцію належності $\mu_1(x), \dots, \mu_n(x)$,

то зразок належить класу i , де x - нечіткі множини. Завданням моделі NEFCLASS є визначення належності вхідного зразка певного до класу. Це виключає подвійну належність об'єкта, тобто перетин двох різних множин є порожнім об'єктом. Розглянемо архітектуру моделі NEFCLASS більш детально [25].

База правил представляє собою апроксимацію невідомої функції і описує класифікаційну задачу, де вказується, до якого класу належить певний об'єкт. Нечіткі множини та лінгвістичні правила визначають результат системи NEFCLASS і походять з множини вибірок шляхом навчання. При цьому обов'язково повинна дотримуватися умова, що для кожного лінгвістичного значення може існувати тільки одне подання нечіткої множини.

Система NEFCLASS має тришарову нейромережеву архітектуру. Перший шар містить вхідні нейрони, які представляють вхідні зразки, при цьому активація нейронів зазвичай не змінює вхідного значення. Прихований шар містить нечіткі правила, а третій шар складається з вихідних нейронів кожного класу. Активація нейронів правил та вихідного шару для зразка g обчислюється як

$$a_R^{(p)} = \min_{x \in U_1} \{W(x, R)(a_x^{(p)})\}$$

$$a_R^{(p)} = \sum_{R \in U_2} W(c, R)(a_R^{(p)})$$

де $W(x, R)$ – нечіткий ваговий коефіцієнт з'єднання вхідного нейрона x з нейроном правила R ;

$W(R, c)$ - нечіткий ваговий коефіцієнт сполучення нейрона правила R з нейроном вихідного шару c . Замість застосування операцій пошуку максимуму і мінімуму може бути використано інші функції – відповідно t -норми і t -конорми.

База правил є апроксимацією невідомої функції та описує класифікаційну задачу, де кожний x належить певному класу. Нечіткі множини та лінгвістичні правила представляють цю апроксимацію та визначають результат системи NEFCLASS (рис. 4). Вони створюються на основі великої кількості вибірок шляхом навчання. Обов'язковим є правило, що для кожного лінгвістичного значення може існувати лише одне подання нечіткої множини.

Система передбачає використання методів оптимізації, тобто пошуку локальних і глобальних максимумів та мінімумів функції в заданому фазовому просторі [26]. Розглянемо методи, які можуть бути використані для задач класифікації гістологічних зображень.

2.3 Метод найкоротшого спуску

Всі методи пошуку рішення задачі глобальної мінімізації відмінностей-відрізняються або вибором напрямку руху, або способом спуску уздовж обраного напрямку. Це дозволяє визначити загальну схему методів спуску.

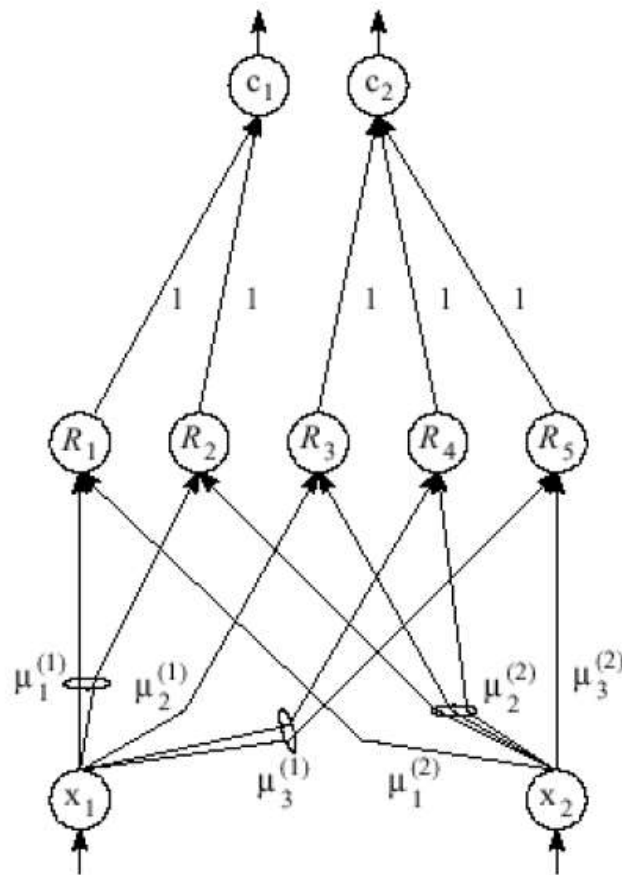


Рисунок 4 – Архітектура системи NEFCLASS

Загальною ознакою є те, що метод вирішує задачу мінімізації функції (x) на всьому заданому просторі E_n . Методи спуску формуються в такій процедурі побудови послідовності x_k . Для відправної точки початкового наближення обирається будь-яка точка x_0 , що належить E . Послідовні наближення $x_1, x_2, \dots, x_n$ знаходяться за такою схемою:

- в точці x_k обирають напрям спуску, позначимо його як S_k ;
- знаходять точку $(k + 1)$ за формулою $x_{k+1} = x_k - h_k S_k$.

Напрямок S_k обирають у такий спосіб, щоб забезпечити виконання нерівності $f(x_{k+1}) < f(x_k)$ принаймні для невеликих значень величини кроку h_k . При цьому вибір способу і напрямку спуску часто є випадковим і залежить від конкретної задачі.

Число h_k описує відстань від точки x_k до точки x_{k+1} . Це число має назву довжина кроку чи просто крок. Основна вимога при виборі значення h_k – це виконання нерівності $R(x_{k+1}) < R(x_k)$.

Величина кроку значно впливає на результативність методу. Більшої ефективності досягає той варіант методу, при якому крок за кожною змінною визначається напрямними косинусами градієнта:

$$x_{k+1} = x_k - h_k \cos \phi_i,$$

$$\cos \phi_i = \frac{(dR / dx_j)}{|grad R(x)|}$$

В такому випадку величина довжини кроку не залежить від значення модуля градієнта, і нею простіше керувати за допомогою зміни h . В районі екстремуму може виникати так зване рискання, рух туди-сюди, тому застосовують різні алгоритми для корекції h в процесі пошуку. Найбільш поширені алгоритми корекції кроку:

- 1) $h_i = const = h$ (без корекції);
- 2) $h_i = h_{i-1}/2$, якщо $R(x_i) = R(x_{i-1})$; $h_i = h_{i-1}$ якщо $R(x_i) > R(x_{i-1})$;
- 3) $h_i = h_{i-1}$, якщо $a_1 \leq \alpha \leq a_2$; $h_i = 2h_{i-1}$, якщо $\alpha > a_2$; $h_i = h_{i-1}/3$, якщо $\alpha < a_1$, де α – кут між векторами градієнту на попередньому і поточному кроці; a_1 і a_2 – задані порогові значення, які обираються користувачем (типові значення $a_1 = \pi / 6$, $a_2 = \pi / 3$).

На значній відстані від оптимуму напрям градієнта змінюється незначно, тому величину кроку можна збільшити (другий вираз), а в околі оптимуму напрям може різко змінюватись (кут між градієнтами $R(x)$ великий), тому значення h зменшується (третій вираз).

Розглянемо на прикладі метод градієнтного спуску.

Візьмемо функцію f , вважаючи, що вона залежна від трьох змінних x, y, z . Обчислимо її частинні похідні $df / dx, df / dy, df / dz$ і знайдемо з їх допомогою вектор, що носить назву градієнту функції:

$$\text{grad } f(x, y, z) = \partial f(x, y, z) / \partial x \cdot \mathbf{i} + \partial f(x, y, z) / \partial y \cdot \mathbf{j} + \partial f(x, y, z) / \partial z \cdot \mathbf{k},$$

де $\mathbf{i}, \mathbf{j}, \mathbf{k}$ – одиничні елементарні вектори, взаємоперпендикулярні та паралельні координатним осям. Частинні похідні характеризують швидкість зміни функції f уздовж кожної осі окремо. Утворений на їх основі вектор градієнта надає загальне розуміння про поведінку функції навколо точки (x, y, z) . Напрямок отриманого вектора є напрямом найбільш швидкого зростання функції у заданій точці. Протилежний йому напрям, що часто називають антиградієнтним, є не що інше, як напрям найбільш швидкого спадання функції. При цьому модуль градієнта

$$|\text{grad } (x, y, z)| = [\partial f / \partial x (x, y, z))^2 + (\partial f / \partial y (x, y, z))^2 + (\partial f / \partial z (x, y, z))^2]^{1/2}$$

визначає швидкість зростання або спадання функції в напрямку градієнта та антиградієнта. Для всіх інших напрямків швидкість зміни функції в точці (x, y, z) менша за модуль градієнта. При переході від однієї точки до іншої змінюються як напрямок градієнта, так і його модуль. Поняття градієнта природним чином переноситься на функції будь-якої кількості змінних.

Перейдемо до опису методу градієнтного спуску. Його основна ідея полягає в тому, щоб рухатися до мінімуму в напрямку найшвидшого зменшення функції, який визначається антиградієнтом. Ця ідея реалізується таким чином.

Обираємо початкову точку будь-яким способом, обчислюємо в ній градієнт даної функції та робимо невеликий крок у зворотному, антиградієнтному напрямку. Як результат, ми потрапимо в точку, в якій

значення функції буде меншим за початкове. У новій точці повторюємо процедуру: знову обчислюємо градієнт функції та робимо крок у зворотному напрямку. Продовжуючи цей процес, ми рухаємося у напрямку зменшення функції. Спеціальний вибір напрямку руху на кожному кроці дозволяє сподіватися, що в цьому випадку наближення до мінімального значення функції буде швидшим, ніж у методі покоординатного спуску.

Метод градієнтного спуску вимагає обчислення градієнта цільової функції на кожному кроці. Якщо функція задана аналітично, то це, як правило, не проблема: для частинних похідних, що визначають градієнт, можна знайти явні формули. В іншому випадку частинні похідні у потрібних точках доводиться обчислювати наближено.

Для оцінки частинних похідних застосовують різницеві методи:

а) алгоритм з центральної пробою:

$$\frac{dR}{dx_i} \approx \frac{f(x_1, \dots, x_i + g_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{g_i};$$

б) алгоритм з парними пробами:

$$\frac{dR}{dx_i} \approx \frac{f(x_1, \dots, x_i + g_i, \dots, x_n) - f(x_1, \dots, x_i - g_i, \dots, x_n)}{2g_i},$$

де g_i – пробний крок уздовж i -ї змінної, значення якого досить мале для різницевого оцінювання значення похідної.

Слід зазначити, що при наведених розрахунках можна g_i взяти занадто малим, тоді як значення функції необхідно обрахувати з досить високою точністю, інакше при обчисленні різниць буде виникати значна помилка:

$$\begin{aligned} \Delta f &= f(x_1, \dots, x_i + g_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n), \\ \Delta f &= f(x_1, \dots, x_i + g_i, \dots, x_n) - f(x_1, \dots, x_i - g_i, \dots, x_n). \end{aligned}$$

Перший алгоритм потребує менших затрат у порівнянні з другим (зазвичай витрати визначаються за кількістю обчислень критерію оптимальності), але дозволяє отримати не таке точне рішення, як другий. Ця похибка залежить від величини пробного кроку.

На рисунку 5 наведено лінії-горизонталі рівня функції двох змінних $u = f(x, y)$ на фазовій площині та наведена траєкторія пошуку її екстремуму за допомогою методу градієнтного спуску.

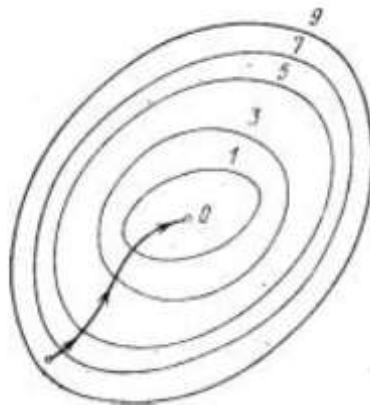


Рисунок 5 – Ілюстрація методу градієнтного спуску

Далі розглянемо метод покоординатного спуску (або методу Гауса-Зейделя). Вважаємо, що потрібно знайти мінімальне значення цільової функції

$$u = f(M) = f(x_1, x_2, \dots, x_n).$$

У формулі літерою M позначена точка у n -мірному просторі з координатами $x_1, \dots, x_i, \dots, x_n$: $M = (x_1, x_2, \dots, x_n)$. Вибираємо будь-яку початкову точку $M = (x_1, x_2, \dots, x_{n0})$ і розглядаємо функцію f при фіксованих значеннях всіх змінних, крім першої: $f(x, \text{const}, \text{const}, \dots, \text{const})$. Тоді вона перетворюється у функцію однієї змінної x . Варіюючи значення цієї змінної, здійснюємо рух від початкової точки в бік зменшення значення функції, доки не досягнемо її

мінімуму, після якого функція починає зростати. Точку зі знайденими координатами позначимо літерою M_1 , при цьому $f(M_0) > f(M_1)$.

Надалі по чергово фіксуються інші змінні, при цьому на кожному кроці функція знову стає залежною лише від однієї з них. Таким чином здійснюємо пошук за кожною зі змінних, на кожному етапі знаходячи мінімальне значення функції, що відповідає умові $f(M_n) > f(M_{n+1})$.

