Computer Vision Laboratory

(EC6673)

Submitted by:
SHAILESH KUMAR
M.TECH (224EC6013)

Department of Electronics and Communications Engineering
National Institute of technology
Rourkela- 769008, INDIA

Sl. No.	Experiment	Page NO.	Date	Remark
1.	To get familiarized with MATLAB Image processing toolbox.			
2.	TO study various pixel-wise and transformation and histogram equalization.			
3.	To implement de-noising and de-blurring algorithms in spatial domain			
4.	To implement de-noising and de-blurring algorithms in frequency domain			
5.	Spatial and frequency domain filtering for colour images.			
6.	Image segmentation using global thresholding			
7.	k-means clustering based segmentation			
8.	Classifier design with principal component analysis			
9.	Classifier design with neural network			
10.	Mini Project			

Experiment no: 1

1. **Aim:** To get familiarized with MATLAB Image processing toolbox.

2. **Software Tool Used:**

MATLAB: is a powerful tool for engineers and scientists, excelling in matrix computations, data analysis, and visualization. It's a cornerstone in computer vision, offering tools for image processing, object detection, feature extraction, and more. With its rich toolbox and intuitive interface, MATLAB accelerates development of complex vision systems, making it indispensable for researchers and industry professionals alike.

Theory:

MATLAB Image Processing Toolbox: This toolbox provides a comprehensive set of reference-standard algorithms and functions for image processing, visualization, and analysis. It includes tools for image segmentation, geometric transformations, filtering, morphological operations, and more. In this project, we explored basic image processing techniques such as image reading, displaying, resizing, cropping, color space conversions, and binary thresholding.

The following functions were used in this project:

1. imread(): Reads an image from a file and stores it in an array.
   - Syntax: I = imread('filename')

2. imshow(): Displays an image in a MATLAB figure window.
   - Syntax: imshow(I)

3. imresize():Resizes an image to a specified size or scale.
   - Syntax: J = imresize(I, scale) or J = imresize(I, [rows cols])

4. imcrop():Crops a portion of the image by specifying the bounding rectangle.
   - Syntax: J = imcrop(I, rect)

5. rgb2gray():Converts an RGB image to a grayscale image.
   - Syntax: I_gray = rgb2gray(I)

6. im2bw():Converts a grayscale image to a binary image based on a specified threshold.
   - Syntax: BW = im2bw(I_gray, threshold)

7. imtool():Opens an interactive tool for image viewing and exploration, allowing for zooming, pixel value inspection, and more.
   - Syntax: imtool(I)

3. Result:

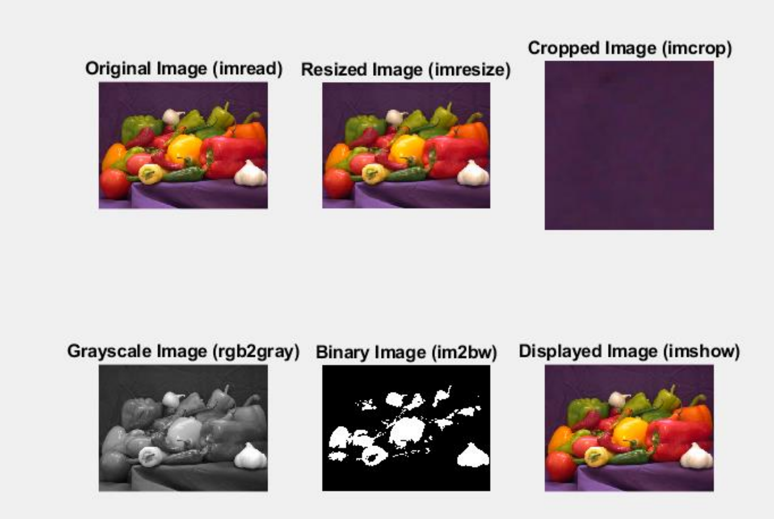

Figure 1: Step-by-step processing of the image ‘peppers.png’ through various operations in MATLAB.

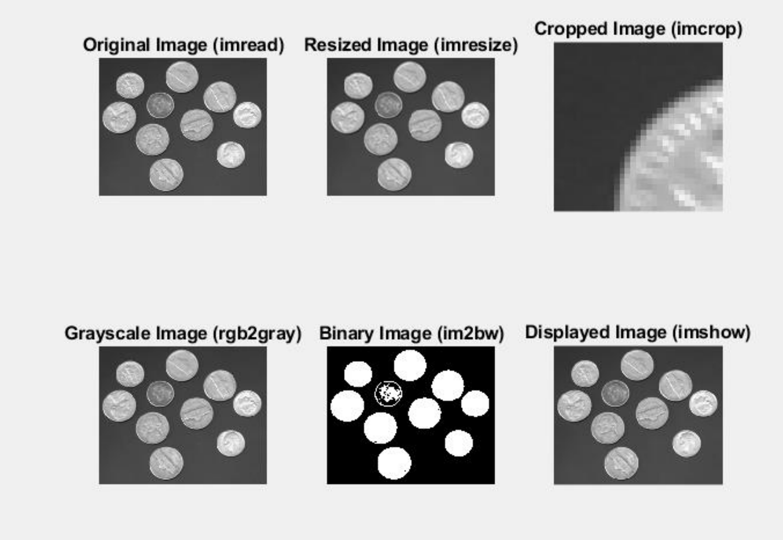

Figure 2: Step-by-step processing of the image ‘coins.png’ through various operations in MATLAB.

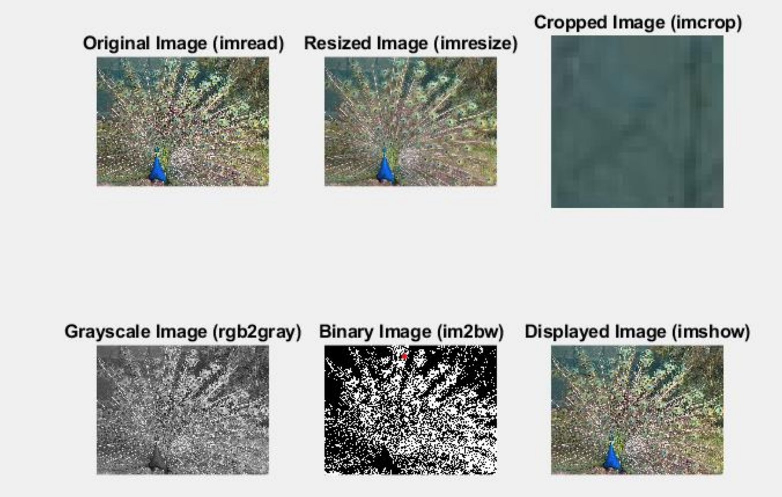

Figure 3: Step-by-step processing of the image ‘peacock.jpg’ through various operations in MATLAB.

4. Discussion:

In this project, we applied several basic image processing operations to a sample image:

1. Image Reading and Displaying:
   The image was read using imread() and displayed using imshow() to understand the structure of the image and its pixel values.
2. Resizing and Cropping:
   Using imresize(), the image was resized to a smaller scale, demonstrating the effect of reducing resolution. imcrop() was used to extract a region of interest (ROI) from the image.
3. Color Space Conversion:
   The image was converted from RGB to grayscale using rgb2gray() to reduce color information and emphasize intensity variation.
4. Binary Conversion:
   The grayscale image was further converted to binary using im2bw() to focus on foreground and background segmentation based on intensity thresholds.
5. Interactive Exploration:
   The imtool() function was used for detailed examination of the image, including pixel intensity values, zoom functionality, and image properties.

These steps helped me understand the fundamental image processing techniques in MATLAB. Each function we used had a specific role in transforming the image, whether by simplifying it, segmenting it, or allowing us to explore it interactively. This hands-on experience reinforced how these basic operations are the building blocks for more advanced image processing tasks.

5. Conclusion:

This project successfully introduced the basic functionalities of the MATLAB Image Processing Toolbox. By applying functions such as imread, imshow, imresize, imcrop, rgb2gray, im2bw, and imtool, we developed an understanding of how MATLAB can be used for basic image manipulation. This knowledge forms a foundation for more advanced image processing tasks and analysis.

6. References:

1. Gonzalez, R.C., & Woods, R.E. (2018). *Digital Image Processing* (4th ed.). Pearson.
2. Pratt, W.K. (2007). *Digital Image Processing: PIKS Scientific Inside* (4th ed.). Wiley-Interscience.

7. **Code:**

Image 1:

```
subplot(2,3,1);x = imread("peppers.png");  imshow(x);  title('Original Image (imread)');
subplot(2,3,2); y = imresize(x, 0.5);  imshow(y);   title('Resized Image (imresize)');

subplot(2,3,3);c = imcrop(x, [20, 20, 30, 30]);  imshow(c); title('Cropped Image (imcrop)');

subplot(2,3,4);l = rgb2gray(x);  imshow(l);  title('Grayscale Image (rgb2gray)');

subplot(2,3,5);m = im2bw(l);  imshow(m);   title('Binary Image (im2bw)');

subplot(2,3,6);imshow(x);   title('Displayed Image (imshow)');
imtool(x);
```

Image 2:

```
subplot(2,3,1);x = imread("coins.png");  imshow(x);  title('Original Image (imread)');

subplot(2,3,2);y = imresize(x, 0.5);  imshow(y);  title('Resized Image (imresize)');

subplot(2,3,3);c = imcrop(x, [20, 20, 30, 30]);  imshow(c);  title('Cropped Image (imcrop)');

subplot(2,3,4);l = im2gray(x);imshow(l);   title('Grayscale Image (rgb2gray)');

subplot(2,3,5);m = im2bw(l); imshow(m);   title('Binary Image (im2bw)');

subplot(2,3,6);imshow(x); title('Displayed Image (imshow)');
imtool(x);
```

image 3:

```
subplot(2,3,1);x = imread("peacock.jpg");  imshow(x);  title('Original Image (imread)');

subplot(2,3,2);y = imresize(x, 0.5);imshow(y);  title('Resized Image (imresize)');

subplot(2,3,3);c = imcrop(x, [20, 20, 30, 30]);  imshow(c);  title('Cropped Image (imcrop)');

subplot(2,3,4);l = rgb2gray(x);imshow(l);   title('Grayscale Image (rgb2gray)');

subplot(2,3,5);m = im2bw(l); imshow(m);    title('Binary Image (im2bw)');

subplot(2,3,6);imshow(x);   title('Displayed Image (imshow)');
imtool(x);
```