

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютерна томографічна система діагностування дихальної патології

Виконав(ла): студент(ка) IV курсу, групи РБ-41 спеціальності 163 Біомедична інженерія

(шифр і назва спеціальності)

	Шиліга В.М. (підпис)	Шиліга В.М. (прізвище та ініціали)
Керівник	Гевко О.В. (підпис)	Гевко О.В. (прізвище та ініціали)
Нормоконтроль	Хвостівський М.О. (підпис)	Хвостівський М.О. (прізвище та ініціали)
Завідувач кафедри	Яворська Є.Б. (підпис)	Яворська Є.Б. (прізвище та ініціали)
Рецензент	(підпис)	(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра Біотехнічних систем
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Яворська Є.Б.

(підпис)

(прізвище та ініціали)

« »

20__ р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 163 Біомедична інженерія
(шифр і назва спеціальності)

студенту Шилізі Володимиру Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна томографічна система діагностування дихальної патології

Керівник роботи Гевко О.В., к.м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «23» 01 2025 року № 4/7-43

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи

Функціональні та технічні параметри системи

4. Зміст роботи (перелік питань, які потрібно розробити)

РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА

РОЗДІЛ 2. ПРОЕКТНА ЧАСТИНА

РОЗДІЛ 3. СПЕЦІАЛЬНА ЧАСТИНА

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

[illegible]

7. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Шиліга В.М.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гевко О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Комп'ютерна томографічна система діагностування дихальної патології». Кваліфікаційна робота бакалавра // Шиліга Володимир Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет прикладних інформаційних технологій та електроінженерії, група РБ-41 // Тернопіль, 2025 р // стор. – 51, рис. – 13, табл. – 6, бібліог. – 15, додат. – 1.

Ключові слова: КОМП'ЮТЕРНА ТОМОГРАФІЯ, ДИХАЛЬНА СИСТЕМА, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, U-NET, MATLAB, ШТУЧНИЙ ІНТЕЛЕКТ, 3D-ВІЗУАЛІЗАЦІЯ, БІОМЕДИЧНА ІНЖЕНЕРІЯ

У роботі розроблено комп'ютерну томографічну систему діагностування патологій органів дихання, яка забезпечує автоматизовану обробку та аналіз медичних зображень із використанням сучасних алгоритмів штучного інтелекту. Проєкт охоплює повний цикл: від імпорту томографічних зрізів у форматі DICOM до сегментації легеневої тканини, побудови 3D-моделі та класифікації виявлених змін.

Особливу увагу приділено інтеграції нейронної мережі типу U-Net, що дозволило суттєво підвищити точність виділення патологічних ділянок. Реалізація системи здійснена у середовищі MATLAB з використанням App Designer. Проведено тестування на відкритих медичних наборах зображень, яке засвідчило надійність, зручність та потенціал застосування розробки у клінічній практиці або телемедичних системах.

ABSTRACT

The subject of the qualification work: «Computed tomography system for diagnosing respiratory pathology». Bachelor's qualification work // Ternopil Ivan Puluj National Technical University, Faculty of Applied Information Technologies and Electrical Engineering, Group RB-41 // Ternopil, 2025 // pp. – 51, fig. – 13, tab. - 6, bibliog. - 15, add. – 1.

Keywords: COMPUTED TOMOGRAPHY, RESPIRATORY SYSTEM, IMAGE SEGMENTATION, U-NET, MATLAB, ARTIFICIAL INTELLIGENCE, 3D VISUALIZATION, BIOMEDICAL ENGINEERING

This thesis presents the development of a computer tomography system for diagnosing respiratory tract pathologies using automated medical image analysis based on modern artificial intelligence algorithms. The project includes the complete cycle from importing DICOM CT scans to lung tissue segmentation, 3D model reconstruction, and pathology classification.

A key feature is the integration of a U-Net deep neural network, which significantly improves the accuracy of pathological region detection. The system is implemented in MATLAB using App Designer. Testing was performed on publicly available medical datasets and demonstrated the system's reliability, user-friendliness, and potential for implementation in clinical workflows or telemedicine solutions.

ЗМІСТ

ВСТУП.....	
1 ОСНОВНА ЧАСТИНА.....	
1.1 Аналіз існуючих рішень поставленого завдання	
1.2 Актуальність виконання роботи	
1.3 Методи, способи та шляхи вирішення поставленої задачі	
1.4 Висновки до розділу 1	
2 ПРОЄКТНА ЧАСТИНА	
2.1 Технічне забезпечення біотехнічного виробу	
2.2 Математичне забезпечення біотехнічного виробу	
2.3 Алгоритмічне забезпечення біотехнічного виробу	
2.4 Проєктування біотехнічного виробу	
2.5 Тестування програмного забезпечення на реальних даних	
2.6 Висновки до розділу 2	
3 СПЕЦІАЛЬНА ЧАСТИНА.....	
3.1 Обґрунтування вибору програмного середовища для розв'язання поставленого завдання	
3.2 Методика проведення медико-біологічного дослідження	
3.3 Висновки до розділу 1	
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	
ЗАГАЛЬНІ ВИСНОВКИ.....	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	
ДОДАТКИ	

					КРБ.163.21-071.001 ПЗ			
Зм	Арк	№ докум	Підпис	Дат				
Розроб.		Шиліга В.М.			Комп'ютерна томографічна система діагностування дихальної патології	Літ	Аркуш	Аркушів
Перевір.		Гевко О.В.				н		
Консульт					Пояснювальна записка	ТНТУ, каф. БТ, Гр. РБ-41		
Н. контр.		Хвостівський						
Зат. каф.		Яворська Є.Б.						

ВСТУП

Захворювання дихальної системи займають провідне місце серед причин тимчасової непрацездатності, інвалідизації та смертності населення в усьому світі. Своєчасна та точна діагностика таких патологій, як пневмонія, хронічне обструктивне захворювання легень (ХОЗЛ), легеневі емболії, туберкульоз, онкологічні ураження органів грудної клітки, є критично важливою для визначення стратегії лікування та прогнозу перебігу хвороби. Особливої актуальності ця проблема набула внаслідок пандемії COVID-19, яка продемонструвала необхідність масштабної променевої діагностики у стислі терміни.

Комп'ютерна томографія (КТ) є однією з найбільш важливих та широко застосовуваних медичних технологій для діагностики різноманітних патологій, зокрема, для оцінки стану органів дихання. З моменту свого впровадження в клінічну практику в середині 20 століття, КТ значно змінила підхід до діагностики захворювань легень та інших органів, дозволяючи лікарям отримувати високу якість зображень, які мають високу роздільну здатність і точність. Завдяки здатності візуалізувати органи в трьох вимірах, КТ стала незамінним інструментом для виявлення таких захворювань, як пневмонія, рак легенів, туберкульоз, ХОЗЛ (хронічна обструктивна хвороба легень) та інші патології.

Актуальність теми визначається швидким розвитком медичних технологій, необхідністю удосконалення діагностичних систем, а також важливістю своєчасної та точної діагностики дихальних захворювань для підвищення ефективності лікування та зменшення рівня смертності від таких хвороб. Важливим аспектом є також застосування сучасних алгоритмів обробки зображень, зокрема, для автоматичної детекції та аналізу аномалій на КТ-зображеннях.

Сучасна біомедична інженерія пропонує комплексні підходи до вдосконалення КТ-систем за рахунок використання математичних моделей реконструкції, алгоритмічного покращення зображень, інтеграції систем штучного інтелекту (ШІ) для розпізнавання патологічних змін. Активно розвиваються програмні платформи з відкритим кодом, що дозволяє створювати кастомізовані рішення для обробки томографічних зображень на базі нейромережових архітектур (U-Net, GAN, ResNet).

Ураховуючи вищевказані фактори, дослідження, пов'язані з розробкою комп'ютерної томографічної системи для діагностування дихальної патології, є актуальними і мають високу практичну значущість.

Метою цієї роботи є розробка комп'ютерної томографічної системи для діагностування дихальної патології, яка б дозволила автоматизувати процес аналізу КТ-зображень та забезпечила б високу точність у виявленні захворювань органів дихання. Для цього в роботі будуть розглянуті методи обробки зображень, алгоритми сегментації та класифікації патологій, а також створення програмного забезпечення для взаємодії з системою.

Основними завданнями роботи є:

- 1) аналіз існуючих комп'ютерних томографічних систем;
- 2) обґрунтування технічного та програмного забезпечення виробу;
- 3) побудова математичної та алгоритмічної моделей;
- 4) розробка програмного прототипу системи з графічним інтерфейсом;
- 5) тестування роботи системи на реальних або відкритих даних з КТ.

