RO4101 – DIAGNOSTIC TECHNOLOGIES

Lab Session #2

Learning Outcomes:
This lab exercise aims to introduce and familiarize you with different image-processing techniques that are crucial for extracting meaningful features from medical images.

By the end of this exercise, you will have the knowledge and tools to perform the following tasks in PYTHON:

- Learn techniques to identify and outline the boundaries of objects within medical images.
- Understand methods to eliminate or minimize background noise, enhancing the focus on relevant features.

Marks Allocated: 2% (20 points (*pts)*)

Computing Tools Required:

- **Python 3.x**
- **Libraries:** pydicom, numpy, nibabel, matplotlib, scipy, opencv-python (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:

Download the data:
https://drive.google.com/file/d/1M8Lol4hkdXQ3Cy75CtMb1FW9CV9YUAS6/view?usp=sharing

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-lab2.py

TASK-1: Load DICOM file and apply Sobel Edge Detection *(5 points)*

Step-1: Download the DICOM data from above link and extract it into your working directory. Load a DICOM file from the Chest-Xray folder by passing the path.

Imort Required Libraries

[1]:
```
import os
import pydicom
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage
import dicom2nifti
import cv2
```

Load a DICOM File

[2]:
```
# Load the DICOM image
dicom_file_path = 'C:/Users/deepanshu.sharma/OneDrive - Plaksha University/Desktop/DT/x1.dicom'  # Replace with your DICOM file path
dicom_data = pydicom.dcmread(dicom_file_path)
# Extract image data
image = dicom_data.pixel_array
```

Step-2: Scale the image pixel values within a specific range for better visualization.

Image Scaling

[3]:
```
# Scale the pixel values of the image to a 0-255 range.
image_norm = (image - np.min(image)) / (np.max(image) - np.min(image)) *255 #min-max scaler

image_norm = np.uint8(image_norm) #convert the image to uint8 for proper visualization
```

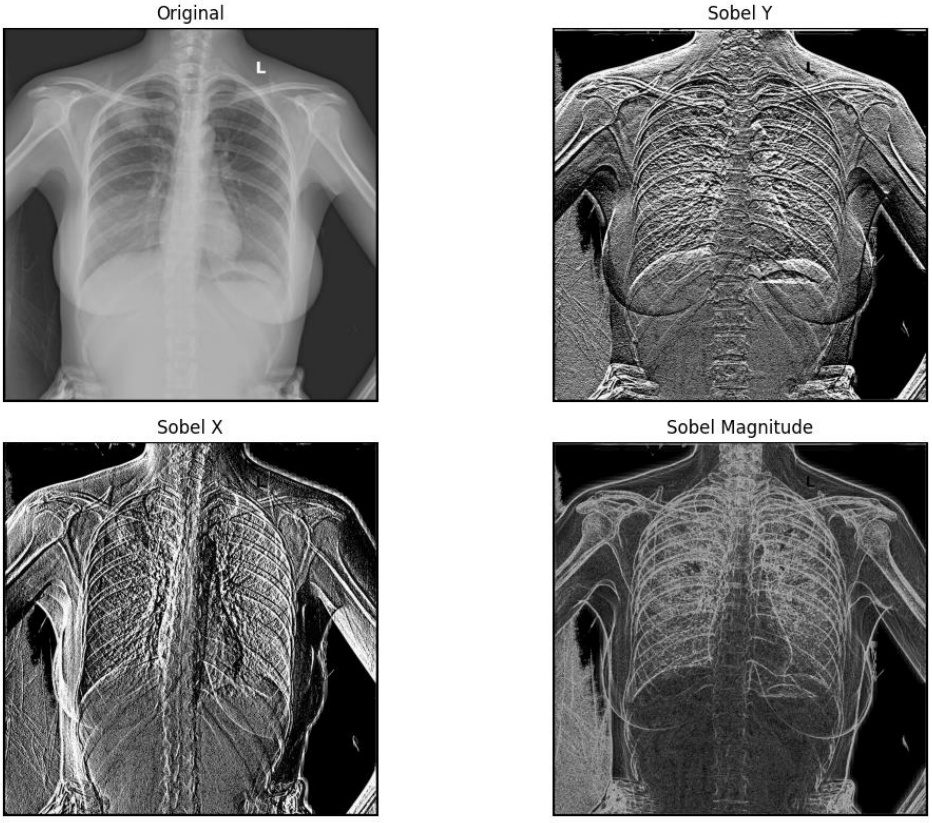
Step-3: Apply Sobel filters on the scaled image and visualize the results.

Applying Sobel Edge Detection

[5]:
```
# Define Sobel kernels
sobel_x = np.array([[-1, 0, 1],
                    [-2, 0, 2],
                    [-1, 0, 1]])

sobel_y = np.array([[-1, -2, -1],
                    [0, 0, 0],
                    [1, 2, 1]])

# Apply Sobel filter in the x direction
sobel_x_result = ndimage.convolve(image_norm, sobel_x)

# Apply Sobel filter in the y direction
sobel_y_result = ndimage.convolve(image_norm, sobel_y)

# Compute the magnitude of gradients
sobel_mag = np.sqrt(sobel_x_result**2 + sobel_y_result**2)
```

Display the results

[5]:
```
# Display the results using matplotlib
plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)
plt.title('Original')
plt.imshow(image_norm, cmap='gray')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.title('Sobel Y')
plt.imshow(sobel_y_result, cmap='gray')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.title('Sobel X')
plt.imshow(sobel_x_result, cmap='gray')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.title('Sobel Magnitude')
plt.imshow(sobel_mag, cmap='gray')
plt.axis('off')

plt.tight_layout()
plt.show()
```



Complete the following tasks:

- 1.1: Load an image from the Mammogram folder.
  - 1.1.1: Print the shape, min pixel value, max pixel value, and the datatype of the image. *(Hint: print(image.dtype)) (1* pt*).*
  - 1.1.2: Scale the image to a range of 0-255 and datatype to uint8 *(1* pt*).*
  - 1.1.3: Apply Sobel Edge Detection on the image *(2* pts*).*
  - 1.1.4: Display the results *(1* pt*).*

TASK-2: Eliminate or minimize background noise and extract a region of interest *(5 points)*

Step-1: Normalize the loaded Mammogram through cv2.

Normalize the image and apply binary thresholding

[7]:
```
# Normalize the image to the range 0-255 through cv2.
image_normalized = cv2.normalize(image, None, 0, 255, cv2.NORM_MINMAX)
image_normalized = np.uint8(image_normalized)
```

Step-2: Apply thresholding to create a binary image.

Apply thresholding

[8]:
```
# Apply thresholding to create a binary image
_, binary_image = cv2.threshold(image_normalized, 70, 255, cv2.THRESH_BINARY)
```

Step-3: Remove the background noise in the scan and extract the required region.

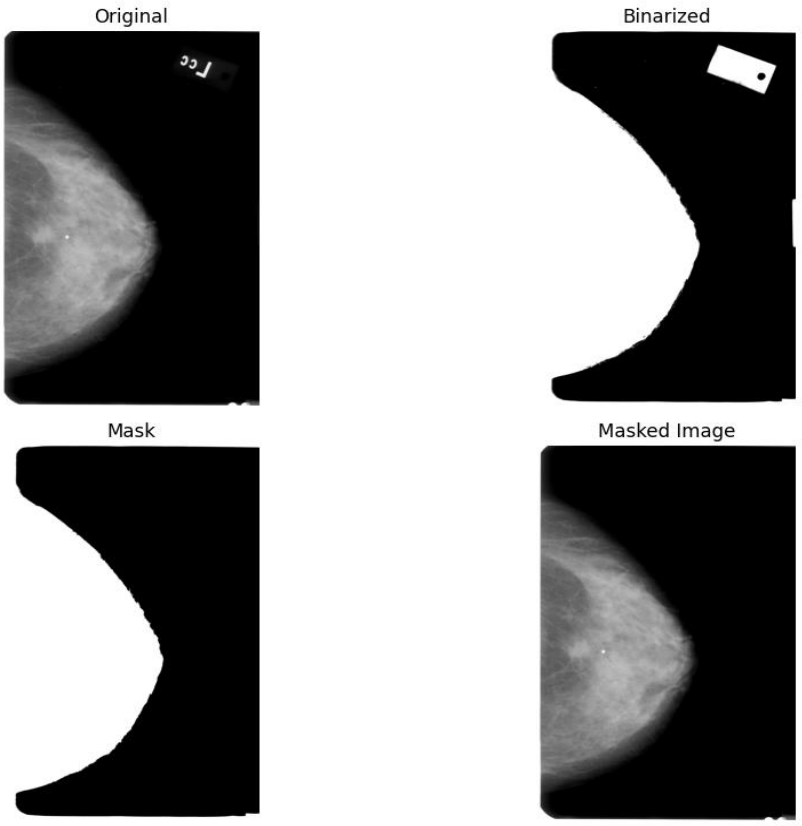
Extract the contours and select the largest contour in the binarized image.

[10]:
```
# Find contours
contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Find the largest contour by area
if contours:
    largest_contour = max(contours, key=cv2.contourArea)
else:
    largest_contour = np.array([])
```

Mask the largest contour and apply the mask to the original image

[12]:
```

# Create a mask with the same dimensions as the image
mask = np.zeros_like(binary_image)

# Draw the largest contour on the mask
if largest_contour.size > 0:
    cv2.drawContours(mask, [largest_contour], -1, (1), thickness=cv2.FILLED)


# Apply the mask to the original image
masked_image = cv2.bitwise_and(image_normalized, image_normalized, mask=mask)
```

Step-4: Display image_normalized, binary_image, mask, and masked_image.



Complete the following tasks:
- 2.1: Perform the above exercise on the chest x-ray. (3 pts)
- 2.2 *Adjust the threshold clip limit to see the changes. (1* pts*)*
- 2.3: Apply the Sobel edge detection on the mammogram image (1 pts)

Home Assignment: Try other edge detection techniques using the open-cv library and compare the results (1 points).

Canny Edge Detection: ***https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html?ref=blog.roboflow.com***

Sobel Edge Detection: ***https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives.html***

Laplacian Edge Detection: ***https://docs.opencv.org/4.x/d5/db5/tutorial_laplace_operator.html***

To do tasks:
- 2.1: Display a 2*2 grid of images containing a normalized image, canny edge detection result, Sobel edge detection result, and Laplacian edge detection result on the x-ray image. (5 pts)

- 2.2: Apply Sobel edge detection on another mammogram scan and apply the background removal exercise on one case. (5 pts)