```
function [ output_args ] = question1( input_args )
    %Author - Guneet Singh Mehta ,ECE Department, UW Madison
     im=imread('saturn.png');
     whos im;% whos im displays the information related to the image 'im' array
     % The size is displayed as pixels along length * pixels along width
     % whos im also displays the class i.e. double/uint8/uint16 etc of the
     % variable im along with the total space occupied in workspace

     % for the image 'saturn.png' - the image size is 328 X 438 pixels and
     % it occupies 143,664 bytes on disc

end
```

```
Name          Size               Bytes  Class     Attributes

  im          328x438            143664  uint8
```

The whos im command displays the information related to the image ‘im’ array. The information displayed by whos im command is

- The image size in pixels pixels along length X pixels along width
- The data type of image i.e uint8/uint16/ double etc
- Size of the image array ‘im’ in the workspace which is equal to = (total pixels in the image)*sizeof (each pixel value). For uint8 sizeof(each pixel value) is 1 byte.

Question 2

```
function[]=question2()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison
  im=double(imread('saturn.png')); % this command converts the uint8 matrix
  % a matrix of double numbers

end
```

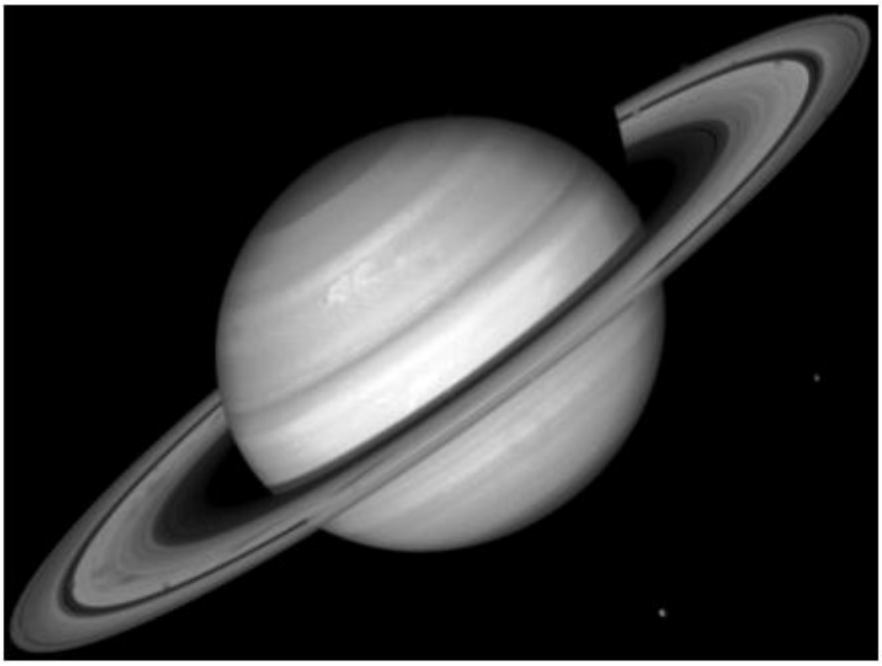
Question 3

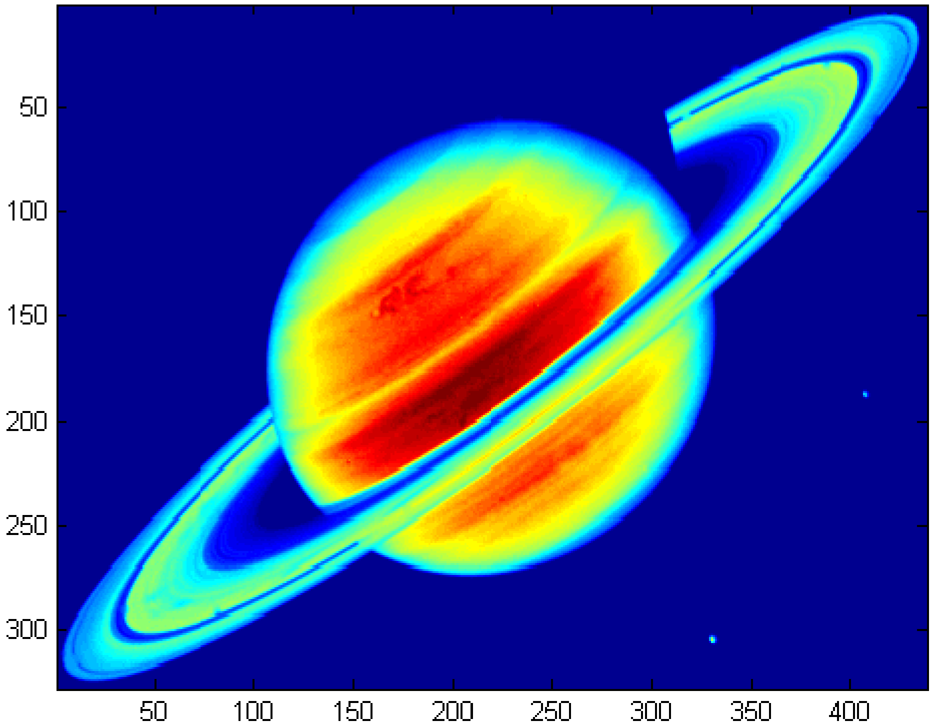
```
function[]=question3()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison
  im=imread('saturn.png');
  Max_pixel_value=max(im(:));
  Min_pixel_value=min(im(:));
  fprintf('in image saturn.png ,Maximum pixel value=%d and Minimum Pixel
Value=%d',Max_pixel_value,Min_pixel_value);

end
```

```
in image saturn.png ,Maximum pixel value=255 and Minimum Pixel Value=0
```

Question 4

```
function[]=question4()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison
  im=imread('saturn.png');
  figure;imshow(im);
  figure;imagesc(double(im));

  % imshow basically shows all images as it is. That is a grayscale image
  % will be displayed as a grayscale image , a color image will be shown as
  % a color image

  % while imagesc replaces the image data (grayscale in the case of
  % 'saturn.png' to colors from colormap. The colormap links each intensity
  % value in the image to a particular color. the default colormap of
  % imagesc is 'jet'
end
```





Step 1 – k1=0;k2=100;

This step just assigns two variables two values

Step 2 – figure(3)

This opens up a new figure.

Step 3 – imagesc(im,[k1,k2]);

This plots a color image with color of each pixel corresponding to the grayscale intensity value in the colormap. The default colormap being 'jet' with blue in the lowest value and red at the highest value.

If k1!=0 or if k2!=255 then, all pixels with grayscale intensities below k1 are plotted with the color corresponding to the lowest value on the colorbar and all pixels above or equal to k2 are plotted with the color corresponding to the maximum value on the colorbar.

Step 4 – colorbar- this command shows the colorbar on the fiure, beside the plotted image using imagesc.

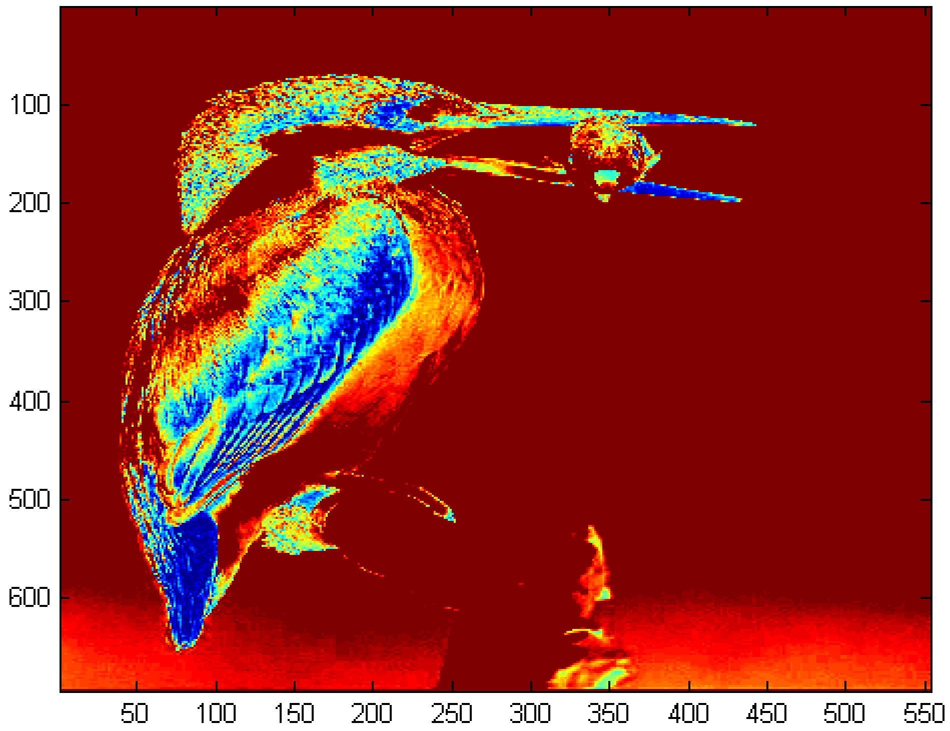
Step 5 – colormap gray – this changes the colorbar from the default 'jet' tp grayscale. So the pixels below k1 are plotted with gray intensity =0 and pixels with gray value greater than k2 are plotted as white. The values between k1 and k2 are stretched from 0 to 255.

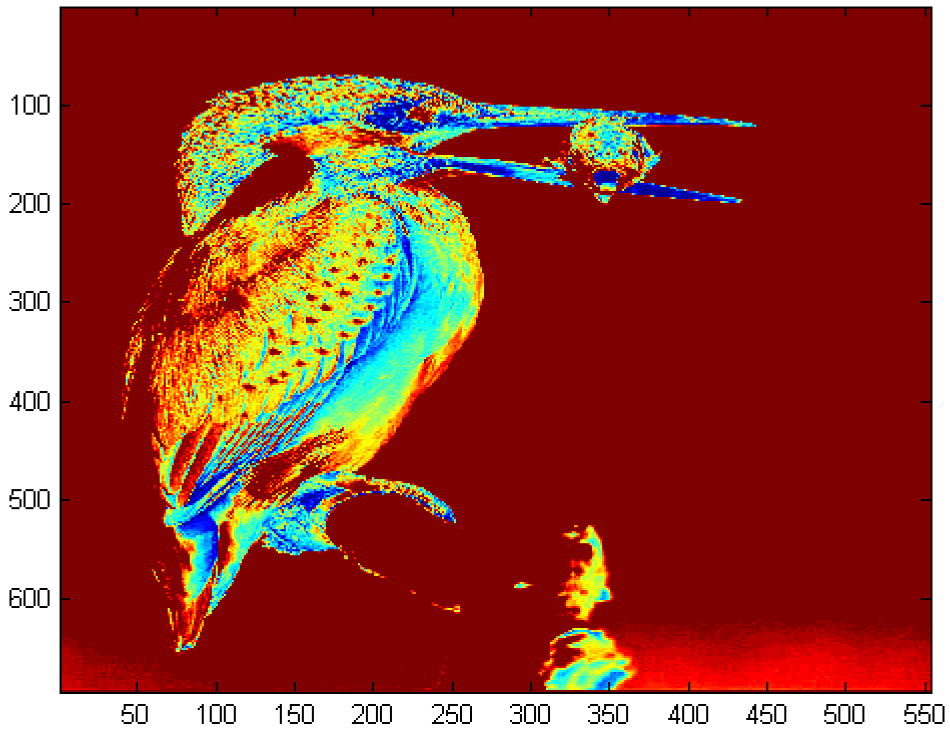
Check This Step 6 – axis image – this plots the image in the aspect ratio as indicated by its size. That is the image is not stretched unlike the image after step 5

