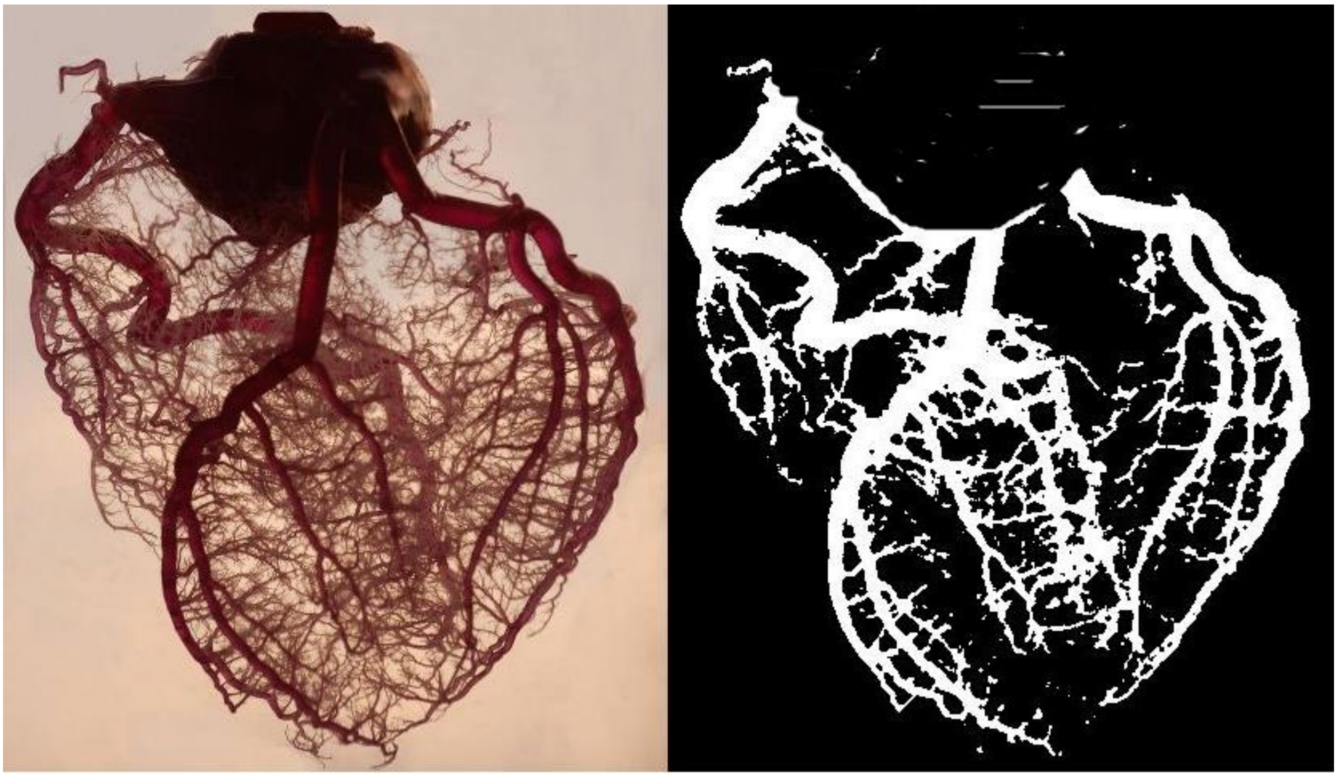
CSC 462 2.0 Digital Image Processing

Assignment 1



Index Number : AS2019358

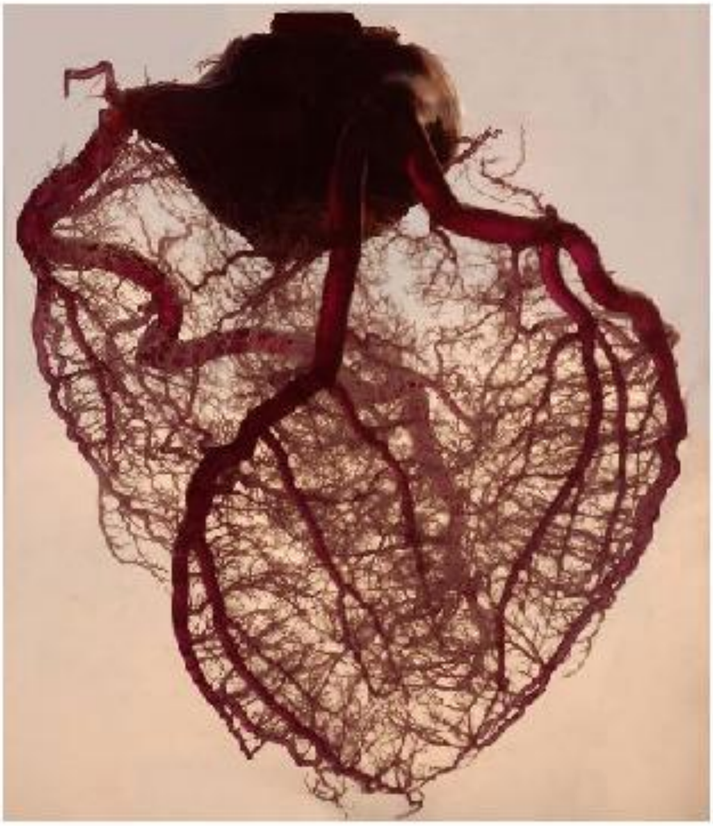
Name : G.L.H.Binoli Gaweshika

Step 1 : Load image

Octave Code:

