

Experiment 1:

Introduction to MATLAB Commands and Functions in Digital Image Processing

MATLAB is a high-level programming environment commonly used for numerical computing and image processing. Here's a brief overview of some essential commands and functions used in MATLAB for image processing:

Reading an Image

- **Command:** `imread`
- **Description:** Reads an image file and stores it in a variable.
- **Example:**

```
img = imread('abcd.jpg');
```

Displaying an Image

- **Command:** `imshow`
- **Description:** Displays an image in a new figure window.
- **Example:**

```
imshow(img);
```

Converting to Grayscale

- **Command:** `rgb2gray`
- **Description:** Converts a color image to a grayscale image.
- **Example:**

```
grayImg = rgb2gray(img);
```

Creating Subplots

- **Command:** subplot
- **Description:** Divides the figure window into a grid and allows you to place multiple plots in different sections.
- **Example:**

```
subplot(1, 2, 1); % Creates a subplot in a 1x2 grid, first position  
imshow(img);  
title('Original Image');
```

```
subplot(1, 2, 2); % Creates a subplot in a 1x2 grid, second position  
imshow(grayImg);  
title('Grayscale Image');
```

Experiment 2:

Writing Programs to Read and Display Images

Objective

Write MATLAB programs to read and display images.

MATLAB Commands

1. Reading an Image

```
img = imread('filename.jpg');
```

- imread: Reads an image from a file.

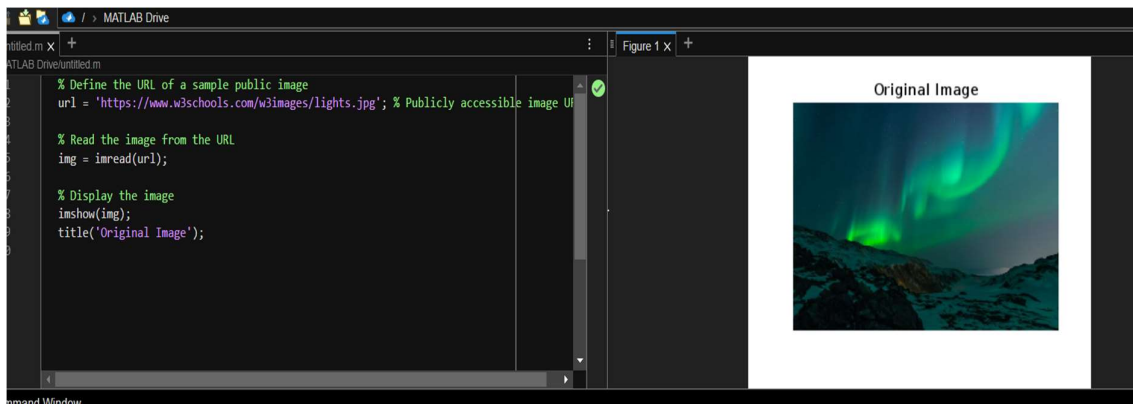
2. Displaying an Image

```
imshow(img);
```

- imshow: Displays the image in a window.

Example Program

```
img = imread('example.jpg');  
imshow(img);  
title('Original Image');
```



Experiment 3:

Writing Programs to Convert to Grayscale and Display Using Subplot

MATLAB Commands

1. Convert to Grayscale

matlab

Copy code

```
grayImg = rgb2gray(img);
```

- `rgb2gray`: Converts a color image to grayscale.

2. Create Subplots

matlab

Copy code

```
subplot(m, n, p);
```

- `subplot`: Divides the figure into a grid of subplots.

3. Displaying Images in Subplots

matlab

Copy code

```
subplot(1, 2, 1);  
imshow(img);  
title('Original Image');
```

```
subplot(1, 2, 2);  
imshow(grayImg);  
title('Grayscale Image');
```

Example Program

matlab

Copy code

```
url =  
'https://www.publicdomainpictures.net/pictures/320000/vel  
ka/background-image.png';  
img = imread(url);  
grayImg = rgb2gray(img);  
  
subplot(1, 2, 1);
```