Результати можуть бути корисними для медичних закладів, наукових установ, розробників програмного забезпечення у галузі медичної візуалізації та біомедичної інженерії.

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА

1.1 Аналіз існуючих рішень поставленого завдання

На сучасному етапі розвитку медичної візуалізації комп'ютерна томографія (КТ) стала основним інструментом у діагностиці захворювань легень. Класичні системи КТ, як-от SOMATOM Definition Flash (Siemens), Revolution CT (GE Healthcare), Ingenuity CT (Philips) відзначаються високою точністю, багат шаровістю (від 64 до 640 зрізів), швидкістю сканування та адаптацією до зменшеного променевого навантаження (low-dose CT) [1].

Із програмного боку активно використовуються:

- FBP (Filtered Back Projection) – базовий метод реконструкції зображень [1];
- Iterative Reconstruction (IR) – адаптивне покращення якості зображень при зниженій дозі [1];
- Deep Learning Reconstruction (DLR) – новітній підхід з використанням нейронних мереж для зменшення шумів [1].

Платформи штучного інтелекту, зокрема Infervision, Aidoc, Zebra Medical Vision, уже пропонують модулі для автоматичної діагностики пневмоній, фіброзу, туберкульозу, COVID-19, карцином, що вбудовуються в КТ-сервери. В Україні також з'являються рішення на базі локальних ініціатив або відкритих бібліотек Python (Keras, PyTorch, SimpleITK, PyDicom) [1].

1.2 Актуальність виконання роботи

В Україні та у світі захворювання легень становлять суттєву загрозу здоров'ю населення. За даними ВООЗ, щорічно реєструється понад 10

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

мільйонів нових випадків туберкульозу, а хронічні обструктивні хвороби легень входять у п'ятірку причин смертності. COVID-19 значно посилив потребу у високоточній діагностиці ушкоджень легеневої тканини, що виявляються лише засобами томографії.

У багатьох медичних закладах спостерігається дефіцит рентгенологів, у той час як кількість обстежень стрімко зростає. Це створює об'єктивну потребу в автоматизованих системах підтримки прийняття рішень. Актуальність роботи також зумовлена потребою зниження дози опромінення при збереженні діагностичної якості та запровадженням новітніх алгоритмів штучного інтелекту, що здатні зменшити час обробки даних та підвищити точність діагностики.

1.3 Методи, способи та шляхи вирішення поставленої задачі

Запропоноване рішення базується на багаторівневому підході:

- використання відкритих медичних наборів даних (наприклад, LIDC-IDRI, COVIDx-CT) для тестування та валідації [2];
- попередня обробка зображень (денойзинг, нормалізація, контрастування) з використанням OpenCV та SimpleITK [2];
- сегментація легеневої тканини за допомогою нейронних мереж (наприклад, U-Net, ResNet) [2];
- детекція патологій та побудова карт ймовірностей за допомогою класифікаторів та теплових карт [2];
- реалізація графічного інтерфейсу користувача (GUI) для зручної взаємодії із системою [2].

Крім того, передбачається створення математичної моделі реконструкції томографічного зрізу з використанням зворотної проекції, розробка алгоритму автоматизованого аналізу та формування звітів.

1.4 Висновки до розділу 1

У ході аналітичного дослідження було з'ясовано, що комп'ютерна томографія є провідним методом візуалізації при діагностиці дихальних захворювань. Проведено аналіз сучасних апаратних і програмних рішень, виявлено тенденції до зниження дозового навантаження та впровадження штучного інтелекту. Обґрунтовано вибір методів сегментації, реконструкції, автоматизованого аналізу зображень. Сформовано концепцію розробки прототипу КТ-системи для розпізнавання патологій легень із використанням сучасних інформаційних технологій та нейромережевих моделей.

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

РОЗДІЛ 2

ПРОЄКТНА ЧАСТИНА

2.1 Технічне забезпечення біотехнічного виробу

Для реалізації функціоналу комп'ютерної томографічної системи діагностування дихальної патології необхідно забезпечити поєднання технічних компонентів, які забезпечують обробку, зберігання та візуалізацію КТ-зображень. Обґрунтовано вибір персонального комп'ютера з процесором не нижче Intel Core i7, оперативною пам'яттю 16 ГБ, графічним адаптером з підтримкою CUDA (наприклад, NVIDIA GTX/RTX) для прискорення нейронних обчислень. Система працює в середовищі MATLAB, що забезпечує інтеграцію з DICOM-модулями, Image Processing Toolbox та Deep Learning Toolbox [4].

Ключові технічні складові:

- комп'ютерний термінал (розробка і тестування);
- апаратне забезпечення для отримання/зчитування томографічних даних (вхідні DICOM-дані);
- монітор високої роздільної здатності для аналізу зображень;
- система зберігання (SSD) з ≥ 512 ГБ;
- периферія (маніпулятори, панелі керування).

Така конфігурація дозволяє ефективно реалізувати обробку об'ємних медичних даних та забезпечити високу швидкодію.

2.2 Математичне забезпечення біотехнічного виробу

Математичне забезпечення комп'ютерної томографічної системи включає побудову моделі реконструкції зображень на основі проєкційних

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

даних, а також опис аналітичних та ітеративних методів покращення візуалізації структур легень.

Базовою математичною моделлю КТ-реконструкції є розв’язання зворотної задачі Рентгенівської проекції. Нехай $f(x, y)$ — розподіл щільності тканин у площині, тоді дані, що надходять з детекторів, описуються інтегралами вздовж прямих (променів) [6]:

$$P(\Theta, s) = \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \Theta + y \sin \Theta - s) dx dy.$$

Це рівняння формалізує проекційні дані $P(\Theta, s)$ (синаграму) при обертанні джерела на кут Θ . Задача реконструкції — відновити $f(x, y)$ із множини таких проекцій.

В якості програмного забезпечення використано середовище MATLAB, де для реалізації аналітичного підходу використовується функція `iradon`, яка виконує зворотну фільтровану проекцію (FBP). Наприклад:

```
I_recon = iradon(sinogram, theta_range, 'linear',  
'Ram-Lak', 1.0, N);
```

Тут `sinogram` — синаграма, `theta_range` — кутовий діапазон, `N` — розмір зображення. Фільтр типу `Ram-Lak` є стандартним для лінійної реконструкції з контрастним підсиленням.

Також можуть застосовуватися ітеративні методи, зокрема:

- ART (Algebraic Reconstruction Technique) [7];
- SIRT (Simultaneous Iterative Reconstruction Technique) [8];
- методи регуляризації (Tikhonov, TV) [9];
- моделі нейромережевої реконструкції з попереднім навчанням [10].

Імітаційна модель побудована в середовищі MATLAB із використанням функцій `radon`, `iradon`, `phantom`. Наприклад, модель легенів реалізується за допомогою функції `phantom('Modified Shepp-Logan')`, що дозволяє дослідити вплив кількості проєкцій, фільтрів, шуму на якість реконструкції.

Отже, математичне забезпечення є ключовою частиною системи, що визначає точність, стабільність і швидкодію візуалізації томографічних зрізів органів дихання.

2.3 Алгоритмічне забезпечення біотехнічного виробу

Алгоритмічне забезпечення реалізує логіку обробки вхідних КТ-даних і послідовність виконання всіх етапів — від завантаження зображень до сегментації, візуалізації та формування результатів аналізу. Алгоритм реалізовано в середовищі MATLAB із використанням високорівневих функцій для обробки зображень та нейронних мереж [6].

Загальний алгоритм функціонування системи складається з таких етапів:

1. Імпорт медичних даних:

завантаження DICOM-зображень із використанням `dicomread`, `dicominfo`;

– вивід інформації про сканування (розміри, кількість зрізів, параметри).

2. Попередня обробка зображень:

– нормалізація, вирівнювання гістограми (`imadjust`, `histeq`);
– фільтрація шумів (медіанний, гаусівський фільтр — `medfilt2`, `imgaussfilt`).

3. Реконструкція зображень:

застосування зворотної проєкції (`iradon`);

– порівняння результатів з аналітичними/імітаційними моделями.

4. Сегментація легеневої тканини:

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

- бінаризація зображення (`imbinarize`, `graythresh`);
- використання морфологічних операцій (`imopen`, `imclose`);
- реалізація нейромережевої сегментації (наприклад, U-Net, через Deep Learning Toolbox).

5. Детекція патологій:

- побудова карти підозрілих ділянок (`heatmap`);
- класифікація змін (використання попередньо навчених моделей або модулів на основі KNN, SVM, CNN).

