

Submit by:

Hina Batool(BIT21251)

Submit to:

Mam Fouqia Zaheer

#### 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.

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

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

4.Test your program on following images

Function main()

```
% Test images (Replace these with actual file paths)
```

```
testImages = {'flower.jpeg', 'house.jpeg', 'nature.jpeg'};
```

```
% Process each image
```

```
For I = 1:length(testImages)
```

```
    Fprintf('Processing: %s\n', testImages{i});
```

```
    processImage(testImages{i});
```

```
end
```

```
end
```

```
function processImage(filePath)
```

```
% Read the input image
```

```
Image = imread(filePath);
```

```
% Step 1: Detect the type of image
```

```
imageType = detectImageType(image);
```

```
fprintf('Image Type: %s\n', imageType);
```

```
% Step 2: Analyze the image for issues
```

```
Issue = analyzeImageIssues(image);
```

```
Fprintf('Detected Issue: %s\n', issue);
```

```
% Step 3: Resolve issues and enhance the image
```

```
enhancedImage = resolveIssues(image, issue);
```

```
% Display the original and enhanced images
```

```
Figure;
```

```
Subplot(1, 2, 1);
```

```
Imshow(image);
```

```
Title('Original Image');
```

```
Subplot(1, 2, 2);
```

```
Imshow(enhancedImage);
```

```
Title('Enhanced Image');
```

```
End
```

```
Function imageType = detectImageType(image)
```

```
% Detect whether the image is Binary, Grayscale, or RGB
```

```

If ndims(image) == 2
    uniqueValues = unique(image);
    if length(uniqueValues) == 2
        imageType = 'Binary';
    else
        imageType = 'Grayscale';
    end
elseif ndims(image) == 3
    imageType = 'RGB';
else
    imageType = 'Unknown';
end
end

```

```

function issue = analyzeImageIssues(image)

% Analyze the image for over dark, over bright, low contrast, or normal

If ndims(image) == 3
    grayImage = rgb2gray(image); % Convert RGB to Grayscale
else
    grayImage = image;
end

% Compute mean and standard deviation of pixel intensities
meanIntensity = mean(grayImage(:));
stdIntensity = std(double(grayImage(:)));

```

```

% Classify the issue based on thresholds

If meanIntensity < 50
    Issue = 'Over Dark';
Elseif meanIntensity > 200
    Issue = 'Over Bright';
Elseif stdIntensity < 40
    Issue = 'Low Contrast';
Else
    Issue = 'Normal';
End

End

End

Function enhancedImage = resolveIssues(image, issue)

% Resolve the detected issue in the image

Switch issue
    Case 'Over Dark'
        enhancedImage = imadjust(image, [], [], 1.2); % Brighten
    case 'Over Bright'
        enhancedImage = imadjust(image, [], [], 0.8); % Darken
    case 'Low Contrast'
        if ndims(image) == 3 % RGB Image
            labImage = rgb2lab(image);

            L = labImage(:, :, 1);

            L = histeq(L / 100) * 100;

            labImage(:, :, 1) = L;

            enhancedImage = lab2rgb(labImage);

```

```
    else % Grayscale Image
        enhancedImage = histeq(image);
    end
otherwise
    enhancedImage = image; % No enhancement needed
end
end
```