Дійшовши до останньої змінної x_n , знову повертаємось до першої змінної x_1 і продовжуємо друге коло процесу. Така процедура цілком відповідає назві методу. За її допомогою можна побудувати послідовність точок M_n , якій буде відповідати монотонна спадаюча послідовність значень функції $f(M_n)$. Її можна зупинити на будь-якому етапі k , тоді наближено можна прийняти значення функції $f(M_k)$ за її мінімальне значення в області, що розглядається.

Зазначимо, що цей метод зводить задачу пошуку найменшого значення функції багатьох змінних до багаторазового розв'язання одновірних задач оптимізації. Якщо цільова функція $f(x_1, x_2, \dots, x_n)$ задана явною формулою і є диференційованою в області пошуку, ми можемо обчислити її частинні похідні і використовувати їх для визначення напрямку зменшення функції по кожній змінній та пошуку відповідних одновірних мінімумів. У випадку, коли явної формули для цільової функції немає, одновірні задачі слід розв'язувати за допомогою одновірних методів.

На рис. 6 зображені лінії-горизонталі рівня деякої функції двох змінних $u = f(x, y)$. Уздовж цих ліній функція зберігає постійні значення, які дорівнюють відповідно 1, 3, 5, 7, 9. На схемі показана траєкторія пошуку її найменшого значення, яке досягається в точці O , за допомогою методу по координатного спуску. При цьому слід чітко розуміти, що рисунок є лише схематичною ілюстрацією методу.

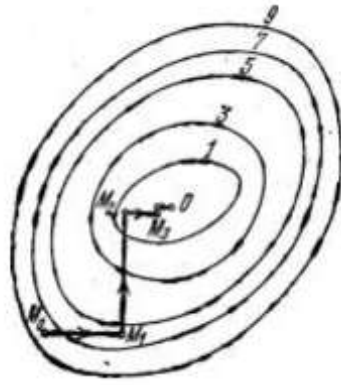


Рисунок 6 – Схематичне представлення методу покоординатного спуску

Покрокова реалізація узагальненої схеми у двовірному випадку (функції двох змінних) визначає траєкторію наближення до точки мінімуму методом покоординатного спуску, що складається з ланок кривої, які поєднують точки знайдених локальних мінімумів. При $k = 0$, виходячи з початкової точки, яку приймають за початок координат $x(0) = (x_1(0), x_2(0))$, при фіксованому значенні $x_2(0)$ знаходять точку мінімуму функції однієї змінної $f(x_1, x_2(0))$. Після цього знаходять точку мінімуму $x(1)$ функції $f(x_1(0), x_2)$ уздовж другої координати. Далі вибирають мінімальне з двох отриманих значень і приймають його за локальний мінімум. Далі виконують наступний крок розрахунків при $k = 1$. Тобто вихідною точкою для другого кроку розрахунку стає $x(1)$.

Нарешті, розглянемо метод найшвидшого спуску, який полягає в наступному. Спочатку в початковій точці визначається антиградієнт функції, що мінімізується. Замість того, щоб робити фіксований крок у напрямку антиградієнта, рухаються в цьому напрямку доти, поки цільова функція зменшується, досягаючи мінімуму в певній точці. У цій новій точці знову обчислюють антиградієнт і визначають нову точку мінімуму цільової функції, і цей процес повторюється на кожному кроці. Таким чином, спуск стає більш цілеспрямованим, здійснюється більшими кроками, а градієнт функції обчислюється у меншій кількості точок, що підвищує швидкодію алгоритму.

2.4 Метод диференціальної еволюції

Диференціальна еволюція - це метод багатовимірної математичної оптимізації, який належить до класу стохастичних алгоритмів оптимізації (тобто використовує випадкові числа) і запозичує деякі ідеї генетичних алгоритмів. Це прямий метод оптимізації, тобто він вимагає лише можливості обчислювати значення цільової функції, але не її похідних. Метод диференціальної еволюції призначений для знаходження глобального мінімуму (або максимуму) недиференційованих, нелінійних, мультимодальних функцій багатьох змінних. Цей метод простий у реалізації та використанні (має небагато керованих параметрів, які потребують налаштування) і легко розпаралелюється.

Метод диференціальної еволюції (differential evolution) був розроблений Р. Сторном і К. Прайсом, вперше опублікований ними у 1995 році [27] та розвинений у їхніх наступних роботах.

Алгоритм у своєму базовому вигляді можна описати наступним чином: спочатку генерується деяка кількість векторів, що називаються поколінням. Під векторами розуміються точки d n -вимірному просторі, в якому визначена цільова функція $f(x)$, яку необхідно мінімізувати. На кожній ітерації алгоритм генерує нове покоління векторів, випадковим чином комбінуючи вектори з попереднього покоління. Кількість векторів у кожному поколінні є сталою і визначається як один з параметрів методу. Нове покоління векторів генерується наступним чином.

Для кожного вектора x_i з попереднього покоління обираються три різних випадкових вектора v_1, v_2, v_3 з-поміж векторів старого покоління, за винятком самого x_i , і формується т. зв. мутантний вектор за допомогою формули:

$$v = v_1 + F \cdot (v_2 - v_3) \quad v = v_1 + F \cdot (v_2 - v_3)$$

де F – один з параметрів методу, деяке додатнє дійсне число в інтервалі $[0, 2]$. Над мутантним вектором v здійснюється операція «перехрещення» (crossover), що полягає в тому, що деякі координати цього вектора замінюються відповідними координатами початкового вектора x_i (причому кожна координата може замінитись з певною ймовірністю, що теж є параметром методу еволюції). Отриманий після перехрещування вектор називають пробним вектором. У випадку, якщо він буде кращим за початковий вектор x_i (інакше, значення цільової функції при цьому зменшилось), то в новому поколінні вектор x_i буде замінено на пробний вектор, інакше – залишається початковий вектор x_i . Метод диференціальної еволюції дозволяє значно покращити процеси ранжирування і систематизації.

Базовий варіант алгоритму диференціальної еволюції полягає в тому, що він використовує популяцію кандидатів на рішення задачі оптимізації. Ці кандидати переміщуються в просторі пошуку, застосовуючи прості математичні формули для об'єднання позицій існуючих агентів з популяції. Якщо нова позиція агента покращує результат, вона приймається і стає частиною популяції, інакше вона відхиляється. Процес повторюється до тих пір, поки не буде знайдене задовільне рішення.

2.5 Алгоритм стрибка в басейн

Алгоритм стрибка в басейн («basin-hopping») є одним із методів глобальної оптимізації для пошуку мінімуму або максимуму нелінійних функцій. Він був розроблений Д. Вейлсом і Д. Доєм у 1997 році [28]. Основна його ідея полягає в тому, щоб уникати застрягання в локальних мінімумах шляхом комбінування методу змінення координат (Metropolis) з локальним методом оптимізації, як наприклад метод Нелдера-Міда (Nelder-Mead). Алгоритм працює таким чином.

Алгоритм *basinhopping* розпочинається з випадкової початкової точки в просторі параметрів. На кожній ітерації алгоритм використовує локальний метод оптимізації (наприклад, метод Нелдера-Міда) для знаходження локального мінімуму від поточної точки.

Після знаходження локального мінімуму, виконується випадковий зсув або зміна координат цієї точки. Відповідно до алгоритму *Metropolis*, крок приймається з ймовірністю, що залежить від різниці у значенні цільової функції перед кроком і після кроку. Якщо значення функції покращується або залишається незмінним, то крок приймається. В іншому випадку, з ймовірністю, яка зменшується з ростом різниці значень функції, крок також може бути прийнятий.

Процес продовжується до досягнення задовільного критерію зупинки, яким може бути, наприклад, певна кількість ітерацій або зниження значення цільової функції нижче певного порогового значення.

Алгоритм *basinhopping* ефективний для пошуку глобальних мінімумів в просторі параметрів, де можуть бути багато локальних мінімумів. Він комбінує в собі переваги методу змінення координат і методу локальної оптимізації, що дозволяє йому ефективно досліджувати фазові простори з великою кількістю локальних екстремумів.

Стохастична складова алгоритму полягає в випадкових зміщеннях або збуреннях в певній точці – місцевому мінімумі. Ці зміщення повинні бути достатньо великими, щоб уникнути поточного місцевого мінімуму, але не настільки великими, щоб стрибок став «абсолютно випадковим». Такий підхід був мотивований фізичною хімією, звідки походить цей алгоритм, адже стабільні конфігурації, тобто локальні мінімуми, розташовані в тому ж регіоні пошукового простору. Таким чином, алгоритм не повинен досліджувати весь простір пошуку, що є однією з основних проблем глобальної оптимізації взагалі. Однак якщо мінімум «потрапляє» в певну область простору, метод «стрибка в басейн» може бути дуже потужним.

Цей метод глобальної мінімізації виявився надзвичайно ефективним для різних фізичних і хімічних задач, особливо коли функція має багато мінімумів, розташованих віддалено один від одного.

Для стохастичної глобальної оптимізації неможливо гарантувати, що справді був знайдений глобальний мінімум. Замість цього, алгоритм може бути запущений з різних випадкових вихідних точок, щоб забезпечити знаходження глобального екстремуму, який збігається з глобальним мінімумом в кожному випадку. Тому за замовчуванням метод *basinhopping* просто запускається на задану кількість ітерацій і повертає найкращий результат. Користувачу залишається переконатися, що цей результат фактично є глобальним мінімумом.

2.6 Згорткові нейронні мережі

Згорткові, або конволюційні, нейронні мережі є спеціальною архітектурою штучних нейронних мереж, яку запропонував Ян Лекун в 1988 році з метою ефективного розпізнавання зображень. Ці мережі використовують особливості організації зорової кори мозку, де виявлено так звані прості клітини, які реагують на прямі лінії під різними кутами, а також складні клітини, активація яких пов'язана з включенням певного набору простих клітин. Таким чином, ідея конволюційних нейронних мереж полягає в чергуванні згорткових шарів і підвибіркових шарів (або шарів пулінгу). Структура мережі є односпрямованою (без зворотних зв'язків) і багатошаровою. Для навчання використовуються стандартні методи, найчастіше метод зворотного поширення помилки. Функція активації нейронів (передавальна функція) може бути будь-якою, в залежності від вибору дослідника. Назва архітектури мережі походить від операції згортки, яка полягає в тому, що кожен фрагмент зображення множиться на матрицю (ядро)

згортки елементно, а результат підсумовується і записується в аналогічну позицію вихідного зображення.

Згорткова нейронна мережа дозволяє здійснити перехід від конкретних особливостей зображення до більш абстрактних деталей, і навіть до високорівневих понять. Мережа автоматично налаштовується і формує необхідну ієрархію абстрактних ознак (послідовності карт ознак), фільтруючи незначні деталі і виділяючи суттєві.

Ця інтерпретація має скоріше метафоричний або ілюстративний характер. Фактично «ознаки», які формуються складною мережею, є важкими для інтерпретації і малозрозумілими. В практичних системах не рекомендується намагатися розуміти значення цих ознак або змінювати їх «на льоту». Замість цього рекомендується вдосконалити саму структуру і архітектуру мережі, щоб отримати кращі результати.

Ігнорування деяких суттєвих явищ системою може свідчити про те, що або було надано недостатньо даних для навчання, або структура мережі має недоліки і не може ефективно створити потрібні ознаки для цих явищ.

У звичайному перцептроні, який є повнозв'язною нейронною мережею, кожен нейрон зв'язаний з усіма нейронами попереднього шару, причому кожен зв'язок має свій власний ваговий коефіцієнт. У згортковій нейронній мережі для операції згортки використовується лише обмежена матриця ваг невеликого розміру, яку «переміщують» по всьому оброблюваному шару (на початку – безпосередньо по вхідному зображенню), формуючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією. Це означає, що для різних нейронів вихідного шару використовується одна і та ж матриця ваг, яку називають ядром згортки. Її інтерпретують як графічне кодування певної ознаки, наприклад, наявність похилої лінії під певним кутом.

Отже, наступний шар, утворений в результаті операції згортки такою матрицею ваг, показує наявність цієї ознаки в оброблюваному шарі та її координати, формуючи так звану карту ознак (feature map). У конволюційній

нейронній мережі набір ваг не один, а цілий спектр, вони кодують різні елементи зображення (наприклад, лінії та дуги під різними кутами). Ці ядра згортки не задаються дослідником наперед, а формуються самостійно мережею шляхом навчання класичним методом зворотного поширення помилки. Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальною (багато незалежних карт ознак на одному шарі).

Варто зазначити, що при переміщенні шару матрицею ваг її зсувають не на повний крок (розмір цієї матриці), а на невелику відстань. Наприклад, при розмірності матриці ваг 5×5 її зсувають на один або два нейрони (пікселі) замість п'яти, щоб не пропустити ознаку, яку шукають.

Операція субдискретизації (або pooling) в згорткових нейронних мережах здійснює зменшення розмірності формованих карт ознак. Для даної архітектури мережі вважають, що інформація про наявність шуканої ознаки є важливішою за точне знання її координат, тому з кількох сусідніх нейронів карт ознак обирається максимальний і приймається за один нейрон ущільненої карти ознак меншої розмірності. Ця операція дозволяє прискорити подальші обчислення і робить мережу більш інваріантною до масштабу вхідного зображення.

Типова структура згорткової нейронної мережі включає велику кількість шарів. На рисунку 7 показано типову архітектуру згорткової нейронної мережі типу VGG-16. Після початкового шару (вхідного зображення) сигнал проходить чергу згорткових шарів, в яких чергуються операції згортки і субдискретизації (або пулінгу). Це чергування шарів дозволяє формувати «карти ознак», де на кожному наступному шарі карта зменшується в розмірі, але збільшується кількість каналів і виявлених ознак. На практиці це означає здатність виявляти складні ієрархії ознак.