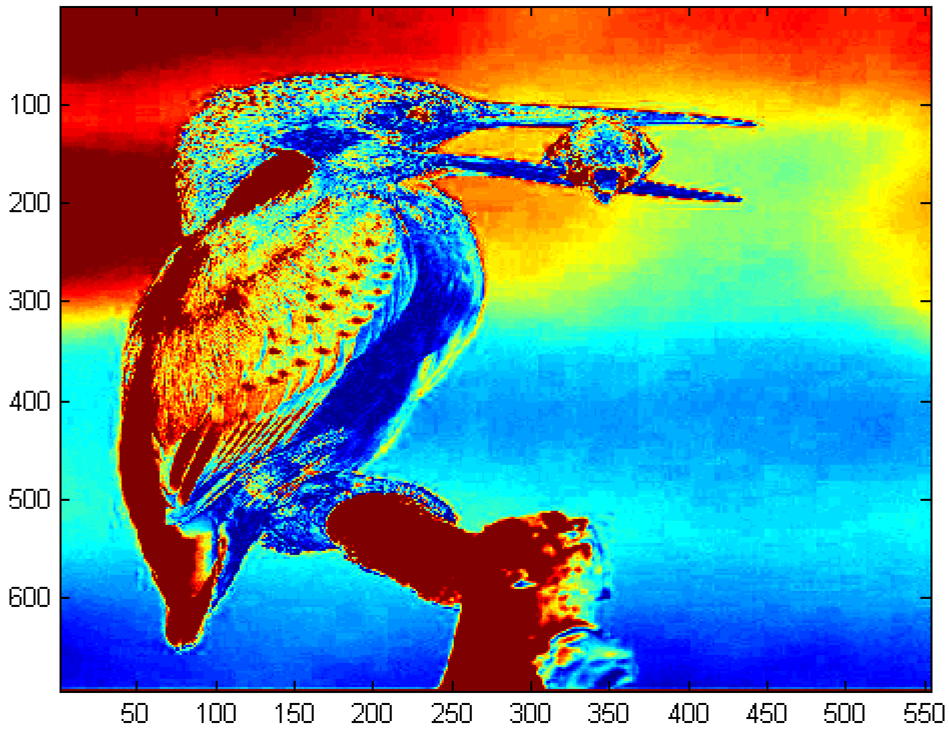
Question 5

```
function[]=question5()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison
  im=imread('BirdFish.jpg');
  min_3channel=min(im(:));
  max_3channel=max(im(:));
  fprintf('Maximum value in the image=%d\nMinimum Value in
image=%d\n',max_3channel,min_3channel);
  if(size(im,3)==3)
     for k=1:3
        min_value(k)=min(min(im(:,:,k)));
        max_value(k)=max(max(im(:,:,k)));
        fprintf('Minimum value for channel %d=%d Maximum value for channel
%d=%d\n',k,min_value(k),k,max_value(k));
     end
  end
  fprintf('Result of whos im command=\n');
  whos im;
  fprintf('plotting results of imagesc for each channel\n');
  k1=0;k2=100;
  if(size(im,3)==3)
     for k=1:3
        figure;imagesc(im(:,:,k),[k1,k2]);
     end
  end
end
```

```
Maximum value in the image=255
Minimum Value in image=0
Minimum value for channel 1=0 Maximum value for channel 1=255
Minimum value for channel 2=0 Maximum value for channel 2=255
Minimum value for channel 3=0 Maximum value for channel 3=255
Result of whos im command=
  Name        Size                Bytes  Class    Attributes

  im        693x553x3            1149687  uint8

plotting results of imagesc for each channel
```


100
200
300
400
500
600
50
100
150
200
250
300
350
400
450
500
550


100
200
300
400
500
600
50
100
150
200
250
300
350
400
450
500
550



Published with MATLAB® R2013a

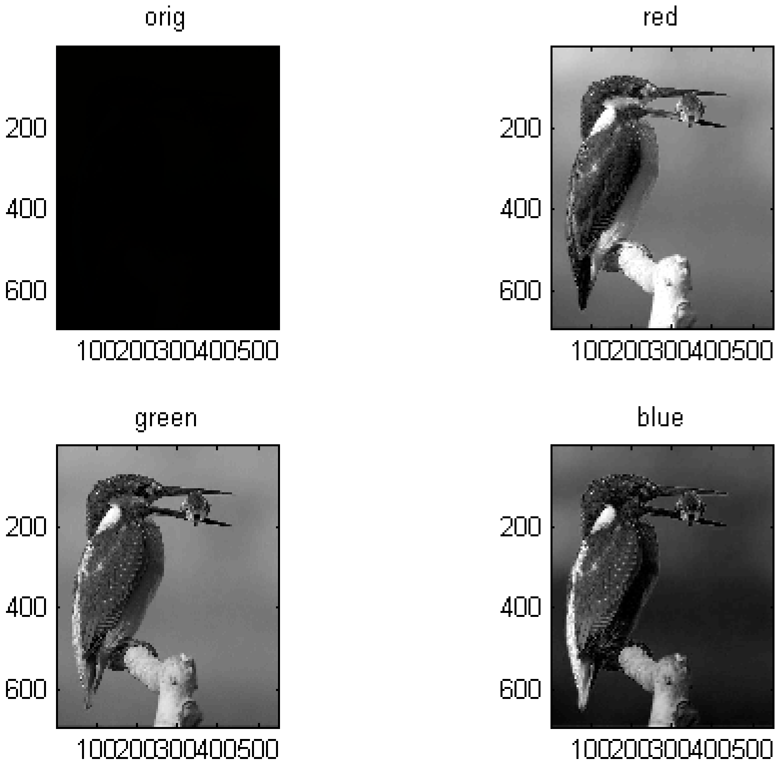
- Using the commands min(im(:)) and max(im(:)) the minimum and maximum value for the color image 'BirdFish.jpg' are 0 and 255 repectively
- Minimum and Maximum values in Red , Green and Blue channels are –
  a.) Red –Min= 0 Max=255
  b.) Green- Min= 0 Max=255
  c.) Blue - Min= 0 Max=255
- whos command displays the following
  a.) Name of the variable
  b.) Size of the variable im – Since the image is a color image , array 'im' is now three dimensional with each of the three z stacks corresponding to the Red , Green and Blue Channels in a color image
  c.) The size of variable im in bytes
  d.) Class of the variable 'im'
  e.) Empty field of Attributes.

imagesc does not work as expected, but using imagesc to display one channel at a time work correctly.

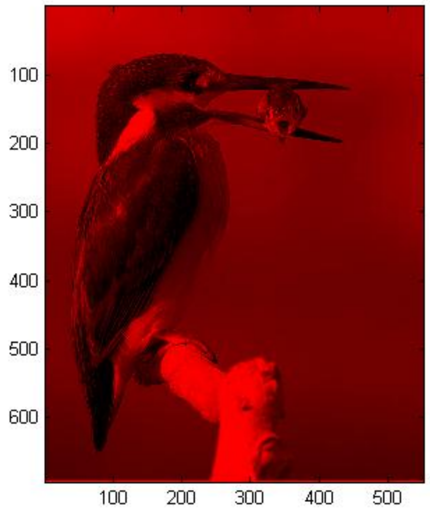
Question 6

```
function[]=question6()
  % Code Taken from - https://ay15-
16.moodle.wisc.edu/prod/pluginfile.php/90512/mod_assign/introattachment/0/Lab1.pdf?forcedownl
```

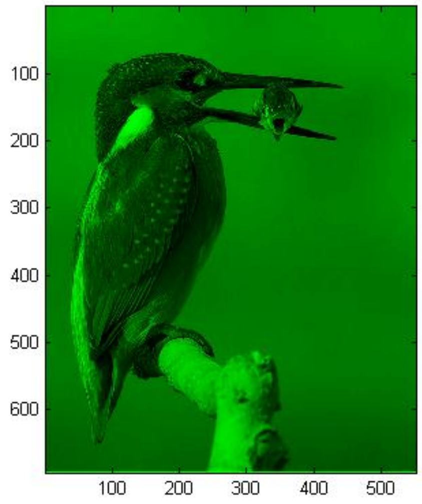
```
oad=1
  % code written courtsey to Prof Rebecca Willett, University of Wisconsin
  % Madison

  imC=imread('BirdFish.jpg');
  figure(15);
    subplot(221);image(imC/255);axis image;title('orig');
    subplot(222);imagesc(imC(:,:,1),[0,255]);axis image;
    title('red');colormap gray;
    subplot(223);imagesc(imC(:,:,2),[0,255]);axis image;
    title('green');colormap gray;
    subplot(224);imagesc(imC(:,:,3),[0,255]);axis image;
    title('blue');colormap gray;
    linkaxes;
    load cmapRGB
    figure(16);imagesc(imC(:,:,1),[0,255]);axis image;
    title('red');colormap(cmap_red)
    figure(17);imagesc(imC(:,:,2),[0,255]);axis image;
    title('green');colormap(cmap_green)
    figure(18);imagesc(imC(:,:,3),[0,255]);axis image;
    title('blue');colormap(cmap_blue)
end
```



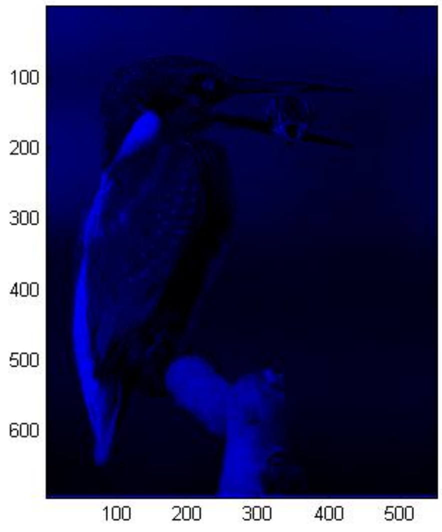
red



green



blue



Published with MATLAB® R2013a

Question 7

```
function[]=question7()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison
  % code snippets for imtransform taken from MATLAB help of function imtransform
  im=imread('saturn.png');

  figure;imshow(imrotate(im,-90));title('Image Rotation using imrotate');
  A=[0 1 0;-1 0 0;0 0 1]; % A is of the form of [cos(angle),sin(angle),0;-sin(angle)
cos(angle) 0;0 0 1]
  tform = maketform('affine',A);
  J = imtransform(im,tform);
  figure, imshow(J);title('Image Rotation using imtransform');

  if(size(im,3)==3)
     for k=1:3
        im2_temp=flipdim(im(:,:,k),2);
        im2_temp=transpose(im2_temp);
        im2_temp=flipdim(im2_temp,1);
        im2_temp=flipdim(im2_temp,2);
        im2(:,:,k)=im2_temp;
     end
  elseif(size(im,3)==1)
      im2=flipdim(im,2);%flipping along columns
      im2=transpose(im2);%transpose
      im2=flipdim(im2,1);% flipping along rows
      im2=flipdim(im2,2);%flipping along columns
  end
  figure;imshow(im2);title('rotating image using matrix operations ');
end
```

