General Sir John Kotelawala Defense University

Electrical, Electronic & Telecommunication Engineering

ET3112 – Image Processing and Machine Vision

Practical II - Simulation for Histogram generation

Reg No - **D/ENG/23/0201-ET**

Name - **W.M.S.F Weerasinghe**

Submission date – 07/01/2025

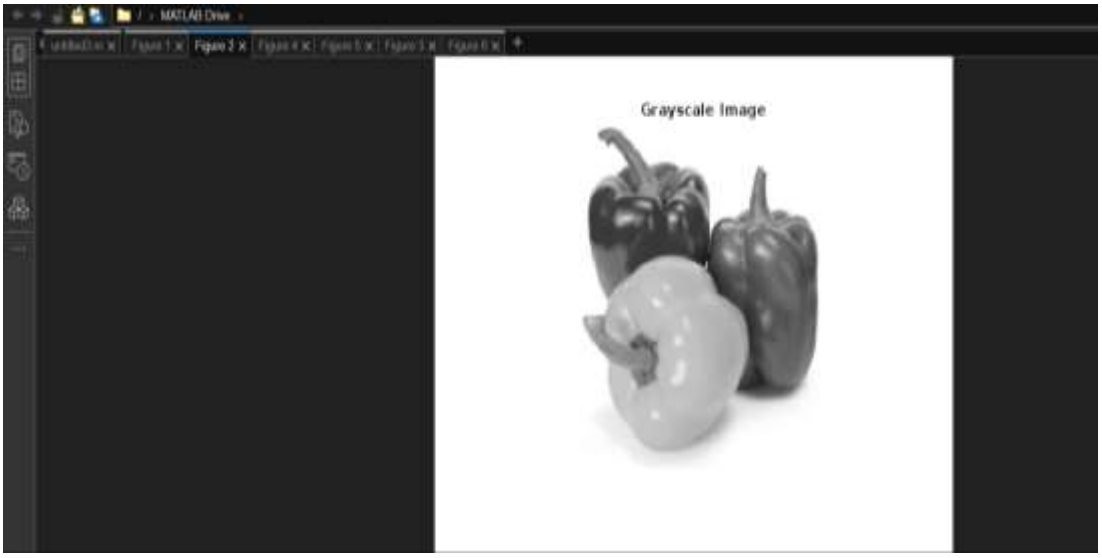
Q1

```
% Clear workspace and load the image
clc;
clear all;
close all;

% Step 1: Load and display the original image
bellpaper = imread('/MATLAB Drive/bellpaper.jpg');
figure;
imshow(bellpaper);
title('Original Image');

% Step 2: Convert the image to grayscale
gray_bellpaper = rgb2gray(bellpaper);
figure;
imshow(gray_bellpaper);
title('Grayscale Image');

% Step 3: Plot the histogram of the grayscale image
figure;
imhist(gray_bellpaper);
title('Histogram of Grayscale Image');
xlabel('Pixel Intensity');
ylabel('Number of Pixels');

% Step 4: Segment the image using a global threshold
threshold = graythresh(gray_bellpaper); % Calculate Otsu's threshold
bw_bellpaper = imbinarize(gray_bellpaper, threshold); % Create binary image
figure;
imshow(bw_bellpaper);
title('Segmented Image (Global Threshold)');

% Step 5: Save the segmented image
imwrite(bw_bellpaper, '/MATLAB Drive/segmented_bellpaper.jpg');

% Step 6: Custom segmentation based on histogram peak
% Find the peak value in the histogram
histo = imhist(gray_bellpaper);
[~, peak] = max(histo);

% Create binary image using the histogram peak as threshold
custom_bw = gray_bellpaper >= peak; % Logical operation for thresholding
figure;
imshow(custom_bw);
title('Segmented Image ');

% Step 7: Subplot all results
figure;
subplot(2, 2, 1);
imshow(bellpaper);
title('Original Image');

subplot(2, 2, 2);
imshow(gray_bellpaper);
```

