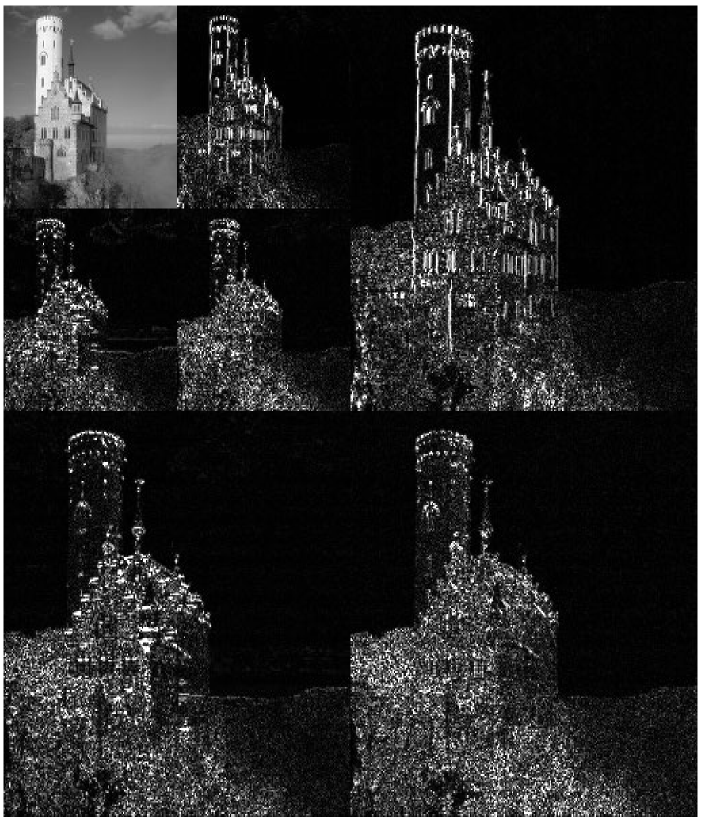


ABSTRACT

This project was made according to the course CEN 590 according to the Term project with topic about Discrete Wavelet Transform implemented on MATLAB.

Prepared By: Baftjar Tabaku
Supervised By:

CEN 590 Term Project

EPOKA UNIVERSITY, TIRANE, ALBANIA

07/29/20

CEN 590 TERM PROJECT

Discrete Wavelet Transform

Discrete Wavelet transform

The DWT provides a sparse representation for many natural signals. In other words, the important features of many natural signals are captured by a subset of DWT coefficients that is typically much smaller than the original signal. This "compresses" the signal. With the DWT, you always end up with the same number of coefficients as the original signal, but many of the coefficients may be close to zero in value.

As a result, you can often throw away those coefficients and still maintain a high-quality signal approximation.

Performing discrete wavelet transform on images will be important to know that the filter bands are one dimensional in the nature, and the object that we want to analyze will be 2D in nature, and will be necessary to develop a technique that we apply one dimensional (1D) filter bands along the rows and then columns of the images or vice versa.

As shown in the figure below, starting with an example of a simple image that will pass though some transformations with the scale 'j+1' and n number of rows and m number of columns;

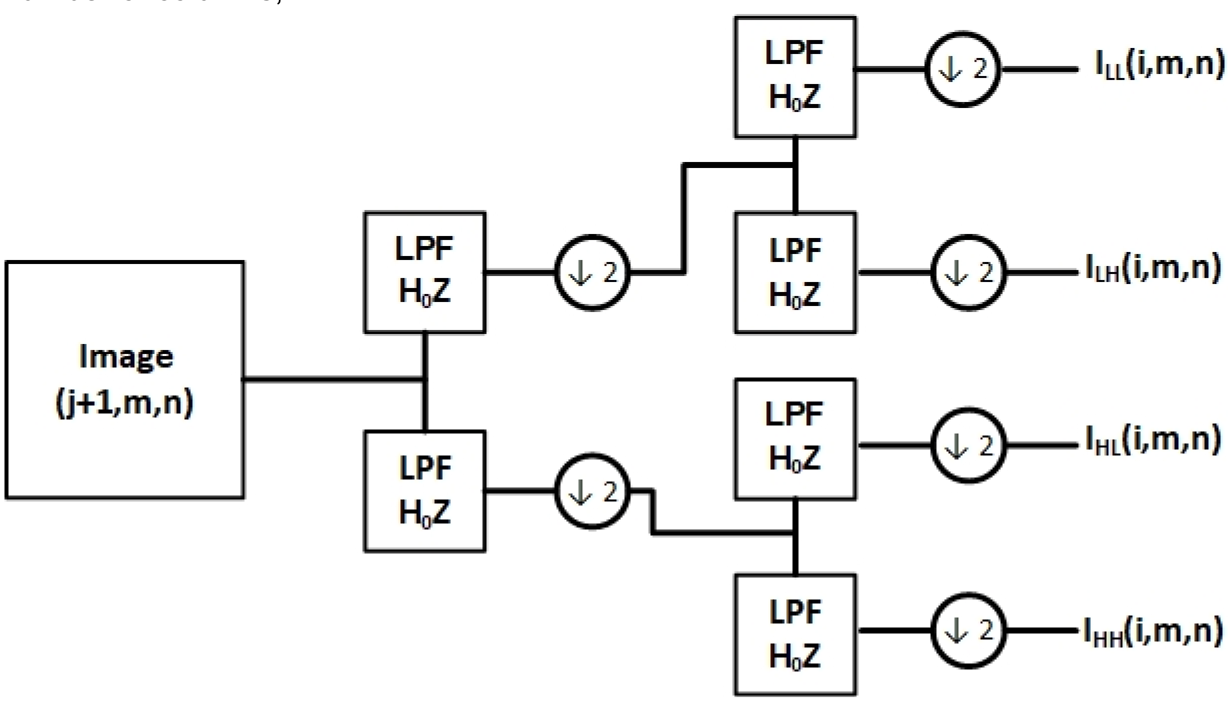


Figure 1 Image passing though the filters

This input I can be any kind of signal, and it will pass thought two filters, Low pass filter and High pass filter, with transfer functions of H_0Z and H_1Z. To avoid losing data and information, we apply substitute by 2 booths of the outputs of the lowpass filter (LPF) as shown in the circles of the upper figure. The high pass filter extracts the edges and Low pass filter does the approximations.

The output sub-images, are at one resolution half of the image that we started with, so the resolution will be affected by a factor of two, booth along the rows and along the columns will mean that we have done a sub-sampling by a factor of four. Therefore, each of these sub-bands I_{LL}, I_{LH}, I_{HL} and I_{HH} contains ¼ of the total number of samples according to our original image.

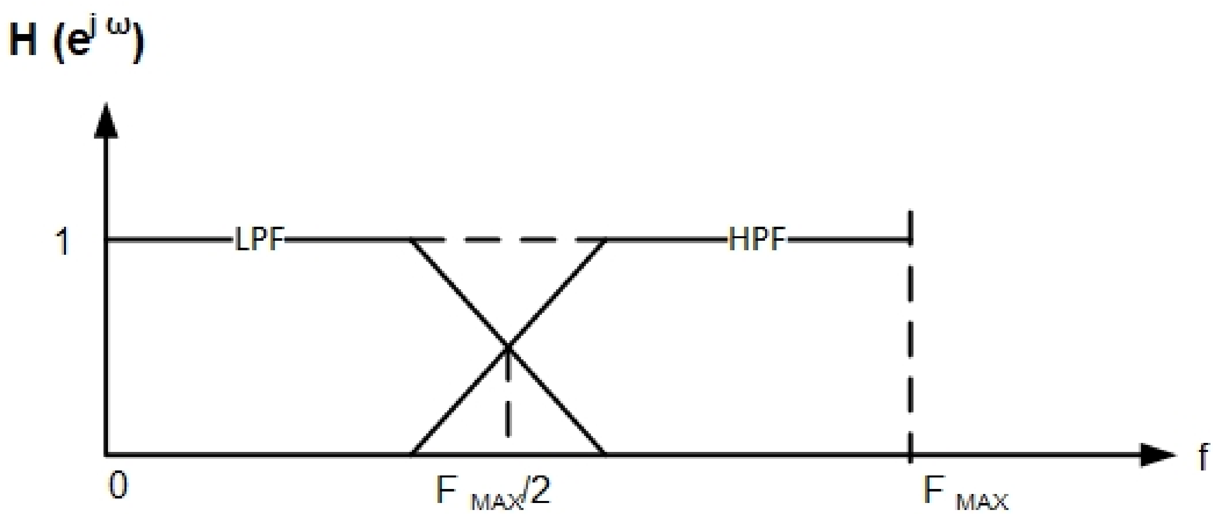


Figure 2 Frequency Response of the filters

Frequency Response of the filter can be shown according the figure below, with frequency f and magnitude H ($e^{j\,\omega}$), with the frequency response of the low pass filter and high pass filter, where HPF (high pass filter) and LPF will have the amplitude 1, the signal will be split into these two bands, and the bandwidth of the signal is half, each of these sub-bands (low pass and high pass band).

Visually this can be interpreted as shown below, where there will be the outputs I_{LL}, I_{LH}, I_{HL}, I_{HH}, whereas output we would get some processed parts of images.