Image Rotation using imrotate

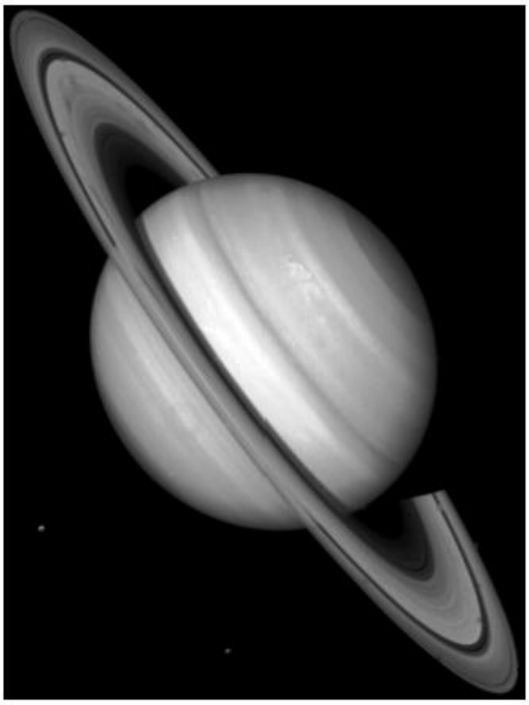
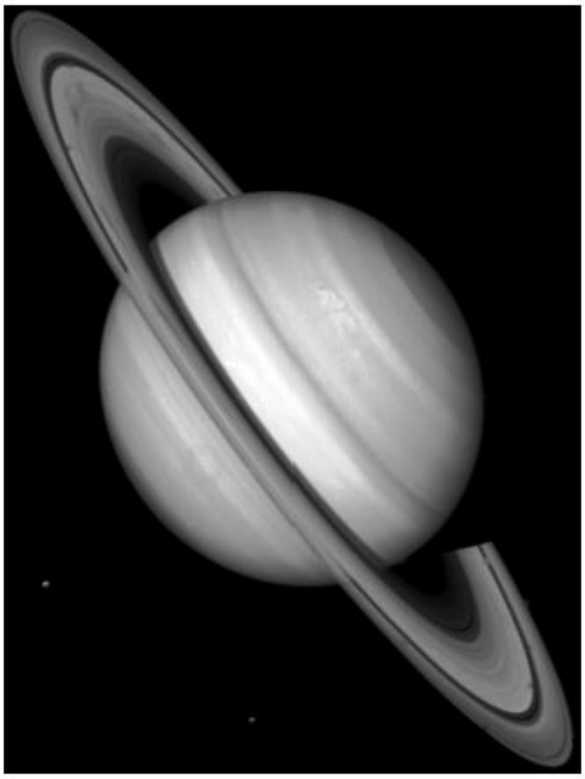
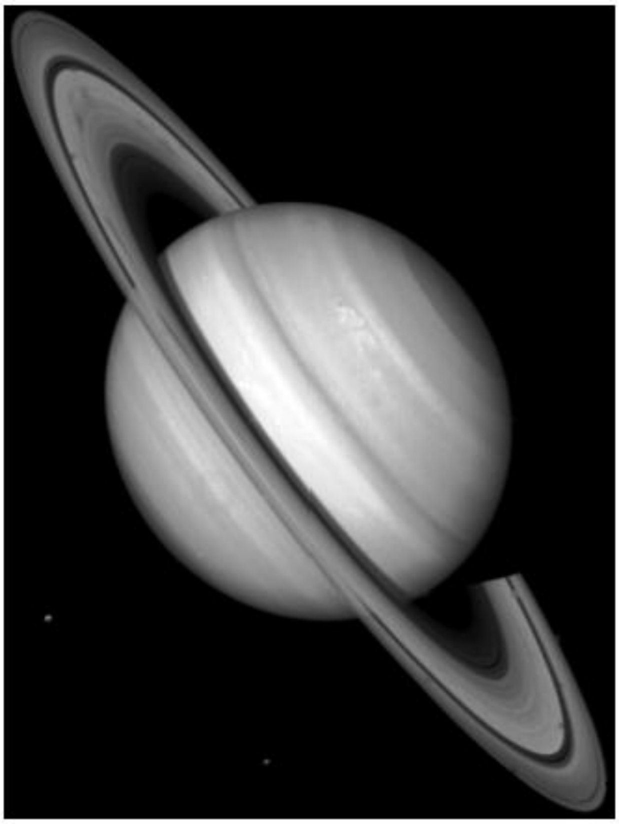


Image Rotation using imtransform



rotating image using matrix operations



Published with MATLAB® R2013a

Question 8

```
function[]=question8()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison

  im=imread('saturn.png');
  figure;imshow(im);title('orignal image');

  steps=5;
  A=[1 0 0;0 1 0;-steps 0 1]; % A is of the form of [cos(angle),sin(angle),0;-sin(angle)
cos(angle) 0;xshift yshift 1]
  % -steps because the shift is towards left
  tform = maketform('affine',A);
  J = imtransform(im,tform);
  figure, imshow(J);title('Image translation using imtransform');

  s1=size(im,1);s2=size(im,2);
  steps=50;
  im2(:,1:s2-steps)=im(:,steps+1:end);
  im2(:,s2-steps+1:s2)=uint8(0);
  figure;imshow(im2);title('shifting image by matrix manipulation');

end
```

orignal image

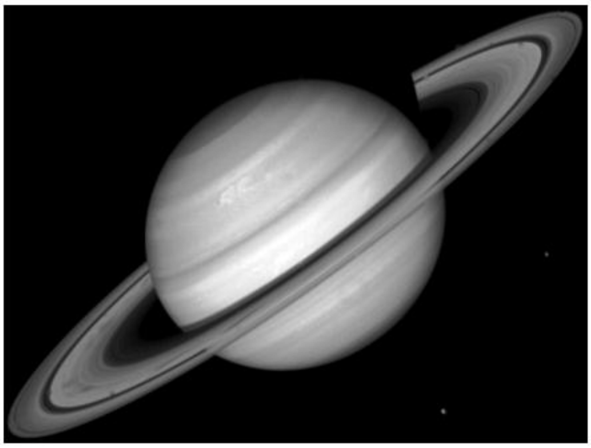
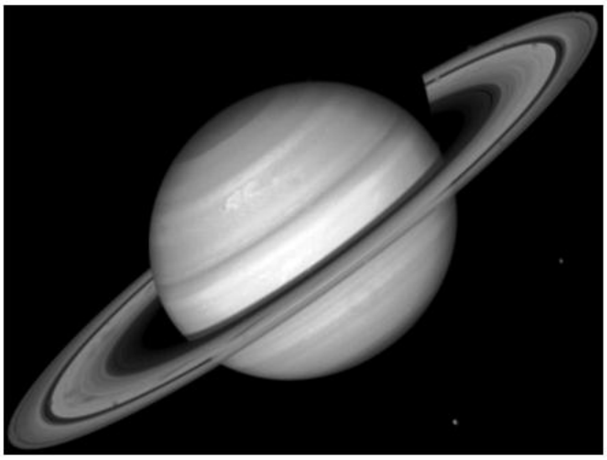
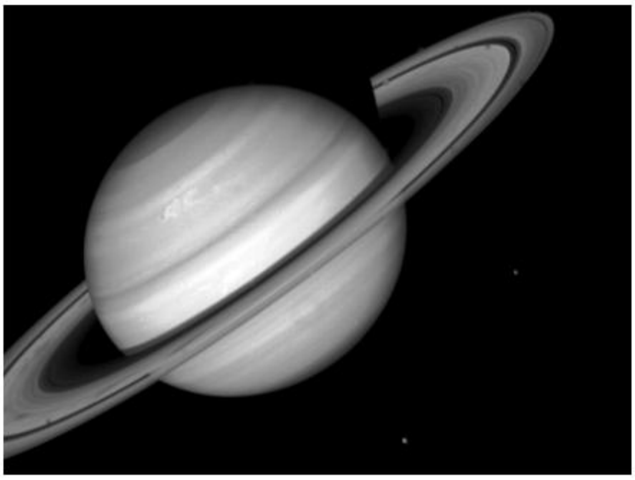


Image translation using imtransform



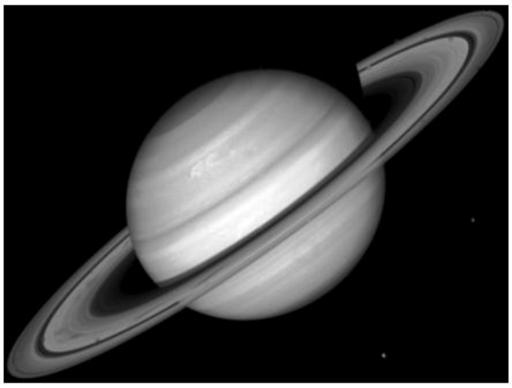
shifting image by matrix manipulation



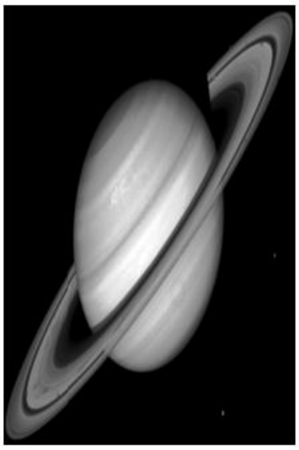
Published with MATLAB® R2013a

Question 9

```
function[]=question9()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison

  im=imread('saturn.png');
  s1=size(im,1);s2=size(im,2);
  figure;imshow(im);title(['orignal image with size=',num2str(s1),'x',num2str(s2)]);
  im2=imresize(im,[s1,s2/2]);
  figure;imshow(im2);title(['modified image with size=',num2str(s1),'x',num2str(s2/2)]);
end
```



orignal image with size=328x438



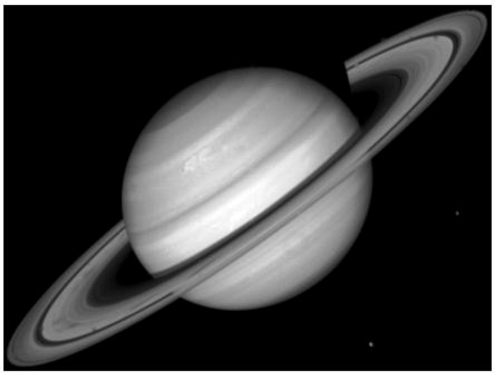
modified image with size=328x219

Published with MATLAB® R2013a

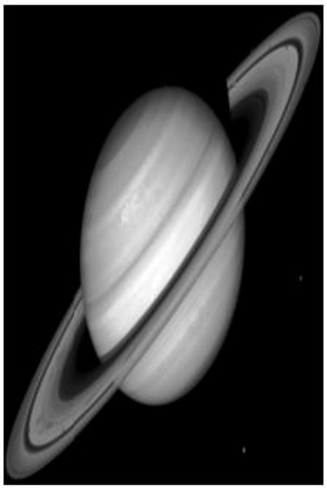
Question 10

```
function[]=question10()
  %Author - Guneet Singh Mehta ,ECE Department, UW Madison

  im=imread('saturn.png');
  s1=size(im,1);s2=size(im,2);
  figure;imshow(im);title(['orignal image with size=',num2str(s1),'x',num2str(s2)]);
  im2=imresize(im,[s1,s2/2]);
  figure;imshow(im2);title(['modified image with size=',num2str(s1),'x',num2str(s2/2)]);

  imwrite(im2,'saturn_modified.png'); %writing the modified image to default directory
end
```