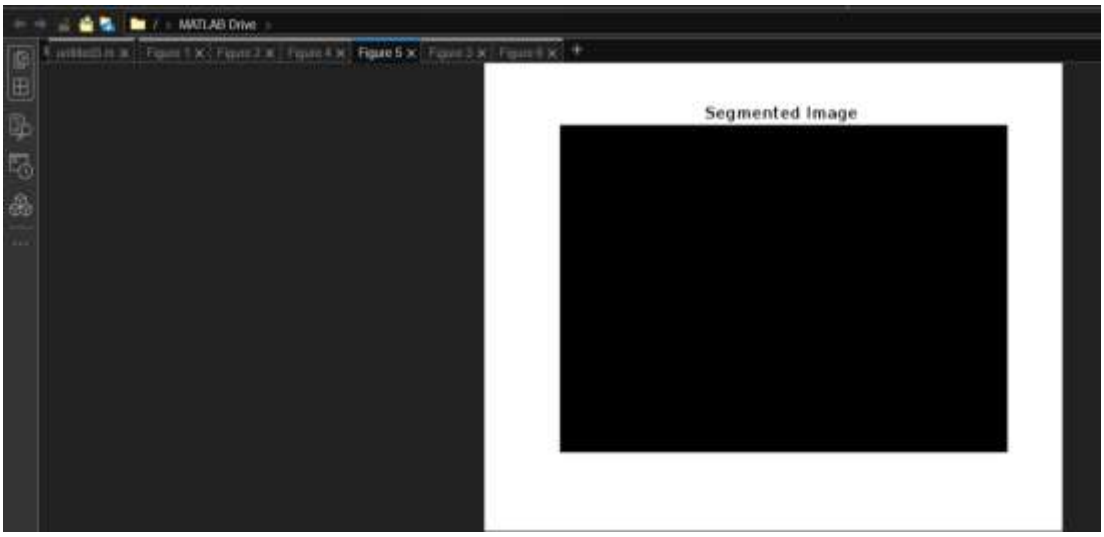
```
title('Grayscale Image');

subplot(2, 2, 3);
imshow(bw_bellpaper);
title('Segmented Image ');

subplot(2, 2, 4);
imshow(custom_bw);
title('Segmented Image (Custom Threshold)');
```