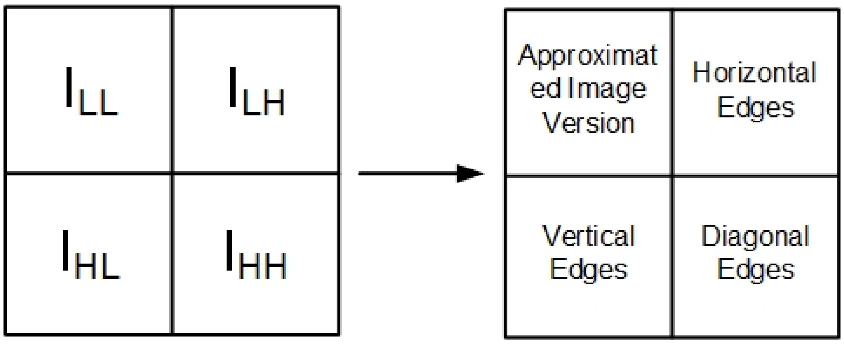


Figure 3 Level 1 Decomposition

Applying Fourier Transform on images

The Fourier transform will deal with continue and infinite functions, the digital images that we want to transform will be of finite length and have discrete pixel values, so the output will be also finite, because of this we have to use a discrete Fourier transform, for N output pixels.

$$X[k] = \sum_{n} x[n]\, e^{-j2\pi nk/N}$$
$$x[n] = \sum_{k} X[k]\, e^{j2\pi nk/N}$$

Figure 4 Finite and discrete input-output Fourier transform

$$X[s,t] = \sum_{n}\sum_{m} x[n,m] e^{-j2\pi mk/M} e^{-j2\pi nk/N}$$
$$x[n,m] = \sum_{n}\sum_{m} X[s,t] e^{j2\pi mk/M} e^{j2\pi nk/N}$$

Figure 5 Fourier transform approximated for two dimensions for image processing

Supposing that we have and image and independently replace each row with its Fourier transform, and replacing each column with its Fourier transform *(as shown in the figure 4)*.

Problem to be treated

Providing the following images;

1. 10 iris images
2. 10 finger print images
3. 10 finger vein images

Then we need to calculate the Discrete Wavelet transform of these images. We will get 4 outputs. Determine the frequency components of these 4 outputs of each image (30 images in total).

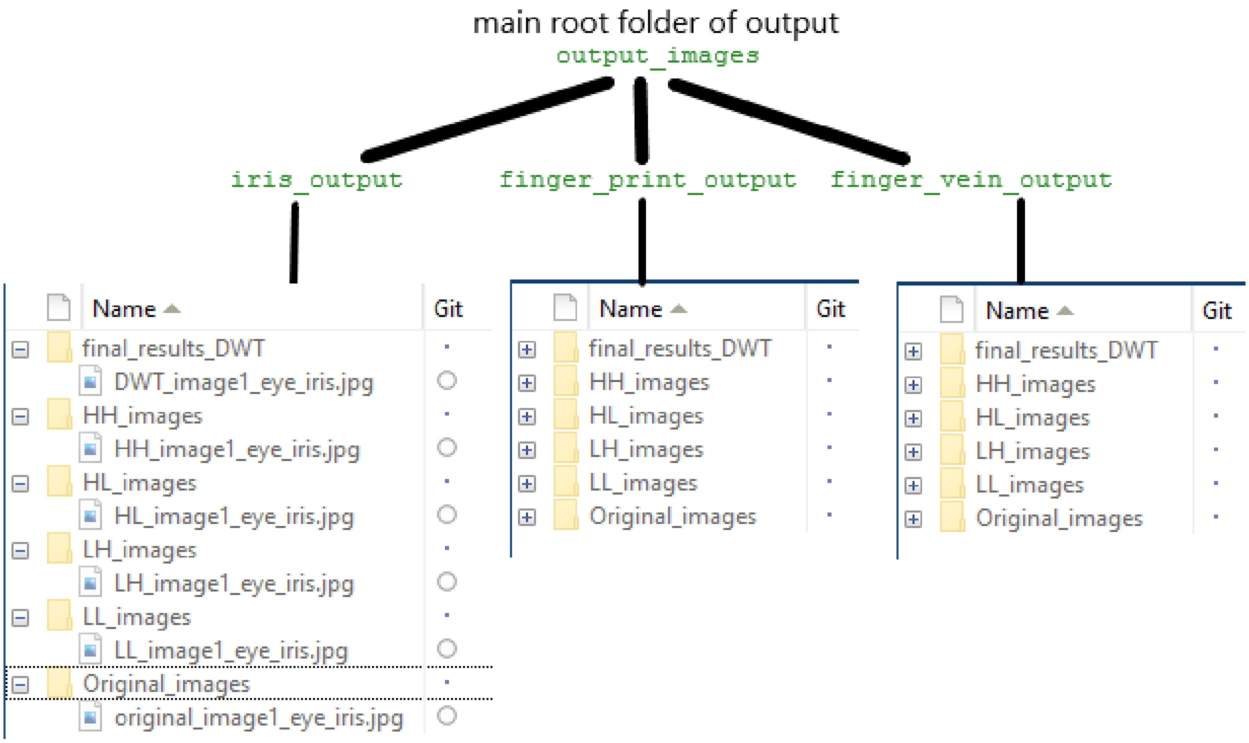
Put next to each other:

1. the original image of an iris and its Fourier transform
2. the 1st output (of discrete wavelet transform) and Fourier transform
3. the 2nd output (of discrete wavelet transform) and its Fourier transform
4. the 3rd output (of discrete wavelet transform) and its Fourier transform
5. the 4th output (of discrete wavelet transform) the Fourier transform

MATLAB Implementation

Code will be separated into three files, one will be the main file where the execution will happen step by step the 'main.m' file, the other one the 'DWT_function.m' file, implemented as a function and the 'furier_function.m' that will calculate the furrier function.

Doing such work, we need to be precise, according to the paths and directories. Images are saved according to a root and folder, with specific names according to the following schema:



The folders will be created automatically if they do not exist, and according to root and sub-root everything will be precise.

Project Code

The 'main.m' file that will call and execute the respective functions.

```
clc, clear all, close all

fprintf('Cen Master, Epoka Univesity, Computer Engenieering\n===Discrete Wavelet
Transform===\n');
% DWT Part
% NOTE: The script images should run one by one, uncomment, run , comment,
% go to next and so on.
% Creating the folders according to the root

% ======== Iris Part - 10 images ========
% creating root folder for the output
if ~exist('output_images\iris_output', 'dir')
   mkdir('output_images\iris_output')
end

DWT_function('output_images\iris_output','images\iris\','image1_eye_iris','.jpg');%1
% DWT_function('output_images\iris_output','images\iris\','image2_eye_iris','.jpg');%2
% DWT_function('output_images\iris_output','images\iris\','image3_eye_iris','.jpg');%3
% DWT_function('output_images\iris_output','images\iris\','image4_eye_iris','.jpg');%4
% DWT_function('output_images\iris_output','images\iris\','image5_eye_iris','.jpg');%5
% DWT_function('output_images\iris_output','images\iris\','image6_eye_iris','.jpg');%6
% DWT_function('output_images\iris_output','images\iris\','image7_eye_iris','.jpg');%7
% DWT_function('output_images\iris_output','images\iris\','image8_eye_iris','.jpg');%8
% DWT_function('output_images\iris_output','images\iris\','image9_eye_iris','.jpg');%9
% DWT_function('output_images\iris_output','images\iris\','image10_eye_iris','.jpg');%10

% ======== Finger Print Part - 10 images ========
% creating root folder for the output

if ~exist('output_images\finger_print_output', 'dir')
   mkdir('output_images\finger_print_output')
end
%
```

```
%
DWT_function('output_images\finger_print_output','images\finger_print\','image1_finger_prin
t','.jpg');%1
%
DWT_function('output_images\finger_print_output','images\finger_print\','image2_finger_prin
t','.jpg');%2
%
DWT_function('output_images\finger_print_output','images\finger_print\','image3_finger_prin
t','.jpg');%3
%
DWT_function('output_images\finger_print_output','images\finger_print\','image4_finger_prin
t','.jpg');%4
%
DWT_function('output_images\finger_print_output','images\finger_print\','image5_finger_prin
t','.jpg');%5
%
DWT_function('output_images\finger_print_output','images\finger_print\','image6_finger_prin
t','.jpg');%6
%
DWT_function('output_images\finger_print_output','images\finger_print\','image7_finger_prin
t','.jpg');%7
%
DWT_function('output_images\finger_print_output','images\finger_print\','image8_finger_prin
t','.jpg');%8
%
DWT_function('output_images\finger_print_output','images\finger_print\','image9_finger_prin
t','.jpg');%9
%
DWT_function('output_images\finger_print_output','images\finger_print\','image10_finger_pri
nt','.jpg');%10




% ======== Finger Vain Part - 10 images ========
if ~exist('output_images\finger_vein_output', 'dir')
   mkdir('output_images\finger_vein_output')
end
```

```
%
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image1_finger_vein',
'.jpg');%1
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image2_finger_vein',
'.jpg');%2
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image3_finger_vein',
'.jpg');%3
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image4_finger_vein',
'.jpg');%4
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image5_finger_vein',
'.jpg');%5
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image6_finger_vein',
'.jpg');%6
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image7_finger_vein',
'.jpg');%7
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image8_finger_vein',
'.jpg');%8
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image9_finger_vein',
'.jpg');%9
%
DWT_function('output_images\finger_vein_output','images\finger_vein\','image10_finger_vein'
,'.jpg');%10
```

In the following code we run each image one by one, in a way not to mix the results and keeping track on what we are doing.