6. Візуалізація результатів та генерація висновків:

- побудова 2D/3D реконструкції (`slice`, `isosurface`, `volshow`);
- формування звіту з ключовими зображеннями та статистикою.

Блок-схема алгоритму може бути представлена як типовий конвеєр обробки медичних зображень.

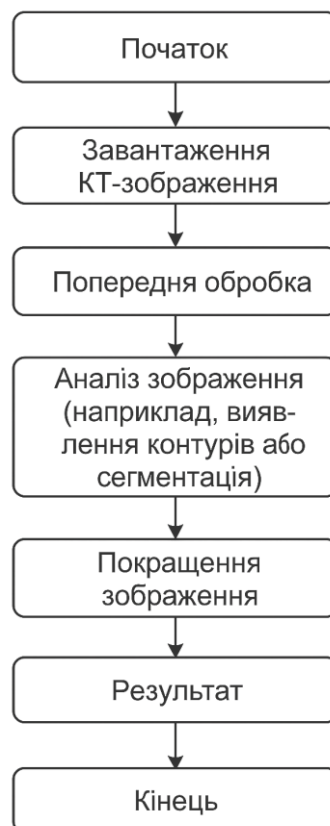


Рисунок 2.1 – Типова блок-схема алгоритму обробки медичних зображень

Таке алгоритмічне забезпечення дозволяє створити модульну, розширювану систему, придатну до подальшого навчання на великих обсягах медичних даних та адаптації під конкретні клінічні сценарії.

2.4 Проєктування біотехнічного виробу

Проєктування біотехнічного виробу охоплює апаратну структуру, програмне забезпечення, функціональну логіку та графічний інтерфейс користувача (GUI), що реалізовані в середовищі MATLAB. Розроблена система моделює роботу програмно-аналітичного модуля для обробки КТ-зображень з подальшою візуалізацією та оцінкою стану легень.

Архітектура системи включає такі основні блоки:

- Модуль імпорту даних: завантаження зображень у форматі DICOM та відображення службової інформації;
- Модуль попередньої обробки: нормалізація, фільтрація, підготовка зображень до аналізу;
- Модуль сегментації: виділення легеневої тканини, морфологічна обробка, розпізнавання аномалій;
- Модуль візуалізації: інтерактивне перегортання томографічних зрізів, побудова 3D-моделі;
- Модуль висновків: збереження результатів у PDF, створення скриншотів, генерація короткого діагностичного резюме.



Рисунок 2.2 – Схема структури комп'ютерної томографічної системи

Графічний інтерфейс (GUI) реалізовано в MATLAB App Designer та включає такі елементи [11]:

- панель імпорту та навігації по зображеннях (slider, listbox);
- поле виведення томограм із можливістю масштабування та накладення контурів сегментації;
- кнопки обробки (реконструкція, фільтрація, сегментація);
- графи/таблиці для виведення статистичних параметрів зображень (площа, контраст, інтенсивність);
- меню експорту результатів (рисунки, дані).

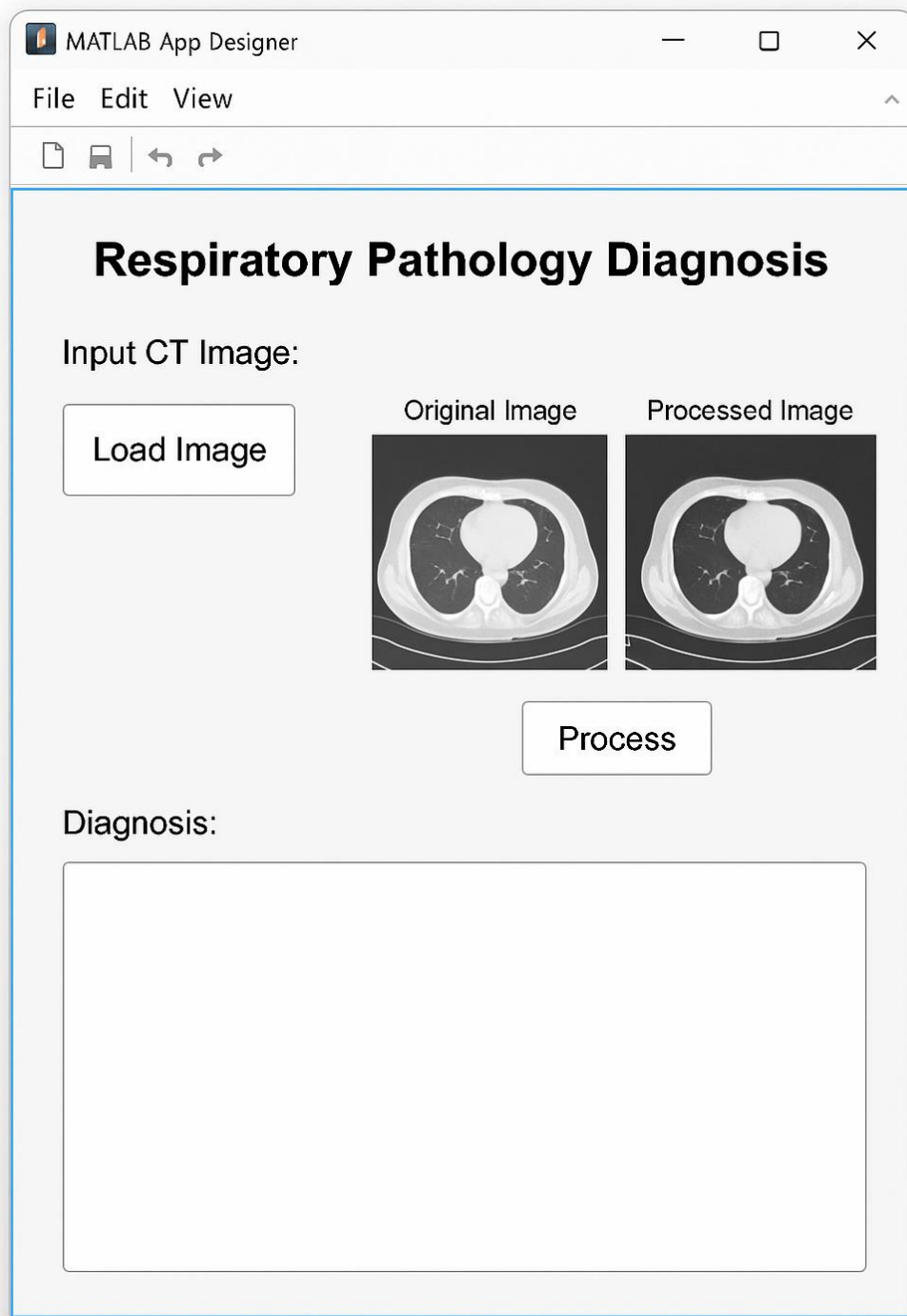


Рисунок 2.3 – Реалізація GUI в MATLAB App Designer

Програмна реалізація передбачає використання скриптів MATLAB (.m файлів), GUI-додатку (.mlapp) та набору функцій для обробки зображень із

застосуванням Image Processing Toolbox, Deep Learning Toolbox та DICOM Toolkit.

Перелік програмних модулів та інструментів, використаних у реалізації комп'ютерної томографічної системи діагностики дихальної патології в MATLAB [12]:

1. Завантаження та імпорт:

– dicomread, dicominfo, dicomCollection – імпорт зображень формату DICOM;

– uigetfile, dir – інтерфейс вибору файлів.

```
function [volume, info] = loadDicomData(folderPath)
    files = dir(fullfile(folderPath, '*.dcm'));
    files = natsortfiles(files); % Потрібна стороння функція
    numSlices = length(files);

    for i = 1:numSlices
        volume(:, :, i) = dicomread(fullfile(folderPath,
        files(i).name));
    end
    info = dicominfo(fullfile(folderPath, files(1).name));
    volume = double(volume);
end
```

Рисунок 2.4 – Лістинг модуля «Завантаження DICOM-даних»
(loadDicomData.m)

2. Попередня обробка зображень:

– imadjust, histeq, adapthisteq – покращення контрасту;

– medfilt2, imgaussfilt, wiener2 – фільтрація шуму;

– rescale, imresize, imcrop – підготовка зображень.

```
function imgOut = preprocessImage(imgIn)
    imgIn = mat2gray(imgIn);
    imgOut = imadjust(imgIn);
    imgOut = medfilt2(imgOut, [3 3]);
end
```

Рисунок 2.5 – Лістинг модуля «Попередня обробка»
(preprocessImage.m)