```
originalImage = imread('heart3.png');
imshow(originalImage);
```

Output:



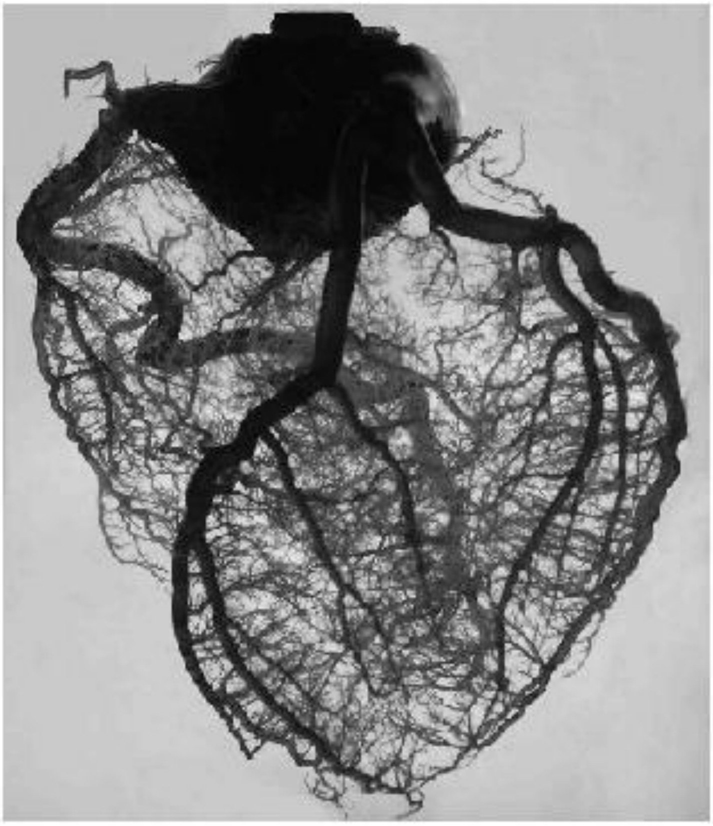
Step 2 : Convert image to gray scale

Convert the image to grayscale to simplify further processing.