The other will be the ‘DWT_function.m’ this will be the part of the Discrete Wavelet transform that will take an image input and will make the respective calculations.

```
function [] =
DWT_function(output_folder,image_path_location,image_to_proceed,image_extensi
on)
```

the first one will be the main folder of output, then the image path, then the image name and the extension at the end, so due to the fact that we had to work on multiple files and high volume we need to save them properly in a automatic mode.

```
function [] =
DWT_function(output_folder,image_path_location,image_to_proceed,image_extensi
on)
%  Join location strings
path_location=strcat(image_path_location,image_to_proceed,image_extension);

input_image = imread(path_location);
input_image=rgb2gray(input_image);% convert Gray
% Original Image
% figure(1)
% imshow(input_image);
% title('Original Image');
% resizing the image
size_of_output_result_image=size(input_image);

[LL,LH,HL,HH]=dwt2(input_image,'db1');

% processingd Discrete Wavelet transform with levels
figure(1)
subplot(2,3,1);imshow(input_image);title('Original Image');
subplot(2,3,2);imshow(LL);title('LL band of image');
subplot(2,3,3);imshow(LH);title('LH band of image');
subplot(2,3,5);imshow(HL);title('HL band of image');
subplot(2,3,6);imshow(HH);title('HH band of image');
output_result_transformed_image =
idwt2(LL,LH,HL,HH,'db1',size_of_output_result_image);
```

```
subplot(2,3,4);imshow(output_result_transformed_image);
title('Transformed Image DWT');
% increase size of image to display united
set(figure(1),'units','points','position',[0,0,820,450])

%0-Saving the original image

original_path=strcat(output_folder,'\Original_images\');
if ~exist(original_path, 'dir')
   mkdir(original_path)
end

original_path=strcat(original_path,'original_',image_to_proceed,image_extensi
on);

original_image = figure(8);
set(original_image,'Visible','off');
imshow(input_image);
saveas(original_image,original_path);




% creating path for final result according to the root, but checking it
% first, ex 'output_images\iris_output'
% Creating final result folder -> \final_results_DWT\
transformed_result_image=strcat(output_folder,'\final_results_DWT\');
if ~exist(transformed_result_image, 'dir')
   mkdir(transformed_result_image)
end
%Adding file with the extension to the string
transformed_result_image=strcat(transformed_result_image,'DWT_',image_to_proc
eed,image_extension);

% Discrete Wavelet transform image
main_output_image = figure(2);
saveas(imshow(output_result_transformed_image),transformed_result_image);
```

```
% imshow(output_result_transformed_image)
% title('Transformed Image DWT');
set(main_output_image,'Visible','off');% close image after if needed to be
closed



% Working with each output step by step
% storing outputs on separated variables
% Saving images LL (low-low), LH(low-high), HL(high-low), HH(high-high)

% Check if path exist, if not we create it

% 1 LL path
LL_path=strcat(output_folder,'\LL_images\');
if ~exist(LL_path, 'dir')
   mkdir(LL_path)
end

LL_path=strcat(LL_path,'LL_',image_to_proceed,image_extension);



image_1_LL = figure(4);
set(image_1_LL,'Visible','off');%image visible off
imshow(LL);
saveas(image_1_LL,LL_path);

% 2 LH Path


LH_path=strcat(output_folder,'\LH_images\');
if ~exist(LH_path, 'dir')
   mkdir(LH_path)
end

LH_path=strcat(LH_path,'LH_',image_to_proceed,image_extension);

image_2_LH = figure(5);
```

```
set(image_2_LH,'Visible','off');
imshow(LH);
saveas(image_2_LH,LH_path);

% 3 HL path creation
HL_path=strcat(output_folder,'\HL_images\');
if ~exist(HL_path, 'dir')
   mkdir(HL_path)
end

HL_path=strcat(HL_path,'HL_',image_to_proceed,image_extension);

image_3_HL = figure(6);
set(image_3_HL,'Visible','off');
imshow(HL);
saveas(image_3_HL,HL_path);

% 4 HH image path creation
HH_path=strcat(output_folder,'\HH_images\');
if ~exist(HH_path, 'dir')
   mkdir(HH_path)
end

HH_path=strcat(HH_path,'HH_',image_to_proceed,image_extension);

image_4_HH = figure(7);
set(image_4_HH,'Visible','off');
imshow(HH);
saveas(image_4_HH,HH_path);

% Fourier Transform part, each
% image output will be included into the function fourier transform

% 1-Low-Low image
furier_function(LL_path,2)
% 2-Low-High image
```

```
furier_function(LH_path,3);
% 3-High-Low image
furier_function(HL_path,4);
% 4-High-high image
furier_function(LH_path,5);

end
```

The upper code will take the image as input, join the folders path and locations under a single location, loading the image, calculating the 4-outptus and the final result image, then images will be plotted and saved.

```
% Fourier Transform part, each
% image output will be included into the function fourier transform

% 1-Low-Low image
furier_function(LL_path,2)
% 2-Low-High image
furier_function(LH_path,3);
% 3-High-Low image
furier_function(HL_path,4);
% 4-High-high image
furier_function(LH_path,5);
```

Each of 4 outputs will be proceed and pass thought a Fourier transform function, according to the outputs that will be taken as input by this function named

```
furier_function(image_path,2)
```

The number 2 will be the image name ex: “figure 2”, the code will be shown below;

```
function [] = furier_function(path_image_location, fig_num)
% First we will be loading image
imdata =imread(path_image_location);
figure(fig_num)
subplot(2,3,1);imshow(imdata); title('Original Image');
```

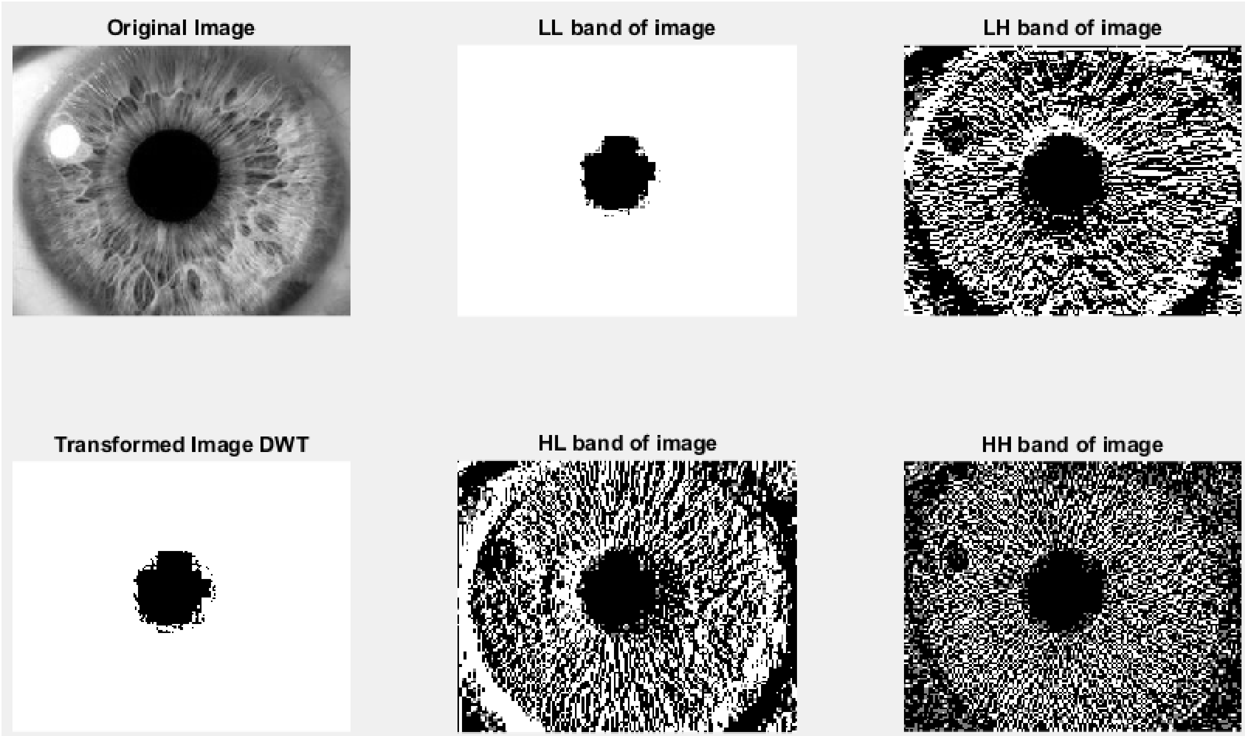
```
imdata = rgb2gray(imdata);
subplot(2,3,2); imshow(imdata); title('Gray Image');
%Get Fourier Transform of an image
F = fft2(imdata);
% Fourier transform of an image
S = abs(F);
subplot(2,3,3);imshow(S,[]);title('Fourier transform of an image');
%get the centered spectrum
Fsh = fftshift(F);
subplot(2,3,4);imshow(abs(Fsh),[]);title('Centered fourier transform of
Image')
%apply log transform
S2 = log(1+abs(Fsh));
subplot(2,3,5);imshow(S2,[]);title('log transformed Image')
%reconstruct the Image
F = ifftshift(Fsh);
f = ifft2(F);
subplot(2,3,6) ;imshow(f,[]),title('reconstructed Image')

% increase image size
set(figure(fig_num),'units','points','position',[0,0,820,450])
end
```

Image will be taken, and proceed, converted in gray color, passing thought Fourier functions and plotted according to the respective calculations.

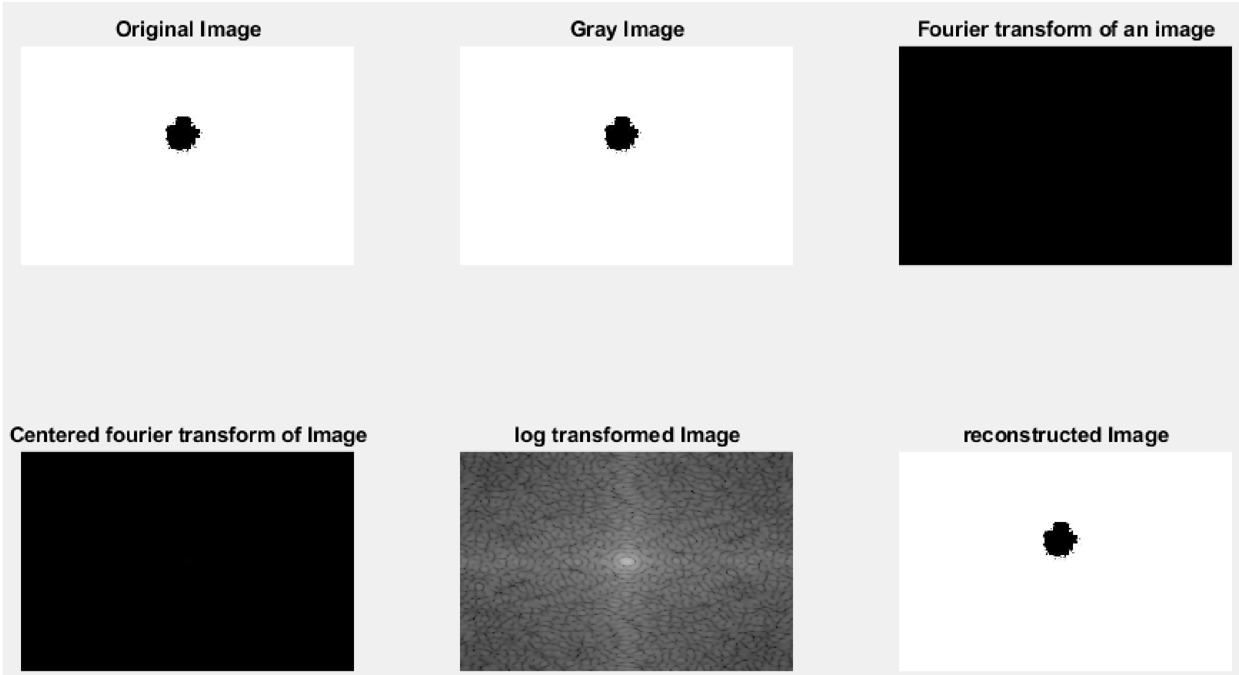
Proceeding with the images

Iris part, the 10 images analysis with the respective Fourier transform part for each output,

