

УДК 537.32.

DOI 10.36910/775.24153966.2021.72.10

Б.О. Решетило¹, В.В. Лотиш¹, О.К. Яковенко²¹Луцький національний технічний університет²КП «Волинська обласна клінічна лікарня» Волинської облради**ВІЗУАЛІЗАЦІЯ ТА ОБРОБКА МЕДИЧНИХ ЗОБРАЖЕНЬ ФОРМАТУ DICOM ЗАСОБАМИ PYTHON**

У статті досліджено сучасні алгоритми обробки медичних зображень формату DICOM засобами Python. Розглянуто проблему низької якості медичного зображення. Висвітлено переваги обраних методів дослідження та їх алгоритм дій.

Ключові слова: медична візуалізація, методи візуалізації, формат Dicom, бібліотеки numpy, scipy, matplotlib, шкала Хаунсфілда, метод вододілу Watershed Transform, Python.

Б.А. Решетило, В.В. Лотыш, О.К. Яковенко**ВИЗУАЛИЗАЦИЯ И ОБРАБОТКА МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ ФОРМАТА DICOM СРЕДСТВАМИ PYTHON**

В статье исследованы современные алгоритмы обработки медицинских изображений формата DICOM средствами Python. Рассмотрена проблема низкого качества медицинского изображения. Освещены преимущества выбранных методов исследования и их алгоритм действий.

Ключевые слова: медицинская визуализация, методы визуализации, формат Dicom, библиотеки numpy, scipy, matplotlib, шкала Хаунсфилда, метод водораздела Watershed Transform, Python.

В.А. Reshetilo, V.V. Lotysh, O.K. Yakovenko**VISUALIZATION AND PROCESSING OF MEDICAL IMAGES IN DICOM FORMAT WITH PYTHON MEANS**

The article reflects modern methods of processing medical images of DICOM format using Python. The problem of low quality of the studied medical image is posed and solved. The advantages of the selected research methods and their algorithm of actions are reflected.

Keywords: medical imaging, imaging techniques, Dicom format, numpy, scipy, matplotlib libraries, Hounsfield scale, Watershed Transform watershed method, Python.

Постановка проблеми. Технології покращення медичного рентгенологічного зображення дали великий поштовх в реалізації сучасного методу діагностики - комп'ютерної томографії високої роздільної здатності (КТВРЗ), яка дозволяє лікарю точніше визначити стан будь-якого органу людини, його розміри, положення, форму, стан поверхні, його функції та щільність будь-якої тканини.

На сьогоднішній день КТВРЗ будь-якого органу людини, в тому числі органів грудної клітки (ОГК) – це один з провідних методів точного встановлення діагнозу, проте, однією з проблем, з якою стикається лікар під час інтерпретації рентгенологічного зображення, є іноді низька якість самого зображення яка зумовлена різними факторами. Зростання обсягів інформації яка зберігається, обробляється і перетворюється, утворюють неодмінну умову в створенні та вдосконаленні методів сучасної діагностики захворювань ОГК, особливо в умовах існуючої пандемії коронавірусної хвороби-2019 (COVID-19) [9].

З розвитком сучасної медичної діагностики вимоги до стандартів візуалізації внутрішніх біологічних структур організму людини постійно зростають, що призвело до розгалуження візуалізації на різні види для точнішого виявлення не лише морфологічних, а й функціональних змін, які відбуваються в здоровому організмі та при різних захворюваннях.

В теперішній час через плівку чи екран лікарі бачать стан органу людини не лише у двовимірному, а й в трьохвимірному вигляді. Віртуально повертаючи, є можливість роздивитись будь-який ракурс органу навіть в реальному часі. Це дало можливість якісніше проводити діагностику з наступним лікуванням різних захворювань хірургічними та не хірургічними методами [1].

У даній статті наведено сучасні методи обробки медичного зображення КТВРЗ ОГК, послідовність дій та засоби вдосконалення результатів візуалізації на основі яких створено програмне забезпечення.

Аналіз останніх досліджень і публікацій. Визначаючи конкретніше дане питання, слід зазначити, що розгляд методів обробки зображення висвітлено в працях Н.В. Туманська, К.С.

Барська, С.В.Скринченко, Т.М. Кічангіна, Прэтт У., Рисована Л.М., Радзішевська Є.Б. та ін. Окремі аспекти роботи з форматом Dicom та теоретичний, практичний курс обробки зображень за допомогою бібліотек мови Python: numpy, matplotlib, skimage, відображаються в роботах Mustra Mario, Delac Kresimir, Grgic Mislav, Van derWalt, S., Colbert, S. C. & Varoquaux, Travis E. O. та ін..

Метою статті є огляд існуючих алгоритмів для обробки медичного зображення КТВРЗ ОГК з використанням програмного середовища Python.

Основна частина. Одним з найпопулярніших загальних форматів зображення та стандартом створення, збереження, архівування, передачі даних та візуалізації медичних зображень є DICOM.

Основною особливістю цього стандарту є те, що він зберігає велику кількість даних про пацієнта, місце, час та методику дослідження [2].

Стандарт DICOM має безліч можливостей ідентифікувати та пов'язувати конкретні набори даних із багатьма різними медичними структурами, такими як інформація про установи чи інформація про відповідальність [3].

Це поштовх в розвитку сфери медичного програмного забезпечення, що дає можливість візуалізувати дані про дослідження організму пацієнта для подальшого лікування та отримати файл стандарту DICOM відразу після сканування.

Стандартний файл формату Dicom містить в собі загальну інформацію, дані про пацієнта, характерну особливість апарату обстеження, назву медичного закладу, інформація про діагностуючого лікаря, вид діагностування, параметри обстеження та певні його особливості, унікальні ключі ідентифікації, цифрове зображення збережене в файлі формату DICOM [4].

Для розробки програмного забезпечення обрано мову програмування Python через певні її особливості: простота читання та написання коду, вирішення різних завдань обробки зображень за допомогою певних пакетів, зрізи масивів.

Для роботи з форматом Dicom обрано бібліотеки: pydicom, numpy, pandas, matplotlib, scipy. Кожна є унікальною в сфері свого застосування.

Перші три етапи для візуалізації медичного зображення: загрузка файлу, перетворення параметрів в певний формат та отримання масиву з певних точок зрізу, проводяться за допомогою бібліотеки pydicom.

Для наступних етапів обробки медичних зображень використовують наступні бібліотеки.