Зазвичай після проходження декількох шарів карта ознак перетворюється у вектор або навіть скаляр, але таких карт ознак може стати

сотні. На виході конволюційних шарів мережі додатково встановлюють декілька шарів повнонейронної мережі (типу перцептрона), на вхід яких подаються кінцеві карти ознак.

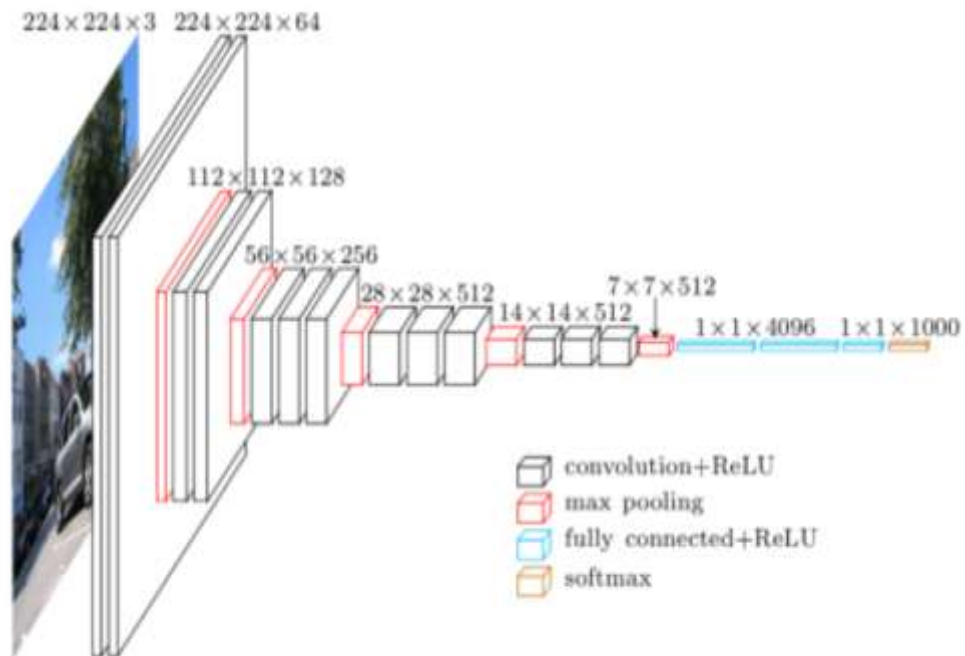


Рисунок 7 – Архітектура згорткової нейронної мережі на прикладі макроструктури CNN VGG-16

Шар згортки (англ. convolutional layer) є основним блоком конволюційної нейронної мережі. Він включає в себе для кожного каналу свій фільтр, ядро згортки якого обробляє попередній шар за фрагментами зображення, підсумовуючи результати матричного представлення для кожного фрагмента. Вагові коефіцієнти ядра згортки, які є невеликою матрицею, встановлюються в процесі навчання.

Особливістю згорткового шару є порівняно невелика кількість параметрів, які встановлюються при навчанні. Наприклад, якщо вхідне зображення має розмірність 100×100 пікселів по трьох каналах (що означає 30,000 вхідних нейронів), а згортковий шар використовує фільтри з ядром 3×3 пікселі з виходом на 6 каналів, тоді в процесі навчання визначається тільки 9 ваг ядра. Однак за всіма комбінаціями каналів, отже, $9 \times 3 \times 6 = 162$,

в такому випадку даний шар вимагає знаходження тільки 162 параметрів, що є значно меншою кількістю, ніж у повнозв'язаній нейронній мережі.

Шар ректифікованих лінійних одиниць (ReLU) використовує елементарну функцію активації для введення нелінійності в мережу.

Шар пулінгу або субдискретизації забезпечує нелінійне ущільнення карти ознак, де група пікселів (зазвичай розміром 2×2) зменшується до одного пікселя шляхом нелінійного перетворення. Найчастіше використовується функція максимуму. Перетворення впливає на непересічні прямокутники або квадрати, де вибирається піксель з максимальним значенням. Операція пулінгу значно зменшує просторовий обсяг зображення. Пулінг інтерпретується так: якщо після попередньої операції згортки вже виявлені деякі ознаки, то для подальшої обробки таке докладне зображення вже не потрібне і воно ущільнюється до меншого за обсягом. Крім того, відфільтровані непотрібні деталі допомагають уникнути перенавчання. Шар пулінгу, як правило, розташовують після шару згортки перед наступним шаром згортки.

З метою ще більш інтенсивного зменшення розміру одержуваних зображень останнім часом все частіше використовують розширення ідеї використання менших фільтрів або повну відмову від шарів пулінгу.

В архітектурі згорткових нейронних мереж після кількох проходжень згортки зображення і ущільнення за допомогою пулінгу система перебудовується від конкретної сітки пікселів з високою роздільною здатністю до більш абстрактних карт ознак. Зазвичай на кожному наступному шарі збільшується кількість каналів і зменшується розмірність зображення в кожному каналі. Залишається великий набір каналів, що зберігають невелику кількість даних, інтерпретованих як абстрактні поняття, виявлені з вихідного зображення.

Ці дані об'єднуються і передаються на звичайну повнозв'язну нейронну мережу, яка також може складатися з декількох шарів. Повнозв'язні шари вже

втрачають просторову структуру пікселів і мають порівняно невелику розмірність (відносно кількості пікселів вхідного зображення).

В подальшій роботі будемо використовувати цей тип ШНН. На рис. 8 показано структуру згорткової мережі у вигляді лінгвістичного опису.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000
dense_1 (Dense)	(None, 16)	16016
Total params: 138,373,560		
Trainable params: 138,373,560		
Non-trainable params: 0		

Рисунок 8 – Опис структури мережі CNN VGG-16

3 ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ АНАЛІЗУ ГІСТОЛОГІЧНИХ ЗОБРАЖЕНЬ

3.1 Методологія дослідження

В роботі було використано попередньо навчену модель на основі згорткових нейронних мереж, які були описані в попередньому розділі, зокрема модель VGG-16. Було використано метод передачі навчання, який включає два основних сценарії.

Перший сценарій – виділення ознак. У цьому випадку останній повнозв'язний шар моделі видаляється, і решта згорткових шарів використовуються для виділення ознак для нового набору даних.

Другий сценарій – точне налаштування. При цьому новий набір даних використовується для точного налаштування ваг попередньо підготовленої моделі CNN. Є можливість точно налаштувати всі шари або ж тільки певні шари моделі.

У нашому експерименті мережа VGG-16 була обрана для виділення ознак. Отримані ознаки подавалися на вхід для навчання і тестування класифікаторів, зокрема на мережу NEFCLASS, яка була описана у попередньому розділі. В якості алгоритмів навчання були використані три алгоритми: алгоритм «стрибка в басейн» (див. підрозділ 2.5), алгоритм диференціальної еволюції (див. підрозділ 2.4) і метод градієнтного спуску (див. підрозділ 2.3).

3.2 Експериментальна частина і обговорення результатів

Для досліджень була використана база даних гістологічних зображень тканин молочних залоз BreakHis. До зображень затосовували попередньо навчену модель для виділення ключових ознак. При цьому навчання моделі

також відбувалось на вказаних зображеннях. З цією метою уся сукупність зображень була розбита у співвідношенні 3 до 1 на відповідно навчальну вибірку (75 % зображень) і тестову вибірку (25 % зображень). В таблиці 3 подано отримані результати класифікації за різними параметрами. Ми змінювали кількість лінгвістичних змінних та правил для того, щоб визначити оптимальне їх значення для досягнення максимальної точності.

Таблиця 3 – Результати досліджень застосування мережі NEFCLASS на тестовій вибірці

Кількість розбиття / кількість правил	Точність класифікації гістологічних зображень зі збільшенням			
	у 40 разів	у 100 разів	у 200 разів	у 400 разів
2 / 2	73%	74%	74.2%	73.5%
4 / 2	75.3%	74.8%	75.7%	75.4%
6 / 2	78.2%	79%	78.4%	78%
8 / 2	76%	75.4%	76.5%	75.8%
2 / 4	75%	74%	73.8%	73%
4 / 4	78.3%	76.3%	75.7%	75.4%
6 / 4	82%	83%	82.4%	83.2%
8 / 4	82.2%	81.5%	81.5%	83.8%
2 / 6	75.4%	73.8%	74.4%	73.2%
4 / 6	90%	91%	90.5%	90%
6 / 6	89%	89.7%	90.2%	89.5%
8 / 6	90.3%	90.5%	92%	91.2%
4 / 8	89.3%	89.8%	89.7%	89.3%
6 / 8	89.2%	88%	89.4%	88.4%
8 / 8	88%	87.2%	87.2%	87%

З наведеної таблиці видно, що при класифікації зі зростанням кількості початкового розбиття та збільшення кількості правил підвищується й точність діагностики. Однак можна виділити два значення кількості розбиттів та визначальних правил (4/6 та 8/6), для яких спостерігається максимальне значення точності діагностування. Однак зі збільшенням цих чисел зростає й складність та тривалість навчання системи. У зв'язку з цим обираємо

мінімальні значення кількості розбиттів та правил, які дають досягати оптимальної точності діагностики на рівні 90–92 %.

Порівняємо отримані дані з результатами роботи інших класифікаторів, отримані в попередніх роботах (табл. 4),

Таблиця 4 – Порівняння точності діагностування при застосуванні різних класифікаторів

Класифікатор	Точність класифікації гістологічних зображень зі збільшенням			
	у 40 разів	у 100 разів	у 200 разів	у 400 разів
NEFCLASS	90 %	91 %	90.5 %	90 %
Linear svm	89 %	89 %	88 %	88 %
Polynomial svm	88 %	90 %	89 %	85 %
Random forest	89.2 %	88 %	87.7 %	80 %

Як бачимо, модель Nefclass виявляється трохи ефективнішою, ніж інші алгоритми.

На вхід для навчальних алгоритмів подавалися витягнуті ознаки, що складаються з 4096 вхідних значень (навчальна вибірка). Ознаки були витягнуті за допомогою попередньо навченої моделі VGG-16 з вилученням останнього вирішального шару.

Для навчання мережі Nefclass було використано три алгоритми навчання, а саме диференціальна еволюція, алгоритм оптимізації «стрибок в басейн» та метод найшвидшого спуску. Під час навчання було досягнуто однакових результатів для алгоритмів «стрибок в басейн» і для методу найшвидшого спуску, що може свідчити про правильно виконану й оптимізовану класифікацію. Також важливо зазначити, що алгоритм «стрибок в басейн» також впорався, що свідчить про те, що мінімум функції було знайдено правильно, а це, в свою чергу, підтверджує, що мінімум є досить явно локалізований.

Оскільки після застосування екстрактора кількість вхідних параметрів була значною (4096), було доцільно зменшити їх розмірність. Для цього було використано метод головних компонент. В таблиці 5 наведені результати такої декомпозиції.

Таблиця 5 – Залежність варіації параметрів від кількості головних компонент

Кількість компонент	Варіація	Приблизний час навчання
100	0.847366730496	2 год
200	0.890587442159	3 год
300	0.914868534936	4 год
500	0.952324353994	9 год

З таблиці можна побачити, що результат декомпозиції параметрів з кількістю компонент 300 є найбільш задовільним через те, що складність навчання підвищується пропорційно зі зростанням обсягу вхідних даних. За умов обмеженого часового ресурсу подальші дослідження було вирішено проводити на зображеннях зі 100-кратним збільшенням (всього 2312 зображення). Результати перевірки наведено у табл. 6.

Таблиця 6 – Точність класифікації при застосуванні 300 ознак

Кількість розбиття / кількість правил	Точність класифікації гістологічних зображень зі збільшенням			
	у 40 разів	у 100 разів	у 200 разів	у 400 разів
4 / 4	-	81.43 %	-	-
4 / 6	-	87.35 %	-	-
4 / 8	-	88.22 %	-	-

Результати експериментів показують, що точність знизилась на декілька відсотків. Ймовірно, це сталось через те, що ми завдяки декомпозиції

відкинули деякі важливі ознаки. Напротивагу, було суттєво зменшено час навчання. А зниження точності не таке критичне.

Також було досліджено вплив кількості ознак на точність діагностування зображень зі збільшенням у 100 разів (табл. 7).

Таблиця 7 – Точність діагностування при різних кількостях ознак

Кількість розбиття / кількість правил	Точність класифікації гістологічних зображень з кількістю ознак			
	100	300	500	4096
4 / 4	74.93 %	81.14 %	-	76.5 %
4 / 6	82.84 %	86.94 %	-	91,2 %
4 / 8	84.12 %	88.21 %	-	89.7 %

Результати експериментів показують, що точність діагностування зростає зі збільшенням кількості ознак, проте дуже відчутно, в середньому на 3-5 % при порівнянні досліджень з 100 та 300 ознак. При порівнянні з зображеннями з 4096 ознак було встановлено, що зміни кількості ознак у 20 разів призводять до незначного падіння точності, в середньому на 2-3 %. Це може свідчити про те, що ознаки мають незначний вплив на результати класифікації.

Необхідно зазначити, що для більш детальних результатів потрібно провести більшу кількість експериментів. Можливо, при додаткових налаштуваннях параметрів систем класифікації можна буде отримати кращий результат.