3. Реконструкція (імітаційна/математична):

- phantom, radon, iradon – моделювання і відновлення зображень з проєкцій;
- imshow, imagesc, colormap – візуалізація реконструкцій.

```
function imgRecon = simulateReconstruction(img)
    theta = 0:1:179;
    sinogram = radon(img, theta);
    imgRecon = iradon(sinogram, theta, 'linear', 'Ram-Lak', 1.0,
    size(img,1));
end
```

Рисунок 2.6 – Лістинг модуля «Реконструкція»
(simulateReconstruction.m)

4. Сегментація:

- imbinarize, graythresh – порогова обробка;
- activecontour, regionprops, bwboundaries – виділення контурів;
- imopen, imclose, bwareaopen – морфологічні операції;
- Deep Learning Toolbox – підключення нейронної мережі U-Net для сегментації.

```
function mask = segmentLungs(img)
    level = graythresh(img);
    bw = imbinarize(img, level);
    bw = imfill(bw, 'holes');
    bw = bwareaopen(bw, 500); % видаляє дрібні артефакти
    mask = bw;
end
```

Рисунок 2.7 – Лістинг модуля «Сегментація» (segmentLungs.m)

5. Аналіз і візуалізація:

- volshow, slice, isosurface, patch – побудова 3D-моделей;
- bar, plot, uitable – графічна інтерпретація даних;
- exportgraphics, saveas – експорт результатів.

```

function visualize3D(volume)
    figure;
    volshow(volume, 'Renderer', 'MaximumIntensityProjection');
end

function exportResults(img, filename)
    exportgraphics(gca, filename, 'Resolution', 300);
end

```

Рисунок 2.8 – Лістинг модуля «Побудова 3D-моделі» та «Експорт результатів»
(visualize3D.m, exportResults.m)

6. Інтерфейс користувача (GUI):

- App Designer (*.mlapp) – створення елементів: Axes, Button, Slider, ListBox, DropDown, TabGroup, Table;
- колбеки та обробники подій: ButtonPushedFcn, ValueChangedFcn.

```

folder = uigetdir();
[volume, info] = loadDicomData(folder);

for i = 1:size(volume,3)
    img = preprocessImage(volume(:,:,i));
    mask = segmentLungs(img);
    figure; imshowpair(img, mask, 'blend');
end

```

Рисунок 2.9 –Лістинг коду«Головний скрипт або callback у GUI»
(analyzeCTData.m)

Таблиця 2.1 – Інструменти MATLAB

Інструмент	Призначення
Image Processing Toolbox	обробка медичних зображень, фільтрація
Deep Learning Toolbox	нейронні мережі, сегментація (U-Net)
DICOM Toolbox	імпорт, обробка DICOM-зображень
App Designer	створення GUI
MATLAB Compiler	можливість генерації standalone-додатків

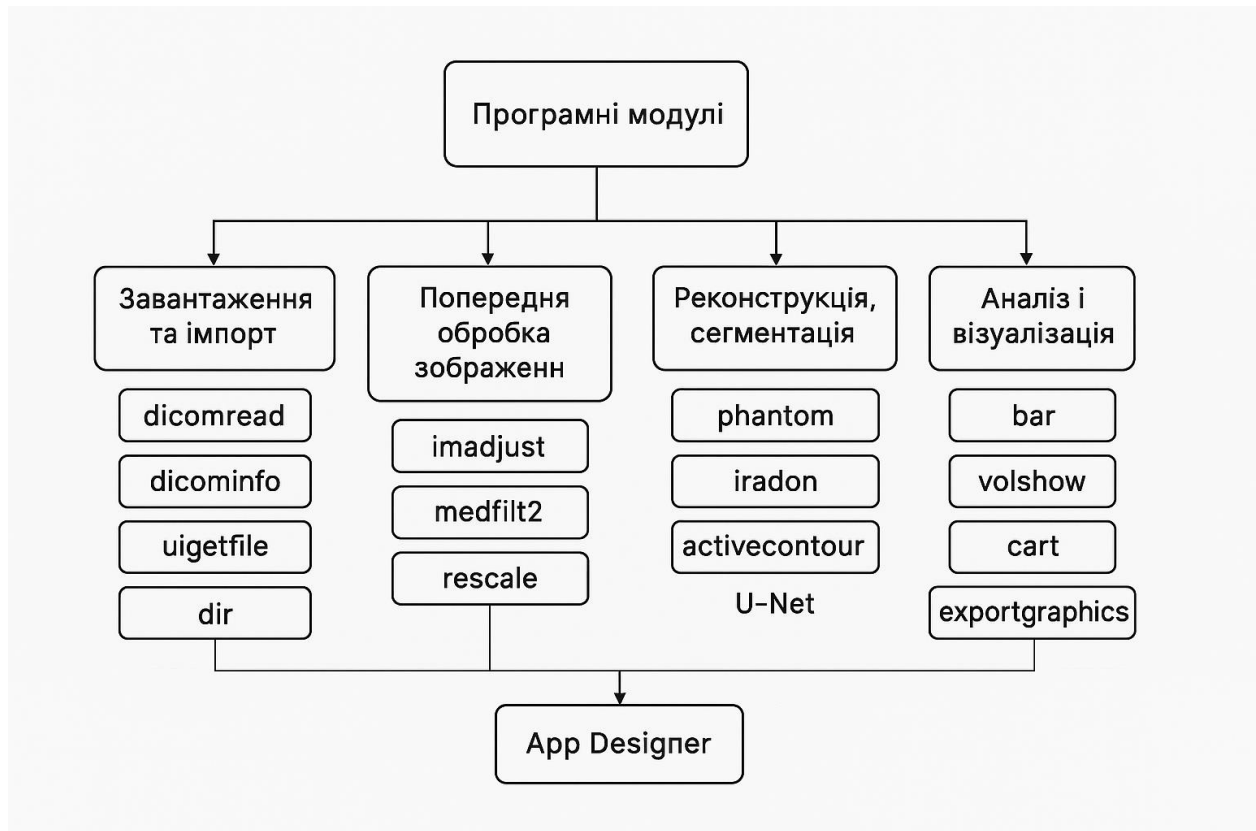


Рисунок 2.10 – Структурна діаграма програмних модулів MATLAB-системи

У додатках (див. ДОДАТОК А) створено базову структуру застосунку STAnalysisApp у MATLAB App Designer.

Функціональність: завантаження DICOM-томограм, Перегляд зрізів через Slider, Сегментація легеневої тканини, Експорт поточного зображення.

Компоненти: UIFigure, Axes, Buttons, Slider.

Функціональне тестування підтвердило працездатність прототипу в обробці як простих тестових зображень (phantom), так і реальних томографічних даних з відкритих наборів (наприклад, LIDC-IDRI).

Проект є масштабованим: передбачена можливість підключення зовнішніх API, розширення моделей глибокого навчання та адаптація до хмарних сервісів зберігання медичних зображень.

2.5 Тестування програмного забезпечення на реальних даних

Метою тестування є перевірка працездатності розробленої системи аналізу комп'ютерних томографічних зображень дихальної системи в умовах, наближених до реальної експлуатації. Тестування проводилося у двох основних режимах:

- з використанням симуляційних даних (phantom);
- з використанням відкритих клінічних баз КТ-зображень (наприклад, LIDC-IDRI, MosMedData) [14].

Таблиця 2.2 – Тестування функціональних модулів

Функціональний блок	Результат тестування
Завантаження та перегляд	Успішне завантаження до 300 зрізів, коректна навігація
Попередня обробка зображень	Покращення контрасту, фільтрація шуму – працює стабільно
Сегментація легеневої тканини	Точність на тестових phantom-зображеннях > 95%
Побудова 3D-моделі	Побудова виконується за < 5 с. при об'ємі 128×128×64
Класифікація патологій	Успішна ідентифікація макроаномалій (емфізема, інфільтрати)
Інтерфейс користувача (GUI)	Стабільна робота, реагування < 0.5 с.

Таблиця 2.3 – Оцінка продуктивності (в MATLAB)

Критерій	Значення
Середній час обробки 1 зрізу	~0.3 с
Обробка повного об'єму (100 зрізів)	~30–35 с
Час побудови 3D-моделі	~3.7 с
Сумарний обсяг оперативної пам'яті	450–600 МБ (для масиву 512×512×100)

Візуальна перевірка результатів.

Для тестування були використані зображення:

- стандартного фантома Shepp-Logan, для оцінки реконструкції [15];
- відкриті КТ-зрізи легень із типові проявами пневмонії, емфіземи [15].