orignal image with size=328x438



modified image with size=328x219



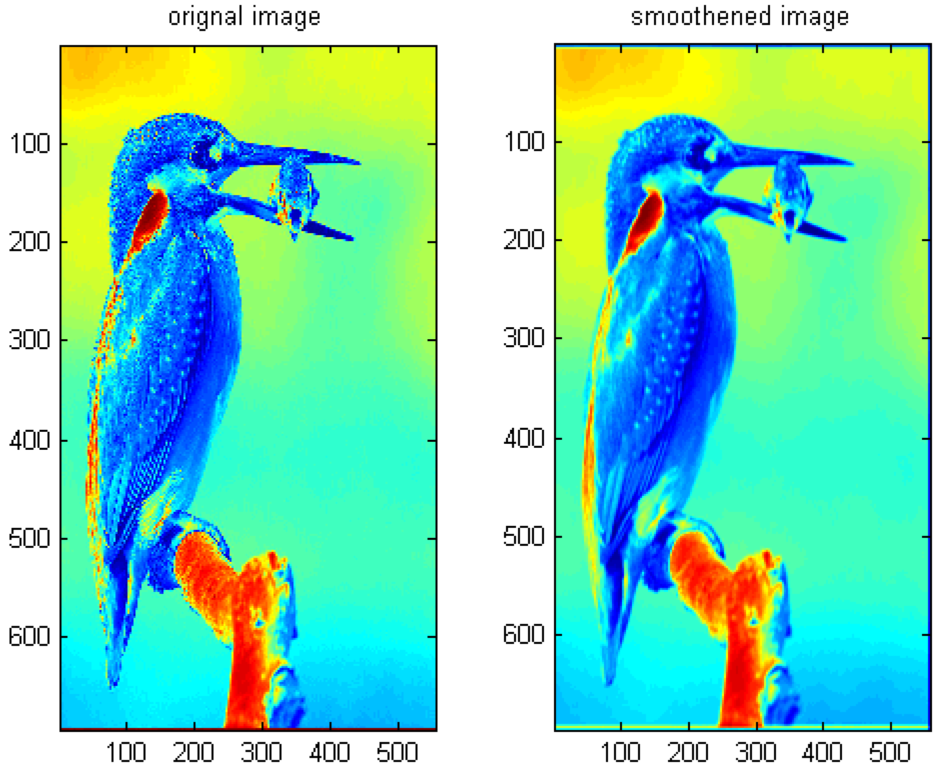
Published with MATLAB® R2013a

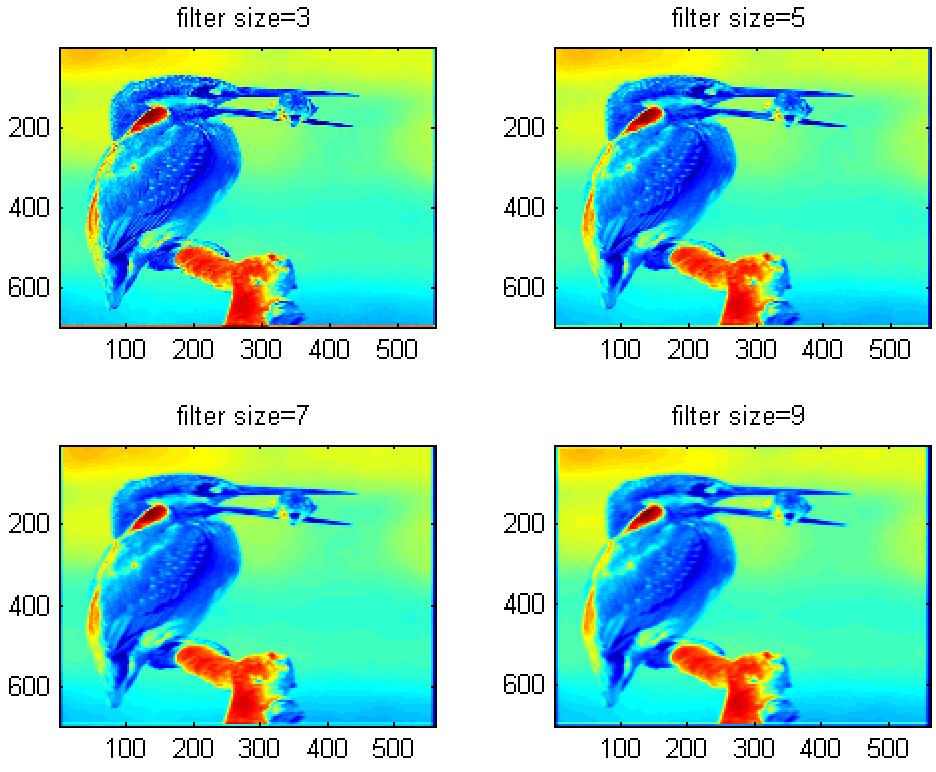
Part B

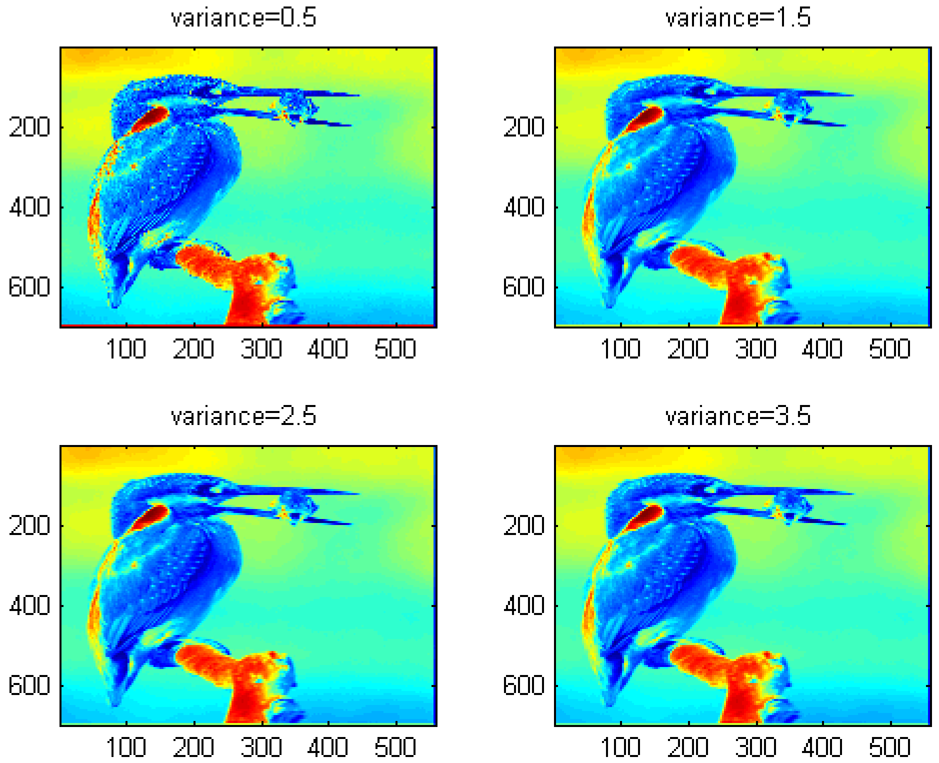
Question 1

```
% Implement a spatial smoothing filter. (HINT: the Matlab command fspecial has several useful
% filters you may use, and conv2 performs two-dimensional convolution.) How does the filter
performance
% change as you change the size of the filter? What if you leave the filter size constant but
```

```
vary
% the values of the filter coefficients? What is the difference between a boxcar and Gaussian
filter?
function[]=Bquestion1()
% Author - Guneet Singh Mehta, ECE Masters Student, UW Madison
% email id- gmehta2@wisc.edu
    h1=fspecial('gaussian',5,2);
    image=imread('BirdFish.jpg');
    if(size(image,3)==3)
      image=rgb2gray(image);
    end

    imout=conv2(double(image),double(h1));
    colormap  jet;
    f1=figure;% part 1 of question
    subplot(1,2,1);imagesc(uint8(image));title('orignal image');
    subplot(1,2,2);imagesc(uint8(imout));title('smoothened image');

    f2=figure;%change of filter size
    var=3;% for more visible filtering
    filter_size_num=4;rows=2;column=ceil(filter_size_num/rows);
    for i=1:filter_size_num
        filter_size=3+2*(i-1);
        h1=fspecial('gaussian',filter_size,var);
        imout=conv2(double(image),double(h1));
        subplot(rows,column,i);imagesc(uint8(imout));
        title_string=['filter size=' num2str(filter_size)];
        title(title_string);
    end
    % as filter size increases the image becomes more blurred

    % for varying variances
    f3=figure;%change of filter size
    filter_size_num=4;rows=2;column=ceil(filter_size_num/rows);
    var_num=4;
    for i=1:var_num
        var=0.5+1*(i-1);
        h1=fspecial('gaussian',5,var);
        imout=conv2(double(image),double(h1));
        subplot(rows,column,i);imagesc(uint8(imout));
        title_string=['variance=' num2str(var)];
        title(title_string);
    end
    % as variance increases for the same filter size , image gets
    % progressively blurred

    % A gaussian filter for image filtering has maximum value of
    % coefficient near the center of the filter , which decreases according
    % to the Gaussian function as distance from the center increases. While
    % in Boxcar filter the value of all the coefficients in the filter
    % remain the same and equal to =1/(number of elements in the filter)

end
```


orignal image
smoothened image
100
200
300
400
500
600
100
200
300
400
500

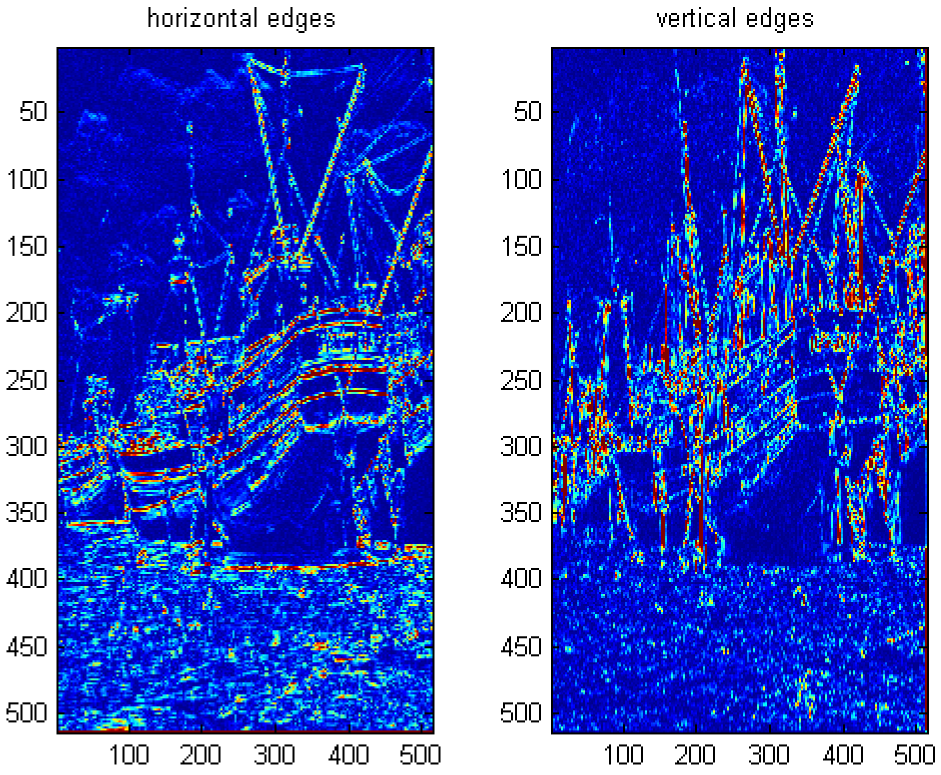

filter size=3
filter size=5
filter size=7
filter size=9
200
400
600
100
200
300
400
500

