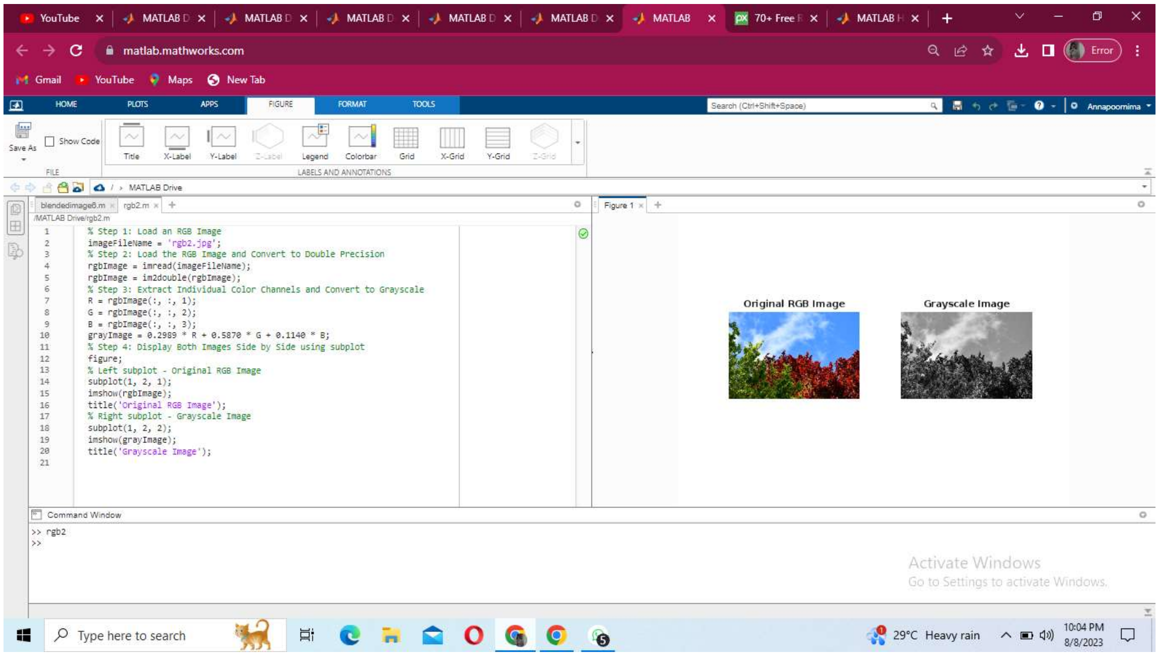
NAME : ANNAPOORNIMA S

225229101

EXERCISE 1: RGB to Grayscale Conversion

Write a MATLAB function that converts an RGB image to grayscale using the formula gray_value = 0.2989* R+0.5870 * G+0.1140 B. Display the original RGB image and the grayscale image side by side.

```
%code :
imageFileName = 'rgb2.jpg';
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
R = rgbImage(:, :, 1);
G = rgbImage(:, :, 2);
B = rgbImage(:, :, 3);
grayImage = 0.2989 * R + 0.5870 * G + 0.1140 * B;
figure;
subplot(1, 2, 1);
imshow(rgbImage);
title('Original RGB Image');
subplot(1, 2, 2);
imshow(grayImage);
title('Grayscale Image');
```



```
% Step 1: Load an RGB Image
imageFileName = 'rgb2.jpg';
% Step 2: Load the RGB Image and Convert to Double Precision
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
% Step 3: Extract Individual Color Channels and Convert to Grayscale
R = rgbImage(:, :, 1);
G = rgbImage(:, :, 2);
B = rgbImage(:, :, 3);
grayImage = 0.2989 * R + 0.5870 * G + 0.1140 * B;
% Step 4: Display Both Images Side by Side using subplot
figure;
% Left subplot - Original RGB Image
subplot(1, 2, 1);
imshow(rgbImage);
title('Original RGB Image');
% Right subplot - Grayscale Image
subplot(1, 2, 2);
imshow(grayImage);
title('Grayscale Image');
```