Таким чином, можна зробити висновок, що для класифікації гістологічних зображень тканин молочних залоз можна застосовувати модель після декомпозиції ознак.

4 ОХОРОНА ПРАЦІ

Згідно зі статтею 13 Закону України "Про охорону праці", роботодавець має зобов'язання забезпечувати працівникам безпечні та здорові умови праці, а також визначає права працівників на створення таких умов. Цей закон встановлює загальні принципи організації охорони праці в Україні та визначає порядок взаємодії між роботодавцем та працівниками з питань безпеки, гігієни праці та виробничого середовища.

Під час дослідження та розробки системи діагностики раку молочної залози на працівника могли мати вплив такі небезпечні та шкідливі виробничі фактори:

а) фізичні:

- підвищена запиленість та загазованість повітря робочої зони;
- підвищений рівень шуму на робочому місці;
- підвищена чи понижена вологість повітря;
- підвищений рівень електромагнітного випромінювання;
- підвищена чи понижена іонізація повітря;
- недостатня освітленість робочої зони;
- підвищена яскравість світла; понижена контрастність;
- пряма і відбита блискість.

б) психофізіологічні: статичне перевантаження та розумове перевантаження [29].

Відповідно до наведених факторів здійснюємо розробку заходів щодо безпечного виконання поставленого завдання.

4.1 Технічні рішення щодо безпечного виконання роботи

Вимоги щодо організації та обладнання робочих місць під час дослідження та розробки системи діагностики раку молочної залози: площа,

відведена на одне робоче місце має становити не менше 6 кв. м., а об'єм – не менше 20 куб. м. Конструкція робочого місця повинна забезпечувати підтримання оптимальної робочої пози (тобто такої, яка дозволяє працівникові виконувати роботу з мінімальним напруженням тіла, і яка дозволяє уникнути перевтоми в ході і після закінчення робочого процесу). Раціональна робоча поза має важливе значення для збереження здоров'я працівника, оскільки тривале перебування його в незручній і напруженій позі може призвести до таких захворювань, як сколіоз (викривлення хребта), варикозне розширення вен, плоскостопість тощо. Установлено, що робота в зігнутому положенні збільшує затрати енергії на 20 %, а при значному нахиленні — на 45 % порівняно з прямим положенням корпусу [30].

За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5 - 2 м.

Робочі місця слід розташовувати відносно джерела природного світла (вікон) таким чином, щоб світло падало збоку, переважно зліва. Також робоче місце має відповідати сучасним вимогам ергономіки:

- стіл повинен мати висоту поверхні 680 - 800 мм., ширину 600 - 1400 мм. і глибину 800 - 1000 мм. (такі параметри забезпечують можливість виконання операцій в зоні досяжності працівника);
- робочий стілець робочий стілець має бути підйомно-поворотним, з можливістю регулювання висоти, бажано зі стаціонарними або змінними підлікотниками і напівм'якою нековзкою поверхнею сидіння, що легко чиститься і не електризується;
- екран комп'ютера має розташовуватися на оптимальній відстані від користувача, що становить 600 – 700 мм., але не менше за 600 мм. з урахуванням літерно-цифрових знаків і символів.

Приміщення, де здійснювалося дослідження та розробки системи діагностики раку молочної залози за небезпекою ураження електричним

струмом належить до приміщень без підвищеної небезпеки (сухе, мало заповишене, з нормальною температурою повітря, ізолюваними підлогами і малим числом заземлених приладів).

Персональні комп'ютери, периферійні пристрої, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію.

Лінія електромережі для живлення персональних комп'ютерів і периферійних пристроїв виконується як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів.

Усі провідники відповідають номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння людини під напругу.

Персональні комп'ютери і периферійні пристрої підключаються до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного

провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним.

4.2 Технічні рішення з гігієни праці та виробничої санітарії

4.2.1 Мікроклімат

Стан повітря робочої зони у виробничому приміщенні називають мікрокліматом або метеорологічними умовами. Мікроклімат або метеорологічні умови виробничих приміщенні, визначаються за такими параметрами:

- температурою повітря у приміщенні, С;
- відносною вологістю повітря, %;
- рухливістю повітря, м/с;
- тепловим випромінюванням, Вт/м³.

Всі ці параметри поодиночі, а також у комплексі впливають на фізіологічну функцію організму його терморегуляцію і визначають самопочуття.

Дослідження та розробки системи діагностики раку молочної залози за енерговитратами відноситься до категорії I а (енерговитрати до 139 Дж/с) [31]. Допустимі параметри мікроклімату для цієї категорії згідно ДСН 3.3.6.042-99 [31] наведені в табл. 8.

Таблиця 8 – Параметри мікроклімату

Період року	Допустимі		
	t, °C	W, %	V, м/с
Теплий	22-28	55	0,1-0,2
Холодний	21-25	75	0,1

Для створення і автоматичної підтримки в приміщенні незалежно від зовнішніх умов допустимих значень температури, вологості, чистоти і швидкості руху повітря обладнані системами опалення та кондиціонування повітря. Систематично проводиться вологе прибирання.

4.2.2 Склад повітря робочої зони

Оточуюче нас повітря (атмосфера) є найважливішим фактором забезпечення життя. В природних умовах повітря, як правило, не забруднене отруйними речовинами і життю людини не загрожує. Органи чутливості людини не дозволяють з достатньою точністю визначати якість повітря і запобігати загрозі отруєння.

В приміщенні, де здійснюється дослідження та розробки системи діагностики раку молочної залози, у повітрі можуть перевищувати ГДК такі речовини як вуглекислий газ, пил та озон. Джерелами цих речовин є офісна техніка. Пил потрапляє у приміщення ззовні через відкриті вікна та заноситься на одязі і взутті працівниками.

ГДК шкідливих речовин, які знаходяться в досліджуваному приміщенні, наведені в таблиці 9.

Таблиця 9 – ГДК шкідливих речовин у повітрі

Назва речовини	ГДК, мг/м ³	Середньо добова	Клас небезпечності
	Максимально разова		
Вуглекислий газ	3	1	4
Пил нетоксичний	0,5	0,15	4
Озон	0,16	0,03	4

Рівні позитивних і негативних іонів у повітрі мають відповідати санітарно-гігієнічним нормам (табл. 10).

Таблиця 10 – Рівні іонізації повітря приміщень при роботі на ПК

Рівні	Кількість іонів в 1 см ³	
	n ⁺	n ⁻
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально необхідні	50000	50000

Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачати установки або прилади зволоження або штучної іонізації, кондиціонування повітря.

4.2.3 Виробниче освітлення

Освітлення відіграє важливу роль у житті людини. Біля 90 % інформації сприймається через зоровий канал, тому правильно виконане раціональне освітлення має важливе значення для виконання всіх видів робіт. Недостатня освітленість або її надмірна кількість знижують рівень збудженості центральної нервової системи і, природно, активність усіх життєвих процесів. Раціональне освітлення є важливим фактором загальної культури виробництва. Неможливо забезпечити чистоту та порядок у приміщенні, в якому напівтемрява, світильники брудні або в занедбаному стані.

Приміщення, в яких встановлені персональні комп'ютери, повинні мати природне та штучне освітлення відповідно до ДБН В.2.5-28-2018 [33]. Норми освітленості при штучному освітленні та КПО (для III пояса світлового клімату) при природному та сумісному освітленні зазначені у таблиці 11:

Штучне освітлення в досліджуваному приміщенні здійснюється системою загального рівномірного освітлення. У разі переважної роботи з документами, допускається застосування системи комбінованого освітлення

(крім системи загального освітлення додатково встановлюються світильники місцевого освітлення).

Таблиця 11 – Норми освітленості в приміщенні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнювання	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фона	Освітленість, лк		КПО, e_n , %			
						Штучне освітлення		Природне освітлення		Сумісне освітлення	
						Комбіноване	Загальне	Верхнє або верхнє	Бокове	Верхнє або верхнє	Бокове
Дуже високої точності	Від 0,15 до 0,3	II	г	великий	світлий	1000	300	7	2,5	4,2	1,5

Для забезпечення достатнього освітлення передбачені такі заходи:

- 1) Максимальне використання бічного природного освітлення.
- 2) Систематичне очищення скла від бруду – не рідше двох разів на рік.
- 3) Штучне освітлення в приміщенні забезпечується світильниками типу РСР08×250 (однолампові) з лампами ДРЛ-250.

4.2.4 Виробничий шум

Виробничий шум – це сукупність різних за гучністю і тоном звуків, які виникають у повітряному середовищі. В досліджуваному приміщенні наявний як постійний, так і непостійний шуми. Нормування непостійного шуму, а також орієнтовна оцінка загального рівня постійного шуму здійснюється скоректованим за частотою загальним рівнем звукового тиску – так званим рівнем звуку, який вимірюється в дБА за шкалою «А» шумоміра.

Непостійний шум характеризується еквівалентним рівнем звуку LA екв., що являє собою середньоквадратичний рівень звуку непостійного шуму, який має такий самий вплив на людину, як і постійний шум.

Для умов виконання роботи допустимі рівні звукового тиску не повинні перевищувати 50 дБА (табл.12).

Таблиця 12 – Допустимі рівні звукового тиску і рівні звуку для постійного широкополосного шуму (згідно ДСН 3.3.6.037-99 [34])

Характер робіт	Допустимі рівні звукового тиску (дБ) в стандартизованих октавних смугах зі середньгеометричними частинами (Гц)									Допуст-мий рівень звуку, дБА
	32	63	125	250	500	1000	2000	4000	8000	
Виробничі приміщення	86	71	61	54	49	45	42	40	38	50

Рівень шуму в приміщенні не перевищує допустимих значень.

4.2.5 Виробничі випромінювання

Розрізняють природні та штучні джерела електромагнітних полів (ЕМП). У процесі еволюції біосфера постійно перебуває під впливом ЕМП природного походження (природний фон): електричне та магнітне поля Землі, космічні ЕМП, передусім ті, що генеруються Сонцем. У період науково-технічного прогресу людство створило і все ширше використовує штучні джерела ЕМП. У теперішній час ЕМП антропогенного походження значно перевищують природний фон і є тим несприятливим чинником, чий вплив на людину з року в рік зростає.

Значення напруженості електростатичного поля на робочих місцях (як у зоні екрана дисплея, так і на поверхнях обладнання, клавіатури, друкувального пристрою) мають не перевищувати гранично допустимих за ГОСТ 12.1.045-84 [35]. Значення напруженості електромагнітних полів на

робочих місцях з ВДТ мають відповідати нормативним значенням (ДСанПіН 3.3.6-2002 [36], ГОСТ 12.1.045-84 [35]). Інтенсивність потоків інфрачервоного випромінювання має не перевищувати допустимих значень відповідно до ГОСТ 12.1.005-88 [37] (табл.13).

Таблиця. 13 - Допустимі параметри електромагнітних неіонізуючих випромінювань і електростатистичного поля

Види поля	Допустима поверхнева щільність потоку енергії, Вт/кв.м
Електромагнітне поле оптичного діапазону в ультрафіолетовій частині спектру УФ-С (220 — 280 нм)	0,001
Електромагнітне поле оптичного діапазону в ультрафіолетовій частині спектру УФ-В (280 — 320 нм)	0,01
Електромагнітне поле оптичного діапазону в ультрафіолетовій частині спектру УФ-А (320 — 400 нм)	10,0
Електромагнітне поле оптичного діапазону в видимій частині спектру 400 — 760 нм	10,0
Електромагнітне поле оптичного діапазону в інфрачервоній частині спектру 0,76 — 10,0 мкм	35,0 — 70,0
Напруженість електричного поля відеодисплейного терміналу	20кВ/м

Для дотримання наведених нормативів слід використовувати офісну техніку з сертифікатом якості та дотримуватися встановлених режимів праці та відпочинку з ПК.

4.3 Пожежна безпека

Пожежна безпека – стан об’єкта, при якому з регламентованою ймовірністю відкидається можливість виникнення та розвиток пожежі, і впливу на людей її небезпечних факторів, а також забезпечується захист матеріальних цінностей.

З урахуванням ДСТУ Б В.1.1-36:2016 [38] приміщення, де здійснюється дослідження та розробки системи діагностики раку молочної залози відноситься до категорії вибухопожежонебезпеки В (тверді горючі та важкогорючі речовини і матеріали, за умови, що приміщення, в яких вони знаходяться, не відносяться до категорій А, Б і питома пожежне навантаження для твердих і рідких легкозаймистих та горючих речовин на окремих ділянках площею не менше 10 м² кожна перевищує 180 МДж/м²).

Клас приміщення і зон з вибухо- і пожежонебезпеки – П-Па (приміщення, у якому знаходяться тверді горючі речовини та матеріали) [39].

4.3.1 Технічні рішення системи запобігання пожежі

Усі заходи системи запобігання пожежі в приміщенні можна поділити на організаційні, технічні, режимні та експлуатаційні.

Організаційні заходи пожежної безпеки передбачають: організацію пожежної охорони на об'єкті, проведення навчань з питань пожежної безпеки (включаючи інструктажі та пожежно-технічні мінімуми), застосування наочних засобів протипожежної пропаганди та агітації, проведення перевірок, оглядів стану пожежної безпеки приміщень, будівель, об'єкта в цілому та ін.

До технічних заходів належать: суворе дотримання правил і норм, визначених чинними нормативними документами при реконструкції приміщень, будівель та об'єктів, технічному переоснащенні виробництва, експлуатації чи можливому переобладнанні електромереж, опалення, вентиляції, освітлення і т. ін.