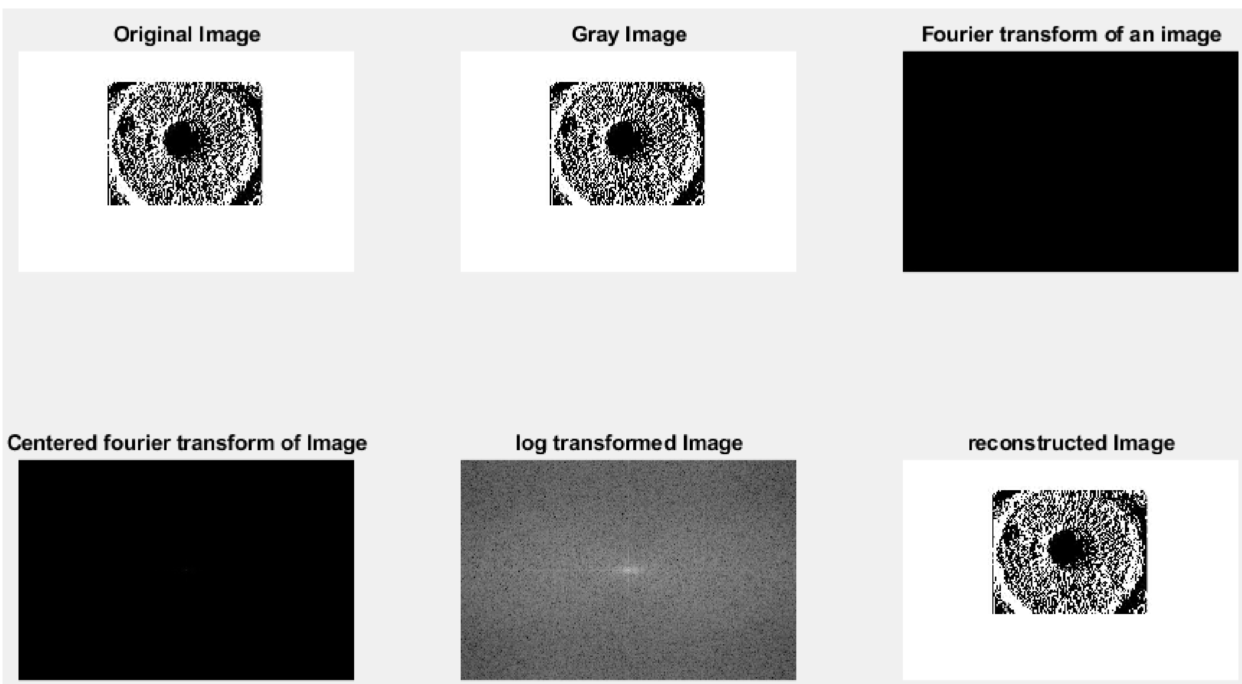
DWT


Original Image
LL band of image
LH band of image
Transformed Image DWT
HL band of image
HH band of image