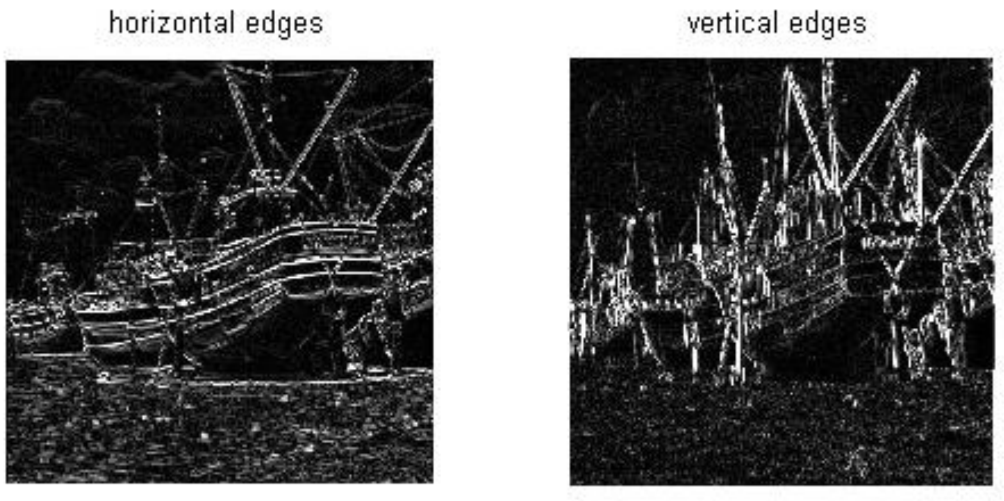


Published with MATLAB® R2013a

Question 2

```
% Question 2 Part B
% Implement a spatial edge-emphasizing filter. Again, what is the impact of different choices
of filter
% coefficients? Does the emphasis of edges depend on the edge orientation? How is this
related to the
% filter coefficients?

function[]=Bquestion2()
% Author - Guneet Singh Mehta, ECE Masters Student, UW Madison
% email id- gmehta2@wisc.edu

    % Using sobel filter to detect edges
    image=imread('boat.png');% using this image becuase of horizontal and vertical lines
    horz_edge_filter=fspecial('sobel');
    vert_edge_filter=transpose(fspecial('sobel'));

    imout_horz=conv2(double(image),double(horz_edge_filter));
    imout_vert=conv2(double(image),double(vert_edge_filter));

    colormap jet;
    f1=figure;
    subplot(1,2,1);imagesc(uint8(abs(imout_horz)));title('horizontal edges');
    subplot(1,2,2);imagesc(uint8(abs(imout_vert)));title('vertical edges');

    f2=figure;
    subplot(1,2,1);imshow(uint8(abs(imout_horz)));title('horizontal edges');
    subplot(1,2,2);imshow(uint8(abs(imout_vert)));title('vertical edges');
end

% A sobel filter with coeffs= [1 2 1; 0 0 0;-1 -2 -1] on encountering a
% horizontal boundary , lets say [ 1 1 1 ; 0 0 0 ; 0 0 0] will result in a
% non zero value while on encountering a vertical edge like
% [1 0 0;1 0 0;1 0 0] will result in a zero value because the coefficients
% times the pixel value will negate each other.
% Similarly the transpose of sobel filter will detect only the vertical
% edges and not horizontal edges
```



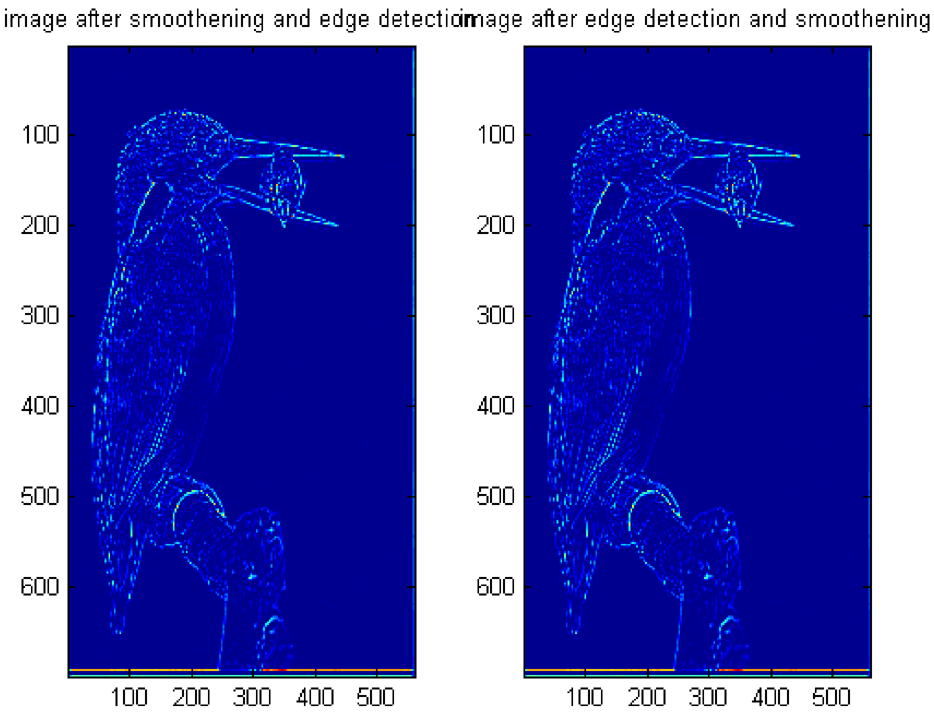


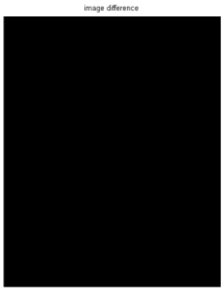
Published with MATLAB® R2013a

Question 3

```
% Apply your smoothing filter to your image, and then apply an edge-emphasizing filter to the
smoothed
% image. Is this result the same or different than if we reversed the order of the two filter
operations?
% Why or why not?

function[]=Bquestion3()
% Author - Guneet Singh Mehta, ECE Masters Student, UW Madison
% email id- gmehta2@wisc.edu
    h_smooth=fspecial('gaussian',5,2);
    h_laplacian=fspecial('laplacian');

    image=imread('BirdFish.jpg');
    if(size(image,3)==3)
       image=rgb2gray(image);
    end

    im_smooth_edge=conv2(double(image),double(h_smooth));
    im_smooth_edge=conv2(double(im_smooth_edge),double(h_laplacian));
    im_edge_smooth=conv2(double(image),double(h_laplacian));
    im_edge_smooth=conv2(double(im_edge_smooth),double(h_smooth));

    f1=figure;
    colormap jet;
    subplot(1,2,1);imagesc(uint8(im_smooth_edge));title('image after smoothening and edge
detection');
    subplot(1,2,2);imagesc(uint8(im_edge_smooth));title('image after edge detection and
smoothening');

    figure;imshow(uint8(im_edge_smooth-im_smooth_edge));title('image difference');
end
% there is no difference between smoothened followed by edge enhanced image
% and edge enhanced followed by smoothened image becuase both operations
% are a result of convolution with different masks, and since convolution
% is a linear operation , the order in which the linear operations are done
% on the image  don't matter
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```





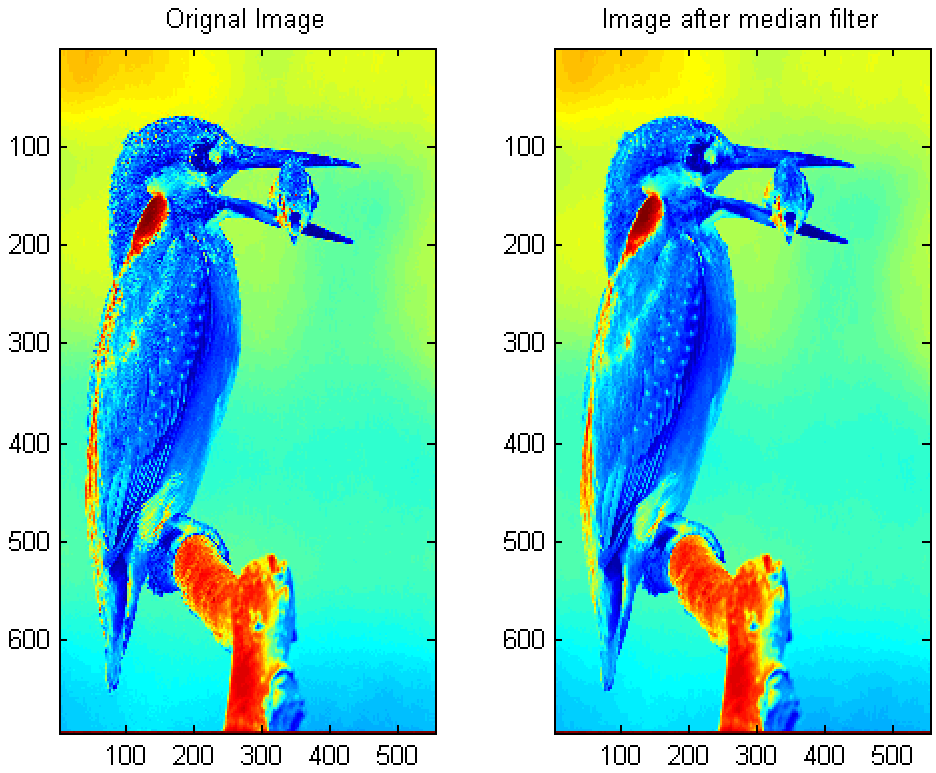
Published with MATLAB® R2013a

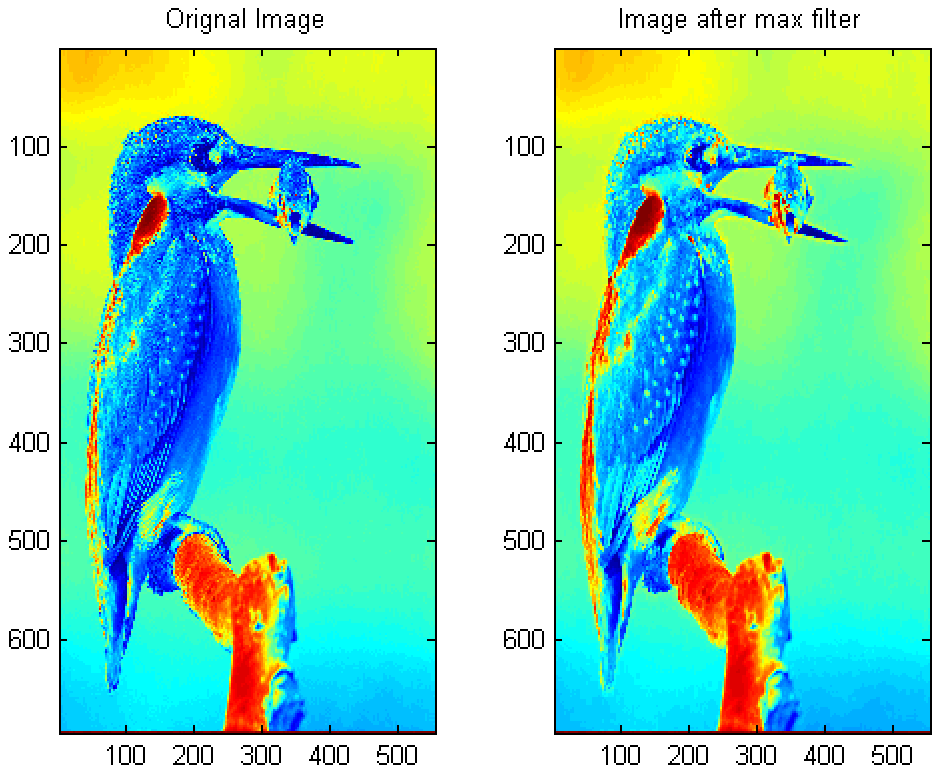
Question 4

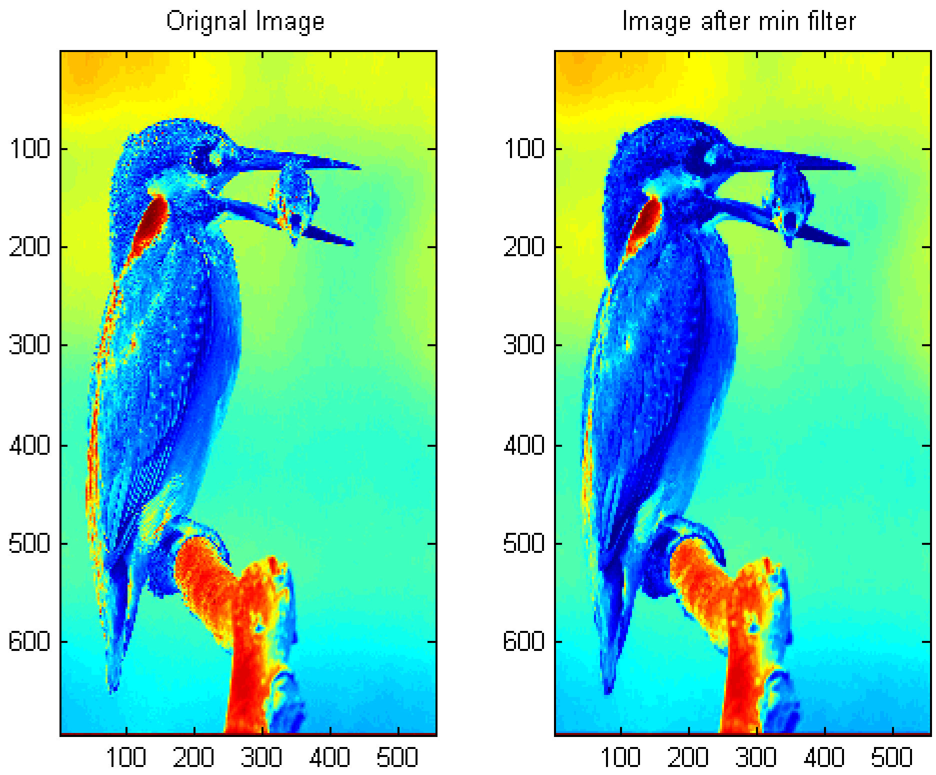
```
% Generate a sharpening filter by combining the above smoothing and edge-emphasizing filters.
HINT:
% this is a kind of unsharp masking. Explain how the same process can be implemented with a
single
% filter.