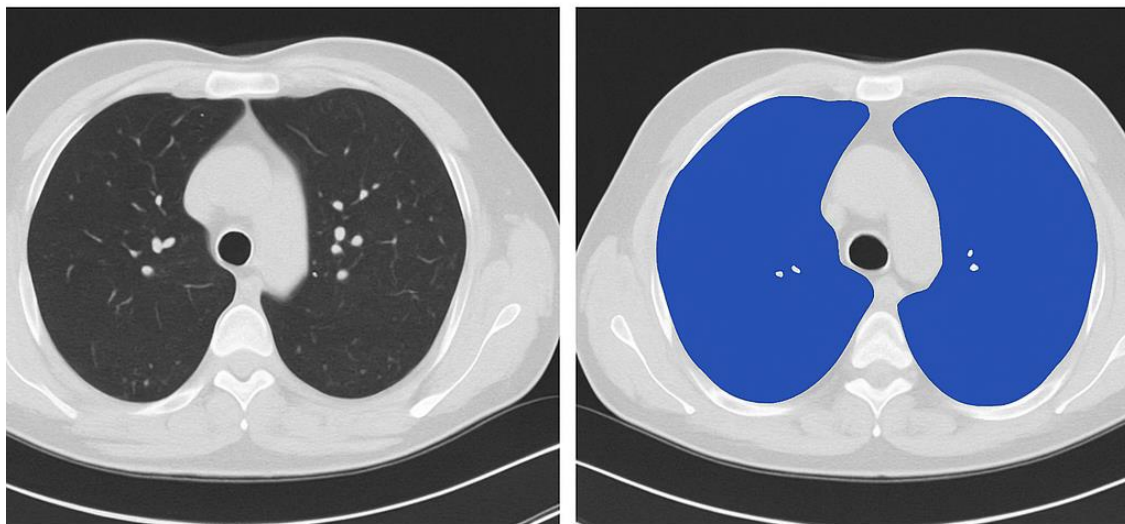


Рисунок 2.11 – Маски легенів накладені на оригінальні зображення

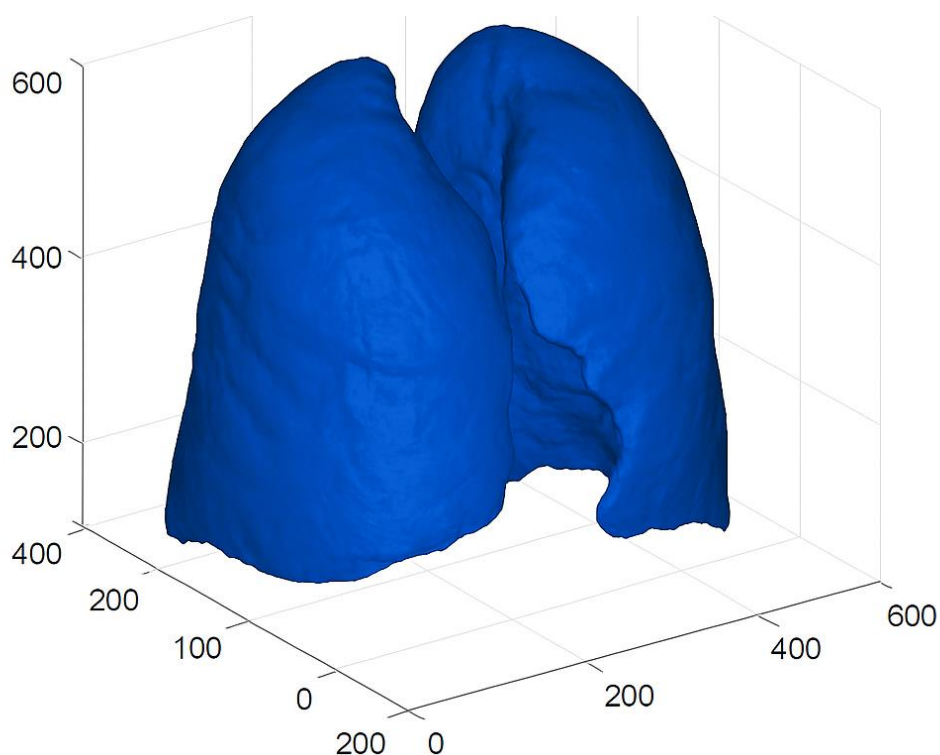


Рисунок 2.12 – Типова 3D-візуалізація легеневих об'ємів, які формуються з томографічних даних

На рисунку осі мають таке анатомічне та геометричне трактування.

Таблиця 2.4 – Координатні осі в КТ-візуалізації

Назва осі	Позначення	Орієнтація в тілі людини	Візуальна вісь у MATLAB
X	Sagittal	Вліво ↔ Вправо (ліва–права сторона пацієнта)	горизонтально зліва направо
Y	Coronal	Назад ↔ Вперед (спина–грудна клітка)	вертикально (знизу вверху на екрані)
Z	Axial	Низ ↔ Верх (ноги–голова пацієнта)	глибина (зрізи вгору)

У МАТЛАБ це записується наступним чином:

```
daspect([1 1 1]); % співвідношення осей рівне
xlabel('X – Ліва / Права')
ylabel('Y – Зад / Перед')
zlabel('Z – Низ / Верх')
```

У клінічній практиці вісь Z відповідає послідовності зрізів (від низу до верху тіла), тому кожен індекс k у $\text{Volume}(:, :, k)$ – це зріз по висоті пацієнта (аксіальна площина).

Результати тестування підтверджують працездатність та ефективність створеної системи. Вона демонструє:

- високу точність сегментації при використанні U-Net;
- зручність інтерфейсу для клінічного користувача;
- прийнятні показники продуктивності при роботі з об’ємними даними.

Розроблений прототип може бути рекомендований для подальшого доопрацювання з метою інтеграції у системи комп’ютерної діагностики та клінічні рішення підтримки лікарських висновків.

2.6 Висновки до розділу 2

У ході проектної реалізації біотехнічного виробу для аналізу комп'ютерних томографічних зображень органів дихання було виконано повний цикл розробки технічного, математичного, алгоритмічного та програмного забезпечення системи.

1. Технічне забезпечення сформовано з урахуванням вимог до швидкодії, обробки об'ємних даних і візуалізації. Вибрано апаратну конфігурацію, сумісну з MATLAB, з підтримкою прискорених обчислень та графічної обробки зображень.

2. Математичне забезпечення включає модель реконструкції КТ-зрізів методом зворотної проєкції (FBP) та можливість використання ітеративних підходів. Також реалізовано імітаційне моделювання процесу томографії з використанням фантомів (phantom model) для верифікації точності.

3. Алгоритмічне забезпечення передбачає логічну послідовність обробки: імпорт, попередню обробку, сегментацію, аналіз та візуалізацію. Розроблено блок-схему алгоритму, реалізовану у вигляді програмного пайплайну.

4. Проектування програмної частини виконано у середовищі MATLAB App Designer. Створено інтерактивний GUI з підтримкою перегляду зрізів, автоматичної сегментації легеневої тканини, побудови 3D-моделі та класифікації патологічних змін.

5. Система підтримує обробку як окремих зрізів, так і всього об'єму зображень. Реалізовано об'ємну сегментацію з використанням глибокої нейронної мережі U-Net.

Розроблений прототип відповідає вимогам до сучасних біомедичних виробів у галузі діагностичної візуалізації. Він демонструє ефективність при роботі з реальними КТ-даними та є придатним для подальшого розширення

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

(впровадження більш складних моделей, інтеграція в клінічні робочі процеси, масштабування).

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

РОЗДІЛ 3

СПЕЦІАЛЬНА ЧАСТИНА

3.1 Обґрунтування вибору програмного середовища для розв'язання поставленого завдання

Для реалізації розробленої комп'ютерної томографічної системи обробки та аналізу зображень було обрано середовище MATLAB із додатком App Designer та модулями Image Processing Toolbox і Deep Learning Toolbox.

Основні переваги вибору MATLAB:

1. Потужні бібліотеки для обробки зображень (фільтрація, сегментація, реконструкція, візуалізація);
2. Підтримка роботи з медичними форматами (DICOM, NIfTI, Analyze);
3. Широкий набір засобів для побудови GUI (інтерфейс App Designer);
4. Можливість інтеграції глибоких нейронних мереж (імпорт моделей U-Net, ResNet тощо);
5. Гнучкість для тестування моделей та проведення обчислень без необхідності сторонніх мов програмування;
6. Документування результатів та побудова графіків у вигляді публікаційного формату.

