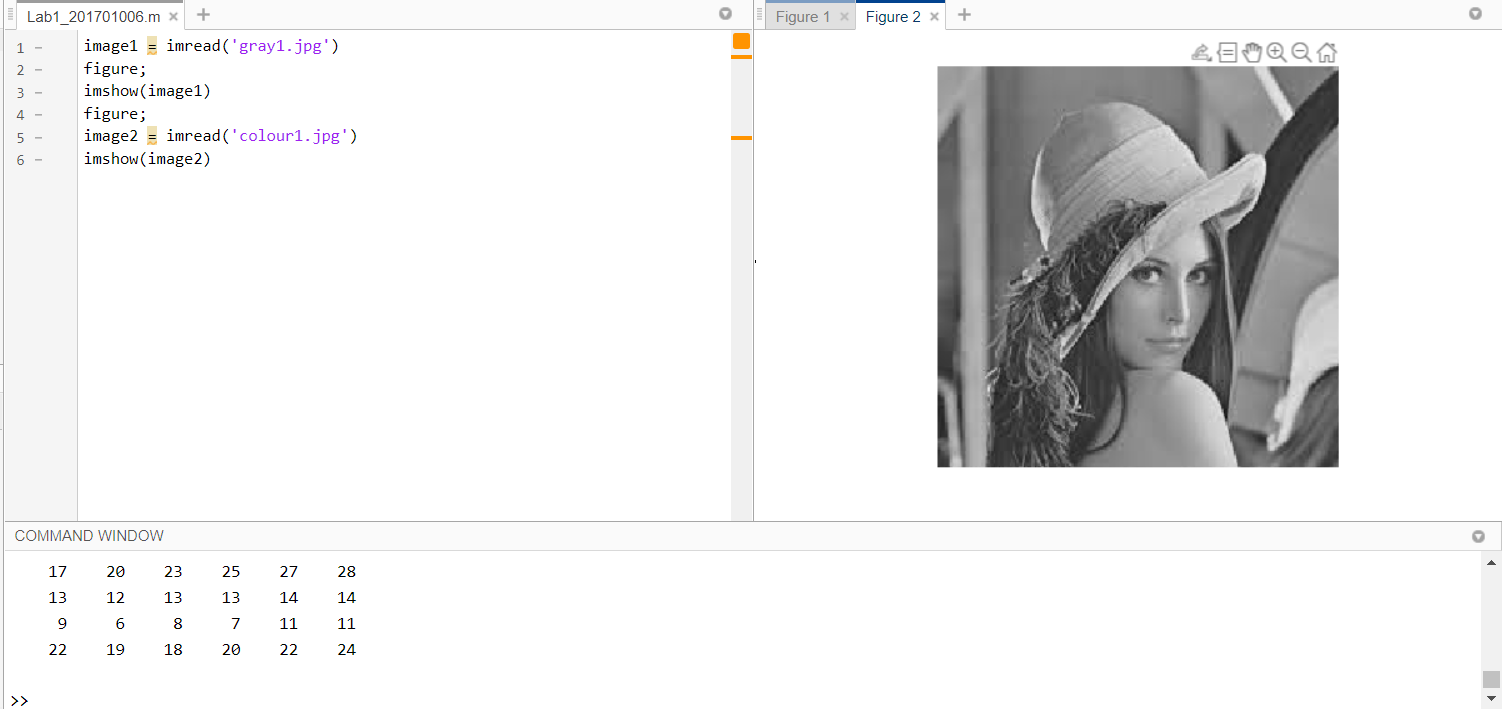
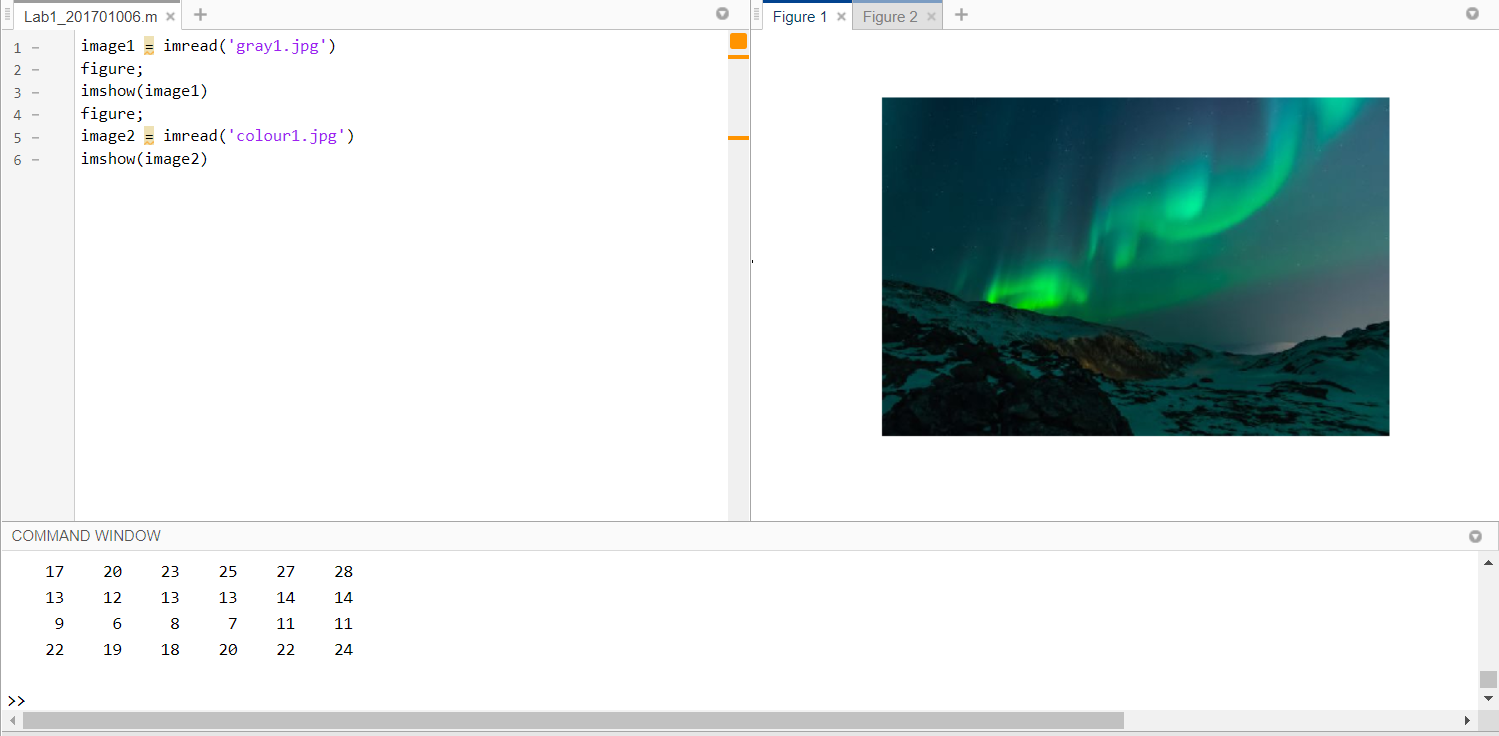
**NAME: SWARNADITYA TAMONASH MAITRA ID: 201701006**