function [ output_args ] = Bquestion4( input_args )
    %Author - Guneet Singh Mehta ,ECE Department, UW Madison
     im=imread('BirdFish.jpg');im=rgb2gray(im);
     %im=imread('saturn.png');
     %im=imread('boat.png');
     h1=fspecial('laplacian');
     sharp=conv2(im,h1);
     h2=fspecial('gaussian',3,0.5);
     smooth=conv2(im,h2);
    figure;imshow(im);title('orignal');
    figure; imshow(uint8(sharp));title('sharp');
    figure;imshow(uint8(smooth));title('smooth');
     figure;imshow(uint8(0.5*sharp+smooth));title('unsharp k=0.5');
    figure;imshow(uint8(0.3*sharp+smooth));title('unsharp k=0.3');
     h3=h1+h2;
     combined=conv2(im,h3);
     figure;imshow(uint8(combined));title('combined');

end
```

Question 5

```
% Implement a median filter. (For this lab, do NOT use a built-in function like medfilt2.
Also
% implement max and min filters
function[]=Bquestion5()
% Author - Guneet Singh Mehta, ECE Masters Student, UW Madison
% email id- gmehta2@wisc.edu

    image=imread('BirdFish.jpg');
    if(size(image,3)==3)
       image=rgb2gray(image);
    end
    s1=size(image,1);s2=size(image,2);
    imout(1:s1,1:s2)=uint8(0);
    imout_max(1:s1,1:s2)=uint8(0);
    imout_min(1:s1,1:s2)=uint8(0);

    colormap jet
    mask_size=3;
    for i=1:s1
        for j=1:s2
            if(i<=floor(mask_size/2)||i>=s1-floor(mask_size/2)||j<=floor(mask_size/2)||j>=s2-
floor(mask_size/2))
               imout(i,j)=image(i,j);
               imout_max(i,j)=image(i,j);
               imout_min(i,j)=image(i,j);
            else
               submatrix=image(i-floor(mask_size/2):i+ floor(mask_size/2),j-
floor(mask_size/2):j+ floor(mask_size/2));
               imout(i,j)=median(submatrix(:));
               imout_max(i,j)=max(submatrix(:));
               imout_min(i,j)=min(submatrix(:));
            end
        end
    end
    figure;
    subplot(1,2,1);imagesc(image);title('Orignal Image');
    subplot(1,2,2);imagesc(imout);title('Image after median filter');

    figure;
    subplot(1,2,1);imagesc(image);title('Orignal Image');
    subplot(1,2,2);imagesc(imout_max);title('Image after max filter');

    figure;
    subplot(1,2,1);imagesc(image);title('Orignal Image');
    subplot(1,2,2);imagesc(imout_min);title('Image after min filter');

end
```


Orignal Image
Image after median filter
100
200
300
400
500
600
100 200 300 400 500
100
200
300
400
500
600
100 200 300 400 500


Orignal Image
Image after max filter
100
200
300
400
500
600
100 200 300 400 500
100
200
300
400
500
600
100 200 300 400 500

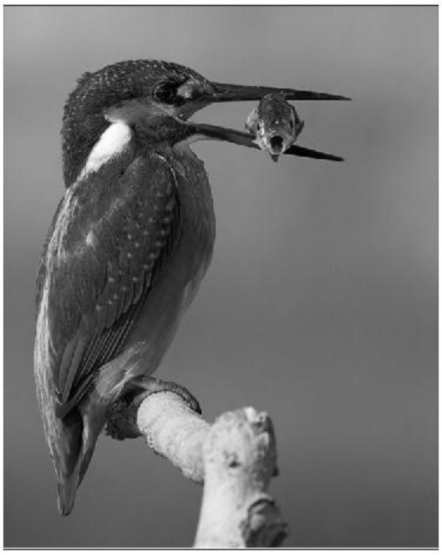


Orignal Image

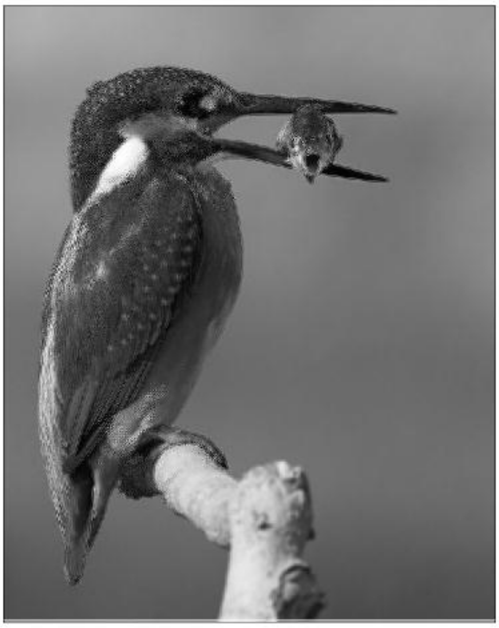
Image after min filter

Published with MATLAB® R2013a

sharp



smooth



unsharp k=0.5

unsharp k=0.3

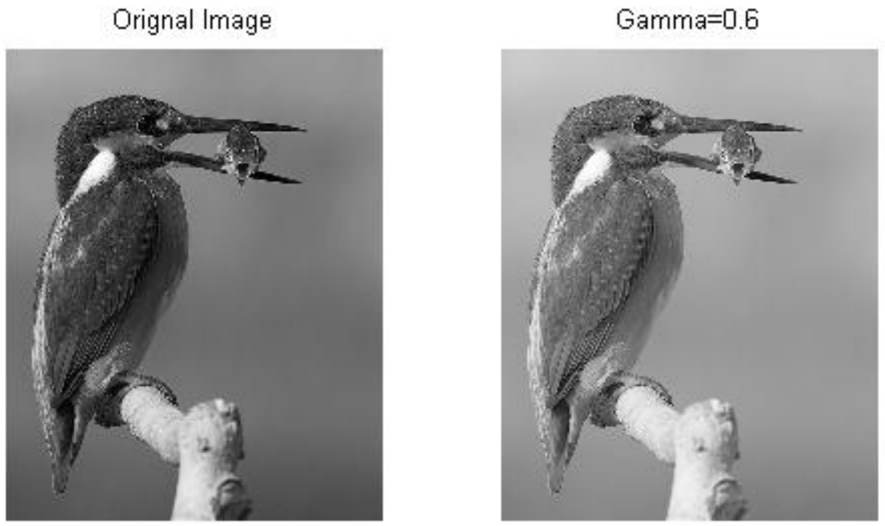
combined

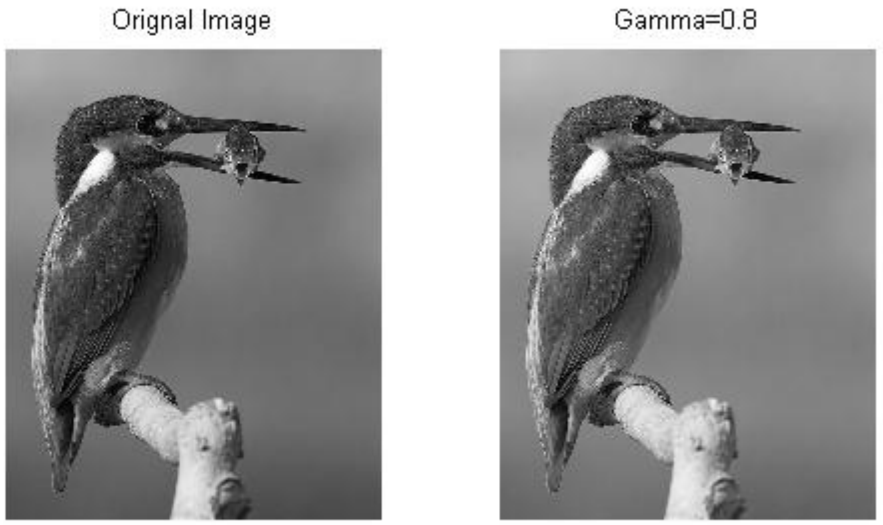
Published with MATLAB® R2013a

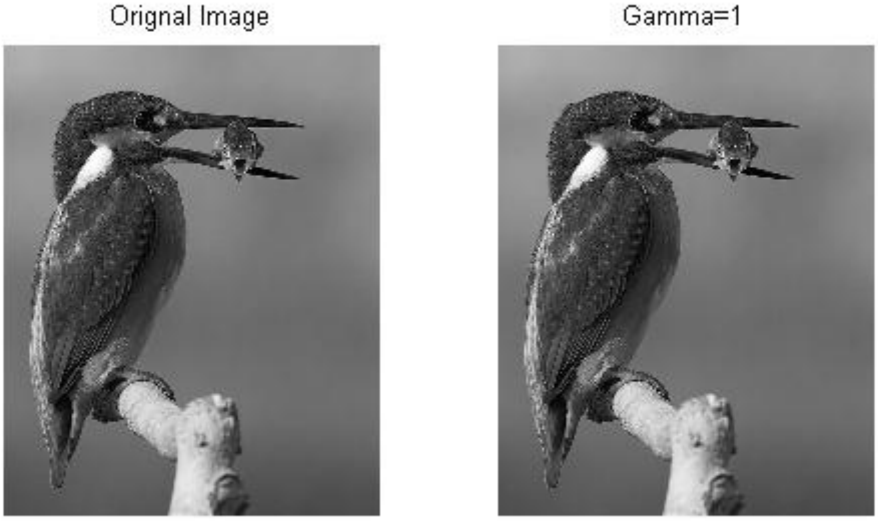
Question 6

