

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

(повна назва інституту/факультету)

кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

(повна назва кафедри)

«До захисту допущено»

В.о. завідувачка кафедри БМК

Світлана АЛХІМОВА

(підпис)

(ініціали, ПРІЗВИЩЕ)

“ ” червня 2025р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Комп'ютерні технології в біології та медицині»

зі спеціальності 122 «Комп'ютерні науки»

на тему:

Програмний застосунок визначення рівня

стеатозу печінки за ультразвуковими зображеннями

Виконав: студент IV курсу, групи БС-11

КОЗАК ГРИГОРІЙ ГРИГОРОВИЧ

(прізвище, ім'я, по батькові)



(підпис)

Науковий керівник:

доцент каф. БМК,

доц., к.т.н., Павлов Володимир Анатолійович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по ініціали)



(підпис)

Рецензент:

доцент каф. біомедичної інженерії (БМІ),

доц., к.б.н., Калашнікова Лариса Євгенівна

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент



(підпис)

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
 Інститут (факультет) БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
 (повна назва)

Кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ
 (повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»
 (код і назва)

ЗАТВЕРДЖУЮ

В. о. завідувачка кафедри БМК

Світлана АЛХІМОВА

« » квітень 2025 р.

ЗАВДАННЯ на дипломну роботу студенту

КОЗАКУ ГРИГОРІЮ ГРИГОРОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи

***Програмний застосунок визначення рівня
стеатозу печінки за ультразвуковими зображеннями***

керівник роботи

Павлов Володимир Анатолійович доцент , к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «26» травня 2025 р. №1758-с

2. Термін подання студентом роботи

05 травня 2025 року

3. Вихідні дані *База даних з ультразвуковими зображеннями печінки*

4. Зміст роботи *Проаналізувати наявну на базі практики літературу про жировий стеатоз печінки. Проаналізувати та обрати засоби реалізації моделей глибокого навчання. Створити та протестувати моделі. Провести розрахунок економічного ефекту за темою ДР. Зробити загальні висновки та визначитись з остаточною темою ДР до наказу.*

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) *14 слайдів (Презентація захисту практики).*

6. Консультанти розділів роботи^{1*}

^{1*} Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

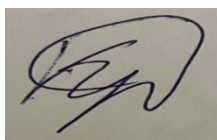
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Дипломної роботи			

7. Дата видачі завдання **17 березня 2025р.**

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 24.12.2024р.	виконано
2	Переддипломна практика. Виконання практичної частини ДР та додаткових розділів	1-4 тиждень за графіком практики	виконано
3	Виконання основних розділів ДР (Огляд існуючих методів визначення стеатозу, Визначення рівня стеатозу печінки з застосуванням нейронних мереж, Програмна реалізація системи визначення рівня стеатозу)	Протягом усієї практики	виконано
4	Перевірка ДР науковим керівником	до 20.05.2025	виконано
5	Подання в електронному вигляді ДР на перевірку нормоконтролера та плагіат (StrikePlagiarism.com). Доопрацювання ДР	Не пізніше 23.05.2025	виконано
6	Отримати відгук від керівника ДР	Не пізніше 05.06.2025	виконано
7	Надати пакету документів по ДР та супровідних до неї документів на засідання кафедри		виконано
8	Подання ДР рецензенту. Отримання рецензії.	09.06 – 13.06.2025	виконано
9	Захист ДР в ЕК	16.06 --21.06.2025	

Студент



(підпис)

Григорій КОЗАК

(ім'я, ПРІЗВИЩЕ)

Керівник ДР

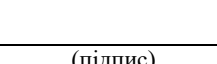


(підпис)

Володимир ПАВЛОВ

(ім'я, ПРІЗВИЩЕ)

Нормоконтролер



(підпис)

Галина КОРНІЄНКО

(ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Дипломна робота за темою «Програмний застосунок визначення рівня стеатозу печінки за ультразвуковими зображенням» виконана студентом кафедри біомедичної кібернетики ФБМІ Козаком Григорієм Григоровичем зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 3-х розділів (Огляд існуючих методів визначення стеатозу, Визначення рівня стеатозу печінки з застосуванням нейронних мереж, Програмна реалізація системи визначення рівня стеатозу), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 28 джерела, та додатків. Загальний обсяг роботи 70 сторінок.

Актуальність теми

Стеатоз печінки є однією з найпоширеніших хвороб, що виявляється приблизно у 25% населення світу. Традиційні методи діагностики (біопсія, МРТ, КТ) точні, але дорогі, інвазивні або малодоступні. УЗД є найбільш доступним, проте залежить від суб'єктивної оцінки лікаря. Тому актуальним є використання нейронних мереж для автоматизованої обробки зображень, що дозволяє знизити суб'єктивність, підвищити точність і пришвидшити скринінг пацієнтів.

Мета і завдання роботи

Метою роботи є вдосконалення методів визначення рівня жирового стеатозу печінки за ультразвуковими знімками засобами згорткових нейронних мереж.

Її досягнення передбачає вирішення наступних завдань:

1. Порівняти сучасні методи діагностики стеатозу.
2. Провести аналіз даних ультразвукових знімків.
3. Натренувати моделі для автоматичного визначення рівня стеатозу.
4. Створити програмне забезпечення для зручного аналізу нових зображень.

5. Перевірити працездатність застосунку на тестових зображеннях.
6. Розглянути можливості вдосконалення методології.

Використані методи

У роботі використано згорткову нейронну мережу DenseNet121 для первинного аналізу ультразвукових знімків печінки, а також кастомну нейронну мережу другого рівня для узагальнення результатів по декількох знімках одного пацієнта. Для розробки програмного застосунку використано мову програмування Python, бібліотеки PyTorch, torchvision, NumPy та PyQt5 для створення графічного інтерфейсу користувача.

Отримані результати

Запропонована двоетапна система для оцінки рівня стеатозу печінки продемонструвала високу ефективність. На першому етапі модель DenseNet121 досягла середньої абсолютної помилки (MAE) 8.70 та кореляції Спірмена 0.916, що перевищує аналогічний показник (0.78), отриманий у роботі Byra et al. (2018) для моделі з використанням Inception-ResNet-v2 та Lasso-регресії. Усереднення результатів по 10 знімках одного пацієнта дозволило знизити MAE до 8.58. Найкращі результати були досягнуті на другому етапі — за допомогою кастомної нейронної мережі, яка обробляє усі 10 прогнозів: MAE становила 7.60, кореляція Спірмена — 0.931, Пірсона — 0.921. Це свідчить про високу точність і доцільність застосування багаторівневої архітектури для автоматизованого аналізу УЗ-знімків.

Бібліографічний опис ДР

Козак Г. Г. Програмний застосунок визначення рівня стеатозу печінки за ультразвуковими зображенням: дипломна робота бакалавра : 122 Комп'ютері науки / Козак Григорій Григорович. — Київ, 2025. — 70 с

ANNOTATION

The bachelor's thesis on the topic "Software application for liver steatosis level assessment based on ultrasound images" was completed by Hryhorii Kozak, a student of the Department of Biomedical Cybernetics at the Faculty of Biomedical Engineering and Medical Informatics, majoring in Computer Science (field of study 122) under the educational-professional program "Computer Technologies in Biology and Medicine". The thesis consists of: an introduction; three chapters (Overview of existing steatosis assessment methods, Liver steatosis level detection using neural networks, Software implementation of the steatosis level assessment system); conclusions to each of these chapters; general conclusions; a list of references comprising 28 sources; and appendices. The total volume of the thesis is 70 pages.

Relevance of the topic

Liver steatosis is one of the most widespread liver diseases, affecting approximately 25% of the global population. Traditional diagnostic methods such as biopsy, MRI, and CT are accurate but often costly, invasive, or limited in accessibility. Ultrasound is the most accessible method but depends heavily on the subjective judgment of the physician. Therefore, the use of neural networks for automated image processing is highly relevant, as it allows for reduced subjectivity, improved diagnostic accuracy, and accelerated patient screening.

Objective and tasks

The objective of the thesis is to improve methods for assessing liver steatosis level using convolutional neural networks applied to ultrasound images.

To achieve this objective, the following tasks were formulated:

1. Compare modern methods for steatosis diagnostics.
2. Analyze data from ultrasound liver images.
3. Train models for automatic liver steatosis level prediction.
4. Develop software for convenient analysis of new images.
5. Evaluate the performance of the application on test images.

6. Explore possibilities for methodology improvement.

Methods used

The work utilizes a convolutional neural network model (DenseNet121) for initial analysis of individual liver ultrasound images, along with a custom secondary neural network to aggregate results across multiple images from a single patient. The software application was developed using the Python programming language with PyTorch, torchvision, NumPy, and PyQt5 libraries for model implementation and graphical user interface development.

Results obtained

The proposed two-stage system for liver steatosis assessment demonstrated high effectiveness. In the first stage, the DenseNet121 model achieved a mean absolute error (MAE) of 8.70 and a Spearman correlation of 0.916, which surpasses the previously reported value of 0.78 by Byra et al. (2018) for their Inception-ResNet-v2 model with Lasso regression. Averaging predictions across 10 images per patient reduced the MAE to 8.58. The best results were achieved using a custom secondary neural network that processed all 10 predictions per patient: MAE = 7.60, Spearman correlation = 0.931, and Pearson correlation = 0.921. These results confirm the high accuracy and practical applicability of the multi-level architecture for automated analysis of ultrasound liver images.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП	11
РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ВИЗНАЧЕННЯ РІВНЯ СТЕАТОЗУ ПЕЧІНКИ	15
1.1 СТЕАТОЗ ПЕЧІНКИ: КЛІНІЧНА КАРТИНА ТА НАСЛІДКИ	15
1.2 МЕТОДИ ВИЗНАЧЕННЯ СТЕАТОЗУ: БІОПСІЯ, КТ, МРТ, УЗД	16
1.3. ПЕРЕВАГИ ТА НЕДОЛІКИ УЛЬТРАЗВУКОВОЇ ДІАГНОСТИКИ	17
1.4. НЕІНВАЗИВНІ МЕТОДИ ТА РОЛЬ АВТОМАТИЗОВАНИХ СИСТЕМ	19
Висновок з розділу 1	20
РОЗДІЛ 2 ВИЗНАЧЕННЯ РІВНЯ СТЕАТОЗУ ПЕЧІНКИ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ	21
2.1. Основи штучних нейронних мереж	21
2.2. Згорткові нейронні мережі для обробки медичних зображень	23
2.3. Ультразвукові зображення	25
2.4. Порівняння популярних архітектур CNN у медичній діагностиці	26
2.5. Принципи регресійного прогнозування в медичних задачах	28
2.6. Огляд сучасних підходів до поєднання декількох зображень для пацієнта	30
2.7. Текстурний аналіз як альтернатива або доповнення CNN	32
Висновки до розділу 2	34
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ СТЕАТОЗУ	36
3.1. Використані технології	36
3.1.1. Мова програмування	36

	9
3.1.2. <i>Фреймворк для глибокого навчання</i>	37
3.1.3. <i>Бібліотеки для роботи із зображеннями</i>	38
3.1.4. <i>Попередньо навчені архітектури моделей</i>	39
3.1.5. <i>Бібліотеки для обробки табличних даних і візуалізації</i>	40
3.1.6. <i>Використання графічного процесора</i>	41
3.1.7. <i>Середовище розробки</i>	42
3.1.8. <i>Бібліотека Cартит для пояснюваного машинного навчання</i>	43
3.2. ПРОГРАМНА РЕАЛІЗАЦІЯ	43
3.2.1. <i>Завантаження та обробка вхідних даних</i>	43
3.2.2. <i>Реалізація архітектури мережі</i>	48
3.2.3. <i>Навчання моделі та оцінка якості</i>	49
3.2.4. <i>Візуалізація важливих областей зображення</i>	56
3.2.5. <i>Реалізація другого етапу прогнозування на основі 10 зображень</i>	57
3.2.6. <i>Реалізація графічного інтерфейсу користувача</i>	59
3.3 РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ	62
Висновок до розділу 3	63
ЗАГАЛЬНІ ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

НАЖГП – неалкогольна жирова хвороба печінки

УЗД – ультразвукова діагностика

КТ – комп'ютерна томографія

МРТ – магнітно-резонансна томографія

ПЗ – програмний застосунок

НАСГ – неалкогольний стеатогепатит

РІ – регіони інтересу

ВСТУП

У сучасних умовах поширеності неалкогольної жирової хвороби печінки (НАЖХП) питання точної, доступної та неінвазивної діагностики стеатозу печінки набуває особливої актуальності. За статистикою, приблизно 25% населення світу страждає на жировий гепатоз, що робить це захворювання однією з найпоширеніших патологій печінки. Небезпека стеатозу полягає не лише в його поширеності, а й у потенційній можливості прогресування до фіброзу, цирозу або навіть гепатоцелюлярної карциноми, якщо вчасно не буде проведено діагностику та не вжито відповідних лікувальних заходів.

На сьогоднішній день золотим стандартом діагностики стеатозу є біопсія печінки, яка дозволяє точно визначити ступінь ураження тканини. Проте цей метод є інвазивним, дороговартісним та асоціюється з ризиками ускладнень, тому не придатний для масового скринінгу. Альтернативними є МРТ та КТ, але вони також залишаються малодоступними у багатьох клінічних умовах. Ультразвукове дослідження (УЗД), навпаки, є дешевим, неінвазивним і широко доступним методом, який активно використовується для первинного виявлення стеатозу. Проте ключовою проблемою УЗД-діагностики є її висока залежність від суб'єктивної оцінки лікаря, що призводить до варіабельності результатів.

У зв'язку з цим особливої актуальності набуває автоматизація процесу аналізу ультразвукових зображень шляхом використання методів глибинного навчання, зокрема згорткових нейронних мереж (CNN). Такі методи дозволяють зменшити вплив людського фактору, покращити точність оцінки, а також забезпечити швидкий і масштабований скринінг великої кількості пацієнтів. В основу даної роботи покладено ідеї, висвітлені у науковій публікації Byra et al. (2018)[7], у якій було застосовано transfer learning з використанням моделі Inception-ResNet-v2 для оцінки стеатозу печінки за УЗ-даними, що дало змогу досягти високої точності класифікації та кореляції з

результатами біопсії.

На базі цієї ідеї у даній дипломній роботі було реалізовано повноцінний програмний застосунок із графічним інтерфейсом для автоматичного визначення рівня стеатозу за ультразвуковими знімками, з використанням двоетапної моделі аналізу: перший етап – передбачення для кожного зображення окремо, другий етап – об'єднання результатів по 10 знімках одного пацієнта за допомогою спеціально натренованої невеликої нейронної мережі.

Мета і завдання роботи

Метою роботи є вдосконалення методів автоматизованого визначення рівня стеатозу печінки за ультразвуковими знімками з використанням глибинного навчання, зокрема згорткових нейронних мереж, і створення програмного застосунку для практичного використання.

Її досягнення передбачає вирішення наступних завдань:

1. Аналіз вітчизняних та зарубіжних наукових публікацій, пов'язаних з оцінкою стеатозу печінки за медичними зображеннями.
2. Огляд сучасних підходів до діагностики стеатозу, зокрема методів комп'ютерної обробки УЗ-даних.
3. Аналіз та попередня обробка УЗ-знімків печінки з відповідними гістологічними еталонами.
4. Побудова та навчання моделі глибокого навчання для оцінки стеатозу за окремими зображеннями.
5. Розробка другого етапу моделі, який об'єднує результати по декількох зображеннях для отримання узагальненої оцінки.
6. Розробка графічного інтерфейсу користувача для зручної роботи із застосунком.
7. Оцінка ефективності системи на тестовому наборі даних та порівняння з відомими методами.