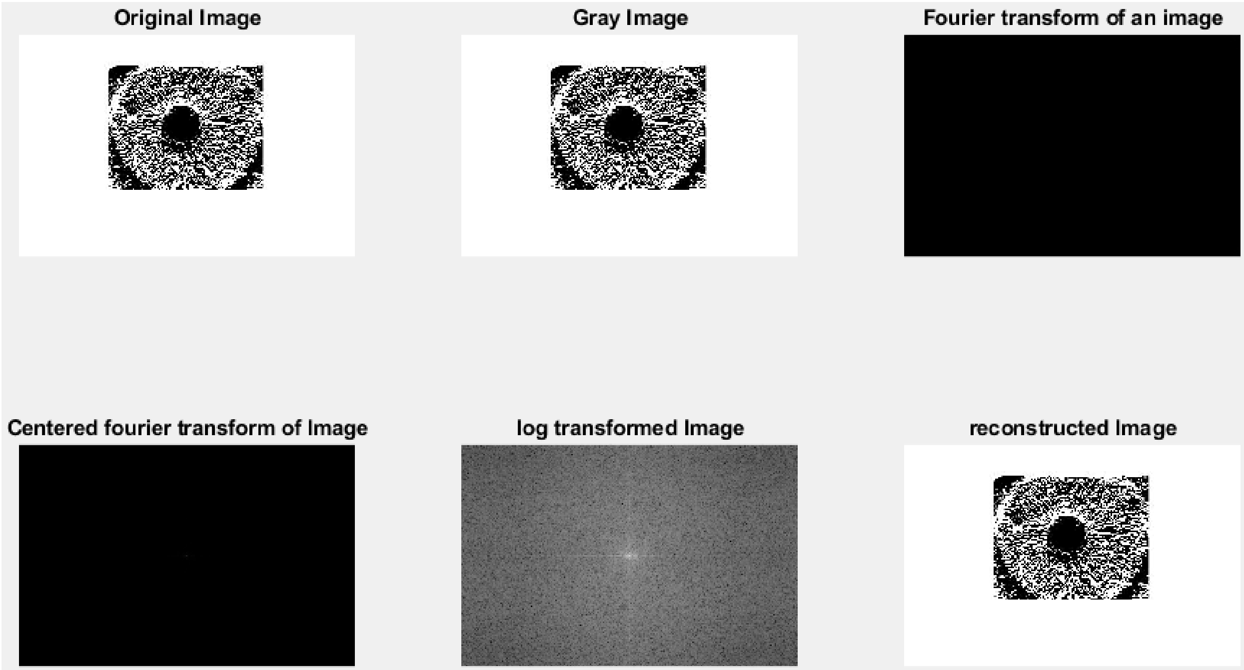
Fourier Part for LL image



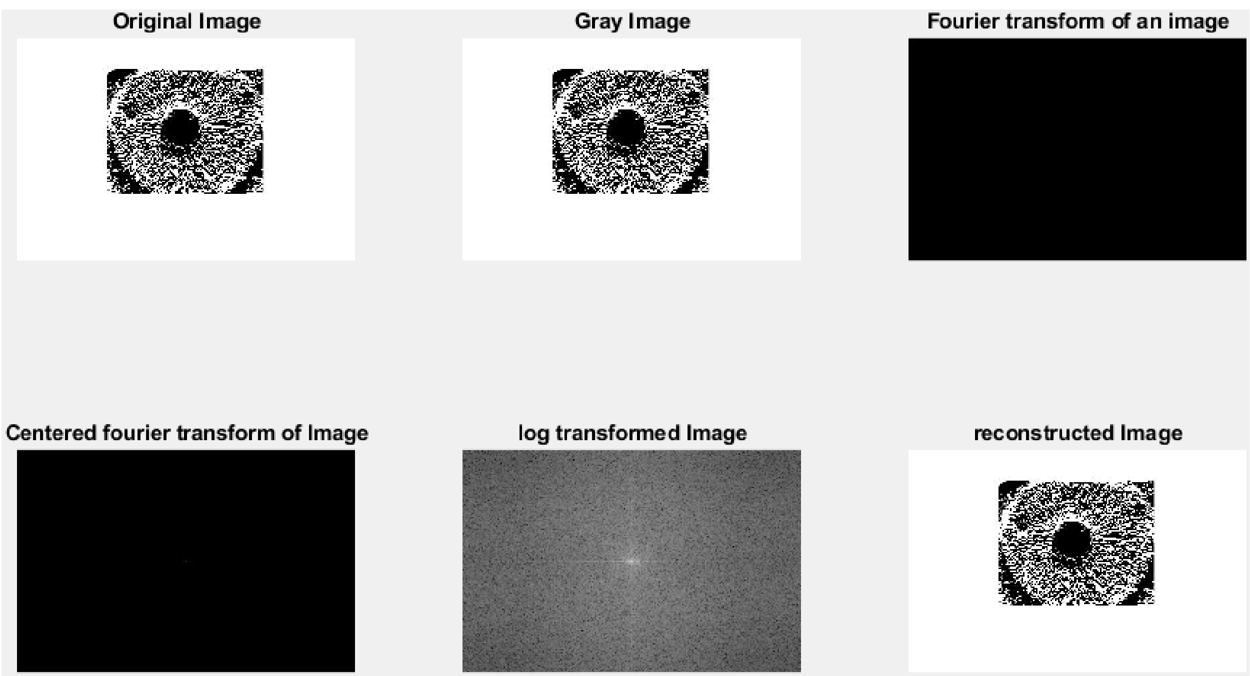
LH Fourier



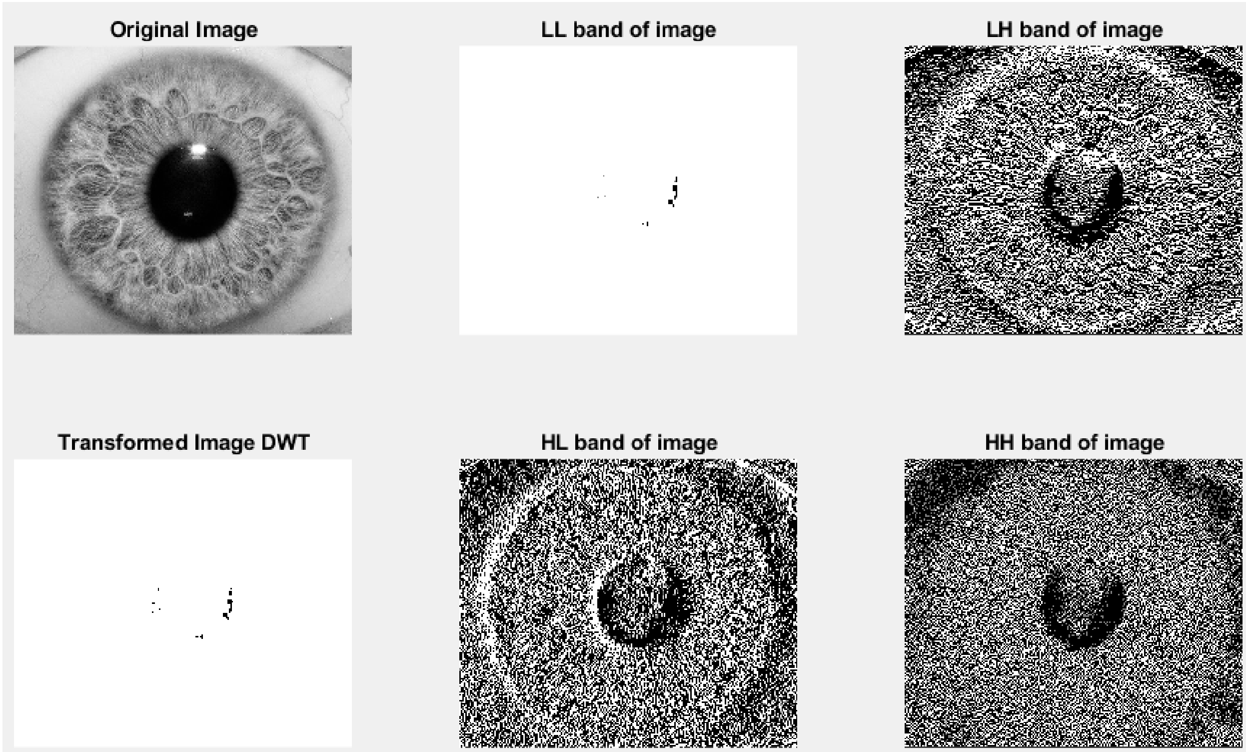
HL Fourier



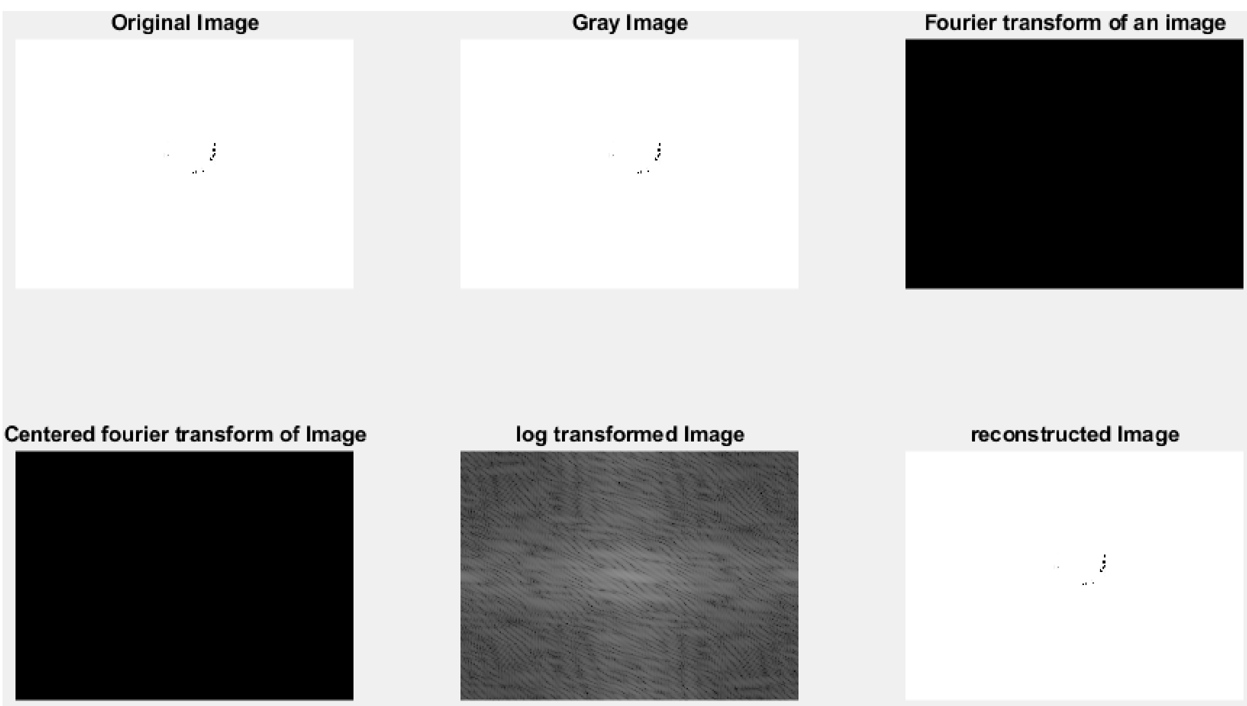
HH Fourier



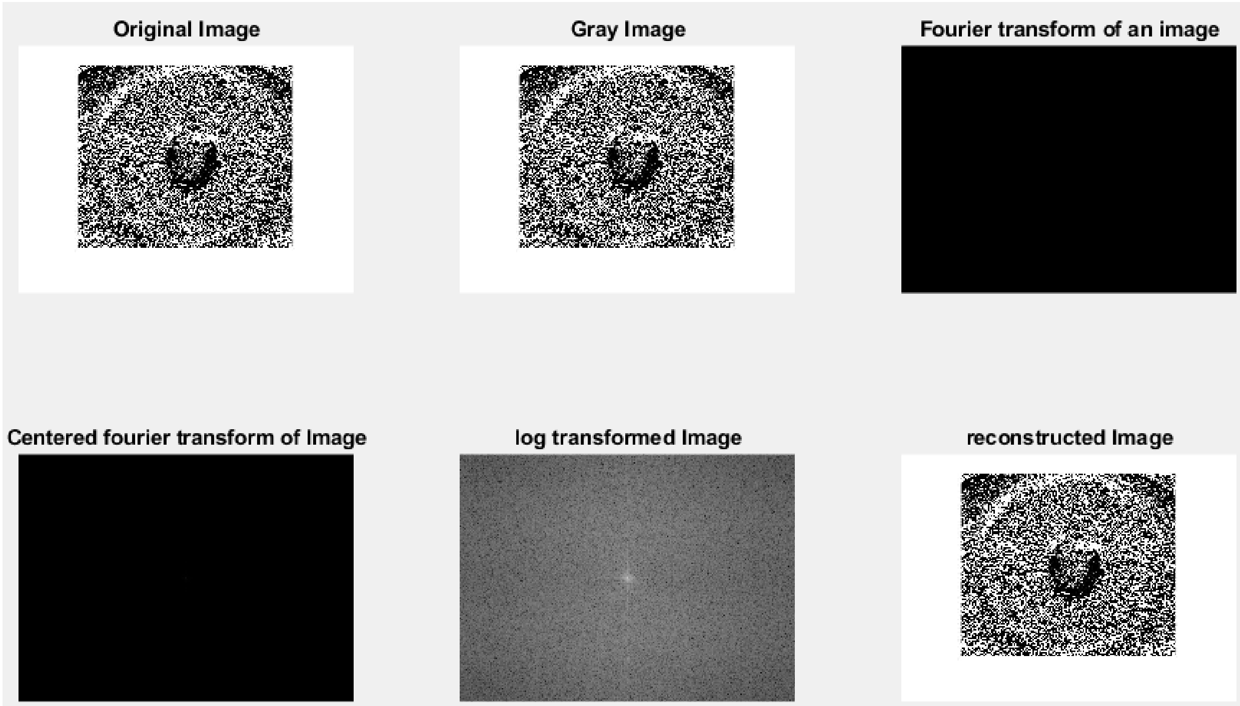
Iris, image 2 DWT



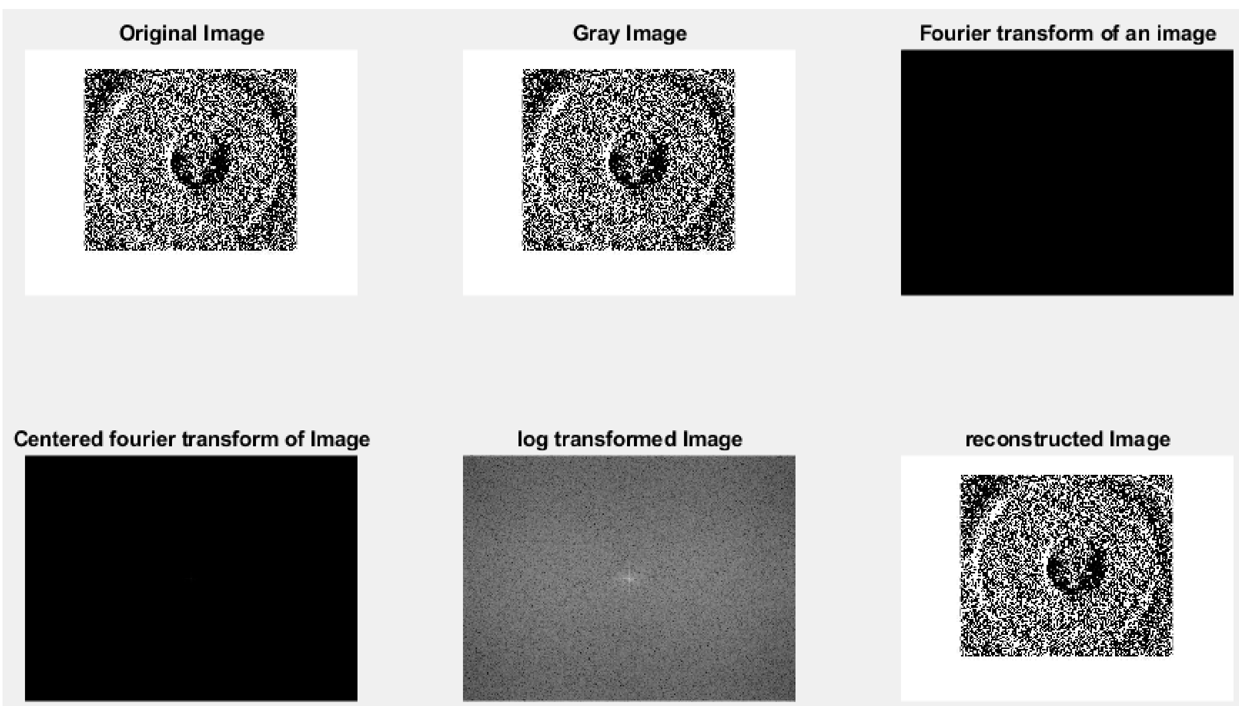