Найбільш популярною є numpy. Цей модуль використовується для роботи з числами, таблицями, матрицями в різних форматах, для аналізу даних у вигляді швидких функцій. Пакет numpy підходить для цифрової обробки зображень, яка базується на маніпулюванні 2D піксельними масивами. Numpy підтримує багатовимірні масиви та надає численні функції для загальних завдань з обробкою зображення. Завдяки певним ознакам: використання алгоритмів, мала кількість циклів для полегшеної індексації ітерацій, чіткий код, складні функції, наявність інструментів для інтеграції та засоби лінійної алгебри, numpy є базовим пакетом фактичного стандарту для багатовимірних масивів у Python [5].

Для маніпулювання даними з величезними можливостями: швидкість проведення досліджень нових даних, тестування гіпотез, отримання звіту, використовують бібліотеку pandas.

Ця бібліотека є однією з головних переваг Python, оскільки вона забезпечує роботу даними та функції для роботи зі структурованими даними. Це найкращий інструмент для запису, читання даних та роботою з великою кількістю табличних даних формату csv.

За допомогою модуля Pandas можливі наступні дії: індексування, маніпулювання, перейменування, сортування, об'єднання фрейму даних; відновлення, додавання, видалення стовпці з фрейму даних; відновлення відсутніх файлів; побудова гістограм. Це робить Pandas незамінною бібліотекою в застосуванні Python для Data Science. Застосування Pandas спрощує представлення зображення, представлених набором даних, методами візуалізації.

Для візуалізації даних двовимірної і тривимірної графіки, представлених в масивах numpy використовують модуль matplotlib [6].

Бібліотекою числових процедур для мови програмування Python, яка забезпечує базові функції для моделювання є scipy. Цей набір інструментів включає алгоритми оптимізації, інтегрування, інтерполяції, задачі власних значень, алгебраїчні рівняння, диференціальні рівняння, підтримують великі багатовимірні масиви та забезпечують численні функції для загальних завдань обробки зображень [7].

Наведені вище бібліотеки є основними засобами для вирішення таких завдань: візуалізації

2D та 3D зображення, робота з масивами даних, функцію моделювання.

За допомогою бібліотеки `pydicom` зчитується всі дані медичного зображення, `numpy` - перетворює зображення в піксельний масив, `pandas` – здійснює запис та маніпуляції, `matplotlib` – для візуалізації графічно, `scipy` – для моделювання.

Розглянемо можливість застосування вищеперелічених модулів для роботи з зображенням грудної клітки людини отриманих комп'ютерним томографом (КТ) і збереженим у форматі DICOM.

Першим етапом обробки медичного зображення грудної клітки є отримання метаданих DICOM.

Для редагування всіх полів метаданих DICOM використовуємо пакет `pydicom` з підключенням модуля `os` для роботи з операційною системою. Вибираємо каталог з файлом формату `Dicom`: `filepath = './data/'`. Вказуємо індивідуальний префікс файлу DICOM перед номером: `dcmprefix = 'Image_'`. Для перевірки полів метаданих вибираємо перше зображення DICOM: `firstdcm = dcmprefix + '%04d.DICOM' %1`.

Нові поля метаданих зберігаємо в нову директорію: `newdir = './data-edited/'`.

Для подальшої роботи використовуємо цикл з методом `os.listdir` для отримання списку всіх файлів у вказаному каталозі з заданою умовою методом `endswith()`, який повертає `true`, якщо рядок закінчується вказаним форматом файлу, інакше повертає `false`.

Завантажуємо набір даних DICOM (`ds`): `ds = dicom.read_file(filepath + firstdcm)`.

Результат етапу представлено на рис. 2.

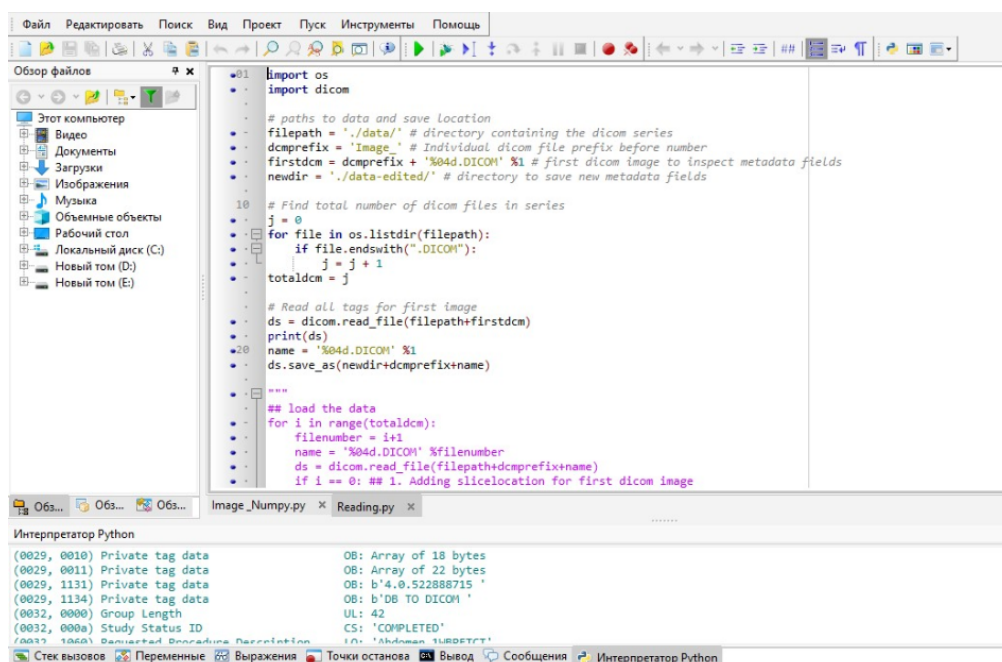
Другим етапом є нормалізація, яка надає медичним зображенням контрастності та усуває залежність зображення від освітлення та кольору.

Для нормалізації зображення формату DICOM імпортуємо такі бібліотеки як `opencv`, `numpy`, `matplotlib`.

Задаємо відповідні значення вікна по центру та ширині командами `window_center`, `window_width`.

Вибираємо файл певного формату командою `dicom_path` та завантажуюмо зріз КТ зображення операцією `slice = dicom.read_file(dicom_path)`.

