

Name: Ananya Nigam	Section: A	Semester: 4
SRN: PES1UG20CS044	Roll number: 47	Date: 1.05.2022

LINEAR ALGEBRA PROJECT

GRAYSCALE IMAGE MANIPULATION

Code:

```

C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\imagemanipulation.m *
imagemanipulation.m * coloured_img_manipulation.m +
1  %: To use the matrix operations to manipulate Grayscale images.
2
3  %% Load a grayscale jpg file and represent the data as a matrix:
4  % 0 corresponds to black and 255 to white: bitmaps
5  ImJPG=imread('aut.jpg'); %loading using imread
6
7  %% checking dimensions of the obtained array ImJPG
8  [m,n]=size(ImJPG)
9
10 %type of the array ImJPG by using the command isinteger
11 %0=false and 1=true
12 isinteger(ImJPG)
13
14 %Finding range of colours in image
15 maxImJPG = max(ImJPG);
16 minImJPG = min(ImJPG);
17
18 % display the image on the screen
19 imshow(ImJPG)
20
21 %% CROP IMG: Select and display a part of the image, paste into a different image
22
23 ImJPG_center = ImJPG(700:m-800,500:n-600);
24 figure;
25
26
27 %% ImJPG_centre is cropped image matrix
28 imshow(ImJPG_center)

```

```

C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\imagemanipulation.m
imagemanipulation.m coloured_img_manipulation.m +
30 %% paste the selected part of the image into another image
31 %first creating a zero matrix
32 ImJPG_border=uint8(zeros(m,n));
33 %paste the preselected matrix ImJPG center into matrix ImJPG border and display the image:
34 ImJPG_border(700:m-800,500:n-600)=ImJPG_center;
35 figure;
36 imshow(ImJPG_border)
37
38
39
40 %% Flip the image horizontally/vertically
41
42 %flipping the image vertically:reverses the order of elements in each column of the matrix
43
44 ImJPG_vertflip=flip(ImJPG);
45 imshow(ImJPG_vertflip)
46
47
48 %% transposing the image matrix is equivalent to rotating the image 90 degrees counterclockwise and flipping it horizontally.
49 ImJPG_transpose=ImJPG';
50 figure;
51 imshow(ImJPG_transpose)
52
53 %% Horizontal images
54 ImJPG_horflip=flip(ImJPG_transpose)';
55 imshow(ImJPG_horflip)
56

```

```

C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\imagemanipulation.m
imagemanipulation.m coloured_img_manipulation.m +
57 %% Rotate the image
58 %rot90 rotates 90 degrees in counterclockwise direction
59 ImJPG90=rot90(ImJPG)
60 figure;
61 imshow(ImJPG90)
62
63
64
65 %% Color inversion
66 %Matlab treats the constant 255 as an array of the same size as ImJPG with all the elements equal to 255.
67 ImJPG_inv = 255-ImJPG;
68 figure;
69 imshow(ImJPG_inv);
70
71 %If out of range values then as array type is uint8,those elements rounded to zero.
72
73 %% darkening: 0 corresponds to black
74 ImJPG_dark=ImJPG-50;
75 imshow(ImJPG_dark);
76
77 %% lightening: 255 corresponds to white
78 ImJPG_light=ImJPG+70;
79 imshow(ImJPG_light);
80
81 %% A naive conversion to black and white
82 %Making grayer shades black
83 %Making the shades lighter than medium gray to white
84 %uint8 to convert to integer format

```

```
C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\imagemanipulation.m *
imagemanipulation.m * coloured_img_manipulation.m +
78     ImJPG_light=ImJPG+70;
79     imshow(ImJPG_light);
80
81     %% A naive conversion to black and white
82     %Making grayer shades black
83     %Making the shades lighter than medium gray to white
84     %uint8 to convert to integer format
85
86     ImJPG_bw=uint8(255*floor(ImJPG/128));
87     figure
88     imshow(ImJPG_bw);
89
90     %% Increasing Contrast
91     %we can increase the difference between the colors of the image, for instance,
92     % by making the dark shades of gray to appear even darker.
93     % scalar multiply the matrix by some constant.
94     % positive constant greater than 1, then the contrast is increased
95     % constant less than 1, then it is decreased
96     ImJPG_HighContrast=uint8(1.25*ImJPG);
97     imshow(ImJPG_HighContrast)
98
99     %% Decreasing contrast
100     ImJPG_LowContrast=uint8(0.25*ImJPG);
101     imshow(ImJPG_LowContrast)
102
```

CODE FOR COLOURED IMAGE MANIPULATION:

```
Editor - C:\Users\anany\OneDrive\Documents\MATLAB
imagemanipulation.m  coloured_img_manipulation.m  +
1  %% Loading the image
2
3  I=imread('car.jpg');
4  % Checking dimensions
5  [m,n]=size(I)
6
7  imshow(I)
8
9  %% Crop image
10
11  targetSize = [300 600];
12  %Create a Rectangle object that specifies the spatial extent of the crop window.
13
14  r = centerCropWindow2d(size(I),targetSize);
15  %Crop the image to the spatial extents. Display the cropped region
16  J = imcrop(I,r);
17  h=[];
18  h(1) = subplot(2,2,1);
19  h(2) = subplot(2,2,2);
20  image(I,'Parent',h(1));
21  image(J,'Parent',h(2));
22
23
24
25  %% Flipping image
26
27  I_vertflip=flip(I);
28  imshow(I_vertflip)
```

```
C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\coloured_img_manipulation.m
imagemanipulation.m coloured_img_manipulation.m
28 imshow(I_vertflip)
29 h(3) = subplot(2,2,3);
30 image(I_vertflip, 'Parent', h(3));
31
32
33 %% Taking transpose
34
35 I_trans=permute(I,[2 1 3])
36
37 imshow(I_trans)
38
39
40 %% Rotating by 90 degrees
41
42 ImJPG90=rot90(I)
43 figure;
44 imshow(ImJPG90)
45
46 %% Converting to black and white
47 gray = 0.2989 * I(:, :, 1) + 0.5870 * I(:, :, 2) + 0.1140 * I(:, :, 3);
48 imshow(gray)
49
50 %% Adjusting contrast image
51 RGB2 = imadjust(I,[.2 .3 0; .6 .7 1],[]);
52 figure
53 imshow(RGB2)
54
55 %% Lightening the image
56
57 ImJPG_light=I+70;
58 imshow(ImJPG_light)
```

```
C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\coloured_img_manipulation.m
imagemanipulation.m coloured_img_manipulation.m
60 %% Darkening the image
61 ImJPG_dark=I-100;
62 imshow(ImJPG_dark)
63
64
65 %% Read in original RGB image.
66 rgbImage = imread('car.jpg');
67 % Extract color channels.
68 redChannel = rgbImage(:,:,1); % Red channel
69 greenChannel = rgbImage(:,:,2); % Green channel
70 blueChannel = rgbImage(:,:,3); % Blue channel
71 % Create an all black channel.
72 allBlack = zeros(size(rgbImage, 1), size(rgbImage, 2), 'uint8');
73 % Create color versions of the individual color channels.
74 just_red = cat(3, redChannel, allBlack, allBlack);
75 just_green = cat(3, allBlack, greenChannel, allBlack);
76 just_blue = cat(3, allBlack, allBlack, blueChannel);
77 % Recombine the individual color channels to create the original RGB image again.
78 recombinedRGBImage = cat(3, redChannel, greenChannel, blueChannel);
79 % Display them all.
80 subplot(3, 3, 2);
81 imshow(rgbImage);
82 fontSize = 20;
83 title('Original RGB Image', 'FontSize', fontSize)
84 subplot(3, 3, 4);
85 imshow(just_red);
86 title('Red Channel in Red', 'FontSize', fontSize)
87 subplot(3, 3, 5);
88 imshow(just_green)
89 title('Green Channel in Green', 'FontSize', fontSize)
90 subplot(3, 3, 6);
```

```
C:\Users\anany\OneDrive\Documents\MATLAB
Editor - C:\Users\anany\OneDrive\Documents\MATLAB\coloured_img_manipulation.m
imagemanipulation.m coloured_img_manipulation.m +
75 just_green = cat(3, allBlack, greenChannel, allBlack);
76 just_blue = cat(3, allBlack, allBlack, blueChannel);
77 % Recombine the individual color channels to create the original RGB image again.
78 recombinedRGBImage = cat(3, redChannel, greenChannel, blueChannel);
79 % Display them all.
80 subplot(3, 3, 2);
81 imshow(rgbImage);
82 fontSize = 20;
83 title('Original RGB Image', 'FontSize', fontSize)
84 subplot(3, 3, 4);
85 imshow(just_red);
86 title('Red Channel in Red', 'FontSize', fontSize)
87 subplot(3, 3, 5);
88 imshow(just_green)
89 title('Green Channel in Green', 'FontSize', fontSize)
90 subplot(3, 3, 6);
91 imshow(just_blue);
92 title('Blue Channel in Blue', 'FontSize', fontSize)
93 subplot(3, 3, 8);
94 imshow(recombinedRGBImage);
95 title('Recombined to Form Original RGB Image Again', 'FontSize', fontSize)
96 % Set up figure properties:
97 % Enlarge figure to full screen.
98 set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0, 1, 1]);
99 % Get rid of tool bar and pulldown menus that are along top of figure.
100 % set(gcf, 'Toolbar', 'none', 'Menu', 'none');
101 % Give a name to the title bar.
102 set(gcf, 'Name', 'After Applying matrix operations', 'NumberTitle', 'Off')
103
```

MATRIX OPERATIONS & IMAGE MANIPULATIONS !!