Заходи режимного характеру передбачають заборону куріння та застосування відкритого вогню в недозволених місцях, недопущення появи сторонніх осіб у вибухонебезпечних приміщеннях чи об'єктах, регламентацію пожежної безпеки при проведенні вогневих робіт тощо.

Експлуатаційні заходи охоплюють своєчасне проведення профілактичних оглядів, випробувань, ремонтів технологічного та допоміжного устаткування, а також інженерного господарства (електромереж, електроустановок, опалення, вентиляції)

4.3.2 Технічні рішення системи протипожежного захисту

Система протипожежного захисту – це сукупність організаційних заходів а також технічних засобів, спрямованих на запобігання впливу на людей небезпечних чинників пожежі та обмеження матеріальних збитків від неї.

До основних засобів пожежогасіння, що використовуються в приміщеннях такого класу відносять вогнегасники, розміщення яких в приміщеннях позначається встановленим поруч вказівним знаком

Згідно категорії пожежовибухонебезпеки будівлі, класу приміщення і за вибухо- і пожежонебезпекою П-Па та Наказу Міністерства внутрішніх справ України «Про затвердження Правил експлуатації та типових норм належності вогнегасників», в приміщенні має бути встановлено 2 порошкових або вуглекислотних вогнегасники із зарядом речовини від 3 до 5 кг. [40]

В цілому приміщення по категорії вибухо- і пожежонебезпечності та ступеню вогнестійкості відповідає нормам, але особливу увагу потрібно звернути на утримання в справному стані засобів протипожежного захисту та своєчасне інформування пожежної охорони про несправність пожежної техніки, впровадження систем протипожежного захисту.

ВИСНОВКИ

В бакалаврській кваліфікаційній роботі було розроблено питання систем автоматизованої діагностики раку молочної залози на основі комп'ютерного аналізу гістологічних мікроскопічних зображень.

Автоматизована діагностика раку молочної залози робить вагомий внесок у підвищення якості медичної допомоги, знижує витрати ресурсів, і сприяє прогресу в лікуванні та профілактиці цього небезпечного захворювання. Основними перевагами використання інформаційних технологій є: підвищення точності діагностики, швидкість обробки даних, підвищення доступності медичної допомоги та зниження вартості медичних послуг, підтримка прийняття рішень лікарями, стандартизація діагностики підтримка наукових досліджень.

Аналіз наявних автоматизованих систем для задач медичної діагностики показав, що вони не повною мірою задовольняють вимоги до вирішення завдань, які потребують складних логічних висновків в умовах високого ступеня невизначеності, неповноти та суперечливості вихідних даних. Перспективним виявився підхід, що полягає в інтелектуалізації цих систем на основі концепції експертних систем, які пропонують рішення для завдань, які важко формалізувати.

Тому для вирішення завдань БКР було розроблено автоматизовану інформаційну систему підтримки прийняття рішень лікарем для класифікації гістологічних зображень тканин молочних залоз, отриманих шляхом біопсії. Матеріалом дослідження стала відкрита база даних зображень BreakHis, яка містить 9109 зображень доброякісних і злоякісних пухлин молочних залоз у різному збільшенні.

Автоматизована система була розроблена на основі штучної нейронної мережі згорткового типу у поєднанні з класичним персептроном на виході. Крім того, для задач оптимізації в системі було використано методи

градієнтного спуску, алгоритм «стрибка в басейн» та алгоритм диференціальної еволюції.

Для навчання системи на основі нейромереж NEFCLASS і VGG-16 було використано 75 % зображень заданої бази даних. Для тестування – решту 25 % зображень. Точність діагностування склала від 90 % до 92 % для зображень з різним збільшенням, що є співмірним результатом у порівнянні з аналогічними системами. Для підвищення швидкодії системи було використано декомпозицію зображень з метою зменшення кількості ознак. Така процедура призводить до зменшення точності діагностування на 2-3 %, однак значно скорочує час діагностування, що особливо важливо для задач скринінгової діагностики в публічних закладах охорони здоров'я.

Також в роботі розглянуті питання охорони праці при розробці інформаційної системи, розраховані відповідні гігієнічні та санітарні показники виробничого приміщення і робочого місця.

Таким чином, всі завдання БКР виконані, мета роботи досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Як визначити рак грудей: симптоми, причини, діагностування захворювання [Електронний ресурс] / Онлайн-медицина. URL : <https://doc.ua/ua/news/articles/mozhno-li-diagnostirovat-u-sebya-rak-grudi>
2. Рак молочної залози [Електронний ресурс] / Онлайн-медицина. URL : <https://doc.ua/ua/bolezni/rak-molochnoj-zhelezy>
3. Дідик І. В. Рак передміхурової залози як світова проблема. Поширеність, фактори ризику, своєчасність діагностики (огляд літератури). *Український медичний часопис*. 2016. № 3 (113). С. 62–64.
4. Aloraifi, F., McDevitt, T., Martiniano, R., et al. Detection of novel germline mutations for breast cancer in non-BRCA ½ families. *The FEBS Journal*. 2015. Vol. 282 (17). P. 3424-3437.
5. Горда О., Цюцюра М. Когнітивні технології предметної області на основі онтології. *Управління розвитком складних систем*, 2023. № 53. С. 30–38.
6. Постевка І. Д. Сучасний стан проблеми скринінгу раку молочної залози (огляд літератури). *Буковинський медичний вісник*/ 2016. № 20 (78). С. 199–203.
7. Wang L. Early diagnosis of breast cancer. *Sensors*. 2017. Vol. 17 (7). Art. 1572. <https://doi.org/10.3390/s17071572>
8. He, Z., Chen, Z., Tan, M., et al. A review on methods for diagnosis of breast cancer cells and tissues. *Cell proliferation*. 2020. Vol. 53 (7). Art. e12822. <https://doi.org/10.1111/cpr.12822>
9. Lei, S., Zheng, R., Zhang, S., et al. Breast cancer incidence and mortality in women in China: temporal trends and projections to 2030. *Cancer biology & medicine*. 2021. Vol. 18 (3). P. 900–909.
10. M. Kowal, P. Filipczuk, A. Obuchowicz, J. Korbicz, R. Monczak, Computer-aided diagnosis of breast cancer based on fine needle biopsy microscopic

images Computers in Biology and Medicine, vol. 43, no. 10, pp. 1563–1572, 2013

11. Kaur, P., Singh, G., Kaur, P. Intellectual detection and validation of automated mammogram breast cancer images by multi-class SVM using deep learning classification. *Informatics in Medicine Unlocked*. 2019. Vol. 16. 100151. <https://doi.org/10.1016/j.imu.2019.01.001>

12. Zhou, L. Q., Wu, X. L., Huang, S. Y., et al. Lymph node metastasis prediction from primary breast cancer US images using deep learning. *Radiology*. 2020. Vol. 294 (1). P. 19–28.

13. Fan, K., Wen, S., Deng, Z. Deep learning for detecting breast cancer metastases on WSI. *Innovation in Medicine and Healthcare Systems, and Multimedia: Proceedings of KES-InMed-19 and KES-IIMSS-19 Conferences*. Springer, Singapore, 2019. P. 137-145.

14. Ye, J., Luo, Y., Zhu, C., Liu, F., Zhang, Y. Breast cancer image classification on WSI with spatial correlations. *ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019. P. 1219-1223.

15. Zeiser, F. A., da Costa, C. A., de Oliveira Ramos, G., Bohn, H. C., Santos, I., Roehe, A. V. DeepBatch: A hybrid deep learning model for interpretable diagnosis of breast cancer in whole-slide images. *Expert Systems with Applications*. 2021. Vol. 185, 115586.

16. Jahn, S. W., Plass, M., Moinfar, F. Digital pathology: advantages, limitations and emerging perspectives. *Journal of clinical medicine*. 2020. Vol. 9 (11), 3697.

17. Wakili, M. A., Shehu, H. A., Sharif, M. H., Sharif, M. H. U., Umar, A., Kusetogullari, H., Ince, I. F., Uyaver, S. Classification of Breast Cancer Histopathological Images Using DenseNet and Transfer Learning. *Computational intelligence and neuroscience*. 2022. 8904768. URL : <https://doi.org/10.1155/2022/8904768>

18. Imagenet classification with deep convolutional neural networks /

Krizhevsky A., Sutskever I., Hinton G. E. *Advances in neural information processing systems*. 2012. P. 1097-1105.

19. Moura, D. C., Guevara López, M. A. An evaluation of image descriptors combined with clinical data for breast cancer diagnosis. *International journal of computer assisted radiology and surgery*, 2013. 8, P. 561-574.

20. LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998. 86 (11). P. 2278—2324.

21. Chirra, V. R. R., Uyyala, S. R., Kolli, V. K. K. Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State. *Rev. d'Intelligence Artif.*, 2019. Vol. 33 (6). P. 461-466.

22. Spanhol, F. A., Oliveira, L. S., Petitjean, C., Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE transactions on biomedical engineering*. 2015. Vol. 63 (7). P. 1455-1462.

23. Boadh, R., Grover, R., Dahiya, M., et al. Study of fuzzy expert system for the diagnosis of various types of cancer. *Materials Today: Proceedings*. 2022. Vol. 56. P. 298-307.

24. Kour, H., Manhas, J., Sharma, V. Usage and implementation of neuro-fuzzy systems for classification and prediction in the diagnosis of different types of medical disorders: a decade review. *Artificial Intelligence Review*, 2020. Vol. 53, P. 4651-4706.

25. Yousefi J. A modified NEFCLASS classifier with enhanced accuracy-interpretability trade-off for datasets with skewed feature values. *Fuzzy Sets and Systems*, 2021. Vol. 413. P. 99-113.

26. Васильєва Л. В., Богдан М. П. Математичні методи дослідження операцій : навчальний посібник. Краматорськ : ДДМА, 2018. 144 с.

27. Storn, R., Price, K. (1995). Differential evolution — a simple and efficient scheme for global optimization over continuous spaces. *FTP. ICSI. Berkeley*. URL : <https://cse.engineering.nyu.edu/~mleung/CS909/s04/Storn95->

012.pdf

28. Wales, D. J., Doye, J. P. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*. 1997. Vol. 101 (28). P. 5111–5116.

29. ДСТУ ОHSAS 18002:2015. Системи управління гігієною та безпекою праці. Основні принципи виконання вимог ОHSAS 18001:2007 (ОHSAS 18002:2008, IDT). К. : ГП «УкрНИИУЦ», 2016. 21 с.

30. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: http://sop.zpr.ua/norm_npraop_0_00-7_15-18_01_ua.php.

31. Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - [Електронний ресурс] - Режим доступу: <http://zakon4.rada.gov.ua/laws/show/v4137400-86>

32. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. - [Електронний ресурс] - Режим доступу: <http://mozdocs.kiev.ua/view.php?id=1972>

33. ДБН В.2.5-28-20018 Природне і штучне освітлення - [Електронний ресурс] - Режим доступу: <http://document.ua/prirodne-i-shtuchne-osvitlennja-nor8425.html>

34. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. - [Електронний ресурс] - Режим доступу: <http://document.ua/sanitarni-normi-virobnichogo-shumu-ultrazvuku-ta-infravzuku-nor4878.html>

35. ДСНіПЗ.3.6.096-2002. Державні санітарні норми і правила при роботі з джерелами електромагнітних полів. URL: <http://zakon2.rada.gov.ua/laws/show/z0203-03>.

36. ДСанПіН 3.3.6-2002 Державні санітарні норми і правила при роботі з джерелами електромагнітних полів - [Електронний ресурс] - Режим доступу:

<http://zakon.nau.ua/doc/?code=z0203-03>

37. НПАОП 0.00-7.11-12. Загальні вимоги стосовно забезпечення роботодавцями охорони праці працівників. URL: <http://zakon.rada.gov.ua/laws/show/z0226-12>.

38. ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпек. URL: https://dbn.co.ua/load/normativy/dstu/dstu_b_v_1_1_36/5-1-0-1759.

39. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги. URL: http://www.poliplast.ua/doc/dbn_v.1.1-7-2002.pdf.

40. Наказ Міністерства внутрішніх справ України «Про затвердження Правил експлуатації та типових норм належності вогнегасників». URL: <https://zakon.rada.gov.ua/laws/show/z0225-18#Text>.