```
% Experiment with
%  filtering for contrast enhancement. In particular, if im is your original image and
% im_gam is your filtered image, then for each pixel i the
%  filter performs the following operation:
% im_gam[i] = im[i]^gamma;
% What is the effect of this filter? How does the effect change with the value of
% , particularly when it
% is less than or greater than 1? Does it help if your image pixel values are between 0 and
1?
```

```
function[]=Bquestion6()
% Author - Guneet Singh Mehta, ECE Masters Student, UW Madison
% email id- gmehta2@wisc.edu
     image=imread('BirdFish.jpg');
    if(size(image,3)==3)
       image=rgb2gray(image);
    end
    s1=size(image,1);s2=size(image,2);
    imout(1:s1,1:s2)=uint8(0);

    for gamma=0.6:0.2:1.4
        c=255/(255)^gamma;
        for i=1:s1
            for j=1:s2
                imout(i,j)=uint8(c*double(image(i,j))^gamma);
            end
        end
        figure;
        title_text=['Gamma=' num2str(gamma)];
        subplot(1,2,1);imshow(image);title('Orignal Image');
        subplot(1,2,2);imshow(imout);title(title_text);
        %colorbar;
%         display(size(imout));
%         display(size(image));
    end
    % for gamma values less than 1 the contrast in the darker regions (low
    % intensity value) is increased while for gamma values greater than 1
    % the contrast in the brighter regions (high intensity value ) is
    % increased

end
```

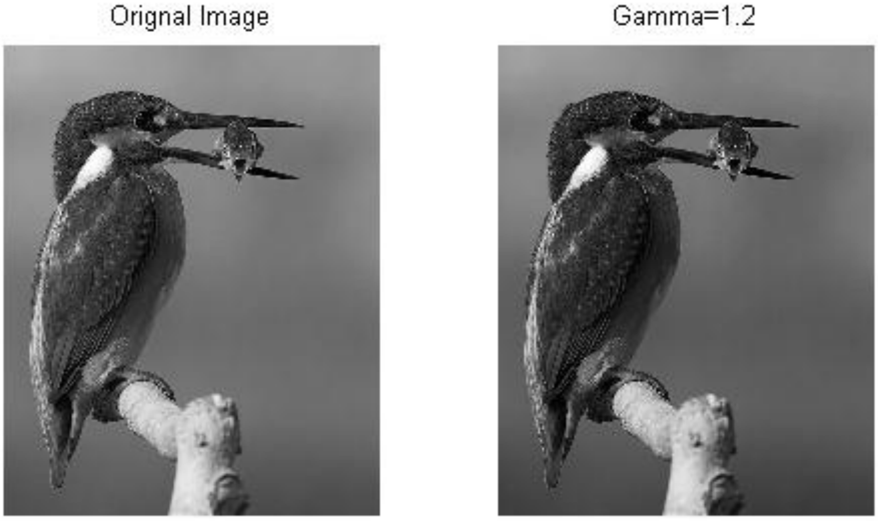

Orignal Image
Gamma=0.6


Orignal Image
Gamma=0.8



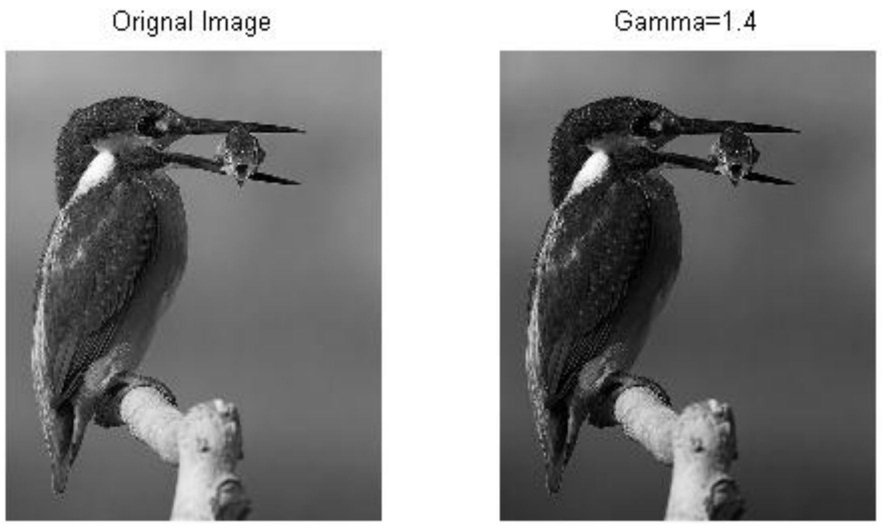
Orignal Image

Gamma=1



Orignal Image

Gamma=1.2



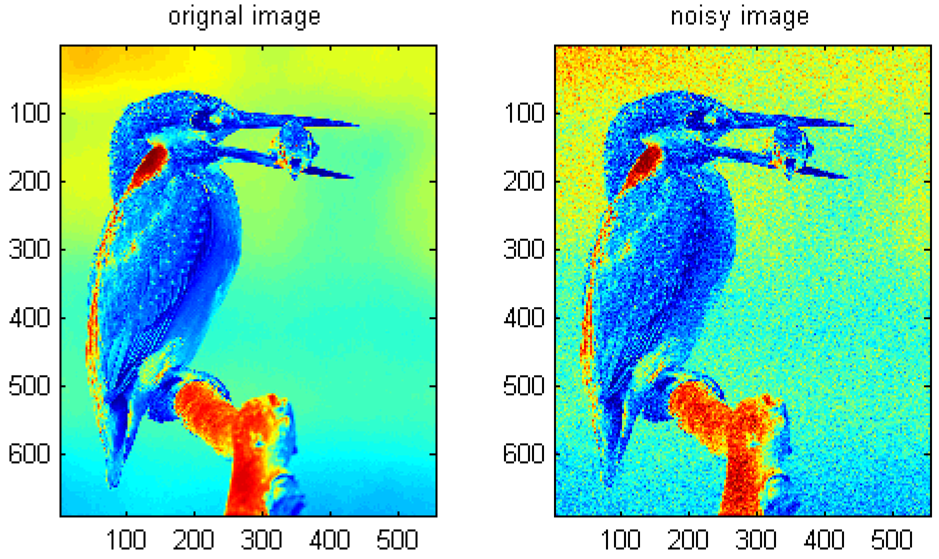
Published with MATLAB® R2013a

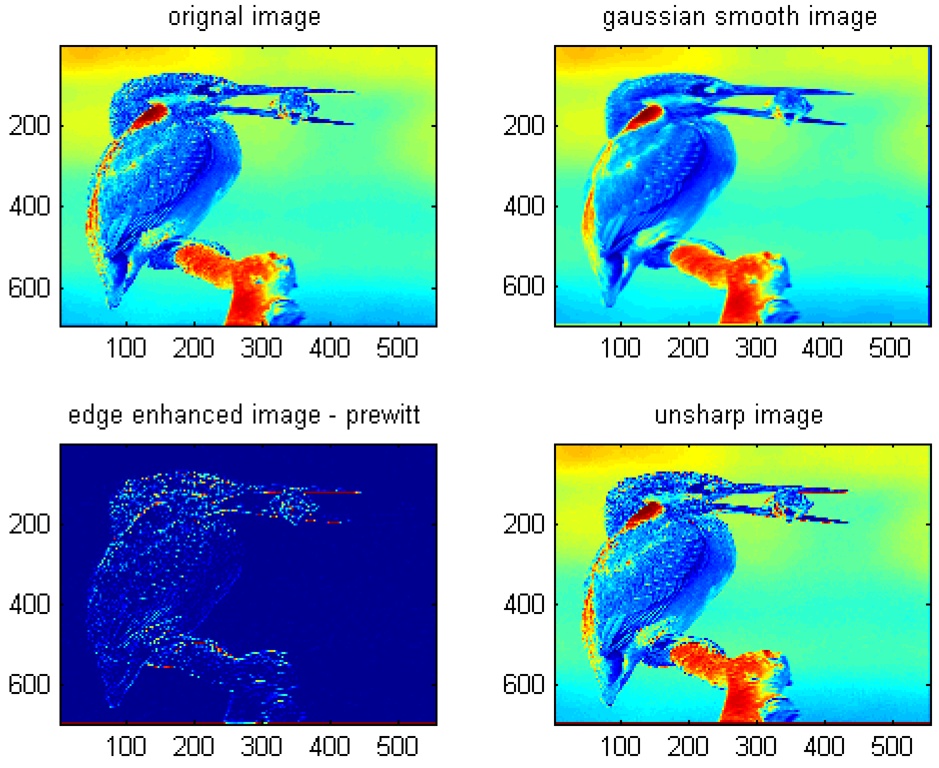
Question 7

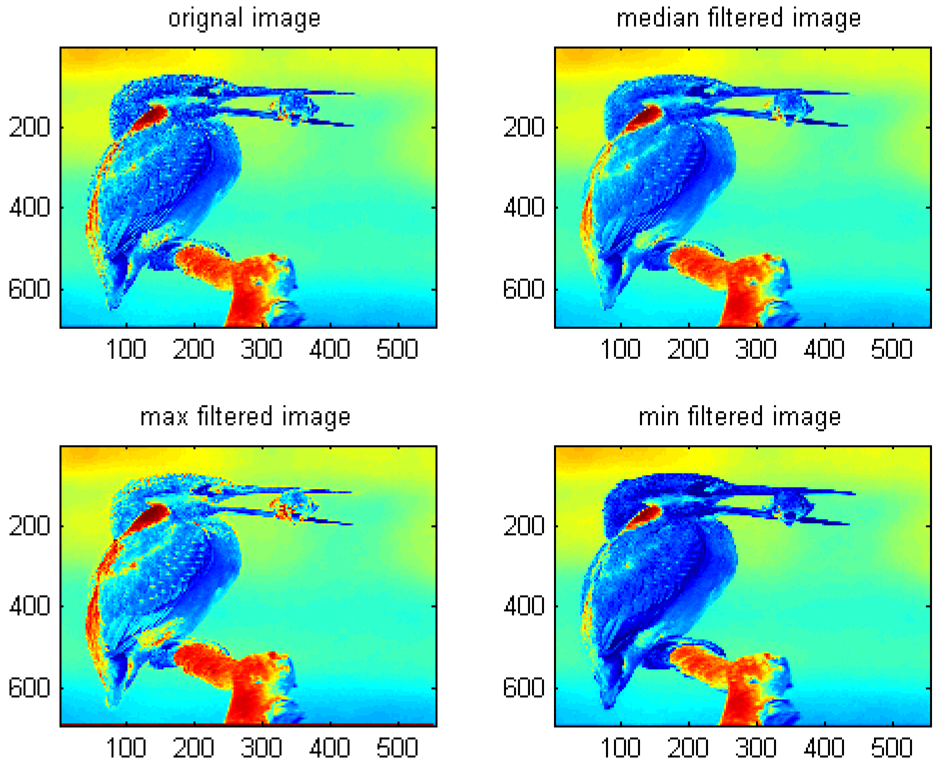
```
% Next experiment with the effect of these filters on a noisy image.
function[]=Bquestion7()
% Author - Guneet Singh Mehta, ECE Masters Student, UW Madison
% email id- gmehta2@wisc.edu