Завантажений зріз перетворюємо в піксельний масив з змінним масштабом операціями: `s = int(slice.RescaleSlope)`; `b = int(slice.RescaleIntercept)`; `image = s * slice.pixel_array + b`. Використовуючи операцію бібліотеки `matplotlib`: `plt.subplots`, `plt.title('DICOM ->array')`, `plt.imshow(image, cmap='gray')` — показали перетворене зображення на графіку у відтинках сірої карти.



```

Файл Редактировать Поиск Вид Проект Пуск Инструменты Помощь
Обзор файлов
Этот компьютер
  Видео
  Документы
  Загрузки
  Изображения
  Музыка
  Объемные объекты
  Рабочий стол
  Локальный диск (C:)
  Новый том (D:)
  Новый том (E:)
01 Import os
import dicom

# paths to data and save location
filepath = './data/' # directory containing the dicom series
dcmprefix = 'Image_' # individual dicom file prefix before number
firstdcm = dcmprefix + '%04d.DICOM' %1 # first dicom image to inspect metadata fields
newdir = './data-edited/' # directory to save new metadata fields

# Find total number of dicom files in series
j = 0
for file in os.listdir(filepath):
    if file.endswith('.DICOM'):
        j = j + 1
totaldcm = j

# Read all tags for first image
ds = dicom.read_file(filepath+firstdcm)
print(ds)
20 name = '%04d.DICOM' %1
ds.save_as(newdir+dcmprefix+name)

###
## load the data
for i in range(totaldcm):
    filename = i+1
    name = '%04d.DICOM' %filename
    ds = dicom.read_file(filepath+dcmprefix+name)
    if i == 0: ## 1. Adding slicelocation for first dicom image

Интерпретатор Python
(0029, 0010) Private tag data      OB: Array of 18 bytes
(0029, 0011) Private tag data      OB: Array of 22 bytes
(0029, 1131) Private tag data      OB: b'4.0.522888715 '
(0029, 1134) Private tag data      OB: b'DB TO DICOM '
(0032, 0000) Group Length         UL: 42
(0032, 0000) Study Status ID      CS: 'COMPLETED'
(0033, 1000) Dispatched BoneDensity Description  LO: 'Abdomen_11BDETT'
Стек вызовов Переменные Выражения Точки останова Вывод Сообщения Интерпретатор Python
  
```

Рис. 1. Фрагмент коду в IDE редакторі

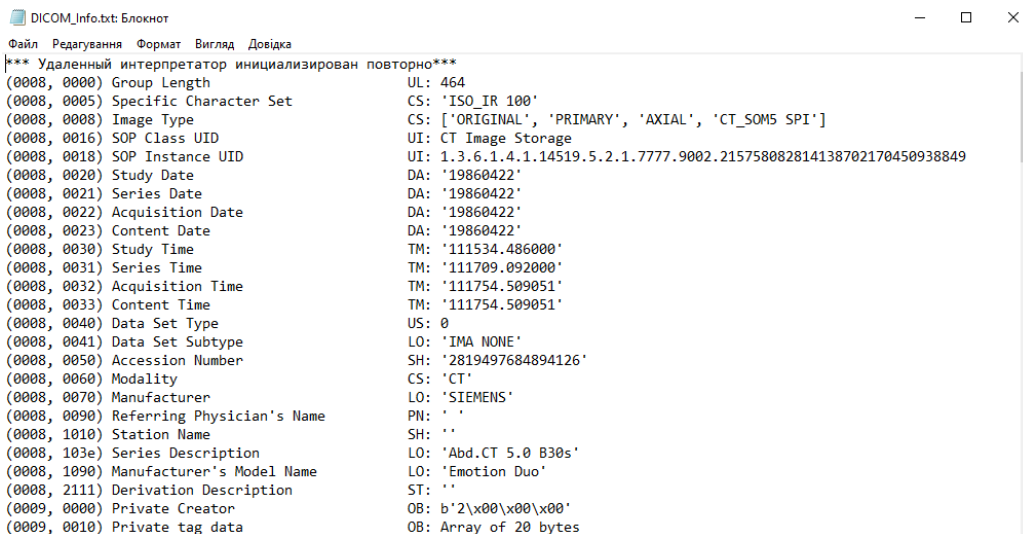
Над отриманим зображенням здійснюємо останній етап: нормалізація, використовуючи операцію бібліотеки `numpy` — `numpy.clip` для відсікання значень у масиві. Нормалізоване

результуюче зображення виводимо разом з операціями `matplotlib.pyplot.subplots`, `plt.title`, `plt.imshow`. Інтерфейс програми та виведені зображення грудної клітки наведено на рисунках 3-5.

Тканини з меншою щільністю будуть перекриті чорним кольором, тканини з більшою густиною - білим.

Для реалізації цього етапу до отриманого цифрового зображення: КТВРЗ ОГК, завантажуюємо бібліотеки Python: `dicom`, `numpy`, `matplotlib`, `tkinter`. Після чого оцифруємо зображення в піксельний масив `512x512` та зберігаємо його на диск. За допомогою операції `get_pixels_hu` перетворюємо необроблені значення в одиниці HU.

В даній роботі зображення — це цифрові масиви `numpy`, як підтримують тип даних `astype`, тому для того, щоб уникнути спотворення використовуємо діапазони `int16` (`image=image.astype(np.int16)`; `image[image<=-2000]=0`). Наступним кроком є представлення вікна зрізу функціями бібліотеки `matplotlib` та зображення гістограми з одиницями вимірювання та частотою для значення шкали HU від `-1000` до `+1000` (рис. 6).