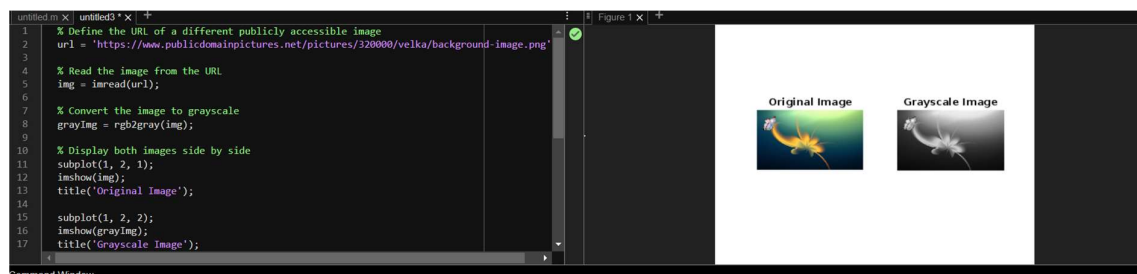
```
imshow(img);  
title('Original Image');  
  
subplot(1, 2, 2);  
imshow(grayImg);  
title('Grayscale Image');
```

Conclusion

In this experiment, you have learned to:

- Use MATLAB commands to read and display images.
- Convert images to grayscale.
- Utilize the subplot function to display multiple images in a single figure.

These basic skills form the foundation for more advanced image processing tasks in MATLAB.



Experiment 4:

To write a program for histogram calculation and equilization
MATLAB code to calculate the histogram and perform
histogram equalization on a grayscale image.

matlab

Copy code

image =

imread('https://wallpapercave.com/wp/wp3028175.jpg');

if size(image, 3) == 3

image = rgb2gray(image);

end

figure;

imshow(image);

title('Original Image');

hist_orig = imhist(image);

figure;

bar(hist_orig);

title('Histogram of Original Image');

xlabel('Pixel Intensity');

ylabel('Frequency');

equalized_image = histeq(image);

figure;

imshow(equalized_image);

```
title('Equalized Image');
```

```
hist_eq = imhist(equalized_image);
```

```
figure;
```

```
bar(hist_eq);
```

```
title('Histogram of Equalized Image');
```

```
xlabel('Pixel Intensity');
```

```
ylabel('Frequency');
```

```
figure;
```

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

```
imshow(image);
```

```
title('Original Image');
```

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

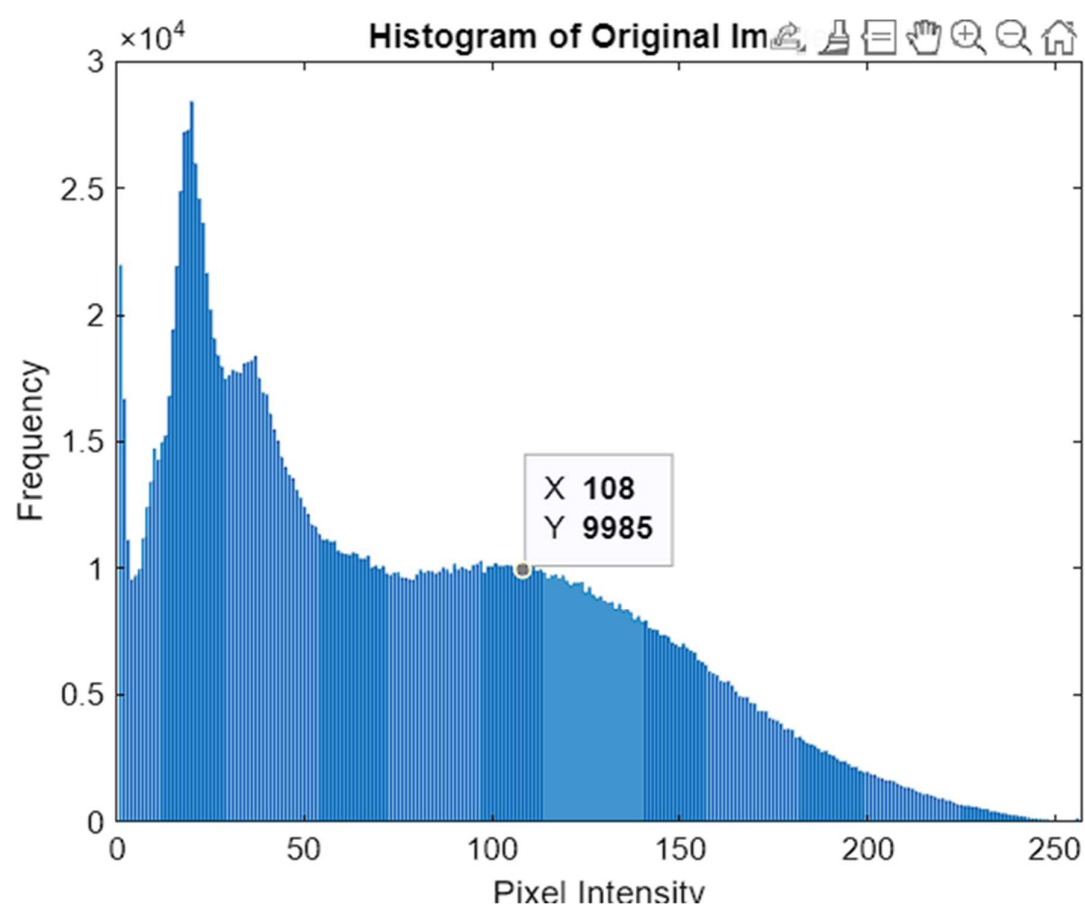
```
imshow(equalized_image);
```

```
title('Equalized Image');
```