% code copied from pdf handout in ECE 533 as illustration
    image = imread('BirdFish.jpg');
    if(size(image,3)>1)
       image=rgb2gray(image);
    end
    figure;imshow(image);
    sigma = 15;
    noise = randn(size(image))*sigma;
    im_noisy = double(image) + noise;
    figure;
    subplot(1,2,1);imagesc(uint8(image));title('orignal image');
    axis image;colormap gray;
    subplot(1,2,2);imagesc(uint8(im_noisy),[min(image(:)), max(image(:))]);title('noisy
image');
    axis image;colormap gray;
    linkaxes;
    %pause(5);

    h_smooth=fspecial('gaussian',5,2);
    h_edge_prewitt=fspecial('prewitt');
    h_unsharp=fspecial('gaussian',3,0.5)+0.4*h_edge_prewitt;
    colormap jet;
    imout_smooth=uint8(conv2(double(image),double(h_smooth)));
    imout_prewitt=uint8(conv2(double(image),double(h_edge_prewitt)));
    imout_unsharp=uint8(conv2(double(image),double(h_unsharp)));
    f1=figure;
    subplot(2,2,1);imagesc(image);title('orignal image');
    subplot(2,2,2);imagesc(imout_smooth);title('gaussian smooth image');
    subplot(2,2,3);imagesc(imout_prewitt);title('edge enhanced image - prewitt');
    subplot(2,2,4);imagesc(imout_unsharp);title('unsharp image');
    %median ,max and min filtering
    s1=size(image,1);s2=size(image,2);
    imout_median(1:s1,1:s2)=uint8(0);
    imout_max(1:s1,1:s2)=uint8(0);
    imout_min(1:s1,1:s2)=uint8(0);
    imout_gamma_half(1:s1,1:s2)=uint8(0);
    imout_gamma_two(1:s1,1:s2)=uint8(0);
    mask_size=3;
    c_gamma_half=double(255)/(255.0)^0.8;
    c_gamma_two=(255.0)^(-0.2);
    for i=1:s1
        for j=1:s2
            if(i<=floor(mask_size/2)||i>=s1-floor(mask_size/2)||j<=floor(mask_size/2)||j>=s2-
floor(mask_size/2))
               imout_median(i,j)=image(i,j);
               imout_max(i,j)=image(i,j);
               imout_min(i,j)=image(i,j);
            else
               submatrix=image(i-floor(mask_size/2):i+ floor(mask_size/2),j-
floor(mask_size/2):j+ floor(mask_size/2));
               imout_median(i,j)=median(submatrix(:));
               imout_max(i,j)=max(submatrix(:));
               imout_min(i,j)=min(submatrix(:));
            end
            imout_gamma_half(i,j)=uint8(c_gamma_half*(double(image(i,j))^0.8));
            imout_gamma_two(i,j)=uint8(c_gamma_two*(double(image(i,j))^1.2));
        end
    end
    f2=figure;
    subplot(2,2,1);imagesc(image);title('orignal image');
    subplot(2,2,2);imagesc(imout_median);title('median filtered image');
    subplot(2,2,3);imagesc(imout_max);title('max filtered image');
    subplot(2,2,4);imagesc(imout_min);title('min filtered image');

    f3=figure;
    subplot(1,3,1);imagesc(image);title('orignal image');
    subplot(1,3,2);imagesc(imout_gamma_half);title('gamma filtered = 0.5');
    subplot(1,3,3);imagesc(imout_gamma_two);title('gamma filtered = 2.0');

end
   % seeing the output images , it can be inferred that Gaussian filter,
   % and median filter are able to remove the noise and are closer to the
   % orignal image.

   % Max and Min filters were able to remove the noise but the resultant
   % image is not close to the orignal image.
```

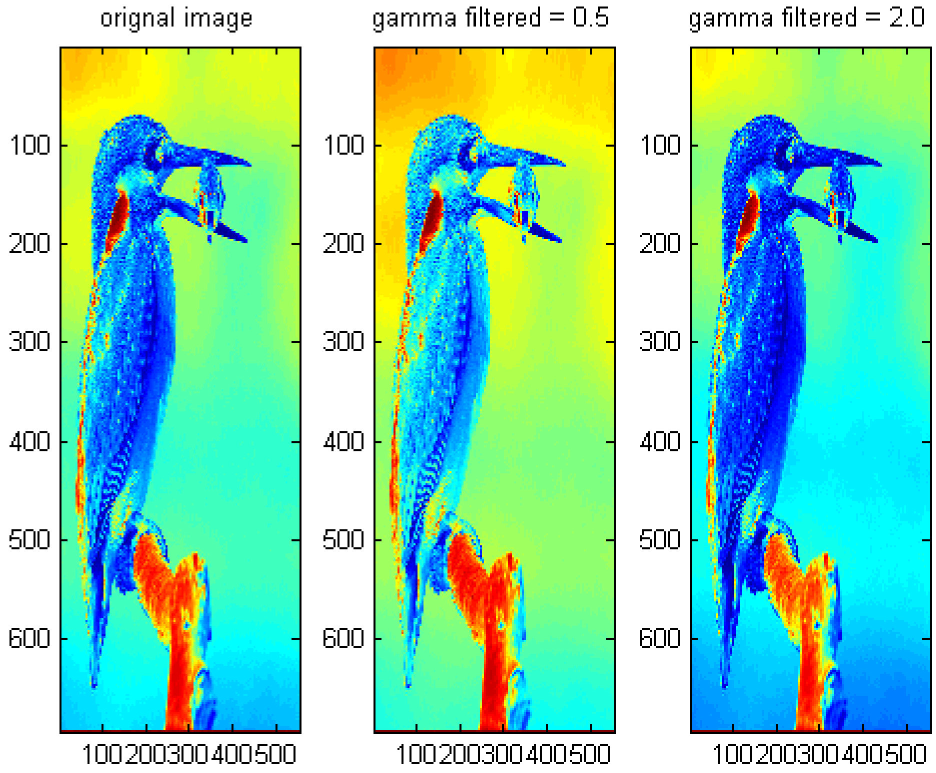
```
    % Edge enhanced, unsharp images added had more noise compared to the
    % orignal image

     %Gamma filtered images for gamma>1 and Gamma<1 were also not able to
     %remove the noise
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```


orignal image
noisy image
100
200
300
400
500
600
100
200
300
400
500

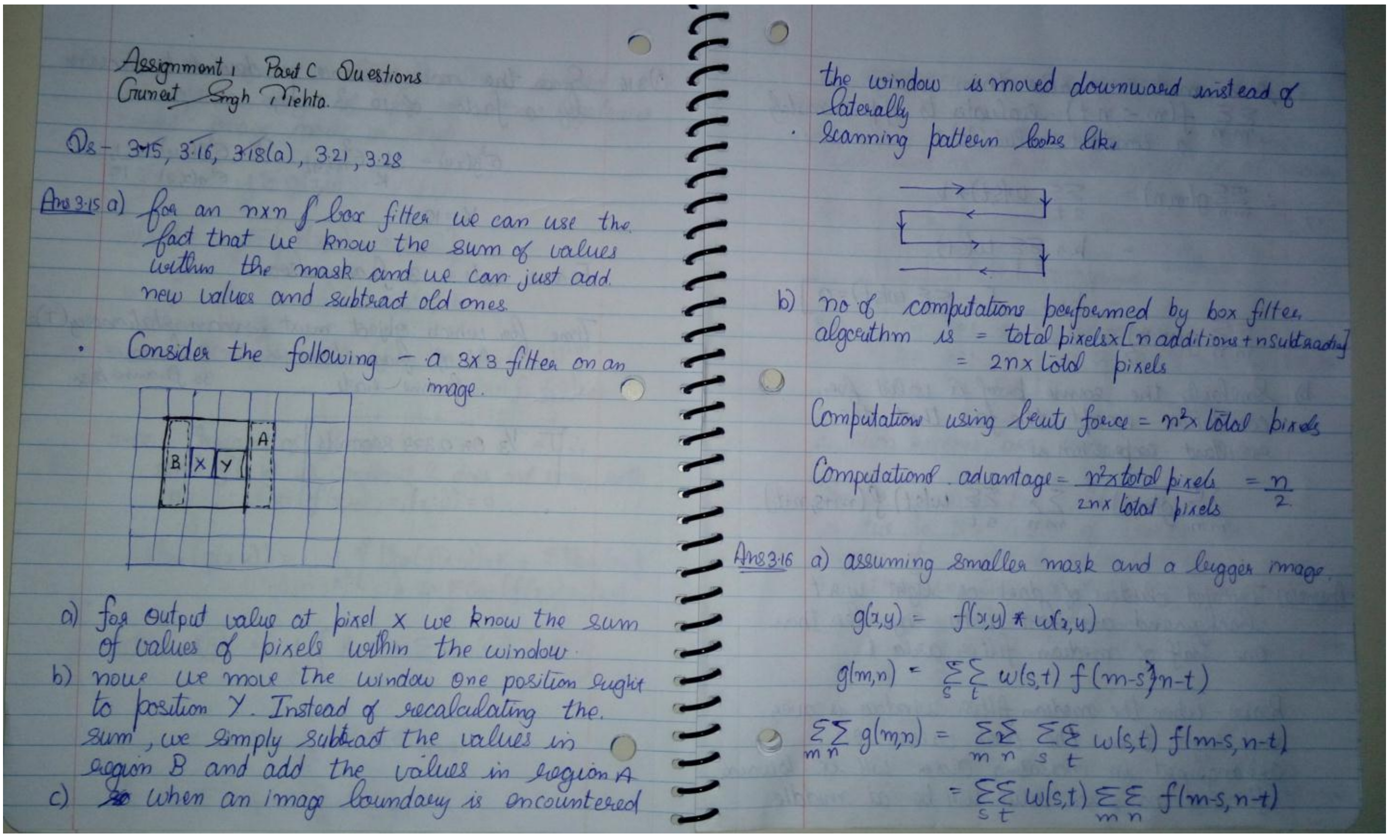

orignal image
gaussian smooth image
edge enhanced image - prewitt
unsharp image
200
400
600
100
200
300
400
500





Published with MATLAB® R2013a

Part C



Assignment, Part C Questions
Gurneet Singh Mehta.

Qs - 3.15, 3.16, 3.18(a), 3.21, 3.28

Ans 3.15 a) for an nxn ~~f~~ box filter we can use the fact that we know the sum of values within the mask and we can just add new values and subtract old ones.

- Consider the following - a 3x3 filter on an image.

a) for output value at pixel x we know the sum of values of pixels within the window.
b) now we move the window one position right to position Y. Instead of recalculating the sum, we simply subtract the values in region B and add the values in region A
c) ~~So~~ when an image boundary is encountered the window is moved downward instead of laterally

- Scanning pattern looks like

b) no of computations performed by box filter algorithm is = total pixels x [n additions + n subtraction]
= 2n x total pixels

Computation using brute force = n^2x total pixels

Computational advantage = $\frac{n^2 \text{x total pixels}}{2n \text{x total pixels}} = \frac{n}{2}$

Ans 3.16 a) assuming smaller mask and a bigger image.

$$g(x,y) = f(x,y) * w(x,y)$$

$$g(m,n) = \sum_s \sum_t w(s,t)\, f(m-s, n-t)$$

$$\sum_m \sum_n g(m,n) = \sum_m \sum_n \sum_s \sum_t w(s,t)\, f(m-s, n-t)$$

$$= \sum_s \sum_t w(s,t) \sum_m \sum_n f(m-s, n-t)$$

Ans 3.28:- unsharp masking = $f(x,y) + k\left[f(x,y) - \bar{f}(x,y)\right]$

unsharp mask = $\begin{bmatrix} -k/9 & -k/9 & -k/9 \\ -k/9 & 8k/9+1 & -k/9 \\ -k/9 & -k/9 & -k/9 \end{bmatrix} = \textcircled{A}$

Image - Laplacian = $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$\Rightarrow \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \textcircled{B}$

for $k = 9$ $\textcircled{A} = \textcircled{B}$