NAME: Ananya Nigam

SRN: PES1UG20CS044

Section : A

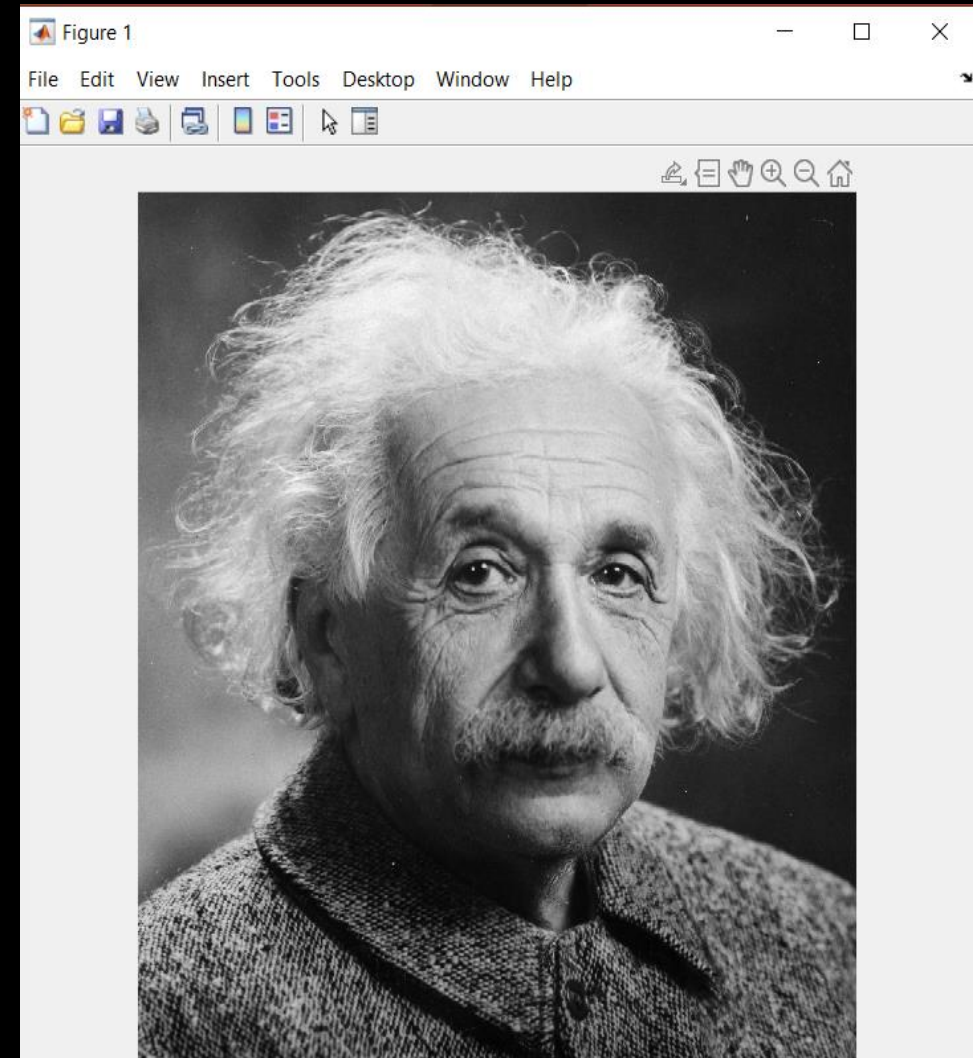
Roll number : 47

INTRODUCTION

- An image in a computer memory can be stored as a matrix with each element of the matrix representing a pixel of the image and containing a number which corresponds to the color of this pixel.
- If the image is a color image, then each pixel is characterized by three numbers corresponding to the intensities of Red, Green, and Blue (the so-called RGB color system). If the image is a grayscale image, then only one number for the intensity of gray is needed.
- The intensity of each color typically ranges from 0 (black) to 255 (white).
- I have used Matlab to build this project.

LOADING A GRAYSCALE IMAGE AND REPRESENTING IT AS A MATRIX

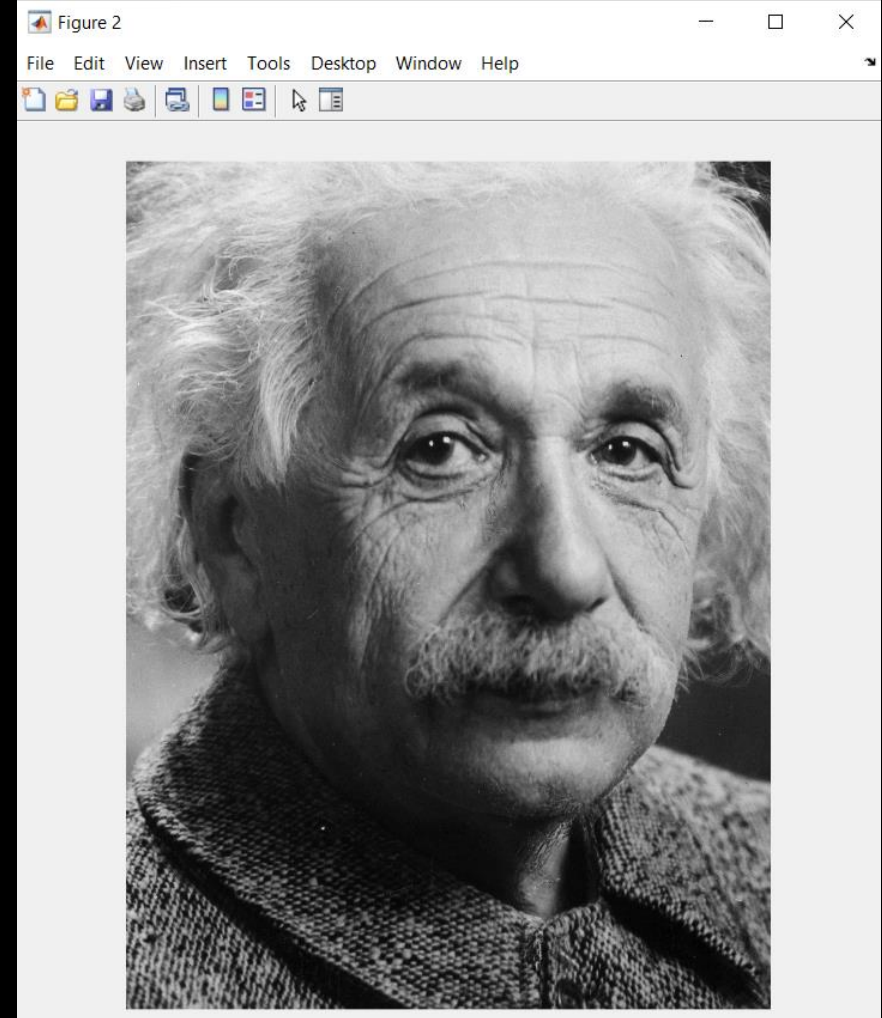
```
%% Load a grayscale jpg file and represent the data as a matrix:  
% 0 corresponds to black and 255 to white: bitmaps  
ImJPG=imread('einstein.jpg'); %loading using imread  
  
%% checking dimensions of the obtained array ImJPG  
[m,n]=size(ImJPG)  
  
%type of the array ImJPG by using the command isinteger  
%0=false and 1=true  
isinteger(ImJPG)  
  
%Finding range of colours in image  
maxImJPG = max(ImJPG);  
minImJPG = min(ImJPG);  
% display the image on the screen  
imshow(ImJPG)
```



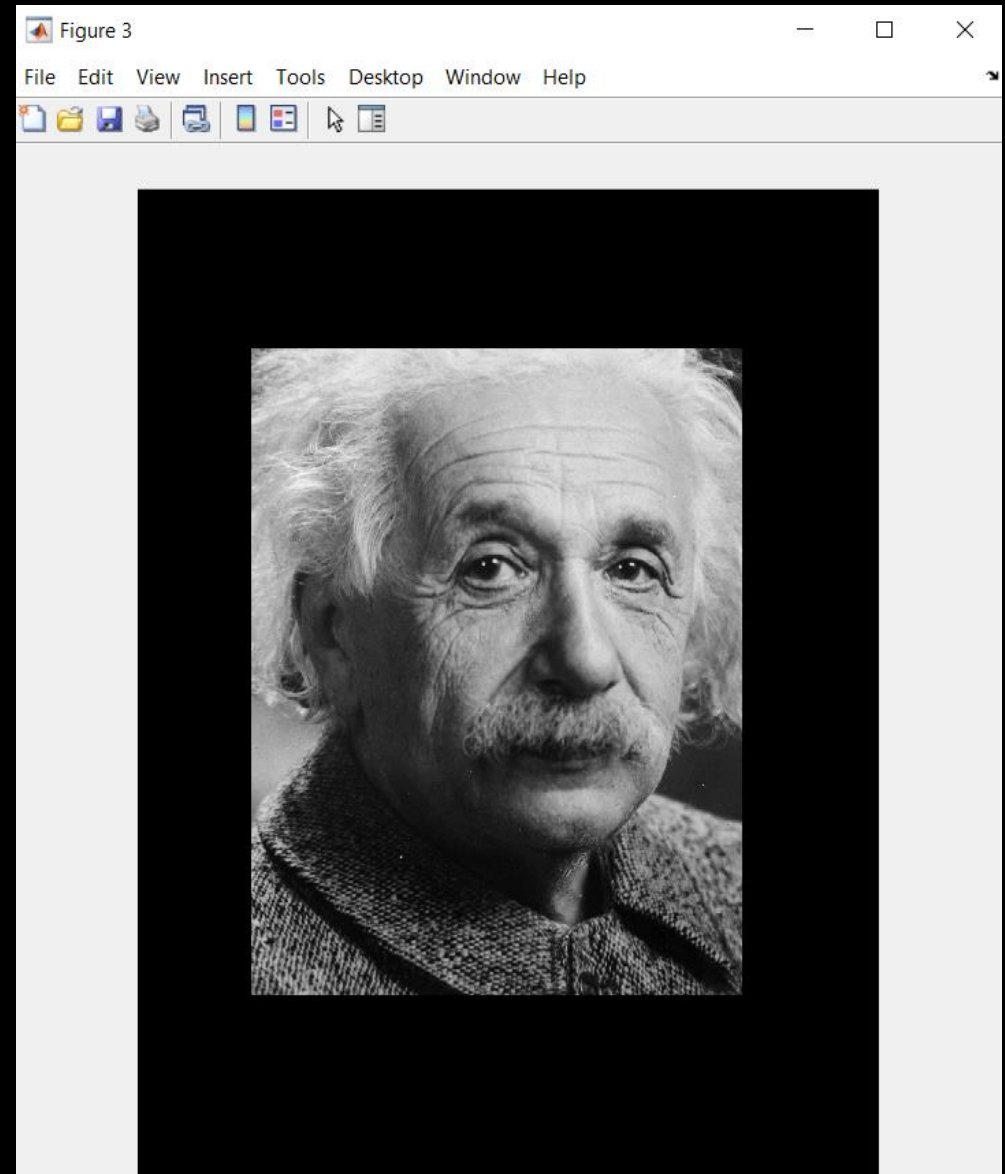
CROPPING IMAGE

```
%% CROP IMG: Select and display a part of the image, paste into a different image

ImJPG_center = ImJPG(700:m-800,500:n-600);%top and bottom,left,right
figure;
%ImJPG_center is cropped image matrix
imshow(ImJPG_center)
%paste the selected part of the image into another image
%first creating a zero matrix
ImJPG_border=uint8(zeros(m,n));
%paste the preselected matrix ImJPG center into matrix ImJPG border and display the image:
ImJPG_border(700:m-800,500:n-600)=ImJPG_center;
figure;
imshow(ImJPG_border)
```



