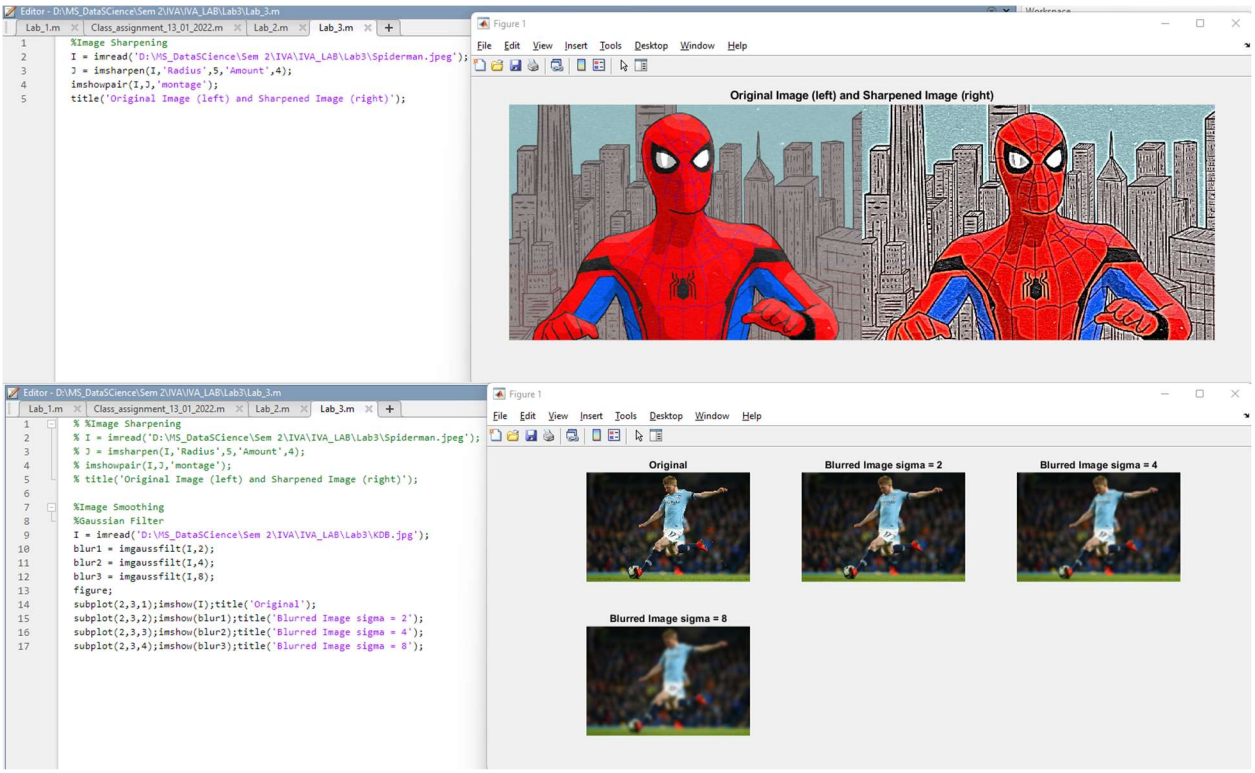
1. Program to demonstrate Spatial filtering using Built-in and user defined functions with different size of kernels. (Smoothing, Sharpening)

CODE(INBUILT):

```
%Image Sharpening
I = imread(Spiderman.jpeg');
J = imsharpen(I,'Radius',5,'Amount',4);
imshowpair(I,J,'montage');
title('Original Image (left) and Sharpened Image (right)');

%Image Smoothing
%Gaussian Filter
I = imread('KDB.jpg');
blur1 = imgaussfilt(I,2);
blur2 = imgaussfilt(I,4);
blur3 = imgaussfilt(I,8);
figure;
subplot(2,3,1);imshow(I);title('Original');
subplot(2,3,2);imshow(blur1);title('Blurred Image sigma = 2');
subplot(2,3,3);imshow(blur2);title('Blurred Image sigma = 4');
subplot(2,3,4);imshow(blur3);title('Blurred Image sigma = 8');
```