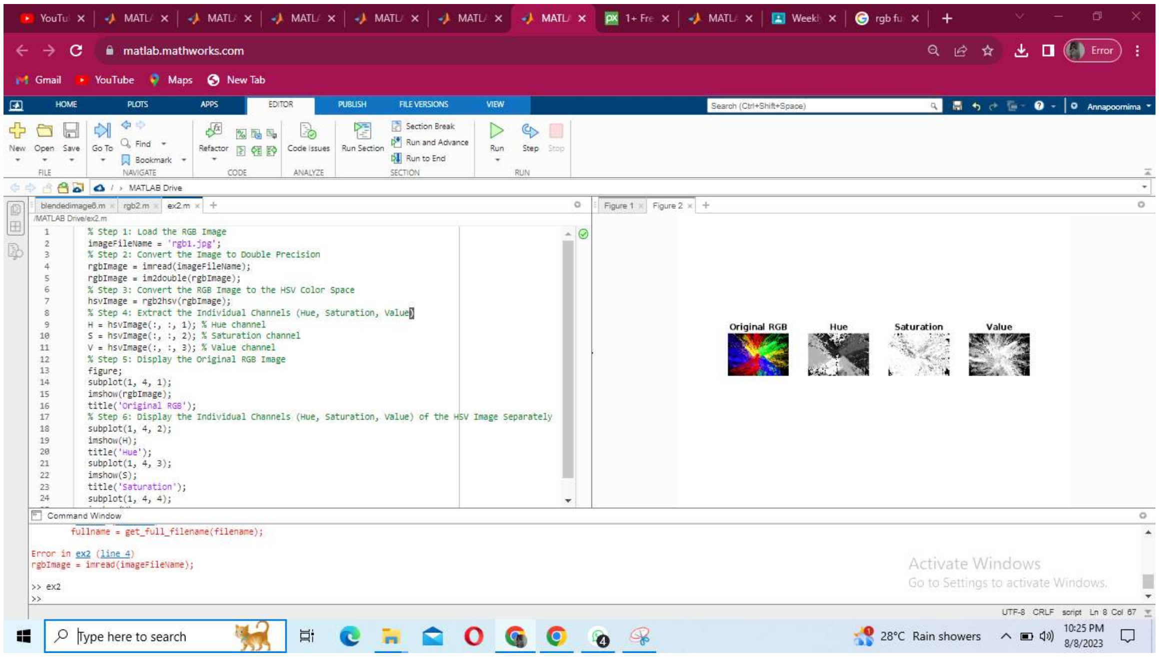
```
>> rgb2
>>
```

EXERCISE 2: RGB to HSV Conversion

Write a MATLAB function that converts an RGB image to the HSV color space using the rgb2hsv function. Display the individual channels (Hue, Saturation, and Value) of the HSV image separately.

```
%Code :
imageFileName = 'rgb1.jpg';
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
hsvImage = rgb2hsv(rgbImage);
H = hsvImage(:, :, 1); % Hue channel
S = hsvImage(:, :, 2); % Saturation channel
V = hsvImage(:, :, 3); % Value channel
figure;
subplot(1, 4, 1);
imshow(rgbImage);
title('Original RGB');
subplot(1, 4, 2);
imshow(H);
title('Hue');
subplot(1, 4, 3);
imshow(S);
```

```
title('Saturation');
subplot(1, 4, 4);
imshow(V);
title('Value');
```



```
% Step 1: Load the RGB Image
imageFileName = 'rgb1.jpg';
% Step 2: Convert the Image to Double Precision
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
% Step 3: Convert the RGB Image to the HSV Color Space
hsvImage = rgb2hsv(rgbImage);
% Step 4: Extract the Individual Channels (Hue, Saturation, Value)
H = hsvImage(:, :, 1); % Hue channel
S = hsvImage(:, :, 2); % Saturation channel
V = hsvImage(:, :, 3); % Value channel
% Step 5: Display the Original RGB Image
figure;
subplot(1, 4, 1);
imshow(rgbImage);
title('Original RGB');
% Step 6: Display the Individual Channels (Hue, Saturation, Value) of the HSV Image Separately
subplot(1, 4, 2);
imshow(H);
title('Hue');
subplot(1, 4, 3);
imshow(S);
title('Saturation');
subplot(1, 4, 4);
```

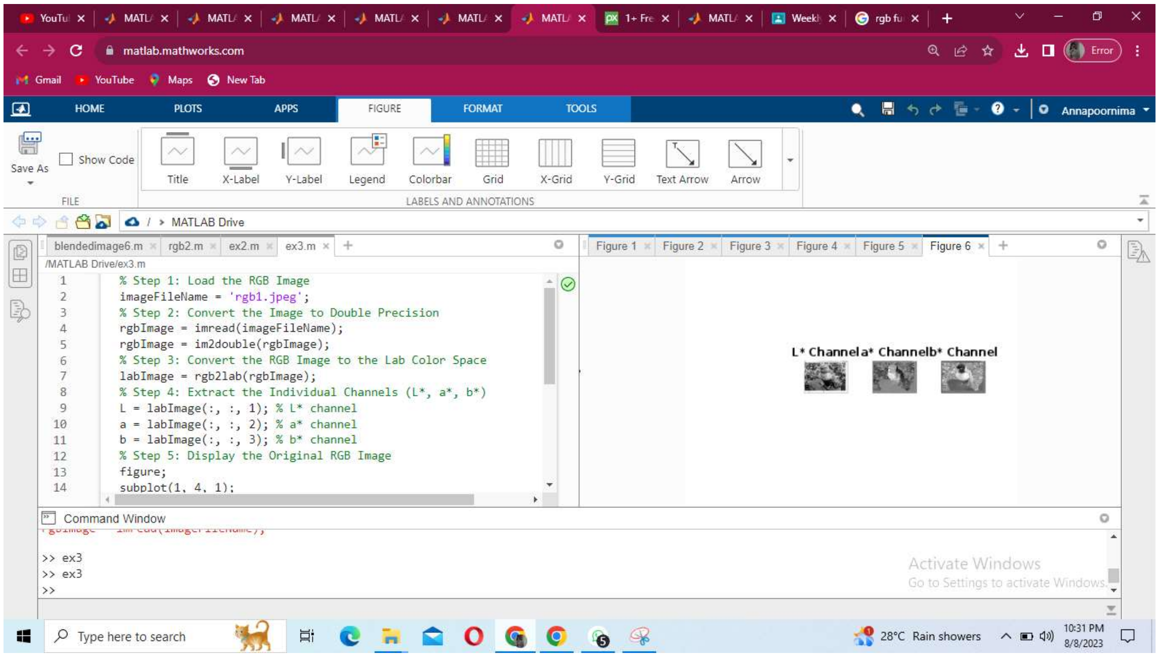
```
        fullname = get_full_filename(filename);

Error in ex2 (line 4)
rgbImage = imread(imageFileName);

>> ex2
>>
```

EXERCISE 3: RGB to Lab Conversion

Write a MATLAB function that converts an RGB image to the Lab color space using the rgb2lab function. Display the individual channels (L*, a*, and b*) of the Lab image separately.

```
%Code :
imageFileName = 'rgb1.jpeg';
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);labImage = rgb2lab(rgbImage);
L = labImage(:, :, 1); % L* channel
a = labImage(:, :, 2); % a* channel
b = labImage(:, :, 3); % b* channel
figure;
subplot(1, 4, 1);
imshow(rgbImage,[]);
title('Original RGB Image');
figure;
subplot(1, 4, 2);
imshow(L, []);
title('L* Channel');
subplot(1, 4, 3);
imshow(a, []);
```

```
title('a* Channel');
subplot(1, 4, 4);
imshow(b, []);
title('b* Channel');
```



/MATLAB Drive/ex3.m

```
% Step 1: Load the RGB Image
imageFileName = 'rgb1.jpeg';
% Step 2: Convert the Image to Double Precision
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
% Step 3: Convert the RGB Image to the Lab Color Space
labImage = rgb2lab(rgbImage);
% Step 4: Extract the Individual Channels (L*, a*, b*)
L = labImage(:, :, 1); % L* channel
a = labImage(:, :, 2); % a* channel
b = labImage(:, :, 3); % b* channel
% Step 5: Display the Original RGB Image
figure;
subplot(1, 4, 1);
```

Command Window

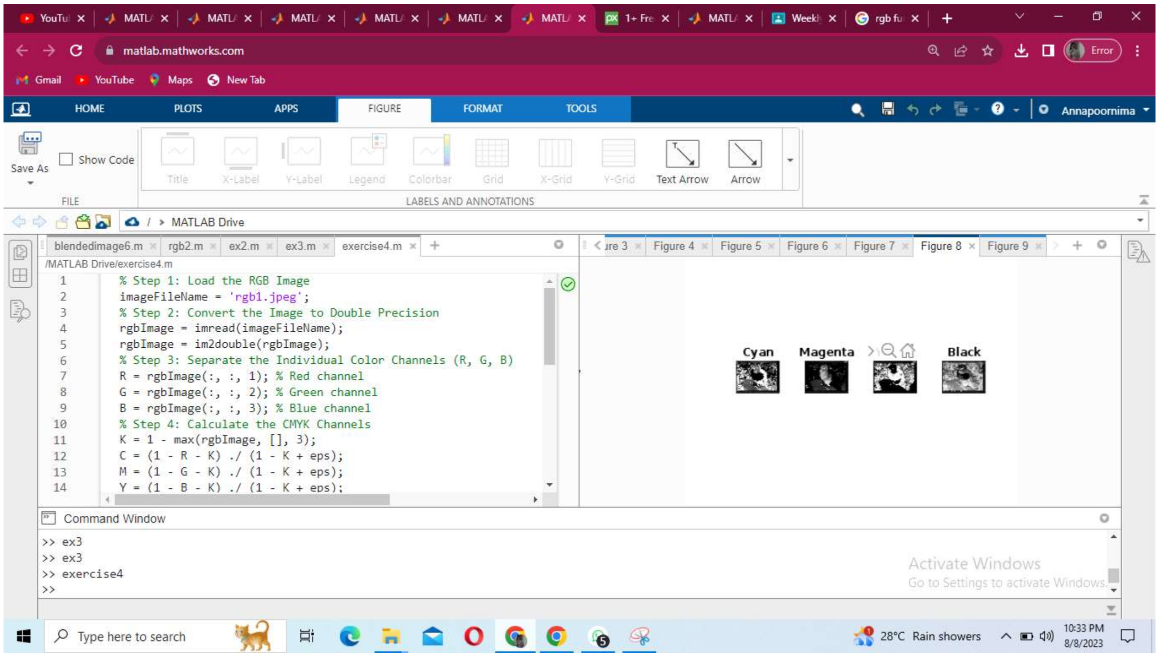
```
>> ex3
>> ex3
>>
```

EXERCISE 4: RGB to CMYK Conversion

Write a MATLAB function that converts an RGB image to the CMYK color space using the formulae:
C=1-R
M=1-G
Y=1-B
K = min(C, M, Y)
Display the individual channels (Cyan, Magenta, Yellow, and Black) of the CMYK image separately.

```
% Code :
imageFileName = 'rgb1.jpeg';
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
R = rgbImage(:, :, 1);
G = rgbImage(:, :, 2);
B = rgbImage(:, :, 3);
K = 1 - max(rgbImage, [], 3);
C = (1 - R - K) ./ (1 - K + eps);
M = (1 - G - K) ./ (1 - K + eps);
Y = (1 - B - K) ./ (1 - K + eps);
```

```
cmykImage = cat(3, C, M, Y, K);
figure;
imshow(rgbImage);
title('Original RGB Image');
figure;
subplot(1, 4, 1);
imshow(C);
title('Cyan');
subplot(1, 4, 2);
imshow(M);
title('Magenta');
subplot(1, 4, 3);
imshow(Y);
title('Yellow');
subplot(1, 4, 4);
imshow(K);
title('Black');
```



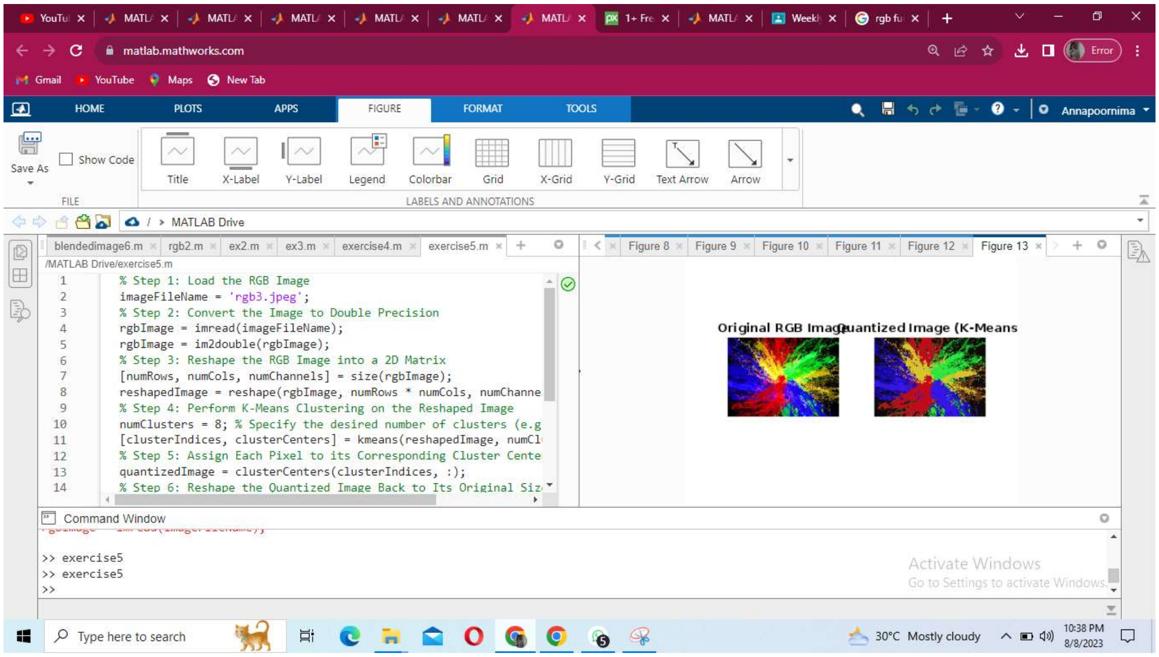
```
% Step 1: Load the RGB Image
imageFileName = 'rgb1.jpeg';
% Step 2: Convert the Image to Double Precision
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
% Step 3: Separate the Individual Color Channels (R, G, B)
R = rgbImage(:, :, 1); % Red channel
G = rgbImage(:, :, 2); % Green channel
B = rgbImage(:, :, 3); % Blue channel
% Step 4: Calculate the CMYK Channels
K = 1 - max(rgbImage, [], 3);
C = (1 - R - K) ./ (1 - K + eps);
M = (1 - G - K) ./ (1 - K + eps);
Y = (1 - B - K) ./ (1 - K + eps);
```

```
>> ex3
>> ex3
>> exercise4
>>
```

EXERCISE 5: Color Quantization

Write a MATLAB function that performs color quantization on an RGB image. Use the kmeans function to cluster the colors into a specified number of clusters (e.g., 8 colors) and replace each pixel's color with the
centroid color of the corresponding cluster. Display the original and quantized images side by side.

```
%Code :
imageFileName = 'rgb3.jpeg';
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
[numRows, numCols, numChannels] = size(rgbImage);
reshapedImage = reshape(rgbImage, numRows * numCols, numChannels);
numClusters = 8;
[clusterIndices, clusterCenters] = kmeans(reshapedImage, numClusters);
quantizedImage = clusterCenters(clusterIndices, :);
quantizedImage = reshape(quantizedImage, numRows, numCols, numChannels);
figure;
subplot(1, 2, 1);
imshow(rgbImage);
title('Original RGB Image');
subplot(1, 2, 2);
imshow(quantizedImage);
title('Quantized Image (K-Means)');
```



/MATLAB Drive/exercise5.m

```
% Step 1: Load the RGB Image
imageFileName = 'rgb3.jpeg';
% Step 2: Convert the Image to Double Precision
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
% Step 3: Reshape the RGB Image into a 2D Matrix
[numRows, numCols, numChannels] = size(rgbImage);
reshapedImage = reshape(rgbImage, numRows * numCols, numChanne
% Step 4: Perform K-Means Clustering on the Reshaped Image
numClusters = 8; % Specify the desired number of clusters (e.g
[clusterIndices, clusterCenters] = kmeans(reshapedImage, numCl
% Step 5: Assign Each Pixel to its Corresponding Cluster Cente
quantizedImage = clusterCenters(clusterIndices, :);
% Step 6: Reshape the Quantized Image Back to Its Original Siz
```

