EAST WEST UNIVERSITY

Lab Report-1

Submitted By:
Name: Yougshree Saha Urmy
Id: 2021-1-60-127
Section: 01

Submitted To:
Dr. Md. Tauhid Bin Iqbal (DTBI)
Assistant Professor, Department of CSE
East West University

Date of submission: 02-03-2025

Here I am Using MATLAB code in this assignment.

1. Grayscale Conversion:

Convert a color image (RGB) to grayscale with any method.

Answer:

```
% Read the RGB image
img = imread('lab_1.jpg');
% Convert to grayscale
gray_img = rgb2gray(img);
% Display the results
figure;
subplot(1,2,1), imshow(img), title('Original RGB Image');
subplot(1,2,2), imshow(gray_img), title('Grayscale Image');
% Save the grayscale image
imwrite(gray_img, 'grayscale_image.jpg');
```

Output:

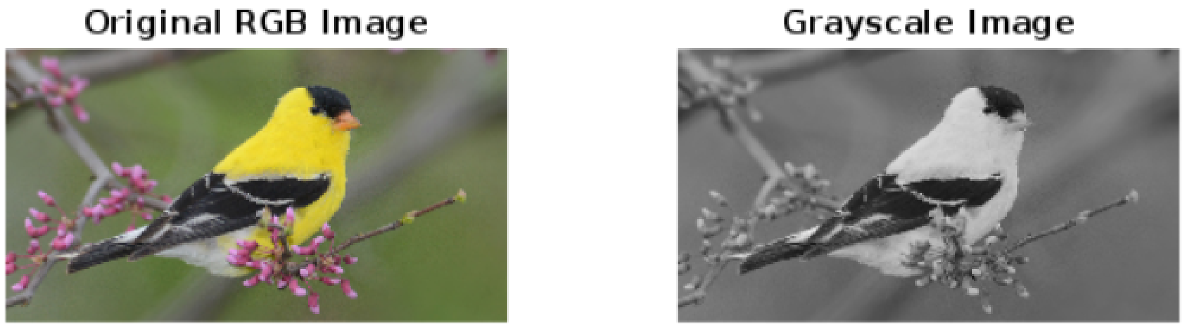


Figure: 1

2. Image Negative:

Generate the negative of an image by applying the transformation: new_pixel =255 - old_pixel.

Answer:

```
img = imread('lab_1.jpg');
% Convert to negative image
negative_img = 255-img;
% Display results:
figure;
```

```
subplot(1,2,1), imshow(img), xlabel('Original Image', 'FontSize', 12,
'FontWeight', 'bold');
subplot(1,2,2),imshow(negative_img),xlabel('Negative Image', 'FontSize', 12,
'FontWeight', 'bold');
% Save the negative image
imwrite(negative_img, 'negative_image.jpg');
```

Output:

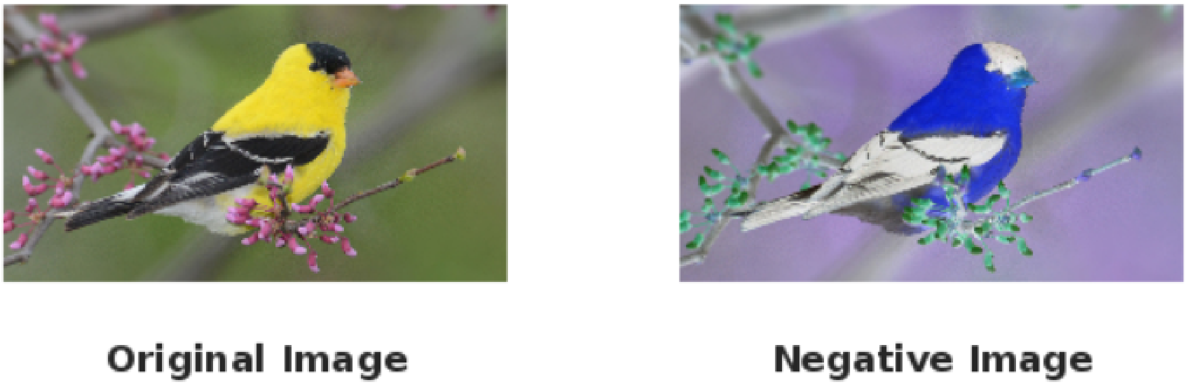


Figure: 2

3. Brightness Adjustment:

Modify the brightness of an image by adding or subtracting a constant from each pixel value.