Використані методи

У роботі реалізовано двоетапний підхід до оцінки стеатозу печінки:

- на першому етапі використано згорткову нейронну мережу

DenseNet121, модифіковану для задачі регресії, яка прогнозує рівень стеатозу за кожним окремим УЗ-зображенням;

- на другому етапі реалізовано кастомну легку нейронну мережу, яка обробляє 10 результатів по одному пацієнту і видає остаточний прогноз.

Для реалізації програмного забезпечення використано мову програмування Python та бібліотеки PyTorch, torchvision, NumPy, SciPy, PyQt5. Розроблено повноцінний графічний інтерфейс, який дозволяє проводити діагностику в одному з двох режимів: за одним знімком або за повним набором із 10 зображень пацієнта.

Отримані результати

Використання двоетапної системи дозволило досягти високої точності прогнозування рівня стеатозу.

- Модель першого рівня (DenseNet121) досягла $MAE = 8.70$, кореляція Спірмена = 0.916, що значно перевищує аналогічний показник у роботі [7] (0.78).
- При усередненні результатів по 10 знімках MAE зменшився до 8.58, а кореляція зросла.
- Найкращий результат було отримано за допомогою нейронної мережі другого рівня: $MAE = 7.60$, Spearman correlation = 0.931, Pearson correlation = 0.921.

Результати свідчать про високу точність запропонованої архітектури, її перевагу над базовими підходами та перспективність застосування у сфері медичної діагностики.

Структура роботи

Дипломна робота за темою «Програмний застосунок визначення рівня стеатозу печінки за ультразвуковими зображенням» виконана студентом *Козаком Григорієм Григоровичем* зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та

медицині», побудована за класичним типом та викладена на 56 сторінках машинописного тексту. Вона складається з: вступу; 3-х розділів (Огляд існуючих методів визначення стеатозу, Визначення рівня стеатозу печінки з застосуванням нейронних мереж, Програмна реалізація системи визначення рівня стеатозу); висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 28 джерел та додатків. В роботі представлено 24 рисунків і 2 таблиці.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ВИЗНАЧЕННЯ РІВНЯ СТЕАТОЗУ ПЕЧІНКИ

Стеатоз, часто спостерігається серед людей з зайвою вагою, являє собою найбільш розповсюджену форму гепатозу печінки, котра полягає у заповненні її відмерлих клітин жиром. У більшості випадків він виникає при дисбалансі надходження чи синтезу жирних кислот у печінці та їх утилізації[1], що часто пов'язано з резистентністю до інсуліну, діабетом 2-го типу чи ожирінням. Точна діагностика кількості жиру в печінці є значущою, адже рівень стеатозу корелює з метаболічним синдромом та серцево-судинними захворюваннями.

1.1 Стеатоз печінки: клінічна картина та наслідки

Стеатоз печінки (жирова хвороба печінки) — це стан, при якому в клітинах печінки (гепатоцитах) накопичується надлишкова кількість жиру, переважно у формі тригліцеридів. Найчастіше зустрічається неалкогольний стеатоз печінки (НАЖХП), поширеність якого у світі становить близько 25% [1].

Клінічно стеатоз зазвичай має безсимптомний перебіг, що ускладнює його раннє виявлення. У деяких випадках пацієнти можуть скаржитися на втому, важкість у правому підребер'ї або дискомфорт у животі, проте ці симптоми є неспецифічними [2].

Найбільшу загрозу становить прогресування стеатозу до фіброзу, цирозу печінки або навіть гепатоцелюлярної карциноми. Також НАЖХП асоціюється з підвищеним ризиком серцево-судинних захворювань [3].

Оскільки стеатоз часто розвивається без явних симптомів, особливу цінність набувають неінвазивні методи діагностики та автоматизовані системи, здатні виявляти патологію на ранніх стадіях [4].

1.2 Методи визначення стеатозу: біопсія, КТ, МРТ, УЗД

Для діагностики стеатозу печінки в клінічній практиці застосовуються як інвазивні, так і неінвазивні методи. Кожен із них має свої переваги та обмеження, що визначають їх доцільність у конкретних умовах.

Біопсія печінки вважається «золотим стандартом» визначення рівня стеатозу, оскільки дозволяє безпосередньо оцінити вміст ліпідів у гепатоцитах і виявити супутні зміни (фіброз, запалення) [1, 2]. Проте цей метод є інвазивним і пов'язаний із ризиками, такими як кровотеча, інфекція та больовий синдром, а також залежить від техніки забору тканини й суб'єктивної інтерпретації результатів.

Комп'ютерна томографія (КТ) забезпечує візуалізацію стеатозу завдяки зниженню щільності печінки порівняно із селезінкою. Метод є неінвазивним, але малочутливим до легких форм жирової інфільтрації й має променеве навантаження, що обмежує його використання для масового скринінгу [3].

Магнітно-резонансна томографія (МРТ), зокрема методи спектроскопії та PDFF (proton density fat fraction), дозволяє кількісно оцінити вміст жиру в печінці з високою точністю. МРТ є безпечним і чутливим методом, проте його висока вартість і недостатня доступність обмежують використання в рутинній практиці [4].

Ультразвукове дослідження (УЗД) — найбільш поширений метод візуалізації печінки, який використовується для первинного виявлення стеатозу. Він базується на зміні ехогенності тканин, яка зростає зі збільшенням вмісту жиру. Метод є доступним, безпечним і неінвазивним, однак має обмежену точність на ранніх стадіях стеатозу та є операторозалежним [5].

Порівняльну характеристику методів подано в табл. 1.1:

Таблиця 1.1.

Порівняння методів визначення стеатозу

Метод	Інвазивність	Точність	Вартість
Біопсія	Висока	Висока	Висока
КТ	Низька	Середня	Середня
MPT	Низька	Висока	Висока
УЗД	Низька	Середня	Низька

1.3. Переваги та недоліки ультразвукової діагностики

Ультразвукове дослідження (УЗД) є найбільш поширеним методом первинної візуалізації печінки в медичній практиці. Його популярність зумовлена неінвазивністю, відносною дешевизною, широкою доступністю та відсутністю променевого навантаження. Згідно з численними дослідженнями, УЗД дозволяє виявляти стеатоз при вмісті жиру понад 20–30% у гепатоцитах [5]. Прикладом ультра-звукового зображення з нашого датасету є рис. 1.1.



Рисунок 1.1 – Ультразвуковий знімок печінки

Переваги УЗД:

- **Безпека.** Метод не використовує іонізуюче випромінювання, тому може багаторазово застосовуватись навіть у вразливих групах пацієнтів (вагітні, діти) [5].
- **Доступність.** УЗД-апарати широко поширені, обстеження не вимагає спеціальної підготовки й триває недовго.
- **Вартість.** Значно нижча у порівнянні з МРТ або КТ, що робить метод придатним для масового скринінгу [3].
- **Можливість оцінки інших патологій.** Окрім стеатозу, УЗД може допомогти виявити інші зміни в структурі печінки (вогнищеві утворення, фіброз, портальну гіпертензію тощо).

Недоліки УЗД:

- **Низька чутливість до легких форм стеатозу.** На ранніх стадіях накопичення жиру ехогенність змінюється незначно, що ускладнює виявлення [5].
- **Операторозалежність.** Якість зображення та інтерпретація результатів залежать від досвіду та кваліфікації лікаря [6].
- **Обмеження при ожирінні.** У пацієнтів із високим індексом маси тіла якість зображення може бути значно зниженою через посилене поглинання ультразвукових хвиль жировими тканинами [6].
- **Суб'єктивність оцінки.** Критерії діагностики базуються на візуальних ознаках (підвищена ехогенність, затінення судин), які можуть бути інтерпретовані неоднозначно.

Ці недоліки УЗД актуалізують потребу в розробці автоматизованих систем аналізу зображень, що можуть зменшити суб'єктивний вплив людського фактора й покращити точність діагностики стеатозу на ранніх етапах розвитку захворювання.

1.4. Неінвазивні методи та роль автоматизованих систем

Оскільки стеатоз печінки у більшості випадків розвивається безсимптомно, особливо на ранніх стадіях, неінвазивні методи діагностики набули особливої актуальності. Їх перевага полягає в безпеці для пацієнта, можливості багаторазового використання, а також потенціалі до автоматизації й стандартизації.

Серед найбільш ефективних неінвазивних підходів — ультразвукова еластографія, яка дозволяє оцінити не тільки наявність жирової інфільтрації, але й жорсткість тканин печінки. Також перспективними є методи, що базуються на магнітно-резонансній томографії, зокрема PDFF, які забезпечують високу точність кількісного аналізу жиру [4].

Попри це, найбільш доступним методом залишається традиційне УЗД, що активно використовується для масового скринінгу. Саме тому у фокусі сучасних досліджень опинилися автоматизовані системи аналізу УЗ-зображень — вони покликані зменшити залежність від людського фактора, підвищити точність оцінки, стандартизувати висновки та пришвидшити діагностику.

Значного розвитку досягли методи на основі штучного інтелекту, зокрема згорткові нейронні мережі (CNN), які здатні навчатися на великій кількості зображень і точно класифікувати чи оцінювати рівень стеатозу [7]. Системи з глибоким навчанням можуть самостійно виявляти патерни, які є складними або малопомітними для людського ока, що особливо важливо у випадках легкого стеатозу.

Крім того, дослідники все частіше поєднують текстурний аналіз (наприклад, на основі GLCM) з глибоким навчанням для підвищення точності моделі [8]. Подібні системи вже демонструють результати, співставні або навіть вищі за точність традиційних методів діагностики, й мають великий потенціал для впровадження у практичну медицину.

Таким чином, неінвазивні методи діагностики стеатозу, доповнені сучасними алгоритмами штучного інтелекту, відкривають нові можливості для ефективного та масштабованого медичного скринінгу.

Висновок з розділу 1

У першому розділі було розглянуто сучасні підходи до діагностики стеатозу печінки. Захворювання є однією з найпоширеніших форм хронічної патології печінки, що часто має безсимптомний перебіг і може прогресувати до серйозних ускладнень, зокрема цирозу або гепатоцелюлярної карциноми. Основними методами виявлення стеатозу є біопсія, комп'ютерна томографія (КТ), магнітно-резонансна томографія (МРТ) та ультразвукове дослідження (УЗД). Попри високу точність біопсії та МРТ, саме УЗД залишається найбільш доступним і безпечним методом для скринінгу, що забезпечує швидку візуалізацію змін у тканинах печінки. Однак цей метод має обмеження, пов'язані з операторозалежністю та недостатньою чутливістю до ранніх стадій стеатозу.

У зв'язку з цим зростає роль **неінвазивних автоматизованих систем**, що базуються на методах штучного інтелекту, зокрема згорткових нейронних мережах. Такі системи мають потенціал підвищити точність діагностики, зменшити суб'єктивність оцінки результатів та забезпечити масштабованість у рамках систем охорони здоров'я.

Таким чином, поєднання традиційних методів УЗ-діагностики з сучасними алгоритмами машинного навчання є перспективним напрямом розвитку автоматизованої діагностики стеатозу печінки.

РОЗДІЛ 2

ВИЗНАЧЕННЯ РІВНЯ СТЕАТОЗУ ПЕЧІНКИ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ

Методи діагностики стеатозу печінки, які базуються на традиційних підходах — зокрема візуальна оцінка УЗ-зображень, біопсія, або обчислення індексу ехогенності (hepatorenal index), — мають певні обмеження. Основною проблемою залишається висока суб'єктивність, залежність від досвіду лікаря та необхідність ручного виділення областей інтересу. Це знижує точність діагностики, особливо на ранніх стадіях захворювання.

У зв'язку з цим усе більшого поширення набувають автоматизовані підходи, що базуються на глибокому навчанні, зокрема на згорткових нейронних мережах (CNN). Ці моделі здатні автоматично виділяти інформативні ознаки з медичних зображень без потреби в ручному втручанні. Крім того, застосування трансферного навчання дозволяє ефективно використовувати обмежені медичні датасети за допомогою моделей, попередньо натренованих на великих колекціях зображень, таких як ImageNet. Один із таких прикладів — робота Byra et al. (2018), де застосовано модель Inception-ResNet-v2 для оцінки рівня стеатозу печінки за В-режимом УЗ-зображень [7].

Цей розділ присвячено теоретичним аспектам використання нейронних мереж у задачі оцінки стеатозу печінки: від загальної архітектури мереж до підходів трансферного навчання, текстурного аналізу зображень та огляду актуальних досліджень у цій сфері.

2.1. Основи штучних нейронних мереж

Штучні нейронні мережі (ШНМ) — це один з основних інструментів сучасного машинного навчання, що імітує роботу людського мозку для

вирішення складних задач розпізнавання образів, класифікації та регресії. Класичні нейронні мережі складаються з пов'язаних між собою штучних нейронів, які об'єднуються у шари: вхідний, приховані та вихідний. Кожен нейрон виконує зважене підсумовування сигналів з попереднього шару та передає результат через активаційну функцію.

Останніми роками найбільшого поширення набула архітектура глибоких згорткових нейронних мереж (CNN, Convolutional Neural Networks), які призначені для обробки двовимірних структурованих даних, таких як зображення. CNN дозволяють автоматично виявляти просторові ознаки завдяки згортковим фільтрам, що аналізують локальні області зображення, поступово формуючи дедалі абстрактніші уявлення на кожному рівні мережі.

Однією з ключових переваг глибокого навчання є здатність моделей самостійно навчатися ефективних ознак із «сирих» даних, що усуває потребу в ручному виділенні характеристик, як це було характерно для класичних методів обробки зображень (GLCM, HI тощо). Проте глибокі нейронні мережі, як правило, потребують великої кількості даних для якісного навчання. У випадках, коли обсяг навчального набору обмежений (що характерно для медицини), часто застосовується трансферне навчання — перенавчання моделей, попередньо натренованих на великих датасетах, наприклад, ImageNet [1].

У межах даної роботи застосовано інший підхід — навчання моделі з нуля, без попередньо натренованих ваг. Це дозволяє повністю адаптувати модель до специфіки медичних зображень, однак вимагає ретельної попередньої обробки даних, використання регуляризаційних методів та потенційно більшого обсягу епох для досягнення збіжності. У роботі використано архітектуру DenseNet — мережу, в якій кожен шар безпосередньо з'єднаний з усіма попередніми шарами в межах блоку. Це дозволяє уникнути проблеми зникнення градієнтів, сприяє повторному використанню ознак і загалом підвищує ефективність навчання [2].

DenseNet належить до класу сучасних архітектур, які демонструють високу точність у задачах медичної діагностики. Завдяки щільним з'єднанням між шарами вона забезпечує покращене поширення інформації, сприяє більш ефективному навчанню та зменшенню кількості параметрів у порівнянні з традиційними мережами, такими як ResNet чи VGG [2].

2.2. Згорткові нейронні мережі для обробки медичних зображень

Згорткові нейронні мережі (CNN) є провідним інструментом для аналізу зображень у багатьох сферах, зокрема в медичній діагностиці. Завдяки здатності автоматично виділяти складні просторові ознаки з вхідного зображення, CNN стали основою для створення систем комп'ютерної діагностики (CAD) при аналізі знімків КТ, МРТ, УЗД, рентгенографії, гістологічних препаратів тощо [3].

Основною перевагою CNN у медичному застосуванні є їх здатність автоматично виявляти інформативні патерни, що можуть бути важкозаметними навіть для досвідченого лікаря. Зокрема, при аналізі УЗ-зображень печінки мережа може виділяти мікротекстурні відмінності, які корелюють зі ступенем жирового ураження, без необхідності попереднього сегментування або ручного вибору області інтересу (ROI).