Equalized Image



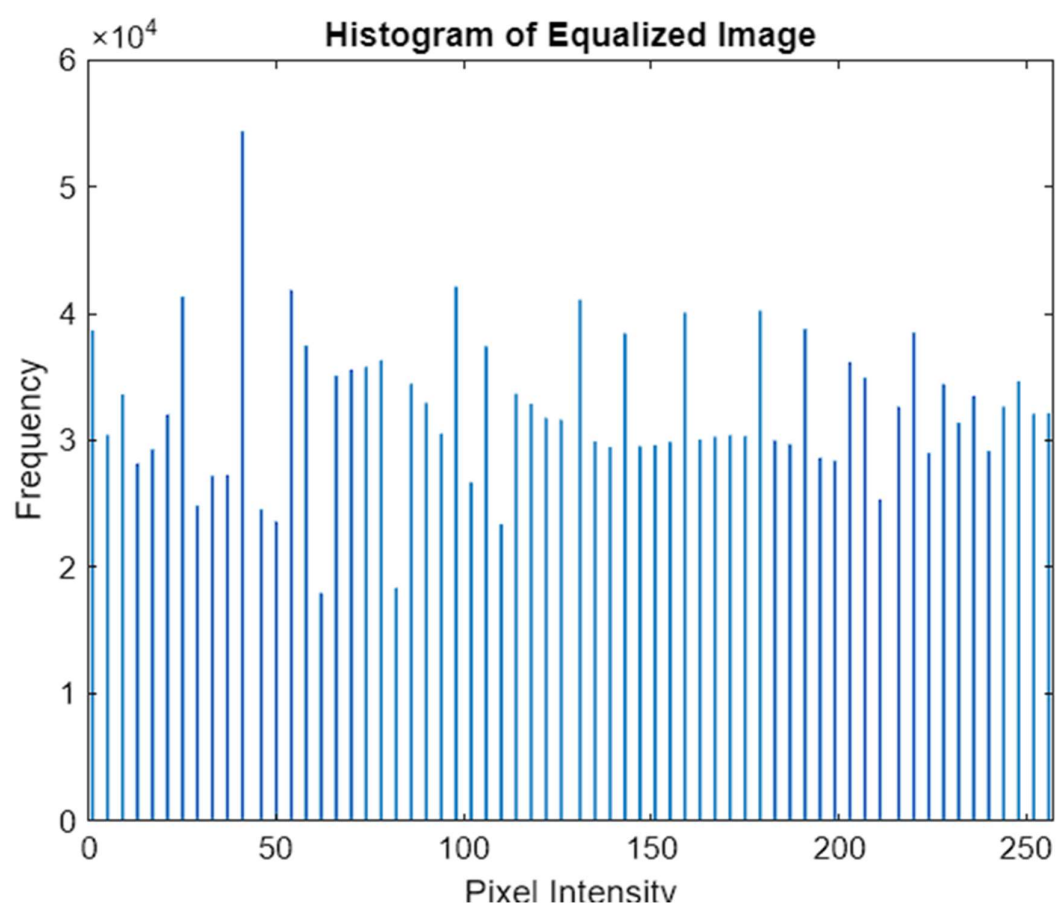


Original Image



Equalized Image





Experiment 5

To write and execute programs for image arithmetic operations addition of 2 images, subtract one image from other , multiply two images, divide two images , image blending , mean value of image:

```
% Read images
image1 =
imread('https://tse1.mm.bing.net/th?id=OIP.Q1xjNu2aibIEeO
SQ20HuQgHaEy&pid=Api&P=0&h=220');
image2 =
imread('https://tse3.mm.bing.net/th?id=OIP.7LNXRUIblBnFCJ
_LQni8lwHaFj&pid=Api&P=0&h=220');

% Resize images to the same size if necessary
image2 = imresize(image2, [size(image1, 1) size(image1, 2)]);

% Perform image addition
image_add = imadd(image1, image2);

% Display the result
figure, imshow(image_add);
title('Image Addition');
% Perform image subtraction
image_subtract = imsubtract(image1, image2);
```

```
% Display the result
figure, imshow(image_subtract);
title('Image Subtraction');
% Perform image multiplication
image_multiply = immultiply(image1, image2);

% Display the result
figure, imshow(image_multiply);
title('Image Multiplication');
% Convert images to double for division
image1_double = im2double(image1);
image2_double = im2double(image2);

% Perform image division
image_divide = imdivide(image1_double, image2_double);

% Display the result
figure, imshow(image_divide);
title('Image Division');
% Alpha blending coefficient
alpha = 0.5;

% Perform image blending
image_blend = alpha * image1 + (1 - alpha) * image2;

% Display the result
figure, imshow(image_blend);
title('Image Blending');
% Convert the image to grayscale
```

```
gray_image = rgb2gray(image1);
```

```
% Calculate the mean value of the grayscale image
```

```
mean_value = mean(gray_image(:));
```

```
% Display the result
```

```
fprintf('Mean value of the image: %.2f\n', mean_value);
```