Answer:

```
img = imread('lab_1.jpg');
brightness_factor= 60; %
% Increase brightness
brighter_img= img+ brightness_factor;
brighter_img(brighter_img> 255)= 255;
% Decrease brightness
darker_img= img- brightness_factor;
darker_img(darker_img< 0)= 0;
% Display results side by side
figure;
subplot(1,3,1),imshow(img), xlabel('Original Image', 'FontSize', 12,
'FontWeight', 'bold');
subplot(1,3,2), imshow(brighter_img),xlabel('Brighter Image', 'FontSize', 12,
'FontWeight', 'bold');
subplot(1,3,3),imshow(darker_img),xlabel('Darker Image', 'FontSize', 12,
'FontWeight', 'bold');
% Save the images
imwrite(brighter_img, 'brighter_image.jpg');
imwrite(darker_img, 'darker_image.jpg');
```

Output:

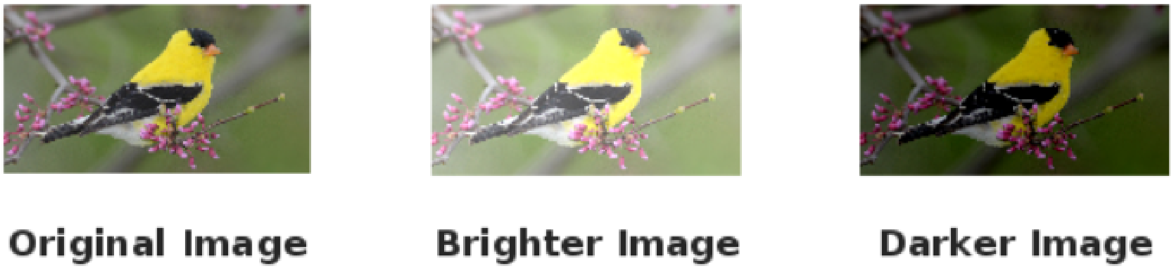


Figure: 3

4. Black-and-White Conversion (Thresholding):

Convert an image to black and white by applying a threshold (e.g., if pixel>128→ white, else black).

Answer:

```
img = imread('lab_1.jpg');
gray_img = rgb2gray(img);

% Set threshold value
threshold = 128;

%Convert black and white by applying a threshold.If pixel>128 → white, else
black).
bw_img= gray_img> threshold; %logical matrix(binary image)
% Display results:

figure;
subplot(1,3,1),imshow(img),xlabel('Original Image', 'FontSize', 12,
'FontWeight', 'bold');
subplot(1,3,2),imshow(gray_img),xlabel('Grayscale Image', 'FontSize', 12,
'FontWeight', 'bold');
subplot(1,3,3),imshow(bw_img),xlabel('Black & White Image', 'FontSize', 12,
'FontWeight', 'bold');
% Save the binary image
imwrite(bw_img,'bw_image.jpg');
```

Output:

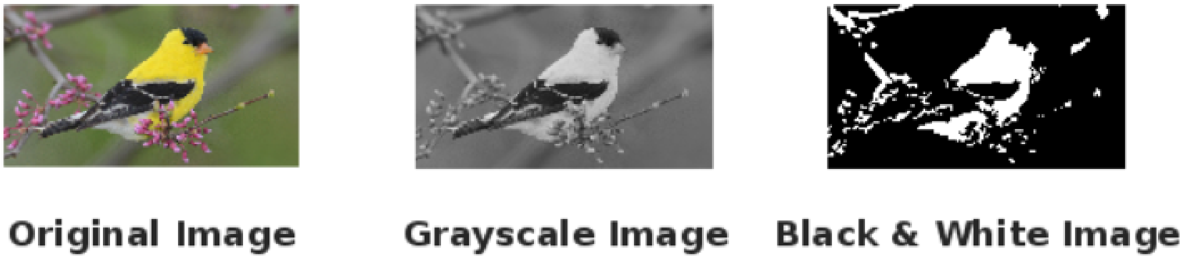


Figure: 4

5. Image Resizing:

Implement image scaling using nearest neighbor interpolation or any other suitable method.

Answer: Here I am using nearest neighbor, bilinear interpolation and bicubic interpolation.

