

DEPARTEMENT OF INFORMATION TECHNOLOGY

UNIVERSITY OF THE PUNJAB

(GUJRANWALA CAMPUS)



Assignment
Computer Vision

Prepared by:

Fatima

BIT21003

BSIT (Morning)

7th Semester

Submitted To:

Ma'am Fouqia Zafeer

Date: Jan 22,2025

EXERCISE 1:

Write a program which can read an image as an input and do the following automatically. Show the results of all steps.

1. Find the type of image: binary, gray or RGB.

```
% Load or use the input image stored in variable A
A = imread('wow.png'); % Replace with your image file name or path

% Display the input image
figure('Name', 'Original Image');
imshow(A);
title('Original Image');

% Step 1: Identify image type
if islogical(A)
    img_type = 'Binary';
elseif ndims(A) == 3
    img_type = 'RGB';
else
    img_type = 'Grayscale';
end

% Display the image type
disp(['Image type: ', img_type]);

% Convert RGB to grayscale for further processing
if strcmp(img_type, 'RGB')
    img_gray = rgb2gray(A);
elseif strcmp(img_type, 'Grayscale')
```

```
img_gray = A;  
else  
img_gray = double(A) * 255; % Convert binary to grayscale  
End
```

2. Find the issue in image, over dark, over bright, low contrast, or normal. (Hint: can use histogram).

```
% Step 2: Analyze histogram
```

```
figure('Name', 'Histogram');  
histogram(img_gray, 256);  
title('Image Histogram');
```

```
% Determine brightness and contrast issues
```

```
mean_intensity = mean(img_gray(:));  
contrast = max(img_gray(:)) - min(img_gray(:));
```

```
if mean_intensity < 50  
issue = 'Over Dark';  
elseif mean_intensity > 200  
issue = 'Over Bright';  
elseif contrast < 50  
issue = 'Low Contrast';  
else  
issue = 'Normal';  
end
```

```
% Display the identified issue
```

```
disp(['Image issue: ', issue]);
```

3. Resolve the issue if any and show the final image after enhancement.

```
% Step 3: Resolve issues
```

```
enhanced_img = img_gray; % Start with the grayscale image
```

```
switch issue
```

```
case 'Over Dark'
```

```
enhanced_img = imadjust(img_gray, stretchlim(img_gray), []);
```

```
case 'Over Bright'
```

```
enhanced_img = imadjust(img_gray, [0.2, 1], []);
```

```
case 'Low Contrast'
```

```
enhanced_img = histeq(img_gray);
```

```
end
```

```
% Display results
```

```
figure('Name', 'Enhanced Image');
```

```
imshow(enhanced_img);
```

```
title('Enhanced Image');
```

4. Test your program on following images

```
% Step 4: Compare original and enhanced images
```

```
if ~strcmp(issue, 'Normal')
```

```
disp(['Enhancement applied: ', issue]);
```

```
figure('Name', 'Comparison');
```

```
subplot(1, 2, 1);
```

```
imshow(img_gray);
```

```
title('Before Enhancement');
```

```
subplot(1, 2, 2);
```