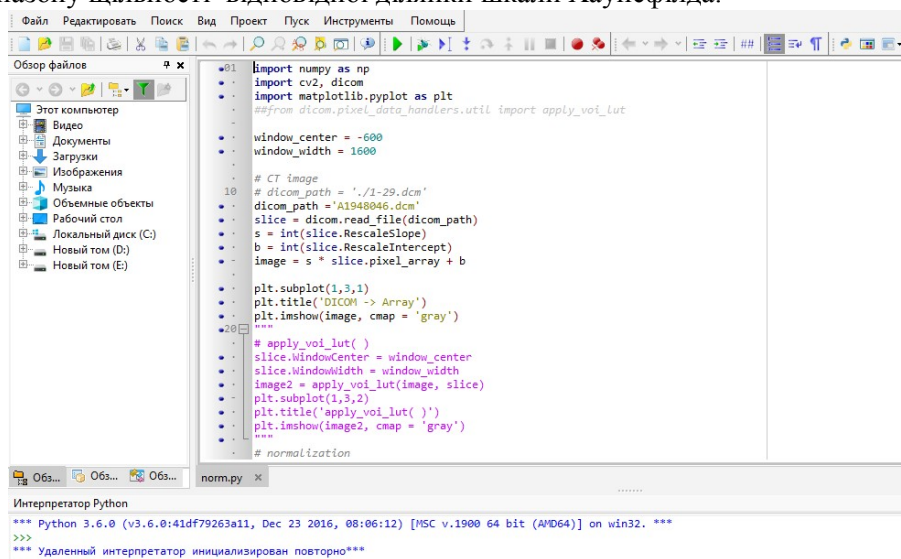


```

DICOM_Info.txt: Блокнот
Файл Редагування Формат Видгляд Довідка
*** Удаленный интерпретатор инициализирован повторно***
(0008, 0008) Group Length          UL: 464
(0008, 0005) Specific Character Set CS: 'ISO_IR 100'
(0008, 0008) Image Type            CS: ['ORIGINAL', 'PRIMARY', 'AXIAL', 'CT_SOM5 SPI']
(0008, 0016) SOP Class UID         UI: CT Image Storage
(0008, 0018) SOP Instance UID      UI: 1.3.6.1.4.1.14519.5.2.1.7777.9002.215758082814138702170450938849
(0008, 0020) Study Date            DA: '19860422'
(0008, 0021) Series Date           DA: '19860422'
(0008, 0022) Acquisition Date      DA: '19860422'
(0008, 0023) Content Date          DA: '19860422'
(0008, 0030) Study Time            TM: '111534.486000'
(0008, 0031) Series Time           TM: '111709.092000'
(0008, 0032) Acquisition Time      TM: '111754.509051'
(0008, 0033) Content Time         TM: '111754.509051'
(0008, 0040) Data Set Type         US: 0
(0008, 0041) Data Set Subtype      LO: 'IMA NONE'
(0008, 0050) Accession Number      SH: '2819497684894126'
(0008, 0060) Modality              CS: 'CT'
(0008, 0070) Manufacturer          LO: 'SIEMENS'
(0008, 0090) Referring Physician's Name PN: ''
(0008, 1010) Station Name          SH: ''
(0008, 103e) Series Description    LO: 'Abd.CT 5.0 B30s'
(0008, 1090) Manufacturer's Model Name LO: 'Emotion Duo'
(0008, 2111) Derivation Description ST: ''
(0009, 0000) Private Creator       OB: b'2\x00\x00\x00'
(0009, 0010) Private tag data      OB: Array of 20 bytes
  
```

Рис. 2. Вивід метаданих Dicom

Шкала Хаунсфілда включає в себе 4096 значень — від `-1024` до `+3071` одиниць Хаунсфілда (HU). За допомогою вікна візуалізації здійснюється зміна яскравості і контрастності КТ зображень таким чином, щоб на екрані монітора в шкалі сірого кольору видно тільки тканини необхідного діапазону щільності відповідної ділянки шкали Хаунсфілда.



```

import numpy as np
import cv2, dicom
import matplotlib.pyplot as plt
#from dicom.pixel_data_handlers.util import apply_voi_lut

window_center = -600
window_width = 1600

# CT image
dicom_path = './1-29.dcm'
dicom_path = 'A1948046.dcm'
slice = dicom.read_file(dicom_path)
s = int(slice.RescaleSlope)
b = int(slice.RescaleIntercept)
image = s * slice.pixel_array + b

plt.subplot(1,3,1)
plt.title('DICOM -> Array')
plt.imshow(image, cmap = 'gray')

# apply_voi_lut()
slice.WindowCenter = window_center
slice.WindowWidth = window_width
image2 = apply_voi_lut(image, slice)
plt.subplot(1,3,2)
plt.title('apply_voi_lut()')
plt.imshow(image2, cmap = 'gray')

# normalization
  
```

Рис. 3. Фрагмент коду нормалізації в IDE редакторі

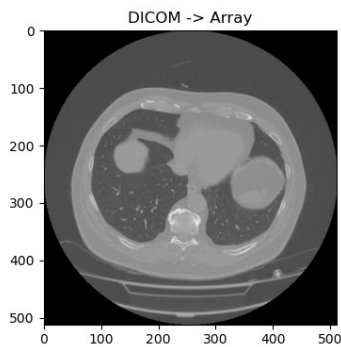


Рис. 4. Початкове зображення

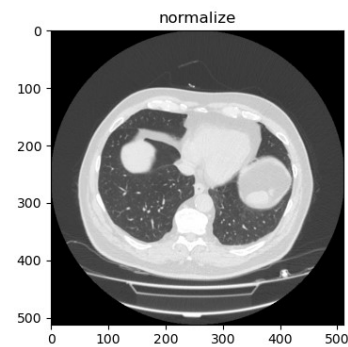


Рис. 5. Нормалізоване зображення

Наступним кроком виводимо зображення формату DICOM в одиницях HU з накладанням двох карт: сірої та теплової для кращої візуалізації. (`fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(18,8)); ax= axs[0] ; ax.imshow(first_patient_pixels, cmap='gray', vmin=vmin, vmax=vmax); ax.set_title("HU", fontsize=20); ax.set_axis_off()`). Даний результат наведено на рисунках 7-8. Щоб побачити певну структуру організму необхідно та достатньо вести мінімальне та максимальне значення по шкалі HU. Використовуючи кісткову тканину з $HU_{MIN}=400$ ТА $HU_{max}=600$ візьмемо її за основу для візуалізації.

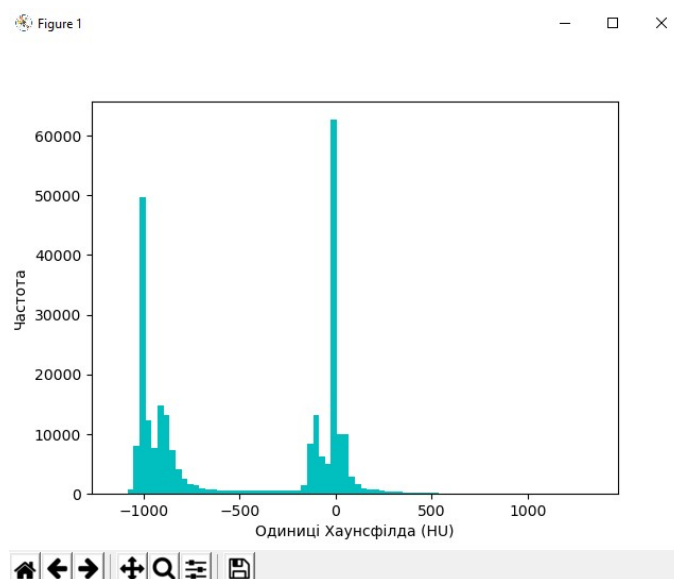


Рис. 6. Гістограма даних в одиницях Хаунсфілда

З отриманим новим піксельним масивом вивили гістограму з видаленням нульових значень та зображення по новій шкалі з накладанням сірої та теплової карт. Даний етап повторюється діями з попереднім.

Виведені результати з інтерфейсом програми наведені на рисунках 9-10.

Щоб ліквідувати шуми з зображення використовуємо метод покращення з пакетами `dicom`, `numpy`, `cv2`, `matplotlib`, `os`, `scipy`, `skimage`. Значення пікселів перетворили в одиниці шкали HU операцією `def_transform_to_hu` вказуючи перетворення масштабу. Будуємо графіки з встановленням границь вікна зображення операцією `def_window_image(image, window_center, window_width)` та перетворенням значень в пікселі матриці. Даний етап проводимо функцією `def_load_and_plot_image`. Виведений результат зображено на рисунку 11.

Покращимо якість отриманих зображень і відрегулюємо їх розмір.

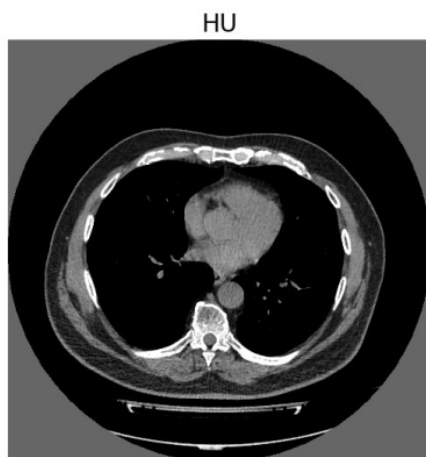


Рис. 7. Початкове зображення

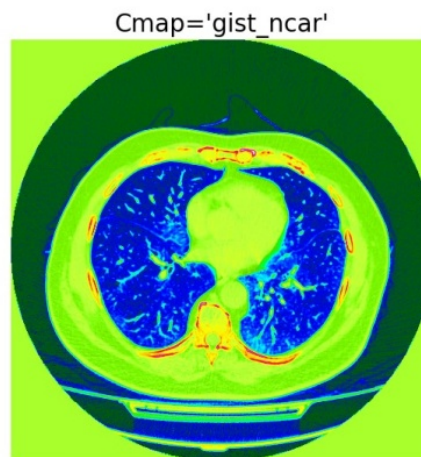


Рис. 8. Зображення з тепловою картою

Завдяки `morphology.dilation` створюється сегментація зображення з масивом квадратного розміру 5×5 . Ці операції виконуємо функцією `def remove_noise`, де зчитане зображення - `dicom.read_file(file_path)`, переводимо в масив - `medical_image.pixel_array` та перетворюємо за допомогою шкали HU. Розмір `label_count` - це кількість знайдених класів / сегментацій.

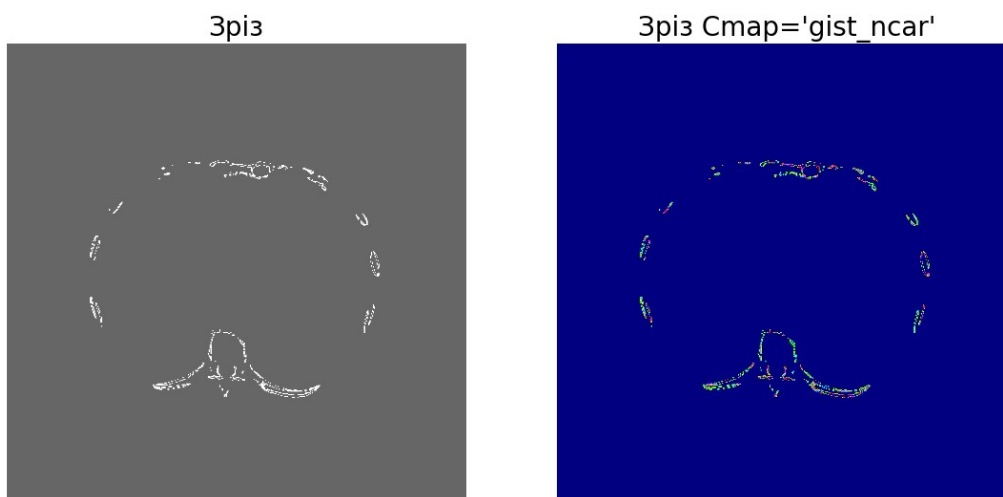


Рис. 9. Результати з інтерфейсом програми

```

Файл  Редактировать  Поиск  Вид  Проект  Пуск  Инструменты  Помощь
Обзор файлов
Этот компьютер
Видео
Документы
Зеркала
Изображения
Музыка
Объемные объекты
Рабочий стол
Локальный диск (C:)
Новый том (D)
Новый том (E)

import dicom
import numpy as np
import matplotlib.pyplot as plt
from tkinter import *
from tkinter import filedialog
import cv2
# ----- завантаження файлу -----
root = Tk()
root.withdraw()
root.update()
root.update()
image_path = filedialog.askopenfile(
    filetypes=(("DICOM", "*.dcm"),
               ("All files", "*.*")))
root.destroy()
# -----
ds = dicom.read_file(image_path.name)
img = ds.pixel_array
# print(ds.pixel_array)  матриця
# print(ds.pixel_array.shape)  shape матриці
np.save("fullimage_temp.npy", img)
file_used = "fullimage_temp.npy"
img_to_process = np.load(file_used).astype(np.float64)
# ----- функція переводу в шкалу Хаунсфілда -----
def get_pixels_hu(medical_image, image):
    image = image.astype(np.int16)

```

Рис. 10. Інтерфейс представлення зображень за допомогою шкали Хаунсфілда

Для покращення зображення використовуємо `mask` бібліотеки `scipy`, збереженням необхідної області та видаленням непотрібного фону операціями `coords`, `top_left`, `bottom_right`. Оброблене зображення виводимо операціями пакета `matplotlib` (рис. 14). Наступними діями відцентрували та вирізали зображення (рис. 15). Для фінальної візуалізації зображення органів грудної клітки міняємо пікселі на пікселі вихідного коду (`final_image[pad_top:pad_top + height, pad_left:pad_left + width] = image`). На рисунках 12-15 наведений результат та фрагмент скрипту.

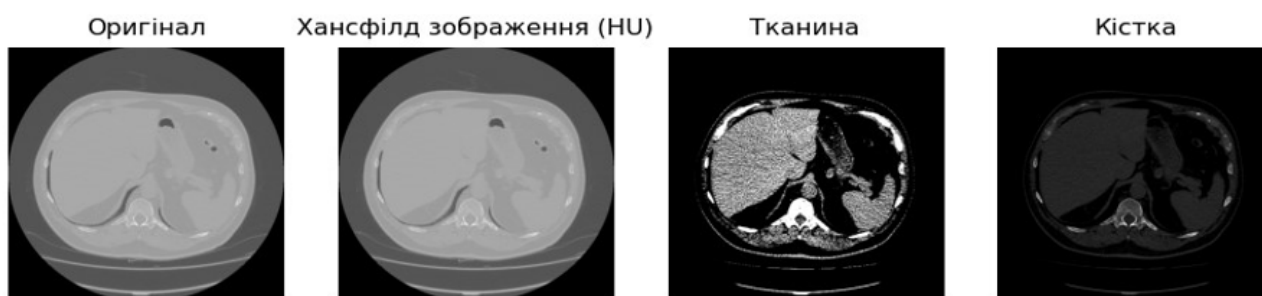


Рис. 11. Виведення оригінального та перетворених зображень

Запропоновані раніше методи сегментації використовують порогове значення зображення на основі HU (значення Хаунсфілда) або інші методи, які базуються на виборі регіонів.

Watershed Transform - це потужний алгоритм сегментації, який базується на “вододілах”. Основна ідея полягає в розміщенні “водного джерела” в кожному локальному мінімумі області, для того щоб затопити всю область при джерелах і побудувати “бар'єри”, коли різні “водні джерела” зустрічаються. Отриманий набір “бар'єрів” становить “вододіл” області [8].

Для реалізації використовуємо наступні бібліотеки: dicom, numpy, pandas, os, scipy, skimage, matplotlib, time.

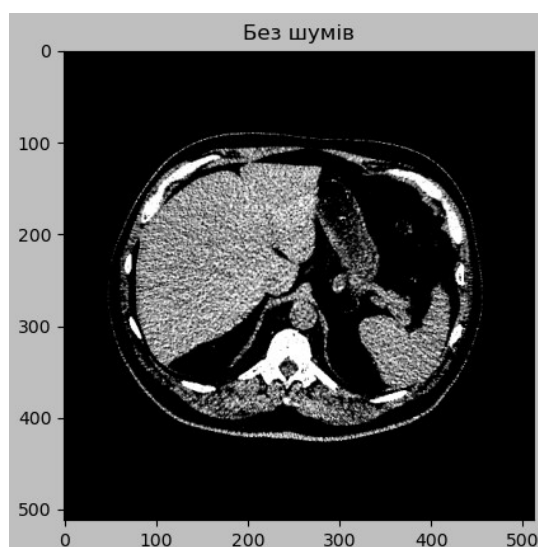


Рис. 12. Покращене зображення формату Dicom без шумів

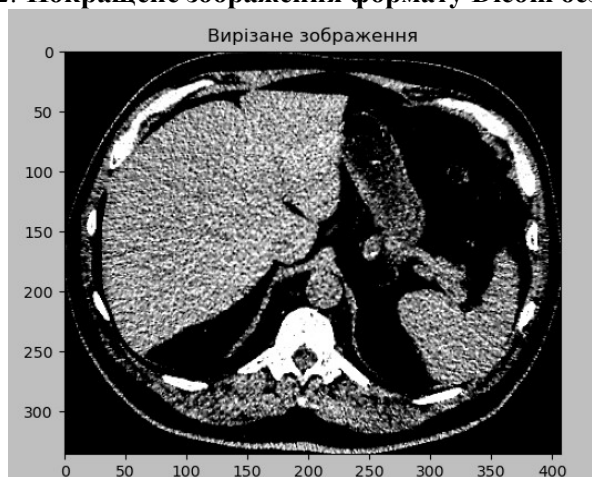


Рис. 13. Вирізане зображення

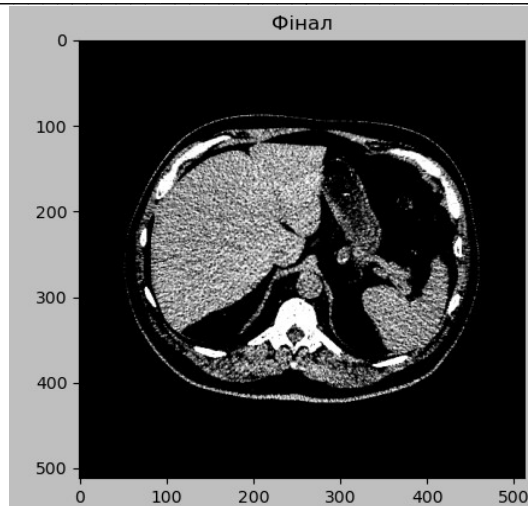


Рис. 14. Покращене зображення формату Dicom

```

001  ## https://vincentblog.xyz/posts/medical-images-in-pyth
.
.
.  import dicom
.  import numpy as np
.  import cv2
.  import os
.  import matplotlib.pyplot as plt
.  import matplotlib.image as mpimg
.
.010  from scipy import ndimage
.  from skimage import morphology
.
.  image_ = 'H1951203' # !!!!!
.
.  def transform_to_hu(medical_image, image):
.  .  intercept = medical_image.RescaleIntercept
.  .  slope = medical_image.RescaleSlope
.  .  hu_image = image * slope + intercept
.
.020  .  return hu_image
.
.  def window_image(image, window_center, window_width):
.  .  img_min = window_center - window_width // 2
.  .  img_max = window_center + window_width // 2
.  .  window_image = image.copy()
.  .  window_image[window_image < img_min] = img_min
.  .  window_image[window_image > img_max] = img_max
.
.  .  return window_image

```

Рис. 15. Фрагмент скрипту

Завантажуємо скановане зображення КТВРЗ ОГК операцією `def load_scan(path)` (рис. 16). Провели перетворення в піксельний масив в одиниці шкали HU операцією `def fet_pixels_hu`. Візьмемо випадковий фрагмент для проведення сегментації. Результат наведений на рис. 17.

```

def load_scan(path):
    print(path)
    slices = [dicom.read_file(path + s) for s in os.listdir(path)]
    slices.sort(key = lambda x: int(x.InstanceNumber))

```

Рис. 16. Скрипт завантаження та сортування

Для використання контрольованої маркерами сегментації вододілу нам потрібно буде ідентифікувати маркери: зовнішній, внутрішній та накладанням їх обох. Для початку створюємо внутрішній маркер: `marker_internal`. Робимо з ним такі дії: сегментацію - `segmentation.clear_border`, встановлюємо міру мітки - `measure.label`, встановлюючи відповідно зону - `areas`. Створюємо зовнішній маркер - `external`, багатовимірним двійковим розширенням із заданим елементом структурування `ndimage.binary_dilation`. Створюємо маркер вододілу - `marker_watershed`, який

повертає масив `pr.zeros`, додаючи внутрішній та зовнішній маркер. Отримані маркери для зразка зрізу виводені на рисунку 18.