ДОДАТКИ

Додаток А
(довідковий)

ПРИКЛАДИ ЗОБРАЖЕНЬ З БАЗИ ДАНИХ BREAKNIS

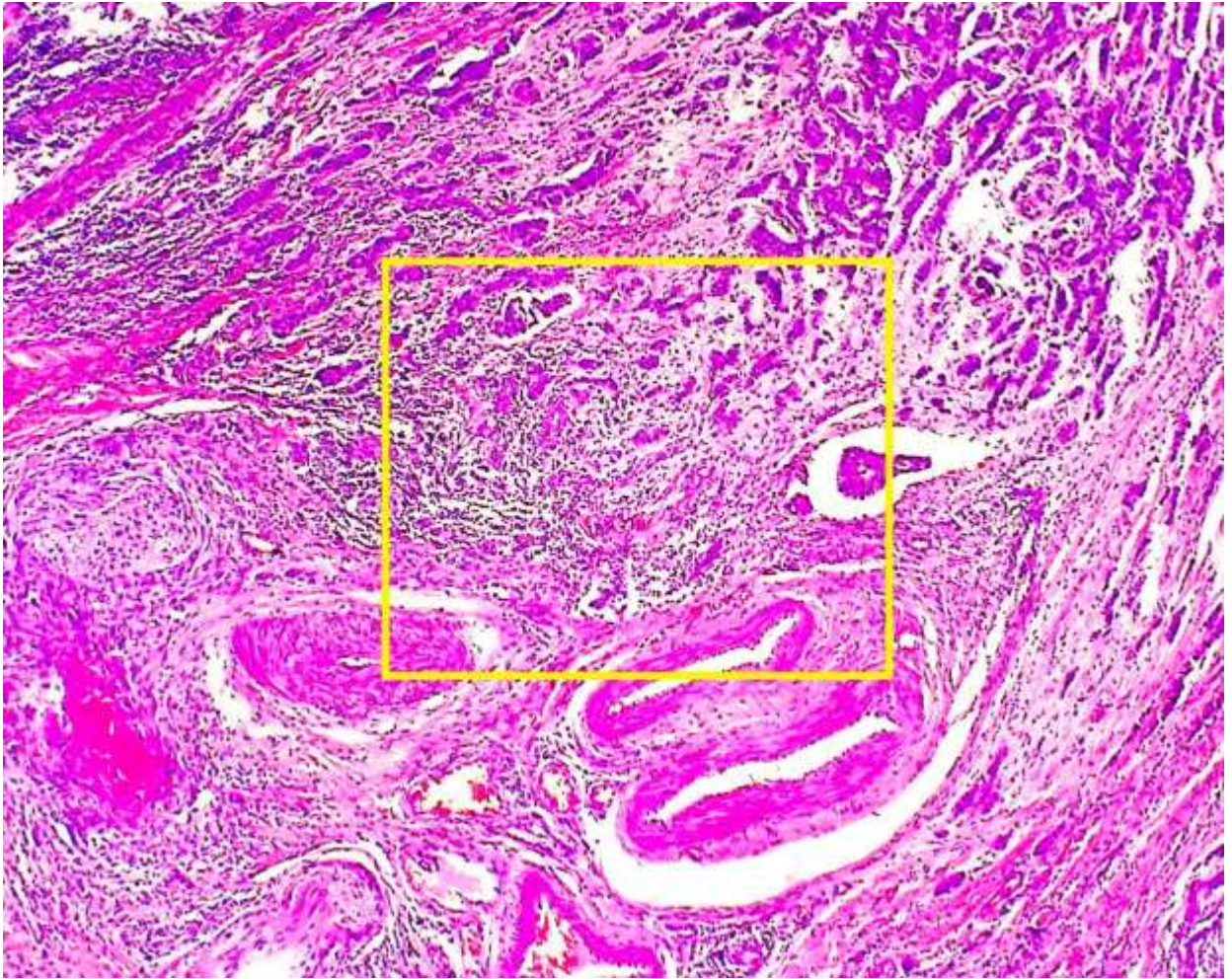


Рисунок А.1 – Гістологічне зображення злоякісної пухлини зі збільшенням у 40 разів. Виділена жовтим квадратом ділянка буде представлена у великому збільшенні на наступних рисунках

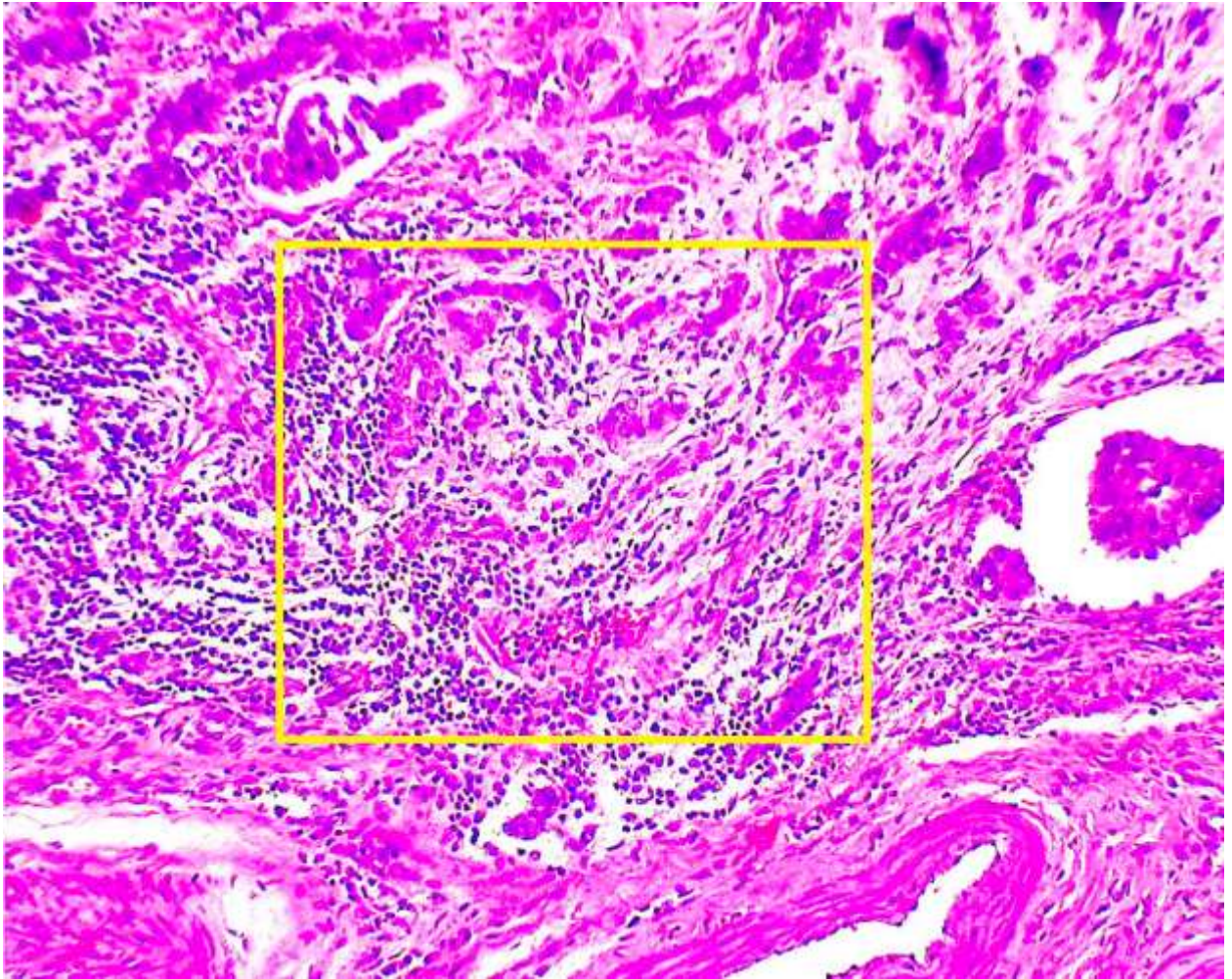


Рисунок А.2 – Гістологічне зображення злоякісної пухлини зі збільшенням у 100 разів. Виділена жовтим квадратом ділянка буде представлена у великому збільшенні на наступних рисунках

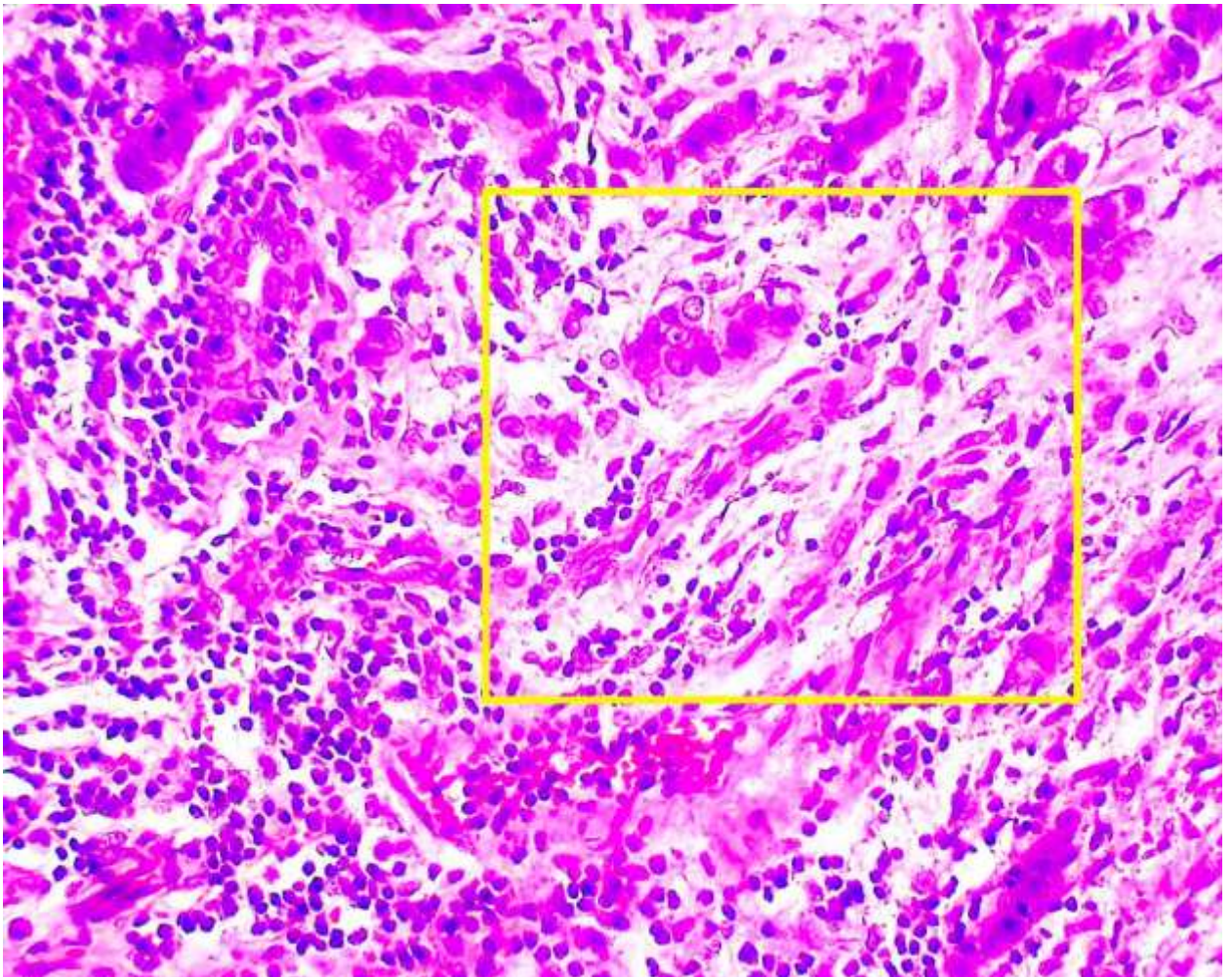


Рисунок А.3 – Гістологічне зображення злоякісної пухлини зі збільшенням у 200 разів. Виділена жовтим квадратом ділянка буде представлена у великому збільшенні на наступному рисунку

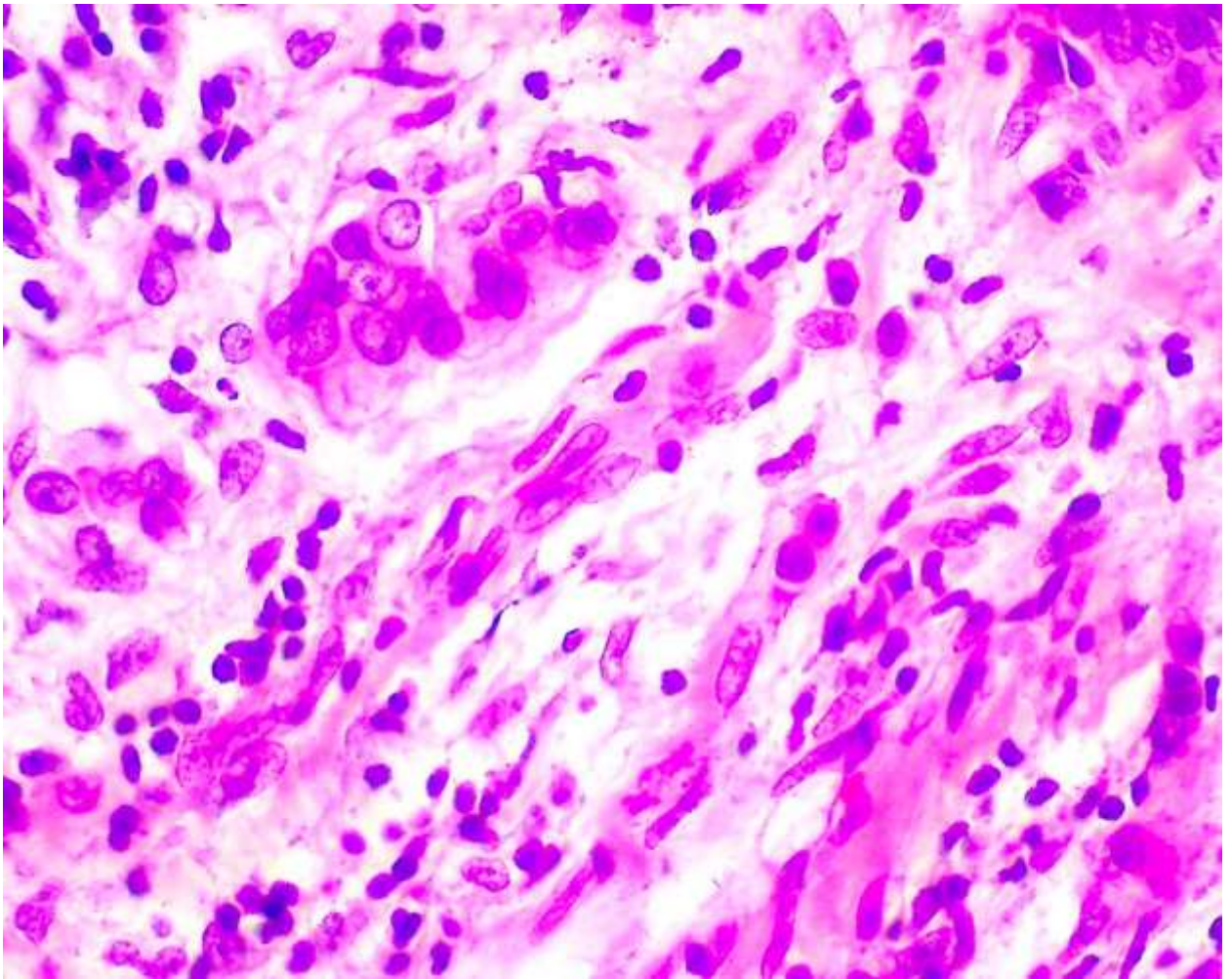


Рисунок А.4 – Гістологічне зображення злоякісної пухлини
зі збільшенням у 400 разів