LL Fourier



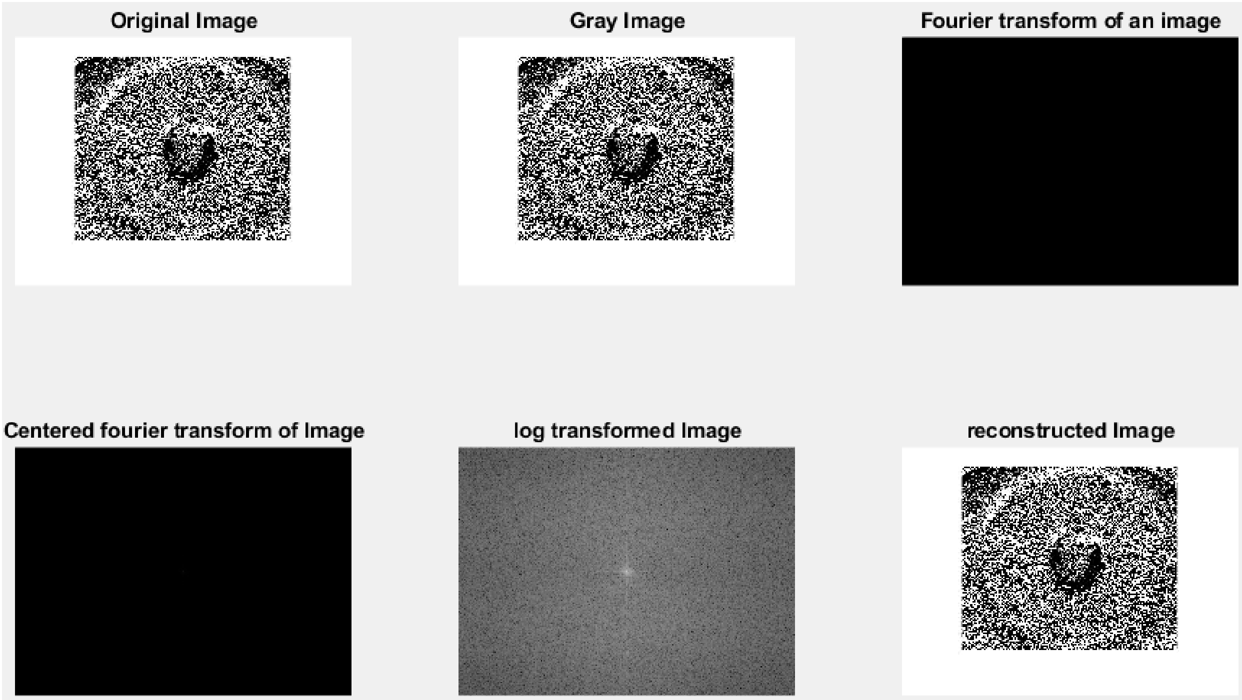
LH Fourier



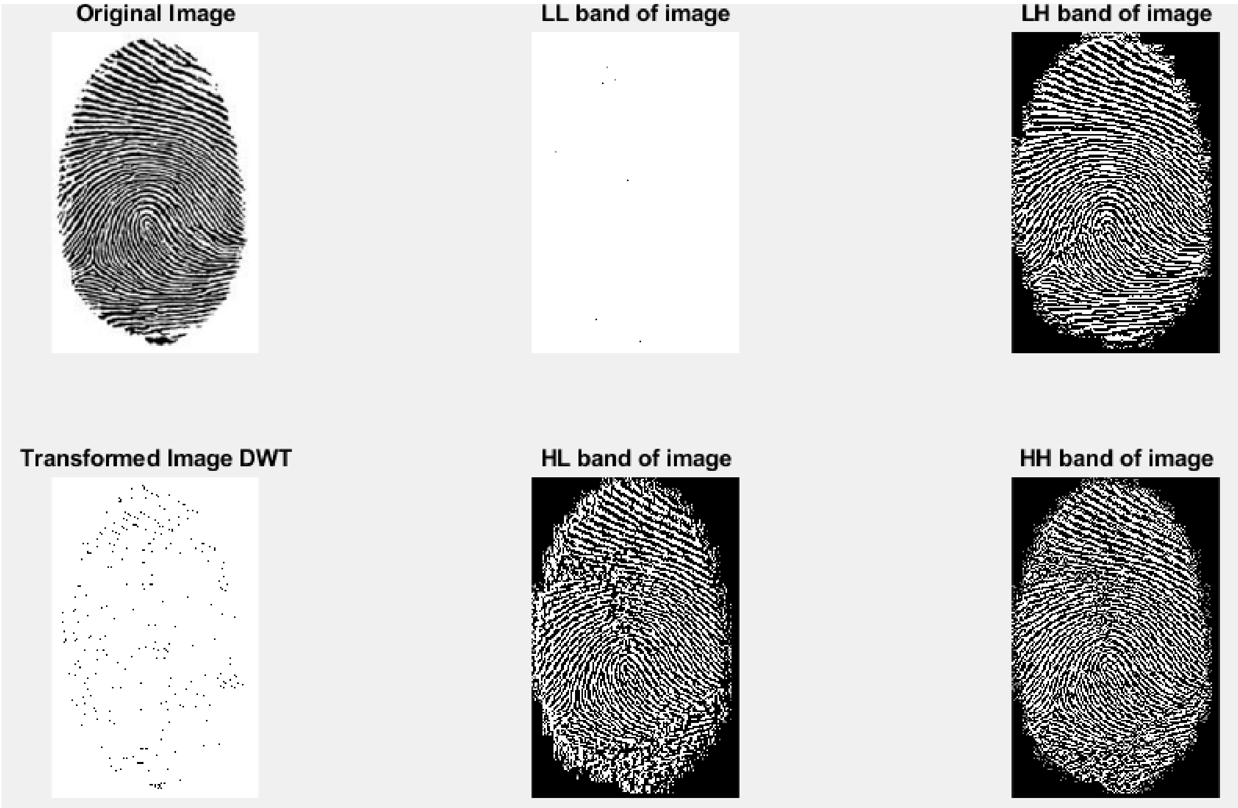
HL Fourier



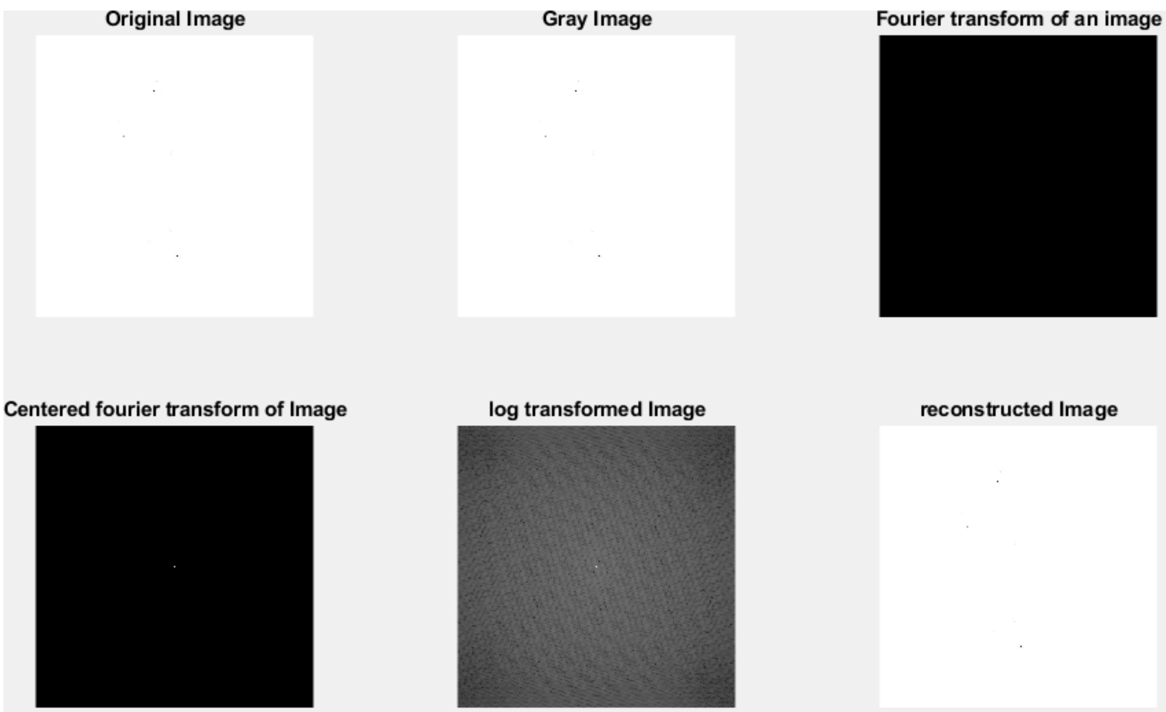
HH Fourier


Original Image
Gray Image
Fourier transform of an image
Centered fourier transform of Image
log transformed Image
reconstructed Image