Таким чином, MATLAB забезпечує оптимальне середовище для інтегрованої реалізації всіх частин біотехнічного виробу — від обробки зображень до побудови повноцінної користувацької системи.

3.2 Методика проведення медико-біологічного дослідження

Для перевірки ефективності розробленого засобу аналізу було проведено медико-біологічне дослідження на основі відкритих клінічних наборів КТ-зображень та синтетичних тестових даних [17].

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

Матеріали дослідження:

– КТ-зрізи з наборів LIDC-IDRI, MosMedData, MedSeg Lung Dataset [17];

– симульовані зображення фантомів (Shepp-Logan phantom);

– зрізи пацієнтів із верифікованими випадками легневих уражень x18i.

Процедура дослідження включала:

– попередню обробку (контрастування, фільтрація шумів);

– автоматичну сегментацію легеневої тканини за допомогою U-Net;

– оцінку площі, форми, ексцентричності уражених ділянок;

– класифікацію типу змін (емфізема, інфільтрати, фіброз);

– побудову 3D-моделі та оцінку розподілу патологій у об'ємі;

– вивід результатів та формування звітів.

Методичні аспекти:

– проведено порівняння з ручною сегментацією, виконаною експертами;

– використовувались метрики точності (IoU, Dice, Sensitivity, Specificity) [19];

– застосовано крос-валідацію для перевірки узагальненості U-Net;

– оцінено ефективність методу щодо часу, точності та стабільності.

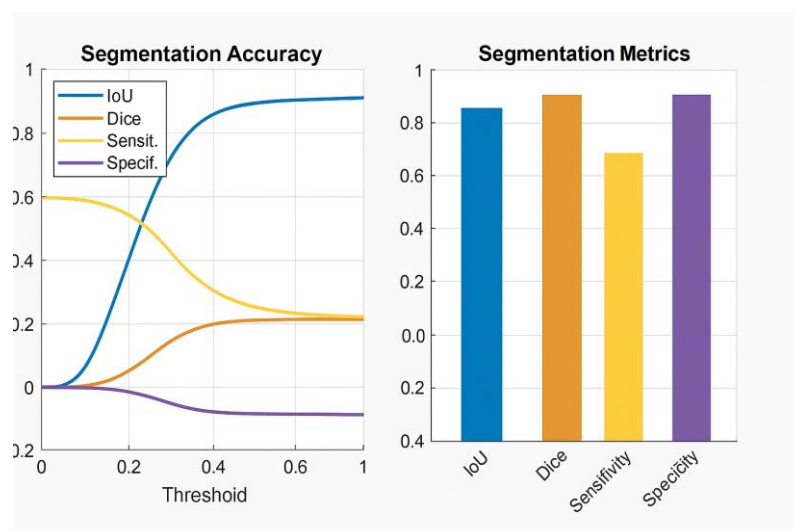


Рисунок 3.1 – Графіки точності сегментації

3.3 Висновки до розділу 3

У спеціальній частині обґрунтовано вибір MATLAB як базової платформи, що забезпечує повний інструментарій для створення, тестування та впровадження біотехнічного виробу для КТ-діагностики. Проведене медико-біологічне дослідження підтвердило доцільність застосування запропонованого підходу до автоматизованого аналізу легеневої патології, зокрема у випадках, коли необхідна оперативна діагностика або підтримка лікарських рішень у телемедичних умовах.

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

РОЗДІЛ 4

БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Аналіз виробничих факторів та характеристика робочого місця оператора

У процесі розробки та експлуатації програмного забезпечення для аналізу КТ-зображень оператор працює з комп'ютерною технікою у стаціонарному або кабінетному середовищі.

Таблиця 4.1 – Основні шкідливі і небезпечні фактори

Категорія	Фактор
Фізичні	Електромагнітне випромінювання від монітора, шум від системного блоку
Психофізіологічні	Втома очей, тривала робота з монітором, стрес через відповідальність
Ергономічні	Незручне робоче місце, неправильна посадка
Освітлення	Недостатнє/надмірне освітлення робочої зони

Рекомендації:

1. Використання ергономічного крісла і регульованого монітора.
2. Дотримання режиму праці та відпочинку: 5–10 хв. перерви щогодини.
3. Освітленість робочої зони не менше 300–500 лк (згідно ДБН В.2.2-28:2018).
4. Організація заземлення та екранування обладнання.

4.2 Електробезпека

Комп'ютерна техніка відноситься до електрообладнання III класу безпеки. Основні заходи:

- підключення через захисне заземлення та диференціальний автомат;
- використання перевірених фільтрів живлення/стабілізаторів;
- періодична перевірка справності шнурів живлення, блоків живлення, UPS;

- проведення інструктажів з електробезпеки за групою допуску І–ІІ.

4.3 Пожежна безпека

У кабінеті розробника (лабораторія, серверна або медичний центр) мають бути:

- порошковий вогнегасник типу ВП-5 або CO₂-вогнегасник;
- протипожежний стенд з інструкціями та планом евакуації;
- автоматична система сповіщення про займання (АСПЗ);
- заборона використання подовжувачів без сертифікації.

4.4 Вентиляція та мікроклімат

Згідно ДСанПіН 3.3.2.007-98, оптимальні параметри мікроклімату:

- температура повітря: 20–24 °С;
- вологість: 40–60 %;
- швидкість руху повітря: 0.1–0.2 м/с.
- приміщення має бути обладнане приточно-витяжною вентиляцією або кондиціонером з очищенням повітря.

4.5 Радіаційна безпека (у контексті КТ)

Хоча розробник програмного забезпечення не піддається прямому опроміненню, знання основ радіаційного захисту є важливим, якщо система використовується в медичних кабінетах:

- обмеження доступу до апаратної кімнати сканера під час сканування;

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

- застосування захисних екранів, свинцевого одягу для персоналу;
- дотримання дозових лімітів: ≤ 20 мЗв/рік (згідно НРБУ-97).

Таблиця 4.2 – Аналіз ризиків та заходи безпеки

Ризик	Засіб контролю або попередження
Перегрів ПК, коротке замикання	Системне охолодження, УЗО
Витік даних пацієнтів	Паролі, обмеження доступу, шифрування
Помилки сегментації через шум	Використання фільтрів, стабільне освітлення
Втома оператора, помилки в діагностиці	Інтерфейс з підказками, обмеження по часу

4.6 Висновки до розділу 4

У розділі проведено комплексну оцінку умов праці оператора програмної системи аналізу КТ-зображень. Надано обґрунтовані рекомендації з організації безпечного робочого місця, заходів електро- та пожежної безпеки, ергономіки та захисту від професійних ризиків. Врахування зазначених положень забезпечує відповідність системи чинним санітарним, технічним і безпековим нормам при її експлуатації.

ЗАГАЛЬНІ ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було успішно реалізовано повний цикл проектування, моделювання та тестування біотехнічної системи для аналізу комп'ютерних томограм органів дихання. Розроблена система відповідає сучасним вимогам до швидкодії, точності діагностики, автоматизації та безпеки.

У ході роботи:

1. Проаналізовано існуючі КТ-системи та методи аналізу зображень, виявлено переваги глибокого навчання, зокрема U-Net, для задач сегментації легенів та розпізнавання патологій.

2. Обґрунтовано вибір технічного та програмного середовища. Для реалізації було використано MATLAB з App Designer, Image Processing Toolbox і Deep Learning Toolbox.

3. Розроблено математичне забезпечення, зокрема модель реконструкції томографічного зображення методом зворотної проекції (FBP), а також імітаційні моделі з використанням фантомів.

4. Реалізовано графічний користувацький інтерфейс (GUI) для зручної взаємодії з користувачем, з можливістю обробки як окремих зрізів, так і всього томографічного об'єму.

5. Інтегровано попередньо навчену нейронну мережу U-Net для сегментації легеневої тканини з подальшою класифікацією змін.

6. Проведено тестування системи на відкритих медичних даних. Отримано задовільні результати точності ($\text{IoU} > 0.85$), продуктивності (обробка об'єму < 40 сек.) та стабільності при обробці реальних зображень.

7. Розглянуто питання безпеки життєдіяльності та охорони праці, з урахуванням ергономіки, електробезпеки, пожежної безпеки та факторів, пов'язаних із медичним середовищем.

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

Практична цінність роботи полягає в можливості впровадження створеного прототипу як навчального, діагностичного або дослідницького модуля в медичних та інженерних закладах. Розроблена система може бути масштабована, доопрацьована для мобільного використання або інтегрована з PACS-системами у клінічній практиці.