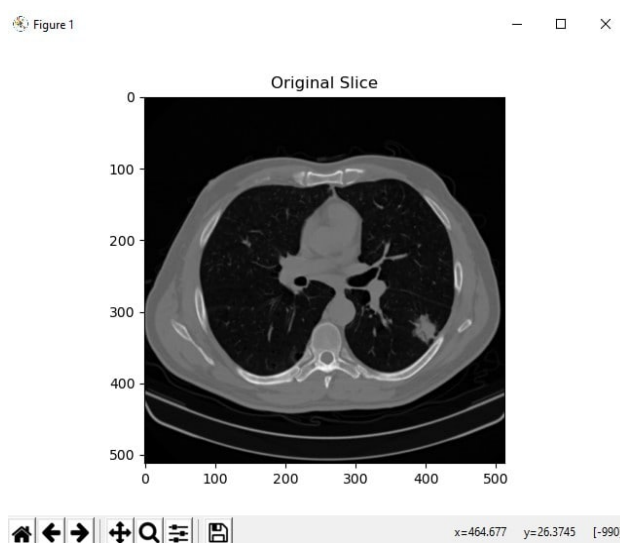


Рис. 17. Оригінальний зріз зображення

Оператор Sobel виконує двовимірне вимірювання просторового градієнта на зображенні і таким чином підкреслює області з високою просторовою частотою, які відповідають краям. Для наступних дій вводимо функцію `def seperate_lungs(image, iterations = 1)`, де буде сегментація легенів за допомогою різних технік. Початковими параметри є скановані зображення, ітерації, а результатом буде сегментовані легені, легеневий фільтр, контур легенів, вододіл легенів та Sobel Gradient. Створимо Sobel Gradient операцією `sobel_filtered` та `sobel_gradient`.

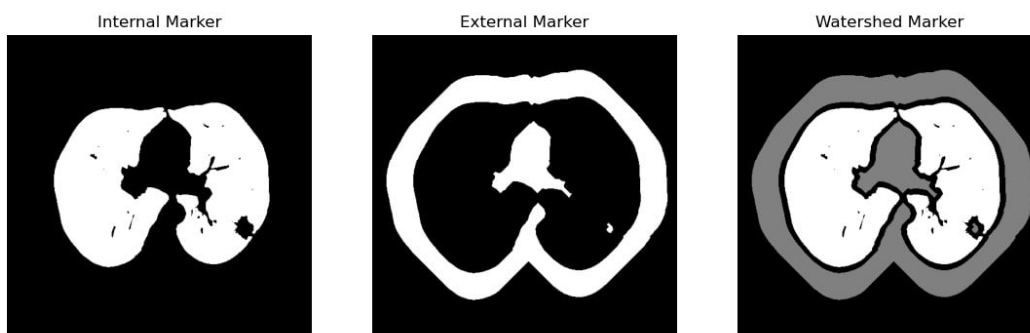


Рис. 18. Створені маркери

Наступним етапом буде використання алгоритму вододілу. Передаємо зображення, згорнене оператором Sobel, та маркер вододілу до `morphology.watershed`. Отримуємо матрицю, позначену за допомогою алгоритму сегментації вододілу. Зменшуємо зображення до контурів після алгоритму Watershed операцією `outline`. Для видалення дрібних та непотрібних деталей на зображення використовуємо операцію `blackhat_struct`. В наших діях дана операція повертає темні плями зображення. Щоб створити легеневий фільтр використовуємо `lungfilter`: внутрішній маркер та контур (рис. 19).

```
lungfilter = np.bitwise_or(marker_internal, outline)
lungfilter = ndimage.morphology.binary_closing(lungfilter, structure=np.ones((5,5)), iterations=3)
...
```

Рис. 19. Скрипт фільтру.

Сегментуємо легені за допомогою операції `segmented`, використовуючи фільтр `lungfilter` та зображення.

Результати цифрової обробки зображення методом вододілу зображено на рисунках 20-23, програмний код зображено на рис. 24.



Рис. 20. Створення градієнта Sobel

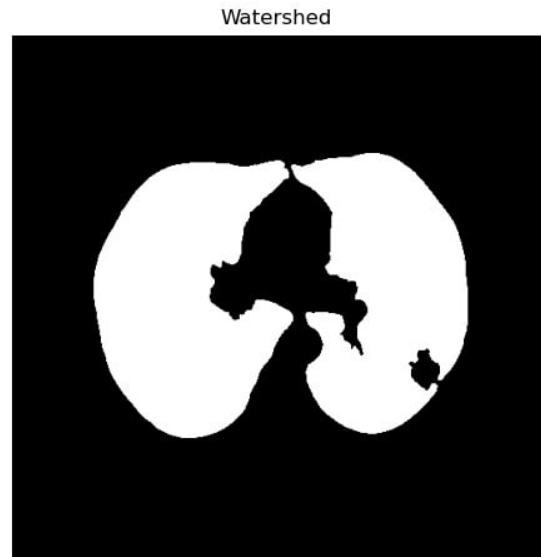


Рис. 21. Використання методу вододілу

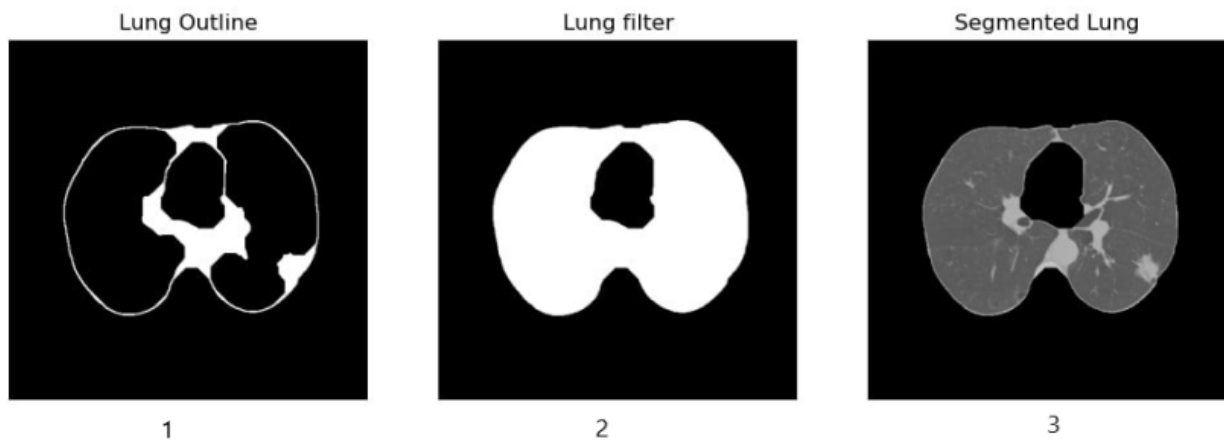


Рис. 22. Обробка зображення:

- 1) зменшення зображення до контурів після алгоритму Watershed;
- 2) створення легеневого фільтру за допомогою внутрішнього маркера та контуру;
- 3) сегментація легенів за допомогою lungfilter.