Архітектура CNN зазвичай включає:

- згорткові шари, які формують локальні ознаки;
- шари підвибірки (pooling), що зменшують розмірність ознак;
- повнозв'язні шари, які об'єднують інформацію для прийняття рішень;
- активаційні функції (ReLU, sigmoid, tanh тощо), що додають нелінійність.

Особливої популярності набули архітектури глибокого навчання, такі як ResNet, Inception, DenseNet, які демонструють високу точність при меншій

кількості параметрів завдяки використанню новітніх концепцій — залишкових зв'язків (residual connections), щільних з'єднань (dense blocks) тощо.

У контексті задач діагностики стеатозу печінки CNN-архітектури активно використовуються для:

- класифікації (виявлення наявності стеатозу),
- регресії (оцінка ступеня ураження печінки у відсотках),
- сегментації (виділення зони ураження на зображенні).

Наприклад, у роботі Byra et al. (2018) було використано модель Inception-ResNet-v2, яка дозволила досягти високої точності при класифікації стеатозу (AUC = 0.977) за допомогою УЗ-зображень [7]. У рамках цієї дипломної роботи було використано іншу сучасну архітектуру — DenseNet (архітектуру наведено на рис. 2.1), яка забезпечує ефективну передачу інформації між шарами і є стійкою до проблеми затухання градієнта [2].

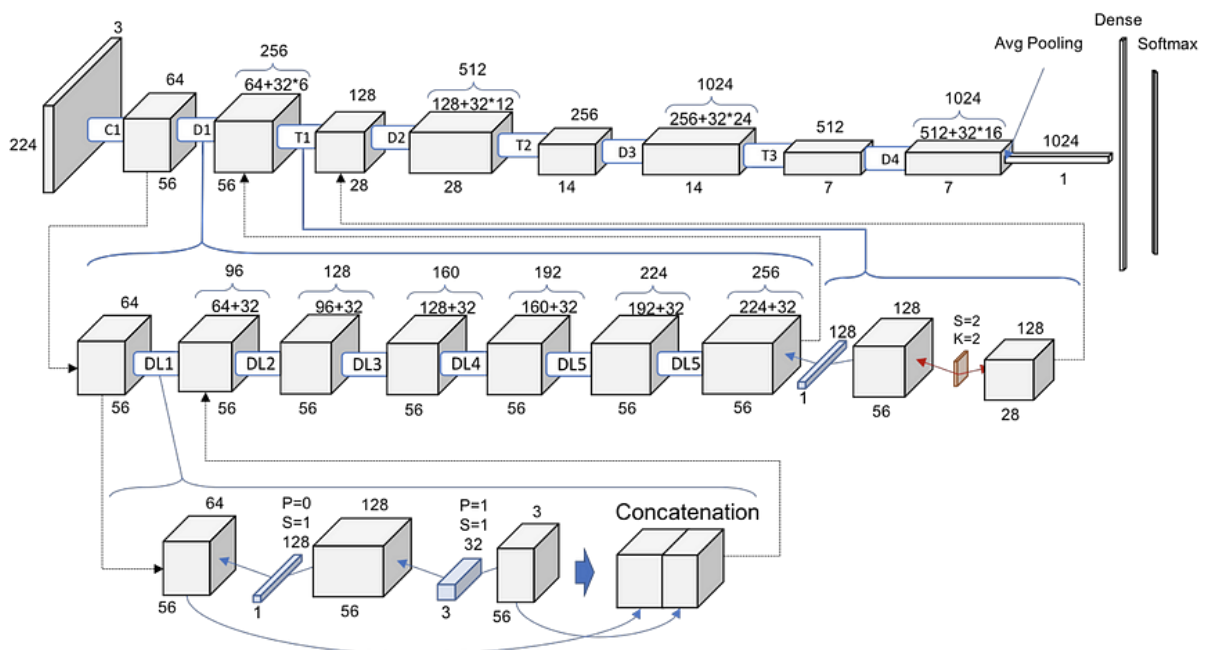


Рисунок 2.1 – Архітектура мережі Densenet

Під час обробки медичних зображень особливу увагу слід приділяти попередній підготовці даних — нормалізації яскравості, масштабуванню, усуненню артефактів. У задачах, де кількість доступних прикладів обмежена,

застосовуються методи аугментації даних (дзеркальне відображення, зміна яскравості, обертання тощо), що дозволяє підвищити узагальнюючу здатність моделі [4].

Таким чином, CNN довели свою ефективність у медицині, а завдяки сучасним архітектурам, таким як DenseNet, їх можна успішно застосовувати навіть на відносно невеликих вибірках, досягаючи високої точності при автоматизованому аналізі УЗ-зображень печінки.

2.3. Ультразвукові зображення

Ультразвукова візуалізація (УЗД) є одним із найпоширеніших методів неінвазивної діагностики печінкових захворювань. Вона ґрунтується на відображенні звукових хвиль від тканин різної щільності, що дозволяє формувати зображення в режимі реального часу. У контексті стеатозу печінки ультразвукове зображення дає змогу виявити зміну ехогенності тканини, що зростає зі збільшенням вмісту жиру в гепатоцитах [5].

Завдяки доступності, безпеці та низькій вартості, УЗД широко використовується як метод первинного скринінгу, а також для моніторингу перебігу захворювання. Однак традиційна інтерпретація УЗ-зображень залежить від досвіду лікаря й має високу суб'єктивність, особливо при легкому стеатозі. Крім того, зображення можуть бути спотворені артефактами, неоднорідною якістю чи індивідуальними особливостями пацієнтів (наприклад, наявність ожиріння) [6].

Попри це, УЗ-зображення мають великий потенціал для автоматизованої обробки, оскільки містять важливу текстурну та структурну інформацію, яка може бути проаналізована за допомогою нейронних мереж. Важливо, що навіть без контрасту або додаткових апаратних засобів, B-mode зображення печінки дають змогу виявити ключові ознаки стеатозу, які може бути важко помітити людському оку, але які здатна розпізнати модель глибокого навчання.

Ультразвукові зображення є сірими, з градацією яскравості пікселів, що відображає відбиту інтенсивність сигналу. Вони відзначаються високою варіативністю — залежно від кута сканування, глибини, налаштувань апарату. Саме ця варіативність ставить виклики, але й відкриває можливості для навчання моделей, здатних узагальнювати різні варіанти візуального представлення стеатозу.

Застосування методів машинного навчання, зокрема згорткових нейронних мереж, дозволяє аналізувати УЗ-зображення не лише на основі яскравості, а й на основі текстурних особливостей, виявляючи закономірності, що корелюють із кількісним рівнем стеатозу. Завдяки цьому УЗД, як джерело вхідних даних, є ідеальною платформою для створення доступних і масштабованих систем автоматизованої діагностики.

2.4. Порівняння популярних архітектур CNN у медичній діагностиці

У медичній діагностиці комп'ютерний зір і глибоке навчання посідають ключове місце в автоматизованому аналізі зображень. Різні архітектури згорткових нейронних мереж (CNN) демонструють різну ефективність залежно від обсягу даних, складності задачі та апаратних можливостей. У цьому підрозділі розглядаються популярні CNN-архітектури, які застосовуються в аналізі медичних зображень.

ResNet (Residual Network)

Архітектура ResNet вперше була запропонована у 2015 році та стала основою для багатьох сучасних моделей завдяки використанню **залишкових зв'язків** (residual connections), які полегшують проходження градієнта в глибоких мережах. ResNet дозволяє ефективно тренувати дуже глибокі моделі (від 50 до 152 шарів), що критично важливо для класифікації складних медичних зображень [11]. Наприклад, ResNet активно використовується для аналізу МРТ, КТ та мамограм, забезпечуючи хорошу генералізацію, особливо у великих датасетах.

Inception

Inception-архітектури, зокрема Inception-v3, базуються на ідеї **паралельної обробки ознак різного масштабу** в межах одного шару. Завдяки використанню фільтрів різного розміру, модель може ефективно фіксувати як локальні, так і глобальні патерни на зображеннях. Це особливо корисно при аналізі зображень зі складною текстурою, зокрема гістологічних або УЗ-знімків печінки [11].

Inception-ResNet-v2

Модель Inception-ResNet-v2 поєднує переваги архітектур Inception та ResNet. Вона була використана в роботі Byra et al. [7], яка стала методологічною основою цієї дипломної роботи. У зазначеному дослідженні модель застосовувалась для класифікації стеатозу печінки за ультразвуковими зображеннями. Автори продемонстрували високі результати точності — **AUC = 0.977** та **кореляцію Спірмена = 0.78**, використовуючи transfer learning і Lasso-регресію. Однак така модель є складною в реалізації, має значну обчислювальну вартість і не завжди підходить для застосування в реальному часі.

EfficientNet

EfficientNet — сучасна архітектура, що використовує оптимізацію глибини, ширини та роздільної здатності моделі одночасно (compound scaling) [Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML 2019] [19]. Ця архітектура забезпечує чудову точність при меншій кількості параметрів, порівняно з ResNet чи Inception. У медицині EfficientNet все частіше використовується для задач класифікації шкірних новоутворень, рентгенограм, сегментації органів. Проте, для регресійних задач зі специфічними обмеженими даними, як у випадку стеатозу печінки, її застосування потребує додаткового налаштування [11].

DenseNet (обрана архітектура)

Архітектура **DenseNet** (Densely Connected Convolutional Network) була запропонована у 2017 році і отримала широке визнання завдяки щільним

з'єднанням між шарами [9]. Кожен шар отримує як вхід всі ознаки, сформовані у попередніх шарах, що сприяє ефективному поширенню градієнтів та повторному використанню ознак. Це дозволяє досягати **високої точності навіть на невеликих датасетах**, що є особливо актуальним у медичних застосуваннях [11].

У цій роботі було використано саме **DenseNet121** як базову архітектуру першого рівня. Основні переваги:

- низьке переобчислення ознак;
- висока продуктивність на обмежених даних;
- менше параметрів у порівнянні з ResNet та Inception;
- швидка збіжність при тренуванні.

Таким чином, DenseNet була обрана як **оптимальне рішення** для побудови моделі автоматичного визначення рівня стеатозу печінки за УЗ-зображеннями — з огляду на її ефективність, економічність, простоту реалізації та стабільність результатів.

2.5. Принципи регресійного прогнозування в медичних задачах

У сфері медичного аналізу зображень задачі машинного навчання поділяються, переважно, на класифікаційні та регресійні. Кожен тип задачі має свої цілі, підходи до побудови моделей та способи оцінки ефективності.

- Класифікація полягає у визначенні до якого класу належить об'єкт. У медичному контексті це можуть бути задачі, як-от: виявлення наявності/відсутності патології, визначення стадії захворювання, або віднесення знімку до певної категорії (наприклад, стеатоз є чи нема) [11].
- Регресія, натомість, полягає у передбаченні неперервного числового значення. Вона особливо важлива у випадках, коли потрібно оцінити ступінь патології, кількість ураженої тканини, рівень біомаркера тощо.

Оцінка стеатозу печінки передбачає визначення ступеня жирового ураження у відсотках або за шкалою, що відображає континуум (наприклад, 0–

100%). Саме тому задача не може бути ефективно зведена до простої класифікації (наприклад, "є/нема стеатозу"), оскільки це призведе до втрати великої кількості важливої діагностичної інформації. Таким чином, модель має передбачити реальне числове значення рівня стеатозу, що і робить це типовою регресійною задачею [7], [13].

Для регресійного навчання в даній роботі була обрана функція втрат L1 Loss (або Mean Absolute Error Loss). Її особливість полягає в тому, що вона менш чутлива до викидів (outliers) у порівнянні з L2 Loss (MSE), що важливо при роботі з медичними зображеннями, які можуть містити артефакти або некоректно розмічені приклади. Крім того, L1 Loss більш інтерпретована з клінічної точки зору, оскільки безпосередньо відображає середнє відхилення прогнозів від істинних значень [11], [9].

У межах цієї роботи для оцінки якості передбачення використовувались такі метрики:

- MAE (Mean Absolute Error) — середнє абсолютне відхилення передбачень від реальних значень. Добре підходить для медичних задач, оскільки дає зрозумілу інтерпретацію похибки (у тій самій одиниці виміру, що і сама величина).
- Spearman correlation coefficient — непараметричний показник монотонної залежності, який дозволяє оцінити, наскільки правильно модель зберігає порядок між передбаченнями та істинними значеннями. Ця метрика особливо цінна у задачах, де абсолютні значення можуть варіюватися, але важливо зберігати ранжування пацієнтів за тяжкістю захворювання [7].
- Pearson correlation coefficient — показує лінійний зв'язок між прогнозом і еталонними значеннями. Високі значення цієї метрики (близько до 1.0) свідчать про хорошу відповідність передбачень реальним даним з точки зору лінійної залежності.

У сукупності використання MAE, Spearman і Pearson дозволяє отримати всебічну оцінку якості моделі: від точності числового передбачення до якості відображення клінічного порядку.

2.6. Огляд сучасних підходів до поєднання декількох зображень для пацієнта

У багатьох задачах медичної діагностики, зокрема при аналізі ультразвукових досліджень, **один пацієнт може мати кілька зображень**, що відображають орган з різних кутів, положень чи фаз дихального циклу. Для отримання надійного і узагальненого результату важливо правильно поєднати інформацію з усіх наявних знімків. Існує кілька поширених підходів до вирішення цієї задачі.

Середнє значення (mean aggregation)

Найпростішим способом об'єднання результатів є **усереднення** вихідних передбачень, зроблених моделлю для кожного окремого зображення пацієнта. Цей підхід дозволяє зменшити вплив шуму, випадкових артефактів або аномальних зображень і є **обґрунтованим варіантом** при невеликій кількості зображень. Він активно використовується в літературі, зокрема в роботі Вуга et al. [7], де середнє значення оцінювалося як базовий метод агрегування результатів при класифікації стеатозу.

Однак цей метод **не враховує вагомість окремих зображень**, не здатен навчитися виділяти більш інформативні кадри і не враховує кореляції між ними.

Рекурентні нейронні мережі (RNN, GRU, LSTM)

Іншим підходом до об'єднання послідовності даних є використання **рекурентних мереж**. Такі архітектури здатні опрацьовувати серії даних із часовою або логічною послідовністю. У контексті медичних зображень це може бути послідовність МРТ-зрізів або кадри УЗ-відео. Для завдань регресії

або класифікації на основі послідовних зображень нерідко застосовуються **GRU** чи **LSTM**, що дозволяє враховувати залежності між кадрами [11].

Попри потенційну ефективність, цей підхід **потребує великих обсягів даних** і складніший у реалізації. Враховуючи обмежений розмір датасету (55 пацієнтів, по 10 зображень), його застосування в межах даної роботи не є доцільним.

Мережі з багатозображувальним входом (stacked input / multi-channel input)

Ще одним підходом є **подача кількох зображень одночасно на вхід моделі**, шляхом об'єднання їх у багатоканальне зображення (наприклад, 10 УЗ-знімків у вигляді тензора розмірності $[10 \times H \times W]$). У такому випадку мережа може навчитися враховувати просторово-контекстуальну інформацію між зображеннями.

Однак для такого підходу потрібно:

- значно модифікувати архітектуру стандартної CNN;
- мати велику кількість тренувальних прикладів;
- нормалізувати порядок і масштаб зображень у кожній серії.

Через зазначені обмеження він також не був обраний у цій роботі.

Кастомна мережа другого етапу (власний підхід)

У рамках цієї роботи реалізовано **компромісний і ефективний підхід**:

1. На першому етапі кожне з 10 зображень обробляється CNN (DenseNet121), яка передбачає числове значення рівня стеатозу.
2. На другому етапі реалізовано **невелику кастомну нейронну мережу**, яка приймає на вхід ці 10 значень та навчається видавати **узагальнене прогнозне значення** для пацієнта.