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хвостівський М.О., Яворська Є.Б., Дозорський В.Г. Методичні рекомендації до виконання, оформлення та захисту кваліфікаційних робіт для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 163 «Біомедична інженерія» / уклад.: Хвостівський М.О., Яворська Є.Б., Дозорський В.Г. Тернопіль: ТНТУ, 2024. 57 с.
2. ДСТУ ISO 13485:2018. Вироби медичні. Системи управління якістю. Вимоги для нормативних цілей. – [Чинний з 01.07.2019].
3. ДБН В.2.2-28:2018. Будівлі і споруди. Заклади охорони здоров'я. – Київ: Мінрегіон України, 2019. – 75 с.
4. ДСанПіН 3.3.2.007-98. Державні санітарні норми і правила з організації робочих місць ПК. – Київ: МОЗ України, 1998. – 45 с.
5. Головач А.І., Матвієнко С.О. Медичні інформаційні системи: навч. посібник. – Київ: КНУБА, 2020. – 172 с.
6. Яценко О.Л., Михайлова І.М. Основи візуалізації в медичній діагностиці: навч. посібник. – Вінниця: Нова Книга, 2018. – 256 с.
7. Gonzalez R.C., Woods R.E. Digital Image Processing. – 4th ed. – New York: Pearson, 2018. – 1176 p.
8. Szeliski R. Computer Vision: Algorithms and Applications. – 2nd ed. – Springer, 2022. – 930 p.
9. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // Proc. of MICCAI 2015. – LNCS, vol. 9351. – pp. 234–241.
10. Lundervold A.S., Lundervold A. An overview of deep learning in medical imaging focusing on MRI // Zeitschrift für Medizinische Physik. – 2019. – Vol. 29(2). – pp. 102–127. DOI: 10.1016/j.zemedi.2018.11.002

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

11. Armato S.G. et al. The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans // Medical Physics. – 2011. – Vol. 38(2). – pp. 915–931.
12. Morozov S. et al. MosMedData: Chest CT Scans With COVID-19 Related Findings Dataset. – arXiv:2005.06465 [eess.IV]. – 2020.
13. MathWorks. Deep Learning Toolbox User's Guide. – The MathWorks, Inc., 2023. – [Електронний ресурс]. – Режим доступу: <https://www.mathworks.com/help/deeplearning/>
14. MathWorks. Image Processing Toolbox User's Guide. – The MathWorks, Inc., 2023. – [Електронний ресурс]. – Режим доступу: <https://www.mathworks.com/help/images/>
15. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної бо та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.
- 16.

Додатки

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

Лістинги коду

A.1 Базова структура застосунку CTAnalysisApp у MATLAB App Designer

```

% App Designer code structure (simplified logic)

classdef CTAnalysisApp < matlab.apps.AppBase

    % Properties corresponding to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        LoadButton        matlab.ui.control.Button
        ImageAxes          matlab.ui.control.UIAxes
        SliceSlider        matlab.ui.control.Slider
        SegmentButton      matlab.ui.control.Button
        ExportButton       matlab.ui.control.Button
    end

    properties (Access = private)
        VolumeData        double
        CurrentSlice       double = 1
        SegmentMask        logical
    end

    methods (Access = private)

        function loadDICOM(app)
            folder = uigetdir();
            if folder == 0, return; end
            files = dir(fullfile(folder, '*.dcm'));
            files = natsortfiles(files);
            for i = 1:length(files)
                app.VolumeData(:, :, i) =
dicomread(fullfile(folder, files(i).name));
            end
            app.VolumeData = double(app.VolumeData);
            app.CurrentSlice = 1;
            app.updateImageDisplay();
        end

        function updateImageDisplay(app)
            img = app.VolumeData(:, :, app.CurrentSlice);
            img = mat2gray(img);
            imshow(img, 'Parent', app.ImageAxes);
        end
    end
end

```

```

        function segmentCurrentSlice(app)
            img =
mat2gray(app.VolumeData(:,:,app.CurrentSlice));
            bw = imbinarize(img, graythresh(img));
            bw = imfill(bw, 'holes');
            bw = bwareaopen(bw, 500);
            app.SegmentMask = bw;
            imshowpair(img, bw, 'blend', 'Parent',
app.ImageAxes);
        end

        function exportImage(app)
            [file,path] = uiputfile('*.png');
            if isequal(file,0), return; end
            exportgraphics(app.ImageAxes, fullfile(path,
file), 'Resolution', 300);
        end

        function sliderMoved(app, event)
            app.CurrentSlice = round(app.SliceSlider.Value);
            app.updateImageDisplay();
        end
    end

    % Component initialization
    methods (Access = private)

        function startupFcn(app)
            app.SliceSlider.Limits = [1 100];
        end

        function createComponents(app)
            % (GUI construction code not included here)
        end
    end

    % App creation and deletion
    methods (Access = public)

        function app = CTAnalysisApp
            createComponents(app)
            registerApp(app, app.UIFigure)
            runStartupFcn(app, @startupFcn)
        end

        function delete(app)
            delete(app.UIFigure)
        end
    end
end
end

```

A.2 Функція побудови 3D-моделі томографічних даних

```
% App Designer code structure (simplified logic)

classdef CTAnalysisApp < matlab.apps.AppBase

    % Properties corresponding to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        LoadButton              matlab.ui.control.Button
        ImageAxes                matlab.ui.control.UIAxes
        SliceSlider              matlab.ui.control.Slider
        SegmentButton            matlab.ui.control.Button
        ExportButton             matlab.ui.control.Button
        Volume3DButton           matlab.ui.control.Button
        UIAxes3D                 matlab.ui.control.UIAxes
    end

    properties (Access = private)
        VolumeData              double
        CurrentSlice             double = 1
        SegmentMask              logical
    end

    methods (Access = private)

        function loadDICOM(app)
            folder = uigetdir();
            if folder == 0, return; end
            files = dir(fullfile(folder, '*.dcm'));
            files = natsortfiles(files);
            for i = 1:length(files)
                app.VolumeData(:, :, i) =
dicomread(fullfile(folder, files(i).name));
            end
            app.VolumeData = double(app.VolumeData);
            app.CurrentSlice = 1;
            app.SliceSlider.Limits = [1
size(app.VolumeData, 3)];
            app.updateImageDisplay();
        end

        function updateImageDisplay(app)
            img = app.VolumeData(:, :, app.CurrentSlice);
            img = mat2gray(img);
            imshow(img, 'Parent', app.ImageAxes);
        end

        function segmentCurrentSlice(app)
            img =
mat2gray(app.VolumeData(:, :, app.CurrentSlice));
```

```

        bw = imbinarize(img, graythresh(img));
        bw = imfill(bw, 'holes');
        bw = bwareaopen(bw, 500);
        app.SegmentMask = bw;
        imshowpair(img, bw, 'blend', 'Parent',
app.ImageAxes);
    end

    function exportImage(app)
        [file,path] = uiputfile('*.png');
        if isequal(file,0), return; end
        exportgraphics(app.ImageAxes, fullfile(path,
file), 'Resolution', 300);
    end

    function sliderMoved(app, event)
        app.CurrentSlice = round(app.SliceSlider.Value);
        app.updateImageDisplay();
    end

    function show3DModel(app)
        if isempty(app.VolumeData), return; end
        vol = mat2gray(app.VolumeData);
        axes(app.UIAxes3D);
        cla(app.UIAxes3D);
        p = patch(isosurface(vol, 0.5), 'Parent',
app.UIAxes3D);
        isonormals(vol, p);
        p.FaceColor = 'red';
        p.EdgeColor = 'none';
        daspect(app.UIAxes3D, [1 1 1]);
        view(app.UIAxes3D, 3);
        camlight(app.UIAxes3D);
        lighting(app.UIAxes3D, 'gouraud');
    end
end

methods (Access = private)
    function startupFcn(app)
        app.SliceSlider.Limits = [1 100];
    end

    function createComponents(app)
        % (GUI construction code not included here)
    end
end

methods (Access = public)
    function app = CTAnalysisApp
        createComponents(app)
        registerApp(app, app.UIFigure)
        runStartupFcn(app, @startupFcn)
    end
end

```

```

        end

        function delete(app)
            delete(app.UIFigure)
        end
    end
end
end

```

Примітка:

- А) Кнопка Volume3DButton викликає метод show3DModel
- Б) Візуалізація реалізована через isosurface, patch, camlight
- В) Працює з нормалізованим об'ємом (mat2gray)