OUTPUT:



CODE(USER DEFINED):

```
%Image sharpening.
% Read the image in variable 'a'
a=imread("cameraman.tif");
% Defined the laplacian filter.
Lap=[0 1 0; 1 -4 1; 0 1 0];
% Convolve the image read
% in 'a' with Laplacian mask.
a1=conv2(a,Lap,'same');
% After convolution the intensity
% Values go beyond the range.
% Normalise the range of intensity.
a2=uint8(a1);
% Display the sharpened image.
imtool(abs(a-a2),[])
% Define strong laplacian filter
lap=[-1 -1 -1; -1 8 -1; -1 -1 -1];
% Apply filter on original image
a3=conv2(a,lap,'same');
% Normalise the resultant image.
a4=uint8(a3);
% Display the sharpened image.
imtool(abs(a+a4),[])
clear cam;
```

```
%Image Smoothening
% I = imread('D:\MS_DataSCience\Sem 2\IVA\IVA_LAB\Lab4\captain.jpg');
% grayImage = rgb2gray(I);
% subplot(1,2,1);
% imshow(grayImage);
% title('Original Image', 'FontSize', 15);
%
% windowSize = 16;
% kernel = ones(windowSize, windowSize) / windowSize ^ 2;
% filtimage= imfilter(grayImage, kernel, 'symmetric');
%
% subplot(1,2,2);
% imshow(filtimage);
% title('Blurred Image', 'FontSize', 15);
```

OUTPUT:

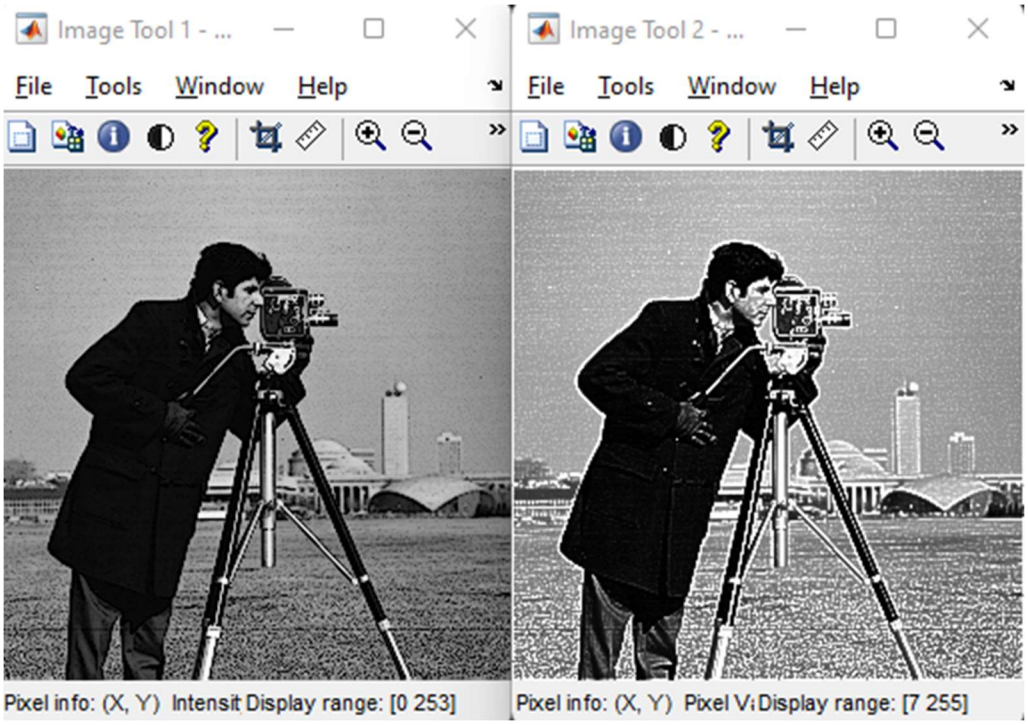
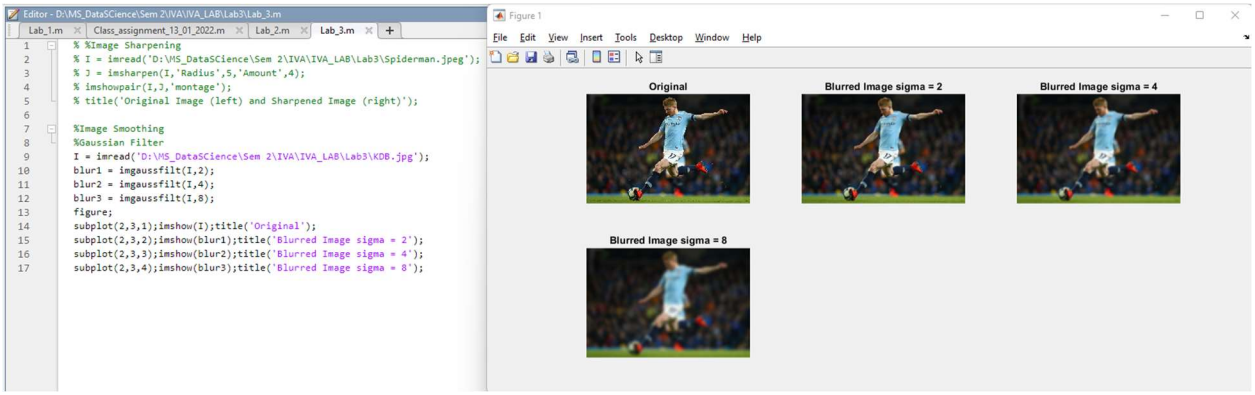