**DIGITAL IMAGE PROCESSING: LAB – 1 | Coded using MATLAB**

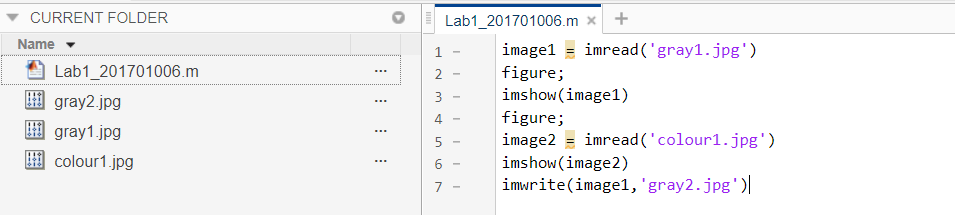
1. Answers to question1
   1. CODE: image1 = imread('gray1.jpg') and image2=imread(‘colour1.jpg’)
   2. CODE: imshow(image1) and imshow(image2)
      1. OUTPUT: Gray-Scale Output



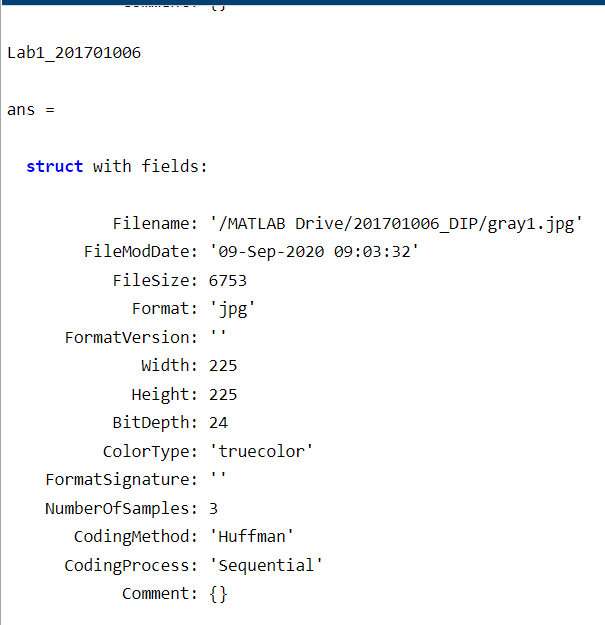
* + 1. OUTPUT: Colour Output



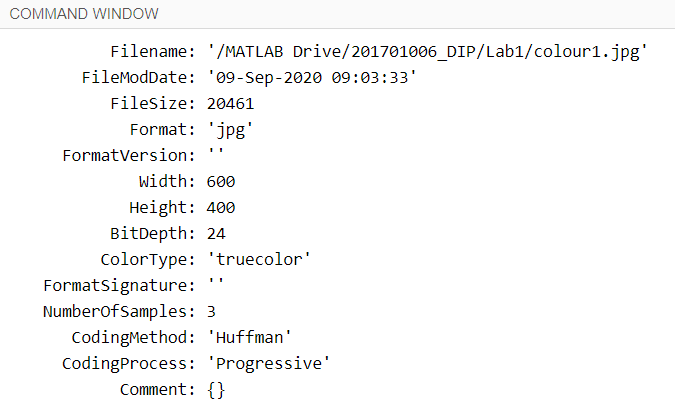
* 1. CODE: imwrite(image1,’gray2.jpg’) OR imwrite(image2,’colour2.png’)
     1. OUTPUT:



* 1. CODE: imfinfo(‘gray1.jpg’) AND imfinfo(‘colour1.jpg’)
     1. OUTPUT (for gray1):



* + 1. OUTPUT (for colour1):



* 1. **CODE:**

**%% Reading**

gray = imread('gray.jpg')

color = imread('colour.jpg')

**%% Convert dataclass of the image**

%Convert dataclass of gray image

gray2=cast(gray>75,'uint8').\*gray;

figure(1)

subplot(1,3,1);

imshow(gray2);

title('Input gray image','FontSize',8);

bool=cast(gray2,'logical');

subplot(1,3,2);

imshow(bool);

title(' "logical" datatype','FontSize',8);

doub=cast(gray2,'double');

subplot(1,3,3);

imshow(doub);

title(' "double" datatype','FontSize',8);

sgtitle('change data class of gray Image');

%Convert dataclass of RGB image

color2=cast(color>75,'uint8').\*color;

figure(2)

subplot(1,3,1);

imshow(color2);

title('Input RGB image','FontSize',8);

bool=cast(color2,'logical');

bool=bool(:,:,1)&bool(:,:,2)&bool(:,:,3);

subplot(1,3,2);

imshow(bool);

title(' "logical" datatype','FontSize',8);

doub=cast(color2,'double');

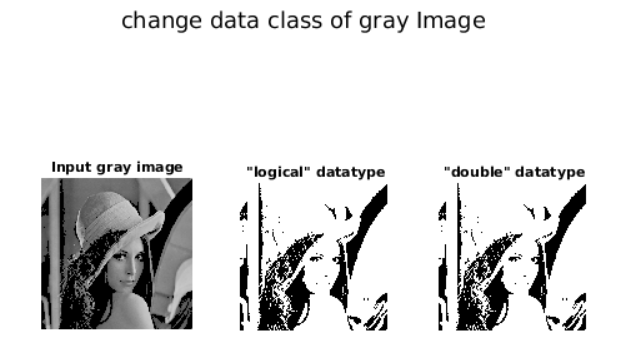
subplot(1,3,3);

imshow(doub);

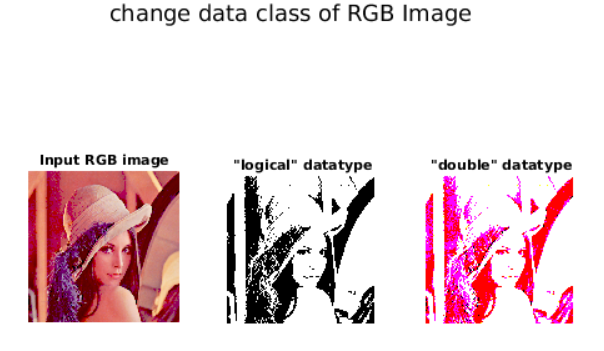
title(' "double" datatype','FontSize',8);

sgtitle('change data class of RGB Image');

* + 1. **OUTPUT 1:**



* + 1. **OUTPUT 2:**



* 1. **CODE:**

% Convert image rgb image to gray

% Average method

figure(1)

subplot(2,2,1);

imshow(I);

title('input RGB image');

new\_gray=(I(:,:,1)+I(:,:,2)+I(:,:,3))/3;

subplot(2,2,2);

imshow(new\_gray);

title('RGB to gray using avg. method');

%Weighted sum method(Luminosity method)

subplot(2,2,3);

new\_gray\_2=(0.3\*I(:,:,1)+0.59\*I(:,:,2)+0.11\*I(:,:,3));

imshow(new\_gray\_2);

title('RGB to gray(weighted avg. method)');