Цей підхід має такі переваги:

- не потребує додаткових архітектурних ускладнень основної CNN;
- дозволяє вчитись розрізняти інформативність різних зображень;
- адаптується до невеликих обсягів даних (кілька десятків пацієнтів);

- покращує результати — як видно з експериментів, **MAE знижується до 7.60**, а кореляція Спірмена підвищується до **0.931**.

Таким чином, реалізована двоетапна система поєднує в собі **простоту реалізації та ефективність агрегування**, забезпечуючи високу точність при невеликих обсягах даних, що особливо важливо для медичних задач.

2.7. Текстурний аналіз як альтернатива або доповнення CNN

Хоча згорткові нейронні мережі (CNN) є потужним інструментом для автоматизованого аналізу зображень, у медичних задачах, зокрема при обробці ультразвукових знімків, ефективним може бути і більш традиційний текстурний аналіз. Його можна застосовувати окремо або як доповнення до CNN, що дозволяє покращити результативність моделей, особливо при обмеженій кількості даних.

Методи текстурного аналізу

Одним з найпоширеніших підходів до аналізу текстури зображень є використання GLCM (Gray Level Co-occurrence Matrix) — матриці співзустрічі сірого рівня. Вона відображає, як часто пікселі певних значень яскравості розташовуються поряд один з одним. На основі GLCM розраховуються так звані ознаки Хараліка (Haralick features), які включають:

- контраст (contrast),
- однорідність (homogeneity),
- енергію (energy),
- ентропію (entropy),
- кореляцію (correlation) тощо [8], [15].

Ці ознаки використовуються для кількісного опису текстури, що особливо цінно в УЗ-зображеннях, де межі структур часто розмиті, а форма — неоднозначна.

Застосування у задачах стеатозу

Ультразвукові знімки печінки демонструють виражені текстурні зміни залежно від ступеня жирового ураження. При стеатозі відбувається підвищення ехогенності та зміна дрібнозернистої структури, що добре відображається на текстурних характеристиках.

У роботі Вуга et al. [7] поряд із CNN-аналізом згадуються і методи текстурної оцінки, як частина гібридного підходу. Автори підкреслюють, що текстурні ознаки можуть бути ефективними у задачах ранньої діагностики або в умовах обмеженої кількості навчальних прикладів.

Інше дослідження Han et al. [8] присвячене класифікації стеатозу саме за текстурними ознаками, отриманими з УЗ-знімків. Було показано, що за допомогою методів машинного навчання (SVM, Random Forest) та текстурних дескрипторів можна досягти AUC понад 0.90, що підтверджує їхню практичну цінність.

Комбіновані підходи

На сьогоднішній день активно розвиваються гібридні підходи, які поєднують:

- CNN для глибокої обробки зображення,
- та статистичні ознаки, зокрема Haralick features, як додаткові входи до повнозв'язних шарів нейронної мережі.

Такий підхід дозволяє мережі поєднувати низькорівневу інформацію про текстуру з високорівневою семантикою, що витягується CNN. У рамках цієї роботи такий підхід не було реалізовано, але він розглядався як перспективний напрямок для подальших досліджень, особливо з урахуванням успішних прикладів у літературі [8], [13].

Висновки до розділу 2

У цьому розділі було розглянуто теоретичне підґрунтя застосування штучних нейронних мереж до задачі оцінки рівня стеатозу печінки за ультразвуковими зображеннями. Особливу увагу приділено згортковим нейронним мережам (CNN), які здатні автоматично виокремлювати інформативні просторові ознаки з медичних зображень без необхідності ручного втручання.

Було обґрунтовано вибір архітектури DenseNet121, яка завдяки щільним зв'язкам між шарами забезпечує ефективне поширення градієнтів, повторне використання ознак і зниження кількості параметрів. Це робить її особливо придатною для задач медичної діагностики в умовах обмеженого обсягу навчальних даних. Водночас у розділі було проведено порівняльний аналіз альтернативних архітектур (ResNet, Inception, Inception-ResNet-v2, EfficientNet), що дозволило чітко обґрунтувати доцільність обраного підходу.

У контексті регресійної постановки задачі було проаналізовано різницю між класифікацією та регресією, що є принципово важливим для коректного оцінювання рівня стеатозу у відсотках. Вибір функції втрат L1 Loss та метрик MAE, Spearman, Pearson базувався на їх здатності відображати як абсолютну похибку, так і монотонність та лінійність зв'язку між прогнозами і реальними значеннями.

Окремий акцент зроблено на сучасних підходах до обробки множинних зображень одного пацієнта. Розглянуто різні стратегії: усереднення, рекурентні мережі, багатоканальний вхід, а також запропоновану в даній роботі двоетапну систему, де другий рівень — це кастомна нейронна мережа, яка узагальнює результати зображень у єдиний прогноз. Цей підхід забезпечує покращення точності без ускладнення основної архітектури.

Також розглянуто текстурний аналіз як альтернативний або допоміжний інструмент до CNN. Методи GLCM і Haralick дозволяють витягти додаткові ознаки, особливо інформативні для УЗ-зображень із характерною зернистістю.

У роботах [7] і [8] показано, що комбінування CNN і текстурних ознак може покращити результати, що є перспективним напрямом для подальших досліджень.

Таким чином, розділ 2 сформував повне концептуальне та методологічне підґрунтя для реалізації програмної частини системи автоматизованого визначення рівня стеатозу печінки, що буде докладно розглянуто у наступному розділі.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ СТЕАТОЗУ

3.1. Використані технології

3.1.1. Мова програмування

Основною мовою програмування, обраною для реалізації системи визначення рівня стеатозу печінки за ультразвуковими зображеннями, є Python. Ця мова є однією з найпоширеніших у сфері наукових досліджень, аналізу даних, машинного навчання та біоінформатики завдяки своїй простоті, гнучкості та великій кількості сторонніх бібліотек.

Python забезпечує високу швидкість розробки, читабельність коду та підтримку об'єктно-орієнтованого, процедурного та функціонального підходів до програмування. У поєднанні з великою екосистемою пакетів і фреймворків, Python дозволяє ефективно реалізовувати як прототипи моделей, так і повноцінні програмні продукти.

Особливу перевагу Python має у сфері глибокого навчання завдяки таким бібліотекам, як PyTorch, TensorFlow, Keras, а також засобам для обробки зображень (наприклад, OpenCV, PIL) і роботи з табличними даними (Pandas). У межах даного проєкту Python використовувався як основна платформа для:

- розробки архітектури нейронної мережі;
- завантаження та попередньої обробки медичних зображень;
- виконання навчання та оцінювання моделі;
- побудови графіків і візуалізації результатів;
- збереження навченої моделі для подальшого використання.

Крім того, Python має активну спільноту розробників і дослідників, що спрощує пошук рішень у разі виникнення труднощів і сприяє швидшому розвитку навичок.

3.1.2. Фреймворк для глибокого навчання

У межах реалізації програмної частини проєкту було використано фреймворк PyTorch, що є одним із провідних інструментів у галузі глибокого навчання. PyTorch, розроблений компанією Meta AI (раніше Facebook AI Research), забезпечує інтуїтивно зрозумілий інтерфейс для побудови, тренування та тестування нейронних мереж різної складності.

Основною перевагою PyTorch є динамічне обчислювальне дерево (dynamic computation graph), що дозволяє змінювати архітектуру моделі "на льоту" під час виконання програми. Це особливо корисно для дослідницьких цілей, коли модель потрібно гнучко адаптувати до специфіки даних або завдання.

Крім того, PyTorch надає:

високорівневі абстракції для створення мереж через модуль `torch.nn`;

- зручний механізм обчислення градієнтів за допомогою `autograd`;
- вбудовані засоби оптимізації через модуль `torch.optim`;
- підтримку різних форматів завантаження та збереження моделей (`.pt`, `.pth`);
- інтеграцію з іншими бібліотеками, такими як `Torchvision`, для роботи із зображеннями;
- просту роботу з GPU через `CUDA` (`.to(device)`), що дозволяє легко перемикатися між CPU і GPU.

У проєкті PyTorch використовувався для реалізації нейронної мережі на основі архітектури `DenseNet121`, налаштування функції втрат, оптимізатора та тренувального циклу, а також для перенесення обчислень на графічний процесор. Завдяки цьому вдалося скоротити час навчання та підвищити продуктивність системи.

3.1.3. Бібліотеки для роботи із зображеннями

У процесі побудови системи визначення рівня стеатозу печінки важливу роль відіграє обробка медичних зображень, отриманих за допомогою ультразвукової діагностики. Для реалізації відповідних операцій було використано дві потужні бібліотеки — OpenCV та модуль `torchvision.transforms` з бібліотеки Torchvision.

OpenCV (Open Source Computer Vision Library) — це широко використовувана бібліотека з відкритим кодом, яка підтримує численні алгоритми комп'ютерного зору й обробки зображень. У рамках цього проєкту OpenCV застосовувалась для виконання таких базових операцій:

- зчитування зображень у форматах PNG або JPG;
- зміна розміру зображень до заданої роздільної здатності, необхідної для подачі в нейронну мережу;
- попередня візуалізація зображень у процесі аналізу.

Завдяки ефективній реалізації алгоритмів та хорошій підтримці бібліотеки на різних платформах, OpenCV дозволяє швидко обробляти великі обсяги даних без значного навантаження на систему.

Модуль `torchvision.transforms` входить до складу бібліотеки Torchvision і призначений для підготовки зображень перед подачею в нейронну мережу. У даному проєкті він використовувався для побудови послідовності трансформацій, які автоматично застосовуються до кожного зображення в процесі завантаження. Зокрема:

- перетворення зображення у тензор (`ToTensor()`), що дозволяє працювати з даними в PyTorch;
- нормалізація піксельних значень (`Normalize()`), яка покращує стабільність і швидкість навчання моделі;
- масштабування до стандартних розмірів (`Resize()`, `CenterCrop()`), щоб гарантувати однорідність вхідних даних.

Завдяки цим бібліотекам було забезпечено ефективне попереднє опрацювання даних, необхідне для успішного навчання глибокої моделі.

3.1.4. Попередньо навчені архітектури моделей

Для вирішення завдання регресії рівня стеатозу за ультразвуковими зображеннями було використано попередньо навчену модель DenseNet121, яка входить до складу модуля torchvision.models. Ця архітектура є однією з варіацій глибоких згорткових нейронних мереж, відомих своєю ефективністю в задачах комп'ютерного зору, зокрема класифікації, сегментації та регресії медичних зображень.

Основні характеристики DenseNet121:

- Щільні з'єднання (Dense connections): кожен шар отримує вхідні дані від усіх попередніх шарів, що сприяє повторному використанню ознак і зменшенню кількості параметрів;
- Менше перенавчання: завдяки ефективному передаванню градієнтів між шарами, мережа краще навчається навіть на невеликих датасетах;
- Покращена узагальнювальна здатність: архітектура демонструє високу якість результатів при меншій складності моделі порівняно з іншими відомими архітектурами (наприклад, ResNet).

У рамках цього проєкту базову модель DenseNet121 було модифіковано для задачі регресії: останній повнозв'язний шар, який зазвичай використовується для класифікації, було замінено на шар Linear(1024, 1), що дозволяє отримати єдине числове значення — передбачуваний рівень стеатозу.

Крім того, використання попередньо навчених ваг (на ImageNet) дозволило моделі стартувати з уже сформованими уявленнями про базові візуальні ознаки, що значно прискорило та стабілізувало процес навчання на медичних зображеннях.

3.1.5. Бібліотеки для обробки табличних даних і візуалізації

У рамках реалізації проєкту було використано дві ключові бібліотеки для роботи з табличними даними та візуального аналізу результатів: Pandas та Matplotlib. Вони забезпечують зручність у роботі з анотаціями, метаданими та результатами навчання моделі.

Pandas — потужна бібліотека для обробки та аналізу структурованих даних. У цьому проєкті вона використовувалася для:

- зчитування анотацій у форматі CSV, які містили інформацію про пацієнтів, рівень стеатозу та шляхи до відповідних зображень;
- фільтрації та попередньої перевірки даних на наявність пропущених або некоректних значень;
- групування зображень за пацієнтами для реалізації крос-валідації;
- підрахунку статистичних характеристик вибірки.

Pandas забезпечує гнучке опрацювання даних у вигляді DataFrame, що значно спрощує інтеграцію з іншими бібліотеками, зокрема PyTorch, при створенні користувацьких класів Dataset.

Matplotlib — популярна бібліотека для створення графіків і візуалізацій. У проєкті вона використовувалася для:

- побудови графіків тренувальних та валідаційних втрат (loss) по епохах, що дозволяло візуально оцінювати ефективність навчання;
- візуалізації помилок моделі на тестовому наборі даних;
- створення графіків для демонстрації результатів регресії — порівняння передбачених і фактичних значень рівня стеатозу.

Такі візуалізації є важливими для аналізу поведінки моделі, виявлення перенавчання або недонавчання, а також для обґрунтування ефективності запропонованого підходу.

3.1.6. Використання графічного процесора

Навчання моделей глибокого навчання зазвичай потребує значних обчислювальних ресурсів, особливо коли йдеться про обробку великої кількості зображень або використання складних архітектур. Для пришвидшення цього процесу у проєкті було використано технологію CUDA (Compute Unified Device Architecture), розроблену компанією NVIDIA.

CUDA — це платформа паралельних обчислень, яка дозволяє розробникам використовувати потужність графічних процесорів (GPU) для загальних обчислювальних завдань. Завдяки використанню CUDA, PyTorch автоматично переносить обчислення з центрального процесора (CPU) на графічний, що суттєво:

- скорочує час навчання моделі;
- дозволяє використовувати більші об'єми даних за один крок;
- покращує продуктивність при роботі з глибокими мережами.

У цьому проєкті навчання DenseNet121 відбувалося на комп'ютері з CUDA-сумісною відеокартою. Для активації обчислень на GPU використовувалася стандартна конструкція PyTorch наведена на рис. 3.1:

```
if torch.cuda.is_available():  
    device = "cuda"  
else:  
    device = "cpu"  
print(device)
```

Рисунок 3.1 – Підключення CUDA

Це дозволило виконувати навчання та валідацію моделі значно швидше, ніж у випадку використання лише центрального процесора, що особливо важливо при експериментуванні з архітектурами, гіперпараметрами та різними підходами до обробки зображень.

3.1.7. Середовище розробки

Розробка програмної частини системи визначення рівня стеатозу печінки здійснювалася в інтегрованому середовищі розробки Visual Studio Code (VS Code) — одному з найпопулярніших редакторів коду серед розробників штучного інтелекту та науковців.

VS Code забезпечує широкий спектр можливостей для зручної та ефективної роботи з Python-кодом, зокрема:

- підсвічування синтаксису;
- інтеграцію з Git для контролю версій;
- запуск та налагодження коду безпосередньо з редактора;
- підтримку роботи з віртуальними середовищами (venv, conda);
- розширення для роботи з Jupyter Notebook прямо в середовищі редактора.

У проєкті активно використовувалися функціональні можливості розширення Python for VS Code, а також підтримка Jupyter notebooks, що дозволяло:

- створювати блоки коду для послідовного виконання;
- візуалізувати графіки втрат та передбачень моделі прямо в редакторі;
- інтерактивно змінювати параметри навчання без необхідності запускати всю програму повторно.

Також VS Code забезпечував зручне керування структурою проєкту, модульністю коду, візуалізацією директорій, автоматичне форматування та підказки під час написання коду, що сприяло підвищенню якості розробки та спрощувало експериментування з різними конфігураціями моделі.