Image Tool 1 - ...
File Tools Window Help
Pixel info: (X, Y) Intensit Display range: [0 253]
Image Tool 2 - ...
File Tools Window Help
Pixel info: (X, Y) Pixel V Display range: [7 255]

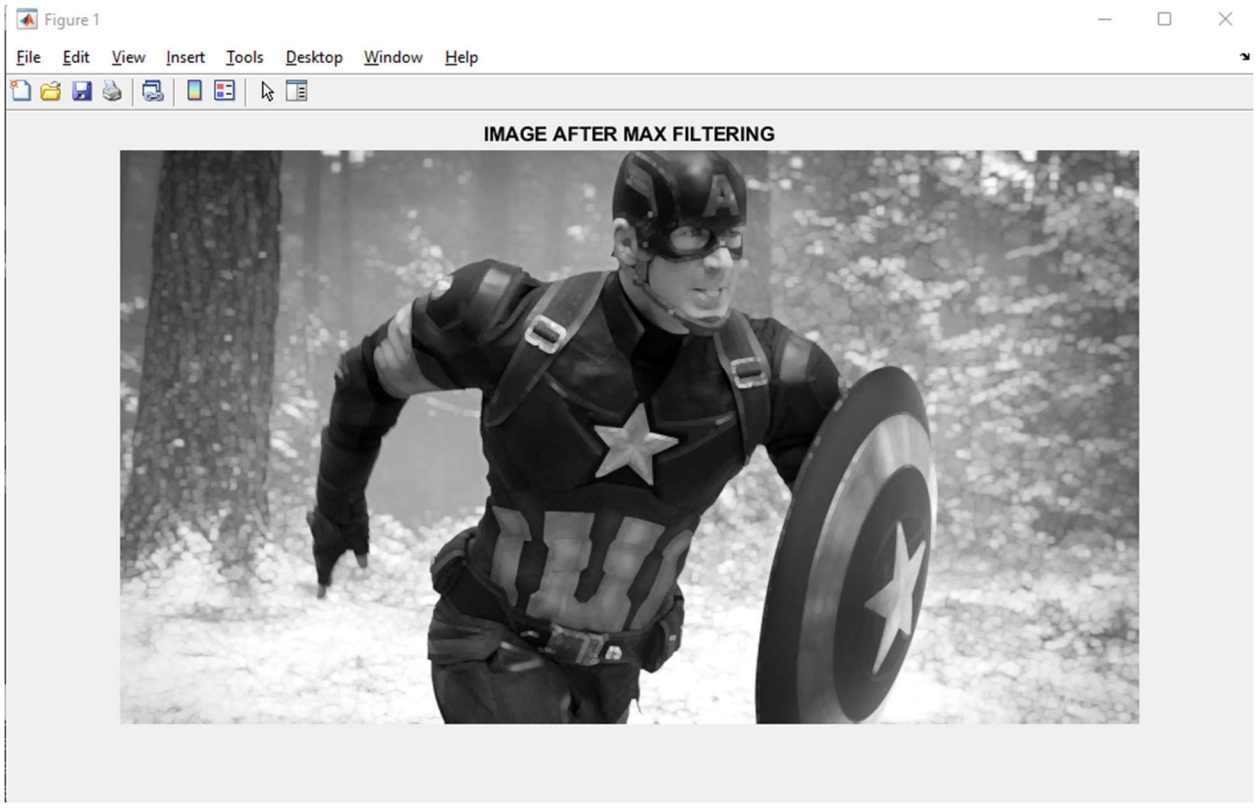

Editor - D:\MS_DataSCience\Sem 2\IVA\IVA_LAB\Lab3\Lab_3.m
Lab_1.m
Class_assignment_13_01_2022.m
Lab_2.m
Lab_3.m
% %Image Sharpening
% I = imread('D:\MS_DataSCience\Sem 2\IVA\IVA_LAB\Lab3\Spiderman.jpeg');
% J = imsharpen(I,'Radius',5,'Amount',4);
% imshowpair(I,J,'montage');
% title('Original Image (left) and Sharpened Image (right)');
%Image Smoothing
%Gaussian Filter
I = imread('D:\MS_DataSCience\Sem 2\IVA\IVA_LAB\Lab3\KDB.jpg');
blur1 = imgaussfilt(I,2);
blur2 = imgaussfilt(I,4);
blur3 = imgaussfilt(I,8);
figure;
subplot(2,3,1);imshow(I);title('Original');
subplot(2,3,2);imshow(blur1);title('Blurred Image sigma = 2');
subplot(2,3,3);imshow(blur2);title('Blurred Image sigma = 4');
subplot(2,3,4);imshow(blur3);title('Blurred Image sigma = 8');
Figure 1
File Edit View Insert Tools Desktop Window Help
Original
Blurred Image sigma = 2
Blurred Image sigma = 4
Blurred Image sigma = 8

2. Program to demonstrate Non linear filtering using Built-in and user defined functions with different size of kernels.

CODE(INBUILT):

```
% built in function
B = ordfilt2(img,9,ones(3,3));
imshow(B);
```

OUTPUT:



CODE(USER DEFINED):

```
im = imread('dog.jpg');
out_lm=zeros(size(im));

%PAD THE MATRIX A WITH ZEROS
padIm=padarray(im,[1 1]);

      x=[1:3]';
      y=[1:3]';

for i= 1:size(padIm,1)-2
   for j=1:size(padIm,2)-2
```

```
    %VECTORIZED METHOD
    window=reshape(padIm(i+x-1,j+y-1),[],1);

    %FIND THE MAXIMUM VALUE IN THE SELECTED WINDOW

    out_lm(i,j) = max(window);

  end
end

%CONVERT THE OUTPUT MATRIX TO 0-255 RANGE IMAGE TYPE
out_lm=uint8(out_lm);
figure,imshow(out_lm),title('IMAGE AFTER MAX FILTERING');
%convert the output into single array
output = out_lm(:);
```

OUTPUT: