# Yash Patel, 201301134 %%

## Table of Contents

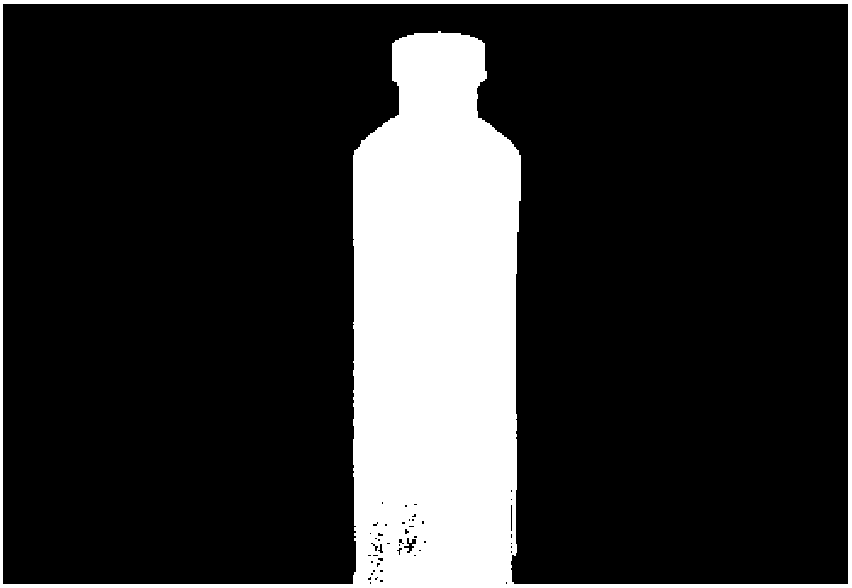
Find the imperfect bottle. % ........ 1
Yash Patel, 201301134 %% ........ 7
Question-3 %% ........ 8
Classifying yoga-assanans. %% ........ 11

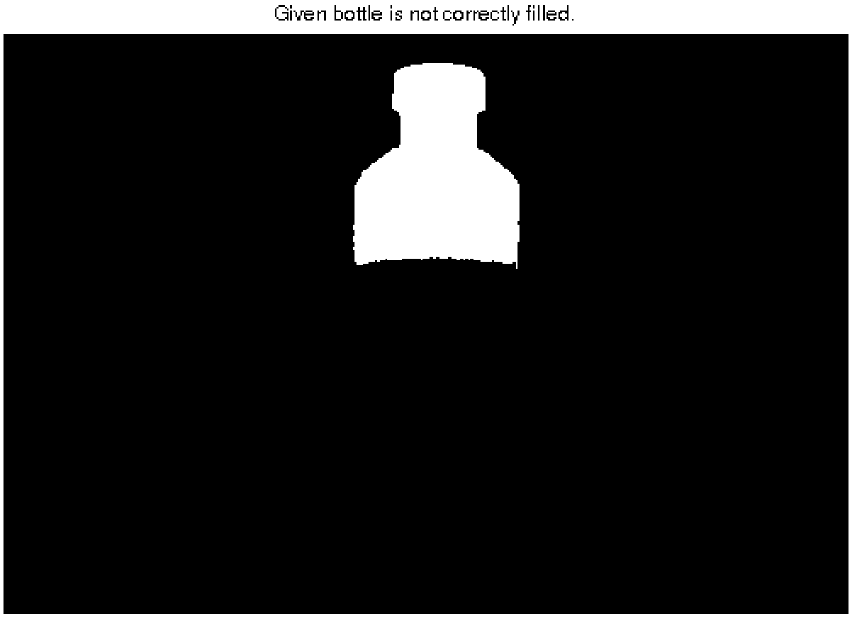
# Find the imperfect bottle. %

```
clc
clear all
% Read the image. %
im = imread('bottles.tif');

% Convert this image to binary image. %
im_bw = im2bw(im, 0.8);
im_bw_bottles = im2bw(im,0.2);


% Iteratively find the connected components. %
for i=1:5

    % Find the connected components. %
    im_cc = bwconncomp(im_bw);
    im_cc_bottles = bwconncomp(im_bw_bottles);

    % Find the largest connected comp. in the image. %
    numPixels = cellfun(@numel,im_cc.PixelIdxList);
    [biggest,idx] = max(numPixels);
    numPixels_bottles = cellfun(@numel,im_cc_bottles.PixelIdxList);
    [biggest_bottles, idx_bottles] = max(numPixels_bottles);


    % Make image, negation of conncomp. %
    im_cnn = zeros(size(im_bw));
    im_cnn_bottles = zeros(size(im_bw_bottles));

    % Make the conn comp pixels of this as one. %
    im_cnn(im_cc.PixelIdxList{idx}) = 255;
    im_cnn_bottles(im_cc_bottles.PixelIdxList{idx_bottles}) = 255;

    % Convert this to binary image and show. %
    im_cnn_bw = im2bw(im_cnn, 0.5);
    im_cnn_bw_bottles = im2bw(im_cnn_bottles, 0.5);

    % Find the num of pixels in given conn components. %
    a = find(im_cnn_bw);
    a_bottle = find(im_cnn_bw_bottles);
    size(a_bottle,1);
    figure, imshow(im_cnn_bw_bottles), title('Considering given bottle.')
    if size(a_bottle,1)/size(a,1) < 5
        val = 'Given bottle is not correctly filled.';
        disp(val)
        figure, imshow(im_cnn_bw), title(val)

    else
        val = 'Given bottle is correctly filled.';
        disp(val)
        figure, imshow(im_cnn_bw), title(val)

    end

    % Change the given image by removing the largest cnncomp. %
    im_bw(im_cc.PixelIdxList{idx}) = 0;
    im_bw_bottles(im_cc_bottles.PixelIdxList{idx_bottles}) = 0;

end
```