- Cropping the image and displaying it with borders

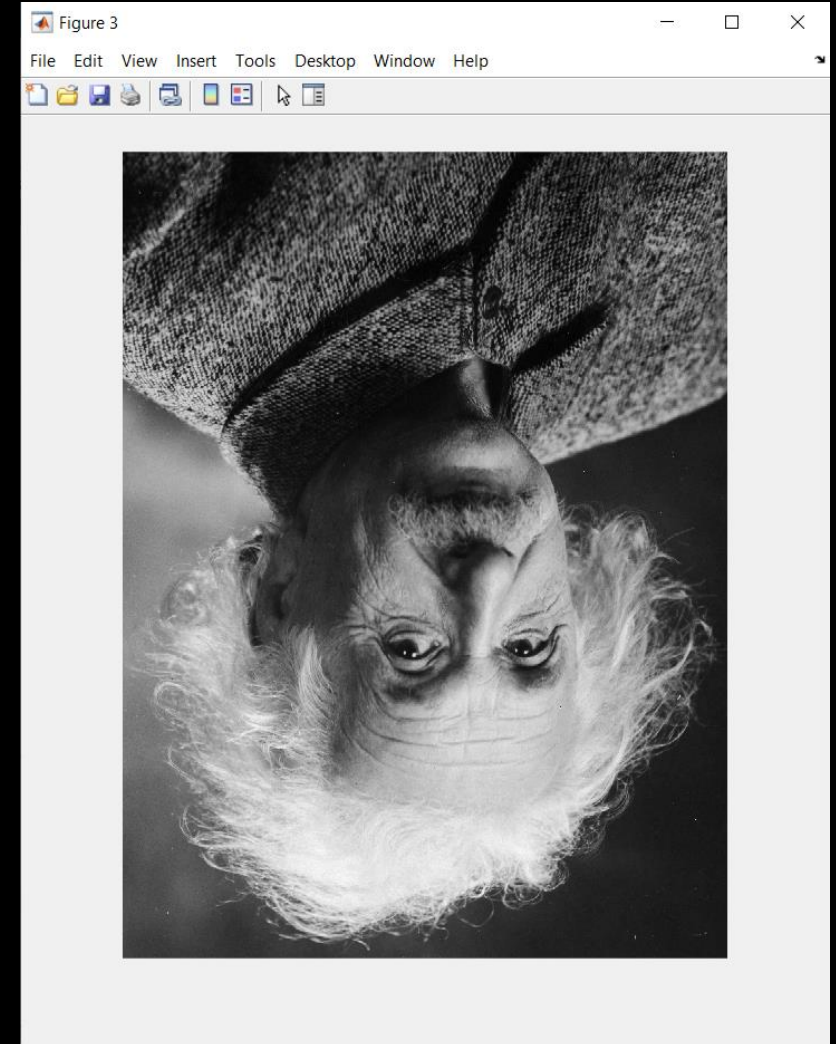


FLIPPING THE IMAGE VERTICALLY

```
%% Flip the image horizontally/vertically

%flipping the image vertically:reverses the order of elements in each column of the matrix

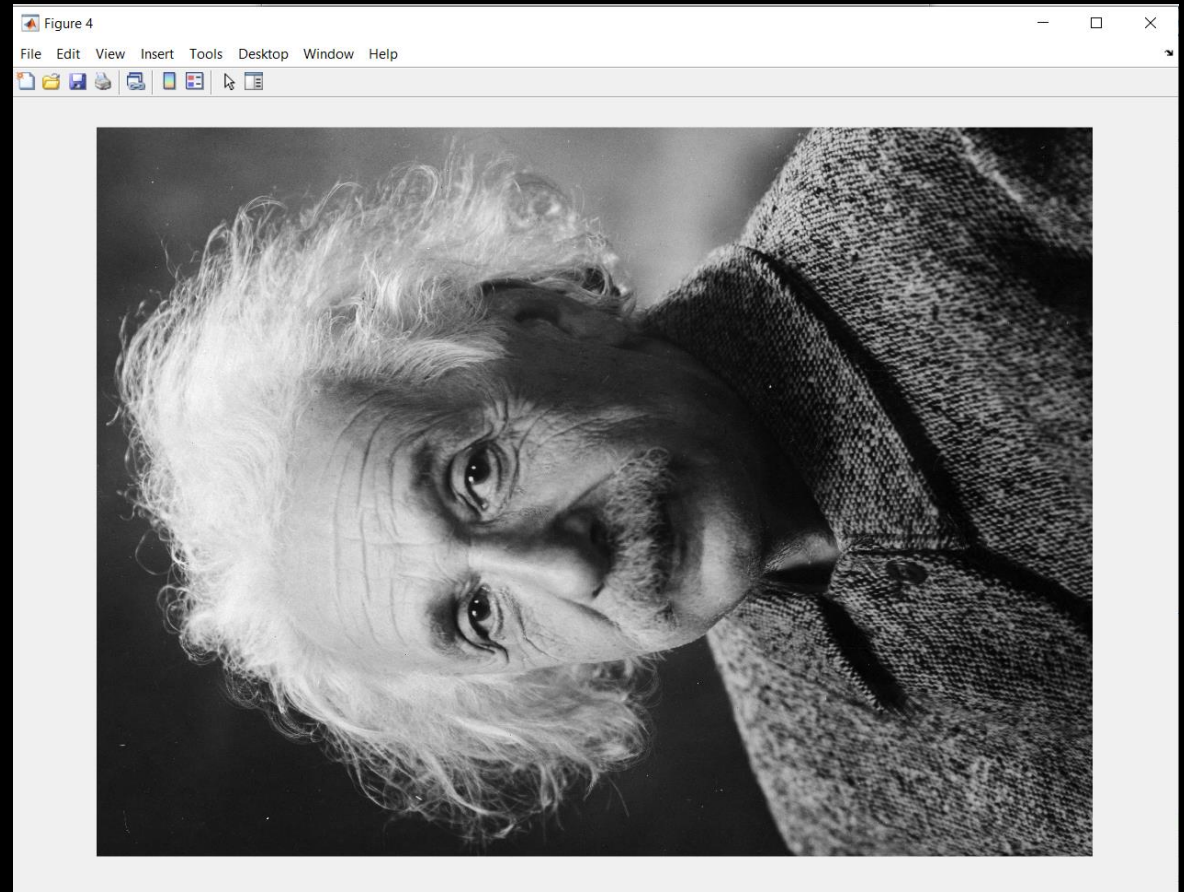
ImJPG_vertflip=flip(ImJPG);
imshow(ImJPG_vertflip)
```



FLIPPING THE IMAGE HORIZONTALLY

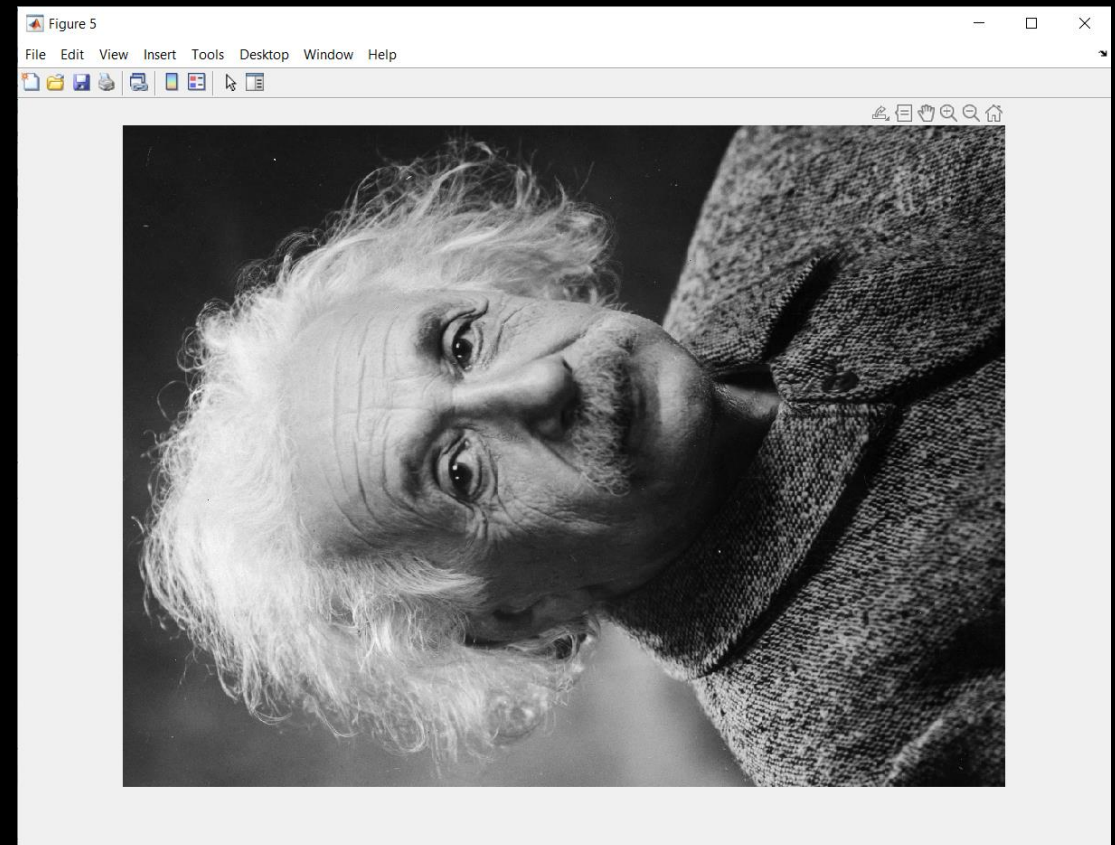
Transposing the image matrix is equivalent to rotating the image 90 degrees counterclockwise and flipping it horizontally

```
ImJPG_transpose=ImJPG';  
figure;  
imshow(ImJPG_transpose)
```