Command Window

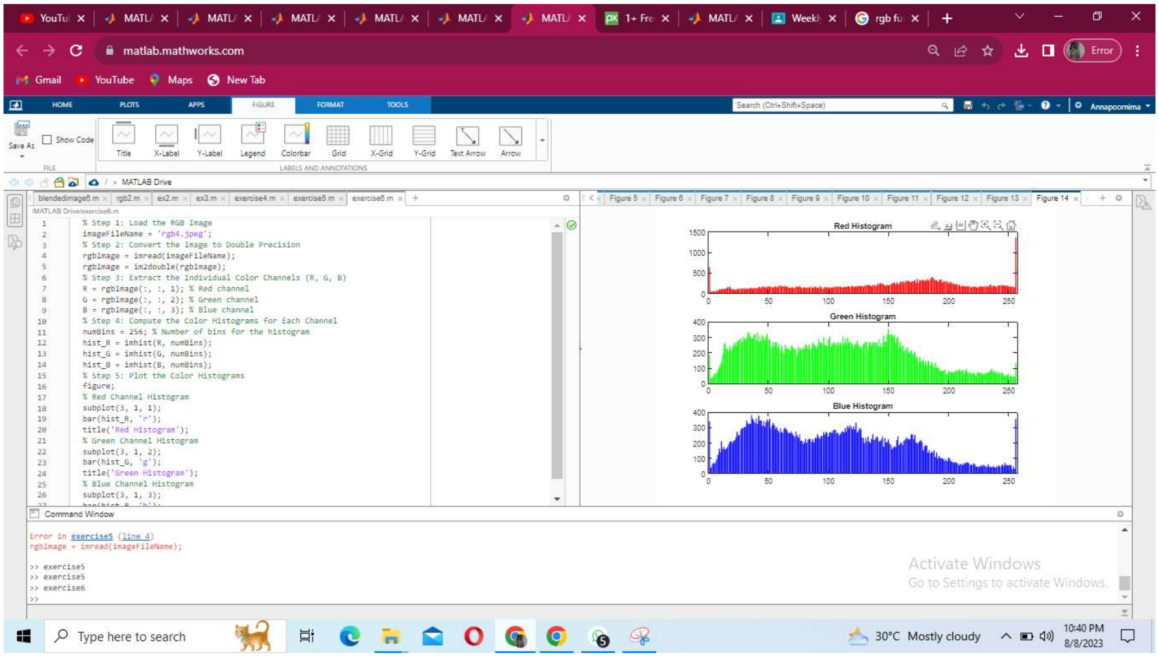
```
>> exercise5
>> exercise5
>>
```

EXERCISE 6: Color Histogram

Write a MATLAB function that computes and plots the color histogram of an RGB image. The histogram should show the distribution of colors in each channel (Red, Green, and Blue) separately.

```
% Code :
imageFileName = 'rgb4.jpg';
```

```
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
R = rgbImage(:, :, 1);
G = rgbImage(:, :, 2);
B = rgbImage(:, :, 3);
numBins = 256;
hist_R = imhist(R, numBins);
hist_G = imhist(G, numBins);
hist_B = imhist(B, numBins);
figure;
subplot(3, 1, 1);
bar(hist_R, 'r');
title('Red Histogram');
subplot(3, 1, 2);
bar(hist_G, 'g');
title('Green Histogram');
subplot(3, 1, 3);
bar(hist_B, 'b');
title('Blue Histogram');
```



```
% Step 1: Load the RGB Image
imageFileName = 'rgb4.jpeg';
% Step 2: Convert the Image to Double Precision
rgbImage = imread(imageFileName);
rgbImage = im2double(rgbImage);
% Step 3: Extract the Individual Color Channels (R, G, B)
R = rgbImage(:, :, 1); % Red channel
G = rgbImage(:, :, 2); % Green channel
B = rgbImage(:, :, 3); % Blue channel
% Step 4: Compute the Color Histograms for Each Channel
numBins = 256; % Number of bins for the histogram
hist_R = imhist(R, numBins);
hist_G = imhist(G, numBins);
hist_B = imhist(B, numBins);
% Step 5: Plot the Color Histograms
figure;
% Red Channel Histogram
subplot(3, 1, 1);
bar(hist_R, 'r');
title('Red Histogram');
% Green Channel Histogram
subplot(3, 1, 2);
bar(hist_G, 'g');
title('Green Histogram');
% Blue Channel Histogram
subplot(3, 1, 3);
```

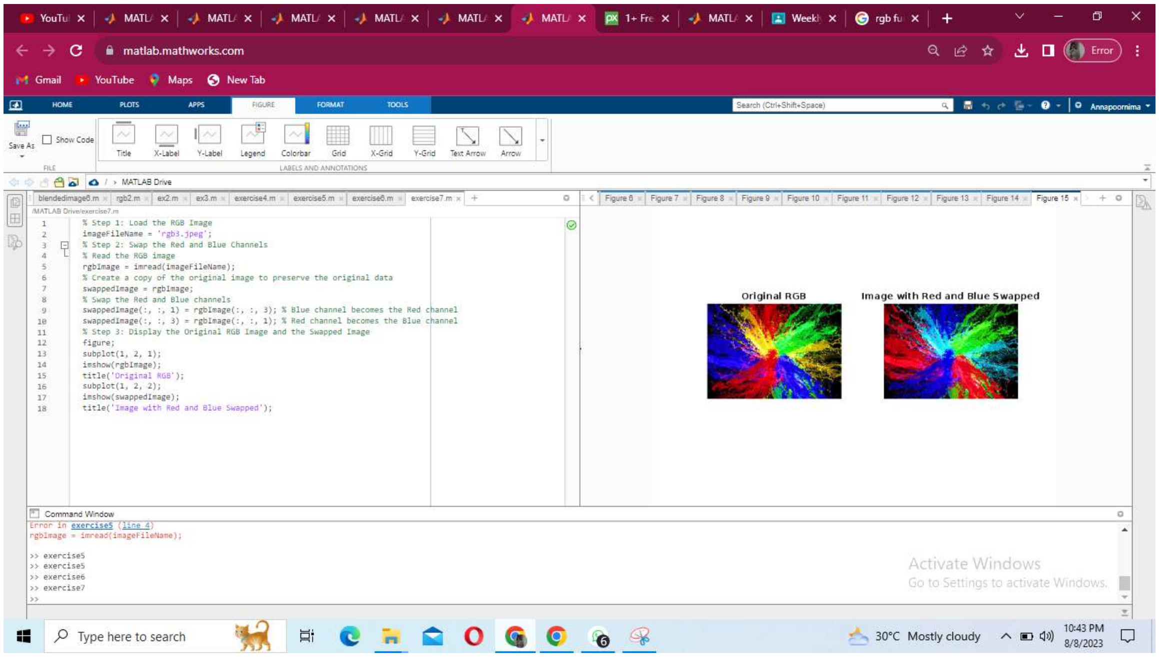
```
Error in exercise5 (line 4)
rgbImage = imread(imageFileName);

>> exercise5
>> exercise5
>> exercise6
>>
```

EXERCISE 7: Color Channel Swap

Write a MATLAB function that swaps the Red and Blue channels of an RGB image. Display the original and swapped images side by side.

```
% Code :

imageFileName = 'rgb3.jpeg';
```

```
rgbImage = imread(imageFileName);
swappedImage = rgbImage;
swappedImage(:, :, 1) = rgbImage(:, :, 3);
swappedImage(:, :, 3) = rgbImage(:, :, 1);
figure;
subplot(1, 2, 1);
imshow(rgbImage);
title('Original RGB');
subplot(1, 2, 2);
imshow(swappedImage);
title('Image with Red and Blue Swapped');
```



matlab.mathworks.com
MATLAB Drive
Original RGB
Image with Red and Blue Swapped
Activate Windows
Go to Settings to activate Windows.

/MATLAB Drive/exercise7.m

```
% Step 1: Load the RGB Image
imageFileName = 'rgb3.jpeg';
% Step 2: Swap the Red and Blue Channels
% Read the RGB image
rgbImage = imread(imageFileName);
% Create a copy of the original image to preserve the original data
swappedImage = rgbImage;
% Swap the Red and Blue channels
swappedImage(:, :, 1) = rgbImage(:, :, 3); % Blue channel becomes the Red channel
swappedImage(:, :, 3) = rgbImage(:, :, 1); % Red channel becomes the Blue channel
% Step 3: Display the Original RGB Image and the Swapped Image
figure;
subplot(1, 2, 1);
imshow(rgbImage);
title('Original RGB');
subplot(1, 2, 2);
imshow(swappedImage);
title('Image with Red and Blue Swapped');
```

Command Window

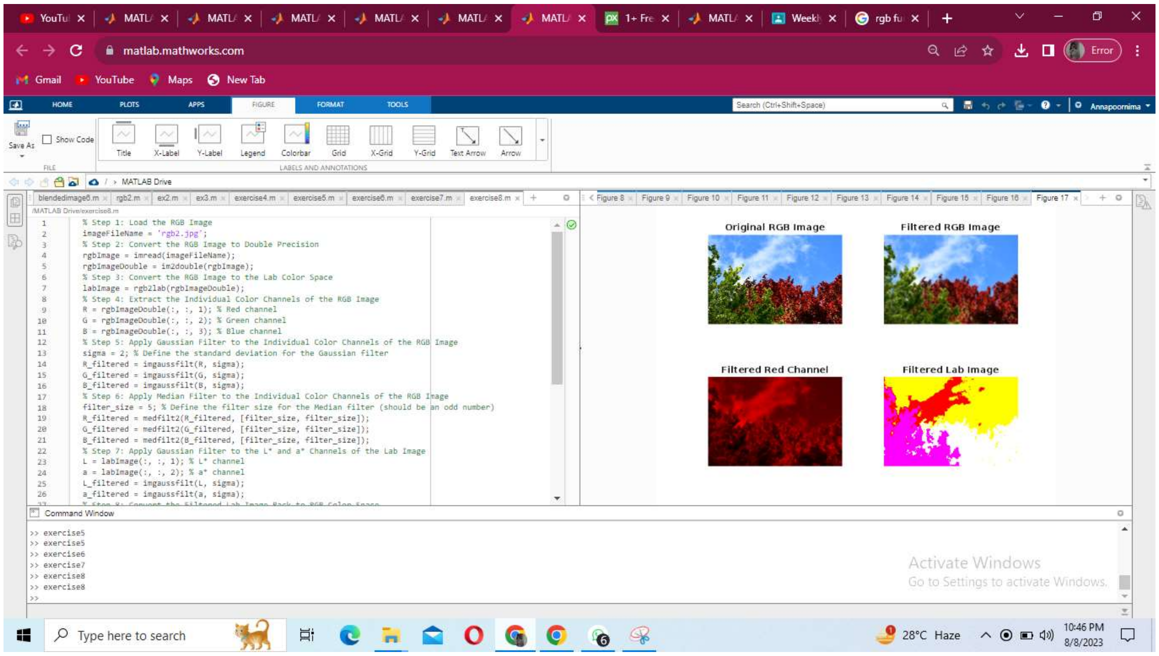
```
Error in exercise5 (line 4)
rgbImage = imread(imageFileName);

>> exercise5
>> exercise5
>> exercise6
>> exercise7
>>
```