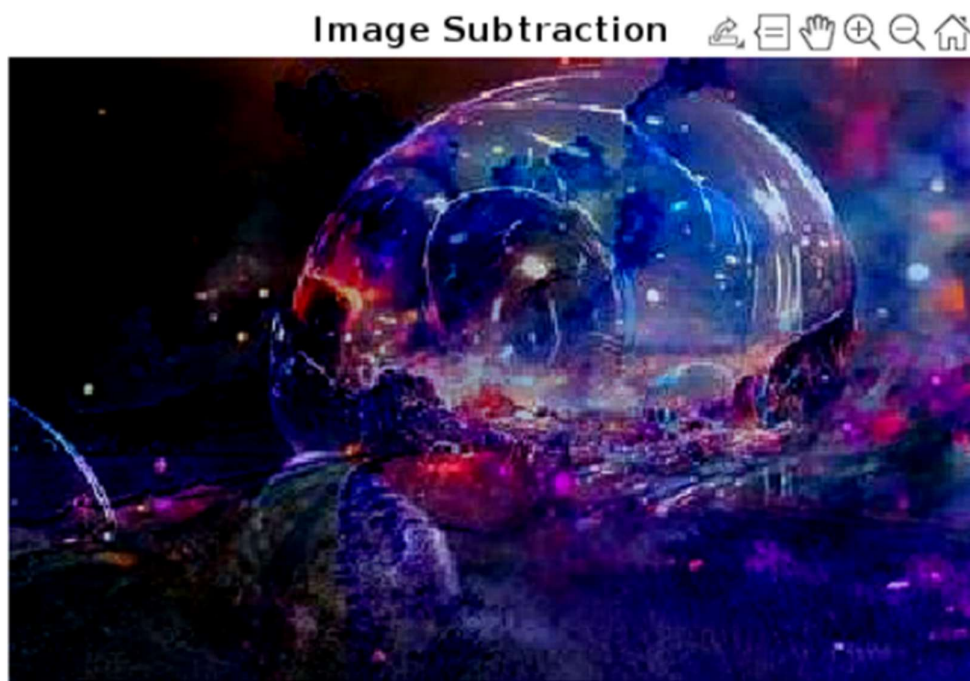


Image Addition



Image Multiplication

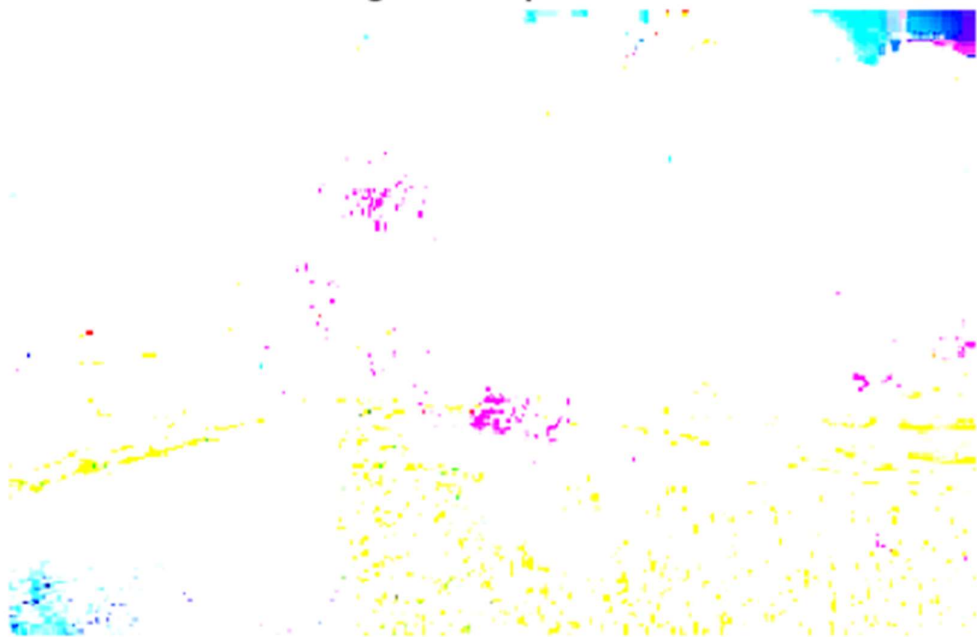
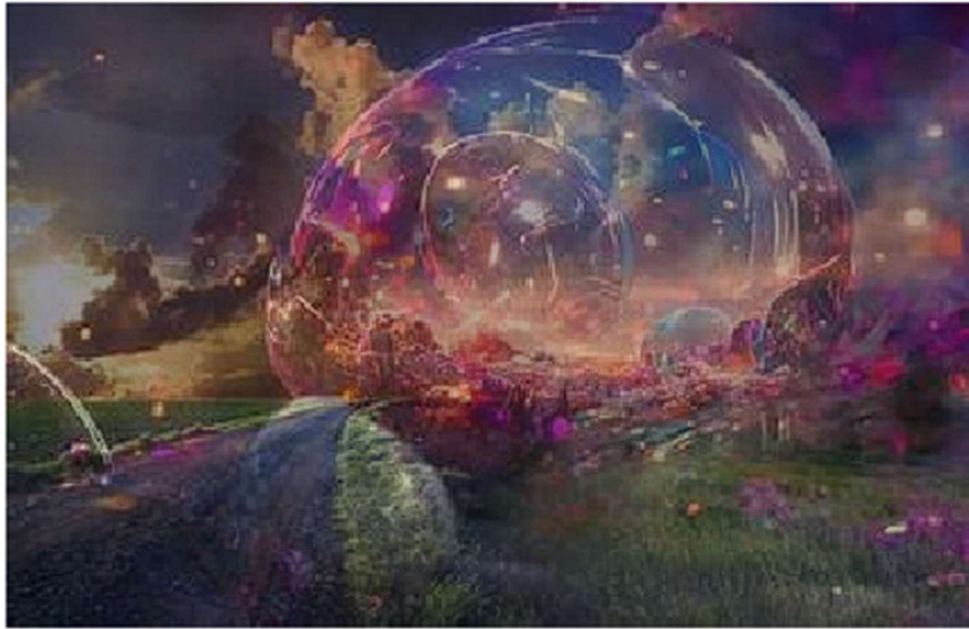


Image Division



Image Blending



Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> untitled  
Mean value of the image: 81.20  
>>
```