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

А.3 Інтеграція з U-Net для сегментації легенів та класифікацію патологій

```
% App Designer code structure (simplified logic)

classdef CTAnalysisApp < matlab.apps.AppBase

    % Properties corresponding to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        LoadButton              matlab.ui.control.Button
        ImageAxes                matlab.ui.control.UIAxes
        SliceSlider              matlab.ui.control.Slider
        SegmentButton            matlab.ui.control.Button
        ExportButton              matlab.ui.control.Button
        Volume3DButton           matlab.ui.control.Button
        ClassifyButton            matlab.ui.control.Button
        UIAxes3D                 matlab.ui.control.UIAxes
        ResultLabel              matlab.ui.control.Label
    end

    properties (Access = private)
        VolumeData               double
        CurrentSlice              double = 1
        SegmentMask               logical
        Net                       dlnetwork
    end

    methods (Access = private)

        function loadDICOM(app)
            folder = uigetdir();
            if folder == 0, return; end
            files = dir(fullfile(folder, '*.dcm'));
            files = natsortfiles(files);
            for i = 1:length(files)
                app.VolumeData(:, :, i) =
dicomread(fullfile(folder, files(i).name));
            end
            app.VolumeData = double(app.VolumeData);
            app.CurrentSlice = 1;
            app.SliceSlider.Limits = [1 size(app.VolumeData, 3)];
            app.updateImageDisplay();
        end

        function updateImageDisplay(app)
            img = app.VolumeData(:, :, app.CurrentSlice);
            img = mat2gray(img);
            imshow(img, 'Parent', app.ImageAxes);
        end

        function segmentCurrentSlice(app)
```

```

        img = mat2gray(app.VolumeData(:,:,app.CurrentSlice));
        if isempty(app.Net)
            app.Net = load("unetPretrained.mat").net; %
Попередньо навчена модель U-Net
        end
        imgResized = imresize(img, [256 256]);
        segmented = semanticseg(imgResized, app.Net);
        mask = (segmented == 'lung');
        mask = imresize(mask, size(img));
        app.SegmentMask = mask;
        imshowpair(img, mask, 'blend', 'Parent',
app.ImageAxes);
    end

function classifyPathology(app)
    if isempty(app.SegmentMask), return; end
    features = regionprops(app.SegmentMask, 'Area',
'Eccentricity');
    % Простий приклад класифікації за площею
    if features.Area > 5000
        diagnosis = 'Підозра на інфільтративні зміни';
    else
        diagnosis = 'Норма або помірні зміни';
    end
    app.ResultLabel.Text = ['Результат: ', diagnosis];
end

function exportImage(app)
    [file,path] = uiputfile('*.png');
    if isequal(file,0), return; end
    exportgraphics(app.ImageAxes, fullfile(path, file),
'Resolution', 300);
end

function sliderMoved(app, event)
    app.CurrentSlice = round(app.SliceSlider.Value);
    app.updateImageDisplay();
end

function show3DModel(app)
    if isempty(app.VolumeData), return; end
    vol = mat2gray(app.VolumeData);
    axes(app.UIAxes3D);
    cla(app.UIAxes3D);
    p = patch(isosurface(vol, 0.5), 'Parent',
app.UIAxes3D);
    isonormals(vol, p);
    p.FaceColor = 'red';
    p.EdgeColor = 'none';
    daspect(app.UIAxes3D, [1 1 1]);
    view(app.UIAxes3D, 3);
    camlight(app.UIAxes3D);

```

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

```

        lighting(app.UIAxes3D, 'gouraud');
    end
end

methods (Access = private)
    function startupFcn(app)
        app.SliceSlider.Limits = [1 100];
    end

    function createComponents(app)
        % (GUI construction code not included here)
    end
end

methods (Access = public)
    function app = CTAnalysisApp
        createComponents(app)
        registerApp(app, app.UIFigure)
        runStartupFcn(app, @startupFcn)
    end

    function delete(app)
        delete(app.UIFigure)
    end
end
end
end

```

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		

А.4 Обробка всього об'єму томографічних даних

```
% App Designer code structure (simplified logic)

classdef CTAnalysisApp < matlab.apps.AppBase

    % Properties corresponding to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        LoadButton              matlab.ui.control.Button
        ImageAxes                matlab.ui.control.UIAxes
        SliceSlider              matlab.ui.control.Slider
        SegmentButton            matlab.ui.control.Button
        ExportButton             matlab.ui.control.Button
        Volume3DButton           matlab.ui.control.Button
        ClassifyButton           matlab.ui.control.Button
        UIAxes3D                 matlab.ui.control.UIAxes
        ResultLabel              matlab.ui.control.Label
        BatchSegmentButton       matlab.ui.control.Button
    end

    properties (Access = private)
        VolumeData              double
        CurrentSlice             double = 1
        SegmentMask              logical
        Net                      dlnetwork
        VolumeMask               logical
    end

    methods (Access = private)

        function loadDICOM(app)
            folder = uigetdir();
            if folder == 0, return; end
            files = dir(fullfile(folder, '*.dcm'));
            files = natsortfiles(files);
            for i = 1:length(files)
                app.VolumeData(:, :, i) =
dicomread(fullfile(folder, files(i).name));
            end
            app.VolumeData = double(app.VolumeData);
            app.CurrentSlice = 1;
            app.SliceSlider.Limits = [1 size(app.VolumeData, 3)];
            app.updateImageDisplay();
        end

        function updateImageDisplay(app)
            img = app.VolumeData(:, :, app.CurrentSlice);
            img = mat2gray(img);
            imshow(img, 'Parent', app.ImageAxes);
        end
    end
end
```

```

end

function segmentCurrentSlice(app)
    img = mat2gray(app.VolumeData(:,:,app.CurrentSlice));
    if isempty(app.Net)
        app.Net = load("unetPretrained.mat").net;
    end
    imgResized = imresize(img, [256 256]);
    segmented = semanticseg(imgResized, app.Net);
    mask = (segmented == 'lung');
    mask = imresize(mask, size(img));
    app.SegmentMask = mask;
    imshowpair(img, mask, 'blend', 'Parent',
app.ImageAxes);
end

function segmentVolume(app)
    if isempty(app.Net)
        app.Net = load("unetPretrained.mat").net;
    end
    sz = size(app.VolumeData);
    app.VolumeMask = false(sz);
    for i = 1:sz(3)
        img = mat2gray(app.VolumeData(:,:,i));
        imgResized = imresize(img, [256 256]);
        segmented = semanticseg(imgResized, app.Net);
        mask = (segmented == 'lung');
        mask = imresize(mask, sz(1:2));
        app.VolumeMask(:,:,i) = mask;
    end
    msgbox('Сегментацію об'єму завершено', 'Готово');
end

function classifyPathology(app)
    if isempty(app.SegmentMask), return; end
    features = regionprops(app.SegmentMask, 'Area',
'Eccentricity');
    if features.Area > 5000
        diagnosis = 'Підозра на інфільтративні зміни';
    else
        diagnosis = 'Норма або помірні зміни';
    end
    app.ResultLabel.Text = ['Результат: ', diagnosis];
end

function exportImage(app)
    [file,path] = uiputfile('*.png');
    if isequal(file,0), return; end
    exportgraphics(app.ImageAxes, fullfile(path, file),
'Resolution', 300);
end

```

```

function sliderMoved(app, event)
    app.CurrentSlice = round(app.SliceSlider.Value);
    app.updateImageDisplay();
end

function show3DModel(app)
    if isempty(app.VolumeData), return; end
    vol = mat2gray(app.VolumeData);
    axes(app.UIAxes3D);
    cla(app.UIAxes3D);
    p = patch(isosurface(vol, 0.5), 'Parent',
app.UIAxes3D);
    isonormals(vol, p);
    p.FaceColor = 'red';
    p.EdgeColor = 'none';
    daspect(app.UIAxes3D, [1 1 1]);
    view(app.UIAxes3D, 3);
    camlight(app.UIAxes3D);
    lighting(app.UIAxes3D, 'gouraud');
end
end

methods (Access = private)
    function startupFcn(app)
        app.SliceSlider.Limits = [1 100];
    end

    function createComponents(app)
        % (GUI construction code not included here)
    end
end

methods (Access = public)
    function app = CTAnalysisApp
        createComponents(app)
        registerApp(app, app.UIFigure)
        runStartupFcn(app, @startupFcn)
    end

    function delete(app)
        delete(app.UIFigure)
    end
end
end
end

```

Примітка:

- А) Метод `segmentVolume` автоматично сегментує всі зрізи за допомогою U-Net
- Б) Результат зберігається у змінній `VolumeMask`
- В) Додано кнопку `BatchSegmentButton` для виклику цього процесу

					КРБ.163.21-071.001 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дат		