EXERCISE 8: Image Filtering in Different Color Spaces

Choose an image and apply different image filters (e.g., Gaussian, Median) to the individual channels of the RGB image and also to the L* and a* channels of the Lab image. Compare the filtered images in different color spaces.

```
% Code :
```

```matlab
imageFileName = 'rgb2.jpg';
rgbImage = imread(imageFileName);
rgbImageDouble = im2double(rgbImage);
labImage = rgb2lab(rgbImageDouble);
R = rgbImageDouble(:, :, 1);
G = rgbImageDouble(:, :, 2);
B = rgbImageDouble(:, :, 3);
sigma = 2;
R_filtered = imgaussfilt(R, sigma);
G_filtered = imgaussfilt(G, sigma);
B_filtered = imgaussfilt(B, sigma);
filter_size = 5;
R_filtered = medfilt2(R_filtered, [filter_size, filter_size]);
G_filtered = medfilt2(G_filtered, [filter_size, filter_size]);
B_filtered = medfilt2(B_filtered, [filter_size, filter_size]);
L = labImage(:, :, 1);
a = labImage(:, :, 2);
L_filtered = imgaussfilt(L, sigma);
a_filtered = imgaussfilt(a, sigma);
labImage_filtered = labImage;
labImage_filtered(:, :, 1) = L_filtered;
labImage_filtered(:, :, 2) = a_filtered;
rgbImageFiltered = lab2rgb(labImage_filtered);
figure;
subplot(2, 2, 1);
imshow(rgbImage);
title('Original RGB Image');
subplot(2, 2, 2);
imshow(cat(3, R_filtered, G_filtered, B_filtered));
title('Filtered RGB Image');
subplot(2, 2, 3);
imshow(cat(3, R_filtered, zeros(size(G_filtered)),
zeros(size(B_filtered))));
title('Filtered Red Channel');
subplot(2, 2, 4);
imshow(labImage_filtered);
title('Filtered Lab Image');
```



```
% Step 1: Load the RGB Image
imageFileName = 'rgb2.jpg';
% Step 2: Convert the RGB Image to Double Precision
rgbImage = imread(imageFileName);
rgbImageDouble = im2double(rgbImage);
% Step 3: Convert the RGB Image to the Lab Color Space
labImage = rgb2lab(rgbImageDouble);
% Step 4: Extract the Individual Color Channels of the RGB Image
R = rgbImageDouble(:, :, 1); % Red channel
G = rgbImageDouble(:, :, 2); % Green channel
B = rgbImageDouble(:, :, 3); % Blue channel
% Step 5: Apply Gaussian Filter to the Individual Color Channels of the RGB Image
sigma = 2; % Define the standard deviation for the Gaussian filter
R_filtered = imgaussfilt(R, sigma);
G_filtered = imgaussfilt(G, sigma);
B_filtered = imgaussfilt(B, sigma);
% Step 6: Apply Median Filter to the Individual Color Channels of the RGB Image
filter_size = 5; % Define the filter size for the Median filter (should be an odd number)
R_filtered = medfilt2(R_filtered, [filter_size, filter_size]);
G_filtered = medfilt2(G_filtered, [filter_size, filter_size]);
B_filtered = medfilt2(B_filtered, [filter_size, filter_size]);
% Step 7: Apply Gaussian Filter to the L* and a* Channels of the Lab Image
L = labImage(:, :, 1); % L* channel
a = labImage(:, :, 2); % a* channel
L_filtered = imgaussfilt(L, sigma);
a_filtered = imgaussfilt(a, sigma);
```

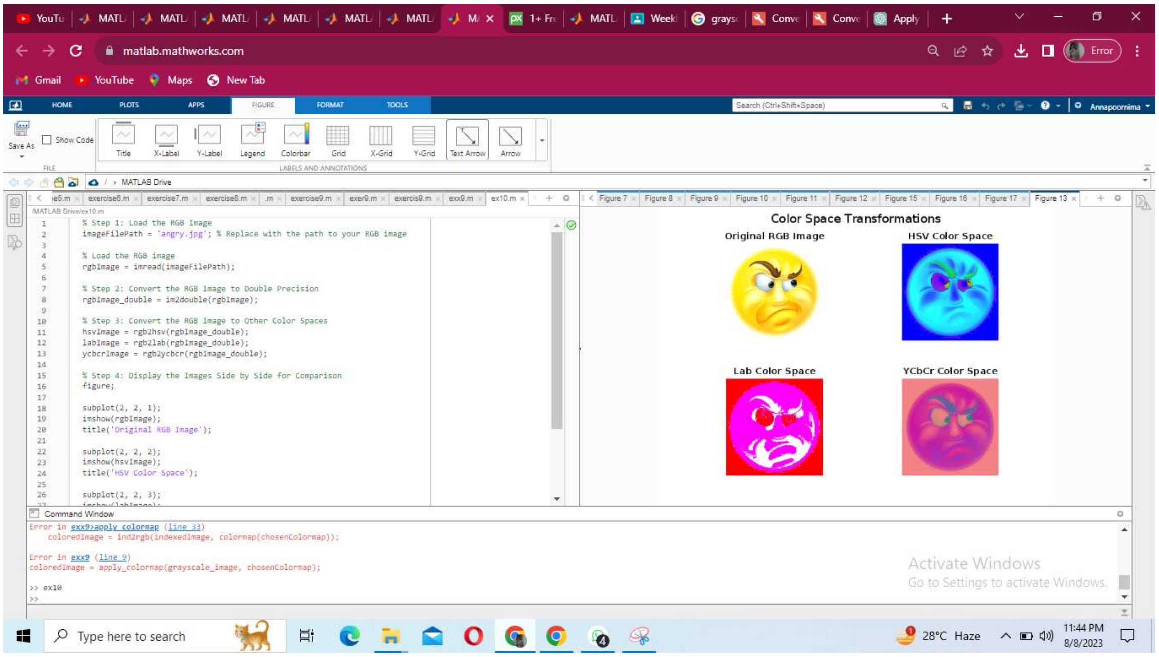
```
>> exercise5
>> exercise5
>> exercise6
>> exercise7
>> exercise8
>> exercise8
>>
```

EXERCISE 9: Color Image Reconstruction

Convert a grayscale image to an RGB image using a colormap of your choice. Display both the original grayscale image and the colormap-based RGB image side by side.

```
% Step 1: Load the Grayscale Image
grayscale_image_path = 'ggg.jpeg'; % Replace with the actual path to your image
grayscale_image = imread(grayscale_image_path);

% Step 2: Choose a Colormap
chosenColormap = 'jet'; % You can change this to any built-in colormap

% Step 3: Apply the Colormap to the Grayscale Image
coloredImage = apply_colormap(grayscale_image, chosenColormap);

% Step 4: Display the Original Grayscale Image and the Colormap-based RGB Image
figure;

subplot(1, 2, 1);
imshow(grayscale_image, []);
```

```
title('Original Grayscale Image');
colormap(gray); % Apply 'gray' colormap

subplot(1, 2, 2);
imshow(coloredImage);
title(['Colormap: ', chosenColormap]);

sgtitle('Grayscale Image and Colormap-based RGB Image');

% Function to apply colormap to grayscale image
function coloredImage = apply_colormap(grayscale_image, chosenColormap)
   colormapSize = 256; % Number of color levels in the colormap

   % Convert grayscale image to indexed image using colormap
   indexedImage = gray2ind(grayscale_image, colormapSize);

   % Apply the chosen colormap to indexed image
   coloredImage = ind2rgb(indexedImage, colormap(chosenColormap));
```