%Convert rgb to binary image

bin=(I(:,:,1)+I(:,:,2)+I(:,:,3))/3;

bin=255\*(bin/127);

subplot(2,2,4);

imshow(bin);

title('RGB to binary image');

figure(2)

subplot(2,2,1);

imshow(C);

title('Input Gray image');

%Convert gray to binary image

bin\_2=255\*(C/127);

subplot(2,2,2);

imshow(bin\_2);

title('gray to binary image');

%Convert gray to rgb(pseudo color) Image. Using a new 512x512 Lena Image for the same (OUTPUT 3)

pseudo=zeros([size(gray),3],'uint8');

pseudo(:,:,3)=255\*tan(pi\*double(gray)/255 - pi/4);

pseudo(:,:,2)=255\*tan(pi\*double(gray)/255 - pi/2);

pseudo(:,:,1)=255\*tan(pi\*double(gray)/255 - pi/3);

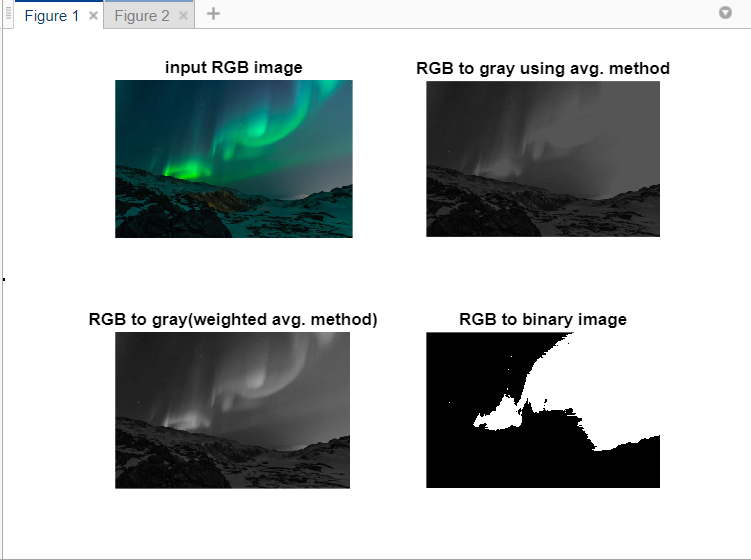
subplot(2,2,3);

imshow(pseudo);

title('gray to RGB(pseudo color) image');

sgtitle('Gray to other types');

**OUTPUT 1:**



**OUTPUT 2:**



**OUTPUT 3:**



* 1. **CODE:**

clc;

clear all;

close all;

**%% Reading Image**

I = imread('colour1.jpg');

C = imread('gray1.jpg');

**%% Flipping Horizontally**

flippedI = I(:,[end:-1:1],:);

figure(1)

subplot(1,2,1);

imshow(flippedI);

flippedC = C(:,[end:-1:1],:);

subplot(1,2,2);

imshow(flippedC);

**%% Flipping Vertically**

flippedI2 = I([end:-1:1],:,:);

figure(2)

subplot(1,2,1);

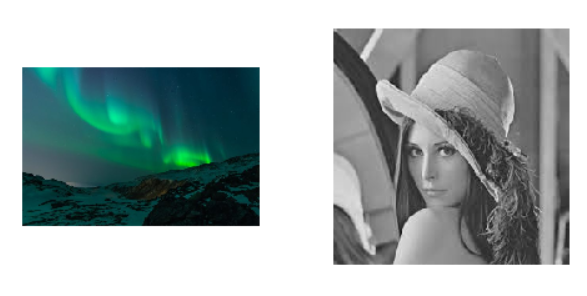
imshow(flippedI2);

flippedC2 = C([end:-1:1],:,:);

subplot(1,2,2);

imshow(flippedC2);

**OUTPUT 1(flipped horizontally):**



**OUTPUT 2(flipped vertically):**



* 1. **CODE**
     1. **For Gray**

%% Perform Arithmetic operations on images

%Perform Arithmetic operations on gray images

close all;

gray2=flip\_horizontal(gray);

figure(1)

subplot(3,3,1);

imshow(gray);

title('Input image 1','FontSize',8);

subplot(3,3,2);

imshow(gray2);

title('Input image 2','FontSize',8);

subplot(3,3,3);

imshow(gray+gray2);

title('Addition','FontSize',8);

subplot(3,3,4);

imshow(gray-gray2);

title('Subtraction','FontSize',8);

subplot(3,3,5);

imshow(gray.\*gray2);

title('Multiplication','FontSize',8);

subplot(3,3,6);

imshow(gray./gray2);

title('division','FontSize',8);

subplot(3,3,7);

imshow(abs(gray-gray2));

title('Absolute difference','FontSize',8);

subplot(3,3,8);

imshow(255-gray);

title('Complement on input 1','FontSize',8);

subplot(3,3,9);

