RO4101 – DIAGNOSTIC TECHNOLOGIES

Lab Session #1

Learning Outcomes:

This exercise is designed to introduce and familiarize you with the Digital Imaging and Communications in Medicine-DICOM format and how it can be utilized and analyzed.
By the end of this exercise, you will have the knowledge and tools to perform the following tasks in PYTHON:

- Load a DICOM file and display it.
- Create a stack of DICOM files, representing a volume.
- Perform simple analysis on the DICOM stack - identification/retrieval etc.

Marks Allocated: 1% (10 points)

Computing Tools Required:

- **Python 3.x**
- **Libraries:** pydicom, numpy, nibabel, matplotlib, scipy (for image processing)

You can install the necessary libraries using pip by running the following command in the terminal:

```
D:\>pip install pydicom numpy nibabel matplotlib scipy
```

Instructions:

You are expected to submit your work as a ***Python script file*** with the following filename after you have been marked in the lab ***(marking in-lab components for these DT labs will be in the last 30mins of any lab session):***

DT-4104-Yourname-lab1.py

TASK-1: Load and Display DICOM file *(5 points)*

Step-1: Download the DICOM data from Moodle and extract it into your working directory. Load a DICOM file by passing the path as given below.

Import Required Libraries

[1]:
```
import os
import pydicom
import numpy as np
import matplotlib.pyplot as plt
```

Load a DICOM file

[2]:
```
dicom_file_path = './T2/1-15.dcm'
dicom_data = pydicom.dcmread(dicom_file_path)
```

Step-2: DICOM files contain a lot of metadata and additional information about the image such as the patient's name and date of birth, the place where the scan was completed, the time and date of the scan, the name of the doctor, and so on. You can access individual fields of the info structure as below:

Print the DICOM header

```
[3]: print(dicom_data)
```

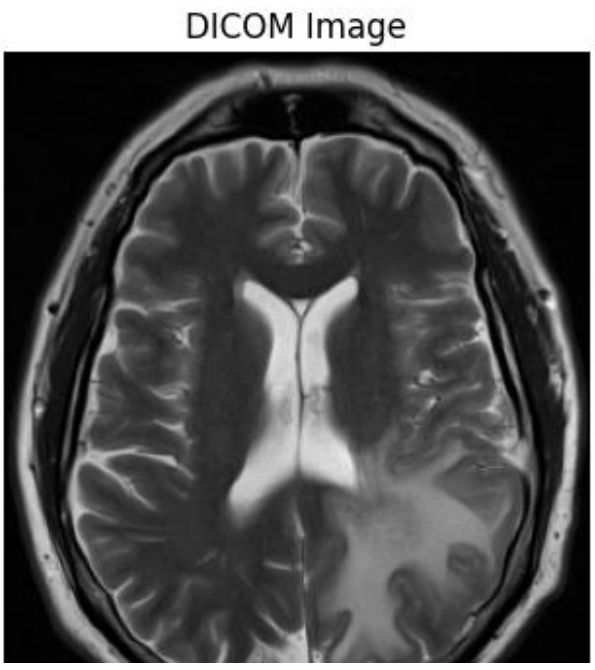
```
(0008, 0018) SOP Instance UID                   UI: 1.3.6.1.4.1.14519.5.2.1.4429.7055.724371543136
(0008, 0020) Study Date                         DA: '19920402'
(0008, 0021) Series Date                        DA: '19920402'
(0008, 0023) Content Date                       DA: '19920402'
(0008, 0030) Study Time                         TM: '085321.125000'
(0008, 0031) Series Time                        TM: '090214.718000'
(0008, 0033) Content Time                       TM: '090214.750000'
(0008, 0050) Accession Number                   SH: '5686274134839343'
(0008, 0060) Modality                           CS: 'MR'
(0008, 0070) Manufacturer                       LO: 'Imaging Biometrics LLC'
(0008, 0090) Referring Physician's Name         PN: ''
(0008, 1030) Study Description                  LO: 'FH-HEAD^Brain Protocols'
(0008, 103e) Series Description                 LO: 'T2_reg'
(0008, 1090) Manufacturer's Model Name          LO: 'IB Delta Suite'
(0010, 0010) Patient's Name                     PN: 'PGBM-001'
(0010, 0020) Patient ID                         LO: 'PGBM-001'
(0010, 0030) Patient's Birth Date               DA: ''
```

Step-3: Display the loaded image using Matplotlib.

Display the DICOM image ¶

```
[5]: dicom_array = dicom_data.pixel_array
```

```
[6]: plt.imshow(dicom_array, cmap='gray')
     plt.title('DICOM Image')
     plt.axis('off')
     plt.show()
```



Complete the following tasks:

- 1.1: Load an image from the MRI folder.
    - 1.1.1: Print the DICOM header*(1)*.

- 1.1.2: Get the DICOM pixel_array and print the shape of the image *(1)*.
- 1.1.3: Display the DICOM image *(1)*.

- 1.2: Load the DICOM image from other folders and repeat the same task *(2)*.

TASK-2: Load all DICOM series and stack them to represent a volume *(5 points)*

Step-1: A) List all the DICOM files in the folder.
B) Load all the files and extract the pixel_array of each file using 'for loop' and store them into a Python list.
C) Stack all the image arrays to form a single volume.

Create a stack of DICOM files representing a volume

```
*[5]:
# Directory containing DICOM files
dicom_dir = './T2'
# List all DICOM files in the directory
dicom_files = [os.path.join(dicom_dir, f) for f in sorted(os.listdir(dicom_dir)) if f.endswith('.dcm')]

# Read all DICOM files
dicom_images = [pydicom.dcmread(file).pixel_array for file in dicom_files]

# Create a 3D volume
volume = np.stack(dicom_images, axis=-1)
```

Step-2: Display the volume.

Function to display 3D volume

```
[6]:
from ipywidgets import interact
def explore_3D_array(arr: np.ndarray, cmap: str = 'gray'):
  """
  Given a 3D array with shape (X,Y,Z) This function will create an interactive
  widget to check out all the 2D arrays with shape (X,Y) inside the 3D array.

  Args:
    arr : 3D array with shape (X,Y,Z) that represents the volume
    cmap : Which color map use to plot the slices in matplotlib.pyplot
  """

  def fn(SLICE):
    plt.figure(figsize=(7,7))
    plt.imshow(arr[:, :, SLICE], cmap=cmap)

  interact(fn, SLICE=(0, arr.shape[2]-1))
```

```
[7]:
explore_3D_array(volume)
```

Complete the following tasks:

- 2.1: Load all the images from the CT folder.
  - 2.1.1: Print the DICOM header of the first image *(1)*.
  - 2.1.2: Get the DICOM pixel_array and stack it into a volume*(1)*.
  - 2.1.3: Print the volume shape and display the DICOM volume *(1)*.
- 2.2: Print the minimum and maximum pixel value of the volume using the min(), max() function *(2)*.