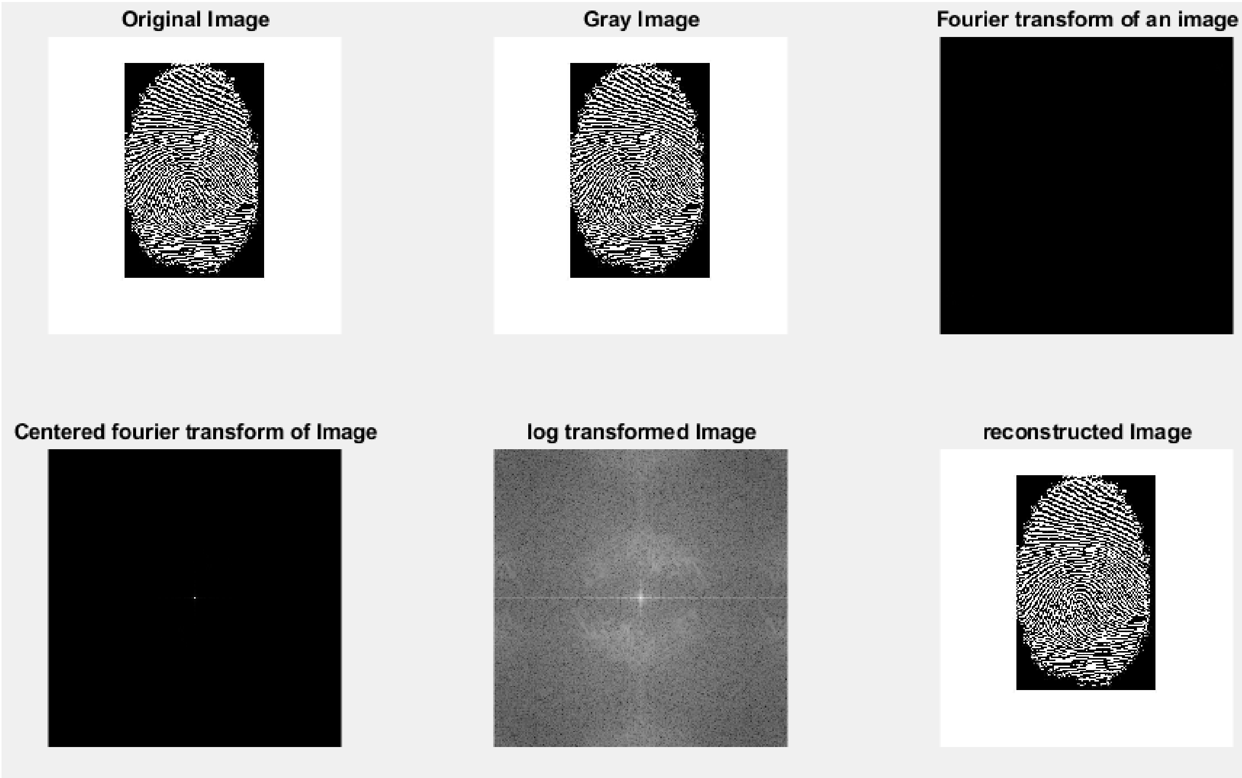
Finger Print DWT,



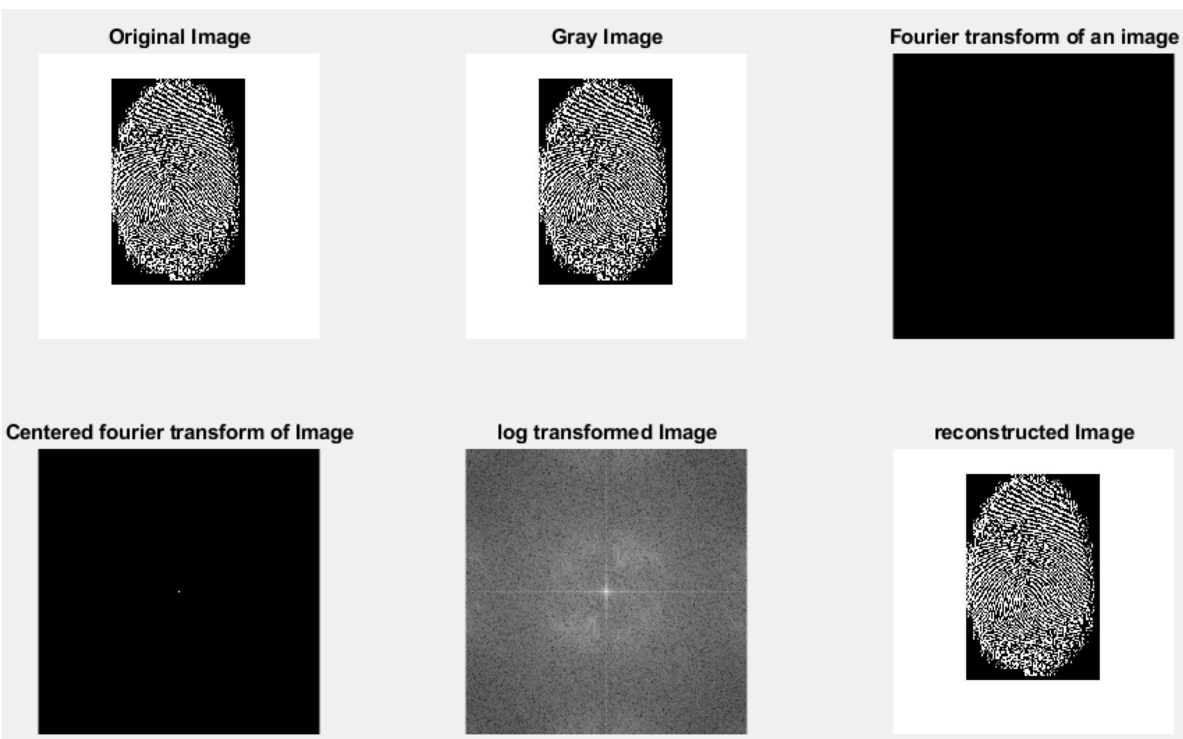
LL Fourier



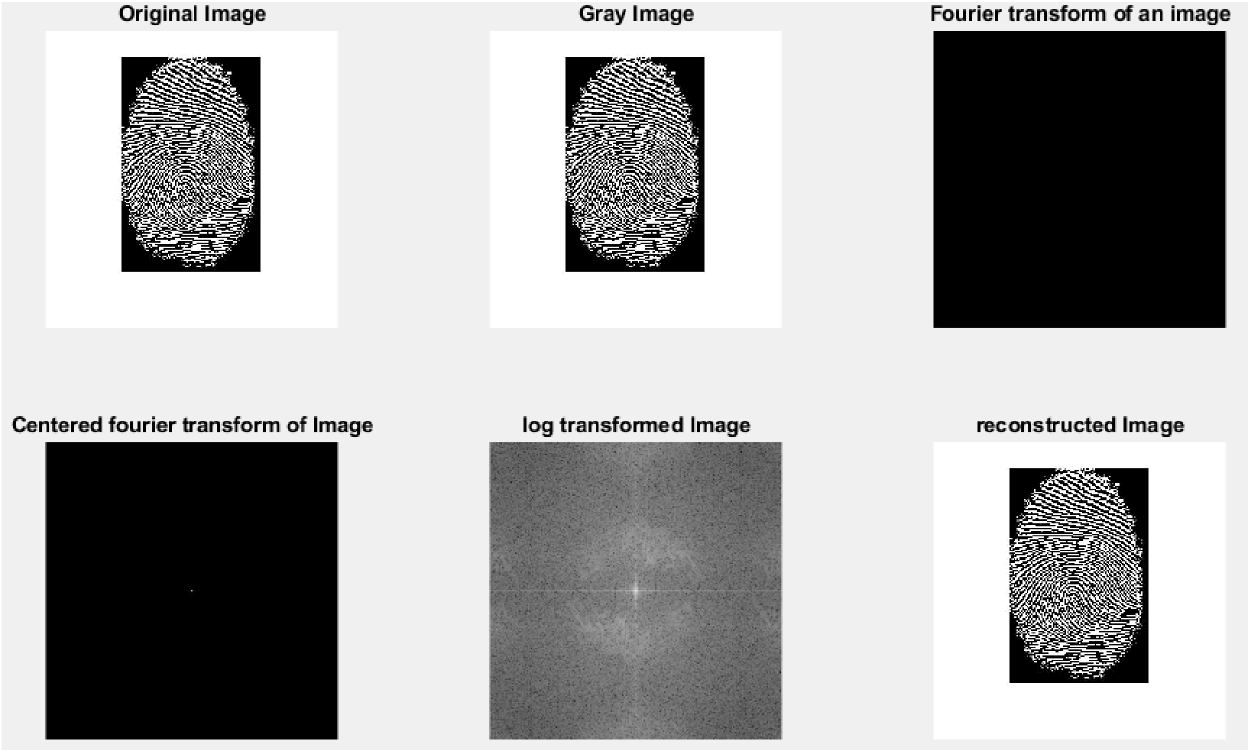
LH Fourier



HL Fourier

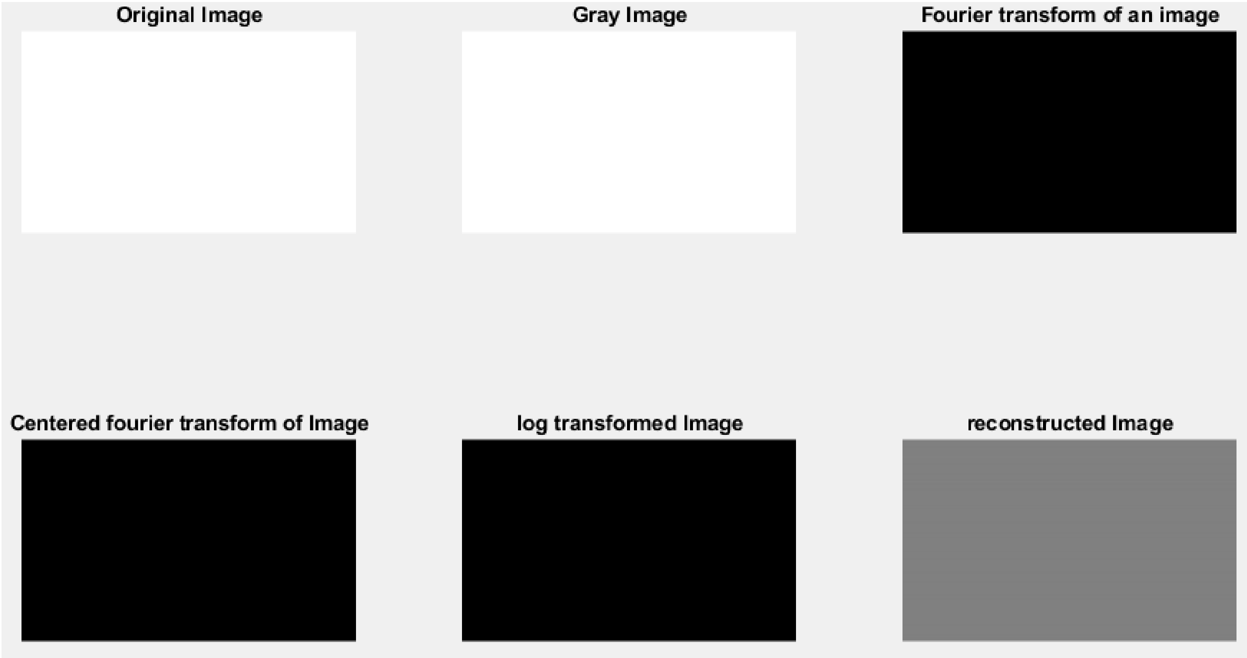


HH Fourier

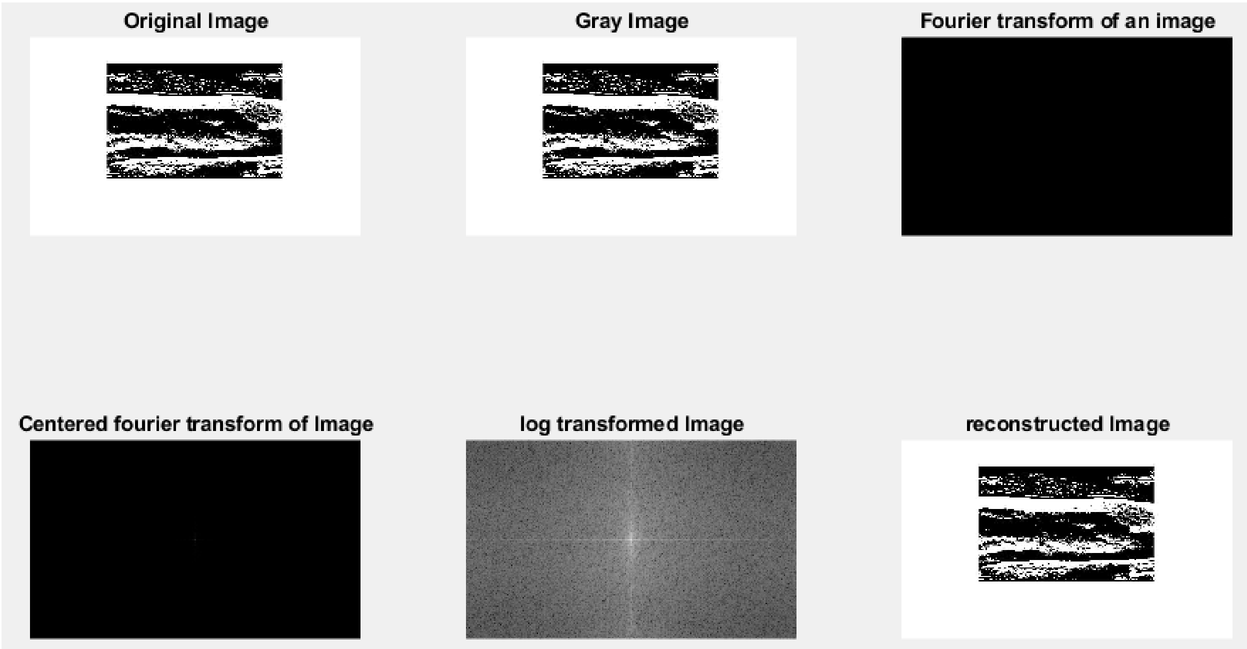


Finger Vein DWT,

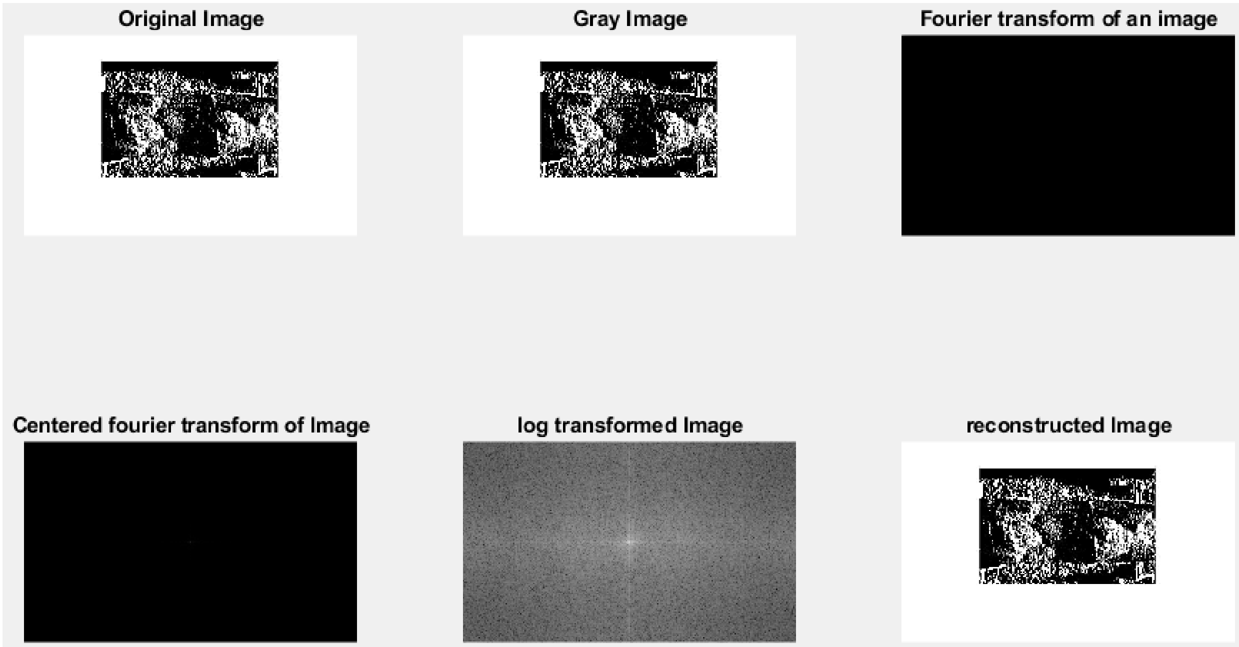
LL Fourier,



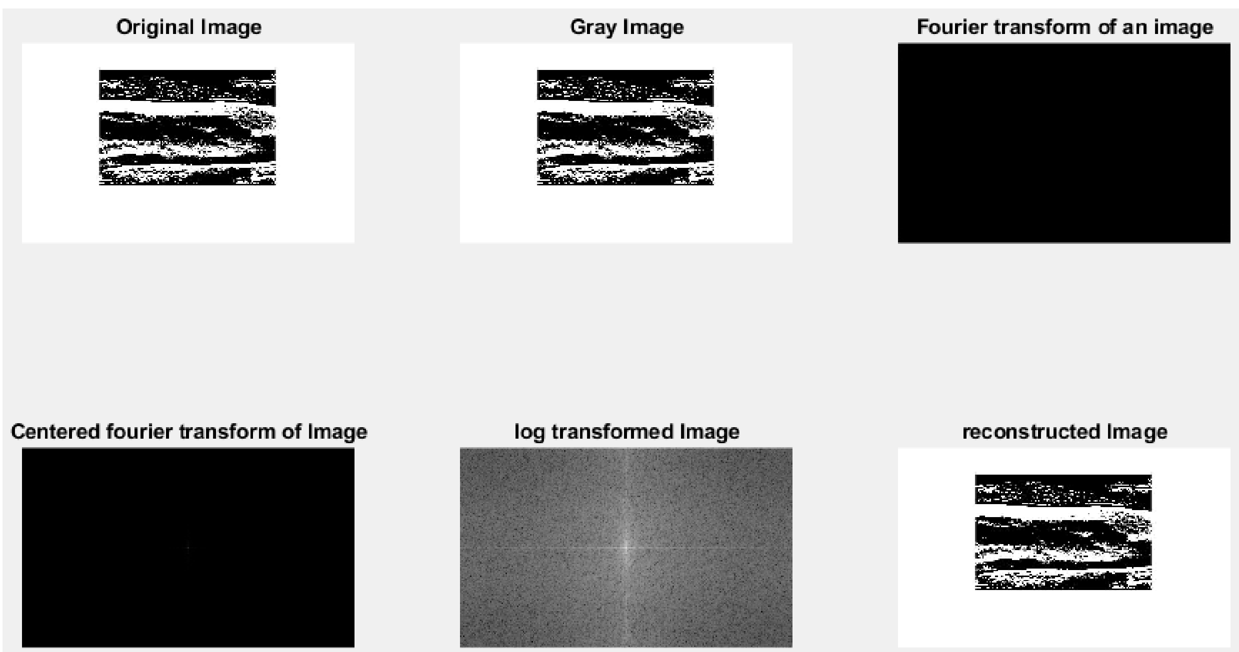
LH Fourier,



HL Fourier,



HH Fourier,



Conclusions

Doing such analyze was an good experience, the studying of the images and how it was processed from the MATLAB, then again produced as output, there would be always images of different sizes as input, and the output would be four images or four sub-bands, where the LL sub-band will output the maximum energy or the information after the first level of decomposition, is contained in the LL (low-low) band.

The output images will be one resolution smaller than the images taken to be processed. We have lost the resolution of a factor of 2 along the rows and columns of the image. In other side The Fourier transform plays a critical role in a broad range of image processing applications, including enhancement, analysis, restoration, and compression.