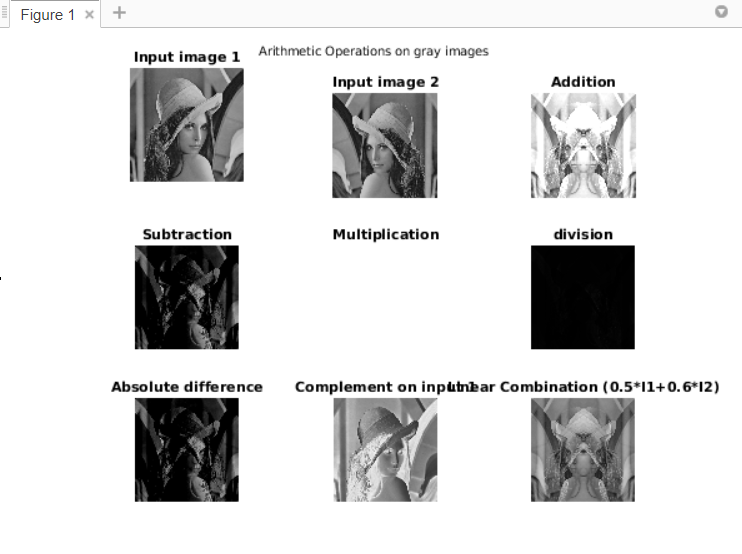
imshow(0.5\*gray+0.6\*gray2);

title('Linear Combination (0.5\*I1+0.6\*I2)','FontSize',8);

sgt.FontSize = 2;

sgtitle('Arithmetic Operations on gray images','FontSize',7,'Position',[5.5 0.4]);

**OUTPUT:**



* + 1. **For Colour**

%% Perform Arithmetic operations on images

%Perform Arithmetic operations on color images

color2=flip\_horizontal(color);

figure(2)

subplot(3,3,1);

imshow(color);

title('Input image 1','FontSize',8);

subplot(3,3,2);

imshow(color2);

title('Input image 2','FontSize',8);

subplot(3,3,3);

imshow(color+color2);

title('Addition','FontSize',8);

subplot(3,3,4);

imshow(color-color2);

title('Subtraction','FontSize',8);

subplot(3,3,5);

imshow(color.\*color2);

title('Multiplication','FontSize',8);

subplot(3,3,6);

imshow(color./color2);

title('division','FontSize',8);

subplot(3,3,7);

imshow(abs(color-color2));

title('Absolute difference','FontSize',8);

subplot(3,3,8);

imshow(255-color);

title('Complement on input 1','FontSize',8);

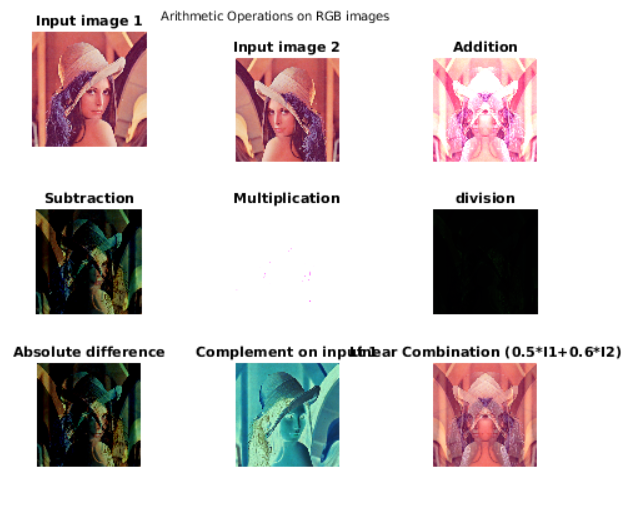
subplot(3,3,9);

imshow(0.5\*color+0.6\*color2);

title('Linear Combination (0.5\*I1+0.6\*I2)','FontSize',8);

sgtitle('Arithmetic Operations on RGB images','FontSize',7);

**OUTPUT:**



* 1. **CODE:**

%% Perform relational operations images

close all;

%Perform relational operations gray images

gray2=imread('Barbara\_gray.jpg');

figure(1)

subplot(3,3,1);

imshow(gray);

title('Input image 1','FontSize',8);

subplot(3,3,3);

imshow(gray2);

title('Input image 2','FontSize',8);

subplot(3,3,4);

imshow(gray==gray2);

title('== operator','FontSize',8);

subplot(3,3,5);

imshow(gray<gray2);

title('< operator','FontSize',8);

subplot(3,3,6);

imshow(gray>gray2);

title('> operator','FontSize',8);

subplot(3,3,7);

imshow(gray>=gray2);

title('>= operator','FontSize',8);

subplot(3,3,8);

imshow(gray<=gray2);

title('<= operator','FontSize',8);

subplot(3,3,9);

imshow(gray~=gray2);

title('~= operator','FontSize',8);

sgt.FontSize = 2;

sgtitle('Relational Operations on gray images','FontSize',7);