ROTATING THE IMAGE BY 90 DEGREES ANTI CLOCKWISE

```
%% Rotate the image
%rot90 rotates 90 degrees in counterclockwise direction
ImJPG90=rot90(ImJPG)
figure;
imshow(ImJPG90)
```



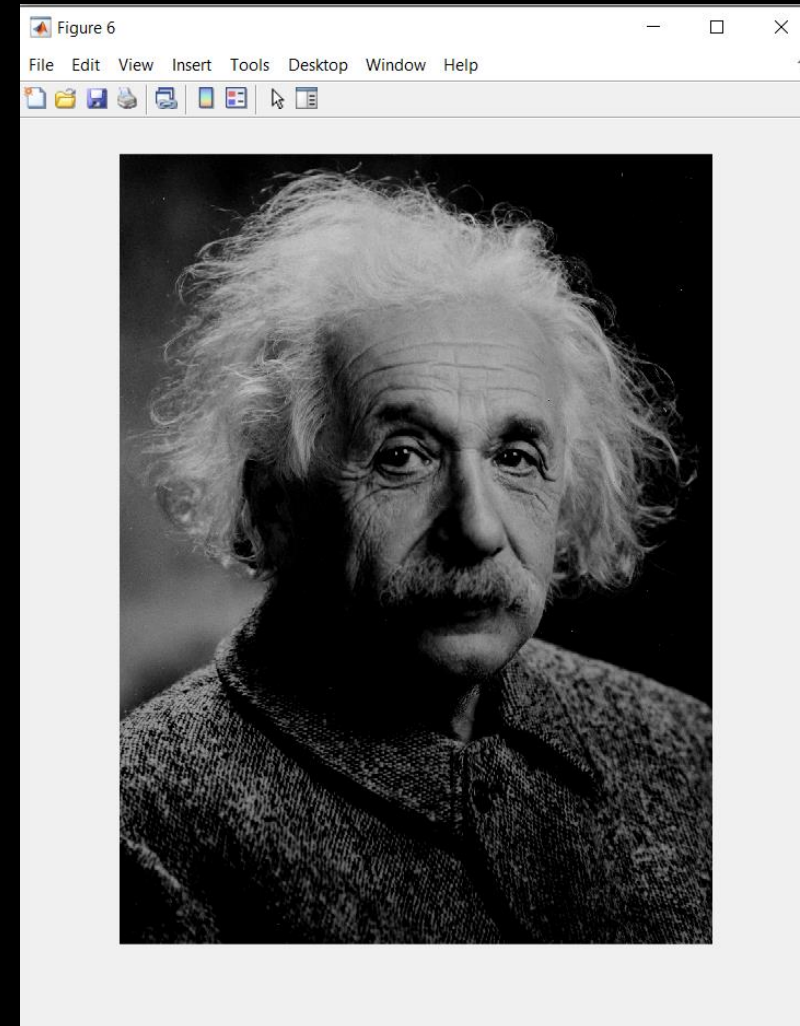
COLOR INVERSION

```
%% Color inversion  
%Matlab treats the constant 255 as an array of the same size as ImJPG with all the elements equal to 255.  
ImJPG_inv = 255-ImJPG;  
figure;  
imshow(ImJPG_inv);
```



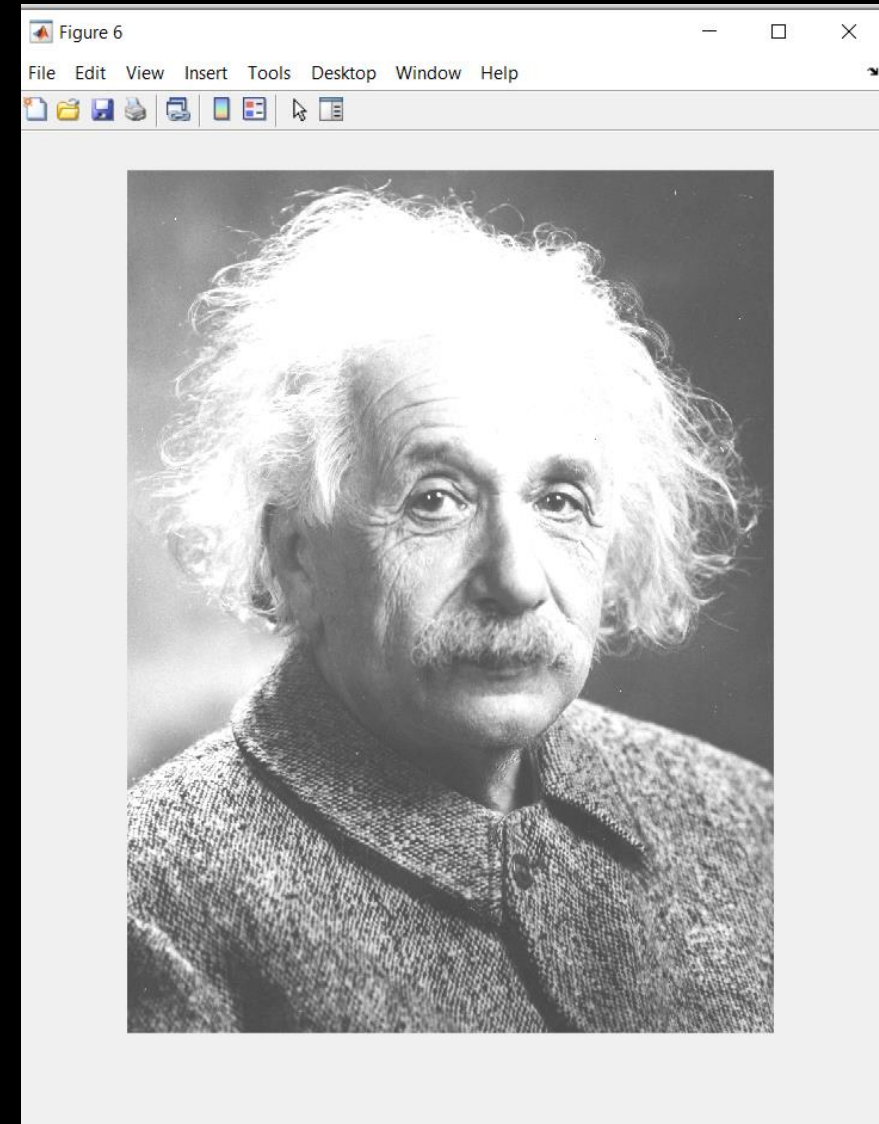
DARKENING THE IMAGE

```
%darkening: 0 corresponds to black  
ImJPG_dark=ImJPG-50;  
imshow(ImJPG_dark);
```



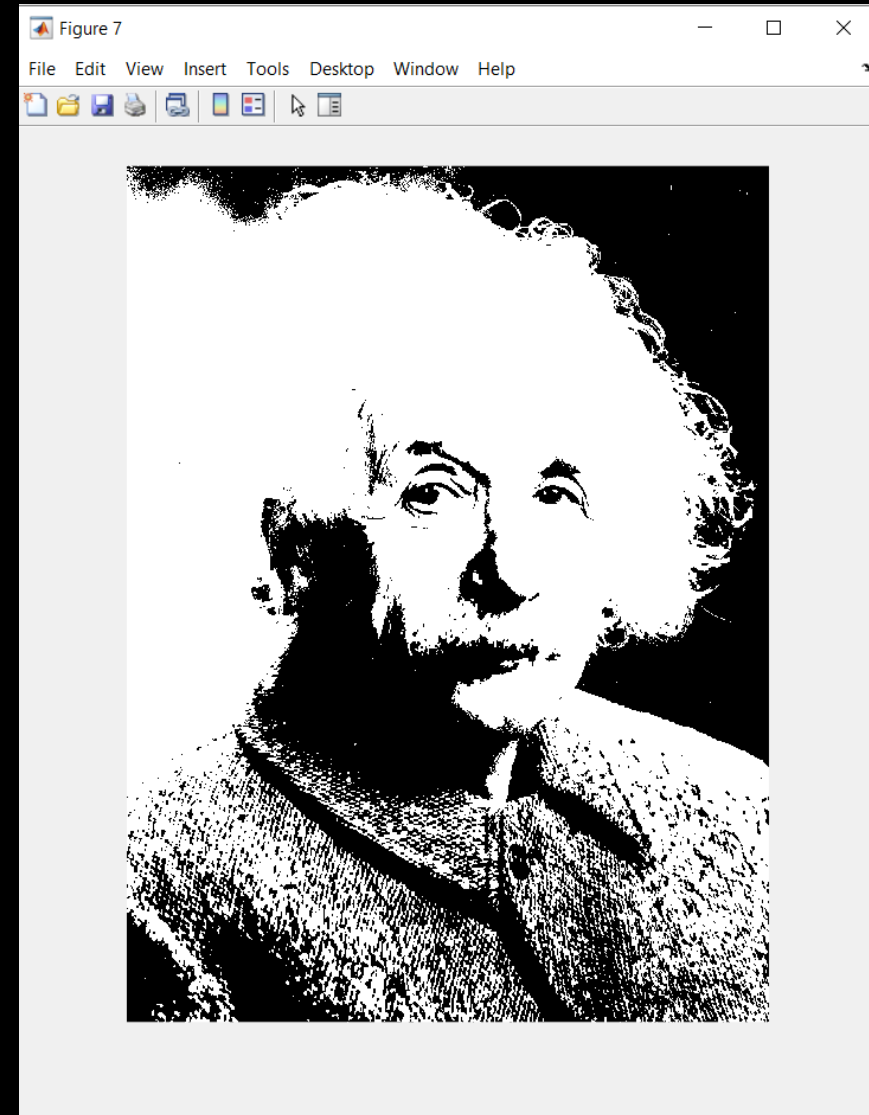
LIGHTENING THE IMAGE

```
%% lightening: 255 corresponds to white  
ImJPG_light=ImJPG+70;  
imshow(ImJPG_light);
```



CONVERTING IMAGE TO BLACK AND WHITE

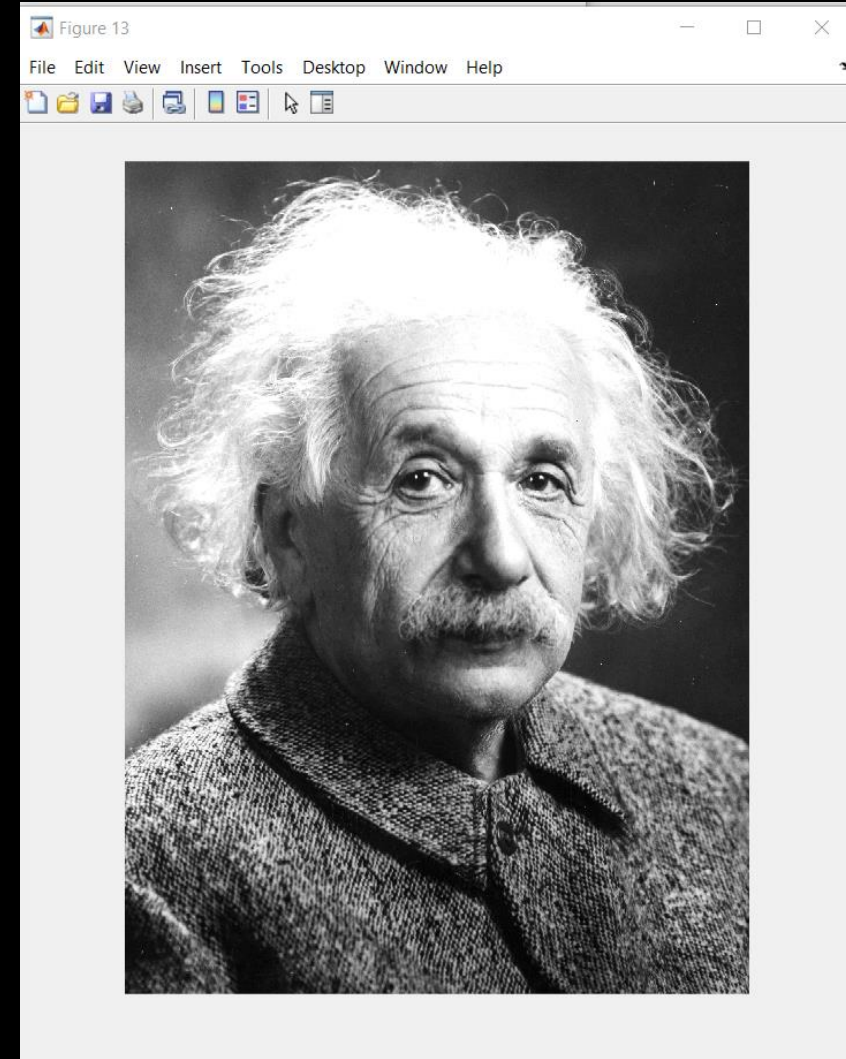
```
%% A naive conversion to black and white  
%Making grayer shades black  
%Making the shades lighter than medium gray to white  
%uint8 to convert to integer format  
  
ImJPG_bw=uint8(255*floor(ImJPG/128));  
figure  
imshow(ImJPG_bw);
```



INCREASING COLOR CONTRAST

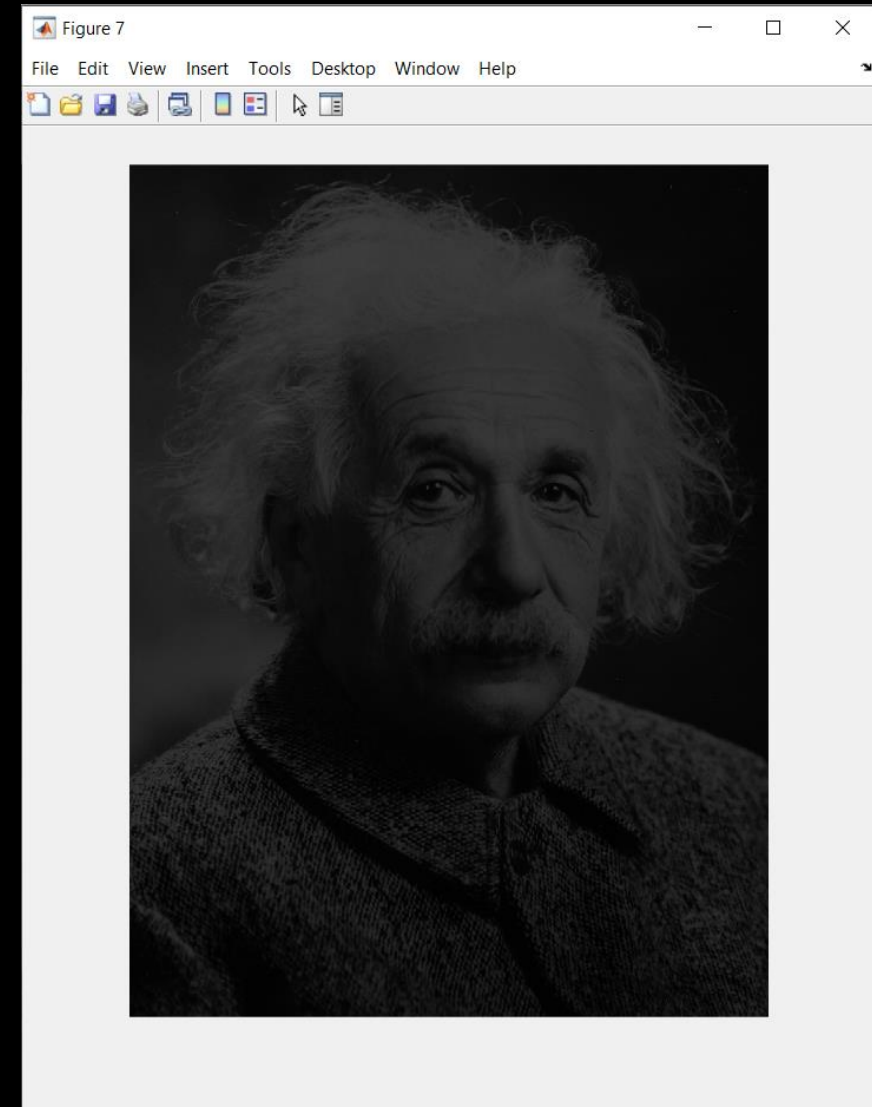
%% Increasing Contrast

```
%we can increase the difference between the colors of the image, f  
% scalar multiply the matrix by some constant.  
% positive constant greater than 1, then the contrast is increased  
% constant less than 1, then it is decreased  
ImJPG_HighContrast=uint8(1.25*ImJPG);  
imshow(ImJPG_HighContrast)|
```



DECREASING COLOUR CONTRAST

```
ImJPG_LowContrast=uint8(0.25*ImJPG);  
imshow(ImJPG_LowContrast)|
```



Thank
you



IMAGE MANIPULATIONS USING MATRIX OPERATIONS ON COLOURED IMAGES

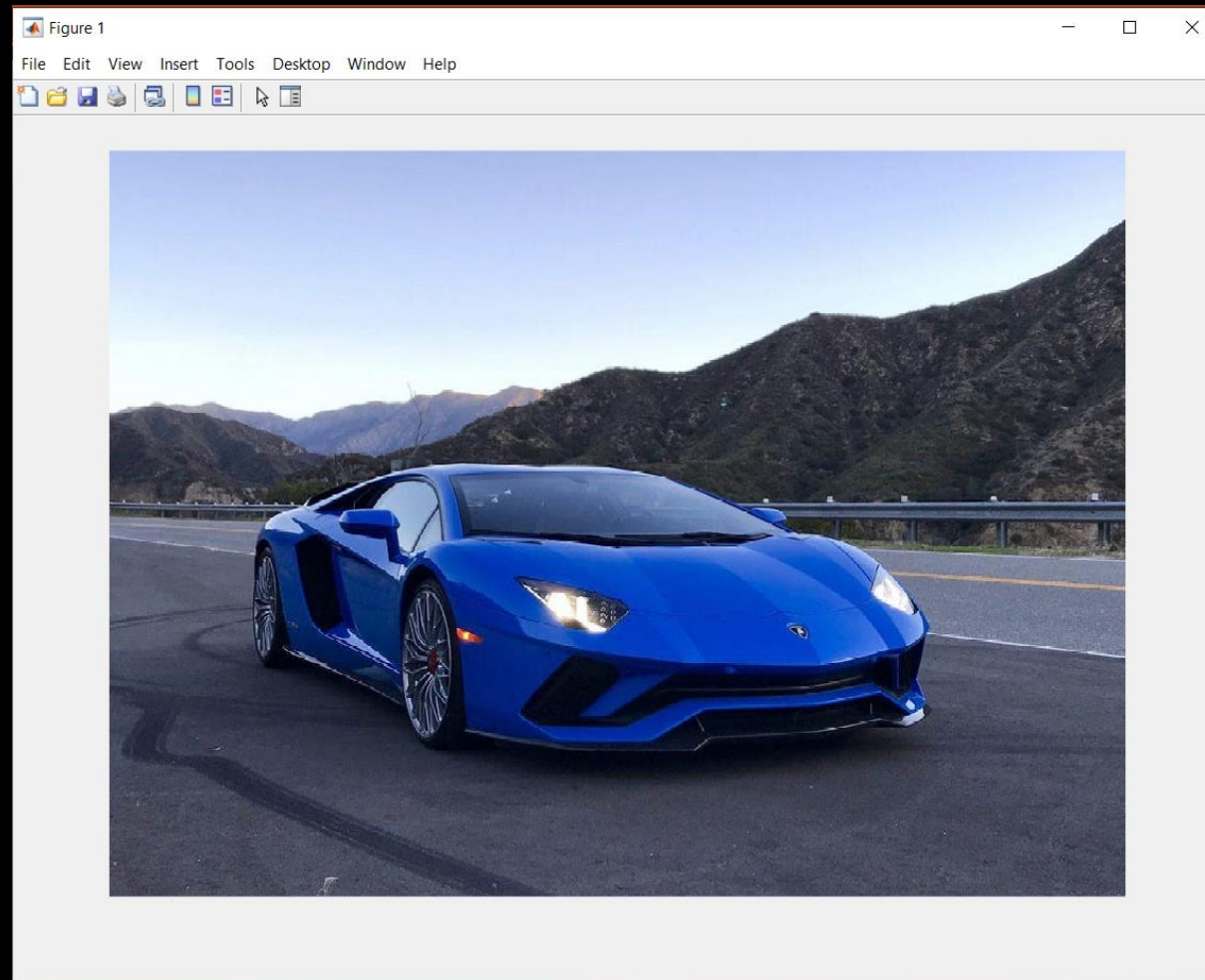
Name: Ananya Nigam

SRN: PES1UG20CS044

Section: A

Roll number: 47

ORIGINAL IMAGE :

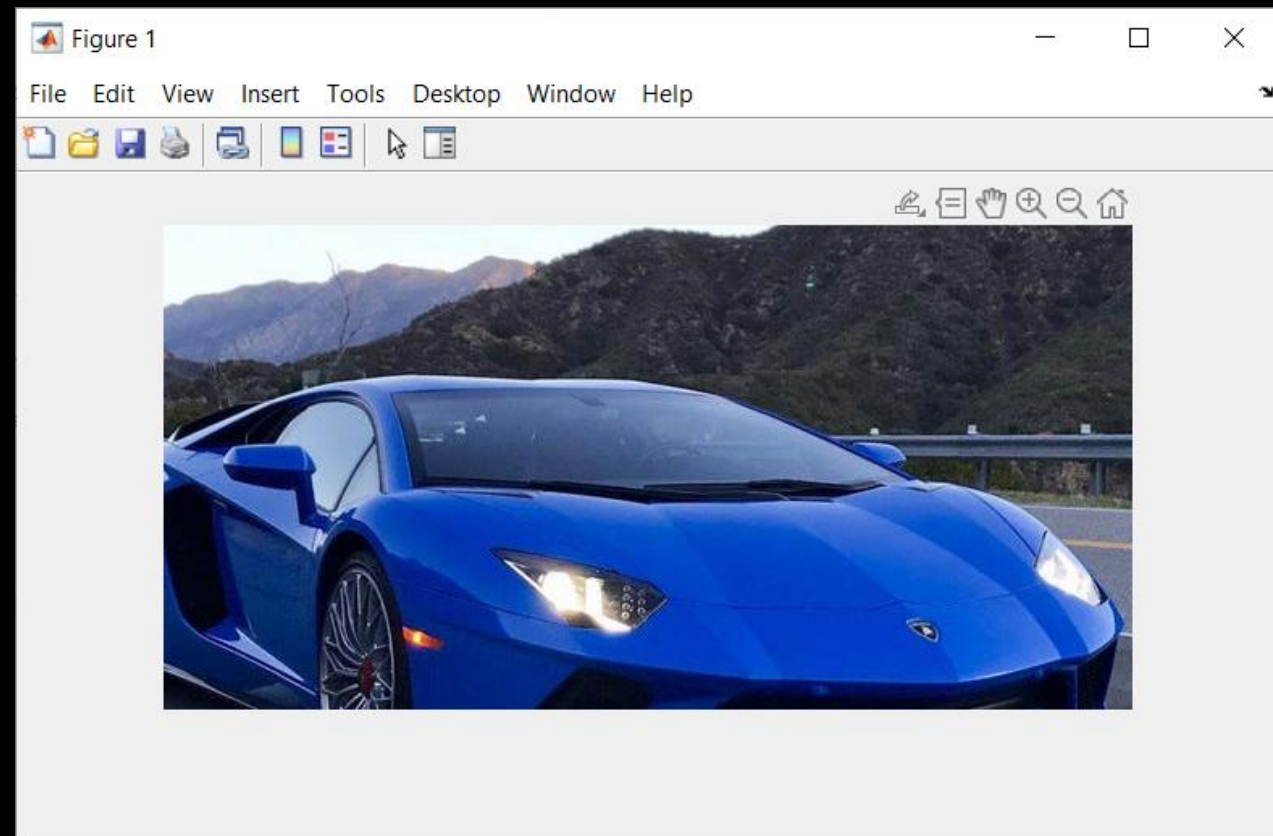


CROPPED IMAGE

```
%% Crop image

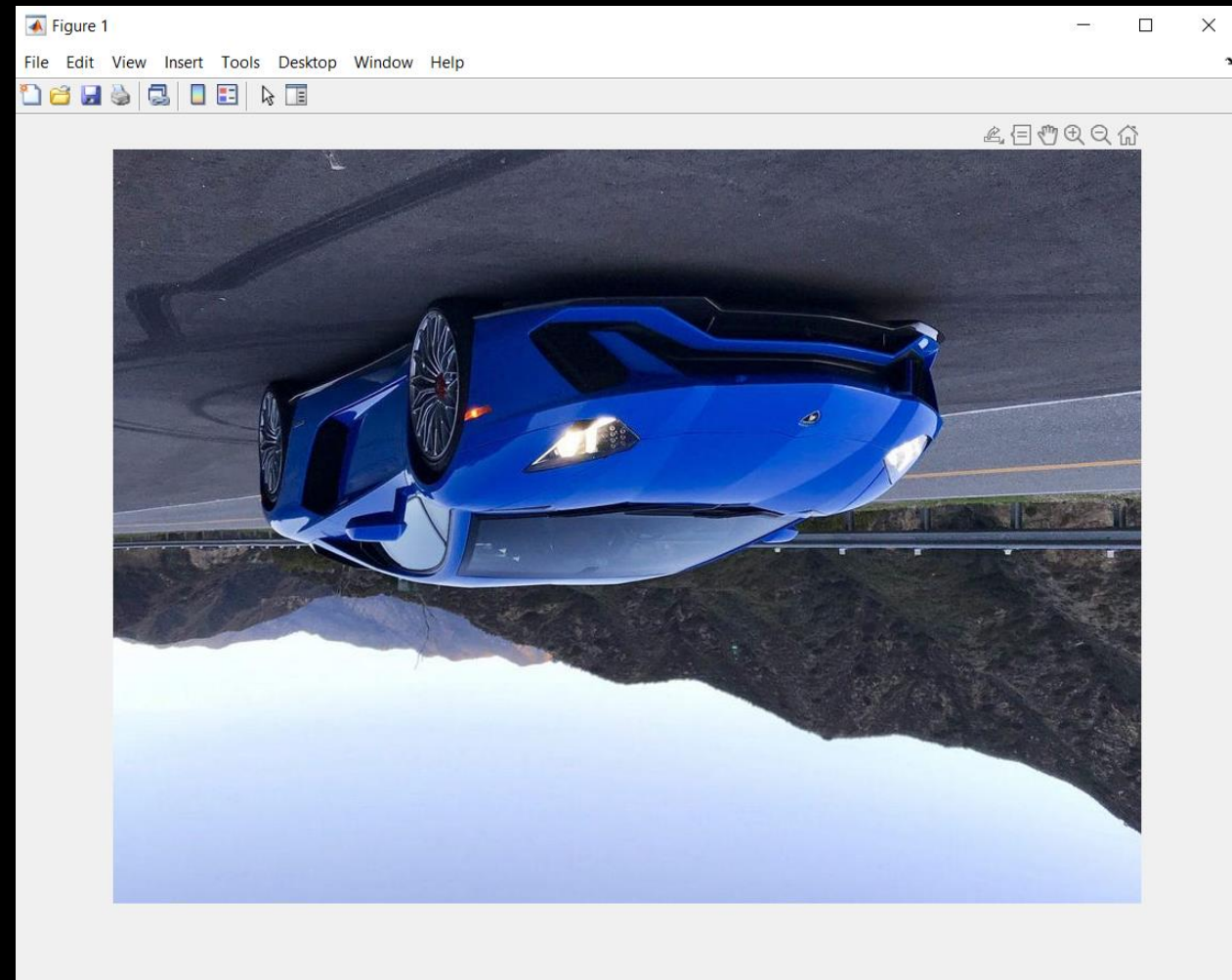
targetSize = [300 600];
%Create a Rectangle object that specifies the spatial extent of the crop window.

r = centerCropWindow2d(size(I),targetSize);
%Crop the image to the spatial extents. Display the cropped region
J = imcrop(I,r);
imshow(J)
```



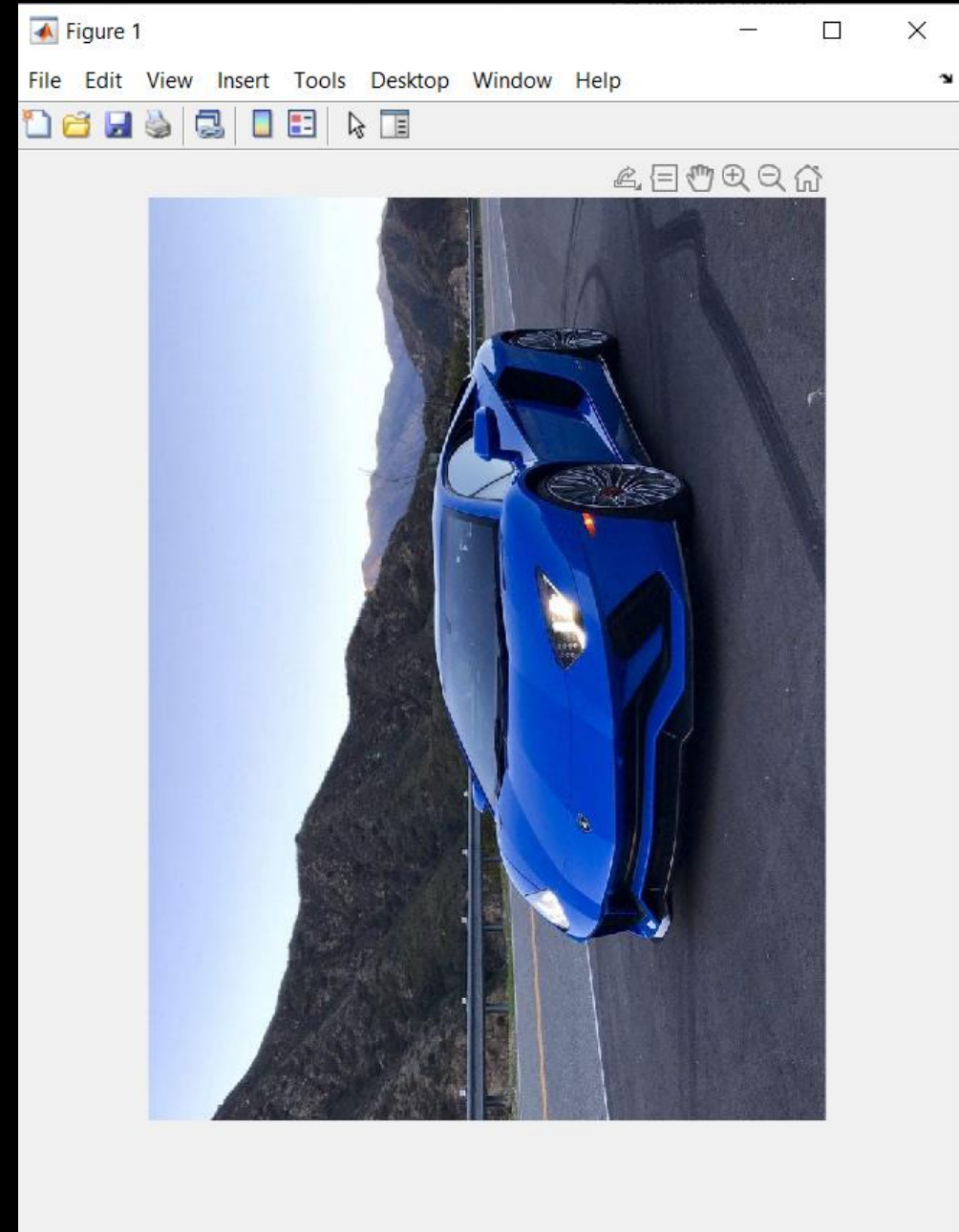
VERTICALLY FLIPPED IMAGE

```
%% Flipping image  
I_vertflip=flip(I);  
imshow(I_vertflip)
```



FLIPPING IMAGE HORIZONTALLY

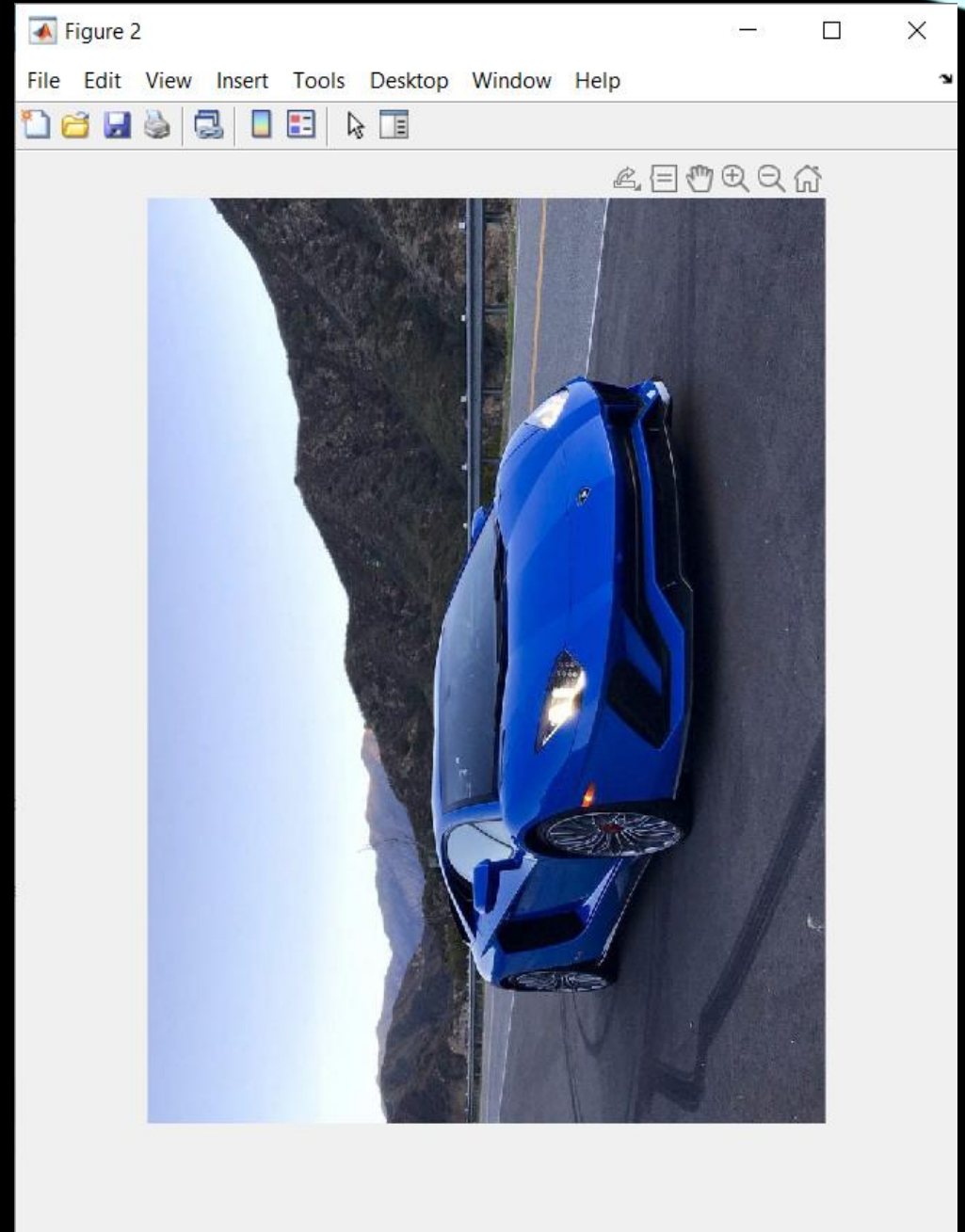
```
I_trans=permute(I,[2 1 3])  
imshow(I_trans)
```



ROTATING 90 DEGREES

%% Rotating by 90 degrees

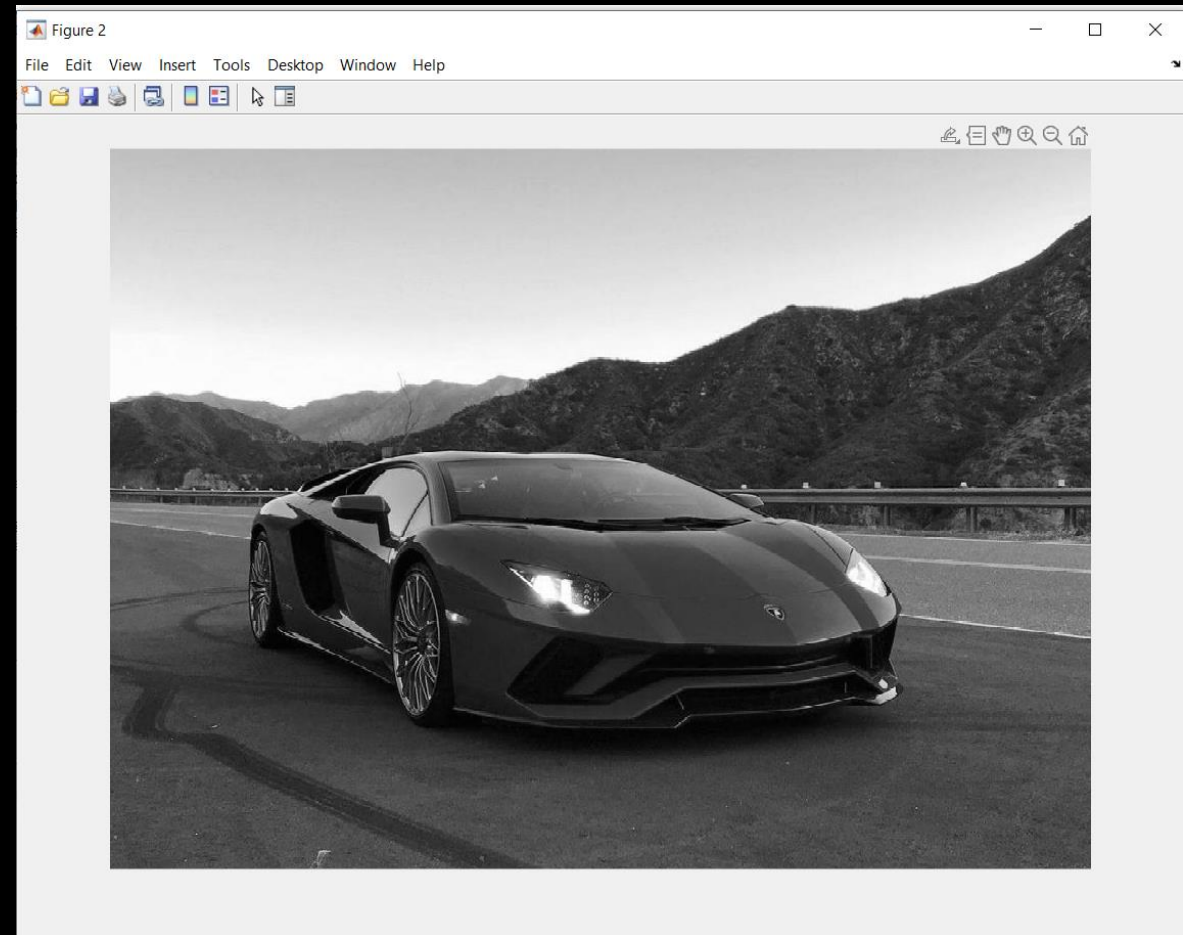
```
ImJPG90=rot90(I)  
figure;  
imshow(ImJPG90)
```



CONVERTING IMAGE TO BLACK AND WHITE

Using Luminicence formula :

```
%% Converting to black and white  
gray = 0.2989 * I(:,:,1) + 0.5870 * I(:,:,2) + 0.1140 * I(:,:,3);  
imshow(gray)
```



SHOWING RGB CHANNELS:

RED CHANNEL

Red Channel in Red



GREEN CHANNEL

Green Channel in Green



BLUE CHANNEL

Blue Channel in Blue



Original RGB Image



Red Channel in Red



Green Channel in Green



Blue Channel in Blue



Recombined to Form Original RGB Image Again



ADJUSTING CONTRAST OF IMAGE

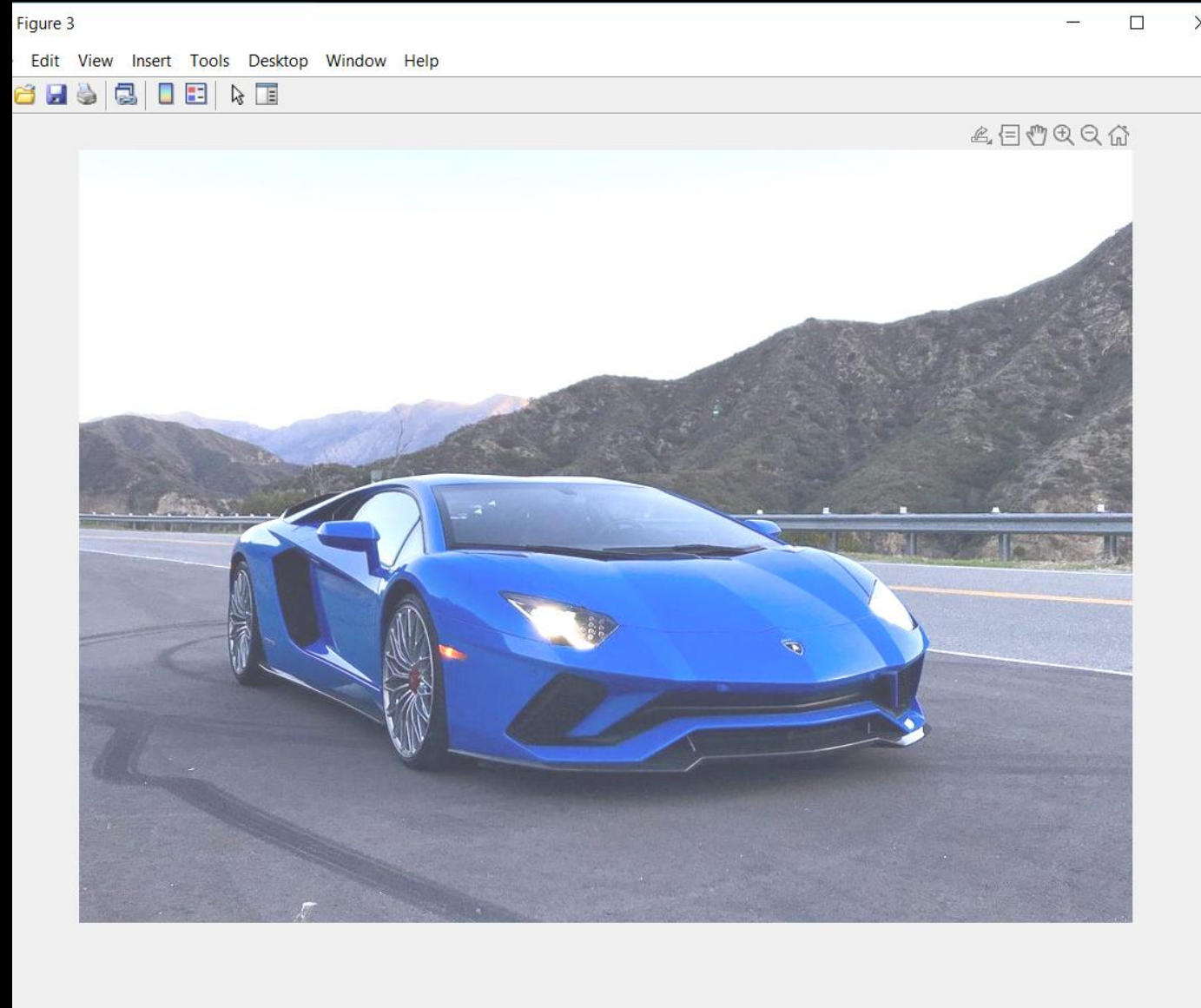
```
%% Adjusting contrast image  
RGB2 = imadjust(I,[.2 .3 0; .6 .7 1],[]);  
figure  
imshow(RGB2)
```



LIGHTENING THE IMAGE

```
%% Lightening the image
```

```
ImJPG_light=I+70;  
imshow(ImJPG_light)
```

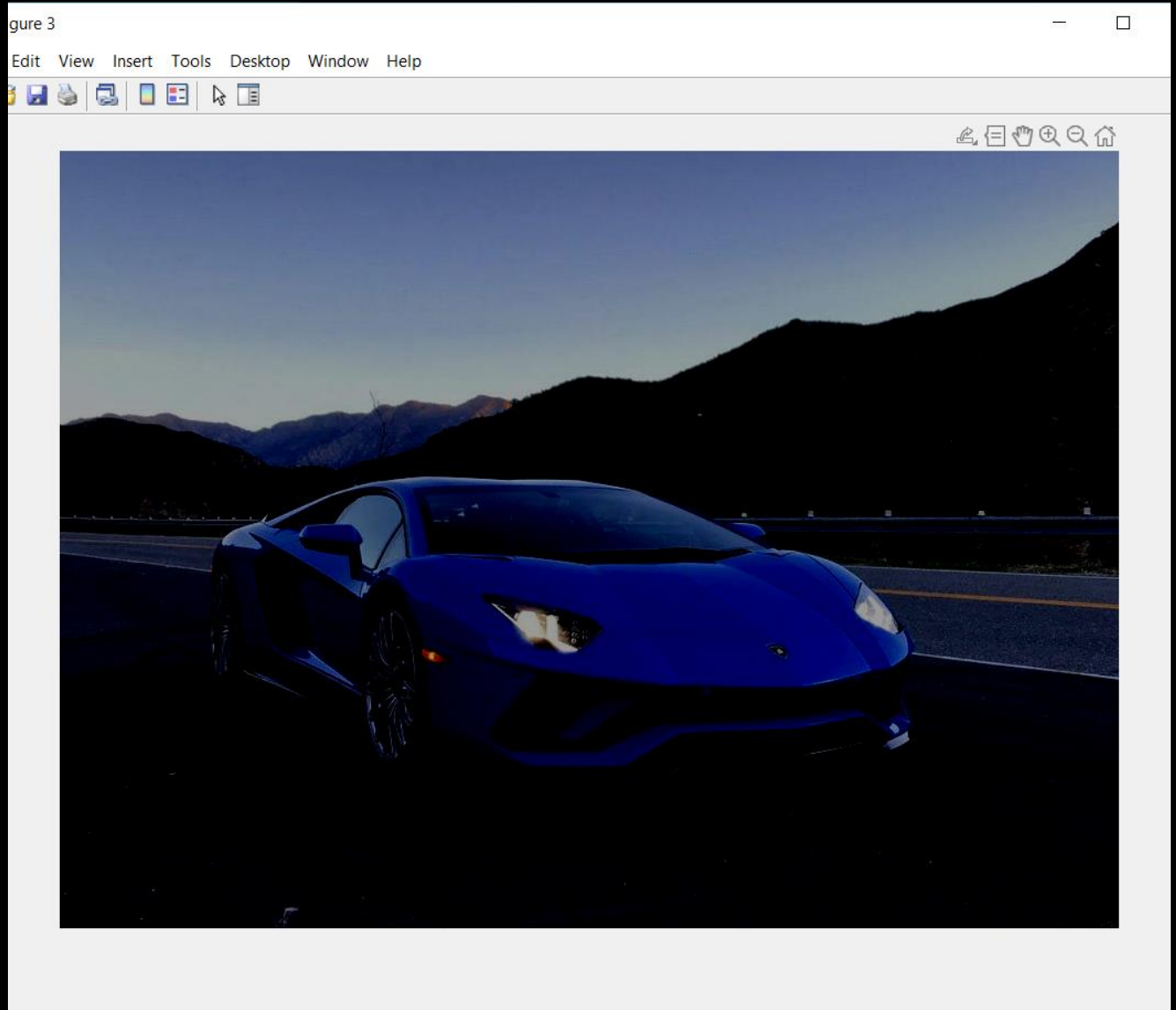


DARKENING THE IMAGE

```
%% Darkening the image
```

```
ImJPG_dark=I-100;
```

```
imshow(ImJPG_dark)
```





THANK YOU