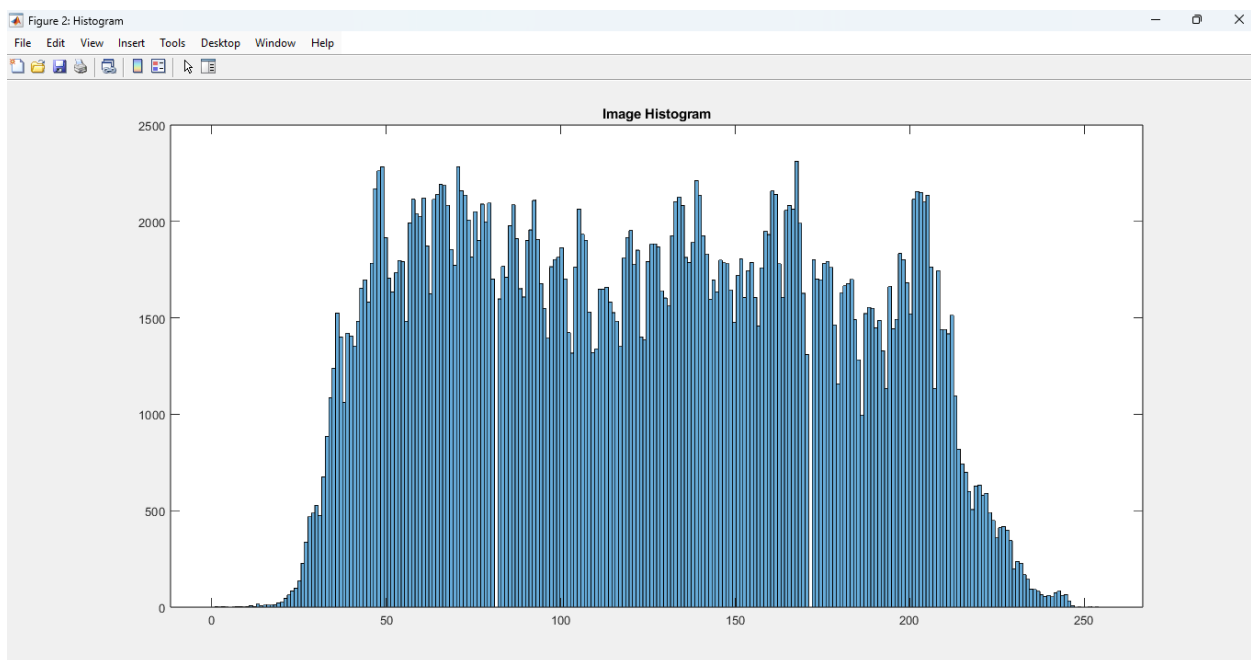
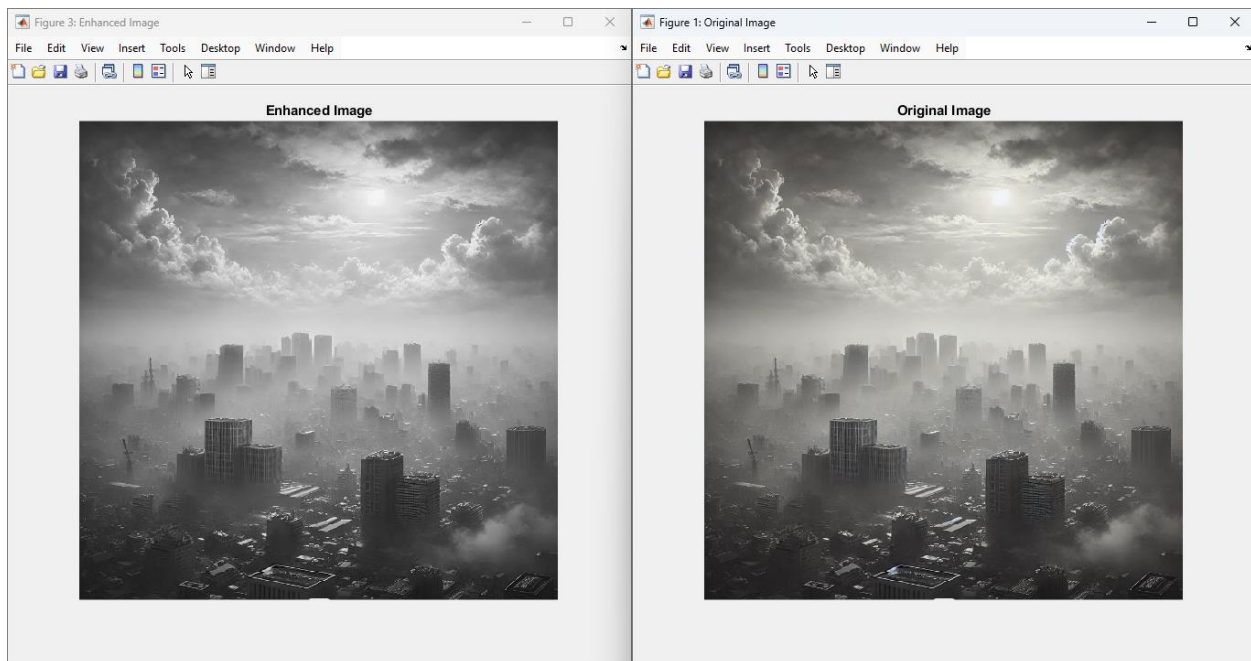
```
imshow(enhanced_img);
```

```
title('After Enhancement');
```

```
else
```

```
disp('No enhancement needed.');
```

```
end
```



```
Image type: RGB  
Image issue: Normal  
No enhancement needed.
```

Name ▲	Value	
A	574x574x3 uint8	
contrast	253	
enhanced_img	574x574 uint8	
img_gray	574x574 uint8	
img_type	'RGB'	
issue	'Normal'	
mean_intensity	123.9652	

EXERCISE 2:

Identify which intensity transformation was used on liftingbody.png to create each of the four results below. Write a script to reproduce the results using the intensity transformation functions.

CODE:

```
function intensityTransformations()
    % Read the original image
    originalImage = imread('house.jpeg');

    % Ensure the image is grayscale
    if ndims(originalImage) == 3
        originalImage = rgb2gray(originalImage);
    end

    % Apply transformations
    % 1. Darkened image (Result 1)
    result1 = imadjust(originalImage, [], [], 0.5); % Gamma correction with gamma < 1

    % 2. Brightened image (Result 2)
    result2 = imadjust(originalImage, [], [], 1.5); % Gamma correction with gamma > 1

    % 3. High contrast image (Result 3)
    result3 = histeq(originalImage); % Histogram equalization

    % 4. Low contrast image (Result 4)
    result4 = imadjust(originalImage, [0.3 0.7], [0.4 0.6]); % Adjust intensity range

    % Display results
    figure;
    subplot(2, 3, 1);
```

```

imshow(originalImage);
title('Original Image');

subplot(2, 3, 2);
imshow(result1);
title('Result 1: Darkened');

subplot(2, 3, 3);
imshow(result2);
title('Result 2: Brightened');

subplot(2, 3, 4);
imshow(result3);
title('Result 3: High Contrast');

subplot(2, 3, 5);
imshow(result4);
title('Result 4: Low Contrast');

```

End