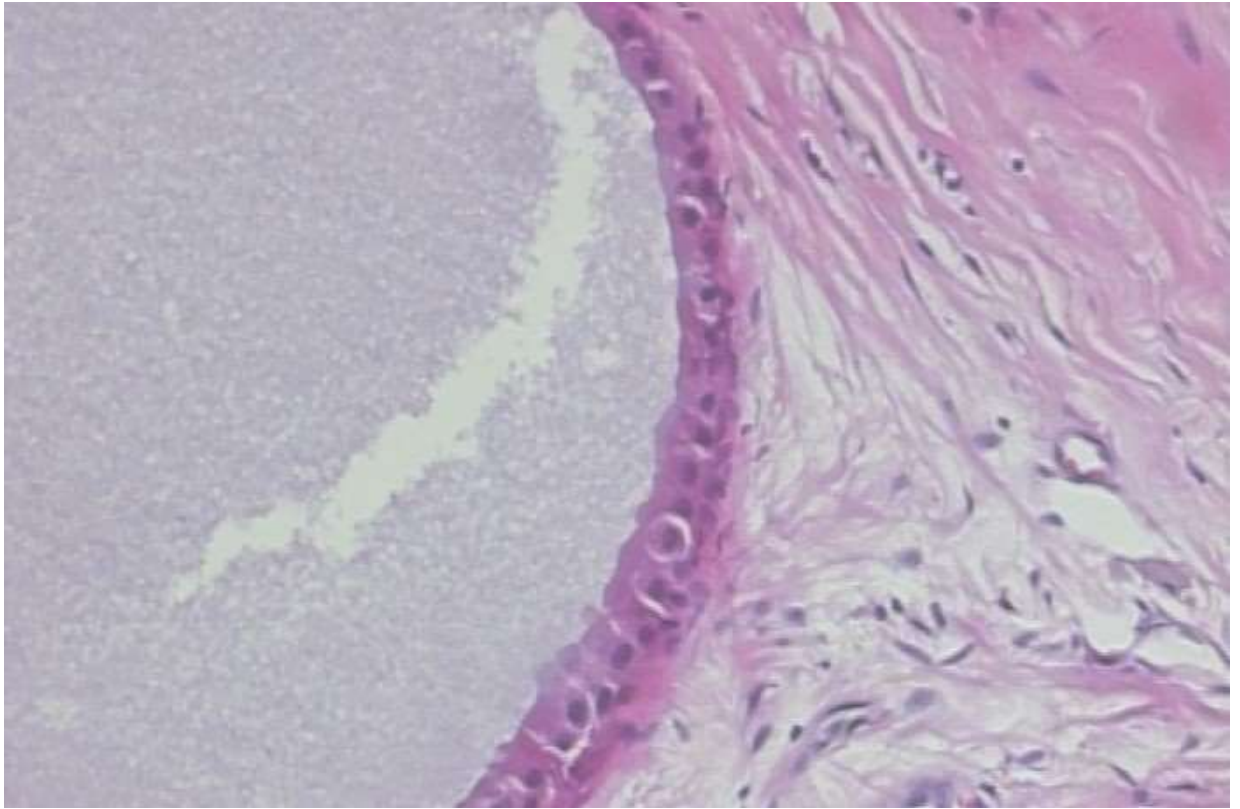


Рисунок А.5 – Гістологічне зображення доброякісної пухлини
зі збільшенням у 100 разів

Додаток Б
(довідковий)

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

Лістинг програми

```

import tensorflow as tf
import numpy as np
from scipy.misc import imread, imresize from keras.applications.vgg16 import VGG16from
keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input, decode_predictions
from imagenet_classes import class_names
import os

class vgg16:
    def __init__(self, imgs, weights=None, sess=None):
        self.imgs = imgs
        self.convlayers()
        self.fc_layers()
        self.probs = tf.nn.softmax(self.fc3l)
        if weights is not None and sess is not None:
            self.load_weights(weights, sess)
    def convlayers(self):
        self.parameters = [] # zero-mean input
        with tf.name_scope('preprocess') as scope:
            mean = tf.constant([123.68, 116.779, 103.939], dtype=tf.float32, shape=[1, 1, 1, 3],
                                name='img_mean')
            images = self.imgs - mean
        # conv1_1
        with tf.name_scope('conv1_1') as scope:
            kernel = tf.Variable(tf.truncated_normal([3, 3, 3, 64], dtype=tf.float32, stddev=1e-1),
                                name='weights')
            conv = tf.nn.conv2d(images, kernel, [1, 1, 1, 1], padding='SAME')
            biases = tf.Variable(tf.constant(0.0, shape=[64], dtype=tf.float32), trainable=True,
                                name='biases')
            out = tf.nn.bias_add(conv, biases)
            self.conv1_1 = tf.nn.relu(out, name=scope)
            self.parameters += [kernel, biases]
        # conv1_2
        with tf.name_scope('conv1_2') as scope:
            kernel = tf.Variable(tf.truncated_normal([3, 3, 64, 64], dtype=tf.float32, stddev=1e-1),
                                name='weights')
            conv = tf.nn.conv2d(self.conv1_1, kernel, [1, 1, 1, 1], padding='SAME')
            biases = tf.Variable(tf.constant(0.0, shape=[64], dtype=tf.float32), trainable=True,
                                name='biases')
            out = tf.nn.bias_add(conv, biases)
            self.conv1_2 = tf.nn.relu(out, name=scope)
            self.parameters += [kernel, biases]
        # pool1
        self.pool1 = tf.nn.max_pool(self.conv1_2, [1, 2, 2, 1], [1, 2, 2, 1], padding='SAME', name='pool1')
        # conv2_1
        with tf.name_scope('conv2_1') as scope:
            kernel = tf.Variable(tf.truncated_normal([3, 3, 64, 128], dtype=tf.float32, stddev=1e-1),
                                name='weights')
            conv = tf.nn.conv2d(self.pool1, kernel, [1, 1, 1, 1], padding='SAME')
            biases = tf.Variable(tf.constant(0.0, shape=[128], dtype=tf.float32), trainable=True,
                                name='biases')
            out = tf.nn.bias_add(conv, biases)
            self.conv2_1 = tf.nn.relu(out, name=scope)
            self.parameters += [kernel, biases]
        # conv2_2
        with tf.name_scope('conv2_2') as scope:
            kernel = tf.Variable(tf.truncated_normal([3, 3, 128, 128], dtype=tf.float32, stddev=1e-1),
                                name='weights')
            conv = tf.nn.conv2d(self.conv2_1, kernel, [1, 1, 1, 1], padding='SAME')
            biases = tf.Variable(tf.constant(0.0, shape=[128], dtype=tf.float32), trainable=True,
                                name='biases')
            out = tf.nn.bias_add(conv, biases)
            self.conv2_2 = tf.nn.relu(out, name=scope)
            self.parameters += [kernel, biases]
        # pool2
        self.pool2 = tf.nn.max_pool(self.conv2_2, [1, 2, 2, 1], [1, 2, 2, 1], padding='SAME', name='pool2')
        # conv3_1
        with tf.name_scope('conv3_1') as scope:
            kernel = tf.Variable(tf.truncated_normal([3, 3, 128, 256], dtype=tf.float32, stddev=1e-1),
                                name='weights')
            conv = tf.nn.conv2d(self.pool2, kernel, [1, 1, 1, 1], padding='SAME')

```

```

biases = tf.Variable(tf.constant(0.0, shape=[256], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv3_1 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# conv3_2
with tf.name_scope('conv3_2') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 256, 256], dtype=tf.float32,
stddev=1e-1), name='weights') conv =
tf.nn.conv2d(self.conv3_1, kernel, [1, 1, 1, 1], padding='SAME') biases =
tf.Variable(tf.constant(0.0, shape=[256], dtype=tf.float32),
trainable=True, name='biases') out = tf.nn.bias_add(conv, biases)
self.conv3_2 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# conv3_3
with tf.name_scope('conv3_3') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 256, 256], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.conv3_2, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[256], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv3_3 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# pool3
self.pool3 = tf.nn.max_pool(self.conv3_3, ksize=[1, 2, 2, 1], strides=[1, 2, 2,
1], padding='SAME', name='pool3')
# conv4_1
with tf.name_scope('conv4_1') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 256, 512], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.pool3, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[512], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv4_1 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# conv4_2
with tf.name_scope('conv4_2') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 512, 512], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.conv4_1, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[512], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv4_2 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# conv4_3
with tf.name_scope('conv4_3') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 512, 512], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.conv4_2, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[512], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv4_3 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# pool4
self.pool4 = tf.nn.max_pool(self.conv4_3, ksize=[1, 2, 2, 1], strides=[1, 2, 2,
1], padding='SAME', name='pool4')
# conv5_1
with tf.name_scope('conv5_1') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 512, 512], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.pool4, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[512], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv5_1 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# conv5_2
with tf.name_scope('conv5_2') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 512, 512], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.conv5_1, kernel, [1, 1, 1, 1], padding='SAME')

```

```

biases = tf.Variable(tf.constant(0.0, shape=[512], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv5_2 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# conv5_3
with tf.name_scope('conv5_3') as scope:
    kernel = tf.Variable(tf.truncated_normal([3, 3, 512, 512], dtype=tf.float32, stddev=1e-1),
name='weights')
conv = tf.nn.conv2d(self.conv5_2, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[512], dtype=tf.float32), trainable=True,
name='biases') out = tf.nn.bias_add(conv, biases)
self.conv5_3 = tf.nn.relu(out, name=scope) self.parameters += [kernel, biases]
# pool5
self.pool5 = tf.nn.max_pool(self.conv5_3, ksize=[1, 2, 2, 1], strides=[1, 2, 2,
1], padding='SAME', name='pool4')
def fc_layers(self):
    # fc1
    with tf.name_scope('fc1') as scope:
        shape = int(np.prod(self.pool5.get_shape()[1:]))
        fc1w = tf.Variable(tf.truncated_normal([shape, 4096], dtype=tf.float32, stddev=1e-1),
name='weights') fc1b = tf.Variable(tf.constant(1.0, shape=[4096],
dtype=tf.float32), trainable=True, name='biases') pool5_flat = tf.reshape(self.pool5, [-1,
shape])
        fc1l = tf.nn.bias_add(tf.matmul(pool5_flat, fc1w), fc1b) self.fc1 = tf.nn.relu(fc1l)
        self.parameters += [fc1w, fc1b]
    # fc2
    with tf.name_scope('fc2') as scope:
        fc2w = tf.Variable(tf.truncated_normal([4096, 4096], dtype=tf.float32, stddev=1e-1),
name='weights') fc2b = tf.Variable(tf.constant(1.0, shape=[4096],
dtype=tf.float32), trainable=True, name='biases') fc2l = tf.nn.bias_add(tf.matmul(self.fc1, fc2w),
fc2b)
        self.fc2 = tf.nn.relu(fc2l) self.parameters += [fc2w, fc2b]
    # fc3
    with tf.name_scope('fc3') as scope:
        fc3w = tf.Variable(tf.truncated_normal([4096, 1000], dtype=tf.float32, stddev=1e-1),
name='weights') fc3b = tf.Variable(tf.constant(1.0, shape=[1000],
dtype=tf.float32), trainable=True, name='biases') self.fc3l = tf.nn.bias_add(tf.matmul(self.fc2,
fc3w), fc3b)
        self.parameters += [fc3w, fc3b]
def load_weights(self, weight_file, sess): weights = np.load(weight_file)
keys = sorted(weights.keys())
for i, k in enumerate(keys): print(i, k, np.shape(weights[k]))
sess.run(self.parameters[i].assign(weights[k]))

if __name__ == '__main__': sess = tf.Session()
imgs = tf.placeholder(tf.float32, [None, 224, 224, 3]) #model = VGG16(include_top=False,
weights='imagenet') #model.save_weights()
vgg = vgg16(imgs, 'vgg16_weights.npz', sess)
#running for classification folder
i=0
for file in os.listdir('/Users/igorvarga/PycharmProjects/classifying-
cancer/cnn_image_classifier/images/predict/benign'):
    if file.endswith('.png'):
        name=os.path.join('/Users/igorvarga/PycharmProjects/classifying-
cancer/cnn_image_classifier/images/predict/benign/',file) base_name = name[:-4]
        img1 = imread(name)
        img1 = imresize(img1, (224, 224))
        #img1 = img1[:, :, 0:-1]
        prob = sess.run(vgg.fc2, feed_dict={vgg.imgs: [img1]})[0] np.save(base_name, prob)
        print(len(prob)) i = i+1
        #for x in prob: # if x == 0.0:
        # i=i+1
        print(i)