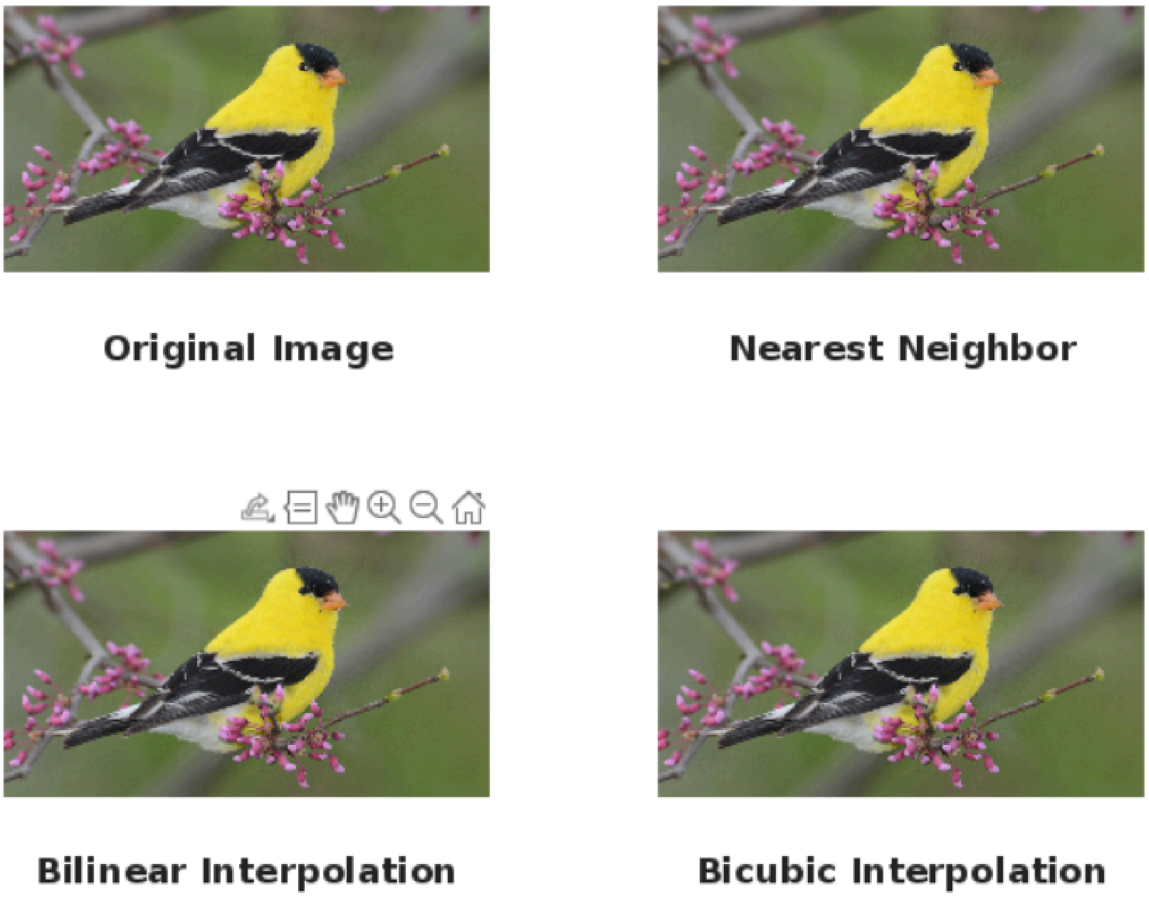
```
img= imread('lab_1.jpg');
% Scaling factor (reduce to half)
scale_factor= 0.5;

% Resize the image using nearest neighbor interpolation
r_n= imresize(img, scale_factor, 'nearest');
% Resize the image using bilinear interpolation
r_b= imresize(img, scale_factor, 'bilinear');
% Resize the image using bicubic interpolation
r_bi= imresize(img, scale_factor, 'bicubic');

% Display results:
figure;
subplot(2,2,1),imshow(img),xlabel('Original Image', 'FontSize', 12,
'FontWeight', 'bold');
subplot(2,2,2),imshow(r_n),xlabel('Nearest Neighbor', 'FontSize', 12,
'FontWeight', 'bold');
subplot(2,2,3),imshow(r_b),xlabel('Bilinear Interpolation', 'FontSize', 12,
'FontWeight', 'bold');
subplot(2,2,4),imshow(r_bi),xlabel('Bicubic Interpolation', 'FontSize', 12,
'FontWeight', 'bold');

% Save the resized images
imwrite(r_n,'resized_nn.jpg');
imwrite(r_b,'resized_bilinear.jpg');
imwrite(r_bi,'resized_bicubic.jpg');
```

Output:



Original Image

Nearest Neighbor

Bilinear Interpolation

Bicubic Interpolation

Figure: 5

THE END