3.1.8. Бібліотека Captum для пояснюваного машинного навчання

Для реалізації пояснень роботи моделі та візуалізації важливих областей на зображеннях, у даній роботі була використана бібліотека Captum — інструмент для інтерпретації моделей, що працюють на основі PyTorch.

Одним із найважливіших методів, реалізованих у Captum, є метод інтегральних градієнтів (Integrated Gradients), який дозволяє визначити внесок кожного пікселя вхідного зображення у фінальне передбачення моделі. Цей метод є модифікацією стандартного градієнтного аналізу і забезпечує стабільні та інтерпретовані значення важливості ознак.

Основні переваги використання Captum:

- Пряма інтеграція з PyTorch;
- Підтримка різноманітних методів пояснення (Integrated Gradients, Saliency, DeepLIFT, Shapley Values тощо);
- Зручний API для візуалізації heatmap'ів на зображеннях;
- Підтримка моделей з кількома входами або складними структурами.

У рамках цієї роботи бібліотека Captum використовувалась для:

- побудови мап важливості (saliency maps);
- демонстрації, які ділянки УЗ-зображення найбільше впливають на рішення моделі;
- підтвердження того, що модель справді зосереджується на областях печінки, а не на фонових або артефактних зонах.

Таким чином, Captum дозволила зробити модель прозорішою та наочною для подальшого клінічного аналізу та верифікації.

3.2. Програмна реалізація

3.2.1. Завантаження та обробка вхідних даних

Для розв'язання задачі автоматизованого визначення рівня стеатозу печінки було використано датасет, описаний у науковій публікації Вуга et al. (2018) [7]. Він був зібраний у рамках клінічного дослідження з метою вивчення

можливості оцінки жирової інфільтрації печінки за допомогою ультразвукових зображень. Зібрані дані дозволяють застосовувати алгоритми машинного навчання для кількісного передбачення стеатозу на основі УЗ-зображень без необхідності інвазивних процедур.

Датасет містить 550 ультразвукових зображень, що були отримані від 55 пацієнтів, тобто по 10 зображень на кожного пацієнта. Усі зображення були зроблені в B-mode (режим сірого зображення) за допомогою ультразвукового апарата Siemens Acuson S2000 із використанням конвексного датчика 6C1 HD. Завдяки однаковим умовам отримання зображень датасет є добре стандартизованим, що важливо для тренування нейронних мереж.

Кожне зображення супроводжується відповідним числовим значенням стеатозу, отриманим за результатами біопсії печінки. Таким чином, модель має навчитися передбачати безперервну величину (відсоток жиру в тканині печінки), що робить задачу регресійною.

Як видно на рис. 3.2, більшість пацієнтів мають стеатоз у межах 0–30%, однак також присутні випадки з високим рівнем ураження — до 85%.

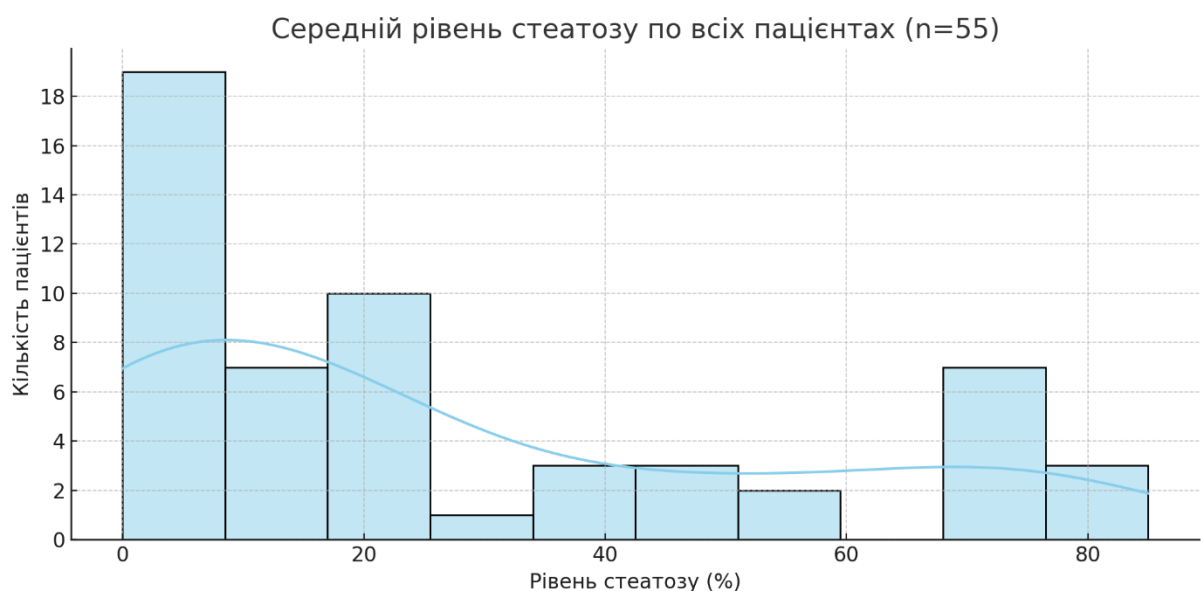


Рисунок 3.2 – Рівень стеатозу по всіх пацієнтах в датасеті

Це створює сприятливі умови для навчання нейронної мережі з гарною узагальнюючою здатністю, оскільки модель бачить приклади із широкого спектру клінічних станів — від здорової до тяжко ураженої печінки.

Ультразвукові зображення в сирому вигляді можуть містити сторонні графічні елементи, які не мають діагностичного значення та можуть погіршити навчання моделі. Серед таких елементів — шкали, мітки часу, значення частоти серця (HR), а також графіки електрокардіограми (ЕКГ).

Щоб уникнути впливу цих артефактів на процес навчання, було застосовано етап попередньої обробки, який передбачав видалення або зафарбовування зайвої службової інформації, що знаходиться за межами клінічно важливої області. При цьому геометрія самої ультразвукової області (fan-shaped) залишалася незмінною, що гарантувало збереження вихідної структури тканини печінки. Приклади сирого та обробленого зображень наведені на рис. 3.3 та рис.3.4 відповідно.

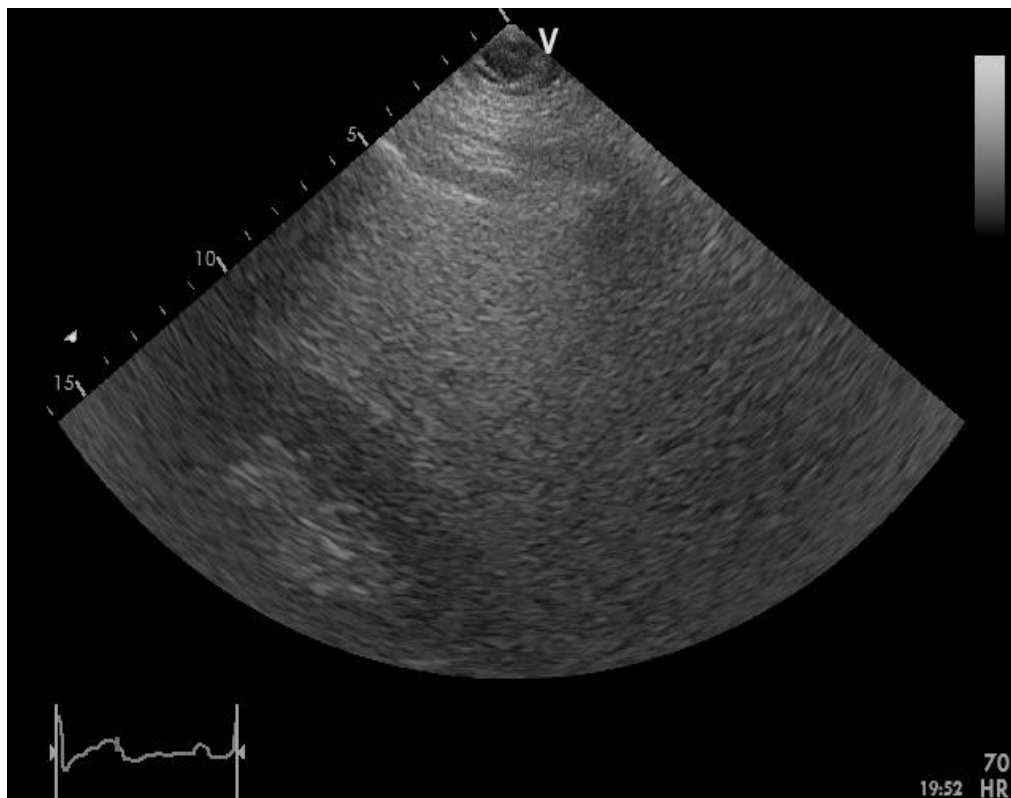


Рисунок 3.3 – Сире УЗ-зображення

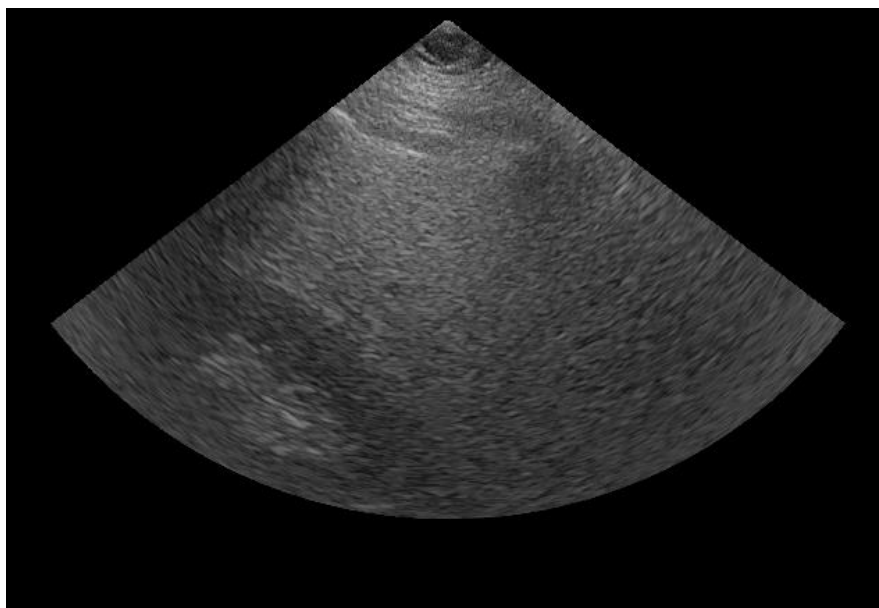


Рисунок 3.4 – Те ж зображення після виділення ROI

Як видно з прикладу, службова інформація, яка не стосується зображення печінки, була усунута або замінена фоном, що дозволяє моделі зосередитися на реальних тканинних структурах при навчанні.

Для роботи з ультразвуковими зображеннями та відповідними мітками рівня стеатозу було реалізовано власний клас `Dataset`, який є розширенням базового інтерфейсу `torch.utils.data.Dataset` з фреймворку `PyTorch`. Такий підхід дозволяє централізовано реалізувати усі необхідні етапи підготовки даних, що критично важливо у задачах глибокого навчання.

Розроблений клас (рис. 3.5) відповідає типовій структурі `PyTorch`-датасетів і включає такі ключові компоненти:

- Ініціалізація: конструктор класу отримує шлях до директорії з даними та файл `annotation.csv`, у якому вказано відповідність між кожним зображенням та його цільовим значенням (рівнем стеатозу). Додатково задається розмір зображень для масштабування (`size`) та обчислювальний пристрій (`device`), на який будуть передаватися тензори.
- Трансформації зображень: усі зображення автоматично масштабуються до заданого розміру за допомогою методу `transforms.Resize`, а також конвертуються у формат тензора за допомогою `transforms.ToTensor()`.

Таким чином, зображення стають сумісними з форматом вхідного шару згорткової нейронної мережі.

- Підготовка тензорів міток: залежно від конфігурації, мітки можуть подаватися як числові значення з плаваючою точкою (float) — для задачі регресії.

```
image_size = (420, 420)

train_dataset = Dataset("data/train", size=image_size, save_in_ram=False)
test_dataset = Dataset("data/test", size=image_size, save_in_ram=False)

train_loader = DataLoader(train_dataset, batch_size=4, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=2, shuffle=True)
```

Рисунок 3.5 – Створення екземплярів нашого класу Dataset для тренувальної і тестової вибірок

Перевага структури класу: дані готуються у вигляді пар (зображення, мітка) і можуть бути передані у стандартний DataLoader з PyTorch для формування міні-батчів, які використовуються під час навчання моделі. Клас також передбачає передачу кожного тензора на GPU або CPU — залежно від конфігурації середовища.

Функціональність класу забезпечує гнучке масштабування, чисту інтеграцію з навчальним циклом моделі, а також дозволяє централізовано контролювати підготовку вхідних даних.

Для забезпечення об'єктивної оцінки якості навченої моделі було реалізовано розподіл даних на тренувальну та тестову вибірки. Згідно з загальноприйнятою практикою, було обрано співвідношення 80:20, яке дозволяє виділити достатньо даних для тренування, зберігаючи при цьому репрезентативну тестову підмножину для валідації.

Оскільки вхідні дані згруповані по пацієнтах, розбиття проводилося на рівні пацієнтів, а не окремих зображень. Це важливо для виключення витоку

даних — модель не повинна бачити навіть частини зображень пацієнтів, які входять до тестової вибірки.

- 44 пацієнти (80%) було включено до тренувальної вибірки;
- 11 пацієнтів (20%) — до тестової вибірки.

Такий підхід гарантує, що модель оцінюється на раніше не бачених пацієнтах, а отже, отримані метрики точності (наприклад, середня абсолютна похибка) мають високу узагальнювальну силу.

3.2.2. Реалізація архітектури мережі

У даній роботі для вирішення задачі регресійного передбачення рівня стеатозу печінки було використано згорткову нейронну мережу DenseNet121, доступну в бібліотеці `torchvision.models`. Ця архітектура була обрана завдяки своїй ефективності при роботі з медичними зображеннями, особливо на невеликих датасетах, а також через здатність до повторного використання ознак (feature reuse), що зменшує ймовірність перенавчання.

DenseNet121 (Densely Connected Convolutional Network) є глибокою нейронною мережею, побудованою на принципі щільного з'єднання шарів: кожен шар отримує на вхід не лише вихід попереднього шару, а й усі виходи попередніх шарів у межах блоку.

Така структура:

- покращує поширення градієнтів;
- сприяє повторному використанню ознак;
- зменшує кількість параметрів порівняно з іншими глибокими архітектурами.

У стандартній реалізації DenseNet121 останній шар (classifier) відповідає за класифікацію 1000 класів (ImageNet). У даному проєкті ця частина була замінена на один повнозв'язний шар (`nn.Linear`) із виходом розмірності 1, що відповідає задачі одновимірної регресії — передбачення рівня стеатозу у відсотках. Визначення об'єкту моделі наведено на рис. 3.6.

```
import torchvision
model = torchvision.models.densenet121()
model.classifier = torch.nn.Linear(1024, 1)
```

Рисунок 3.6 – Створення класу моделі

3.2.3. Навчання моделі та оцінка якості

У процесі навчання моделі було підібрано відповідні гіперпараметри (рис. 3.7), які забезпечували стабільну та ефективну роботу мережі. В якості функції втрат було обрано `L1Loss()` — середню абсолютну похибку, яка обчислює середнє відхилення між передбаченим і реальним значенням рівня стеатозу.

```
loss_fn = nn.L1Loss()
optimizer = torch.optim.AdamW(model.parameters(), lr=1e-4)
```

Рисунок 3.7 – Визначення гіперпараметрів навчання

Такий вибір обґрунтований тим, що `L1Loss` є менш чутливою до викидів порівняно з середньоквадратичною похибкою (`MSE`) і тому краще підходить для медичних задач, де важливо мінімізувати саме середнє відхилення у фізично вимірюваних величинах.