```

```
import math
```

```
class FuzzyNeuralNetwork(object):conj = None
```

```
out_f = None
```

```
"""docstring for FuzzyNeuralNetwork"""def _init_(self, rules):
```

```
    super(FuzzyNeuralNetwork, self)._init_()self.rules = rules
```

```
def _init_(self, rules,conjunct_func,out_func):super(FuzzyNeuralNetwork, self)._init_() self.rules = rules
```

```
    self.conj = conjunct_funcself.out_f = out_func
```

```
def activate(self, vec, fn=None,fo=None):
```

```
    if fn is None:
```

```
        return self.activate(vec,self.conj,self.out_f)res = []
```

```
    for i in range(len(self.rules)): res.append(self.rules[i](vec,fn))
```

```
    tres = []
```

```
    for x in zip(*res) :tv = 0.0
```

```
        for y in x :
```

```
            tv = fo(tv,y)tres.append(tv)
```

```
    return tres
```

```
def mape(self, test):tmp = 0
```

```
    for i in range(len(test)):v = test[i]
```

```
        clas = v[1]
```

```
        x = self.activate(v[0],self.conj,self.out_f)v = max(x)
```

```
        c = x.index(v)
```

```
        if c == clas:tmp += 1
```

```
return tmp/len(test)
```

```
#def activate2(self, vec):
```

```
# return self.activate(vec,self.conj,self.out_f)
```

```
class Rule(object):
```

```
    """docstring for Rule"""
```

```
def _init_(self, inp_w, out_w = None,cls = None): super(Rule, self)._init_()
```

```
    self.inp_weights = inp_wself.out_weights = out_wself.cl = cls
```

```
def _eq_(self, other):a = True
```

```
    if type(other) != type(self):
```

```
        return False
```

```
    if len(self.inp_weights) != len(other.inp_weights):
```

```
        return False
```

```
    for i in range(len(self.inp_weights)):
```

```
        a = a and (self.inp_weights[i] == other.inp_weights[i])
```

```
    for i in range(len(self.out_weights)):
```

```
        a = a and (self.out_weights[i] == other.out_weights[i])
```

```
    return a
```

```
def __call__(self, *args, **kwargs):
```

```
    """
```

```
    :param args: 1 - vector of input 2 - conjunction func
```

```
    :param kwargs:
```

```
    :return: vector output"""
```

```
    t_val = []
```

```
    if len(args[0]) != len(self.inp_weights) :
```

```
        raise Exception("lengths is not valid")for x in range(0,len(args[0])):
```

```
            t_val.append( self.inp_weights[x]( args[0][x] ) )
```

```
    #conjunction
```

```
    t_val1 = 1.0
```

```

for x in range(0, len(t_val)): t_val1 = args[1](t_val[x],t_val1)

t_val = [0.0] * len(self.out_weights)

for x in range(0,len(self.out_weights)): t_val[x] = self.out_weights[x] * t_val1
return t_val

def activate1(self, *args):
    """
    :param args: 1 - vector of input 2 - conjunction func
    :param kwargs:
    :return: vector output"""
    t_val = []
    if len(args[0]) != len(self.inp_weights) :
        raise Exception("lengths is not valid")
    for x in range(0,len(args[0])):
        t_val.append( self.inp_weights[x]( args[0][x] ) )
    #conjunction
    t_val1 = 1.0
    for x in range(0, len(t_val)): t_val1 = args[1](t_val[x],t_val1)
    return t_val1
    pass

class Gauss(object): """docstring for Gauss"""
    def __init__(self, a, b):
        super(Gauss, self).__init__()
        self.a = a
        self.b = b

    def __call__(self, *args):
        x = args[0]
        return math.exp(0 - ((x - self.a)*(x - self.a))/(self.b * self.b)))

    def __repr__(self):
        return "a : " + str(self.a) + " b: " + str(self.b)

    def __eq__(self, other):
        a = True
        if type(other) != type(self):
            return False
        a = a and (self.a == other.a) and (self.b == other.b)
        return a

def evaluate_interval(train_set):
    interv = []

    for x in zip(*train_set):
        lv = [min(x), max(x)]
        interv.append(lv)
    return interv

if __name__ == '__main__':
    lala = Rule([lambda x: x*x, lambda x: x*x*x],[123])
    lala([12.132434,11.4345],lambda x,y: x*y)
    import copy
    copya = 4
    ala = Gauss(a,1)
    ala2 = copy.copy(ala)
    a = 5
    print(ala2)
    #print lala.inp_weights
    class StandardDeviationNefclassAdditive():
        fnn = None
    desc = []
    rule_classes = []
    t_vals = []

    def initiate(self, descr, rul_cls, trn):
        self.desc = descr
        self.rule_classes = rul_cls
        self.t_vals = trn
        self.cnt = 0

    def restoreFNN(self,weights):
        self.cnt = self.cnt+1
        W = weights

```

```

wrc = W[0:len(self.rule_classes):] wgts = W[:0:len(self.rule_classes)]
wgts = list(chunks(wgts,int(len(wgts)/len(self.rule_classes))))

rules = []
for i in range(len(self.rule_classes)):rc = self.rule_classes[i]
    inter = wgts[i]
    b_s = inter[int(len(inter)/2):]a_s = inter[:int(len(inter)/2)]
    gaussses = [ns.Gauss(a_s[x],b_s[x]) for x in range(len(a_s))]ow = [0.0] * self.desc[0]
    ow[rc] = wrc[i]
    rule = ns.Rule(gaussses,ow)rule.rc = rc rules.append(rule)

con = lambda x,y : x*y out_fu = lambda x,y : x+y
fnn = ns.FuzzyNeuralNetwork(rules,con,out_fu)
return fnn

```

```

pass
def call(self, *args, **kwargs):self.cnt = self.cnt+1
W = args[0]
wrc = W[0:len(self.rule_classes):] wgts = W[:0:len(self.rule_classes)]
wgts = list(chunks(wgts,int(len(wgts)/len(self.rule_classes))))

rules = []
for i in range(len(self.rule_classes)):rc = self.rule_classes[i]
    inter = wgts[i]
    b_s = inter[int(len(inter)/2):]a_s = inter[:int(len(inter)/2)]
    gaussses = [ns.Gauss(a_s[x],b_s[x]) for x in range(len(a_s))]
    ow = [0.0] * self.desc[0]ow[rc] = wrc[i]
    rule = ns.Rule(gaussses,ow)rule.rc = rc rules.append(rule)

con = lambda x,y : x*y out_fu = lambda x,y : x+y
fnn = ns.FuzzyNeuralNetwork(rules,con,out_fu)

result = 0.0
#k = fnn.mape(train)#j = fnn.mape(test) #print(j)
for i in range(len(self.t_vals)):vec = self.t_vals[i]
    t_res = fnn.activate(vec[0])

    for j in range(len(t_res)):
        if j == vec[1]:
            result = result + (1 - t_res[j])*(1 - t_res[j])
        else:
            result = result + (0 - t_res[j]) * (0 - t_res[j])
    pass

return result/len(self.t_vals)

```

```

def train_weights():global t_rules import pickle

fi = open('/Users/igorvarga/nefclass-jupyter/fo.out','rb')
try:
    unpickler = pickle.Unpickler(fi)t_rules = unpickler.load()

except EOFError:
    return
#t_rules = pickle.load(fi)#t_rules = list(t_rules) global cl_cnt
global db global dc global test
#set output weight#

wins = [] wouts = [] r_classes = []

```

```

for i in range(len(t_rules)):v = t_rules[i]
    a_s = []
    b_s = []
    for j in range(len(v.inp_weights)):g = v.inp_weights[j] a_s.append(g.a)
        b_s.append(g.b)wins += a_s
    wins += b_s wouts.append(v.out_weights[v.cl])r_classes.append(v.cl)
devi = StandardDeviationNefclassAdditive()devi.initiate([cl_cnt],r_classes,train)
x0 = wins + wouts
import scipy.optimize as opt
res = opt.fmin_cg(devi,x0,maxiter=4000,gtol=0.003)resfnn = devi.restoreFNN(res)
print("'" + str(resfnn.mape(train)) + "," + str(resfnn.mape(test)))print("ddffkeffddfv")

def grad_func():
    pass

def train_rules():

    global inp_dimglobal perc global cl_cnt global train global test
    global training_algorithm
    global kmax
    global t_rules

    ##### TRAIN_RULES

    inp_dim = len(train[0][0])ts = [x[0] for x in train]
    interv = evaluate_interval(ts) t_interv = copy.deepcopy(interv)
    fir_gauss = evaluate_gauss_first(t_interv)

    antedescents = []

    for i in range(len(train)):cc = train[i][1]
        rule = create_antedescend(fir_gauss,train[i][0])ow = [0.0] * cl_cnt
        ow[cc] = 1.0 rule.out_weights = ow

        if rule in antedescents:
            pass else:
                rule.cl = cc antedescents.append(rule)

    #activation count per rule
    acc = [[0.0] * cl_cnt for x in range(len(antedescents))]

    for i in range(len(train)):cc = train[i][1] print(i)
        for j in range(len(antedescents)):x = acc[j][cc]
            x = x + antedescents[j].activate1(train[i][0],mult)acc[j][cc] = x
    print(len(antedescents))

    #evaluate performance
    perf = []
    for i in range(len(antedescents)):ant = acc[i]
        tmp_r = antedescents[i]t = max(ant)
        #antedescents[i].cl = ant.index(t)
        rs = ant[tmp_r.cl]
        for j in range(len(ant)):
            if j != tmp_r.cl: rs = rs - ant[j]
        perf.append(rs)

    #rules T
    t_rules = []

    if training_algorithm == "BEST_PER_CLASS":ppc = [] for x in range(cl_cnt)]

```

```

rpc = [[] for x in range(cl_cnt)]
for i in range(len(antedescents)): r = antedescents[i]
    rc = r.cl rpc[rc].append(r) ppc[rc].append(perf[i])

for i in range(cl_cnt): p = ppc[i]
    r = rpc[i]

    for j in range(int(kmax // cl_cnt)): v = max(p)
        x = p.index(v) t_rules.append(r[x])

        p.pop(x)
        r.pop(x)

    pass

if training_algorithm == "BEST":

    pass
    #print(fir_gauss)#return t_rules
    #print(num_of_weights)#return 43
    #fnn = ns.FuzzyNeuralNetwork(rules=21)
    pass

def evaluate_interval(data): interv = []
    #print(data)
    for x in (zip(*data)): lv = [min(x), max(x)] interv.append(lv)
    return interv

def evaluate_gauss_first(interv):
    global inp_dim
    global num_of_weightsres = []
    for i in range(0, inp_dim): tl = []
        delta = (interv[i][1] - interv[i][0]) / (num_of_weights - 1)
        for j in range(0, num_of_weights):
            g = ns.Gauss(interv[i][0] + delta * j, delta) tl.append(g)
        res.append(tl)
    return res

    pass

def create_ant descent(v_gauss, inp): vg = []
    for i in range(0, len(inp)): tmp = []
        for j in range(0, len(v_gauss[i])): tmp.append(copy.copy(v_gauss[i][j]))
        vec = find_max_in_weights(tmp, inp[i]) vg.append(vec)

    vg = ns.Rule(vg)
    return vg

gr = (math.sqrt(5) + 1) / 2
def goldensection_method(f, a, b, tol=1e-5): c = b - (b - a) / gr
    d = a + (b - a) / gr
    while abs(c - d) > tol:
        if f(c) < f(d): b = d
        else:
            a = c

    # we recompute both c and d here to avoid loss of precision which may lead to incorrect results or
    # infinite loop
    c = b - (b - a) / gr d = a + (b - a) / gr

    return (b + a) / 2

def find_max_in_weights(vec_gauss, x): max = 0

```

```

maxtmp = vec_gauss[0](x)
for i in range(len(vec_gauss)):
    if i > 0:
        tmp_v = vec_gauss[i](x)
        if tmp_v > maxtmp: maxtmp = tmp_v
        max = copy.copy(i)
    return copy.copy(vec_gauss[max])

if _name_ == '__main__': i = 2
j = i
i = i + 1 print(j)
td = file_reader("/Users/igorvarga/Documents/WBC/wdbc.data.txt") train_set = td[0]
lv = evaluate_interval(td[0]) print(lv)

```

Додаток В
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Дослідження та розробка системи діагностики раку
молочної залози

Тип роботи: БКР

Підрозділ: кафедра біомедичної інженерії та оптико-електронних систем

Показники звіту подібності Unichesk

Оригінальність 90,0 %

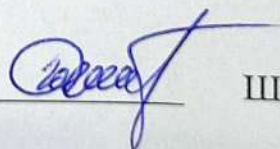
Схожість 10,0 %

☒ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

☐ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

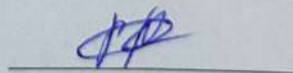
Особа, відповідальна за перевірку



Штофель Д. Х.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи



Подольнчук П. С.

Керівник роботи



Карась О. В.