/MATLAB Drive/ex10.m

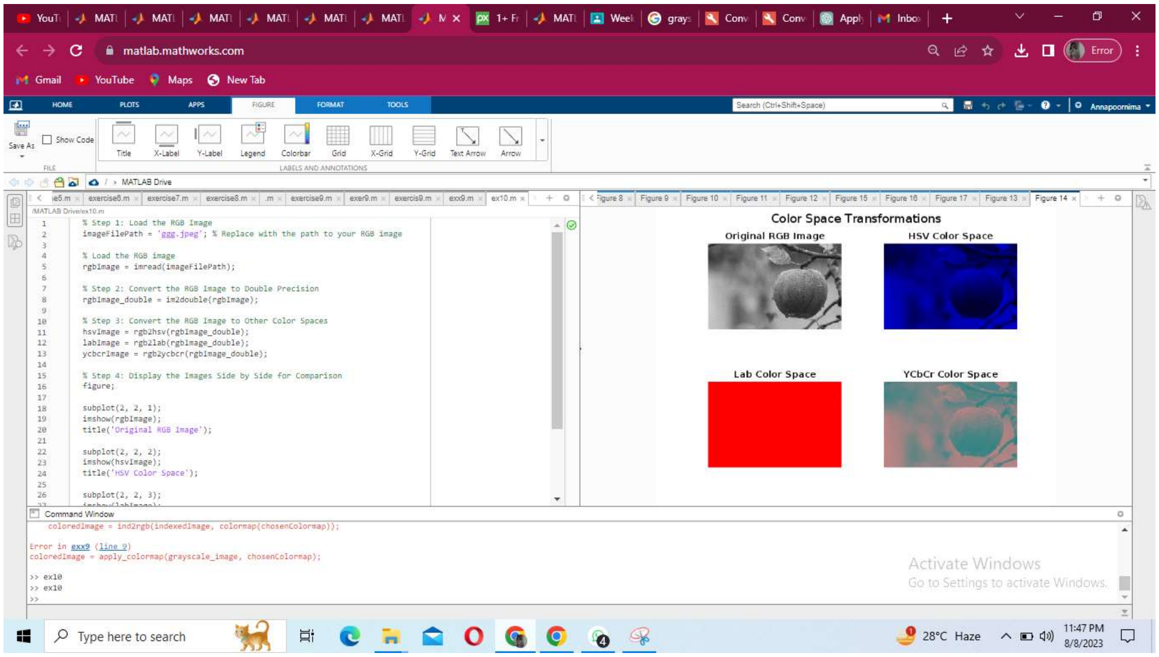
```
% Step 1: Load the RGB Image
imageFilePath = 'angry.jpg'; % Replace with the path to your RGB image

% Load the RGB image
rgbImage = imread(imageFilePath);

% Step 2: Convert the RGB Image to Double Precision
rgbImage_double = im2double(rgbImage);

% Step 3: Convert the RGB Image to Other Color Spaces
hsvImage = rgb2hsv(rgbImage_double);
labImage = rgb2lab(rgbImage_double);
ycbcrImage = rgb2ycbcr(rgbImage_double);

% Step 4: Display the Images Side by Side for Comparison
figure;

subplot(2, 2, 1);
imshow(rgbImage);
title('Original RGB Image');

subplot(2, 2, 2);
imshow(hsvImage);
title('HSV Color Space');

subplot(2, 2, 3);
```

```
Error in exx9>apply_colormap (line 33)
    coloredImage = ind2rgb(indexedImage, colormap(chosenColormap));

Error in exx9 (line 9)
coloredImage = apply_colormap(grayscale_image, chosenColormap);

>> ex10
>>
```

EXERCISE 10: Color Space Transformation Visualization

Load an RGB image and visualize the effect of transforming it between different color spaces (e.g., RGB to HSV, RGB to Lab, etc.) by displaying the images side by side for comparison.

```
% Code :
imageFilePath = 'ggg.jpeg';
rgbImage = imread(imageFilePath);
rgbImage_double = im2double(rgbImage);
hsvImage = rgb2hsv(rgbImage_double);
labImage = rgb2lab(rgbImage_double);
ycbcrImage = rgb2ycbcr(rgbImage_double);
figure;
subplot(2, 2, 1);
imshow(rgbImage);
title('Original RGB Image');
subplot(2, 2, 2);
imshow(hsvImage);
title('HSV Color Space');
subplot(2, 2, 3);
imshow(labImage);
title('Lab Color Space');
subplot(2, 2, 4);
imshow(ycbcrImage);
title('YCbCr Color Space');
sgtitle('Color Space Transformations');
```



```
% Step 1: Load the RGB Image
imageFilePath = 'ggg.jpeg'; % Replace with the path to your RGB image

% Load the RGB image
rgbImage = imread(imageFilePath);

% Step 2: Convert the RGB Image to Double Precision
rgbImage_double = im2double(rgbImage);

% Step 3: Convert the RGB Image to Other Color Spaces
hsvImage = rgb2hsv(rgbImage_double);
labImage = rgb2lab(rgbImage_double);
ycbcrImage = rgb2ycbcr(rgbImage_double);

% Step 4: Display the Images Side by Side for Comparison
figure;

subplot(2, 2, 1);
imshow(rgbImage);
title('Original RGB Image');

subplot(2, 2, 2);
imshow(hsvImage);
title('HSV Color Space');

subplot(2, 2, 3);
```

```
    coloredImage = ind2rgb(indexedImage, colormap(chosenColormap));

Error in exx9 (line 9)
coloredImage = apply_colormap(grayscale_image, chosenColormap);

>> ex10
>> ex10
>>
```