Для оптимізації ваг мережі застосовувався оптимізатор `AdamW` з бібліотеки `PyTorch` (`torch.optim.AdamW`), який є вдосконаленою версією класичного алгоритму `Adam`. Його особливість полягає в окремому регуляризаторі `weight decay`, що дозволяє точніше контролювати загальну складність моделі та зменшує ризик перенавчання.

Початкова швидкість навчання була встановлена на рівні `0.0001` (`1e-4`), що відповідає типовим рекомендаціям для регресійних задач на невеликих

датасетах. Навчання виконувалося з розміром батчу 4 для тренувальної вибірки та 2 — для тестової, що забезпечувало прийнятний компроміс між стабільністю градієнтів і обмеженнями графічного процесора.

Загальна кількість епох навчання складала 10, однак реалізована система дозволяє за необхідності продовжити тренування, наприклад, у випадку виявлення подальшого зниження тестової похибки або застосування методів ранньої зупинки.

Процес тренування моделі організовано у вигляді епохального циклу, в межах якого виконуються дві основні фази — навчання (train, рис. 3.8) та тестування (test, рис. 3.9). Це дозволяє контролювати як динаміку зниження функції втрат на тренувальній вибірці, так і загальну здатність моделі до узагальнення.

```
def train(dataloader, model, loss_fn, optimizer):  
    model.train()  
    train_loss = 0  
    for x, y in tqdm(dataloader, total=len(dataloader)):  
        pred = model(x)  
        loss = loss_fn(pred, y)  
        loss.backward()  
        optimizer.step()  
        optimizer.zero_grad()  
        train_loss += loss.item()  
    train_loss /= len(dataloader)  
    return train_loss
```

Рисунок 3.8 – Функція тренування

```
def test(dataloader, model, loss_fn, epoch):  
    model.eval()  
    test_loss = 0  
    with torch.no_grad():  
        for x, y in tqdm(dataloader, total=len(dataloader)):  
            pred = model(x)  
            test_loss += loss_fn(pred, y).item()  
    test_loss /= len(dataloader)  
    return test_loss
```

Рисунок 3.9 – Функція тестування

У першій фазі (`model.train()`), модель переходить у режим навчання, у якому активуються механізми оновлення ваг (наприклад, Dropout або BatchNorm поведуться відповідно до тренувального режиму).

Далі дані поетапно подаються в модель у вигляді батчів, і для кожного батчу:

- обчислюються передбачення моделі;
- обчислюється похибка за допомогою функції втрат (`loss_fn(pred, y)`), яка порівнює передбачене значення з реальним;
- обчислюється градієнт втрат за допомогою `loss.backward()`;
- виконується оновлення ваг мережі методом `optimizer.step()`;
- градієнти скидаються (`optimizer.zero_grad()`) перед наступною ітерацією, щоб уникнути накопичення.

Загальна втрата по епосі нормується на кількість батчів, що дозволяє відстежувати середню втрату на етапі навчання. Основний цикл тренування наведено на рис. 3.10.

```
epochs = 10
for i in range(epochs):
    print("Epoch:", epoch+1)
    train_loss = train(train_loader, model, loss_fn, optimizer)
    test_loss = test(test_loader, model, loss_fn, epoch)
    train_losses.append(train_loss)
    test_losses.append(test_loss)
    if test_loss < min_test_loss:
        print("Saving model...")
        min_test_loss = test_loss
        best_model = model.state_dict().copy()
    print("train_loss:", train_loss)
    print("test_loss:", test_loss)
    epoch += 1
print("Done!")
```

Рисунок 3.10 – Основний цикл тренування і тестування

Після кожної епохи модель переходить у режим оцінки (`model.eval()`), що відключає стохастичні елементи та забезпечує стабільність передбачень. У цій фазі також вмикається контекст `torch.no_grad()`, який блокує обчислення градієнтів і знижує споживання пам'яті. Модель проходить ті ж батчі з тестової вибірки, проте без оновлення ваг. Для кожного прикладу обчислюється похибка, а в кінці епохи виводиться середнє значення тестової втрати, що служить головним критерієм для оцінки ефективності.

Таким чином, описаний цикл навчання дозволяє здійснювати поетапне вдосконалення моделі, контролюючи її здатність до генералізації на нових даних.

Для уникнення перенавчання та забезпечення збереження найоптимальніших параметрів нейронної мережі в процесі тренування було реалізовано механізм контролю якості на тестовій вибірці. Після завершення кожної епохи виконувалося порівняння поточної тестової втрати із найменшим зафіксованим її значенням (визначення на рис. 3.11). У разі покращення результату (тобто зменшення похибки), модель вважалася кращою, і її ваги зберігалися в оперативній пам'яті:

```
if test_loss < min_test_loss:
    print("Saving model...")
    min_test_loss = test_loss
    best_model = model.state_dict().copy()
```

Рисунок 3.11 – Визначення найкращої моделі

Таким чином, у змінній `best_model` зберігалася найкраща версія моделі, яка демонструвала найнижчу середню абсолютну похибку на тестовій вибірці протягом усього процесу навчання. Цей підхід забезпечує більш надійну генералізацію моделі та дозволяє уникнути ситуацій, коли пізні епохи дають менший `train-loss`, але гірші результати на `unseen-даних`.

Для моніторингу ефективності тренування моделі протягом епох було реалізовано візуалізацію графіків втрат (loss), наведену на рис. 3.12, для обох фаз: тренувальної та тестової. Це дозволяє оперативно виявляти проблеми в навчанні, наприклад: нестабільність градієнтів, перенавчання (overfitting), або досягнення плато.

```
# plot train and test loss
plt.plot(train_losses, label="train_loss")
plt.plot(test_losses, label="test_loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()
plt.title("Train and Test Loss")
plt.show()
```

Рисунок 3.12 – Код візуалізації тренувальної і тестової історії

На графіку (рис. 3.13) зображено динаміку зміни середньої абсолютної похибки (MAE) на тренувальних та тестових вибірках впродовж 20 епох:

- З аналізу графіка можна зробити кілька важливих висновків:
- Тренувальна похибка монотонно зменшується, що свідчить про поступове покращення здатності моделі відтворювати залежності на відомих даних.
- Тестова похибка коливається, зберігаючи певну тенденцію до зниження, однак демонструє флуктуації, що може бути пов'язано з обмеженим обсягом тестової вибірки.
- Значення тренувальної та тестової втрат не розходяться критично, отже модель не переобучена — вона не просто запам'ятовує тренувальні приклади, а демонструє здатність до узагальнення.

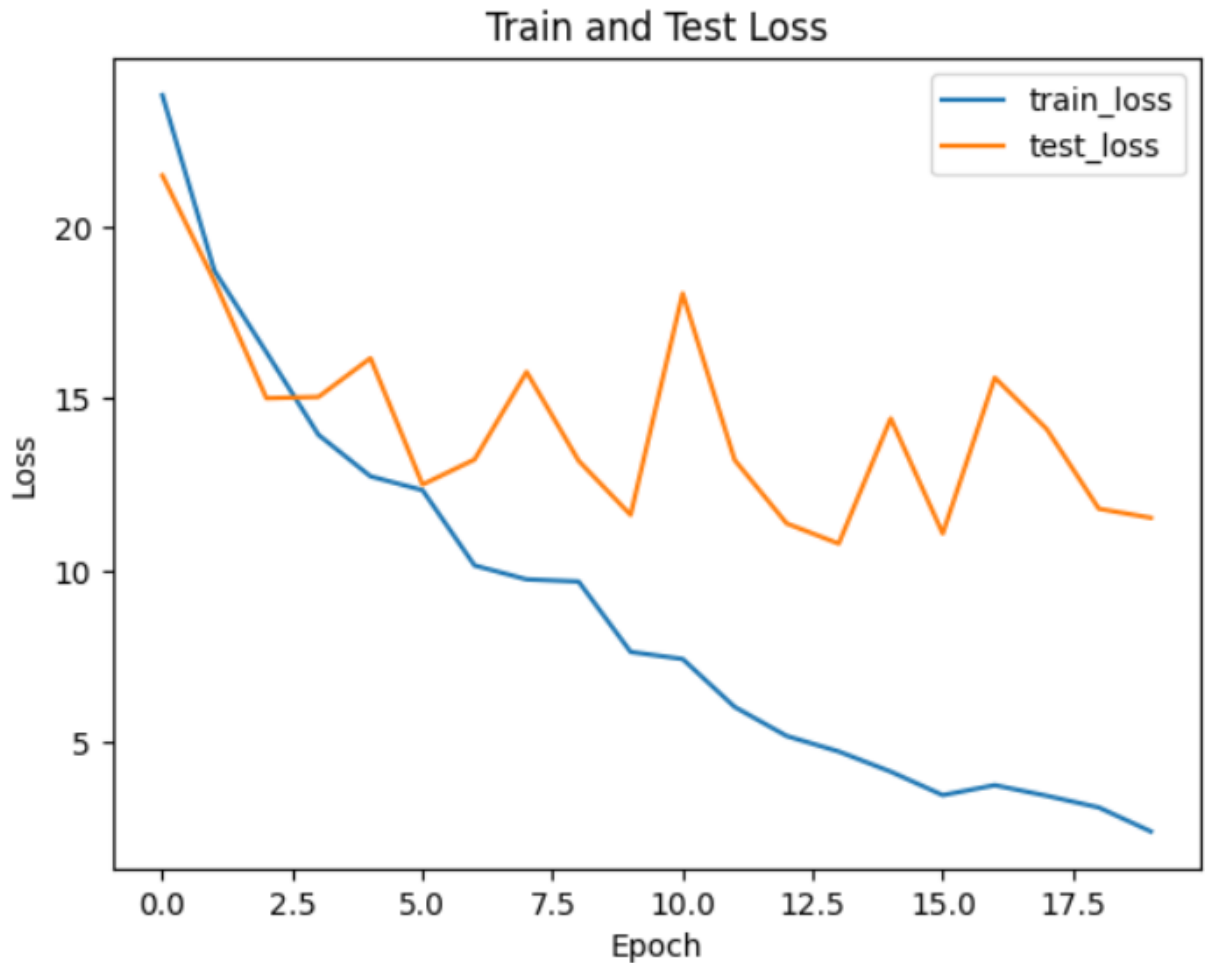


Рисунок 3.13 – Тренувальна та тестова історії

Таким чином, графік підтверджує загальну стабільність процесу навчання та ефективність вибраної архітектури й гіперпараметрів.

Основною метрикою в процесі навчання використовувалася середня абсолютна похибка (MAE), яка є зручною в медичному контексті, оскільки відображає середнє абсолютне відхилення між передбаченим та фактичним рівнем стеатозу у відсотках. Це дозволяє безпосередньо інтерпретувати помилку в клінічно значущих одиницях.

Крім того, після завершення тренування було проведено оцінку кореляції Спірмана (реалізація наведена на рис. 3.14) між передбаченими значеннями та реальними результатами біопсії. Цей показник дозволяє оцінити, наскільки модель зберігає відносне ранжування пацієнтів за ступенем стеатозу — тобто, чи коректно вона визначає, у кого показник вищий, а в кого — нижчий. Для

оцінки було зібрано усі справжні мітки та передбачення на тестовій вибірці, після чого обчислено коефіцієнт Спірмана:

```
test_targets = [i[1].item() for i in test_dataset]

model_outputs = []
for x, y in tqdm(test_dataset):
    with torch.no_grad():
        pred = model(x.unsqueeze(0).to(device))
        model_outputs.append(pred[0][0].item())

spearman = spearmanr(test_targets, model_outputs)
print("Spearman correlation coefficient:", spearman.statistic)
print("Spearman p-value:", spearman.pvalue)
```

Рисунок 3.14 – Перевірка кореляції Спірмана тестової вибірки для нашої моделі

Результат оцінки (рис. 3.15) склав $\rho = 0.901$ при р-значенні 4.19×10^{-81} , що свідчить про високу статистичну значущість та сильну монотонну залежність між передбаченнями моделі та біопсійними результатами.

100%  220/220 [00:04<00:00, 48.29it/s]

```
Spearman correlation coefficient: 0.9011939823075421
Spearman p-value: 4.189972588423631e-81
```

Рисунок 3.15 – Значення кореляції Спірмана для наших тестових даних

Для порівняння: у роботі [7], на яку орієнтувалася дана дипломна робота, значення коефіцієнта Спірмена для згорткової нейронної мережі становило 0.78, а для класичного гепаторенального індексу (HI) — 0.80 [7]. Таким чином, отримані результати демонструють вищу кореляцію, ніж у попередніх дослідженнях, що може бути зумовлено вдосконаленням структури навчання,

використанням новіших реалізацій моделей або оптимізацією попередньої обробки вхідних даних.

3.2.4. Візуалізація важливих областей зображення

Для підвищення інтерпретованості результатів та перевірки того, чи модель дійсно спирається на клінічно релевантні структури під час передбачення рівня стеатозу, було застосовано метод візуалізації інтегрованих градієнтів (Integrated Gradients). Цей метод дозволяє оцінити внесок кожного пікселя у фінальний прогноз моделі, отже, виділяє ділянки зображення, які найбільше впливають на прийняте рішення (рис. 3.16 – рис.3.17).

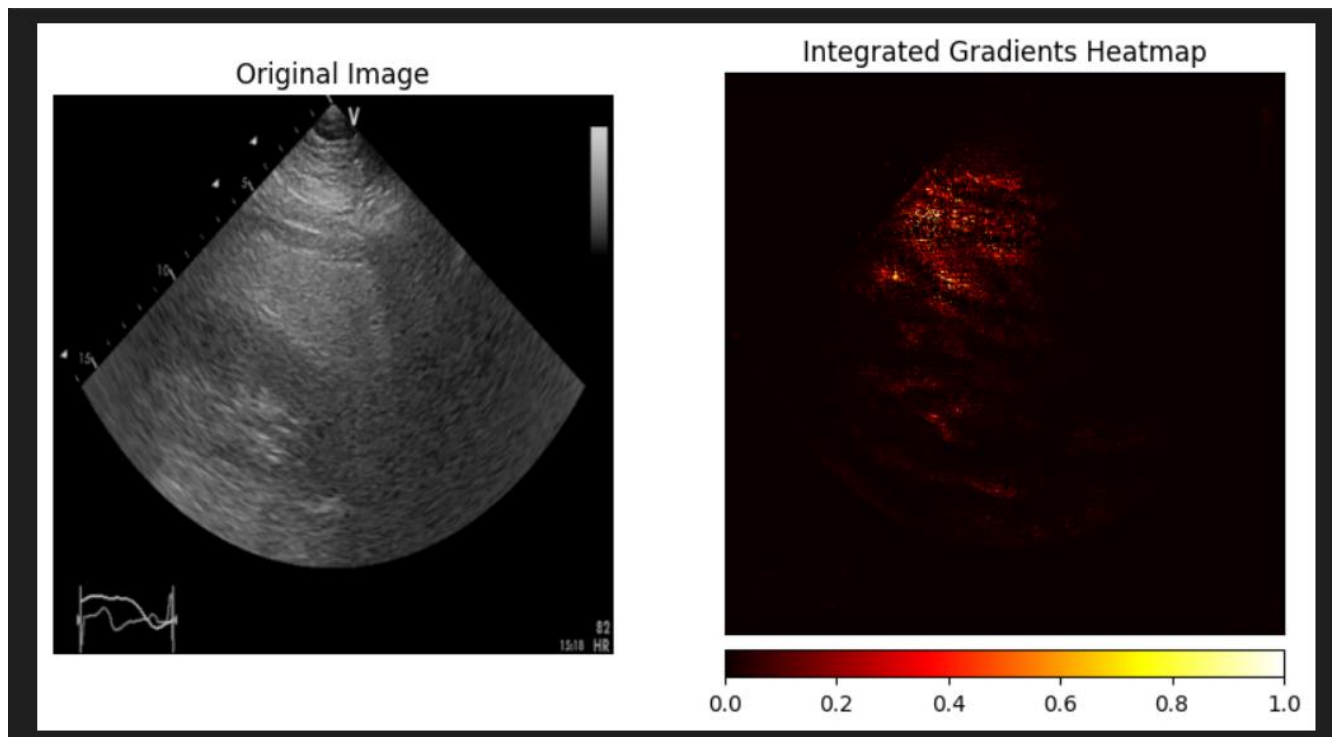


Рисунок 3.16 – Оригінальне вхідне зображення та теплова карта важливості пікселів вхідного зображення отримана методом інтегрованих градієнтів