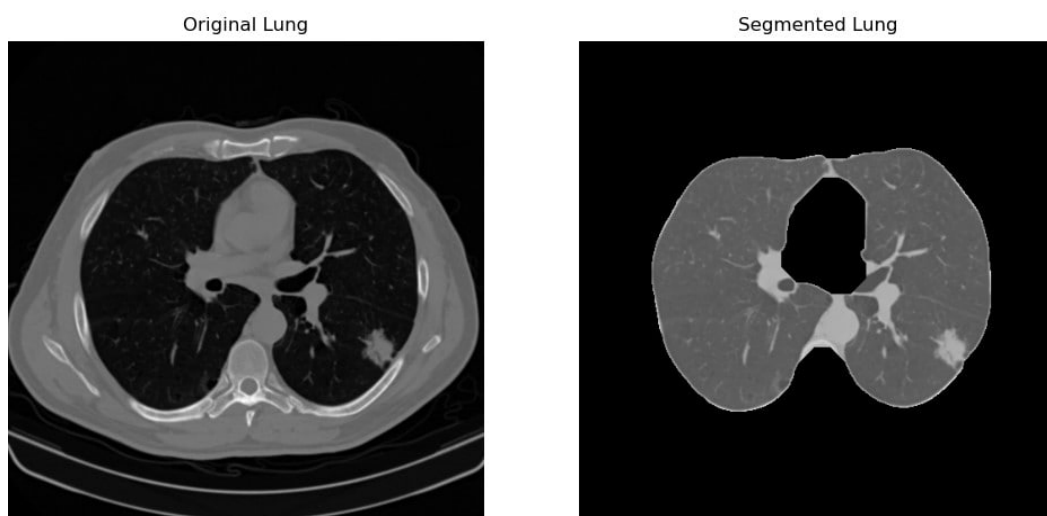


Рис. 23. Порівняння цифрової обробки між первинним зображенням та сегментованим.

```

# https://www.kaggle.com/aadhavignesh/Lung-segmentation-by-marker-controlled-watershed
# http://www.cmm.mines-paristech.fr/~beucher/wtshed.html
import numpy as np
import pandas as pd
import dicom
import os
import scipy.ndimage as ndimage
from skimage import measure, morphology, segmentation
import matplotlib.pyplot as plt

@110
import time

# INPUT_FOLDER = "d:/www/Gotovo/Сегментация изображений за вододілками/My_soft_watersheds/train/"
@116 INPUT_FOLDER = "d:/ProPython/Lung_Segmentation/train/"

#print( os.listdir(INPUT_FOLDER))

@120 patients = os.listdir(INPUT_FOLDER)
patients.sort()

print("Some examples of patient IDs:")
print("\n".join(patients[:5]))

def load_scan(path):
    """
    Loads scans from a folder and into a list.
    Parameters: path (Folder path)

```

Интерпретатор Python

```

Some examples of patient IDs:
ID0000_AGE_0060_CONTRAST_1_CT,
ID0001_AGE_0069_CONTRAST_1_CT,
ID00024_AGE_0060_CONTRAST_1_CT.dcm,
ID0002_AGE_0074_CONTRAST_1_CT,
ID0003_AGE_0075_CONTRAST_1_CT
d:/ProPython/Lung_Segmentation/train/

```

Рис. 24. Фрагмент коду реалізації алгоритму Watershed

Висновок. Розглянуто алгоритми та проведена програмна реалізація роботи з зображеннями отриманими за допомогою КТВРЗ і збереженими у форматі DICOM. Мовою реалізації алгоритмів обрано Python.

В якості тестових зображень використано КТВРЗ ОГК людини.

Предмет подальших досліджень. Проведення досліджень обробки зображень з отриманням цифрових моделей інших структур. Розробка спеціалізованих інструментів сегментації та поліпшення їх функціоналу зі спільним використанням результатів різних видів медичної діагностики.

Література:

1. Василенко В. Х. Пропедевтика внутренних болезней / В. Х. Василенко, А. Л. Гребенева, Н. Д Михайловой. – Москва: Медицина, 1974
2. Стандарт DICOM [Електронний ресурс] – Режим доступу до ресурсу: <http://dicom.nema.org>
3. Технология построения твердотельных моделей бедренных костей на основе данных компьютерной томографии [Електронний ресурс] / В. М.Соловьёв, П. В. Ирматов, М. С. Ирматова, М. Г. Щербаков – Режим доступу до ресурсу: <https://cyberleninka.ru/article/n/tehnologiya-postroeniya-tverdotelnyh-modeley-bedrennyh-kostey-na-osnove-dannyh-kompyuternoy-tomografii/viewer>
4. Mustra Mario, Delac Kresimir, Grgic Mislav. Overview of the DICOM Standard // ELMAR, 2008. 50th International Symposium. Zadar, Croatia. P. 39–44
5. NumPy, часть 1: начало работы [Електронний ресурс] – Режим доступу до ресурсу: <https://pythonworld.ru/numpy/1.html>
6. Ravishankar C. Image Processing and Acquisition using Python / C. Ravishankar, P. Sridevi., 2020. – 451 с.
7. Van derWalt, S., Colbert, S. C. & Varoquaux, G. The NumPy array: a structure for efficient numerical computation. Comput. Sci. Eng. 13, 22–30 (2011).
8. Beucher S. The watershed transformation applied to image segmentation[J]. SCANNING MICROSCOPY-SUPPLEMENT-, 1992: 299-299.
9. Yakovenko O., Khanin O., Lotysh V., CLINICAL FEATURES OF SEVERE COVID-19 WITH LETHAL OUTCOME IN VOLYN REGION RESIDENTS. Ukr. Pulmonol. J. 2021; 2: 16–24. DOI: 10.31215/2306-4927-2021-29-2-16 -24