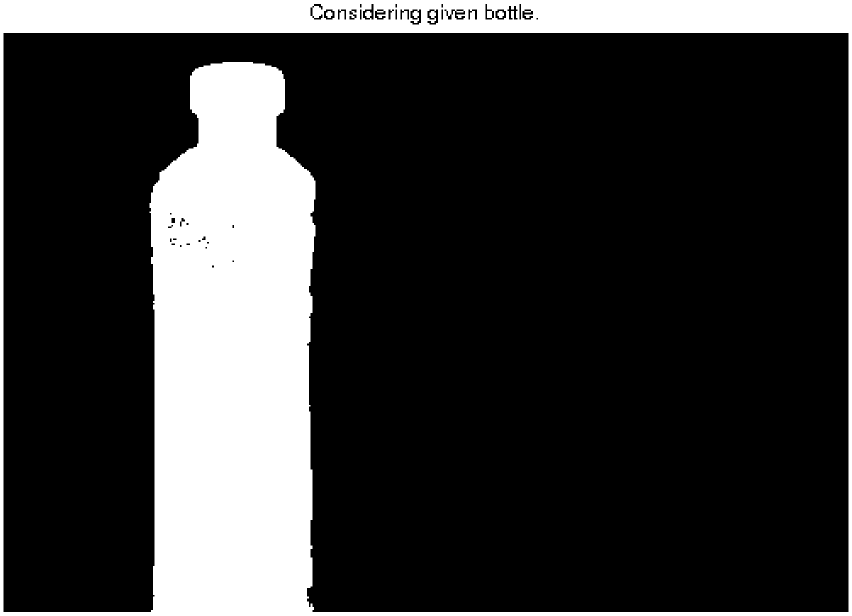
```
        Given bottle is not correctly filled.

        Given bottle is correctly filled.
        Given bottle is correctly filled.
        Given bottle is correctly filled.
        Given bottle is correctly filled.
```

Considering given bottle.

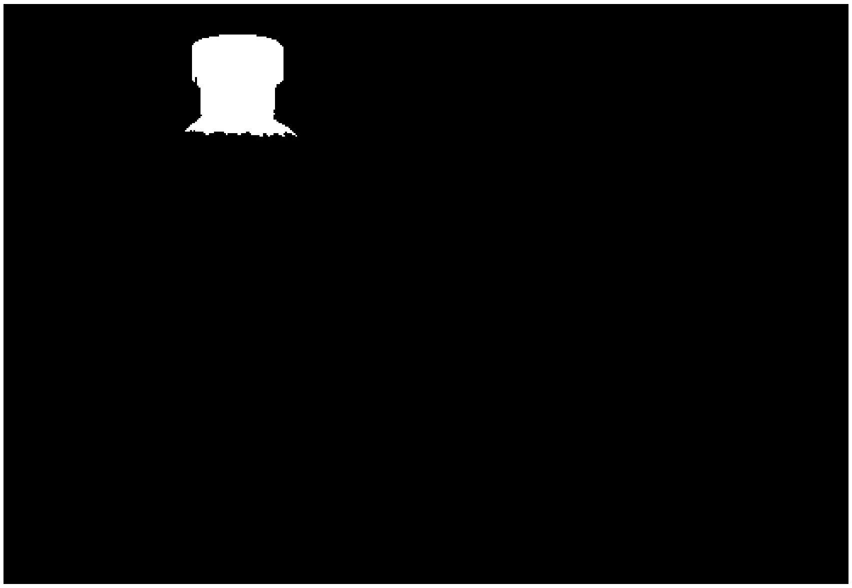




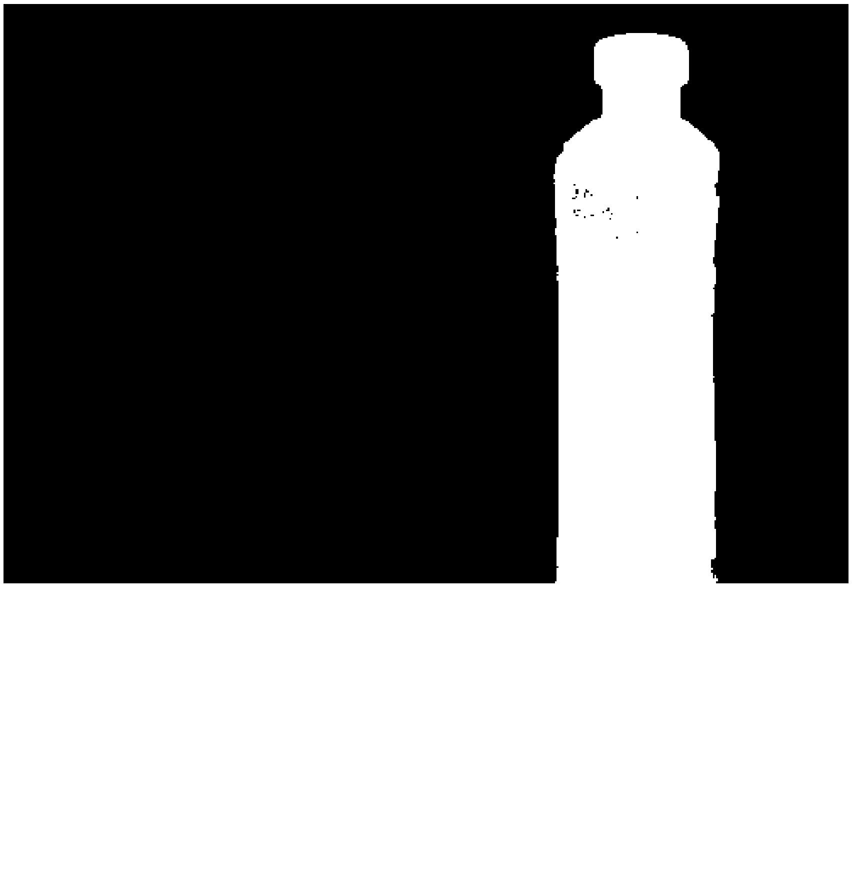
Given bottle is not correctly filled.



Considering given bottle.



Given bottle is correctly filled.



Considering given bottle.

Given bottle is correctly filled.

Considering given bottle.

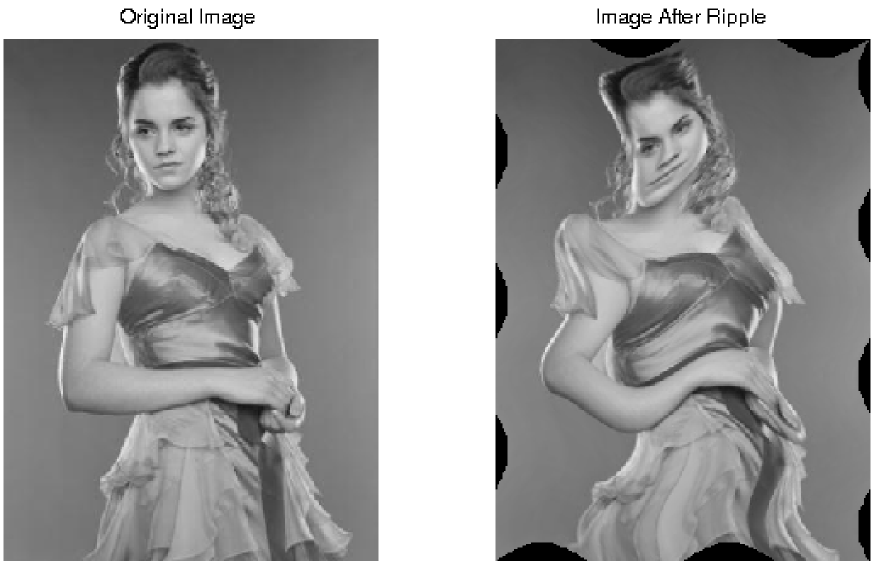
Given bottle is correctly filled.

Considering given bottle.

Given bottle is correctly filled.

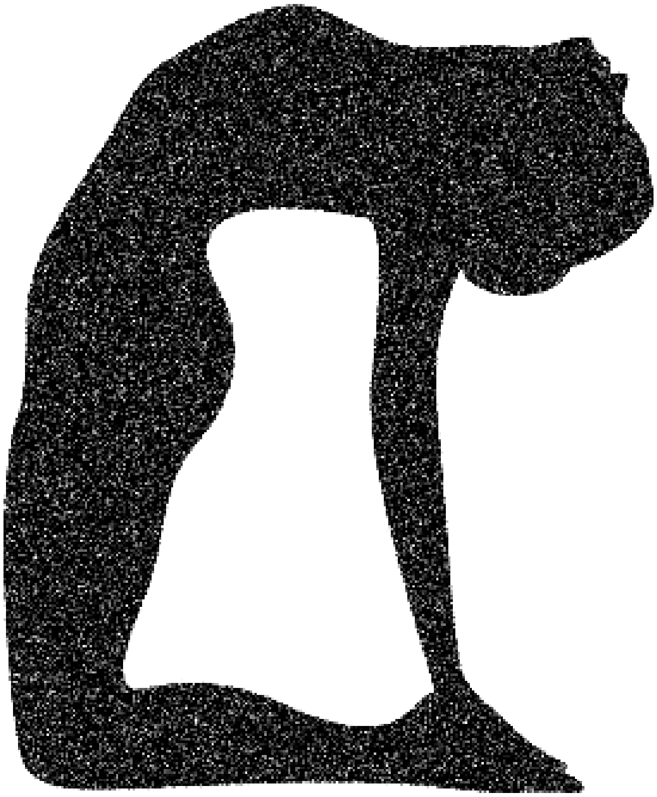
# Yash Patel, 201301134 %%

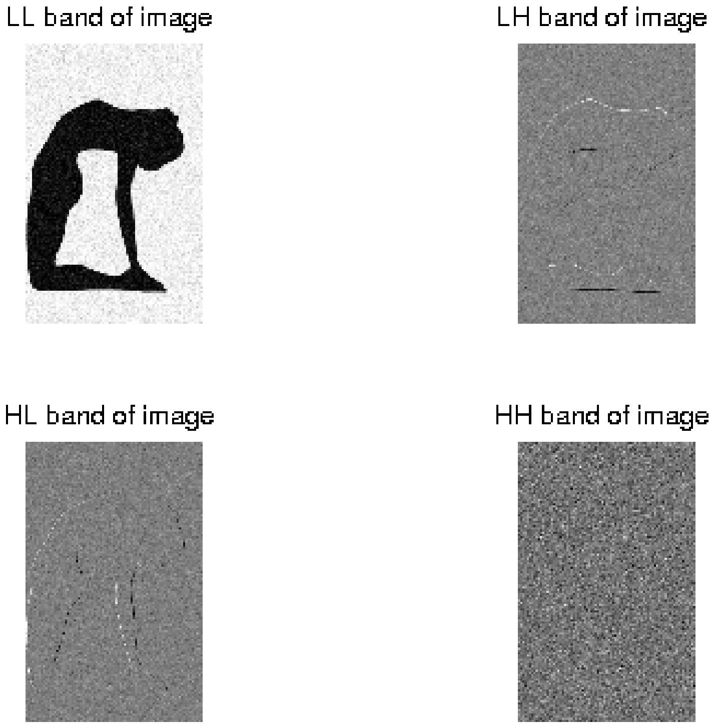
```
clc
clear all
% Ripple tranform. %


ax = 10;
ay = 15;
tx = 120;
ty = 150;
im = im2double(rgb2gray((imread('q4_image.jpg'))));
subplot(1,2,1); imshow(im); title('Original Image');
[x, y] = meshgrid(1:size(im,2), 1:size(im,1));
x_new = x + ax*sin(2*pi*y/tx);
y_new = y + ay*sin(2*pi*x/ty);
final = interp2( x, y, im, x_new, y_new, 'linear');
subplot(1,2,2); imshow(final); title('Image After Ripple');
```



# Question-3 %%

```
clc
clear all
% Read the Image. %
im = im2double(rgb2gray(imread('./yogasan/1.jpg')));

% Add Gaussian noise. %
im_noise = imnoise(im,'gaussian',0,0.1);
im = im + im_noise;
figure;imshow(im);


[cA,cH,cV,cD]=dwt2(im,'haar');
figure, subplot(2,2,1);imshow(cA,[]);title('LL band of image');
subplot(2,2,2);imshow(cH,[]);title('LH band of image');
subplot(2,2,3);imshow(cV,[]);title('HL band of image');
subplot(2,2,4);imshow(cD,[]);title('HH band of image');

%Thresholding
cH = zeros(size(cH));
cV = zeros(size(cV));
cD = zeros(size(cD));

figure;imshow(idwt2(cA,cH,cV,cD,'haar'));
```



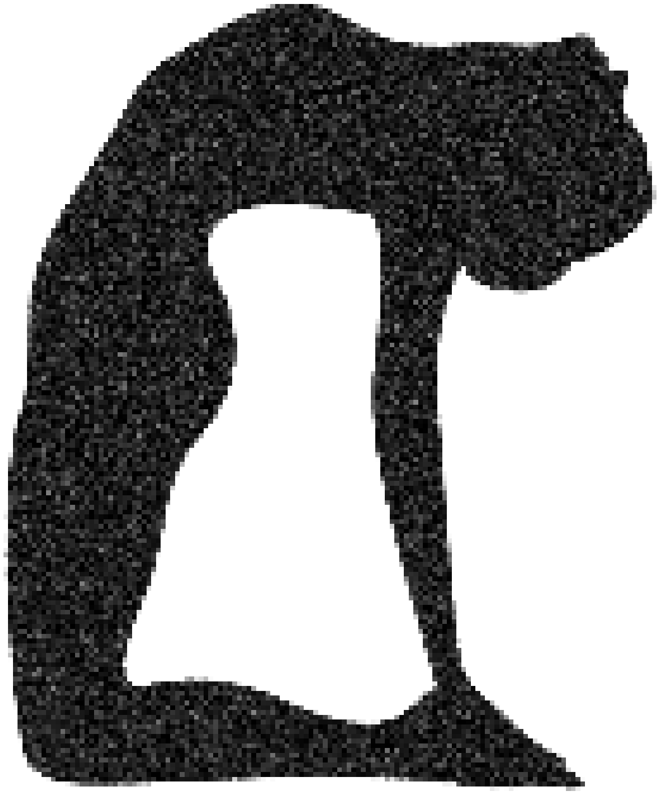


LL band of image

LH band of image

HL band of image

HH band of image



# Classifying yoga-assanans. %%

```
clc
clear all

% Make a global hash table, mapping index to asan name. %
match = containers.Map;
match('1') = 'Ustrasana';
match('2') = 'Veerbhadrasan';
match('3') = 'Vrikhsasana';
match('4') = 'Trikonasana';

% Learn the boundary discriptor. %
boundary_des = containers.Map;
for i=1:4

    im = imread(strcat('./yogasan/', num2str(i),'.jpg'));
    im = 255*not(im);
    im = im2bw(im);
    b = bwboundaries(im, 'noholes');
    boundary_des(num2str(i)) = b(1);

end

% Using a random function pick and image and classify. %
for j=1:10

    random_i = randi([1,4]);
    im = imread(strcat('./yogasan/', num2str(random_i),'.jpg'));
    imshow(im)
    im = 255*not(im);
    im = im2bw(im);
    b = bwboundaries(im, 'noholes');
    for i=1:4
        if(isequal(boundary_des(num2str(i)),b(1)))
            val = strcat('Given asana is ',' : ' ,match(num2str(i)));
            disp(val)
        end
    end
end

        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Veerbhadrasan
        Given asana is :Ustrasana
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Veerbhadrasan
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Veerbhadrasan
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Veerbhadrasan
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Veerbhadrasan
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Vrikhsasana
        Given asana is :Ustrasana
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Vrikhsasana
        Warning: Image is too big to fit on screen; displaying at 67%
        Given asana is :Trikonasana
```

*Published with MATLAB® 7.14*