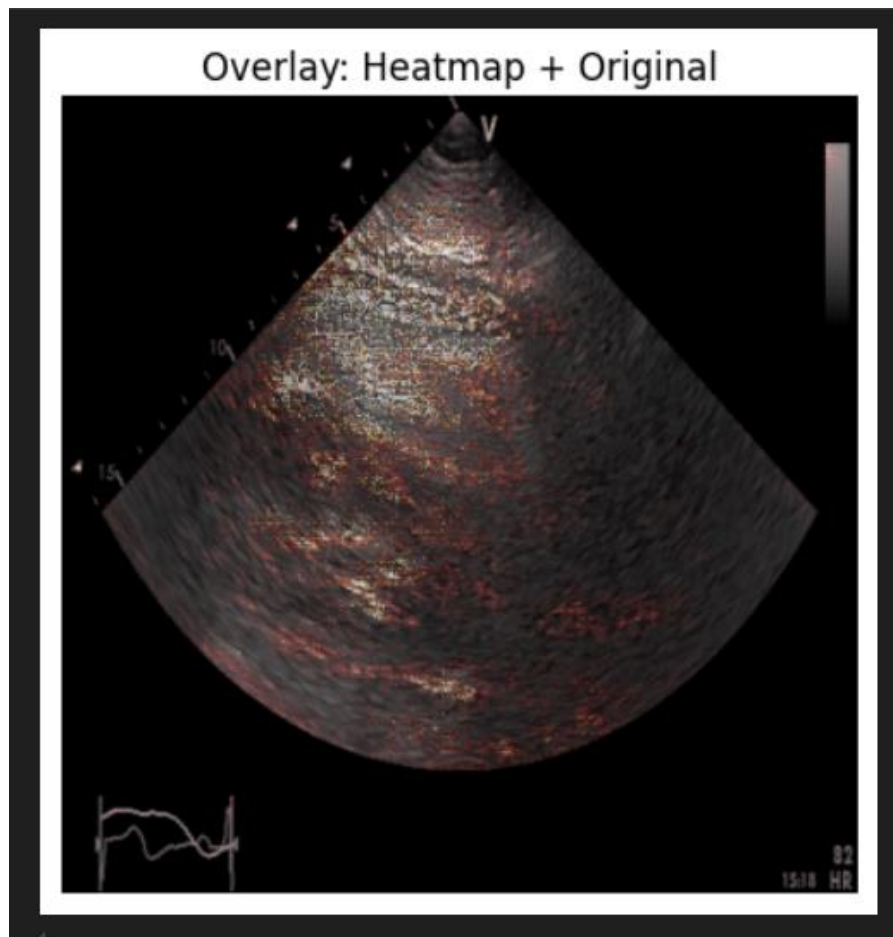


Рисунок 3.17 - Накладення цієї теплової карти поверх оригінального зображення

Таке накладення дозволяє легко інтерпретувати, на які ділянки зображення модель орієнтується при прогнозуванні рівня стеатозу. Як видно з прикладу, найбільший внесок у прийняття рішення мають центральні зони печінки, що узгоджується з клінічними уявленнями про ділянки для оцінки стеатозу на УЗ.

3.2.5. Реалізація другого етапу прогнозування на основі 10 зображень

У рамках реалізації запропонованої системи діагностики рівня стеатозу печінки було впроваджено двоетапну архітектуру обробки даних. Перший етап передбачає отримання окремих прогнозів рівня стеатозу для кожного з 10 ультразвукових зображень пацієнта за допомогою модифікованої згорткової нейронної мережі DenseNet121. Проте для практичного застосування

необхідно сформулювати єдине підсумкове значення для кожного пацієнта на основі множини часткових передбачень.

З метою агрегування результатів першого етапу було реалізовано другий етап системи — спеціалізовану повнозв'язну нейронну мережу, яка приймає на вхід вектор з десяти числових значень (прогнозів моделі першого рівня) та здійснює регресійне передбачення інтегрального рівня стеатозу.

Архітектура другоетапної моделі складається з вхідного шару розмірністю 10, прихованого шару з 1024 нейронами з активацією GELU, та вихідного шару з одним нейроном, який формує фінальний регресійний прогноз. Реалізацію моделі подано нижче (рис. 3.18):

```
class Stage2Model(nn.Module):
    def __init__(self, embed_size=1024):
        super(Stage2Model, self).__init__()
        self.fc1 = nn.Linear(10, embed_size)
        self.last = nn.Linear(embed_size, 1)
        self.activation = nn.GELU()
    def forward(self, x):
        x = self.activation(self.fc1(x))
        x = self.last(x)
        return x
```

Рисунок 3.18 – Архітектура мережі другого рівня

Особливістю запропонованого підходу є інваріантність моделі до порядку елементів вхідного вектора, що було досягнуто шляхом навчання мережі на випадково перемішаних масивах з 10 передбачень. Такий підхід дозволив не лише знизити ризик переобучення на специфічному порядку вхідних даних, але й штучно розширити навчальну вибірку за рахунок генерації нових комбінацій входів для кожного пацієнта. Враховуючи обмежений розмір датасету (55 пацієнтів), це стало одним із ключових факторів підвищення узагальнювальної здатності моделі.

Навчання моделі другого етапу здійснювалося на рівні пацієнтів, де кожному входу з 10 прогнозованих значень відповідало одне еталонне значення рівня стеатозу. Було використано функцію втрат L1Loss (MAE) та оптимізатор Adam, з навчанням на 44 пацієнтах (тому ж тренувальному наборі що й для першого рівня) протягом 10 епох.

У порівнянні з альтернативними підходами, результати, досягнуті другоетапною моделлю, засвідчили її ефективність. Наведена нижче таблиця демонструє порівняльні характеристики трьох підходів (табл. 3.1):

Таблиця 3.1

Порівняння результатів трьох варіантів отримання рівня стеатозу

Підхід	MAE	Spearman	Pearson
Один знімок	8.70	0.916	0.914
Усереднення 10 передбачень	8.58	0.922	0.919
Модель другого етапу	7.60	0.931	0.921

Таким чином, запропонована модель другого етапу дозволяє суттєво покращити якість прогнозування шляхом обробки серії часткових оцінок. Її використання забезпечує гнучкість, масштабованість та адаптивність системи, зберігаючи при цьому низьку обчислювальну складність і високу інтерпретованість результатів.

3.2.6. Реалізація графічного інтерфейсу користувача

З метою підвищення зручності використання програмного засобу, розробленого для автоматизованого визначення рівня стеатозу печінки, було реалізовано графічний інтерфейс користувача (GUI). Реалізація здійснена за допомогою бібліотеки PyQt5, яка є інструментом для створення кросплатформених десктопних застосунків на мові програмування Python.

Інтерфейс орієнтований на максимально просту взаємодію користувача з системою та не потребує спеціальних технічних знань для роботи. Користувачу достатньо завантажити одне або декілька зображень для отримання результату оцінки стеатозу.

Інтерфейс реалізовано у вигляді віконної системи з головним меню (рис. 3.19), в якому передбачено вибір одного з двох режимів:

- Оцінка за одним зображенням (рис. 3.20) — дозволяє завантажити окреме ультразвукове зображення печінки та отримати відповідний прогноз рівня стеатозу;
- Оцінка за серією з 10 зображень (рис. 3.21) — призначена для обробки повного набору знімків одного пацієнта, відповідно до структури навчального датасету, і дозволяє отримати фінальний прогноз за допомогою моделі другого етапу.

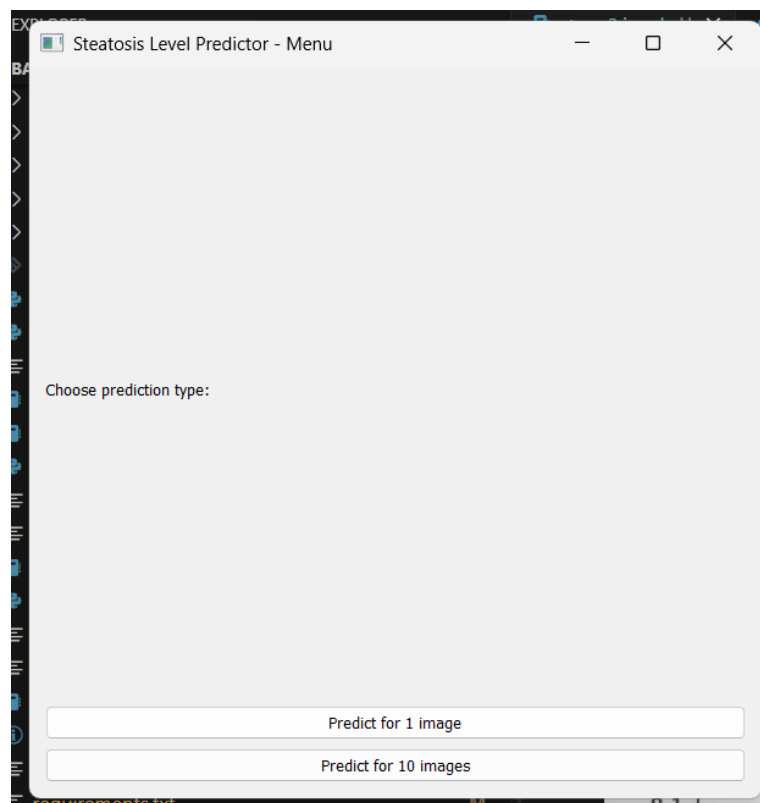


Рисунок 3.19 – Головне меню програмного застосунку

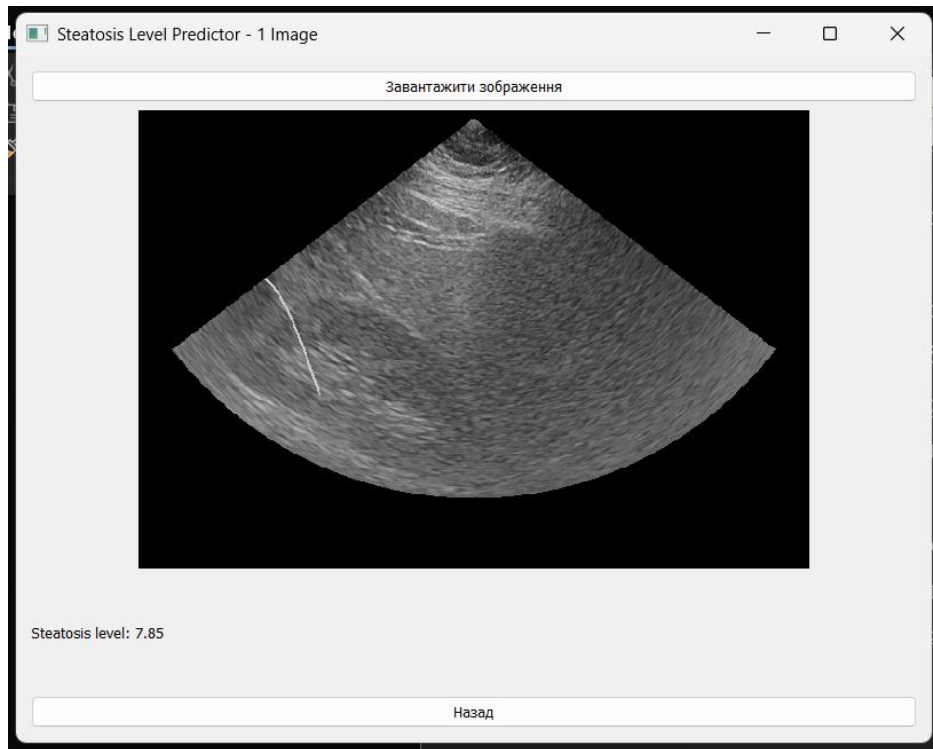


Рисунок 3.20 – Вікно оцінки за одним зображенням

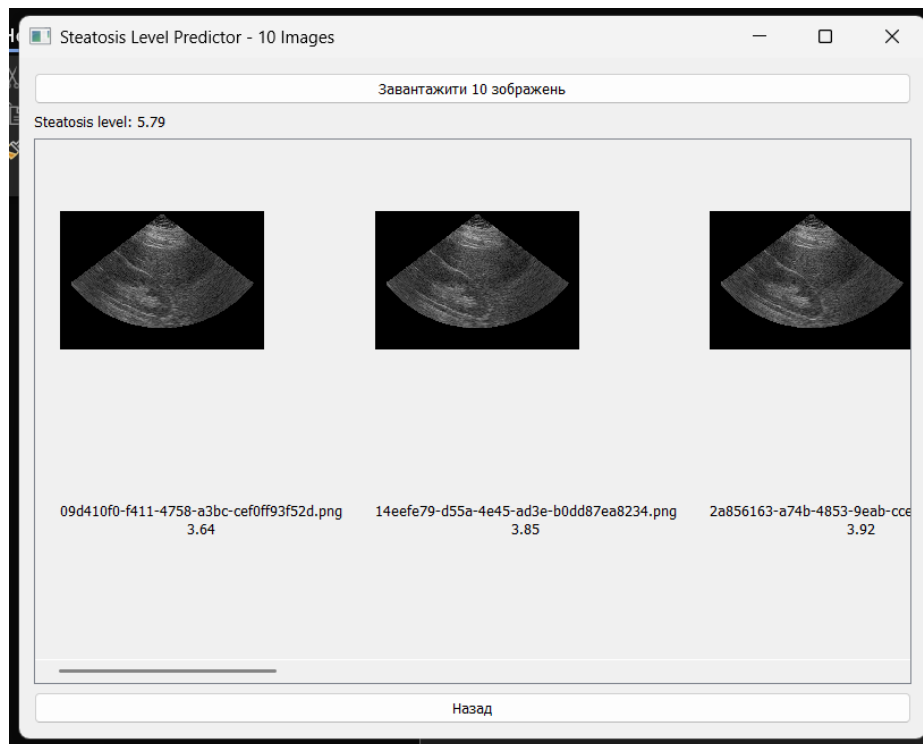


Рисунок 3.21 – Вікно оцінки за 10-ма зображеннями

У першому режимі після завантаження зображення на екрані відображається саме зображення та обчислене значення стеатозу. У другому

режимі система виводить мініатюри завантажених зображень, індивідуальні передбачення для кожного з них, а також узагальнене значення, отримане за допомогою моделі Stage2Model.

3.3 Розрахунок економічного ефекту

В результаті проведеного функціонально-вартісного аналізу розробки та впровадження нашого застосунку, що було розроблено в рамках цієї роботи, сформовано висновки стосовно ефективності як обраних технологій так і стратегій економічних рішень.

Головна функція F_0 – розробка програмного продукту, що забезпечує автоматичне визначення ступеня стеатозу печінки за ультразвуковими зображеннями шляхом використання нейронної мережі. Беручи за основу цю головну функцію, можна виділити наступні підфункції:

- F_1 – вибір способу обробки вхідного зображення (preprocessing);
- F_2 – вибір фреймворку для розробки нейронної мережі;
- F_3 – вибір типу інтерфейсу для взаємодії з користувачем.