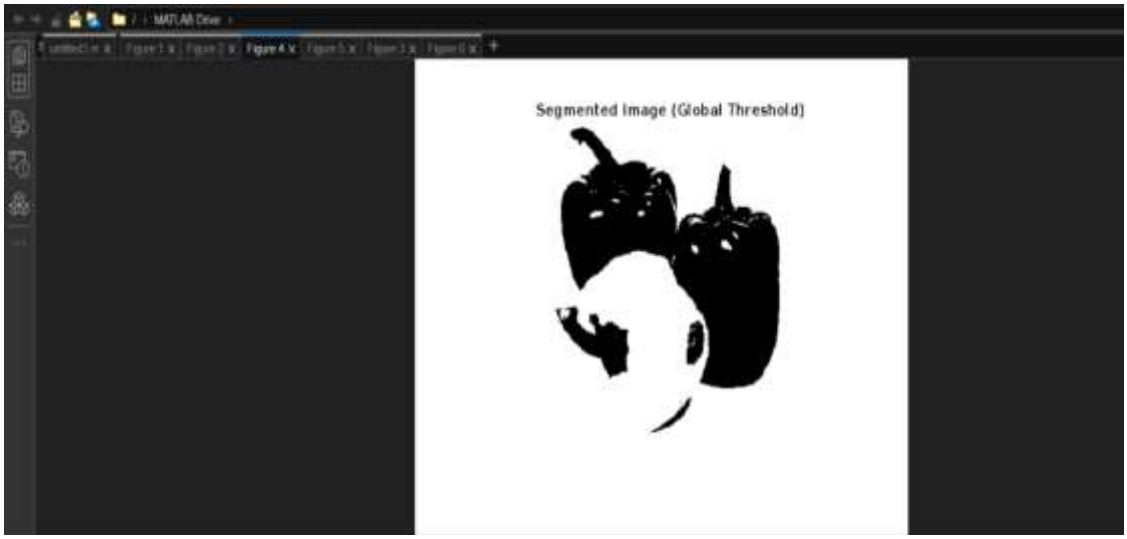
Original Image

Grayscale Image

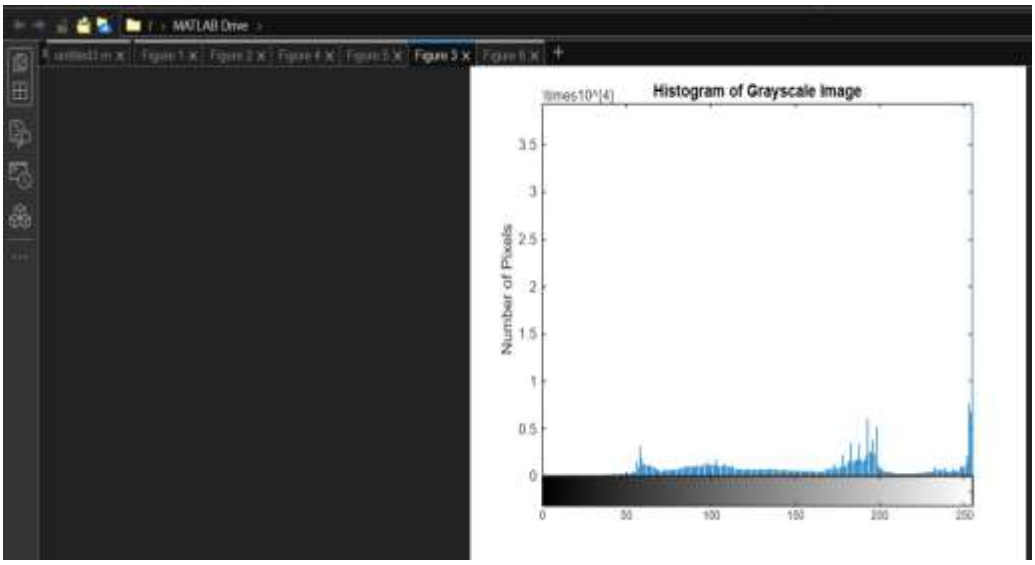


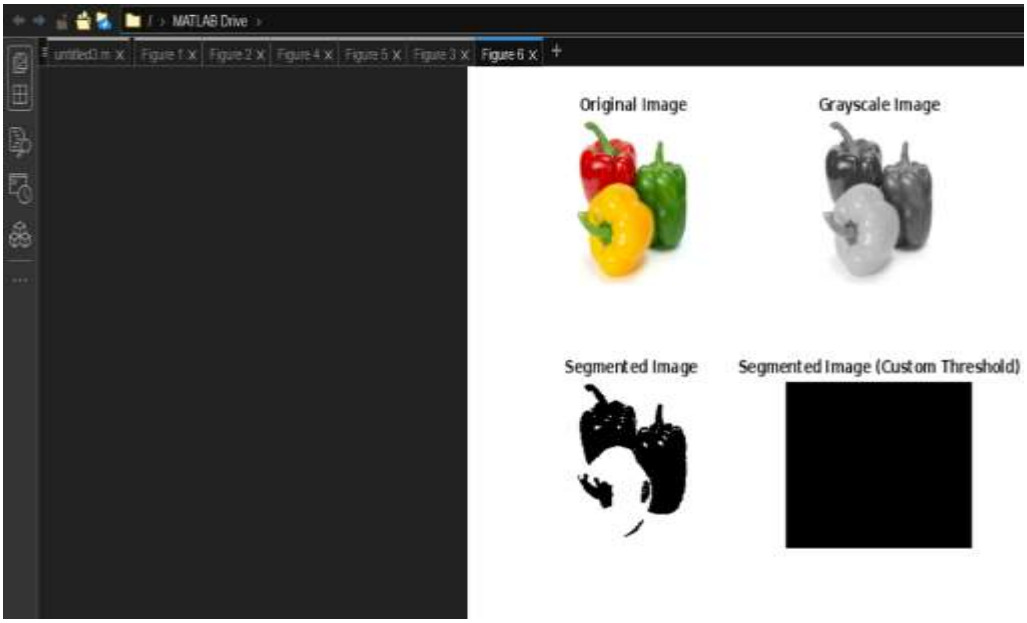
Segment


MATLAB Drive
Figure 5
Segmented Image


MATLAB Drive
Figure 4
Segmented Image (Global Threshold)

Histogram

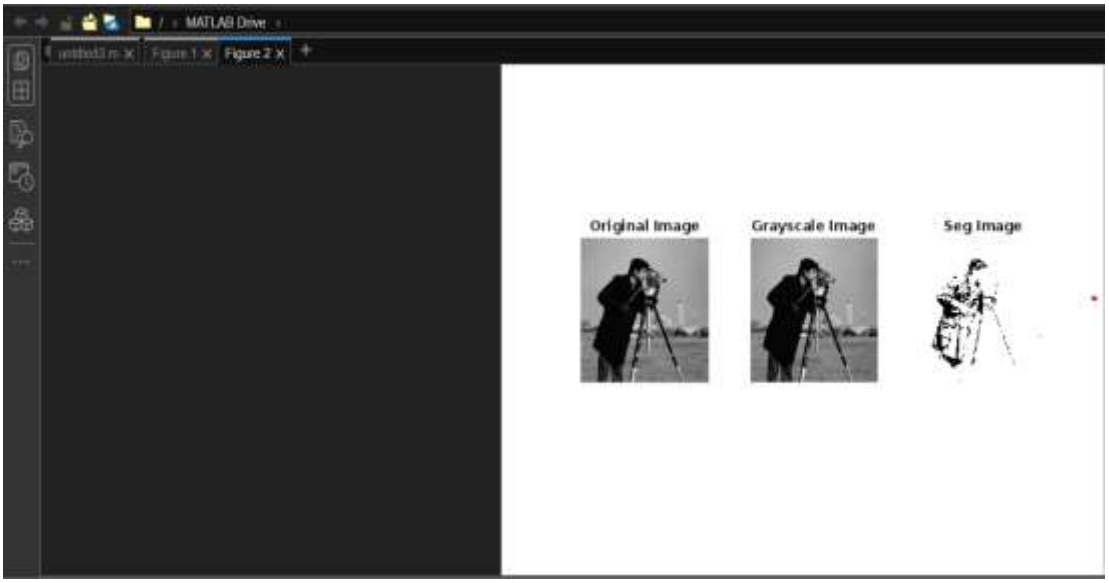

Histogram of Grayscale Image
Number of Pixels
0
0.5
1
1.5
2
2.5
3
3.5
0
50
100
150
200
250

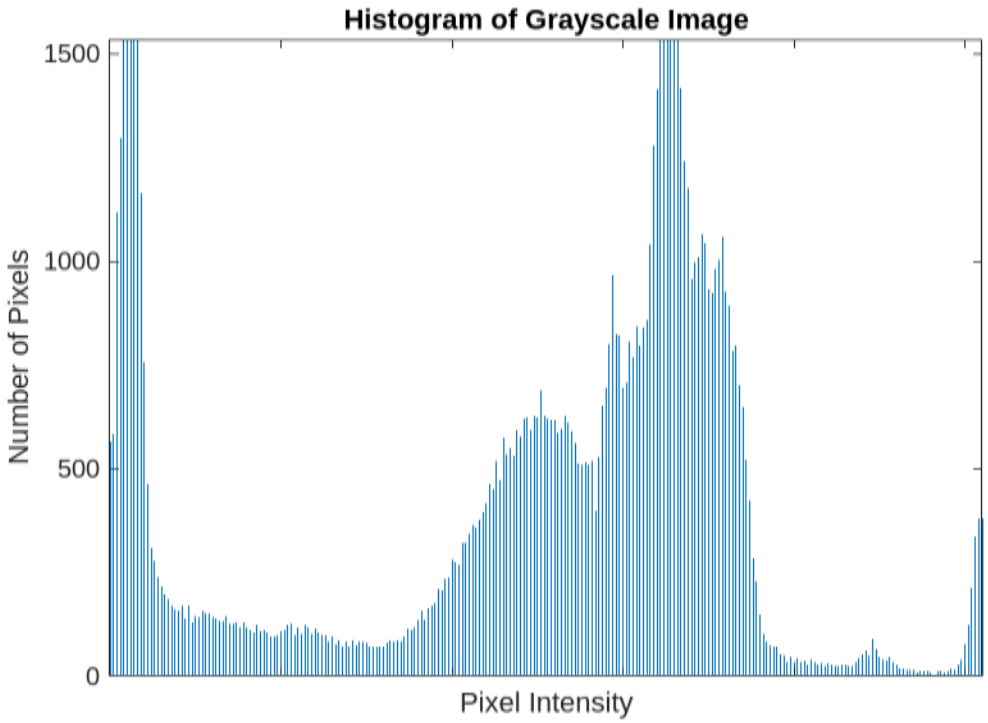

Original Image
Grayscale Image
Segmented Image
Segmented Image (Custom Threshold)

Q2

```
% Clear the workspace
clc;
clear all;
close all;

% Step 1: Load the color image
man_2 = imread('/MATLAB Drive/man.jpg');

% Step 2: Convert the image to grayscale
gray_man_2 = rgb2gray(man_2);

% Step 3: Plot the histogram of the grayscale image
figure;
imhist(gray_man_2);
title('Histogram of Grayscale Image');
xlabel('Pixel Intensity');
ylabel('Number of Pixels');

% Step 4: Get the peak of the histogram
histo = imhist(gray_man_2); % Histogram of grayscale image
[~, peak] = max(histo); % Get the peak intensity value

% Step 5: Segment the image using custom thresholding with loops
% Initialize a segmented image of the same size
[x, y] = size(gray_man_2);
segmented_image = zeros(x, y); % Create a black image of the same size

% Loop through each pixel and apply thresholding
for i = 1:x
    for j = 1:y
        if gray_man_2(i, j) >= peak
            segmented_image(i, j) = 1; % Assign '1' for pixels >= threshold
        else
            segmented_image(i, j) = 0; % Assign '0' otherwise
        end
    end
end

% Convert the segmented image to uint8 for display purposes
segmented_image = uint8(segmented_image * 255);

% Step 6: Sub-plot the original, grayscale, and segmented images
figure;
subplot(1, 3, 1);
imshow(man_2);
title('Original Image');

subplot(1, 3, 2);
imshow(gray_man_2);
title('Grayscale Image');

subplot(1, 3, 3);
imshow(segmented_image);
```

```
title('Seg Image');

% Step 7: Save the segmented image as a JPEG
imwrite(segmented_image, '/MATLAB Drive/segmented_man.jpg');
```

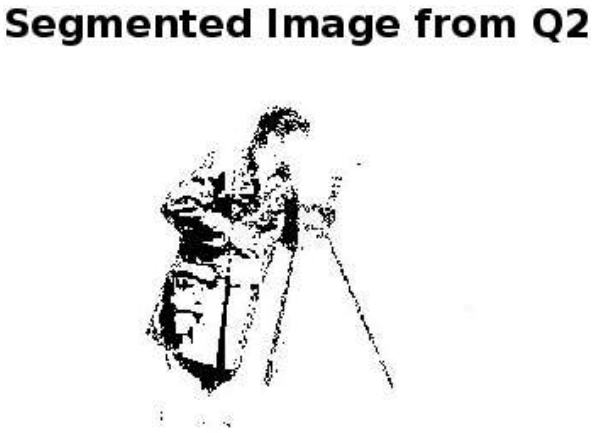



Histogram of Grayscale Image
1500
1000
500
0
Number of Pixels
Pixel Intensity

Q3

```
% Clear the workspace
clc;
clear all;
close all;

% Step 1: Load the color image
man_2 = imread('/MATLAB Drive/man.jpg');

% Step 2: Convert the image to grayscale
gray_man_2 = rgb2gray(man_2);

% Step 3: Plot the histogram of the grayscale image
figure;
imhist(gray_man_2);
title('Histogram of Grayscale Image');
xlabel('Pixel Intensity');
ylabel('Number of Pixels');

% Step 4: Define two threshold values
threshold_low = 100; % Lower threshold value
threshold_high = 150; % Higher threshold value

% Step 5: Segment the image using two thresholds
[x, y] = size(gray_man_2);
segmented_image_two_thresh = zeros(x, y); % Initialize binary image

% Apply segmentation using the two thresholds
for i = 1:x
   for j = 1:y
      if gray_man_2(i, j) >= threshold_low && gray_man_2(i, j) <= threshold_high
         segmented_image_two_thresh(i, j) = 1; % Assign '1' if pixel is within thresholds
      else
         segmented_image_two_thresh(i, j) = 0; % Assign '0' otherwise
      end
   end
end

% Convert the segmented image to uint8 for display purposes
segmented_image_two_thresh = uint8(segmented_image_two_thresh * 255);

% Step 6: Sub-plot all results for comparison
figure;
subplot(2, 2, 1);
imshow(man_2);
title('Original Image');

subplot(2, 2, 2);
imshow(gray_man_2);
title('Grayscale Image');

subplot(2, 2, 3);
imshow(segmented_image_two_thresh);
title('Segmented Image (Two Thresholds)');
```

```
subplot(2, 2, 4);
imshow('/MATLAB Drive/segmented_man.jpg'); % Load previously saved segmented image
title('Segmented Image from Q2');

% Step 7: Save the segmented image as a JPEG
imwrite(segmented_image_two_thresh, '/MATLAB Drive/segmented_man_two_thresholds.jpg');
```



Compare the segmented image with the segmented images in Q1 and Q2.

Using two thresholds for image segmentation, as done in Question 3, gives a more detailed result compared to the simpler methods in Questions 1 and 2. Instead of separating the image into just black and white like in the previous methods, this approach highlights the pixels that fall within a specific range of intensity values. This means it keeps more details and gives a clearer picture of certain parts of the image. In contrast, Question 1 used an automatic single threshold (im2bw), and Question 2 manually applied one threshold, both resulting in more basic, two-color images. The two-threshold method is helpful when you need to focus on certain intensity levels and preserve finer details in the image.