Octave Code:

```
grayImage = rgb2gray(originalImage);
imshow(grayImage);
```

Output:



Step 3 : Load image package Octave Code : pkg load image

Octave Code:

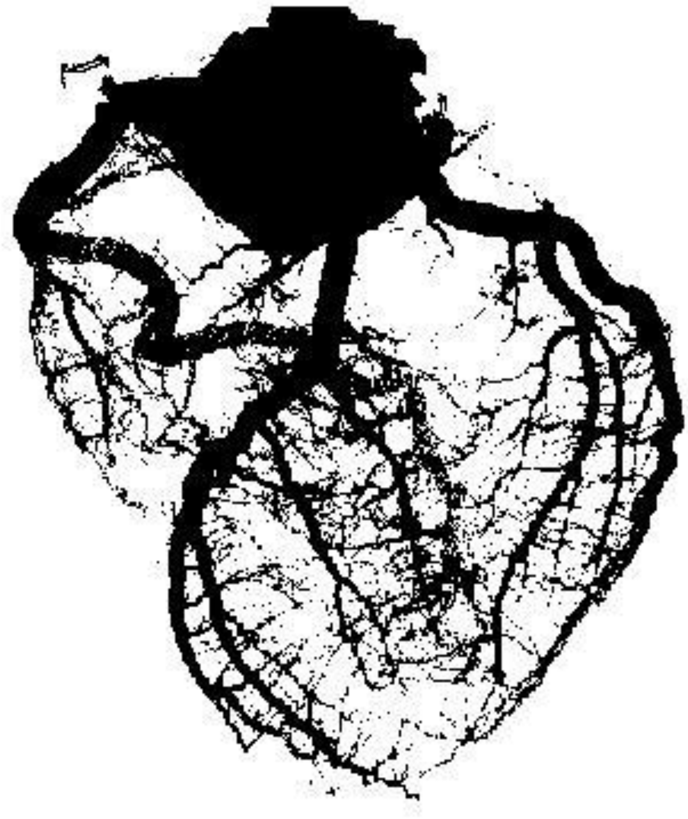
```
pkg load image;
```

Step 4 : Convert image to binary with threshold of 0.32

Octave Code:

```
binaryImage = im2bw(grayImage, 0.32);
imshow(binaryImage);
```

Output:



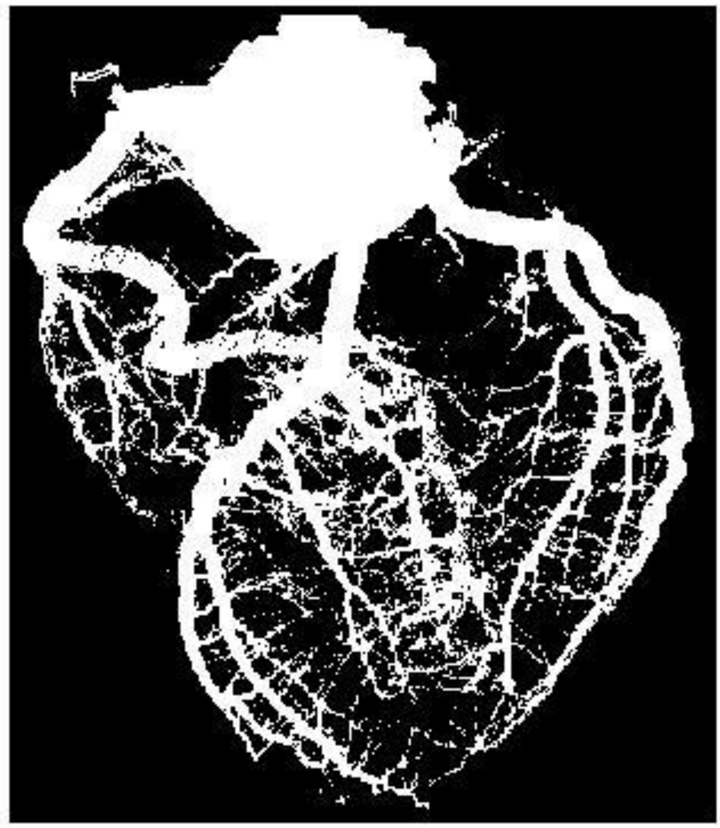
Step 5 : Invert the binary image

Getting the complement of binary image.