Кожна з цих функцій має кілька варіантів реалізації:

Функція F_1 :

- a) OpenCV
- b) TorchVision
- c) PIL (Python Imaging Library)

Функція F_2 :

- a) PyTorch
- b) TensorFlow
- c) Keras

Функція F_3 :

- a) PyQt5
- b) Tkinter

с) Веб-інтерфейс на Flask

Варіанти реалізації основних функцій наведено у морфологічній карті системи (рисунок 3.22).

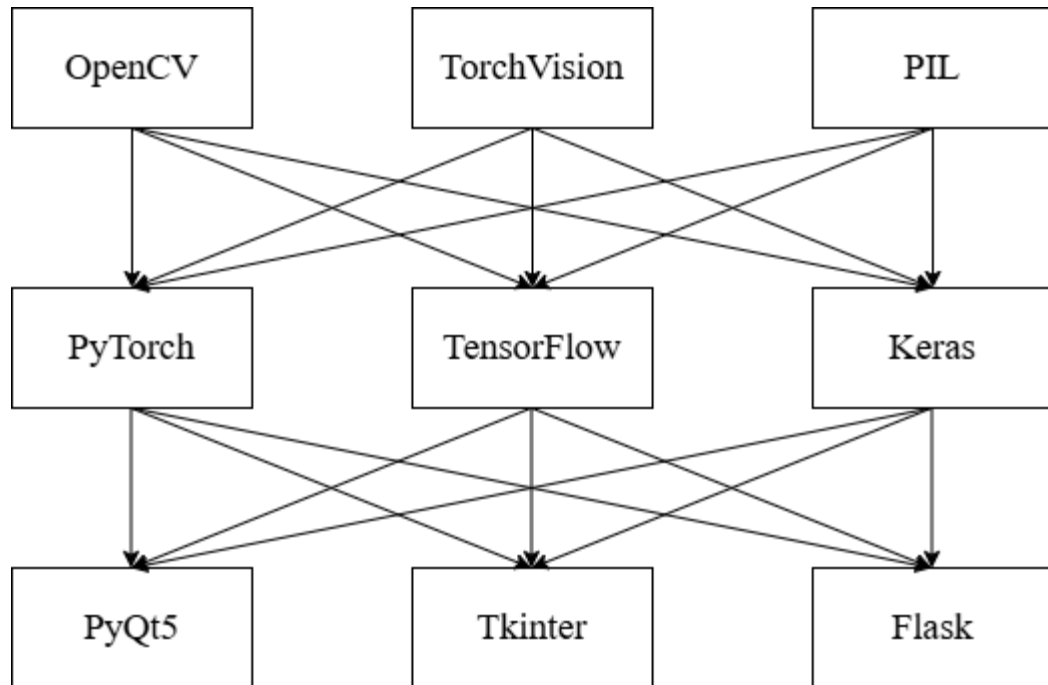


Рисунок 3.22 -- Морфологічна карта

Серед усіх можливих варіантів було обрано найкращий, виходячи з коефіцієнту техніко-економічного рівня, а саме $6.24 \cdot 10^{-5}$, адже якраз він вказує на ефективність. Вартість даного варіанту реалізації становить 312 478 грн. Таким чином ми маємо такі опції:

- TorchVision
- PyTorch
- PyQt5

Висновок до розділу 3

У цьому розділі було представлено повну програмну реалізацію системи визначення рівня стеатозу печінки за ультразвуковими зображеннями із застосуванням методів глибокого навчання. Основу реалізації склала

архітектура згорткової нейронної мережі DenseNet121, адаптована для задачі регресії шляхом модифікації вихідного шару. У якості інструментального середовища обрано фреймворк PyTorch, що забезпечив необхідну гнучкість у побудові та навчанні моделей, а також ефективну обробку медичних зображень.

Було реалізовано повний цикл обробки вхідних даних: від завантаження та попередньої обробки зображень до формування тензорів і їх подачі в модель. Для цього створено власний клас Dataset, який дозволив централізовано контролювати підготовку зображень та формувати батчі для тренування і тестування. Проведене навчання моделі супроводжувалося візуалізацією функції втрат та метрик, що дозволило оперативно контролювати якість результатів.

З метою підвищення точності прогнозування було реалізовано двоетапну систему. На першому етапі DenseNet121 формувала оцінки стеатозу для кожного з 10 зображень пацієнта. Другий етап передбачав агрегування цих результатів за допомогою окремої кастомної повнозв'язної нейронної мережі, яка навчалася на векторах із 10 передбачень. Особливістю підходу стало використання випадкових перестановок елементів вхідного вектора при тренуванні, що забезпечило емпіричну інваріантність до порядку входів та дозволило штучно збільшити розмір тренувальної вибірки. У результаті модель другого рівня досягла найвищої точності серед усіх протестованих підходів: $MAE = 7.60$, $\rho = 0.931$, $r = 0.921$.

Для підвищення прозорості прийняття рішень було реалізовано метод інтегрованих градієнтів за допомогою бібліотеки Captum, який дозволив візуалізувати важливі області зображення, що вплинули на прогноз. Побудовані теплові карти показали, що модель фокусується на клінічно значущих ділянках, що підтверджує релевантність внутрішніх представлень моделі.

З метою зручності використання системи у прикладних сценаріях було розроблено графічний інтерфейс користувача на базі PyQt5, який реалізує два

режими роботи — для одного або десяти зображень. Інтерфейс забезпечує просту взаємодію з системою, відображення результатів прогнозу та можливість перегляду відповідних зображень.

Отримані результати свідчать про ефективність запропонованого двоетапного підходу, заснованого на згорткових нейронних мережах і повнозв'язній моделі для агрегування прогнозів, а також про можливість практичного застосування створеної системи в умовах реальної клінічної практики завдяки високій точності, інтерпретованості та зручності використання.

ЗАГАЛЬНІ ВИСНОВКИ

У межах дипломної роботи було розроблено програмну систему для автоматизованого визначення рівня стеатозу печінки за ультразвуковими зображеннями із застосуванням методів глибокого навчання. Запропоноване рішення базується на двоетапній архітектурі, що поєднує згорткову нейронну мережу DenseNet121, яка здійснює оцінку рівня стеатозу на основі окремих зображень, та повнозв'язну нейронну мережу другого етапу, яка узагальнює передбачення для кожного пацієнта на основі серії з десяти знімків.

У процесі реалізації було забезпечено повний цикл обробки медичних зображень: від етапу попередньої обробки й трансформації вхідних даних до безпосереднього моделювання та інтерпретації результатів. Особливістю моделі другого рівня є її інваріантність до порядку вхідних даних, досягнута шляхом навчання на випадкових перестановках, що дозволило не лише підвищити стійкість системи, а й ефективно розширити тренувальну вибірку.

Отримані результати підтверджують високу точність розробленої системи: середня абсолютна похибка (MAE) становила 7.60, коефіцієнт кореляції Спірмана — 0.931, а коефіцієнт Пірсона — 0.921. Для підвищення інтерпретованості результатів було застосовано метод інтегральних градієнтів, що дозволив візуалізувати найбільш значущі ділянки зображення, на які орієнтується модель під час прогнозування.

Крім того, розроблено графічний інтерфейс користувача на базі PyQt5, який підтримує два режими роботи: прогнозування на основі одного зображення або серії з десяти. Це забезпечує зручність практичного застосування розробленої системи та свідчить про її придатність для подальшого впровадження в клінічні або дослідницькі середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Younossi Z.M., Koenig A.B., Abdelatif D., Fazel Y., Henry L., Wymer M. Global epidemiology of nonalcoholic fatty liver disease—Meta-analytic assessment of prevalence, incidence, and outcomes. *Hepatology*. 2016;64(1):73–84. DOI: 10.1002/hep.28431.
2. Cohen J.C., Horton J.D., Hobbs H.H. Human fatty liver disease: old questions and new insights. *Science*. 2011;332(6037):1519–1523. DOI: 10.1126/science.1204265.
3. Targher G., Day C.P., Bonora E. Risk of cardiovascular disease in patients with nonalcoholic fatty liver disease. *New England Journal of Medicine*. 2010;363(14):1341–1350. DOI: 10.1056/NEJMra0912063.
4. Anderson E.L., Howe L.D., Jones H.E., Higgins J.P.T., Lawlor D.A., Fraser A. The Prevalence of Non-Alcoholic Fatty Liver Disease in Children and Adolescents: A Systematic Review and Meta-Analysis. *PLOS ONE*. 2015;10(10):e0140908. DOI: 10.1371/journal.pone.0140908.
5. Hernaez R., Lazo M., Bonekamp S., Kamel I., Brancati F.L., Guallar E., Clark J.M. Diagnostic accuracy and reliability of ultrasonography for the detection of fatty liver: a meta-analysis. *Hepatology*. 2011;54(3):1082–1090. DOI: 10.1002/hep.24452. Google Scholar+1 Clinical Radiology Online+1
6. Dasarathy S., Dasarathy J., Khiyami A., Joseph R., Lopez R., McCullough A.J. Validity of real time ultrasound in the diagnosis of hepatic steatosis: a prospective study. *Journal of Hepatology*. 2009;51(6):1061–1067. DOI: 10.1016/j.jhep.2009.09.001. Google Scholar+1 ResearchGate+1
7. Byra M., Styczynski G., Szmigielski C., Kalinowski P., Paluszkiewicz R., Ziarkiewicz-Wroblewska B., Wojcicki M., Ziobrowski H. Transfer learning with deep convolutional neural network for liver steatosis assessment in ultrasound images. *International Journal of Computer Assisted Radiology and Surgery*. 2018;13(12):1895–1903. DOI: 10.1007/s11548-018-1843-2.

8. Han A., Yoon J.H., Kim M.J., Kim E.K., Moon H.J., Kwak J.Y. Texture analysis of ultrasound images for liver steatosis classification using machine learning. *BMC Medical Imaging*. 2021;21(1):125. DOI: 10.1186/s12880-021-00638-7.
9. Huang G., Liu Z., Van Der Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017:4700–4708. DOI: 10.1109/CVPR.2017.243.
10. Shorten C., Khoshgoftaar T.M. A survey on image data augmentation for deep learning. *Journal of Big Data*. 2019;6(1):60. DOI: 10.1186/s40537-019-0197-0. Directory of Open Access Journals – DOAJ+1SpringerOpen+1
11. Wang K., Lu X., Zhou H., Gao Y., Zheng J., Tong M., Zhang Y. Deep learning for image-based liver analysis: a comprehensive review. *Computerized Medical Imaging and Graphics*. 2022;93:102143. DOI: 10.1016/j.compmedimag.2021.102143. ScienceDirect
12. Reddy D.S., Rajalakshmi P. A Novel Web Application Framework for Ubiquitous Classification of Fatty Liver Using Ultrasound Images. In: *IEEE 5th World Forum on Internet of Things (WF-IoT)*. 2019:502–506. DOI: 10.1109/WF-IoT.2019.8767251. medultrason.ro
13. Gatos I., Tsantis S., Spiliopoulos S., Karnabatidis D., Kagadis G.C. A cascaded deep learning neural network for automated liver steatosis classification. *Sensors*. 2021;21(16):5304. DOI: 10.3390/s21165304. MDPI
14. Zhou Y., He L., Huang Y., Chen H., Zhu Y., Chen Y., Wang W. Ultrasound-based artificial intelligence in gastroenterology and hepatology: current status and future perspectives. *World Journal of Gastroenterology*. 2021;27(38):6531–6549. DOI: 10.3748/wjg.v27.i38.6531.
15. Lee J., Kim D., Kim H., Kim H.J., Lee C.H., Kim H. Texture analysis of gray-scale ultrasound images for staging of hepatic fibrosis. *Journal of Korean Society of Radiology*. 2020;81(2):338–347. DOI: 10.3348/jksr.2019.0185. JKSR Online

16. Matplotlib for Python Developers [Электронный ресурс] // Packt Publishing.
– 2009. – Режим доступа до ресурсу:
[https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/Python/Matplotlib%20for%20Python%20Developers%20\(2009\).pdf](https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/Python/Matplotlib%20for%20Python%20Developers%20(2009).pdf)
17. Waskom M. Seaborn: statistical data visualization [Электронный ресурс] / Michael L. Waskom. – 2021. – Режим доступа до ресурсу:
<https://doi.org/10.21105/joss.03021>
18. Os — Miscellaneous operating system interfaces [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.python.org/uk/3/library/os.html>
19. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML), 2019. PMLR 97:6105–6114.
<https://arxiv.org/abs/1905.11946>
20. Li B., Tai D.-I., Yan K., Chen Y.-C., Huang S.-F., Hsu T.-H., Yu W.-T., Xiao J., Lu L., Harrison A.P. Accurate and Generalizable Quantitative Scoring of Liver Steatosis from Ultrasound Images via Scalable Deep Learning [Электронный ресурс] arXiv preprint arXiv:2110.05664. 2021. Режим доступа до ресурсу: <https://arxiv.org/abs/2110.05664>
21. Delfan N., Ketabi Moghadam P., Khoshnevisan M., Hosseini Chagahi M., Hatami B., Asgharzadeh M., Zali M., Moshiri B., Momeni Moghaddam A., Khalafi M.A., Dehnad K. AI-Driven Non-Invasive Detection and Staging of Steatosis in Fatty Liver Disease Using a Novel Cascade Model and Information Fusion Techniques [Электронный ресурс]. arXiv preprint arXiv:2412.04884. 2024. Режим доступа до ресурсу: <https://arxiv.org/abs/2412.04884>
22. Kokhlikyan N., Miglani V., Martin M., Wang E., Alsallakh B., Reynolds J., Melnikov A., Kliushkina N., Araya C., Yan S., Reblitz-Richardson O. Captum: A unified and generic model interpretability library for PyTorch [Электронный ресурс]. arXiv preprint arXiv:2009.07896. 2020. Режим доступа до ресурсу: <https://arxiv.org/abs/2009.07896>

23. Han A., Byra M., Heba E., Andre M.P., Erdman J.W. Jr., Loomba R., Sirlin C.B., O'Brien W.D. Jr. Noninvasive Diagnosis of Nonalcoholic Fatty Liver Disease and Quantification of Liver Fat with Radiofrequency Ultrasound Data Using One-dimensional Convolutional Neural Networks. *Radiology*. 2020;295(2):342–350. DOI: 10.1148/radiol.2020191160.
24. Ojo J., Aborisade D.O., Amole A.O., Durodola A.O. Comparative Analysis of Textural Features Derived from GLCM for Ultrasound Liver Image Classification. *International Journal of Emerging Trends & Technology in Computer Science*. 2014;11(6):239–243. DOI: 10.14445/22312803/IJCTT-V11P151.
25. Haralick R.M., Shanmugam K., Dinstein I. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*. 1973;SMC-3(6):610–621. DOI: 10.1109/TSMC.1973.4309314.
26. Nam K., Lee J., Kim D., Kim H.J., Lee C.H., Kim H. Improving diagnostic accuracy of ultrasound texture features in hepatic steatosis detection using various beamforming speeds of sound. *Phys Med Biol*. 2023;68(3):035001. DOI: 10.1088/1361-6560/acb3d3.
27. Ojo J., Aborisade D.O., Amole A.O., Durodola A.O. Comparative Analysis of Textural Features Derived from GLCM for Ultrasound Liver Image Classification. *International Journal of Computer Trends and Technology*. 2014;11(6):239–243. DOI: 10.14445/22312803/IJCTT-V11P151.
28. Kiani F. Texture features in medical image analysis: a survey. *arXiv preprint*. 2022. arXiv:2208.02046.