**%% Perform relational operations on images**

%Perform relational operations on color images

color2=imread('Barbara\_color.jpg');

figure(2)

subplot(3,3,1);

imshow(color);

title('Input image 1','FontSize',8);

subplot(3,3,3);

imshow(color2);

title('Input image 2','FontSize',8);

subplot(3,3,4);

o=color==color2;

o=255\*cast(o,'uint8');

imshow(o);

title('== operator','FontSize',8);

subplot(3,3,5);

o=color<color2;

o=255\*cast(o,'uint8');

imshow(o);

title('< operator','FontSize',8);

subplot(3,3,6);

o=color>color2;

o=255\*cast(o,'uint8');

imshow(o);

title('> operator','FontSize',8);

subplot(3,3,7);

o=color>=color2;

o=255\*cast(o,'uint8');

imshow(o);

title('>= operator','FontSize',8);

subplot(3,3,8);

o=color<=color2;

o=255\*cast(o,'uint8');

imshow(o);

title('<= operator','FontSize',8);

subplot(3,3,9);

o=color~=color2;

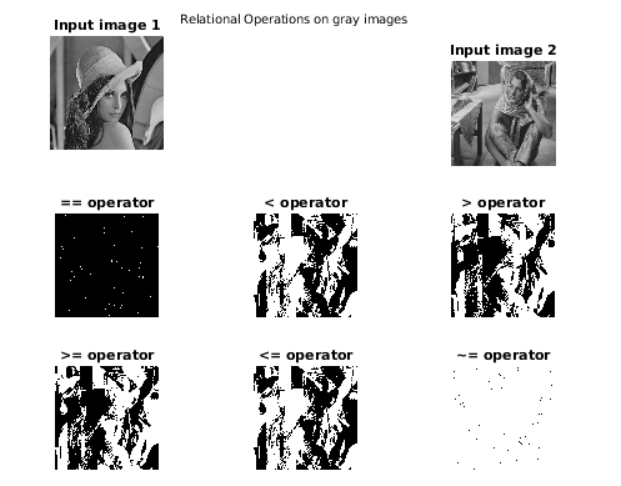
o=o(:,:,1)&o(:,:,2)&o(:,:,3);

imshow(o);

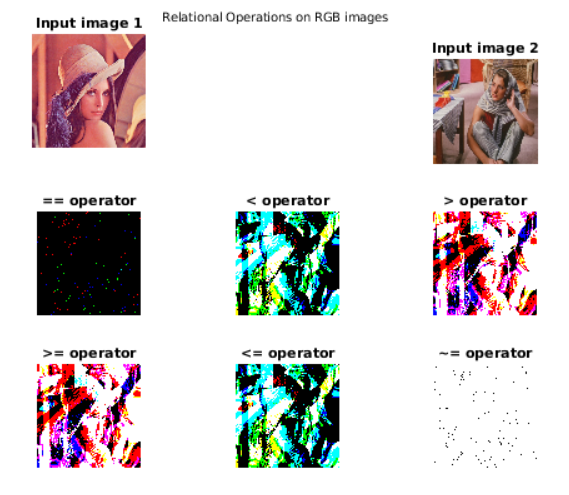
title('~= operator','FontSize',8);

sgtitle('Relational Operations on RGB images','FontSize',7);

**OUTPUT1**



**OUTPUT 2:**



* 1. **CODE:**

%% Perform logical operators on image

% Perform logical operators on gray image

close all;

gray1=cast(gray>75,'uint8').\*gray;

gray2=imread('Barbara\_gray.jpg');

gray2=cast(gray2>75,'uint8').\*gray2;

figure(1)

subplot(2,3,1);

imshow(gray1);

title('Input image 1','FontSize',8);

subplot(2,3,3);

imshow(gray2);

title('Input image 2','FontSize',8);

subplot(2,3,4);

imshow(gray1|gray2);

title('OR operator','FontSize',8);

subplot(2,3,5);

imshow(gray1&gray2);

title('AND operator','FontSize',8);

subplot(2,3,6);

imshow(~gray1);

title('NOT Operator on input 1','FontSize',8);

sgt.FontSize = 2;

sgtitle('Logical Operations on gray images','FontSize',7);

**%% Perform Logical operations on images**

%Perform Logical operations on color images

color1=cast(color>75,'uint8').\*color;

color2=imread('Barbara\_color.jpg');

color2=cast(color2>75,'uint8').\*color2;

figure(2)

subplot(2,3,1);

imshow(color1);

title('Input image 1','FontSize',8);

subplot(2,3,3);

imshow(color2);

title('Input image 2','FontSize',8);

subplot(2,3,4);

o=color1|color2;

o=255\*cast(o,'uint8');

imshow(o);

title('OR operator','FontSize',8);

subplot(2,3,5);

o=color1&color2;

o=255\*cast(o,'uint8');

imshow(o);

title('AND operator','FontSize',8);

subplot(2,3,6);

o=~color1;