Octave Code:

```
negImage = imcomplement(binaryImage);
imshow(negImage);
```

Output:



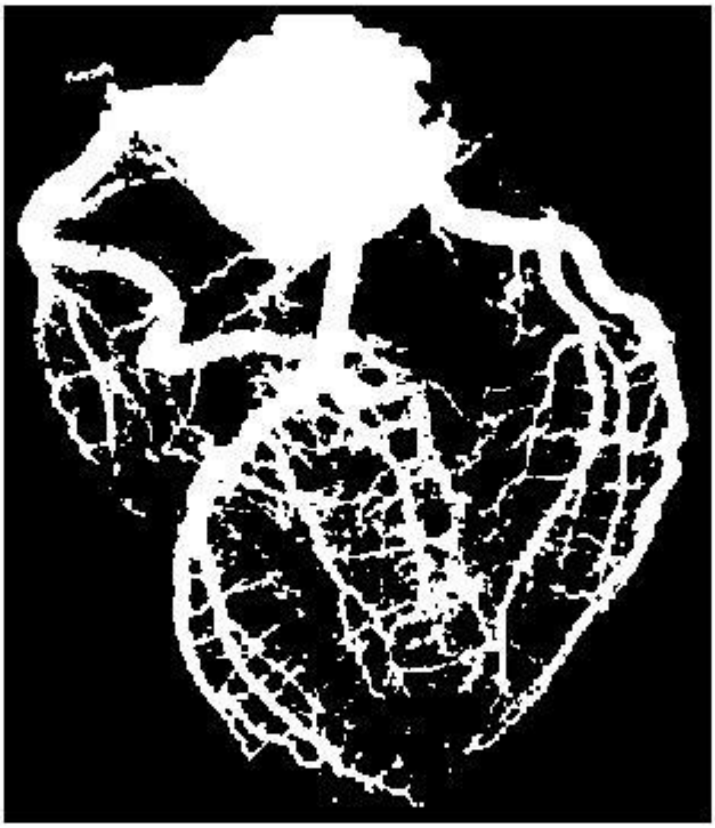
Step 6 : Filter the image removing noises

Apply Median filter to reduce noise and enhance blood vessel features.

Octave Code:

```
noiseFreeImage = medfilt2(negImage,[3 3]);
imshow(noiseFreeImage);
```

Output:



Step 7 : Define the structuring element

Octave Code:

```
se = strel('disk', 15 ,0);
```

Step 8 : Apply morphological closing operator

This morphological operation that helps to remove small pats of the image & keep only the lagest connected component.

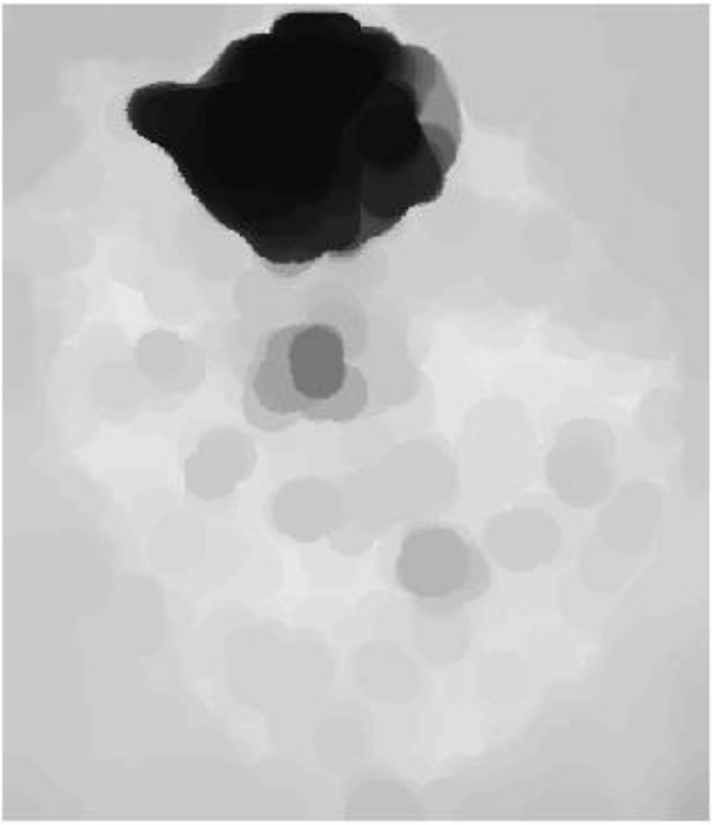
Octave Code:

```
se = strel('disk', 15 ,0);

closedImage = imclose(grayImage, se);

imshow(closedImage);
```