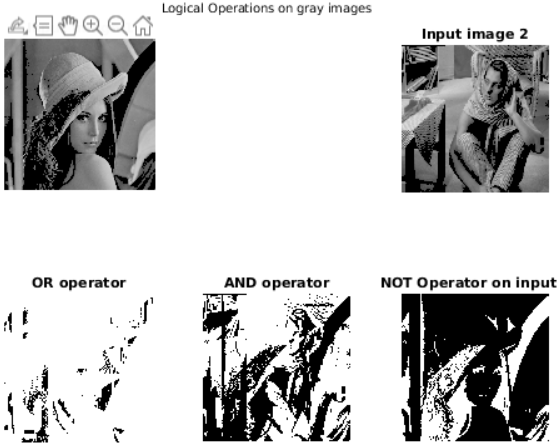
o=255\*cast(o,'uint8');

imshow(o);

title('NOT operator on Input 1','FontSize',8);

sgtitle('Logical Operations on RGB images','FontSize',7);

**OUTPUT 1:**



**OUTPUT 2:**



* 1. **CODE:**

**%% Compute average value for image**

n1=numel(gray);

gray=cast(gray,'double');

avg=0;

for i=1:n1

avg=avg+gray(i);

end

avg=avg/n1;

disp(avg);

n2=numel(color);

color=cast(color,'double');

avg=0;

for i=1:n2

avg=avg+color(i);

end

avg=avg/n2;

disp(AverageValue(color));

**OUTPUT 1 and OUTPUT 2:**



* 1. **CODE:**

**%% Extract subimage from an image**

%Compute subimage of gray image

close all;

figure(1)

subplot(2,2,1)

imshow(gray);

title('gray image','FontSize',8);

tl=input('Enter top-left corner of the subimage');

br=input('Enter bottom-right corner of the subimage');

subplot(2,2,2);

imshow(Subimage(gray,tl,br));

title(['subimage from (',num2str(tl(1)),',',num2str(tl(2)),') to (',num2str(br(1)),',',num2str(br(2)),')'],'FontSize',8);

subplot(2,2,3)

imshow(color);

title('color image','FontSize',8);

tl=input('Enter top-left corner of the subimage');

br=input('Enter bottom-right corner of the subimage');

subplot(2,2,4)

imshow(Subimage(color,tl,br));

title(['subimage from (',num2str(tl(1)),',',num2str(tl(2)),') to (',num2str(br(1)),',',num2str(br(2)),')'],'FontSize',8);

sgtitle('Subimages','FontSize',7);

%Code for Subimage function:

function sub=Subimage(img,top\_left,bottom\_right)

if any(top\_left>bottom\_right)

ME = MException('Subimage:invalidSizeException','n\_pixels in subimage must be non-zero.');

throw(ME)

end

sub=img(top\_left(1):bottom\_right(1),top\_left(2):bottom\_right(2),:);

end

**OUTPUT:**



1. Answers to question 2
   1. CODE:

gray\_matrix=cast(randi(256,10,10)-1,'uint8');

color\_matrix=cast(randi(256,10,10,3)-1,'uint8');

figure(1)

subplot(2,2,1);

imshow(gray\_matrix);

title('input gray Image','FontSize',8);

subplot(2,2,2);

imshow(ZoomPixelReplication(gray\_matrix));

title('Zoomed Image','FontSize',8);

subplot(2,2,3);

imshow(color\_matrix);

title('input color Image','FontSize',8);

subplot(2,2,4);

imshow(ZoomPixelReplication(color\_matrix));

title('Zoomed Image','FontSize',8);

sgtitle('Pixel Replication','FontSize',8);

%where the ZoomPixelReplication(matrix) is as follows:

function zoom=ZoomPixelReplication(img)

sz=2\*size(img);

if ~ismatrix(img)

sz(end)=sz(end)/2;

end

zoom=zeros(sz,class(img));

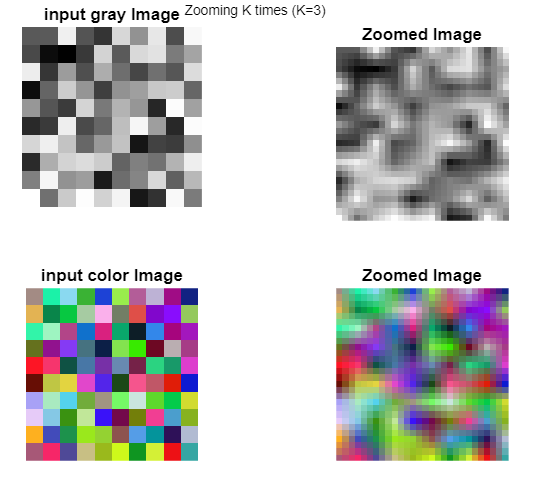
zoom(1:2:end,1:2:end,:)=img;

zoom(1:2:end,2:2:end,:)=img;

zoom(2:2:end,:,:)=zoom(1:2:end,:,:);

end

**OUTPUT:**



* 1. **CODE:**

figure(2)

subplot(2,2,1);

imshow(gray\_matrix);

title('input gray Image','FontSize',8);

subplot(2,2,2);

imshow(ZoomZeroOrderHold(gray\_matrix));

title('Zoomed Image','FontSize',8);

subplot(2,2,3);

imshow(color\_matrix);

title('input color Image');

subplot(2,2,4);

imshow(ZoomZeroOrderHold(color\_matrix));

title('Zoomed Image','FontSize',8);

sgtitle('Zero Order Hold','FontSize',8);

where ZoomZeroOrderHold() is as follows:

function zoom=ZoomZeroOrderHold(img)

sz=(2\*size(img)-1);

if ~ismatrix(img)

sz=[sz(1:end-1) size(img,ndims(img))];

end

zoom=zeros(sz,class(img));

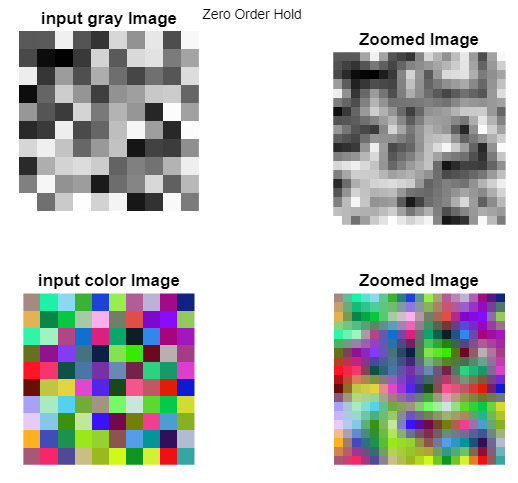
zoom(1:2:end,1:2:end,:)=img;

zoom(1:2:end,2:2:end,:)=floor((double(img(:,1:end-1,:))+double(img(:,2:end,:)))/2);

zoom(2:2:end,:,:)=floor((double(zoom(1:2:end-1,:,:))+double(zoom(3:2:end,:,:)))/2);

end

**OUTPUT:**



* 1. **CODE:**

for k=2:6

figure(k-1)

subplot(2,2,1);

imshow(gray\_matrix);

title(['input gray Image ',num2str(size(gray\_matrix,1)),'X',num2str(size(gray\_matrix,2))],'FontSize',8);

subplot(2,2,2);

zoom=ZoomKTimes(gray\_matrix,k);

imshow(zoom);

title(['Zoomed Image ',num2str(size(zoom,1)),'X',num2str(size(zoom,2))],'FontSize',8);

subplot(2,2,3);

imshow(color\_matrix);

title(['input RGB Image ',num2str(size(color\_matrix,1)),'X',num2str(size(color\_matrix,2))],'FontSize',8);

subplot(2,2,4);

zoom=ZoomKTimes(color\_matrix,k);

imshow(zoom);

title(['Zoomed Image ',num2str(size(zoom,1)),'X',num2str(size(zoom,2))],'FontSize',8);

sgtitle(['Zooming K times (K=',num2str(k),')'],'FontSize',8);

end

%where ZoomKTimes(matrix) is as follows:

function zoom=ZoomKTimes(img,K)

sz=(K\*(size(img)-1)+1);

if ~ismatrix(img)

sz=[sz(1:end-1) size(img,ndims(img))];

end

zoom=zeros(sz,class(img));

zoom(1:K:end,1:K:end,:)=img;

img=cast(img,'double');

cols1=img(:,1:end-1,:);

cols2=img(:,2:end,:);

filter=cols2>=cols1;

for k=1:K-1

temp=(cols1+k\*floor((cols2-cols1)/K)).\*filter;

zoom(1:K:end,1+k:K:end,:)=temp+(cols2+(K-k)\*floor((cols1-cols2)/K)).\*(~filter);

end

cols2=double(zoom(1+K:K:end,:,:));

cols1=double(zoom(1:K:end-1,:,:));

filter=cols2>=cols1;

for k=1:K-1

temp=(cols1+k\*floor((cols2-cols1)/K)).\*filter;

zoom(1+k:K:end,:,:)=temp+(cols2+(K-k)\*floor((cols1-cols2)/K)).\*(~filter);

end

end

**OUTPUTS:**