Output:



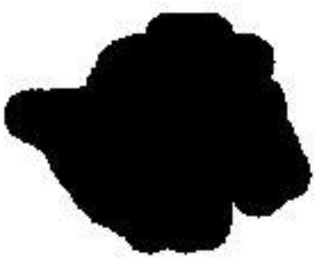
Step 9 : Threshold the image

Thresholding based on the pixel values of the image, to keep only the pixels with highest intensities.

Octave Code:

```
thresholdValue = 43;
higherPixelValuesImage = closedImage > thresholdValue;
imshow(higherPixelValuesImage);
```

Output:



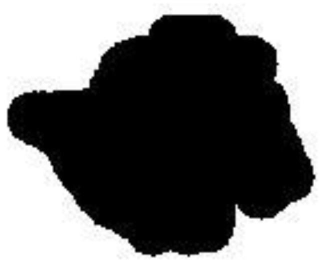
Step 10 : Convert to a binary image

This step identifies the largest connected component as a binary image. This step is done mainly due to need of the getting the complement.

Octave Code:

```
binaryImage1 = im2bw(higherPixelValuesImage, 0.35);
imshow(binaryImage1);
```

Output:



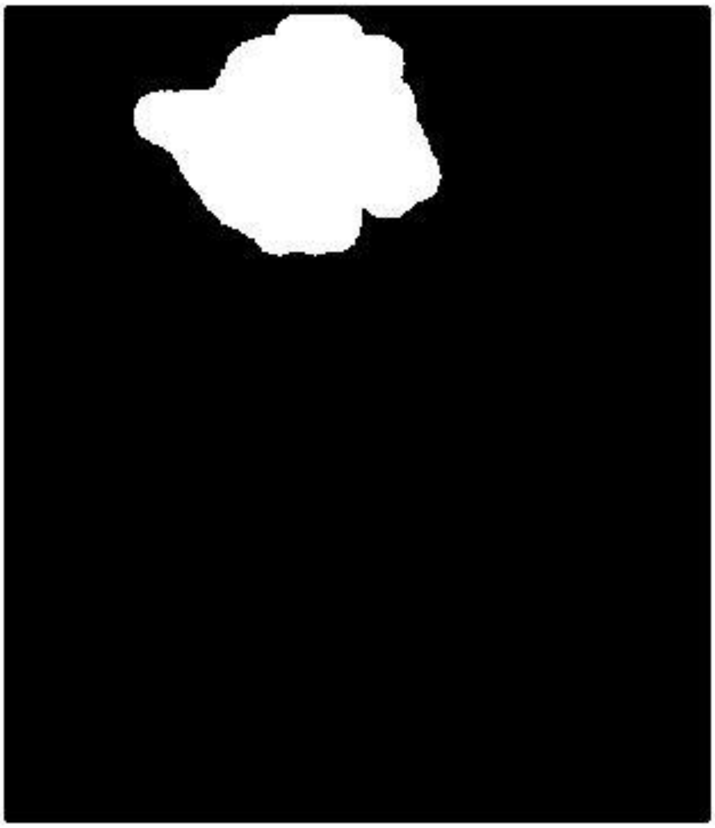
Step 11 : Invert the binary image

This step inverts the binary image and results a complemented largest connected component.

Octave Code:

```
negImage1 = imcomplement(binaryImage1);
imshow(negImage1);
```

Output:



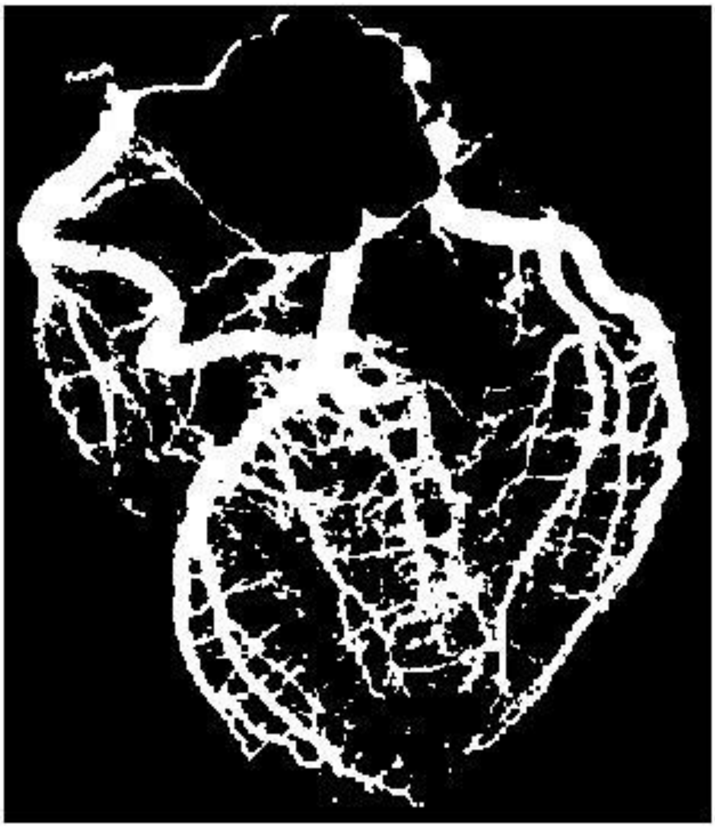
Step 12 : the largest connected component eliminated from the preprocessed image

This step removes the largest connected component with the highest intensities from the noise reduced preprocessed image and results only the blood vessels.

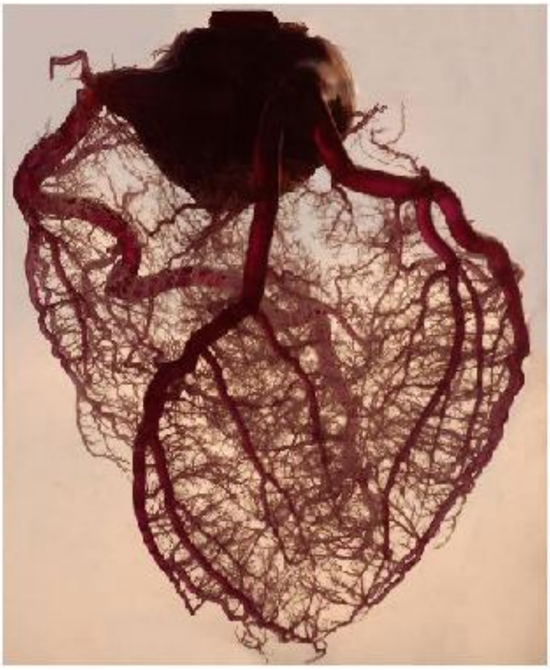
Octave Code:

```
output = noiseFreeImage - negImage1;
imshow(output);
```

Output:



Original Image



Final Output